MATHEMATICS OF COMPUTATION, VOLUME 30, NUMBER 133
JANUARY 1976, PAGES 48–57

A Nearest Point Algorithm for Convex Polyhedral Cones and Applications to Positive Linear Approximation

By Don R. Wilhelmsen

Abstract. Suppose K is a convex polyhedral cone in E_n and is defined in terms of some generating set $\{e_1, e_2, \ldots, e_N\}$. A procedure is devised so that, given any point $q \in E_n$, the nearest point p in K to q can be found as a positive linear sum of $N^* \leqslant n$ points from the generating set. The procedure requires at most finitely many linear steps.

The algorithm is then applied to find a positive representation

$$Lf = \sum_{i=1}^{N^*} \lambda_i f(x_i), \qquad \lambda_i > 0,\ f \in \Phi,$$

for a positive linear functional L acting on a suitable finite-dimensional function space Φ.

1. Introduction. Let K be a closed, convex set in Euclidean space E_n and q an arbitrary point in E_n. Given the usual inner product and associated Euclidean norm, we may speak of the unique point $p \equiv p(q, K)$ in K which is nearest to q.

Consider the case in which K is a polyhedral cone generated by a finite set of points $E = \{e_1, e_2, \ldots, e_N\}$. That is,

$$K = K(E) \equiv \left\{ \sum_{i=1}^{N} \lambda_i e_i : \lambda_i \geqslant 0,\ i = 1, \ldots, N \right\}.$$

Then it is possible, using the algorithm of this paper, to find p in a finite number of linear steps.† More importantly, the algorithm gives the barycentric coordinates of p with respect to $N^* \leqslant n$ linearly independent points of E.

This latter feature makes application to positive linear approximation possible. Given a linear functional L defined on a finite-dimensional function space Φ, the functions having a common domain D, the positive linear approximation problem consists of finding points $x_1, x_2, \ldots, x_{N^*}$ in D, $N^* \leqslant n$, and positive weights $\lambda_1, \lambda_2, \ldots, \lambda_{N^*}$ so that

$$Lf = \sum_{i=1}^{N^*} \lambda_i f(x_i) \tag{1}$$

for all $f \in \Phi$. Positivity of the weights is not necessary to achieve a representation (1),

Received March 11, 1974; revised March 4, 1975.
AMS (MOS) subject classifications (1970). Primary 52A25, 65D15; Secondary 65D30.
Key words and phrases. Convex set, nearest point, projection, positive linear approximation, linear algorithm, cubature.

†The algorithm can easily be modified to apply to convex polytopes (i.e., $\Sigma \lambda_i = 1$). It becomes, then, identical to a method developed independently by Wolfe [8].

but it is necessary if the representation is to be a good one with respect to convergence and computational stability (see Davis [2, p. 352]).

If $\varphi_1, \varphi_2, \ldots, \varphi_n$ is a basis for Φ, then the imbedding

$$M \equiv (L\varphi_1, \ldots, L\varphi_n)^T \quad \text{and} \quad e(x) \equiv (\varphi_1(x), \ldots, \varphi_n(x))^T, \qquad x \in D,$$

converts the positive approximation problem into a two-part representation problem in E_n:

A. Find an integer N and vectors $e_i = e(x_i)$, $i = 1, 2, \ldots, N$, (if they exist) so that M is contained in the convex cone they generate.

B. Find the barycentric coordinates $\lambda_1, \lambda_2, \ldots, \lambda_{N^*}$ of M with respect to $N^* \leqslant n$ points of the generating set $E = \{e_1, \ldots, e_N\}$.

The existence question implicit in A can be answered affirmatively under quite general conditions which we shall state in the next section.

Given existence of E, the algorithm may be used to solve A–B. At no point is there need to handle a linear system larger than $n \times n$, and each pass through the algorithm produces an intermediate solution. Each intermediate solution is itself an approximation to the final solution of A–B. This will be discussed in more detail in Section 5.

A related method for solving A–B has been given by Wilson [6]. It employs a sequence of applications of the simplex algorithm to progressively larger and larger systems. It can be shown, as in Wilson [6], [7] and Wilhelmsen [5], that the size of such systems in certain cases is asymptotically proportional to n^2. Furthermore, no intermediate solutions are obtained.

We discuss some background for the approximation problem and describe the tools needed for the algorithm in the next section. The algorithm is described in Section 3 and stated in Section 4. Section 5 contains some remarks on the application to A–B, and Section 6 has numerical examples.

2. Background and Preliminaries. Most interest in the positive approximation problem centers on the integration functional

$$Lf = \int_D \omega(x) f(x)\, dx, \qquad \omega(x) \geqslant 0. \tag{2}$$

The basic existence theorem is due to Tchakaloff [4]. Under rather general circumstances, there always exists a positive representation

$$Lf = \sum_{i=1}^{N^*} \lambda_i f(x_i), \qquad f \in \Phi, \tag{3}$$

where $\lambda_i > 0$ and $x_i \in D$, $i = 1, \ldots, N^* \leqslant n$.

We shall refer to Eq. (3) as a Tchakaloff representation for L. If T is a subset of D and there exists a Tchakaloff representation for L which uses points only in T, then we shall call it a Tchakaloff set. The Tchakaloff base $\mathcal{T}_L$, of L is the aggregate of all Tchakaloff sets in D.

A constructive proof of the Tchakaloff theorem was given by Davis [1]. Although his paper deals only with the integration functional (2), his results are easily

adapted to more general functionals. We shall state his theorem and its conditions for the more general case.

If $\varphi_1, \ldots, \varphi_n$ are continuous and real-valued on D, we say Φ satisfies the Krein condition if there is at least one $f \in \Phi$ which does not vanish on D. A linear functional L is said to be nonnegative if $Lf \geqslant 0$ whenever $f(x) \geqslant 0$ on D. L is (strictly) positive if $Lf > 0$ whenever $f(x) \geqslant 0$ on D and f is not identically zero.

THEOREM 1 (DAVIS). *Let Φ be the linear span of continuous, real-valued, linearly independent functions $\varphi_1, \varphi_2, \ldots, \varphi_n$ defined on a compact set D. Assume Φ satisfies the Krein condition and that L is a positive linear functional on Φ. If $\{x_i\}_{i=1}^{\infty}$, is an everywhere dense subset of D, then for sufficiently large m, the set $\{x_i\}_{i=1}^{m}$ is a Tchakaloff set.*

What Theorem 1 says is that under suitable circumstances $\mathcal{T}_L$ is nonempty. In fact, there are at least as many Tchakaloff sets as there are mutually disjoint dense sequences in D.

Following are some well-known properties of nearest points and support hyperplanes which are used in the algorithm. K is understood to be a convex polyhedral cone in E_n, and $\text{int}(K)$ denotes its relative interior.

Property P1. For $q \notin K$, $p = p(q, K)$ if and only if $H \equiv \{y \in E_n : \langle q - p, y\rangle = 0\}$ is a support hyperplane of K and $p \in H \cap K$. That is, $p \in H \cap K$ and $\langle q - p, k\rangle \leqslant 0$ for all $k \in K$. Observe that $q - p \perp H$.

Property P2. If $p \in \text{int}(K)$, then $K \subset H$.

Property P3. If $e_1, \ldots, e_N$ are linearly independent and $K = K(\{e_1, \ldots, e_N\})$, then

$$\text{int}(K) = \left\{ \sum_{i=1}^{N} \lambda_i e_i : \lambda_i > 0,\ i = 1, \ldots, N \right\}.$$

Let S be a linear subspace of E_n. The principal computational step in the algorithm is to compute $p = p(q, S)$ as a linear sum of a given basis $e_1, e_2, \ldots, e_N$ of S. This is a restricted form of the classical least squares problem and may be solved in a variety of ways.

Property P4. If S is one-dimensional, then $p = \langle q, e_1\rangle e_1 / \|e_1\|^2$ for any $e_1 \in S$.

Property P5. In general, $p = \Sigma_{i=1}^{N} \lambda_i e_i$, where $\lambda_1, \lambda_2, \ldots, \lambda_N$ are the unique solutions of the $N \times N$ linear system (known as the normal equations)

$$\sum_{i=1}^{N} \lambda_i \langle e_i, e_j\rangle = \langle q, e_j\rangle, \qquad j = 1, 2, \ldots, N. \tag{4}$$

3. Description of the Algorithm. We are given a point $q \in E_n$ and a convex polyhedral cone $K \subset E_n$ generated by the set $E = \{e_1, \ldots, e_N\}$. The object is to compute $p = p(q, K)$, the nearest point in K to q, in terms of E.

Briefly stated, the algorithm consists of computing a sequence of nearest points $p_1, p_2, \ldots$, to q in subcones $K_1, K_2, \ldots$ of K. Each subcone K_j is chosen so that $p_j \in \text{int}(K_j)$ and is closer to q than is p_{j-1}. Since there are at most finitely many subcones, the sequence must terminate at some step with $p = p_j$.

Finding K_1 and p_1 is simple. We examine E for a vector e_1 such that the scalar product $\langle q, e_1\rangle$ is positive and choose the half-ray containing e_1 as K_1. Then according to Property P4, $p_1 = \langle q, e_1\rangle e_1/\|e_1\|^2$.

The key step, of course, is to find p_{j+1}, given p_j. Suppose $p_j = p(q, K_j) \in \text{int}(K_j)$, where K_j is generated by some linearly independent subset $E_j \subset E$. If $p_j = q$, then we are done. Otherwise, there is a hyperplane H_j which contains K_j and is orthogonal to $q - p_j$. Now, either H_j is a support hyperplane of K, in which case $p_j = p$, or H_j is not, in which case at least one point $e^* \in E$ lies on the near side of H_j with respect to q; i.e., $\langle e^*, q - p_j\rangle > 0$. In the latter case, we adjoin e^* to E_j and begin a subcycle of steps designed to extract from this union a generating subset E_{j+1} for the next subcone K_{j+1}.

In a given step of the subcycle we have a cone C, a smallest subspace $S \equiv S(C)$ containing C, and a point $Q \in C$. Initially, for example, $C = K(\{e^*\} \cup E_j)$ and $Q = p_j$. Now, we compute $P = p(q, S)$. If $P \in C$, then $p_{j+1} = P$ and K_{j+1} is taken to be the smallest subcone, or face, of C which contains P. E_{j+1} consists of the generators of K_{j+1}. If $P \notin C$, then there is a unique point R in the interval (Q, P) which intersects the boundary of C in S. This can be computed, and we can determine the smallest face $C' \subset C$ that contains R. Notice that $\|q - R\| < \|q - Q\|$, and $R \in \text{int}(C')$. Furthermore, the generating set of C' is a strict subset of the generating set of C. Finally, we make the reassignments $C \leftarrow C'$, $Q \leftarrow R$, $S \leftarrow S(C')$ and repeat the step.

Because $\{e^*\} \cup E_j$ is finite and each step in the subcycle causes a reduction in the number of retained generators, the subcycle must eventually terminate successfully with p_{j+1} and K_{j+1} determined.

4. The Algorithm. Begin with a point $q \in E_n$ and a convex polyhedral cone $K \subset E_n$ generated by the set $E = \{e_1, e_2, \ldots, e_N\}$.

Step 0. Find a point $e_i \in E$ such that $\langle q, e_i\rangle > 0$. Set $E_1 = \{e_i\}$ and compute $p_1 = \langle q, e_i\rangle e_i/\|e_i\|^2$. If no such point exists, then take the origin as the nearest point in K to q. Otherwise, go to Step 1.

Step 1. Set $\eta_j = q - p_j$. If $\eta_j = 0$, then $q = p$, so stop. Otherwise, find $e^* \in E$ such that $\langle \eta_j, e^*\rangle > 0$. If no such point exists, then $p_j = p$, so stop. Otherwise, let $F = \{f_1, f_2, \ldots, f_m\}$ be a reindexing of $\{e^*\} \cup E_j$, let $\lambda_1, \ldots, \lambda_m$ be the barycentric coordinates of p_j in terms of F; that is, $p_j = \sum_{i=1}^m \lambda_i f_i$, and go to Step 2.

Step 2. Denote $S = \text{span}\{f_1, \ldots, f_m\}$ and compute (using Property P5, for example) $P = p(q, S) = \sum_{i=1}^m \beta_i f_i$. If $\beta_i \geqslant 0$, $i = 1, \ldots, m$, then set $E_{j+1} = \{f_i \in F: \beta_i > 0\}$, $p_{j+1} = P$, and go to Step 1. Otherwise, compute

$$\rho_i = \begin{cases} 1, & \beta_i \geqslant \lambda_i, \\ \lambda_i/(\lambda_i - \beta_i), & \beta_i < \lambda_i, \end{cases} \quad i = 1, \ldots, m,$$

$$\rho = \min_{1 \leqslant i \leqslant m} \lambda_i,$$

and

$$\gamma_i = (1 - \rho)\lambda_i + \rho\beta_i, \qquad i = 1, \ldots, m.$$

Go to Step 3.

Step 3. Set $F' = \{f_i \in F\colon \gamma_i > 0\}$ and $\Gamma = \{\gamma_i\colon \gamma_i > 0, i = 1, \ldots, m\}$. Reset m as the cardinality of F', $F = \{f_1, \ldots, f_m\}$ as a reindexing of F', $\{\lambda_1, \ldots, \lambda_m\}$ as a reindexing of Γ, and go to Step 2.

When p is not the origin, it is clear that if the algorithm terminates successfully, it will do so from Step 1. In this step we always have p_j as a candidate known in terms of its barycentric coordinates with respect to $E_j \subset E$. That is, after reindexing,

$$p_j = \sum_{i=1}^{N_j} \lambda_i e_i, \tag{5}$$

where $\lambda_1, \ldots, \lambda_{N_j}$ are positive constants, given by construction. For instance, if $j = 1$, then $\lambda_1 = \langle q, e_1\rangle / \|e_1\|^2$. If $j > 1$, then the coefficients are supplied from Step 2.

What has to be shown is that Step 2 supplies the appropriate p_{j+1} and E_{j+1} in a finite number of steps upon each successful completion of Step 1. If Step 1 cannot be completed, then we must have a solution $p = p_j$ given by (5). The reason is obvious if $q = p_j$. Otherwise, by Property P1, $H_j \equiv \{y \in E_n\colon \langle \eta_j, y\rangle = 0\}$ is a support hyperplane of K_j with normal $q - p_j$. If e^* cannot be found, then H_j is also a support hyperplane of K with normal $q - p_j$. So, $p = p_j$.

By showing E_j is always linearly independent, we obtain $N_j \leqslant n$. Finally, in showing that $\|q - p_{j+1}\| < \|q - p_j\|$, we can conclude that the algorithm will terminate in a finite time, since the number of distinct subsets $E_j \subset E$ is finite.

LEMMA 1. *E_j is linearly independent for all j.*

Proof. Using induction, we assume E_j is independent; certainly E_1 is. By Property P3, $p_j \in \operatorname{int}(K_j)$, since its coefficients $\lambda_1, \ldots, \lambda_{N_j}$ are all positive. Property P2 implies $E_j \subset H_j$, but $e^* \notin H_j$. Consequently, $\{e^*\} \cup E_j$ is linearly independent, and the lemma follows because $E_{j+1} \subset \{e^*\} \cup E_j$.

Step 2 consists of taking a point $Q = \Sigma_{i=1}^m \lambda_i f_i$ in C, the convex cone generated by F, computing $P = \Sigma_{i=1}^m \beta_i f_i$ as the nearest point to q in the subspace spanned by F, and finding $R = \rho Q + (1 - \rho)P$ as the unique point between P and Q which intersects the relative boundary of C. This is repeated as often as possible, letting R be the new Q and diminishing F by those generators which correspond to zero coefficients in the expansion of R. The step terminates as soon as $P \in C$. This must happen eventually, since F cannot be diminished indefinitely.

The next lemma describes what happens in Step 2.

LEMMA 2. *If $P \notin C$, then $0 < \rho < 1$, R is a nonzero point in the relative boundary of C, and $\|q - R\| < \|q - Q\|$.*

Proof. Notice that ρ is computed only if at least one β_i is negative, and each coefficient λ_i in the expansion of Q is positive except when $Q = p_j$. In the latter case, $\lambda^* = 0$, where λ^* is the coefficient of e^*. In any event, $\rho < 1$. To show $0 < \rho$, we must show that $\beta^* > 0$ when $Q = p_j$. Here, β^* is the coefficient of e^* in the expansion of P.

Recall that $P = \beta^* e^* + h$, where $h \in H_j$. If $\beta^* \leqslant 0$, then $\langle \eta_j, P\rangle \leqslant 0$. If $Q = p_j$

$= p(q, H_j)$, it follows that $\|q - P\| \geqslant \|q - Q\|$. But P is the nearest point to q in the linear span of $\{e^*\} \cup E_j$ and is certainly as close or closer to q than any point on the segment (Q, e^*). Such a point, say $U = \mu e^* + (1 - \mu)Q$, satisfies

$$\|q - U\| = [\|\eta_j\|^2 - 2(1 - \mu)\langle \eta_j, e^* \rangle + (1 - \mu)^2 \|e^* - Q\|^2]^{1/2}.$$

For small enough positive μ, $\|q - U\| < \|\eta_j\| = \|q - Q\|$. Thus, $\|q - P\| < \|q - Q\|$, so $\beta^* > 0$.

It is clear from the definition of ρ that the coefficients in the expansion of R are nonnegative but not all positive. Property P3 and Lemma 1 then imply that R belongs to the relative boundary of C.

Now, $q - P$ is normal to the linear span of F, so

$$\begin{aligned} \|q - R\| &= [\|q - P\|^2 + (1 - \rho)^2 \|P - Q\|^2]^{1/2} \\ &< [\|q - P\|^2 + \|P - Q\|^2]^{1/2} \\ &= \|q - Q\|. \end{aligned}$$

Finally, if $R = 0$, then $\|q\| < \|q - Q\| \leqslant \|q - p_j\|$, a contradiction of the fact that $p_j = p(q, K_j)$.

The condition $R \neq 0$ is important. It shows that F, hence E_{j+1}, is always nonempty. At worst, Step 2 might reduce F to a singleton $F = \{f_1\}$. In this event, we obtain $P = \langle q, f_1 \rangle f_1 / \|f_1\|^2$. If $\langle q, f_1 \rangle \leqslant 0$, then $R = 0$. This cannot happen, of course, so $P \in C$ and Step 2 terminates.

In summary, we have

THEOREM 2. *The algorithm described above supplies in a finite number of steps positive constants* $\lambda_1, \lambda_2, \ldots, \lambda_{N^*}$, $N^* \leqslant n$, *and points* $e_1, \ldots, e_{N^*}$ *in* E (*after reindexing*) *such that*

$$p = \sum_{i=1}^{N^*} \lambda_i e_i. \tag{6}$$

Proof. Lemmas 1 and 2 and the arguments preceding this theorem.

5. The Positive Linear Approximation Problem. Suppose T is a known finite Tchakaloff set. Then the positive linear approximation problem is solved by using the algorithm in Section 4 with $q = M$ and $E = \{e(x_i)\colon x_i \in T\}$.††

In practice, however, all we know is that any given everywhere dense (in D) sequence $S = \{x_i\}_{i=1}^{\infty}$ contains a finite Tchakaloff set. According to Theorem 1, given suitable conditions on D, Φ and L, each set $T = \{x_i\}_{i=1}^{m}$ is a Tchakaloff set for sufficiently large m. Let m^* be the smallest such number, and denote $T^* = \{x_i\}_{i=1}^{m^*}$. It is not necessary to know what m^* is, only that it exists.

Suppose we apply the algorithm to the infinite set $\{e(x_i)\}_{i=1}^{\infty}$ in an attempt to

---

††Notice that this provides an alternative to the Steinitz algorithm used in [1] to reduce the size of a positive representation of M.

find M. If the search for e_i in Step 0 and for e^* in Step 1 is always carried out in the order $e(x_1), e(x_2), \ldots$, then the fact that $T^* \in T_L$ guarantees that e_i or e^* will be encountered within the set $e(x_1), e(x_2), \ldots, e(x_{m^*})$. Since M is in the convex cone of these points, the algorithm must terminate eventually with $M = p_j$, yielding a Tchakaloff representation of L from the set T^*.

When T^* is large it might be profitable to stop the algorithm early; that is, accept an approximation to L rather than carry out what might be time-consuming computations to find an exact representation. This is a feasible alternative due to the "intermediate solution" characteristic of $p_1, p_2, \ldots$.

For example, suppose we have computed $p_j = \Sigma_{i=1}^{m} \lambda_i e(x_i)$. Let $f \in \Phi$ have the expansion $f(x) = a_1\varphi_1(x) + \cdots + a_n\varphi_n(x)$. Denote the vector $(a_1, \ldots, a_n)^T$ by A_n and define the functional L_j by

$$L_j f = \sum_{i=1}^{m} \lambda_i f(x_i).$$

Then, we have

$$|(L - L_j)f| = |\langle A_n, M - p_j\rangle| \leqslant \|A_n\| \, \|M - p_j\|.$$

As j increases, $\|M - p_j\|$ decreases (eventually vanishing), so at some stage a reasonable approximation to L is given by L_j. This reasonableness becomes more apparent if we view the problem in a larger setting, that in which L is the restriction to Φ of a larger operator.

Let $\varphi_1, \varphi_2, \ldots$ be a basis for the infinite-dimensional linear space A. Assume that evaluation functionals $(f \longrightarrow f(x))$ are bounded and that each $f \in \mathsf{A}$ has a uniformly convergent expansion

$$f(x) = \sum_{k=1}^{\infty} a_k \varphi_k(x).$$

Let L be the restriction to Φ of a bounded linear operator L on A, and define L_j by $L_j f = \Sigma_{i=1}^{m} \lambda_i f(x_i)$. Then

$$(L - L_j)f = \langle A_n, M - p_j\rangle + R_{nj}f, \tag{7}$$

where

$$R_{nj}f = \sum_{k=n+1}^{\infty} a_k(L - L_j)\varphi_k.$$

It is normally the case in practice that the order of magnitude of $R_{nj}f$ is small relative to that of Lf and changes very little as j increases. So in order to make $(L - L_j)f$ have approximately the same order of magnitude as that of $(L - L_J)f$, where J is such that $p_J = M$, it suffices to make $\langle A_n, M - p_j\rangle$ small. This, of course, happens automatically as j increases; in fact, we have an estimate for the size of $\langle A_n, M - p_j\rangle$, since $\|M - p_j\|$ is known.

The selection of e^* in Step 1 should be made with some concern about the resulting computational difficulties inherent in the finding of P. The smaller $\langle \eta_j, e^* \rangle$ is with respect to the size of η_j, the more ill-conditioned will be the system whose solution yields P. The next theorem states a guideline which can be used to maximize $\langle \eta_j, e^* \rangle$.

THEOREM 3. *Assume that at least one function* $P \in \Phi$ *is bounded below by* $\mu > 0$ *on* D. *Choose any* $\delta \in (0, 1)$. *If* T *is a Tchakaloff set and* $\eta_j \neq 0$, *then it is always possible to fine* e^* *in Step* 1 *so that*

$$\langle \eta_j, e^* \rangle > \delta \mu \|\eta_j\|^2 / nLP. \tag{8}$$

Proof. The Tchakaloff representation (3) may be written as $M = \sum_{i=1}^{N^*} \lambda_i c_i$, where $c_i = e(x_i)$, $i = 1, \ldots, N^*$. Since $p_j = p(M, K_j)$, we have $\langle \eta_j, p_j \rangle = 0$. But $\|\eta_j\|^2 = \langle \eta_j, M - p_j \rangle = \langle \eta_j, M \rangle$, or

$$\|\eta_j\|^2 = \sum_{i=1}^{N^*} \lambda_i \langle \eta_j, c_i \rangle.$$

At least one term in the sum must satisfy

$$\lambda_i \langle \eta_j, c_i \rangle \geqslant \|\eta_j\|^2 / n. \tag{9}$$

Now,

$$LP = \sum_{i=1}^{N^*} \lambda_i P(x_i) \geqslant \mu \sum_{i=1}^{N^*} \lambda_i,$$

showing that no λ_i can exceed LP/μ. Using (9), we get

$$\langle \eta_j, c_i \rangle \geqslant \mu \|\eta_j\|^2 / nLP > \delta \mu \|\eta_j\|^2 / nLP.$$

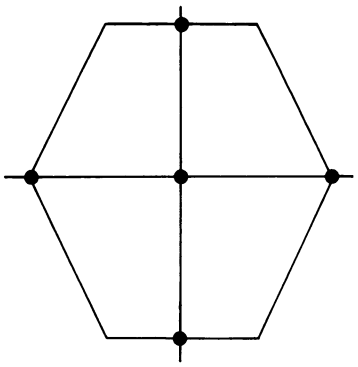
It suffices to take $e^* = c_i$.

6. Numerical Results. We used the algorithm to obtain positive numerical integration rules of polynomial precision $k = 3, 5, 7$ for the hexagon, $k = 2, 3, 4, 5$ for the quarter disc, and $k = 3$ for the 3-simplex (Figures 1–6).

Rather than use a dense sequence $S = \{x_i\}_{i=1}^{\infty}$ (see Theorem 1), we employed a set sequence $S_1, S_2, \ldots$ such that $\bigcup_{j=1}^{\infty} S_j$ was dense in D. These sets are defined as follows: when D is a bounded set in E_r, then there is a hypercube (or "pie slice" for polar coordinates) $C \equiv \{s_k \leqslant x_k \leqslant s_k + d_k,\ k = 1, \ldots, r\}$ which contains D. We denote by (md) the set of all points $(m_1 d_1, \ldots, m_r d_r)$ obtained as $m = (m_1, m_2, \ldots, m_r)$ ranges through the lattice of points in E_r which have nonnegative integer coordinates. Taking $s = (s_1, \ldots, s_r)$, we can define

$$S_j = \{s + \alpha_j (md)\} \cap D,$$

where α_j is a positive scalar. If $\alpha_j \longrightarrow 0$, then $\bigcup_{j=1}^{\infty} S_j$ is dense in D. In applying the algorithm, we arranged for the search in Step 1 to exhaust first the points in S_1, then S_2 and so on.



$s = (-1, -1)$
$d = (2, 2)$
$\alpha_j = 2^{1-j}$

FIGURE 1

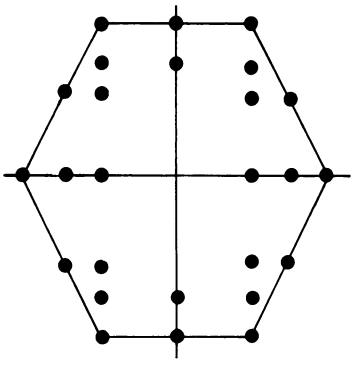
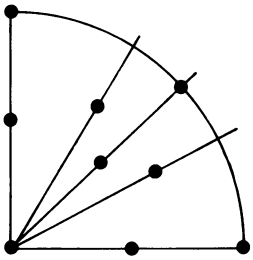


FIGURE 2



$s = (0, 0)$
$d = (1, \pi/2)$
$\alpha_j = 1/(1 + j)$

FIGURE 3

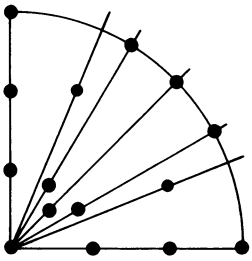
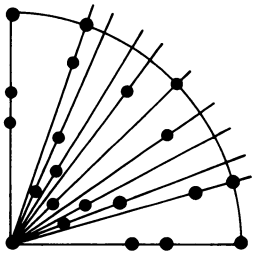


FIGURE 4



$s = (0, 0, 0)$
$d = (1, 1, 1)$
$\alpha_j = 3^{1-j}$

FIGURE 5

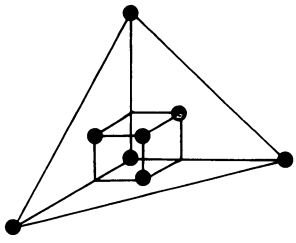


FIGURE 6

The test results can be found in Table 1 and the figures. The weights of the rules have not been recorded, since these are easily computed once the points are known. Observe the striking symmetry present in each example. This appears to occur whenever the sets $S_1, S_2, \ldots$ reflect the symmetry characteristics of D. Also, note that the rule for the 3-simplex is a minimum point rule (see Stroud [3]).

In Table 1, k refers to the polynomial precision, n to the dimension of the polynomial space Φ, and N^* to the number of points used in the rule. The number of passes through Step 1 is denoted by N_1. We found that Step 2 almost always yielded $p_{j+1} = P$ on the first try; consequently, each pass through Step 1 corresponded to the solution of slightly more than one linear system on the average. The size of the largest linear system encountered in Step 2 is given by $N_{\max}$, and N_{ave} denotes the average system size. This average was computed as $[(\Sigma N^3)/N_1]^{1/3}$, where the sum ranges over all systems solved in Step 2.

TABLE 1

Figure	Region	k	N^*	n	N_{max}	N_{ave}	N_1
1	Hex	3	5	10	6	4	6
not shown	"	5	13	21	13	10	16
2	"	7	27	36	28	20	36
not shown	Disc	2	5	6	5	3	5
3	"	3	9	10	10	7	10
4	"	4	15	15	15	13	29
5	"	5	21	21	21	19	60
6	Simplex	3	8	20	11	8	13

Department of Mathematics
The University of Georgia
Athens, Georgia 30602

1. P. J. DAVIS, "A construction of nonnegative approximate quadratures," *Math. Comp.*, v. 21, 1967, pp. 578–582. MR 36 #5584.

2. P. J. DAVIS, *Interpolation and Approximation*, Blaisdell, New York, 1963. MR 28 #393.

3. A. H. STROUD, *Approximate Calculation of Multiple Integrals*, Prentice-Hall Ser. in Automatic Computation, Prentice-Hall, Englewood Cliffs, N. J., 1971. MR 48 #5348.

4. V. TCHAKALOFF, "Formules de cubatures mécaniques à coefficients non-négatifs," *Bull. Sci. Math.* (2), v. 81, 1957, pp. 123–134. MR 20 #1145.

5. D. R. WILHELMSEN, "Nonnegative cubature on convex sets," *SIAM J. Numer. Anal.*, v. 11, 1974, pp. 332–346.

6. M. W. WILSON, "A general algorithm for nonnegative quadrature formulas," *Math. Comp.*, v. 23, 1969, pp. 253–258. MR 39 #3705.

7. M. W. WILSON, "Necessary and sufficient conditions for equidistant quadrature formula," *SIAM J. Numer. Anal.*, v. 7, 1970, pp. 134–141. MR 43 #8241.

8. P. WOLF, "Finding the nearest point in a polytope," *Math. Programming.* (To appear.)