Spherical Averages and Applications to Spherical Splines and Interpolation

Samuel R. Buss*
Department of Mathematics
University of California, San Diego

Jay P. Fillmore
Department of Mathematics
University of California, San Diego

June 11, 2001

Abstract

This paper introduces a method for computing weighted averages on spheres based on least squares minimization that respects spherical distance. We prove existence and uniqueness properties of the weighted averages, and give fast iterative algorithms with linear and quadratic convergence rates. Our methods are appropriate to problems involving averages of spherical data in meteorological, geophysical and astronomical applications. One simple application is a method for smooth averaging of quaternions, which generalizes Shoemake's spherical linear interpolation.

The weighted averages methods allow a novel method of defining Bézier and spline curves on spheres, which provides direct generalization of Bézier and B-spline curves to spherical spline curves. We present a fast algorithm for spline interpolation on spheres. Our spherical splines allow the use of arbitrary knot positions; potential applications of spherical splines include smooth quaternion curves for applications in graphics, animation, robotics and motion planning.

Keywords: *spherical mean, spherical average, least squares minimization, barycentric coordinates, quaternions, quaternion interpolation, Bézier curve, B-spline, spline curve, spline interpolation, spherical interpolation.*

1 Introduction

Let S^d be the d-dimensional unit sphere in $\mathbb{R}^{d+1}$. For points x and y on S^d, let $\mathrm{dist}_S(x, y)$ denote the length of the shortest geodesic on S^d from x to y.

*Supported in part by NSF grant DMS-9503247. Email: `sbuss@ucsd.edu`

Let $||x-y||$ denote the usual Euclidean distance between x and y in $\mathbb{R}^{d+1}$; note $||x-y|| < \mathrm{dist}_S(x,y)$ for any distinct points x and y on S^d.

Let $p_1,\ldots,p_n$ be points on S^d: the main purpose of the present paper is to define a weighted average or centroid of these n points using as weights values $w_1,\ldots,w_n$ such that each $w_i \geq 0$ and and such that $\sum_i w_i = 1$. This weighted average will be denoted

$$C \;=\; \sum_{i=1}^{n} w_i \cdot p_i,$$

and is to be defined in terms of distances in S^d, using $\mathrm{dist}_S(-,-)$ instead of Euclidean distance. Since S^d is not a linear space, it is not immediately obvious that there is a sensible way to define this weighted average; however, in section 2 below, we shall show that it makes sense to define the weighted average as the result of a least squares minimization, namely as the point C on S^d which minimizes the value

$$f(C) \;=\; \tfrac{1}{2}\sum_i w_i \cdot \mathrm{dist}_S(C,p_i)^2.$$

This method of calculating the centroid C will be shown to enjoy a number of nice properties: firstly, it is a natural analogue of weighted averages in Euclidean space; secondly, it generalizes Shoemake's spherical linear interpolation, which provides a definition of a spherical average of two points; and thirdly, the centroid value C exists and is unique in many common situations. In section 3, we describe efficient methods of calculating the centroid C using iterative methods with linear and quadratic convergence rates.

To the best of our knowledge, this kind of weighted average or centroid on spheres has not been considered in the past. Indeed, there is an extensive literature on averaging points on spheres and on statistical analysis of points that lie on spheres, e.g., see [34, 35, 1] or the book [37] and the references cited therein. Most of that work performs the averages in $\mathbb{R}^{d+1}$ and renormalizes to place the result on the sphere; namely, they compute the centroid as

$$\frac{\sum_i w_i p_i}{||\sum_i w_i p_i||}$$

with the summation being taken inside $\mathbb{R}^{d+1}$. Shoemake [30] noted that even when taking weighted averages of only two points, this kind of renormalized $\mathbb{R}^{d+1}$-average causes undesirable effects. Shoemake used the quaternion representation of rigid-body orientations as points on S^3 to perform smooth interpolation between two orientations of a rigid body in 3-space. He pointed out that the renormalized $\mathbb{R}^{d+1}$-averaging method did not produce uniform interpolation of orientations and was therefore unsuitable for many applications: his solution was to use "spherical linear interpolation" or "slerp"-ing to compute the weighted average of only two points on the sphere based on geodesic length in S^3. Our spherical averaging method provides a smooth generalization of Shoemake's spherical linear interpolation to allow computing weighted averages of more than two points.

A few authors consider statistics of spherical data that do not merely embed the results in $\mathbb{R}^{d+1}$: for instance Clark and Thompson [6] perform averages using a stereographic projection (the axis for the stereographic projection is calculated by averaging in $\mathbb{R}^{d+1}$) — this suffers from distorsion for data points located away from the center of projection. Later, in [33], they used averages based on longitude and latitude values: this causes distorsion for data located away from the equator.

A number of authors, including [3, 14, 39], have considered averages in fairly general metric space and Banach space settings. These settings are too general to be applicable to averages on the sphere S^d.

Spherical splines. The second main topic of the present paper is the the application of spherical averaging to generate spline curves on spheres. Spherical spline curves have potential applications in computer graphics, in animation and in robotics and motion planning based on quaternions.

In the Euclidean case, a spline consists of control points $p_1, \ldots, p_n$ (in $\mathbb{R}^{d+1}$, say) and of blending functions $f_1(t), \ldots, f_n(t)$ (also called "basis functions") defined for t in the domain $[a, b]$ for some $a < b$. The functions f_i have the property that, for all $t \in [a, b]$,

$$f_1(t) + f_2(t) + \cdots + f_n(t) \;=\; 1, \qquad \text{and} \qquad f_i(t) \geq 0, \text{ for all } i. \tag{1}$$

In the most common applications, the functions $f_i(t)$ are defined by specifying knot positions $t_1, \ldots, t_{k+d}$ and constructing the blending functions $f_i(t)$ from the knot positions. This provides a framework for defining blending functions which give curves with desired smoothness properties, and even sharp bends or cusps where desired (see e.g., [10]). Once the blending functions have been defined, the spline curve is defined by

$$s(t) \;=\; f_1(t)p_1 + f_2(t)p_2 + \cdots + f_n(t)p_n.$$

This gives a curve parameterized by t.

Generalizing this approach to spline curves to the d-sphere S^d becomes straightforward after the development of spherical averages in sections

2 and 3 below. Namely, let the control points $p_1, \ldots, p_n$ lie on S^d. Given blending functions $f_i(t)$ which satisfy (1), a spline curve on the sphere can be defined by using the spherical weighted average:

$$s(t) \;=\; \textstyle\sum\!\!\!\!\bigcirc_{i=1}^{n} f_i(t) \cdot p_i.$$

In this way, many of the techniques of defining B-splines in Euclidean space are generalized to the sphere. As a special case, this allows the definition of spherical analogues of piecewise cubic B-spline curves.

Spherical spline interpolation. The control points used to define a spline curve typically do not lie on the resulting curve. However, one often wishes to use spline curves to interpolate points, i.e., one has points $q_1, \ldots, q_n$ and has n distinct time values $t_1, \ldots, t_n$ and one wishes to ensure that $s(t_i) = q_i$ for all $1 \le i \le n$. There are a variety of well-known approaches for picking blending functions f_i so as to ensure the spline has the desired smoothness properties and to control the behavior of the spline curve at its endpoints. Given the interpolation points q_i and the time values t_i and the blending functions f_i, one then needs to find control points $p_1, \ldots, p_n$ which cause the curve to interpolate the points q_i.

For curves on the sphere, this means choosing points p_i on S^d so that

$$q_j \;=\; \sum\!\!\!\!\!\bigcirc_{i=1}^{n} f_i(t_j) \cdot p_i \qquad \text{for } 1 \le j \le n.$$

In most common applications (where one generates spline curves which are piecewise degree 3 polynomials), the values $f_i(t_j)$ are non-zero only for $j \in \{i-1, i, i+1\}$. Therefore in the Euclidean case, the control points p_i can be found very simply by solving a tridiagonal matrix equation. The matrix cannot be so simply inverted for spherical spline interpolation; instead we use an iterative procedure to determine $p_1, \ldots, p_n$. Two such procedures are described in section 4.2, and experimental results show them to be quite fast in practice. Indeed, in realistic small examples, the p_i's can be found in a few milliseconds or less — section 5 lists a variety of experimental measurements of runtime.

Prior work. There has been extensive prior work on spherical splines; we survey here a major portion of it, with an eye to how our methods improve on the prior methods.

The earliest description of spherical spline interpolation was by Parker-Denham [25], who described spherical splines on S^d by defining a spline curve in $\mathbb{R}^{d+1}$ and normalizing to map the curve to the sphere. Since the curve may deviate a long way from the sphere, the normalization process is mathematically unnatural and can cause significant distorsion.

Thompson and Clark [33] (see also the references cited therein) applied spherical spline curves to the problem of determining the Gondwanan apparent polar wonder path to study the past movements of continents. They defined spherical spline curves following a method of Gould by interpolating in latitude and longitudinal space. This method causes distorsion near the poles. Fisher-Lewis [11] proposed interpolating points on S^d with a smooth curve by combining smooth segments so as to ensure continuous second derivatives and suggested using either third-order natural splines or loxodromes.

Spherical splines on the 3-sphere S^3, the unit sphere in $\mathbb{R}^4$, have become quite important in computer graphics and animation because of the correspondence between orientations of solid bodies and quaternions (see [7, 30]). Quaternions can be viewed as (pairs of antipodal) points on S^3 and thus spherical spline curves can be used to specify a smooth transition of solid orientations. Shoemake [30, 31] was the first apply quaternions to graphics and animation: he stressed the importance of using spherical distance instead of Euclidean distance. In [30] he introduced the now widely used method of spherical linear interpolation ("slerping") for two points. He also used analogues of the de Casteljau method of calculating Bézier curves to define curves on the sphere, and in [31] introduced a similar, but faster method called *Squad*. A number of other authors have proposed similar methods, also based on Catmull-Rom splines: these methods allow a general spline to be defined in terms of multiple linear interpolations between pairs of points and in this way slerping can be used to define higher-order spherical splines. Duff [9] gives the most in-depth development of this kind of method and proves a variation diminishing property for these splines. A disadvantage to these methods on the sphere, at least as described in the extant literature, is that they work well only for equally spaced knot positions. However, it should be possible to give more sophisticated spherical spline curves based on the de Castaljau method which are computed using multiple slerps between pairs of points and which work well for arbitrary knot positions (indeed, knot insertion methods for spline curves should suffice for this, c.f. [10]). Moreover, these methods should give curves satisfying the properties a.-f. below. These kinds of algorithms, however, would suffer from the fact that resulting curve would depend on the details of the order in which slerps are performed: for instance, the results of repeated knot insertion would depend on the order in which the knot were inserted. By contrast, spline curves based on our spherical averages provide a mathematically natural way of blending multiple control points in a single step without having to arbitrarily choose an order for averaging.

Wang-Joe [36] suggest forming smooth spherical curves which are piecewise small circles. A more sophisticated approach was taken by Roberts-Bishop-Ganapathy [29] and Kim-Nam [20, 21] who proposed defining interpolating splines using spline segments which are obtained by blending small circles on the sphere. This method provides nice smoothness properties, but performs poorly when data points are not equally spaced, and does not allow specification of knot positions.

Ge-Ravani [13] and Jüttler [17] suggest the use of dual quaternion curves to control solid body orientation and position simultaneously. The subsequent paper of Jüttler-Wagner [18] uses piecewise rational functions for the entries of orientation matrices; this method suffers from having control points which are only affine (non-orthogonal) matrices, More recently, Alfeld et al. [1] discuss defining barycentric coordinates and spherical spline functions with good continuity properties using renormalized-$\mathbb{R}^d$ methods. In all four of these papers, the interpolation methods are not intrinsic to the sphere in the sense of respecting spherical distance.

Kim-Kim-Shin [19] give a method of constructing quaternions curves by using general blending functions. Like our method introduced in this paper, they allow the use of arbitrary blending functions and thus arbitrarily spaced knots and control points. Their use of blending functions is mathematically quite different from ours: instead of using a general method of computing weighted averages on the sphere, they compose a series of rotations based on slerps between the control points. This is mathematically somewhat unnatural; for instance, reversing the order of the control points can alter the spline curve. Another undesirable feature is that their spline curves may not satisfy the "convex hull" property; for instance it is possible for the control points to lie on vertices of a triangle in a hemisphere and for the spline curve to lie partly outside the triangle. In addition they do not address the problem of interpolation. Nonetheless, their spline curves enjoy many of the advantages we list below for our spherical splines, particularly if the control points are not very widely spaced and if the blending functions have sufficiently local support.

Gabriel-Kajiya [12], Jupp-Kent [16], Noakes-Heinzinger-Paden [23], Park-Ravani [24], and Dam-Koch-Lillhom [8] proposed using natural splines which minimize covariant (tangential) acceleration. The first paper [12] gave a very general method which works on any manifold as well as described the specialization of their method to the sphere. Curves obtained by minimizing covariant acceleration have nice smoothness, but suffer from not having local control, since changing one control point changes the whole curve. In addition, they are computationally somewhat difficult; e.g., [12] report that it takes a "few tens of seconds" to find an interpolating spline curve on an IBM 4341. Barr et al.[4] proposed minimizing covariant accelerations while relaxing the requirement that the curve lie exactly on the sphere, because this allowed faster computation. They noted that a curve could be computed in "a few minutes per interpolation"; although later [28] characterized the run time of [4] as "several minutes to hours." Ramamoorthi-Barr [28] describe a numerically fast algorithm that can get close to an optimal curve within about 4 seconds (machine type and programming language unspecified). This run time is quite adequate for interactive applications, but unlike our algorithms is not fast enough for real-time applications. Like other natural spline methods, the spline curves of [4, 28] do not allow local control or (without some kind of modification of their energy functions) specification of knot positions. Their curves lie close to, but not on, the 3-sphere and have to be normalized to lie on the sphere. The papers [40, 24] suggest using curves that minimize covariant acceleration and/or jerk. [40] give closed form solutions for special cases of the two point interpolation problems, e.g., when the initial and final velocity are both zero; [24] give approximations that work well in restricted cases, e.g., when the initial and final rotation are not too different, and give applications to interpolation with knot positions.

The definition of spherical splines and spherical spline interpolation in the present paper provides several advantages:

a. It allows construction of spherical splines with non-interpolated control points, and also allow interpolating curves by suitably choosing control points.

b. They can be computed fast enough for some real-time applications: see section 5 for details.

c. The spherical averages and spherical splines are invariant under rotations of the sphere; there is no distorsion near the poles for instance.

d. The algorithms are completely intrinsic to the sphere: we do not compute averages in Euclidean space and then renormalize back to the sphere. Spherical distances, not Euclidean distances, are the basis for averaging and blending. The spherical weighted averages are mathematically natural. If the points all lie on a geodesic (great circle), then our algorithms correspond to weighted averages on a line, but using geodesic arc length.

e. The spline curve lies in the convex hull of the control points.

f. Since our spline curves are defined in terms of weighted averages of spherical data, the well-known methods for generating spline curves in Euclidean space immediately generalize to the sphere. This immediately gives several advantages which carry over from the Euclidean case. First, the well-known strategies for choosing knot positions can now be applied to spherical splines to control the smoothness of the curve. It is therefore easy to define spherical splines with any number of continuous derivatives. Second, knot positions need not be equally spaced or distinct. This allows defining spline curves with sharp localized bends and even discontinuous derivatives if desired. Third, spherical splines can be defined with local control; i.e., changing one control point can affect the curve only in segments close to the control point.

The outline of the rest of the paper is as follows. Section 2 gives the mathematical definition of spherical weighted averages and includes proofs of existence and uniqueness. Section 2 also establishes smoothness of the spherical weighted average as a function of the weights and the points p_i, and proves a convexity property for spherical weighted averages. Only the first part of section 2 is needed for the rest of the paper: the reader is who is interested more in implementation than in mathematical proofs, may skip sections 2.1 through 2.3. Section 3 describes two algorithms for computing spherical weighted averages. Section 4 describes applications to spherical spline curves and to spherical spline interpolation. Section 5 describes the execution speed of our algorithms and compares them to other approaches. We conclude with some open problems in section 6.

2 Spherical Weighted Averages

We consider the following problem: given n points $p_1, \ldots, p_n$ lying on the sphere and n non-negative real weights $w_1, \ldots, w_n$ with sum equal to 1, and we wish to define a weighted average function

$$Avg(w_1, p_1; w_2, p_2; \ldots w_n, p_n) \;=\; \sum\nolimits_i w_i \cdot p_i \tag{2}$$

which computes a point on the sphere which is a "weighted average" of these given n points. This kind of problem is important for a number of applications including statistics of spherical valued functions (as might arise in astronomical, geophysical, meteorological applications), and applications to solid body orientations (quaternions), etc. There are a number of natural properties the *Avg* function should satisfy: (1) the averaging should be intrinsic to the sphere, and depend on spherical distances, not Euclidean distances, between points on the sphere, (2) in the degenerate case where the points p_i lie on a geodesic in a hemisphere, the weighted average should be computed in the usual fashion based on distances along the geodesic, (3) the value of *Avg* should be a C^∞-continuous function of the weights w_i and the points p_i, at least in the case when all the points are contained a hemisphere of the sphere. When averaging quaternions, condition (2) states that the average of two points on the 3-sphere should corresponds to Shoemake's slerping algorithm. A final condition is that the *Avg* function should act qualitatively like a weighted average.

It is not possible to directly define the *Avg* as a linear combination of the points $p_1, \ldots, p_n$ and respect geodesic distance on the sphere. However, there is an alternative definition of Euclidean weighted sums that can be adapted to the sphere. Letting $||p - q||$ denote the Euclidean distance between points p and q, the Euclidean weighted average

$$w_1p_1 + w_2p_2 + \cdots + w_np_n$$

can be defined as the point q which minimizes the value of the function

$$f(q) \;=\; \tfrac{1}{2}\sum\nolimits_{i=1}^{n} w_i||q - p_i||^2.$$

In other words, in Euclidean space, the weighted average of the points p_i is equal to the point where a weighted sum of squares of distances is minimized. To prove this fact, let the Euclidean space be $\mathbb{R}^d$, and each p_i be in $\mathbb{R}^d$ and q range over points in $\mathbb{R}^d$: let the k-th component of q and of p_i be denoted q_k and $p_{i,k}$, respectively ($1 \le k \le d$). By the Pythagorean theorem,

$$f(q) \;=\; \tfrac{1}{2}\sum\nolimits_{i,k} w_i(q_k - p_{i,k})^2.$$

Differentiating with respect to q_k, any critical point of f must satisfy

$$\sum\nolimits_{i=1}^{n} w_i(q_k - p_{i,k}) = 0,$$

whence, using the fact that the weights sum to 1, there is a unique critical point and it is given by

$$q_k \;=\; \sum\nolimits_{i=1}^{n} w_ip_{i,k}.$$

It is clear that this critical point is a global minimum and of course it is the weighted average of the points p_i.

To transfer the least squares minimization to the sphere, we let $\mathrm{dist}_S(p,q)$ denote the spherical distance from points q and p on the d-sphere, namely, the length of the shortest geodesic seqment from q to p. Now let the function f be defined on the d-sphere by

$$f(q) \;=\; \tfrac{1}{2}\sum\nolimits_{i=1}^{n} w_i \cdot \mathrm{dist}_S(q,p_i)^2. \tag{3}$$

We thus define the value of $Avg(w_1,p_1;\ldots w_n,p_n)$ to be the point q on S^d which minimizes $f(q)$. Since the notation Avg is fairly inconvenient and non-intuitive, we henceforth use in its place the modified summation notation

$$\sum\!\!\!\!\!\circ\nolimits_{i=1}^{n} w_i \cdot p_i$$

where the new summation symbol means that it is not a traditional sum, but instead represents the value minimizing equation (3). Sometimes, we use a superimposed '+' and small circle to represent the same quantity, namely, we sometimes denote $\sum\!\!\!\!\!\circ_{i_1}^{n} w_i \cdot p_i$ by

$$w_1 \cdot p_1 \oplus w_2 \cdot p_2 \oplus \cdots \oplus w_n \cdot p_n$$

This modified summation and addition notation is suggestive, but the reader should be aware that many common properties of ordinary summations are not shared by spherical weighted averages; for instance, it is usually the case that the values $\frac{2}{3}\cdot(\frac{1}{2}\cdot p_1 \oplus \frac{1}{2}\cdot p_2) \oplus \frac{1}{3}\cdot p_3$ and $\frac{1}{3}\cdot p_1 \oplus \frac{2}{3}\cdot(\frac{1}{2}\cdot p_2 \oplus \frac{1}{2}\cdot p_3)$ are distinct. This failure of associativity is an example of how our weighted average Avg differs from using progressive slerping, that is to say, the Avg value cannot be computed by taking successive spherical averages of pairs of points. Indeed, by Brown and Worsey [5], there is no way to define spherical averages which is intrinsic to the sphere so that associativity holds.

Fix points p_i on the d-sphere S^d and non-negative weights w_i with sum equal to 1 and let f be the weighted sum of squares of spherical distances (3). By the compactness of the sphere, the function f has a minimum value: in order for the weighted average $\sum\!\!\!\!\!\circ_i w_i \cdot p_i$ to be well-defined, the function f must have attain its minimum value at a unique point on S^d. Of course this condition may not always hold; for instance if the points p_1,p_2,p_3 are equally spaced around the equator of S^2 and if $w_1=w_2=w_3=1/3$, then f is minimized at both the north and south poles. As another example, if there are six equally weighted points on S^2 distributed at the intersections of S^2 with the x,y,z-axes, then f attains its minimum value at the eight points $(\pm 1/\sqrt{3}, \pm 1/\sqrt{3}, \pm 1/\sqrt{3})$.

However, in many situations the function f does attain a unique minimum: the easiest case, treated in the next theorem, is when the points p_i all lie in a common hemisphere of S^d (by convention, a hemisphere is always closed). By a 'critical point', we mean a point where the first derivatives of f are all zero.

Theorem 1 *Suppose the points $p_1, \ldots, p_n$ all lie in a hemisphere $\mathcal{H}$ of S^d, with at least one point p_i in the interior of $\mathcal{H}$ with $w_i \neq 0$. Then the function f has a single critical point q in $\mathcal{H}$, and this point q is the global minimum of f.*

It will be clear from the proof that the assumption that the points all lie in a single hemisphere can be relaxed significantly. Theorem 5 states examples of other conditions that imply the uniqueness of a global minimum for f.

Before we prove Theorem 1, we introduce the exponential and logarithmic functions, and derive formulas for the partial derivatives of f: these will also be useful for the development of algorithms for spherical weighted averages in section 3. The partial derivatives of f are with respect to displacements in the sphere around a point $q \in S^d$. To make this formal, we define T_q to be the hyperplane tangent to S^d at the point q. The hyperplane T_q is d-dimensional and can be coordinatized by letting the point q be the origin of T_q and letting the axes of T_q be arbitrary orthonormal axes which are tangent to the sphere. We define the *exponential map at q* to be the mapping from the tangent hyperplane T_q to the sphere which preserves distances and angles from q.* To make this precise, we need to choose an appropriate set of coordinates for T_q and $\mathbb{R}^{d+1}$. Let points in the space T_q be specified using coordinates $x_1, \ldots, x_d$ (so that the point q is the origin). Let $x'_1, \ldots, x'_{d+1}$ be the coordinate axes for $\mathbb{R}^{d+1}$. Without loss of generality (choose new axes for $\mathbb{R}^{d+1}$ if necessary), the point q is at the point $(0, \ldots, 0, 1)$ in $\mathbb{R}^{d+1}$ and the axes x'_i are parallel to x_i for all $i \leq d$. The exponential map is denoted $exp_q(\cdot)$ and is a function mapping a point p with coordinates $(x_1, \ldots, x_d)$ to the point $exp_q(p) = (x'_1, \ldots, x'_{d+1})$ where

$$x'_i \; = \; x_i \cdot \frac{\sin r}{r}$$

for $1 \leq i \leq d$, where $r = \sqrt{\sum x_i^2}$ is the distance from q to p, and $x'_{d+1} = \cos r$. We let $(\sin 0)/0$ equal 1, and thus $exp_q(q) = q$. Furthermore, since

*The terminology "exponential map" comes from Lie theory; this paper does not depend on knowledge of Lie theory however. The exponential map has been discussed in the setting of computer graphics by Hart et al. [15] and Dam et al. [8].

the sphere has unit radius, it is clear that r is equal to the spherical distance from q to $exp_q(p)$ provided $r \leq \pi$. Thus the exponential map takes points in the tangent plane to points on the sphere, preserving distance from q; it also preserves the tangential direction from q. Actually, to be precise, the exponential map only preserves angles and distances for points in the tangent plane which have distance $< \pi$ from q; however, we shall implicitly assume this condition holds whenever it is needed.

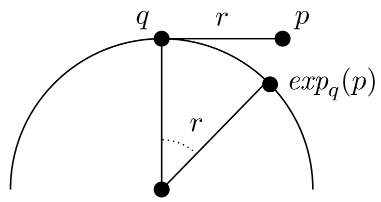


Figure 1: The exponential map preserves distances $< \pi$ and directions from q

The inverse of the exponential map is denoted $\ell_q(p)$ and it maps a point p on S^d to the tangent hyperplane, provided p is not antipodal to q. We have $exp_q(\ell_q(p)) = p$ and the formula governing the inverse map is

$$x_i \;=\; x_i' \cdot \frac{\theta}{\sin\theta}$$

where $\theta = \cos^{-1}(x_{d+1})$ is the spherical angle between p and q, $0 \leq \theta < \pi$. Note θ is also equal to the spherical distance from q to p.

To define the partial derivatives of the function f at a point q on S^d, let $F(s) = f(exp_q(s))$ for points on the tangent hyperplane T_q, and choose axes $x_1, \ldots, x_d$ for T_q. Then the first-order derivatives of f at q are defined to equal $(\partial F/\partial x_i)_q$; its second-order derivatives at q are equal to $(\partial^2 F/\partial x_i \partial x_j)_q$, etc. The best description of the derivative of f is as a gradient vector ∇f in R_{d+1} which is tangent to the sphere at q; namely, let $\nabla f(q)$ equal the vector

$$\left(\frac{\partial F}{\partial x_1}\right)_q u_1 + \left(\frac{\partial F}{\partial x_2}\right)_q u_2 + \cdots + \left(\frac{\partial F}{\partial x_d}\right)_q u_d,$$

where $u_1, \ldots, u_d$ are the unit vectors pointed in the directions of the axes $x_1, \ldots, x_d$. We think of the vector $\nabla f(q)$ being attached to the sphere at q. Similarly, the second-derivatives of f at q are best described as the $d \times d$ Hessian matrix $H = (h_{ij})$ where $h_{ij} = (\partial^2 F/\partial x_i \partial x_j)_q$. For v a unit vector tangent to S^d, the second derivative of f in the direction v is equal to $(v^T)Hv$, where v is viewed as a column vector and v^T is its transpose.

For $1 \leq i \leq n$, let the function $g_i(q)$ equal $\mathrm{dist}_S(q, p_i)$ so that

$$f(q) \;=\; \tfrac{1}{2} \sum\nolimits_{i=1}^{n} (g_i(q))^2.$$

We also let $f_i(q) = \frac{1}{2}(g_i(q))^2$ so that $f(q) = \sum_i f_i(q)$. To describe the derivatives of f, it will be sufficient to describe the derivatives of f_i.

The derivatives of f_i are easily calculated if we choose the correct set of axes for T_q: namely, let x_1 be in the direction pointing from q away from p_i, tangent to the shortest geodesic from p_i to q, and let the rest of the axes be arbitrary orthonormal vectors. The next lemma tells us how to compute the derivatives of f_i relative to these axes. By doing this for each p_i, and then converting to a common coordinate system, we can add up the derivatives of the f_i's to obtain the derivatives of f.

Lemma 2 *Let the coordinate axes $x_1, \ldots, x_d$ for T_q be chosen with x_1 pointing away from p_i. Let $\rho = \mathrm{dist}_S(q, p_i)$ be the spherical distance from p_i to q. Let $F_i(s) = f_i(exp_q(s))$. Then*

a. $\left(\dfrac{\partial F_i}{\partial x_1}\right)_q = \rho$ *and, for $j \neq 1$,* $\left(\dfrac{\partial F_i}{\partial x_j}\right)_q = 0$.

b. $\left(\dfrac{\partial^2 F_i}{\partial x_1^2}\right)_q = 1$ *and, for $j \neq 1$,* $\left(\dfrac{\partial^2 F_i}{\partial x_j^2}\right)_q = \rho \cot \rho$.

c. *For every $j \neq k$, the mixed partial* $\left(\dfrac{\partial^2 F_i}{\partial x_j \partial x_k}\right)_q$ *equals* 0.

Proof (a): Define $G_i(s) = g_i(exp_q(s)) = \mathrm{dist}_S(q, exp_q(s)) = ||q - s||$. Since the exponential map preserves distances and the x_1 axis points in the direction away from p_i, it is obvious that $(\partial G_i/\partial x_1)_q = 1$. For $j \neq 1$, let u_j be the unit vector pointing along the x_j axis. By symmetry, we have $G_i(q + tu_j) = G_i(q - tu_j)$ for all $t \in \mathbb{R}$. Thus $(\partial G_i/\partial x_j)_q = 0$. We have established that

$$\left(\frac{\partial G_i}{\partial x_1}\right)_q = 1 \quad \text{and, for } j \neq 1, \quad \left(\frac{\partial G_i}{\partial x_j}\right)_q = 0.$$

Since $F_i(r) = \frac{1}{2}(G_i(r))^2$, (a) follows immediately.

(b): The fact that $(\partial^2 F_i/\partial x_1^2)_q = 1$ follows easily by the same reasoning, since along the axis x_1, F_i equals half of the square of the distance from $\ell_q(p_i)$. To compute $(\partial^2 F_i/\partial x_j^2)_q$, we need to derive the formula for F_i of points along the x_j-axis. Let $r(t)$ denote the point $exp_q(q + tu_j)$. Thus $r(t)$ is the point on the sphere with coordinates $(0, \dots, 0, \sin t, 0, \dots, 0, \cos t)$. Let $\varphi = \varphi(t)$ denote the spherical distance from $r(t)$ to p_i, so that $f_i(r(t)) = \frac{1}{2}(\varphi(t))^2$. Differentiating with the chain rule shows that $(\partial f_i/\partial x_j) = \varphi \cdot (d\varphi/dt)$.

The points $p_i, q, r(t)$ form a right spherical triangle with the right angle at vertex q, so a spherical identity states that

$$\cos\varphi \;=\; \cos t\, \cos\rho.$$

Differentiating implicitly gives $-\sin\varphi\, d\varphi = -\cos\rho\, \sin t\, dt$, whence

$$\begin{aligned} \frac{d\varphi}{dt} &= \frac{\sin t}{\sin\varphi}\cos\rho, \\ \frac{d^2\varphi}{dt^2} &= \frac{\cos t}{\sin\varphi}\cos\rho - \frac{\sin t\, \cos\varphi}{\sin^2\varphi}\frac{d\varphi}{dt}\cos\rho \\ &= \frac{\cos t}{\sin\varphi}\cos\rho - \frac{\sin^2 t\, \cos\varphi\, \cos^2\rho}{\sin^3\varphi}. \end{aligned}$$

At the point q, we have $t = 0$ and $\varphi = \rho$, thus $\cos t = 1$ and $\sin t = 0$ and

$$\left(\frac{\partial^2 f_i}{\partial x_j^2}\right)_q \;=\; \left(\frac{d\varphi}{dt}\right)^2_{t=0} + \varphi \cdot \left(\frac{d^2\varphi}{dt^2}\right)_{t=0} \;=\; 0 + \rho\frac{\cos\rho}{\sin\rho} \;=\; \rho\cot\rho.$$

Finally, (c) follows immediately from the second part of (a), since at least one of j, k is not equal to 1. □

Lemma 2(a) has the immediate consequence that any critical point (and in particular, any local minimum) of the function f looks like the weighted average of the points p_i from the point of view of the tangent hyperplane at the critical point:

Theorem 3 *Let q be a critical point of f, with q not antipodal to any p_i. Then*

$$\sum\nolimits_{i=1}^{n} w_i(\ell_q(p_i) - q) \;=\; 0. \tag{4}$$

Proof Let f^* be the function defined on T_q by

$$f^*(x) \;=\; \frac{1}{2}\sum\nolimits_{i=1}^{n} w_i||\ell_q(p_i) - q||^2.$$

At the point q, the first derivatives of f are given by Lemma 2(a) and are equal to the first derivatives of f^*. Therefore, since q is a critical point of f, it is also a critical point of f^*. As discussed earlier, the unique critical point of f^* is the Euclidean weighted average $\sum_{i=1}^n w_i \ell_q(p_i)$. Thus, q is the Euclidean weighted average, and equation (4) is satisfied. $\square$

One might expect that Theorem 3 already implies Theorem 1; however, we have been unable to give such a simple proof of Theorem 1: the problem is the inverse exponential mapping $\ell_q()$ depends on q so there is no way to a priori rule out the possibility of having more than one point q inside the hemisphere $\mathcal{H}$ which satisfies equation (4). Furthermore, Theorem 3 is not enough to prove that every critical point q is a local minimum of f.

2.1 Proof of The Uniqueness Theorem

We next prove Theorem 1 concerning the uniqueness of the spherical weighted average. The details of the proofs in the following sections are not needed for the rest of the paper, so the reader who is not interested in the proofs may skip ahead to section 3.

Since f is a continuous function on the compact space S^d, f attains its minimum value at at least one point q. We show that q is in the interior of the hemisphere $\mathcal{H}$. First suppose that q lies completely outside $\mathcal{H}$: by reflecting q across the boundary of H, we get a point q' in the interior of $\mathcal{H}$. It is easy to verify that for every point p_i in the interior of $\mathcal{H}$, the point q' is closer to p_i than q is. Points p_i on the boundary of $\mathcal{H}$ are equidistant from q and q'. Therefore the value of $f(q') < f(q)$, contradicting the choice of q. Next, we show that the gradient of f at the boundary is always non-zero and is pointing outwards from $\mathcal{H}$. To prove this, suppose that q is on the boundary of $\mathcal{H}$. If p_i lies in the interior of $\mathcal{H}$, then Lemma 2(a) implies that the gradient of f_i points outward from $\mathcal{H}$. If p_i is a point on the boundary, then the gradient of f_i points parallel to the boundary of $\mathcal{H}$, by the second part of Lemma 2(a). The gradient of f is a weighted sum of the gradients of the f_i's and thus f points outward from the boundary of $\mathcal{H}$. Therefore, a global minimum q can lie neither on the boundary of $\mathcal{H}$ nor in the exterior of $\mathcal{H}$.

Let q be in the interior of $\mathcal{H}$ and a critical point of f, i.e., a point where the first derivatives of f are all zero. We claim that q is a local minimum of f. In fact, the second derivative test will show that f is concave up at q. To make this property precise, fix a geodesic through q and let the scalar u measure distance along this geodesic, and view f as a function of u. We shall prove that $(\partial^2 f/\partial u^2)_q > 0$, so f is concave up at q.

The function f is the sum of the functions f_i; unfortunately, the individual functions f_i are not necessarily concave up, since for a point p_i at distance $\rho_i > \pi/2$ from q, Lemma 2 states that some of the second derivatives are equal to $\rho_i \cot \rho_i < 0$. However, we shall see that it is possible to consider the points p_i in pairs, so that the sum of pairs of functions f_i will be concave up at q.

Formally, our proof will proceed by induction on the number n of points. In the base case, when $n = 1$, we have $q = p_1$ and clearly $f(s) = \frac{1}{2}(\mathrm{dist}_S(s, p_1))^2$ is concave up at $s = q$. For the induction step, let z be the geodesic which passes through both q and the center of the hemisphere $\mathcal{H}$. For any point p_i we define the angle α_i to be the angle between z and the geodesic from q to p_i, with the convention that $\alpha_i = 0$ if p_i lies on geodesic z in the direction from q to the center of the hemisphere (see Figure 2). First suppose there is some p_i with $\alpha_i = \pi/2$. Then by choice of z and since p_i lies in the hemisphere $\mathcal{H}$, the distance from q to p_i is $\rho_i \le \pi/2$ (or, if p_i is the interior of $\mathcal{H}$, then $\rho_i < \pi/2$). Thus $\cot \rho_i \ge 0$. We claim that this implies that the second derivative $\partial^2 f_i / \partial u^2$ of f_i is non-negative (resp, positive if $\rho_i < \pi/2$). This claim follows immediately from the next lemma.

Lemma 4 *Let α be the angle between the geodesic from q to p_i and the geodesic of u. Then $\partial^2 f_i/\partial u^2 \ = \ (\sin\alpha)^2 \rho_i \cot\rho_i + (\cos\alpha)^2$.*

Proof of Lemma: Let local axis v be the geodesic from q to p_i and let y be a perpendicular local axis so that y makes angle $\pi/2 - \alpha$ with u. Note the images of v, t, y under the exponential map are coplanar. Lemma 2 implies that $(\partial^2 f_i/\partial v^2)_q = 1$ and that $(\partial^2 f_i/\partial y^2)_q = \rho_i \cot \rho_i$. Using the chain rule, and differentiating twice,

$$\frac{\partial f_i}{\partial u} \ = \ \frac{\partial f_i}{\partial v}\frac{\partial v}{\partial u} + \frac{\partial f_i}{\partial y}\frac{\partial y}{\partial u} \ = \ \frac{\partial f_i}{\partial v}\cos\alpha + \frac{\partial f_i}{\partial y}\sin\alpha.$$

$$\frac{\partial^2 f_i}{\partial u^2} \ = \ \frac{\partial^2 f_i}{\partial v^2}\cos^2\alpha + 2\frac{\partial^2 f_i}{\partial v \partial y}\cos\alpha\,\sin\alpha + \frac{\partial^2 f_i}{\partial y^2}\sin^2\alpha.$$

By Lemma 2, the mixed partial is zero, so Lemma 4 is proved. □

To complete the proof of Theorem 1, it remains to prove the induction step when there are no points which make angle $\pi/2$ with the geodesic z. By Theorem 3, we know that the weighted sum of the vectors $\ell_q(p_i) - q$ in the tangent hyperplane is equal to zero. Therefore, letting $\ell_q(z)$ be the image of the axis z under the inverse exponential map, the sum of the projections of the vectors onto $\ell_q(z)$ is also zero; i.e.,

$$\sum\nolimits_{i=1}^{n} w_i \rho_i \cos\alpha_i \ = \ 0. \tag{5}$$

Since $\cos\alpha_i \neq 0$ for all i, there are a pair of points, w.l.o.g. the points p_1 and p_2, so that $\cos\alpha_1 > 0$ and $\cos\alpha_2 < 0$. Thus, $0 \leq \alpha_1 < \pi/2 < \alpha_2 \leq \pi$.

We shall prove by induction on n, the base case being $n = 2$, that when (5) is satisfied for points in $\mathcal{H}$, that $\partial^2 f/\partial u^2 \geq 0$ (and > 0 if at least one point lies in the interior of $\mathcal{H}$). For the induction step, express f as $f = f' + f''$ as follows. Choose w_1' and w_2' so that $0 \leq w_1' \leq w_1$ and $0 \leq w_2' \leq w_2$ and

$$w_1'\rho_1 \cos\alpha_1 + w_2'\rho_2 \cos\alpha_2 \ = \ 0$$

and so that either $w_1' = w_1$ or $w_2' = w_2$. Define two functions $f'(x) = \frac{1}{2}(w_1'\mathrm{dist}_S(x, p_1) + w_2'\mathrm{dist}_S(x, p_2))$ and $f''(x) = f(x) - f'(x)$. Since one of the first two coefficients in the formula for $f''(x)$ is zero, $f''(x)$ is a weighted sum of squares of distances of only $n - 1$ points from x. By construction, (5) holds for f'', so the induction hypothesis applies to f''. Similarly, f' satisfies (5) as a weighted sum of two points. Thus it will suffice to prove our claim for the $n = 2$ base case.

In the base case we have

$$w_1\rho_1 \cos\alpha_1 + w_2\rho_2 \cos\alpha_2 = 0. \tag{6}$$

Since $\rho_i \cot\rho_i \leq 1$ and since $\cos^2\alpha_i + \sin^2\alpha_i = 1$, Lemma 4 implies that

$$\partial^2 f/\partial u^2 \ \geq \ w_1\rho_1 \cot\rho_1 + w_2\rho_2 \cot\rho_2 \tag{7}$$

We define λ to equal the distance from q to the boundary of the hemisphere $\mathcal{H}$, i.e., if the geodesic z from q to the center of $\mathcal{H}$ is extended past the center of $\mathcal{H}$ to the boundary of $\mathcal{H}$ at a point r, then λ is the geodesic length from q to r. Likewise let s_1 and s_2 be the points on the boundary of $\mathcal{H}$ which are reached by extending the geodesics from q through p_1 and p_2 and define γ_1 and γ_2 to be the distances from q to s_1 and s_2, respectively.

Consider the spherical right triangle with vertices q, r, s_1, which has a right angle at r. The angle at vertex q is α_1, and a spherical right triangle identity tells us that

$$\cot\gamma_1 \ \tan\lambda = \cos\alpha_1,$$

so therefore $\cot\gamma_1 = (\tan\lambda)^{-1}\cos\alpha_1$. Since $\rho_1 \leq \gamma_1$ and the cotangent function is deceasing for angles in the interval $[0, \pi)$, we have

$$\cot\rho_1 \geq (\tan\lambda)^{-1}\cos\alpha_1$$

with strict inequality if p_1 is in the interior of $\mathcal{H}$. This plus the corresponding inequality for $\cot\rho_2$ and equations (7) and (6) imply that

$$\partial^2 f/\partial u^2 \ \geq \ (\tan\lambda)^{-1}(w_1\rho_1 \cos\alpha_1 + w_2\rho_2 \cos\alpha_2) \ = \ 0$$

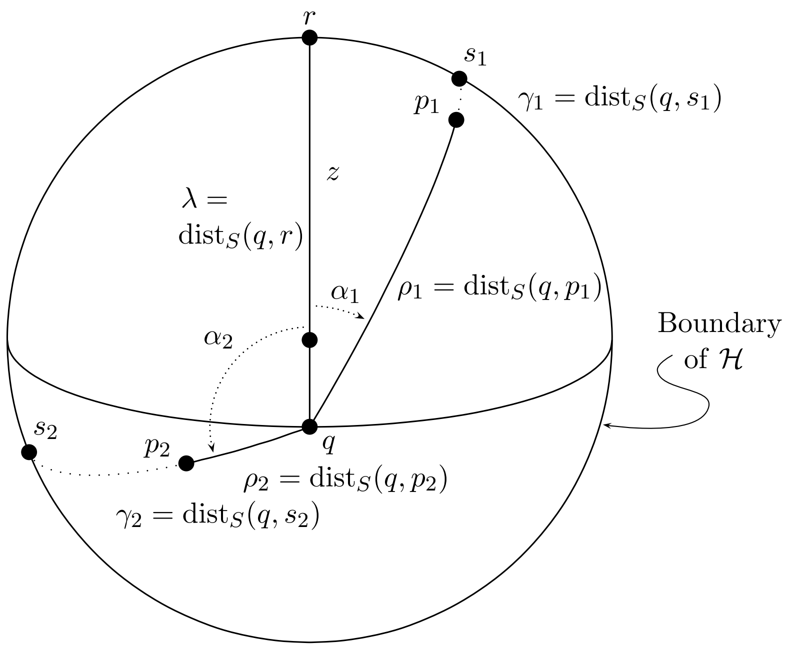


Figure 2: Top view of hemisphere $\mathcal{H}$ (orthogonal projection). All curves represent geodesics on the sphere.

(where as usual the inequality is strict if at least one of ρ_1 or ρ_2 is in the interior of $\mathcal{H}$). We are almost done, but our proof breaks down when q is the center of $\mathcal{H}$, in which case $\tan \lambda = 0$: for this case, however, $\rho_i \leq \pi/2$ for all i and therefore $\rho_i \cot \rho_i \geq 0$. So in any event, $\partial^2 f / \partial u^2 \geq 0$. Thus the second derivative test shows that f is concave up at q.

We have established that any critical point of f inside $\mathcal{H}$ is a local minimum of f. In addition, we showed that the gradient of f at any point on the boundary of $\mathcal{H}$ points outward from $\mathcal{H}$. Therefore, it follows from index theory (see [22], for example) that there is only one critical point in $\mathcal{H}$. This critical point is clearly the unique point inside $\mathcal{H}$ where f attains its minimum value; therefore, it is also the unique point on S^d where f attains its minimum value, which completes the proof of Theorem 1. $\square$

The proof of Theorem 1 involved splitting points p_i and then pairing up points so that each pair defined a concave up distance function at q. Obviously the hypothesis that the points all lie in a hemisphere is stronger than is needed: instead, all we really used was that $\sum_i \rho_i \cot \rho_i > 0$ where ρ_i is the distance of p_i from q. And even this condition is stronger than is really necessary except in the degenerate case where the points all lie in a $(d-1)$-dimensional subspace of the sphere.

Thus there are various ways that the hypotheses of Theorem 1 can be weakened: two of these are stated in the next theorem. If q is a point on the sphere, then $B_q(t)$ denotes the ball of radius t around q, namely, the set of points on the sphere of distance $\leq t$ from q.

Theorem 5 *Let f, p_i, w_i be as usual.*

(a) *Let $0 < \varphi \leq \pi/2$. Suppose q is a critical point of f and $B_q(\varphi)$ contains points p_i of total weight ≥ 0.5 and then all the points p_i are contained in the $B_q(\pi - \varphi)$. Then q is a local minimum of f according to the second derivative test.*

(b) *Let $0 < \varphi < \psi \leq \pi/2$. Suppose $0 < w \leq 1$ and that $w\varphi + (1-\psi)\pi \leq \psi$. Further suppose there is a point v so that $B_v(\varphi)$ contains points p_i of total weight $\geq w$ and that all the points p_i are in $B_v(\pi - 2\psi - \varphi)$. Then f has a unique minimum. This unique minimum is inside $B_v(\psi)$ and is the only critical point inside $B_v(\psi)$.*

Proof Part (a) is immediate from the method used to prove Theorem 1: just match points outside of $B_q(t)$ with points inside $B_q(t)$.

To prove (b), note that the condition on w ensures that the global minimum lies inside $B_v(\psi)$. And any critical point inside $B_v(\psi)$ satisfies the condition of part (a), with φ replaced by $\varphi + \psi$. □

Part (b) of the theorem has the advantage that it is a test that can be applied without having already found the critical point q. We conjecture that part (b) can be substantially strengthened (see the final section).

2.2 Continuity of Spherical Weighted Averages

Theorem 6 *Let values for $p_1, \ldots, p_n$ and $w_1, \ldots, w_n$ and q be chosen that satisfy the hypotheses of Theorem 1 or 5. Then there is a neighborhood of $p_1, \ldots, p_n, w_1, \ldots, w_n$ in which the weighted average q is a C^∞-function of $p_1, \ldots, p_n, w_1, \ldots, w_n$.*

This theorem is proved as a corollary of the Implicit Function Theorem (see Spivak [32]). Indeed, the weighted average q is equal to a root of ∇f. Except at points antipodal to one of the p_i's, the function ∇f is clearly a C^∞-function of the points $p_1, \ldots, p_n$ and of the weights $w_1, \ldots, w_n$. Furthermore, the matrix of first derivatives of ∇f is the Hessian matrix H.

Since the proof of Theorems 1 and 5 establishes that the second derivative test shows that f is concave up, we know that the Hessian matrix H is positive definite and thus non-singular. Therefore the conditions of the Implicit Function Theorem are satisfied at any point q satisfying the conditions of Theorems 1 or 5.

Note that the Inverse Function and Implicit Function Theorems can be used to calculate the first-order and higher-order derivatives of the weighted average function Avg.

It will be helpful in section 3 to know that the Hessian matrix H is positive definite and non-singular in a neighborhood of the weighted average q. One further property: at every point q and every unit vector v, we have $v^T H v \leq 1$. This is proved by noting that lemma 2 shows that this holds for the Hessians of the each of the F_i functions, since $\rho \cot \rho \leq 1$.

2.3 A Convexity Property

In this section we show that the points q which can be written as a weighted average of $x_1, \ldots, x_k$ form a convex set; in fact, they form precisely the convex hull of the points $x_1, \ldots, x_k$.

Definition A subset C of S^d is convex iff for any two points $x, y \in C$ there is a shortest geodesic from x to y which lies entirely in C.

The subset C is the *convex hull* of a set D iff C is the unique smallest convex set containing D.

The above definition allows antipodal points to be in a convex set C, as long as there is at least one geodesic between the antipodal points which lies in C. Note that if x_1 and x_2 are antipodal, then they do not have a convex hull. However, for any third point x_3, $\{x_1, x_2, x_3\}$ does have a convex hull, namely, the geodesic from x_1 through x_3 to x_2.

To state the next theorem with maximum generality, we define that a point q is a *proper weighted average* of $x_1, \ldots, x_k$ if there are non-negative weights $w_1, \ldots, w_k$ which sum to 1 such that $q = \sum_i w_i \cdot x_i$ and there is a hemisphere $\mathcal{H}$ such that for each non-zero w_i, we have $x_i \in \mathcal{H}$, and that for at least one non-zero w_i, x_i is in the interior of $\mathcal{H}$. The proper weighted averages are of course precisely the weighted averages which are guaranteed by Theorem 1 to be uniquely defined. We say that the weighted average is *strongly proper* provided the hemisphere $\mathcal{H}$ can be chosen so that each x_i with non-zero weight is in the interior of $\mathcal{H}$.

Theorem 7 *Suppose that $x_1, \ldots, x_k$ are distinct points, and that it not the case that $k = 2$ with x_1 and x_2 antipodal. Then the convex hull C of $\{x_1, \ldots, x_k\}$ exists and is equal to the set of proper weighted averages of $x_1, \ldots, x_k$. If $x_1, \ldots, x_k$ lie in a hemisphere, then the convex hull C is a subset of the hemisphere. If they do not lie in a hemisphere, then the convex hull is the entire sphere S^d.*

Proof Let C be the intersection of all the hemispheres which contain $\{x_1, \ldots, x_k\}$, or $C = S^d$ if there are no such hemispheres. (Recall that hemispheres are always closed.) Clearly C is convex.

To show that every proper weighted average is in C, suppose that $\mathcal{H}$ is a hemisphere containing all of $x_1, \ldots, x_k$ and that a point q is a proper weighted average of $x_1, \ldots, x_k$: we must show that $q \in \mathcal{H}$. If some x_i with non-zero weight is in the interior of $\mathcal{H}$, then Theorem 1 implies that $q \in \mathcal{H}$. However, if every x_i with non-zero weight is on the boundary of $\mathcal{H}$, then it is obvious that q also lies on the boundary of H since q is a point where the weighted sum of squares of spherical distances is minimized. Hence $q \in \mathcal{H}$ in either case.

It remains to to prove that every point q in C can be expressed as a proper weighted average of $x_1, \ldots, x_k$. Since it is much simpler, we first prove this in the special case where the points x_i all lie in the interior of a hemisphere $\mathcal{H}$. Let $x_i' = \ell_q(x_i)$ be the point in the tangent plane T_q corresponding to x_i. The fact that $q \in C$ implies that q is in the convex hull of the points x_i', since otherwise there is a closed halfplane $\mathcal{G}$ in T_q with q on its boundary which contains none of the points x_i', and its image $exp_q(\mathcal{G})$ in S^d is a hemisphere which contains q but none of the x_i's, contradicting the fact that $q \in C$. Therefore q may be written as a Euclidean weighted average $q = \sum_i w_i x_i'$, and now Theorems 1 and 3 imply that q is equal to the proper, spherical weighted average $q = \mathop{\textstyle\sum\kern-0.9em\bigcirc}_i w_i \cdot x_i$.

Now we prove the more difficult general case. To handle the degenerate cases, our proof proceeds by induction on the dimension d. For the base case, $d = 1$, S^1 is a circle. Let x_1 and x_2 be the first points clockwise and counterclockwise (respectively) from q. Letting δ equal the sum of the arclengths from q to x_1 and to x_2, it is clear that $q \in C$ iff $\delta < \pi$ and this holds iff q is a strongly proper weighted average of x_1 and x_2.

Next we argue the induction step, $d > 1$. For r a point on S^d, let $\mathcal{H}_r$ be the hemisphere centered at r. When q and x_i are in $\mathcal{H}_r$ we define δ_{r,x_i} to be equal to the distance from x_i to the boundary of $\mathcal{H}_r$ in the direction away from q: i.e., measured along the geodesic containing q and x_i. We shall only consider points r which are in $\mathcal{H}_q$, and letting $\delta_{r,x_i} = 0$ for x_i not in $\mathcal{H}_r$, δ_{r,x_i} is always defined for these r's. For each $r \in \mathcal{H}_q$, define $f(r)$ to equal the maximum value of δ_{r,x_i}. Since f is a continuous function and has compact domain, f attains a minimum value, $f(s)$, at some point $s \in \mathcal{H}_q$.

Let I_r denote the set of points x_i which are in the interior of $\mathcal{H}_r$. First suppose that I_s is non-empty or, equivalently, $f(s) > 0$. Let $y_1, \ldots, y_\ell$ enumerate all the points x_i which are in I_s, and let $z_i = \ell_q(y_i)$ be the corresponding points in the tangent plane T_q. We claim that q is in the (Euclidean) convex hull of $\{z_1, \ldots, z_\ell\}$. To prove this claim, suppose it is false: then there is a vector $\vec{v}$ so that the dot product $(z_i - q) \cdot \vec{v} \leq 0$ for all z_i. Displace the point s infinitesimally in the direction $\vec{v}$; i.e., let $s' = exp_q(\ell_q(s) + \epsilon \cdot \vec{v})$ for sufficiently small $\epsilon > 0$. Clearly, for small ϵ, we have $\delta_{s',y_i} < \delta_{s,y_i}$ for each $y_i \in I_s$. Also, for $\epsilon < f(s)$, we have $\delta_{s',x} < \epsilon < f(s)$ for each $x \in I_{s'} \setminus I_s$. Therefore, $f(s') < f(s)$ which contradicts the choice of s. That proves the claim that q is in the convex hull of $\{z_1, \ldots, z_\ell\}$ and then exactly as argued in the preceding special case, Theorems 1 and 3 imply that q is a (strongly) proper spherical weighted average of the points $y_1, \ldots, y_\ell$.

Secondly, consider the case where $I_s = \emptyset$, so none of the points x_i are in the interior of $\mathcal{H}_s$. In this case, since $q \in C$, the point q must also lie on the boundary B of $\mathcal{H}_s$ (since the hemisphere with center antipodal to s contains all the points x_i). The boundary B is itself a $(d-1)$-sphere. It is easy to check that q is in every $(d-1)$-hemisphere subset of B which contains every point $x_i \in B$ — this is because any such hemispheric subset of B can be "lifted" to a hemispherical subset of S^d which contains every point x_i. Therefore, by the induction hypothesis, q can be written as a proper weighted average of the points x_i which lie in B. □

The next theorem is a corollary to the method of proof of the previous theorem,

Theorem 8 *Suppose that $x_1, \ldots, x_k$ are distinct points, and that it not the case that $k = 2$ with x_1 and x_2 antipodal. Then the convex hull C of $\{x_1, \ldots, x_k\}$ exists and is equal to the set of strongly proper weighted averages of $x_1, \ldots, x_k$.*

In particular, q can be written as a proper weighted average of the points x_i if and only if it can be expressed as a strongly proper weighted average of the points. This can be further strengthened as follows:

Theorem 9 *Every point in the convex hull C of $\{x_1, \ldots, x_k\}$ can be written as a strongly proper weighted average of at most $d+1$ many of $x_1, \ldots, x_k$.*

Proof This follows from Theorems 8, 1 and 3 and corresponding fact for weighted averages in d-dimensional Euclidean spaces.

3 Algorithms for Spherical Weighted Averages

We now present two algorithms for computing a spherical weighted average $q = \sum_{i=1}^{n} w_i \cdot p_i$. The first, Algorithm A1, is a linear convergence rate algorithm which iteratively searches for a point q which satisfies the condition of Theorem 3. The second, Algorithm A2, is a quadratic convergence rate algorithm which uses Newton descent to find a critical point of the function f. Runtimes for these algorithms are reported in section 5.

Both algorithms are iterative, i.e., they start with an estimate for the value of q and produce a better estimate. To start the process, we choose an initial value q_0 on the sphere taking the Euclidean weighted average of the points p_i and normalizing to place on the sphere; namely,

$$q_0 := \sum_{i=1}^{n} w_i p_i \,/\, ||\sum_{i=1}^{n} w_i p_i||.$$

If the hypotheses of Theorems 1 or 5 hold, then $\sum_i w_i p_i$ is non-zero.

Given an estimate for q, Algorithm A1 maps all the points p_i to the tangent hyperplane at q, then calculates their Euclidean weighted average u in the hyperplane and maps this back to the sphere by the exponential map.

Algorithm A1:

Inputs: Points $p_1, \ldots, p_n$ on S^d

Non-negative weights $w_1, \ldots, w_n$ with sum 1.

Output: The spherical weighted average of the inputs.

Initialization: Set $q := \sum_{i=1}^{n} w_i p_i / || \sum_{i=1}^{n} w_i p_i||$.

Main Loop:

For $i = 1, \ldots, n$

Set $p_i^* := \ell_q(p_i)$

Set $u := \sum_{i=1}^{n} w_i (p_i^* - q)$.

Set $q := exp_q(q + u)$.

If $||u||$ is sufficiently small, output q and halt.

Otherwise continue looping.

By Lemma 2 and the observation in the proof of Theorem 3, the vector u in the algorithm is the *negative* of the gradient $\nabla f(q)$ of f at q. Thus, the loop in Algorithm A1 is setting $q = q - \nabla f$, i.e., it moves q in the direction of steepest descent. We argue below that A1 converges linearly.

Next we give the Algorithm A2 which has a quadratic convergence rate. The main loop for Algorithm A2 updates an estimate q for the spherical weighted average by computing the first-order and second-order derivatives of the function f at the point q. It then inverts the Hessian matrix and computes a better estimate of the point where the first-derivative of f is zero.

Algorithm A2:

Inputs: Points $p_1, \ldots, p_n$ on S^d
Non-negative weights $w_1, \ldots, w_n$ with sum 1.
Output: The spherical weighted average of the inputs.
Initialization: Set $q := \sum_{i=1}^n w_i p_i / || \sum_{i=1}^n w_i p_i ||$.
Main Loop:
Set up a local Euclidean coordinate system C at q.
For $i = 1, \ldots, n$
Set $p_i^* := \ell_q(p_i)$
Let $\rho_i := ||p_i^* - q||$.
Set up a local coordinate system C_i at q such that the first axis $x_1^{(i)}$ points in the direction away from p_i.
Let M_i be the matrix that converts C_i coordinates to C coordinates.
Let H_i be the $d \times d$ diagonal matrix with first entry 1 and the rest of the entries $\rho_i \cot \rho_i$.
Set $u := \sum_{i=1}^n w_i (p_i^* - q)$.
Set $H := \sum_{i=1}^n M_i H_i M_i^T$. (Note $M_i^T = M_i^{-1}$, the transpose of M)
Set $v := H^{-1} u$.
Set $q := exp_q(q + v)$.
If $||v||$ is sufficiently small, output q and halt.
Otherwise continue looping.

By Lemma 2, each H_i is the Hessian matrix of f_i at the point q in the coordinate system C_i; and H is the Hessian matrix of f at q in the coordinate system C. The points p^* and the negative gradient u are to be represented in the coordinate system C. To see that the convergence rate of Algorithm A2 is really quadratic, note that it is performs Newton's algorithm for finding a root of ∇f; namely it updates q by $q = q - H(\nabla f)$. As noted after Theorem 6, the Hessian H is positive definite in a neighborhood of the weighted average q_{soln}. In addition, by finite dimensionality, $u \cdot Hu$ is bounded away from zero in a neighborhood of q_{soln}. These facts are sufficient for Newton's algorithm to converge quadratically, see e.g. Theorem 1.4.6 of Polak [26].

We have $u = Hv$, so by comments after Theorem 6, $1 \geq v \cdot u > \epsilon$ for some $\epsilon > 0$ in some neighborhood of q_{soln}. From this it can be shown, again in a sufficiently small neighborhood of q_{soln}, that if $q' = exp_q(q+u)$, then $f(q') < \delta f(q)$ for some $0 < \delta < 1$. This implies that A1 converges at least linearly.

The above discussion on convergence rates only covers local convergence, i.e., convergence assuming that q is already sufficiently close to q_{soln}. It is possible to convert these algorithms into ones with guaranteed global convergence (c.f. [26]), but in practice we have never seen a situation where Algorithms A1 and A2 failed to converge well.

4 Applications to Splines

4.1 Splines Based on Weighted Spherical Averages

Spline curves are curves which are specified by a small set of parameters and which have nice smoothness properties, such as having continuous k-th order derivatives. They are widely used in drafting and computer aided manufacturing to specify curves and surfaces. In robotics and animation, spherical splines are often used to specify orientations for smooth kinematic motion of solid bodies.

We consider the problem of specifying spline functions which take values on the unit d-sphere S^d: these are useful for a variety of applications. In graphics and robotics, the most prominent applications are based on the equivalence between quaternions and (pairs of antipodal) points on the 3-sphere. A spline function taking values on the 3-sphere can be used as a spline function taking quaternion values and, if the curve is parameterized by time t, then the spline curve specifies the orientation of a solid body as a smooth function of time.

The most common method of using splines in Euclidean space $\mathbb{R}^{d+1}$ involves the selection of control points $p_1, \ldots, p_n$ in $\mathbb{R}^{d+1}$ and *blending functions* $f_1(t), \ldots, f_n(t)$, also called *basis functions*. The Euclidean spline curve $s(t)$ is defined by

$$s(t) \;=\; \sum\nolimits_{i=1}^{n} f_i(t) p_i.$$

The blending functions must always satisfy the properties

$$f_1(t) + f_2(t) + \cdots + f_n(t) \;=\; 1 \qquad \text{and} \qquad f_i(t) \;\geq\; 0, \text{ for all } i, \tag{8}$$

for t in the interval $[a, b]$. Usually the blending functions enjoy additional properties such as having continuous k-th order derivatives; or being 'bitonic', i.e., being increasing on an initial part of the interval $[a, b]$ and decreasing on the the rest of the interval; or having local support.

In order to define spherical splines, let $p_1, \ldots, p_n$ now be points on S^d, and let $f_1(t), \ldots, f_n(t)$ be functions which satisfy (8). We define a spline curve $s(t)$ which takes values on the unit sphere by setting

$$s(t) \;=\; \sum\nolimits_{i=1}^{n} f_i(t) \cdot p_i.$$

In practice, we should ensure that for each value of the parameter t, the set of control points p_i for which $f_i(t) \neq 0$ is contained inside a hemisphere, or at least is mostly contained inside a hemisphere so as to satisfy the conditions of Theorems 1 or 5. This condition is most readily met provided that either (a) the blending functions have local support and consecutive control points are not too widely spaced, or (b) the control points all lie inside a single hemisphere.

The most common applications of splines use B-splines with the blending functions f_i being piecewise cubic curves with continuous second derivatives. To define the B-spline curves, one picks control points $a = t_1, t_2, \ldots, t_{n-1}, t_n = b$ (to be precise, one may also pick additional control points, but this does not impact the present discussion), which then determine the blending functions. The blending functions have the property that for $t_j \le t \le t_{j+1}$, the only non-zero $f_i(t)$ are for $i \in \{j-1, j, j+1, j+2\}$. In particular, $f_i(t_j)$ is non-zero only if $i \in \{j-1, j, j+1\}$. In this case the spherical spline points $s(t)$ will be well-defined provided that any four consecutive control points lie in a hemisphere. Of course, this requirement that any four consecutive control points lie in a hemisphere can be relaxed somewhat, in light of Theorem 5.

Theorem 6 implies that if the the blending functions f_i have continuous k-th derivatives, then the spline curve $s(t)$ also has continuous k-th derivatives.

Since Bézier curves are special cases of B-spline curves, one can define spherical Bézier curves in terms of our B-spline curves. These curves will generally be different than the spherical Bézier curves generated by de Casteljau methods.

4.2 Interpolation with Spherical Splines

The previous discussion concerned B-spline curves defined with control points — in general, the curve does not pass through the control points. We now take up the problem of defining a spline curve that interpolates a desired set of points.

Suppose we are given points $c_1, \ldots, c_n$ on the d-sphere, and are given time values $t_1 < t_2 < \cdots < t_n$: we wish to find a smooth curve s lying on the sphere, parameterized by t, so that $s(t_1) = c_i$ for all i. The basic problem is to choose additional knot positions and control points p_i which defines a spherical spline curve which satisfies these conditions.

There are of course a variety of possible ways to define B-spline curves: for experimental purposes, we have implemented one kind of B-spline curve that gives blending functions which are piecewise cubic polynomials and which have continuous second derivatives. We implemented two sets of algorithms, one with a linear convergence rate and one with a quadratic convergence rate, for both the 2-sphere and the 3-sphere. We report timing results below, but since the linear convergence rate algorithm worked almost as fast as the quadratic convergence rate algorithm, and since the former algorithm is much easier to describe and to implement, we will describe only the first algorithm in detail.

We used a standard B-spline implementation (see, e.g., [38]) having as knot positions the values $t_0, t_1, t_2, \ldots, t_{n+4}, t_{n+5}$ with $t_0 = t_1 = t_2 = t_3$ and $t_{n+5} = t_{n+4} = t_{n+3} = t_{n+2}$. We 'double' the first and last control points, requiring that $p_0 = p_1$ and $p_{n+1} = p_n$. Our blending functions are defined as usual for piecewise-cubic B-splines, namely, define

$$B_{i,1}(t) \;=\; \begin{cases} 1 & \text{if } t_i \le t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

and

$$B_{i,n+1}(t) \;=\; \frac{t - t_i}{t_{i+n} - t_i} B_{i,n}(t) + \frac{t_{i+n+1} - t}{t_{i+n+1} - t_{i+1}} B_{i+1,n}(t),$$

using the convention that $0/0 = 0$. The blending functions are defined by $f_i(t) \;=\; B_{i,4}(t)$, are defined for $t \in [t_3, t_{n+2}]$, are piecewise cubic, and have continuous second derivatives. The support of f_i is in the interval $[t_i, t_{i+4}]$. The spline curve is given by $s(t) = \sum_{i=0}^{n+1} f_i(t) \cdot p_i$.

Once we have the blending functions, it remains to choose control points p_i so that the points c_i are interpolated correctly, with $s(t_{i+1}) = c_i$. In the Euclidean setting, this yields a set of linear equalities given by a tri-diagonal matrix (see, for instance, [38]), namely,

$$\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & 0 & \cdots & 0 \\ 0 & \alpha_3 & \beta_3 & \gamma_3 & 0 & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \alpha_{n-1} & \beta_{n-1} & \gamma_{n-1} \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_{n-1} \\ p_n \end{pmatrix}$$

The row sums of the matrix are all equal to 1.

In the Euclidean case, one can easily solve this tridiagonal matrix for the control points p_i. In the spherical setting, we interpret the matrix equation as n spherical weighted averages:

$$c_i \;=\; \alpha_i \cdot p_{i-1} \oplus \beta_i \cdot p_i \oplus \gamma_i \cdot p_{i+1}.$$

where $\beta_1 = \beta_n = 1$ and $\alpha_1 = \alpha_n = \gamma_1 = \gamma_n = 0$. However, since the sphere is not a linear space under the definition of spherical weighted averages, it is not sufficient to invert the tridiagonal matrix to solve for the control points p_i. Instead it is necessary to use an iterative method of solving for the control points p_i. The simplest method is to use one of the following two methods of iteratively solving for the control points:

Algorithm S1:

Inputs: Interpolation points $c_1, \ldots, c_n$,
and real coefficients $\alpha_i, \beta_i, \gamma_i$ $(1 \leq i \leq n)$
Output: Control points $p_1, \ldots, p_n$

Initialization: Set $p_i := c_i$, for $i = 1, \ldots, n$.
Main Loop:
For $i = 1, \ldots, n$
Set $p_i^* := -(\alpha_i \cdot \ell_{c_i}(p_{i-1}) + \gamma_i \cdot \ell_{c_i}(p_{i+1}))/\beta_i$.
Let $\delta_i := ||p_i^* - \ell_{c_i}(p_i)||$.
For $i = 1, \ldots, n$
Set $p_i := exp_{c_i}(p_i^*)$
If the values of δ_i are all sufficiently small, halt.
Otherwise, continue looping.

Note that in Algorithm $S1$, the points c_i and p_i lie on the d-sphere, and each value p_i^* lies in the d-dimensional Euclidean space T_{c_i}, which is the hyperplane tangent to the d-sphere at the point c_i. The only purpose of computing the scalar values δ_i is to measure the distance between the old and new values of p_i, so as to have a stopping criterion. The general idea behind the algorithm is quite simple: the new value of p_i is set so that its weighted average with the old values of p_{i-1} and p_{i+1} would correctly give the interpolation point c_i. Of course, since all the p_i values are being updated at once, a single iteration of the loop does not yield a solution. But since the diagonal entries of the matrix dominate the off-diagonal entries, the iteration converges towards a solution.

In practice, we use a modified version of the above algorithm which converges approximately twice as fast:

Algorithm S2:

Inputs: Interpolation points $c_1, \ldots, c_n$,
and real coefficients $\alpha_i, \beta_i, \gamma_i$
Output: Control points $p_1, \ldots, p_n$

Initialization: Set $p_i := c_i$, for $i = 1, \ldots, n$.
Main Loop:
For $i = 1, \ldots, n$
Set $p^* := -(\alpha_i \cdot \ell_{c_i}(p_{i-1}) + \gamma_i \cdot \ell_{c_i}(p_{i+1}))/\beta_i$.
Let $\delta = ||p^* - \ell_{c_i}(p_i)||$.
Set $p_i := exp_{c_i}(p^*)$.
If all n values for δ were sufficiently small, halt.
Otherwise, continue looping.

The algorithms above both have a linear convergence rate; i.e., in order to get k digits of accuracy, the loop must be iterated $O(k)$ times. We also implemented algorithms with quadratic convergence rates: these algorithms computed discrepancy vectors equal to

$$\delta_i \ = \ \ell_{c_i}(\alpha_i \cdot p_{i-1} \oplus \beta_i \cdot p_i \oplus \gamma_i \cdot p_{i+1})$$

which are vectors in the Euclidean space T_{c_i} tangent to the sphere at c_i. The goal is to find points p_i for which the discrepancy vectors equal zero. In addition, the algorithm computes $d \times d$ matrices $M_{i,j}$ which are Jacobian matrices giving the rate of change of the discrepancy vector δ_i with respect to the changes in the control point p_j. (Changes in the control point p_j are measured by vectors in the d-dimensional tangent space T_{p_j}. Thus, using $M_{i,j}$ requires setting up a local coordinate system each p_i which provides a basis for the tangent space T_{p_i}.) The algorithm then solves a linear of equations to implement a Newton-method estimate for improved control points.

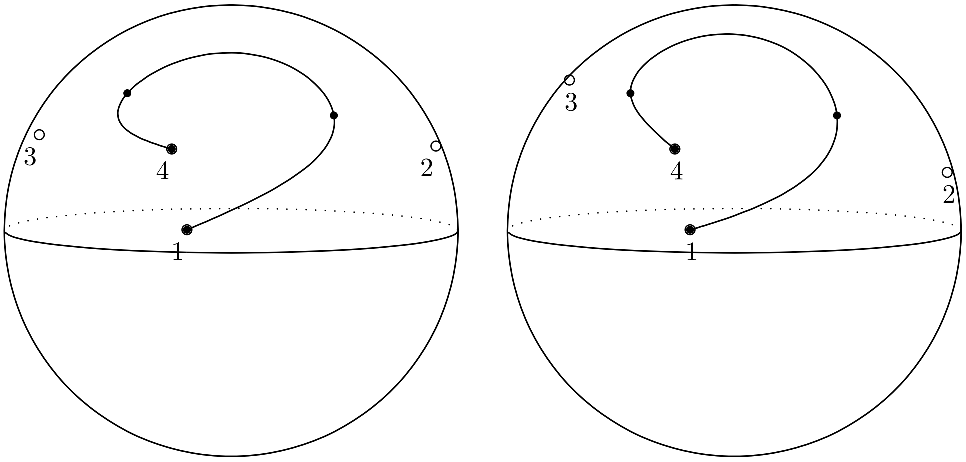
The disadvantage of the quadratic convergence rate method is that it is much more difficult to implement than Algorithms $S1$ or $S2$. Furthermore the runtime is not greatly improved over the linear convergence rate algorithms, since each loop iteration is much slower. We tested the runtimes of the various algorithms to obtain approximately 16 digits of accuracy (the IEEE floating point double precision limits). When $d = 2$ (for finding splines on the 2-sphere embedded in $\mathbb{R}^3$), the quadratic convergence rate method was experimentally observed to be twice the speed of the linear convergence rate method. However, for $d = 3$ (for splines on S^3 or for quaternion spline curves), the quadratic convergence rate method was slightly slower than the linear convergence rate algorithm.

In view of the relatively lackluster performance advantage of the quadratic convergence rate algorithm, and in view of the substantial difficulty of implementing the quadratic convergence rate methods, we recommend use of the linear convergence rate methods only.

4.3 Examples of Spline Interpolation

Figures 3 and 4 show some experimental results of spline interpolation. Figures 3a and 3b show curves which interpolate four points on the sphere — the control points for the curve are shown as open circles and are seen to lie completely within one hemisphere. For Figure 3a, the knots were equally spaced; and for Figure 3b, the knots were spaced proportionally to spherical distance between the interpolation points. Clearly the second version with unequally spaced knots yields a more rounded, smoothly varying curve. It is beyond the scope of the present paper to investigate the best methods of choosing knot positions; rather, we are using this as an example of how techniques for generating Euclidean spline curves can now be applied to curves on spheres.

Figure 4 shows curves that interpolate 10 points that circumnavigate the 2-sphere. Again, Figure 4a shows the result of choosing equally spaced knots and Figure 4b shows the curve obtained with knots spaced proportionally to the spherical distance between the interpolation points. Once again, the use of unequally spaced knots yields a smoother looking curve. One interesting feature of the control points shown in the curves Figure 4 is that they do not satisfy the hypotheses of Theorem 1. For example, the control points numbered 6,7,8,9 in Figure 4a do not lie in a single hemisphere: nonetheless, the spherical distance is still well-defined for the points on the spline curve and the spline curve still varies smoothly. The control points



(a)   (b)

Figure 3: Interpolating four points on S^2. (a) shows the curve generated with equally spaced knots. (b) is the curve generated with knots spaced proportionally to the spherical distance between interpolation points. The interpolated points are drawn as small disks; the control points are drawn as small circles.

numbered 2,3,4,5 are another example of this. However, it should be noted that if the interpolation points were much more widely spaced, then the well-definedness and the smoothness of the spline curve could breakdown; for instance, if the interpolation points were more widely spaced, then the control points 6 and 9 could "wrap around" the sphere past each other.

5 Experimental results

This section reports experimentally observed runtimes for the algorithms for weighted averages and for spline interpolation. Our algorithms were implemented in C++ (in Microsoft's Visual C++), and run on a 400MHz Pentium II. We tried to code our algorithms relatively efficiently, but did use classes for vectors, matrices, etc., which of course impacts performance due to the overhead of constructors and destructors.

Table 1 shows the observed average computation times for the two algorithms for computing spherical weighted averages. The run times are reported for both the linear convergence rate Algorithm A1 and the quadratic convergence rate Algorithm A2, on both the 2-sphere and the 3-sphere. Both

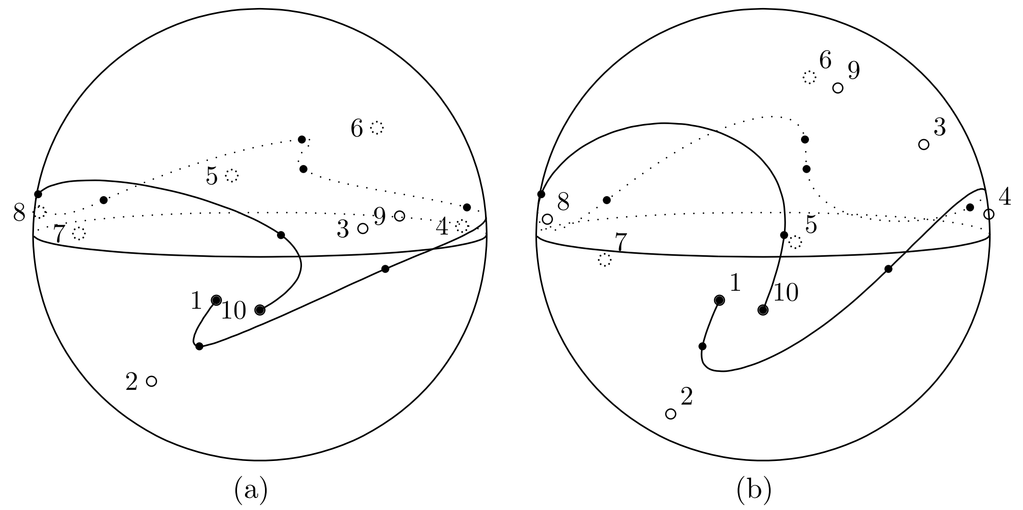


Figure 4: Interpolating ten points on S^2. (a) shows the curve generated with equally spaced knots. (b) is the curve generated with knots spaced proportionally to the spherical distance between interpolation points. Lines and control points that are on the back side of the sphere are drawn dotted.

algorithms were quite fast, always less than a millisecond per computation of weighted average of small sets of points: this is more than adequate for interactive applications and is adequate for many real-time applications. It was observed that the quadratic convergence rate algorithm is substantially faster than the linear convergence rate algorithm. Note that the speed advantage of algorithm A2 over A1 was less for the 3-sphere than for the 2-sphere; we expect that the speed advantage would decline even further for higher dimensional spheres, since Algorithm A2 has to perform a relatively expensive $d \times d$ matrix inversion, and this will tend to dominate the run time as d increases.

It should be noted that the times reported in Table 1 are for the computation of weighted averages to approximately 16 digits of accuracy, the limits of the IEEE double-precision floating point specification. If one is interested in single precision accuracy, the linear convergence algorithm is likely to be approximately as fast as the quadratic convergence rate algorithm.

Now we turn to the run time for the computation of interpolating spherical spline curves. Here we are given n points c_i on the sphere along with knot positions (values of the parameter t) and the algorithm

Algorithm	Averaging 4 points		Averaging 12 points	
	on S^2	on S^3	on S^2	on S^3
A1	0.1538	0.196	0.3626	0.423
A2	0.0396	0.103	0.0846	0.265

Table 1: Run times in milliseconds of the computation of spherical weighted averages to approximately 16 digits of accuracy. Algorithm A1 is the linear convergence rate algorithm. Algorithm A2 is the quadratic convergence rate algorithm. The time reported is the average elapsed real time achieved over 5,000 computations.

computes the control points which define a spline curve based on piecewise cubic blending functions which interpolates the control points at the specified values of t. Column 1 of Table 2 reports the time required to compute the control points from the interpolated points, using Algorithm $S2$.

The run times of Algorithm S2 are quite satisfactory; however, generally one wishes to not only compute the control points, but also to compute points along the curve. These points along the curve are to be computed as spherical weighted averages of the control points, i.e., are computed using Algorithm A1 or A2. The runtimes of Algorithms A1 and A2 are only one order of magnitude less the run time of Algorithm S2, so the time to calculate a large number of points along the curve dominates the time needed to calculate the control points. Therefore, we report in Table 2, the total time needed to compute control points and to compute a reasonably large number of points along the curve. The points along the curve were computed using the faster Algorithm A2: we used the previous two points on the curve to make a linear prediction of the next point on the curve, which was used as the initial value q_0 in $A2$. The linear prediction gets better as the points on the curve are more closely spaced; hence, the computation of 256 equally spaced points on the spline curve is faster per point than the computation of 64 equally spaced points on the curve.

It is difficult to compare our runtimes to that of previous authors; in part, because very few authors report detailed run times. A number of authors have reported runtimes for algorithms which find curves minimizing an energy function; the best of these run times are those of Ramamoorthi-Barr [28] who compute good curves within approximately four seconds. The only other authors who report such detailed run times were Gabriel-Kajiya [12] who report run times of “a few tens of seconds” for an APL2 implementation on an IBM 4341. Thus our algorithms are faster than

Number of interpolated points	Computation of control points	Computation also of 64 curve points	Computation also of 256 curve points
4 points on S^2	0.276	2.716	7.801
12 points on S^2	1.768	4.213	10.943
4 points on S^3	0.318	6.388	20.133
12 points on S^3	2.582	9.037	27.107

Table 2: Run times in milliseconds of the computation of interpolating spherical spline curves to approximately 16 digits of accuracy. The first column reports the time needed to compute the control points. The second and third columns report this time plus the time needed to compute 64 and 256 points on the interpolating curve for equally spaced values of t. Algorithm S2 was used to compute the control points, and Algorithm A2 used to compute the points equally spaced on the curve. Algorithm A2 was seeded with good estimates for the successive points (see text).

theirs by perhaps one or two orders of magnitude. To be fair, we should note that the energy minimizing approach, although substantially slower, can potentially give better quality results, since the energy function can be customized for each particular application.

Kim and Nam [20] report detailed runtimes for computing spherical splines based on circular blending: their experiments were carried out on SGI's with clock rates between 33 MHz and 50 MHz, however they do not describe the programming language used. Adjusting for processor clock rates, their runtimes are approximately twice as fast as ours: this difference in run time partly reflects the fact that our algorithms use an iterative solution, and their circular blending function computes the spline explicitly. Kim and Nam also implemented several other fast methods of generating spherical spline curves, including the Bézier method and Shoemake's Squad method. These were roughly equivalent in runtime to their circular blending function, in that the runtimes of various methods ranged from 25% faster to slightly worse than two times slower. Thus these methods all have runtime either comparable to or approximately twice as fast as our spherical spline curves based on spherical weighted average. We were somewhat surprised at the speed of our algorithm in comparison to the other algorithms, since our algorithm seems to need a good deal more computation; so the relative speeds probably also represent differences in implementations. In our opinion, the advantages of our splines as outlined in the introduction outweigh their slower computation time in many potential applications; and in any event, our spherical spline curve algorithms are sufficiently fast for all but the most time critical applications.

6 Open Problems

We conclude with a few remarks about open problems and the possibility of improving the analysis and applications of our spherical spline curves in future work.

First, we have not yet derived formulas for computing the covariant derivatives of our spherical spline curves. In particular, the proof of Theorem 6 based on the Implicit Function theorem should yield formulas for the derivatives of the function $\sum_i w_i \cdot p_i$ with respect to changes in the values of w_i and of p_i. Given formulas for the first- and second-order derivatives of the spline curve, it should be possible to use iterative methods for finding control points that define "natural splines" which minimize the curvature or the second derivatives of the spline curve. Likewise, we expect that one could minimize energy functions, etc.

Second, it would be useful to extend methods of knot insertion from the Euclidean domain to the spherical domain. In the Euclidean domain, knot insertion is a technique that allows the insertion of additional knots and control points without altering the spline curve. This is useful for several purposes including efficiently rendering curves to any desired level of detail and for optimizing trajectories specified with splines (see, e.g., [27] for the latter application). The standard knot insertion algorithms all depend strongly on the linearity of Euclidean space, so they are not immediately applicable to spherical spline curves based on spherical weighted averages. It is plausible that techniques similar to those of Brown and Worsey [5] could be used to prove that knot insertion is not possible on the sphere.

Third, it would be nice to prove stronger forms of Theorem 5, particularly forms that would be more useful for spline curves by relaxing the condition that any four successive control points lie in a hemisphere. For instance, we conjecture that if each pair of successive control points of a B-spline curve are separated by a distance of no more than $\pi/2$, then the spherical weighted averages in the definition of the spline curve are always uniquely defined and are local minima according to the second derivative test.

Fourth, there is another, yet unexplored alternative to defining spherical averages, based on the characterization of barycentric coordinates in terms of areas of triangles (or in higher dimensions, simplices). This could be used to give an alternate definition of spherical averages, and it would be interesting to understand how this is related to the spherical averages of the present paper: it is not hard to give examples where they differ. We have not pursued since alternative for two reason: first, the area-based definition would seemingly only allow averages of $d+1$ points in the d-sphere and, second, since the definition based on minimization of square distance seemed more compelling and elegant. Along these lines, there may be some connection Arvo's algorithm [2] for area-uniform sampling of spherical triangles.

Fifth, and perhaps most interesting, it would be desirable to extend the spherical averages to more general manifolds. The definition of an average as the point which minimizes the weighted sum of the squares of the distances from the p_i's could apply to any manifold (or subset of a manifold) in which, for any two given points, there is a shortest geodesic joining the points. It would be nice to have general conditions on manifolds, or the distribution of points p_i on the manifold, which imply that the weighted average is well-defined in this way. Ideally, this would allow the notion of weighted averages to be extended to a broad range of manifolds.

Acknowledgements We thank B. Ravani for comments on a draft of the paper, and especially for the suggestion that we prove convexity properties. The first author thanks S. Rotenberg for useful discussions, and for introducing him to slerping. Finally, we thank the referees for useful suggestions.

References

[1] P. Alfeld, M. Neamtu, and L. L. Shumaker, *Bernstein-Bézier polynomials on spheres and sphere-like surfaces*, Computer Aided Geometric Design, 13 (1996), pp. 333–349.

[2] J. Arvo, *Startified sampling of spherical triangles*, Computer Graphics, 29 (1995), pp. 437–438. SIGGRAPH '95.

[3] M. Baronti, E. Casini, and P. L. Papini, *Centroids, centers and medians: What is the difference?*, Geometriae Dedicata, 68 (1997), pp. 157–168.

[4] A. H. Barr, B. Currin, S. Gabriel, and J. Hughes, *Smooth interpolation of orientations with angular velocity constraints using quaternions*, in Computer Graphics: SIGGRAPH'92 Conference Proceedings, 1992, pp. 313–320.

[5] J. L. Brown and A. J. Worsey, *Problems with defining barycentric coordinates for the sphere*, Mathematical Modelling and Numerical Analysis (Modélisation mathématique et Analysé numérique), 26 (1992), pp. 37–49.

[6] R. M. Clark and R. Thompson, *Statistical comparison of palaeomagnetic directional records from lake sediment*, Geophys. J. R. Astr. Soc., 76 (1984), pp. 337–368.

[7] H. S. M. Coxeter, *Quaternions and reflections*, American Mathematical Monthly, 53 (1946), pp. 136–146.

[8] E. B. Dam, M. Koch, and M. Lillholm, *Quaternions, interpolation and animation*, Tech. Rep. DIKU 98/5, Institute of Computer Science, Univ. of Copenhagen, 1998. http://www.diku.dk/students/myth/quat.html.

[9] T. Duff, *Splines in animation and modeling*, in SIGGRAPH'86 Course Notes on State of the Art in Image Synthesis, ACM, July 1986.

[10] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, Academic Press, Boston, 3rd ed., 1993. Contains chapters by P. Bézier and W. Boehm.

[11] N. I. Fisher and T. Lewis, *A note on spherical splines*, J. R. Statist. Soc. B, 47 (1985), pp. 482–488.

[12] S. Gabriel and K. Kajiya, *Spline interpolation in curved space*, in SIGGRAPH'85 Course Notes on State of the Art Image Synthesis, ACM, 1985, pp. 1–14.

[13] Q. J. Ge and B. Ravani, *Computer aided geometric design of motion interpolants*, Transactions of the ASME: Journal of Mechanical Design, (1994), pp. 756–762. Earlier version in *Adv. in Design Automation*, ASME, DE-Vol32-2, 1991, pp.33-41.

[14] O. Gross, *The rondezvous value of a metric space*, in Advances in Game Theory, Princeton University Press, 1964, pp. 49–53.

[15] J. C. Hart, G. K. Francis, and L. H. Kauffman, *Visualizing quaternion rotation*, ACM Transactions on Graphics, 13 (1994), pp. 256–276.

[16] P. E. Jupp, *Fitting smooth paths to spherical data*, Applied Statistics, 36 (1987), pp. 34–46.

[17] B. Jüttler, *Visualization of moving objects using dual quaternion curves*, Comput. and Graphics, 18 (1994), pp. 315–326.

[18] B. Jüttler and M. G. Wagner, *Computer-aided design with spatial rational B-spline motions*, Journal of Mechanical Design, 118 (1996), pp. 193–201.

[19] M.-J. Kim, M.-S. Kim, and S. Y. Shin, *A general construction scheme for unit quaternion curves with simple high order derivatives*, in Computer Graphics: SIGGRAPH'95 Conference Proceedings, 1985, pp. 369–376.

[20] M.-S. Kim and K.-W. Nam, *Interpolating solid orientations with circular blending quaternion curves*, Computer Aided Design, 27 (1995), pp. 385–398.

[21] ——, *Hermite interpolation of solid orientations with circular blending quaternion curves*, Journal of Visualization and Computer Animation, 7 (1996), pp. 95–110.

[22] J. W. Milnor, *Topology from the Differentiable Viewpoint*, University Press of Virginia, Charlottesville, 1965. Based on notes by D.W. Weaver.

[23] L. Noakes, G. Heinzinger, and B. Paden, *Cubic splines on curved surfaces*, IMA Journal of Mathematical Control and Information, 6 (1989), pp. 465–473.

[24] F. C. Park and B. Ravani, *Smooth invariant interpolation of rotations*, ACM Transactions on Graphics, 16 (1997), pp. 227–295.

[25] R. L. Parker and C. F. Denham, *Interpolation of unit vectors*, Geophysical Journal of the Royal Astronomical Society, 58 (1979), pp. 685–687.

[26] E. Polak, *Optimization: Algorithms and Consistent Approximations*, Applied Mathematical Sciences #124, Springer-Verlag, New York, 1997.

[27] R. Ramamoorthi, C. Ball, and A. H. Barr, *Dynamic splines with constraints for animation*, Tech. Rep. CS-TR-97-03, Dept. of Computer Science, CalTech, 1997.

[28] R. Ramamoorthi and A. H. Barr, *Fast constuction of quaternion splines*, in Computer Graphics: SIGGRAPH'97 Conference Proceedings, 1997, pp. 287–292.

[29] K. S. Roberts, G. Bishop, and S. K. Ganapathy, *Smooth interpolation of rotational matrices*, in Proceedings CVPR'88: Computer Vision and Pattern Recognition, IEEE Computer Science Press, 1988, pp. 724–729.

[30] K. Shoemake, *Animating rotation with quaternion curves*, Computer Graphics, 19 (1985), pp. 245–254. SIGGRAPH'85.

[31] ———, *Quaternion calculus and fast animation*, in SIGGRAPH'87 Course Notes on State of the Art Image Synthesis, ACM, 1987, pp. 101–121.

[32] M. Spivak, *Calculus on Manifolds*, Benjamin/Cummings Publishing, Menlo Park, 1965.

[33] R. Thompson and R. M. Clark, *A robust least-squares Gondwanan apparent polar wander path and the question of paleomagnetic assessment of Gondwanan reconstructions*, Earth and Planetary Science Letters, 57 (1982), pp. 152–158.

[34] G. Wagner, *On means of distances on the surface of a sphere (lower bounds)*, Pacific Journal of Mathematics, 144 (1990), pp. 389–398.

[35] ———, *On means of distances on the surface of a sphere. II (lower bounds)*, Pacific Journal of Mathematics, 154 (1992), pp. 381–396.

[36] W. Wang and B. Joe, *Orientation interpolation in quaternion space using spherical biarcs*, in Graphics Interface'93, 1993, pp. 24–32.

[37] G. S. Watson, *Statistics on Spheres*, J. Wiley & Sons, New York, 1983.

[38] A. Watt and M. Watt, *Advanced Animation and Rendering Techniques*, Addison-Wesley, Reading, Massachusetts, 1992.

[39] R. Wolf, *On the average distance property of spheres in Banach spaces*, Archives of Mathematics, 62 (1994), pp. 338–344.

[40] M. Žefran and V. Kumar, *Planning of smooth motions on $SE(3)$*, in Proc. IEEE International Conference on Robotics and Animation, IEEE, 1996, pp. 121–126.