Z. Wahrscheinlichkeitstheorie verw. Geb. 5, 55—70 (1966)

A Quadratic Programming Algorithm*

JOHAN PHILIP

Received Dezember 16, 1964 / January 17, 1966

Abstract. Given a point d and a convex polyhedron or polyhedral cone in a real complete inner product space. We shall describe a numerical method to find a point in the polyhedron (cone) which has minimum distance to d. The characteristics of our method are the description of the polyhedron (cone) by its extreme points (rays) and the introduction of a one-parameter family of problems including a trivially solvable problem and the given problem. The knowledge of the solution of the problem corresponding to one value of the parameter makes it easy to find a larger parameter value for which the solution can again be found. Starting with the trivially solvable problem, the given problem is reached in a finite number of steps. Computational experience shows that the computation time is about proportional to the product of the dimension of the space and the number of extreme points in the polyhedron, when these two quantities are of the same order of magnitude.

1. Introduction

Define an inner product in R^s by $(x, y) = x^\top C y$, where C is a positive semidefinite $(s \times s)$-matrix and let $\|x\| = \sqrt{(x, x)}$. When C is strictly positive definite, this is a norm and when it is only semidefinite, it is a seminorm. Let $A = \{a_1, a_2, \ldots, a_N\}$ be a finite set of points in R^s, d a given point (in R^s) and consider the problem

$$\inf\{\|x - d\| : x \in \operatorname{conv}(A)\}, \tag{I}$$

where conv stands for the convex hull of.

We shall describe a numerical method (algorithm) to solve (I) and also a slight modification of it that solves

$$\inf\{\|x - d\| : x \in \operatorname{cone}(A)\}, \tag{II}$$

where cone (A) means the convex cone with vertex at the origin that is generated by A (that is the conical hull of A).

Our method is constructed to handle problems in which the constraining polyhedron is described by its extreme points and not as the intersection of halfspaces, which is the description used in most other quadratic programming methods. See e.g. P. WOLFE, who describes his own and other methods in [*1*]. See also HOUTHAKKER [*2*], who has presented a technique resembling ours. Of course, our formulations can be transformed to such with halfspaces, but we think that each kind of problem shall be solved by a method that takes advantage of the special character of its formulation.

Our interest in quadratic programming problems formulated as (I) and (II) originates from a study of the wide class of problems that arise when one has to extract information from data obtained in measuring positive quantities. We

* The Swedish Natural Science Research Council has partly sponsored the work.

have shown in [3] how such problems in many cases are least square problems with inequalities as subsidiary conditions and that these problems have the forms (I) and (II), that is, they are quadratic programming problems. In a second paper [4], we have shown how an approximate solution with error estimate of a problem like (I) but with nonfinite A can be obtained by solving (I) with A as a suitably chosen finite subset of the originally given A.

The description of the algorithm and the proofs will be carried through in detail for problem (I) when $\| \ \|$ is a norm. The same algorithm works when $\| \ \|$ is a seminorm, but a completion of the proofs is needed. The seminorm case is treated in section 8. The change of the algorithm needed for problem (II) is described in section 7.

Remark 1. By a solution to (I) resp. (II), we mean a point $x_d \in \operatorname{conv}(A)$ resp. $\operatorname{cone}(A)$ that satisfies $\|x_d - d\| \leqq \|x - d\|$ for all $x \in \operatorname{conv}(A)$ resp. $\operatorname{cone}(A)$. When $\| \ \|$ is a norm, x_d is unique.

Remark 2. It is only the number of points N in A that must be finite to assure the finiteness of the algorithm. Since it is only the inner products (a_i, a_m) and (d, a_m) that are used in the calculations, our method can even be used in an infinite-dimensional real Hilbert space. The "dimension of the calculations" is the smaller of the numbers s and N.

2. A General Outline of the Quadratic Programming Method for Problem (I)

We shall describe a "continuity method" which considers a one-parametric set (with parameter λ) of problems of type (I):

$$\inf \{\|x - c - \lambda(d - c)\| : x \in \operatorname{conv}(A)\}, \qquad (\mathrm{I}_\lambda)$$

where c is a point in R^s.

For $\lambda = 1$ (I_λ) equals (I). We shall choose c suitably so that we know the solutions of (I_0) and can "continue" the problem and its solution across the interval $0 \leqq \lambda \leqq 1$ to get the solution of (I). We shall show that this interval can be divided by a finite number of points λ_i

$$0 = \lambda_0 \leqq \lambda_1 \leqq \lambda_2 \leqq \ldots . \leqq \lambda_t = 1,$$

such that the solution of (I_λ) is an affine function of λ in each closed interval $(\lambda_i, \lambda_{i+1})$. Moreover, if $f_i(\lambda)$ are these affine functions, we shall show that it is fairly easy to determine λ_{i+1} and $f_{i+1}(\lambda)$ when $f_i(\lambda)$ is known.

3. Proofs for Problem (I) when $\| \ \|$ is a Norm

It is well known that our problems have unique solutions when $\| \ \|$ is a norm. This uniqueness makes the proofs neater and easier to understand in the norm than in the seminorm case. This is the reason why we confine ourselves to the norm case in this section. The changes in the proofs and definitions needed for the seminorm case are small and postponed to section 8.

We shall introduce some definitions and notations, which all depend on the set $A = \{a_1, a_2, a_3, \ldots, a_N\}$.

1. By an index set, we mean a subset of $\{1, 2, 3, \ldots, N\}$. An index set $K = \{m_1, m_2, \ldots, m_p\}$ is defined to be degenerate if it is possible to find numbers $\varkappa_m$, not all zero, such that $\Sigma_m \varkappa_m = 0$ and $\Sigma_m \varkappa_m a_m = 0$, e.g. if the points a_m $(m \in K)$ are affinely dependent.

2. An index set K is nondegenerate if it is not degenerate.

Note: Degeneracy of K implies linear dependence of the points a_m, $(m \in K)$. Thus, linear independence implies nondegeneracy.

3. For an index set K, we define

$$L(K) = \{z : z = \Sigma_m \varkappa_m a_m, \Sigma_m \varkappa_m = 1, m \in K\},$$

that is the affine variety spanned by $\{a_m : m \in K\}$.

4. We write $C(K)$ for the interior relative to $L(K)$ of $\operatorname{conv}(a_m : m \in K)$. If there is only one $m \in K$, $C(K) = L(K) = a_m$.

$$C(K) = \{z : z = \Sigma_m \varkappa_m a_m, \varkappa_m > 0, \Sigma_m \varkappa_m = 1, m \in K\}$$

The sets $C(K)$ may overlap.

5. We consider the solution x_d of (I) as a function of d and define the nonlinear operator P by $P: d \to x_d$. We call x_d the projection of d and P the projection operator.

6. For an index set K, we define

$$B(K) = \{z : Pz = \Sigma_m \varkappa_m a_m, \varkappa_m > 0, \Sigma_m \varkappa_m = 1, m \in K\}.$$

$B(K)$ is the set of all points whose projection on $\operatorname{conv}(A)$ are in $C(K)$. Since A is finite, there are only a finite number of sets $B(K)$. When two sets $C(K)$ overlap, the corresponding $B(K)$ do so too.

We list now some more or less well known propositions and give also the proofs, since we want to refer to these proofs when we generalize to the case of a seminorm.

Proposition 1. *In s-dimensional space, a set K of more than $s+1$ indices is degenerate.*

Proof. Let K contain $s+2$ indices among which n is one. Consider the $s+1$ points $a_m - a_n$ $(m \neq n)$. Since $s+1$ points are linearly dependent, there exist ν_m, not all zero, such that $\Sigma_m \nu_m (a_m - a_n) = 0$, $(m \neq n, m \in K)$. Putting $\nu_n = -\Sigma_m \nu_m$, we get $\Sigma_m \nu_m a_m = 0$ and $\Sigma_m \nu_m = 0$ $(m \in K)$. QED.

Proposition 2. *The barycentric representation of a point $x \in L(K)$ is unique if K is nondegenerate.*

Proof. Assume two representations

$$x = \Sigma_m \mu_m a_m, \quad \Sigma_m \mu_m = 1, \quad (m \in K)$$
$$x = \Sigma_m \nu_m a_m, \quad \Sigma_m \nu_m = 1, \quad (m \in K).$$

Subtraction gives

$$0 = \Sigma_m (\mu_m - \nu_m) a_m, \quad \Sigma_m (\mu_m - \nu_m) = 0,$$

which implies $\mu_m = \nu_m$ when K is nondegenerate. QED.

Proposition 3. *A point z belongs to $B(K)$ if and only if there exists a point x (intended to be Pz) satisfying:*

1. $x = \Sigma_m \mu_m a_m, \quad \mu_m > 0, \quad \Sigma_m \mu_m = 1, \quad (m \in K)$
2. $(z - x, a_n - x) \leqq 0$ for $1 \leqq n \leqq N$.

Proof. First, it is well known (see e.g. [*5*]) that 2. is the necessary and sufficient condition for x to be the projection of z on $\operatorname{conv}(A)$ also in the semidefinite case. If $z \in B(K)$, we take $x = Pz$. Then x satisfies 1. by the definition of $B(K)$ and 2. since it is the projection on $\operatorname{conv}(A)$. Conversely, if there exists a point x satisfying 2., x is the projection of z on $\operatorname{conv}(A)$, that is $x = Pz$. Since x satisfies 1., we have $z \in B(K)$. QED.

Note: We have $(z - x, a_m - x) = 0$ for $m \in K$.

Proposition 4. *The operator P is normdecreasing, that is*

$$\|Px - Py\| \leqq \|x - y\|.$$

Proof. By 2. of prop. 3 we have

$$(x - Px, Py - Px) \leqq 0$$

and

$$(y - Py, Px - Py) \leqq 0.$$

Changing the signs in the first inequality and adding it to the second, we obtain

$$(y - x, Px - Py) + \|Px - Py\|^2 \leqq 0,$$

or

$$\|Px - Py\|^2 \leqq (x - y, Px - Py) \leqq \|x - y\| \cdot \|Px - Py\|.$$

If $\|Px - Py\| \neq 0$, this gives $\|Px - Py\| \leqq \|x - y\|$, and if $\|Px - Py\| = 0$ the proposition is trivially true. QED.

Corollary. *P is continuous.*

Proposition 5. *The restriction of P to $B(K)$, which we shall denote P_K is an affine operator, that is*

$$P_K \Sigma_i \varkappa_i x_i = \Sigma_i \varkappa_i P_K x_i \quad \text{if} \quad \Sigma_i \varkappa_i = 1.$$

Proof. Since for every $z \in B(K)$, the point $Pz = x_z$ "closest" to z is in $C(K)$, which is an open subset of $L(K)$, it coincides with the point in $L(K)$ "closest" to z. Thus, for the points in $B(K)$, P is the orthogonal projection operator on the affine subvariety $L(K)$. This operator, whose properties are well known, is among other things affine. QED.

Proposition 6. *The sets $B(K)$ are convex.*

Proof. $B(K)$ is the inverse image under P_K of $C(K)$, which is convex. The inverse image of a convex set under an affine transformation is itself convex.

Proposition 7. *The sets $B(K)$ with nondegenerate K cover the space.*

Proof. Let z be an arbitrary point. We have to prove that there exists a *nondegenerate* index set K (depending on z) such that

$$Pz = x_z = \Sigma_m \mu_m a_m, \quad \mu_m > 0, \quad \Sigma_m \mu_m = 1, \quad m \in K.$$

Since every z has a projection on $\operatorname{conv}(A)$, and this projection must be in some simplex generated by points of A, the existence of an index set K (degenerate or not) is selfevident. We shall show that every degenerate K, such that $z \in B(K)$ contains an index q such that $z \in B(K - \{q\})$.

If K is degenerate, there exist ν_m, not all zero, such that $\Sigma_m \nu_m = 0$ and $\Sigma_m \nu_m a_m = 0$, $m \in K$.

Define

$$\varkappa = \min_{\nu_m < 0} - \mu_m / \nu_m ,$$

and put

$$\varkappa_m = \mu_m + \varkappa \nu_m .$$

Then $\varkappa_m \geqq 0$ and $\varkappa_q = 0$ for at least one index $q \in K$. Further, $\Sigma_m \varkappa_m = 1$ and $\Sigma_m \varkappa_m a_m = \Sigma_m \mu_m a_m + \varkappa \Sigma_m \nu_m a_m = x_z \in \operatorname{conv}(a_m : m \in K - \{q\})$. QED.

We shall study the solutions of (I_λ) for $0 \leqq \lambda \leqq 1$ by considering the projections of

$$y(\lambda) = (1-\lambda)c + \lambda d = c + \lambda(d-c) = c + \lambda b \qquad (0 \leqq \lambda \leqq 1)$$

as a function of λ. We have introduced the notation $b = d - c$.

Proposition 8. *When $y(\lambda) \in B(K)$ with K nondegenerate, we have a unique representation*

$$P y(\lambda) = \Sigma_m (\alpha_m + \beta_m \lambda) a_m , \quad \Sigma_m \alpha_m = 1, \quad \Sigma_m \beta_m = 0, \quad m \in K.$$

Proof. $P y(\lambda) = P_K y(\lambda) = P_K[(1-\lambda)c + \lambda d] = (1-\lambda) P_K c + \lambda P_K d = P_K c + \lambda (P_K d - P_K c)$. We can write

$$P_K c = \Sigma_m \alpha_m a_m , \quad \Sigma_m \alpha_m = 1, \quad m \in K$$

and

$$P_K d = \Sigma_m \gamma_m a_m, \quad \Sigma_m \gamma_m = 1, \quad m \in K .$$

Putting $\beta_m = \gamma_m - a_m$ so that $\Sigma_m \beta_m = 0$, we get the representation. Its uniqueness follows from prop. 2.

Corollary. *By prop. 4, corollary, the representation*

$$P y(\lambda) = \Sigma_m (\alpha_m + \lambda \beta_m) a_m$$

is valid also on the boundary of $B(K)$. (Points on the boundary may or may not belong to $B(K)$.)

Theorem. *To every starting point c there exists a finite sequence of parameter values*

$$0 = \lambda_0 \leqq \lambda_1 \leqq \lambda_2 \cdots \leqq \lambda_t = 1$$

and a corresponding sequence of nondegenerate index sets K_i such that

$y(\lambda) \in B(K_i)$ for $\lambda_i < \lambda < \lambda_{i+1}$ if $\lambda_i < \lambda_{i+1}$

and such that $y(\lambda_i)$ is in at least one of $B(K_i)$ and $B(K_{i+1})$. We shall refer to such a sequence of pairs as a (λ, K)-sequence.

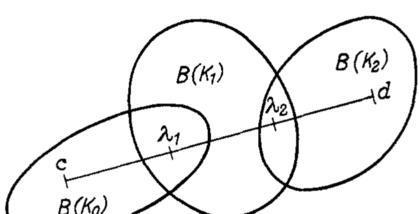

Fig. 1

Proof (Cf. Fig. 1). By proposition 7 we can take a covering of the space by $B(K)$ with nondegenerate K. Since the sets $B(K)$ are convex, those who intersect the line-segment between c and d, cut out intervals of it. Since there are only a finite number of sets $B(K)$, these intervals are finite in number and they cover the line-segment. Thus, we can choose points λ_i so that the theorem is true. QED.

4. The Geometry of an Algorithm for Problem I

Construction of a (λ, K)-sequence

We shall describe an algorithm for the construction of a (λ, K)-sequence. A step of this algorithm is to determine λ_{i+1} and K_{i+1} when λ_i and K_i are known.

Since we always choose c in or on the boundary of $\operatorname{conv}(A)$ in our method, we confine ourselves to describe an algorithm that works with such a c. In the discussion, cf. Fig. 2. First, we describe the procedure for such λ that $y(\lambda) \notin \operatorname{conv}(A)$, so that $y(\lambda) \neq Py(\lambda)$. Suppose that $y(\lambda') \in B(K_i)$, where $\lambda' \geqq \lambda_i$. We shall start

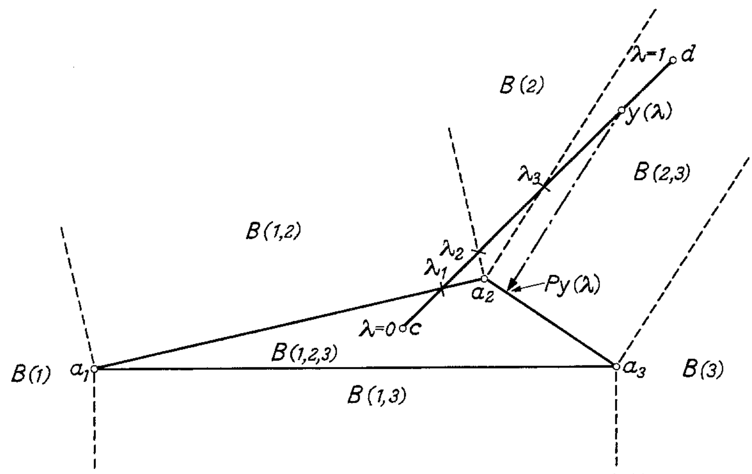

Fig. 2

by determining λ_{i+1}. If we increase λ from λ', $y(\lambda)$ moves on a straight line and by proposition 5, $Py(\lambda)$ moves on a straight line in $L(K_i)$. Consider the hyperplane $H(\lambda)$: $(y(\lambda) - Py(\lambda), x - Py(\lambda)) = 0$. If $L(K_i)$ has dimension $s-1$ (is a hyperplane), $H(\lambda)$ equals $L(K_i)$, but if the dimension of $L(K_i)$ is less than $s-1$, $H(\lambda)$ "rotates around" $L(K_i)$ when λ is changed. By proposition 3, $y(\lambda)$ remains in $B(K_i)$ as long as $Py(\lambda) \in C(K_i)$ (condition 1) and $H(\lambda)$ separates all the points $a_n \in A$ from $y(\lambda)$ (condition 2). Since $Py(\lambda) \in C(K_i)$, we can write (cf. prop. 8)

$$Py(\lambda) = \Sigma_m \mu_m(\lambda) a_m, \quad \mu_m(\lambda) \geqq 0, \quad \Sigma_m \mu_m(\lambda) = 1, \quad m \in K_i.$$

Thus, $y(\lambda)$ reaches the boundary of $B(K_i)$ either when $\mu_m(\lambda)$ becomes zero for an index $m \in K_i$ (condition 1) or when $H(\lambda)$ is turned so much that it touches a point $a_n \notin K_i$, that is

$$h_n(\lambda) = (y(\lambda) - Py(\lambda), a_n - Py(\lambda)) \leqq 0$$

becomes zero for an index $n \notin K_i$ (condition 2).

We define

$$\xi_{i+1} = \max_{\mu_m(\lambda) \geqq 0} \lambda$$

$$\eta_{i+1} = \max_{h_n(\lambda) \leqq 0} \lambda.$$

If we put

$$\lambda_{i+1} = \min(\xi_{i+1}, \eta_{i+1})$$

and if this $\lambda_{i+1} > \lambda_i$, we have $y(\lambda) \in B(K_i)$ for $\lambda_i < \lambda < \lambda_{i+1}$. Now, we turn to the determination of K_{i+1}.

If $\xi_{i+1} < \eta_{i+1}$ and there is only one index $m \in K_i$ such that $\mu_m(\lambda_{i+1}) = 0$, we have by definition that $y(\lambda_{i+1}) \in B(K_i - \{m\})$, so we put

$$K_{i+1} = K_i - \{m\}.$$

Since K_i is nondegenerate, its subset K_{i+1} is too.

If on the other hand, $\eta_{i+1} < \xi_{i+1}$, and there is only one index $n \notin K_i$ such that $h_n(\lambda_{i+1}) = 0$ we put

$$K_{i+1} = K_i + \{n\}.$$

The motivation for this is that $y(\lambda_{i+1})$ is on the boundary of $B(K_{i+1})$ and $y(\lambda)$ is moving into this set since it comes from the adjacent set $B(K_i)$. Since, for $\lambda_i < \lambda < \lambda_{i+1}$, $y - Py$ is orthogonal to $L(K_i)$ and $Py \in L(K_i)$, $h_n(\lambda) \equiv 0$ if $a_n \in L(K_i)$. If a_n determines η_{i+1}, $h_n(\lambda)$ must vary with λ, that is $a_n \notin L(K_i)$, so K_{i+1} is nondegenerate.

If $\xi_{i+1} = \eta_{i+1}$ or if there are more than one index determining ξ_{i+1} or η_{i+1}, we call $y(\lambda_{i+1})$ a degenerate point. In section 10 we describe how to choose K_{i+1} at such a point.

It may happen that $\lambda_{i+1} = \lambda_i$, namely when $L(K_{i-1})$ contains a point a_q, $q \notin K_{i-1}$ and $K_i = K_{i-1} - \{m\}$. The hyperplane $H(\lambda)$ which turned around $L(K_{i-1})$ for $\lambda \leqq \lambda_i$, then should turn around the smaller variety $L(K_i)$ for $\lambda \leqq \lambda_i$, but it may be locked by the point $a_q \in H(\lambda_i)$, so that $\eta_{i+1} = \lambda_i$. By the rules described above, we then put $K_{i+1} = K_i + \{q\}$ and obtain $\lambda_{i+2} > \lambda_{i+1} = \lambda_i$. (Some reasoning using the continuity of $\mu_m(\lambda)$ and $h_n(\lambda)$ shows that we always have $\xi_{i+1} > \lambda_i$, and that $\eta_{i+1} > \lambda_i$ in all situations except the one just described.)

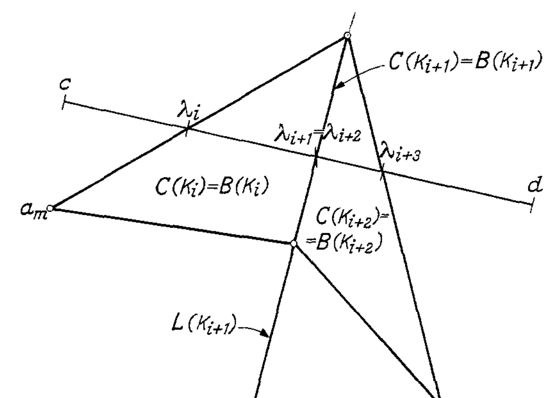

Fig. 3

Now, we shall describe the geometry when $y(\lambda) \in \operatorname{conv}(A)$. Then $Py(\lambda) = y(\lambda)$ and $B(K_i) = C(K_i)$. If K_i has $s+1$ elements, $L(K_i)$ is the whole space, and $h_n(\lambda) = 0$ for all n. Then $y(\lambda)$ reaches the boundary of $B(K_i)$ when one $\mu_m(\lambda)$ becomes zero. We put $K_{i+1} = K_i - \{m\}$ since $y(\lambda_{i+1}) \in C(K_{i+1})$. $L(K_{i+1})$ is now a hyperplane bounding $C(K_i)$ at $y(\lambda_{i+1})$. The vector $y(\lambda) - Py(\lambda)$ for $\lambda > \lambda_{i+1}$ is the perpendicular to $L(K_{i+1})$ pointing out of $C(K_i)$ (cf. Fig. 3). The quantities $h_n(\lambda)$ will increase from zero at λ_{i+1} to positive values when λ is increased for all a_n strictly on the $y(\lambda)$ side of $L(K_{i+1})$, so that all these a_n will act to make $\eta_{i+2} = \lambda_{i+1}$. By taking any of these n, and form $K_{i+2} = K_{i+1} + \{n\}$, we get a nondegenerate simplex $C(K_{i+2}) = B(K_{i+2})$ into which $y(\lambda) = Py(\lambda)$ moves for $\lambda \geqq \lambda_{i+2} = \lambda_{i+1}$, that is $\lambda_{i+3} > \lambda_{i+2} = \lambda_{i+1}$. We shall use the rule to choose that n for which a_n is most far away from $L(K_{i+1})$. If there is no a_n strictly on the $y(\lambda)$ side of $L(K_{i+1})$, this hyperplane is a supporting hyperplane of $\operatorname{conv}(A)$, and we have the case $y(\lambda) \neq Py(\lambda)$ described first. Thus, when $y(\lambda) \in \operatorname{conv}(A)$, we use the same formulas and rules to determine λ_{i+1} and K_{i+1} as when $y(\lambda) \notin \operatorname{conv}(A)$, but we need an extra rule to determine which n to choose every second step.

5. The Algebra of the Algorithm for Problem (I)

By prop. 8, the barycentric coordinates of $Py(\lambda)$ have the following form when $y(\lambda) \in B(K_i)$:

$$\mu_m(\lambda) = \alpha_m + \lambda \beta_m \qquad (m \in K_i)$$

with

$$\Sigma_m \alpha_m = 1 \qquad (m \in K_i) \tag{1}$$

and

$$\Sigma_m \beta_m = 0 \qquad (m \in K_i). \tag{2}$$

We begin with the determination of β_m. We use the fact that $c - P_{K_i}c$ and $d - P_{K_i}d$ are orthogonal to $L(K_i)$, that is

$$(c - P_{K_i}c,\, a_r - a_s) = 0 \qquad (r, s \in K_i)$$

$$(d - P_{K_i}d,\, a_r - a_s) = 0 \qquad (r, s \in K_i).$$

With the notations of prop. 8, this gives

$$(b - \Sigma_m \beta_m a_m,\, a_r - a_s) = 0 \qquad (m, r, s \in K_i)$$

or

$$(b - \Sigma_m \beta_m a_m,\, a_r) = \text{const.} = \beta_0 \qquad (m, r \in K_i). \tag{3}$$

The relations (2) and (3) give a system of linear equations, which is solvable since K_i is nondegenerate.

A similar discussion gives

$$(c - \Sigma_m \alpha_m a_m,\, a_r) = \alpha_0 \qquad (m, r \in K_i). \tag{4}$$

The relations (4) together with (1) give α_0 and α_m $(m \in K_i)$. However, the following discussion leads to an easier way to find α_0 and α_m.

By putting $\alpha_n = \beta_n = 0$ for $n \notin K_i$, we can write

$$Py(\lambda) = \Sigma_m (\alpha_n + \lambda \beta_n) a_n \qquad (1 \leqq n \leqq N).$$

Let α_n and β_n $(1 \leqq n \leqq N)$ be the coefficients corresponding to $B(K_i)$, and let α_n' and β_n' be those corresponding to $B(K_{i-1})$. By the corollary of prop. 8, we have

$$Py(\lambda_i) = \Sigma_n (\alpha_n + \lambda_i \beta_n) a_n = \Sigma_n (\alpha_n' + \lambda_i \beta_n') a_n \qquad (1 \leqq n \leqq N).$$

Since one of the varieties $L(K_i)$ and $L(K_{i+1})$ contains the other, we have by prop. 2

$$\alpha_n + \lambda_i \beta_n = \alpha_n' + \lambda_i \beta_n' \qquad (1 \leqq n \leqq N). \tag{5}$$

Using (3) and (4) we can write

$$\mu_0(\lambda) = \alpha_0 + \lambda \beta_0 = (y(\lambda) - Py(\lambda), a_m) \quad \text{for all } m \in K_i. \tag{6}$$

Since $y(\lambda)$ and $Py(\lambda)$ are continuous functions of λ, we obtain

$$\mu_0(\lambda_i) = \alpha_0 + \lambda_i \beta_0 = \alpha_0' + \lambda_i \beta_0'. \tag{7}$$

Thus, we can use

$$\alpha_n = \alpha_n' - \lambda_i(\beta_n - \beta_n') \qquad (0 \leqq n \leqq N) \tag{8}$$

to calculate α_n when β_n has been computed from (2) and (3).

The condition $\mu_m(\lambda) \geqq 0$ gives at once

$$\xi_{i+1} = \min_{\beta_m < 0} - \alpha_m/\beta_m \qquad (m \in K_i). \tag{9}$$

To get η_{i+1}, we insert the explicit expressions in

$$h_n(\lambda) = (y(\lambda) - Py(\lambda), a_n - Py(\lambda)).$$

It would seem that $h_n(\lambda)$ is a quadratic function of λ, but using (6) we get

$$\begin{aligned}(y(\lambda) - Py(\lambda), Py(\lambda)) &= (y(\lambda) - Py(\lambda), \Sigma_m(\alpha_m + \lambda\beta_m)a_m) \\ &= \Sigma_m(\alpha_m + \lambda\beta_m)(y(\lambda) - Py(\lambda), a_m) = \alpha_0 + \lambda\beta_0.\end{aligned}$$

Thus,

$$\begin{aligned}h_n(\lambda) &= (y(\lambda) - Py(\lambda), a_n) - \alpha_0 - \lambda\beta_0 \\ &= (c + \lambda b - \Sigma_m(\alpha_m + \lambda\beta_m)a_m, a_n) - \alpha_0 - \lambda\beta_0 \\ &= [(c - \Sigma_m\alpha_m a_m, a_n) - \alpha_0] + \lambda[(b - \Sigma_m\beta_m a_m, a_n) - \beta_0] = u_n + \lambda v_n,\end{aligned}$$

where we have defined u_n and v_n.

The condition $h_n(\lambda) \leqq 0$ gives

$$\eta_{i+1} = \min_{v_n > 0} - u_n/v_n \qquad (n \notin K_i). \tag{10}$$

Also here it is obvious that $h_n(\lambda)$ are continuous functions of λ, so that we have (with the usual notation)

$$u_n + \lambda_i v_n = u_n' + \lambda_i v_n',$$

from which equations we get u_n when

$$v_n = (b, a_n) - \beta_0 - \Sigma_m\beta_m(a_m, a_n) \qquad (m \in K_i, n \notin K_i)$$

are computed.

6. The Starting Point

To be able to start our algorithm, we need a nondegenerate index set K_0 and a point $c \in B(K_0)$, such that $h_n(\lambda) \leqq 0$ for $n \notin K_0$ in an open interval $(0, \lambda_1)$.

If $N > s$, this can be accomplished by choosing for K_0 any nondegenerate index set consisting of $s+1$ indices and as c any point

$$c = \Sigma_m \mu_m a_m, \mu_m > 0, \Sigma_m \mu_m = 1, \qquad m \in K_0. \tag{11}$$

With this choice of c we have $Py = y$ in $C(K_0)$, so that $h_n(\lambda) \leqq 0$ in an open λ-interval.

If $N \leqq s$, we take as K_0 a maximal nondegenerate index set formed from $1, 2, 3, \ldots, N$, and choose c by (11).

If we have any idea of what the solution of (I) is like, of course we choose c close to the expected Pd. Without any information on a likely form of the solution, we suggest the starting algorithm discribed below to be used. This algorithm works with almost the same computational scheme as the main algorithm.

Try successively for K_0 an increasing sequence of index sets:

$$Q_1 = \{q_1\}, \quad Q_2 = \{q_1, q_2\}, \quad Q_3 = \{q_1, q_2, q_3\}, \ldots, \quad Q_i = \{q_1, q_2, \ldots, q_i\}, \ldots$$

and a corresponding sequence of starting points

$$c_i = (1/i)\, \Sigma_m\, a_{q_m} \qquad (1 \leqq m \leqq i).$$

Start the algorithm with any index as q_1, for instance one such that $\| d - a_q \|$ is minimum.

When q_i is chosen, start step i by calculating β_{q_m} $(1 \leqq m \leqq i)$ and β_0 from (2) and (3) with $b = d - c_i$. Since $\alpha_{q_m} = 1/i > 0$, we have $\xi_1 > 0$, so Q_i is admissible as K_0 if $\eta_1 > 0$. Since $c_i = Pc_i$, we have $h_n(0) = 0$ for all n, that is $u_n = 0$ for all n, and we have $h_n(\lambda) = \lambda v_n$. The condition $h_n(\lambda) \leqq 0$ then reduces to $v_n \leqq 0$, so we calculate v_n for all $n \notin Q_i$ and if any of these is strictly positive, we take as q_{i+1} that n which corresponds to the largest v_n.

We can find a way to simplify this algorithm a little by noting that the only property of c_i that was used to get $h_n(\lambda) = \lambda v_n$, was $c_i = Pc_i$. Thus, since $a_{q_1} \in L(Q_i)$, we get the same sequence of q_i if we use $c_i = c_1 = a_{q_1}$. When K_0 is found and has r elements, we put $c = (1/r)\, \Sigma_m a_m$, $m \in K_0$.

7. The Cone Problem

In this section, we shall show that our second problem

$$\inf \{\| x - d \| : x \in \operatorname{cone}(A)\}, \tag{II}$$

can be treated in almost the same way as problem (I). For its solution we shall choose a starting point c and consider the family of problems

$$\inf \{\| x - c - \lambda (d - c) \| : x \in \operatorname{cone}(A)\}. \tag{II$_\lambda$}$$

The necessary transcriptions of the formulas and definitions for problem (I) to make them valid for problem (II) will not be given in detail. We just survey

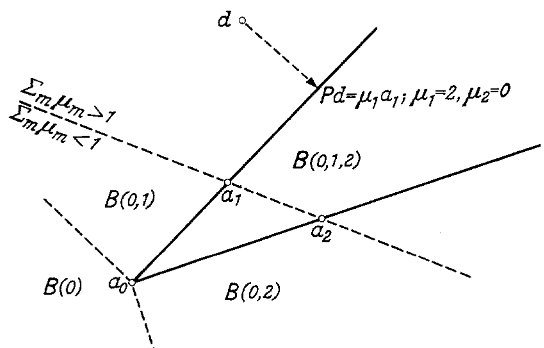

Fig. 4

the main changes. Thus, we get the cone problem from problem (I) by adding the origin as a point a_0 to A and exclude the condition $\Sigma_m \mu_m = 1$ on the representation of the projection (we keep the conditions $\mu_m > 0$). All index sets K shall contain the index 0, but this is not visible in the formulas because we never have to sum over the term $\mu_0 a_0$ since it does not contribute to the sums. (Cf. Fig. 4.)

As to the geometry of the problem, all affine varieties $L(K)$ become linear subspaces, and degeneracy becomes linear dependence (in the norm case).

The algebra of problem (II) is obtained from that of problem (I) by excluding the conditions $\Sigma_m a_m = 1$ and $\Sigma_m \beta_m = 0$, and putting $\alpha_0 = \beta_0 = 0$.

8. Proofs for Problem (I) when $\| \ \|$ is a Seminorm

When $\| \ \|$ is a seminorm, there may be more than one point $x_d \in \operatorname{conv}(A)$ satisfying $\|x_d - d\| \leqq \|x - d\|$ for all $x \in \operatorname{conv}(A)$. To treat the seminorm case, one can map the whole problem on the quotientspace R^s/S, where S is the linear subspace

$$S = \{z\colon \|z\| = 0\}.$$

The map of the seminorm becomes a real norm there, and theoretically there is nothing more to prove. However, our algorithm works well when $\| \ \|$ is a seminorm, so we need not do the computationally cumbersome mapping on the quotientspace. Instead, we apply the inverse of the map on the quotientspace to our proofs so that we get proofs for the seminorm case.

Thus, define an index set K to be degenerate (mod S) if it is possible to find $\varkappa_m$ not all zero such that $\Sigma_m \varkappa_m = 0$ and $\|\Sigma_m \varkappa_m a_m\| = 0$. An index set is defined to be nondegenerate (mod S) if it is not degenerate (mod S).

Using the definitions of $L(K)$ and $C(K)$ of section 3, we get:

Proposition 9. *If* $x_1, x_2 \in L(K)$ *with* K *nondegenerate* (mod S) *and* $\|x_1 - x_2\| = 0$, *then* $x_1 = x_2$.

Proof. Let

$$x_1 = \Sigma_m \varkappa_m a_m, \quad \Sigma_m \varkappa_m = 1, \quad m \in K$$

and

$$x_2 = \Sigma_m \nu_m a_m, \quad \Sigma_m \nu_m = 1, \quad m \in K.$$

Writing $\mu_m = \varkappa_m - \nu_m$, we get $\Sigma \mu_m = 0$ and $0 = \|x_1 - x_2\| = \|\Sigma_m (\varkappa_m - \nu_m) a_m\| = \|\Sigma_m \mu_m a_m\|$. Since K is nondegenerate (mod S), this means $\mu_m = 0$. for all $m \in K$. QED.

Let X_d be the set of solution points of (I) and define $\bar{P}$ by $\bar{P}: d \to X_d$.

Proposition 10. *Let* $x_1 \in X_d$. *A point* $x_2 \in \operatorname{conv}(A)$ *is in* X_d *if and only if* $\|x_1 - x_2\| = 0$.

Proof. If $\|x_1 - x_2\| = 0$, we get $\|x_2 - d\| \leqq \|x_2 - x_1\| + \|x_1 - d\| = \|x_1 - d\|$, giving $x_2 \in X_d$. Conversely, if $x_2 \in X_d$, we have by 2. of prop. 3 that

$$(d - x_1, x_2 - x_1) \leqq 0.$$

and

$$(d - x_2, x_1 - x_2) \leqq 0.$$

Changing the signs in the second inequality and adding it to the first, we obtain $\|x_2 - x_1\|^2 \leqq 0$. QED.

Corollary 1. *If* x_d *is a solution, we have* $X_d = (x_d + S) \cap \operatorname{conv}(A)$.

Corollary 2. *For an arbitrary point z, the value of* $\| x_d - z \|$ *is the same for all* $x_d \in X_d$.

In view of the last corollary, we can define

$$\bar{B}(K) = \{z : \| \bar{P}z - \Sigma_m \varkappa_m a_m \| = 0, \varkappa_m > 0, \Sigma_m \varkappa_m = 1, m \in K\}.$$

$\bar{B}(K)$ is the set of all points whose projection on conv(A) intersects $C(K)$.

Examining the proofs of prop. 1 and 2, we easily see that these props. are true when degeneracy is replaced by degeneracy (mod S). In the props. 3 and 4, degeneracy is not mentioned.

Much of the explanation of why our method works well when $\| \ \|$ is a seminorm is contained in

Proposition 11. *For* $z \in \bar{B}(K)$, $\bar{P}z$ *intersects* $C(K)$ *in a single (unique) point when K is nondegenerate* (mod S).

Proof. Follows immediately from props. 9 and 10. QED.

We shall soon prove that our algorithm works with sets K which are non-degenerate (mod S). From prop. 11, it then follows that we work all the time with unique projections in the algorithm, so that the discussions of section 4 also can be done when we have a seminorm.

Yet we have to prove the existence of a finite algorithm when $\| \ \|$ is a seminorm, that is we have to prove the generalizations of prop. 5, 6 and 7.

For K nondegenerate (mod S), we define $\bar{P}_K$ with domain $B(K)$ as the operator which takes a point to its unique projection on $C(K)$.

Thus, when K is nondegenerate (mod S), the definition of $\bar{B}(K)$ can be written

$$\bar{B}(K) = \{z : \bar{P}_K z = \Sigma_m \varkappa_m a_m, \varkappa_m > 0, \Sigma_m \varkappa_m = 1, m \in K\}.$$

Proposition 12. $\bar{P}_K$ *is an affine operator.*

Proof. The same as for prop. 5.

Proposition 13. $\bar{B}(K)$ *with K nondegenerate* (mod S) *is convex.*

Proof. Follows immediately from prop. 12.

Note: In fact, every $\bar{B}(K)$ is convex. They are actually the inverse images for the map of the convex sets $B(K)$ in R^s/S.

Proposition 14. *The sets* $\bar{B}(K)$ *with K nondegenerate* (mod S) *cover the space.*

Proof. We follow the line of proof of prop. 7. Thus let K be an index set (degenerate (mod S) or not) such that

$$Pz \ni x_z = \Sigma_m \mu_m a_m, \mu_m > 0, \Sigma_m \mu_m = 1, m \in K.$$

If K is degenerate (mod S), there exist ν_m, not all zero, such that $\Sigma_m \nu_m = 0$ and $\|\Sigma_m \nu_m a_m\| = 0, m \in K$. Define $\varkappa$ and $\varkappa_m$ and $\varkappa_q$ as in prop. 7. Put $x_z' = \Sigma_m \varkappa_m a_m$. Then, $\| x_z - x_z' \| = \| \varkappa \Sigma_m \nu_m a_m \| = 0$, so x_z' is a solution by prop. 10. QED.

To show that our algorithm described in section 4 works we have to prove that it chooses index sets that are nondegenerate (mod S) and that the system of equations (2) and (3) are solvable. Nothing else can cause trouble.

Thus, assume that K_i is nondegenerate (mod S), that n has been chosen with the aid of formula (10) which implies that $v_n > 0$. We shall prove that $K_{i+1} = K_i + \{n\}$ is nondegenerate (mod S). Assume against the hypothesis that it is degenerate (mod S). Then, there exist ν_r, not all zero, $\Sigma_r \nu_r = 1$, $r \in K_i$ such that

$\| a_n - \Sigma_r v_r a_r \| = 0$ $(r \in K_i)$. For any point x, the Cauchy-Schwarz inequality gives

$$|(x, a_n - \Sigma_r v_r a_r)| \leqq \|x\| \cdot \| a_n - \Sigma_r v_r a_r \| = 0 .$$

From $v_n = (b - \Sigma_m \beta_m a_m, a_n) - \beta_0 > 0$ we get

$$\beta_0 < (b - \Sigma_m \beta_m a_m, a_n) = \Sigma_r v_r (b - \Sigma_m \beta_m a_m, a_r) = \Sigma_r v_r \beta_0 = \beta_0 ,$$

Since this is impossible, K_{i+1} is nondegenerate (mod S).

Further, suppose that there are two solutions β_{01}, β_{m1} and β_{02}, β_{m2} of (2) and (3). From (3) we get

$$(b, a_r) = \beta_{01} + \Sigma_m \beta_{m1} (a_m, a_r) = \beta_{02} + \Sigma_m \beta_{m2} (a_m, a_r) \qquad (m, r \in K),$$

or

$$\beta_{01} - \beta_{02} = \Sigma (\beta_{m2} - \beta_{m1}) (a_m, a_r) \qquad (m, r \in K). \tag{12}$$

Introducing $\varkappa_m = \beta_{m2} - \beta_{m1}$ we get $\Sigma_m \varkappa_m = 0$. Multiplying by $\varkappa_r$ in (12) and adding we get

$$0 = \Sigma_r \Sigma_m \varkappa_r \varkappa_m (a_r, a_m) = \| \Sigma_m \varkappa_m a_m \|^2$$

contradicting the hypothesis that K is nondegenerate (mod S).

9. Computational Aspects and Experiences

The main numerical work in a step of our algorithm is needed for the solution of the system of equations (2) and (3). Introducing the symmetric matrix M_i and the vectors $\bar{\beta}_i$ and $\bar{g}_i$

$$M_i = \begin{pmatrix} 0 & 1 \; 1 \; 1 \; . \; . \; 1 \\ 1 & \\ 1 & \\ 1 & (a_m, a_n) \\ . & \\ . & \\ 1 & \end{pmatrix} \qquad \bar{\beta}_i = \begin{pmatrix} \beta_0 \\ \cdot \\ \cdot \\ \beta_m \\ \cdot \\ \cdot \\ \cdot \end{pmatrix} \qquad \bar{g}_i = \begin{pmatrix} 0 \\ \cdot \\ \cdot \\ (a_m, b) \\ \cdot \\ \cdot \\ \cdot \end{pmatrix} \qquad m, n \in K_i ,$$

we can write the system (2) and (3) in the form

$$M_i \bar{\beta}_i = \bar{g}_i .$$

For the determination of $\bar{\beta}_i$, we suggest the calculation of the inverse of M_i. We do this because M_i^{-1} is easily calculated when M_{i-1}^{-1} is known. When $K_i = K_{i-1} + \{q\}$, this can be done by the bordering method (see e.g. [6]). When $K_i = K_{i-1} - \{q\}$, a method analogous to the bodering method gives M_i^{-1} with very little computation. Straightforward matrix manipulations also show that $\bar{\beta}_i$ can be obtained from $\bar{\beta}_{i-1}, \bar{g}_i$ and only the last column of M_i^{-1} or the deleted column of M_{i-1}^{-1}.

The algorithm has been tested on 150 problems generated by random numbers. The largest problems had $s = 20$ and $N = 40$. As a condensed description of the tests, we give the average computing time $\bar{T}$ as a function of N and s:

$\bar{T} \sim C_1 (Ns)^{0.67}$ when d was far away from conv(A),

$\bar{T} \sim C_1 (Ns)^{0.95}$ when d was close to or inside conv(A).

($C_1 \sim 0.2$ sec on a Ferranti Mercury computor.)

10. Degeneracy Procedures

We shall give two ways of dealing with a degeneracy encountered at $y(\lambda_{i+1})$. Our first method is to displace the starting point c. This does not mean that we have to restart the calculations from the beginning. First, we describe a way of displacing c, and then we prove that nothing much is lost in doing so.

We denote quantities pertaining to the displaced starting point by the subscript $_1$. Let

$$c_1(\varepsilon) = c + \varepsilon \cdot w$$

be a preliminary new starting point, where w is a vector not parallel with b and $\varepsilon > 0$ a number to be determined so that $y_1(\bar{\lambda}) \in B(K_i)$, where $\bar{\lambda} = \frac{1}{2}(\lambda_{i+1} + \lambda_i)$. Take for instance w as a unit vector $(w = (0, 0, \ldots, 1, \ldots, 0, 0))$ and calculate

$$\alpha_{m1} = \alpha_m + \varepsilon \alpha_m' \qquad m \in K_i$$
$$\beta_{m1} = \beta_m + \varepsilon \beta_m' \qquad m \in K_i .$$

The quantities α_m' and β_m' are easily found when we have the inverse of the matrix M corresponding to K_i. A short calculation shows that

$$\begin{bmatrix} \alpha_0' \\ \cdots \\ \cdot \\ \alpha_m' \\ \cdot \\ \cdot \end{bmatrix} = - \begin{bmatrix} \beta_0' \\ \cdots \\ \cdot \\ \beta_m' \\ \cdot \\ \cdot \end{bmatrix} = M^{-1} \begin{bmatrix} 0 \\ \cdots\cdots \\ \cdot \\ (a_m, w) \\ \cdot \\ \cdot \end{bmatrix}$$

For $y_1(\bar{\lambda})$ to be in $B(K_i)$, the conditions 1. and 2. of prop. 3 must be satisfied, giving the following conditions on ε:

$$\mu_{1m}(\bar{\lambda}) = \alpha_m + \bar{\lambda}\beta_m + \varepsilon(1 - \bar{\lambda})\alpha_{1m}' \geqq 0 \quad \text{for} \quad m \in K_i$$
$$h_{1n}(\bar{\lambda}) = u_n + \bar{\lambda} v_n + \varepsilon(1 - \bar{\lambda})[(a_n, w) - \alpha_0' - \Sigma_m \alpha_m'(a_m, a_n)] \leqq 0$$
$$\text{for} \quad n \notin K_i .$$

If $\bar{\varepsilon}$ is the largest value of ε satisfying these conditions, choose as a new starting point $c_1(\varepsilon_1)$ where $\varepsilon_1 < \bar{\varepsilon}$, so that $y_1(\bar{\lambda})$ is strictly inside $B(K_i)$. Start the algorithm again by determining $\lambda_{1(i+1)}$.

Since we have changed c, our proof of the finiteness of the algorithm breaks down. This can be remedied, however, if we can prove that no $B(K_j)$ with $j < i$ intersects $y_1(\lambda)$ for $\lambda \geqq \lambda_{1(i+1)}$. This is a two-dimensional problem in the plane through c, c_1 and d. The intersection between $B(K_j)$ and this plane is a convex set. Now, consider the interval that $B(K_j)$ cuts out of the halfline through $c_1(\varepsilon)$ with d as endpoint. Let $f_j(\varepsilon)$ be the length of this interval. Since $B(K_j)$ is a connected set, $f_j(\varepsilon)$ is strictly positive in *one* interval (connected set). Since $y(\lambda)$ intersects $B(K_j)$ for some $\lambda < \lambda_{i+1}$, we have $f_j(0) > 0$. If $f_j(\varepsilon)$ becomes zero for some $\varepsilon < \varepsilon_1$, $f_j(\varepsilon_1) = 0$, that is $B(K_j)$ does not intersect $y_1(\lambda)$ for $\lambda \geqq \lambda_{1(i+1)}$. Cf. Fig. 5. If c has to be displaced several times, the proof of the finiteness holds provided the same w is chosen every time.

Our second way of dealing with a degeneracy consists in solving an auxiliary subproblem of kind (II).

Write $y(\lambda_{i+1}) = y$ for the degeneracy point and $Py(\lambda_{i+1}) = f$ for its projection. Let M be the set of indices m for which $(y - f, a_m - f) = 0$. We have $K_i \subset M$. Consider the problem

$$\inf\{\|z - b\| : z \in \operatorname{cone}(a_m - f, m \in M)\}. \qquad \text{(III)}$$

Its solution has the form

$$z_0 = \Sigma_m \varkappa_m (a_m - f), \quad \varkappa_m \geqq 0, \quad m \in M.$$

We claim that the main problem can be continued across λ_{i+1} by putting

$$K_{i+1} = \{m : m \in M, \varkappa_m > 0\}$$

and moreover that

$$P_{K_{i+1}}(\lambda) = f + (\lambda - \lambda_{i+1}) z_0 .$$

Note: If $y = f$, which is the case when $y \in \operatorname{conv}(A)$ resp. $\operatorname{cone}(A)$, M contains all indices. Then, problem (III) is equally hard to solve as any of the problems (I) and (II), so in this case the first method of this section shall be used.

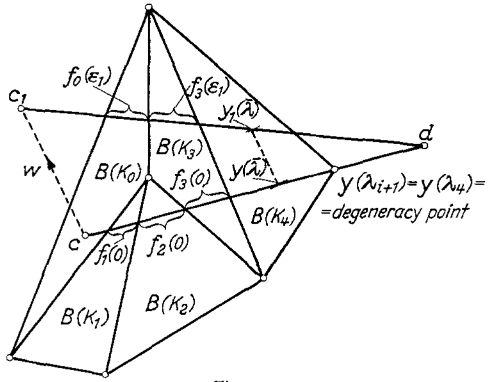

Fig. 5

First, we note that if $f = \Sigma_m \mu_m a_m$, $m \in K_i$, we have

$$f_{i+1}(\lambda) = P_{K_{i+1}}(\lambda) = \text{const.} + \lambda\, \Sigma_m [\varkappa_m - (\Sigma_n \varkappa_n)\, \mu_m]\, a_m, \quad m, n \in M.$$

Thus, if $\mu_m = 0$, the new $\beta_m = \varkappa_m - (\Sigma_n \varkappa_n)\, \mu_m \geqq 0$, so that $\xi_{i+2} > \lambda_{i+1}$. Further, we prove the existence of a $\eta_{i+2} > \lambda_{i+1}$ such that $h_n(\lambda) \leqq 0$ for all n in the interval $\lambda_{i+1} \leqq \lambda \leqq \eta_{i+2}$. Since z_0 is the solution of (III), we have

$$(b - z_0, z_0) = 0 \qquad (13)$$

$$(b - z_0, z_0 - a_m + f) \geqq 0 \quad m \in M,$$

or in view of (13)

$$(b - z_0, f - a_m) \geqq 0 \qquad m \in M \qquad (14)$$

and

$$(y - f, z_0) = \Sigma_m \varkappa_m (y - f, a_m - f) = 0. \qquad (15)$$

By (13) and (15) we get

$$\begin{aligned} h_n(\lambda) &= (y(\lambda) - f_{i+1}(\lambda), a_n - f_{i+1}(\lambda)) \\ &= (y - f, a_n - f) + (\lambda - \lambda_{i+1})(b - z_0, a_n - f) - (\lambda - \lambda_{i+1})(y - f, z_0) + \\ &\quad + (\lambda - \lambda_{i+1})^2 (b - z_0, z_0) = (y - f, a_n - f) + (\lambda - \lambda_{i+1})(b - z_0, a_n - f). \end{aligned}$$

Since $(y - f, a_n - f) < 0$ for $n \notin M$, there exists a $\eta_{i+2} > \lambda_{i+1}$ such that $h_n(\lambda) \leqq 0$ for $\lambda_{i+1} \leqq \lambda \leqq \eta_{i+2}$ and $n \notin M$. For $n \in M$, $h_n(\lambda) \leqq 0$ by (14) and the definition of M.

References

[1] WOLFE, PH.: Methods of nonlinear programming. Recent advances in mathematical programming. Edited by GRAVES and WOLFE. New York: McGraw-Hill 1963.

[2] HOUTHAKKER, H. S.: The capacity method of quadratic programming. Econometrica **28**, 62—87 (1960).

[3] PHILIP, J.: Reconstruction from measurements of positive quantities by the maximum-likelihood method. J. math. Analysis Appl. **7**, 327—347 (1963).

[4] — A quadratic programming method with error estimates for approximate solutions. Z. Wahrscheinlichkeitstheorie verw. Geb. **1**, 301—314 (1963).

[5] BOURBAKI, N.: Espaces-vectoriels topologiques. Chap. V, § 1, 1955.

[6] FADEEVA, V. N.: Computational methods of linear algebra. Dover 1959.

Institut för Matematisk Statistik
Kungl. Tekniska Högskolan
Stockholm 70 (Schweden)