Freeform Curves on Spheres of Arbitrary Dimension

S. Schaefer[1], R. Goldman[1]
1. Rice University
E-mail: sschaefe@rice.edu, rng@rice.edu

Abstract

Recursive evaluation procedures based on spherical linear interpolation and stationary subdivision algorithms based on geodesic midpoint averaging are used to construct the analogues on spheres of arbitrary dimension of Lagrange and Hermite interpolation and Bezier and B-spline approximation.

Keywords: curves, spheres, quaternions.

1. Introduction

Spheres in 3-dimensions are common in Computer Graphics, but spheres in 4-dimensions are also important, since the unit sphere in 4-dimensions is the space of unit quaternions and unit quaternions represent rotations in 3-dimensions. The goal of this paper is to generate curves on spheres of arbitrary dimension by extending two standard techniques for generating curves in Euclidean *n*-space: recursive evaluation procedures and stationary subdivision algorithms.

In a landmark paper, Shoemake replaced linear interpolation with spherical linear interpolation in the de Casteljau algorithm in order to construct analogues of Bezier curves in the space of unit quaternions [5]. Here we show that spherical linear interpolation is compatible not only with approximating methods such as the de Casteljau algorithm for Bezier curves, but also with interpolatory techniques such as Neville's algorithm for Lagrange and Hermite interpolation.

Generating analogues for B-splines on *n*-dimensional spheres is not so simple, since replacing linear interpolation with spherical linear interpolation in the de Boor algorithm does not necessarily generate piecewise smooth curves [7]. Stationary subdivision algorithms are procedures for building smooth curves from discrete data by recursively inserting values between the data. To construct smooth curves in Euclidean space, we commonly insert new data by averaging the given data. To extend subdivision algorithms to generate smooth curves on spheres, we shall use geodesics (great circles) to interpret what averaging means on *n*-dimensional spheres. We will then show how to apply subdivision to construct analogues of uniform B-splines on spheres of arbitrary dimension. Other researchers have applied geodesics to extend the theory of subdivision from Euclidean spaces to arbitrary manifolds [1, 4]; our contribution here is to provide simple explicit formulas for subdivision algorithms on spheres of arbitrary dimension. An alternative approach to generating quite different analogues of Bezier and B-spline curves on the space of unit quaternions is presented in [7].

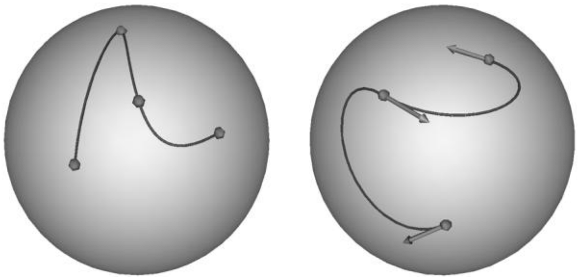

Figure 1. A Lagrange interpolant (left) and a C^1 Hermite curve with two segments (right) on the sphere.

2. Spherical Linear Interpolation (Slerp)

Many well-known schemes for generating freeform curves in *n*-dimensional Euclidean space can be evaluated by recursive procedures based on repeated linear interpolation [2, 6]. Linear interpolation (*lerp*) in Euclidean space takes in two points P, Q and a scaling parameter α and creates a point that splits the line from P to Q into 2 segments whose ratio is $\frac{1-\alpha}{\alpha}$.

$$lerp(P,Q,\alpha) = (1-\alpha)P + \alpha Q$$

Spherical linear interpolation (*slerp*) operates on two points on a unit sphere represented by unit vectors u, v and generates a point along the geodesic connecting u, v that splits the arc into two arcs whose ratio is $\frac{1-\alpha}{\alpha}$.

$$slerp(u,v,\alpha) = \frac{\sin(\phi(1-\alpha))}{\sin(\phi)}u + \frac{\sin(\phi\alpha)}{\sin(\phi)}v$$

where $\phi = \cos^{-1}(u \cdot v)$ [5].

2.1. Lagrange Interpolation

Consider the Lagrange interpolating polynomial

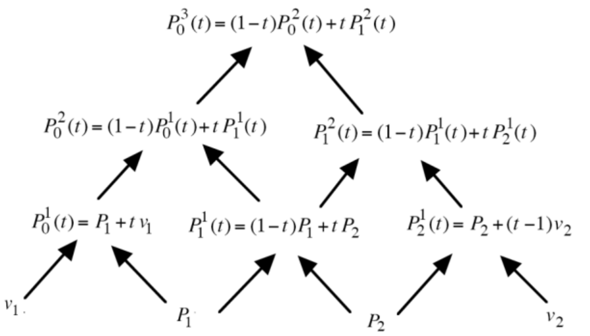

Figure 2. Neville's algorithm for cubic Hermite interpolation in Euclidean space.

$L(t)$ of degree n with nodes $t_0, \ldots, t_n$ and interpolation points $P_0, \ldots, P_n$ -- that is, the unique degree n polynomial curve $L(t)$ such that $L(t_i)=P_i$, $i=0, \ldots, n$. Neville's algorithm for Lagrange interpolation is given by setting:

$$P_j^0(t) = P_j \qquad j = 0,\ldots,n$$

$$P_j^k(t) = lerp\left(P_j^{k-1}(t), P_{j+1}^{k-1}(t), \frac{t-t_j}{t_{j+k}-t_j}\right) \quad j = 0,\ldots,n-k.$$

The function $L(t) = P_0^n(t)$ interpolates the points $P_0, \ldots, P_n$ in Euclidean space at the parameter values $t_0, \ldots, t_n$.

To generate interpolating curves on spheres, simply replace linear interpolation by spherical linear interpolation in the recursive step of the evaluation algorithm. Thus if $u_0, \ldots, u_n$ are a collection of unit vectors representing points on a sphere and $t_0, \ldots, t_n$ are a set of parameter values, the recursive step for Lagrange interpolation on the sphere is

$$su_j^k(t) = slerp\left(su_j^{k-1}(t), su_{j+1}^{k-1}(t), \frac{t-t_j}{t_{j+k}-t_j}\right) \quad j = 0,\ldots,n-k.$$

The function $SL(t) = su_0^n(t)$ is the analogue of a Lagrange interpolating curve on the sphere -- that is, $SL(t_i)=u_i$, $i=0, \ldots, n$ (see Figure 1).

The proof that this spherical version of Neville's algorithm generates an interpolant on the sphere is essentially the same as the inductive proof that the standard version of Neville's algorithm generates an interpolant in Euclidean space [2].

2.2. Hermite Interpolation

Hermite polynomials interpolate derivatives as well as positions at the nodes [2]. The simplest and probably the most important version of Hermite interpolation is cubic Hermite interpolation, where a single cubic polynomial $H(t)$ interpolates two points P_1, P_2 as well as the first derivative vectors v_1, v_2 at each of the points -- that is,

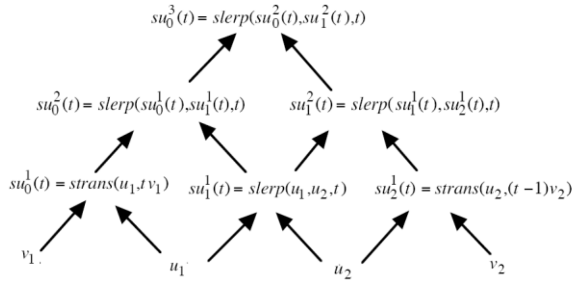

Figure 3. The analogue of Neville's algorithm for cubic Hermite interpolation on the sphere, replacing linear interpolation by spherical linear interpolation and replacing translation by spherical translation.

$$H(0) = P_1 \qquad H(1) = P_2$$
$$H'(0) = v_1 \qquad H'(1) = v_2$$

Given a sequence (P_j, v_j) of points and derivative vectors, cubic Hermite interpolation can be applied to generate a C^1 cubic spline that interpolates all the data.

Neville's algorithm for Lagrange interpolation can be generalized to handle Hermite interpolation [2]. Figure 2 illustrates Neville's algorithm for cubic Hermite interpolation in Euclidean space. Notice that except for the first and last steps on the first level of this algorithm, each step is once again just linear interpolation. To generalize cubic Hermite interpolation to spheres, we shall, as usual, simply replace these linear interpolations by spherical linear interpolations. The expressions $P_0^1(t) = P_1 + t\,v_1$ and $P_2^1(t) = P_2 + t\,v_2$ that appear on the first level of Neville's algorithm for cubic Hermite interpolation represent translations in Euclidean space. Therefore to extend cubic Hermite interpolation to spheres, we shall need to replace these Euclidean translations by translations on spheres.

To perform translation on a sphere, we define an operator *strans* that takes as input a point u on the sphere and a tangent vector v at u, and translate u along the geodesic in the direction of v by the distance $|v|$. Equivalently, *strans* rotates u in the plane determined by u, v by the angle $|v|$. Thus, since v is perpendicular to u,

$$strans(u,v) = \cos(|v|)u + \sin(|v|)v\frac{|u|}{|v|}.$$

We illustrate this algorithm for Hermite interpolation on the sphere in Figure 3. In Figure 1, we show an example of a C^1 spline on the sphere generated by Hermite interpolation. Once again the proof that this algorithm generates a Hermite interpolant on the sphere is much the same as the proof in Euclidean space.

3. Stationary Subdivision Algorithms: Midpoint Averaging on Spheres

The two most famous subdivision algorithms are the

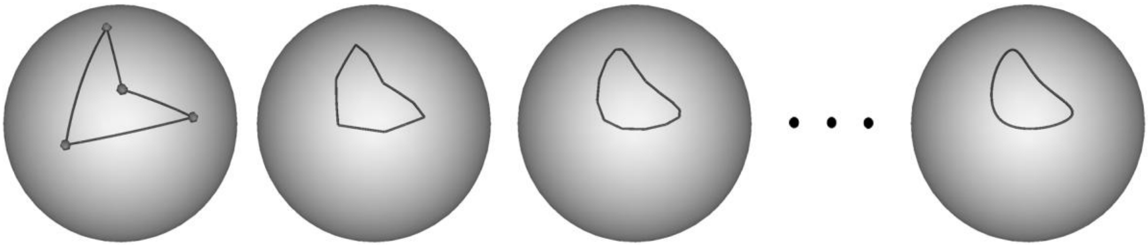

Figure 4. Subdivision for a uniform cubic B-spline on a sphere.

de Casteljau subdivision algorithm for Bezier curves and the Lane-Riesenfeld subdivision algorithm for uniform B-splines [3]. The main step in each of these subdivision procedures is midpoint averaging. For example, in the Lane-Riesenfeld algorithm

$$P^0_{2j} = P^0_{2j+1} = P_j$$

$$P^k_j = \frac{1}{2}\left(P^{k-1}_j + P^{k-1}_{j+1}\right)$$

Therefore to extend subdivision algorithms to spheres, we need to extend the notion of midpoint averaging to spheres of arbitrary dimension.

The midpoint between two points on a sphere is the midpoint of the shortest arc of the great circle joining these two points. (To make the midpoint unique, we shall exclude pairs of antipodal points.) Therefore, we need a simple procedure for finding this midpoint. We could invoke *slerp* and evaluate at s=0.5, but there is an easier way.

A point on the sphere can be represented by a unit vector. Two points on the sphere are represented by two unit vectors, and their midpoint is simply the vector from the origin to the arc through the tips of the two vectors pointing midway between the two vectors. Thus the midpoint between two vectors on the sphere can be represented by the vector that bisects the angle between the two vectors normalized to unit length. Therefore, if v_1, v_2 are two unit vectors representing points on a unit sphere, then their midpoint on the sphere is represented by the unit vector

$$v_m = \frac{(v_1 + v_2)/2}{|(v_1 + v_2)/2|} = \frac{(v_1 + v_2)}{\sqrt{2(1 + v_1 \cdot v_2)}}. \qquad (1)$$

It is straightforward to verify that the right hand side of Equation 1 is identical to the value of *slerp*(v_1, v_2, s) at s=0.5.

Now all we need to generate the analogues of uniform B-splines on spheres is to reinterpret the midpoint averaging rule in the Lane-Riesenfeld algorithm -- that is, we replace the Euclidean midpoint averaging rule

$$P_m = \frac{P_0 + P_1}{2}$$

by the spherical midpoint averaging rule in Equation 1. Notice that there is no need for *slerp* here. We illustrate this subdivision procedure for generating the analogues of B-spline curves on spheres in Figure 4. It is not difficult to prove that the curves generated by this algorithm are indeed smooth; for details, see [1].

Since a circle is a sphere in 2-dimensions, if we start with evenly spaced points on a circle in the plane, then the curve generated by this spherical subdivision algorithm is a circle with a uniform parameterization.

Finally, since midpoint averaging work for spheres of arbitrary dimension, we can use subdivision to generate analogues of uniform B-splines in the space of unit quaternions. This technique may be useful for performing key frame animation in 3-dimensions.

4. References

[1] N. Dyn, Three families of nonlinear subdivision schemes, *Private communication*, 2005.

[2] R. Goldman, *Pyramid Algorithms: A Dynamic Programming Approach to Curves and Surfaces for Geometry Modeling*, Morgan Kaufmann, 2002.

[3] J. Lane and R. Riesenfeld, A theoretical development for the computer generation and display of piecewise polynomial surfaces, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2, 1980, pp. 35–46.

[4] L. Noaks, Nonlinear corner cutting, *Advances in Computational Mathematics* 8, 1998, pp. 165–177.

[5] K. Shoemake, Animating rotation with quaternion curves, *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer Graphics and interactive techniques*, ACM Press, 1985, pp. 245–254.

[6] K. Shoemake, Linear form curves, *Graphics Gems V*. AP Professional, 1995.

[7] M-J. Kim, M-S Kim, and S. Y. Shin, A general construction scheme for unit quaternion curves with simple high order derivatives; *Proceedings of SIGGRAPH '95*, pp. 369–37