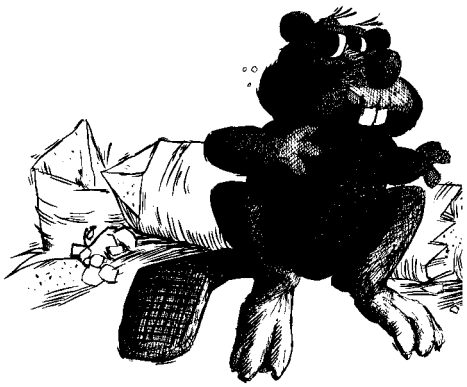
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT



Primal Methods are Better Than Dual Methods for Solving Overdetermined Linear Systems in the l_∞ Sense?

Richard H. Bartels
Andrew R. Conn
Yuying Li

Research Report
CS-87-12

February 1987

Primal Methods are Better Than Dual Methods for Solving Overdetermined Linear Systems in the l_∞ Sense?

Richard H. Bartels

Andrew R. Conn

Yuying Li

Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada

Abstract.

It is generally believed that methods based upon solving the dual of the linear programming formulation of the linear l_∞ problem are superior to methods based upon the primal formulation.

We present a primal approach along with evidence that for random problems primal methods are to be preferred. We then show that the method that is generally considered the best algorithm for discrete linear approximation in the l_∞ norm actually owes much of its efficiency to effective choice of a starting point. This phenomenon is explained with the aid of results from classical minimax theory, which are then used to determine a suitable initial point for a primal method.

With this enhancement it would appear that the primal approach is also preferable for data-fitting problems.

Abbreviated Title: Primal Linear l_∞ Algorithms.

Keywords: linear Chebychev approximation; primal descent algorithms; linear l_∞ approximation.

1. Introduction

We are interested in solving the following problem:

Given an $n \times m$ matrix

$$A = \begin{bmatrix} a_1 & \cdots & a_m \end{bmatrix} \in \mathbf{R}^{n \times m}, \quad (m > n \geq 2) \tag{1.1}$$

and a right hand side vector

$$b = \begin{bmatrix} \beta_1 & \ldots & \beta_m \end{bmatrix}^T \in \mathbf{R}^m \tag{1.2}$$

we wish to find a vector $x \in \mathbf{R}^n$ such that

$$||A^T x - b||_\infty = \max_{1 \leq i \leq m} |a_i^T x - \beta_i| \tag{1.3}$$

is minimized.

This problem is termed the discrete, linear l_∞ (or Chebychev) problem.

In [7] we discussed the solution of this problem by so-called primal techniques. All such methods essentially reduce the maximum modulus of the residual components at each iteration and are finite algorithms.

Most other techniques formulate a related linear programming problem and then solve its dual. (See for example [1], [12] and [15].) It is generally believed that the dual approach is superior to the primal approach. However, it is the authors' contention that this belief is not justified.

We will first provide evidence to show that for random problems the primal approach appears to be distinctly superior to the dual approach (see also [6] and [7]).

In [7] it appeared, however, that for function approximation problems, an important source of Chebychev problems, the dual method of [1] performed better than the primal algorithms. Indeed, Barrodale and Phillips is the usual algorithm of preference.

Motivated by the curious discrepancy between random and data fitting problems, we conjectured that the apparent superiority of Barrodale and Phillips for function approximation problems was due, not to an inherent superiority of the approach, but rather as a consequence of the structure of function approximation problems, it being obvious that the random problems were essentially non-structured. Furthermore, we conjectured that this structure, characterized theoretically by the alternating sign property (the details of which will be summarized below) manifests itself particularly in the choice of starting point.

In this paper we shall show that the method of Barrodale and Phillips typically chooses an excellent starting point in the following sense. Suppose k is the rank of the coefficient matrix, A, of (1.1). Then the algorithm of [1], at the end of iteration $k+1$, has a superior initial basic feasible solution for the subsequent (exchange) iterations of the dual linear program. When appropriate attention is given to the choice of starting point for the primal approach, as

applied to function approximation problems, it once again appears that the primal method is to be preferred.

2. Description of the Barrodale and Phillips Algorithm

In [2] the authors suggest that the algorithm converges in relatively few iterations, in general, because of its automatic construction of an initial approximation which is already quite close to the best solution. However, they give no proof that the initial approximation is predominantly responsible for this efficiency, nor do they offer any explanation as to why the initial point is likely to be a good approximation to the solution.

Since we are particularly interested in the function approximation problem it is convenient to use a slightly different notation.

Let $L = < \phi_1 \, \phi_2 \ldots \phi_n >$ denote the linear space spanned by the functions ϕ_i , $i = 1 , \ldots , n$ on the closed interval of the real line, $[a, b]$, where the ϕ_i are continuous on $[a, b]$.

A function ϕ is in L, if and only if

$$\phi(z) = \sum_{i=1}^{n} \alpha_i \, \phi_i(z)$$

where $z \in [a, b]$. Given $m > n \geq 2$ data points (z_i, y_i), we are to determine a discrete l_∞ approximation function in L (equivalently, determine $\alpha = (\alpha_1 , \ldots , \alpha_n)$) which minimizes

$$(2.1) \qquad e(\alpha_1 \cdots \alpha_n) = \left|\left| y_i - \sum_{j=1}^{n} \alpha_j \, \phi_j \, (z_i) \right|\right|_\infty$$

$$= \max_{1 \leq i \leq m} \left| y_i - \sum_{j=1}^{n} \alpha_j \, \phi_j \, (z_i) \right| .$$

Thus, in the notation of (1.3) we have that

$$(2.2) \qquad x = \alpha, \quad A = \begin{bmatrix} \phi_1(z_1) & \cdots & \phi_1(z_m) \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ \phi_n(z_1) & \cdots & \phi_n(z_m) \end{bmatrix} \quad \text{and} \quad b = (y_1, \ldots, y_m)^T .$$

It is clear that the optimal α^* that solves (2.1) can be determined from the linear programming problem, in $n+1$ variables $\alpha_1 \cdots \alpha_n$ and ξ, given by

$$(2.3) \qquad \begin{array}{ll} \text{Minimize} & \xi \\ \left.\begin{array}{ll} \text{subject to} & y_i - \sum_{j=1}^{n} \alpha_j \, \phi_j(z_i) \leq \xi \\ \text{and} & -y_i + \sum_{j=1}^{n} \alpha_j \, \phi_j(z_i) \leq \xi \end{array}\right\}, & i = 1, 2, \ldots, m . \end{array}$$

Moreover, it is evident that we may add the (redundant) linear constraint $\xi \geq 0$.

The corresponding dual linear program in $2m$ variables $\underline{\sigma}_1, \ldots, \underline{\sigma}_m, \underline{\tau}_1, \ldots, \underline{\tau}_m$ is given by

$$(2.4) \qquad \begin{array}{ll} \text{Maximize} & \sum_{i=1}^{m} (\underline{\sigma}_i - \underline{\tau}_i) y_i \\ \text{subject to} & \sum_{i=1}^{m} (\underline{\sigma}_i - \underline{\tau}_i) \, \phi_{ji} = 0 , \qquad j = 1, 2, \ldots, n \\ & \sum_{i=1}^{m} (\underline{\sigma}_i + \underline{\tau}_i) \quad \leq 1 , \end{array}$$

and

$$\underline{\sigma}_i, \underline{\tau}_i \geq 0 , \quad i = 1, 2, \ldots, m ,$$

where $\phi_{ji} = \phi_j(z_i)$, and we use the underscore character to signify dual variables.

The algorithm of Barrodale and Phillips consists of solving (2.4) whilst taking advantage of the structure to suppress some of the calculations. More specifically, inserting artificial variables $\underline{\alpha}_j$ in each of the n equality constraints and a slack variable $\underline{\xi}$ in the inequality constraint we have

$$
\begin{aligned}
&\text{maximize} && \sum_{i=1}^{m} (\underline{\sigma}_i - \underline{\tau}_i) y_i \\
(2.5) \qquad &\text{subject to} && \sum_{i=1}^{m} (\underline{\sigma}_i - \underline{\tau}_i)\, \phi_{j,i} + \underline{\alpha}_j = 0 \ , \\
& && \sum_{i=1}^{m} (\underline{\sigma}_i + \underline{\tau}_i) + \underline{\xi} = 1 \ ,
\end{aligned}
$$

where it is understood that the $\underline{\alpha}_j$ will ultimately be zero. [In fact, they always have the value zero — even in the basis. The problem is degenerate.]

Denoting the column vectors formed from the coefficients of $\underline{\sigma}_i$, $\underline{\tau}_i$, $\underline{\alpha}_j$ and $\underline{\xi}$ by $\underline{s}_i$, $\underline{t}_i$, $\underline{a}_j$ and $\underline{w}$ respectively, it follows that

$$
(2.6) \qquad \underline{s}_i + \underline{t}_i = 2\underline{w} \ , \qquad i = 1 \ , \ldots , \ m \ .
$$

Furthermore, letting $\underline{b}$ denote the right hand side of the constraints (2.5) we have that

$$
(2.7) \qquad \underline{b} = \underline{w} = [0 \ , \ldots , \ 0, \ 1]^T = e_{n+1}^T
$$

where e_{n+1} denote the $n+1$st column of the $(\overline{n+1} \times \overline{n+1})$ identity matrix.

Using the relations (2.6) and (2.7) Barrodale and Phillips condense the standard simplex tableau associated with (2.5) (which would be expected to have dimensions $(n + 1)\times(2m + n + 1)$) to a tableau that is $(n+1)\times m$.

Let Φ denote the $n\times m$ matrix given by

$$
\Phi = (\phi_{ij})_{\substack{1\leq i\leq n \\ 1\leq j\leq m}} \ .
$$

Suppose Φ is of rank k, then the algorithm of Barrodale and Phillips starts with the initial basis $\underline{\alpha}$, $\underline{\xi}$ and can be divided into three consecutive stages. Stage 1 refers to the first k simplex iterations. Stage 2 refers to the single $k+1$st simplex iteration and stage 3 consists of all the remaining (finite) iterations.

In Stage 1, only $\underline{\sigma}_i$ can enter the basis matrix and only $\underline{\alpha}_j$ can leave the basis matrix. Corresponding to each $\underline{\sigma}_t$, say, that enters the basis, we have that the tth residual is zero, i.e.

$$y_t - \sum_{j=1}^{n} \alpha_j \, \phi_j(z_t) = \xi \, .$$

Moreover, the $\underline{\sigma}_t$ that enters the basis corresponds to that with the largest absolute reduced cost. With respect to the primal problem this corresponds to chosing the residual that is largest in magnitude. Since $\underline{\xi}$ and the chosen $\underline{\sigma}_i$'s remain in the basis during this stage it follows that the selected residuals are reduced to zero at each iteration and remain there during subsequent iterations. Thus at the end of Stage 1 we have k residuals at zero, (and $n-k$ α's and ξ still basic).

In Stage 2 the variable to leave the basis is $\underline{\xi}$, corresponding to the primal variable ξ becoming non-zero. Another $\underline{\sigma}_i$ (or $\underline{\tau}_i$) enters the basis, along with possibly an exchange of $\underline{\sigma}_j$'s and $\underline{\tau}_j$'s to maintain feasibility. Consequently, at the end of stage $k+1$ the approximating function possesses $k+1$ residuals of magnitude equal to the resulting value of ξ.

Stage 3 is equivalent to the well-known exchange algorithm for linear minimax approximation (see [14], chapter 8, for example).

In the light of the above comments it is clear that an equivalent formulation of the algorithm of Barrodale and Phillips can be stated in a direct way based on the original statement of the problem as follows.

Let

$$r_i^k \triangleq y_i - \sum_{j=1}^{n} \alpha_j^k \, \phi_j(z_i)$$

and

$$\sigma_i = \text{sign}\,(r_i) \, .$$

Stage 1

{Initialize}

$$k = 0 \, ,$$

$$P^{(k)} = I \ , \quad \mathcal{A} = \emptyset \ , \quad \text{the empty set} \ .$$

{Iteration k}

$$t \triangleq \{ i \mid |r_i| \text{ is maximal} \} \ ,$$

{choose search direction}

$$d^k = -\sigma_t \, P^{(k)} \nabla r_t(\alpha^k) = \sigma_t \, P^{(k)} \begin{bmatrix} \phi_1(z^t) \\ \cdot \\ \cdot \\ \cdot \\ \phi_n(z^t) \end{bmatrix} = \sigma_t \, P^{(k)} \, a_t \ ,$$

where

$$a_t = [\phi_1(z^t) \ \cdots \ \phi_n(z^t)]^T \ .$$

{choose stepsize}

$$\text{choose} \quad \lambda_t \quad \text{so that}$$

$$r_t^k(\alpha^k + \lambda_t d^k) = 0 \ ,$$

or equivalently

$$\lambda_t = \frac{r_t^k(\alpha^k)}{[d^k]^T a_t} \ .$$

{update the orthogonal projection matrix and loop}

$$\mathcal{A} = \mathcal{A} \cup \{t\} \ ,$$

$$k = k + 1 \ ,$$

$$P^{(k)} = I - A^k (A^{k_T} A^k)^{-1} A^{k_T} \ ,$$

where

$$A^k = [a_1 \ \cdots \ a_s] \quad \text{with} \quad \mathcal{A} = \{1, 2 \ , \ldots , \ s\}.$$

{Return to iteration k until $|\mathcal{A}| =$ rank of A.}

Note: The above is not meant to be a practical statement of the implementation but merely a statement of the necessary algebra. This remark applies equally to Stage 2 and Stage 3, below. Moreover, we mean equivalent in the sense that all primal versions of the simplex method for linear programming move from a primal feasible basic solution to an adjacent primal feasible basic solution whilst improving the objective function and are, in this sense, equivalent. However, the path to an optimal solution is not thereby uniquely defined. In an analogous manner $P^{(k)}$ above could be replaced by any projection onto the null space of A^k — not necessarily an orthogonal projection.

Stage 2

{iteration $k = |\mathcal{A}| + 1$}

Determine $t \triangleq \{i \mid |r_i| \text{ is maximal}\}$.

{Choose search direction}

$$d^k = \sigma_t \, Pa_t \ ,$$

where

$$P = I - N(N^TN)^{-1} N^T \ \text{ with } \ N = [\sigma_1 a_1 - \sigma_2 a_2, \sigma_1 a_1 - \sigma_3 a_3 \ \cdots \ \sigma_1 a_1 - \sigma_k a_k]$$

and

$$\mathcal{A} = \{1 \ , \ldots , \ k\}$$

from stage 1. Thus d^k decreases the magnitude of a_t whilst changing $a_1 \ , \ldots , \ a_k$ simultaneously.

{Choose the stepsize}

choose λ_t so that

$$\sigma_t r_t(\alpha^k + \lambda_t d^k) = \sigma_1 r_1(\alpha^k + \lambda_t d^k) \ ,$$

or equivalently

$$|r_t(\alpha^k)| + \lambda_t \sigma_t (d^k)^T a_t = |r_1(\alpha^k)| + \lambda_t \sigma_1 (d^k)^T a_1 \ .$$

Thus,

$$\lambda_t = \frac{|r_t(\alpha^k)|}{(d^k)^T[\sigma_t a_t - \sigma_1 a_1]} .$$

Consequently, r_t and r_1 have the same magnitude at $\alpha^k + \lambda_t d^k$ and hence, by the choice of d^k, so do $r_2, r_3 \cdots r_k$. In other words we now have $k+1$ residuals at the same value.

Stage 3

For the third stage, we have

$$\mathcal{A} = \{1, 2, \ldots, k+1\} ,$$

say. Put $l = |\mathcal{A}| + 1$.

{iteration l}

Let A_s^k denote the matrix, $n \times k$, whose columns are given by $\sigma_s a_s - \sigma_i a_i$, $i \in \mathcal{A}$, $i \neq s$.

{test for optimality}

Let $t \in \{i \mid |r_i| \text{ is maximal and } t \notin \mathcal{A}\}$. If $|r_t| \leq |r_i|$, $i \in \mathcal{A}$ stop (optimal, see below) .

{determine multipliers}

Solve

$$\sigma_t a_t = A_1^k u , \tag{2.8}$$

{update the projection matrix and iteration count}

Clearly (2.8) has an exact unique solution since the rank of A is k. Let $P_{s,j}$ denote the projection given by

$$P_{s,j} = I - A_{s,j}^k \left(\left[A_{s,j}^k\right]^T A_{s,j}^k \right)^{-1} \left[A_{s,j}^k\right]^T , \tag{2.9}$$

where $A_{s,j}$ denotes the $n \times (k-1)$ matrix whose columns are given by $\sigma_s a_s - \sigma_i a_i$ $i \in \mathcal{A}$, $i \neq s$ or j.

Put $l = l + 1$

{choose search direction}

If u given by (2.8) is such that

$$u_j < 0 \ , \tag{2.10}$$

define $d^k = -\sigma_t \ P_{1,j} \ a_t$ (i.e. supposing s=1).

Clearly $[d^k]^T \ a_t = -\sigma_t \, ||P_{1,j} \ a_t \, ||^2$, which implies that $|r_t \, |$ descends.

Moreover,

$$\left(\sigma_1 a_1 - \sigma_s a_s\right)^T d^k = 0 \ , \ \ s \neq j \ , \ \ s \in \mathcal{A} \ ,$$

$$\left(\sigma_1 a_1 - \sigma_j a_j\right)^T d^k = -\sigma_t (\sigma_1 a_1 - \sigma_j a_j)^T \ P_{1,j} \ a_t$$

$$= -\sigma_t \ u_j \, ||P_{1,j}(\sigma_1 a_1 - \sigma_j a_j) \, ||^2 > 0 \ , \qquad \text{[using (2.8), (2.9) and 2.10]} \ .$$

Thus, the first equation implies that

$$r_1(\alpha^k + \lambda d^k) = r_s(\alpha^k + \lambda d^k) \ , \ \ s \neq 1 \ , \ \ s \in \mathcal{A} \ ,$$

and the second equation implies that

$$|r_j(\alpha^k + \lambda d^k)| \ < \ |r_1(\alpha^k + \lambda d^k)| \ ,$$

for λ sufficiently small and positive.

{choose stepsize}

The stepsize is that which determined the first residual whose value equals that of $|r_s \, |$, $s \in \mathcal{A}$, $s \neq j$.

{return to iteration l}

Equation (2.9) was developed on the assumption that the current point is not optimal and $j \neq 1$. However it may be that we need to "drop" a_1. That is, we may require that

$$r_2(\alpha^k + \lambda d^k) = r_s(\alpha^k + \lambda d^k) \ , \qquad s \neq 2$$

and

$$|r_1(\alpha^k + \lambda d^k)| \quad < \quad |r_2(\alpha^k + \lambda d^k)| .$$

From (2.8) we have that

$$\sigma_t a_t = \sum_{\substack{i=1 \\ i \neq 2}}^{k} \bar{u}_i(\sigma_2 a_2 - \sigma_i a_i) , \tag{2.11}$$

where

$$\bar{u}_1 = -\sum_{i=2}^{k} u_i \quad \text{and} \quad \bar{u}_i = u_i \quad i \neq 1 .$$

Thus we are able to drop a_1 if $\bar{u}_1 < 0$, or in terms of the original multipliers u_i, if

$$0 > -\sum_{i=2}^{k} u_i .$$

By hypothesis $u_i \geq 0, i = 2, \ldots, k$ and at least one of the u_i is strictly positive (as follows from (2.8)). Hence $\bar{u}_1 < 0$ is always satisfied. Note this implies that we know immediately in this case that we can drop a_1. Thus, we are optimal when there is no maximal residual (in modulus) greater than $|r_i|$, $i \in \mathcal{A}$.

Thus, assuming we were not optimal, once again at the end of the iteration we have $k+1$ residuals of equal value. Moreover the maximum residual has clearly descended since no absolute residual $|r_i|$, $i \in \mathcal{A}$, is allowed to excede $|r_t|$ by the choice of stepsize, and $|r_t|$ has descended by the choice of d.

More specifically we choose λ to be the minimum positive λ such that

$$|\sigma_s r_s(\alpha^k + \lambda d^k)| = |\sigma_i r_i(\alpha^k + \lambda d^k)| , \text{ where } s \in \mathcal{A} , \ i \notin \mathcal{A}$$

i.e.

$$\lambda_i = \frac{-\sigma_s r_s(\alpha^k) + \sigma_i r_i(\alpha^k)}{[\sigma_s a_s - \sigma_i a_i]^T d^k}$$

if r_s and r_i do not change sign from α^k to $\alpha^k + \lambda d^k$, with analogous formulae if r_s and r_i (or both) change sign. One then chooses

$$\lambda = \min \{\lambda_i \mid \lambda_i > 0\}.$$

Thus, in summary the method of Barrodale and Phillips initially chooses the residual of maximum modules and brings it down to zero. In the next iteration, the residual of maximum modules is again chosen (which possibly is larger than the maximum modules of the previous iteration) and brought down to zero whilst maintaining any residuals already at zero. This is repeated for k iterations where k is the rank of the coefficient matrix A. A single iteration follows where the residual of maximum modulus is reduced (in absolute value) whilst the residuals at zero are brought up (in absolute value) together, until one has $k+1$ residuals all at the same value. These residuals are not necessarily the maximum residuals.

All subsequent iterations correspond to choosing that residual which has the largest absolute value and reducing it whilst maintaining k of the $k+1$ identified residuals at a common magnitude. One continues until once again one has $k+1$ residuals all at the same absolute value. Since the maximum magnitude has decreased, improvement is assured. One terminates when no residual is identified with a larger value in magnitude than the current $k+1$ residuals, whereupon optimality is achieved.

This is in marked contrast to the proposed primal method of this article which can be summarized as follows.

Initially, choose the residual of maximum modules, reduce it as much as possible until one "crosses" an increasing (in the absolute sense) residual. Now determine a search direction such that the residuals of equal maximum modules descend together until a new residual is "crossed". After at most k steps (where k again refers to the rank of A) we will have $k+1$ residuals of equal magnitude with all other residuals smaller in magnitude. It now follows from a theorem of the alternative equivalent to Farkas' Lemma (Gordan's Theorem) that either we can find a search direction that reduces the magnitude of k of the $k+1$ residuals together, with the $k+1^{st}$ remaining residual reducing more rapidly, or we are optimal. In the former case one reduces the k residuals until a new (increasing in magnitude) residual is met, whereupon, once again, we have $k+1$ equal residuals. We have thus established the prototype of all remaining iterations.

Intuitively, we believe the direct approach of the primal method is much more natural than the dual approach and the primal interpretation of both algorithms suggest that the primal approach, contrary to popular belief, is superior. We hope to justify this intuition below.

3. Basic Classical Chebychev Approximation Theory

In order to understand the significance of the initial point obtained at the end of stage two of Barrodale and Phillips' algorithm we shall first review some concepts and theorems of classical minimax approximation theory.

Definition 3.1

An n dimensional linear function space $L = <\phi_1 \cdots \phi_n>$ defined on the interval $[a, b]$ is said to satisfy **the Haar condition** if and only if, for every non-zero ϕ in L, the number of roots of the equation $\phi(z) = 0$ on $[a, b]$ is less than n, the dimension of L.

Definition 3.2

A reference on $[a, b]$ is a set $\{q_i\}_{i=1}^{n+1}$ of $n+1$ distinct points in $\mathbf{R}$ where we assume that

$$a < q_1 < q_2 \cdots < q_{n+1} \leq b \tag{3.1}$$

and it is understood that the reference is defined with respect to an n dimensional linear function space.

Definition 3.3

Let $\{q_i\}_{i=1}^{n+1}$ be a reference. The corresponding values $\phi(q_i)$ of **any** function $\phi(q)$ in L are related by the equation

$$\sum_{i=1}^{n+1} \theta_i \, \phi(q_i) = 0 \, , \tag{3.2}$$

for some suitable $\theta = (\theta_1 \cdots \theta_{n+1})^T \in \mathbf{R}^{n+1}$.

Equation (3.2) is called the **characteristic relation**.

Definition 3.4

Let $\phi(q)$ be any function in L and $\rho_i = y_i - \phi(q_i)$ be the errors of the approximation at the points q_i of the reference $\{q_i\}_{i=1}^{n+1}$. The function $\phi(q)$ is called a **reference function** with respect to the reference set $\{q_i\}_{i=1}^{n+1}$ if

$$\text{sgn}\,(\rho_i) = \text{sgn}\,(\theta_i) \qquad i = 1\,,\ldots,\,n+1 \tag{3.3}$$

or

$$\text{sgn}\,(\rho_i) = -\text{sgn}\,(\theta_i) \qquad i = 1\,,\ldots,\,n+1\,.$$

where the θ_i are defined by the characteristic relation (3.2).

Definition 3.5

The **levelled reference function** with respect to a given reference $\{q_i\}$ is characterized by the property that the errors ρ_i have the same absolute value. The common absolute value $|\rho_i|$ of the approximation errors is called the **reference deviation**.

For these definitions, and much more, the reader is referred to the excellent text [14].

We shall state, without proof (see [14] page 77) the following fundamental characterization theorem.

Theorem 3.1

Let L be an n-dimensional subspace of $C[a,b]$, the set of continuous real valued functions that are defined on the interval $[a,b]$ of the real line. Furthermore, we assume that L satisfies the Haar condition and $f \in C[a,b]$. Then ϕ^* is the best minimax approximation from L to f if and only if there exists a reference $\{q_i\}_{i=1}^{n+1}$ on $[a,b]$ such that

$$|f(q_i) - \phi^*(q_i)| = ||f - \phi^*||_\infty \tag{3.4}$$

and

$$f(q_{i+1}) - \phi^*(q_{i+1}) = -\,[f(q_i) - \phi^*(q_i)] \qquad i = 1, 2, \ldots, n\ .$$

Equivalently the theorem can be stated as follows: The function of best approximation is the reference function with maximal reference deviation.

We will also require

Theorem 3.2

Let L be an n-dimensional subspace of $C[a, b]$ that satisfies the Haar condition. Furthermore, let $\{q_i\}_{i=1}^{n+1}$ be a reference on $[a, b]$ and let $\{\theta_i\}_{i=1}^{n+1}$ be a set of real multipliers that are not all zero and satisfy the characteristic equation

$$\sum_{i=1}^{n+1} \theta_i\ \phi(q_i) = 0$$

for all functions ϕ in L. Then every multiplier is non-zero and their signs alternate. Once again, the reader is referred to ([14], page 18) for a proof.

We now wish to establish certain key properties of the point determined by the algorithm of Barrodale and Phillips at the end of stage two.

4. Barrodale and Phillips' Initial Point for the Exchange Algorithm

We wish to show that the trial approximation obtained after stage two of the algorithm of Barrodale and Phillips is a levelled reference function that satisfies the alternating sign property.

Recalling the dual linear program (2.4), we introduce several algebraic transformations to map points in the dual space into corresponding points in the primal space.

Let $\underline{B}^k$ denote the current dual basis matrix, $[(n+1)\times(n+1)]$, whose columns are given by $\underline{B}_1^k, \ldots, \underline{B}_{n+1}^k$ at iteration k. In what follows, the iteration number is omitted whenever no confusion results. Let $\underline{c}_B$ denote the cost vector associated with the basis matrix $\underline{B}$.

Define $f_i^k : \mathbf{R} \to \mathbf{R}$ by

$$f_i^k(\underline{a}_i) = \underline{c}_B^T(\underline{B}^k)^{-1}\underline{a}_i \ , \quad i = 1, 2 \ , \ldots , n \ . \tag{4.1}$$

As above, we may omit the iteration index k and write

$$f_i(\underline{a}_i) = \underline{c}_B^T\underline{B}^{-1}\underline{a}_i \ . \tag{4.2}$$

Similarly we write

$$g_i(\underline{s}_i) = \underline{c}_B^T\underline{B}^{-1}\underline{s}_i - y_i \ , \quad i = 1, 2 \ , \ldots , m \ , \tag{4.3}$$

$$h_i(\underline{t}_i) = \underline{c}_B^T\underline{B}^{-1}\underline{t}_i + y_i \ , \quad i = 1, 2 \ , \ldots , m \ , \tag{4.4}$$

and

$$i(\underline{w}) = \underline{c}_B^T\underline{B}^{-1}\underline{w} \ . \tag{4.5}$$

We will now show that if we write

$$\begin{aligned} \alpha_i &= f_i(\underline{a}_i) \ , \quad i = 1, 2, \cdots n \ , \\ \sigma_i &= g_i(\underline{s}_i) \ , \quad i = 1, 2 \ , \ldots , n \ , \\ \tau_i &= h_i(\underline{t}_i) \ , \quad i = 1, 2 \ , \ldots , n \ , \end{aligned} \tag{4.6}$$

and

$$\xi = i(\underline{w}) \ ,$$

then provided the rank of A is n, these α_i, σ_i, τ_i and ξ satisfy the constraints of the primal problem.

Lemma 4.1

Assuming the rank of Φ is n. Consider the mapping M defined by

$$M(\underline{a}_1 \ \cdots \ \underline{a}_n \ \underline{s}_1 \ \cdots \ \underline{s}_m \ \underline{t}_1 \ \cdots \ \underline{t}_m \ \underline{b}) = [\alpha^T \ \sigma^T \ \tau^T \ \xi^T] \ .$$

Then $[\alpha^T \ \sigma^T \ \tau^T \ \xi]$ satisfies the constraints

$$y_i - \sum_{j=1}^{n} \alpha_j \ \phi_j(z_i) + \sigma_i = \xi \ , \tag{4.7}$$

$$-y_i + \sum_{j=1}^{n} \alpha_j \, \phi_j(z_i) + \tau_i = \xi \, , \tag{4.8}$$

$$\xi \geq 0 \, . \tag{4.9}$$

Proof

Now

$$\underline{s}_i \triangleq \underline{S} e_i = \begin{bmatrix} \Phi \\ e^T \end{bmatrix} e_i \, .$$

Thus $\underline{c}_B^T \underline{B}^{-1} \underline{S} = y + \sigma$ [using (4.3) and (4.6)]. But $\underline{a}_i = e_i$, $i = 1 \, , \ldots , \, n$, and $\underline{w} = e_{n+1}$. Hence $C_B^T \, B^{-1} \, I = [\alpha, \, \xi]$, [using (4.2) and (4.5) and (4.6)].

It now follows that $y_i + \sigma_i = [\alpha, \, \xi] \, \underline{S} e_i = \sum_{j=1}^{n} \alpha_j \, \phi_{ji} + \xi$,

proving (4.7).

Similarly

$$\underline{t} = - \begin{bmatrix} \Phi \\ -e^T \end{bmatrix} e_i$$

and

$$\underline{c}_B^T \, \underline{B}^{-1} \, \underline{T} = \tau - y$$

$$\Rightarrow \; \tau_i - y_i = [\alpha, \, \xi] \, \underline{T} e_i = - \sum_{j=1}^{n} \alpha_j \, \phi_{ji} + \xi$$

proving (4.8).

To prove that $\xi \geq 0$ we note that, by definition of $\underline{B}$, the dual objective function $\sum_{i=1}^{m} (\underline{\sigma}_i - \underline{\tau}_i) y_i$ has value $\underline{c}_B^T \, \underline{B}^{-1} \, e_{n+1} = \underline{C}_B^T \, \underline{B}^{-1} \, \underline{w} = \xi$, using (4.5) and (4.6). However, by construction, the initial value of the dual objective function is zero, and this value never decreases. Consequently, $\xi \geq 0$.

We are now able to prove the following key theorem.

Theorem 4.2

Let L be an n-dimensional subspace of $C[a, \, b]$ spanned by the basis functions ϕ_i, $i = 1 \, , \ldots , \, n$. Furthermore assume that L satisfies the Haar condition. Then the point, $[\alpha^T, \, \sigma^T, \, \tau^T, \, \xi]$, given by the algorithm of Barrodale

and Phillips at the end of stage two, determines a levelled reference function $\phi(z) = \sum_{j=1}^{n} \alpha_j \, \phi_j(z)$, with respect to the reference set $\{z_{i_k}\}_{k=1}^{n+1}$, $\{i_1, i_2 \cdots i_{n+1}\} \subset \{1, 2, \ldots, m\}$.

In other words

$$\operatorname{sgn}\,[y_{i_k} - \phi(z_{i_k})] = \operatorname{sgn}\,(\theta_{i_k}) \quad k = 1, \ldots, \overline{n+1},$$

or

$$\operatorname{sgn}\,[y_{i_k} - \phi(z_{i_k})] = -\operatorname{sgn}\,(\theta_{i_k}) \quad k = 1, \ldots, \overline{n+1},$$

where

$$\sum_{k=1}^{n+1} \theta_{i_k} \, \phi(z_{i_k}) = 0,$$

and

$$|y_{i_k} - \phi(z_{i_k})| = |y_{i_1} - \phi(z_{i_1})| \qquad k = 2 \cdots \overline{n+1}.$$

Furthermore,

$$y_{i_{k+1}} - \phi(z_{i_{k+1}}) = -[y_{i_k} - \phi(z_{i_k})] \qquad k = 1, \ldots, n.$$

Proof

The algorithm of Barrodale and Phillips only allows $\underline{\sigma}_j$'s to enter the basis in Stage 1. Stage 2 consists of one pivot in which either a $\underline{\sigma}_j$ or a $\underline{\tau}_j$ enters the basis, after possibly, some trivial pivots in which one or several $\underline{\sigma}_j$'s are swapped with their corresponding $\underline{\tau}_j$'s (see below).

Thus $\underline{w}$ can be written as a linear combination of the columns of $\underline{B}$, and by assumption $\underline{B}$ is $\overline{n+1} \times \overline{n+1}$. To simplify the notation somewhat, without loss of generality we can assume that at the end of Stage 2 $\{i_1 \cdots i_{n+1}\} = \{1, \ldots, n+1\}$.

Consequently, $\underline{w} = \sum_{i=1}^{n+1} \overline{\theta}_i \, \underline{B}_i$ and the basis consists of $\underline{\sigma}_i$'s and $\underline{\tau}_i$'s only. Furthermore, at least one $\lambda_i \neq 0$. Equivalently, $\underline{B}_i = \underline{s}_i$ or $\underline{t}_i$ and thus

$$0 = \sum_{i=1}^{n+1} \theta_i \, \phi_j(z_i) \, , \qquad j = 1 \, , \ldots , n \, , \tag{4.10}$$

where

$$\theta_i = \overline{\theta}_i \, , \quad \text{if } i \text{ corresponds to } \underline{\sigma}_j \text{ being in the basis,}$$

$$= -\overline{\theta}_i \, , \quad \text{if } i \text{ corresponds to } \underline{\tau}_j \text{ being in the basis.}$$

We used here the definition of the first n rows of $\underline{s}_j$ or $\underline{t}_j$. Moreover, without loss of generality we may assume that $\overline{\theta}_i \geq 0$, $i = 1 \, , \ldots , n+1$, since if $\overline{\theta}_i < 0$ and $\underline{B}_j = \underline{s}_j$ $(\underline{t}_j)$ then we may pivot (trivially) making $\underline{\sigma}_j$ (or $\underline{\tau}_j$) non-basic and $\underline{\tau}_j$ (or $\underline{\sigma}_j$) basic. Thus $\underline{B}_j = \underline{t}_j$ $(\underline{s}_j)$ and $\overline{\theta}_i$ changes sign. It is clear from (2.6), and the definition of $\underline{w}$ that this merely requires that the original $\overline{\theta}_i$ be scaled by $\left(1 - 2 \sum_{j=1}^{k} \overline{\theta}_{i_j}\right)^{-1}$ where k are the number of swaps required and $\theta_{i_j} < 0$, $j = 1 \, , \ldots , k$. But now, for arbitrary $\hat{\phi}$ in L, writing $\hat{\phi}(z) = \sum_{j=1}^{n} x_j \, \phi_j(z_i)$, we have that

$$\sum_{i=1}^{n+1} \theta_i \, \hat{\phi}(z_i) = \sum_{i=1}^{n+1} \theta_i \sum_{j=1}^{n} x_j \, \hat{\phi}_j(z_i) = 0 \, , \tag{4.11}$$

the characteristic equation.

Let $\rho_i \triangleq y_i - \phi(z_i)$.

For each basic variable $\underline{\sigma}_j$, say, we have

$$\underline{c}_B^T \, \underline{B}^{-1} \, \underline{s}_j = \underline{c}_B^T \, e_i = y_i \, ,$$

and similarly, for each basic variable $\underline{\tau}_j$

$$\underline{c}_B^T \, \underline{B}^{-1} \, \underline{t}_j = \underline{c}_B^T \, e_i = -y_i \, ,$$

since

$$c^T \begin{bmatrix} \underline{\alpha} \\ \hline \underline{\sigma} \\ \hline \underline{\tau} \\ \hline \underline{\xi} \end{bmatrix} = \sum_{i=1}^{m+1} (y_i \, \underline{\sigma}_i - y_i \, \underline{\tau}_i) \, .$$

But it now follows from (4.3), (4.4) and (4.6) that the corresponding primal variables σ_i and τ_i have the value zero.

Thus, in the notation of the primal, using Lemma 4.1,

$$y_i - \sum_{j=1}^{n} \alpha_j \, \phi_j(z_i) = y_i - \phi(z) = \xi \, , \qquad \text{i.e. } [\rho_i = \xi] \, , \tag{4.12}$$

if σ_i is in the basis, and

$$-y_i + \sum_{j=1}^{n} \alpha_j \, \phi_j(z_i) = -y_i + \phi(z) = \xi \, , \qquad \text{i.e. } [\rho_i = -\xi] \, , \tag{4.13}$$

if t_i is in the basis, where $\alpha_j = \underline{C}_B^T \, \underline{B}^{-1} \, \underline{a}_j$.

Thus we have $n+1$ of the primal variables σ_i, τ_i, zero and $\phi(z) = \sum_{j=1}^{n} \alpha_j \, \phi_j(z)$ is a levelled reference function with respect to the given reference $\{z_i\}_{i=1}^{n+1}$ with reference deviation ξ.

Now from Theorem 3.2 we have that every θ_i is non-zero and their signs alternate with i.

We will now complete the proof by showing that $\operatorname{sgn}(\rho_i) = \operatorname{sgn}(\theta_i)$ $i = 1, \ldots, n$, or $\operatorname{sgn}(\rho_i) = -\operatorname{sgn}(\theta_i)$ $i = 1, \ldots, n$.

Using (4.12) and (4.13) we have that

$$\rho_i = \xi \, , \text{ if } \sigma_i \text{ is in the basis,}$$

$$= -\xi \, , \text{ if } \tau_i \text{ is in the basis.}$$

Using (4.10) we have that

$$\theta_i \text{ is positive, if } \sigma_i \text{ is in the basis,}$$

and

$$\theta_i \text{ is negative, if } \tau_i \text{ is in the basis.}$$

Thus sgn (θ_i) = sgn (ρ_i). But, clearly we could have

$$0 = \sum_{i=1}^{n+1} \theta_i \, \phi_j(z_i) \, , \quad j = 1 \, , \ldots , \, n \, ,$$

where

$$\theta_i = -\overline{\theta}_i \, , \text{ if } i \text{ corresponds to } \underline{\sigma_j} \text{ being in the basis,}$$

$$= \overline{\theta}_i \, , \text{ if } i \text{ corresponds to } \underline{\tau_j} \text{ being in the basis,}$$

merely by multiplying (4.10) through by -1. Thus the theorem is proved.

5. The Proposed Direct Algorithm

Returning to the notation of Section 1, we will describe a method that is a generalization of [7].

Consider the primal linear program

$$\begin{array}{ll} \text{Minimize} & \xi \\ \text{subject to} & \begin{array}{l} \xi - a_i^T x \geq -\beta_i \, , \\ \xi + a_i^T x \geq +\beta_i \, , \end{array} \quad i = 1, 2 \, , \ldots , \, m \, , \end{array}$$

which we will write as

$$\begin{array}{ll} \text{Minimize} & c_0^T v \, , \\ \text{subject to} & c_j^T v \geq \delta_j \, , \quad j = 1 \, , \ldots , \, 2m \, , \end{array}$$

where

$$
\begin{aligned}
c_0 &= \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}, \\
c_i &= \begin{bmatrix} 1 \\ -a_i \end{bmatrix}, \\
c_{m+i} &= \begin{bmatrix} 1 \\ +a_i \end{bmatrix}, \\
\delta_i &= -\beta_i\,, \quad i = 1\,,\ldots,\,m \\
\delta_{m+i} &= +\beta_i\,, \quad i = 1\,,\ldots,\,m \\
v &= \begin{bmatrix} \xi \\ x \end{bmatrix}.
\end{aligned}
\tag{5.2}
$$

The approach is based upon solving (5.2) via a piecewise linear penalty function.

Let $\mu > 0$ be a fixed parameter. Define

$$
p(v,\,\mu) = \mu\, c_0^T\, v - \sum_{j=1}^{2m} \min\,(0,\, c_j^T\, v - \delta_j)\,. \tag{5.3}
$$

For any arbitrary $v \in \mathbf{R}^{n+1}$, (5.3) can be expanded into

$$
\begin{aligned}
p(v,\,\mu) = \mu\, c_0^T\, v &- \sum_{j\in I^0} \min\,(0,\, c_j^T\, v - \delta_j) \\
&- \sum_{j\in I^+} \min\,(0,\, c_j^T\, v - \delta_j) \\
&- \sum_{j\in I^-} \min\,(0,\, c_j^T\, v - \delta_j)\,,
\end{aligned}
$$

where the index sets are given by,

$$
\begin{aligned}
I^0 = I^0(v) &= \{j \mid c_j^T\, v = \delta_j\} = \{j_1\,,\ldots,\,j_k\}\,, \\
I^+ = I^+(v) &= \{j \mid c_j^T\, v > \delta_j\}\,, \\
I^- = I^-(v) &= \{j \mid c_j^T\, v < \delta_j\}\,.
\end{aligned}
\tag{5.4}
$$

For any $d \in \mathbf{R}^{n+1}$, and any $\lambda \geq 0$ sufficiently small, we have

$$p(v + \lambda d, \mu) = p(v, \mu) + \lambda \left[\mu\, c_0^T d - \sum_{j \in I^-} c_j^T d - \sum_{j \in I^0} \min\,(0, c_j^T d) \right]$$

$$= p(v, \mu) + \lambda h^T d + \lambda \sum_{j \in I^0} \sigma_j^-\, c_j^T d\ , \tag{5.5}$$

where

$$h = \mu\, c_0 - \sum_{j \in I^-} c_j\ , \tag{5.6}$$

and

$$\sigma_j^- \begin{array}{ll} = 0 & \text{if } c_j^T d \geq 0 \\ = -1 & \text{if } c_j^T d < 0 \end{array}, \quad j \in I^0\ .$$

We define the matrix

$$N = [c_{j_1} \ \cdots \ c_{j_k}]\ , \tag{5.7}$$

and let

$$P = I - N(N^T N)^{-1} N^T\ . \tag{5.8}$$

Thus P is the orthogonal projector onto the null space of N^T. Once again we emphasize that, in practise, we do not propose to compute P via (5.8).

Case i) $Ph \neq 0$.

If we define

$$d = -Ph \tag{5.9}$$

then it is easy to see, from the definition of P that

$$p(v + \lambda d, \mu) = p(v, \mu) + \lambda h^T d \quad \text{and} \quad h^T d < 0\ .$$

Thus d will serve as a descent direction for the function p.

Case ii) $Ph = 0$.

Assuming that the columns of N are linearly independent (guaranteed by the Haar condition, for example) h can be expressed uniquely as a linear combination of the c_{j_i}'s, $i = 1\ , \ldots ,\ k$. Otherwise x is a point of degeneracy and the reader is referred to Section 8 below for a brief discussion of how this

degeneracy may be handled. [The situation is equivalent to standard degeneracy in linear programming.]

Thus writing

$$h = \sum_{i=1}^{k} \eta_i \, c_{j_i} \, , \tag{5.10}$$

(5.5) reduces to,

$$p(v + \lambda d, \mu) = p(v, \mu) + \lambda \sum_{i=1}^{k} \left[\eta_i + \sigma_{j_i}^{-} \right] c_{j_i}^T \, d \, . \tag{5.11}$$

There are two subcases to consider.

Subcase a)

Suppose $\eta_{i_*} < 0$ for some $i_* \in I^0$. Then we choose d to satisfy

$$\begin{aligned} c_j^T d &= 0 \, , \quad j \in I^0 \, , \quad j \neq i_* \, , \\ c_{i_*}^T \, d &= 1 \, . \end{aligned} \tag{5.12}$$

Noting that (5.12) implies that $\sigma_j^- = 0$, $j \in I^0$, (5.5) becomes

$$p(v + \lambda d, \mu) = p(v, \mu) + \lambda h^T d \, . \tag{5.13}$$

Furthermore $h^T d < 0$. Since $h^T d = \eta_{i_*} \, c_{i_*}^T \, d = \eta_{i_*} < 0$, [using (5.10), (5.12)].

Subcase b)

Suppose $\eta_i \geq 0$ for all $i \in I^0$. Then no choice of d exists which will reduce the value of p further without causing at least one of the constraints of (5.11), which is currently satisfied, to become violated.

If Subcase b) holds and no constraints are violated then $v = [\xi, x]^T$ is optimal for (5.1). In fact Subcase b) is then equivalent to the Kuhn-Tucker conditions.

On the other hand, if any of the constraints are violated, then μ can be reduced ($\mu \leftarrow \mu/8$, say) and the function p can be minimized again starting from the current point z. In [11] it is stated that the linear programming problem (which in our case must be feasible) will be optimized by a point z attained after a finite (and usually very small) number of such reductions of μ. In fact, more

precisely one might notice that, in Subcase b), if there exists a $i \in I^0$ such that $\eta_i > 1$, then choosing d so as to satisfy

$$\begin{aligned} c_j^T d &= 0 \ , \quad j \in I^0 \ , \quad j \neq i \\ c_i^T d &= -1 \ , \end{aligned} \tag{5.14}$$

d is a descent direction for p.

However, $\eta_i \geq 0$ for all $i \in I^0$, and $Ph = 0$ indicates that, with the possible exception of the feasibility condition not being satisfied, one is at a Kuhn Tucker point. Thus either one is optimal (feasibility is satisfied) or the parameter μ is above the threshold value (feasibility is not satisfied). Consequently there seems to be little that can be gained by not reducing μ immediately.

If λ is increased slightly from zero then the index sets change as follows:

$$\begin{aligned} I^0(v + \lambda d) &= I^0(v) \ , \quad \text{for } d \text{ given by (5.9)} \ , \\ &= I^0(v) - j_{i_*} \ , \quad \text{for } d \text{ given by (5.12)} \ , \\ I^+(v + \lambda d) &= I^+(v) \ , \quad \text{for } d \text{ given by (5.9)} \ , \\ &= I^+(v) + j_{i_*} \ , \quad \text{for } d \text{ given by (5.12)} \ , \\ I^-(v + \lambda d) &= I^-(v) \ . \end{aligned} \tag{5.15}$$

For both choices of d, the change in p can be expressed as

$$p(v + \lambda d, \mu) = p(v, \mu) + \lambda h^T d \ .$$

We now need to discuss the choice of the stepsize λ.

To date in this section we have assumed that $\lambda \geq 0$ is sufficiently small. In fact we have required that $0 \leq \lambda \leq \lambda^{(1)}$, where $\lambda^{(1)}$ is the minimum element of

$$\Lambda^{(1)} = \{\lambda \mid \lambda = (\delta_j - c_j^T v) / c_j^T d \ \text{ and } \ \lambda > 0 \ \text{ and } \ j \in I^+ \cup I^-\} \ . \tag{5.16}$$

For a nondegenerate problem, each of the ratios $(\delta - c_j^T v) / c_j^T d$ defining $\Lambda^{(1)}$ will have a unique value.

If

$$\lambda^{(1)} = (\delta_{j_*} - c_{j_*}^T v) / (c_{j_*}^T d) \ , \ \text{for} \ \ j_* \in I^+ \ , \tag{5.17}$$

then an increase of λ past $\lambda^{(1)}$ will result in the constraint j_* being violated. This would cause the index j_* to be removed from I^+ and transferred to I^-, and it would require that the vector h be readjusted according to

$$h := h - c_{j_*} \ . \tag{5.18}$$

On the other hand, if

$$\lambda^{(1)} = (\delta_{j_*} - c_{j_*}^T v) / (c_{j_*}^T d) \ , \ \text{for} \ \ j_* \in I^- \ , \tag{5.19}$$

then an increase of λ past $\lambda^{(1)}$ will result in the constraint j_* being transferred from I^- to I^+ and a readjustment of h according to

$$h := h + c_{j_*} \ . \tag{5.20}$$

The increase of λ past $\lambda^{(1)}$ will be useful if $h^T d < 0$ for the readjusted h; that is, if d continues to be a descent direction in the region $\lambda > \lambda^{(1)}$, λ sufficiently small.

As was done in [11], we may continue increasing λ until the value $\lambda^{(t)}$ is reached, with λ the minimum of the set $\Lambda^{(t)}$, where

$$\Lambda^{(l)} = \Lambda^{(l-1)} - \min \{\Lambda^{(l-1)}\} \ , \ \text{for} \ \ l = 2 \ , \ldots , \ t \ . \tag{5.21}$$

The index t is the least for which d fails to be a descent direction after the corresponding correction, (5.18) or (5.20), is made to h. We redefine v as

$$v := v + \lambda^{(t)} d$$

We note that the index j_*, must be transferred to the index set $I^0(v)$ at the redefined point v, where j_* is the index, (5.17) or (5.19), to which the value $\lambda^{(t)}$ corresponds.

The algorithm of Bartels, Conn and Charalambous corresponds to the above algorithm where we always take $\lambda^{(t)} = \lambda^{(1)}$; that is we stop at the first possible λ.

That this corresponds to the primal description of Section 2 is easy to verify. One need merely realize that d given by (5.9) corresponds to keeping all current activities active, whereas d given by (5.12) keeps all but one of the activities active, with the "dropped" activity, i_* becoming feasible. Once one notes further the form of the objective function and constraints of (5.1), it is readily appreciated that maintaining feasibility (i.e. talking $\lambda = \lambda^{(1)}$ at each iteration) ensures that, assuming the Haar condition holds, one "picks up" a single activity for each of the initial iterations until one has $n+1$ activities. Henceforth one "drops" and "adds" a single activity. Maintaining feasibility ensures that the dropped constraint corresponds to one of the maximal residuals at the start of the iteration descending more rapidly than the maintained n activities. It is equally evident that activities always correspond to a maximum residual (in magnitude). Finding an initial feasible point is trivial. For arbitrary x one merely defines $\xi = \operatorname*{Max}_i \{a_i^T x - \beta_i \ , \ -a_i^T x + \beta_i\}$.

The more general form of the stepsize described above introduces a degree of flexibility analogous to that described in [11] for the general linear programming framework (see also the comments of Section 8, below). It also has the advantage that since it is no longer necessary to start at a feasible point, multiple related, or perturbed problems, can be solved using the previous solution as the new starting point.

6. Implementation

We will now describe briefly a stable implementation of the above primal algorithm. The basic ideas are relatively standard, since essentially one is doing no more than is required in linear programming.

The algorithm has the outline

```
{initialize}
repeat
    μ: = μ/8
    repeat
        opt: = false;
        {update}
            d = −Ph;
            if {d = 0} then
                begin
                    {obtain η}
                        if {η ≥ 0} then
                            begin opt: = true; break loop; end
                        {obtain d}
                end;
                {choose λ}
                z: = z + λd
    until opt
until {feasible}
```

The section between {update} and {choose λ} will be referred to, in brief, as {choose d}.

For {initialize} we must choose z and μ, identify the index sets I^0, I^+ and I^- and construct N. The choice of z and μ, in theory, can be arbitrary, although, as we will see below and have already alluded to, some choices of z are significantly better than others when A and b arise from function approximation problems.

Instead of constucting N explicitly it is useful to construct a factorization of N which facilitates the {choose d} operation. Our choice has been to use the factorization

$$N = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{6.1}$$

where Q is $\overline{n+1} \times \overline{n+1}$, orthonormal and R is a non-singular upper triangular matrix of order k.

In {update} we must transfer indices amongst I^0, I^+ and I^- as described in Section 5 above, to account for the just completed modification $z := z + \lambda d$. The net effect of this modification on N will be to cause one column to be added and (possibly) one column to be deleted.

Updating the QR decomposition of N under the addition or deletion of a column is straightforward and is discussed, for example in [8].

In {choose d} we must carry out the operation Ph as well as (possibly) solving $N\eta = h$ for η and $N^T d = e_*$ for d, where e_* denotes an appropriate elementary vector.

To obtain

$$d = -Ph$$

$$\text{set } d = -Q_2\, Q_2^T h \ , \tag{6.2}$$

where Q_2 consists of the last $(n+1) - k$ columns of Q.

We note that $d = 0$ iff $Q_2^T h = 0$, so we do not have to compute d in this case. For most of the algorithm, this is in fact the case and then computing $Q_2^T h$ is not necessary.

To solve

$$N\eta = h \ ,$$

set

$$u = Q_1^T h \ , \tag{6.3}$$

and backsolve

$$R\eta = u \ ,$$

for η, where Q_1 consists of the first k columns of Q.

To solve

$$N^T d = e_* ,$$

forward solve

$$R^T u = c_* , \tag{6.4}$$

and set

$$d = Q_1 u ,$$

where advantage should be taken of the special form of c_*, the column of N corresponding to e_*.

If the components of η are non-negative, the inner loop is broken, and a check for feasibility is made (which will indicate whether the linear programming problem has been solved).

If the components of η are not non-negative the {choose λ} step is to be carried out as indicated in Section 5. Concurrent with the determination of λ we may adjust I^0, I^+ and I^-, leaving only N, or more appropriately its QR factorization, to be updated at the beginning of the loop.

It should be evident that linear constraints can be incorporated into this scheme with no substantive change to the discussion.

Constraints of the form

$$g_r^t x \geq \gamma_i ,$$

need only be accommodated as

$$[0 : g_r^t] \begin{bmatrix} \xi \\ x \end{bmatrix} \geq \gamma_r .$$

If we make the definitions

$$c_{2m+r} = \begin{bmatrix} -0 \\ g_r \end{bmatrix} \text{ and } \delta_{2m+r} = \gamma_r$$

for all given r and add the inequalities $c_{2m+r}^T v \geq \delta_{2m+r}$ to (5.2) then Section 5 onward can be read essentially without change. Only the implementation of the proposed software requires slight modification. Each column a_i which is stored must spawn two constraints $c_j^T v \geq \delta_j$, while columns g_r give rise to one. An occasional "**if-then-else**" suffices to treat both a's and g's if they are stored in a single matrix array.

Constraints of the form

$$g_\rho^T x = \gamma_\rho$$

can be accommodated by defining $c_\rho = \begin{bmatrix} 0 \\ g_\rho \end{bmatrix}$ and $\delta_s = \gamma_s$ and by considering the penalty function

$$p(v, \mu) = \mu c_0^T v - \sum_j \min(0, c_j^T v - \rho_j) + \sum_s | c_s^T v - \delta_s |$$

as an extension of the earlier defined $p(v, \mu)$ of (5.3). This is essentially the function minimized in [5] and the reader is referred to that paper for details. The algorithm given there is only slightly different from the one just described. The major change is to be found in the {choose d} step. If a component η_* of the vector η given in (5.10) corresponds to a column of N arising from an equality constraint, the subcase a) corresponds to $|\eta_*| > 1$ and subcase b) corresponds to $|\eta_*| \leq 1$. Thus (5.12) will change to $C^T d = -\text{sgn}(\eta_*) e_*$ and (5.13) will hold if h is redefined by

$$h := h - \text{sgn}(\eta_*) c_*$$

where the columns of C are the gradients of the active constraints.

7. The Choice of an Initial Point

We have shown that assuming the Haar condition holds, the trial approximation function obtained from the Barrodale and Phillips algorithm after stage two is a levelled reference function with respect to a determined reference set. Moreover, the error terms of this reference set satisfy the alternating sign property.

The classical approximation theory indicates that the best approximation function is the one whose reference deviation is the maximum residual over all the data points and the error terms alternate sign.

Given that the primal methods appear to be superior to the dual methods on random problems [see below for some numerical evidence in support of this statement] and the classical approximation theory is inappropriate for random problems, it seems reasonable to conjecture that the apparent demise of primal algorithms for data fitting approximations is a direct consequence of the fact that no account of the rich structure of data fitting problems was considered in the choice of starting points.

Thus we will now investigate the question of choosing a suitable starting point for primal methods applied to data fitting problems.

A starting point such that the corresponding error terms alternate signs on the reference set is emphasized. The reason being that this structure would appear to be more fundamental than the residual value, as the direct algorithm above is more likely to make the necessary value adjustments rather than making a sign adjustment. However, we are currently investigating methods that attempt to take direct account of the sign during intermediate iterations.

In choosing a suitable starting point there are two subproblems to be considered. One is the choice of the reference set and the other is the choice of the corresponding parameters so that the approximating errors satisfies the alternating sign property.

We use classical theory to determine our reference set and we solve a suitable system of linear equations to satisfy the alternating sign property.

In an attempt to verify the significance of the conditions necessary to invoke the classical theory, we also determine a reference set that is uniformly distributed over the interval of approximation [a,b] whilst still satisfying an analogous system of linear equations.

We first summarize the required classical results.

Define

$$T_n(z) = \cos\left(n \cos^{-1}(z)\right), \quad -1 \le z \le 1 . \tag{7.1}$$

T_n, called a Chebychev polynomial, is an algebraic polynomial of degree n. It is obvious that the Chebychev polynomial T_n has the following properties.

Lemma 7.1

There are $n+1$ points

$$z_k = \cos\left(\frac{(k-1)\pi}{n}\right) \quad k = 1, 2, \ldots, n+1 \tag{7.2}$$

on the interval $[-1, 1]$ such that $T_n(z)$ achieves its maximum value 1 and minimum value -1, alternating with k.

We will also require the following

Theorem 7.2 [14]

Let f be a function defined on $[-1, 1]$ and ϕ be an approximation function of f in P^{n-1}, the space of all polynomials with the degree less than or equal to $n-1$, that is calculated via

$$\phi(z) = \sum_{i=1}^{n} \alpha_i \, \phi_i(z) , \tag{7.3}$$

with basis functions $\phi_i(z)$ where the α_i, are chosen to satisfy the linear system

$$f(z_i) - \phi(z_i) = (-1)^i \, \xi , \qquad i = 1, 2, \ldots, n+1 , \tag{7.4}$$

in the $n+1$ unknowns $\alpha_1 \cdots \alpha_n$ and ξ.

Furthermore, our reference set $\{z_i\}_{i=1}^{n+1}$ is given by (7.2). Then, if f is a polynomial of degree n, ϕ is the best minimax approximation from P^{n-1} to f.

For a general function, suitably scaled to have range [-1, 1], we are using the $n+1$ term Taylor's expansion as justification for using the Chebychev initial point.

In the numerical results that follow we use the terminology "the Chebychev initial point of degree n" to denote that point $(\alpha_1 \cdots \alpha_n)$ that satisfies (7.3) and (7.4) with reference set (7.2), suitably scaled over the interval [a, b].

We use the terminology, "the uniformly distributed initial point of degree n" to denote that point $(\alpha_1 \cdots \alpha_n)$ that satisfies (7.3) and (7.4) with reference set

$$z_i = a + \frac{b-a}{n}(i-1), \quad i = 1, \ldots, n+1. \tag{7.5}$$

8. Degeneracy

Degeneracy is a difficulty whose importance is gradually being recognized, and even for linear problems it cannot simply be waived away with the statement that a computer's finite precision automatically results in degeneracy being a theoretical rather than practical difficulty.

From the point of view of the linear Chebychev problem the degeneracy with which we are concerned is that of the η_i of (5.10) not being uniquely defined (or equivalently, the activities being linear dependent).

Two approaches that are suitable for overcoming the difficulties is that of perturbation and solving the optimality conditions explicitly. Our implementation took the former approach, details of which are given [7].

An example of the latter approach in the context of nonlinear l_1 problems is given by [10]. In the case of the linear Chebychev problem, one solves the bounded least squares problem in η (corresponding to 5.10) given by

$$\text{minimize} \quad ||h - \sum_{i=1}^{k} \eta_i \, c_{j_i}||_2$$

$$\text{subject to} \quad \eta_i \geq 0$$

or alternatively, one can use the l_1 formulation

$$\text{minimize} \quad ||h - \sum_{i=1}^{k} \eta_i \, c_{j_i}||_1$$

$$\text{subject to} \quad \eta_i \geq 0 \; .$$

In either case, if a solution with zero objective function exists, one is optimal for the penalty function, and if feasible, optimal to the original Chebychev problem. If not feasible, as before, μ will be reduced. Otherwise one finds a solution that satisfies the positivity constraints but for which $h \neq \sum_{i=1}^{k} \eta_i \, c_{j_i}$. In this case $\sum_{i=1}^{k} \eta_i \, c_{j_i} - h$ is a usable descent direction.

9. Numerical Results

Our first intention is to indicate that primal methods appear superior to dual methods on problems that do not correspond to discretizations of continuous function approximation problems (which we will term random problems).

Problem 1:

With this end in view we ran a series of random problems (1.3). More specifically we generated (pseudo-) random elements in the interval $(-100, +100)$ for A and b using the IMSL routine GGUBF. These random problems were solved by the Barrodale and Phillips routine [1] (BP) and the Bartels Golub implementation of Stiefel's method (BGS), [9], as the two best known representatives of the dual approach. As representative of the primal approach we will consider the algorithm of Section 5 with two different stepsize choices. The first corresponds to taking the minimum stepsize $\lambda^{(1)}$, giving the algorithm of [7] (BCC) and the second corresponds to taking the maximum stepsize consistent

with descent on the maximum absolute residual, that was first presented in [6] (BC).

These results are tabulated below.

Each of the methods can be programmed to require only $O(mn)$ work per iteration step. Thus numbers of iteration steps can be translated, roughly, into comparative running times. For this reason, and because of the disparate nature of the codes to which we had access (for example, the BP code uses $O(mn)$ work per iteration and BGS uses $O(n^3 + mn)$), we report the number of iteration steps rather than run times in our comparisons below and in subsequent results.

BP, BCC and BC codes had $x = 0$ for their initial point. BGS selects its own starting point by finding a nonsingular submatrix of $[A^T : e]$, where $e = [1, \ldots, 1]^T$.

$n = 2$

Number of Steps				
m	BP $x=0$	BC $x=0$	BCC $x=0$	BGS
50	4	2	3	4
100	5	2	3	6
150	6	2	3	8
200	6	2	3	6

$n = 6$

Number of Steps				
m	BP $x=0$	BC $x=0$	BCC $x=0$	BGS
50	14	6	7	14
100	14	6	7	9
150	17	8	9	17
200	18	7	8	14

$n = 10$

Number of Steps				
m	BP $x=0$	BC $x=0$	BCC $x=0$	BGS
50	19	12	13	20
100	29	13	14	19
150	30	14	15	25
200	24	13	14	24

We also include a random problem set with which we can compare the BC results using three initial starting points, $\alpha = 0$, the Chebychev initial point (C-points) and the uniform distributed initial point (D-points), as described in Section 7.

Problem 2:

Solve the non-data fitting problem

$$\alpha_1 + \sum_{j=2}^{n} \alpha_j \, \beta_{ji} \, z_i^{j-1} = \sqrt{1+z_i} * (4\beta_i - 2)$$

where

$$\beta_i, \beta_{ji} \, , \quad i = 1, 2 \, , \ldots , \, m \, , \quad j = 2 \, , \ldots , \, n \, ,$$

are random numbers in [0,1] and the z_i are chosen to be distinct points in [0,1], distributed uniformly

$n = 2$

Number of Steps				
m	BP	BC α=0	BC D-points	BC C-points
50	5	3	4	2
100	5	3	4	3
150	6	3	5	3
200	5	2	4	4

$n = 6$

Number of Steps				
m	BP	BC α=0	BC D-points	BC C-points
50	12	7	14	12
100	23	6	28	24
150	15	9	23	25
200	18	7	24	17
250	22	12	19	23

$n = 10$

Number of Steps				
m	BP	BC α=0	BC D-points	BC C-points
50	30	16	37	30
100	40	12	37	47
150	28	15	47	48
200	32	14	39	49
250	27	16	34	42

$n = 14$

Number of Steps				
m	BP	BC α=0	BC D-points	BC C-points
50	38	22	31	30
100	62	24	55	45
150	40	22	54	47
200	55	22	69	50
250	59	38	73	57

The results of Problem 1 make both BC and BCC (the primal methods) appear distinctly superior to (the dual methods) BGS and BP.

As the theory predicts, Problem 2 confirms that D-points and C-points are of no value for non-data approximation problems. Indeed, the results for these starting points are quite similar to those of Barrodale and Roberts. However, in all cases, BC with the zero initial point is noticeably superior.

We now wish to consider approximation problems. Consistent with the arguments of this paper, we expect that an "intelligent" choice of starting point would more than eradicate the believed superiority of dual methods in this case. Thus, in the case of BCC we consider three starting points, 0, C and D. As before, 0 corresponds to the initial point x=0, C corresponds to Chebychev initial points and D corresponds to the uniform distributed initial point.

Since the algorithm of Barrodale and Phillips is generally considered to be superior to Stiefel's algorithm, in what follows we report only the dual results corresponding to BP.

Problem 3:

Approximate $f(z) = e^z$, evaluated for $z = 0.0\ (0.1)\ 2.0$ (i.e., $m = 21$) by a polynomial of degree $n-1$.

$$p(z) = \xi_1 + \xi_2 z + \cdots + \xi_n z^{n-1} .$$

Let $x = [\xi_1, \xi_2, \ldots, \xi_n]$.

Number of Steps				
n	BP	BC $x=0$	BC D-point	BC C-point
4	4	8	4	4
6	6	12	10	6
8	16	16	14	9

Problem 4:

Approximate $f(z) = e^z$, evaluated for $z = 0.0\ (0.01)\ 2.0$ (i.e., $m = 201$) by a polynomial of degree $n-1$.

$$p(z) = \xi_1 + \xi_2 z + \cdots + \xi_n z^{n-1} .$$

Let $x = [\xi_1, \ldots, \xi_n]$.

Number of Steps				
n	BP	BC $x=0$	BC D-point	BC C-point
2	3	5	2	2
4	8	27	7	4
6	12	40	13	6
8	18	30	17	9

Problem 5:

Approximate $f(z) = \sin(z)\, e^{-z}$, evaluated for $z = 0.0\ (0.02)\ 4.0$ (i.e., $m = 201$) by a polynomial of degree $n-1$.

Number of Steps				
n	BP	BC $x=0$	BC D-points	BC C-points
2	5	5	4	4
4	11	24	7	5
6	17	22	26	16

We also include the Barrodale problems in [1]. Once again, the comparisons are made with results from the direct method of Bartels and Conn with $x=0$, the initial point produced by Chebychev point, the uniformly-distributed initial points, and the BP algorithm.

Problem 6:

Approximate following $f(z)$, evaluated for $z = 0\ (0.01)\ 1$ (i.e., $m = 101$) by a polynomial of degree $n-1$:

$$\phi(z) = \sum_{j=1}^{n} \alpha_j\, z^{j-1}\ .$$

Let $x = (\alpha_1, \ldots, \alpha_n)$.

1. $f(z) = \sqrt{1+z}$.

Number of Steps				
n	BP	BC $x=0$	BC D-points	BC C-points
2	3	8	2	2
3	6	7	4	3
4	8	17	6	4
5	9	16	8	5
6	12	23	13	6
7	14	18	12	8
8	16	29	13	14

2. $f(z) = \sin\left(\frac{\pi z}{2}\right)$.

Number of Steps				
n	BP	BC $x=0$	BC D-points	BC C-points
2	4	7	2	2
3	8	12	3	3
4	11	17	6	4
5	12	14	9	5
6	12	27	11	6
7	18	32	14	7
8	15	29	19	12

3. $f(z) = \log(1+z)$.

Number of Steps				
n	BP	BC $x=0$	BC D-points	BC C-points
2	3	10	2	2
3	7	10	3	3
4	7	20	5	4
5	12	16	8	6
6	13	36	9	6
7	15	20	14	9
8	15	33	22	10

4. $f(z) = \sinh(z)$.

Number of Steps				
n	BP	BC $x=0$	BC D-points	BC C-points
2	3	9	2	2
3	6	9	5	3
4	8	23	8	4
5	9	23	6	5
6	12	23	13	6
7	14	20	16	12
8	16	26	23	19

5. $f(z) = erf\ (z)$.

Number of Steps				
n	BP	BC x=0	BC D-points	BC C-points
2	5	8	2	2
3	6	12	4	3
4	9	20	5	4
5	10	13	16	10
6	14	31	15	6
7	12	22	12	10
8	17	25	20	9

6. $f(z) = e^{\frac{z^2}{2}}$.

Number of Steps				
n	BP	BC x=0	BC D-points	BC C-points
2	5	8	2	2
3	8	16	4	3
4	12	21	6	4
5	10	15	7	5
6	13	30	11	7
7	13	34	14	7
8	16	28	12	13

From the results tabulated above, we notice that the Chebychev initial point is indeed a better choice if the basis function space $L = < \phi_i\ ; 1 = 1\ , \ldots ,\ n >$ satisfies the Haar condition. [L satisfies the Haar condition in all the above examples.] Comparing the last column (C-points) with BP, we observe that BC algorithm performed much better than BP in most cases and a significant improvement over x=0 is achieved. If we compare the D-point column with BP and x=0, we note that in addition, the initial D-point has a significant improvement over x=0 although it is typically, as expected, inferior to the Chebychev initial point.

We now consider some data fitting problems for which either the Haar conditions are not satisfied or the function being approximated is not continuous.

Problem 7:

A. Approximate $f(z) = 1 + z + z^2 + z^3 + z^4 + g(z)$

$$g(z) = \begin{cases} 5 & \text{for } 0.94 \leq z \leq 1.0 \\ 0 & \text{for all others} \end{cases}$$

evaluated on the points $z = 0.0\ (0.02)\ 1.0$ using a polynomial of degree $n-1$.

Number of Steps				
n	BP	BC $x=0$	BC D-points	BC C-points
2	3	8	2	2
4	10	21	14	15
6	11	44	19	23
8	14	45	37	16

B. Approximate $f(z) = 1 + z + z^2 + z^3 + z^4 + g(z)$

$$g(z) = \begin{cases} 5 & \text{for } 0.5 \leq z \leq 0.8 \\ 0 & \text{for all others} \end{cases}$$

evaluated on the points $z = 0.0\ (0.02)\ 1.0$ using a polynomial of degree $n-1$.

Number of Steps				
n	BP	BC $x=0$	BC D-points	BC C-points
2	6	3	3	3
4	9	9	6	6
6	13	21	11	15
8	24	37	17	18

Problem 8:

A. Approximate the following functions evaluated for $z = 0.0\ (0.02)\ 1.0$ (i.e., $m = 51$) by the cubic spline

$$\sum_{j=1}^{4} (\xi_j \, z^{j-1} + \xi_{j+4} \max [(z - \gamma_j)^3, 0])$$

where γ_j are given by

$$\gamma_1 = 0.1,\ \gamma_2 = 0.2,\ \gamma_3 = 0.4,\ \gamma_4 = 0.7,\ f_1(z) = \sqrt{z},\ f_2(z) = \sqrt{1+z},$$
$$f_3(z) = \sin\left(\frac{\pi z}{2}\right),\ f_4(z) = \log(1+z)\ .$$

Number of Steps				
F	BP	BC $x=0$	BC D-points	BC C-points
f_1	12	33	25	28
f_2	13	26	18	21
f_3	13	24	19	19
f_4	12	19	16	20

B. Approximate the following functions evaluated for $z = 0.0\ (0.02)\ 1.0$ (i.e., $m = 51$) by a piecewise linear function:

$$\phi(z) = g_1(z) + g_2(z)\ ,$$
$$g_1(z) = \begin{cases} c_1 + m_1 z & \text{if } z \le \alpha\ ; \\ 0 & \text{otherwise} \end{cases}$$
$$g_2(z) = \begin{cases} \dfrac{(m_1 - m_2)}{2} + c_1 + m_2 z & \text{if } z \le \alpha\ ; \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha = 0.5,\ f_1(z) = z^2,\ f_2(z) = \sqrt{z},\ f_3(z) = \sin\left(\frac{\pi z}{2}\right),\ f_4(z) = \log(1+z)$

and c_1, m_1, m_2 are the variables, i.e. $x = (c_1, m_1, m_2)$.

Number of Steps				
F	BP	BC $x=0$	BC D-points	BC C-points
f_1	6	6	10	11
f_2	5	6	3	4
f_3	5	6	5	5
f_4	7	5	6	8

Problem 9:

Approximate $y = x + 2$ by

$$y = \alpha_0 + \alpha_1 x^2 + \alpha_2 x^4$$

on the interval $[-2, 2]$. Let $\alpha = (\alpha_0, \alpha_1, \alpha_2)$.

Number of Steps				
m	BP	BC $\alpha=0$	BC D-points	BC C-points
4	3	2	2	2
10	6	3	3	5
20	8	4	5	5
60	11	4	8	8
100	12	4	7	7

In Problem 7, the functions to be approximated are not continuous. The numerical results are inconclusive as to which method is preferable.

In Problems 8, and 9 the Haar condition doesn't hold for the basis space. For Problem 8 B), the BP algorithm indicated that the solution may not be unique while the BC algorithm found different solutions for different starting points. The residuals for both algorithms did not satisfy the alternating sign property at termination.

9. Further Remarks and Conclusions

It is interesting to note the connection between the linear l_1 and l_∞ problems.

Firstly, they are dual problems in the functional analysis sense. That is, they correspond to l_p and l_q normed spaces, where $\frac{1}{p} + \frac{1}{q} = 1$, in the limiting case $p = 1 \quad q = \infty$.

Consequently, all things being equal, one would expect to be able to solve both problems equally well — or at least, failing to do so, one might always solve one via its dual. In actual fact, until around 1972, the most effective algorithm was most probably that of Barrodale and Young [4], for the l_∞ problem, although l_1 and l_∞ algorithms were almost comparable. However, in 1974, the l_1 algorithm of Barrodale and Roberts [3], was clearly superior. In other words, at this time we were more proficient at solving the l_1 problem than the l_∞ problem. Arguably, the fact that the algorithm of Barrodale and Roberts was, in linear programming terms, not restricted to choosing a stepsize corresponding to the first feasible vertex, was a significant contributor to this superiority.

We now note that the penalty function approach enables us to incorporate a similar relaxation in the case of the l_∞ problem. This is best illustrated by looking at a simple example.

In the case of the l_1 problem — a typical line search along d for $F(x) = \sum_{i=1}^{n} |a_i^T x - b_i|$ might be as depicted in Figure 1

A standard linear programming approach (or that of Barrodale and Young, which is essentially a standard linear programming approach that incorporates some efficiencies because of the special structure of the l_1 problem) would choose the stepsize $\lambda = \lambda_1$. By contrast, the algorithm of Barrodale and Roberts would choose the stepsize $\lambda = \lambda_3$.

Analogously, if we were to consider the l_∞ problem with $F(x) = \max_{1 \le i \le m} |f_i(x)|$ we might have $F(x)$ as depicted in Figure 2

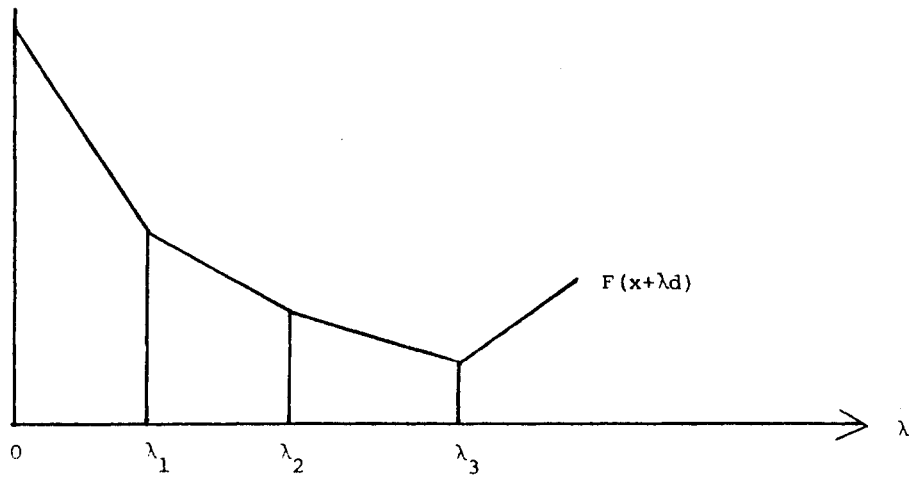


Figure 1

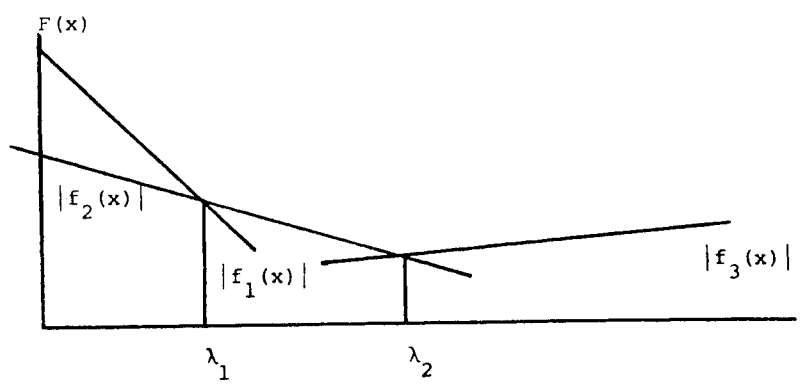


Figure 2

A method based on a standard linear programming primal formulation would take a stepsize $\lambda = \lambda_1$. However, consider the primal linear program solved via a penalty approach. At $\lambda = 0$, our objective ξ would have the value $|f_1(x)|$ and $\xi - f_1(x)$ or $\xi + f_1(x)$ would be active. Now past $\lambda = \lambda_1$, $\xi - |f_2(x)| \geq 0$ would become violated, but since $|f_2(x)|$ is decreasing for $\lambda > \lambda_1$ our penalty function would continue to decrease [it corresponds to $\xi - (\xi - |f_2(x)|) = |f_2(x)|$] and thus we would take $\lambda = \lambda_2$ for our stepsize.

With respect to the numerical results presented above, it is noteworthy that, for many of the examples presented, the results of the primal approach of Bartels and Conn is close to the best possible, in that normally one would expect to require at least $n+1$ iterations.

In conclusion, it has been observed that indirect (i.e. dual) methods perform better on data fitting problems. See for example [13] page 157, in addition to the earlier cited references. We have shown that the initial point generated by the algorithm of Barrodale and Phillips gives a trial approximation whose error terms possess the alternating sign property, if the Haar condition is satisfied for the basis space and the function to be approximated is continuous. It is the claim of this paper that this is the most significant aspect of the apparently superior performance of dual methods on data fitting problems.

We are able to generate equivalent starting points for primal methods, in an amount of work equivalent to solving a square linear system in the number of unknowns. Consequently it is worth giving some thought to the representation of the basis functions. For example, if one is approximating by a polynomial of degree n one could likely choose a Newton form of the type $1, (x-z_1), (x-z_1)(x-z_2), \ldots, \prod_{i=1}^{n} (x-z_i)$ for basis functions. One is then able to generate the starting point in $O(n^2)$ work, which compares favourably to the $O(mn)$ of a single iteration. We then demonstrate that, with such a choice of initial point, the primal methods are superior to dual methods in the case of data-fitting problems that satisfy the same Haar and continuity conditions.

In the case where we have a data fitting problem which does not satisfy either the Haar-condition or the continuity condition a choice of a good initial point is unclear and, as is to be expected, there are examples for which any of the considered algorithms might be superior.

References

1. Barrodale, I. and Phillips, C., "Algorithm 495; Solutions of an Overdetermined System of Linear Equations in the Chebychev Norm", ACM Trans. on Math. Software, 1 (1975) 3, pp. 264-2 .
2. Barrodale, I. and Phillips, C., "An Improved Algorithm for Discrete Chebychev Linear Approximation", Proc. 4th Manitoba Conf. on Numer. Math. U. of Manitoba, Winnipeg, Canada, 1974, pp. 177-190.
3. Barrodale, I and Roberts, F.D.K., "An Improved Algorithm for Discrete l_1 Linear Approximation", SIAM J. Num. Anal., 10 1972, pp. 839-848.
4. Barrodale, I. and Young, A., "Algorithms for best L_1 and L_∞ Linear Approximations on a Discrete Set", Numer. Math., 8 (1966), pp. 295-306.
5. Bartels, R.H. and Conn, A.R., "Linearly Constrained Discrete l_1 Problems", ACM Trans. on Math. Software, 6 (1980) 4, pp. 594-608.
6. Bartels, R.H. and Conn, A.R., "A Primal, Penalty Linear Programming Method for Solving Overdetermined Linear Systems in the L_∞ Sense", Tech. Rept. CORR 77/143 (1977), Dept. of Combinatorics and Optimization, Univ. of Waterloo, Waterloo, Ontario, Canada, N2L 3G1.
7. Bartels, R.H., Conn, A.R. and Charalambous, C., "On Cline's Direct Method for Solving Overdetermined Linear Systems in the l_∞ Sense", SIAM J. Numer. Anal., 15 (1978), pp. 255-270.
8. Bartels, R.H., Conn, A.R. and Sinclair, J.W., "Minimization Techniques for Piecewise Differentiable Functions: the l_1 Solution to an Overdetermined Linear System", SIAM J. Numer. Anal., 15 (1978), pp. 224-241.

9. Bartels, R.H., and Golub, G.H., "Stable Numerical Methods for Obtaining the Chebychev Solution to an Overdetermined System of Equations", Comm. ACM, 11 (1968), pp. 401-406.

10 Busovača, S., "Handling Degeneracy in a Nonlinear L_1 Algorithm", Tech. Rept. CS-85-34, Dept. of Computer Science, Univ. of Waterloo, Waterloo, Ontario N2L 3G1 (1985).

11. Conn, A.R., "Linear Programming via a Nondifferentiable Penalty Function", SIAM J. Numer. Anal., 13 (1976), pp. 145-154.

12. Joe, B. and Bartels, R.H., "An Exact Penalty Method for Constrained, Discrete, Linear l_∞ Data Fitting", SIAM J. Sci. Stat. Comput., 4 (1983), pp. 69-84.

13. Osborne, M.R., "Finite Algorithms in Optimization and Data Analysis", John Wiley (1985).

14. Powell, M.J.D., "Approximation Theory and Methods", Cambridge University Press (1981).

15. Stiefel, E.L., "Note on Jordan Elimination, Linear Programming and Tschebyscheff Approximation", SIAM J. Numer. Math., 2 (1960), pp. 1-17.

16. Watson, G.A., "Approximation Theory and Numerical Methods", John Wiley (1980).