EXISTENCE AND UNIQUENESS OF BEST APPROXIMANTS,  
WITH NUMERICAL APPLICATIONS

BY M. PLANITZ

DIVISION OF MATHEMATICS  
THAMES POLYTECHNIC  
LONDON SE18 6PF

THESIS SUBMITTED TO THE COUNCIL FOR NATIONAL ACADEMIC AWARDS IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

MAY 1985
ABSTRACT

EXISTENCE AND UNIQUENESS OF BEST APPROXIMANTS, WITH NUMERICAL APPLICATIONS

BY M. PLANITZ

Part I of the thesis deals with existence and uniqueness theorems. Strengthening a result due to J. Blatter, it is proved in chapter 3 that a normed linear space is complete if every closed, bounded, and convex set is proximinal. It is also shown, that in a semi-reflexive, locally convex, real linear metric space, every closed, bounded and convex set is proximinal. An example is constructed which proves that not every reflexive space is sequentially convex. In chapter 4, sequential and local uniform convexity are shown to be independent properties. It is proved that a sequentially convex space can be equivalently renormed with a locally uniformly convex norm. Various spaces are shown to be incapable of uniformly convex renorming. In chapter 5, a number of convexity properties and a class of convergence processes are generalized to metric spaces. It is shown that Clarkson's renorming technique can be extended to metrics and that each closed subset of a metric space can be made proximinal by introducing an equivalent metric. Chapter 6 provides a link between the abstract material of previous chapters and the numerical applications of part II. A unified theory is developed which comprises both discrete and continuous Chebyshev approximation.

Part II of the thesis contains numerical applications to the approximation of functions, data analysis, mathematical modelling, and optimization. Chapter 7 deals with a modified exchange algorithm for Chebyshev approximation. In chapter 8, closed formulae for linear Chebyshev approximants are derived. A computer approximation is obtained which is subject to restrictions on the number of non-zero bits in its binary representation. In chapter 9, an algorithm is developed which determines the $L_1$ solution set and selects a strictly best solution. Chapter 10 deals with the problem of balancing the input and output streams of mineral processing plants. A comparison is made of various existing methods and some new algorithms are suggested. In chapter 11, an integer programming algorithm is developed which allows the user to search for sub-optimal and alternative optimal solutions. Codings of the algorithms in chapters 7, 9, 10, and 11 are listed in the appendix of programs. A separate pocket at the end of the thesis contains two papers published in advance.
ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisors, Dr. D. C. Handscomb (University of Oxford) and Dr. M. R. Mehdi (University of London), for their guidance, understanding and encouragement. Thanks are also due to Prof. Dr. H.-P. Blatt (University of Eichstätt) for advice and encouragement during the initial stages of preparing this thesis, and to Edie McFall for expertly typing the manuscript.
CONTENTS

Chapter 0. Introduction ............................................. 1

I. EXISTENCE AND UNIQUENESS THEOREMS

Chapter 1. Strict and Uniform Convexity ................. 11
Chapter 2. Clarkson's Method of Equivalent Renorming 19
Chapter 3. Further Convexity Properties ................. 23
Chapter 4. The Relationship Between Sequential
Convexity and Local Uniform Convexity ... 34
Chapter 5. Convexity and Best Approximation
in Metric Spaces ........................................... 39
Chapter 6. Best Approximation in the $L_p$ Norms ...... 51

II. NUMERICAL APPLICATIONS

Chapter 7. A Modified Exchange Algorithm For
Best Chebyshev Approximation ................. 82
Chapter 8. Segmented Linear Chebyshev Approximation 90
Chapter 9. Strict Approximation in the $L_1$ Norm ..... 100
Chapter 10. An Application to Mineral Processing .... 114
Chapter 11. An Algorithm for Alternative Optimal
and Sub-Optimal Solutions in
Integer Programming ................................. 124

REFERENCES ...................................................... 139
APPENDIX OF PROGRAMS ........................................ 147
Chapter 0

Introduction

This thesis is divided into two main parts. Part I is largely theoretical and deals with abstract existence and uniqueness theorems. Part II contains various numerical applications to functional approximation, data analysis, mathematical modelling, and optimization. The theoretical part centres on the following problem of approximation theory: let \((E, \| \cdot \|)\) be a normed linear space and \(M\) a subset of \(E\). For each \(y \in E\) we define the distance of \(y\) from \(M\) by

\[
d(y, M) = \inf_{x \in M} \| y - x \|
\]

If there is an element \(x \in M\) so that \(d(y, M) = \| y - x \|\), then \(x\) is said to be a proximum in \(M\) of \(y\). \(M\) is called proximinal if every \(y \in E\) has a proximum \(x \in M\). The set of proxima is defined by the metric projection \(P_M : E \to 2^M\), with

\[
P_M(y) = \{ x \in M : \| y - x \| = d(y, M) \}.
\]

General conditions for the existence and uniqueness of proxima are of interest in many areas of pure and applied mathematics. For the special case that \(M\) is a subspace of \(E\), there is now a more or less complete theory. (For a good account of this theory, see the comprehensive book by I. Singer [22].) In recent years, research has therefore concentrated on the problem of non-
linear approximation. It soon became apparent that many results of the linear theory remain valid if $M$ is assumed to be a convex set, but not necessarily a subspace, and if certain additional conditions are satisfied. Thus we have the following generalization of a well known result about closed subspaces.

**Theorem 0.1** Let $M$ be a closed convex subset of a uniformly convex Banach space $B$. Then $M$ is a Chebyshev set, i.e. for each $y \in B - M$ there is exactly one proximum in $M$.

An introduction to the elements of linear and convex approximation is given in chapter 1 of this thesis. Chapter 2 deals with Clarkson's method of equivalent renorming. In his seminal paper of 1936, Clarkson [2] showed that each separable Banach space can be given a strictly convex norm which is equivalent to the original norm. Clarkson also showed that this result cannot be extended to uniformly convex norms. His proof consists of constructing a non-differentiable function of bounded variation from $[0,1]$ into $L_1$ (theorem 2.4). Since differentiability is a necessary condition for the existence of an equivalent, strictly convex norm, the space $L_1$ is not strictly or uniformly convex renormable. Using similar constructions, Clarkson drew the same conclusion for certain other spaces. More generally, we show in Chapter 4 that non-reflexive spaces cannot be given an equivalent norm which is uniformly or sequentially convex.
(theorem and corollary 4.5). Chapter 3 deals with convexity properties which lie between strict and uniform convexity, such as local uniform convexity and sequential convexity.

The former is due to Lovaglia [6], the latter was introduced by Ky Fan and Glicksberg [6], who pointed out that in theorem 0.1, uniform convexity can be replaced by sequential convexity but left the exact logical relationship between sequential and locally uniform convexity unresolved. An important contribution was then made by R.R. Phelps [17] and R.C. James who proved the following characterization theorem.

**Theorem 0.2** A Banach space $B$ is reflexive if and only if every closed convex subset of $B$ is proximinal.

J. Blatter [30] proved that a normed linear space is complete if every closed convex subset is proximinal. Since a reflexive space is always complete, this follows immediately from theorem 0.2. However, similar arguments can be used to establish a slightly stronger result: a normed linear space is complete if every closed, bounded and convex subset is proximinal (see theorem 3.5). It also follows that a sequentially convex Banach space is reflexive and strictly convex (Corollary 3.9). At the end of chapter 3 we construct a counterexample (example 3.1) which proves that, conversely, not every reflexive space is sequentially convex. Starting from theorem 0.2,
we then conclude in chapter 4 that sequential convexity and local uniform convexity are independent properties. In 1978, M.A. Smith [20] proved, that sequential convexity does not imply local uniform convexity. His counterexample consists of a norm which is strictly and sequentially convex, but not locally uniformly convex. In chapter 4 we construct a norm which achieves the same result, without being strictly convex (example 4.1). In the opposite direction we use a theorem by Kadetz (theorem 4.3) and the results of chapter 3 to complete the proof that sequential and local uniform convexity are independent properties. We also show in this chapter that a sequentially convex space can be given an equivalent norm which is uniformly convex.

The following fundamental existence theorem of approximation theory also applies to metric spaces.

**Theorem 0.3** If $M$ is a compact subset of a metric space, then $M$ is proximinal.

In order to generalize other existence and uniqueness theorems to metric spaces without a linear structure, it will be necessary to modify various convexity properties of normed space theory. Some results of this type appear in an article by Ahuja, Narang and Trehan [13]. Taking this paper as a starting point, we shall discuss certain weak convexity properties and investigate a problem suggested by I. Singer (see [22, p.378]): "It would be
interesting to study in metric spaces the problem of best approximation by elements of sets \( G \) belonging to certain special classes of sets, for instance convex sets \( G \subseteq E \) in the sense of K. Menger, i.e. having the property that for any distinct \( x, z \in E \) there exists a \( y \in E \) with \( x \neq y \neq z \) such that \( d(x, z) = d(x, y) + d(y, z) \).

We shall refer to Menger's convexity as semi-convexity and show that, if \( M \) is an approximatively compact semi-convex set in a strictly convex metric space, then \( M \) is a Chebyshev set (theorem 5.1). We shall also modify the concept of \( \tau \)-compactness, which is due to F. Deutsch [27] and L. P. Vlasov [26], and prove a suitable generalization of theorem 5.1. The fact that a compact or complete semi-convex set in a metric space is convex in the usual sense, can already be found in Menger's paper (see [23, p. 83 ff.]). It is therefore impossible to replace approximative compactness by compactness or completeness. Accordingly, theorems 5.3 and 5.4 deal with convex subsets.

We conclude Chapter 5 by showing how the geometric properties of a metric space can be improved by introducing an equivalent metric. Using the universality of the space of continuous functions we can apply Clarkson's ideas to certain metric spaces in order to guarantee uniqueness of best approximations. As for existence, it will be demonstrated that each closed subset can be made proximinal by introducing an equivalent, "almost" isometric metric.
This result has no analogue in a normed space.

Chapter 6 provides a link between the abstract material of part I and the numerical applications of subsequent chapters. The emphasis lies on best approximation in the $L_1$ and $L_\infty$ norms. It is shown that the minimax solution of a linear system can be regarded as a best continuous approximation on a compact metric space. A unified theory is developed which comprises both continuous and discrete approximation (theorems 6.5 - 6.8). The treatment of certain discrete $L_1$ results draws on material in the book by J.R. Rice [58, vol.I]. In particular, the proofs of lemmas 6.10 and 6.11 follow the line of reasoning used by Rice to establish the corresponding interval results. The alternation property of $L_p$ approximants is shown to extend to generalized polynomials (lemma 6.16, theorem 6.17).

Chapter 7 contains a modified exchange algorithm for best Chebyshev approximation. The basic idea is to obtain an initial reference for subsequent exchange iterations by considering certain features of the error vector of the $L_2$ approximant. A FORTRAN version of this algorithm, subroutine MINMAX, is included in the appendix of programs. MINMAX is about three times faster than the linear programming algorithm by Barrodale and Phillips.

Chapter 8 deals with some aspects of segmented Chebyshev approximation. Although the best polynomial
approximation to a given continuous function is known to exist, uniquely in fact, it remains an open question whether such a polynomial can be obtained by a general finite-step method. Such a method is feasible if the approximant is linear and if the continuous function satisfies certain additional conditions. It is shown that these sufficient conditions can be slightly weakened so as to satisfy only convexity and differentiability. The method is then applied to a problem of computer approximation, which imposes an upper bound on the number of non-zero bits in the binary representation of the coefficient of \( x \), in order to minimize the execution time for linear approximants. The remainder of Chapter 8 is concerned with segmented linear approximation to functions of two variables. While the usual arguments for the existence of polynomial approximants carry over from the single variable case, the uniqueness theory breaks down. However, it was shown by Collatz [38], that a linear best approximant is unique if the approximated function has continuous partial derivatives at all interior points of a closed, strictly convex set of the plane. It is shown that, if the approximated function satisfies certain convexity and differentiability conditions, then it is possible to generalize the single variable case and derive closed expressions for the coefficients and maximum error of a best linear approximant.

Chapter 9 is concerned with the problem of non-
unique linear approximants in the $L_1$ norm. Approximation packages generally supply only one $L_1$ solution and ignore alternative optima. At most, an exit code indicates that alternative optima may exist. As is detailed in chapter 6, the solutions form a two-dimensional convex set. In chapter 9, an algorithm is developed which determines this solution set and then proceeds to select from it a unique "best" of infinitely many best solutions. This is done by minimizing the $L_2$ norm of the error vector, with the parameters constrained to belong to the $L_1$ solution set. A FORTRAN coding of this algorithm is included in the appendix of programs (see the subroutines SOLVE and STRICT). A similar criterion, due to J.R.Rice, of choosing a "best" of all best Chebyshev approximants, is described in chapter 6 (see the remarks following theorem 6.5). Chapter 9 also includes a refinement of the usual linear programming technique for determining a best linear $L_1$ approximation. It consists of forcing the line through two interpolating points during the first two iterations. The interpolating points are chosen so that their $L_2$ errors $r_j, r_k$ are numerically minimal, with $\text{sgn}(r_j, r_k) \leq 0$.

Chapter 10 deals with the problem of balancing the input and output streams of a mineral processing plant. A comparison is made of various existing computational techniques, emphasizing microcomputer implementation. A new algorithm is developed and coded in BASIC (see the
program MINBAL in the appendix). The inconsistent systems arising from the material balance problem are traditionally solved by least squares methods. An adaptive package, incorporating other norms, is suggested and these norms are applied to a test problem.

An algorithm for alternative optimal and sub-optimal solutions in integer programming is developed in chapter 11. It is based on some elementary number theory and deals with the following problem: determine non-negative integers $x_1, \ldots, x_n$ such that $f(x) = c_1 x_1 + \ldots + c_n x_n = \min!$, subject to linear constraints of the form $A x \leq b$. There may also be secondary constraints of the type $||x|| = \min!$. Initially, it is assumed that the constrained minimum $c$ of $f$ is known. If the $c_i$ are non-negative, then $0 \leq x = D t + k \leq s$, where $D$ is triangular, the $t_i$ are arbitrary parameters, $k$ is constant, and $s_i = c/c_i$. Upper and lower bounds for $t$ define a superset of the feasible parameter set. Infeasible solutions are eliminated by a simple test for $A x \leq b$ and $x \geq 0$. An adaptive version of the algorithm is outlined which may be used as an alternative to standard integer programming packages.
I. EXISTENCE AND UNIQUENESS THEOREMS

We next show that if $X$ is a finite-dimensional sub-space of a normed space $V$, then $P_X(y)$ is non-empty.

Let $x \in X$ and $\|x\| \leq \|y\|$. Then $\|x + \|y\|^2 y\| = \|x\| + \|y\| = \|y\|$.

Now let $x \in X$ and $\|x\| > \|y\|$. Then $\|x - \|y\|^2 y\| = \|x\| - \|y\|$. If $\|x\| > \|y\|$, then $\|x - \|y\|^2 y\| < \|x\|$. Therefore, $\|x - \|y\|^2 y\|$ is the minimum distance from $x$ to $y$ and $P_X(y)$ is non-empty.

Proof. First note that the intersection of a family of convex sets is convex and that $P_X(y) = \{0\}$ whenever $x \in X$ and $y \in X$. Moreover, if $x_1 \leq x_2$ and $y_1 \leq y_2$, then $x_1 y_1 \leq x_2 y_2$. If $x_1 \geq x_2$ and $y_1 \geq y_2$, then $x_1 y_1 \geq x_2 y_2$. It follows that $P_X(y)$ is convex.
Chapter 1

Strict and Uniform Convexity

Let $E$ be a normed linear space with real or complex scalars, let $M$ be a subset of $E$ and $y \in E$. If

$$
\|y - x\| = d(y, M),
$$

then $x$ is called a **proximum**, best approximant, or element of best approximation, to $y$ in $M$. The set of all proxima to $y$ in $M$ will be denoted by $P_M(y)$. If $P_M(y)$ contains at least (at most) one proximum, $M$ will be referred to as a **proximinal** (semi-Chebyshev) set. If $P_M(y)$ is a singleton set for every $y \in E$, $M$ is said to be a **Chebyshev** set. A subset $K$ of $E$ will be called **convex** if $x, y \in K$ implies $\alpha x + \beta y \in K$ for all $\alpha, \beta > 0$ such that $\alpha + \beta = 1$.

**Theorem 1.1** If $M$ is a convex set in a normed space, then the set $P_M(y)$ is convex.

**Proof** First note that the intersection of convex sets is convex and that $P_M(y) = M \cap S$, where $S = \{x : \|x - y\| \leq d(x, M)\}$. Moreover, if $x_1, x_2 \in S$ and $0 \leq \alpha \leq 1$, then $\|x_1 - y\|, \|x_2 - y\| \leq d(y, M)$ and $\|\alpha x_1 + (1 - \alpha)x_2 - y\| = \|\alpha(x_1 - y) + (1 - \alpha)x_2 - y\| \leq d(y, M)$, i.e. $S$ is also convex. It follows that $P_M(y)$ is convex. //

We next show that if $M$ is a finite-dimensional subspace of a normed space $E$, then $P_M(y)$ is non-empty. Let $x \in M$ and $\|x\| > 2\|y\|$. Then $\|x - y\| \geq \|x\| - \|y\| > \|y\|$. 

- 11 -
d(y, M). It therefore suffices to consider the function 
\( \phi(x) = ||x-y|| \) on the set \( B = \{ x \in M : ||x|| \leq 2||y|| \} \). This 
function is continuous, since 
\[ |\phi(x_1) - \phi(x_2)| \leq \phi(x_1 - x_2) \leq ||x_1 - x_2||. \]
B is a closed and bounded subset of a finite dimensional space and therefore compact, which implies 
that \( \phi(x) \) assumes its minimum \( d(y, M) \) for some \( x_0 \in B \). We therefore have the following result.

**Theorem 1.2** If M is finite-dimensional, then M is
proximinal.

**Example 1.1** Consider the space \( (\mathbb{R}^2, || \cdot ||_1) \) and let 
\( M = \{(x_1, x_2) : x_1 = x_2 \} \). Then with \( y = (1,0) \), 
\( \inf_{x \in M} ||y - x||_1 = \inf_{x_1 \in \mathbb{R}} (|x_1 - 1| + |x_1|) = 1 \) and 
\( p_M(y) = \{(x_1, x_2) : 0 \leq x_1 \leq 1\} \). //

The example demonstrates that a proximum is not
necessarily unique. In order to guarantee uniqueness, we
impose a restriction on the norm of E:

**Definition 1.1** Let \( x, y \) be points in the normed linear
space E. We say that E, or more precisely the norm of E,
is strictly convex (or rotund) if the unit sphere
\( S = \{ x \in E : ||x|| = 1 \} \) contains no line segment, i.e.

\[ ||x|| = ||y|| = ||x+y||/2 = 1 \]

implies that \( x = y \).

Equivalently, \( ||x+y|| = ||x|| + ||y|| \) implies \( x = \alpha y, \alpha > 0 \).

**Example 1.2** The space \( E = C[0,1] \) is not strictly convex.
To see this, let \( 0 < c \leq 1 \) and define

\[ f_c(t) = \begin{cases} 
  t/c, & 0 \leq t \leq c \\
  1, & c \leq t \leq 1.
\end{cases} \]
Then \( \|f_c\|_\infty = 1 \). If \( 0 < c < d \leq 1 \) and 
\[
g = \lambda f_c + (1-\lambda)f_d, \quad 0 \leq \lambda \leq 1,
\]
then \( \|g\|_\infty \leq 1 \), and since \( g(t_0) = 1 \) for \( d \leq t_0 \leq 1 \), we actually have \( \|g\|_\infty = 1 \) for \( 0 \leq \lambda \leq 1 \). Thus the unit sphere of \( C \) contains the segment \([f_c, f_d] \) and every point on this segment is a point of minimum distance 1 from 0. //

**Theorem 1.3** If \( M \) is a finite-dimensional subspace of a strictly convex normed linear space \( E \), then \( M \) is a Chebyshev subspace.

**Proof** We only have to prove uniqueness. If \( x_1, x_2 \in M \) such that \( x_1 \neq x_2 \) and if 
\[
\|y-x_1\| = \|y-x_2\| = d(y,M),
\]
then 
\[
\|y-x_1 + y - x_2\| < 2d(y,M),
\]
since \( E \) is strictly convex. Hence 
\[
\|y - (x_1 + x_2)/2\| < d(y,M),
\]
contradicting the definition of \( x_1 \) and \( x_2 \). //

The last two theorems cannot be extended to infinite-dimensional spaces as the following example demonstrates.

**Example 1.3** (see Cheney [21,p.21]). Let \( s = (s_1, s_2, \ldots) \in c_0 \), the space of sequences which converge to zero, and define a norm on \( c_0 \) by \( \|s\| = \max |s_i| \). Then
M = \{ s \in c_0 : \sum_{i=1}^{\infty} 2^{-i}s_i = 0 \} is an infinite-dimensional subspace of \( c_0 \). Let \( t = (t_1, t_2, \ldots) \in c_0 - M \) and put \( \sum 2^{-i}t_i = \delta \).

Then \( \delta \neq 0 \) and the sequences \( u^{(1)} = (-2/1)(\delta, 0, 0, 0, \ldots) + t \),

\( u^{(2)} = (-4/3)(\delta, \delta, 0, 0, \ldots) + t \),

\( u^{(3)} = (-8/7)(\delta, \delta, \delta, 0, \ldots) + t, \ldots \)

all lie in \( M \). Moreover,

\[ ||u^{(n)} - t|| = 2^n|\delta|/(2^n - 1) \to |\delta|. \]

For any \( v \in P_M(t) \), \( ||v-t|| \leq |\delta| \). If \( N \) is now chosen so that \( |v_i - t_i| < |\delta|/2 \) for all \( i \geq N \), then

\[
|\sum 2^{-i}t_i| \leq |\sum 2^{-i}(t_i - v_i)| \leq |\sum 2^{-i}t_i - v_i| \\
\leq |\delta| \sum_{i=1}^{N-1} 2^{-i} + \left| \frac{|\delta|}{2} \right| \sum_{i \geq N} 2^{-i} < |\delta|,
\]

which is a contradiction. Hence \( P_M(t) = \emptyset \). //

It is interesting to note where the proof of the theorem 1.2 breaks down if \( M \) is infinite-dimensional. Consider the subspace

\[ M = \{ s \in l_2 : s = (0, s_2, s_3, \ldots) \}. \]

If \( B = M \cap \{ s : ||s||_2 = 1 \} \), then \( B \) is a closed and bounded subset of \( M \). But it is easy to see that \( B \) is not compact, by noting that the sequence \( s_1 = (0,1,0,\ldots), s_2 = (0,0,1,0,\ldots), \ldots \) has no convergent subsequence since \( ||s_i - s_j|| = \sqrt{2} \) for \( i \neq j \).

In order to extend theorem 1.3 to infinite-dimensional subspaces we require the completeness of \( E \) and a stronger
form of convexity.

**Definition 1.2** (Clarkson [2]). Let $E$ be a normed linear space. Then $E$ is called **uniformly convex** if for all $\varepsilon (0 < \varepsilon \leq 2)$ there is a $\delta (\varepsilon) > 0$ so that the conditions

$$
\|x\| = \|y\| = 1 \text{ and } \|x + y\|/2 > 1 - \delta \quad (x, y \in E)
$$

imply

$$
\|x - y\| < \varepsilon.
$$

**Theorem 1.4** Let $B$ be a uniformly convex Banach space.

If $M$ is a closed convex subset of $B$ (in particular, a closed subspace) then $M$ is a Chebyshev set.

**Proof** Let $y \in B - M$ and assume w.l.o.g. that $y = 0$. Put

$$
\inf_{x \in M} \|x\| = \alpha. \quad \text{Since } M \text{ is closed, } \alpha > 0 \text{ and there is a sequence } (x_n) \text{ in } M \text{ such that } \|x_n\| \to \alpha. \quad \text{Setting}
$$

$$
u_n = x_n/\alpha, \quad \text{we have } \|u_n\| \to 1. \quad \text{For a given } \varepsilon > 0 \text{ we now choose } \delta(\varepsilon) > 0 \text{ according to definition 1.2. Next let}
$$

$$
\|u_n\| - 1 < \delta \text{ for } n \geq N, \text{ say, and define } v_n = u_n/\|u_n\|.
$$

Then $\|v_n\| = \|v_m\| = 1$. Since $M$ is convex,

$$
\|u_n + u_m\| \geq 2. \quad \text{We therefore have } \|v_n + v_m\|/2 \geq \|u_n + u_m\| - (\|u_n\| - 1)/2 - (\|u_m\| - 1)/2 > 1 - \delta.
$$

It now follows from the uniform convexity of $B$ that $\|v_n - v_m\| < \varepsilon$, i.e. $(v_n)$ is a Cauchy sequence. Since $B$ is a Banach space, there exists a $v \in B$ such that $v_n \to v$. Moreover,

$$
\|u_n - v\| \leq \|u_n - v_n\| + \|v_n - v\|
$$
\[ \leq \|u_n\|(1 - \|u_n\|^{-1}) + \|v_n - v\|, \]

i.e. \( u_n = x_n/\alpha + v \) and \( x_n + \alpha v \).

But \( \alpha v \in M \), since \( M \) is closed. The uniqueness part now follows from the fact that a uniformly convex space is strictly convex (see the next theorem).

**Theorem 1.5** If \( E \) is a uniformly convex normed linear space, then \( E \) is strictly convex. The converse holds if \( E \) is finite-dimensional.

**Proof** If \( E \) is uniformly convex and \( \|x\| = \|y\| = \|x + y\|/2 = 1 \), then \( \|x - y\| < \varepsilon \) for any \( \varepsilon > 0 \).

It follows that \( x = y \).

Conversely, suppose \( E \) is finite-dimensional and strictly convex. For a given \( \varepsilon > 0 \) define the set

\[ S = \{ (x, y) \in E \times E : \|x\| = \|y\| = 1 \& \|x - y\| \geq \varepsilon \}. \]

Clearly, \( S \) is compact. Define a function \( f \) by

\[ f(x, y) = 1 - \|x + y\|/2. \]

Then \( f \) is continuous. Moreover \( f(x, y) > 0 \), since \( E \) is strictly convex. Hence there is a \( \delta > 0 \) so that

\[ \inf_{x, y \in S} f(x, y) = \delta. \]

For \( \|x\| = \|y\| = 1 \), it therefore follows from \( \|x - y\| \geq \varepsilon \) that \( 1 - \|x + y\|/2 \geq \delta \), i.e. \( \|x + y\|/2 > 1 - \delta \) implies \( \|x - y\| < \varepsilon \). \(/

The next theorem is due to Clarkson [2].

**Theorem 1.6** A Hilbert space is uniformly convex.

**Proof** Put \( \|x\| = \|y\| = 1 \) in the parallelogram identity.
Then
\[ \|x + y\|^2 + \|x - y\|^2 = 2\|x\|^2 + 2\|y\|^2. \]
It follows that if \( \|x + y\| \to 2 \), then
\[ \|x - y\|^2 = 4 - \|x + y\|^2 = 4(1 - \|x + y\|^2/4) \to 0. // \]
Clarkson showed in the same paper that the spaces \( L_p \) and \( l_p \)
(\( 1 < p < \infty \)) are also uniformly convex. His proof is based
on the following result.

**Lemma** For the spaces \( L_p \) and \( l_p \), with \( p \geq 2 \), we have
\begin{align*}
(i) \quad \|x + y\|^p + \|x - y\|^p & \leq 2^{p-1}(\|x\|^p + \|y\|^p) \\
(ii) \quad 2(\|x\|^p + \|y\|^p)^{q-1} & \leq \|x + y\|^q + \|x - y\|^q,
\end{align*}
where \( q = p/(p-1) \). For \( 1 < p < 2 \), these inequalities
hold in the reverse sense.

For a proof of the lemma, see Hewitt and Stromberg [3], p. 225.
To prove uniform convexity for \( p > 2 \), put \( \|x\| = \|y\| = 1 \)
in (i). Then
\[ \|x + y\|^p + \|x - y\|^p \leq 2^p \text{ and } \|x - y\|^p \leq 2^p(1 - \|x + y\|^p/2^p) \to 0 \text{ as } \|x + y\|/2 \to 1. \]
For \( 1 < p < 2 \), reverse
the sense of inequality (ii). We therefore have

**Theorem 1.7** (Clarkson). The spaces \( L_p \) and \( l_p \) (\( 1 < p < \infty \))
are uniformly convex.

Theorem 1.7 is not true for \( p=1 \) or \( \infty \). The spaces
concerned also lack the weaker property of strict convexity.
To prove this for \( L_1[0,1] \), say, put \( f=2x, g=2-2x \). Then
\[ \|f\| = \|g\| = \|f + g\|/2, \text{ but } f \neq g. \]
By choosing \( f=x, \)
\( g=1 \) we can show that \( L_\infty[0,1] \) is not strictly convex.
The same result is established for \( l_1 \) by considering the
sequences \( s = (1/2, 0, 1/4, 0, 1/8, 0, \ldots) \),
\( t = (0, 1/2, 0, 1/4, 0, \ldots) \) and for \( \ell_\infty \) by choosing
\( s = (1, 1/2, 1/4, \ldots) \) and \( t = (1, 1, 1, \ldots) \).
Chapter 2

Clarkson's Method of Equivalent Renorming

In his paper on uniformly convex spaces, Clarkson [2] considered the following problem: given a Banach space 
\((B, \|\cdot\|)\), is there an equivalent norm \(\|\cdot\|'\) which satisfies a certain convexity property such as strict convexity. (Recall that two norms \(\|\cdot\|, \|\cdot\|'\) are equivalent if there exist positive constants \(k, K\) such that 
k\|x\| \leq \|x\|' \leq K\|x\| \text{ for all } x \in B.) Clarkson found that any separable Banach space can be given an equivalent strictly convex norm.

**Theorem 2.1 (Clarkson).** If \((B, \|\cdot\|)\) is a separable Banach space, then there exists a strictly convex norm \(\|\cdot\|'\) which is equivalent to \(\|\cdot\|\).

The sequence space \(\ell_1(\|\cdot\|_1)\), the space of integrable functions \((L_1[0,1], \|\cdot\|_1)\) and the space of continuous functions \((C[0,1], \|\cdot\|_\infty)\) are separable and can be renormed in this way. We first prove the theorem for \(C[0,1]\) and then apply a result due to Banach and Mazur, which will be stated here without proof.

**Theorem 2.2 (Banach and Mazur).** If \((E, \|\cdot\|_E)\) is a separable normed linear space, then \(E\) is isometric to a closed linear manifold of \(C[0,1]\).

**Proof of theorem 2.1** Let \((x_n)\) be a sequence which is dense in \([0,1]\) and define a sequence \((F_n)\) of bounded linear
functionals by $F_n(f) = f(x_n)$ for all $f \in C[0,1]$. It is easy to see that if $F_n(f) = 0$ for $n = 1,2,3,\ldots$, then $f = 0$. Now let

$$||f||_C = (||f||_\infty^2 + \sum_{n=1}^{\infty} 2^{-2n} |F_n(f)|^2)^{\frac{1}{2}}.$$ 

To see that $|| \cdot ||_C$ satisfies the triangle inequality, note that

$$||f + g||_C = (||f + g||_\infty^2 + \sum_{n=1}^{\infty} 2^{-2n} |F_n(f) + F_n(g)|^2)^{\frac{1}{2}} \leq$$

$$||f||_\infty^2 + 2||f||_\infty^2 ||g||_\infty + ||g||_\infty^2 +$$

$$+ \sum_{n=1}^{\infty} 2^{-2n} |F_n(f) + F_n(g)|^2$$

$$< ||f||_\infty^2 + 2||f||_\infty^2 ||g||_\infty + \sum_{n=1}^{\infty} 2^{-2n} |F_n(f)|^2 +$$

$$+ 2(||f||_\infty^2 + \sum_{n=1}^{\infty} 2^{-2n} |F_n(f)|^2)^{\frac{1}{2}} (||g||_\infty^2 + \sum_{n=1}^{\infty} 2^{-2n} |F_n(g)|^2)^{\frac{1}{2}}$$

$$= ||f||_C + ||g||_C. \quad (2.1)$$

Clearly, the remaining axioms of a norm are also satisfied. Since

$$||f||_\infty \leq ||f||_C \leq (||f||_\infty^2 + ||f||_\infty^2 \sum_{n=1}^{\infty} 2^{-2n})^{\frac{1}{2}} = (2/3) ||f||_\infty,$$

the norms $|| \cdot ||_\infty$ and $|| \cdot ||_C$ are equivalent. We see from (2.1) that

$$||f + g||_C = ||f||_C + ||g||_C \quad (2.2)$$

implies

$$||f + g||_\infty = ||f||_\infty + ||g||_\infty.$$
Thus if \( f, g \neq 0 \) are functions in \( C[0,1] \) which satisfy equation (2.2), we can write

\[
\left( \| f \|_\infty + \| g \|_\infty \right)^2 + \sum 2^{-2n} |F_n(f) + F_n(g)|^2 \frac{1}{4} = \\
\left( \| f \|_\infty^2 + \sum 2^{-2n} |F_n(f)|^2 \right) + \left( \| g \|_\infty^2 + \sum 2^{-2n} |F_n(g)|^2 \right) \frac{1}{2}.
\]

It follows from the equality condition of the Cauchy-Schwarz inequality that there is a positive number \( k \) so that \( kF_n(f) = F_n(g) \), i.e. \( kf = g \). This concludes the proof that \( \| \cdot \|_C \) is strictly convex. By theorem 2.2, there exists an isometry \( T: E \rightarrow C[0,1] \), with \( \| x \|_E = \| T(x) \|_\infty \) for all \( x \in E \). If we now define a new norm on \( E \) by

\[
\| x \|_E' = \| T(x) \|_C,
\]

then \( \| \cdot \|_E \) and \( \| \cdot \|_E' \) are equivalent. If

\[
\| x \|_E' = \| y \|_E' = \| x + y \|_E' / 2 = 1,
\]

then

\[
\| T(x) \|_C = \| T(y) \|_C = \| T(x + y) \|_C / 2 = 1,
\]

and since \( \| \cdot \|_C \) is strictly convex, it follows that \( T(x) = T(y) \). Using the fact that an isometry is injective, we deduce that \( x = y \), i.e. \( \| \cdot \|_E' \) is strictly convex. //

Theorem 2.1 was strengthened by Kadetz [5], who proved that a separable Banach space can be given an equivalent norm which is locally uniformly convex. (For a definition of local uniform convexity, see chapter 3). A number of negative results in Clarkson's paper demonstrate that the renorming technique cannot be extended to uniform convexity. The argument is based on the following theorem which will be stated without proof.
Theorem 2.3 (Clarkson). Let $F$ be a function of bounded variation from a Euclidean space to a Banach space which can be given an equivalent strictly convex norm. Then $F$ is differentiable almost everywhere.

Consider the function $F: [0,1] \rightarrow L_1[0,1]$: 

$$t \mapsto \phi_t(s) = \begin{cases} 1, & 0 \leq s \leq t \\ 0, & t < s \leq 1. \end{cases}$$

Let $\frac{\delta F}{\delta t} = (F(t + \delta t) - F(t))/\delta t$, $\delta t \neq 0$. Then 

$$\|\frac{\delta F}{\delta t}\|_{L_1} = \| (\phi_{t+\delta t} - \phi_t)/\delta t \|_{L_1} = 1,$$

i.e. $F$ is of bounded variation. On the other hand, $F$ is nowhere differentiable on $[0,1]$. We therefore have

Theorem 2.4 The space $L_1[0,1]$ cannot be renormed so as to be uniformly convex.

Using similar arguments, Clarkson drew the same conclusion for the spaces $L_\infty$ (bounded, measurable functions), $C$, $\ell_\infty$ (bounded sequences) and $c$ (convergent sequences).

We shall see in the next section that theorem 2.4 holds, in fact, for all non-reflexive spaces. We finally conclude from this discussion that the converse of theorem 1.5 cannot be extended to infinite dimensional spaces:

Example 2.1 Let $(x_n)$ be a sequence which is dense in $[0,1]$ and define a sequence $F_n$ by $F_n(f) = f(x_n)$ for all $f \in C[0,1]$ as in the proof of theorem 2.1. Then

$$\|f\|_C = (\|f\|_\infty^2 + \sum_{n=1}^{\infty} 2^{-2n} |F_n(f)|^2)^{\frac{1}{2}}$$

is a norm which is strictly, but not uniformly convex. //
Chapter 3
Further Convexity Properties

The convexity properties discussed in this section are stronger than strict convexity and weaker than uniform convexity. The first definition goes back to Lovaglia [9]:

Definition 3.1 A normed linear space is called **locally uniformly convex** if \( x \in \mathcal{E}, \|x\| = 1 \) and \( \epsilon > 0 \) implies there exists \( \delta(\epsilon, x) > 0 \) such that \( \|x - y\| < \epsilon \) if \( \|y\| = 1 \) and \( \|x + y\|/2 > 1 - \delta \).

Theorem 3.1 Uniform convexity implies local uniform convexity which in turn implies strict convexity. In a finite-dimensional space all three are equivalent.

**Proof** The first implication follows immediately from the definition. Now suppose \( \|a\| = \|b\| = \|a + b\|/2 = 1 \) and \( a \neq b \). Take \( \epsilon = \|a - b\| \). Since the space is locally uniformly convex there exists a \( \delta(\epsilon, a) > 0 \) so that if \( \|y\| = 1, \|a + y\|/2 > 1 - \delta \), then \( \|a - y\| < \epsilon \). But \( \|a + b\|/2 = 1 > 1 - \delta \) and \( \|b\| = 1 \). Therefore \( \|a - b\| < \epsilon \), contradiction. It follows that the space is strictly convex. The equivalence of all three properties in the finite-dimensional case follows from theorem 1.5. //

We shall later see that local uniform convexity is not sufficient to ensure proximinality of closed convex sets. But Lovaglia found a relationship between differentiability of the norm and local uniform convexity. Thus, if the dual space \( \mathcal{B}^* \) is locally uniformly convex, then the norm in
the Banach space $B$ is strongly differentiable, i.e.
\[ \lim_{h \to 0} \frac{\|x_0 + hx\| - \|x_0\|}{h} \]
exists uniformly on $\|x\| \leq 1$. Moreover, if $B$ is locally uniformly convex and linear functionals attain their maximum on the unit sphere in $B$, then the norm in $B^*$ is strongly differentiable.

Another important convexity property is due to Fan and Glicksberg [6]:

**Definition 3.2** Let $K$ be a convex set. If $(x_n)$ is a sequence in $K$ such that
\[ \lim_{n \to \infty} \|x_n\| = \inf_{x \in K} \|x\|, \]
then it is called a minimizing sequence for $K$. A normed linear space is said to be sequentially convex if every minimizing sequence is a Cauchy sequence.

The relationship between sequential convexity and locally uniform convexity was not fully clarified in [6]. We shall see in Chapter 4 that the two properties are in fact independent. Since, a fortiori, sequential convexity does not imply uniform convexity, it follows that the next result represents a strengthening of theorem 1.4.

**Theorem 3.2** (Fan and Glicksberg). If $M$ is a closed convex subset of a Banach space $B$ and if $B$ is sequentially convex, then $M$ is a Chebyshev set.

**Proof** As in the proof of theorem 1.4 we put $\inf_{x \in M} \|x\| = \alpha$. Then $\alpha > 0$ and there exists a minimizing sequence $(x_n)$ in $M$. 

- 24 -
Since $B$ is sequentially convex, $(x_n)$ is a Cauchy sequence. But $B$ is complete and $M$ is closed. Hence $x_n \to x \in M$. To prove uniqueness, suppose $||x|| = ||y|| = \alpha$ for $x, y \in M$. Then $(x, y, x, y, \ldots)$ is a minimizing sequence and therefore a Cauchy sequence, i.e. $x = y$. //

Further to the sufficient conditions of theorems 1.4 and 3.2 we next state necessary and sufficient conditions for the proximality of all closed, convex subsets of a Banach space.

**Lemma 3.3** Let $E$ be a normed linear space, $H$ a hyperplane given by $f(x) = \alpha$, where $f \in E^*$, i.e. $f$ is a continuous linear functional on $E$, and let $y \in E$. Then
$$d(y, H) = \frac{|f(y) - \alpha|}{||f||}.$$

**Proof** Since $||y-x|| > \frac{|f(y-x)|}{||f||} = \frac{|f(y) - \alpha|}{||f||}$,
$$d(y, H) \geq \frac{|f(y) - \alpha|}{||f||}. \text{ Now let } 0 < \varepsilon < ||f||. \text{ By the definition of } ||f|| \text{ there exists } z \in E \text{ such that } |f(z)| > (||f|| - \varepsilon)||z||. \text{ Multiplying this inequality by } |f(y) - \alpha|/|f(z)|, \text{ we obtain } |f(y) - \alpha| > (||f|| - \varepsilon)||z|| \times |f(y) - \alpha|/|f(z)|. \text{ Now put } x = y - ((f(y) - \alpha)/f(z))z. \text{ Then } x \in H \text{ and the inequality becomes } ||y - x|| < |f(y) - \alpha|/ (||f|| - \varepsilon). \text{ Since } \varepsilon > 0 \text{ is arbitrary small, we also have } d(y, H) \leq |f(y) - \alpha|/||f||.$$

**Theorem 3.4** A Banach space $B$ is reflexive if and only if each closed convex subset of $B$ is proximinal.

**Proof** First assume that $B$ is not reflexive. Then $B$ has a non-reflexive, closed subspace $M$. By a well-known
theorem of James [10], a Banach space is reflexive if and only if each continuous linear functional attains its supremum on the unit sphere of every closed linear subspace. Let

$$S = \{ x \in B : \| x \| = 1 \} \text{ and } S^*_{M} = \{ f \in M^* : \sup_{S \cap M} |f(x)| = 1 \}.$$

Then there exists $F \in S^*_{M}$ such that $F^{-1}(1)$ does not meet $S \cap M$. Clearly $F^{-1}(1)$ is a closed convex subset of $B$.

We show that $F^{-1}(1)$ is not proximinal. By lemma 3.3, the distance $d$ from the origin to the hyperplane $F^{-1}(1)$ is given by

$$d = |F(0) - 1|/\|F\| = 1.$$ 

But this distance is not achieved by any element of $F^{-1}(1)$ since $F^{-1}(1)$ does not intersect the set $S \cap M$. It follows that $F^{-1}(1)$ is not proximinal.

We use the well-known result that reflexivity is equivalent to weak compactness of the unit ball $U = \{ x \in B : \| x \| \leq 1 \}$, see Day [1, p.69]. Let $M$ be a closed convex set and $y \in B \sim M$. Define a sequence $(B_n)$ of balls with centre $y$ and radius $d(y, M) + n^{-1}$. Then $(B_n \cap M)$ is a decreasing sequence of non-empty, weakly compact, convex subsets of $M$. By a theorem of Smulian (see Dunford and Schwartz [18, p.433]) there is therefore an element $z \in \cap (B_n \cap M)$. It is easy to see that $z \in M$ and $\|z - y\| = d(y, M)$. //

J. Blatter [30] proved that if $X$ is a normed linear space in which every closed convex subset is proximinal,
then \( X \) is complete. Since a reflexive space is always complete, Blatter's result immediately follows from theorem 3.4. We can, however, use his line of reasoning to obtain a stronger result.

**Theorem 3.5** If \( X \) is a normed linear space in which every closed, bounded and convex subset is proximinal, then \( X \) is complete.

**Proof** We prove the contrapositive. Suppose \( X \) is not complete and let \( \hat{X} \) be the completion of \( X \). Then 
\[
(\hat{X})^* = X^* \quad \text{(see Koethe [28, p. 261])}
\]
If \( \hat{y} \in \hat{X} \sim X \) and 
\[
||\hat{y}|| = d, \text{ then } \hat{x} = \hat{y}/d \in \hat{X} \sim X \text{ and } ||\hat{x}|| = 1.
\]
By the Hahn-Banach theorem there is an \( f \in X^* \) such that 
\[
||f|| = 1
\]
and \( f(\hat{x}) = ||\hat{x}|| = 1 \). Let \( (z_n) \) be a sequence in \( X \) such that 
\[
z_n \to \hat{x}. \text{ Then } f(z_n) + f(\hat{x}) = 1. \text{ Putting } f(z_n) = \delta_n, \text{ it is easy to see that } x_n = (1+1/n)z_n/\delta_n \to \hat{x} \text{ and } f(x_n) = 1+1/n.
\]

Let \( M_1 = H(x_1, x_2, \ldots) \), the convex hull of the sequence \( (x_n) \). Then \( \bar{M}_1 \) is a closed, bounded and convex set. We show that \( \bar{M}_1 \) is not proximinal. Note that, if \( x \in M_1 \), then 
\[
x = \sum \theta_i x_i, \quad \theta_i \geq 0, \sum \theta_i = 1, \quad \text{and}
\]
\[
f(x) = f(\sum \theta_i x_i) = \sum \theta_i f(x_i) = \sum \theta_i (1+1/i) > 1.
\]
Moreover, if \( x \in \bar{M}_1 \), then \( f(x) \geq 1 \) by the continuity of \( f \). Next note that
\[
1 \leq |f(x)| \leq ||f|| ||x|| = ||x|| \quad \text{for all } x \in M_1
\]
and that
\[
\lim ||x_n|| = ||\hat{x}|| = 1.
\]
Hence \( d(0, M_1) = \inf_{x \in \bar{M}_1} ||x|| = 1 \).
If we can show that $f(x) > 1$ for all $x \in \bar{M}_1$, then
\[ 1 < |f(x)| \leq ||x||, \]
i.e. $\bar{M}_1$ is not proximinal. Suppose, to the contrary, there exists some $x_1 \in \bar{M}_1$ such that $f(x_1) = 1$. Define $M_k = H(x_k, x_{k+1}, \ldots)$. By choosing $K$ sufficiently large we can ensure that $x_1 \notin \bar{M}_K$. Now put $P = H(x_1, \ldots, x_{K-1})$ and $Q = M_K$. If $(y_n)$ is any sequence in $M_1$ such that $y_n \to x_1$, then
\[ y_n = \alpha_n p_n + \beta_n q_n \]
for some $\alpha_n, \beta_n \geq 0$, $\alpha_n + \beta_n = 1$, $p_n \in P$, $q_n \in Q$.
If $p \in P$, then
\[ f(p) = f(\sum \theta_i x_i) = \sum \theta_i f(x_i) = \sum \theta_i (1 + 1/i) \geq 1 + 1/(K-1) \]
for $\theta_i \geq 0$, $\sum \theta_i = 1$. Similarly, if $q \in Q$, then
\[ f(q) = f(\sum \theta_i' x_i) > 1. \]
Hence
\[ f(y_n) = \alpha_n f(p_n) + \beta_n f(q_n) > \alpha_n (1 + 1/(K-1)) + \beta_n \]
\[ = 1 + \alpha_n/(K-1). \]
But $f(y_n) + f(x_1) = 1$. It follows that $\alpha_n \to 0$ and $\beta_n \to 1$. Moreover, $\lim(\alpha_n p_n) = 0$ since $P$ is a bounded set. Hence
\[ \lim(\alpha_n p_n + \beta_n q_n) = \lim q_n = x_1, \]
where $q_n \in Q = M_K$, i.e. $x_1 \in \bar{M}_K$, which contradicts the definition of $K$. //

The 'only if' part of the proof of theorem 3.4 is essentially due to M.M.Day [19] and was first published in 1941. The 'if' part is outlined in a paper by R.R.Phelps [17] who attributes it to R.C.James. We can exploit the concept of weak compactness in a more general setting.
Definition 3.3 A linear topological space $E$ is said to be \textbf{locally convex} if for each $a \in E$ and for each neighborhood $N(a)$ of $a$, there is a convex neighborhood $M(a)$ such that $a \in M \subset N$. The dual $E^*$ and bidual $E^{**}$ of $E$ are defined as usual. If $E = E^{**}$ then $E$ is called \textbf{semi-reflexive}. (A semi-reflexive Banach space is reflexive).

The 'only if' part of theorem 3.4 depends on Smulian's characterization of weak compactness in Banach spaces, which does not carry over to general locally convex spaces. However, one half of Smulian's result can be generalized for our purposes. We require two theorems from convexity theory.

Definition 3.4 Two convex sets $A$ and $B$ are said to be \textbf{strongly separated} by a hyperplane $f(x) = \alpha$ if for some $\varepsilon > 0$ and for all $a \in A$, $b \in B$

$$f(a) \leq \alpha - \varepsilon < \alpha + \varepsilon \leq f(b).$$

Theorem 3.6 Two convex sets $A$ and $B$ in a locally convex space $X$ can be strongly separated by a closed hyperplane if and only if $0 \notin \overline{B-A}$.

For a proof see Holmes [29,p.64]. The proof of the next theorem is an adaptation of arguments used in [29,p.146].

Theorem 3.7 If a convex subset $M$ of a real locally convex linear space $X$ is $w$-compact, then every decreasing sequence of non-empty closed convex subsets of $M$ has a non-empty intersection.
Proof Let $M$ be $w$-compact. Take any decreasing sequence of non-empty closed convex subsets $K_n$ of $M$. Select $x_n \in K_n$ for $n=1,2,...$ and suppose, choosing a sub-sequence if necessary, that $x_n \rightharpoonup x_0 \in M$. (The half-arrow denotes weak convergence.) We shall show that for any $f \in X^*$

$$
\lambda = \lim f(x_n) \leq f(x_0) \leq \overline{\lim} f(x_n) = L. \quad (3.1)
$$

Let $f(x_0) > L + \varepsilon$ for some $\varepsilon > 0$. Then there are only finitely many $x_n$ such that $|f(x_n) - f(x_0)| < \varepsilon$, contradicting $x_n \rightharpoonup x_0$. Hence $f(x_0) \leq L$ and similarly $f(x_0) \geq \lambda$. Now suppose $x_0 \notin \bigcap_{n=1}^{\infty} K_n$. Then there exists $N$ such that $x_0 \notin K_N$. Using the previous theorem with $B = \{x_0\}$ and $A = K_N$, we see that there is some $f \in X^*$ such that $f(x_n) < f(x_0)$ for all $n \geq N$, i.e. $\lim_{n \to \infty} f(x_n) < f(x_0)$, contradicting (3.1). Hence $x_0 \in \bigcap_{n=1}^{\infty} K_n$. //

We now use the fact that every bounded subset $M$ of a locally convex semi-reflexive space $E$ is relatively $w$-compact. (The converse is also true, see Koethe [28, p.299].) If we also assume that $M$ is closed and convex, it follows from the last theorem that a decreasing sequence of non-empty closed convex subsets $B_n$ of $M$ has a non-empty intersection. Now let $y \in E \sim M$ and put

$$
B_n = \{x : d(x,y) \leq d(y,M) + 1/n\}.
$$

Then there exists some $x_0 \in \bigcap_{n=1}^{\infty} (B_n \cap M)$ and $d(x_0,y) = d(x_0,M)$. We therefore have

Theorem 3.8 If $E$ is a semi-reflexive, locally convex, real linear metric space, then every closed bounded and convex subset $M$ of $E$ is proximinal.
We can easily extend theorem 3.4 to include a uniqueness criterion. This result appears in Singer [22] and Cudia [8].

**Theorem 3.9** A Banach space $B$ is reflexive and strictly convex if and only if each closed convex subset $M$ of $B$ is a Chebyshev set.

**Proof** If the space $B$ in theorem 3.4 is also strictly convex, then the proximum must be unique. Conversely, suppose $B$ is not strictly convex. Then the boundary of the unit sphere contains a line segment with at least two best approximations to the origin. //

**Corollary 3.9** If a Banach space is sequentially convex, then it is reflexive and strictly convex.

We now give a counterexample to demonstrate that a reflexive space is not necessarily sequentially convex. The norm used appears in a different context in the paper by M.A. Smith [20].

**Example 3.1** Let $x = (x_j) \in \ell_2$ and define

$$
\|x\|_S = \max \{ |x_1|, \left( \sum_{j=2}^{\infty} x_j^2 \right)^{1/2} \}.
$$

It is easy to see that $\| \cdot \|_S$ is a norm on $\ell_2$. It follows from

$$(1/\sqrt{2}) \|x\|_2 \leq \|x\|_S \leq \|x\|_2$$

that $\| \cdot \|_S$ is equivalent to the usual $\ell_2$ norm. We define a linear map

$$T : \ell_2 \rightarrow \ell_2 : (x_1, x_2, \ldots) \mapsto \left( x_1/1, x_2/2, \ldots \right)$$

and a new norm
\[ \| x \|_W = (\| x \|_S^2 + \| T x \|_2^2)^{\frac{1}{2}}. \]

We then have
\[ \| x \|_S \leq \| x \|_W = \| x \|_S (1 + \| T x \|_2^2 / \| x \|_2^2)^{\frac{1}{2}} \]
\[ \leq \| x \|_S (1 + 2\| T \|_2^2 / \| x \|_S^2) \leq \| x \|_S (1 + 2\| T \|_2^2), \]

i.e. \( \cdot \) \( W \) and \( \cdot \) \( 2 \) are equivalent norms.

Now let
\[ x = (1/\sqrt{2}, 0, 0, \ldots), \quad y = (2/\sqrt{2}, 0, 0, \ldots) \]
and
\[ K = \{ x : \| x \|_W \leq 1 \}. \]

Then
\[ \| x \|_S = 1/\sqrt{2}, \quad T x = (1/\sqrt{2}, 0, 0, \ldots), \]
\[ \| T x \|_2 = 1/\sqrt{2} \quad \text{and} \quad \| x \|_W = 1. \]

From \( \| y \|_W = 2 \) and \( \| x-y \|_W = 1 \) we have
\[ d(y,K) = 1, \]

where \( d \) is calculated according to the norm \( \cdot \) \( W \).

Next define a sequence \( (x^{(n)}) \) by
\[ x^{(n)} = (1/\sqrt{2} - 1/n, 0, 0, \ldots, 1/\sqrt{2} - 1/n, 0, 0, \ldots), \quad n \geq 2. \]

Then
\[ T x^{(n)} = (1/\sqrt{2} - 1/n, 0, 0, \ldots, (1/\sqrt{2} - 1/n), 0, 0, \ldots), \]
\[ \| x^{(n)} \|_S = 1/\sqrt{2} - 1/n \quad \text{and} \quad \| x^{(n)} \|_W = \]
\[ = [2(1/\sqrt{2} - 1/n)^2 + (1/\sqrt{2} - 1/n)^2 / n^2]^{\frac{1}{2}} \]
\[ = [1 - 4/(n/2)^2 + 5/(2n^2) - 2/(n^3/2) + 1/n^4]^{\frac{1}{2}} < 1, \]

i.e. \( x^{(n)} \in K \). Moreover,
\[ x^{(n)} - y = (-1/\sqrt{2} - 1/n, 0, 0, \ldots, 1/\sqrt{2} - 1/n, 0, 0, \ldots) \]

- 32 -
\( T(x(n) - y) = (-1/\sqrt{2} - 1/n, 0, 0, \ldots, (1/\sqrt{2} - 1/n)/n, 0, 0, \ldots) \),
\[ \|x(n) - y\|_S = 1/\sqrt{2} + 1/n \] and
\[ \|x(n) - y\|_W = \]
\[ = [2(1/\sqrt{2} + 1/n)^2 + (1/\sqrt{2} - 1/n)^2/n^2]^{\frac{1}{2}} \]
\[ = [1 + 4/(n\sqrt{2}) + 5/(2n^2) - 2/(n^3\sqrt{2}) + 1/n^4]^{\frac{1}{2}} - 1. \]

It follows that \((x(n))\) is a minimizing sequence.

But \[ x(n) - x(n+1) = \]
\[ = (1/(n+1) - 1/n, 0, 0, \ldots, 1/\sqrt{2} - 1/n, -1/\sqrt{2} + 1/(n+1), 0, 0, \ldots) \]
and \[ T(x(n) - x(n+1)) = \]
\[ = (1/(n+1) - 1/n, 0, 0, \ldots, (1/\sqrt{2} - 1/n)/n, (-1/\sqrt{2} + 1/(n+1))/n, \ldots). \]

It follows that \[ \|x(n) - x(n+1)\|_S \]
\[ = [(1/\sqrt{2} - 1/n)^2 + (1/\sqrt{2} - 1/(n+1))^2]^{\frac{1}{2}} \] and
\[ \|x(n) - x(n+1)\|_W = [(1/\sqrt{2} - 1/n)^2 + (1/\sqrt{2} - 1/(n+1))^2 + \]
\[ + (1/(n+1) - 1/n)^2 + (1/n^2)(1/\sqrt{2} - 1/n)^2 + \]
\[ + (1/(n+1))^2(1/(n+1) - 1/\sqrt{2})^2]^{\frac{1}{2}}, \]
i.e. \((x(n))\) is not a Cauchy sequence. The result now follows in view of theorem 4.2 //
Chapter 4
The Relationship Between Sequential Convexity and Local Uniform Convexity

In order to disprove the conjecture that local uniform convexity implies sequential convexity we first establish some properties of equivalent norms (theorems 4.1 and 4.2) and then apply a theorem by Kadetz [5], which will be stated without proof.

Theorem 4.1 Let $B$ be a Banach space and $\| \cdot \|$, $\| \cdot \|'$ equivalent norms on $B$, with

$$k \| x \| \leq \| x \|' \leq K \| x \|$$

for all $x \in B$.

If the corresponding operator norms on $B^*$ and $B^{**}$ are $\| \cdot \|_*$, $\| \cdot \|'_*$ and $\| \cdot \|_{**}$, $\| \cdot \|'_{**}$ respectively, then

(i) $(1/K) \| f \|_* \leq \| f \|'_* \leq (1/k) \| f \|_*$ for all $f \in B^*$,

(ii) $k \| g \|_{**} \leq \| g \|'_{**} \leq K \| g \|_{**}$ for all $g \in B^{**}$.

Proof For $f \in B^*$, $\| f \|'_* = \sup \{ f(x) \mid \| x \| \leq 1 \}$

$$= \frac{1}{k} \sup \{ f(x) \mid \| x \| \leq 1 \} = \frac{1}{k} \| f \|_* .$$

Moreover, $\| f \|'_* \geq \sup \{ f(x) \mid \| x \| \leq 1 \}$, which proves (i).

Repeating the argument for the second dual gives (ii). //

Theorem 4.2 Let $B$ be a Banach space and $\| \cdot \|$, $\| \cdot \|'$ equivalent norms on $B$. Then $(B, \| \cdot \|)$, $(B, \| \cdot \|')$ are either both reflexive or both non-reflexive.
Proof This follows from the definition of reflexivity and the previous theorem.

Theorem 4.3 (Kadetz). If \((B, \|\cdot\|)\) is a separable Banach space, then there exists a locally uniformly convex norm \(\|\cdot\|',\) which is equivalent to \(\|\cdot\|\).

It now follows that if local uniform convexity implied sequential convexity we could make the separable space \(\ell_1\) sequentially convex by equivalent renorming. But Corollary 3.9 would then imply that \(\ell_1\) is reflexive, which disproves the conjecture.

We can also prove that, conversely, sequential convexity does not imply local uniform convexity. A supposed counter-example due to Anderson, which is cited in the survey paper by Cudia [8, p.83] was shown to be fallacious by M.A. Smith (private communication, 1976). In 1978 Smith [20] succeeded in constructing a norm which is not locally uniformly convex and has the following properties: (i) strict convexity, (ii) reflexivity, (iii) convergence property \((H)\). Property \((H)\) is well-known to hold in any Hilbert space:

Definition 4.1 A normed linear space \(E\) has property \((H)\) (also called the Radon-Riesz property) if \(x,x^{(n)} \in E,\)
\[\|x^{(n)}\| + \|x\| \text{ and } x^{(n)} \rightharpoonup x \text{ implies } x^{(n)} \rightharpoonup x.\] (\(\rightharpoonup\) denotes weak convergence in \(E\).)

Fan and Glicksberg proved [6, p.560] that a Banach space is sequentially convex if and only if it is reflexive and has
property (H). Smith's example therefore demonstrates that sequential convexity does not imply local uniform convexity. We shall now derive the same result in a different way, using a norm which is not strictly convex.

**Example 4.1** Let \( x = (x_1, x_2, \ldots) \in \ell_2 \) and define

\[
\|x\|_F = |x_1| + \left( \sum_{j=2}^{\infty} x_j^2 \right)^{\frac{1}{2}}.
\]

It is easy to see that \( \| \cdot \|_F \) is a norm on \( \ell_2 \). We prove that \( \| \cdot \|_F \) has the following properties

(i) \( \| \cdot \|_F \) and \( \| \cdot \|_2 \) are equivalent,

(ii) \( \| \cdot \|_F \) is not strictly convex,

(iii) \( \| \cdot \|_F \) is sequentially convex.

(i) Follows from \( \|x\|_2 \leq \|x\|_F \leq 2\|x\|_2 \).

(ii) Let \( x = (1, 0, 0, \ldots), \ y = (0, 1/\sqrt{2}, 1/\sqrt{2}^2, 1/\sqrt{2}^3, \ldots) \).

Then \( \|x\|_F = \|y\|_F = 1 \) and \( \|x+y\|_F = 2 \).

(iii) We first show that \( \| \cdot \|_F \) has property (H). Let \( x, x^{(n)} \in \ell_2 \), with \( \|x^{(n)}\|_F \rightarrow \|x\|_F \) and \( x^{(n)} \rightharpoonup x \).

If \( x = 0 \), then \( x^{(n)} \rightarrow 0 \). For \( x \neq 0 \) assume w.l.o.g. that \( \|x^{(n)}\|_F = \|x\|_F = 1 \). Then

\[
|x_1^{(n)}| + \left( \sum_{j=2}^{\infty} |x_j^{(n)}|^2 \right)^{\frac{1}{2}} = 1 = |x_1| + \left( \sum_{j=2}^{\infty} x_j^2 \right)^{\frac{1}{2}} \tag{4.1}
\]

Let \( y(n) = (x_2^{(n)}, x_3^{(n)}, \ldots) \) and \( y = (x_2, x_3, \ldots) \).

Since \( x^{(n)} \rightharpoonup x \), we have \( x_1^{(n)} \rightarrow x_1 \). We can now deduce from (4.1) that \( \|y(n)\|_2 \rightarrow \|y\|_2 \).
Since also $y^{(n)} \to y$, we obtain
\[ \|y^{(n)} - y\|_2 \to 0 \quad \text{and} \quad \]
\[ \|x^{(n)} - x\|_F = |x_1^{(n)} - x_1| + \|y^{(n)} - y\|_2 \to 0, \]
i.e. $\| \cdot \|_F$ has property (H). Moreover, the space
$(\ell_2', \| \cdot \|_F)$ is reflexive. Hence it is sequentially convex
as required. //

While the example shows that a sequentially convex space
need not be locally uniformly convex, it can nevertheless be
renormed with a locally uniformly convex norm. This follows
from corollary 3.9 and a result by Lindenstrauss, Asplund,
Troyanski et al.

Theorem 4.4 Each reflexive normed linear space can be
renormed with an isometric norm which is both locally
uniformly convex and strongly differentiable.

Proof See Day [1, p.72].

In view of theorems 3.2, 3.4 and 3.9 we can deduce
from theorem 4.2 another important result about the renorming
of Banach spaces.

Theorem 4.5 A non-reflexive Banach space cannot be renormed
with a uniformly convex or a sequentially convex norm.

The converse of theorem 4.5 is false, i.e. there are
reflexive spaces which are not uniformly convex renormable.
An example, due to M.M.Day can be found in Koethe [28,p.361].
It remains an open question whether every reflexive Banach
space can be given a strictly convex norm.

We are now in a position to replace Clarkson's proofs of various renorming results (see the remarks following theorem 2.4) by a simple corollary to the last theorem.

Corollary 4.5 The spaces \(L_1, L_\infty, C, \ell_\infty\) and \(c\) cannot be renormed with uniformly convex norms.

It is clear that we can add other spaces such as \(\ell_1, c_0\) and \(BV\), the space of functions of bounded variation, to Clarkson's original list.

The results of this chapter can be summarized by saying that only certain weaker convexity properties, e.g. strict and local uniform convexity, can be obtained by Clarkson's method. While equivalent renorming can improve uniqueness properties and certain local properties of a space, it is impossible to affect global proximality in this way.
Chapter 5
Convexity and Best Approximation in Metric Spaces

We now discuss the problem of best approximation by elements of a subset $M$ of a metric space $X$. In trying to generalize the theory of metric spaces we first note that the convexity properties of a normed linear space will have to be modified to allow for the lack of linear structure. Non-linear spaces are not just of theoretical interest as the following examples demonstrate: the space of all non-decreasing functions on $[a,b]$, the space of functions $f$ on $[a,b]$ with $f(a) = c \neq 0$ and the space of functions $f$ with $\int_a^b f = c \neq 0$. Definitions of this type appear in a paper by Ahuja, Narang and Trehan [13]. These authors generalize the notions of strict and uniform convexity and show that certain approximation results, such as theorem 1.4, remain true for metric spaces. Ahuja et al. make the assumption that $M$ is convex, i.e. for any two points $x, y \in M$ any point between $x$ and $y$ is also in $M$, and that the space $X$ is strongly convex which means that if $x, y \in X$, then there is a unique $z \in X$ such that $d(x, z) = d(z, y) = d(x, y)/2$. (See also Rolfsen [14].) We shall see that these convexity properties can sometimes be replaced by weaker conditions.

Definition 5.1 A set $M$ will be called semi-convex if for all $x, y \in M$ there exists at least one intermediate point $z \in M$ such that
\[ d(x, z) + d(z, y) = d(x, y). \]
A metric space is called strictly convex, if $x \neq y$, $d(x, x_o) \leq r$
and \( d(y,x_0) \leq r \) implies \( d(z,x_0) < r \) whenever \( z \) is an intermediate point of \( x \) and \( y \).

**Definition 5.2** Let \((X,d)\) be a metric space and \(M\) a semi-convex subset of \(X\). If \((x_n)\) is a sequence in \(M\) such that

\[
\lim_{n \to \infty} d(x_n,x_0) = \inf_{x \in M} d(x,x_0)
\]

for some point \(x_0 \in X - M\), then \((x_n)\) is called a minimizing sequence. \((X,d)\) is called sequentially convex, if every minimizing sequence is a Cauchy sequence.

**Definition 5.3** (see Efimov and Stechkin [24]). A set \(M\) is said to be approximatively compact if every minimizing sequence in \(M\) has a sub-sequence convergent in \(M\).

**Theorem 5.1** Let \(M\) be an approximatively compact semi-convex subset of a strictly convex metric space \((X,d)\). Then \(M\) is a Chebyshev set.

**Proof** The proximinality of \(M\) was proved by Efimov and Stechkin [24] and does not depend on the semi-convexity of \(M\): first note from

\[
|d(x,y_1) - d(x,y_2)| \leq d(y_1,y_2), \quad (x,y_1,y_2 \in X),
\]

that for any given \(x\), the functional \(f(y) = d(x,y)\) is uniformly continuous in \(y\). For any \(x_0 \in X - M\) there exists a sequence \((d(x_0,y_n))\) with \(y_n \in M\), such that

\[
\lim_{n \to \infty} d(x_0,y_n) = d(x_0,M).
\]

Since \(M\) is approximatively compact, there is a subsequence \((y_{n_k})\) of \((y_n)\) which converges to some \(y \in M\). The uniform continuity of \(d(x,y)\) now gives
It follows that $M$ is proximinal.

Now let $y, y' \in M$, with $d(x, y) = d(x, y') = d(x, M)$, and let $z \in M$ be an intermediate point of $y, y'$. Since $(X, d)$ is strictly convex we have $d(x, z) < d(x, M)$, which contradicts the definition of $d(x, M)$. Hence $y = y'$ and $M$ is a Chebyshev set. //

Example 5.1 Let $A = \{ x \in \mathbb{Q} : x \in (-1, 1) \}$, $X = \mathbb{R} \setminus A$, and $M = [-1, 1] \setminus A$. Then $M$ is approximatively compact and semi-convex, but not convex. This shows that the above result is stronger than Ahuja's theorem 2 [13, p.95], in which $M$ is assumed to be convex. //

If in definition 5.3 convergence is replaced by weak convergence, we obtain a generalization of approximative compactness which was first proposed by W. Breckner [11].

Corresponding to three types of weak compactness, we obtain in this way three weak types of approximative compactness, which enable us to deduce that the following sets are proximinal

2. Weak* closed subsets of the dual space.
3. Weakly closed sets of operators on a Hilbert space.

L.P. Vlasov [26] introduced the concept of $\tau$-compactness which includes the various forms of compactness mentioned above. Similar ideas are contained in an article by F. Deutsch [27], who obtains a very general approximation
theorem which adds the following subsets of $C[a,b]$ to the above list:

4. Spline functions with free knots.
5. Exponential sums.
6. Rational functions.

We now adapt the definition of approximative $\tau$-compactness for metric spaces and prove a corresponding generalization of theorem 5.1. Recall that $(A, \preceq)$ is called a directed set if for all $\alpha, \beta \in A$ there is some $\gamma \in A$ so that $\alpha \preceq \gamma$ and $\beta \preceq \gamma$, where the relation '$\preceq$' is reflexive, transitive, and antisymmetric.

**Definition 5.4** Let $M$ be an arbitrary set. If $\alpha \in A$ defines an element $x_\alpha \in M$, then the $(x_\alpha)$ form a net in $M$. A subset $B$ of $A$ is said to be cofinal if for all $\alpha \in A$ there is some $\beta \in B$ such that $\beta \geq \alpha$. The corresponding net $(x_\beta)$ will be called a cofinal subnet of $(x_\alpha)$.

Now let $(X, d)$ be a metric space. We define a class of convergence processes called $\tau$-convergence in the following way. Each $\tau$-convergent net $(x_\alpha)$ in $X$ is associated with a unique element $x \in X$ so that for all $y \in X$.

(i) $d(x_\alpha, x) \uparrow 0 \Rightarrow d(x_\alpha, y) \uparrow d(x, y),$  
(ii) $d(x_\alpha, x) \uparrow 0 \Rightarrow d(x, y) \leq \lim d(x_\alpha, y).$

The following are examples of $\tau$-convergence.

**Example 5.2**

1. Convergence in a metric space: $d(x_\alpha, x) \to 0.$
2. Weak convergence: \( f(\mathbf{x}_Q) \to f(\mathbf{x}) \) for each \( f \in X^* \).

3. Weak* convergence in \( X^* \): \( f_\alpha(x) \to f(x) \) for each \( x \in X \).

4. Pointwise convergence in \( C(X) \) on a dense subset \( X_0 \) of the compact Hausdorff space \( X : f_\alpha(x) \to f(x) \) for each \( x \in X_0 \).

If \( M \) is a subset of \( X \), then \( M \) is called approximatively \( \tau \)-compact, if for all \( x \in X - M \) and any minimizing net \( (\mathbf{x}_\alpha) \) such that \( d(x_\alpha, x) \to d(x, M) \), there is a cofinal subnet of \( (\mathbf{x}_\alpha) \), which \( \tau \)-converges to some point in \( M \). A set \( F \) is \( \tau \)-closed if it contains the limit of each \( \tau \)-convergent net. The metric projection \( P_M : X \to 2^M \) will be called upper \( \tau \)-metric semi-continuous at \( x_0 \) if for any \( (\mathbf{x}_\alpha) \) with \( d(x_\alpha, x_0) \to 0 \) and for any \( \tau \)-open set \( U \supset P_M(x_0) \), there is some \( \beta \) such that \( U \supset P_M(x_\alpha) \) for all \( \alpha \geq \beta \).

**Theorem 5.2** Let \( M \) be an approximatively \( \tau \)-compact, semi-convex subset of a strictly convex metric space \( X \). Then \( M \) is a Chebyshev set and the metric projection \( P_M \) is upper \( \tau \)-metric semi-continuous.

**Proof** Let \( x_0 \in X - M \) and \( (y_\alpha) \) be a net in \( M \) so that \( d(y_\alpha, x_0) \to d(x_0, M) \). Then there exists a cofinal subnet \( (y_\beta) \) which \( \tau \)-converges to some \( y \in M \). Hence
\[
d(y_\beta, x_0) \uparrow d(y, x_0)
\]
and
\[
d(y, x_0) \leq \lim d(y_\beta, x_0) = d(x_0, M),
\]
i.e. \( y \in P_M(x_0) \) and \( M \) is proximinal. The uniqueness of \( y \) follows as in the proof of theorem 1.4.
Now suppose $P_M$ is not upper semi-continuous. Then there exists a net $(x_\alpha)$ with $d(x_\alpha, x_o) \to 0$ and a $\tau$-open set $U \supset P_M(x_o)$, so that for any $\beta$ there is some $\alpha \geq \beta$ with $P_M(x_\alpha) \sim U \neq \emptyset$. If one element $p_\alpha$ is selected from each set $P_M(x_\alpha) \sim U$, then

$$d(x_o, M) \leq d(x_o, p_\alpha) \leq d(x_o, x_\alpha) + d(x_\alpha, p_\alpha),$$

$$= d(x_o, x_\alpha) + d(x_\alpha, M) + d(x_o, M),$$

i.e. $(p_\alpha)$ is a minimizing net. Next let $(p_\beta)$ be a cofinal subnet of $(p_\alpha)$, with $p_\beta \overset{\tau}{\to} p_o \in M$.

Then

$$d(p_o, x_o) \leq \lim_{\beta} d(p_\beta, x_o) = d(x_o, M),$$

i.e. $p_o \in P_M(x_o) \subset U$. Since $p_\beta$ is an element of the $\tau$-closed set $X \sim U$, we have $p_o \in X \sim U$. This contradiction shows that $P_M$ is upper $\tau$-metric semi-continuous. //

Since compactness implies approximative compactness we can extend theorem 0.3 to include a uniqueness condition. We also state a metric space version of theorem 3.1. It can be proved that a compact or complete semi-convex set is convex in the usual sense. The two theorems are therefore stated for convex sets.

**Theorem 5.3** Let $M$ be a compact, convex set in a strictly convex metric space. Then $M$ is a Chebyshev set.

**Theorem 5.4** If $M$ is a closed convex subset of a complete sequentially convex metric space $(X,d)$, then $M$ is a Chebyshev set.
Proof Let $x_o \in X - M$ and $\alpha = d(x_o,M)$. Then $\alpha > 0$ and there is a minimizing sequence $(y_n)$ in $M$. Existence and uniqueness of the proximum now follow as in the proof of theorem 3.2. //

We next show how the approximation properties of a metric space can be improved by introducing an equivalent metric. The following theorem will be needed (see Kantorowitch and Akilow [16, p.235]):

**Theorem 5.5** Every separable metric space $(X,d)$ is isometric to a subset of the space $C[0,1]$.

**Proof** Let $M = \{x_1,x_2,\ldots\}$ be dense in $X$. Define a mapping

$$U : X \to l_\infty : x \mapsto y = (y_1,y_2,\ldots)$$

by

$$y_j = d(x,x_j) - d(x_1,x_j)$$

for $j = 1,2,3,\ldots$. Since

$$|y_j| = |d(x,x_j) - d(x_1,x_j)| \leq d(x,x_1)$$

we have $y \in l_\infty$. Now let $U(x) = y$ and $U(x') = y'$. It is easy to see that

$$\|y - y'\|_\infty = \sup |y_j - y'_j| \leq \sup |d(x,x_j) - d(x',x_j)| \leq d(x,x') \quad (5.1)$$

It now follows from the definition of $M$ that there exists $x_n \in M$ so that $d(x,x_n) \leq \varepsilon/2$, with $0 < \varepsilon < d(x,x')$. But $d(x',x_n) \geq d(x',x) - d(x_n,x) \geq d(x',x) - \varepsilon/2 > 0$. 

- 45 -
Hence
\[ |y_n - y'_n| = d(x_n, x') - d(x_n, x) = |d(x_n', x') - d(x_n, x)| \]
\[ \geq d(x', x_n) - \varepsilon/2 \geq d(x', x) - \varepsilon, \]
i.e. \[ \|y - y'\|_\infty \geq d(x, x') - \varepsilon. \]
Since \( \varepsilon > 0 \) is arbitrary, we have
\[ \|y - y'\|_\infty > d(x, x'). \]
Using (5.1) we see that \( \|y - y'\|_\infty = d(x, x') \), which shows that \( X \) is isometric to a subset of \( \ell_\infty \). The linear hull of this subset is clearly separable and the result now follows from theorem 2.2. //

We are now in a position to generalize Clarkson's method to semi-convex metric spaces.

Theorem 5.6 Let \((X, d)\) be a separable, semi-convex metric space. Then there is a strictly convex metric \( d' \) which is equivalent to \( d \).

Proof By theorem 5.5 there is an isometry \( T : X \to C[0,1] \), with \( d(x_1, x_2) = \|T(x_1) - T(x_2)\|_\infty \).
Now let \( d'(x_1, x_2) = \|T(x_1) - T(x_2)\|_{C} \), where \( \|\cdot\|_C \) is defined as in the proof of theorem 2.1. Then \( \|\cdot\|_C \) is strictly convex and we have
\[ \|x\|_\infty \leq \|x\|_C \leq (2/\sqrt{3})\|x\|_\infty. \]
Let \( \varepsilon > 0 \) be given. If \( d(x_1, x_2) < \varepsilon \sqrt{3}/2 \), then
\[ d'(x_1, x_2) \leq (2/\sqrt{3})d(x_1, x_2) < \varepsilon. \]
Conversely, if \( d'(x_1, x_2) < \varepsilon \sqrt{3}/2 \), then
\[
d(x_1, x_2) \leq d'(x_1, x_2) < \varepsilon \sqrt{3}/2 < \varepsilon,
\]
i.e. \( d \) and \( d' \) are equivalent metrics.

We finally show that \( d' \) is strictly convex. Let \( d'(x, x_0), d'(y, x_0) \leq r \) and \( d'(z, x_0) = r \), where \( z \) is an intermediate point of \( x, y \). It follows from the definition of \( d' \) and the strict convexity of \( \| \cdot \|_C \) that \( T(x) = T(y) \). But \( T \) is injective. Hence \( x = y \) and \( d' \) is strictly convex. //

Equivalent metrization can be used to make a given closed set proximinal. We require the following

**Lemma 5.6** (see Singer [22, p.391]). Let \((X, d)\) be a metric space and \( M \) a subset of \( X \). Then
\[
|d(x, M) − d(y, M)| ≤ d(x, y)
\]
for all \( x, y \in X \).

**Proof** Let \( x, y \in X \) and \( \varepsilon > 0 \). Then there exists an element \( m \in M \) such that \( d(y, m) \leq d(y, M) + \varepsilon \).

Hence \( d(x, M) ≤ d(x, m) ≤ d(x, y) + d(y, m) \)
\[
≤ d(x, y) + d(y, M) + \varepsilon.
\]

Since \( \varepsilon > 0 \) is arbitrary,
\[
d(x, M) − d(y, M) ≤ d(x, y). //
\]

We now define a new metric \( d^{(n)} \) by
\[
d^{(n)}(x, y) = \max \{ d(x, y), (1+1/n)|d(x, M) − d(y, M)| \}.
\]
It is easy to see that $d$ and $d(n)$ are equivalent.

First note that $d(n)(x,y) \geq d(x,y)$. If

$$d(x,y) \geq (1+1/n)|d(x,M) - d(y,M)|$$

then $d(n)(x,y) = d(x,y)$. But if

$$d(x,y) < (1+1/n)|d(x,M) - d(y,M)|$$

then

$$d(n)(x,y) = (1+1/n)|d(x,M) - d(y,M)| \leq (1+1/n)d(x,y)$$

by the lemma. Hence

$$d(x,y) \leq d(n)(x,y) \leq (1+1/n)d(x,y),$$

i.e. $d$ and $d(n)$ are equivalent.

Now let $M$ be a closed proper subset of $X$ and $y \in X - M$. Then for any $x \in M$,

$$d(n)(x,y) = \max\{d(x,y), (1+1/n)d(y,M)\} \geq$$

$$\geq (1+1/n)d(y,M) > 0.$$  

If we now choose a point $m \in M$ such that

$$d(m,y) < (1+1/n)d(y,M),$$

then

$$d(n)(m,y) = (1+1/n)d(y,M) = \max\{d(y,M),(1+1/n)d(y,M)\}$$

$$= \inf_{x \in M} \max\{d(x,y), (1+1/n)d(y,M)\}$$

$$= \inf_{x \in M} d(n)(y,x),$$

i.e. $M$ is proximinal. We therefore have the following

**Theorem 5.7** Let $(X,d)$ be a metric space and $M$ a closed
proper subset of $X$. Then $M$ is proximinal with respect to the metric

$$d_n(x,y) = \max \{d(x,y), (1+1/n)|d(x,M) - d(y,M)|\},$$

which is equivalent to $d$, with

$$d(x,y) \leq d_n(x,y) \leq (1+1/n)d(x,y)$$

for all $x, y \in X$.

**Example 5.2** Let $X = \ell_1$, $y = (0,0,0,...)$ and $M = \{x = (0,x_2,x_3,...) \in \ell_1 : \sum_{n=2}^{\infty} nx_n/(n+1) = 1\}$. Then $M$ is a closed convex subset of $\ell_1$. Let $d$ be the metric defined by the $\ell_1$ norm. Then $d(x,0) > 1$ for all $x \in M$. Since $m_n = (0,0,...,(n+1)/n,0,0,...) \in M$ and $d(m_n,0) = 1 + 1/n$, we see that $d(y,M) = 1$, i.e. $M$ is not proximinal with respect to $d$. On the other hand, if $x \in M$ then

$$d_n(x,0) = \max \{d(x,0), (1+1/n)d(y,M)\} = \max \{d(x,0), 1+1/n\}.$$

If $p$ is chosen so that $p > n$, then

$$d(m_p,0) = 1+1/p < 1 + 1/n \quad \text{and} \quad d_n(m_p,0) = 1 + 1/n = d_n(y,M). \quad \text{Hence } M \text{ is proximinal with respect to } d_n. \quad //$$

The example demonstrates that the proximity obtained in this way are not generally unique. Since a proximinal set is always closed we can use theorem 5.7 to characterize the closed sets in a metrizable topological space. Using the
metric $d_{(1)}$, this was done by V.L. Klee [25], who also showed that if $M$ is proximinal with respect to all equivalent metrics, then $M$ is compact.
Chapter 6

Best Approximation in the $L_p$ norms

The purpose of this chapter is to provide a link between the abstract material of part I of this thesis and the numerical applications of part II. We shall concentrate on $L_1$ and $L_\infty$ approximation, with occasional references to the $L_2$ norm. Historically, the three norms date back to the early 1800s. The earliest reference to discrete $L_1$ and $L_\infty$ can be found in Laplace's "Mécanique Céleste", which was published in 1799. Laplace's ideas gained a certain notoriety for arithmetic unwieldiness and were soon eclipsed by the least squares technique of Gauss and Legendre. Although the period from about 1850 to 1950 saw considerable advances in $L_1$ and $L_\infty$ theory through the work of Chebyshev, Weierstrass, de la Vallée-Poussin, Banach, Jackson and others, the practical importance of these results remained somewhat limited until the arrival of electronic computers in the early 1950s. Computers created an urgent need for efficient methods of functional approximation, an area in which the $L_\infty$ norm offers distinct advantages over other norms. At the same time, the spectacular increase in computing power revived research into a number of algorithms which had hitherto been regarded as computationally too expensive. Laplace's ideas on the solution of inconsistent linear systems as well as the algorithms of Remes belong to this category.

In subsequent chapters, frequent use will be made of an
important alternation property, which characterizes polynomials of best $L_{\infty}$ approximation. This property was discovered by Chebyshev in the 1850s. For a proof see Cheney [21,p.75]. We first require the following definition.

**Definition 6.1** A set of functions $\{g_1, \ldots, g_n\}$ satisfies the Haar condition if every set of vectors of the form

$$g(x_i) = (g_1(x_i), \ldots, g_n(x_i)), \quad i = 1(1)n,$$

is linearly independent for any distinct $x_i$, i.e. if the determinant

$$\begin{vmatrix}
g_1(x_1) & \cdots & g_n(x_1) \\
g_1(x_2) & \cdots & g_n(x_2) \\
\vdots & \cdots & \vdots \\
g_1(x_n) & \cdots & g_n(x_n)
\end{vmatrix}$$

does not vanish for distinct $x_1, \ldots, x_n$.  

It is easy to show that the Haar condition holds if and only if every generalized polynomial

$$g(x) = \sum_{i=1}^{n} c_i g_i(x) \neq 0$$

has at most $n-1$ distinct zeros.

**Theorem 6.1** Let $g_1, \ldots, g_n \in C[a,b]$ and $f \in C(X)$, where $X$ is a closed subset of $[a,b]$. If $\{g_1, \ldots, g_n\}$ satisfies the Haar condition, then the generalized polynomial

$$g(x) = \sum_{i=1}^{n} c_i g_i(x)$$

is a best uniform approximant to $f$ on $X$ if and only
if there are \( n+1 \) points \( x_1, \ldots, x_{n+1} \in X \), with 
\( x_1 < \ldots < x_{n+1} \), such that 
\[
|g(x_i) - f(x_i)| = \|g - f\|_\infty
\]
and \( g(x_i) - f(x_i) \) alternates in sign for \( i=1, \ldots, n+1 \).

In the language of chapter 1, the linear space 

\[ M = \langle g_1, \ldots, g_n \rangle \]

is a finite-dimensional subspace of 

\[ C[a, b] \]. It is clear from theorem 1.2 that \( M \) is proximinal. Although the \( L_\infty \) norm is not strictly convex, the polynomial approximant in theorem 6.1 is in fact unique. The function subspaces which have this uniqueness property are characterized by theorem 6.2, which is due to A. Haar [59].

The proof given below follows Achieser [12, p. 67 ff.], who considers "\( n \) linearly independent real functions of the point \( P \) of a bounded closed set in ordinary space of any number of dimensions". This terminology suggests that the author refers to finite-dimensional domains, but the proof easily carries over to compact Hausdorff spaces and in particular to compact metric spaces.

**Theorem 6.2** Let \( M = \langle f_1, \ldots, f_n \rangle \) be an \( n \)-dimensional subspace of \( C(X) \), where \( X \) is a compact metric space. Then \( M \) is a Chebyshev subspace if and only if the set \( \{f_1, \ldots, f_n\} \) satisfies the Haar condition.

**Proof** Suppose the Haar condition is not satisfied. Then there exist \( n \) distinct points \( x_1, \ldots, x_n \) in \( X \) so that
and we can find scalars \( a_1, \ldots, a_n \) (not all zero),
with
\[
\sum_{k=1}^{n} a_k \frac{f_k(x_1)}{\|f_k\|_\infty} + \ldots + a_n \frac{f_n(x_n)}{\|f_n\|_\infty} = 0,
\]
for any function \( f \) in \( M \).
Now let
\[
F(x) = b_1 f_1(x) + \ldots + b_n f_n(x)
\]
be a function in \( M \) with \( \|F\|_\infty < 1 \) and \( F(x) \geq 0 \). If \( g \in C(x) \) with
\[
|g(x)| \leq 1 \quad \text{on } X \quad \text{and } \quad g(x_i) = \text{sgn } a_i \quad \text{for } a_i \neq 0 \quad (i=1, \ldots, n),
\]
then the function
\[
h(x) = g(x) \left[ 1 - |F(x)| \right]
\]
satisfies
\[
|h(x)| \leq 1 \quad \text{and } \quad h(x_i) = \text{sgn } a_i
\]
for \( a_i \neq 0 \quad (i=1, \ldots, n) \). If for some \( f \in M \), \( |F|_{\infty} < 1 \),
then \( \text{sgn } f(x_i) = \text{sgn } a_i \) for \( a_i \neq 0 \quad (i=1, \ldots, n) \),
contradicting equation (6.1). It follows that \( \|h-f\|_{\infty} \geq 1 \)
for all \( f \) in \( M \).
Conversely, let \( |\varepsilon| \leq 1 \). Then
\[
|h(x) - \varepsilon F(x)| \leq |h(x)| + |\varepsilon F(x)|
\]
\[
= |g(x)| \left[ 1 - |F(x)| \right] + |\varepsilon F(x)|
\]
\[
\leq 1 - |F(x)| + |\varepsilon||F(x)| \leq 1.
\]
Hence εF is a best approximant to h for all |ε| ≤ 1, i.e. \( P_M(h) \) is an infinite set and \( M \) is not semi-Chebyshev. //

To prove sufficiency, a number of lemmas are required. In each case the Haar condition is assumed.

**Lemma 6.3** Let

\[
\begin{vmatrix}
 f_i(x_1) & f_{i+1}(x_1) & \ldots & f_k(x_1) \\
 \vdots & \vdots & \ddots & \vdots \\
 f_i(x_k) & f_{i+1}(x_k) & \ldots & f_k(x_k)
\end{vmatrix} = 0 \quad (1 \leq i < k < n). \tag{6.2}
\]

Then for any \( q, k < q \leq n \), there exist points \( x_{k+1}', x_{k+2}', \ldots, x_q' \) such that

\[
\begin{vmatrix}
 f_i(x_1) & f_{i+1}(x_1) & \ldots & f_q(x_1) \\
 \vdots & \vdots & \ddots & \vdots \\
 f_i(x_q) & f_{i+1}(x_q) & \ldots & f_q(x_q)
\end{vmatrix} \neq 0.
\]

**Proof** It follows from (6.2) and the Haar condition that the non-trivial generalized polynomial

\[
f(x) = \begin{vmatrix}
 f_i(x_1) & f_{k+1}(x_1) \\
 \vdots & \vdots \\
 f_i(x_k) & f_{k+1}(x_k) \\
 f_i(x) & f_k(x)
\end{vmatrix}
\]

has at most \( n-1 \) zeros. Hence there is a point \( x_{k+1} \) such that \( f(x_{k+1}) \neq 0 \). //

**Lemma 6.4** If \( x_1, \ldots, x_k (k < n) \) are distinct points, then the matrix
\[
\begin{pmatrix}
  f_1(x_1) & \ldots & f_n(x_1) \\
  \vdots & & \vdots \\
  f_1(x_k) & \ldots & f_n(x_k)
\end{pmatrix}
\]

has at least one non-zero minor of order \( k \).

**Proof** We first prove the result for \( k=1 \). If \( f_1(x_1) = 0 \) for \( i=1, \ldots, n \), choose \( y_2 \) such that \( f_2(y_2) \neq 0 \) and use lemma 6.3 to determine \( y_3, \ldots, y_n \) such that

\[
\begin{vmatrix}
  f_2(y_2) & \ldots & f_n(y_2) \\
  \vdots & & \vdots \\
  f_2(y_n) & \ldots & f_n(y_n)
\end{vmatrix} \neq 0.
\]

Then

\[
\begin{vmatrix}
  f_1(x) & f_2(x) & \ldots & f_n(x) \\
  f_1(y_2) & f_2(y_2) & \ldots & f_n(y_2) \\
  \vdots & \vdots & & \vdots \\
  f_1(y_n) & f_2(y_n) & \ldots & f_n(y_n)
\end{vmatrix}
\]

has \( n \) distinct zeros \( x_1, y_2, \ldots, y_n \) which contradicts the Haar condition. Hence \( f_1(x_1) \neq 0 \) for some \( i, 1 \leq i \leq n \).

Next suppose the lemma is true for \( k=1, \ldots, m-1 \). W.l.o.g. we assume

\[
\begin{vmatrix}
  f_2(x_2) & \ldots & f_m(x_2) \\
  \vdots & & \vdots \\
  f_2(x_m) & \ldots & f_m(x_m)
\end{vmatrix} \neq 0.
\]

By the previous lemma we can find points \( y_{m+1}, \ldots, y_n \) such that

\[
\begin{vmatrix}
  f_2(x_2) & \ldots & f_n(x_2) \\
  \vdots & & \vdots \\
  f_2(y_n) & \ldots & f_n(y_n)
\end{vmatrix} \neq 0.
\]
If the assertion was false, then

\[
\begin{bmatrix}
f_1(x) & \cdots & f_n(x) \\
f_1(x_2) & \cdots & f_n(x_2) \\
\vdots & & \vdots \\
f_1(y_n) & \cdots & f_n(y_n)
\end{bmatrix}
\]

would be a non-trivial polynomial with \( n \) zeros \( x_1, \ldots, x_m, y_{m+1}, \ldots, y_n \), which contradicts the Haar condition. //

Lemma 6.5 Let \( F(x) = \alpha_1 f_1(x) + \cdots + \alpha_n f_n(x) \) be a function in \( M \) and \( f \in C(X) \).

If \(|f(x) - F(x)| = \|f(x) - F(x)\|_\infty\) (6.3)

for fewer than \( n \) values of \( x \), then \( F \notin \mathcal{P}_M(f) \).

Proof Suppose \( x_1, \ldots, x_m (m < n) \) are distinct points in \( X \) for which (6.3) holds. Then by lemma 6.4 we can solve the underdetermined system

\[
\beta_1 f_1(x_k) + \cdots + \beta_n f_n(x_k) = f(x_k) - F(x_k)
\]

\((k=1, \ldots, m)\) for \( \beta_1, \ldots, \beta_n \).

Let

\[
G(x) = \beta_1 f_1(x) + \cdots + \beta_n f_n(x)
\]

and

\[
r(x) = f(x) - F(x).
\]

For each \( x_k (k=1, \ldots, m) \), choose a closed neighbourhood \( N_k \) such that

\[
\mu_k(F) = \min_{x \in N_k} |r(x)| > 0 \quad \text{and} \quad \min_{x \in N_k} |G(x)| \geq \|f-F\|_\infty/2.
\]

- 57 -
Next suppose that $M_k = \max_{x \in N_k} |G(x)|$, $M = \max_{x \in N^*} |G(x)|$, and $L^*(F) = \max_{x \in N^*} |r(x)|$, where $N^* = N_1 \sim \ldots \sim N_M$.

Clearly,

$$u = u(F) = \max_{x \in X} |r(x)| - \max_{x \in N^*} |r(x)| > 0.$$ 

Now choose $\varepsilon$ such that

$$0 < \varepsilon < \min \left( \frac{u}{M}, \frac{u_1}{M_1}, \ldots, \frac{u_m}{M_m} \right).$$

Put

$$Y_i = \alpha_i + \varepsilon \beta_i \quad (i = 1, \ldots, n)$$

and

$$H(x) = \gamma_1 f_1(x) + \ldots + \gamma_n f_n(x).$$

Then

$$|f(x) - H(x)| = |f(x) - F(x) - \varepsilon G(x)| = |r(x) - \varepsilon G(x)|.$$ 

Hence

$$|f(x) - H(x)| \leq |r(x)| \left( 1 - \varepsilon \frac{G(x)}{r(x)} \right) \leq \|f - F\|_{\infty} (1 - \varepsilon/2)$$

whenever $x \in N_k \quad (k = 1, \ldots, m)$, and

$$|f(x) - H(x)| \leq |r(x)| + \varepsilon |G(x)| \leq L^*(F) + \varepsilon M < \|f - F\|_{\infty}$$

whenever $x \in N^*$. We therefore have

$$\|f - H\|_{\infty} = \max_{x \in X} |f(x) - H(x)| < \|f - F\|_{\infty}. \quad \Box$$

We can now prove the sufficiency of the Haar condition.

**Proof** Suppose $F(x) = \alpha_1 f_1(x) + \ldots + \alpha_n f_n(x)$ and $G(x) = \beta_1 f_1(x) + \ldots + \beta_n f_n(x) \in \mathcal{P}_M(f)$. Since

- 58 -
\begin{align*}
|\frac{1}{2}(F+G) - f| & \leq \frac{1}{2} |F - f| + \frac{1}{2} |G - f|, \\
\end{align*}
we also have $\frac{1}{2}(F+G) \in P_M(f)$. By lemma 6.5, the equation
\begin{equation}
|f(x) - \frac{[F(x) + G(x)]}{2}| = L
\end{equation}
has at least $n$ zeros $x_1, \ldots, x_n \in X$, where

\begin{equation*}
L = \|(F+G)/2 - f\|_\infty = \|F - f\|_\infty = \|G - f\|_\infty.
\end{equation*}
But for $|f(x_i) - \frac{[F(x_i) + G(x_i)]}{2}|$ to equal $L$, it is necessary that

\begin{align*}
f(x_i) - F(x_i) &= f(x_i) - G(x_i) = \pm L.
\end{align*}
It follows that the non-trivial polynomial

\begin{equation*}
(\alpha_1 - \beta_1) f_1(x) + \ldots + (\alpha_n - \beta_n) f_n(x)
\end{equation*}
has $n$ distinct zeros, which proves the sufficiency of the Haar condition. //

Although theorem 6.2 is a result about functions defined on a compact Hausdorff space, its practical importance is restricted to the single variable case, because functions of several variables do not in general satisfy the Haar condition. This can be established by the following simple argument (see A.Haar [59, p.311]). Suppose the function

\begin{equation}
g(x) = \sum_{i=1}^{n} \lambda_i g_i(x) \neq 0
\end{equation}
satisfies the Haar condition on the unit square $X = [0,1]^2$. Then there exist at most $n-1$ distinct points $x_j \in X$ such that $\sum \lambda_i g_i(x_j) = 0$. It follows that, if $\sum \lambda_i g_i(x_j) = 0, j=1(1)n$, holds for $n$ distinct points, then $\lambda_i = 0$, i.e.
If we now interchange $x_1$ and $x_2$, say, keeping all $x_j$ distinct in the process, then the above determinant changes its sign and therefore must equal zero for some position of $x_1$ and $x_2$, contradicting the Haar condition.

A characterization of the set $X$ was first given by Mairhuber [60] in 1956. Similar results hold for $C(X)$ when $X$ is a compact Hausdorff space and for $C_0(X)$ when $X$ is a locally compact Hausdorff space (see Phelps [17] and Lutts [65]).

Theorem 6.4 (Mairhuber) Let $g_1, \ldots, g_n \in C(X)$, where $X$ is a compact subset of $\mathbb{R}^k$, containing at least $n$ points ($n \geq 2$). Then the set $\{g_1, \ldots, g_n\}$ satisfies the Haar condition if and only if $X$ is homeomorphic to a closed subset of the circumference of a circle.

The alternation property of theorem 6.1 also characterizes discrete best approximants. Discrete and continuous Chebyshev approximation are usually treated as separate topics, each with its own existence, uniqueness, and characterization theorems. (See for example chapters 2 and 3 in Cheney [21] or Watson [66].) However, it is possible to develop a unified theory in which discrete approximation is regarded as a special case of continuous approximation. We give a brief outline of such a theory.
Let $X = \{x_1, \ldots, x_m\} = \{1, \ldots, m\}$. Define a function $f : X \to \mathbb{R}$ by

$$f(x_i) = a_i, \quad i = 1(1)m.$$  

If $X$ is given the discrete topology, then each singleton set $\{x_i\}$ is open, i.e. $f$ is continuous. This topology is induced by the discrete metric $d$ defined by

$$d(x, y) = \begin{cases} 
0 & \text{if } x = y \\
1 & \text{if } x \neq y.
\end{cases}$$

It is clear that $(X, d)$ is a compact metric space and $C(X) = \mathbb{R}^m$.

Let $f, f_1, \ldots, f_n \in C(X)$. We can write

$$f = (a_1, \ldots, a_m)^T, \quad f_i = (a^i_1, \ldots, a^i_m)^T,$$

i.e. $f(k) = a_k$, $f_i(k) = a^i_k$ for $k = 1, \ldots, m$. The Haar condition demands that for any $n$ distinct points $x_i = k_i$ in $X$, where $i = 1, \ldots, n$ and $1 \leq k_i \leq m$, the vectors

$$(f_1(x_i), \ldots, f_n(x_i)) = (a^i_{k_1}, \ldots, a^i_{k_n})$$

are linearly independent. Denote the $m \times n$ matrix $(f_1, \ldots, f_n)$ by $A$. The Haar condition can then be expressed by saying that every $n \times n$ submatrix of $A$ is non-singular. We retain the equivalent definition that any generalized polynomial

$$\sum_{i=1}^n \alpha_i f_i \neq 0$$

has at most $n-1$ zeros in $X$. Clearly, $\{f_1, \ldots, f_n\}$ always satisfies the Haar condition if $m < n$.  

- 61 -
The problem of determining \( \alpha = (\alpha_1, \ldots, \alpha_n)^T \) so that
\[
\| A \alpha - b \|_\infty = \min
\]
where \( A \) is a given \( m \times n \) matrix, can now be interpreted as a problem of continuous approximation. If we identify \( A \) with \((f_1, \ldots, f_n)\) and \( b \) with \( f \), we require
\[
\| \sum_{i=1}^{n} \alpha_i f_i - f \|_\infty = \min
\]

Example 6.1 Find the minimax solution of the system

\[
\begin{pmatrix}
1 & -1 \\
1 & 1 \\
2 & 1
\end{pmatrix}
\begin{pmatrix}
\alpha_1 \\
\alpha_2
\end{pmatrix}
=
\begin{pmatrix}
2 \\
4 \\
8
\end{pmatrix}
\]

we have \( m=3, n=2, X = \{1,2,3\} \). The matrix \( A \) satisfies the Haar condition and
\[
\| \alpha_1 (1,1,2)^T + \alpha_2 (-1,1,1)^T - (2,4,8)^T \|_\infty
\]
is a minimum for the unique solution \((\alpha_1, \alpha_2) = (10/3,1)\).

Note that
\[
g(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x)
\]
has only one zero in \( X \). We find
\[
f(1) = \alpha_1 - \alpha_2 = 0 \quad \text{for} \quad \alpha_1 = \alpha_2.
\]
But
\[
f(2) = \alpha_1 + \alpha_2 \neq 0
\]
and
\[
f(3) = 2\alpha_1 + \alpha_2 \neq 0. \quad //
\]

Example 6.2 An example in Watson [66,p.33] is intended to show that a linear system can have a (strongly) unique solution when the Haar condition is not satisfied. However, the system
\[
\begin{pmatrix} 1 \\ 0 \end{pmatrix} \alpha = \begin{pmatrix} 0 \\ 0 \end{pmatrix}
\]

is equivalent to \( 1.\alpha = 0 \), i.e. the Haar condition is satisfied and the solution is unique.

**Example 6.3** Solve \( \alpha_1 + 3\alpha_2 = 6 \) in the minimax sense.

This is an underdetermined system, with \( m=1, n=2, X=\{1,2\} \).

The Haar condition is satisfied, and we have

\[ \|\alpha_1 + 3\alpha_2 - 6\|_\infty = 0 \quad \text{for} \quad (\alpha_1,\alpha_2) = (\alpha_1, 2-\alpha_1/3) \text{ and any } \alpha_1. \]

The unique element of \( M \) is

\[ \alpha_{11} f_1 + \alpha_{12} f_2 = \alpha_1 1 + (2-\alpha_1/3).3 = 6 = \sum f \in M, \]

i.e. the approximated function coincides with the approximant. //

Note that the word "solution" can denote the vector \( \alpha \) or the generalized polynomial \( \sum \alpha_i f_i \). By theorem 6.2, the latter is unique if and only if \( A \) satisfies the Haar condition. For uniqueness of the former, the Haar condition is necessary but, as example 6.3 demonstrates, not sufficient.

To clarify the situation, we distinguish between consistent and inconsistent systems. \( A\alpha = b \) is consistent if and only if \( b \) lies in the linear span of the columns \( f_i \) of \( A \), i.e. if and only if \( f \in M \). A consistent system has a unique solution \( \alpha \) if and only if the columns of \( A \) are linearly independent. For an inconsistent system, the minimax solution \( \alpha \) is unique if and only if \( A \) satisfies the Haar condition.

We now show that the characterization theorem for the
The minimax solution of $A\alpha = b$ can be deduced from the corresponding continuous result. In the literature, these theorems are usually treated independently of each other, with separate proofs for the continuous and discrete case (see for example Cheney [21, pp. 35 and 73]).

**Theorem 6.5 (Continuous Characterization Theorem)**

Let $f, f_1, \ldots, f_n \in C(X)$, where $X$ is a compact metric space. $\left\| \sum_{i=1}^{n} \alpha_i f_i - f \right\|_\infty$ is a minimum if and only if

$$0 \in H \{ r(x)(f_1(x), \ldots, f_n(x)) : |r(x)| = \|r\|_\infty \}$$

where $r(x) = \sum_{i=1}^{n} \alpha_i f_i(x) - f(x)$ and $H$ denotes the convex hull of a set.

To obtain the discrete version, let

$$M = \{ x \in K : |r(x)| = \|r\|_\infty \},$$

where $X = \{1, \ldots, m\}$. If $x = j$, $1 \leq j \leq m$, then

$$(f_1(x), \ldots, f_n(x)) = (a_{1j}, \ldots, a_{nj}) = A^j,$$

the $j$th row of $A$, i.e. the necessary and sufficient condition becomes

$$0 \in H \{ r(j)A^j : j \in M \}.$$

Let $\sigma_j = \text{sgn} r(j)$. Then there exist numbers $\theta_j \geq 0$ such that $\sum_{j \in M} \theta_j = 1$ and

$$0 = \sum_{j \in M} \theta_j r(j)A^j = \sum_{j \in M} \theta_j \sigma_j r(j)A^j.$$
Put $\sum_{j \in M} \theta_j \sigma_j r(j) = k$. Then

$$0 = \sum_{j \in M} \phi_j \sigma_j A^j,$$

where $\phi_j = \theta_j \sigma_j r(j)/k \geq 0$ and $\sum_{j \in M} \phi_j = 1$.

Hence $0 \in H\{\sigma_j A^j : j \in M\}$ and we obtain Theorem 6.6 (Discrete Characterization Theorem)

Let $\alpha = (\alpha_1, \ldots, \alpha_n)^T$. $\|A\alpha - b\|_\infty$ is a minimum if and only if

$$0 \in H\{\sigma_j A^j : j \in M\},$$

where $M = \{j : |r_j(\alpha)| = \|A\alpha - b\|_\infty\}$ and $A^j$ is the $j$th row of $A$.

The next result is usually stated as a theorem about inconsistent systems of equations (see Cheney [21, p. 36]).

Theorem 6.7 Let $g(x) = \sum_{i=1}^n c_i g_i(x)$ be a best Chebyshev approximation to $f$ on a compact metric space $X$.

Then there exists a finite subset $X_0$ of $X$, containing at most $n+1$ points such that $g$ is a best Chebyshev approximation to $f$ on $X_0$. If, in addition, \{g_1, \ldots, g_n\} satisfies the Haar condition, then $X_0$ contains exactly $n+1$ points.

Proof By theorem 6.5, $0 \in H(S)$, where $S = \{r(x)(g_1(x), \ldots, g_n(x)) : |r(x)| = \|r\|_\infty\}$. It now follows from Carathéodory's theorem (see Cheney [21, p. 17]) that we can find (at most) $n+1$ points, $x_1, \ldots, x_k \in X$.
(k ≤ n+1), so that 0 = \sum_{i=1}^{k} \theta_i r(x_i)(g_1(x_i), \ldots, g_n(x_i)) for some \( \theta_i \geq 0 \) and \( \sum \theta_i = 1 \). Hence

0 \in \text{H} \{r(x_i)(g_1(x_i), \ldots, g_n(x_i)): \|r(x_i)\| = \|r\|_\infty, \ i=1(1)k\}.

The result now follows, using theorem 6.5 in the opposite direction. If \{g_1, \ldots, g_n\} satisfies the Haar condition, we require \( k \geq n+1 \), i.e. \( k=n+1 \). //

It is clear from our previous discussion that theorem 6.7 covers inconsistent systems of equations. Thus if \( \alpha = (\alpha_1, \ldots, \alpha_n)^T \) is a minimax solution of the overdetermined mxn system \( A\alpha = b \), then \( \alpha \) is a minimax solution of a subsystem comprising at most \( n+1 \) equations. The subsystem has exactly \( n+1 \) equations if \( A \) satisfies the Haar condition. We finally obtain a discrete version of the alternation property (Theorem 6.1), which also applies to \( A\alpha = b \). In view of Theorem 6.7 we shall assume that \( X \) contains \( n+1 \) points.

**Theorem 6.8** Let \( f, g_1, \ldots, g_n \in C(X) \), where \( X = \{x_0, \ldots, x_n\} \) is a set of \( n+1 \) points in \([a,b]\). If \( \{g_1, \ldots, g_n\} \) satisfies the Haar condition, then

\[
g(x) = \sum_{i=1}^{n} c_i g_i(x)
\]

is a best uniform approximant to \( f \) if and only if there is an ordering \( x_{k_1} < \ldots < x_{k_{n+1}} \) of the points of \( X \) so that

\[
|g(x_{k_i}) - f(x_{k_i})| = \|g-f\|_\infty
\]

and \( g(x_{k_i}) - f(x_{k_i}) \) alternates in sign for \( i=1, \ldots, n+1 \).
Proof Let $g = (b_1, \ldots, b_{n+1})^T$, $g(x_{k_i}) = b_i$, $i=1, \ldots, n+1$.

Let $G(x) = a_0 + a_1 x + \ldots + a_n x^n$ be the interpolating polynomial of degree $n$ for the points $(x_{k_i}, b_i)$, $i=1, \ldots, n+1$.

Then $g = G$ on $X$ and $g$ is a best approximant to $f$ on $X$ if and only if $G$ is. The result now follows from theorem 6.1. //

The algorithm of chapter 7 is based on the fact that the error components of the best $L_\infty$ approximant agree in sign with those of the $L_2$ approximation. More precisely, we have the result stated below. First recall that a norm $|| \cdot ||$ on $R^n$ is called monotone if

$$|x_i| \leq |y_i| \quad (i=1, \ldots, n) \quad \text{implies} \quad ||x|| \leq ||y||.$$

All $L_p$ norms ($1 \leq p \leq \infty$) are monotone.

Theorem 6.9 The points of a hyperplane $H$ in $R^n$ which minimize two monotone norms have components of equal sign.

Proof See Cheney [21, p.40].

The following method of selecting a unique best of all best (or "strict") Chebyshev approximations is due to J.R.Rice [42]. Disregarding the $n+1$ components $r_i$ of equal maximal magnitude $||r||_\infty$, the maximum error of the remaining components is minimized. If necessary, the process is repeated.
Example 6.4  The minimax solution of the system

\[ \begin{align*}
  x_2 &= 0 \\
  x_2 &= 1 \\
  x_1 + x_2 &= 0
\end{align*} \]

are given by \((x_1, x_2) = (\lambda, \frac{1}{2}), \lambda \in [-1, 0]\), i.e. \(r_1 = 1/2, r_2 = -1/2, r_3 = \lambda + 1/2\). Since \(|r_3|\) is minimal for \(\lambda = -1/2\), \((x_1, x_2) = (-1/2, 1/2)\) is the required strict solution.

It was proved by J.Déscoux [60] that the strict approximation is the limit of the best \(L_p\) approximation as \(p \to \infty\) (Pólya's algorithm). A limitation of Rice's definition is that it only applies to finite point sets. Chapter 9 contains a definition of strict \(L_1\) approximation which can be extended to intervals. In this context, some results of discrete \(L_1\) approximation are required. The treatment below draws on material in the book by T.R. Rice [58, vol.1]; the proofs of lemmas 6.10 and 6.11 follow the line of reasoning used by Rice to establish the corresponding interval results.

Consider the following problem. The data points \((x_i, f(x_i)), i=1(1)m\), are to be approximated in the \(L_1\) norm by a function of the form

\[ L(A, x) = \sum_{i=1}^{n} a_i \phi_i(x), \]

where \(A\) denotes the unknown parameters \((a_1, \ldots, a_n)\).
I.e. we wish to minimize the function

$$\Delta_1(f, x) = \sum_{i=1}^{m} |f(x_i) - L(A, x_i)|,$$

which is equivalent to selecting a point \((a_1, \ldots, a_n, d)\) from the set

$$K = \{(A, d) \in \mathbb{R}^{n+1} : \Delta_1(f, A) \leq d\},$$

so that \(d\) is minimal. It is easy to see that \(K\) is convex: if \((A_1, d_1), (A_2, d_2) \in K\) and \(\lambda, \mu \geq 0\), with \(\lambda + \mu = 1\), then

$$\Delta_1(f, \lambda A_1 + \mu A_2) = \sum_{i=1}^{m} |f(x_i) - L(\lambda A_1 + \mu A_2, x_i)|$$

$$= \sum_{i=1}^{m} |(\lambda + \mu)f(x_i) - \lambda L(A_1, x_i) - \mu L(A_2, x_i)|$$

$$\leq \lambda \sum_{i=1}^{m} |f(x_i) - L(A_1, x_i)| + \mu \sum_{i=1}^{m} |f(x_i) - L(A_2, x_i)|$$

$$\leq \lambda d_1 + \mu d_2, \text{ i.e. } \lambda(A_1, d_1) + \mu(A_2, d_2) \in K.$$

We now define a plane \(H\) in \(\mathbb{R}^{n+1}\) by

$$H(g(x_i), a) = \{(A, d) : \sum_{i=1}^{m} L(A, x_i)g(x_i) = a - d\},$$

where \(a\) is the distance of \(H\) from the origin. Then

$$n = (\sum_{i=1}^{m} \phi_1(x_i)g(x_i), \ldots, \sum_{i=1}^{m} \phi_n(x_i)g(x_i), 1)$$

is a vector perpendicular to \(H\), since

$$(a_1, \ldots, a_n, d).n = \sum_{i=1}^{m} L(A, x_i)g(x_i) + d = a,$$

where "." denotes the inner product in \(\mathbb{R}^{n+1}\) and

\((a_1, \ldots, a_n, d)\) is any point in \(H\). \(H\) is called a plane of support of \(K\) at the boundary point \((A_o, d_o)\) of \(K\), if
Lemma 6.10 Let \( (A_0, d_0) \) be any point on the boundary of \( K \), i.e.
\[
\sum_{i=1}^{m} |L(A_0, x_i) - f(x_i)| = d_0.
\]
Then \( H(s(x_i), a_f) \) is a plane of support of \( K \) at \( (A_0, d_0) \), where
\[
s(x_i) = \text{sgn} \left[ f(x_i) - L(A_0, x_i) \right] \quad (6.6)
\]
and \( a_f = \sum_{i=1}^{m} f(x_i) s(x_i) \).

Proof Since \( d_0 = \sum_{i=1}^{m} |f(x_i) - L(A_0, x_i)| \)
\[
= \sum_{i=1}^{m} \left[ f(x_i) - L(A_0, x_i) \right] s(x_i),
\]
we have
\[
L(A_0, x_i) s(x_i) = a_f - d_0,
\]
i.e. \( (A_0, d_0) \in H(s(x_i), a_f) \). To show that \( K \subseteq H^+(s(x_i), a_f) \),

note that, if \( (A, d) \in K \), then
\[
d \geq \Delta_1(f, A) = \sum_{i=1}^{m} |L(A, x_i) - f(x_i)|
\]
\[
= \sum_{i=1}^{m} \left[ f(x_i) - L(A, x_i) \right] s(x_i)
\]
\[
\geq \sum_{i=1}^{m} \left[ f(x_i) - L(A, x_i) \right] s(x_i)
\]
\[
= a_f - \sum_{i=1}^{m} L(A, x_i) s(x_i).
\]
Hence
\[ \sum_{i=1}^{m} L(A, x_i)s(x_i) \geq a_f - d, \]
i.e. \((A, d) \in H^+(s(x_i), a_f). // \]

In the following discussion, the assumption is made that for all \( x_i \) there exists an \( L \), such that \( L(A, x_i) \neq 0 \). We define the sets
\[
X = \{x_1, \ldots, x_m\}, \\
X_0 = \{x_i \in X : \phi_j(x_i) = 0, j=1(1)n\}, \\
X_1 = X - X_0, \\
Z(A) = \{x_i \in X : f(x_i) - L(A, x_i) = 0\}, \\
Z_0(A) = \{x_i \in X_1 : f(x_i) - L(A, x_i) = 0\}. 
\]
The number of elements in any subset \( S \) of \( X \) will be denoted by \( v(S) \).

**Lemma 6.11**
\[ \Delta_1(f, A^*) \leq \Delta_1(f, A^* + tA), \text{ for all } t, \quad (6.7) \]
if and only if
\[ \sum_{X} L(A, x_i) \text{sgn}[f(x_i) - L(A^*, x_i)] \leq \sum_{Z_0(A^*)} |L(A, x_i)|. \quad (6.8) \]
Inequality (6.7) is strict for all \( t \neq 0 \), if inequality (6.8) is.

**Proof** \( \Longleftrightarrow \) Let \( s(x_i) = \text{sgn}[f(x_i) - L(A^*, x_i)] \) and \( s_t(x_i) = \text{sgn}[f(x_i) - L(A^*, x_i) - tL(A, x_i)] \). Then
\[ \Delta_1(f, A^* + tA) = \Delta_1(f, A^*) \]

- 71 -
\[
\sum_{x} \left[ \frac{f(x_{i}) - L(A^{*}, x_{i}) - tL(A, x_{i})}{s_{t}(x_{i})} \right] s(x_{i})
- \sum_{x} \left[ \frac{f(x_{i}) - L(A^{*}, x_{i})}{s(x_{i})} \right] s(x_{i})
= \sum_{x_{1} \sim Z_{0}(A^{*})} |tL(A, x_{1})| - \sum_{x_{1} \sim Z_{0}(A^{*})} [tL(A, x_{1})] s(x_{i})
+ \sum_{x_{1} \sim Z_{0}(A^{*})} |f(x_{i}) - L(A^{*}, x_{i}) - tL(A, x_{i})| - \sum_{x_{1} \sim Z_{0}(A^{*})} [f(x_{i}) - L(A^{*}, x_{i}) - tL(A, x_{i})] s(x_{i})
\]

(6.9)

The first difference on the R.H.S. of (6.9) is non-negative because of (6.8), the second difference is non-negative by definition of \( s(x_{i}) \), which proves inequality (6.7).

\( \rightarrow \) Suppose (6.8) is false. Then

\[
| \sum_{x} L(A, x_{i}) \text{sgn}[f(x_{i}) - L(A^{*}, x_{i})] | > \sum_{Z_{0}(A^{*})} |L(A, x_{i})|. \quad (6.10)
\]

Let \( E_{\varepsilon} = \{ x_{i} \in X : |f(x_{i}) - L(A^{*}, x_{i})| \leq \varepsilon \} \). Taking \( \varepsilon = tK \), we obtain

\[
\Delta_{1}(f, A^{*} + tA) - \Delta_{1}(f, A^{*})
= - \sum_{x} [tL(A, x_{1})] s(x_{i}) + \sum_{Z_{0}(A^{*})} |tL(A, x_{i})| + \sum_{E_{\varepsilon} \sim Z_{0}(A^{*})} [f(x_{i}) - L(A^{*}, x_{i}) - tL(A, x_{i})] [s_{t}(x_{i}) - s(x_{i})],
\]

where \( K = \max_{x} |L(A, x_{i})| \).

(6.11)

If \( x_{i} \in E_{\varepsilon} \sim Z_{0}(A^{*}) \), then

\[
|f(x_{i}) - L(A^{*}, x_{i}) - tL(A, x_{i})| \leq 3\varepsilon/2 = 3tK/2,
\]

i.e. the absolute value of the third \( \sum \)-term on the R.H.S. of (6.11) is bounded by \( (3/2)tKv(E_{\varepsilon} \sim Z_{0}(A^{*})) \).
Hence \( \Delta_1(f, A^* + tA) - \Delta_1(f, A^*) \)
\[= |t| \sum_{Z_0(A^*)} |L(A, x_i)| - t \sum_{X} L(A, x_i)s(x_i) + o(t) \quad (6.12) \]

If \( t \) and \( \sum_{X} L(A, x_i)s(x_i) \) have the same sign, it follows from (6.10) that the R.H.S. of (6.12) is negative for some small \( t \), contradicting (6.7).  //

**Theorem 6.12** \( L(A^*, x) \) is a best \( L^1 \) approximation to \( f(x) \) on \( X = \{x_1, \ldots, x_m\} \) if and only if

\[ |\sum_{X} L(A, x_i) \text{sgn}[f(x_i) - L(A^*, x_i)]| \leq \sum_{Z(A^*)} |L(A, x_i)| \]

for all \( A \).  \( (6.13) \)

\( L(A^*, x) \) is unique if inequality (6.13) is strict.

**Theorem 6.12** follows immediately from the preceding lemma. We are now in a position to prove the main result of this section.

**Remark** Let \( K \) be a convex set. Recall that a point \( k \) in \( K \) is said to be an extreme point of \( K \) if it cannot be expressed as a convex combination of two other points in \( K \).

**Theorem 6.13** Let \( \{\phi_1(x), \ldots, \phi_n(x)\} \) satisfy the Haar condition. Then the set \( P_M(f) \) of best \( L_1 \) approximants from \( M = \langle \phi_1, \ldots, \phi_n \rangle \) to \( f \) on \( X = \{x_1, \ldots, x_m\} \) is a closed convex set. The extreme points of \( P_M(f) \) are the best \( L_1 \) approximants to \( f \) for which \( \nu(Z(A^*)) \geq n \).

**Proof** Let \( L(A_1, x), L(A_2, x) \in P_M(f) \), \( \alpha + \beta = 1 \) and \( \alpha, \beta \geq 0 \).
Then \[ \sum_{X} |f(x_i) - L(aA_1 + bA_2, x_i)| \leq \alpha \sum_{X} |f(x_i) - L(A_1, x_i)| + \beta \sum_{X} |f(x_i) - L(A_2, x_i)|, \]
which shows that \( P_\alpha(f) \) is convex. If \( \lim A_k = A_0 \), then by the continuity of \( L(A_k, x) \) and of the \( L_1 \) norm,
\[ \lim \sum_{k} |f(x_i) - L(A_k, x_i)| = \sum |f(x_i) - L(A_0, x_i)|, \]
which shows that \( P_M(f) \) is closed. To prove the second part of the theorem, suppose \( \forall(Z(A^*)) = k < n \). By the Haar condition, there exists an approximant \( L \) such that \( L(A_0, x_i) = 0 \) for \( x_i \in Z(A^*) \). Let
\[ M = \max_{X} |L(A_0, x_i)| \quad \text{and} \quad \varepsilon = \min_{X \sim Z(A^*)} \left( |f(x_i) - L(A^*, x_i)| \right). \]
If \( |t| < \varepsilon/(2M) \), then
\[ \text{sgn} [f(x) - L(A^* + tA_0, x)] = \text{sgn} [f(x) - L(A^*, x)]. \] (6.14)
It now follows from theorem 6.12 that, if \( L(A^*, x) \in P_M(f) \), then (6.13) is satisfied. Using (6.14), we replace \( \text{sgn} [f(x) - L(A^*, x)] \) in (6.13) by \( \text{sgn} [f(x) - L(A^* + tA_0, x)] \) and deduce that \( L(A^* + tA_0, x) \in P_M(f) \). We similarly show that \( L(A^* - tA_0, x) \in P_M(f) \). Since
\[ L(A^*, x) = \frac{1}{2}L(A^* + tA_0, x) + \frac{1}{2}L(A^* - tA_0, x), \]
\( L(A^*, x) \) is not an extreme point of \( P_M(f) \). //

We restate theorem 6.13 in a form which will be used in chapter 9.
Corollary 6.13 The set $P_M(f)$ of theorem 6.13 is the convex hull of best approximations which interpolate $f$ in at least $n$ points of $X$.

In particular, the parameters $a$ and $b$ of a best linear $L_1$ approximation $ax+b$ to $m$ data points $(x_i, y_i), i=1(1)m,$ form a two-dimensional set whose extreme points interpolate the data in at least two points.

As might be expected from the convexity properties of the $L_1$ norm, best $L_1$ approximants are not necessarily unique, but uniqueness can be guaranteed by imposing additional conditions either on the norm or the approximating functions. Uniqueness via the first method is the subject of chapter 9. It is not known whether the second method is feasible in the discrete case. As for interval approximation, the hypothesis which guarantees uniqueness of best $L_\infty$ approximants also works for $L_1$ approximants. This result was proved by D.Jackson [61] in 1921, three years after the publication by Haar of the corresponding $L_\infty$ result. The proof given below follows E.W.Cheney [62].

Lemma 6.14 Let $r,g \in C[a,b]$. If $r$ has a finite number of zeros in $[a,b]$ and
\[
\int_a^b g(x) \text{sgn} r(x) dx \neq 0,
\]
then there exists a real number $\lambda$ such that
\[
\int_a^b |r(x) - \lambda g(x)| dx < \int_a^b |r(x)| dx.
\]
Proof Let \( x_1, \ldots, x_k \in (a, b) \) be zeros of \( r \). Choose \( \varepsilon > 0 \) sufficiently small so that

\[
I = [a + \varepsilon, x_1 - \varepsilon] \cup \cdots \cup [x_k + \varepsilon, b - \varepsilon]
\]

consists of \( k + 1 \) disjoint closed intervals. Let

\[
J = [a, b] - I
\]

and assume w.l.o.g. that \( \int_a^b g \operatorname{sgn} r > 0 \).

For \( \varepsilon > 0 \) sufficiently small,

\[
\int_I g \operatorname{sgn} r \, dx > \int_J |g| \, dx. \tag{6.15}
\]

Since \( I \) is closed and contains no zeros of \( r \),

\[
\delta = \min_{x \in I} |r(x)| > 0.
\]

If \( M = \max_{a \leq x \leq b} |g(x)| \) and \( 0 < \lambda < \delta/M \), then

\[
|\lambda g(x)| < \delta \leq |r(x)| \quad \text{for all } x \in I.
\]

Now let \( x \in I \). If \( r(x) > 0 \), then \( \lambda |g(x)| < r(x) \),

i.e. \( 0 < r(x) - \lambda g(x) \).

If \( r(x) < 0 \), then \( \lambda |g(x)| < -r(x) \),

i.e. \( r(x) - \lambda g(x) < 0 \).

Hence \( \operatorname{sgn}[r(x) - \lambda g(x)] = \operatorname{sgn} r(x) \) for all \( x \in I \).

It follows that

\[
\int_a^b |r - \lambda g| \, dx
\]

\[
= \int_I (r - \lambda g) \operatorname{sgn} r \, dx + \int_J |r - \lambda g| \, dx
\]

\[
= \int_I |r| \, dx - \lambda \int_I g \operatorname{sgn} r \, dx + \int_J |r - \lambda g| \, dx
\]

\[
= \int_a^b |r| \, dx - \lambda \int_I g \operatorname{sgn} r \, dx - \int_J |r| \, dx + \int_J |r - \lambda g| \, dx
\]

\[
= \int_J (|r - \lambda g| - |r|) \, dx + \int_a^b |r| \, dx - \lambda \int_I g \operatorname{sgn} r \, dx
\]

- 76 -
Theorem 6.15 Let $M = \langle f_1, \ldots, f_n \rangle$ be an $n$-dimensional subspace of $C[a, b]$. If $M$ satisfies the Haar condition, then it is a Chebyshev subspace.

Proof In view of theorem 1.2, we only have to prove that $M$ is semi-Chebyshev. Suppose $g_1, g_2$ are two best approximants to $f \in C[a, b]$. Since $P_M(f)$ is convex, $g_0 = (g_1 + g_2)/2$ is also a best approximant. Hence

$$\int (|f - g_0| - |f - g_1|/2 - |f - g_2|/2)dx = 0.$$ 

Since the integrand is non-positive and continuous on $[a, b]$, it must equal the zero function, i.e. $|f - g_0| = |f - g_1|/2 + |f - g_2|/2$.

If $f - g_0$ has $m$ zeros, with $m \geq n$, then $f - g_1, f - g_2$ and $g_1 - g_2$ have the same $m$ zeros. Hence $g_1 = g_2$ by the Haar condition.

Now suppose $r = f - g_0$ has at most $n-1$ zeros. These will be a subset of the $n+1$ points $a = x_0 < x_1 < \ldots < x_n = b$. Take any $g = \sum a_i f_i \in M$ and let

$$\int_{x_{i-1}}^{x_i} g \, dx = \phi_i(g).$$

Then for suitably chosen $\sigma_i = 0$ or $\pm 1$,

$$\int_a^b g \, \text{sgn} \, r \, dx = \sum_{i=1}^n \sigma_i \int_{x_{i-1}}^{x_i} g \, dx = \sum_{i=1}^n \sigma \phi_i(g) = 0.$$
For if this expression did not vanish, the lemma would give
\[ \int |r - \lambda g| \, dx < \int |r| \, dx, \]
contradicting the definition of \( r \). In particular,
\[ \sum_{i=1}^{n} \sigma_i \phi_i (f_j) = 0, \text{ i.e. the matrix } (\phi_i(f_j)) \text{ and its} \]
transpose are singular. Hence there exist scalars \( \beta_1, \ldots, \beta_n \) (not all zero) so that
\[ \sum_{i=1}^{n} \beta_i \phi_j (f_i) = 0, \]
i.e. for the non-zero function \( h = \sum_{i=1}^{n} \beta_i f_i \), we have
\[ 0 = \phi_i (h) = \int_{x_{i-1}}^{x_i} h \, dx. \]
Hence \( h \) has \( n \) roots, contradicting the Haar condition. //

The algorithm in chapter 7 is based on the fact that a
best \( L_2 \) approximant satisfies the alternating sign property,
if not the equal error property, of theorem 6.1. This result
seems to be due to E. Stiefel [63]. A generalization to \( L_p \)
polynomial approximation can be found in the book by Werner
[39]. In the version given below, the result is extended to
generalized polynomial approximants satisfying the Haar
condition.

**Lemma 6.16** Let \( f, g_i \in C[a,b] \), \( i=1(1)n \).

Suppose \( L(A,x) = \sum_{i=1}^{n} a_i g_i(x) \) and \( \{g_1, \ldots, g_n\} \)
satisfies the Haar condition on \([a,b]\). Set
\[ \Delta_p(A) = \left[ \int_a^b |f(x) - L(A,x)|^p \, dx \right]^{1/p} \quad (1 \leq p < \infty) \]  

and assume that \( f \) is not a generalized polynomial.

Then \( \Delta_p(A) \) is continuously differentiable and the best \( L_p \) approximant \( L(A^*,x) \) to \( f \) is given by the system

\[ \frac{\partial \Delta_p(A)}{\partial a_i} = \]

\[ \left[ \Delta_p(A) \right]^{1-p} \int_a^b |f-L(A,x)|^{p-1} \text{sgn} \left[ L(A,x) - f(x) \right] \cdot g_i(x) \, dx = 0 \]  

(6.17)

**Proof**

\[ \frac{\partial}{\partial a_i} |f-L(A,x)|^p \]

\[ = \frac{\partial}{\partial a_i} \left[ |f-L(A,x)|^p \text{sgn} \left[ f-L(A,x) \right] \right] \]

\[ = \begin{cases} p |f-L(A,x)|^{p-1} \text{sgn} \left[ f-L(A,x) \right], & \text{if } f \neq L(A,x) \\ 0, & \text{if } f = L(A,x). \end{cases} \]

But \( p |f-L(A,x)|^{p-1} \text{sgn} \left[ f-L(A,x) \right] \rightarrow 0 \) as \( f \rightarrow L(A,x) \),

i.e. \( |f-L(A,x)|^p \) is continuously differentiable.

Differentiating under the integral sign, we find that \( \Delta_p(A) \) is also continuously differentiable. Equation (6.17) now follows as a necessary condition. By the convexity of the set

\[ K = \{(A,d) \in \mathbb{R}^{n+1} : \Delta_p(A) \leq d \}, \]

the parameter \( A^* \) defined by (6.17) must be a minimum.

Since the set \( \{g_1, \ldots, g_n\} \) satisfies the Haar condition it is linearly independent, which ensures the existence of
a solution for the system (6.17). //

**Theorem 6.17** The error function \( f - L(A^*, x) \) of the best \( L_p \) approximant \( L(A^*, x) \) defined in lemma 6.6 changes sign at least \( n \) times.

**Proof** Let \( \mathbf{v} \in \mathbb{R}^n \) be an arbitrary unit vector. Since \( \Delta_p(A^*) \) is minimal,

\[
\frac{\partial \Delta_p(A)}{\partial \mathbf{v}} = 0 \quad \text{for} \quad A = A^*.
\]

Hence

\[
\left[ \Delta_p(A^*) \right]^{1-p} \int_{a}^{b} |f - L(A^*, x)|^{p-1} \text{sgn} [L(A^*, x) - f] \cdot L(h, x) dx = 0. \quad (6.18)
\]

Since \( \Delta_p(A^*) \neq 0 \) by hypothesis,

\[
\int_{a}^{b} |f - L(A^*, x)|^{p-1} \text{sgn} [L(A^*, x) - f] L(h, x) dx = 0. \quad (6.19)
\]

Now suppose that \( f - L(A^*, x) \) changes sign \( m \) times, with \( m < n \). Then we can find a generalized polynomial

\[
L(b, x) = \sum_{i=1}^{n} b_i g_i(x)
\]

in \([a, b]\) as \( f - L(A^*, x) \). Hence the function

\[
[L(A^*, x) - f] \sum_{i=1}^{n} b_i g_i(x)
\]

does not change sign in \([a, b]\), contradicting (6.19). It follows that \( f - L(A^*, x) \) changes sign at least \( n \) times. //
II. NUMERICAL APPLICATIONS
Chapter 7

A Modified Exchange Algorithm for Best Chebyshev Approximation

Given \( n \) data points \((x_i, y_i), i = 1(1)n\), a line \( y = ax + b \) is to be determined so that the norm of \( r = (r_1, \ldots, r_n) \) is a minimum, where

\[ r_i = ax_i + b - y_i. \]

An \( L_1, L_2 \) or \( L_\infty \) line is obtained according as the norm is defined by

\[ \|r\|_1 = \|r\|, \quad \|r\|_2 = \left( \sum r_i^2 \right)^{\frac{1}{2}} \quad \text{or} \quad \|r\| = \max |r_i|. \]

These methods are maximum likelihood for the double exponential, the Gaussian and the uniform distribution, respectively. The \( L_\infty \) norm can be used if the data are thought not to contain any outliers. The \( L_1 \) norm, on the other hand, is the least sensitive to outliers and gives good results if some of the data points are suspect. In order to ensure the uniqueness of \( L_2 \) and \( L_\infty \) approximations we assume that the \( x_i \) are distinct.

In certain applications especially to the social sciences, little or nothing is known about the underlying error distribution, and the customary compromise of choosing the \( L_2 \) norm can lead to inappropriate results. There is, therefore, a need for adaptive regression packages, which allow the user to experiment with different norms and all possible solutions. Such a package could, for example, include a facility for "robust" economic forecasting, by computing a band of \( L_p \) approximations and deducing
upper and lower bounds for each forecast. These bounds are given by the parameters YMAX and YMIN of subroutine EXTRAP in the appendix of programs. The output parameters ICODE and JCODE indicate the norm used to obtain YMAX and YMIN, respectively. If the outlying points are wildly inaccurate, the $L_1$ approximation should provide the most accurate forecasts. If, on the other hand, the outliers herald a new trend, then the $L_\infty$ approximation can be expected to yield better results. Similar adaptive packages could be designed for the solution of inconsistent linear systems. A possible application for such a package is outlined in chapter 10.

There are also computational advantages in this unified approach, since the amount of arithmetic involved in obtaining the $L_1$ and $L_\infty$ lines can be substantially reduced by using the $L_2$ line as an initial estimate. We briefly describe the $L_\infty$ theory.

It is well known (see theorem 6.17) that a best polynomial $L_p(p>1)$ approximation satisfies the alternating sign property. Thus in the linear case these are points $P_k^*, P_\lambda^*, P_m$ such that

$$\text{sgn}(r_k^* r_{\lambda}^*) = \text{sgn}(r_{\lambda}^* r_m) = -1.$$ (7.1)

If these points are chosen so that the absolute errors are as large as possible, one or two exchange iterations will normally suffice to obtain the $L_\infty$ line. As in example 7.1 below, the $L_2$ line frequently leads
immediately to the required solution and the exchange method need not be applied.

With \( P_k, P_l, P_m \) defined as above, we determine the equal error line through the points \((x_k, y_k + e), (x_l, y_l - e), (x_m, y_m + e)\). The resulting system

\[
ax_k + b - (y_k + e) = 0
\]
\[
ax_l + b - (y_l - e) = 0
\]
\[
ax_m + b - (y_m + e) = 0
\]

has a non-trivial solution if

\[
e = \frac{[(x_k - x_l)(y_l - y_m) + (x_m - x_l)(y_k - y_l)]}{2(x_k - x_m)}.
\]

(7.2)

The required equal error line has the equation

\[
(y - y_k - e)(x_m - x_k) = (y_m - y_k)(x - x_k).
\]

(7.3)

If \( \max|r_i| = e \), the equal error line is also the required \( L_\infty \) line. Otherwise there exists an integer \( M \), \( 1 \leq M \leq n \), such that \( \max|r_i| = r_M \). \( x_M \) now replaces one of the \( x_k, x_l, x_m \) in such a way that the resulting triple satisfies the alternating sign property (7.1). The process can be shown to terminate in a finite number of steps when \( \max|r_i| = e \).

Example 7.1 The \( L_2 \) line for the 31 data points in Table 7.1 is given by

\[
y = 0.370565x + 0.054435.
\]

We note from the table that \( r_9, r_{24}, r_{30} \) is an alter-
nating error triple with maximum absolute values. We therefore choose the initial reference $P_9$, $P_{24}$, $P_{30}$. Using (7.2) and (7.3), the error line is found to have the equation

$$y = \frac{8}{21}x - \frac{6}{21}, \text{ with } e = -\frac{39}{21}.$$

Since $\max|r_i| = e$, this is also the required $L_\infty$ line, i.e. the exchange method is not needed. If, on the other hand, $P_0$, $P_1$, $P_2$ are chosen as the initial reference, three exchange iterations are required to compute the solution (see Scheid [32, p.271]).

<table>
<thead>
<tr>
<th>$x_i$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_i$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0.1</td>
<td>-0.6</td>
<td>-0.2</td>
<td>-0.8</td>
<td>0.5</td>
<td>-1.1</td>
<td>0.3</td>
<td>0.7</td>
<td>0.1</td>
<td>-1.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x_i$</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_i$</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>$r_i$</td>
<td>0.8</td>
<td>0.1</td>
<td>-0.5</td>
<td>0.9</td>
<td>0.2</td>
<td>-0.4</td>
<td>-0.0</td>
<td>1.4</td>
<td>-0.3</td>
<td>1.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x_i$</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_i$</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>12</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>$r_i$</td>
<td>-0.5</td>
<td>0.8</td>
<td>1.2</td>
<td>0.6</td>
<td>2.0</td>
<td>0.3</td>
<td>-1.3</td>
<td>0.1</td>
<td>1.6</td>
<td>-0.2</td>
<td>-1.8</td>
</tr>
</tbody>
</table>

SUBROUTINE MINMAX(N,X,Y,ITER,ERROR, A,B,C,D) in the appendix is a FORTRAN IV version of the modified exchange method described above. If double precision is required, the REAL declaration should be changed to DOUBLE PRECISION, E to D and ABS to DABS. The formal parameters are as follows
In Table 7.2, the running time (in seconds) of a double precision version of MINMAX is compared with that of CHEB, an LP-based subroutine due to Barrodale and Phillips [33]. The 31 points refer to example 1 above, the 201 and 1001 points are given by \( y = e^x \), with \( x = 0(0.01)2 \) and \( 0(0.01)10 \), respectively. The figures for MINMAX include CPU time for the \( L_2 \) lines. A flowchart for the subroutine is given below.

<table>
<thead>
<tr>
<th>Number of points</th>
<th>CHEB</th>
<th>MINMAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>0.03</td>
<td>0.01</td>
</tr>
<tr>
<td>201</td>
<td>0.19</td>
<td>0.06</td>
</tr>
<tr>
<td>1001</td>
<td>0.85</td>
<td>0.33</td>
</tr>
</tbody>
</table>

(My attention has just been drawn by a referee to a recent algorithm by Sklar and Armstrong [71], which appears to be about 5 times faster than Barrodale and Phillips.) See note in pocket.
Flowchart for subroutine MINMAX

START

1. Compute \( L_2 \) line

2. Compute errors \( e_i \) of \( L_2 \) line and determine \( k \) so that \( \max |e_i| = |e_k| \)

3. Is \( e_k = e_1 \)?
   - YES: \( 1 + i_1 \)
     - select \( i_2 \) and \( i_3 \) so that \( e(i_2), e(i_3) \) are numerically maximal and satisfy alternating sign property
   - NO: \( k + i_2 \)
     - select \( i_1 \) and \( i_3 \) as in 4

4. \( n + i_3 \)
   - select \( i_1 \) and \( i_2 \) as in 4

5. Is \( e_k = e_n \)?
   - YES: compute equal error line for points with indices \( i_1, i_2, i_3 \)
   - NO: compute errors \( e_i \) and \( k \) so that \( \max |e_i| = |e_k| \)

6. Is \( \max |e_i| = |e_k| \)?
   - YES: STOP
   - NO: Exchange \( (x_k, y_k) \) and a point from the old triple so that the new triple has alternating signs
Example 7.2 demonstrates that the modified exchange technique of subroutine MINMAX can also be used to obtain best approximating polynomials of higher degree.

**Example 7.2**  
Find the minimax parabola \( y=ax^2+bx+c \) for the points \( P_1(0,0), P_2(0.25,0.015625), P_3(0.5,0.125), P_4(0.75,0.421875), P_5(1,1) \).

Taking \( P_1P_2P_3P_4 \) as the initial reference, four iterations are required to obtain the required parabola \( y=1.5x^2-0.5625x+0.03125 \). The subsequent references are \( P_2P_3P_4P_5, P_3P_4P_5, P_1P_2P_4P_5 \).

Alternatively, we first compute the \( L_2 \) parabola \( y=1.5x^2-0.5375x+0.01875 \). By inspection of the error vector

\[
\mathbf{r} = (0.01875, -0.0375, 0, 0.0375, -0.01875)
\]

\( P_1P_2P_4P_5 \) is chosen as initial reference, which gives the required answer in only one iteration. To obtain the \( L_\infty \) parabola by the first method takes twice as long as computing both \( L_2 \) and \( L_\infty \) parabolas by the second method. //

Finally note that in the continuous case, the \( L_2 \) error function can be analyzed in a similar way in order to obtain good starting values for an iterative exchange method such as the second algorithm of Remes. The \( L_2 \) method is computationally more expensive than the usual technique of taking the values of \( x \) which maximize \( |T_{n+1}| \) (\( n \) is the order of the approximant and \( T_{n+1} \) the Chebyshev polynomial of degree \( n+1 \)). However, when
the approximated function is odd or even, the $L_2$ method gives better results as the following example shows.

**Example 7.3** Minimizing

$$\int_{-1}^{1} \left[ ax^3 + bx^2 + cx + d - \sin(\pi x/2) \right]^2 dx$$

in the usual way, we find $a = -0.562228$, $b = 0$, $c = 1.553191$, $d = 0$. Searching the error function

$$r(x) = -0.562228x^3 + 1.553191x - \sin(\pi x/2)$$

for maximal absolute values, the reference

$$\{-0.8, -0.3, 0.3, 0.8, 1\}$$

(7.4)

is obtained. Alternatively,

$$x_i = \cos[(i-1)\pi/4], \ i = 1(1)5,$$

defines the initial reference $\{-1, -0.7, 0, 0.7, 1\}$. The next two references are $\{-0.9, -0.4, 0.4, 0.9, 1\}$ and $\{-0.8, -0.3, 0.3, 0.8, 1\}$, i.e. two Remes iterations are needed before reference (7.4) is reached. //
Chapter 8

Segmented Linear Chebyshev Approximation

Segmented approximation provides useful initial estimates for fast and efficient techniques of computing function values. It remains an open question whether there is a general finite-step method of constructing the best approximating polynomial for a given continuous function. In the linear case, such a method exists for a restricted class of functions. The single-variable case is discussed in Natanson [34, p.34 f.], where it is proved that, if a function $f$ can be differentiating twice and if $f''$ does not alter its sign for $a \leq x \leq b$, then the best linear Chebyshev approximation $g(x) = Ax + B$ over the interval $[a, b]$ is given by

$$A = \frac{f(b) - f(a)}{(b-a)} = f'(c) \quad (8.1)$$
$$B = \frac{[f(a) + f(c)]}{2} - \frac{A}{2} \left( \frac{a+c}{2} \right) \quad (8.2)$$

for some $c \in (a, b)$. We prove a slightly stronger version of Natanson's result.

Theorem 8.1  If $f$ is a strictly convex function which is differentiable on $(a, b)$ and continuous on $[a, b]$, then $g(x) = Ax + B$ as defined by (8.1) and (8.2) is the best linear approximation to $f$.

To prove the theorem we establish the existence of a number $c \in (a, b)$ such that

$$\frac{[f(b) - f(a)]}{(b-a)} = f'(c)$$

- 90 -
by the mean value theorem. Now let \( L \) be the line parallel to and equidistant from the tangent to \( f \) at \( P(c, f(c)) \) and the chord through the points \( Q(a, f(a)) \), \( R(b, f(b)) \). \( L \) is determined by its gradient \( f'(c) \) and the midpoint \( \frac{1}{2}(a + c, f(a) + f(c)) \) of \( PQ \), i.e. its equation is

\[
y = f'(c)x + \frac{[f(a) + f(c)]}{2} - f'(c)(a+c)/2,
\]

which agrees with (8.1) and (8.2). Since \( f \) is convex, the maximum error occurs with alternating signs at \( x = a,c,b \). Its absolute value is

\[
|f(a) - f(c) - f'(c)(a-c)|/2.
\]

To see that the theorem is stronger than Natanson’s result we note that the function

\[
f(x) = \begin{cases} 
  x^2, & -1 \leq x \leq 0 \\
  x^3, & 0 \leq x \leq 1
\end{cases}
\]

is convex and differentiable on \([-1,1]\) but \( f''(0) \) does not exist.

Before considering a generalization of the above theorem to functions of several variables, we briefly consider an application to computer approximation. (An earlier version of the ideas set out below can be found in M. Planitz[35]). The following square root routine for the now extinct Hewlett-Packard 2000F computer has appeared, without explanations, in Unit 10 of Numerical Computation [36], an Open University text on approximation theory. The process of evaluating \( \sqrt{x} \) is carried out in
four steps:

(i) Determine a real number \( t \in [0.25, 1] \), such that

\[ x = 4^k t, \text{ where } k \text{ is an integer.} \]

(ii) Use the formula

\[
y(t) = \begin{cases} 
0.27863 + 0.875t, & t \in [0.25, 0.5) \\
0.421875 + 0.578125t, & t \in [0.5, 1) 
\end{cases}
\]

to obtain a first approximation for \( \sqrt{t} \).

(iii) Apply Newton's method in the form

\[
y_{n+1} = \frac{y_n + t/y_n}{2}
\]

with \( y_0 = y(t) \) and \( n = 0, 1 \).

(iv) Compute \( \sqrt{x} = 2^ky_2 \).

This algorithm, which seems cumbersome at first sight, is in fact remarkably efficient. The result is correct to 6 significant figures, and a binary computer requires only 2 "long" operations (i.e. multiplications or divisions). These are needed to compute \( t/y_n \) in step (iii). Steps (i) and (iv) as well as the division by 2 in step (iii), only involve shifts. Less obviously, step (ii) can be regarded as a "short" operation, since

\[ 0.875 = 0.111_2 \text{ and } 0.578125 = 0.100101_2, \text{ i.e. only 4 additions and 3 shifts are required to find } y(t). \]

The selection of the function \( y(t) \) for step (ii) poses an interesting non-trivial problem. First note that for greater accuracy, the approximation on \( [0.25, 1] \) is segmented. Since our computer uses binary arithmetic, a power of 2 is chosen as a point of sub-division.
follows from theorem 6.2, that there is a unique best linear approximation to $\sqrt{t}$ on each of the two subintervals. It is not clear how Hewlett-Packard arrived at the formula in (ii), but the following approach leads to similar, in fact slightly better, results. We first use (8.1) and (8.2) to determine the best segmented approximant

$$y^*(t) = \begin{cases} 
0.297335 + 0.828427t, & t \in [0.25, 0.5) \\
0.420495 + 0.585786t, & t \in [0.5, 1), 
\end{cases}$$

with approximate errors of 0.004 on [0.25, 0.5) and 0.006 on [0.5, 1). Some of the accuracy of $y^*$ is now sacrificed in order to reduce the execution time of step (ii). This is done by approximating the coefficients of $t$ by numbers whose binary expansions contain only three non-zero bits. The resulting formula is

$$y(t) = \begin{cases} 
a_0 + 0.875t, & t \in [0.25, 0.5) \\
b_0 + 0.578125t, & t \in [0.5, 1). 
\end{cases}$$

To adjust the value of $a_0$ we apply theorem 6.1 to the function

$$g(t) = \sqrt{t} - 0.875t.$$

This time the required best approximation is a constant and a simple argument will show that this constant is given by

$$a_0 = (m + M)/2,$$

where $m = \min g(t)$ and $M = \max g(t)$ on [0.25, 0.5]. Since $a_0$ has degree 0, we have to show that the error function alternates on two points. If we define $t_1, t_2$ by
m = g(t_1) and M = g(t_2), then
\[ a_o - g(t_1) = (M - m)/2 \quad \text{and} \quad a_o - g(t_2) = (m - M)/2. \]

Moreover,
\[ |a_o - g(t_i)| = \max|a_o - g(t)|, \quad i=1 \text{ or } 2, \quad t \in [0.25, 0.5], \]
i.e. \( a_o = (m + M)/2 \) satisfies the alternation property of theorem 6.1. It is now easy to show that

\[ m = 0.269 \, 068 \quad \text{and} \quad M = 0.285 \, 714 \, 3. \]

Hence \( a_o = 0.277 \, 661 \). This gives a maximum absolute error of 0.008 on \([0.25, 0.5]\), compared with an error of 0.009 in Hewlett-Packard's original formula. We similarly find \( b_o = 0.425 \, 008 \) with an error of 0.007, which reduces Hewlett-Packard's error by 0.003. Thus the formula in (ii) should be replaced by

\[
y(t) = \begin{cases} 
0.277 \, 661 + 0.875t, & t \in [0.25, 0.5) \\
0.425 \, 008 + 0.578 \, 125t, & t \in [0.5, 1).
\end{cases}
\]

A further reduction in the number of long operations could be achieved by introducing a k-fold segmented approximation to \( \sqrt{t} \), with \( k > 2 \), and applying the above technique to each of the \( k \) subintervals. The decreasing costs of integrated circuit technology have now made it economically feasible to save CPU time by permanently installing a large number of constants in read-only memory chips. If \( k \) is sufficiently large, step (iii) can be eliminated and execution times should approach those of a single multiplication, even for transcendental functions which at present are still computationally expensive.
We now derive a generalization of theorem 8.1 to functions of two variables. Let the strictly convex function \( f \) be differentiable on the open rectangle \( S = (a,b) \times (c,d) \) and continuous on the corresponding closed rectangle \( \bar{S} \). A best approximation

\[
g(x,y) = Ax + By + C \tag{8.3}
\]
to \( f \) on \( \bar{S} \) clearly exists. As to uniqueness, we know from the remarks following theorem 6.2 that the Haar theory does not automatically carry over to multivariate approximation. However, for the special case of linear polynomial approximants we have the following result due to L. Collatz [38].

**Theorem 8.2** If \( f \) has continuous partial derivatives at all interior points of a closed, strictly convex set \( X \) of the plane, then there exists a unique linear polynomial \( Ax + By + C \) of best approximation to \( f \) on \( X \).

In the book by J.R. Rice [58, Vol.II, p.237], theorem 8.2 appears with the weaker hypothesis that \( X \) is closed and convex. To disprove this version, consider the convex (but not strictly convex) function

\[
f(x,y) = (2y^2 - 1)(1 - x/2),
\]
with \( 0 \leq x \leq 1 \) and \( -1 \leq y \leq 1 \). Then \( X \) is convex (but not strictly convex) and \( g(x) = kx/2 \) is a best approximant to \( f \) for any \( k \) such that \( |k| \leq 1 \).
Let $L$ be the best approximating plane (8.3). If 
\((x, y, f(x, y)), (x+h, y+k, f(x+h, y+k))\) are points on the
intersection of $L$ with the surface $z=f(x, y)$, then
\[f(x+h, y+k) - f(x, y) = Ah+Bk = (h, k).(A, B).\]
By the mean-value theorem, this expression is equal to
\[(h, k).\nabla f(x+\theta h, y+\theta k)\]
for $0<\theta<1$, i.e. there exists a point $(\alpha, \beta)$
\[= (x+\theta h, y+\theta k) \in S, \text{ such that}\]
\[\nabla f(\alpha, \beta) = (A, B).\]
Thus the tangent plane $T$ to $z=f(x, y)$ at $P(\alpha, \beta, f(\alpha, \beta))$
is parallel to $L$. Since $f$ is strictly convex, $P$ is at
maximum distance from the best approximation $L$, and the
point $(\alpha, \beta)$ must be a minus-point, i.e. a point with
negative maximum error
\[f(\alpha, \beta) - A\alpha - B\beta - C.\]

Now let $L$ be parallel to and equidistant from the
tangent plane $T$ and a third plane $U$, say. By definition
of $U$, none of the points $P_1(a, c, f(a, c)), P_2(b, c, f(b, c)), P_3(b, d, f(b, d)), P_4(a, d, f(a, d))$ lie above $U$. Now suppose
they all lie below $U$. Then there exists a plus-point
\((x_p', y_p')\) i.e. a point with positive maximum error, which
is not one of the $Q_i$, where $Q_i$ denotes the projection
of $P_i$ onto the $xy$-plane. Suppose \((x_p', y_p')\) lies on
the boundary of $S$ between $Q_1$ and $Q_2$, say. Then
\((x_p', y_p', f(x_p', y_p'))\) lies above the chord $P_1P_2'$. 

- 96 -
contradicting the convexity of $f$. Suppose next $(x_p, y_p) \in S$ and draw a line from $Q_1$, say, through $(x_p, y_p)$. If this line meets the boundary of $S$ at $(x_b, y_b)$, then $(x_p, y_p, f(x_p, y_p))$ lies above the chord from $P_1$ to $(x_b, y_b, f(x_b, y_b))$, again contradicting the convexity of $f$. It follows that at least one of the $P_i$ lies in $U$. We next prove that at least two more of the points $P_i$ lie in $U$. First recall the following

**Definition 8.1** A set of points $M \subset S$ will be called a reference of $z = Ax + By + C$ if there is no triple $(D, E, F)$ so that

$$(Dx + Ey + F)[f(x) - Ax - By - C] > 0$$

for all $x \in M$, i.e. there is no plane $z = Dx + Ey + F$ whose sign on $M$ agrees with that of the error $f(x) - Ax - By - C$. The reference is said to be a **Chebyshev alternant** if for all $x \in M$

$$|f(x) - Ax - By - C| = \|f(x) - Ax - By - C\|_\infty.$$ 

We also require the following result.

**Theorem 8.3** $z = Ax + By + C$ is a best Chebyshev approximation to $f$, if and only if there is a Chebyshev alternant.

For a proof of this theorem, see for example Werner [39, p.141]. If only one of the $P_i$ ($P_1$, say) lies in $U$, we can clearly determine a plane whose sign is positive for $Q_1$ and negative for $(\alpha, \beta)$, contradicting the assumption that the two points form an
alternant. The same argument shows that the only possible constellations with two of the $P_i$ in $U$ are $P_1, P_3$ (or $P_2, P_4$), with $Q_1, Q_3$ (or $Q_2, Q_4$) as plus-points and the minus-point $(\alpha, \beta)$ on the diagonal $Q_1 Q_3$ (or $Q_2 Q_4$), leading to non-unique approximations. Now assume that there are at least three points in $U$ ($P_1, P_2, P_3$, say). To determine $L$, note that, if $z = Ax + By + N$ is the equation of $U$, then
\[
\begin{align*}
  f(a, c) &= Aa + Bc + N \\
  f(b, c) &= Ab + Bc + N \\
  f(b, d) &= Ab + Bd + N.
\end{align*}
\]
Hence
\[
A = \frac{[f(b, c) - f(a, c)]}{(b-a)} = f_x(\alpha, \beta) \tag{8.4}
\]
and
\[
B = \frac{[f(b, d) - f(a, c)]}{(d-c)} - \frac{A(b-a)}{(d-c)} = f_y(\alpha, \beta) \tag{8.5}
\]
Since $U$ is given by $z = Ax + By + f(a, c) - Aa - Bc$, the vertical distance between $U$ and $P(\alpha, \beta, f(\alpha, \beta))$ is given by
\[
d = A\alpha + B\beta + f(a, c) - f(\alpha, \beta) - Aa - Bc.
\]
But $L$ is equidistant from $T$ and $U$. Hence
\[
d/2 = A\alpha + B\beta + C - f(\alpha, \beta).
\]
Eliminating $d$ from these expressions we find
\[
C = \frac{\{f(a, c) + f(\alpha, \beta) - A(a+\alpha) - B(c+\beta)\}}{2}. \tag{8.6}
\]
Equations (8.4), (8.5), (8.6) define $L$. The maximum error is given by
\[ e = \frac{f(a,c) - f(a,b) - A(a-a) - B(c-b))}{2}. \quad (8.7) \]

**Example 8.1** Determine the best approximation of the form

\[ z = Ax + By + C \]

to the function

\[ f(x,y) = x^2 + 6y^2 + 4x - 8y - 143 \]

on the unit square \([0,1]^2\).

The points \(P_1(0,0, -143), P_2(1,0, -138), P_3(1,1, -140),\)

\(P_4(0,1, -145)\) all lie in the plane \(U\) whose equation is

\[ z = 5x - 2y - 143. \]

We have \(A = 5, B = -2, \alpha = \beta = 1/2,\) and \(C = -143.875.\)

\(L\) is given by

\[ z = 5x - 2y - 143.875, \]

with \(e = 0.875.\) //
Chapter 9

Strict Approximation in the $L_1$ Norm

We return in this chapter to the problem of linear approximation to a set of data $(x_i, y_i)$, $i = 1(1)n$. In contrast to the $L_2$ line and $L_\infty$ line, an $L_1$ line need not be unique even if the $x_i$ are distinct. In fact, if $ax+b$ and $a'x + b'$ are two best linear $L_1$ approximations, then so is any convex combination

$$\alpha(ax+b) + \beta(a'x+b')$$

for $\alpha, \beta \geq 0$ and $\alpha + \beta = 1$. This follows from the inequality

$$\sum |\alpha(ax_i+b) + \beta(a'x_i+b') - y_i|$$

$$\leq \alpha \sum |ax_i+b-y_i| + \beta \sum |a'x_i+b'-y_i|.$$

Thus, if there is more than one $L_1$ solution, then there are infinitely many. The purpose of this chapter is to develop an algorithm, which determines this infinite solution set and then selects a unique "best" of all best solutions by minimizing $\|r\|_2$ over the $L_1$ solution set.

We restrict our attention to the non-unique case and assume that an $L_1$ line

$$L(A_1, x) = a_1x + b_1$$

has been obtained, using the subroutine $L1$ by Barrodale and Roberts [40] or any other suitably adapted LP-based package. The subroutine SOLVE (see appendix of programs) is then activated to compute the remaining simplex vertices which represent optimal solutions. Denote these solutions
by \( L(A_2, x), \ldots, L(A_n, x) \). We know from chapter 6 that any convex combination of the form

\[
L(A, x) = \alpha_1 L(A_1, x) + \ldots + \alpha_n L(A_n, x),
\]

(9.1)

with \( \alpha_i \geq 0 \) and \( \sum \alpha_i = 1 \), is also an \( L_1 \) solution and that the locus of all solution parameters \( A(a, b) \) is the convex hull \( H \) of the points \( A_1(a_1, b_1), \ldots, A_n(a_n, b_n) \).

**Example 9.1** For the data points \((0,1), (1,0), (2,0), (3,1)\), the LP method yields four interpolating \( L_1 \) lines: \( y = 0, y = 1, y = 0.5x - 0.5, y = -0.5x + 1 \), with \( \sum |r_i| = 2 \). The set \( H \) is the quadrilateral whose vertices are \((-0.5,1), (0,1), (0.5,-0.5), (0,0)\). //

Contrary to Sadovski's [41, p. 245] claim that the \( L_1 \) norm fit must pass through at least two data points, we note from the example that \( y = 0.5 \) is an \( L_1 \) line which misses all four data points. \( y = 0.5 \) is also an \( L_\infty \) and \( L_2 \) line and clearly satisfies the additional requirement that \( \| r \|_2 \) should be minimal on \( H \). We shall refer to this line as a strict \( L_1(L_2) \) approximation. The term "strict approximation" was first used by J.R. Rice [42] to denote a unique "best" of all best Chebyshev approximations. An exchange algorithm to determine the strict Chebyshev approximation can be found in the paper by Duris and Temple [43].

Example 9.1 is exceptional in that the strict \( L_1(L_2) \) approximation coincides with the \( L_2 \) and \( L_\infty \) approximations. In general, the problem is to minimize the function
$f(a, b) = \Sigma (ax_i + b - y_i)^2,$

subject to the constraint $(a, b) \in \mathbb{H}$. We proceed as follows.

The subroutine STRICT first computes the (unique) $L_2$ line $y = cx + d$ and determines whether the point $(c, d)$ lies in $\mathbb{H}$. Two cases arise:

(i) $(c, d) \in \mathbb{H}$. Then $y = cx + d$ is clearly the required strict $L_1(L_2)$ approximation and the algorithm stops (see example 9.1).

(ii) $(c, d) \not\in \mathbb{H}$. Then $f$ has its global minimum at $(c, d)$. But the convex function $f$ has a positive-definite Hessian matrix

$$
\begin{pmatrix}
2\Sigma x_i^2 & 2\Sigma x_i \\
2\Sigma x_i & 2n
\end{pmatrix},
$$

since for distinct $x_i$, $n\Sigma x_i^2 - (\Sigma x_i)^2 > 0$ by Hölder's inequality. It follows that its constrained minimum is unique and occurs on the boundary of $\mathbb{H}$ at $(a, b)$, say (see example 9.2).

In either case, STRICT returns a unique $L_1(L_2)$ approximation.

Example 9.2 For the data points $(0, 2), (1, 2.5), (2, 2), (3, 5)$, the convex hull $\mathbb{H}$ has vertices $(1.25, 1.25), (1, 2), (0.5, 2)$. The $L_2$ line is given by $y = 0.85x + 1.6$, the point $(0.85, 1.6)$ lies outside $\mathbb{H}$ and the required strict approximation is $y = 0.83x + 1.6$. //

We now give a description of the subroutines SOLVE and STRICT. Subroutine SOLVE is preceded by a driver program.
which computes an optimal simplex tableau $A$ and the initial optimal $L_1$ solution $(a_1, b_1)$. The idea of using linear programming methods to obtain an $L_1$ approximation for discrete data is due to H.M.Wagner [44]. The LP method is based on the following theory. Set

$$r_i = ax_i + b - y_i, \quad a = \alpha_1 - \alpha_2, \quad b = \beta_1 - \beta_2,$$

where $\alpha_1, \alpha_2, \beta_1, \beta_2 \geq 0$. In order to minimize $\sum |r_i|$, put $r_i = v_i - u_i$, with $u_i, v_i > 0$. To ensure non-singularity of the basis matrix, $u_i$ and $v_i$ may not both be present in the basis. It follows that $u_i v_i = 0$ and hence

$$|r_i| = |v_i - u_i| = (u_i^2 + 2u_i v_i + v_i^2)^{\frac{1}{2}} = u_i + v_i.$$

Thus the problem can be restated in the form

$$\sum (u_i + v_i) = \min,$$

subject to the constraints

$$y_i = \beta_1 - \beta_2 + (\alpha_1 - \alpha_2)x_i + u_i - v_i,$$

$i=1(1)n$. In the subroutine SOLVE and its driver program, a numerical code is used to identify the variables: the numbers $1, 2, 3, \ldots, n+2$ denote the variables $\alpha_1, \beta_1, u_1, \ldots, u_n$, respectively; $-1, -2, -3, \ldots, -n-2$ denote $\alpha_2, \beta_2, v_1, \ldots, v_n$, respectively. The efficiency of the driver program can be improved by combining the method used in chapter 7 with linear programming techniques. As in subroutine MINMAX, we first compute the $L_2$ line and then use the errors $r_i$ to estimate the position of two interpolating points of the $L_1$ line. The following strategy will be employed:
if \(|r_j|, |r_k| \) are the smallest absolute \(L_2\) errors with \(\text{sgn}(r_j r_k) \leq 0\), we apply two LP iterations to ensure that the line goes through the points \((x_j, y_j), (x_k, y_k)\). This step corresponds to phase I of subroutine \(L1\) by Barrodale and Roberts [40]. When the interpolation step is complete, we continue with the usual simplex method or apply phase II of the Barrodale-Roberts algorithm. For the straight line to interpolate \((x_j, y_j)\) and \((x_k, y_k)\) we remove \(u_j, u_k\) from the basis without allowing \(v_j, v_k\) to enter. Any negative entries in column \(y\) are made positive by multiplying the appropriate rows by \(-1\) and making the corresponding \(u,v\)-interchanges. The data in the example below appear in Barrodale and Roberts [46].

Example 9.3 For the points \((1,1), (2,1), (3,2), (4,3), (5,2)\) we find the \(L_2\) line \(y = 0.4x + 0.6\). On inspection of the error vector

\[
\mathbf{r} = (0, 0.4, -0.2, -0.8, 0.6),
\]

\((1,1)\) and \((3,2)\) are chosen as interpolation points. \(u_1\) and \(u_3\) will therefore be removed from the basis. The condensed tableaux are as follows. (Pivots are indicated by asterisks.)

\[
\begin{array}{cccc}
\text{basis} & y & \beta_1 & \alpha_1 \\
\hline
u_1 & 1 & 1 & 1 \\
u_2 & 1 & 1 & 2 \\
u_3 & 2 & 1 & 3^* \\
u_4 & 3 & 1 & 4 \\
u_5 & 2 & 1 & 5 \\
\hline
& 9 & 5 & 15
\end{array}
\]

\[
\begin{array}{cccc}
\text{basis} & y & \beta_1 & u_3 \\
\hline
u_1 & 1/3 & 2/3^* & -1/3 \\
v_2 & 1/3 & -1/3 & 2/3 \\
\alpha_1 & 2/3 & 1/3 & 1/3 \\
u_4 & 1/3 & -1/3 & -4/3 \\
v_5 & 4/3 & 2/3 & 5/3 \\
\hline
& 7/3 & 2/3 & -1/3
\end{array}
\]

\[
\begin{array}{cccc}
\beta_1 & 1/2 & 3/2 & -1/2 \\
v_2 & 1/2 & 1/2 & 1/2 \\
\alpha_1 & 1/2 & -1/2 & 1/2 \\
u_4 & 1/2 & 1/2 & -3/2 \\
v_5 & 1 & -1 & 2 \\
\hline
& 2 & -1 & 0
\end{array}
\]
Note that pivot $3^*$ was chosen by applying the usual criteria of the simplex method to rows 1 and 3 of the first tableau. The Barrodale-Roberts technique is computationally more expensive: starting with the usual simplex pivot 5, $\alpha_1$ is increased until the marginal cost becomes negative. In the above example, the two methods give rise to identical tableaux, which seems fairly typical of small data sets. //

Subroutine SOLVE is summarized in the macroscopic flowchart below; the formal parameters are as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Input/Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDIM</td>
<td>Integer</td>
<td>input</td>
</tr>
<tr>
<td>A</td>
<td>Real array (IDIM,6)</td>
<td>input: optimal simplex tableau</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_{1,5}$ = residuals, $i=1(1)n$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_{n+1,j}$ = marginals, $j=1(1)5$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_{i,6}$ = basis identifiers, $i=1(1)n$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_{n+2,j}$ = variable identifiers, $j=1(1)4$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_{n+2,5}=1, a_{n+2,6}=0, a_{n+1,6}=0$</td>
</tr>
<tr>
<td>X</td>
<td>Real array (2)</td>
<td>input: initial optimal solution</td>
</tr>
<tr>
<td>TOLER</td>
<td>Real</td>
<td>input: $1.0 \cdot 10^{-D}$, where $D$ is the number of accurate decimal digits available</td>
</tr>
<tr>
<td>IFAIL</td>
<td>Integer</td>
<td>output: fault indicator equal to</td>
</tr>
<tr>
<td>ISC</td>
<td>Integer</td>
<td>output: number of solutions</td>
</tr>
<tr>
<td>T</td>
<td>Real array (2,10)</td>
<td>output: solutions $T(1,I) = a_i, T(2,I) = b_i$.</td>
</tr>
</tbody>
</table>
Flowchart for subroutine SOLVE

START

Initialization of T, IBASIS, TEMP, AA, AD
A(I,J) ← AA(1,I,J), I=1,...,ID1M, j=1,...,6
A(M1,5) ← ERROR, 1 ← KAA, 0 ← KAD

1

2

1 → IC, 1 → IR

3

Does A(IR,IC) qualify as positive-entry pivot?

4

NO

IR + 1 → IR

5

YES

Is IR ≤ M?

6

NO

IC + 1 → IC

7

YES

Is IC ≤ 4?

8

YES

7

NO

NO

3

- 106 -
YES Does $A(IR, IC)$ qualify as zero-entry pivot? 

IF $IR + 1 \leq M$ 

YES NO

IF $IC + 1 \leq 4$ 

YES NO

IF $IDEN = 1$ 

YES NO

ICOUNT + 1 \rightarrow ICOUNT

IF $ICOUNT \leq KAD$ 

YES AD(ICOUNT, J, K) \rightarrow A(J, K) 

J = 1, ..., IDIM, K = 1, ..., 6

YES STOP

NO

YES KAA = 0? 

NO

0 \rightarrow KAD, 1 \rightarrow IDEN 

1 \rightarrow ICOUNT

AA(ICOUNT, J, K) \rightarrow A(J, K) 

J = 1, ..., IDIM, K = 1, ..., 6

YES

NO

STOP
ICOUNT+1 → ICOUNT

Is I COUNT ≤ KAA? YES → 22, NO → 24

Is KAD=0? YES → 23, NO → 25

0 → KAA, 0 → IDEN, 1 → I COUNT

AD(I COUNT, J, K) + A(J, K)
J=1, ..., IDIM, K=1, ..., 6

Use non-zero entry pivot to compute new tableau

Ensure column 5 is non-negative

Compute marginal costs

Is basis new? YES → 30, NO → 28

IBC+1 → IBC

Has sum of absolute errors increased? YES → 32, NO → 33

Compute solution

Is solution new? YES → 34, NO → 35

Use zero-entry pivot to compute new tableau

Ensure column 5 is non-negative

Compute marginal costs

Is basis new? YES → 39, NO → 38

IBC+1 → IBC

- 108 -
10 Store new solution in T

11 Is IFLAG = 0?

11 NO KAA+1 → KAA

11 43 Store new tableau in AA: TEMP(J,K) → AA(KAA,J,K)

11 J=1,...,IDIM, K=1,...,6

11 44 YES KAD+1 → KAD

11 45 Store new tableau in AD: TEMP(J,K) → AD(KAD,J,K)

11 J=1,...,IDIM, K=1,...,6

11 46

Remarks (Box numbers are indicated on the left.)

3 For positive-entry pivots the usual simplex criteria apply; only the minimum-ratio rule is disregarded.

9 Zero-entry pivots do not lead to new solutions, but subsequent tableaux may do so.

27-34 Non-zero entry pivoting.

35-39 Zero entry pivoting.

18, 24 KAA, KAD are counters for the tableaux stored in AA, AD respectively.

44, 46 Pivots from tableaux in AA lead to new tableaux which are stored in AD. When the AA-pivots have been exhausted, the process is reversed and any new tableaux are stored in AA. The algorithm terminates when no new pivots and tableaux can be found.

Subroutine STRICT is preceded by a driver program.
which uses a standard wrapping technique to determine the convex hull \( H \) of the solution points \((a_i, b_i), i=1(1)ISC\), computed by SOLVE. Suppose the vertices \(A, B, C\) have been found in that order. Let \( \alpha \) be the angle of \(CB\) from the horizontal (see FIG. 9.1). The next point \(P\) is chosen so that the angle \(CP\) from the horizontal is a minimum. To avoid the inclusion of interior points such as \(E\), we ignore angles not greater than \(\alpha\). We also ignore angles not less than \(2\pi\). In FIG. 9.1, the next point found in this way is \(D\). Intermediate points such as \(F\) are eliminated. Since the point sets encountered in the present context are small, no attempt has been made to include a "quicksort" technique, but a machine-dependent improvement in running time of about 30% was achieved for the driver program by avoiding the function \(\text{ATAN}\). Instead of measuring the angle by \(\text{ATAN}(Y/X)\), where \(X\) and \(Y\) are the horizontal and vertical steps between consecutive vertices, the "angle" is defined by

\[
0.5 \times \pi Y / (\text{ABS}(X) + \text{ABS}(Y)),
\]

which preserves the ordering of angles.

Subroutine STRICT first computes the \(L_2\) solution \((c, d)\) for the given data points \((x_i, y_i)\) and then determines whether \((c, d) \in H\). This is done by considering
the intersections of the sides of $H$ with the line segment
defined by $(c,d)$ and the centroid $(x_0,y_0)$ of the vertices
of $H$. If double precision is required, the REAL declaration
should be changed to DOUBLE PRECISION, E to D and
ABS to DABS in either subroutine. In addition, SIGN
becomes DSIGN in SOLVE, and FLOAT(.) becomes
DBLE(FLOAT(.)) in STRICT. The formal parameters of
STRICT are as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IHC</td>
<td>Integer</td>
<td>input : number of vertices $(a_i, b_i)$</td>
</tr>
<tr>
<td>HULL</td>
<td>Real array</td>
<td>input : vertices $(a_i, b_i), i = 1(1)IHC$</td>
</tr>
<tr>
<td>M</td>
<td>Integer</td>
<td>input : number of data points $(x_i, y_i)$</td>
</tr>
<tr>
<td>T1</td>
<td>Real array</td>
<td>input : $T1(i) = x_i, i = 1(1)m$</td>
</tr>
<tr>
<td>T2</td>
<td>Real array</td>
<td>input : $T2(i) = y_i, i = 1(1)m$</td>
</tr>
<tr>
<td>TOLER</td>
<td>Real</td>
<td>input : as for subroutine MINMAX</td>
</tr>
<tr>
<td>ICODE</td>
<td>Integer</td>
<td>output : indicates status of solution;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 strict and $L_2$ solution are identical;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 otherwise</td>
</tr>
<tr>
<td>A</td>
<td>Real</td>
<td>output : gradient of strict $L_1$ line</td>
</tr>
<tr>
<td>B</td>
<td>Real</td>
<td>output : intercept of strict $L_1$ line</td>
</tr>
<tr>
<td>C</td>
<td>Real</td>
<td>output : gradient of $L_2$ line</td>
</tr>
<tr>
<td>D</td>
<td>Real</td>
<td>output : intercept of $L_2$ line</td>
</tr>
</tbody>
</table>

If $(c,d) \in H$, this is also the strict solution and the
exit code will be set to 0. If $(c,d) \notin H$, the subroutine
determines analytically the minimum of

$$f(a,b) = \sum (ax_i + b - y_i)^2$$  \hspace{1cm} (9.2)
on the boundary of $H$. Consider the side with endpoints $(a_j, b_j), (a_{j+1}, b_{j+1})$. The line through these points is given by

$$ b = g(a - a_j) + b_j, $$

where $g = (b_{j+1} - b_j)/(a_{j+1} - a_j), a_{j+1} \neq a_j$. Hence (9.2) becomes

$$ f(a) = \Sigma(ax_i + ag - ajg + b_j - y_i)^2. $$

From $f'(a)$ we find

$$ a = \frac{\Sigma[g(a_j x_i + a_j g - b_j + y_i) - b_j x_i + x_i y_i]}{\Sigma(x_i^2 + 2gx_i + g^2)} $$

If $a_j = a_{j+1}$, put $a = a_j$ in (9.2). Then

$$ f(b) = \Sigma(a_jx_i + b - y_i)^2, $$

and $f'(b) = 0$ gives

$$ b = \Sigma(y_i - a_jx_i)/d. $$

In either case, a check is made to ensure that the point $(a, b)$ lies between $(a_j, b_j)$ and $(a_{j+1}, b_{j+1})$. The local minima found in this way compete with the values of $f(a, b)$ at the vertices of $H$ to determine the global minimum on the boundary.

Note that strict $L_1$($L_2$) approximations can also be defined for continuous approximants as the following example shows.
Example 9.4 The function

\[ f(x) = \begin{cases} 
0, & -1 \leq x \leq 2 \\
-1, & 2 < x \leq 3 
\end{cases} \]

has infinitely many best \( L_1 \) approximants of the form \( g(x) = ax + b \). These are given by

\[ g(x) = tx, \quad -\frac{1}{2} \leq t \leq 0. \]

To determine a strict approximation we minimize

\[ F(t) = \int_{-1}^{3} \left[ f(x) - tx \right]^2 \, dx, \]

subject to the constraint \( -\frac{1}{2} \leq t \leq 0 \). From \( F'(t) = 0 \), \( t = -15/52 \), i.e. the required strict approximation is \( y = (-15/52)x \). //
It is well known that the general solution of a linear system

\[ Ax = b \]  \hspace{1cm} (10.1)

is given by

\[ x = A^\# b + (I - A^\# A)w, \]  \hspace{1cm} (10.2)

where \( w \) is an arbitrary vector in \( \mathbb{R}^n \) and \( A^\# \) is any generalized inverse of the \( m \times n \) matrix \( A \). If the coefficient matrix \( A \) contains inaccurate measurements or observations, we may find there is no solution, i.e. there is no vector \( x \) such that \( Ax - b = 0 \). It then seems natural to consider the following modification of the original problem: choose \( x \) such that \( ||Ax - b|| \) is a minimum. The most elegant result is obtained if we interpret \( ||\cdot|| \) as the Euclidean norm, because \( x \) then has the same form as the general solution of the consistent system. Thus (10.2) represents the general solution if the system (10.1) is consistent and the best approximation if it is inconsistent. As has been observed before, the \( L_2 \) solution of an inconsistent linear system is not necessarily unique. However, if \( A^\# \) is interpreted as the Moore-Penrose inverse, then \( x = A^\# b \) is the unique vector of smallest Euclidean norm minimizing \( ||Ax - b||_2 \). (For a proof, see for example M. Planitz [48, p.183].)
A large number of physical and technological applications lead to inconsistent linear systems. Such an application is the problem of balancing the input streams (feeds) and output streams (products) of a mineral processing plant. (Some of the material of this chapter has appeared in the paper by Voller, Planitz and Reid [47].) Following the article by Wiegel [49] in 1972, a number of computer packages have been designed, which determine material balances from sets of inconsistent measurements. A survey of existing packages can be found in the paper by K.J. Reid [50]. Although most of these packages have been designed for mainframe computers, more recently attention has focused on microcomputers implementations.

The purpose of this chapter is to compare existing techniques for the solution of the fundamental material balance problem and to propose alternatives, with particular reference to microcomputer implementation. We consider a single processing unit with a feed stream (1) and two product streams (2) and (3) as shown in fig.10.1.

FIG.10.1
It will be assumed that each stream has been assayed for \( n \) distinct species. We calculate the mass flow in each stream for the material in the processing unit to balance. This is usually done by obtaining a best least squares solution for the following overdetermined system of \( n+1 \) equations:

\[
M_1 = M_2 + M_3, \tag{10.3}
\]

\[
M_1 x_1^k = M_2 x_2^k + M_3 x_3^k, \quad k=1(1)n, \tag{10.4}
\]

where \( M_i \) denotes the mass flow rate in stream \( i \) and \( x_i^k \) the assayed percent value of species \( k \) in stream \( i \). Eliminating \( M_3 \) from (10.3) and (10.4) gives the so-called two-product balance formula

\[
M_2 = M_1 (x_1^k - x_3^k)/(x_2^k - x_3^k). \tag{10.5}
\]

The data in the tables below demonstrate that in practice it is not feasible to use (10.5) in order to determine \( M_2 \). Table 10.1 contains a typical set of inconsistent measured assays. Given that \( M_1 = 1 \), we use (10.5) and (10.3) to obtain the corresponding values of \( M_2 \) and \( M_3 \) shown in table 10.2.

<table>
<thead>
<tr>
<th>( i )</th>
<th>( x_1^1 )</th>
<th>( x_1^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23.8</td>
<td>52.1</td>
</tr>
<tr>
<td>2</td>
<td>5.3</td>
<td>40.7</td>
</tr>
<tr>
<td>3</td>
<td>53.9</td>
<td>63.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( k )</th>
<th>( M_2 )</th>
<th>( M_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6193</td>
<td>0.3807</td>
</tr>
<tr>
<td>2</td>
<td>0.4978</td>
<td>0.5022</td>
</tr>
</tbody>
</table>

We therefore modify our problem in the following way.
Denote the unknown exact value of species \( k \) in stream \( i \) by \( \hat{x}_i^k \) and replace the \( x_i^k \) by \( \hat{x}_i^k \) in equation (10.4). In order to minimize the error in the least squares sense, we require, subject to the constraints (10.3) and (10.4) that

\[
J_2 = \sum_{k=1}^{n} J^k = \min \, ,
\]

where \( J^k = \sum_{j=1}^{3} w^k_j (\hat{x}_j^k - x_j^k)^2 \) and \( w_j^k \) is a suitable weighting factor. On defining a relative mass flow

\[
D = \frac{M_2}{M_1} \, ,
\]

the \( n+1 \) constraints reduce to \( n \) constraints

\[
\hat{x}_1^k = D \hat{x}_2^k + (1-D) \hat{x}_3^k
\]

There are various ways of solving the problem defined by (10.6) and (10.7). In packages designed for the minerals industry, methods ranging from Lagrange multipliers to direct search techniques have been employed (see Mular [51]). Most solutions start by introducing Lagrange multipliers \( \lambda^k \), combining (10.6) and (10.7) into a single auxiliary function

\[
L = J_2 + \sum_{k=1}^{n} \lambda^k (\hat{x}_1^k - D \hat{x}_2^k - (1-D) \hat{x}_3^k) .
\]

This approach has been used in a number of large mineral processing material balance packages. These packages then employ a variety of methods to minimize (10.8). Wiegel [49], Cutting [52], and Laguitton and Wilson [53] use a gradient
method deriving a set of non-linear equations, which are solved by a linearizing iterative technique. Smith and Ichiyen [54] and Hockings and Callen [55] also employ the gradient method, but combine it with a search over the independent relative mass-flows in the circuit. Hodouin and Everall [56] employ a hierarchical procedure in which the problem is decomposed and a combination of gradient, search, and Newton methods are adopted for maximum efficiency.

Setting the partial derivatives of $L$ to zero and re-writing the constraint equations (10.7), we obtain the following $4n+1$ equations:

$$2w^k_j (\hat{x}^k_j - x^k_j) - g_j^x^k = 0, \quad (10.9a)$$

$$\sum_{k=1}^{n} \lambda_j^k (\hat{x}_3^k - \hat{x}_2^k) = 0, \quad (10.9b)$$

$$\sum_{j=1}^{3} g_j^x \hat{x}_j^k = 0, \quad (10.9c)$$

where $g_1 = -1$, $g_2 = D$, $g_3 = 1-D$. In terms of $D$, equations (10.9a) and (10.9c) give

$$\hat{x}_j^k = x_j^k + g_j^r r^k / (w_j^k h^k), \quad (10.10)$$

where

$$r^k = x_1^k - Dx_2^k - (1-D)x_3^k \quad (10.11)$$

is called the residue or imbalance equation and

$$h^k = 1/w_1^k + D^2/w_2^k + (1-D)^2/w_3^k. \quad (10.12)$$
On substitution of (10.10) into (10.9b), the following polynomial in $D$ is obtained:

$$\sum_{k=1}^{n} \left( \frac{r^k}{h^k} \right) \left( x_2^k - x_3^k + \frac{r^k}{h^k} \left( \frac{D}{w^k_2} - \frac{(1-D)}{w^k_3} \right) \right) = 0 \quad (10.13)$$

Solving (10.13) iteratively, by Newton's method for example, will give the value of $D$ which minimizes $L$. The corresponding adjusted assays are then obtained from equation (10.10). This method will be referred to as "LMP" for Lagrange Multiplier Polynomial method.

An alternative method in Voller, Planitz, Reid [47] consists of minimizing

$$J_2^* = \sum_{k=1}^{n} w^* k (r^k)^2, \quad (10.14)$$

where $w^* = 1/h^k$. $\partial J_2^*/\partial D = 0$ leads back to equation (10.13) and the LMP method. If, on the other hand, $w^* k$ is treated as a constant by choosing an estimate for $D$ in $h^k$ (via equation (10.5), for example), $\partial J_2^*/\partial D = 0$ gives

$$D = \frac{\sum_{k=1}^{n} w^* k (x_2^k - x_3^k)(x_1^k - x_3^k)}{\sum_{k=1}^{n} w^* k (x_2^k - x_3^k)^2}. \quad (10.15)$$

The values for $D$ given by (10.15) are substituted into (10.10) in order to obtain the adjusted assay values $\hat{x}_j^k$. This method will be referred to as "MWR" for minimum of weighted residues method. It has obvious computational advantages over LMP, but requires field trials to establish whether it is sufficiently accurate for practical purposes.
Alternatively, we can use penalty functions, an optimization technique which has not yet been employed in the solution of material balance problems. Introducing a large positive constant \( K \), we now minimize

\[
L^p_2 \equiv J_2 + K \sum_{k=1}^{n} \left( \hat{x}_1^k - D \hat{x}_2^k - (1-D)\hat{x}_3^k \right)^2
\]

The constant \( K \) ensures that in the minimization of \( L^p_2 \), selections of \( D \) and \( \hat{x}_j^k \) which violate the mass balance constraints are penalized. The usual gradient method for minimizing \( L^p_2 \) gives

\[
\hat{x}_j = x_j + \left( \frac{g_j r^k}{w_j} \right) K \left( 1 + Kh^k \right),
\]

for the calculation of adjusted assay values, and

\[
\sum_{k=1}^{n} Kr^k/(1+Kh^k) \left( x_2^k - x_3^k + Kr^k/(1+Kh^k)(D/w_2^k - (1-D)/w_3^k) \right) = 0
\]

for the calculation of \( D \). For large \( K \), equations (10.17) and (10.18) give values for \( D \) and \( \hat{x}_j^k \) which are close to those obtained via (10.10), (10.13). Thus for the simple stream process unit, the two methods are roughly equivalent. In the solution of larger problems, the penalty function approach requires further investigation.

The above methods are all gradient methods involving derivatives. In contrast, the flowchart below outlines a hierarchical direct search routine for the solution of the material balance problem defined by equations (10.6) and (10.7). This method will be referred to as "DSM" for direct search method. Steps 1 and 3 were carried out using the Powell quadratic interpolation technique (see G.R. Walsh [57]).
The data of table 10.1 have been reproduced in table 10.3, adding typical percentage standard deviations, $\sigma_{ij}^k$, associated with the measurements. The weights $w_{ij}^k$ are inversely proportional to the $(\sigma_{ij}^k)^2$.

<table>
<thead>
<tr>
<th>TABLE 10.3</th>
<th>TABLE 10.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$x_1^i$</td>
</tr>
<tr>
<td>1</td>
<td>23.8</td>
</tr>
<tr>
<td>2</td>
<td>5.3</td>
</tr>
<tr>
<td>3</td>
<td>53.9</td>
</tr>
</tbody>
</table>
### TABLE 10.5

<table>
<thead>
<tr>
<th>Input</th>
<th>LMP</th>
<th>LMS</th>
<th>MWR</th>
<th>DSM</th>
<th>ASSAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>23.8</td>
<td>23.85</td>
<td>23.89</td>
<td>23.85</td>
<td>23.89</td>
<td>Type 1</td>
</tr>
<tr>
<td>5.3</td>
<td>5.29</td>
<td>5.30</td>
<td>5.29</td>
<td>5.30</td>
<td></td>
</tr>
<tr>
<td>53.9</td>
<td>53.88</td>
<td>53.87</td>
<td>53.88</td>
<td>53.87</td>
<td></td>
</tr>
<tr>
<td>52.1</td>
<td>49.94</td>
<td>49.96</td>
<td>49.95</td>
<td>49.96</td>
<td>Type 2</td>
</tr>
<tr>
<td>40.7</td>
<td>41.51</td>
<td>41.51</td>
<td>41.51</td>
<td>41.51</td>
<td></td>
</tr>
<tr>
<td>63.4</td>
<td>63.59</td>
<td>63.59</td>
<td>63.59</td>
<td>63.59</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE 10.6

<table>
<thead>
<tr>
<th>CPU time in seconds</th>
<th>LMP</th>
<th>LMS</th>
<th>MWR</th>
<th>DSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.6</td>
<td>0.1</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

| Number of BASIC lines | 40 | 45 | 35 | 70 |

The four algorithms considered above were coded in BASIC. The results of a comparison between these algorithms are summarized in tables 10.4-10.6. The values of $D$ and $x_k^j$ were compatible with those obtained from the mainframe package MATBAL by R.L. Wiegel [49]. For the simple material balance problem, MWR is clearly superior both in CPU time and number of BASIC lines. This is an interesting result, since none of the existing packages use this approach. The BASIC code (MINBAL) for LMS can be found in the appendix of programs. As might be expected, the direct search method (DSM) emerges as the least efficient of the four algorithms. It is unlikely that a more sophisticated search technique would alter the order of merit.

From our results, the MWR method looks promising,
and the development along these lines of a full-scale microcomputer package for more complicated processing units seems worthwhile. Such a package could also incorporate adaptive features as suggested in chapter 7. As an example of the use of alternative adjustment criteria, the values of

\[ J_1 = \sum_{k=1}^{n} |r^k| \quad \text{and} \quad J_\infty = \max_k |r^k| \quad (10.20) \]

have been minimized, using the test data in table 10.3 to compare various best approximations for the relative mass flow rate \( D \). In table 10.7, these approximations are compared with the values of \( D \) obtained by minimizing the weighted sum of squares \( J^*_2 \) and unweighted sum of squares \( J^*_2 \).

<table>
<thead>
<tr>
<th>Adjustment criterion</th>
<th>( J_1 )</th>
<th>( J^*_2 )</th>
<th>( J_2 )</th>
<th>( J_\infty )</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimizing value of ( D )</td>
<td>0.6324</td>
<td>0.6181</td>
<td>0.5975</td>
<td>0.5806</td>
</tr>
</tbody>
</table>

The results of table 10.7 indicate that the values of \( D \) derived from minimizing \( J_1 \) and \( J_\infty \) define upper and lower bounds for least squares solutions.
Chapter 11

An Algorithm for Alternative Optimal and Sub-Optimal Solutions in Integer Programming

Standard packages for integer linear programming, such as algorithm H02BAF in the N.A.G. library and algorithm 263A in the C.A.C.M. collection, are based on Gomery's cutting plane method and enhanced by a technique known as Wilson's cuts. The purpose of this chapter is to develop an algorithm, which allows the user to search for alternative optimal solutions and for sub-optimal solutions, e.g. all second best solutions, and to solve certain two-stage optimization problems.

More precisely, we wish to determine non-negative integers \( x_1, \ldots, x_n \) such that
\[
f(x) = c_1 x_1 + \ldots + c_n x_n
\] (11.1)
is a minimum (or maximum), subject to linear constraints of the form
\[
A \mathbf{x} \leq b
\] (11.2)
where \( \mathbf{x} = (x_1, \ldots, x_n)^T \) and \( A \) is an \( m \times n \) matrix. There may also be secondary constraints, e.g. \( \| \mathbf{x} \| = \min \). The difficulty with this problem lies in the condition that the \( x_i \) should be integers, which is equivalent to a non-linear constraint of the form \( \sin(\pi x) = 0 \). (11.1-2) belongs to a class of so-called NP-complete problems. These are known to be either collectively capable, or collectively incapable, of solution by polynomial-time
algorithms. Thus if (11.1-2) could be shown to be polynomial-time solvable, this would be automatically true of other important problems such as Boolean satisfiability or the travelling salesman problem.

We first discuss the existence of an integer solution for the linear diophantine equation

\[ c_1 x_1 + \ldots + c_n x_n = c, \]  

(11.3)

where \( c, c_i \in \mathbb{Z} \), the set of integers. If such a solution exists, then the greatest common divisor \( g = (c_1, \ldots, c_n) \) of the \( c_i \) must be a factor of \( c \). To show that the converse is also true we require some results from number theory. (Theorems 11.1-4 follow the treatment in Niven and Zuckerman [68].) It will be convenient to begin with the two-variable case.

**Theorem 11.1** Let \( b, c \in \mathbb{Z} \). If \( g = (b, c) \), then there exist integers \( x_0, y_0 \) such that

\[ g = bx_0 + cy_0. \]

**Proof** Choose \( x_0, y_0 \) so that \( m = bx_0 + cy_0 \) is the smallest positive integer of the form \( bx + cy \), where \( x, y \in \mathbb{Z} \). We show that \( m \mid b \), i.e. \( m \) is a factor of \( b \). To obtain a contradiction, assume that \( m \nmid b \). Then there are integers \( q, r \) such that

\[ b = mq + r, \quad 0 < r < m, \]

i.e.

\[ r = b - mq = b - (bx_0 + cy_0)q \]

\[ = bx_1 + cy_1 < bx_0 + cy_0, \]
where \( x_1 = 1 - x_0, y_1 = -qy_0 \).

But this inequality contradicts the definition of \( m \), hence \( m | b \). We can similarly show that \( m | c \). Since \( g = (b, c) \), there are integers \( k_1, k_2 \) such that \( b = gk_1, c = gk_2 \) and \( m = bx_0 + cy_0 = g(k_1x_0 + k_2y_0) \). It follows that \( g | m \), i.e. \( g \leq m \). But \( g < m \) contradicts \( g = (b, c) \). Hence \( g = m \). //

From the above proof we immediately obtain

**Theorem 11.2** \( g = (b, c) \) is the least positive value of \( bx + cy \), where \( x, y \) range over \( \mathbb{Z} \).

The theorem below is a generalization of theorem 11.2 to \( n \) variables and will be stated without proof.

**Theorem 11.3** Given any integers \( c_1, \ldots, c_n \) (not all 0), with \( g = (c_1, \ldots, c_n) \), there exist integers \( x_1, \ldots, x_n \) such that

\[ g = \sum_{i=1}^{n} c_i x_i. \]

\( g \) is the least positive value of the linear form \( \sum_{i=1}^{n} c_i y_i \), with \( y_i \) ranging over \( \mathbb{Z} \).

Now suppose \( g | c \). By the above theorem there exist integers \( x_1, \ldots, x_n \) such that

\[ g = \sum_{i=1}^{n} c_i x_i \]

Since \( g | c \), there exists \( k \in \mathbb{Z} \) such that \( c = kg \). Hence
\( y_i = kx_i \) is a solution of (11.3). We therefore have the following result.

**Theorem 11.4** The linear diophantine equation

\[
c_1 x_1 + \ldots + c_n x_n = c \tag{11.3}
\]

has an integer solution if and only if \((c_1, \ldots, c_n)|c\).

The integer solutions of (11.3) are obtained by reduction to the two-variable case. We therefore first consider the equation

\[
a x + b y = c. \tag{11.4}
\]

Note that if \((a, b) = 1\) and \(x_0, y_0\) is any integer solution of (11.4), then all integer solutions are of the form

\[
x = x_0 - bt
\]

\[
y = y_0 + at
\]

\(t = 0, \pm 1, \pm 2, \ldots\). To see this, let \(x, y\) be any other integer solution. Then

\[
ax - ax_0 + by - by_0 = 0
\]

i.e.

\[
y - y_0 = (a/b)(x_0 - x).
\]

Since \((a, b) = 1\), \(b|(x_0 - x)\), i.e. \(x_0 - x = bt\), for some \(t\), and \(y - y_0 = at\). The converse follows by direct substitution of (11.5) into (11.4). The problem of solving (11.4) is now reduced to finding an initial solution \(x_0, y_0\). The algorithm for determining \(x_0, y_0\) involves continued fractions. Recall that any rational number \(a/b\) \((a, b \in \mathbb{Z}, b \neq 0)\) can be written as a finite continued fraction of the form
\[ [a_0, a_1, \ldots, a_n] = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \ldots + \frac{1}{a_{n-1} + \frac{1}{a_n}}} \]

The \((n-1)\)th convergent of \([a_0, a_1, \ldots, a_n]\) is the rational number defined by

\[ \frac{p_{n-1}}{q_{n-1}} = \left[ a_0, a_1, \ldots, a_{n-1} \right], \]

with \(p_{n-1}, q_{n-1} \in \mathbb{Z}, q_{n-1} \neq 0\).

We assume that \((a, b) = 1\) and apply the Euclidean algorithm:

\[
\begin{align*}
a/b & = a_0 + 1/(b/r_1), \\
b/r_1 & = a_1 + 1/(r_1/r_2), \\
r_1/r_2 & = a_2 + 1/(r_2/r_3), \\
& \vdots \\
r_{n-2}/r_{n-1} & = a_{n-1} + 1/(r_{n-1}/r_n), \quad 0 < r_n < r_{n-1}, \\
r_{n-1}/r_n & = a_n.
\end{align*}
\]

Hence \(a/b = [a_0, a_1, \ldots, a_n]\). If we define the convergents

\[ \Delta_0 = [a_0], \quad \Delta_1 = [a_0, a_1], \ldots, \quad \Delta_k = [a_0, \ldots, a_k], \]

with \(\Delta_k = p_k/q_k\), \(1 \leq k \leq n\),

then

- 128 -
\[ \Delta_0 = a_0, \text{ i.e. } P_0 = a_0 \text{ and } Q_0 = 1, \]
\[ \Delta_1 = a_0 + 1/a_1, \text{ i.e. } P_1 = a_0a_1 + 1 \text{ and } Q_1 = a_1, \]
\[ \Delta_2 = a_0 + 1/(a_1 + 1/a_2), \text{ i.e. } P_2 = a_0a_1a_2 + a_0 + a_2 = a_2P_1 + a_0 \text{ and } Q_2 = a_1a_2 + 1 = a_2Q_1 + a_0. \]

Using induction, it is easy to prove that
\[ P_k = a_kP_{k-1} + P_{k-2} \text{ and } Q_k = a_kQ_{k-1} + Q_{k-2}, \quad (11.6) \]
for \( k = 2, \ldots, n. \) Applying (11.6) repeatedly gives
\[ \Delta_k - \Delta_{k-1} = \frac{P_kQ_{k-1} - Q_kP_{k-1}}{Q_kQ_{k-1}} = \frac{(-1)(P_{k-1}Q_{k-2} - Q_{k-1}P_{k-2})}{Q_{k-1}Q_{k-2}} \]
\[ ... = \frac{(-1)^{k-2}(P_2Q_1 - Q_2P_1)}{Q_{k-1}Q_{k-2}} = \frac{(-1)^{k-1}}{Q_kQ_{k-1}}. \]

Hence \( \Delta_n - \Delta_{n-1} = a/b - \Delta_{n-1} = (-1)^{n-1}/(bQ_{n-1}), \)
\[ \text{ i.e. } aQ_{n-1} - bP_{n-1} = (-1)^{n-1} \]
\[ \text{ i.e. } a[(-1)^{n-1}cQ_{n-1}] + b[(-1)^n cP_{n-1}] = c. \]

It follows that \( x_o = (-1)^{n-1}cQ_{n-1}\text{sgn } a \) and \( y_o = (-1)^{n}cP_{n-1}\text{sgn } b. \) We therefore have the following result.

**Theorem 11.5** If \((a, b) = 1,\) then all integer solutions of
\[ ax + by = c \]
are given by
\[ x = (-1)^{n-1}c Q_{n-1}\text{sgn } a - bt, \]
\[ y = (-1)^n c P_{n-1}\text{sgn } b + at, \quad (11.7) \]
with \( P_{n-1}, Q_{n-1} \) defined as above.
Example 11.1 Determine all non-negative integer pairs \((x, y)\) such that \(5x + 3y = c\) is minimal subject to the constraints

\[
\begin{align*}
11x + 15y &\leq 100 \\
-5x - 3y &\leq -20 \\
3x - y &\leq 12.
\end{align*}
\]

The N.A.G. routine H02BAF gives \(c_{\text{min}} = 20\) for \((x, y) = (4, 0)\). Since \(5/3 = [1, 1, 2]\), we have \(n = 2, P_1 = 2, \) and \(Q_1 = 1\). Hence \(x = -20 - 3t\) and \(y = 40 + 5t, t = 0, \pm 1, \pm 2, \ldots\).

The constraints \(x, y \geq 0\) imply \(t = -8\) or \(-7\), which gives \((x, y) = (4, 0)\) or \((1, 5)\). Both points are seen to be feasible.

To find any second best, sub-optimal solutions, set \(c = 21\). Then \(x = -21 - 3t\) and \(y = 42 + 5t, t = 0, \pm 1, \pm 2, \ldots\), and we similarly obtain \((x, y) = (3, 2)\) or \((0, 7)\).

Now consider the general case

\[c_1 x_1 + \ldots + c_n x_n = c, \quad (11.3)\]

\(n > 2\). A simple inductive proof can be found in Niven and Zuckerman [68]. We proceed constructively: set

\[x_{n-1} = \alpha_1 v_1 + \alpha_2 v_2, \quad x_n = \beta_1 v_1 + \beta_2 v_2.\]

If \(\alpha_1, \alpha_2, \beta_1, \beta_2\) are chosen so that

\[\alpha_1 \beta_2 - \alpha_2 \beta_1 = 1, \quad (11.8)\]

then

\[
\begin{pmatrix}
\alpha_1 \\
\beta_1
\end{pmatrix}
\begin{pmatrix}
\alpha_2 \\
\beta_2
\end{pmatrix}
\begin{pmatrix}
v_1 \\
v_2
\end{pmatrix}
= \begin{pmatrix}
x_{n-1} \\
x_n
\end{pmatrix}, \quad (11.9)
\]

- 130 -
and
\[
\begin{pmatrix}
 v_1 \\
 v_2 \\
 x_{n-1} \\
 x_n
\end{pmatrix}
= \begin{pmatrix}
 \beta_2 & -\alpha_2 \\
 -\beta_1 & \alpha_1
\end{pmatrix}
\begin{pmatrix}
 x_{n-1} \\
 x_n
\end{pmatrix}.
\] (11.10)

Putting \( \alpha_2 = c_n/(c_{n-1},c_n) \), \( \beta_2 = -c_{n-1}/(c_{n-1},c_n) \), we have \((\alpha_2, \beta_2) = 1\) and can use the two-variable method to determine \( \alpha_1, \beta_1 \) so that (11.8) is satisfied.

It then follows that
\[
c_{n-1}x_{n-1} + c_n x_n = (c_{n-1} \alpha_1 + c_n \beta_1)v_1,
\]
i.e. (11.3) has been reduced to \( n-1 \) unknowns
\( x_1, \ldots, x_{n-2}, v_1 \). Next put
\[
x_{n-2} = \alpha_3 v_3 + \alpha_4 v_4, \quad v_1 = \beta_3 v_3 + \beta_4 v_4
\]
to reduce the unknowns to \( x_1, \ldots, x_{n-3}, v_3 \). The process is continued until only two unknowns remain, \( x_1 \) and \( v_{2n-5} \).

The last two substitutions are as follows
\[
x_3 = \alpha_{2n-7} v_{2n-7} + \alpha_{2n-6} v_{2n-6}, \quad v_{2n-9} = \beta_{2n-7} v_{2n-7} + \beta_{2n-6} v_{2n-6},
\]
\[
x_2 = \alpha_{2n-5} v_{2n-5} + \alpha_{2n-4} v_{2n-4}, \quad v_{2n-7} = \beta_{2n-5} v_{2n-5} + \beta_{2n-4} v_{2n-4}.
\]
(11.3) now takes the form
\[
ax_1 + bv_{2n-5} = c, \quad (11.11)
\]
where \( a \) and \( b \) are integer coefficients. We apply (11.7) to (11.11) and obtain the general solution
\[
x_1 = k - bt_1, \quad v_{2n-5} = l + at_1
\]
\[ t_1 = 0, \pm 1, \pm 2, \ldots \] Backsubstitution now gives the solution of (11.3) in terms of \( n-1 \) arbitrary parameters \( t_1, \ldots, t_{n-1} \). We find

\[ x_2 = \alpha_{2n-5} v_2n-5 + \alpha_{2n-4} v_2n-4 = \alpha t_1 + \beta t_2 + \gamma, \]
where \( \alpha = \alpha_{2n-5} \), \( \beta = \alpha_{2n-4} \), \( \gamma = \alpha_{2n-5} \) and \( t_2 = v_2n-4 \).

Similarly \( x_j \) is seen to be a linear form in \( t_1, t_2, t_3 \), where \( t_3 = v_2n-6 \). Putting \( t_4 = v_2n-8, \ldots, t_{n-1} = v_2 \) and continuing in this way, we find the solution of (11.3) in triangular form

\[
\begin{pmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_{n-1} \\
  x_n
\end{pmatrix} =
\begin{pmatrix}
  d_{11} & 0 & 0 & \cdots & 0 \\
  d_{21} & d_{22} & 0 & \cdots & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  d_{n-1,1} & d_{n-1,2} & d_{n-1,3} & \cdots & d_{n-1,n-1} \\
  d_{n,1} & d_{n,2} & d_{n,3} & \cdots & d_{n,n-1}
\end{pmatrix}
\begin{pmatrix}
  t_1 \\
  t_2 \\
  \vdots \\
  t_{n-2} \\
  t_{n-1}
\end{pmatrix} + \mathbf{k},
\]

where

\[
\begin{align*}
  d_{11} &= -b, & k_1 &= k \\
  d_{21} &= a_2n-5, & d_{22} &= \alpha_{2n-4}, & k_2 &= \alpha_{2n-5} \ell; \\
  d_{31} &= a_2n-7 \beta_{2n-5}, & d_{32} &= \alpha_{2n-7} \beta_{2n-4}, \\
  d_{33} &= \alpha_{2n-6}, & k_3 &= \alpha_{2n-7} \beta_{2n-5} \ell; \\
  \vdots & & \vdots & & \vdots \\
  d_{r,1} &= a_{2n-(2r+1)} \beta_{2n-(2r-1)} \beta_{2n-(2r-3)} \cdots \beta_{2n-5}, \\
  d_{r,s} &= a_{2n-(2r+1)} \beta_{2n-2s} \beta_{2n-(2r-1)} \beta_{2n-(2r-3)} \cdots \beta_{2n-(2s+3)}, & s &= 2, \ldots, r-1,
\end{align*}
\]
\[ d_r, r = \alpha_{2n - 2r}, k_r = d_{r1} \lambda/a, \quad r = 4, \ldots, n - 1; \]
\[ d_{n1} = \alpha_1 \beta_3 \eta_5 \ldots \beta_{2n - 5}, \]
\[ d_{n, s} = \beta_{2n - 2s} \beta_1 \beta_3 \eta_5 \ldots \beta_{2n - (2s + 3)}, \quad s = 2, \ldots, n - 1, \]
\[ k_n = d_{n1} \lambda/a. \]

In many operations research problems, the \( c_i \) are non-negative, which allows us to determine upper bounds \( s_i \) for the \( x_i \):

\[ 0 \leq x = D \lambda + k \leq s, \]

where \( s = (c/c_1, \ldots, c/c_n) \). Hence

\[ \lambda \leq D \lambda \leq L, \]

where \( \lambda = -k \) and \( L = s - k \). To obtain bounds \( m, M \) such that

\[ m \leq \lambda \leq M \]

we proceed as follows. First assume that

\( d_{11}, \ldots, d_{n-1, n-1}, d_{n, n-1} > 0 \); if any of these are negative, upper and lower bounds are interchanged.

From \( \lambda \leq d_{11} \lambda_1 \leq L_1 \), we find

\[ m_1 = \lambda_1/d_{11}, \quad M_1 = L_1/d_{11}. \]

For \( k = 2, \ldots, n - 1, \)

\[ (\lambda_k - d_k, l_1 - \ldots - d_k, k_1 t_{k-1})/d_{kk} \leq t_k \leq \]

\[ (L_k - d_k, l_1 - \ldots - d_k, k_1 t_{k-1})/d_{kk}. \]  \hspace{1cm} (11.12)

In the LHS (RHS) of (11.12) set

\[ t_r = m_r, M_r, \text{ or } 0 \]

according as \(-d_{kr} (+d_{kr})\) is positive, negative, or zero.
Finally use
\[\frac{a_n - d_n, t_1 - \cdots - d_{n-2} t_{n-2}}{d_n, n-1} \leq t_{n-1} \leq \frac{b_n - d_n, t_1 - \cdots - d_{n-2} t_{n-2}}{d_n, n-1}\]

to adjust \( m_{n-1}, M_{n-1} \) if necessary.

The bounds \( m_i, M_i \) found in this way are then used to limit the search for feasible optimal points. The method is demonstrated in the example below.

**Example 11.2** Solve the equation

\[2x_1 + x_2 + 5x_3 + 3x_4 = 13, \quad (11.13)\]

subject to the constraints

\[x_1 - x_2 + 5x_3 + x_4 \leq 10\]
\[3x_1 + x_2 - x_3 + 2x_4 \leq 15\]
\[-x_1 + 2x_2 + x_3 - x_4 \leq 5\]
\[x_1 + x_2 + x_3 + x_4 \leq 8.\]

Putting \( x_3 = \alpha_1 v_1 + \alpha_2 v_2, \quad x_4 = \beta_1 v_1 + \beta_2 v_2, \)
\[\alpha_2 = 3/(5,3) = 3, \quad \beta_2 = -5/(5,3) = -5, \quad \text{we have}\]
\[\alpha_1 \beta_2 - \alpha_2 \beta_1 = -5\alpha_1 - 3\beta_1 = 1. \quad \text{Now solve}\]
\[5\alpha_1 + 3\beta_1 = -1.\]
\[5/3 = [1,1,2], \quad P_1 = 2, \quad Q_1 = 1,\]
i.e. \( \alpha_1 = 1, \quad \beta_1 = -2. \quad \text{Hence (11.13) becomes}\]
\[2x_1 + x_2 - v_1 = 18. \quad (11.14)\]
Now put \( x_2 = \alpha_3 v_3 + \alpha_4 v_4 \), \( v_1 = \beta_3 v_3 + \beta_4 v_4 \),

\( \alpha_4 = \beta_4 = -1 \), i.e. \( \alpha_3 \beta_4 - \alpha_4 \beta_3 = -\alpha_3 + \beta_3 = 1 \).

From \( 1/1 = [0,1] \), \( P_0 = 0 \), \( Q_0 = 1 \) we obtain \( \alpha = -1 \), \( \gamma = 0 \). Hence (11.14) becomes

\[
2x_1 - v_3 = 18
\]

(11.15)

\( 2/1 = [1,1] \), \( P_0 = 1 \), \( Q_0 = 1 \) gives

\[
x_1 = 18 + t_1, \quad v_3 = 18 + 2t_1.
\]

Backsubstitution with \( t_2 = v_4 \), \( t_3 = v_2 \) now gives

\[
x_2 = -18 - 2t_1 - t_2, \quad x_3 = -t_2 + 3t_3, \quad x_4 = 2t_2 - 5t_3.
\]

Using

\( 0 \leq x_1 \leq 18/c_1 \),

We find

\[
\begin{pmatrix}
-18 \\
18 \\
0
\end{pmatrix} \leq \begin{pmatrix}
1 & 0 & 0 \\
-2 & -1 & 0 \\
0 & -1 & 3
\end{pmatrix} \begin{pmatrix}
t_1 \\
t_2 \\
t_3
\end{pmatrix} \leq \begin{pmatrix}
-9 \\
36 \\
6
\end{pmatrix},
\]

and hence

\[
-18 \leq t_1 \leq -9, \quad -18 \leq t_2 \leq 18, \quad -8 \leq t_3 \leq 8. \quad (11.16)
\]

(11.16) defines a superset of the feasible parameter set, but any infeasible solutions are easily eliminated. The above bounds give 10 feasible solutions (see appendix of programs). //

Example 11.3 This problem is a worked example for the N.A.G. library routine HØ2BAF: minimize \( c = x_1 + 2x_2 \), subject to the constraints \( 2x_1 + 2x_2 \leq 11 \), \( -x_1 + 3x_2 \leq 10 \), \( x_1 - x_2 \leq 2 \), \( -5x_1 - 15x_2 \leq -33 \), \( -16x_1 - 8x_2 \leq -33 \),
where $x_1, x_2$ are non-negative integers. The answer given is $c_{\text{min}} = 6$ for $(x_1, x_2) = (2, 2)$. Using ILP (see appendix), we obtain suboptimal solutions $(x_1, x_2) = (3, 2), (1, 3)$ for $c = 7$, and $(2, 3)$ for $c = 8$. //

For many applications, secondary constraints of the form $\|\mathbf{x}\| = \text{min}$ are of interest. Thus, if in example 11.2, we require $\sum x_i^2$ to be minimal, the (unique) solution is $\mathbf{x} = (0, 2, 2, 2)$.

**Example 11.4** $c = 2x_1 + x_2 + 5x_3 + 3x_4 = \text{min}$, subject to the constraints of example 11.2 and additional constraints

\[
\begin{align*}
-x_1, -x_2, -x_3, -x_4 &\leq -1 \\
-x_2 - x_4 &\leq -3 \\
-x_1 - x_4 &\leq -5.
\end{align*}
\]

The N.A.G. library routine H02BAF gives $c_{\text{min}} = 18$ for $\mathbf{x} = (4, 2, 1, 1)$. Searching the region defined by (11.16), produces a second optimal solution $(3, 1, 1, 2)$ which is, in fact, the unique minimum-norm solution. //

A FORTRAN version of the algorithm, called ILP (coded by P.J. Watts), is included in the appendix of programs. ILP, like other integer programming routines, is liable to exceed available time resources. An obvious partial remedy lies in speeding up the tree search by parallel processing. Further savings could perhaps be achieved by sharpening the bounds $m_i, M_i$. It is clear that ILP can also function as an ordinary integer programming routine. A lower bound $b$
for the minimum of \( f(x) \) can sometimes be obtained from physical considerations or by solving the associated continuous problem. ILP is then run with \( c = b, b+1, b+2, \ldots \) until an optimal feasible solution is found. For problems with many variables, this is obviously an inefficient process. However, if some estimate \( \hat{x} \) of the optimal solution \( x \) is available, we can solve \( \hat{x} = D t + k \) for \( t \) and then restrict the search to some neighbourhood of \( \hat{x} \), keeping within the limits of our time resources. We conclude this chapter with a brief description of ILP.

```
START

Input objective function: \( c_i, c \) and constraints: \( a_{i,j}, b_j \)

i=1, \ldots, n; \ j=1, \ldots, m.

Compute continued fraction and \( \alpha_i, \beta_i \), \( i=1, \ldots, 2n-4 \)

Determine \( d_{i,j} \) and \( k_{i,j} \), \( i=1, \ldots, n \), \( j=1, \ldots, n-1 \).

Compute bounds \( m_i, M_i \) for parameter \( t_i \), \( i=1, \ldots, n-1 \).

Search region defined by \( m_i, M_i \) for feasible solutions.

STOP
```

All variables are integers: \( \text{XK} = k, \text{VL} = \lambda, \text{T(I)} = t_i, \text{BE(I)} = \beta_i, \text{AL(I)} = \alpha_i, \text{IMN(I)} = m_i, \text{IMX(I)} = M_i, \text{D(I,J)} = d_{i,j}, \text{CON(I)} = k_i \).

There are 3 subroutines:
INPUT(M,N,A,B,C), CONVGT(I,J,PN,QN,NC), CONFRA(A,B,E,RN,NC).
CONVGT computes the convergents $P_n, Q_n$; CONFRA determines the continued fraction $A/B = [a_0, ..., a_k]$. Formal parameters:
M = m, N = n, A(A,J) = $a_{ij}$, B(I) = $b_i$, C(I) = $c_i$, PN = $P_n$,
QN = $Q_n$, E(I) = $a_i$, RN = highest common factor, NC = k.
REFERENCES


Ahuja, G.C., Narang, T.D. and Swaran Trehan [13],
Best approximation on convex sets in a metric space,

Barrodale, I. and Phillips, C. [33], Solution of an over-
determined system of linear equations in the Chebyshev
norm, ACM Trans. on Math. Software, 1, No.3, 264-270
(1975).

Barrodale, I. and Roberts, F.D.K. [40], An improved algorithm
for discrete $L_1$ linear approximation, SIAM J. Numer.

Barrodale, I. and Roberts, F.D.K. [40], Solution of an
overdetermined system of equations in the $L_1$ norm,

Blatter, J. [30], Reflexivity and the existence of best
approximations. In: Approximation Theory II,

Breckner, W.W. [11], Bemerkungen über die Existenz von
Minimallösungen in normierten Räumen, Matematika
(Cluj), 10, 223-228 (1968).

Cheney, E.W. [62], Tchebycheff approximation and related
extremal problems, JMM, 14, 87-98 (1965).


Hadley, G [45], Linear Programming, Addison-Wesley (1978).


Menger, K. [23], Untersuchungen über allgemeine Metrik, Math. Annalen, 100, 75-163 (1928).


Planitz, M. [47], see Voller, V.

Planitz, M. [37], Linear $L_1, L_2$ and $L_\infty$ approximation for discrete data (as Algorithm 524 under revision for Algorithms Section of Appl. Statistics).
Planitz, M. and Watts, P.J. [70], An application of diophantine theory to integer linear programming (in preparation).

Reid, K.J. [50], A survey of material balance computer packages in the minerals industry. In: T.B. Johnson and R.J. Barnes (Eds), 17th Application of computers and operations research in the minerals industry, AIME, New York (1982).


Sadovski, A.N. [41], L1-norm fit of a straight line, Appl. Statistics, 23, 244-248 (1974).


Sklar, M.G. and Armstrong, R. [71], An algorithm for
discrete curve fitting for the simple model using a
dual linear programming approach, Commun. Statist. -

Smith, H.W. and Ichiyen, N. [54], Computer adjustment of
metallurgical balances, The Canad. Inst. of Mining


Smith, M.A. [20], Some examples concerning rotundity in

Stiefel, E. [63], Über diskrete und lineare Tschebyscheff-

Taha, H.A. [67], Operations Research, An Introduction,

Vlasov, L.P, [26], The concept of approximative compactness

Voller, V.R., Planitz, M. and Reid, K.J. [47], A comparison
of the algorithms for automated data adjustment and
material balance around mineral processing equipment,
Proc. Fourth Int. Conf. Math. Modelling, Zürich,


<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINMAX</td>
<td>A 1</td>
</tr>
<tr>
<td>EXTRAP</td>
<td>A 6</td>
</tr>
<tr>
<td>SOLVE</td>
<td>A 7</td>
</tr>
<tr>
<td>STRICT</td>
<td>A 15</td>
</tr>
<tr>
<td>MINBAL</td>
<td>A 21</td>
</tr>
<tr>
<td>ILP</td>
<td>A 24</td>
</tr>
</tbody>
</table>
SUBROUTINE MINMAX
DRIVER PROGRAM

IMPLICIT REAL (A-H, O-Z)
INTEGER *4 K10, K20
PARAMETER (N = 1001)
DIMENSION X(N), Y(N), E(N)
COMMON X, Y, E
WRITE(1, 1)
1 FORMAT(//"INPUT NUMBER OF POINTS")
READ(1, *) NPTS
WRITE(1, 7)
7 FORMAT(//"TYPE 1 TO INPUT INTERACTIVELY/
" TYPE 2 FOR EXponential FUNCTION")
READ(1,**) ERR = 6)IREPLY
IF (IREPLY.EQ.2) GO TO 9
IF (IREPLY.NE.1) GO TO 6
CALL INTER(X, Y, NPTS)
GO TO 11
DO 10 I = 1, NPTS
X(I) = 0.01 * FLOAT(I - 1)
P = X(I)
Y(I) = EXP(P)
CONTINUE
CALL CTIMEA(K10)
CALL MINMAX(NPTS, X, Y, E, ITER, ERROR, A, B, C, D)
CALL CTIMEA(K20)
TI = FLOAT(K20 - K10) / 1000.
WRITE(1, 20) C, 0, A, B, ERROR, ITER, TI
20 FORMAT(//"L2 LINE/
"INTERCEPT: *, 1PE12.3/
"GRADIENT: *, 1PE12.3/
"MINMAX LINE/
"INTERCEPT: *, 1PE12.3/
"GRADIENT: *, 1PE12.3/
"MAXIMUM ERROR: *, 1PE12.3/
"NUMBER OF ITERATIONS: *, 14/
"CPU TIME IN SECONDS: *, 1PE9.2)
WRITE(1, 30)
30 FORMAT(//"DO YOU WISH TO CONTINUE*/
"TYPE Y OR N")
READ(1, 35) IREPLY
35 FORMAT(A1)
IF (IREPLY.EQ."Y") GO TO 5
IF (IREPLY.EQ."N") STOP
GO TO 25
END

SUBROUTINE MINMAX(N, X, Y, E, ITER, ERROR, A, B, C, D)

THIS SUBROUTINE COMPUTES THE L2 LINE
FOR A GIVEN SET OF DATA AND THEN USES
A MODIFIED EXCHANGE ALGORITHM TO OBTAIN
THE MINMAX LINE FOR THE SAME DATA

REAL X(N), Y(N), E(N), A, C, SX, SY, SXY,
SX, N, C, AMAX, ITEST, ERROR, B,
SIZE, SGNIND, SGNONE
LOGICAL L1, L2, L3, L4

FIND THE LEAST SQUARES LINE

SX = 0.0
SY = 0.0
SXY = 0.0
DO 40 I = 1, N
SX = SX + X(I)
SY = SY + Y(I)
SXY = SXY + X(I) * Y(I)
SXX = SXX + X(I) * X(I)
CONTINUE
D = (SY * SXX - SX * SXY) / (N * SXX - SX * SX)
C = (N * SXY - SX * SY) / (N * SXX - SX * SX)

THIS GIVES THE L2 LINE Y = C*X + D
NEXT OBTAIN ERRORS OF L2 LINE

-A1-
DO 60 I = 1, N
E(I) = D + C * X(I) - Y(I)
CONTINUE

THE NUMERICALLY LARGEST ERRORS OBEYING
SIGN PROPERTY ARE CHOSEN

CASE 1: MAX ABS ERROR IS NEITHER E(1) NOR E(N).
CASE 2: ABS(E(1)) IS MAX.
CASE 3: ABS(E(N)) IS MAX.
FIRST OBTAIN MAX ERROR

AMAX = ABS(E(1))
IND = 1
DO 70 I = 2, N
TEST = ABS(E(I))
IF (AMAX .GE. TEST) GO TO 70
AMAX = TEST
IND = I
70 CONTINUE
IF (IND .EQ. 1) GO TO 200
IF (IND .EQ. N) GO TO 300

CASE 1: NEITHER E(1) NOR E(N) GIVE MAX ABS ERROR

I2 = IND
AMAX = - 1.0
IFIRST = IND + 1
DO 80 I = IFIRST, N
TEST = ABS(E(I) - E(I))
IF (AMAX .GE. TEST) GO TO 80
AMAX = TEST
I3 = I
80 CONTINUE
AMAX = - 1.0
ILAST = IND - 1
DO 90 I = 1, ILAST
TEST = ABS(E(I) - E(I))
IF (AMAX .GE. TEST) GO TO 90
AMAX = TEST
I3 = I
90 CONTINUE
GO TO 320

CASE 2: E(1) GIVES MAX ABS ERROR

I1 = 1
AMAX = - 1.0
DO 100 I = 2, N
TEST = ABS(E(I) - E(I))
IF (AMAX .GE. TEST) GO TO 100
AMAX = TEST
I2 = I
100 CONTINUE
AMAX = - 1.0
IFIRST = I2 + 1
DO 110 I = IFIRST, N
TEST = ABS(E(I) - E(I))
IF (AMAX .GE. TEST) GO TO 110
AMAX = TEST
I3 = I
110 CONTINUE
GO TO 320

CASE 3: E(N) GIVES MAX ABS ERROR

I3 = N
AMAX = - 1.0
ILAST = N - 1
DO 310 I = 2, ILAST
TEST = ABS(E(N) - E(I))
IF (AMAX .GE. TEST) GO TO 310
AMAX = TEST
I2 = I
310 CONTINUE
AMAX = - 1.0
ILAST = I2 - 1
DO 315 I = 2, ILAST
TEST = ABS(E(I2) - E(I))
- A2 -
THE INITIAL TRIPLE IS X(I1), Y(I1); X(I2), Y(I2); X(I3), Y(I3). THE FIRST EQUAL ERROR LINE WILL NOW BE FOUND. ITER IS THE NUMBER OF EQUAL ERROR LINES NEEDED TO OBTAIN THE MINIMAX LINE Y = A*X + B

ITER = 0

ERROR = 0.5 * ((X(I1) - X(I2)) * (Y(I2) - Y(I3)) + (X(I3) - X(I2)) * (Y(I1) - Y(I2)))

ITER = ITER + 1

A = (Y(I3) - Y(I1)) / (X(I3) - X(I1))
B = Y(I1) + ERROR - A * X(I1)
AMAX = -1.0

DO 350 I = 1, N

ERROR = B + A * X(I) - Y(I)
SIZE = ABS(ERROR)

IF (AMAX .GE. SIZE) GO TO 350

AMAX = SIZE

350 CONTINUE

CONTINUE IF (IND .EQ. I1 .OR. IND .EQ. I2 .OR.
* IND .EQ. I3) GO TO 500

THE ABOVE DETERMINES IF THE MINIMAX LINE HAS BEEN FOUND. IF NOT, THE EXCHANGE ALGORITHM WILL NOW BE ACTIVATED

SGNIND = DSIGN(1.0, E(IND))
SGNONE = DSIGN(1.0, E(I1))

X(IND) TO LEFT OF X(I1)

L1 = X(IND) .LT. X(I1)

X(IND) BETWEEN X(I1) AND X(I2)

L2 = X(IND) .LT. X(I1) .AND. X(IND) .LT. X(I2)

X(IND) BETWEEN X(I2) AND X(I3)

L3 = X(IND) .LT. X(I2) .AND. X(IND) .LT. X(I3)

X(IND) TO RIGHT OF X(I3)

L4 = X(IND) .LT. X(I3)

IF (SGNIND .EQ. SGNONE) GO TO 400

E(IND) AND E(I1) HAVE THE SAME SIGN

IF (.NOT. L1) GO TO 380

I3 = I2
I2 = I1
I1 = IND

GO TO 330

380 IF (L4) GO TO 390

I2 = IND

GO TO 330

390 I1 = I2
I2 = I3
I3 = IND

GO TO 330

E(IND) AND E(I1) HAVE SAME SIGN

400 IF (L3 .OR. L4) GO TO 410

I1 = IND

GO TO 330

410 I3 = IND

GO TO 330

EXCHANGE IS NOW COMPLETE

500 ERROR = ABS(ERROR)

RETURN

END
SUBROUTINE INTER(X, Y, NPTS)
IMPLICIT REAL (A - H, O - Z)
DIMENSION X(NPTS), Y(NPTS)

CALL END OF SUBROUTINE MINMAX

I N P U T  N U M B E R  O F  P O I N T S
31

T Y P E  I N  3 1  P A I R S
1
0. 0. 2
1. 1. 3
2. 1. 4
3. 2. 5
4. 1. 6
5. 3. 7
6. 2. 8
7. 2. 9
8. 3. 10
9. 5. 11
10. 3. 12
11. 4. 13
12. 5. 14
13. 4. 15
14. 5. 16
15. 6. 17
16. 6. 18
17. 5. 19
18. 7. 20
19. 6. 21
20. 8. 22
21. 7. 23
22. 7. 24
23. 8. 25
24. 7. 26
25. 9.
26. 11.
27. 10.
28. 12.
29. 11.
30. 13.

L2 LINE
GRADIENT: 3.706E-01
INTERCEPT: 5.444E-02

MINMAX LINE
GRADIENT: 3.810E-01
INTERCEPT: -2.857E-01
MAXIMUM ERROR: 1.857E 00

NUMBER OF ITERATIONS: 1
CPU TIME IN SECONDS: 1.00E-03

DO YOU WISH TO CONTINUE
TYPE Y OR N
Y

INPUT NUMBER OF POINTS
210

TYPE 1 TO INPUT INTERACTIVELY
ELSE TYPE 2 FOR EXPONENTIAL FUNCTION
2

L2 LINE
GRADIENT: 3.169E 00
INTERCEPT: 8.332E-02

MINMAX LINE
GRADIENT: 3.390E 00
INTERCEPT: 1.257E-01
MAXIMUM ERROR: 8.743E-01

NUMBER OF ITERATIONS: 3
CPU TIME IN SECONDS: 6.00E-03

DO YOU WISH TO CONTINUE
TYPE Y OR N
Y

INPUT NUMBER OF POINTS
1001

TYPE 1 TO INPUT INTERACTIVELY
ELSE TYPE 2 FOR EXPONENTIAL FUNCTION
2

L2 LINE
GRADIENT: 1.061E 03
INTERCEPT: -3.092E 03

MINMAX LINE
GRADIENT: 2.203E 03
INTERCEPT: -7.375E 03
MAXIMUM ERROR: 7.376E 03

NUMBER OF ITERATIONS: 3
CPU TIME IN SECONDS: 2.70E-02
SUBROUTINE EXTRAP

WRITE(1,10)
FORMAT(*)
READ(1,*) A,B,C,D,E,F
WRITE(1,10)
FORMAT(*)
READ(1,*) X
CALL EXTRAP(A,B,C,D,E,F,X,YMIN,YMAX,ICODE,JCODE)
WRITE(1,40)
FORMAT(*)
CODE = 1, 2, 3 DENOTES L1, L2, MINIMAX LINE RESPECTIVELY
STOP
END

SUBROUTINE EXTRAP(A,B,C,D,E,F,X,YMIN,YMAX,ICODE,JCODE)
ICODE = 1
YMIN = A*X + B
TEMP = C*X + D
IF (YMIN .LE. TEMP) GO TO 10
ICODE = 2
TEMP = E*X + F
IF (YMIN .LE. TEMP) GO TO 20
ICODE = 3
JCODE = 1
YMAX = A*X + B
TEMP = C*X + D
IF (YMAX .GE. TEMP) GO TO 30
YMAX = TEMP
JCODE = 2
TEMP = E*X + F
IF (YMAX .GE. TEMP) GO TO 40
YMAX = TEMP
JCODE = 3
RETURN
END

TYPE IN GRADIENT AND INTERCEPT OF L1, L2 AND MINIMAX LINES
.5 .5 -.1666667 .5 -.23
TYPE IN VALUE OF X
3
YMIN = 1.2500 00
CODE = 3
YMAX = 1.5000 00
CODE = 1
CODE = 1, 2, 3 DENOTES L1, L2, MINIMAX LINE RESPECTIVELY
SUBROUTINE SOLVE
DRIVER PROGRAM
PARAMETER ( IDIM = 6 , TOLER = 1.0E-6)
ADJUST IDIM = M + 2 , WHERE M IS THE
NUMBER OF DATA POINTS
DIMENSION A(IDIM, 6) , X(2) , T(2, 10) ,
DATA BIG /1.0E30/
M = IDIM - 2
M1 = IDIM - 1
DO 20 I = 1 , M
WRITE(1, 10) I
10 FORMAT ( * TYPE IN POINT NUMBER * , I5 )
READ (1, *) A(I, 2) , A(I, 5)
A(I, 1) = 1.
A(I, 3) = -1.
A(I, 4) = -A(I, 2)
A(I, 6) = FLOAT(I - 2)
CONTINUE
ENSURE THAT COLUMN 5 IS NON - NEGATIVE
DO 40 I = 1 , M
IF (A(I, 5) .GE. 3.) GO TO 40
DO 30 J = 1 , 6
A(I, J) = -A(I, J)
30 CONTINUE
CONTINUE
COMPUTE MARGINAL COSTS
DO 60 J = 1 , 5
SUM = 0.
DO 50 I = 1 , M
SUM = SUM + A(I, J)
50 CONTINUE
A(M1, J) = SUM
CONTINUE
A(IDIM, 5) = 1.
A(IDIM, 6) = 0.
IDENTIFY COLUMNS
A(IDIM, 1) = 1.
A(IDIM, 2) = 2.
A(IDIM, 3) = -1.
A(IDIM, 4) = -2.
CHOOSE PIVOT
IN = 1
DO 70 J = 2 , 4
IF (A(M1, IN) .LT. A(M1, J)) IN = J
70 CONTINUE
IF (A(M1, IN) .LE. ?) GO TO 300
DO 80 I = 1 , M
IF (A(I, IN) .LE. 0. OR. A(I, 5) .LE. 0.)
* RATIO(I) = BIG
IF (A(I, IN) .GT. C)
* RATIO(I) = A(I, 5) / A(I, IN)
80 CONTINUE
IOUT = 1
DO 90 I = 2 , M
IF (RATIO(I) .LT. RATIO(IOUT)) IOUT = I
90 CONTINUE
PIV = A(IOUT, IN)
IF (PIV .GT. TOLER) GO TO 95
IFAIL = 4
GO TO 330
COPY A INTO TEMP
DO 100 I = 1 , IDIM
DO 100 J = 1 , 6
   - A7 -
COMPUTE TABLEAU USING PIVOT PIV

DO 110 J = 1, 5
IF (J .EQ. IN) GO TO 110
TEMP(IOUT, J) = A(IOUT, J) / PIV
CONTINUE

DO 130 I = 1, M
IF (I .EQ. IOUT) GO TO 130
DO 120 J = 1, 5
IF (J .EQ. IN) GO TO 120
TEMP(I, J) = A(I, J) - D * TEMP(IOUT, J)
CONTINUE

DO 140 I = 1, M
IF (I .EQ. IOUT) GO TO 140
TEMP(I, IN) = - A(I, IN) / PIV
CONTINUE

TEMP(IOUT, IN) = 1. / PIV
TEMP(IOUT, 6) = A(IDIM, IN)
TEMP(IDIM, IN) = A(IOUT, 6)
DO 160 J = 1, 4
IF (A(IDIM, J) .NE. A(IOUT, IN)) GO TO 160
CONTINUE

DO 180 I = 1, M
IF (TEMP(I, 5) .GE. 0.) GO TO 180
DO 170 J = 1, 6
TEMP(I, J) = - TEMP(I, J)
CONTINUE

COLUMN 5 MUST BE NON-NEGATIVE

DO 190 I = 1, M
IF (TEMP(I, 5) .GE. 0.) GO TO 190
DO 170 J = 1, 6
TEMP(I, J) = - TEMP(I, J)
CONTINUE

DO 200 J = 1, 5
SUM = 0.
DO 190 I = 1, M
CR = 1.
IF (ABS(TEMP(I, 6)) .EQ. 1. .OR. ABS(TEMP(I, 6)) .EQ. 2.) CR = 0.
SUM = SUM + TEMP(I, J) + CR
CONTINUE

DO 210 J = 1, 6
A(I, J) = TEMP(I, J)
CONTINUE

COPY TEMP INTO A

DO 210 I = 1, IDIM
DO 210 J = 1, 6
A(I, J) = TEMP(I, J)
CONTINUE

CHECK IF TABLEAU IS OPTIMAL

IOPT = 0
DO 220 J = 1, 4
IF (A(M1, J) .GT. TOLER) IOPT = 1
CONTINUE
IF (IOPT .EQ. 1) GO TO 250
TABLEAU IS OPTIMAL
STORE SOLUTION IN X

IFLAG1 = 0
IFLAG2 = 0
DO 270 I = 1, M
IF (ABS(A(I, 6)) .NE. 1.) GO TO 270
D = SIGN(A(I, 5), A(I, 6))
Y(I) = D
IFLAG1 = 1
IF (ABS(A(I, 6)) .NE. 2.) GO TO 270
A8 -
5)

70

300

330

360

370

C

ccc

5000

5050

5100

5200

 SuPplimentary SOLVe(IDIM, A, X, Tolep, IFAIL, ISC, T)

GIVEN THE L1 LINE OF A SET OF DATA,
THIS ALGORITHM COMPUTES ANY OTHER
L1 LINE WHICH MAY EXIST

REAL A(IDIM, 6), X(2), Tolep, T(2, 10),
* AA(300, 102, 6), AD(300, 102, 6),
* TEMP(102, 6), CR, PIV, SUM
INTEGER IDIM, IFAIL, ISC, ISASIS(200, 102),
* IT, IPC, IC, ICOUNT, IEN, IFLAG, IFAIL1, IFAIL2, IN, Icut, IR, KA0,
* KAD, M, 11
COMMON /BLK/ AA, AD, IBASIS, TEMP
DATA 316/1.E30/
M = IDIM - 2
M1 = IDIM - 1
IFAIL = 0

I IS = 1
T(1, 1) = X(1)
T(2, 1) = X(2)
DO 5030 I = 1, 2
DO 5000 J = 2, 10
5000

5050

5100

5200

IC = 1
DO 5050 I = 1, 200
DO 5050 J = 1, IDIM
5050

5100

5200

5300

5400

5500

5600

COPY INTO AA(1, I, J),
INITIALIZE TEMP AND AD(1, I, J)
DO 5220 J = 1, 6
DO 5220 I = 1, IDIM
D = AA(I, J)
AA(1, I, J) = 0
TEMP(I, J) = 0.
5200

5300

5400

5500

5600

INITIALIZE AA AND AD
DO 5600 I = 2, 300
DO 5600 J = 1, IDIM
DO 5600 K = 1, 6
AA(I, J*K) = 0.
AD(I, J*K) = 0.
5600

ERROR = A(M1, 5)

- A9 -
KAA = 1
KAD = 0
IDEN = 1

CC
ICOUNT IS TABLEAU COUNTER

CC
ICOUNT = 1
GO TO 5755

CC
INITIALIZATION COMPLETE
SEARCH AA AND STORE IN AD

5670 KAC = 0
IDEN = 1
ICOUNT = 1
DO 5750 J = 1, IDIM
DO 5750 K = 1, 6
D = AA(ICOUNT, J, K)
A(J, K) = D
CONTINUE
GO TO 7140

5755 ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. 323) IFAIL = 1
IF (IFAIL .EQ. 1) RETURN
IF (ICOUNT .LE. KAA) GO TO 5675
IF (KAD .EQ. 0) RETURN
SEARCH AA AND STORE IN AA

5770 KAA = 0
IDEN = 1
ICOUNT = 1
DO 5850 J = 1, IDIM
DO 5850 K = 1, 6
D = AD(ICOUNT, J, K)
A(J, K) = D
CONTINUE
GO TO 7140

5850 ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. 330) IFAIL = 1
IF (IFAIL .EQ. 1) RETURN
IF (ICOUNT .LE. KAD) GO TO 5775
IF (KAA .EQ. 0) RETURN
GO TO 5670
ENTRY

SEARCH FOR POSITIVE PIVOTS

7040 IC = 1
7050 IR = 1
7059 IF (ABS(A(MI, IC)) .GT. TOLER .OR. ABS(A(IR, 5)) .LE. TOLER .OR.
   ABS(A(IR, IC)) .LE. 0) GO TO 7070
   IOUT = IR
IN = IC
IF (A(IN, IN) .GT. TOLER) GO TO 8310
IFAIL = 4
RETURN
7070 IR = IR + 1
IF (IR .LE. M) GO TO 7055
IC = IC + 1
IF (IC .LE. 4) GO TO 7050
SEARCH FOR ZERO ENTRY PIVOTS

7130 IC = 1
7150 IR = 1
7159 IF (ABS(A(MI, IC)) .EQ. 0 .OR. ABS(A(IR, 5)) .GT. TOLER) GO TO 7170
   IOUT = IR
IN = IC
IF (ABS(A(IOUT, IN)) .GT. TOLER) GO TO 9910
IFAIL = 4
RETURN
7170 IR = IR + 1
IF (IR .LE. M) GO TO 7150
IC = IC + 1
7180 IF (IDEN .EQ. 1) GO TO 5760
IF (IDEN .EQ. 0) GO TO 5855
NON-ZERO ENTRY PIVOTING

PIV=AI ICUT, IN)

COPY A INTO TEMP

DO 8020 I = 1, IDIM
DO 8020 J = 1, 6
TEMP(I, J) = A(I, J)

COMPUTE TABLEAU USING PIVOT PIV

DO 8025 J = 1, 5
IF (J .GE. IN) GO TO 8025
TEMP(IOUT, J) = A(IOUT, J) / PIV
CONTINUE
DO 8100 I = 1, M
IF (I .GE. IOUT) GO TO 8100
D = A(I, IN)
DO 8000 J = 1, 5
IF (J .GE. IN) GO TO 8050
TEMP(I, J) = A(I, J) - D * TEMP(IOUT, J)
CONTINUE
DO 8050 I = 1, M
IF (I .GE. IOUT) GO TO 8150
TEMP(I, IN) = -A(I, IN) / PIV
CONTINUE

COLUMN 5 MUST BE NON-NEGATIVE

DO 8200 I = 1, M
IF (TEMP(I, 5) .GE. 0.) GO TO 8200
DO 8170 J = 1, 6
TEMP(I, J) = -TEMP(I, J)
CONTINUE

COMPUTE MARGINAL COSTS

DO 8300 J = 1, 5
SUM = 0.
DO 8250 I = 1, M
CP = 1.
IF (ABS(TEMP(I, 6)) .GE. 1. * CR.
+ ABS(TEMP(I, 6)) .GE. 2.) CR = 0.
SUM = SUM + TEMP(I, J) * CR
CONTINUE
CP = 1.
IF (ABS(TEMP(IDIM, J)) .GE. 1. * CR.
+ ABS(TEMP(IDIM, J)) .GE. 2.) CR = 0.
TEMP(I, J) = SUM - CP
CONTINUE

CHECK IF BASIS IS NEW

DO 8301 I = 1, IDIM
IBASIS(IBC + 1, I) = 0
DO 8303 I = 1, M
IB = IFIX(TEMP(I, 6))
IBASIS(IBC + 1, IABS(IB)) = ISGN(1. * IB)
CONTINUE
DO 8307 I = 1, IDIM
IF (IBASIS(IBC + 1, J) .NE. 0) GO TO 8305
CONTINUE
CONTINUE

CONTINUE SEARCH FOR PIVOTS

GO TO 7070
CONTINUE

- A11 -
IF \( IBC \geq 1 \) \( 200 \) \( IFAIL = 2 \)
IF \( IFAIL = 0 \) \( 2 \) \( RETURN \)

SUM OF ABSOLUTE ERRORS SHOULD NOT INCREASE
* \( \text{GO TO 8500} \)

Determine Solution

IFLAG1 = 0
IFLAG2 = 0
DO 8350 I = 1, M
IF (ABS(TEMP(I, 5)) \( \cdot \text{NEG} \cdot 1 \)) \( \text{GO TO 8320} \)
D = SIGN(TEMP(I, 5), TEMP(I, 6))
X(1) = D
IFLAG1 = 1
DO 8350 I = 1, M
IF (ABS(TEMP(I, 6)) \( \cdot \text{NEG} \cdot 2 \)) \( \text{GO TO 8350} \)
D = SIGN(TEMP(I, 5), TEMP(I, 6))
X(2) = D
IFLAG2 = 1
CONTINUE

IF (IFLAG1 \( \cdot \text{EQ} \cdot 0 \)) \( X(1) = 0 \)
IF (IFLAG2 \( \cdot \text{EQ} \cdot 0 \)) \( X(2) = 0 \).

CHECK IF X CONTAINS NEW SOLUTION
GO 8400 J = 1, ISC
IF (ABS(T(1, J)) = X(1)) \( \cdot \text{LT} \cdot \text{TOLER} \) \( \text{AND} \)
* \( \cdot \text{ABS(T(2, J)) = X(2)} \) \( \cdot \text{LT} \cdot \text{TOLER} \) \( \text{GO TO 8500} \)
CONTINUE

STORE NEW SOLUTION IN T
IFLAG = 1
ISC = ISC + 1
IF (ISC \( \geq 10 \)) \( IFAIL = 3 \)
IF (IFAIL \( \cdot \text{EQ} \cdot 3 \)) \( RETURN \)
D = X(1)
T(1, ISC) = D
D = X(2)
T(2, ISC) = D
CONTINUE

IFLAG = 0 \( \text{MEANS PIVOT IS NO USE} \)

CONTINUE SEARCH FOR PIVOTS
GO 7070

STORE IN AA
KAA = KAA + 1
IF (KAA \( \geq 300 \)) \( IFAIL = 1 \)
IF (IFAIL \( \cdot \text{EQ} \cdot 1 \)) \( RETURN \)
DO 8650 J = 1, IDIM
DO 8650 K = 1, 6
D = TEMP(J, K)
AA(KAA, J, K) = D
CONTINUE

CONTINUE SEARCH FOR PIVOTS
GO 7070

STORE IN AD
KAD = KAD + 1
IF (KAD \( \geq 300 \)) \( IFAIL = 1 \)
IF (IFAIL \( \cdot \text{EQ} \cdot 1 \)) \( RETURN \)
DO 8750 J = 1, IDIM
DO 8750 K = 1, 6
D = TEMP(J, K)
AD(KAD, J, K) = D
CONTINUE
CONTINUE SEARCH FOR PIVOTS

ZERO-ENTRY PIVOTING

PIV = A(IOUT, IN)

COPY A INTO TEMP

DO 9020 I = 1, IDIM
   DO 9020 J = 1, 6
      TEMP(I, J) = A(I, J)
   CONTINUE

COMPLETE TABLEAU USING PIVOT PIV

DO 9025 J = 1, 5
   IF (J * EQ * IN) GO TO 9025
   TEMP(IOUT, J) = A(IOUT, J) / PIV
   CONTINUE

DO 9100 I = 1, M
   IF (I * EQ * IOUT) GO TO 9100
   D = A(I, IN)
   DO 9050 J = 1, 5
      IF (J * EQ * IN) GO TO 9050
      TEMP(I, J) = A(I, J) - D * TEMP(IOUT, J)
   CONTINUE

DO 9150 I = 1, M
   IF (I * EQ * IOUT) GO TO 9150
   TEMP(I, IN) = - A(I, IN) / PIV

DO 9150 J = 1, 4
   IF (A(IDIM, J) * NE. - A(IDIM, IN)) GO TO 9160
   TEMP(I, J) = - TEMP(I, IN)
   CONTINUE

COLUMN 5 MUST BE NON-NEGATIVE

DO 9200 I = 1, M
   IF (TEMP(I, 5) * GE. 0.) GO TO 9200
   DO 9170 J = 1, 6
      TEMP(I, J) = - TEMP(I, J)
   CONTINUE

COMPLETE MARGINAL COSTS

DO 9300 J = 1, 5
   SUM = 0.
   DO 9250 I = 1, M
      CR = 1.
      IF (ABS(TEMP(I, 6)) * EQ. 1. * OR.
         A3S(TEMP(I, 6)) * EQ. 2.) CR = 0.
      CONTINUE
      SUM = SUM + TEMP(I, J) * CR
   CONTINUE

CR = 1.
   IF (ABS(TEMP(IDIM, J)) * EQ. 1. * OR.
      ABS(TEMP(IDIM, J)) * EQ. 2.) CR = 0.
   TEMP(M1, J) = SUM - CR

CHECK IF BASIS IS NEW

DO 9301 I = 1, IDIM
   IBASIS(IBC + 1, I) = 0
   CONTINUE

DO 9303 I = 1, M
   IB = IFIX(TEMP(I, 6))
   IBASIS(IB + 1, IABS(IB)) = ISIGN(1, IB)
   CONTINUE

DO 9307 I = 1, IBC
   DO 9305 J = 1, IDIM
IF (IBASIS(I8C + 1, J) - IBASIS(I, J) .NE. 0)
   CONTINUE
GO TO 9307

9305 CONTINUE
CONTINUE SEARCH FOR PIVOTS

GO TO 7170

9307 CONTINUE

9309 IBC = IBC + 1
IF (IBC .GT. 200) IFAIL = 2
IF (IFAIL .EQ. 2) RETURN

IF ICEN = 0, TEMP IS STORED IN AA
IF ICEN = 1, TEMP IS STORED IN AC

IF (ICEN .EQ. 1) GO TO 9700

STORE IN AA

KAA = KAA + 1
IF (KAA .GT. 500) IFAIL = 1
IF (IFAIL .EQ. 1) RETURN
DO 9650 J = 1, IDIM
   D = TEMP(J, K)
   AA(KAA, J, K) = D
9650 CONTINUE

CONTINUE SEARCH FOR PIVOTS

GO TO 7170

STORE IN AD

9700 KAD = KAD + 1
IF (KAD .GT. 300) IFAIL = 1
IF (IFAIL .EQ. 1) RETURN
DO 9750 J = 1, IDIM
   D = TEMP(J, K)
   AD(KAD, J, K) = D
9750 CONTINUE

CONTINUE SEARCH FOR PIVOTS

GO TO 7170

END

OK, SEG SOLVE
TYPE IN POINT NUMBER 1
0.1.
TYPE IN POINT NUMBER 2
1.0.
TYPE IN POINT NUMBER 3
2.0.
TYPE IN POINT NUMBER 4
3.1.
IFAIL = 1
NUMBER OF SOLUTIONS: 5
7.000000E-01 0.000000E-01
1.000000E-01 0.000000E-01
0.000000E-01 0.000000E-01
1.000000E-01 2.000000E-01
-5.000000E-01 5.000000E-01

**** STOP

OK.

SEG SOLVE
TYPE IN POINT NUMBER 1
0.2.
TYPE IN POINT NUMBER 2
1.2.
TYPE IN POINT NUMBER 3
2.2.
TYPE IN POINT NUMBER 4
3.5.
IFAIL = 0
NUMBER OF SOLUTIONS: 3
2.000000E 01 5.000000E-01
SUBROUTINE STRICT
DRIVER PROGRAM

PARAMETER(M = 5, TOLER = 1.E-6)
M SHOULD EQUAL THE NUMBER OF DATA POINTS

DIMENSION HULL(2, 26), T1(M), T2(M)
WRITE(1, 10)
FORMAT('TYPE IN ISC>1, THE NUMBER OF SOLUTIONS')
READ(1, *) ISC
IF (ISC .LE. 1) GO TO 5
DO 30 J = 1, ISC
WRITE(1, 20) J
30 CONTINUE
FORMAT('TYPE IN SOLUTION POINT NUMBER *, IS)
READ(1, *) T1(J), T2(J)
DO 35 I = 1, ISC
WRITE(1, 34) I
35 CONTINUE
FORMAT('TYPE IN DATA POINT NUMBER *, IS)
READ(1, *) T1(I), T2(I)

INITIALIZE COLUMNS ISC + 1 TO 26
ISCP1 = ISC + 1
DO 40 J = ISCP1, 26
40 HULL(I, J) = 0.

CHOOSE MINIMUM 8
MIN = 1
DO 50 J = 2, ISC
IF (HULL(2, J) .GE. HULL(2, MIN)) GO TO 50
MIN = J
50 CONTINUE
MIN IS THE NUMBER OF POINTS IN THE HULL
IHC = 0
HULL(1, ISC + 1) = HULL(1, MIN)
HULL(2, ISC + 1) = HULL(2, MIN)
ANGMIN = -1.

CHOOSE MINIMUM ANGLE
PI = A TAN(1.0) * 4.*
IHC = IHC + 1
TEMP1 = HULL(1, IHC)
TEMP2 = HULL(2, IHC)
HULL(1, IHC) = HULL(1, MIN)
HULL(2, IHC) = HULL(2, MIN)
HULL(1, MIN) = TEMP1
HULL(2, MIN) = TEMP2
MIN = ISC + 1
ALPHA = ANGMIN
ANGMIN = PI * 2.0
ISCP1 = ISC + 1
DO 70 J = IHC, ISCP1
X = HULL(1, J) - HULL(1, IHC)
Y = HULL(2, J) - HULL(2, IHC)
CALL ANGLE(X, Y, ANG)
IF (ANG .LE. ALPHA) GO TO 73
IF (ANG .GE. ANGMIN) GO TO 70
MIN = J
X = HULL(1, MIN) - HULL(1, IHC)
Y = HULL(2, MIN) - HULL(2, IHC)
CALL ANGLE(X, Y, ANG)
ANGMIN = ANG
CONTINUE

ELIMINATE INTERMEDIATE POINTS
X = HULL(1, MIN) - HULL(1, IHC)
Y = HULL(2, MIN) - HULL(2, IHC)
AGL = ANGMIN
DIS = X * X + Y * Y
ISCP1 = ISC + 1
DO 80 J = ISCP1, ISCP1
- A15 -
80 CONTINUE

IF (MIN .LT. ISCP) GO TO 60

PREPARE OUTPUT FOR HULL

WRITE(1, 90) IHC

FORMAT('NUMBER OF VERTICES OF HULL: ', I5)

WRITE(1, 100)

FORMAT('COORDINATES OF VERTICES: ')

DO 110 J = 1, IHC

WRITE(1, 120) HULL(1, J), HULL(2, J)

120 FORMAT(1P2E14.6)

COMPUTE STRICT APPROXIMATION

CALL STRICT(IHC, HULL, M, T1, T2, TCLER,
+ ICODE, A, B, C, D)

WRITE(1, 130) C, D

FORMAT(TH E L2 APPROXIMATION IS GIVEN BY/
+ *GRADIENT: *, 1PE14.6/
+ *INTERCEPT: *, 1PE14.6)

IF (ICODE .EQ. 0) GO TO 150

WRITE(1, 140) A, B

FORMAT(TH E STRICT APPROXIMATION IS GIVEN BY/
+ *GRADIENT: *, 1PE14.6/
+ *INTERCEPT: *, 1PE14.6)

GO TO 170

150 WRITE(1, 160)

FORMAT(TH E L2 AND STRICT APPROXIMATION ARE IDENTICAL*)

170 STOP

END

COMPUTE ANGLE ANG

SUBROUTINE ANGLE(X, Y, ANG)

IF (X .EQ. 0.0 .AND. Y .EQ. 0.0) ANG = 4.0

IF (X .NE. 0.0 .OR. Y .NE. 0.0) ANG = Y / (ABS(X) + ABS(Y))

IF (X .LT. 0.0) ANG = 2.0 - ANG

IF (X .GE. 0.0 .AND. Y .LT. 0.0) ANG = ANG

ANG = ANG .2 .0 . ATAN(1.0)

RETURN

END

END OF DRIVER PROGRAM

SUBROUTINE STRICT(IHC, HULL, M, T1, T2, TCLER,
+ ICODE, A, B, C, D)

REAL HULL(2, IHC), T1(M), T2(M), A, SE, C, D,
+ SX, SY, SXY, SXX, DENOM, X2, Y2, DENOM2, SMIN, A,
+ S3, ANUM, GRD, TEMPM, TEMPB, SUM

DETERMINE L2 LINE WITH
+ GRADIENT C AND INTERCEPT D

SX = 0.0
SY = 0.0
SXY = 0.0
SXX = 0.0

DO 10 I = 1, M

SX = SX + T1(I)

SY = SY + T2(I)

SXY = SXY + T1(I) * T2(I)

SXX = SXX + T1(I) ** T1(I)

10 CONTINUE

DENOM = FLOAT(M) * SXX - SX * SX

D = (SY * SXX - SX * SXY) / DENOM

- A16 -
Determine centroid \((x_0, y_0)\) of hull

\[ x_0 = 0.0 \]
\[ y_0 = 0.0 \]

Continue

\[ x_3 = x_0 + \text{hull}(1, \text{itop}) \]
\[ y_0 = y_0 + \text{hull}(2, \text{itop}) \]

Determine intersection \((x_2, y_2)\) of sides of hull with line through \((x_0, y_0)\) and \((c, d)\)

\[ \text{code} = 0 \]
\[ \text{kount} = 0 \]
\[ \text{itop} = \text{kount} + 1 \]

\[ \text{if} (\text{kount} \leq \text{ihc}) \text{ go to 40} \]

\[ \text{return} \]

\[ \text{if} (\text{kount} \geq \text{ihc}) \text{ itop} = \text{kount} + 1 \]

Denom = \((c - x_0) / \text{hull}(1, \text{itop}) - \text{hull}(1, \text{kount})\)

\[ \text{if} (\text{denom} \geq 0.0) \text{ go to 50} \]

\[ \text{go} = (d - y_0) / \text{denom} \]

\[ x_2 = \text{hull}(2, \text{itop}) - \text{hull}(2, \text{kount}) - \text{hull}(2, \text{kount}) - d \times x_0 + y_0 / \text{denom} \]

\[ \text{go to 52} \]

Case: infinite \(g_1\) and finite \(g_0\)

\[ g_0 = (d - y_0) / \text{denom} \]

\[ x_2 = \text{hull}(1, \text{kount}) \]
\[ y_2 = g_0 \times (x_2 - x_0) + y_0 \]

Test if \((c, d)\) is inside hull

\[ \text{if} (x_2 \lt \text{lt} \times x_0 \land x_2 \gt \text{lt} \times c \lor x_2 \lt \text{lt} \times c \land x_2 \gt \text{lt} \times x_0) \text{ go to 53} \]

\[ \text{go to 30} \]

\[ \text{if} (x_2 \geq \text{lt} \times \text{hull}(1, \text{kount}) \land x_2 \geq \text{lt} \times \text{hull}(1, \text{itop}) \land x_2 \geq \text{lt} \times \text{hull}(1, \text{kount}) \land \text{go to 110} \]

\[ \text{go to 30} \]

Case: infinite \(g_0\) and finite \(g_1\)

\[ g_1 = (\text{hull}(2, \text{itop}) - \text{hull}(2, \text{kount})) / \text{denom} \]

\[ x_2 = x_0 \]
\[ y_2 = g_1 \times (x_2 - \text{hull}(1, \text{kount})) + \text{hull}(2, \text{kount}) \]

Test if \((c, d)\) is inside hull

\[ \text{if} (x_2 \leq \text{lt} \times x_0 \land x_2 \geq \text{lt} \times c \lor x_2 \leq \text{lt} \times c \land x_2 \geq \text{lt} \times x_0) \text{ go to 60} \]

\[ \text{go to 30} \]

\[ \text{if} (y_2 \lt \text{lt} \times d \land y_2 \geq \text{lt} \times y_0 \lor y_2 \lt \text{lt} \times y_0 \land y_2 \geq \text{lt} \times d) \text{ go to 110} \]

\[ \text{go to 30} \]

Case: \(\text{ihc} = 2\)

\[ \text{denom} \times \text{hull}(1, 2) - \text{hull}(1, 1) \]

\[ \text{if} (\text{denom} \geq 0.0) \text{ go to 90} \]

\[ g_2 = (\text{hull}(2, 2) - \text{hull}(2, 1)) / \text{denom} \]

\[ \text{ddd} = g_2 \times (c - \text{hull}(1, 1)) + \text{hull}(2, 1) \]

\[ \text{if} (\text{abs} \times \text{ddd} \times \text{gt} \times \text{toler}) \text{ go to 110} \]

\[ \text{if} (c \geq \text{hull}(1, 2) \land c \geq \text{hull}(1, 1)) \lor (c = \text{hull}(1, 2) \land c \lt \text{hull}(1, 1)) \]

\[ \text{return} \]

- A17 -
CASE: IH C = 2 AND GRADIENT G2 IS INFINITE

IF (ABSCC - HULL(1, 1), GT, TOLER) GO TO 110

IF (CD, LE, HULL(2, 2) AND D, GE, HULL(2, 1), OR
+ D, GE, HULL(2, 2) AND D, LE, HULL(2, 1))

RETURN

ICODE = 1

COMPLETE STRICT SOLUTION

SMIN = 1.0E30
A = 0.0
B = 0.0
DO 160 J = 1, IH C
ITOP = J + 1
IF (J, EQ, IH C) ITOP = 1

CHECK IF GRADIENT IS FINITE

IF (HULL(1, ITOP) - HULL(1, J), EQ, 0.0) GO TO 136

GRD = (HULL(2, ITOP) - HULL(2, J))/
+ (HULL(1, ITOP) - HULL(1, J))

ANUM = 0.0
DENOM = 0.0
DO 120 I = 1, M
ANUM = ANUM + GRD *
+ (HULL(1, J) + T1(I) + GRD * HULL(1, J) +
- HULL(2, J) + T2(I)) - HULL(2, J) *
+ T1(I) + T1(I) + T2(I)
DENOM = DENOM + T1(I) + 2.0 * T1(I) *
+ GRD + GRD + GRD

CONTINUE

TEMPM = ANUM / DENOM
TEMPB = GRD * (TEMPM - HULL(1, J)) + HULL(2, J)

CHECK IF LOCAL MINIMUM LIES ON CURRENT SIDE

IF (TEMPM, GT, HULL(1, J), AND, TEM P M, GT,
+ HULL(1, ITOP), OR, TEM PM, LT,
+ HULL(1, ITOP)) GO TO 150

SUM = 0.0
DO 125 K = 1, M
SUM = SUM + (TEMPM * T1(K) + TEMPB - T2(K)) ** 2
IF (SUM, GE, SMIN) GO TO 150
SMIN = SUM
A = TEM PM
B = TEMPB
GO TO 150

CASE: GRADIENT GRD IS INFINITE

TEMPM = HULL(1, J)
TEMPB = 0.0
DO 140 I = 1, M
TEMPB = TEMPB + T2(I) - T1(I) + HULL(1, J)
CONTINUE

TEMPB = TEMPB / FLOAT(M)

CHECK IF LOCAL MINIMUM LIES ON CURRENT SIDE

IF (TEMPB, GT, HULL(2, J), AND, TEMPF B, GT,
+ HULL(2, ITOP), OR, TEMPB, LT, HULL(2, J)
+ AND, TEMPB, LT, HULL(2, ITOP))

GO TO 150

SUM = 0.0
DO 145 K = 1, M
SUM = SUM + (TEMPM * T1(K) + TEMPB - T2(K)) ** 2
IF (SUM, GE, SMIN) GO TO 150
SMIN = SUM
A = TEM PM
B = TEMPB

TEST VERTEX J + 1

TEMPM = HULL(1, ITOP)
TEMPB = HULL(2, ITOP)
SUM = 0.0

- A18 -
DO 155 K = 1, M
SUM = SUM + (TEMPM * T1(K) + TEMPB - T2(K)) ** 2
IF (SUM + CE. SMIN) GO TO 160
SMIN = SUM
A = TEMPM
B = TEMPB
160 CONTINUE
RETURN
END

OUTPUT
EXAMPLE 1

TYPE IN ISC>1, THE NUMBER OF SOLUTIONS
5
TYPE IN SOLUTION POINT NUMBER 1
0.5 0.
TYPE IN SOLUTION POINT NUMBER 2
0.5 -0.5
TYPE IN SOLUTION POINT NUMBER 3
0.1
TYPE IN SOLUTION POINT NUMBER 4
-0.5 1.
TYPE IN SOLUTION POINT NUMBER 5
3.333333 1.
TYPE IN DATA POINT NUMBER 1
0.1
TYPE IN DATA POINT NUMBER 2
1.0.
TYPE IN DATA POINT NUMBER 3
2.0.
TYPE IN DATA POINT NUMBER 4
3.1.

NUMBER OF VERTICES OF HULL: 4
COORDINATES OF VERTICES:
5.0000E-01 -5.03333E-01
3.33333E-01 1.00000E 00
-5.00000E-01 1.00000E 00
0.00000E-01 0.200000E-01

THE L2 APPROXIMATION IS GIVEN BY
GRADIENT: 0.000000E+01
INTERCEPT: 5.000000E+00
L2 AND STRICT APPROXIMATION ARE IDENTICAL

EXAMPLE 2

TYPE IN ISC>1, THE NUMBER OF SOLUTIONS
3
TYPE IN SOLUTION POINT NUMBER 1
0.5 2.
TYPE IN SOLUTION POINT NUMBER 2
1.25 1.25
TYPE IN SOLUTION POINT NUMBER 3
1.2 2.
TYPE IN DATA POINT NUMBER 1
0.2.
TYPE IN DATA POINT NUMBER 2
1.25
TYPE IN DATA POINT NUMBER 3
2.2
TYPE IN DATA POINT NUMBER 4
3.5.

NUMBER OF VERTICES OF HULL: 3
COORDINATES OF VERTICES:
1.250000E 00 1.250000E 00
1.000000E 39 2.000000E 00
5.999999E-01 2.000000E 00
The L2 approximation is given by
Gradient: \(8.499999e-01\)
Intercept: \(1.600000e 00\)

The strict approximation is given by
Gradient: \(8.333333e-01\)
Intercept: \(1.666667e 00\)

EXAMPLE 3
SET \( m = 5 \)

TYPE IN ISC\( \geq 1\), THE NUMBER OF SOLUTIONS
2

TYPE IN SOLUTION POINT NUMBER 1
0.5 2.5

TYPE IN SOLUTION POINT NUMBER 2
0.25 2.75

TYPE IN DATA POINT NUMBER 1
1.3

TYPE IN DATA POINT NUMBER 2
2.3

TYPE IN DATA POINT NUMBER 3
3.4

TYPE IN DATA POINT NUMBER 4
4.5

TYPE IN DATA POINT NUMBER 5
5.4

NUMBER OF VERTICES OF HULL: 2
COORDINATES OF VERTICES:
\(5.000000e-01\) \(2.500000e 00\)
\(2.500000e-01\) \(2.750000e 00\)

The L2 approximation is given by
Gradient: \(4.000000e-01\)
Intercept: \(2.600000e 00\)

L2 and strict approximation are identical
10 REM SUBROUTINE "MINBAL"
20 REM ----------------------
30 @%10
40 F0=0
50 DIM A(3,10)
60 DIM E(3,10)
70 DIM X(3,10)
80 DIM P(3)
90 DIM W(3)
100 REM INPUT ROUTINES
110 REM ----------------------------
120 FOR I%=1 TO 3
130 PRINT"INPUT DIRECTION OF STREAM ";I%
140 INPUT P(I%)
150 NEXT I%
160 PRINT
170 PRINT "INPUT MASS FLOW RATE"
180 INPUT "DEFAULT RATE 100", W(1)
190 PRINT
200 INPUT "INPUT NUMBER OF ASSAYS", N
210 PRINT
220 FOR J%=1 TO N
230 FOR I%=1 TO 3
240 PRINT "STREAM ";I%;" ASSAY ";J%;
250 INPUT A(I%,J%)
260 INPUT "INPUT ASSOCIATED ERROR (PERCENT)",E(I%,J%)
270 E(I%,J%)=(E(I%,J%)*A(I%,J%)/100)^2/2
280 PRINT
290 NEXT I%
300 PRINT:PRINT
310 NEXT J%
320 TEMPS=TIME
330 A=1:B=1
340 W(2)=W(1)*(A(1,1)-A(3,1))/(A(2,1)-A(3,1))
350 360 REM MINIMIZATION ROUTINE FOR W(2) GUESS
370 REM --------------------------------------
380 F=0
390 FOR J%=1 TO N
400 S1=0:S2=0
410 FOR I%=1 TO 3
420 S1=S1+E(I%,J%)*W(I%)^2
430 S2=S2+P(I%)*W(I%)*A(I%,J%)
440 NEXT I%
450 FOR I%=1 TO 3
460 X(I%,J%)=A(I%,J%)-P(I%)*E(I%,J%)*W(I%)*S2/S1
470 F=F+(X(I%,J%)-A(I%,J%))^2/2/E(I%,J%)
480 NEXT I%
490 NEXT J%
500 IF F0=F OR F<=F0 THEN GOTO 590
530 REM SEARCH ROUTINE ON W(2)
540 REM -------------------------------
550 W(2) = W(2) - B * A
560 A = A / 2
570 B = -B
580 IF A < .009 GOTO 620
590 F0 = F
600 W(2) = W(2) + B * A
610 GOTO 360
620 REM OUTPUT
630 REM ------
640 FOR J% = 1 TO N
650 @% = 10
660 PRINT "ASSAY TYPE "; J%
670 PRINT "STREAM", "MEASURED", "ADJUSTED", "MASSFLOW"
680 PRINT
690 @% = &0002030A
700 FOR I% = 1 TO 3
710 PRINT; I%; TAB(10) A(I%, J%); TAB(19) X(I%, J%); TAB(30) W(I%)
720 NEXT I%
730 PRINT
740 NEXT J%
750 @% = 10
760 PRINT "RUNNING TIME IN CSEC: "; TIME-TEMPUS
770 END
INPUT DIRECTION OF STREAM 1
?1
INPUT DIRECTION OF STREAM 2
?1
INPUT DIRECTION OF STREAM 3
?1

INPUT MASS FLOW RATE
DEFAULT RATE 100?100

INPUT NUMBER OF ASSAYS?2

STREAM 1 ASSAY 1?23.8
INPUT ASSOCIATED ERROR (PERCENT)?5

STREAM 2 ASSAY 1?5.3
INPUT ASSOCIATED ERROR (PERCENT)?5

STREAM 3 ASSAY 1?53.9
INPUT ASSOCIATED ERROR (PERCENT)?2

STREAM 1 ASSAY 2?52.1
INPUT ASSOCIATED ERROR (PERCENT)?10

STREAM 2 ASSAY 2?40.7
INPUT ASSOCIATED ERROR (PERCENT)?10

STREAM 3 ASSAY 2?63.4
INPUT ASSOCIATED ERROR (PERCENT)?4

ASSAY TYPE 1

<table>
<thead>
<tr>
<th>STREAM</th>
<th>MEASURED</th>
<th>ADJUSTED</th>
<th>MASSFLOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000</td>
<td>23.800</td>
<td>22.297</td>
<td>100.000</td>
</tr>
<tr>
<td>2.000</td>
<td>5.300</td>
<td>5.597</td>
<td>398.356</td>
</tr>
<tr>
<td>3.000</td>
<td>53.900</td>
<td>53.900</td>
<td>0.000</td>
</tr>
</tbody>
</table>

ASSAY TYPE 2

<table>
<thead>
<tr>
<th>STREAM</th>
<th>MEASURED</th>
<th>ADJUSTED</th>
<th>MASSFLOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000</td>
<td>52.100</td>
<td>62.398</td>
<td>100.000</td>
</tr>
<tr>
<td>2.000</td>
<td>40.700</td>
<td>15.663</td>
<td>398.356</td>
</tr>
<tr>
<td>3.000</td>
<td>63.400</td>
<td>63.400</td>
<td>0.000</td>
</tr>
</tbody>
</table>

RUNNING TIME IN CSEC: 8676
**A PROGRAM TO DETERMINE ALTERNATIVE OPTIMAL AND SUB-OPTIMAL INTEGER SOLUTIONS OF A LINEAR OBJECTIVE FUNCTION SUBJECT TO LINEAR CONSTRAINTS**

**IMPLICIT INTEGRER(A-Z)**

REAL M, N, P, R, X(N, N), OF, CS, GS, MNA, MLA, MA, MB

DIMENSION C(20), AL(N, N), BS(10), D(2, 2), CON(20),
MN(20), MX(N, N), OF(CF), QF(CF), F(20, 20),
A(20, 20), B(20), S(20, 20), X(N), K(20),

**INITIALIZE ARRAYS**

DO 10 ICU = 1, 2
DO 10 ICU = 1, 2
DO ICU = 1, 2

CONTINUE

CALL INPUT(M, N, A, B, C)

IF (V, V, 2) GO TO 15

DO (I, I) = -C(1)
DO (I, J) = C(I)
KAP1 = C(I)
KAP2 = C(J)
GO TO 50

**FIND ALPHA AND BETA ARRAYS**

50 IC = 2: N = 4
KAP2 = C(N)
KAP1 = C(N - 1)
M2 = N - 2

DO 10 IC = 3, 2
K = M + 3
M = M + 3
CALL CONPFA(KAP2, KAP1, E, RN, NC)

AL(5) = KAP2/5
B6 = -KAP1/3
ALF = -AL(3)

CALL CONVCT(RET, ALF, RN, GN, NC)

AL(5) = (-1) * AL(3) * GN
IF (BET.LT.0) AL(5) = AL(5) - 1
BET(5) = (-1) * BET(3) * GN
IF (ALF.LT.0) BET(5) = BET(5) - 1
KAP1 = KAP1 + AL(5) - 1 + KAP2 * BET(5) - 1
KAP1 = C(4) + 2

CONTINUE

**NOW COMPUTE MATRIX D**

BE(2* N - 2) = KAP1
BE(1) = -KAP2
DO 120 K1 = 2, N
K = I1 - 1
K3 = K2 - 1
IF (K1.GT.0) K3 = 1
DO 110 RI = 1, K3
IF (RI.GT.1) GO TO 88
R2 = RI - 1
F(R2) = 1

88 IF (K1.EQ.0) GO TO 100
DO 90 S1 = K1, K3

- A24 -
IF (S1 .NE. 1) GO TO 89
F(1) = BE(2*N-(2*S1+1))
GO TO 92

89 CONTINUE
IF (K1 .GE. N) GO TO 105
D(K1, R1) = AL(2*N-(2*K1+1)) * BE(2*N-2*R1) * F(K2)
GO TO 110

100 CONTINUE
D(K1, R1) = BE(2*N-2*R1) * F(K3)

105 CONTINUE
IF (K1 .GE. N) GO TO 115
D(K1, R1) = AL(2*N-(2*K1+1)) * BE(2*N-2*K2)
D(K1, R1) = AL(2*N-2*K1)
GO TO 120

110 CONTINUE
VL = 1 - 1
DO 49 I = 1, N
CONTINUE

49 CONTINUE
FORMAT (28*NPUT C*)
READ (1, *)
IF (Z .LE. -9099) GO TO 52
STOP

C FIND XK*VL THE INITIAL SOLUTIONS
C
32 CALL CONVEX(KAP1, KAP2, PN, SM, HC)
XK = (1, -1, -1, NC+1) * PN 2
VL = (1, -1, -1, (NC+1)) * PN 2
IF (KAP2 .LT. 2, X) VL = Y

C FIND THE CONSTANT VECTOR K
C
CON(1) = XK
GO TO 21

21CONTINUE
CON(K1) = 0(K1, 1) * VL/KAP1

C DETERMINE BOUNDS OF T
C
T(1) = IFLOAT(CON(1))/2(1, 1)
T(1) = (FLOAT(1))/2(1, 1)
IF (T(1) .LE. Y(1)) GO TO 315
VX = Y(1)
VX(1) = N
IF (F .LE. 2, 1) GO TO 351

31 CONTINUE
IF (K .EQ. 0) VX = VX
M = VX
IF (K .LE. 1) VX = VX(K1)

35 CONTINUE
IF (K .GT. 1) VX = VX*(K-1)
M = VX
IF (K .LE. 0) VX = VX(K1)
GO TO 34

34 IF (K .EQ. 0) VX = VX(K1)
M = VX
IF (K .LE. 0) VX = VX(K1)

33 CONTINUE
IF (K .GT. 1) VX = VX*(K-1)
M = VX
GO TO 335

335 CONTINUE
S = F(K, J1)
OS=O(K*J1)
P(K)=P(K*J1)
Q(K)=Q(K*J1)+GS

CONTINUE
MNA=PF(K)
MXA=OF(K)
IF(DD.GE.0) GO TO 345

M'NA=PF(K)
M'XA=OF(K)
MN(K)=FLOAT(CON(K))/DD*MNA
MX(K)=FLOAT(Z)/C(K)-FLOAT(CON(K))/DD+M'N)

IF(MN(K)+LE.MX(K)) GO TO 350

MS=MN(K)
MN(K)=MX(K)

CONTINUE
IF(MN .LT. MN(N+1)) MN(N+1)=MN(N)

M1=N+1

DO 354 J=1,M1

IF(K.EQ.2) M=MX(J)
MS=ABS(M)

C TEST IF ROWN IS ALREADY INTEGER
IF((M=IFIX(MA)),EQ.0.) GO TO 352

C IF MIN IS POSITIVE OR MAX IS NEGATIVE
C ROWN DOWN
IF((K.EQ.2).AND.(M.LE.0.).)
GO TO 352

M1=IFIX(MB)
IF(K.EQ.2)GO TO 360

MN(J)=M1
IF(M1.GT.0)IMX(J)=M1
GO TO 355

IMX(J)=M1
IF(M1.GF.0)IMX(J)=M1

CONTINUE

C SET INITIAL T VALUES AND COUNT T-SET
M1=N+1
N1=1

DO 71 J=1,N1

T(J)=IMN(J)
NT(J)=IMX(J)-IMN(J)+1
M10=M10-NT(J)

CONTINUE

WRITE(1,71)

71 FORMAT(1X,71/'NUMBER OF POINTS SEARCHED/II1")(T(M1)=IMN(M1)-1

C USE SOLUTION X AND CHECK FEASIBILITY
C

WRITE(1,401)

401 FORMAT(1X,402/'FEASIBLE SOLUTIONS X(I)')

J1=0

DO 46 J1=1,N1

IT=1

IF(T(IT).GT.IMX(IT)) GO TO 42

IF(IT.NE.M1) GO TO 45

T(IT)=T(IT)+1
IF(Y(J).LE.IMX(IT)) GO TO 300
IF(Y(IT).GT.IMX(IT)) GO TO 405
CONTINUE
E=0

C SETS FEASIBILITY INDICATORS
C
T(M)=0
DO 520 J=1,N
XA=0
DO 510 K=1,J
XA=XA+C(J,K)*T(K)
510 CONTINUE
X(J)=CON(J)*XA
IF(X(J).GE.0) GO TO 520
J=N
GO TO 525
520 CONTINUE
FE=1
525 CONTINUE
IF(FE.EQ.0) GO TO 80
C IF SOLUTIONS IS FEASIBLE
C
FE=0
DO 620 I=1,M
S(I,1)=A(I,1)*X(1)
DO 610 J=2,M
S(I,J)=S(I,J-1)+A(I,J)*X(J)
610 CONTINUE
IF(S(I,J).LE.B(I)) GO TO 620
I=M
GO TO 630
620 CONTINUE
FE=1
630 CONTINUE
IF(FE.EQ.0) GO TO 80
C PRINT IF FEASIBLE
C
WRITE(1,75)(X(K),K=1,M)
FORMAT(15(2X,I1))
85 J1=J+1
IF(J1.LT.100000) 56 TO 85
J1=0
85 CONTINUE
GO TO 56
END
C
SUBROUTINE CONVGT(I,J,FE,M1,VC)
C THIS FINDS FIRST THE CONTINUED FRACTION OF I,J
C THEN THE (M-1)TH CONVERGENT OF P,Q
C
IMPLICIT INTEGER(4-2)
DIMENSION F(20),G(20),E(20)
IF(I.EQ.J,.EQ.1) GO TO 26
IF(I.EQ.J,.EQ.1) GO TO 27
CALL CONFRA(I,J,E,F,VC)
C
CONVERGENTS
C
P(1)=F(1)
Q(1)=1
P(2)=E(2)*E(1)+1
\[ q(2) = e(2) \]
\[ \text{IF} (N_C > GT.3) \text{GO TO 20} \]
\[ N_C = P(2) \]
\[ Q_N = 0(2) \]
\[ 20 \]
\[ N_S = N_C - 1 \]
\[ \text{DO 25} \]
\[ M = 1 \]
\[ P(M) = C(M) \times P(M - 1) \div Q(M - 2) \]
\[ Q(M) = F(M) \times Q(M - 1) \div Q(M - 2) \]
\[ 25 \text{ CONTINUE} \]
\[ P_N = P(M) \]
\[ Q_N = Q(M) \]
\[ \text{GO TO 30} \]
\[ 26 \]
\[ N_S = 1 \]
\[ N_C = 2 \]
\[ N_C = 2 \]
\[ \text{RETURN} \]

SUBROUTINE CONFRA(A*,E*,E*,PN*,NC*)

C FINDS H.C.F. OF A & E AND CONTINUED FRACTION OF A/B

C IMPLICIT INTEGER(A-Z)
C DIMENSION E(20),R(20)
C NC=1
C I1= IABS(A)
C J1= IABS(B)
C R(1)=1
C IF ((I1.EQ.1).OR. (J1.EQ.1)) GO TO 45

C CONTINUED FRACTIONS

C IF (I1.LE.J1) GO TO 35
C E(1)=I1/J1
C R(2)=I1-E(1)*J1
C GO TO 40

C 35
C E(1)=E(1)
C R(2)=1
C NC=NC+1
C E(NC)=R(NC-1)/R(NC)
C R(NC)=E(NC-1)-E(NC)*R(NC)
C IF (R(NC+1).LT.0.0) GO TO 4

C 40
C NC=NC+1
C E(NC)=R(NC-1)/R(NC)
C R(NC)=E(NC-1)-E(NC)*R(NC)
C IF (R(NC+1).LT.0.0) GO TO 4

C 45
C RETURN

END

SUBROUTINE INPUT(K*,M*,A,B,C)

C INPUTS AX\< KER A IS AN M BY M MATRIX
C X IS THE OBJECTIVE FUNCTION

C IMPLICIT INTEGER(A-Z)
C DIMENSION A(20,20),P(20),C(20)
C WRITE(1,14)
C FORMAT(//"ENTER NO. OF VARIABLES")
C READ(1,14)
C WRITE(1,14)
C FORMAT(//"ENTER COEFFICIENTS C(J)")
C READ(1,14)(C(J),J=1,N)
C WRITE(1,25)
C FORMAT(//"HOW MANY CONSTRAINTS")

- A28 -
0(2) = E(2)
IF (NC GT 1) GO TO 20
PM = P(0)
QN = Q(0)
GO TO 30

20 
N5 = NC - 1
DO 25 M1 = 3 * N5
   P(M1) = F(M1) * P(M1 - 1) + P(M1 - 2)
   Q(M1) = F(M1) * Q(M1 - 1) + Q(M1 - 2)
25 CONTINUE
QN = P(1)
QN = Q(1)
GO TO 30

26 
PM = IABS(I / J) - 1
QN = 1
NC = 2
GO TO 30

27 
PM = 0
QN = 0
NC = 2
RETURN

END

SUBROUTINE CONFR(A, E, E, PN, NC)
FINDS HCF OF A, B AND CONTINUED FRACTION OF A/B

IMPLICIT INTEGER(A-Z)
DIMENSION E(20), R(20)
NC = 1
I1 = IABS(A)
J1 = IABS(B)
R(1) = I1
IF ((I1 LE 1) OR (J1 LE 1)) GO TO 45

CONTINUED FRACTIONS

IF (I1 LE J1) GO TO 30
(1) = I1 / J1
(2) = I1 - E(1) - J1
GO TO 40

30 (1) = 0
(2) = 1
NC = NC + 1
E(NC) = F(NC - 1) / R(NC)
R(NC + 1) = R(NC - 1) - E(NC) * R(NC)
IF (R(NC + 1) NE 0) GO TO 45

R(NC) = R(NC) / R(NC + 1)
GO TO 50

45 R = 1
RETURN

END

SUBROUTINE INP(A, N, A, B, C)
INPUTS A, B, C
B, C IS AN N BY N MATRIX
B, C IS THE OBJECTIVE FUNCTION

IMPLICIT INTEGER(A-Z)
DIMENSION A(20, 20), B(20), C(20)
WRITE (1, 5)
FORMAT ("ENTER NO. OF VARIABLES")
READ (1, *)
WRITE (1, 10)
FORMAT ("ENTER COEFFICIENTS C(J)")
READ (1, 10) (C(J), J = 1, N)
WRITE (1, 25)
FORMAT ("HOW MANY CONSTRAINTS")

- A29 -
ENTER NO. OF VARIABLES 4
ENTER COEFFICIENTS C(J) 2 1 5 3
HOW MANY CONSTRAINTS 4
ENTER COEFFICIENTS A(I,J) 1 -1 5 1
3 1 -1 2
-1 2 1 -1
1 1 1 1
TYPE IN COEFFICIENTS B(I) 10 15 5 8
IS INPUT O.K. ENTER "Y" OR "N" Y
INPUT C 18
NUMBER OF POINTS SEARCHED 5650
FEASIBLE SOLUTIONS X(I)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
86. The referee's estimate was not confirmed by a direct comparison of MINMAX with subroutine LINES by Sklar and Armstrong. In order to clarify the role of the advanced L₂ start, a third program (subroutine SHORT) was included in this comparison. SHORT is a subset of MINMAX which works without the L₂ start. The following CPU times in csec are representative for small and large data sets, respectively. The figures in brackets denote the number of statements in each subroutine.

<table>
<thead>
<tr>
<th>Number of points</th>
<th>MINMAX(121)</th>
<th>SHORT(47)</th>
<th>LINES(149)</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
</tr>
<tr>
<td>450</td>
<td>12</td>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>

The figures show that, in general, no time advantage is gained by applying the L₂ start to large data sets. An optimal code (usually faster than LINES) would combine MINMAX and SHORT, activating the SHORT option for large data sets.
A SQUARE ROOT ALGORITHM

The integration of this, and the numerical series which ensue, provide exercises for the interested reader.

J. M. H. PETERS

Department of Mathematics, Liverpool Polytechnic, Byrom Street, Liverpool L3 3AF

A square root algorithm

MAX PLANITZ

Although the approximation of elementary functions is a well-researched area of mathematics, very little has been published on built-in routines in any particular calculator or computer. Computer manufacturers are, understandably perhaps, reluctant to publicise details about their algorithms. Hewlett-Packard have been less secretive than most and allowed an occasional glimpse behind the scene in their own Journal and various other sources. The following square root routine for one of their machines, the now extinct 2000F series, has appeared in an Open University text on numerical computation [1]. The process of evaluating $\sqrt{x}$ is carried out in four steps:

(i) Determine a real number $t \in [0.25, 1]$, such that $x = 4^k t$, where $k$ is an integer.

(ii) Use the formula

$$y(t) = \begin{cases} 0.27863 + 0.875t & t \in [0.25, 0.5) \\ 0.421875 + 0.578125t & t \in [0.5, 1] \end{cases}$$


(iii) Apply Newton’s method in the form

$$y_{n+1} = (y_n + t/y_n)/2$$

with $y_0 = y(t)$ and $n = 0, 1$.

(iv) Compute $\sqrt{x} = 2^k y_2$. The result is correct to 6 significant figures!!

This algorithm, which seems cumbersome at first sight, is in fact remarkably efficient. To obtain $\sqrt{x}$ to 6 significant figures, a binary computer requires only 2 “long” operations (i.e. multiplications or divisions). These are needed to compute $t/y_n$ in step (iii). Steps (i) and (iv), as well as the division by 2 in step (iii), only involve shifts. Less obviously, step (ii) can be called a “short” operation, since $0.875 = 0.111_2$ and $0.578125 = 0.100101_2$, i.e. only 4 additions and 3 shifts are required to find $y(t)$. 
The selection of an approximating function for step (ii) is an interesting, non-trivial problem. First note that for greater accuracy, the approximation on [0.25, 1] is segmented. Since our computer is assumed to use binary arithmetic, a power of 2 is chosen as the point of subdivision. We require the best linear approximation for $\sqrt{t}$ on each subinterval. The precise meaning of “best approximation” depends on the norm used to measure the “distance” between two functions. For the purposes of evaluating a function on a closed interval, an approximation is as good as its maximum error on that interval. It therefore makes sense to minimize the Chebyshev norm of $y(t) - \sqrt{t}$, i.e. the expression

$$\max_{a \leq t \leq b} |y(t) - \sqrt{t}|.$$

More generally, we make the following definition:

**Definition.** Let $y(t)$ belong to the set $P_{n+1}$ of polynomials whose degree does not exceed $n$. If

$$\max_{a \leq t \leq b} |y(t) - f(t)| \leq \max_{a \leq t \leq b} |x(t) - f(t)|$$

for all $x$ in $P_{n+1}$, then $y$ is called a polynomial of best approximation to $f(t)$.

Polynomials of best approximation are characterised by an alternation property, which was discovered by Chebychev in the 1850s. We state this property for polynomials in $P_{n+1}$; for a proof the reader is referred to [2, p. 75].

**Theorem.** Let $f \in C[a, b]$ and $y \in P_{n+1}$. Then $y$ is a polynomial of best approximation to $f$ on $[a, b]$ if and only if there are $n + 1$ points $t_0, \ldots, t_{n+1}$ in $[a, b]$, with $t_0 < \ldots < t_{n+1}$, such that

$$|y(t_i) - f(t_i)| = \max_{a \leq t \leq b} |y(t) - f(t)|$$

and $y(t_i) - f(t_i)$ alternates in sign for $i = 1, \ldots, n + 1$.

Although $y$ is known to exist, uniquely in fact, for any continuous $f$, it remains an open question whether there is a general finite step algorithm for the construction of best approximations. (An iterative method can be found in the book by Cheney [2, p. 96].) Such a construction is feasible, however, if $f$ has a second derivative whose sign does not alter on $[a, b]$ and if $y$ is linear. These conditions are satisfied by $f(t) = \sqrt{t}$ and $y(t) = a_0 + a_1 t$, and we proceed to deduce the values of $a_0$ and $a_1$ from the alternation property. First consider the interval [0.25, 0.5]. The theorem tells us that the error function

$$y(t) - f(t) = a_0 + a_1 t - \sqrt{t}$$

attains its maximum in the following way:

$$y(t_i) - f(t_i) = a_0 + a_1 t_i - \sqrt{t_i}$$

i.e.

$$y(t) = a_0 + a_1 t$$

Since $f''(t) = -\frac{1}{4t^{3/2}}$, if $f$ satisfies the alternation property, then

$$a_1 = \frac{f''(t_0)}{2}$$

(where $a_0$ is the constant term of the resulting polynomial of best approximation $y(t)$.)

This time, we show that the maximum error is attained at the following value:

$$g(t) = \frac{a_0}{a_1}$$

where $m = g(t_0)$.

Moreover, $a_0 = m a_1$. Hence

$$a_0 = \frac{m a_1}{a_1} = m a_1 = \ldots$$

This time, the construction is as follows:

$$a_0 = \frac{m a_1}{a_1} = \ldots$$

where $m = g(t_1)$ and

$$a_0 = \frac{m a_1}{a_1} = \ldots$$

Moreover, $a_0 = \ldots$.
is an interesting, e approximation to use binary n. We require the precise meaning to measure the rating a function maximum error on byshev norm of

als whose degree

imation to f(t), y an alternation 3). We state this referred to [2, p.

y polynomial of best points t_1, \ldots, t_{n+1}

r continuous f, it step algorithm for nod can be found usable, however, if and if y is linear. t_0 + a_1 t, and we imation property, us that the error

A SQUARE ROOT ALGORITHM

attains its extrema for three values t_1, t_2, t_3 \in [0.25, 0.5], with t_1 < t_2 < t_3. It follows that

\[ y'(t_2) - f'(t_2) = a_1 - 1/(2\sqrt{t_2}) = 0 \]

i.e.

\[ a_1 = 1/(2\sqrt{t_2}). \]

Since f''(t) < 0 on [0.25, 0.5], y'(t) - f'(t) cannot vanish for a second value of t inside the interval, i.e. t_1 = 0.25 and t_3 = 0.5. Using the alternation property, we can now write

\[ a_0 + 0.25a_1 - 0.5 = a_0 + 0.5a_1 - \sqrt{0.5} = \sqrt{t_2} - a_0 - t_2a_1, \]

(where a_1 = 1/(2\sqrt{t_2}). Solving, we find a_0 = 0.297335, a_1 = 0.828427, t_2 = 0.363277. We then apply the same technique to the interval [0.5, 1]. The approximation formula obtained in this way is

\[ y^*(t) = \begin{cases} 
0.297335 + 0.828427t & t \in [0.25, 0.5) \\
0.420495 + 0.585786t & t \in [0.5, 1). 
\end{cases} \]

with approximate errors of 0.004 on [0.25, 0.5) and 0.006 on [0.5, 1).

Some of the accuracy of y^* is now sacrificed in order to reduce the execution time of step (ii). This is done by approximating the coefficients of t by numbers whose binary expansions contain only three non-zero bits. The resulting formula is

\[ y(t) = \begin{cases} 
a_0 + 0.875t & t \in [0.25, 0.5) \\
b_0 + 0.578125t & t \in [0.5, 1). 
\end{cases} \]

It is not clear how the coefficients a_0, b_0 were originally obtained, but the following approach leads to similar, in fact slightly better, results. To adjust first the value of a_0, we apply our theorem to the function

\[ g(t) = \sqrt{t} - 0.875t. \]

This time, the best approximation is a constant, and a simple argument will show that this constant is given by

\[ a_0 = (m + M)/2, \]

where m = min g(t) and M = max g(t) on [0.25, 0.5]. Since n = 1, we have to show that the error function alternates on two points. If we define t_1, t_2 by m = g(t_1) and M = g(t_2), then

\[ a_0 - g(t_1) = (M - m)/2 \quad \text{and} \quad a_0 - g(t_2) = (m - M)/2. \]

Moreover,

\[ |a_0 - g(t_i)| = \max |a_0 - g(t)|, \quad i = 1 \text{ or } 2, \quad 0.25 \leq t \leq 0.5, \]
i.e. \( a_n = (m + M)/2 \) satisfies the alternation property of the theorem. It is now easy to show that \( m = 0.2696068 \) and \( M = 0.2857143 \). Hence \( a_n = 0.277661 \). This gives a maximum absolute error of 0.008 on \([0.25, 0.5] \), compared with an error of 0.007 in Hewlett Packard's original formula. We similarly find \( b_n = 0.425008 \) with an error of 0.007, which represents an improvement by 0.003. A further reduction in the number of long operations could be achieved by introducing a \( k \)-fold segmented approximation to \( \sqrt{t} \), with \( k > 2 \).

The basic strategy of our square root routine is to reduce execution time by using shifts, additions and recall of prestored constants in preference to long operations. With the dramatic fall in the cost of computer memory, this technique has become widely used, especially in a group of algorithms referred to as "CORDIC". The **Coordinate Rotation Digital Computer** was designed by J. E. Volder in the 1950s. Its purpose was to perform real-time navigational calculations at high speed. We conclude this article by outlining a CORDIC algorithm for \( \tan \alpha \).

Let \( x = \sigma_1 a_1 + \cdots + \sigma_n a_n + \epsilon_n \) (0 < \( x \) < \( \pi/2 \)), where \( \sigma_1 = \pm 1 \) and \( \epsilon_n \) is small. The angles \( a_1, \ldots, a_n \) (\( n \) depends on the accuracy required) will be defined later; they represent the rotations from which the method takes its name. Using the usual addition formulae for the sine and cosine functions, we obtain

\[
\begin{align*}
\cos x &= \cos a_1 \left( 1 - \tan a_1 \right) \cos (x - a_1) \\
\sin x &= \sin a_1 \left( 1 - \tan a_1 \right) \sin (x - a_1)
\end{align*}
\]

and

\[
\begin{align*}
\cos (x - a_1) &= \cos a_2 \left( 1 - \tan a_2 \right) \cos (x - a_1 - a_2) \\
\sin (x - a_1) &= \sin a_2 \left( 1 - \tan a_2 \right) \sin (x - a_1 - a_2)
\end{align*}
\]

Denoting \( \cos a_i \) by \( c_i \), \( \tan a_i \) by \( t_i \) and \( \left( \begin{array}{c} 1 \\ t_i \\ 1 \end{array} \right) \) by \( T_i \), we can write

\[
\begin{align*}
\cos x &= c_1 c_2 c_n T_1 T_2 \cdots T_n \cos (x - a_1 - a_n) \\
\sin x &= c_1 c_2 c_n T_1 T_2 \cdots T_n \sin (x - a_1 - a_n)
\end{align*}
\]

Continuing this process we find

\[
\begin{align*}
\begin{array}{c}
\frac{\cos x}{\sin x} = c_1 c_n T_1 T_n \\
\frac{\cos x}{\sin x} = c_1 c_n T_1 T_n \frac{\cos \epsilon_n}{\sin \epsilon_n}
\end{array}
\end{align*}
\]

or

\[
\begin{align*}
\begin{array}{c}
\frac{\cos x}{\sin x} = T_1 T_n \frac{\cos \epsilon_n}{\sin \epsilon_n} \\
\frac{\cos x}{\sin x} = T_1 T_n \frac{\cos \epsilon_n}{\sin \epsilon_n}
\end{array}
\end{align*}
\]

where \( c = c_1 \cdots c_n \) and \( \frac{\cos \epsilon_n}{\sin \epsilon_n} \approx 1 \). Note that \( c \) need not be evaluated since \( \tan x = \sin x / \cos x \).

A clever choice of the angle \( a_i \) enables the values of \( t_i \) and \( a_i \) to be chosen so that the \( T_i \) are available as special processor instructions, including the square root. The algorithm can be found in special processor with the maximum execution time of the number of long divisions.

**References**

1. Numerical computation, Un
2. E. W. Cheney, Introduction
3. J. E. Volder, The CORDIC
4. J. S. Walther, A Unified alg

**Data permutation test**

JOHN K. BACKH

This article is about a permuta

**Systematic data permutation test**

The method is introduced, and it is perceived. Examples 1 i

**Example 1.** A horticultu
since \( \tan x = \sin x / \cos x \). The power of the CORDIC method lies in the clever choice of the angles \( a_1, \ldots, a_n \). We define

\[
a_i = \tan^{-1} 2^{1-i},
\]

\( i = 1, \ldots, n \), which enables us to compute \( \tan x \) by shifts and additions only, if the values of \( t_i \) and \( a_i \) are prestored in read-only memory.

CORDIC techniques have been developed for other elementary functions, including the square root function. An interesting unified CORDIC algorithm can be found in a paper by J. S. Walther [4]. By constructing a special processor with three parallel adders, Walther achieved the same maximum execution time of 100 \( \mu \text{sec} \) for square roots, multiplications and divisions.

References

1. Numerical computation, Unit 10, Approximation II. The Open University (1976).

MAX PLANITZ
Thames Polytechnic, Division of Mathematics, Wellington Street, London SE18 6PF

Data permutation tests
JOHN K. BACKHOUSE

This article is about a powerful method of carrying out tests of statistical significance. Its rationale depends on simple ideas of probability and is suitable for pupils at school. Practical applications require a considerable amount of computation but now that schools are equipped with computers this should not be a problem.

Systematic data permutation

The method is introduced by means of examples; in these the sample sizes are kept unrealistically small so that the method used can be more easily perceived. Examples 1 to 3 illustrate different types of data permutation appropriate to three common situations.

Example 1. A horticulturalist has managed to raise 5 seeds of a rare plant and wishes to compare two methods of feeding the plants. He assigns at
random 2 seeds to method A and 3 to method B, and finds that the percentage increases in heights of the plants are 53, 97 for method A, and 3, 6, 11 for method B.

There are 10 ways in which the 5 seeds could have been divided into two groups of sizes 2 and 3; since the sampling was random all 10 ways are equally likely. On the null hypothesis that the percentage increase in height is independent of the method of feeding, each of the following results is equally likely. Here $n_A, n_B$ are the numbers in groups A and B; $m_A, m_B$ are the means of the groups; $\Sigma X_A, \Sigma X_B$ are the totals for the two groups and $t$ is calculated from the formulae

$$t = (m_A - m_B)\left(\frac{s^2}{n_A} + \frac{s^2}{n_B}\right)^{1/2}$$

where

$$s^2 = \frac{(\Sigma(X_A - m_A)^2 + \Sigma(X_B - m_B)^2)}{(n_A + n_B - 2)}$$

<table>
<thead>
<tr>
<th>Method A</th>
<th>Method B</th>
<th>$m_A - m_B$</th>
<th>$\Sigma X_A$</th>
<th>$\Sigma X_B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 6</td>
<td>11 53 97</td>
<td>-49.2</td>
<td>-1.53</td>
<td>9</td>
</tr>
<tr>
<td>3 11</td>
<td>6 53 97</td>
<td>-45.0</td>
<td>-1.32</td>
<td>14</td>
</tr>
<tr>
<td>6 11</td>
<td>3 53 97</td>
<td>-42.5</td>
<td>-1.21</td>
<td>17</td>
</tr>
<tr>
<td>3 53</td>
<td>6 11 97</td>
<td>-10.0</td>
<td>-0.24</td>
<td>56</td>
</tr>
<tr>
<td>6 53</td>
<td>33 11 97</td>
<td>-7.5</td>
<td>-0.18</td>
<td>59</td>
</tr>
<tr>
<td>11 53</td>
<td>3 6 97</td>
<td>-3.3</td>
<td>-0.08</td>
<td>64</td>
</tr>
<tr>
<td>3 97</td>
<td>6 11 53</td>
<td>26.7</td>
<td>0.67</td>
<td>100</td>
</tr>
<tr>
<td>6 97</td>
<td>3 11 53</td>
<td>29.2</td>
<td>0.74</td>
<td>103</td>
</tr>
<tr>
<td>11 97</td>
<td>3 6 53</td>
<td>33.3</td>
<td>0.87</td>
<td>108</td>
</tr>
<tr>
<td>53 97</td>
<td>3 6 11</td>
<td>68.3</td>
<td>4.10</td>
<td>150</td>
</tr>
</tbody>
</table>

The observed result has the largest value of $t = 4.10$, and so the probability that $t$ should be as large as this (or greater) is 1/10. One would expect the experimenter to specify in advance a level of probability at which to reject the null hypothesis. However, the examples are artificial and we shall not pretend to do this. The experimenter should also be clear whether a one- or two-tailed test is appropriate but in this case a two-tailed test, and a one-tailed test in favour of method A, have the same probability.

It was not necessary to list the results in full but the table does help to illustrate some of the following points:

(1) We considered every possible pair of samples of the specified sizes which could have been drawn from the 5 values of the variate. This is one example of systematic data permutation.

(2) We made no assumption that the original sample was random, only that the assignment to the methods was made on a random basis. The test which followed is an example of a randomisation test. As really random samples are very rare, there is a clear advantage in such tests.
A COMPARISON OF THE ALGORITHMS FOR AUTOMATED DATA ADJUSTMENT AND MATERIAL BALANCE AROUND MINERAL PROCESSING EQUIPMENT

V.R.Voller and M. Planitz

K.J. Reid
Mineral Resources Research Centre, University of Minnesota, Minneapolis, MN 55455, U.S.A.

Abstract. The data adjustment problem that occurs in the mass balance of a mineral processing plant is outlined. By means of a simple problem of one process unit, with one feed and two output streams, the basic techniques of data adjustment and material balance are presented. Algorithms that employ these techniques are then examined from the programming point of view, with particular emphasis on microcomputer applications.

Keywords. Automated material balance; Computer aided circuit analysis; Minerals industries; Optimisation.

INTRODUCTION

One area where the increasing availability of micro-computers can make a large impact is in the automation of tedious engineering calculations. An example is the calculation of a material balance in a mineral processing plant (i.e. what flows where and how much of it flows there). The problem associated with this task is that the measured data give rise to an overdetermined inconsistent set of equations. Therefore in a large process flow-sheet, consisting of multiple process streams and equipment, achieving a reliable material balance depends on the mineral engineer's experience in making the necessary adjustments to the measured assay data. For complicated flowsheets this may demand many man hours of work.

Since the early 70's, following in the wake of the work of Wiegel (1972), there have been a number of computer packages designed to produce a material balance around a mineral processing circuit from inconsistent measurements (see Reid et al (1982) for a review). The basic principle in all of these packages is the automated adjustment of the input data via the minimisation of a weighted sum of squares.

It is true to say that the majority of available packages are intended for implementation on main frame computers. There is, however, a strong argument to be made for fitting data adjustment and material balance routines on to microcomputers. For relatively simple problems this has already been achieved, Reid and Voller (1983). In fact, a data adjustment algorithm for hydrocyclone size data has been implemented on a hand held programmable calculator, Voller and Ryan (1983).

The aim of this paper is to return to the "fundamental" material balance problem, that is the balance around a single three stream (one feed, two products) process unit. In this way the basic engineering and mathematical problems can be clearly stated and examined. Furthermore, a comparative study of the possible techniques for solving the data adjustment and material balance problem in terms of computer and engineering requirements can be made.

THE MATERIAL BALANCE PROBLEM

Consider a single processing unit with a feed stream (1) and two product streams (2) and (3) shown schematically in Fig. 1.

![Fig. 1. A three stream process unit.](image)

Further, assume that each stream has been assayed for n distinct species (e.g. % copper, % weight of particles with size less than 100mm, etc.). In order to close the material balance around this process unit, i.e. calculate the mass flows in each stream n+1 mass balance equations have to be satisfied, viz.

\[ M_1 - M_2 - M_3 = 0 \]  

(1)
and
\[ M_1 x_1^k - M_2 x_2^k - M_3 x_3^k = 0 \quad (k=1(1)n) \] (2)

where \( M_i \) is the mass flow in stream \( i \) and \( x_i^k \) is the assayed value of species \( k \) on stream \( i \). In terms of the feed mass flow rate \( M_f \), solutions to equations (1) and (2) will be of the form
\[ M_2 = M_f (x_1^k - x_3^k/x_2^k - x_3^k) \quad (3) \]

the so called "two product balance" formula. The major drawback in using equation (3) is that due to sampling errors etc. the values of \( M_2 \) and \( M_3 \) will depend on the assay species used. The data in Table 1 represents a typical set of inconsistent measured assays. Using

<table>
<thead>
<tr>
<th>TABLE 1 Assay Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>(1)</td>
</tr>
<tr>
<td>(2)</td>
</tr>
<tr>
<td>(3)</td>
</tr>
</tbody>
</table>

each of the two assay species in turn in two product balance formula will give marked differences in the estimates for \( M_2 \) and \( M_3 \), see Table 2.

<table>
<thead>
<tr>
<th>TABLE 2 Two Product Balance Massflow Estimates.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assay Type</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

To solve this system of overdetermined inconsistent equations the measured assayed data need to be adjusted. A technique that adjusts the assayed data would have to perform in a constant manner, i.e., the criteria by which adjustments are made should be independent of the problem. A suitable criterion for adjustment would be one in which the adjusted assays: (1) satisfied all the mass balance equations; and (2) the adjustments made were in some sense the minimum possible. In addition, adjustments of the assays would also have to take account of the relative errors generated in the sampling and assaying. Errors of this type are often assumed to be normal, unbiased and independent. With these assumptions the minimum adjustment may be taken to occur when
\[ J = \sum_{j=1}^{n} \left( x_j^k - x_j^{k} \right)^2 \rightarrow \text{a minimum} \quad (4) \]

In the case of a single three stream process unit we have
\[ J^k = \sum_{j=1}^{3} w_j x_j^{k} - x_j^{k} x_j^{k} \quad (5) \]

where \( x_j^{k} \) is the adjusted assay on stream \( j \) and \( w_j^k \) is a weighting factor. The material balance problem may be stated as: find the set of assays \( x_j^{k} \) \((j=1(1)3, k=1(1)n)\) that minimise \( J \) subject to the mass balance constraints equations (1) and (2). On defining a relative mass flow rate \( D \) to be
\[ D = M_1/M_2 \]

the \( n+1 \) constraints on the minimisation of \( J \) can be reduced to the \( n \) constraints
\[ x_1^k - D x_2^k - (1-D)x_3^k = 0 \quad (6) \]

THE METHODS

There are a number of alternative approaches for solving the optimisation problem defined by equations (4)-(6). In packages designed for the minerals industry methods ranging from Lagrange multipliers to direct search techniques have been employed, Mular (1980). In applications to the simple one process test problem some of the subtlety of these methods is lost. Nevertheless the basic differences of the varying approaches can be clearly illustrated.

The most common approach of solving equations (4)-(6) is by the introduction of Lagrange multipliers, \( \lambda^k \), such that the mass balance constraints and functional \( J \) are combined in a single functional, viz
\[ L = J + \sum_{j=1}^{n} \left( x_j^{k} - D x_2^k - (1-D)x_3^k \right) \quad (7) \]

which requires minimisation. This approach has been used in a number of large mineral processing material balance packages. These packages, however, employ a variety of methods to minimise equation (7). Wieg (1972), Cutting (1976) and Lajomton and Wilson (1979) use a gradient method deriving a set of non-linear equations which are solved by a linearising iterative technique. Smith and Ichiyen (1973) and Hockings and Callen (1977) also employ the gradient method coupled with a search over the independent relative massflows in the circuit. Hodouin and Everell (1980) employ a so-called "hierarchical procedure" in which the problem is decomposed and a combination of gradient, search, and Newton-Raphson methods are adopted for maximum efficiency.

For the case of a balance around a single process unit application of a gradient method (ie differentiation w.r.t. each unknown) results in a set of \( 4n+1 \) equations viz,
\[ 2w_j x_j^k - x_j^k + g_j x_j^k = 0 \quad (8a) \]
\[ \sum_{j=1}^{3} g_j x_j^k = 0 \quad (8b) \]
where $q_1 = -1$, $q_2 = D$, and $q_3 = 1-D$. In terms of $D$ equations (8) give
\[ x_j^k = x_j^k + g_{jk}^k/(w_j^k) \] (9)
where
\[ r_j^k = x_j^k - DX_j^k - (1-D)X_j^k \] (10)
is referred to as the residue or imbalance equation and
\[ h_j^k = 1/w_1^k + D^2/w_2^k + (1-D)^2/w_3^k \] (11)
On substitution of equation (9) into equation (8) the following polynomial in $D$ is obtained
\[ \sum_{j=1}^n \frac{1}{w_j^k} \left[ x_j^k - x_j^k - \left( D - (1-D)X_j^k \right) \right] = 0 \] (12)
Solution of this polynomial, via Newton-Raphson for example, will give the value of $D$ that minimises $L$. The corresponding adjusted assays may then be calculated from equation (9). This method will be referred to as LMP for Lagrange Multiplier Polynomial.

The value of $D$ that minimises $L$ may alternatively be found via a search technique. For any choice of $D$ corresponding minimum adjusted assays may then be calculated from equation (9). Therefore on performing a search on $D$ calculating adjusted assays at each step the values of $D$ and $X_j^k$ that minimise $L$ may be found. This method will be referred to as LMS for Lagrange Multiplier Search.

Minimisation of the weighted sum of squares
\[ J^* = \sum_{j=1}^n w_j^k (x_j^k - x_j^k)^2 \] (13)
where the weighting factor $w_j^k = 1/h_j^k$ provides yet one more way of determining a "best" value for $D$. A gradient method to minimise $J^*$ gives
\[ D = 1 - \sum_{j=1}^n \frac{w_j^k (x_j^k - x_j^k)^2}{\sum_{j=1}^n \frac{w_j^k (x_j^k - x_j^k)^2}{1}} \] (14)
and this value may be substituted into equation (9) to calculate the set of adjusted assays that along with $D$ minimise $L$ and hence solve the material balance problem. This method will be referred to as MMR for minimum of weighted residues.

An alternative, but similar, approach to using Lagrange multipliers is the use of penalty functions, Walsh (1976). An optimisation technique that has not yet been employed in the solution of material balance problems. Using penalty functions the solution of the material balance problem defined by equations (4) to (6) reduces to minimizing the functional
\[ L^* = J + K \sum_{j=1}^n \frac{D_j^k}{w_j^k} - DX_j^k - (1-D)X_j^k \] (15)
where $K$ is a large positive constant. The constant $K$ may be regarded as a "numerical" Lagrange multiplier. Its role is to ensure that in the minimisation of $L^*$ selections of $D$ and $X_j^k$ that contravene the mass balance constraints are penalised by introducing a large constant in $L^*$. The major advantage in the penalty function approach is that the number of unknowns in the problem are reduced by $n$ (i.e. there are no unknown multipliers). In a large problem this could be significant. In the single process unit balance under consideration here, however, a gradient method to minimise $L^*$ gives equations for the adjusted assays in the form
\[ x_j^k = x_j^k + g_{jk}^k/w_j^k \] (16)
with the "best" value for $D$ calculated from the polynomial
\[ \sum_{j=1}^n \frac{w_j^k}{1+Kw_j^k} \left( x_j^k - x_j^k - \left( D - (1-D)X_j^k \right) \right) = 0 \] (17)

For large $K$, equations (16) and (17) will give values for $D$ and $X_j^k$ very close to those obtained from equations (9) and (12). Hence for the single three stream process unit there is no advantage in using penalty functions.

**TABLE 3 Hierarchical Search Technique**

| STEP 1 | search on D to minimise $J^*$
|        | (Powell Quadratic Interpolation PQI)
|        | Walsh (1976) |
| **"Best" D** | |
| STEP 2 | for each of the n assays one variable at a time PQI search to minimise $J^*$ subject to mass balance constraint Eq(6) |
| **"Best" $x_j^k$** | |
| Solution to Problem. | |
All the above techniques for solution of the material balance problem employ differentiation. This problem may be solved without resorting to derivatives. Table 3 shows the main steps in a hierarchical direct search routine for the solution of the material balance problem defined by equations (4)-(6). This method will be referred to as DSM for direct search method.

RESULTS

The data of Table 1 is reproduced in Table 4 along with a typical error model, consisting of the percent relative standard deviations associated with each measurement, \( \sigma_i \). It is this value which is used to determine the weighting factors, ie, \( \omega_k = \sigma_{ik}^2 / \sum_{j=1}^{n} \sigma_{jk}^2 / 100 \). Table 4 shows the main steps in a hierarchical direct search routine for the solution of the material balance problem defined by equations (4)-(6). This method will be referred to as DSM for direct search method.

The direct search method, ie, DSM, is inferior in all departments. Obviously the search technique used in this method could be improved. It is difficult, however, to see a ten fold improvement in CPU time on using an alternative search technique.

DISCUSSION

From the results the MWR method looks promising. The limiting nature of the test problem must be considered, however. It is possible that alternative approaches may be more appropriate for large scale problems. This point requires some investigation before the MWR method can be confidently used in a full scale microcomputer package.

Another area of interest worth exploring is the choice of adjustment criteria. In all methods so far discussed the criterion has been that of minimisation of a sum of squares. In a case where the errors in the measured data are not normally distributed, it may be more appropriate to find a "minimax" or "least absolute sum" solution. As an example of the use of these criteria the values

\[
J_a = \sum |x_j^k| \quad \text{and} \quad J_m = \max |x_j^k|
\]

have been minimised, using the test data in Table 4, to find "best" values for the relative mass flow rate \( D \). These values are compared with the values of \( D \) obtained by minimising the weighted sum of squares \( J_a \) and an unweighted sum of squares in Table 6.

The results in Table 6 indicate that the values of \( D \) derived from minimising \( J_a \) and \( J_m \) bound the sum of squares solutions which suggests that use of these alternative criteria provides upper and lower bounds on the least squares results.

CONCLUSIONS

In this paper techniques of solving a simple mineral processing material balance problem have been investigated. Some of the optimisation techniques tested are currently used in automated material balance computer packages. Others (eg, penalty functions) have not been used in the solution of mineral processing material
balance problems before. The main conclusion that can be drawn from the study is that there are efficient means of solving material balance problems that as of yet have not been exploited. It is these methods, in particular the MWR method coupled with the penalty function approach that need to be developed in building microcomputer software for solution of material balance problems in the minerals industry.

REFERENCES


