Markov Chains and Computer-Aided Geometric Design: Part I—Problems and Constraints

RONALD N. GOLDMAN
Control Data Corporation

The connection between probability theory and computer-aided geometric design is explored further. Markov chains are shown to be associated with solutions to the following three geometric problems: (1) Given a curve $B[P](t)$, alter the shape of the curve by changing the control points P. (2) Given a curve $B[P](t)$, alter the shape of the curve by changing the blending functions $B(t)$. (3) Given a curve $B[P](t)$ and blending functions $D(t)$, find control points Q so that $D[Q](t) = B[P](t)$. Constraints imposed on these Markov chains by computer-aided geometric design are also derived.

CR Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*curve, surface, solid, and object representations*

General Terms: Design, Theory

Additional Key Words and Phrases: Markov chain, probability distribution, stochastic process

1. INTRODUCTION

To construct a free-form shape with the aid of a computer, a designer generally specifies only a relatively small collection of control points. The burden then shifts to the computer to blend these points together to form the desired free-form curve or surface. If the user is dissatisfied with the initial shape generated by the computer, he can alter the number and the location of his control points. The computer will then generate a new shape from this new data. Typically a designer will iterate this procedure many times before he is completely satisfied with the final shape.

To blend an array of control points $P = (P_0, \ldots, P_n)$ into a free-form parametric curve, the system is endowed internally with a collection of blending functions $B(t) = (B_0(t), \ldots, B_n(t))$. These blending functions are not arbitrary; they must satisfy certain very specific conditions to ensure that the corresponding curves defined parametrically by the equation

$$B[P](t) = \sum_j B_j(t)P_j, \qquad 0 \leq t \leq 1,$$

Author's address: Control Data Corporation, 4201 Lexington Avenue North, Arden Hills, MN 55112.

are well-behaved. Minimally, the blending functions are generally required to satisfy the following two conditions:

$$\sum_j B_j(t) = 1, \qquad 0 \le t \le 1, \tag{1}$$

$$B_j(t) \ge 0, \qquad 0 \le t \le 1. \tag{2}$$

The first condition ensures that the curve is well defined, that is, that it depends only on the choice of the points P and not on the location of the coordinate origin. The second condition forces the curve to lie somewhere in the general proximity of the control points, specifically, within their convex hull.

Conditions (1) and (2) are the defining characteristics of discrete probability distributions. Thus in computer-aided geometric design the blending functions are often discrete probability distributions. For example, the blending functions for the Bezier curves are simply the binomial distribution. Since blending functions are often probability distributions, there is, of necessity, a deep and fundamental connection between classical probability theory and computer-aided geometric design [5–8, 10]. Markov chains play an important role in the theory of discrete stochastic processes. This suggests that they too may have a prominent place both in approximation theory and in computer-aided geometric design. Indeed, in [14] the authors use Markov chains to study iterates of Bernstein and B-spline operators. In this paper we shall explore some possible applications for Markov chains in the field of computer-aided geometric design.

Because of its length, this paper is divided into two parts. In Part I we discuss some standard problems in computer-aided geometric design whose solutions lead to Markov chains, and we derive some analytic constraints that computer-aided geometric design imposes on these Markov chains. Part II is devoted to examples of Markov chains that either satisfy the constraints or solve the problems posed in Part I.

2. PROBLEMS

A Markov chain is a square matrix $M = (M_{jk})$ such that

$$\sum_k M_{jk} = 1, \tag{1}$$

$$M_{jk} \ge 0. \tag{2}$$

If a matrix M satisfies these conditions but has more rows than columns, we shall, when necessary, simply annex additional columns of zeros to form a square matrix. The standard probabilistic interpretation of a Markov chain is that the indices j represent a set of physical states and the entries M_{jk} represent the probability of a direct transition from state j to state k independent of any history of past states [2]. Since probabilities are always positive and the probabilities of a collection of mutually exclusive events one of which must occur necessarily sum to 1, the entries of M must be positive and the rows of M must sum to 1. Thus probabilistic models naturally generate conditions (1) and (2). We shall not be too concerned with this probabilistic interpretation here except to note that because of this interpretation Markov chains arise naturally in many physical situations (see Example 1 of Part II [9]).

Let us return now to the context of computer-aided geometric design. Consider a collection of control points $P = (P_0, \ldots, P_n)$ and a collection of blending functions $B(t) = (B_0(t), \ldots, B_n(t))$. Let ${}^tP$ denote the transpose of P, and let $*$ denote matrix multiplication. Then the curve determined by the control points P and the blending functions $B(t)$ is defined parametrically by the equation

$$B[P](t) = \sum_j B_j(t)P_j = B(t) * {}^tP.$$

If a designer is dissatisfied with the curve $B[P](t)$, he has two options: he can change the control points P, or he can change the blending functions $B(t)$.

Suppose he decides to change the control points. How should he proceed? He can, of course, simply replace each point P_j by a new point Q_j. In fact, this is typically what designers do today. However, this procedure has several drawbacks. First, it is certainly tedious to alter each control point individually. Second, changing a single control point is a local operation, but the affect on the curve is not local. A new control point may achieve a desirable result locally only to have unwanted effects globally. This fact accounts for the recent popularity of B-spline curves in which the overall affect of individual control points is clearly localized. Third, altering each control point independently takes no account of the shape of the current curve even though this shape may have been arrived at only after many laborious iterations. Fourth, it is hard to achieve good global effects with this technique.

There is another way. We can alter all the points at once and at the same time achieve both good global and good local effects. The technique is simply to define all the new control points $Q = (Q_0, \ldots, Q_n)$ simultaneously as linear combinations of the old control points $P = (P_0, \ldots, P_n)$. That is, simply set

$$Q_j = \sum_k M_{jk}P_k, \qquad {}^tQ = M * {}^tP$$

for some predefined matrix $M = (M_{jk})$. In order for the points Q to be well defined—that is, independent of the coordinate origin—the matrix M must satisfy

$$\sum_k M_{jk} = 1. \tag{1}$$

If, in addition, we insist that the new control points Q lie within the convex hull of the old control points P—which is certainly reasonable, since we have already insisted that the curve itself lie inside this convex hull—then the matrix M must also satisfy

$$M_{jk} \geq 0. \tag{2}$$

Thus the matrix M that transforms the old control points P into the new control points Q must be a Markov chain.

Rather than alter the control points P, it is possible to achieve the same effect by changing the blending functions $B(t)$. Define a new collection of blending functions $D(t) = (D_0(t), \ldots, D_n(t))$ by setting

$$D_k(t) = \sum_j M_{jk}B_j(t), \qquad D(t) = B(t) * M.$$

Since M is a Markov chain, $D(t)$ is a distribution. In fact, all the terms in the summation formula for $D(t)$ are positive, and

$$\sum_k D_k(t) = \sum_k \sum_j M_{jk} B_j(t)$$
$$= \sum_j \left[\sum_k M_{jk} \right] B_j(t) = \sum_j B_j(t).$$

Therefore $D(t)$ satisfies

$$\sum_k D_k(t) = 1, \qquad 0 \le t \le 1,$$
$$D_k(t) \ge 0, \qquad 0 \le t \le 1.$$

Moreover, by construction,

$$D[P](t) = D(t) * {}^tP = [B(t) * M] * {}^tP$$
$$= B(t) * [M * {}^tP] = B(t) * {}^tQ$$
$$= B[Q](t).$$

Therefore the D-curve with control points P is identical to the B-curve with control points Q. Thus it is really just a matter of convenience whether we decide to alter the control points P or the blending functions $B(t)$. The final effect is the same, though different users may prefer one approach over the other.

So far we have considered only Markov chains M whose entries M_{jk} are constants. Suppose, however, that the entries, and hence the matrix, depend on some scalar parameters $r = (r_1, \ldots, r_m)$. We could then alter the curve simply by varying these scalars. The new control points, or alternatively the new blending functions, would then vary automatically with r. Thus once we understand precisely how the choice of r affects the general shape of the curve, we would have an elementary technique for changing this shape simply by adjusting some scalar parameters. This is a very exciting prospect; we shall return to this topic again in Part II [9].

In our previous constructions, we altered only the position of the control points. However, sometimes we wish to achieve greater control simply by increasing the number of control points. We then need to change the distribution as well as the control points, since each distribution allows only a fixed number of control points. The problem then is this: Given a curve $B[P](t)$ and a distribution $D(t)$ that allows a greater number of control points than $B(t)$, find control points Q such that

$$D[Q](t) = B[P](t).$$

If we can duplicate B-curves $B[P](t)$ with D-curves $D[Q](t)$, then later we can manipulate the additional control points Q to alter the shape of the original curve. This type of problem arises naturally in many situations in computer-aided geometric design. For Bezier curves it is equivalent to the problem of increasing the degree of the Bernstein polynomials [4]; for B-splines it is equivalent to the problem of expanding the knot vector [3].

Let $D(t) = (D_0(t), \ldots, D_m(t))$ be a distribution consisting of $m + 1$ linearly independent functions, and let span$[D(t)]$ be the set of all functions that are

linear combinations of $D_0(t), \ldots, D_m(t)$. If

$$\text{span}[D(t)] \supseteq B(t),$$

then it is always possible to solve the preceding problem. In fact, in this case there is a matrix $M = (M_{jk})$ that

$$B_k(t) = \sum_j M_{jk} D_j(t), \qquad B(t) = D(t) * M.$$

Moreover, by construction,

$$\sum_j D_j(t) = 1 = \sum_k B_k(t) = \sum_k \sum_j M_{jk} D_j(t)$$
$$= \sum_j \left[\sum_k M_{jk} \right] D_j(t).$$

Now since the functions $D_0(t), \ldots, D_m(t)$ are, by assumption, linearly independent, it follows that the coefficients of $D_j(t)$ on both sides of this equation must be identical. Therefore the matrix $M = (M_{jk})$ must satisfy

$$\sum_k M_{jk} = 1. \tag{1}$$

Define

$$Q_j = \sum_k M_{jk} P_k, \qquad {}^tQ = M * {}^tP.$$

Because the matrix $M = (M_{jk})$ satisfies condition (1), it follows that the points $Q = (Q_0, \ldots, Q_m)$ are well defined. Moreover we again have

$$\begin{aligned} D[Q](t) &= D(t) * {}^tQ = D(t) * [M * {}^tP] \\ &= [D(t) * M] * {}^tP = B(t) * {}^tP \\ &= B[P](t). \end{aligned}$$

Thus the points Q are indeed the required control points. Furthermore, the problem of finding these control points has been reduced to finding the matrix M that transforms the distribution $D(t)$ into the distribution $B(t)$. Hence if $\text{span}[D(t)] \supseteq B(t)$, appropriate points Q always exist, in theory. However, a practical technique for constructing Q or M may still be hard to find—see Part II [9], Examples 6 and 7.

We have not yet insisted that the entries of M satisfy

$$M_{jk} \geq 0. \tag{2}$$

Thus M is not necessarily a Markov chain. However, if this condition is satisfied, then M is a Markov chain and the D control points Q will necessarily lie inside the convex hull of the B control points P. Since the curve is automatically confined to the convex hull of its control points, this construction further restricts the exact location of the curve. Narrowing the convex hull in which a curve lies is useful for determining whether two curves actually intersect, for clearly two curves cannot intersect if the convex hulls of their respective control points fail to intersect.

If two B curves do intersect, then subdivision algorithms may be used to locate their points of intersection. A subdivision algorithm is a technique for generating control points $Q(r)$, $R(r)$ for any parameter r such that

$$B[Q(r)](t) = B[P](rt), \qquad 0 \le t \le 1,$$
$$B[R(r)](t) = B[P](r + [1 - r]t), \qquad 0 \le t \le 1.$$

By repeatedly subdividing two curves and determining which segments have intersecting convex hulls, the actual intersection points may be computed to any desired accuracy. We would like to know under what conditions control points $Q(r)$, $R(r)$ that subdivide the curve $B[P](t)$ at r are guaranteed to exist. In addition, when these control points do exist, we would like to know how to find them.

Actually, we have already solved this problem. If we regard $B(rt)$, $B(r + [1 - r]t)$ as the old blending functions and $B(t)$ as the new blending functions, then what we seek to do is to duplicate $B(rt)$ and $B(r + [1 - r]t)$ curves with $B(t)$ curves. But we have just shown that this can be done whenever

$$\text{span}[B(t)] \supseteq B(rt), B(r + [1 - r]t).$$

In this case, there are matrices $M(r)$, $N(r)$ such that

$$B_k(rt) = \sum M_{jk}(r)B_j(t),$$
$$B(rt) = B(t) * M(r),$$

and

$$B_k(r + [1 - r]t) = \sum N_{jk}(r)B_j(t),$$
$$B(r + [1 - r]t) = B(t) * N(r),$$

and the matrices $M(r)$, $N(r)$ satisfy the identity

$$\sum_k M_{jk}(r) = \sum_k N_{jk}(r) = 1. \tag{1}$$

The required control points are given by

$$Q_j(r) = \sum M_{jk}(r)P_k, \qquad {}^tQ(r) = M(r) * {}^tP,$$

and

$$R_j(r) = \sum N_{jk}(r)P_k, \qquad {}^tR(r) = N(r) * {}^tP.$$

Thus the problem of finding the desired control points $Q(r)$, $R(r)$ has been reduced to finding the matrices $M(r)$, $N(r)$ that respectively transform the distribution $B(t)$ into the distributions $B(rt)$, $B(r + [1 - r]t)$. Although this argument shows that if $\text{span}[B(t)] \supseteq B(rt), B(r + [1 - r]t)$, subdivision is always possible in principle, practical subdivision algorithms may still be hard to find.

Again we have not yet insisted that the matrices $M(r)$, $N(r)$ satisfy the relation

$$M_{jk}(r), N_{jk}(r) \ge 0. \tag{2}$$

Thus these matrices are not necessarily Markov chains. However, if this condition is satisfied, then these matrices are Markov chains and the new control points $Q(r)$, $R(r)$ will necessarily lie in the convex hull of the original control points P.

Now this condition is actually critical for our application. Indeed, if the new control points fail to lie in the convex hull of the original control points, then the subdivision algorithm will not necessarily narrow the convex hull. Such curves do exist [1], and for these curves subdivision algorithms are not particularly useful for locating points of intersection. We will return to the subdivision problem again in Part II [9].

To recapitulate, Markov chains arise in computer-aided geometric design in three types of problems:

Problem 1. Given a curve $B[P](t)$, alter the physical shape of the curve by changing the control points P.

Problem 2. Given a curve $B[P](t)$, alter the physical shape of the curve by changing the blending functions $B(t)$.

Problem 3. Given a curve $B[P](t)$ and a discrete distribution $D(t)$, find new control points Q so that the curve $D[Q](t)$ is identical to the curve $B[P](t)$.

A solution to the last problem is called a conversion algorithm, since it allows the user to convert B-curves into D-curves. In this context, subdivision algorithms are simply algorithms for converting $B(rt)$ and $B(r + [1 - r]t)$ curves into $B(t)$ curves.

These three problems are summarized schematically in the following table:

	Curve	Blending Functions	Control Points
(1)	New	Old	New
(2)	New	New	Old
(3)	Old	New	New

To solve the first two problems, we must define a new curve by changing either the control points or the blending functions of the original curve. These two constructions are equivalent. In one, a Markov chain is applied to transform the control points; in the other, the same Markov chain is applied to transform the blending functions. The method that is actually used will depend on what is more sacred to the designer—his control points or his curve type (blending functions). If, in addition, the Markov chain is a function of some scalar parameters r, then these constructions open up the possibility of changing the shape of the curve simply by varying some scalar parameters.

To solve the third problem, we seek a conversion algorithm; that is, we wish to change the curve type without actually altering the shape of the curve. This is usually done to give the designer additional flexibility by increasing the number of control points. However, it might be done just because the new curve type is more convenient, or more natural, or simply more well known. In any event, we assume in this case that the original control points as well as both the old and the new blending functions are known, and we seek only the control points corresponding to the new blending functions. To find these control points, we need only find the Markov chain that transforms the new blending functions back into the original blending functions. By applying this matrix to the old control points, we generate the new control points. This converts the curve type without affecting the actual curve. We can then alter these new control points to change the shape of the original curve.

Surfaces in computer-aided geometric design are created in much the same way as curves. Given a square array of points $P = (P_{jk})$ and a distribution $B(t) = (B_0(t), \ldots, B_n(t))$, we define the tensor product surface $B[P](u, v)$ by

$$B[P](u, v) = \sum_{j,k} B_j(u)B_k(v)P_{jk} = B(u) * P * {}^tB(v).$$

If M is a Markov chain, then we can alter the control points or the blending functions of this surface simply by inserting M into this product. This generates the new surface

$$B(u) * (M * P * {}^tM) * {}^tB(v) = (B(u) * M) * P * {}^t(B(v) * M).$$

By using two distinct distributions and two distinct Markov chains, this technique can easily be generalized to arbitrary rectangular arrays of points. Since the theory for tensor product surfaces is much the same as the theory for simple curves, we shall not pursue this topic explicitly any further in this paper.

3. CONSTRAINTS

Suppose we adopt the perspective of Problem 2 of Section 2; that is, we wish to alter the curve $B[P](t)$ by changing its blending functions $B(t)$ but not its control points P. We can do this by introducing a matrix $M = (M_{jk})$ and defining new blending functions $D(t)$ by setting

$$D_k(t) = \sum M_{jk}B_j(t), \qquad D(t) = B(t) * M.$$

As we have seen, $B(t)$, $D(t)$ must both be discrete probability distributions. However, to generate really well-behaved curves, we must assume that $B(t)$, $D(t)$ satisfy some additional conditions; completely arbitrary discrete distributions do not always generate very well-behaved curves. The list of properties in Table I is adapted from [8], where the reader will find detailed explanations and derivations of each of these conditions.

The properties of the blending functions listed in this table are not all independent. Indeed, it is easy to show that

$$\text{Properties 1, 2, 5} \Rightarrow \text{Property 3},$$
$$\text{Property 6} \Rightarrow \text{Property 7}.$$

Again for details see [8].

The question we now pose is this: Suppose that the original curve $B[P](t)$ and the original blending functions $B(t)$ are known to satisfy some property on this list; if we insist that the new curve $D[P](t)$ and the new blending functions $D(t)$ also satisfy this same property, what constraint does this impose on the transformation matrix M? We have already seen in Section 2 that, if the blending functions $B_0(t), \ldots, B_n(t)$ are linearly independent and $\sum_k B_k(t) = 1$, then

$$(1) \qquad \sum_k D_k(t) = 1 \quad \Leftrightarrow \quad \sum_k M_{jk} = 1.$$

Moreover, it is certainly true that, if $B_k(t) \geq 0$, then

$$(2) \qquad D_k(t) \geq 0 \quad \Leftarrow \quad M_{jk} \geq 0.$$

Table I

	Curve		Blending functions
1.	$B[P](t)$ is well defined	$\Leftrightarrow$	$\sum_k B_k(t) = 1$
2.	$B[P](t) \subseteq$ convex hull (P)	$\Leftrightarrow$	$B_k(t) \geq 0$
3.	$B[P](t)$ interpolates P_0, P_n	$\Leftrightarrow$	$B_k(0) = \begin{cases} 0, & k \neq 0 \\ 1, & k = 0 \end{cases}$ $B_k(1) = \begin{cases} 0, & k \neq n \\ 1, & k = n \end{cases}$
4.	$B[P](t)$ is symmetric	$\Leftrightarrow$	$B_k(t) = B_{n-k}(1-t)$
5.	$B[P](t)$ exactly reproduces straight lines	$\Leftrightarrow$	$\sum_k kB_k(t) = nt$
6.	$B[P](t)$ is variation diminishing	$\Leftarrow$	$B(t)$ satisfies Descartes' Law of Signs
7.	$B[P](t) = B[R](t)$ iff $P = R$	$\Leftrightarrow$	$B_0(t), \ldots, B_n(t)$ are linearly independent

Notice, however, that this second implication is only valid in one direction; in Part II [9] we shall give a counterexample to show that the reverse implication is not necessarily valid. These two conditions together assert that M is a Markov chain. We shall now explore what other conditions are imposed on this Markov chain by the other properties in our list.

To begin, suppose that $B_0(t), \ldots, B_n(t)$ are linearly independent functions. Since

$$D_k(t) = \sum M_{jk} B_j(t), \qquad D(t) = B(t) * M,$$

it follows that the functions $D_0(t), \ldots, D_n(t)$ are linearly independent iff the transformation matrix M is nonsingular. Thus

$$D_0(t), \ldots, D_n(t) \text{ are linearly independent} \quad \Leftrightarrow \quad \text{Det}(M) \neq 0.$$

Throughout the remainder of this discussion, we shall always assume that the functions $B_0(t), \ldots, B_n(t)$ are linearly independent.

Now suppose that $B(t)$ satisfies property 3. Then

$$M_{0k} = \sum_j M_{jk} B_j(0) = D_k(0),$$
$$M_{nk} = \sum_j M_{jk} B_j(1) = D_k(1).$$

Therefore

$$D_k(0) = \begin{cases} 0, & k \neq 0 \\ 1, & k = 0 \end{cases} \quad \Leftrightarrow \quad M_{0k} = \begin{cases} 0, & k \neq 0, \\ 1, & k = 0; \end{cases}$$

$$D_k(1) = \begin{cases} 0, & k \neq n \\ 1, & k = n \end{cases} \quad \Leftrightarrow \quad M_{nk} = \begin{cases} 0, & k \neq n, \\ 1, & k = n. \end{cases}$$

This condition can be interpreted in the following manner. Let

$$u_0 = (1, 0, \ldots, 0), \qquad u_n = (0, \ldots, 0, 1).$$

Then

$$M_{0k} = \begin{cases} 0, & k \neq 0 \\ 1, & k = 0 \end{cases} \quad \Leftrightarrow \quad u_0 * M = u_0,$$

$$M_{nk} = \begin{cases} 0, & k \neq n \\ 1, & k = n \end{cases} \quad \Leftrightarrow \quad u_n * M = u_n.$$

Thus u_0, u_n are eigenvectors of M corresponding to the eigenvalue 1. In the language of Markov chains, these conditions assert that 0, n are absorbing states of the Markov chain M because the probability of a transition out of these states is 0.

Next suppose that $B(t)$ satisfies property 4. Then

$$D_k(t) = \sum_j M_{jk} B_j(t),$$

$$D_{n-k}(1-t) = \sum_j M_{n-j,n-k} B_{n-j}(1-t) = \sum_j M_{n-j,n-k} B_j(t).$$

Since the functions $B_0(t), \ldots, B_n(t)$ are linearly independent, it follows immediately that

$$D_k(t) = D_{n-k}(1-t) \quad \Leftrightarrow \quad M_{n-j,n-k} = M_{j,k}.$$

Now suppose that $B(t)$ satisfies property 5. Then

$$\sum_j jB_j(t) = nt,$$

$$\sum_k kD_k(t) = \sum_k k \sum_j M_{jk} B_j(t) = \sum_j \left[\sum_k kM_{jk} \right] B_j(t).$$

Again since $B_0(t), \ldots, B_n(t)$ are linearly independent functions, it follows immediately that

$$\sum_k kD_k(t) = nt \quad \Leftrightarrow \quad \sum_k kM_{jk} = j.$$

This constraint on M can be interpreted in the following manner. Let $v = (0, 1, \ldots, n)$; then

$$\sum_k kM_{jk} = j \quad \Leftrightarrow \quad M * {}^t v = {}^t v.$$

Thus ${}^t v$ is an eigenvector of M corresponding to the eigenvalue 1. Analogously, if $w = (1, 1, \ldots, 1)$, then the first condition is equivalent to the statement that ${}^t w$ is also an eigenvector of M corresponding to the eigenvalue 1. That is,

$$\sum_k M_{jk} = 1 \quad \Leftrightarrow \quad M * {}^t w = {}^t w.$$

Finally, suppose that $B(t)$ satisfies property 6; what can we say about the matrix M? To begin, we recall Descartes' Law of Signs.

Descartes' Law of Signs. A collection of functions $B(t) = (B_0(t), \ldots, B_n(t))$ is said to satisfy Descartes' Law of Signs in the interval (a, b) iff for every set of

constants $C = (c_0, \ldots, c_n)$,

$$\text{zeros in } (a, b)\left[\sum_k c_k B_k(t)\right] \leq \text{sign alternations } (c_0, \ldots, c_n).$$

It is well known, for example, that the power functions satisfy Descartes' Law of Signs in the interval $(0, \infty)$, and that the Bernstein polynomials satisfy Descartes' Law of Signs in the interval $(0, 1)$ [8]. Notice too that any collection of functions $B(t)$ that satisfy Descartes' Law of Signs is necessarily linearly independent.

Now let $M = (M_{jk})$ be a matrix. For $j_0 < j_1 < \cdots < j_m$ and $k_0 < k_1 < \cdots < k_m$, define

$$M\begin{pmatrix} j_0 & \cdots & j_m \\ k_0 & \cdots & k_m \end{pmatrix} = \text{Det} \begin{vmatrix} M_{j_0 k_0} & \cdots & M_{j_0 k_m} \\ \vdots & & \vdots \\ M_{j_m k_0} & \cdots & M_{j_m k_m} \end{vmatrix}.$$

We shall say that M is a Descartes matrix, or simply a D-matrix, iff

(1) for each m the determinants $M\binom{j_0 \cdots j_m}{k_0 \cdots k_m}$ are all of one sign $s_m = \pm 1$;

(2) for each $j_0 \cdots j_m$ there exists a $k_0 \cdots k_m$ such that

$$M\begin{pmatrix} j_0 & \cdots j_m \\ k_0 & \cdots k_m \end{pmatrix} \neq 0,$$

(3) for each $k_0 \cdots k_m$ there exists a $j_0 \cdots j_m$ such that

$$M\begin{pmatrix} j_0 & \cdots & j_m \\ k_0 & \cdots & k_m \end{pmatrix} \neq 0.$$

The identity matrix is clearly a D-matrix. Additional examples of D-matrices are given in Part II [9] and in [11].

Now let $T = (t_0, \ldots, t_n)$, and define

$$B(T) = \begin{vmatrix} B_0(t_0) & \cdots & B_n(t_0) \\ \vdots & & \vdots \\ B_0(t_n) & \cdots & B_n(t_n) \end{vmatrix}$$

$$B\begin{pmatrix} t_{j_0} & \cdots & t_{j_m} \\ k_0 & \cdots & k_m \end{pmatrix} = B(T)\begin{pmatrix} j_0 & \cdots & j_m \\ k_0 & \cdots & k_m \end{pmatrix}$$

A collection of functions $B(t) = (B_0(t), \ldots, B_n(t))$ is called a Descartes system, or simply a D-system, on the interval $[a, b]$ iff for each m the determinants $B\binom{t_{j_0} \cdots t_{j_m}}{k_0 \cdots k_m}$ are all nonzero and of one strict sign s_m ± 1 whenever $a \leq t_0 < t_1 < \cdots < t_n \leq b$.

THEOREM 1 (GANTMACHER–KREIN). *A set of functions $B(t)$ satisfy Descartes' Law of Signs iff $B(t)$ is a D-system.*

PROOF. See [13]. □

THEOREM 2 (CAUCHY–BINET). *Let $M = M_1 * M_2$. Then*

$$M\begin{pmatrix} j_0 & \cdots & j_m \\ k_0 & \cdots & k_m \end{pmatrix} = \sum_{i_0<i_1<\cdots<i_m} M_1\begin{pmatrix} j_0 & \cdots & j_m \\ i_0 & \cdots & i_m \end{pmatrix} M_2\begin{pmatrix} i_0 & \cdots & i_m \\ k_0 & \cdots & k_m \end{pmatrix}.$$

PROOF. See [12]. □

COROLLARY 1. *Let $M = M_1 * M_2$. Then*

$$M_1 \text{ and } M_2 \text{ are D-matrices} \Rightarrow M \text{ is a D-matrix.}$$

COROLLARY 2. *Let $D(t) = B(t) * M$. Then*

$$B(t) \text{ is a D-system and } M \text{ is a D-matrix} \Rightarrow D(t) \text{ is a D-system.}$$

Now suppose that the distribution $B(t)$ satisfies Descartes' Law of Signs. Then, by Theorem 1, $B(t)$ is a D-system. If we want the distribution $D(t) = B(t) * M$ also to satisfy Descartes' Law of Signs, then $D(t)$ must also be a D-system. By Corollary 2 we can guarantee that $D(t)$ is a D-system if M is a D-matrix. Thus we conclude that

$$M \text{ is a D-matrix} \Rightarrow D(t) \text{ satisfies Descartes' Law of Signs.}$$

Notice, however, that the converse is not necessarily valid (see Part II [9], Example 6).

We summarize our results in Table II. Notice again that the conditions on the transformation matrix listed in this table are not all independent. Indeed

$$\text{Conditions 1, 2, 5} \Rightarrow \text{Condition 3,}$$
$$\text{Condition 6} \Rightarrow \text{Condition 7.}$$

The second implication follows directly from the definition of a Descartes matrix. The first implication follows because if

$$\sum_k M_{jk} = 1, \qquad \sum_k kM_{jk} = j,$$

then multiplying the first equation by j and subtracting the result from the second equation, we obtain

$$\sum_k (k - j)M_{jk} = 0.$$

Now if $j = 0, n$, then by Condition (2) the terms in this summation are either all nonnegative or all nonpositive. Therefore for $j = 0, n$ the only way to satisfy this equation and Condition (1) is for

$$M_{0k} = \begin{cases} 0, & k \neq 0, \\ 1, & k = 0, \end{cases}$$

$$M_{nk} = \begin{cases} 0, & k \neq n, \\ 1, & k = n. \end{cases}$$

Table II

	Blending functions = distribution		Transformation matrix = Markov chain
1.	$\sum_k D_k(t) = 1$	$\Leftrightarrow$	$\sum_k M_{jk} = 1$
2.	$D_k(t) \geq 0$	$\Leftarrow$	$M_{jk} \geq 0$
3.	$D_k(0) = \begin{cases} 0, & k \neq 0 \\ 1, & k = 0 \end{cases}$		$M_{0k} = \begin{cases} 0, & k \neq 0 \\ 1, & k = 0 \end{cases}$
		$\Leftrightarrow$	
	$D_k(1) = \begin{cases} 0, & k \neq n \\ 1, & k = n \end{cases}$		$M_{nk} = \begin{cases} 0, & k \neq n \\ 1, & k = n \end{cases}$
4.	$D_k(t) = D_{n-k}(1-t)$	$\Leftrightarrow$	$M_{n-j,n-k} = M_{jk}$
5.	$\sum_k kD_k(t) = nt$	$\Leftrightarrow$	$\sum_k kM_{jk} = j$
6.	$D(t)$ satisfies Descartes' law of signs	$\Leftarrow$	M is a Descartes matrix
7.	$D_0(t), \ldots, D_n(t)$ are linearly independent	$\Leftrightarrow$	$\text{Det}(M) \neq 0$

Let us now change our perspective to that of Problem 1 of Section 2; that is, we want to alter the curve $B[P](t)$ by changing the control points P. We already know that the same Markov chain M that solves Problem 2 also solves this problem. Indeed, we can define new control points Q simply by setting

$$Q_j = \sum_k M_{jk} P_k, \qquad {}^tQ = M * {}^tP.$$

We now ask: How do the additional constraints on M, which are summarized in Table II, affect the new control points Q?

We have already seen that

$$\sum_k M_{jk} = 1 \quad \Leftrightarrow \quad Q \text{ is well defined,}$$

and

$$M_{jk} \geq 0 \quad \Leftrightarrow \quad Q \subseteq \text{convex hull } (P).$$

In addition, it is obvious that

$$M_{0k} = \begin{cases} 0, & k \neq 0 \\ 1, & k = 0 \end{cases} \quad \Leftrightarrow \quad Q_0 = P_0,$$

$$M_{nk} = \begin{cases} 0, & k \neq n \\ 1, & k = n \end{cases} \quad \Leftrightarrow \quad Q_n = P_n.$$

Thus this condition implies that P and Q share the same end points.

The next condition,

$$M_{n-j,n-k} = M_{jk},$$

has a more subtle affect. Let

$$P'_k = P_{n-k},$$
$$P' = (P'_0, \ldots, P'_n) = (P_n, \ldots, P_0).$$

The corresponding new control points are defined by setting

$$^tQ = M * {}^tP, \qquad {}^tQ' = M * {}^tP'.$$

Therefore

$$\begin{aligned} Q_{n-j} &= \sum M_{n-j,k} P_k, \\ Q'_j &= \sum M_{jk} P'_k = \sum M_{jk} P_{n-k} \\ &= \sum M_{j,n-k} P_k. \end{aligned}$$

Hence

$$M_{n-j,n-k} = M_{jk} \qquad \Leftrightarrow \qquad Q'_j = Q_{n-j}.$$

Thus this condition implies that reversing the order of the old control points P also reverses the order of the new control points Q.

Now consider the condition

$$\sum_k k M_{jk} = j.$$

Let P consist of a collection of points that are equally spaced along a straight line $L(t)$. Then

$$L(t) = tA + B, \qquad P_k = \frac{k}{n} A + B.$$

Therefore

$$\begin{aligned} Q_j &= \sum_k M_{jk} P_k = \sum_k M_{jk} \left(\frac{k}{n} A + B \right) \\ &= \left(\sum_k k M_{jk} \right) \frac{A}{n} + B. \end{aligned}$$

Hence

$$\sum_k k M_{jk} = j \qquad \Leftrightarrow \qquad Q_j = P_j.$$

Thus this condition implies that if the old control points are equally spaced along a straight line, then the new control points are identical to the old control points. Another way to say this is that the transformation M exactly reproduces straight lines. Analogously, notice that the first condition,

$$\sum_k M_{jk} = 1,$$

implies that the transformation M exactly reproduces individual points. For suppose that for all k, $P_k = P_0$; then

$$Q_j = \sum_k M_{jk} P_k = \left(\sum_k M_{jk} \right) P_0.$$

Hence

$$\sum_k M_{jk} = 1 \qquad \Leftrightarrow \qquad Q_j = P_0.$$

Transformations that reproduce both individual points and straight lines are fundamental in computer-aided geometric design.

Next suppose that M is a Descartes matrix. To see what effect this condition has on the control points Q, we first need some preliminary results.

PROPOSITION 1. *Let $M = (M_{jk})$ be a matrix, and let $C = (c_0, \ldots, c_n)$ be a set of constants. Define new constants $D = (d_0, \ldots, d_n)$ by setting*

$$d_j = \sum_k M_{jk} c_k, \qquad {}^tD = M * {}^tC.$$

If M is a D-matrix, then

$$\textit{sign alternations } D \leq \textit{sign alternations } C.$$

PROOF. The proof of this result is very similar to the proof of the theorem of Gantmacher–Krein given in [13]. Let p be the number of sign alternations in C. Then we can partition C into $p + 1$ subsets $(c_0, \ldots, c_{k_1}), \ldots, (c_{k_p+1}, \ldots, c_n)$ such that each subset has at least one nonzero element and

$$\text{sign } c_m = (-1)^q \qquad \text{if} \quad c_m \text{ is in the } q\text{th subset.}$$

Suppose that the proposition is false. Then there must be $p + 2$ nonzero constants $d_{j_0}, \ldots, d_{j_{p+1}}$ such that sign $d_{j_i} = (-1)^i$. Consider the system of $p + 2$ linear equations with $p + 1$ unknowns

$$(*) \qquad d_{j_i} = \sum_{q=1}^{p+1} \left[\sum_{k=k_{q-1}+1}^{k_q} M_{j_i k} |\, c_k \,| \right] x_q, \qquad i = 0, \ldots, p+1.$$

By construction,

$$\begin{aligned} d_{j_i} &= \sum_k M_{j_i k} c_k \\ &= \sum_{q=1}^{p+1} (-1)^q \left[\sum_{k=k_{q-1}+1}^{k_q} M_{j_i k} |\, c_k \,| \right], \qquad i = 0, \ldots, p+1. \end{aligned}$$

Therefore the linear system $(*)$ has a solution. Indeed, we can choose $x_q = (-1)^q$. Now a system of $p + 2$ linear equations with $p + 1$ unknowns can have a solution if and only if the determinant of the entire system is zero. Expanding this determinant by cofactors of the first column, we get

$$\text{Det} \begin{vmatrix} d_{j_0} & \sum_{k=0}^{k_1} M_{j_0 k} |\, c_k \,| & \cdots & \sum_{k=k_p+1}^{n} M_{j_0 k} |\, c_k \,| \\ \vdots & \vdots & & \vdots \\ d_{j_{p+1}} & \sum_{k=0}^{k_1} M_{j_{p+1} k} |\, c_k \,| & \cdots & \sum_{k=k_p+1}^{n} M_{j_{p+1} k} |\, c_k \,| \end{vmatrix}$$

$$= \sum_{i=0}^{p+1} (-1)^i d_{j_i} \sum_{m_0 \cdots m_p} |\, c_{m_0} \cdots c_{m_p} \,| M \begin{pmatrix} j_0 & \cdots & \hat{j}_i & \cdots & j_{p+1} \\ m_0 & & \cdots & & m_p \end{pmatrix},$$

where $\hat{j}_i$ means that j_i is omitted. But if M is a D-matrix, this expression cannot be zero. In fact, if M is a D-matrix, all the terms in this expression have the same sign, and at least some terms are nonzero. Thus we have reached a contradiction. Therefore it follows that if M is a D-matrix,

$$\text{sign alternations } D \leq \text{sign alternations } C \qquad \text{Q.E.D.}$$

In 3-space we say that a collection of points $Q = (Q_0, \ldots, Q_n)$ oscillates less than a collection of points $P = (P_0, \ldots, P_n)$ if and only if for every plane S the number of times the polygon determined by the vertices Q crosses S is less than or equal to the number of times the polygon determined by the vertices P crosses S. In this case we shall write

$$\text{oscillations } Q \leq \text{oscillations } P.$$

PROPOSITION 2. *Let $M = (M_{jk})$ be a Markov chain, and let $P = (P_0, \ldots, P_n)$ be a collection of points. Define a new set of points $Q = (Q_0, \ldots, Q_n)$ by setting*

$$Q_j = \sum M_{jk} P_k, \qquad {}^tQ = M * {}^tP.$$

If M is a D-matrix, then

$$\textit{oscillations } Q \leq \textit{oscillations } P.$$

PROOF. Let S be a plane, N a vector normal to S, and R a point on S. Then with respect to S,

$$\text{oscillations } P = \text{sign alternations } [(P_0 - R)\cdot N, \ldots, (P_n - R)\cdot N],$$
$$\text{oscillations } Q = \text{sign alternations } [(Q_0 - R)\cdot N, \ldots, (Q_n - R)\cdot N].$$

But since M is a Markov chain

$$(Q_j - R)\cdot N = \left[\sum_k M_{jk} P_k - R\right]\cdot N = \left[\sum_k M_{jk}(P_k - R)\right]\cdot N$$
$$= \sum_k M_{jk}[(P_k - R)\cdot N].$$

Therefore it follows immediately from Proposition 1 that, with respect to S,

$$\text{oscillations } Q \leq \text{oscillations } P.$$

Since this inequality holds for every plane S, the general result is valid. Q.E.D.

Thus we have shown that

$$M \text{ is a D-matrix} \Rightarrow Q \text{ oscillates less than } P.$$

Finally, suppose that $\mathrm{Det}(M) \neq 0$. Let P, P' be two sets of control points. The corresponding new control points Q, Q' are defined by

$${}^tQ = M * {}^tP, \qquad {}^tQ' = M * {}^tP'.$$

Therefore

$$\mathrm{Det}(M) \neq 0 \quad \Leftrightarrow \quad Q' = Q \quad \text{iff} \quad P' = P.$$

We summarize our results in Table III.

To complete this discussion, let us adopt the perspective of Problem 3 of Section 2. Here we have a curve $B[P](t)$—defined by a collection of blending

Table III

Transformation matrix			Control points
1.	$\sum_k M_{jk} = 1$	$\Leftrightarrow$	Q is well defined
2.	$M_{jk} \geq 0$	$\Leftrightarrow$	$Q \subseteq$ convex hull (P)
3.	$M_{0k} = \begin{cases} 0, & k \neq 0 \\ 1, & k = 0 \end{cases}$		$Q_0 = P_0$
		$\Leftrightarrow$	
	$M_{nk} = \begin{cases} 0, & k \neq n \\ 1, & k = n \end{cases}$		$Q_n = P_n$
4.	$M_{n-j,n-k} = M_{jk}$	$\Leftrightarrow$	Reversing P reserves Q
5.	$\sum_k kM_{jk} = j$	$\Leftrightarrow$	Straight lines are reproduced exactly
6.	M is a D-matrix	$\Rightarrow$	Oscillations $Q \leq$ oscillations P
7.	$\text{Det}(M) \neq 0$	$\Leftrightarrow$	$Q' = Q$ iff $P' = P$

functions $B(t) = (B_0(t), \ldots, B_n(t))$ and a set of control points $P = (P_0, \ldots, P_n)$—and a new distribution $D(t) = (D_0(t), \ldots, D_m(t))$, and we seek a new set of control points $Q = (Q_0, \ldots, Q_m)$ such that the curve $D[Q](t)$ is identical to the curve $B[P](t)$. As we have already seen, if $\text{span}[D(t)] \supseteq B(t)$, then there is a matrix $M = (M_{jk})$ such that

$$B_k(t) = \sum_j M_{jk} D_j(t), \qquad B(t) = D(t) * M,$$

and the new control points Q are defined simply by setting

$$Q_j = \sum_k M_{jk} P_k, \qquad {}^tQ = M * {}^tP.$$

Now suppose that the blending functions $B(t)$, $D(t)$ are known to satisfy some property in Table I. What implications does this have for the transformation matrix M and the control points Q?

Clearly this question is very similar to those that we have already answered. Thus all we really need to do to answer this question is to combine Tables II and III. There are, however, two slight differences. In our current problem the number m of D blending functions and control points is not necessarily the same as the number n of B blending functions and control points. Also, in our previous discussion, $D(t) = B(t) * M$, but in our current discussion, $B(t) = D(t) * M$. These differences will affect, in particular, conditions 3–5. However, the analysis of these conditions is still essentially the same, so we need not repeat this analysis again here. Instead we shall simply summarize our results in Table IV. In this table we use the Kronecker delta symbol

$$\delta_{jk} = \begin{cases} 0, & j \neq k, \\ 1, & j = k. \end{cases}$$

4. CONCLUSIONS AND FUTURE WORK

In this paper we have tried to elaborate upon the deep connection between classical probability theory and computer-aided geometric design. The fact that Markov chains are often associated with solutions to some of the standard problems of computer-aided geometric design further reinforces this link.

Table IV

Curve		Blending functions		Transformation matrix		Control points
1. $B[P](t)$ is well defined	$\Leftrightarrow$	$\sum_k B_k(t) = 1$	$\Leftrightarrow$	$\sum_k M_{jk} = 1$	$\Leftrightarrow$	Q is well defined
2. $B[P](t) \subseteq$ convex hull (P)	$\Leftrightarrow$	$B_k(t) \geq 0$	$\Leftarrow$	$M_{jk} \geq 0$	$\Leftrightarrow$	$Q \subseteq$ convex hull (P)
3. $B[P](t)$ passes through P_0, P_n	$\Leftrightarrow$	$B_k(0) = \delta_{k0}$ $B_k(1) = \delta_{kn}$	$\Leftrightarrow$	$M_{0k} = \delta_{k0}$ $M_{mk} = \delta_{kn}$	$\Leftrightarrow$	$Q_0 = P_0$ $Q_m = P_n$
4. $B[P](t)$ is symmetric	$\Leftrightarrow$	$B_k(t) = B_{n-k}(1-t)$	$\Leftrightarrow$	$M_{m-j,n-k} = M_{jk}$	$\Leftrightarrow$	Reversing P reverses Q
5. $B[P](t)$ exactly reproduces straight lines	$\Leftrightarrow$	$\sum_k kB_k(t) = nt$	$\Leftrightarrow$	$\sum_k kM_{jk} = \frac{nj}{m}$	$\Leftrightarrow$	M exactly reproduces straight lines
6. $B[P](t)$ is variation diminishing	$\Leftarrow$	$B_0(t), \ldots, B_n(t)$ satisfy Descartes' Law of Signs	$\Leftarrow$	M is a D-matrix	$\Rightarrow$	Q oscillates less than P
7. $B[P](t) = B[R](t)$ iff $P = R$	$\Leftrightarrow$	$B_0(t), \ldots, B_n(t)$ are linearly independent	$\Leftrightarrow$	$\mathrm{Det}(M) \neq 0$	$\Leftrightarrow$	$Q' = Q$ iff $P' = P$

In Part II of this paper [9], we shall flesh out this theme by introducing specific examples of Markov chains that either satisfy some of the constraints or solve one of the problems analyzed here. Particular applications to subdivision will also be discussed.

REFERENCES

1. BARSKY, B.A., AND DEROSE, T.D. The Beta2-spline: A special case of the Beta-spline curve and surface representation. Tech. Rep. UCB/CSD 83/52, Computer Science Div., Univ. of California, Berkeley, Nov. 1983.
2. CHUNG, K.L. *Elementary Probability Theory with Stochastic Processes.* Springer-Verlag, New York, 1975.
3. COHEN, E., LYCHE, T., AND RIESENFELD, R. Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics. *Comput. Graph. Image Proc. 14* (1980), 87–111.
4. FORREST, A.R. Interactive interpolation and approximation by Bezier polynomials. *Comput. J. 15* (1972), 71–79.
5. GOLDMAN, R.N. An urnful of blending functions. *IEEE Comput. Graph. Appl. 3*, 7 (1983), 49–54.
6. GOLDMAN, R.N. An intuitive approach to Bezier and other random curves and surfaces. Siggraph Tutorial on Freeform Surfaces, 1983.
7. GOLDMAN, R.N. Geometry and probability. Siggraph Tutorial on Freeform Surfaces, 1984.
8. GOLDMAN, R.N. Polya's urn model and computer aided geometric design. *SIAM J. Alg. Discr. Meth. 6*, 1, (Jan. 1985), 1–28.
9. GOLDMAN, R.N. Markov chains and computer-aided geometric design: Part II—Examples and subdivision matrices. *ACM Trans. Graph.* To be published.
10. GORDON, W.J., AND RIESENFELD, R.F. Bernstein–Bezier methods for the computer-aided design of free-form curves and surfaces. *J. ACM 21*, 2 (Apr. 1974), 293–310.
11. KARLIN, S. Total positivity, absorption probabilities and applications. *Trans. Amer. Math. Soc. 3*, 1 (1964), 33–107.
12. KARLIN, S. *Total Positivity*, Vol. 1. Stanford University Press, Stanford, Calif., 1968.
13. KARLIN, S., AND STUDDEN, W. *Tchebycheff Systems: With Applications in Analysis and Statistics.* Interscience Publishers, New York.
14. NIELSON, G.M., RIESENFELD, R.F., AND WEISS, N.A. Iterates of Markov operators. *J. Approx. Theory 17*, 4 (1976), 321–331.

Received October 1982; revised July 1984