27/8/78
24 cy to NTIS

3151

SAND77-0552
Unlimited Release

An Algorithm for Linear Least Squares Problems with Equality and Nonnegativity Constraints

Karen H. Haskell, Richard J. Hanson

Prepared by Sandia Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550 for the United States Department of Energy under Contract AT(29–1)-789

Printed June 1978

Sandia Laboratories

SAND77-0552
Unlimited Release
Printed June 1978

An Algorithm for Linear Least Squares Problems With Equality and Nonnegativity Constraints

Karen H. Haskell
Applied Mathematics Division 2613

Richard J. Hanson
Numerical Mathematics Division 5642
Albuquerque, New Mexico 87185

Printed in the United States of America

Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161
Price: Printed Copy $4.00; Microfiche $3.00

[illegible]

Table of Contents

A B S T R A C T

We present a new algorithm for solving a linear least squares problem with linear constraints. These can be equality constraint equations and nonnegativity constraints on selected variables. This problem, while appearing to be quite special, is the core problem arising in the solution of the general linearly constrained linear least squares problem. The reduction process of the general problem to the core problem can be done in many ways. We discuss three such techniques.

The method employed for solving the core problem is based on combining the equality constraints with differentially weighted least squares equations to form an augmented least squares system. This weighted least squares system is solved with nonnegativity constraints on selected variables.

Seven small examples, including a constrained least squares curve fitting example, are presented. A reference to user instructions for subprograms to compute solutions of constrained least squares problems is included.

1. INTRODUCTION

The primary purpose of this paper is to present a new and numerically stable algorithm for solving the following linearly constrained linear least squares problem.

(1) Problem NNLSE

$$\begin{aligned} E\underset{\sim}{x} &= \underset{\sim}{f} \quad \text{(equations to be exactly satisfied)} \\ A\underset{\sim}{x} &\cong \underset{\sim}{b} \quad \text{(equations to be approximately satisfied, least squares sense)} \\ x_i &\geq 0 \ , \ i = \ell + 1, \ldots, n \ , \ 0 \leq \ell \leq n \end{aligned}$$

The real matrices E and A are of respective dimensions m_E by n and m_A by n. Variables $x_1, \ldots, x_\ell$ are free to take on either sign.

In case the equations $E\underset{\sim}{x} = \underset{\sim}{f}$ are inconsistent, we take $\underset{\sim}{x} = \underset{\sim}{x}^+ + \underset{\sim}{y}$, where $\underset{\sim}{x}^+$ is the minimal length solution of the equations and $\underset{\sim}{y}$ is some solution of $E\underset{\sim}{y} = \underset{\sim}{0}$. The remaining freedom in $\underset{\sim}{y}$ is chosen to minimize the residual vector length $\underset{\sim}{b} - A\underset{\sim}{x}$ subject to the nonnegativity constraints $x_i \geq 0$, $i > \ell$.

The problem that is seen frequently in practical computations is often stated

(2) Problem LSEI

$$\begin{aligned} E\underset{\sim}{x} &= \underset{\sim}{f} \\ A\underset{\sim}{x} &\cong \underset{\sim}{b} \\ G\underset{\sim}{x} &\geq \underset{\sim}{h} \quad \text{(inequality constraints that the solution must satisfy)} \end{aligned}$$

where G is an m_G by n real matrix.

While Problem LSEI appears to be a more general problem than Problem NNLSE, it really is not. There are a number of ways to transform Problem LSEI into one of the form of Problem

NNLSE, Eq. (1). Thus, once the computational capability for solving the system of Eq. (1) is obtained, the system of Eq. (2) can be solved, or it can be shown that no solution exists. FORTRAN language subprograms for solving the problems of Eqs. (1) and (2) are described in Ref. [15].

The remainder of Section 1 is devoted to the discussion of three methods for reducing Problem LSEI to the core problem, Problem NNLSE. This material is not essential to the understanding of Sections 2-4.

Sections 2 and 3 develop necessary theory and computational algorithms for solving Problem NNLSE. Section 4 discusses a rank deficient constrained curve-fitting problem and six small problems illustrating the various relations between problem discussion and rank.

It is well known that by introducing <u>slack variables</u> Eq. (2) can be converted to a system of the form of Eq. (1). To this end, an m_G-vector $\underset{\sim}{w}$ of nonnegative variables is introduced into the inequality constraints of Eq. (2) so that they become equality constraints $G\underset{\sim}{x} - \underset{\sim}{w} = \underset{\sim}{h}$. With the expanded matrices and vectors

$$\bar{E} \equiv \begin{array}{c} \begin{array}{ccc} \underbrace{}_{n} & & \underbrace{}_{m_G} \end{array} \\ \left[\begin{array}{ccc} E & : & 0 \\ G & : & -I \end{array}\right] \begin{array}{l} \} m_E \\ \} m_G \end{array} \end{array}$$

$$\bar{f} \equiv \begin{bmatrix} \underset{\sim}{f} \\ \underset{\sim}{h} \end{bmatrix}$$

$$\bar{A} \equiv \begin{array}{c} \begin{array}{ccc} \underbrace{}_{n} & & \underbrace{}_{m_G} \end{array} \\ \left[\begin{array}{ccc} A & : & 0 \end{array}\right] \} \; m_A \end{array}$$

and

(3) $$\bar{x} \equiv \begin{bmatrix} \underset{\sim}{x} \\ \underset{\sim}{w} \end{bmatrix} \begin{matrix} \}\ n \\ \}\ m_G \end{matrix}$$

we obtain the constrained least squares system

$$\begin{aligned} \bar{E}\,\underset{\sim}{\bar{x}} &\cong \underset{\sim}{\bar{f}} \\ \bar{A}\,\underset{\sim}{\bar{x}} &\cong \underset{\sim}{b} \\ \underset{\sim}{w} &\geqslant \underset{\sim}{0} \end{aligned} \tag{4}$$

The problem of Eq. (4) is of type NNLSE, with the parameters (m_E, m_A, n, ℓ) of Eq. (1) identified with the dimensions $(m_E + m_G, m_A, n + m_G, n)$ of Eq. (4).

While this transformation is conceptually straightforward, the techniques we present next are often more efficient for solving Problem LSEI of Eq. (2).

The important special case of Problem LSEI of Eq. (2) with $m_G \leqslant n$ inequalities can be more efficiently transformed directly to Problem NNLSE. This apparently new development results in a core problem with about the same number of parameters as the original problem. Perhaps the most important fact about this idea is that the transformations required can all be computed and stored in essentially the same storage as that needed for the original data.

Suppose that G is of rank $g \leqslant m_G \leqslant n$. As before, introduce the m_G slack variables

$$\underset{\sim}{0} \leqslant \underset{\sim}{w} \equiv G\underset{\sim}{x} - \underset{\sim}{h}$$

Using m_G by m_G and n by n orthogonal matrices H and K, compute an orthogonal decomposition of the m_G by n matrix G so that

$$H^T G K = \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix}$$

where T is a g by g lower triangular nonsingular matrix.

We change variables by letting

$$\underset{\sim}{x} = K\underset{\sim}{y}$$

Using the matrices for the equality constraints and least squares equations in Eq. (2), we have

(5) $$\begin{vmatrix} E \\ A \end{vmatrix} K \equiv \begin{bmatrix} \overbrace{E_1'}^{g} & : & \overbrace{E_2'}^{n-g} \\ A_1' & : & A_2' \end{bmatrix}$$

We also introduce the partitioned vector and matrix

$$\underset{\sim}{y} \equiv \begin{array}{l} \left[\begin{array}{c} \underset{\sim}{y}_1 \\ \underset{\sim}{y}_2 \end{array}\right] \begin{array}{l} \}\, g \\ \}\, n-g \end{array} \end{array}$$

(6) and

$$H \equiv \left| \overbrace{H_1}^{g} \;:\; \overbrace{H_2}^{m_G - g} \right|$$

Following the change of variables for $\underset{\sim}{x}$ and the use of the orthogonal decomposition for G, the equation $G\underset{\sim}{x} - \underset{\sim}{h} = \underset{\sim}{w}$ becomes

$$T\underset{\sim}{y}_1 = H_1^T \left| \underset{\sim}{w} + \underset{\sim}{h} \right|$$

(7) and

$$0 = H_2^T \left| \underset{\sim}{w} + \underset{\sim}{h} \right|$$

Making these substitutions in the equality constraint and least squares equations of Eq. (2) leads to the constrained least squares problem for the $n + m_G - g$ vector

$$\hat{\underset{\sim}{x}} \equiv (\underset{\sim}{y}_2, \underset{\sim}{w}^T)^T$$

(8)
$$\begin{cases} \hat{E}_1\underset{\sim}{y}_2 + \hat{E}_2\underset{\sim}{w} = \hat{\underset{\sim}{f}} \\ \hat{A}_1\underset{\sim}{y}_2 + \hat{A}_2\underset{\sim}{w} = \hat{\underset{\sim}{b}} \\ \underset{\sim}{w} \geq \underset{\sim}{0} \end{cases}$$

Some easy algebra and Eqs. (5)-(7) show that the matrices and vectors of Eq. (8) satisfy

$$\hat{E}_1 \equiv \begin{array}{c} \overbrace{}^{n-g} & \\ \left[\begin{array}{c} E_2 \\ 0 \end{array}\right] & \begin{array}{l} \} \; m_E \\ \} \; m_G - g \end{array} \end{array}$$

$$\hat{E}_2 \equiv \begin{array}{c} \overbrace{\phantom{E_1 T^{-1} H_1^T}}^{m_G} & \\ \left[\begin{array}{c} E_1 T^{-1} H_1^T \\ H_2^T \end{array}\right] & \begin{array}{l} \} \; m_E \\ \} \; m_G - g \end{array} \end{array}$$

$$\underset{\sim}{\hat{f}} \equiv \begin{bmatrix} \underset{\sim}{f} \\ \underset{\sim}{0} \end{bmatrix} - \hat{E}_2 \underset{\sim}{h}$$

and

$$\hat{A}_1 = A_2'$$

$$\hat{A}_2 = A_1' T^{-1} H_1^T$$

$$\underset{\sim}{\hat{b}} = \underset{\sim}{b} - \hat{A}_2 \underset{\sim}{h}$$

The relative ordering of the vectors $\underset{\sim}{y}_2$ and $\underset{\sim}{w}$ in the vector $\underset{\sim}{\hat{x}}$ was chosen so that the problem statement of Eq. (8) would be of type NNLSE of Eq. (1). This is Problem NNLSE with the parameters (m_E, m_A, n, ℓ) of Eq. (1) identified with the dimensions $(m_E + m_G - g, m_A, n + m_G - g, n - g)$ of Eq. (8).

It is easy to see that an L-shaped additional strip of storage beyond the storage for E and A of Problem LSEI is required for the problem of Eq. (8). The number of additional storage locations required is $(m_E + m_A + m_G - g + n + 1)(m_G - g)$. Since we anticipate that $g \doteq m_G$, very little additional storage (none if $g = m_G$) is required.

A third technique for transforming Eq. (2) into a problem of the type NNLSE is described by Cline in [4]. Briefly, this method reduces Problem LSEI to

(9) Problem LPDP

$$\text{minimize} \sum_{i=1}^{k} x_i^2 \quad , \quad k \leq n$$

$$\text{subject to } G\underset{\sim}{x} \geq \underset{\sim}{h}$$

We have used the same notation for the inequality constraints. The integer k is determined during the reduction process.

Problem LPDP is solved using a formulation which involves the solution of two problems of type NNLSE of Eq. (1). Further details are given in [4] .

Chapters 20 and 23 of [3] present Algorithm LDP for solving the special case of Eq. (9) where k = n. In Chapter 23 an error occurred in the development of the reduction of Problem LSEI of Eq. (2) to Problem LDP. This development is valid only in the special case where the matrix $\begin{vmatrix} E \\ A \end{vmatrix}$ is of full column rank k = n. Cline in [4] has given a mathematical development for solving Problem LPDP of Eq. (9) for all values $k \leq n$. In principle this allows one to solve Problem LSEI for any value of the rank of $\begin{bmatrix} E \\ A \end{bmatrix}$.

Numerically stable methods for solving certain cases of the constrained system of Eq. (2) have also been proposed by Stoer [1], Golub and Saunders [2], and Eldén [10].

2. NONNEGATIVITY CONSTRAINTS AND EQUALITY CONSTRAINTS BY DIFFERENTIAL WEIGHTING

In this section we discuss a mathematical method for solving Problem NNLSE of Eq. (1) using differential weighting of the equality constraints and least squares equations. Lemma 4 establishes the convergence of the mathematical algorithm to a solution of Problem NNLSE, provided one exists.

Problem NNLSE of Eq. (1) has nonnegativity constraints on variables $\ell + 1, \ldots, n$. By eliminating the variables $1, \ldots, \ell$ it can be reduced to a form where all variables are constrained to be nonnegative. Thus, to make this section easier for the reader to understand, we assume that $\ell = 0$ in Eq. (1). However, in the Algorithm WNNLS described in Section 3, we solve Problem NNLSE for any ℓ, $0 \leq \ell \leq n$.

Our approach to solving Problem NNLSE of Eq. (1) is based on solving the differentially weighted least squares problem of Eq. (10). We will show that a solution to Problem WNNLS of Eq. (10) has a limit as $\varepsilon \to 0+$. Furthermore, if the set $\{\underset{\sim}{x} : E\underset{\sim}{x} = \underset{\sim}{f}, \underset{\sim}{x} \geq 0\}$ is nonempty, this limit vector is a solution of Problem NNLSE of Eq. (1). This development and its proof are summarized in Lemma 4.

(10) <u>Problem WNNLS</u>

$$A_\varepsilon \underset{\sim}{x} \cong \underset{\sim}{b}_\varepsilon \quad \text{with} \quad \begin{bmatrix} A_\varepsilon & : & \underset{\sim}{b}_\varepsilon \end{bmatrix} = \begin{bmatrix} E & : & \underset{\sim}{f} \\ \varepsilon A & : & \varepsilon \underset{\sim}{b} \end{bmatrix} \begin{matrix} \}\, m_E \\ \}\, m_A \end{matrix}$$
$$\underset{\sim}{x} \geq \underset{\sim}{0}$$

where ε is a small positive real parameter.

Another way of stating this is that the function to be minimized in Eq. (10) is

$$\|E\underset{\sim}{x} - \underset{\sim}{f}\|^2 + \varepsilon^2\|A\underset{\sim}{x} - \underset{\sim}{b}\|^2$$
$$\text{subject to } \underset{\sim}{x} \geq \underset{\sim}{0}$$

Here the vector norms used are euclidean lengths. Our approach, therefore, amounts to a penalty function minimization method that is implemented in a numerically stable way.

In principle, Problem WNNLS, for each $\varepsilon > 0$, can be solved using Algorithm NNLS of Lawson and Hanson, Chapter 23, [3] . In practical computations there are other considerations that must be made when solving Problem WNNLS. These consist of using weighted euclidean lengths in tests for linear independence of column vectors of the problem matrix generated during the orthogonal decomposition process. We must, in fact, use the values of m_E, m_A and ℓ to compute this weighted euclidean norm in the algorithm.

Thus, the <u>mathematical</u> solution of Problem WNNLS of Eq. (10) is obtained using Algorithm NNLS of [3] . Computational distinctions are further discussed in Section 3.

In Lemmas 1-4 we show that the solution of Problem WNNLS is well-defined in its dependence on the small parameter $\varepsilon > 0$.

The discussion in these lemmas refers only to the mathematical description of the algorithms.

In Lemma 1, the statement and proof refer to notation and discussion of Algorithm NNLS, Chapter 23, loc. cit.

Lemma 1

The solution of $A_\varepsilon x \cong b_\varepsilon$, $x \geq 0$ obtained using Algorithm NNLS is a vector $x = \hat{x}_\varepsilon \geq 0$ for each ε. The set $Z_\varepsilon = \{i : x_i(\varepsilon) = 0\}$ is fixed for $0 < \varepsilon \leq \varepsilon_0$, ε_0 sufficiently small.

Proof: Each of the choices of indices in steps 4 and 7-11 are determined by the sign, min or max of functions that are meromorphic in the parameter ε near $\varepsilon = 0$. Once we agree always to take the smallest indexes for the choice of min or max, it follows by the finite convergence of Algorithm NNLS and the meromorphicity of the choice functions, that for some $\varepsilon_0 > 0$, all the indices are fixed for $0 < \varepsilon \leq \varepsilon_0$.

Lemma 2

The constrained least squares problem $A_\varepsilon u \cong b_\varepsilon$, $u \geq 0$, has a solution u_ε. This solution has a limit u_0 as $\varepsilon \to 0$.

Proof: Those components of u_ε which are positive are a fixed set for $0 < \varepsilon < \varepsilon_0$, if ε_0 is sufficiently small. This follows from Lemma 1. Eliminating those components of u which are identically zero for all ε and reverting to the original notation, we must show that the solution of a least squares problem

$$A_\varepsilon \underset{\sim}{u} \cong \underset{\sim}{b}_\varepsilon$$

without constraints has a limit as $\varepsilon \to 0$.

The matrix A_ε has full column rank, by virtue of its choice in Algorithm NNLS. Now using the methods of Chapter 22, loc. cit., e.g. Eq. (22.34) and the related development, we see that $\underset{\sim}{u}_\varepsilon \to \underset{\sim}{u}_0$ as $\varepsilon \to 0$.

Lemma 3

Suppose we consider the two constrained least squares problems

$E\underset{\sim}{u} = \underset{\sim}{f}$		$A_\varepsilon \underset{\sim}{u} \cong \underset{\sim}{b}_\varepsilon$	
$A\underset{\sim}{u} \cong \underset{\sim}{b}$	Problem NNLSE		Problem WNNLS
$\underset{\sim}{u} \geq \underset{\sim}{0}$		$\underset{\sim}{u} \geq \underset{\sim}{0}$	

A solution: $\underset{\sim}{u}' \geq \underset{\sim}{0}$ A solution: $\underset{\sim}{u}_\varepsilon \geq \underset{\sim}{0}$; (obtained with Algorithm NNLS)

Suppose that $E\underset{\sim}{u} = \underset{\sim}{f}$ is consistent for a feasible $\underset{\sim}{u}$; that is, there exists $\hat{\underset{\sim}{u}} \geq 0$ such that $E\hat{\underset{\sim}{u}} = \underset{\sim}{f}$.
Then

(a) $E\underset{\sim}{u}' = \underset{\sim}{f}$

and for any $\varepsilon > 0$

(b) $\|E\underset{\sim}{u}_\varepsilon - \underset{\sim}{f}\|^2 + \varepsilon^2\|A\underset{\sim}{u}_\varepsilon - \underset{\sim}{b}\|^2 \leq \|E\underset{\sim}{u} - \underset{\sim}{f}\|^2 + \varepsilon^2\|A\underset{\sim}{u} - \underset{\sim}{b}\|^2$

for any $\underset{\sim}{u} \geq \underset{\sim}{0}$

(c) $\|E\underset{\sim}{u}_\varepsilon - \underset{\sim}{f}\|^2 \leq \varepsilon^2\|A\underset{\sim}{u}' - \underset{\sim}{b}\|^2$

(d) $\|A\underset{\sim}{u}_\varepsilon - \underset{\sim}{b}\|^2 \leq \|A\underset{\sim}{u}' - \underset{\sim}{b}\|^2$

Proof:

Part (a) Let $\underset{\sim}{u}' = E^+\underset{\sim}{f} + \underset{\sim}{\chi}$, where $E\underset{\sim}{\chi} = \underset{\sim}{0}$ and $E^+\underset{\sim}{f} + \underset{\sim}{\chi} \geq \underset{\sim}{0}$.
Here E^+ is the Moore-Penrose pseudoinverse of E, [3].
Then $0 \leq \|E\underset{\sim}{u}' - \underset{\sim}{f}\| \leq \|E\hat{\underset{\sim}{u}} - \underset{\sim}{f}\| = 0$, implies $E\underset{\sim}{u}' = \underset{\sim}{f}$.

Part (b) This follows from the fact that $\underset{\sim}{u}_\varepsilon$ is the least squares solution of $A_\varepsilon \underset{\sim}{u} \cong \underset{\sim}{b}_\varepsilon$ in the region $\underset{\sim}{u} \geq \underset{\sim}{0}$.

Part (c) Set $\underset{\sim}{u} = \underset{\sim}{u}'$ in the right member of the inequality of (b) and use (a).

Part (d) Similar to proof of (c).

Lemma 4

Assume the same hypothesis as Lemma 3, and the same notation.

Then

(a) $\lim_{\varepsilon \to 0} \underset{\sim}{u}_\varepsilon = \underset{\sim}{u}_0 \geq \underset{\sim}{0}$ exists

(b) $E\underset{\sim}{u}_0 = \underset{\sim}{f}$

(c) $\| A\underset{\sim}{u}_0 - \underset{\sim}{b} \| = \| A\underset{\sim}{u}' - \underset{\sim}{b} \|$

Proof:
Part (a) is Lemma 2.

Part (b) follows from using the bound for $\| E\underset{\sim}{u} - \underset{\sim}{f} \|^2$ of Lemma 3, (c) as $\varepsilon \to 0$.

Part (c) follows by observing that the bound of Lemma 3, (d) implies that $\| A\underset{\sim}{u}_0 - \underset{\sim}{b} \| \leqslant \| A\underset{\sim}{u}' - \underset{\sim}{b} \|$. But since $\underset{\sim}{u}' \geqslant \underset{\sim}{0}$ is a least squares solution of $A\underset{\sim}{u} \cong \underset{\sim}{b}$ for all $\underset{\sim}{u} \geqslant \underset{\sim}{0}$ satisfying $E\underset{\sim}{u} = \underset{\sim}{f}$, we must have $\| A\underset{\sim}{u}' - \underset{\sim}{b} \| \leqslant \| A\underset{\sim}{u}_0 - \underset{\sim}{b} \|$. These last inequalities together show that $\| A\underset{\sim}{u}_0 - \underset{\sim}{b} \| = \| A\underset{\sim}{u}' - \underset{\sim}{b} \|$.

This completes the proofs of the lemmas.

Lemma 4 provides the mathematical basis of an algorithm for solving Problem WNNLS, and thus solving Problem NNLSE. In computing a solution of Problem NNLSE <u>numerically</u>, the system $A_\varepsilon \underset{\sim}{x} \cong \underset{\sim}{b}$, $(\underset{\sim}{x} \geqslant \underset{\sim}{0})$ of Eq. (10) is solved just <u>once</u> using a sufficiently small positive ε. This value of ε is chosen by the algorithm to compute the solution to full working accuracy.

3. THE WNNLS ALGORITHM

In this section we will discuss the overall description of the numerical algorithm used to solve Problem WNNLS of Eq. (10) in Section 2. This discussion will include the choice of the small parameter ε and the norms used in rank determination. Because of the complexity of the algorithm, only the essential points are given here.

To ease notational problems in this section, we restate the problem of Eq. (10). The augmented matrix $[A_\varepsilon : \underset{\sim}{b}_\varepsilon]$ of Eq. (10) is written as a new augmented matrix $[DA : D\underset{\sim}{b}]$. Thus, we define the vector and matrix

$$\underset{\sim}{x} = \begin{bmatrix} \underset{\sim}{y} \\ \underset{\sim}{w} \end{bmatrix} \begin{matrix} \}\ \ell, & \underset{\sim}{y} \text{ unconstrained} \\ \}\ n-\ell, & \underset{\sim}{w} \geqslant \underset{\sim}{0} \end{matrix}$$

$$D = \text{diag}(\overbrace{1,\ldots,1}^{m_E},\ \overbrace{\varepsilon,\ldots,\varepsilon}^{m_A})$$

and Problem WNNLS becomes the constrained least squares problem

$$\begin{aligned} &DA\underset{\sim}{x} \cong D\underset{\sim}{b} \qquad A \text{ is } m \text{ by } n,\ m \equiv m_E + m_A \\ &\underset{\sim}{w} \geqslant \underset{\sim}{0}\,. \end{aligned} \tag{11}$$

We partition the matrix of Eq. (11) as

$$DA \equiv \begin{bmatrix} \overbrace{DA_1}^{\ell} & : & \overbrace{DA_2}^{n-\ell} \end{bmatrix}$$

corresponding to the dimensions of the vectors $\underset{\sim}{y}$ and $\underset{\sim}{w}$.

In the following discussion Algorithm WNNLS will be presented in a top-down, structured format. In addition to giving a clear description of the algorithmic steps at the outer level of detail, the structured algorithm very closely parallels the actual source code that was used to implement the algorithm. The notation used below closely follows the description of a similar algorithm by C. L. Lawson in [6].

Algorithm WNNLS

```
INITIALIZE-VARIABLES-AND-INITIALLY-TRIANGULARIZE
Until (done)
    COMPUTE-SEARCH-DIRECTION-AND-FEASIBLE-POINT
    If (hit-a-constraint) then
        ADD-CONSTRAINTS
    Else
        PERFORM-KUHN-TUCKER-TEST-AND-DROP-A-CONSTRAINT
COMPUTE-FINAL-SOLUTION
```

The names in capital letters denote procedure calls. The italicized parameters are logical variables manipulated within the procedures.

We will discuss each of the procedures used by Algorithm WNNLS in Sections 3.1 - 3.3.

3.1. Initialization, Initial Triangularization and Final Solution Procedures

The initial phase of the algorithm performs three main functions. First, the weight ε is chosen. Second, the matrix DA_1 is triangularized. Third, dependent equality constraints are eliminated.

We have found any choice of the weight ε in the range $(L/\eta)^{\frac{1}{2}} < \varepsilon < \varepsilon_0 = \eta^{\frac{1}{2}}$ to be satisfactory. Here L is the smallest positive machine number and η is the relative machine precision. The upper bound of $\eta^{\frac{1}{2}}$ is motivated by Eq. (22.37), and the preceding development, of [3]. A weight of at most $\eta^{\frac{1}{2}}$ is needed to achieve full working accuracy in an equality constrained least squares problem. The lower bound of $(L/\eta)^{\frac{1}{2}}$ is to prevent damaging underflows.

Using an orthogonal decomposition we initially triangularize the m by ℓ submatrix DA_1. Premultiplying transformations are applied to the remaining columns and right-side vector. We triangularize DA_1 using essentially the Golub-Businger algorithm with extensions for rank deficiency, [3], Chapter 14. Givens orthogonal transformations are used to compute the forward triangularization. As in the Golub-Businger algorithm, column interchanges are performed to maximize the size of the resulting diagonal term at each major step. When DA_1 is rank deficient, Householder orthogonal transformations are applied from the right to obtain an upper triangular, square, nonsingular matrix.

Thus, we compute Q_1 and H_1 satisfying

$$Q_1(DA_1)\,H_1^T = \left[\begin{array}{c|c} \overbrace{\diagdown}^{k} & \overbrace{0}^{\ell-k} \\ \hdashline 0 & \end{array}\right]\begin{array}{l}\}\,m_E \\ \}\,m-m_E\end{array} = \begin{bmatrix} R_1 & 0 \\ 0 & 0 \end{bmatrix} \tag{12}$$

Here k is the rank of DA_1. The horizontal dotted line in the middle term of Eq. (12) corresponds to the value of m_E. Any relation can hold between the integer values of ℓ, k, m_E and m satisfying $0 \leqslant k \leqslant \ell$ and $0 \leqslant m_E \leqslant m$.

When $k < m_E$, we again use the Golub-Businger algorithm in the submatrix consisting of rows $k + 1, \ldots, m_E$ and columns $\ell + 1, \ldots, n$. If any dependent equality constraint equations are found, they are removed from the problem and m_E is reduced corresponding to the rank deficiency of this submatrix. As a result of this step, the equality constraints are linearly independent when we enter the nonnegativity constraint part of the algorithm discussed in Section 3.2.

During the elimination process we want the weighting of rows 1, ..., m_E and rows $m_E + 1$, ..., m to remain disparate at every step. We eliminate the subdiagonal terms in the nonzero column j as follows:

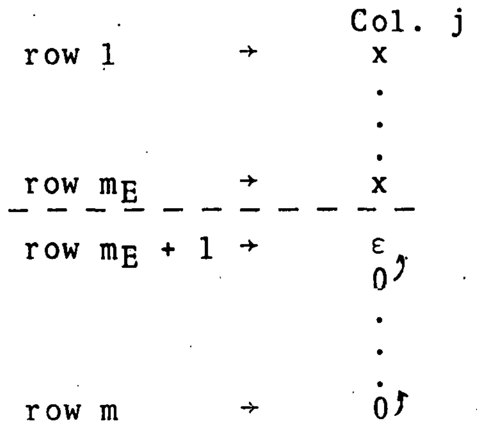

Figure 1. Elimination Order in Column j

The elimination proceeds with row i being "chased" to row i - 1, i = m, ..., $m_E + 2$, using Givens transformations. Next, the element in the $(m_E + 1)^{st}$ row, shown schematically in Fig. 1, is chased into the row of largest index, p, which is nonzero, where $j \leqslant p \leqslant m_E$. Then the elimination of the subdiagonal terms in column j is completed by chasing row i to row i - 1, i = p, ..., j + 1. This pivoting-like step is necessary to avoid mixing weighted and nonweighted rows.

In the exceptional case where components j, ..., m_E are zero, we express the fact that we have solved for y_j in terms of y_{j+1}, ..., y_ℓ and $\underset{\sim}{w}$ as a new and additional equality constraint. This is accomplished by an interchange of rows $m_E + 1$ and j and an increase in the value of m_E by 1.

In this initial phase of the algorithm the rank of the matrix DA_1 is determined by declaring column j, $j \leq m_E$, to be dependent if the euclidean length of components j, ..., m is insignificant compared to the euclidean length of components 1, ..., j - 1. For $j > m_E$ we use the same criterion but the weighted euclidean length $\|D(.)\|$. We have used a rather large tolerance of 10^{-4} for this test in the FORTRAN subprogram WNNLS( ) that solves Problem WNNLS of Eq. (11).

Once the vector $\hat{\underset{\sim}{w}} > \underset{\sim}{0}$ has been computed using the procedures discussed in Section 3.2, the final solution is computed. We want to solve the least squares problem

$$DA_1\underset{\sim}{y} = \underset{\sim}{b} - DA_2\hat{\underset{\sim}{w}}$$

for $\underset{\sim}{y} = \hat{\underset{\sim}{y}}$.

Using the orthogonal decomposition of Eq. (12) we define $\underset{\sim}{g} = Q_1(\underset{\sim}{b} - DA_2\hat{\underset{\sim}{w}})$. Then we compute

$$(13) \qquad \hat{\underset{\sim}{y}} = H_1\begin{bmatrix} R_1 & 0 \\ 0 & 0 \end{bmatrix}^{+}\underset{\sim}{g} = H_1\begin{bmatrix} R_1^{-1} & 0 \\ 0 & 0 \end{bmatrix}\underset{\sim}{g}$$

The final solution of Eq. (11) is given by $\hat{\underset{\sim}{x}} = [\hat{\underset{\sim}{y}}^T \; \hat{\underset{\sim}{w}}^T]^T$, with corresponding residual vector length $\|D^{-1}[0, \ldots, 0, g_{k+1}, \ldots, g_m]^T\|$.

This completes the discussion of the essential features of the first and last subprocedures of Algorithm WNNLS.

3.2. Distinctions Between Algorithm WNNLS and Algorithm NNLS

In this section we consider the remaining subproblem of the Until loop in Algorithm WNNLS where all the variables are constrained to be nonnegative. This amounts to solving Problem WNNLS of Eq. (11) with $\ell = 0$. Mathematically, this is Problem NNLS of [3], Chapter 23. Because of the disparate weighting matrix D in Problem WNNLS, a number of additional details are required to successfully implement Algorithm WNNLS. As in Section 3.1, most of these considerations are based on maintaining the disparate weighting of the data matrix at each step of the computation.

The algorithm begins with all constraints active, $\underset{\sim}{w} = \underset{\sim}{0}$. At each iteration a feasible point $\underset{\sim}{w} \geq \underset{\sim}{0}$ is generated with a reduction in the length of the residual vector $\underset{\sim}{r} = D(\underset{\sim}{b} - A\underset{\sim}{w})$. When no reduction is possible the flag done is set true.

During the course of the algorithm as constraints are dropped and added, it is possible for some of the rows $1, \ldots, m_E$ to become disparately scaled. Suppose there are j positive components in the solution, corresponding to the first j columns. Prior to dropping a new nonnegativity constraint, rows $j + 1, \ldots, m_E$ are tested for disparate scaling. If such a row is found, it is reclassified as a least squares equation and the value of m_E is reduced.

Each variable which is to enter the solution set must be positive. As each such variable enters, the orthogonal decomposition is updated. This amounts to triangularizing a matrix which is already upper triangular except for the last column, which may be entirely nonzero. The updated triangular form is obtained using exactly the same chasing procedure described for the unconstrained problem in Secton 3.1, Fig. 1.

Recall that this may involve solving for some variables (using least squares equations) in terms of the remaining variables and an increase in the value of m_E.

The independence of the additional column vector, corresponding to the constraint just dropped, is decided by solving for w_j and removing this constraint only when

$$0 < w_j < \| \underset{\sim}{b} \| / n^{\frac{1}{2}} \text{ and } a_{jj} \neq 0$$

The idea here is to use solution components that are positive but "not too large." This test was used because it accounts for the consistency of the right-side vector with respect to the disparate row scaling of the matrix D of Eq. (11). A test for independence of this new vector based only on the condition number of the j columns of the triangularized matrix was found to be inappropriate for our algorithm. This is because of the disparate row scaling. Nevertheless, the linear system has a well-determined solution because the scaling is consistently applied to the right-side vector.

At each major step of the algorithm we test to determine if any solution components are nonpositive. Thus, we see if a constraint has been hit or violated. If this occurs, an interpolation to a feasible point is made. If this interpolation step is necessary, at least one column must be removed from the triangularization. The orthogonal decomposition of the nonactive constraints must be retriangularized. In the implementation of Algorithm NNLS of [3] this redecomposition was accomplished simply by chasing the subdiagonal elements to the diagonal of the Hessenberg matrix resulting from removing the column. The use of Givens transformations is particularly efficient because of the special structure of the upper Hessenberg matrix. In the implementation of Algorithm WNNLS this retriangularization is necessarily more complicated than in Algorithm NNLS. Again, this is due to the disparate row scaling shown in Eq. (11). The retriangularization of the Hessenberg matrix proceeds as in Algorithm NNLS except that we pivot to avoid mixing of disparately weighted and nonweighted rows.

Mathematically, when a constraint is hit or violated, the interpolation performed will always cause a move to a nonnegative point. Numerically, it is possible for one or more solution components to remain nonpositive. Thus, after a constraint has been added, other constraints must be added until all components are positive. Each such step results in retriangularization of a Hessenberg matrix as described above.

3.3. Miscellaneous Algorithmic Details in WNNLS

The premultiplying Givens transformations used are the modified Givens transformations discussed in [8], [3], and [7]. This is a natural choice here because of the premultiplying scaling matrix λD, where D appears in Eq. (11). The modified Givens algorithm updates the square of this matrix at each elimination step of the WNNLS ( ) subprogram.

In our FORTRAN implementation of Algorithm WNNLS we multiplied both sides of Eq. (11) by the large parameter $\lambda = \varepsilon^{-1}$. This mathematically equivalent weighting scheme was chosen because we anticipate that the majority of uses will have m_E small compared to m_A. This reduces the amount of extra work that must be done for rescaling during the modified Givens transformations.

Eldén [10] has remarked that Stoer's constrained least squares algorithm [1] includes Algorithm NNLS as a special case. In a strict mathematical sense this is true, at least for problems of full column rank. (Algorithms NNLS and WNNLS do not require this full rank assumption). As Eldén implies, there are a number of additional details in the FORTRAN subprogram NNLS( ) that are included to cope with finite precision arithmetic. Without such practical considerations the code would be unreliable. The FORTRAN subprogram WNNLS ( ) implementing Algorithm WNNLS also includes such needed practical details.

The subprogram WNNLS ( ) is coded using FLECS [9], a structured FORTRAN preprocessor. In particular, the outer level of coding detail using FLECS can be stated exactly as the structured outline of Algorithm WNNLS given at the first of this section.

In addition to the use of FLECS, the use of the BLAS [8] further assists in modularizing the code.

4. ILLUSTRATIVE EXAMPLES

In this section we give examples that are intended to show that the new algorithm presented for solving constrained least squares problems has wider applications than existing algorithms. These sample problems can also be used to aid those doing future algorithmic development on constrained least squares problems.

4.1. A Rank Deficient Constrained Curve Fitting Example

Here we present an example of fitting data points (x_i, y_i), $i = 1, \ldots, 7$, by a piecewise cubic function parameterized using hermite cubics, [13]. Due to additional problem-related information, the curve is to be convex downward, nonincreasing and nonnegative. This leads to a constrained least squares problem that is underdetermined and rank deficient.

We used piecewise cubics because the convexity can be achieved by a linear constraint. We used the hermite cubic basis functions rather than de Boor's B-splines, [14], because of their simplicity. The particular choice of breakpoints shown in Table 1 makes no claim to being optimal in any sense. The rank deficiency comes from the fact that intervals 3, 4 and 5 have, in total, only one data point. Fewer intervals would avoid this particular difficulty.

i	Data Independent Variable Values, x_i	Data Dependent Variable Values, y_i	Nodes of the Hermite Cubics
1	0.2066	0.1510	0.2066
2	0.2762	0.1410	1.2050
3	0.4367	0.1250	3.1380
4	0.7407	0.1030	5.0710
5	0.5848	0.0970	8.9360
6	2.4390	0.0500	16.6670
7	16.6670	0.0140	--

Table 1. Discrete Data and Piecewise Cubic Nodes

In order to constrain the piecewise continuous second derivative to be nonnegative globally it is necessary and sufficient to constrain it to be nonnegative at the left and right limits at nodes 2, 3, 4, 5 and at the right for node 1 and at the left for node 6. A nonpositive constraint for the first derivative and a nonnegative constraint for the function at node 6 guarantee that the fitted curve has the desired shape.

This leads to a linear constrained least squares problem of the form

$$\underset{\sim}{x} = \left|f_1,\ f_1',\ldots,f_6,\ f_6',\right|^T$$

$$A\underset{\sim}{x} \cong \underset{\sim}{b}\ ,\qquad A_{7x12},\quad \underset{\sim}{b} = \left[y_1,\ldots,y_7\right]^T$$

$$G\underset{\sim}{x} \geqslant \underset{\sim}{h}\ ,\qquad G_{12x12},\quad \underset{\sim}{h} = \underset{\sim}{0}$$

where f_i is the piecewise polynomial function value and f_i' is its derivative value at node i.

We used an orthogonal decomposition of the matrix A based on the subprogram HFTI( ), [3], to reduce the problem to one of type LPDP of Eq. (9). A tolerance of $10^{-4}\,\|A\|$ was used in the rank test, and the pseudorank of A was computed as 6 by HFTI( ) using that value.

i	Constrained Solution Vector f_i	f_i'	Residual Vector Length $/\sqrt{7}$ = R.M.S. error
1	0.01514	-0.1626	
2	0.0875	-0.0248	4.76×10^{-3}
3	0.0389	-0.0247	
4	0.0140	0.0	
5	0.0140	0.0	
6	0.0140	0.0	

Table 2. Values of the Constrained Curve and its Derivative the Nodes

For purposes of comparison we plotted the piecewise hermite cubics obtained without constraints. This corresponds to the solution of minimal length to $A\tilde{x} \cong \tilde{b}$. The R.M.S. error with the unconstrained curve is 2.49×10^{-3}, about half the R.M.S. error as in the constrained case. The graph of this curve is shown in Figure 2. Using the coefficients resulting from the constrained least squares problem found in Table 2, we have the graph of Figure 3.

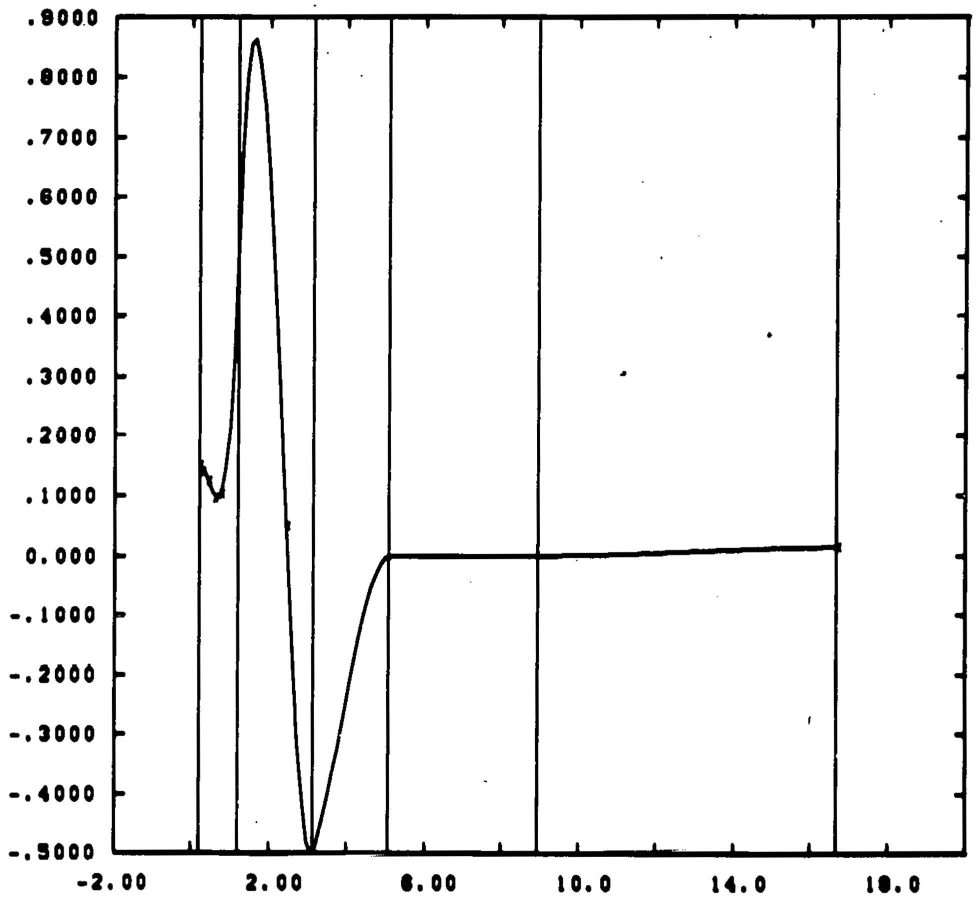

Figure 2. Hermite Piecewise Cubic Fit of Discrete Data Using Minimal Length Unconstrained Solution

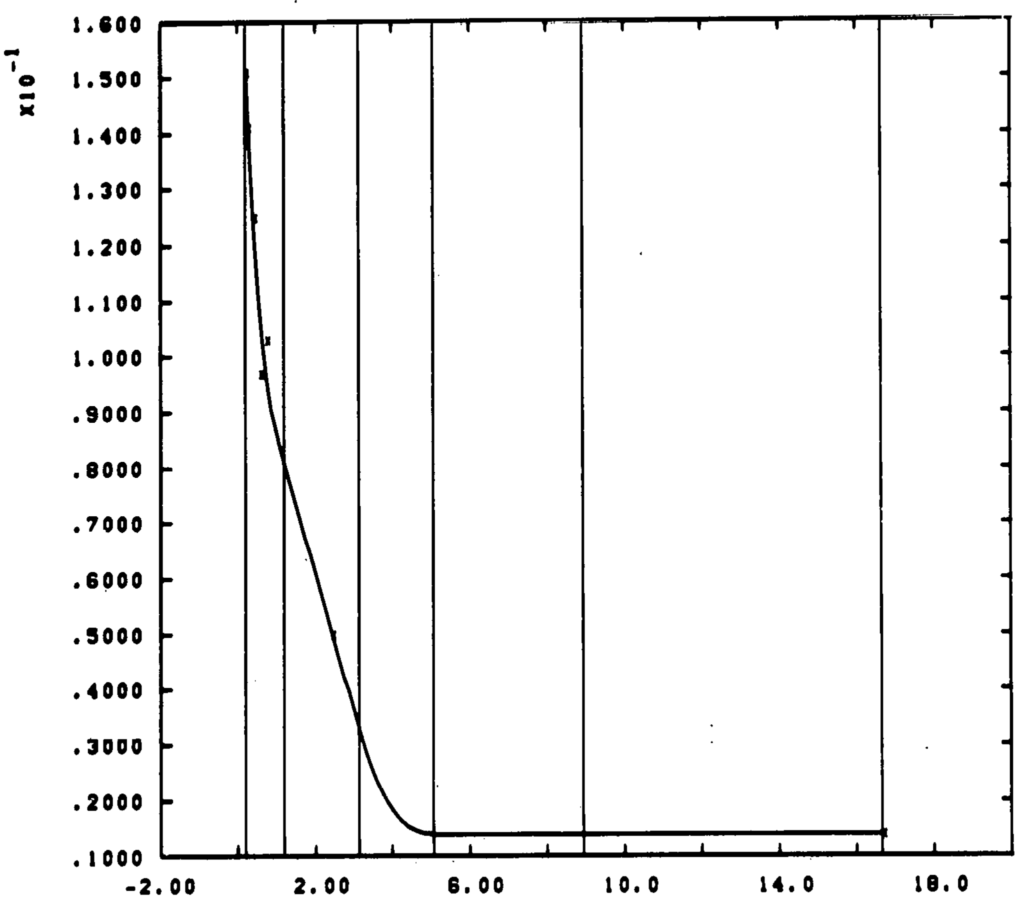

Figure 3. Hermite Piecewise Cubic Fit of Discrete Data with Constraints on the Curve

In Figures 2 and 3 the interior vertical lines mark the nodes shown in Table 1. The 7 data points are marked with an "x." Notice that the dependent variable scales are different from Figure 2 to Figure 3.

Without any additional information there is no reason to prefer the curve of Figure 3 to the curve of Figure 2. But with the requirement that the curve "have the same shape as the data" (or, more precisely, be convex downward, decreasing and nonnegative), the constrained least squares formulation is necessary and the constrained solution satisfies these requirements.

4.2 <u>Examples Formulated Using Slack Variables</u>

In [3], p. 3, there is a rough breakdown of an m by n least squares problem $\underset{\sim}{A x} \cong \underset{\sim}{b}$ of rank k into six cases.

	Case a	Case b
1.	$m = n = \text{rank}(A)$	$k = \text{rank}(A) < m = n$
2.	$k = n < m$	$k < n < m$
3.	$k = m < n$	$k < m < n$

Table 3. Various Relations Possible Between m, n, and k = rank (A).

In (3.22) - (3.27) of [3] there are six matrices illustrating 1.a, 2.a, 3.a, 1.b, 2.b, and 3.b, respectively. The last three cases are classified as rank deficient by Algorithn HFTI, [3], using a tolerance $\tau = 10^{-4}\|A\|$. These provide us with a set of test problems for each of the six categories of Table 3. The right-side vector $\underset{\sim}{b}$ used in all cases was the m-vector

$[1, \ldots, 1]^T$. Each of the systems was scaled by the constant 10^4 so that all of the data in the matrices could be input as integers.

To generate inequality constraints the unconstrained solution of minimal length, $\underset{\sim}{x}$, determined using Algorithm HFTI, [3], was used, together with the 6 by 6 matrix 10^4Q using Q of (3.26), [3]. The six-vector $\underset{\sim}{h}' =$ (first n columns of $10^4Q)\underset{\sim}{x} \equiv Gx$ was computed. Then we defined the right-side vector $\underset{\sim}{h}$ of the constraints by

$$h_{2i-1} = h'_{2i-1} \quad , \qquad i = 1, 2, 3$$

$$h_{2i} = h'_{2i} - 10^{i-1} \quad , \qquad i = 1, 2, 3$$

Thus the constrained least squares problem formulated using six slack variables as in Eqs. (3)-(4) will have a solution $(\underset{\sim}{x}^T, \underset{\sim}{w}^T)^T = (\underset{\sim}{x}^T, 0, 1, 0, 10, 0, 100)^T$. In the sample cases 3.a, 1.b, 2.b, and 3.b we cannot expect our algorithm to obtain that solution, however. This is due to the fact that the solution (with $\underset{\sim}{w} \geqslant \underset{\sim}{0}$) is not uniquely determined in those cases which are rank deficient. However, the residual vector length will be minimized at the solution obtained.

Tables 4-9 show the results obtained using Algorithm WNNLS described in Section 3. These tables show that $\underset{\sim}{w} \geqslant \underset{\sim}{0}$ and that the residual vector length is acceptably minimized.

i	$\hat{\tilde{x}}$	$\hat{\tilde{w}}$
1	-3.21200	$1. \times 10^{-9}$
2	-0.07557	1.000
3	3.52600	0.
4	--	10.000
5	--	$2. \times 10^{-9}$
6	--	100.000

|(Residual Vector Length HFTI) - (Residual Vector Length WNNLS)| = $2. \times 10^{-11}$

Table 4. Sample Case 1.a $m = n = k = 3$

i	$\hat{\tilde{x}}$	$\hat{\tilde{w}}$
1	-9.077×10^{-3}	0.4639
2	0.35880	1.4990
3	0.49770	0.
4	0.61560	10.8900
5	0.27320	0.
6	--	100.1000

|(Residual Vector Length HFTI) - (Residual Vector Length WNNLS)| = $3. \times 10^{-5}$

Table 5. Sample Case 1.b $m = n = 5,\ k = 3$

i	$\hat{\tilde{x}}$	$\hat{\tilde{w}}$
1	1.77000	0.
2	0.34940	1.0000
3	--	0.
4	--	10.000
5	--	0.
6	--	100.0000

|(Residual Vector Length HFTI) - (Residual Vector Length WNNLS)| = 0.

Table 6. Sample Case 2.a $m = 3,\ n = 2,\ k = 2$

i	$\hat{\underset{\sim}{x}}$	$\hat{\underset{\sim}{w}}$
1	0.04648	0.
2	0.34270	0.3611
3	0.51270	0.
4	0.58510	8.4270
5	0.29040	1.9730
6		99.5100

|(Residual Vector Length HFTI) - (Residual Vector Length WNNLS)| = 3. x 10^{-5}

Table 7. Sample Case 2.b m = 6, n = 5, k = 3

i	$\hat{\underset{\sim}{x}}$	$\hat{\underset{\sim}{w}}$
1	-1.34700	5. x 10^{-10}
2	3.16200	1.0000
3	0.30450	0.
4	--	10.0000
5	--	6. x 10^{-10}
6	--	100.0000

|(Residual Vector Length HFTI) - (Residual Vector Length WNNLS)| = 2. x 10^{-10}

Table 8. Sample Case 3.a m = 2, n = 3, k = 2

i	$\hat{\underset{\sim}{x}}$	$\hat{\underset{\sim}{w}}$
1	0.44080	0.
2	0.04467	0.9417
3	1.26000	0.0585
4	0.04087	10.0200
5	-0.18260	0.
6	--	99.9500

|(Residual Vector Length HFTI) - (Residual Vector Length WNNLS)| = 2. x 10^{-6}

Table 9. Sample Case 3.b m = 4, n = 5, k = 3

BIBLIOGRAPHY

[1] Joseph Stoer, "On the Numerical Solution of Constrained Least Squares Problems," SIAM J. Numer. Anal., 8, No. 2, (1971), 382-411.

[2] G. H. Golub and Michael A. Saunders, "Linear Least Squares and Quadratic Programming," Integer and Nonlinear Programming, II, J. Abadie (ed.), North-Holland Publ. Co. (1970), 229-256.

[3] C. L. Lawson and R. J. Hanson, Solving Least Squares Problems, Prentice-Hall (1974).

[4] Alan Cline, "The Transformation of a Quadratic Programming Problem into Solvable Form," ICASE Report No. 75-14, August 1975.

[5] P. Wolfe, "Finding the Nearest Point in a Polytope," Math. Prog. 11, No. 2 (1976), 128-149.

[6] C. L. Lawson, "On the Discovery and Description of Mathematical Programming Algorithms," Lecture Notes in Mathematics, Vol. 506, A. Dold, B. Eckmann (eds.), SpringerVerlag (1976), 157-165.

[7] M. Gentleman, "Least Squares Computations by Givens Transformations without Square Roots," J. Inst. Maths. Applics., 12, (1973), 329-336.

[8] C. L. Lawson, R. J. Hanson, D. R. Kincaid, F. T. Krogh, "Basic Linear Algebra Subprograms for FORTRAN Usage," Trans. Math. Software (to appear).

[9] T. Beyer, "FLECS--FORTRAN Language with Extended Control Structures. User's Manual," University of Oregon Computing Center, Eugene, OR, September 1974.

[10] Lars Eldén, "Numerical Analysis of Regularization and Constrained Least Squares Methods," Part V, Linkoping Studies in Science and Technology, Dissertations. No. 21, LiTH-MAT-R-1977-20, 1977.

[11] C. L. Lawson, "The Covariance Matrix for the Solution Vector of an Equality-Constrained Least-Squares Problem," Jet Propulsion Laboratory, Tech. Memo. 33-807, December 1976.

[12] R. J. Hanson, "Selected Algorithms for Least Squares Computations - A User's Guide," SAND77-1090, February 1978.

[13] G. Strang and G. Fix, An Analysis of the Finite Element Method, Prentice-Hall (1973), p. 56.

[14] Carl de Boor, "Package for Calculating with B-Splines," SIAM J. Numer. Anal., 14, No. 3, (1977), 441-472.

[15] Haskell, K. H., Hanson, R. J., "Selected Algorithms for the Linearly Constrained Least Squares Problem - A User's Guide," SAND78-1290.

KH:0443A:jc:06/17/78

Distribution:

J. A. Wisniewski
528 Schroeder Ave., Apt. 1
Peotone, IL 60468

Dept. of Computer Science
Report Section
Royal Institute of Technology
5-100 44
Stockholm 70, SWEDEN

1116	J. D. Plimpton
1223	R. B. Easterling
1223	I. J. Hall
1223	R. L. Iman
1223	R. R. Prairie
2600	L. E. Hollingsworth
2610	R. E. Detry
2613	E. A. Aronson
2613	L. A. Bertram
2613	D. Ghiglia
2613	B. Marder
2613	K. Haskell (25)
2613	R. E. Jones
2613	M. R. Scott
2613	W. Vandevender
2613	H. A. Watts
2614	A. R. Iacoletti
2640	J. L. Tischhauser
4231	F. Biggs
4231	P. J. McDaniel
4410	R. K. Cole
4410	S. L. Daniel
4410	A. W. Frazier
4732	L. C. Bartel
5532	D. Hicks
5600	D. B. Shuster
	Attn: 5610 A. A. Lieber
	5620 M. M. Newsom
	5630 R. C. Maydew
5640	G. J. Simmons
5641	H. T. Davis
5641	C. A. Morgan
5641	C. P. Steck
5641	R. J. Thompson
5642	D. E. Amos
5642	R. J. Hanson (25)
5642	K. L. Hiebert
5642	B. L. Hulme
5642	L. F. Shampine
8322	R. J. Kee
8325	R. E. Chang
8325	R. E. Huddleston
8325	T. H. Jefferson
8325	T. A. Manteuffel
8325	D. Crawford
8327	J. F. Lathrop
8326	A. A. Aas
3141	T. L. Werner (5)
3151	W. L. Garner (3)
	For DOE/TIC (Unlim. Release)
3172-3	R. Campbell (25)
	For DOE/TIC

Org.	Bldg.	Name	Rec'd by *	Org.	Bldg.	Name	Rec'd by *

* Recipient must initial on classified documents.