2003

Designing Smooth Motions of Rigid Objects: Computing Curves in Lie Groups
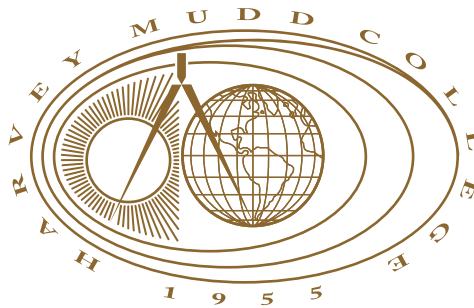
Ross Monet Richardson
Harvey Mudd College

Designing Smooth Motions of Rigid Objects
Computing Curves in Lie Groups

by
Ross Monet Richardson
Weiqing Gu, Advisor

Advisor: _____

Second Reader: _____

(Zachary Dodds, Computer Science)

Abstract

Designing Smooth Motions of Rigid Objects

Computing Curves in Lie Groups

by Ross Monet Richardson

May 2003

Consider the problem of designing the path of a camera in 3D. As we may identify each camera position with a member of the Euclidean motions, $SE(3)$, the problem may be recast mathematically as constructing interpolating curves on the (non-Euclidean) space $SE(3)$.

There exist many ways to formulate this problem, and indeed many solutions. In this thesis we shall examine solutions based on simple geometric constructions, with the goal of discovering well behaved and *computable* solutions. In affine spaces there exist elegant solutions to the problem of curve design, which are collectively known as the techniques of Computer Aided Geometric Design (CAGD). The approach of this thesis will be the generalization of these methods and an examination of computation on matrix Lie groups. In particular, the Lie groups $SO(3)$ and $SE(3)$ will be examined in some detail.

Table of Contents

List of Figures

Acknowledgments

Without the extensive efforts of my advisor, Weiqing Gu, this thesis would never have gotten off the ground, or indeed kept going. I'm extremely grateful to her for her instruction, advice, and constant prodding. I'd also like to thank Zach Dodds for agreeing to read my thesis, and Elizabeth 'Z' Sweedyk for encouraging me to work in this field as well as listening to me gripe when things went wrong.

Finally, I'd like to thank Meghan Powers, for her constant support in my senior year.

Chapter 1

Introduction

1.1 The Problem

In computer animation, it is often desirable to be able to specify the motion of some rigid object in space. For example, if a camera view is to move in some complicated manner, one would like to be able to describe this motion analytically. Moreover, for the purposes of computer animation, one would like to be able to *design* such a motion in a way which is easily controlled and computed.

In the mathematical study of rigid motion, it is shown that the position in space of any rigid object may be equated to an element of the group of Euclidean motions in $\mathbb{R}^3$, $SE(3)$. This group is in fact a *Lie group*, having the structure of both a differentiable manifold as well as that of a group. Hence, we can view the problem of designing the motion of some rigid object as equivalent to the problem of designing a curve in the space $SE(3)$.

The group of Euclidean motions, $SE(3)$, and the group of rotations in $\mathbb{R}^3$, $SO(3)$, are of special interest because of their relation to rigid motion. As such, some research has been done over the past few decades which examines curve construction in these spaces. Research in this area was begun by Shoemake in [15]. Shoemake's work utilized the quaternion representation of $SO(3)$ to construct curves analogous to Bézier curves of CAGD. His idea was to replace linear interpolation with angular interpolation. Others, including [3] made approaches along the same lines. A common limitation of this approach is that the curves arrived at are not

appropriate for real-time computation.

Recent research in the topic has produced a number of new approaches. The focus has ranged from the very theoretical work of Park and Ravani, who construct generalized Bézier curves in Riemannian manifolds in [13], to the more heavily numeric approaches found in [4] and [14]. While these approaches are greatly increased in sophistication, they still are limited by numerical difficulties inherent in their construction.

In this paper we explore a generalization of Bézier curves and spline curves to the setting of compact Lie groups and related spaces. With our attention focused on possibly computing such curves, we disregard the classical analytic notion of polynomial curves and instead generalize the underlying algorithm of Bézier curves, the *de Casteljau* algorithm, to provide a constructive definition for our desired curves. Crouch, Kun, and Leite explore an equivalent approach to curve design in [8] with respect to Bézier curves (though we provide some characterizations of such curves not explored by the preceding authors). The primary novelty of this thesis lies in our work on splines and spline interpolation, which does not seem to appear in the literature. Additionally, we present here a unified approach to CAGD using notions of geodesic interpolation.

The organization of this thesis is as follows: We begin in this chapter with a brief review of Bézier curves to give the flavor of the classical theory of CAGD. We then present the basics of Lie groups and matrix groups, as this is the setting for the majority of the thesis. In chapter 2 we discuss the technical difficulties surrounding one obvious generalization of CAGD onto curved space. In the following chapters, we build up an alternate theory of Bézier curves on curved spaces which is mathematically and computationally tenable. We then introduce a definition of spline curves which is compatible with the Bézier theory thus developed. Finally, we investigate the topics of knot insertion and spline interpolation using these splines.

1.2   Classical Bézier Curves

Computer Aided Geometric Design is concerned with curves and surfaces which are specified by a discrete set of points and are easily computed. In what follows we present the construction of the simplest objects in CAGD, Bézier curves. For a full treatment see [9] and [10].

1.2.1   Multiaffine Maps and Polynomials

We begin with some preliminary definitions about multiaffine functions and polynomials.

Definition 1.2.1 *Let E, F be affine spaces (typically $\mathbb{R}^n$). Let $f : E \to F$. We say that f is an* affine map *if the following holds*

$$f\left(\sum_{i \in I} \lambda_i x_i\right) = \sum_{i \in I} \lambda_i f(x_i),$$

where the set I is finite and the $x_i \in E$. The quantity $\sum_{i \in I} \lambda_i x_i$ is referred to as a barycentric *combination of the points x_i.*

Definition 1.2.2 *Let E, F be affine spaces. A function $f : E^m \to F$ is a* multiaffine map *(specifically, m-affine) if the map is affine in each of its arguments. That is, the map $x \mapsto f(x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_m)$ is an affine map for all i.*

Definition 1.2.3 *We say that the an arbitrary function $f : E^m \to F$ (where E and F are arbitrary sets) is symmetric iff the following holds*

$$f(x_1, \ldots, x_m) = f(x_{\pi(1)}, \ldots, x_{\pi(m)}),$$

where π is a permutation of the set $\{1, \ldots, m\}$.

Definition 1.2.4 *Let E, F be affine spaces. An affine polynomial function of polar degree m is a map $h : E \rightarrow F$ such that there is some symmetric m-affine polynomial $f : E^m \rightarrow F$, called the m-polar form, with*

$$h(x) = f(x, \ldots, x),$$

for all $x \in E$.

The above preliminary definitions allow us to define polynomial curves, which shall be the geometric objects of concern to us.

Definition 1.2.5 *A polynomial curve in polar form of degree m is an affine polynomial map $F : \mathbb{R} \rightarrow \mathcal{E}$ of polar degree m defined by its m-polar form, which is some m-affine map $f : \mathbb{R}^m \rightarrow \mathcal{E}$. Here $\mathcal{E}$ is some affine space of dimension at least 2.*

1.2.2 Polynomial Curves and Control Points

Let f be a symmetric, m-affine map from $\mathbb{R}^m \rightarrow \mathcal{E}$. Choose some affine basis for $\mathbb{R}$, (r, s), where $r < s$. We thus have the following

$$
\begin{aligned}
f(t_1, \ldots, t_m) &= f((1 - \lambda_1)r + \lambda_1 s, t_2, \ldots, t_m) \\
&= (1 - \lambda_1)f(r, t_2, \ldots, t_m) + \lambda_1 f(s, t_2, \ldots, t_m) \\
&= \sum_{k=0}^{m} \sum_{\substack{I \cup J = \{1, \ldots, m\} \\ I \cap J = \emptyset, |J| = k}} \prod_{i \in I}(1 - \lambda_i) \prod_{j \in J} \lambda_j f(\underbrace{r, \ldots, r}_{m-k}, \underbrace{s, \ldots, s}_{k}).
\end{aligned}
$$

The above equalities follow since f is symmetric and m-affine.

We may express the λ_i's as

$$\lambda_i = \frac{t_i - r}{s - r}, 1 \leq i \leq m,$$

to obtain

$$
f(t_1, \ldots, t_m) = \sum_{k=0}^{m} \sum_{\substack{I \cup J = \{1, \ldots, m\} \\ I \cap J = \emptyset, |J| = k}} \prod_{i \in I}\left(\frac{s - t_i}{s - r}\right) \prod_{j \in J}\left(\frac{t_j - r}{s - r}\right) f(\underbrace{r, \ldots, r}_{m-k}, \underbrace{s, \ldots, s}_{k}).
$$

Written in this form it is easy to see that the $m + 1$ points

$$a_k = f(\underbrace{r, \ldots, r}_{m-k}, \underbrace{s, \ldots, s}_{k}),$$

completely determine the m-affine map. The converse is true as well. The points a_k are called *Bézier control points*.

If we set $t_i = t$ for all i, then the coefficients of the control points become

$$\binom{m}{k} \left(\frac{s - t}{s - r} \right)^{m-k} \left(\frac{t - r}{s - r} \right)^{k}.$$

These are polynomials in t which are called the *Bernstein polynomials of degree m over $[r, s]$*, labeled as $B_k^m[r, s](t)$. Observe that the polynomial function associated to f is given by $h(t) = f(t, \ldots, t)$, and so by our discussion is given by

$$h(t) = \sum_{k=0}^{m} \binom{m}{k} \left(\frac{s - t}{s - r} \right)^{m-k} \left(\frac{t - r}{s - r} \right)^{k} f(\underbrace{r, \ldots, r}_{m-k}, \underbrace{s, \ldots, s}_{k}).$$

Definition 1.2.6 *To any $m + 1$ points $\{a_1, \ldots, a_m\}$ in an affine space $\mathcal{E}$ we define the* Bézier curve *determined by these points, $B(t)$, as*

$$B(t) = \sum_{k=0}^{m} B_k^m[r, s](t) a_k.$$

Bézier curves posses many useful geometric properties that make them useful for design purposes. For instance, a Bézier curve lies entirely within the convex polygon determined by the Bézier control points. See [10] for more details.

1.2.3   The de Casteljau Algorithm

The definition of a Bézier curve via a polar form is more than just mathematically convinient. Rather, Bézier curves are quickly and efficiently computable using simple computer arithmetic.

It is clear from the discussion in the previous section that the $m+1$ control points $\{a_1, \ldots, a_m\}$ determine the value of an m-affine map. However, because the map is

multi-affine, we may make a stronger statement. Observe that if $t = (1 - \lambda)r + \lambda s$, and f is a bi-affine map, then

$$f(t, t) = (1 - \lambda)f(r, t) + \lambda f(s, t),$$

and so $f(t, t)$ is an affine combination of the points $f(r, t)$ and $f(s, t)$. Further,

$$f(r, t) = (1 - \lambda)f(r, r) + \lambda f(r, s),$$

and so $f(r, t)$ (similarly $f(s, t)$) is an affine combination of the points $f(r, r) = a_1$ ($f(s, s) = a_3$) and $f(r, s) = a_2$. Hence, we observe that starting from the Bézier control points we may obtain intermediate terms by affine combination, arriving in a finite number of iterations at $f(t, t) = F(t)$, the polynomial curve at an arbitrary parameter point t.

The above process is easily generalized to m-affiine maps. The algorithm described by this process is referred to as the de Casteljau algorithm.

1.3 Lie Groups and Lie Algebras

The following is a minimal introduction to Lie groups and Lie algebras. For a basic introduction see Chapter 1 of [6]. A more complete introduction may be found in [7].

1.3.1 Definitions

Definition 1.3.1 *A Lie group is a group G with a differentiable structure such that the mapping $G \times G \to G$ given by $(x, y) \to xy^{-1}, x, y \in G$, is differentiable.*

Definition 1.3.2 *The tangent space of the identity e of a Lie group G, T_eG, is called the Lie algebra, $\mathfrak{g}$, of the Lie group G when equipped with a certain bilinear operation $[,]$: $T_eG \times T_eG \to T_eG$.*

We shall ignore the bilinear operation referenced in the definition, sufficing to note that such an operation always exists and is unique, and thus every Lie group possesses a unique Lie algebra. This thesis requires only the metric structure on T_eG.

Recall the exponential map is a map from the tangent space of a differentiable manifold to the manifold itself. Further, this map is always defined in some neighborhood of every point. As such, the exponential map provides a useful mapping from the Lie algebra of a group to the Lie group itself. In general, this map is neither injective nor surjective.

Another useful mapping in the theory of Lie groups is the *adjoint* mapping. Consider the following action of $h \in G$ on elements $g \in G$.

$$g \mapsto hgh^{-1}.$$

The differential of this action is denoted by Ad_h and is known as the adjoint mapping.

A related map to the adjoing mapping is the confusingly named adjoint *action*, denoted $\mathrm{ad}A$ ($A \in \mathfrak{g}$), given by

$$\mathrm{ad}A : X \mapsto [A, X], \qquad A, X \in \mathfrak{g},$$

where the reader is reminded that $[\cdot, \cdot]$ is the bilinear form associated with the Lie algebra $\mathfrak{g}$.

1.3.2 Matrix Groups

The group $GL_n(\mathbb{R})$ consisting of non-singular elements of $M_{n \times n}(\mathbb{R})$ has a natural differentiable structure inherited from $\mathbb{R}^{n^2}$, and is thus a Lie group. Many examples of Lie groups may be realized as subgroups of $GL_n(\mathbb{R})$. Further, the exponential map when restricted to matrix groups (inheriting the Euclidean metric from $GL_n(\mathbb{R})$) is the standard matrix exponential defined by

$$\exp A = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots.$$

The Adjoint mapping Ad_A is given by

$$\mathrm{Ad}_A : X \mapsto AXA^{-1}, \qquad X \in \mathfrak{gl}(n, \mathbb{R})$$

in the matrix case.

The following two matrix groups are of use to questions in kinematics.

1.3.3 $SO(3)$: The Group of Rotations in $\mathbb{R}^3$

A rigid rotation is a map which preserves length and angle and fixes the origin. Such rotations in $\mathbb{R}^3$ may be realized as orientation-preserving orthogonal matrices

$$SO(3) = \left\{ A \in M_{3\times 3} | A^T A = I, \det A = 1 \right\}.$$

If we let $A(t) \in SO(3)$ be a curve which passes through I at $t = 0$, then since $A(t)^T A(t) = I$ for all t, we differentiate to obtain

$$
\begin{aligned}
\frac{d}{dt} I|_{t=0} &= \frac{d}{dt}(A(t)^T A(t))|_{t=0} \\
0 &= A'(0)^T A(0) + A(0)^T A'(0) \\
&= H^T + H = 0
\end{aligned}
$$

Here H is the tangent vector $H = A'(0)$. Thus every member of the tangent space $T_e(SO(3)) = \mathfrak{so}(3)$ is skew symmetric. Further, if H is skew-symmetric, we may set $\gamma(t) = \exp(tH)$, which is a curve in $SO(3)$ with $\gamma'(0) = H$. Thus we have the characterization

$$\mathfrak{so}(3) = \left\{ H \in M_{3\times 3} | H = -H^T \right\}.$$

Consider the operator $\hat{\ }$ defined on vectors in $\mathbb{R}^3$ as:

$$\hat{\ } : \omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \mapsto \hat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}.$$

Simple computation shows that

$$\hat{\omega}v = \omega \times v, \qquad \forall v \in \mathbb{R}^3.$$

The elements of $\mathfrak{so}(3)$ may be interpreted physically, such that an element $\hat{\omega}$ represents a rotation about the axis ω with angle $\|\omega\|$. Thus we may view elements in the Lie algebra as generators of rotations. The elements,

$$L_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \qquad L_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \qquad L_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

are thus generators of rotations about the 3 coordinate axes. They also form a basis for $\mathfrak{so}(3)$, and thus we see that any rotation is generated by a linear combination of these rotations.

Finally, $SO(3)$ is a compact Lie group, and thus may be endowed with a bi-invariant metric.

1.3.4 $SE(3)$: The Group of Ridid Motions in $\mathbb{R}^3$

A rigid motion is an orientation preserving map which also preserves length and angle. A fundamental result of differential geometry states that every such motion may be represented as the composition of a rigid rotation and a translation. Hence, the rigid motions of $\mathbb{R}^3$ are elements of the the set $SO(3) \times \mathbb{R}^3$. We may give this set the product differentiable structure, and further we may consider this set as a group $SO(3) \ltimes \mathbb{R}^3$ formed by a semi-direct product. It can be shown that this forms a (non-compact) Lie group.

We may represent $SE(3)$, the group of rigid motions (with the group operation of composition) as a matrix group. Each element of $M \in SE(3)$ corresponds to a rotation and a translation, and hence an element $R \in SO(3)$ and $\vec{v} \in \mathbb{R}^3$. We may

set

$$M = \begin{bmatrix} R & \vec{v} \\ 0 & 1 \end{bmatrix}.$$

It is easily verified that the set of all such matrices form the proper subgroup under matrix multiplication. It is also easy to verify that the Lie algebra $\mathfrak{se}(3)$ is composed of all matrices of the form

$$\begin{bmatrix} H & \vec{v} \\ 0 & 0 \end{bmatrix}, \qquad H \in \mathfrak{so}(3), \vec{v} \in R^3.$$

We may provide a basis for $\mathfrak{se}(3)$ as follows: The basis elements $\{L_1, L_2, L_3\}$ of $\mathfrak{so}(3)$ may also be considered basis elements of $\mathfrak{se}(3)$ corresponding to the pure rotations. Represented in matrix form we have:

$$L_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad L_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad L_3 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Similarly, the standard basis of $\mathbb{R}^3$ extends to a basis of the subspace of pure translations as follows:

$$L_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad L_5 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad L_6 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Chapter 2

A Simple Approach to Curve Design

Our goal is the generalization of the techniques of Computer Aided Geometric Design to curved spaces. In this chapter we discuss an approach to generalizing CAGD by utilizing the local affine structure of the coordinate neighborhoods of our curved space.

2.1 Local Parameterization Method

Consider some curve $c : (-\epsilon, \epsilon) \to M$, where M^n is a Riemannian manifold. For each t on which c is defined, let $(\mathbf{x}, U)$ denote some chart in the differentiable structures of M which contains the point $c(t)$. Then the map $b = \mathbf{x} \circ c : (-\epsilon, \epsilon) \to \mathbb{R}^n$ is the continuous image of $c(t)$ in the affine space $\mathbb{R}^n$. Hence, $\mathbf{x}^{-1} \circ b = c$. Thus, we observe that designing the curve c is equivalent to the problem of designing the curve b locally in an affine space.

Example 2.1.1 *Consider the matrix Lie group $SO(3)$. In a neighborhood about the identity, the exponential mapping provides a diffeomorphism from $\mathfrak{so}(3)$ onto $SO(3)$. Thus, a curve $A(t) \in \mathfrak{so}(3)$ is mapped to a curve $e^{A(t)}$ via this mapping. This is attractive from a design and computational view point. As $\mathfrak{so}(3)$ is a (finite dimentional) vector space, and thus an affine space, we can utilize the techniques of CAGD in this setting[1].*

In general, matrix groups provide this sort of computational structure. The Lie algebra of the group is a finite dimensional vector space of matrices. The exponential mapping may be computed numerically for any such matrix, and in the case of

---

[1]See the appendix for an account of CAGD in a vector space composed of matrices.
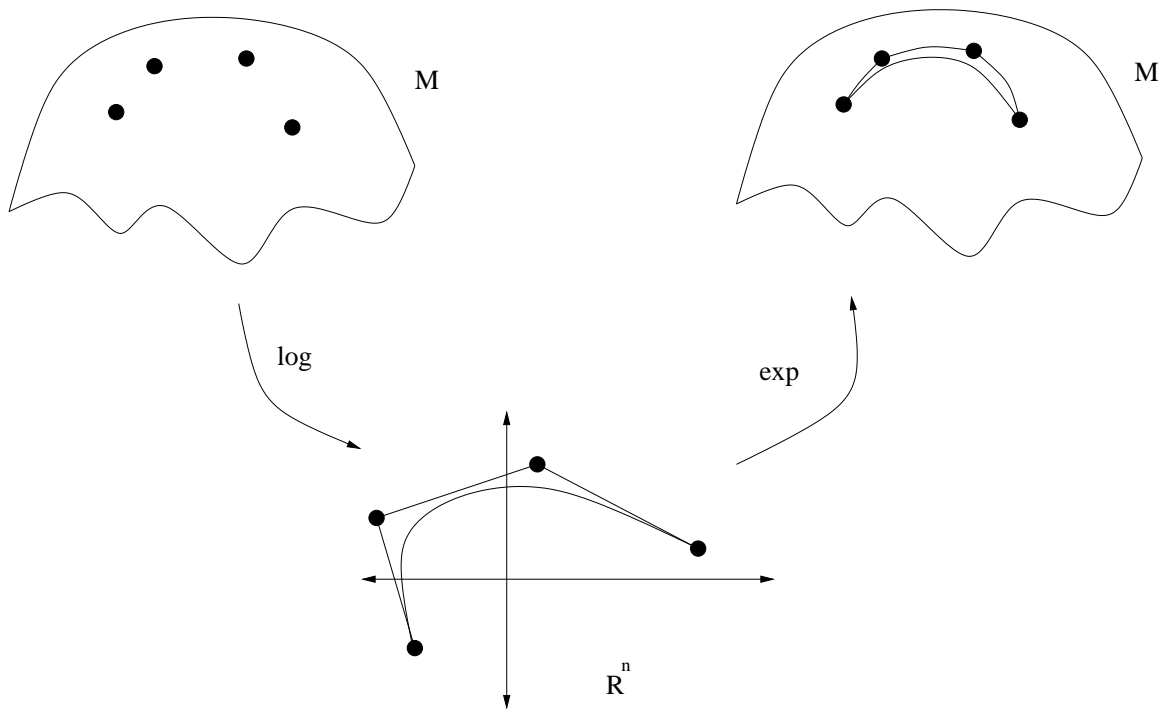
Figure 2.1: Mapping neighborhoods of a manifold under the logarithm mapping (inverse to the exponential mapping) maps the control points into the affine space $\mathbb{R}^n$ on which we may do CAGD.

some groups such as $SO(3)$ and $SE(3)$ exact closed forms for the exponential map allow for efficient, high-precision computation. Moreover, on matrix groups the existence of the logarithm mapping, which is inverse to the exponential mapping, allows for control points on the group to be mapped to pre-images in the Lie algebra. Thus, curve design on such a group follows the program outlined in figure 2.1.

2.2 Difficulties

In practice, the program for curve design outlined in the last section is rife with many technical difficulties. The choice of the exponential map, while advantageous from a computational viewpoint, is problematic. First, in general the ex-

ponential map is neither surjective nor injective. The space $SO(3)$ provides an example of a Lie group on which the exponential map is surjective but not injective (for $\omega \in R^3$, we have $\exp(\hat{\omega}) = \exp(-\hat{\omega})$, expressing the fact that rotations of $\pm\pi$ about some axis result in the same rotation). $SL(n)$, the group of matrices with determinant 1, provides an example of a group for which the exponential map is not surjective.

A related problem is the multivalued nature of the logarithm mapping. The choice of preimage of some group element under the exponential mapping will affect the resultant curve. For example, if an element of $SO(3)$ has $\hat{\omega}$ for a preimage under the exponential map, it will also have

$$\frac{\|\omega\| + 2n\pi}{\|\omega\|}\hat{\omega}, \quad n \in \mathbb{Z}$$

as a preimage.

The most troubling difficulty of this program, however, is the dependence on a particular parameterization of the manifold for constructing curves. The intrinsic geometry of the manifold is thus not used in the construction. It is known that for spaces of non-zero sectional curvature, the exponential mapping is not in general a local isometry. Intuitively, this expresses the fact that the geometry of the curve constructed in affine space is not preserved when it is mapped onto the manifold via the exponential map. Moreover, even in spaces of constant sectional curvature, the geometry of the curve in the affine space becomes "warped" as a function of the distance from the curve to the origin of tangent space (or Lie algebra).

To see this, let p be a point in M. Define f to be the parameterized surface

$$f(t, s) = \exp_p tv(s), \qquad 0 \le t \le 1, \qquad -\epsilon \le s \le \epsilon,$$

where $v(s) \in T_pM$ with $v(0) = v$ and $v'(0) = w \in T_v(T_pM) \approx T_pM$. If we choose w such that $\langle v, w \rangle = 0$, and restrict $|v| = |w| = 1$, then we obtain the following formula:
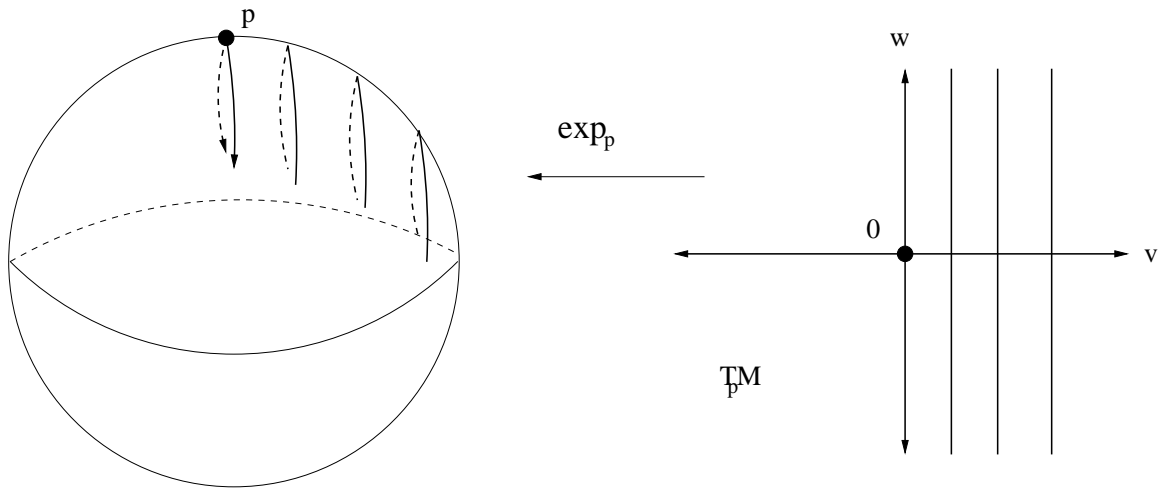
Figure 2.2: An illustration of how translates of a line segment through the origin of T_pM "warp" under the exponential map. The deviation of these segments from geodesics increases with the distance from the origin.

$$\left|\frac{\partial f}{\partial s}(t,0)\right| = t - \frac{1}{6}K(p,\sigma)t^3 + R(t), \qquad \lim_{t\to 0}\frac{R(t)}{t^3} = 0,$$

about $t = 0$, where σ is the subspace spanned by v and w in T_pM and $K(p,\sigma)$ is the sectional curvature at p of the subspace σ. (see [6] for proof).

This provides a measure of how fast the geodesics from p spread apart. Alternately, consider the two lines spanned by v and w in T_pM. If we translate the line spanned by w along v so that it remains parallel to the original line spanned by w, then we may view $\frac{\partial f}{\partial s}(t,0)$ to be the degree of "warping" of this line in a neighborhood of the line spanned by v (see figure 2.2). Hence, the image of a curve in the tangent space of some manifold (under the exponential map) will in general depend on the basepoint of the exponential map.

Hence, any generalization of CAGD to curved spaces which is based on specific parameterizations of a manifold will meet with the difficulties outlined above. If we seek to avoid these difficulties, then, we must appeal to a construction which

relies only on the intrinsic geometry of the manifold.

Chapter 3

Curve Design on $SO(3)$

In this chapter we discuss a different method than that of the last chapter for promoting the techniques of CAGD onto the group $SO(3)$. Instead of relying on the use of actual affine spaces, and thus the resulting dependence on specific parameterizations, we shall instead seek a generalization of the notion of linear interpolation on curved spaces and use this notion to extend the geometric constructions of CAGD. In this way we shall construct curves defined using only the *intrinsic* geometry of the space. This method shall prove more tenable for our purposes, and thus shall serve to introduce the techniques discussed in the remainder of this thesis.

3.1   Interpolation

3.1.1   Riemannian Manifolds

In [13] Park and Ravani seek to extend the notion of Bézier curves in $\mathbb{R}^n$ to the more general setting of Riemannian manifolds. Noting that Bézier's original construction of these curves involved the notion of an osculating plane, which in turn requires one to view a manifold as embedded in some ambient Euclidean space, Park and Ravani were led to instead consider the de Casteljau algorithm to define Bézier curves as such a definition uses only the intrinsic geometry of the manifold.

The de Casteljau algorithm relies on the notion of linear interpolation in Euclidean space. The natural generalization of straight lines in a manifold is the *(arclength) minimizing geodesic*. If there exists a unique minimizing geodesic between

two points in a Riemannian manifold, then interpolation between these points be-
comes well-defined. Hence, the generalization of Park and Ravani prescribes the
use of exactly this manner of interpolation. This provides a complete mathematical
description of how one generates generalized Bézier curves in a Riemannian man-
ifold if unique minimizing geodesics exist between the points to be interpolated.

The existence of minimizing geodesics between two points, however, is not
guaranteed in general for a Riemannian manifold (see [6] for a good discussion).
Park and Ravani restrict their attention to *complete* manifolds, a class of Riemannian
manifolds for which such geodesics always exist. The uniqueness of minimizing
geodesics, however, is still not guaranteed.

3.1.2 Lie groups and $SO(3)$

Let us consider the question of interpolation on the Lie group $SO(3)$. As $SO(3)$
is compact, it is complete, and so we have the existence of minimizing geodesics
between any two points by connectivity. We also require uniqueness of such a
geodesic if we are to define (geodesic) interpolation. To answer this question, we
examine the exponential map at the identity e. If we have an element $\hat{\omega} \in \mathfrak{so}(3)$
then we have the following analytic expression (known as the *Rodriguez formula*)
for its image under the exponential map,

$$\exp(\hat{\omega}) = I + \frac{\sin\|\omega\|}{\|\omega\|}\hat{\omega} + \frac{1 - \cos\|\omega\|}{\|\omega\|^2}\hat{\omega}^2. \tag{3.1}$$

For $R \in SO(3)$ with $\mathrm{Tr}(R) \neq -1$, the inverse *logarithm* map is given by the
formula,

$$\log(R) = \frac{\phi}{2\sin\phi}(R - R^T), \tag{3.2}$$

where ϕ is such that $1 + 2\cos\phi = \mathrm{Tr}(R)$ and $|\phi| < \pi$. The constraint $\mathrm{Tr}(R) \neq -1$
corresponds to the geometric fact that rotations of $\pm\pi$ degrees about some axis are

equivalent.

Thus, we see that if we restrict the exponential map at the identity to the ball $B(0, \pi)$ of radius less than π centered around the origin then the exponential map is in fact an embedding. Hence, for every point $p \in U = exp_e(B(0, \pi))$ there exists a unique minimizing geodesic in $SO(3)$ with endpoints e and p. As such, we may define interpolation between the identity and point in U.

Definition 3.1.1 *The $\lambda-$interpolant, interp$_\lambda(R)$, from the identity to a point $R = \exp_e(\hat{\omega}) \in U$ is the unique point given by*

$$\text{interp}_\lambda(R) = \exp_e(\lambda \hat{\omega}) \qquad 0 \leq \lambda \leq 1. \tag{3.3}$$

It is then a simple matter to extend the notion of interpolation to any points which are sufficiently close in $SO(3)$.

Definition 3.1.2 *The $\lambda-$interpolant, interp$_\lambda(A, B)$ of $A, B \in SO(3)$ such that $A^{-1}B \in U$ is given by*

$$\text{interp}_\lambda(A, B) = A \cdot \text{interp}_\lambda(A^{-1}B). \tag{3.4}$$

Remark 3.1.3 *The definitions given here may be viewed as interpolating along the one-parameter subgroups of $SO(3)$ and their translates. The conditions on the elements A, B enforce the geometric condition that they are not related by a rotation of $\pm\pi$ about some common axis.*

3.2 de Casteljau's Algorithm on $SO(3)$

With an adequate notion of interpolation, we are ready to construct an algorithm on $SO(3)$ analogous to de Casteljau's algorithm.

3.2.1 The Algorithm

Algorithm 3.2.1 (Modified de Casteljau on $SO(3)$)

Input: A sequence of distinct elements $\{R_0, \ldots, R_n\}$ in $SO(3)$ which obey the condition $R_i^{-1} R_{i+1} \in U, \quad 0 \leq i < n$ and real numbers r, s, t with $r \leq t \leq s$ and $r \neq s$.

Initialization: Set $b_{i,0} = R_i, \quad 0 \leq i \leq n$, and put $\lambda = (t - r)/(s - r)$.

Iteration: For stage j (proceeding from $j = 1$ to $j = n$) compute the $b_{i,j}$ using the formula

$$b_{i,j} = \text{interp}_\lambda(b_{i,j-1}, b_{i+1,j-1}), \qquad 0 \leq i \leq n - j. \tag{3.5}$$

Output: Return the point $b_{0,n}$.

In other words, we have defined a curve which is valued at $b_{0,n}$ constructed as above with parameter value t.

Remark 3.2.2 *The observant reader will note that at the iteration step in the above algorithm, there remains the subtle issue of whether a $\lambda-$interpolant exists for all consecutive points $b_{i,j}, b_{i+1,j}$. The conditions on the input points certainly guarantee the existence of such interpolants for $j = 1$, but it is not clear as to whether $b_{i,j}^{-1} b_{i+1,j} \in U$, as is necessary for interpolation. We shall see in the following chapter that this condition does indeed hold, making the above algorithm well defined.*

3.3 Geometry of $SO(3)$

One may introduce many metrics on $SO(3)$ corresponding to physical or other desirable properties. For robotics applications, it is often the case that some actuators move more easily than others. Thus, we might assign different weights to the coordinate axes. In the basis $\{L_1, L_2, L_3\}$ of $\mathfrak{so}(3)$, one obtains a metric that looks

like

$$\begin{bmatrix} w_x & 0 & 0 \\ 0 & w_y & 0 \\ 0 & 0 & w_z \end{bmatrix}$$

for suitable chosen weights w_x, w_y, w_z.

For efficient computational purposes, however, there is a clear choice of metric. Consider the following inner product defined by setting

$$\langle X, Y \rangle = -\operatorname{Tr}(X^T Y), \qquad X, Y \in \mathfrak{so}(3), \tag{3.6}$$

and propagating it to the entire group by left translation. It is readily verified that this is in fact a bi-invariant metric on $SO(3)$. Further, one may show that this metric satisfies:

$$\langle \operatorname{Ad}_A X, \operatorname{Ad}_A Y \rangle = \langle X, Y \rangle, \qquad \forall A \in SO(3).$$

Thus, we may say that such a metric is Ad-invariant. Indeed, this characterizes all Ad-invariant metrics on $SO(3)$ up to an arbitrary constant.

By assuming the matrix exponential mapping (in the case of $SO(3)$ given by the Rodriguez formula) corresponds to the geometric exponential map, we have actually assumed a metric structure on $SO(3)$. Specifically, we have assumed the structure inherited from $GL(n)$ thought of as the manifold $\mathbb{R}^{n^2}$ with standard Euclidean structure. This metric is in fact an Ad-invariant metric, and so it takes the general form of equation 3.6.

Note that under this metric we may compute:

$$\langle L_i, L_i \rangle = 2, \qquad i = 1, 2, 3.$$

Thus, if we desire this metric to agree with the Euclidean metric on $\mathbb{R}^3$ (that is, we desire $\|\hat{\omega}\| = \|\omega\|, \omega \in \mathbb{R}^3$), then we need rescale the above metric. Thus, we shall understand to use the following metric on $SO(3)$ unless otherwise specified:

$$\langle X, Y \rangle = -\frac{1}{2} \operatorname{Tr}(X^T Y), \qquad X, Y \in \mathfrak{so}(3). \tag{3.7}$$

We shall see in the following chapter how the notions of bi-invariance and Ad-invariance are critical to certain geometric properties we desire in Bézier and spline curves.

Chapter 4

Bézier and Spline Curves on Compact Lie Groups

In this chapter we shall define Bézier and spline curves on compact Lie groups, following the methodology of the previous chapter. In particular, we generalize the *de Casteljau* and *de Boor* algorithm in this setting, and use these algorithms to provide a constructive definition of such curves. Additionally, we provide some analysis of these curves, especially in relation to properties produced via the classical theory.

4.1 Bézier Curves on Compact Lie Groups

In the previous chapter we introduced a modification of the de Casteljau algorithm using the notion of *geodesic* interpolation. Consider the following more general setting. Let G be a compact Lie group (with associated Lie algebra $\mathfrak{g}$) equipped with a bi-invariant metric. As some open neighborhood about the identity, e, is diffeomorphic to a ball centered about $T_e(G)$, define $U = \exp_e(B(0, r))$, where r is the largest radius such that $\exp_e(B(0, r))$ is an embedding.

Our notion of geodesic interpolation is easily generalized:

Definition 4.1.1 *The $\lambda-$interpolant*, $\operatorname{interp}_\lambda(g_1, g_2)$ *of the points* $g_1, g_2 \in G$ *is defined as follows if* $g_1^{-1} g_2 \in U$:

Let $V \in B(0, r) \subset T_e(G)$ *be the unique element such that*

$$g_1^{-1} g_2 = \exp_e(V).$$

Then

$$\mathrm{interp}_\lambda(g_1, g_2) = g_1 \exp_e(\lambda V). \tag{4.1}$$

We then have the following algorithm:

Algorithm 4.1.2 (Modified de Casteljau on compact Lie groups)

Input: A sequence of distinct elements $\{g_0, \ldots, g_n\}$ (called *control points*) in G which obey the condition $g_i^{-1} g_{i+1} \in U$, $\quad 0 \le i < n$, and real numbers r, s, t with $r \le t \le s$ and $r \ne s$.

Initialization: Set $b_{i,0} = g_i$, $\quad 0 \le i \le n$, and put $\lambda = (t - r)/(s - r)$.

Iteration: For stage j (proceeding from $j = 1$ to $j = n$) compute the $b_{i,j}$ using the formula

$$b_{i,j} = \mathrm{interp}_\lambda(b_{i,j-1}, b_{i+1,j-1}), \qquad 0 \le i \le n - j. \tag{4.2}$$

Output: Return the point $b_{0,n}$.

Proposition 4.1.3 *Algorithm 4.1.2 is well defined.*

Proof. We need only demonstrate that if the $\{b_{i,j}\}_i$, j fixed, satisfy the relationship

$$b_{i,j}^{-1} b_{i+1,j} \in U$$

then the $\{b_{i,j+1}\}_i$ satisfy

$$b_{i,j+1}^{-1} b_{i+1,j+1} \in U.$$

For points $g_1, g_2 \in G$, set the function $d(g_1, g_2)$ to be the infimum of the arclength of all piecewise differentiable curves joining g_1 to g_2. This function is a metric on the path components of G (see [6]).

The condition that the point $g_i^{-1} g_{i+1} \in U$ is exactly the condition that the points g_i and g_{i+1} are connected by a geodesic of length at most R, where R is fixed for

G. Hence, it is sufficient to demonstrate that two points a and b have distance less than R in the metric d if they are to satisfy $a^{-1}b \in U$.

Given points $a, b, c \in G$ with $a^{-1}b \in U$ and $b^{-1}c \in U$, we shall demonstrate that $\operatorname{interp}_\lambda(a,b)^{-1}\operatorname{interp}_\lambda(b,c) \in U$.

Examine $d(\operatorname{interp}_\lambda(a,b), b)$. By definition, there exists a geodesic from a to b of length $l_1 < R$ on which $\operatorname{interp}_\lambda(a,b)$ lies. The subpath of the geodesic from $\operatorname{interp}_\lambda(a,b)$ to b is of length $(1-\lambda)l_1$. Thus, $d(\operatorname{interp}_\lambda(a,b), b) \leq (1-\lambda)l_1 \leq (1-\lambda)R$. Similarly, $d(b, \operatorname{interp}_\lambda(b,c)) \leq \lambda R$. By the triangle inequality, we have

$$
\begin{aligned}
d(\operatorname{interp}_\lambda(a,b), \operatorname{interp}_\lambda(b,c)) &\leq d(\operatorname{interp}_\lambda(b,c), b) + d(b, \operatorname{interp}_\lambda(b,c)) \\
&\leq (1-\lambda)R + \lambda R \\
&= R
\end{aligned}
$$

Thus, our result follows. $\qquad\square$

Definition 4.1.4 *We call curves generated by algorithm 4.1.2* Bézier curves *on the group G. It is clear that in Euclidean space, the generalized curves agree with traditional Bézier curves.*

Proposition 4.1.5 *Bézier curves on G are smooth.*

Proof. As the constructed curve is the finite composition of smooth operations, the resultant curve is thus smooth. $\qquad\square$

4.2 Properties of Bézier Curves on Compact Lie Groups

In this section we examine control properties of Bézier curves on G which are related to the standard properties of classical Bézier curves.

Proposition 4.2.1 (Left-invariance) *If $\gamma(t)$ is the Bézier curve on G generated on points $\{g_0, \ldots, g_n\}$ and $h \in G$, then $h\gamma(t)$ is the curve generated on points $\{hg_0, \ldots, hg_n\}$.*

Proof. It suffices to show that $\operatorname{interp}_\lambda(g_i, g_{i+1})$ is left-invariant in the sense that

$$h \operatorname{interp}_\lambda(g_i, g_{i+1}) = \operatorname{interp}_\lambda(hg_i, hg_{i+1}).$$

$$
\begin{aligned}
\operatorname{interp}_\lambda(hg_i, hg_{i+1}) &= hg_i \exp(\lambda \log((hg_i)^{-1}hg_{i+1})) \\
&= hg_i \exp(\lambda \log(g_i^{-1}h^{-1}hg_{i+1})) \\
&= hg_i \exp(\lambda \log(g_i^{-1}g_{i+1})) \\
&= h \operatorname{interp}_\lambda(g_i, g_{i+1})
\end{aligned}
$$

$\square$

Proposition 4.2.2 (Invariance under affine parameter change) *If the Bézier curve on G, $\gamma(t)$, generated on points $\{g_0, \ldots, g_n\}$ is parameterized on $[r, s]$, then $\gamma(t)$ agrees with $\gamma'((t-r)/(s-r))$, where $\gamma'(t)$ is the curve generated on the same points and parameterized on $[0, 1]$.*

Proof. This follows simply from how the parameter λ is generated in algorithm 4.1.2. $\square$

Proposition 4.2.3 (Symmetry) *If $\gamma(t)$ (parameterized on $[r, s]$) is the Bézier curve on G generated on points $\{g_0, \ldots, g_n\}$, then the Bézier curve on G generated on points $\{g_n, \ldots, g_0\}$ parameterized by $[r, s]$ is $\gamma(r + s - t)$.*

Proof. One need only observe that the geodesics between adjacent points are unique. Hence, it is clear that

$$\operatorname{interp}_\lambda(a, b) = \operatorname{interp}_{1-\lambda}(b, a).$$

Thus, at each iteration of algorithm 4.1.2 the same points are generated, and thus the resultant curves agree for each λ. $\square$

Proposition 4.2.4 (Geodesic Precision) *If the control points lie sequentially on a geodesic such that the geodesic is minimizing between neighboring points, then the resulting curve lies on the geodesic.*

Proof. As the geodesic is minimizing between neighboring points, the interpolant of any two points will thus lie on the geodesic. The points produced by the first round of interpolation will lie in a configuration which satisfies the conditions of the proposition (by a proof similar to proposition 4.1.3), and so we may iterate this process. □

Note that in the above proposition the conditions on the configuration of points is very strict. In the general manifold setting, it is possible to have points lie on a geodesic such that the subarc of the geodesic between them is not length minimizing (the cylinder is a tradition example of this). It is not known, at least to this author, if such geodesics exist in compact Lie groups. If not, the requirements of the above propositions might be simplified significantly. In any case, the contrast between the above proposition and the linear interpolation property of Bézier curves in affine space shows the complications of curved geometry.

Proposition 4.2.5 *Let G be a Lie group. If $g \in G$ is in U implies that $g_0 g g_0^{-1} \in U$ for all $g_0 \in G$, then geodesic interpolation is right-invariant.*

Proof.

Lemma 4.2.6 *Let $A \in G$ and $V \in \mathfrak{g}$ such that $\exp(V) \in U$. Then,*

$$\log(A \exp(V) A^{-1}) = \mathrm{Ad}_A \log(V).$$

Proof. It is true that $A \exp(V) A^{-1} = \exp(\mathrm{Ad}_A V)$ (see for example [16]). Hence, as $A \exp(V) A^{-1} \in U$ by the hypothesis of the proposition, the logarithm mapping is

well defined on this element. Application of $\log$ to both sides yields the desired equality.

□

Let g_i and g_{i+1} satisfy $g_i^{-1}g_{i+1} \in U$ (hence there is a unique length-minimizing geodesic between them). Then, if $h \in G$, we have the following:

$$
\begin{aligned}
\text{interp}_\lambda(g_i h, g_{i+1} h) &= g_i h \exp(\lambda \log((g_i h)^{-1}(g_{i+1} h))) \\
&= g_i h \exp(\lambda \log(h^{-1}(g_i^{-1} g_{i+1})h))) \\
\text{(by lemma 4.2.6)} &= g_i h \exp(\lambda \text{Ad}_{h^{-1}} \log(g_i^{-1} g_{i+1}))) \\
&= g_i h h^{-1} \exp(\lambda \log(g_i^{-1} g_{i+1})))h \\
&= g_i \exp(\lambda \log(g_i^{-1} g_{i+1})))h \\
&= \text{interp}_\lambda(g_i, g_{i+1})h
\end{aligned}
$$

□

Thus we obtain the immediate corollary.

Corollary 4.2.7 *Let G be a Lie group with an Ad-invariant metric. Then the de Casteljau algorithm produces curves which are bi-invariant.*

Proof. This is a simple combination of propositions 4.2.1 and 4.2.5, since an Ad-invariant metric satisfies the premises for 4.2.5.

□

We note that, in particular, bi-invariant metrics are Ad-invariant. Thus, as every compact Lie group may be endowed with a bi-invariant metric, such a Lie group will allow for bi-invariant Bézier curves on G according to our construction.

For the purposes of further analysis, it is useful to provide an alternate (though equivalent) characterization of Bézier curves on compact Lie groups. The remainder of this section is an exposition of the work of Crouch et. al. in [8].

Given distinct points $g_0, \ldots, g_n \in G$, define $V_k^1, k = 0, \ldots, n-1$ by:

$$x_{k+1} = g_k \exp(V_k^1), \qquad k = 0, \ldots, n-1. \tag{4.3}$$

Further, for every $t \in [0,1]$ and $k = 0, \ldots, n-1$, define the elements of the Lie algebra $V_k^j(t)$ by:

$$\exp(V_k^j(t)) = \exp((1-t)V_k^{j-1}(t)) \exp(tV_{k+1}^{j-1}(t)), \qquad j \geq 2. \qquad (4.4)$$

Furthermore, for every $t \in [0,1]$ we define a sequence of points in G by the following recurrance:

$$
\begin{aligned}
p_0(t) &= g_0 \\
p_k(t) &= p_{k-1}(t) \exp(tV_0^k(t)), \qquad k \geq 1.
\end{aligned}
$$

The expression when $k = n$ gives

$$p_n(t) = g_0 \exp(tV_0^1) \exp(tV_0^2(t)) \cdots \exp(tV_0^{n-1}(t)) \exp(tV_0^n(t)). \qquad (4.5)$$

It will turn out that $p_n(t)$ is exactly the Bézier curve defined on the points $g_0, \ldots, g_n$. The following propositions give a characterization of the derivative of such a curve. The mechanics and statement of such results assume our group is a matrix group. For compact Lie groups, this offers no difficulty as we may consider such groups represented as matrix groups.

Proposition 4.2.8 *The derivatives of the polynomial curve $t \to p_n(t) \in G$ defined by (4.5) satisfy the following boundary conditions:*

$$\frac{d}{dt}p_n(t)\bigg|_{t=0} = ng_0 V_0^1, \qquad \frac{d}{dt}p_n(t)\bigg|_{t=1} = ng_n V_{n-1}^1.$$

Thus, we may characterize the derivative at the endpoints of the Bézier curve.

A characterization of the second covariant derivative is also available via the following proposition.

Proposition 4.2.9 *If $t \to p_n(t)$ is the polynomial curve in G defined in (4.5), then:*

$$\left.\frac{D^2}{dt^2}p_n(t)\right|_{t=0} = \frac{n!}{(n-2)!}g_0\Upsilon_0(V_1^1 - V_0^1),$$

$$\left.\frac{D^2}{dt^2}p_n(t)\right|_{t=1} = \frac{n!}{(n-2)!}g_n\Upsilon_1(V_{n-1}^1 - V_{n-2}^1),$$

where Υ_0 and Υ_1 are respectively the inverses of the operators

$$\int_0^1 \exp(u\,\mathrm{ad}V_0^1)du \qquad and \qquad \int_0^1 \exp(-u\,\mathrm{ad}V_{n-1}^1)du.$$

Finally, for completeness, we demonstrate that the curve defined by $t \to p_n(t)$ is equivalent to the Bézier curve defined on the points $g_0, \ldots, g_n \in G$. In the following proposition we relate the V_i^j to points produced by the de Casteljau algorithm.

Proposition 4.2.10 *Let the points $\{g_0, \ldots, g_n\} \in G$ be given such that $g_i^{-1}g_{i+1} \in U$, and let $t \in [0,1]$. Generate the points $b_{a,b}(t), b = 0, \ldots, n, a = 0, \ldots, n-b$, according to algorithm 4.1.2 on these points. Similarly, generate the Lie algebra elements $V_i^j(t)$ according to equations 4.3 and 4.4 on these points. Then we have the following:*

$$\exp(V_i^j(t)) = b_{i,j}(t)^{-1}b_{i+1,j}(t). \tag{4.6}$$

Proof. For $i = 0$ this is readily verified. Assume equation 4.6 holds for all $i = 0, \ldots, k-1$, and we shall proceed by induction. From equation 4.4 we have that

$$
\begin{aligned}
\exp(V_i^k(t)) &= \exp((1-t)V_i^{k-1}(t))\exp(tV_{i+1}^{k-1}(t)) \\
\text{(induction hypothesis)} &= \exp((1-t)\log(b_{i,k-1}^{-1}b_{i+1,k-1}))\exp(t\log(b_{i+1,k-1}^{-1}b_{i+2,k-1})) \\
&= \exp(-t\log(b_{i,k-1}^{-1}b_{i+1,k-1}))b_{i,k-1}^{-1}b_{i+1,k-1}\exp(t\log(b_{i+1,k-1}^{-1}b_{i+2,k-1})) \\
&= b_{i,k}^{-1}b_{i+1,k}.
\end{aligned}
$$

Thus the proof follows by induction. $\square$

We thus obtain the immediate result showing equality of the two constructions.

Corollary 4.2.11 *The Bézier curve defined on points $\{g_0, \ldots, g_n\} \in G$ according to algorithm 4.1.2 which is parameterized on $[0,1]$ agrees with the curve $t \to p_n(t)$ defined by equation 4.5.*

Proof. The above corollary implies that

$$p_i(t) = b_{0,i}(t).$$

Setting $i = n$ furnishes the result. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

4.3 Spline Curves on Compact Lie Groups

In the prior sections we showed how the algorithmic construction of Bézier curves is naturally generalized to compact Lie groups via the de Casteljau algorithm, with the resulting curves sharing many properties with affine Bézier curves. In this section we offer a constructive definition of spline curves using the de Boor algorithm.

Definition 4.3.1 *We define a* knot sequence *to be a nondecreasing sequence $\{u_k\}_{k\in\mathbb{Z}}$ with $u_k \in \mathbb{R}$ such that every u_k (knot) has finitely many occurrence's. A knot u_k in a knot sequence has* multiplicity $n \geq 1$ *iff it occurs exactly n times. For any natural number $m \geq 1$, a knot sequence has* degree [1] m *iff every knot has multiplicity at most m. A knot u_k of multiplicity m is a* discontinuity *(knot). A knot of multiplicity 1 is a* simple *knot. A knot sequence is uniform iff $u_{k+1} = u_k + h$ for some fixed $h \in \mathbb{R}^+$. Every knot in such a sequence is thus simple.*

Definition 4.3.2 *Given a Lie group G, there exists some maximum radius $r \in \mathbb{R}$ such that $\exp_e |_{B(0,r)}$ is a diffeomorphism (here $\exp_e$ denotes the exponential map at the identity). A* control point sequence *is a sequence $\{d_k\}_{k\in\mathbb{Z}}$ of points $d_k \in G$ such that*

$$d_k^{-1} d_{k+1} \in \exp_e(B(0,r)) = U, \qquad \forall k.$$

---

[1] Also referred to as *degree of multiplicity*.

Spline curves in affine space are computed using the *de Boor* algorithm, a modification of the *de Casteljau* algorithm. We shall proceed to define *spline curves on G* to be those curves produced by the following generalization of the *de Boor* algorithm.

Algorithm 4.3.3 (Modified de Boor on compact Lie groups)

Input: A control point sequence $\{d_k\}_{k \in \mathbb{Z}}$ with $d_k \in G$.

A knot sequence $\{u_i\}$ corresponding to the control points.

A natural number $m \geq 1$ specifying the degree of the sequence and a parameter t.

Initialization: Set $I = \max \{k | u_k \leq t < u_{k+1}\}$.

If $t = u_I$ then set $r := \text{multiplicity}(u_I)$, else $r := 0$.

Set $d_{i,0} := d_i - 1, \qquad I - m + 1 \leq i \leq I + 1 - r$.

Iteration: For stage j (proceeding from $j = 1$ to $j = m - r$) compute the $d_{i,j}$ using the formula

$$d_{i,j} = \text{interp}_{\lambda_i}(d_{i-1,j-1}, d_{i,j-1}) \qquad I - m + j + 1 \leq i \leq I + 1 - r. \qquad (4.7)$$

Where $\lambda_i = \left(\frac{t - u_{i-1}}{u_{m+i-j} - u_{i-1}} \right)$.

Output: Return the point $d_{I+1-r, m-r}$.

Remark 4.3.4 *Note that in the above definitions we have constructed an analogue to* Open B-spline curves, *those defined on an infinite number of control points. We might also choose to construct generalized versions of finite and cyclic B-spline curves, though for the sake of simplicity of exposition we shall not detail such curves herein. The interested reader is directed to chapter 6 of [10], where such curves are discussed at length. The generalizations to the compact Lie group setting are clear.*

We delay the analysis of these curves to chapter 6, where we may discuss the parallel topic of knot insertion and address some important related geometric issues.

Chapter 5

$SE(3)$ and Non-Compact Lie Groups

In this chapter we shall examine methods of lifting the techniques of the previous chapters onto the group of Euclidean motions, $SE(3)$. Our exposition in this regard is a recapitulation of the work of Altafini [1]. We shall also examine the possible generalizations of these methods to more general Lie groups, as well as discuss the limitations of these techniques.

5.1 The Problem of Bi-Invariance on $SE(3)$

In the previous chapter the setting for our curve construction techniques was the *compact* Lie groups. The group of Eucliean motions, however, is non-compact. Indeed, as a differentiable manifold we may represent the group as $SE(3) = SO(3) \times \mathbb{R}^3$, and $\mathbb{R}^3$ is not compact.

The prior restriction to compact Lie groups allowed us to assume our group carried a bi-invariant metric, which in turn showed that our curve construction process was bi-invariant. On non-compact Lie groups no such guarantee is made, and indeed it is a theorem that $SE(3)$ carries no bi-invariant metric.

To understand the geometric consequences of this fact, it is useful to construct elements of $SE(3)$ in the following way:

Let F be a fixed reference frame in $\mathbb{R}^3$ (we may say that F is an inertial reference frame). Let M be a second reference frame. There exists a unique rotation R_T and translation t_T whose composition map the frame M to F. Together, they form a map from the space of frames on $\mathbb{R}^3$ to itself (see figure 5.1). Indeed, to each frame

$$T = R_T + t_T$$

F

M
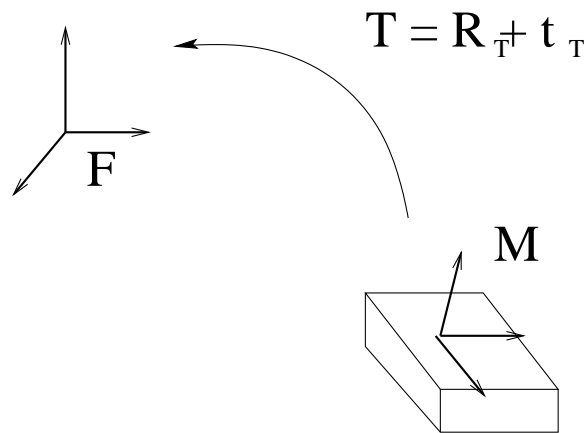
Figure 5.1: A construction of the element $T \in SE(3)$ as the rotation R_T followed by translation t_T which maps the moving frame M to the fixed frame F.

M on $\mathbb{R}^3$ there is a unique map of this form, thus giving a correspondence between frames on $\mathbb{R}^3$ and maps of this kind. Such maps form a group under composition, and indeed this group has an analytic group action. We refer to this as the group of Euclidean motions, which is the Lie group $SE(3)$ discussed in the introduction.

Now it is useful to consider some change of the frame M to another frame M'. There exists a Euclidean motion $T_{M' \to M}$ which maps the frame M' to M. The composition $T \circ T_{M' \to M}$ is the Euclidean motion corresponding to the frame M'. Considering composition as the operation on $SE(3)$, we see that a change of the moving frame M is accomplished by right multiplication. Similarly, if we were to choose a different fixed frame, we would discover that this is equivalent to multiplication from the left.

We now return to the issue of a lack of bi-invariant metric on $SE(3)$. As $SE(3)$ is a Lie group, we can equip it with either a left or right invariant metric. If we choose a left invariant metric, then any geodesics are independent of the choice of where we fix the frame F. However, as such a metric will not be right invariant, we can choose some euclidean motion T' such that the geodesic joining points

$T_1 \circ T'$ and $T_2 \circ T' \in SE(3)$ will not agree with the geodesic joining points $T_1, T_2 \in SE(3)$ right multiplied by T'. Thus, in the case of the motion of a rigid object the choice of where to place the frame on the object affects its trajectory on a geodesic. Alternately, with the choice of a right invariant metric, we can obtain invariance of the choice of where the moving frame is affixed, but at the cost of the trajectory being influenced by the choice of fixed frame.

In general, the choice of one type of invariance over another is not prescribed. However, as a practical matter, there may be a natural choice for the moving frame. For example, in the context of the motion of a rigid body, often the geometry of the body itself dictates the choice of such a frame. Thus, left invariance, or invariance of the fixed frame is generally more desirable for such applications.

5.2 Metric Structures for $SE(3)$

Consider the group structure on $SE(3)$. The Levi decomposition gives that $SE(3)$ is the following semidirect product:

$$SE(3) = SO(3) \ltimes \mathbb{R}^3.$$

Hence, while we can consider the group to be the direct product $SO(3) \times \mathbb{R}^3$ as a manifold, the group structure is more complicated. We shall investigate two possible metric structures, of which one will include this group structure and one will not. It should be emphasized that neither of the metrics we investigate is a natural choice for $SE(3)$ in the sense of providing a natural concept of distance. However, both of the following metrics are useful in that they are feasible from a computational point of view, as shall be discussed.

5.2.1   Ad-invariant pseudo-Riemannian structure

We may choose a metric by structure on $SE(3)$ by retaining the idea of one-parameter subgroups. The metric one obtains, however, is pseudo-Riemannian in the sense that it is non-degenerate but not necessarily positive definite.

Let $X, Y \in \mathfrak{se}(3)$. We may represent $X = \begin{bmatrix} \omega_x \\ v_x \end{bmatrix} \in \mathbb{R}^6$ and $Y = \begin{bmatrix} \omega_y \\ v_y \end{bmatrix} \in \mathbb{R}^6$, where ω_x, ω_y are represented in the basis $\{L_1, L_2, L_3\}$ and v_x, v_y are represented in the standard basis of $\mathbb{R}^3$.

Then we may write down the most general form of an Ad-invariant metric on $SE(3)$ as

$$\langle X, Y \rangle_{Ad-inv} = \alpha \operatorname{Tr}(\hat{\omega_x}^T \hat{\omega_y}) + \beta \langle v_x, \omega_y \rangle_{\mathbb{R}^3} + \beta \langle v_y, \omega_x \rangle_{\mathbb{R}^3}, \qquad \alpha, \beta \in \mathbb{R} \setminus \{0\}.$$

Note that $\langle \cdot, \cdot \rangle_{\mathbb{R}^3}$ is the standard inner product on $\mathbb{R}^3$. Here, we observe that on the right hand side the first term on the left is the bi-invariant metric on $SO(3)$. The latter two summands demonstrate the interaction of $SO(3)$ and $\mathbb{R}^3$ in the direct product. Note that α, β can be either positive or negative.

This metric is well studied. The geodesics produced with this metric are re-ferred to as *screw motions*, and the parameters α, β correspond to whether the geodesic curves have positive or negative energy. Moreover, as this metric is Ad-invariant, by corollary 4.2.7 the curves produced by the de Casteljau algorithm will be bi-invariant in the sense of chapter 4.

5.2.2   Double Geodesic

A much simpler metric structure may be obtained by disregarding the group struc-ture on $SE(3)$ and considering the product metric of the bi-invariant metric on $SO(3)$ and the standard metric on $\mathbb{R}^3$.

Thus, we obtain:

$$\langle X, Y \rangle_{d-geo} = \alpha \operatorname{Tr}(\hat{\omega}_x{}^T \hat{\omega}_y) + \beta \langle v_x, v_y \rangle_{\mathbb{R}^3}, \qquad \alpha, \beta > 0.$$

5.2.3   Comparison of Metric Structures

Of the two metric structures given above, the former is left invariant while the latter is neither. Both, however, rely heavily on the standard metric structures of the underlying manifolds $SO(3)$ and $\mathbb{R}^3$. The reason for this choice is motivated by computation.

The former choice is indeed the Ad-invariant metric inherited by $SE(3)$ from $GL(4)$ (when $SE(3)$ is viewed as a matrix group as discussed in the introduction). Thus the exponential map, as well as the logarithmic map, on $SE(3)$ corresponds to the matrix version of each of these.

Indeed, as we had a closed form for the exponential map on $SO(3)$, we also have a closed form on $SE(3)$, given by the following:

$$\exp_{SE(3)} : \quad \begin{matrix} \mathfrak{se}(3) & \rightarrow & SE(3) \\ V = \begin{bmatrix} \hat{\omega} & v \\ 0_{3\times1} & 0 \end{bmatrix} & \mapsto & g = \begin{bmatrix} \exp_{SO(3)}(\hat{\omega}) & A(\hat{\omega})v \\ 0_{3\times1} & 1 \end{bmatrix} \end{matrix}$$

where

$$A(\hat{\omega}) = I + \frac{1 - \cos \|\omega\|}{\|\omega\|^2} \hat{\omega} + \frac{\|\omega\| - \sin \|\omega\|}{\|\omega\|^3} \hat{\omega}^2.$$

Similarly, we have a closed form for the logarithmic map:

$$\log_{SE(3)} : \quad \begin{matrix} SE(3) & \rightarrow & \mathfrak{se}(3) \\ g = \begin{bmatrix} R & p \\ 0_{3\times1} & 1 \end{bmatrix} & \mapsto & \begin{bmatrix} \hat{\omega} & A^{-1}(\hat{\omega})p \\ 0_{3\times1} & 0 \end{bmatrix} \end{matrix}$$

where here

$$\hat{\omega} = \log_{SO(3)}(R)$$

and
$$A^{-1}(\hat{\omega}) = I - \frac{1}{2}\hat{\omega} + \frac{2\sin\|\omega\| - \|\omega\|(1 + \cos\|\omega\|)}{2\|\omega\|^2 \sin\|\omega\|}\hat{\omega}^2.$$

For the latter metric, the geometry splits into the geometry of the manifolds $SO(3)$ and $\mathbb{R}^3$. Thus, computing geodesics in each of these requires computing geodesics separately on the rotation and translation components. As the Rodriguez formula gives a closed formula for the geodesics on $SO(3)$ (and the geodesics in $\mathbb{R}^3$ are straight lines), such computations are feasible.

Hence, for either choice of a metric structure on $SE(3)$ we have all the tools necessary to efficiently compute geodesics. Thus, the techniques for constructing Bézier and spline curves in compact Lie groups via geodesic interpolation are easily generalized to $SE(3)$. Altafini provides in [1] a good comparison of Bézier curves generated in this way under the above metrics.

5.3   Non-matrix Lie groups and Other Limitations

As illustrated in our discussion of $SE(3)$, non-compact Lie groups do not in general have a natural choice of metric. On an arbitrary Lie group, it remains in the hands of the researcher to determine the important geometric properties of the problem and to understand the limitations of a given metric.

From a computational perspective, the matrix exponential always agrees with the metric induced on a matrix group from $GL(n)$. In practice, this may be the only metric for which we may compute the exponential, and indeed for any matrix we always can compute the exponential. However, we may not in general expect a closed formula for the exponential, and thus such a computation may require the more general tools of numerical analysis. Of course, the computational feasibility of such a general scheme is dubious at best.

As a final note, we observe the simple fact from Lie group theory that there exist groups which admit no matrix representation. The traditional example of such a

group is the *Heisenberg group*, which is discussed in texts such as [2]. While this presents no problem to the theoretical developments of this thesis, the possibility for actual computation on such a group requires a different approach than that which we have discussed. This highlights the fact that matrix groups, as opposed to all Lie groups, are the proper setting on which to discuss computation.

Chapter 6

Knot Insertion and the de Boor Algorithm

In this chapter we explore the interplay between the de Boor algorithm, knot sequences, and knot *insertion* in splines on curved spaces. In particular, we discuss the underlying geometry which allows for a consistent knot insertion procedure in affine space and show how this geometry affects the generalizations of knot insertion in curved spaces.

6.1   Knot Insertion

In chapter 4 we introduced a notion of spline curves in a (compact) Lie group. For the purposes of this section, let $\{u_i\}_{i\in\mathbb{Z}}$ denote a knot sequence and $\{d_i\}_{i\in\mathbb{Z}}$ denote the associated control points.

6.1.1   Classical Knot Insertion

Before delving into knot insertion in Lie groups, it is worth considering the classical theory of knot insertion. So, for the moment, we shall consider our d_i to live in an affine space.

In classical knot insertion, one "inserts" the knot u into the knot sequence $\{u_i\}_{i\in\mathbb{Z}}$. As the knot sequence is ordered, u has a well defined position in the sequence. The knot insertion problem then is to find a corresponding sequence of new control points $\{d_i^u\}_{i\in\mathbb{Z}}$ which define the same degree n spline as did the previous control points before the new knot was inserted.

The classical knot insertion algorithm to do this is the following:

Algorithm 6.1.1 (Knot Insertion Algorithm)

Input: A knot sequence $\{u_i\}_{i \in \mathbb{Z}}$, a control point sequence $\{d_i\}_{i \in \mathbb{Z}}$, and a knot u.

Initialization: Set $I = \max_{u_k \leq u \leq u_{k+1}} k$. If $u = u_I$, then let $r = \text{multiplicity}(u)$. Else $r = 0$.

Output: The control point sequence $\{d_i^u\}$ given by

$$
d_i^u = \begin{cases}
d_i, & i < I - n + 1 \\
\frac{u_{i+n-1}-u}{u_{i+n-1}-u_{i-1}} d_{i-1} + \frac{u-u_{i-1}}{u_{i+n-1}-u_{i-1}} d_i, & I - n + 2 \leq i \leq I + 1 - r \\
d_{i-1}, & i > I + 1 - r
\end{cases}
$$

and the knot sequence including u reordered such that $u = u_{I+1}$.

Depending on how the theory is developed, the fact that the above algorithm is a *consistent* process is non-trivial. By this we mean that we desire the control point sequence obtained by first inserting a knot u and then a knot v to agree with the sequence obtained by first inserting v and then u.

In the classical, affine setting, this procedure is consistent. Following Farin in [9], a proof of consistency is as follows:

Let u be in the interval $[u, u_{I+1}]$ and v in the interval $[u_J, u_{J+1}]$. Inserting either u or v via the above algorithm gives two new knot sequences and control point sequences. By the algorithm we have the following:

$$
d_i^u = \frac{u_{i+n-1}-u}{u_{i+n-1}-u_{i-1}} d_{i-1} + \frac{u-u_{i-1}}{u_{i+n-1}-u_{i-1}} d_i, \qquad I - n + 2 \leq i \leq I + 1 - r
$$

and

$$
d_i^v = \frac{u_{i+n-1}-v}{u_{i+n-1}-u_{i-1}} d_{i-1} + \frac{v-u_{i-1}}{u_{i+n-1}-u_{i-1}} d_i, \qquad J - n + 2 \leq i \leq J + 1 - r.
$$

If the intervals $[u_{I-n+2}, u_{I+1-r}]$ and $[u_{J-n+2}, u_{J+1-r}]$ are disjoint, then the processes above are disjoint, and so we may insert the knots u and v in either order
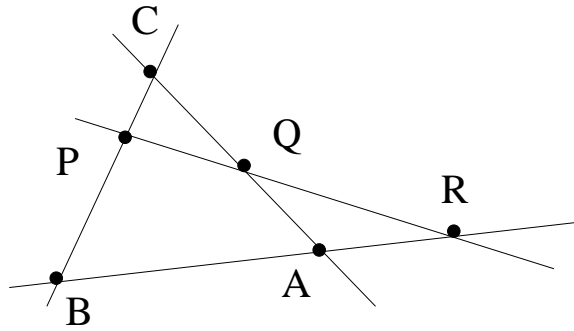
Figure 6.1: A planar diagram for Menelaus' Theorem.

and obtain the same control point sequences. Otherwise, we need to show that the expressions

$$d_{i+1}^{uv} = \frac{u_{i+n-1} - v}{u_{i+n-1} - u_i}d_i^u + \frac{v - u_i}{u_{i+n-1} - u_i}d_{i+1}^u \qquad (6.1)$$

and

$$d_{i+1}^{vu} = \frac{u_{i+n-1} - u}{u_{i+n-1} - u_i}d_i^v + \frac{u - u_i}{u_{i+n-1} - u_i}d_{i+1}^v \qquad (6.2)$$

are equivalent.

While direct computation can show these expressions to be equal, there is a geometric proof of the above which is worth exploring. To demonstrate this proof, we shall first require a classical result in geometry known as Menelaus' Theorem, named for Menelaus of Alexandria[1].

Theorem 6.1.2 (Menelaus) *Referring to figure 6.1, we have the following equality:*

$$\frac{\mathrm{vol}(B, P)}{\mathrm{vol}(P, C)} \cdot \frac{\mathrm{vol}(C, Q)}{\mathrm{vol}(Q, A)} \cdot \frac{\mathrm{vol}(A, R)}{\mathrm{vol}(R, B)} = 1,$$

where $\mathrm{vol}(X, Y)$ *denotes the one dimensional volume (length) of the line segment between* X *and* Y.

---

[1]This result was probably known by Euclid. Menelaus showed in book three of *Sphaerica* a similar result for spherical triangles.
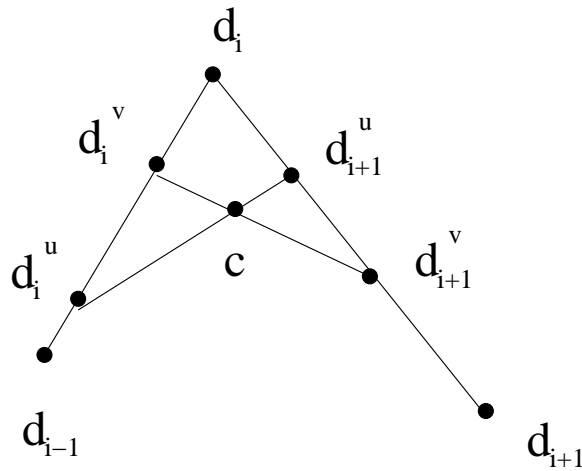
Figure 6.2: For knot insertion to be well defined, we need to show that $c = d_{i+1}^{uv} = d_{i+1}^{vu}$.

Returning to the task of showing that knot insertion is well defined, we refer to figure 6.2. By Menelaus' theorem, we have the following relations:

$$\frac{\text{vol}(c, d_i^v)}{\text{vol}(d_{i+1}^v, c)} = \frac{\text{vol}(d_i^v, d_i^u)}{\text{vol}(d_i, d_i^v)} \cdot \frac{\text{vol}(d_{i+1}^u, d_i)}{\text{vol}(d_{i+1}^v, d_{i+1}^u)}$$

and

$$\frac{\text{vol}(c, d_i^u)}{\text{vol}(d_{i+1}^u, c)} = \frac{\text{vol}(d_i^v, d_i^u)}{\text{vol}(d_i, d_i^u)} \cdot \frac{\text{vol}(d_{i+1}^v, d_i)}{\text{vol}(d_{i+1}^u, d_{i+1}^v)}.$$

Calculating the ratios on the right hand side is a simple task given the definitions of the points. The resulting computation gives that

$$\frac{\text{vol}(c, d_i^v)}{\text{vol}(d_{i+1}^v, c)} = \frac{u - u_i}{u_{i+n-1} - u} \quad \text{and} \quad \frac{\text{vol}(c, d_i^u)}{\text{vol}(d_{i+1}^u, c)} = \frac{v - u_i}{u_{i+n-1} - v}.$$

This of course is just another expression of equations 6.1 and 6.2, as so provides the desired identity.
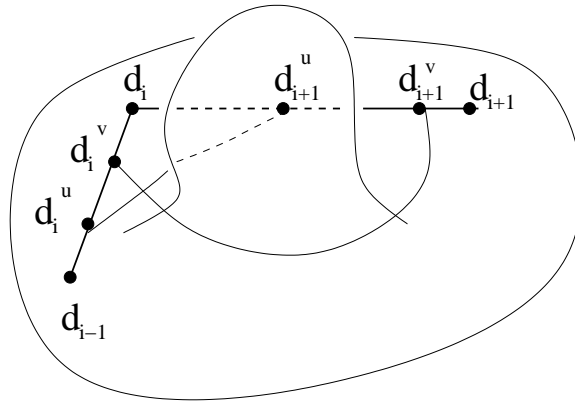
Figure 6.3: A surface obtained by deforming figure 6.2 locally. The geodesics no longer meet in a point prescribed by Menelau's theorem.

6.1.2 Curved Spaces

Now that we have a notion of how a consistent knot insertion algorithm is achieved in affine space, we consider the same problem in the context of curved spaces.

We may define a knot insertion process in curved space in a similar manner to the classical theory by replacing linear interpolation with geodesic interpolation. As the above use of Menelaus' theorem uses only ratios of arclength, one observes that the ratios involving only the points $\left\{ d_{i-1}, d_i^u, d_i^v, d_i, d_{i+1}^u, d_{i+1}^v, d_{i+1} \right\}$ are the same as those of the classical case. Hence, if there exists some analogue to Menelaus' theorem in curved space, then the above proof of consistency goes through for curved spaces.

Unhappily, for arbitrary curved space this is not true. Indeed, even on a surface we should not expect such a condition to hold. For instance, in reference to figure 6.2, if we place a metric on the plane which is flat except in a neighborhood of the point c we obtain a surface like the one pictured in figure 6.3. Thus, we may force the geodesics to intersect in a point such that Menelaus' theorem no longer holds.

Even more troubling is that in dimension at least 3, we are no longer guaranteed

that the geodesics connecting d_i^u to d_{i+1}^u and d_i^v to d_{i+1}^v even intersect! Clearly, for general curved spaces we are not guaranteed a consistent knot insertion process.

We call then a space on which Menelaus' theorem is true (using geodesic interpolation) in convex neighborhoods a *Menelaus space*. One is then led to seek a classification of these spaces. As the exponential map is a local isometry for spaces of vanishing curvature we see that all such spaces are Menelaus spaces. What of the nonzero constant sectional curvature spaces, of which the Lie groups and general symmetric spaces are of most interest to this thesis? At present, we have no answer for such spaces.

And what about spaces which are not Menelaus. It is clear that such spaces will not have consistent knot insertion procedures using geodesic interpolation. It might still be useful to characterize the degree to which the order of the knots inserted affects the outcome. Indeed, if the constant curvature spaces are not Menelaus, the possible change due to knot order might be bounded. Again, more research is required so that we might answer these questions.

6.2 The de Boor Algorithm

We introduced the de Boor algorithm in chapter 4 as a means of defining splines on a group G. One of the classical results of CAGD is that the insertion of a knot repeatedly to raise the degree of the knot to n (the degree of the spline) is in fact just the de Boor algorithm. Our definition of knot insertion on curved space, then, is meant to correspond with the de Boor algorithm in just this way. However, unless the knot insertion procedure is consistent as discussed above, this correspondence holds only if no intermediary knot-insertions are performed.

It is worth noting that the de Boor algorithm, in contrast to the knot insertion algorithm, does not alter the sequences but instead just produces a single resultant point. Thus, as the sequences remain unchanged the de Boor algorithm does not

suffer from the consistency problem, contrasting with the process of knot insertion.

6.3 Knot Insertion and Further Analysis

Knot insertion is related to more than the de Boor algorithm. Indeed, certain methods of spline analysis are assisted with the tools of knot insertion. We present two such methods here, those of polar forms and polygonal convergence.

6.3.1 Polar Forms

One of the great advances in the theory of Bézier curves is the *polarization* technique. One may associate to each $m-$degree polynomial curve $F(t)$ (in affine space) an associated polar form $f(x_1, \ldots, x_m)$ which is symmetric and $m-$affine and for which $F(t) = f(t, \ldots, t)$. One may in fact build up the theory of polynomial curves using polar forms, and then the theory of Bézier curves becomes a series of results about the structure of these forms (Gallier does exactly this in his book [10]). The theory of spline curves, or piecewise polynomial curves, is similarly built up in this manner. Indeed, analysis of the derivatives of polynomial curves is most elegantly stated in the language of polar forms, and this in turn leads to an elegant characterization of the continuity of joining polynomials.

With these advantages, one is led to question why this thesis and other literature on curve design in curved space forgoes the polar form technique. To answer this question one need observe that polar forms rely heavily on notions of affine space, and for spaces with non-zero curvature no affine-like notion makes sense.

Knot insertion, however, provides a partial link to a concept of polar forms on curved space. Consider a spline curve on the group G. Then to each control point d_i we may associate the subsequence $\langle u_i, \ldots, u_{i+m} \rangle$ of the knot sequence, where m is the degree of the spline. This is best illustrated with the following example.
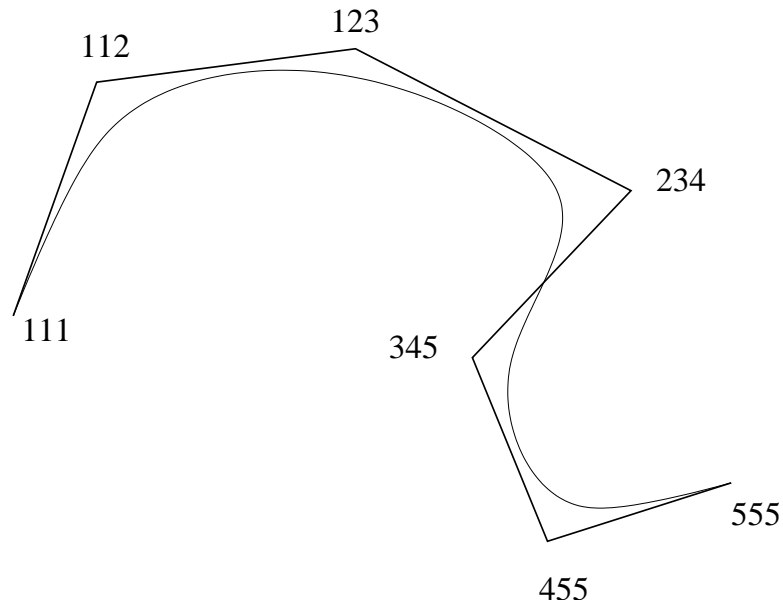
Figure 6.4: A cubic spline corresponding to the knot sequence $1, 1, 1, 2, 3, 4, 5, 5, 5$. Note that every control vertex is associated to 3 (the degree of the spline) consecutive knots.

Example 6.3.1 *Consider the (finite) cubic spline given in figure 6.4 associated to the knot sequence $1, 1, 1, 2, 3, 4, 5, 5, 5$. Note that to each consecutive triple in the knot sequence there is an associated control vertex. Indeed, if we wanted to find the point associated to an arbitrary triple, say the point $2, 2.5, 2.7$, we need only insert the knots 2.5 and 2.7 into the knot sequence through the knot insertion algorithm. The new knot sequence will be $1, 1, 1, 2, 2.5, 2.7, 3, 4, 5, 5, 5$, and so one of the consecutive triples will correspond to $2, 2.5, 2.7$, and thus a new control point of the polygon.*

As we see in this example, we may create a correspondence between $m-$tuples of parameters and points in our space.[2]. As such, there is some hope of defining polar forms through a *consistent* knot insertion process. However, any such theory is predicated on the geometry of knot insertion, and thus an understanding of

---

[2]This correspondence is not total. Indeed, in example 6.3.1 we see that the triple $1, 1.5, 3$ cannot be created, since the knot 2 will always lie betwen 1.5 and 3.

Menelaus spaces. Conversely, if one were to find an adequate theory of polar forms in curved spaces, such a theory would give an understanding of Menelaus spaces.

6.3.2   *Subdivision and Convergence*

One of the most useful applications of knot insertion is in the process of subdivision, that is, inserting knots to create more control points and thus refining the control polygon. Subdivision schemes are used both theoretically to show that the control polygon converges to the desired curve, as well as in practice as a way of quickly finding a good approximation to the curve using straight line segments.

These purposes are of similar interest in the case of curved spaces. However, as subdivision is based on knot insertion, the standard analysis of convergence will require a space to be Menelaus, and thus this interesting question also defers to the larger issue of classifying Menelaus spaces. It would be interesting, however, to see if convergence of different subdivision schemes is possible without the machinery of knot insertion. For spaces which are not Menelaus, we do not expect convergence to the original spline. Indeed, the conditions under which convergence to *some* curve happens are still unknown, as are the different affects of distinct subdivision schemes. Such questions seem fertile ground for future exploration.

Chapter 7

Interpolation

The motivating question for this thesis is more specific than the construction of a curve in non-Euclidean space. Rather, the question sought the construction of curves which *interpolate* given elements, subject to certain geometric constraints such as continuity. One of the great successes of spline curves is their ability to interpolate arbitrary points in affine space. Thus, this chapter seeks to extend the techniques used to develop interpolating curves in CAGD to Lie groups.

7.1 Overview of Interpolation in CAGD

In computer aided geometric design, there are many possible interpolation problems one may hope to solve. At a basic level, one is given a sequence of points and asked to find a curve, typically a spline curve, which interpolates the points. Additionally, one might specify the order of the spline curve, the desired degree of continuity at each of the interpolated points, and an end condition (i.e. a specification of behavior of the curve at the endpoints).

The traditional solution to this problem is to assign an indeterminate Bézier curve (and associated control polygon) to interpolate every consecutive set of points. One may then determine conditions on Bézier polygons so that the specified degree of continuity may be achieved at each point to be interpolated. Because splines in affine space are such that every interpolated point may be specified as a linear combination of n of the indeterminate control points, the interpolation problem may be transfered to a matrix equation and thus in some cases efficiently

solved. Indeed, the industry standard for interpolation are cubic splines of C^2 continuity, due to the efficiency of tri-diagonal matrix solvers.

7.2 Specific Interpolation Problem

Unfortunately, for the case of spline curves in a Lie group G, even the simplest of interpolation problems presents difficulty. Certainly C^0 splines, those with only simple continuity at the interpolation points, are possible since we may specify that sequential Bézier curves share endpoints. However, the only degree spline for which the C^0 condition fully specifies the curve is $n = 1$, which reduces the problem to piecewise geodesic interpolation.

If we seek greater continuity at the interpolation points, say C^1 continuity, the task becomes more difficult. Let us choose the degree of the spline to be $n = 2$, so that this continuity condition specifies the spline. Given points $\{x_0, \ldots, x_k\}$, and associated knots $\{u_0, \ldots, u_k\}$, let us attempt to interpolate with the above spline type.

We now seek to find a sequence of points, $\{b_i\}$ which define the interpolating spline. If we set $b_{2i} = x_i$ for $0 \leq i \leq k$, we then need only discover the points $b_{2i+1}, 0 \leq i < k$. For C^1 continuity we have that

$$b_{2i} = \text{interp}_\lambda(b_{2i-1}, b_{2i+1}), \qquad 0 < i < k, \tag{7.1}$$

where $\lambda = \frac{u_i - u_{i-1}}{u_{i+1} - u_{i-1}}$.[1]

From a computational point of view, we are interested only in matrix groups. Thus, the problem reduces to a system of matrix equations determined by equation 7.1 as

---

[1]Technically, an *end condition* need be specified in order to determine all the $\{b_i\}$. For clarity of exposition we shall ignore this issue.

$$b_{2i} = b_{2i-1} \exp(\lambda \log(b_{2i-1}^{-1} b_{2i+1})).$$

Unlike the affine case, where linear interpolation reduced the problem to a matrix equation, the above problem is decidedly non-linear. Hence, the problem is reduced to a problem of numerical analysis, albeit a challenging one.

7.3 Affine Embeddings and Interpolation

7.3.1 An Interpolation Program

As seen in the previous section, the general interpolation problem remains difficult due to the nonlinearity inherent in equation 7.1. In contrast, the affine case proved tenable due to the fact that points in the desired curve could be expressed as linear combinations of points in the space. One is then led to wonder whether we might "linearize" our problem by embedding our curved space into an affine space in such a way that points in the curved space are expressible as linear combinations of other points.

If such an embedding exists, then we have the following program for computing interpolating curves:

1. Embed our Lie group G in $\mathbb{R}^n$, for some n.

2. Find an expression for minimizing geodesics between two points in G which is expressible as a linear combination of the endpoints.

3. For the given degree spline desired, express the points to be interpolated as a linear combination of the desired control points.

4. Transform the above constraints to a matrix equation and solve, provided a solution exists.

Remark 7.3.1 *Note that in the above program, we desire to express geodesics as* linear combinations *of points in G. Note that this requirement is different from expressing geodesics as* linear interpolants *of the points. Indeed, for curved G we expect such geodesics to differ from linear interpolation in general.*

7.3.2 S^3, $SO(3)$, and Difficulties

We now attempt the above program on the group S^3, the unit sphere in $\mathbb{R}^4$. Our interest in S^3 arises due to its relation to $SO(3)$. To see this, reinterpret S^3 as the set of unit quaternions, i.e.

$$S^3 = \left\{ q = a + b\hat{\boldsymbol{\imath}} + c\hat{\boldsymbol{\jmath}} + d\hat{\boldsymbol{k}} \in \mathbb{H} \mid a^2 + b^2 + c^2 + d^2 = 1 \right\}.$$

If one views a vector $x \in \mathbb{R}^3$ as a pure imaginary unit quaternion, then for any $q \in S^3$ it can be shown that the linear map $L_q : \operatorname{Im} \mathbb{H} \to \operatorname{Im} \mathbb{H}$ given by

$$L_q : x \in \operatorname{Im} \mathbb{H} \mapsto qx\bar{q}$$

corresponds to a rotation of $x \in \mathbb{R}^3$. Moreover, the antipodal points q and $-q$ correspond to the same rotation. Indeed, the map $\pi : S^3 \to SO(3)$ given by

$$\pi : q \in S^3 \mapsto L_q \in SO(3)$$

is surjective, so to every rotation in $SO(3)$ there are two preimages in S^3. Moreover, π is a local isometry, so specifically minimizing geodesics between two points in a convex neighborhood of S^3 become minimizing geodesics between corresponding rotations under composition with π. As a result, interpolating on $SO(3)$ may be viewed as interpolation on S^3, a space whose geometry is better understood. Many references discuss these issues, including [17]. [2]

---

[2]Geometrically speaking, S^3 is the orientable double cover of $SO(3)$ endowed with the covering metric.

On the $3-$sphere the minimizing great circle arcs (the geodesics) between points q_1 and q_2 are given by

$$\frac{\sin{(1-t)\theta}}{\sin\theta}q_1 + \frac{\sin t\theta}{\sin\theta}q_2,$$

where $\cos\theta = \langle q_1, q_2\rangle$ (see [15]). Thus the utility of working on S^3 embedded in this manner becomes clear; geodesics are expressible as linear combinations of their endpoints.

Unfortunately, there is an added difficulty which disrupts our interpolation program. To see this, consider the problem of quadratic spline interpolation as discussed in section 7.2. To enforce C^1 continuity, we have the following set of equations:

$$x_i = \frac{\sin{(1-t)\theta_i}}{\sin\theta_i}b_{2i-1} + \frac{\sin t\theta_i}{\sin\theta_i}b_{2i+1}, \qquad 0 < i < L,$$

with $\cos\theta_i = \langle b_{2i-1}, b_{2i+1}\rangle$. As we are solving for the b_{2i-1}, we have no *a priori* information about their relative geometry, and thus in addition to the resulting linear system of equations we must add the additional constraints $\cos\theta_i = \langle b_{2i-1}, b_{2i+1}\rangle$ in order to obtain a solution. However, such constraints make the numerical problem non-linear. Hence, in the case of the three-sphere, we see that our "linearization" program only serves to reformulate the numerical problem. It remains open as to whether such a problem admits a tenable solution for computation. However, as we shall now demonstrate, imposing some boundary data may actually resolve this problem.

7.4 Boundary Constraints

7.4.1 Quadratic Splines

Considering again the case of quadratic spline curves, we observe that we desire to find the points $b_{2i+1}, 0 \le i < k$, of which there are k (given k points to interpolate

between). Equation 7.1 gives $k - 1$ equations relating these points. Thus, there is one degree of freedom in this system. This degree of freedom is traditionally taken care of by imposing an *end condition*, boundary data such as the tangent vector at the beginning of the spline.

Let us impose a boundary condition for some quadratic spline on a compact Lie group G. We shall specify the point b_1 in some manner. Now, in combination with equation 7.1, the spline is fully determined. Further, we note that b_3 is related to b_1 by the equation

$$b_2 = \mathrm{interp}_\lambda(b_1, b_3), \tag{7.2}$$

where $\lambda = \frac{u_2 - u_1}{u_3 - u_1}$, and b_2 is given. It is worth noting that b_2 lies on the interpolating geodesic between b_1 and b_3, and thus the tangent vector at b_1 which generates the geodesic to b_2 also generates the geodesic to b_3. Moreover, since the distance between b_1 and b_2 is well defined, the continuity condition expressed in equation 7.2 specifies the distance between b_2 and b_3. Thus, we see that from b_1 we may find b_3 by

$$b_3 = \mathrm{interp}_\Lambda(b_1, b_2),$$

where here $\Lambda = 1 + \frac{u_3 - u_2}{u_2 - u_1}$. Observe that here Λ exceeds unity, so the above quantity is actually an extrapolation of the arc between the points b_1 and b_2.

We may iterate the above process. As a result, we can calculate *a priori* the geometry between the points b_{2i+1} and b_{2i+3}, which allows us to sidestep the non-linear constraint in the last section. However, this method gives more geometry information than pairwise distance; by repeating this process the points b_{2i+1} are discovered! Thus, this process forms a self-contained interpolation algorithm for quadratic splines.

7.4.2 Cubic Splines

Next, we may consider cubic, degree 3 splines. In this case, given $k + 1$ points $x_0, \ldots, x_k$ and $k + 1$ knots $u_0, \ldots, u_k$ we seek points $d_{-1}, \ldots, d_{k+1}$ which define a cubic spline f for which $f(u_i) = x_i$ for all i. The endpoints give that $b_{-1} = x_0$ and $b_{k+1} = x_k$. The remaining points are given by the relationship:

$$d_{i-1,1} = \text{interp}_{\lambda i-1}(d_{i-1}, d_i) \tag{7.3}$$

$$d_{i,1} = \text{interp}_{\lambda i}(d_i, d_{i+1}) \tag{7.4}$$

$$x_i = \text{interp}_{\lambda'_{i-1}}(d_{i-1,1}, d_{i,1}), \quad 1 \leq i < k, \tag{7.5}$$

where $\lambda_{i-1} = \frac{u_i - u_{i-2}}{u_{i+1} - u_{i-2}}$, $\lambda_i = \frac{u_i - u_{i-1}}{u_{i+2} - u_{i-1}}$, and $\lambda'_{i-1} \frac{u_i - u_{i-1}}{u_{i+1} - u_{i-1}}$. It is easy to show that these relationships leave two degrees of freedom in the spline. In the traditional theory, these degrees of freedom are fixed by specifying the tangents at the endpoints or equivalently by fixing the points d_0 and d_k.

If we fix the points d_0 and d_k, we may then attempt to iteratively find the remaining d_i in a similar manner to that used in the quadratic spline case. We note that d_1 is related to d_0 by equation 7.3. However, unlike the quadratic case the point $d_{i-1,1}$ is not given. More generally, we see that the constraints given by equations 7.3, 7.4, 7.5 depend on the points $\{d_{i-1}, d_i, d_{i+1}, x_i\}$. Thus, if we want to solve for one of these points we need to know the other three, and so specifically knowledge of d_0 and x_1 are not sufficient to determine d_1.

Now consider an alternate end condition for the cubic spline. Instead of fixing the points d_0 and d_k, we may instead fix the points d_0 and d_1, corresponding to a choice of velocity and acceleration at the beginning of the spline. By our arguments above, this satisfies the two degrees of freedom in the spline, and thus fully determines the spline. Moreover, as the points $\{d_{i-1}, d_i, x_i\}$ fully specify the point d_{i+1} we thus obtain an iterative method for determining all the d_i.

We summarize this process for cubic spline interpolation in the following algorithm:

Algorithm 7.4.1 (Cubic Spline Interpolation with Beginning Boundary Data)

Input: A sequence of $k+1$ points $\{x_0, \ldots, x_k\}$ in the compact Lie group G. A knot sequence $\{u_0, \ldots, u_k\}$. Points d_0 and d_1 in G such that $x_0^{-1} d_0$ and $d_0^{-1} d_1$ both lie in some convex neighborhood about the identity.

Iteration: For stage i (proceeding from $i = 1$ to $i = k$) compute the b_i as follows:

$$
\begin{aligned}
d_{i-1,1} &= \operatorname{interp}_{\lambda_i}(d_{i-1}, d_i) \\
d_{i,1} &= \operatorname{interp}_{\Lambda_i}(d_{i-1,1}, x_i) \\
d_{i+1} &= \operatorname{interp}_{\Lambda_i'}(d_i, d_{i,1}),
\end{aligned}
$$

where $\lambda_i = \frac{u_i - u_{i-2}}{u_{i+1} - u_{i-2}}$, $\Lambda_i = 1 + \frac{u_{i+1} - u_i}{u_i - u_{i-1}}$, and $\Lambda_i' = 1 + \frac{u_{i+2} - u_i}{u_i - u_{i-1}}$.

Output: Return the points $\{d_{-1}, \ldots, d_{k+1}\}$.

Remark 7.4.2 *The resulting spline defined on the points $\{d_{-1}, \ldots, d_{k+1}\}$ and the knot sequence $\{u_0, u_0, u_0, u_1, u_2, \ldots, u_{k-2}, u_{k-1}, u_k, u_k, u_k\}$ interpolates the points x_i in the sense*

$$
f(u_i) = x_i, \qquad 0 \leq i \leq k.
$$

7.4.3   Further Technical Details

The methods presented for quadratic and cubic splines generalizes easily. That is, for degree n splines, if the first $n - 1$ control points are specified than one may find the remaining control points constructively through a series of interpolations and extrapolations. Moreover, there is no particular requirement that the points specified need be the first control points. It is easy to see that this method requires only $n - 1$ consecutive control points different from the endpoints to be specified. However, the choice of how one fixes appropriate control points leaves room for much research.

A more pressing technical concern is the choice of knot sequence. Because of our use of extrapolation, it is possible to construct a knot sequence which forces two consecutive control points to lie far enough apart such that they are not in any common convex neighborhood. This will cause the resulting spline to be ill-defined, since our use of geodesic interpolation requires all consecutive control points to lie sufficiently close together. The remedy for this problem is to construct or modify a knot sequence while the control point sequence is constructed in order to force all control points to lie close enough together. Again, "good" methods for constructing such a sequence need research.

Chapter 8

Conclusion

In this thesis we have shown an approach to curve design on Lie groups which is based on the notion of geodesic interpolation. We have demonstrated that in the case of Bézier curves, the theory of CAGD generalizes easily using geodesic interpolation, and further we have provided analysis to complement that already existing in the field.

Additionally, we have shown that one may also define a notion of spline curves on Lie groups. However, our analysis shows that the tools and applications of spline curves, especially knot insertion and interpolation, do not generalize as easily. Indeed, the underlying geometry of curved spaces increases the complexity of each of these topics. As a result, new routes of investigation have been opened which should help to further generalize the theory here developed as well as solve the specific kinematic questions relating to $SO(3)$ and $SE(3)$. Moreover, we have shown how the central interpolation problem motivating the thesis may be solved in an efficient manner.

8.1 Further Work

The work in this thesis suggests new routes of investigation. One of the major pressing questions this thesis raises is the classification of Menelaus spaces, as this will resolve many issues relating to knot insertion and related topics. Paralleling this question is the possibility of developing error bounds on knot insertion in non-Menelaus spaces.

The interpolation problem still leaves many questions to be researched. "Good" methods for choosing boundary conditions and knot sequences require some work. A more careful attention to the computational efficiency of our approach is needed for optimized performance.

The analytical tools of the theory still might admit significant improvement. In particular, a better characterization of derivatives would significantly improve the theory.

Finally, we suggest that the class of symmetric spaces is perhaps the most general setting for the theory we promote here. Indeed, Crouch et. al. demonstrate how curve design on $SO(n+1)$ and on S^n are easily related, and it seems natural that their work should generalize.

8.2 Recommended Reading

For a good introduction to the classical theory, the books by Farin [9] and Gallier [10] are recommended as comprehensive introductions. The former is a more traditional text, geared to applications of spline curves while the later is a more mathematical treatise which develops the theory through the framework of polar blossoms.

The most current work relating to geodesic interpolation-based curve design is the work of Crouch, Kun, and Leite in [8]. Their work is especially geared toward application on rotation groups and spheres. Altafini, in [1], applies some of the techniques from [8] to the special case of $SE(3)$.

Appendix A

A Matrix Polynomial Approach to Bézier Curves

In the following appendix we develop a theory of Bézier curves which is based on matrix polynomials (contrasting with the traditional theory developed with simple polynomials). As the material here is not used in the body of the thesis, it is presented as a more or less self contained work which may be read independently of the thesis.

A.1 Introduction

A.1.1 Historical Perspective

The purpose of an appendix is to include material which may not be incorporated into the linear flow of the main document. Indeed, the material presented here meets this criterion well. However, it is not intended that the reader should see this material as completely separate from the main corpus of the thesis. Rather, this material arose in the investigation of the central problem of this thesis, and maintains connections to the work of the thesis. Thus we present a very brief historical perspective detailing how this work came about.

In considering the central problem of this thesis, interpolation on Lie groups (especially those which are realized as matrix groups), our methodology has been to promote the extremely successful techniques of CAGD in affine space to the Lie group setting. In this regard, it appeared natural to consider one of the foundations of the theory, namely *polynomials*, and consider how the theory might work if we replaced these with *matrix polynomials*. Our hope in this regard was to obtain a very

analogous theory in matrix spaces which would allow for simple computation in Lie groups. For reasons which should become clear upon reading this appendix, this approach did not bear fruit as desired. However, the theory is still interesting in its own right, and still retains applications to curve planning in useful settings.

As is the nature of mathematical research, the earlier, less fruitful ideas are often tossed aside in the presentation of the final theory. Such an exposition, however, reflects little of the journey through which the results were achieved, often at the expense of disregarding work which is useful and interesting in its own right. Hence, we take liberties with flexible structure of a senior thesis and present the following work.

A.1.2   Overview

The appendix unfolds as follows: We begin first by examining matrix polynomials. We then examine the results of a simpleminded attempt to define a blossom of such polynomials. Following this, we present a more sophisticated and fully developed generalization of these concepts. This in turn allows us to define *Matrix Bézier curves*. We demonstrate some subtleties about these curves which differ from the traditional theory. Additionally we show how many of the properties of such curves relate to those of the traditional theory. Finally, we conclude with some applications of the theory as well as possible areas of future work.

Lastly, we note that the notation in this appendix is meant to correspond with that found in Gallier's book on CAGD [10]. We expect the reader to encounter little trouble with this choice, except in our rather technical use of homogenized polar forms (we denote the homogenized form of a polar form f as $\widehat{f}$).

A.2 A First Look at Matrix Polynomials and Polar Forms

Let $A \in M_{n \times n}(\mathbb{R})$. Consider a matrix polynomial (an element of $\mathbb{R}[A]$) F where

$$F(A) = aA^2 + bA + cI.$$

Polynomials provide a natural class of curves on matrices. Thus, we seek efficient means to control and compute such curves. We use the technique of *polarization* as our point of departure for the study of such curves.

Analogizing to the traditional theory, we may *polarize* the above matrix quadratic in the following manner:

$$f(A_1, A_2) = aA_1 A_2 + b\frac{A_1 + A_2}{2} + cI.$$

Specifically, observe that this polarization recovers the original matrix polynomial when we let $A = A_1 = A_2$. Indeed, one may also verify that the above polarization is also bi-affine in each of its arguments if we view each matrix as living in an affine space isomorphic to $\mathbb{R}^{n^2}$. Further, we may generalize the concept of linear interpolation to apply to matrices. Let $M, N \in M_{n \times n}(\mathbb{R})$ such that the difference $M - N$ is invertible (the reason to be made clear shortly). For $\lambda \in \mathbb{R}$, we see that

$$A = (1 - \lambda)M + \lambda N$$

is an affine combination of the matrices M, N.

Denote by $A_i, i = 1, 2$, the affine combination determined by λ_i. Thus, as f is bi-affine we have that

$$
\begin{aligned}
f(A_1, A_2) &= f((1 - \lambda_1)M + \lambda_1 N, (1 - \lambda_2)M + \lambda_2 N) \\
&= (1 - \lambda_1)(1 - \lambda_2)\, f(M, M) + (1 - \lambda_1)\lambda_2\, f(M, N) \\
&\quad + \lambda_1(1 - \lambda_2)\, f(N, M) + \lambda_1 \lambda_2\, f(N, N).
\end{aligned}
$$

Solving for the λ_i, we may rearrange the the equation

$$A_i = (1 - \lambda_i)M + \lambda_i N$$

such that we obtain

$$\lambda_i I \;=\; (N-M)^{-1}(A_i-M)=(A_i-M)(N-M)^{-1} \qquad \text{(A.1)}$$

$$(1-\lambda_i)I \;=\; -(N-M)^{-1}(A_i-N)=-(A_i-N)(N-M)^{-1}. \qquad \text{(A.2)}$$

The above follows since we note that we assumed $N-M$ to be invertible.

Using these identities and letting A_1, A_2 equal A, we obtain the formula

$$\begin{aligned}
f(A,A) \;=\; & (N-M)^{-1}(A-N)(N-M)^{-1}(A-N)\ f(M,M)\\
& -(N-M)^{-1}(A-N)(N-M)^{-1}(A-M)\ f(M,N)\\
& -(N-M)^{-1}(A-M)(N-M)^{-1}(A-N)\ f(N,M)\\
& (N-M)^{-1}(A-M)(N-M)^{-1}(A-M)\ f(N,N).
\end{aligned}$$

We obtain a further simplification by noting that $(N-M)^{-1}$ and $(A-M)$ commute (similarly $(N-M)^{-1}$ and $(A-N)$), as can be seen by equations A.1 and A.2. Hence, the above becomes

$$\begin{aligned}
f(A,A) \;=\; & ((N-M)^{-1})^2(A-N)^2\ f(M,M)\\
& -((N-M)^{-1})^2(A-N)(A-M)\ f(M,N)\\
& -((N-M)^{-1})^2(A-M)(A-N)\ f(N,M)\\
& +((N-M)^{-1})^2(A-M)^2\ f(N,N).
\end{aligned}$$

We note that the curve defined by $f(A,A)$ now completely depends on the points $\{f(M,M), f(M,N), f(N,M), f(N,N)\}$. Thus, we may view the expressions below as coefficient functions of these points, corresponding to the Bernstein polynomials of the classical theory.

$$((N-M)^{-1})^2(A-N)^2$$

$$((N-M)^{-1})^2(A-N)(A-M)$$

$$((N-M)^{-1})^2(A-M)(A-N)$$

$$((N-M)^{-1})^2(A-M)^2$$

Thus, this simple polarization technique gives many shades of the classical theory. However, there is a lack of symmetry in the constructed polarized form f, namely, $f(N, M) \neq f(M, N)$. Moreover, the coefficient functions above corresponding to these points differ. In the classical theory, we desire the symmetry that we lack here, since this symmetry allows us to polynomial curves of degree n with $n + 1$ points instead of the 2^n points required in a construction as above. Thus, in the following section we discuss a polarization technique which overcomes this difficulty and lines up well with the classical theory.

A.3 Multiaffine Symmetric Maps and Matrix Bernstein Polynomials

A.3.1 Symmetric Matrix Functions

We desire to construct the *symmetric polar blossom* of some matrix polynomial. That is, we desire an $m-$affine function $f : (M_{n \times n}(\mathbb{R}))^m \to M_{n \times n}(\mathbb{R})$ which is symmetric (to be defined in a moment) and which recovers the matrix polynomial when all arguments are set equal to one another.

Given matrix variables $X_1, \ldots, X_n$, we first introduce the *elemetary symmetric matrix functions* $\varsigma(X_1, \ldots, X_n)$ as follows:

$$\varsigma_0 = I$$

$$\varsigma_1 = X_1 + \ldots + X_n$$

$$\varsigma_2 = X_1 X_2 + X_1 X_3 + \ldots + X_1 X_n + X_2 X_3 + X_2 X_4 + \ldots + X_{n-1} X_n$$

$$+ X_n X_{n-1} + X_n X_{n-2} + \ldots + X_n X_1 + X_{n-1} X_{n-2} + \ldots + X_3 X_2 + X_3 X_1 + X_2 X_1$$

$$\varsigma_k = \sum_{1 \leq i_1 \leq \ldots \leq i_k \leq n} \left(\sum_{\pi \in S_k} X_{i_{\pi(1)}} \cdots X_{i_{\pi(k)}} \right).$$

By construction, the above functions are *symmetric*, i.e. for any permuatation $\pi \in S_n$:

$$\varsigma_k(X_1, \ldots, X_n) = \varsigma_k(X_{\pi(1)}, \ldots, X_{\pi(n)}).$$

We further observe that the above matrix functions are multiaffine, that is, for every $X_i = (1 - \lambda)M + \lambda N$ we have that

$$
\begin{aligned}
\varsigma_k &= (X_1, \ldots, (1 - \lambda)M + \lambda N, \ldots, X_n) \\
&= (1 - \lambda)\varsigma_k(X_1, \ldots, M, \ldots, X_n) + \lambda\varsigma_k(X_1, \ldots, N, \ldots, X_n).
\end{aligned}
$$

A.3.2   Matrix Polynomials and Blossoms

Given the degree m matrix polynomial

$$
F(X) = a_m X^m + a_{m-1} X^{m-1} + \ldots + a_1 X + a_0 I,
$$

we seek to construct the polar blossom. We define this blossom by replacing the jth order term in $F(X)$,

$$
a_j X^j
$$

with

$$
a_j \left[\binom{n}{j} j! \right]^{-1} \varsigma_j(X_1, \ldots, X_n)
$$

for all $0 \leq j \leq m$. Note that the coefficient change reflects the fact that

$$
\varsigma_k(X, \ldots, X) = \left[\binom{n}{k} k! \right] X^k.
$$

Hence, we see that setting $X_i = X$ for all i in the blossom gives

$$
f(X, \ldots, X) = F(X).
$$

By construction, we observe that this function f is symmetric and multiaffine, as we desired.

A.3.3 Linear Interpolation and the Matrix Bernstein Polynomials

We now examine polar forms evaluated on matrices which are interpolated linearly.

Let $A_i = (1 - \lambda_i)M + \lambda_i N$. Then we have the following expression by the symmetry and $m-$affine structure of f:

$$
\begin{aligned}
f(A_1, \ldots, A_n) &= (1 - \lambda_1)\, f(M, A_2, \ldots, A_n) + \lambda_1\, f(N, A_2, \ldots, A_n) \\
&= \ldots \\
&= \sum_{k=0}^{m} \sum_{\substack{I \cup J = \{1, \ldots, m\} \\ I \cap J = \emptyset, |J| = k}} \prod_{i \in I}(1 - \lambda_i) \prod_{j \in J} \lambda_j\, f(\underbrace{M, \ldots, M}_{m-k}, \underbrace{N, \ldots, N}_{k}).
\end{aligned}
$$

As $A_i = (1 - \lambda_i)M + \lambda_i N$, we have that

$$
\begin{aligned}
\lambda_i I &= (N - M)^{-1}(A_i - M) = (A_i - M)(N - M)^{-1} \\
(1 - \lambda_i)I &= -(N - M)^{-1}(A_i - N) = -(A_i - N)(N - M)^{-1}.
\end{aligned}
$$

$$\text{(A.3)}$$

Thus, we can use these equalities to examine the matrix coefficients of the points $f(\underbrace{M, \ldots, M}_{m-k}, \underbrace{N, \ldots, N}_{k})$. However, before we proceed, we shall demonstrate a useful characterization of these matrices.

Lemma A.3.1 *Both* $(A_i - N)$ *and* $(A_i - M)$ *are scalar multiples of* $(N - M)$.

Proof. Expanding according to the definitions gives us

$$
\begin{aligned}
A_i - N &= (1 - \lambda_i)M + \lambda_i N - N = -(1 - \lambda_i)(N - M) \\
A_i - M &= (1 - \lambda_i)M + \lambda_i N - M = \lambda_i(N - M).
\end{aligned}
$$

$\square$

As an immediate corollary, we see that the above matrices commute.

Now, if we let $A_i = A$ for all i, and substitute the matrix equivalents for λ_i and $1 - \lambda_i$, we obtain the following coefficient for $f(\underbrace{M, \ldots, M}_{m-k}, \underbrace{N, \ldots, N}_{k})$:

$$\binom{m}{k}(-1)^{m-k}\left[(N - M)^{-1}\right]^m(A - N)^{m-k}(A - M)^k.$$

We shall denote this as the *kth Matrix Bernstein polynomial of degree m*, or $B_k^m(A)$.

Further, if we apply the equalities from the lemma, we see that this polynomial equals:

$$\binom{m}{k}(1 - \lambda)^{m-k}\lambda^k I.$$

This is the identity matrix multiplied by the standard Bernstein polynomials. As such, many of the identities that hold for the Bernstein polynomials also hold for their matrix versions. We list some of these properties in the following section.

A.3.4 Properties of Matrix Bernstein Polynomials

Here we develop properties of the Matrix Bernstein polynomials analogous to the traditional Bernstein polynomials.

Proposition A.3.2

$$B_k^n(t) = (N - M)^{-1}(A - M)B_{i-1}^{n-1}(t) - (N - M)^{-1}(A - N)B_i^{n-1}(t).$$

Proof.

$$
\begin{aligned}
B_i^n(t) &= \binom{n}{i}(1 - t)^{n-i}t^i I \\
&= \binom{n-1}{i}(1 - t)^{n-i}t^i I + \binom{n-1}{i-1}(1 - t)^{n-i}t^i I \\
&= (1 - t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t) \\
&= -(N - M)^{-1}(A - N)B_i^{n-1}(t) + (N - M)^{-1}(A - M)B_{i-1}^{n-1}(t).
\end{aligned}
$$

$\square$

Proposition A.3.3

$$\sum_{i=0}^{n} B_i^n(t) = I.$$

Proof.

$$
\begin{aligned}
\sum_{i=0}^{n} B_i^n(t) &= \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i I \\
&= \left(\sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i \right) I = I.
\end{aligned}
$$

$\square$

Proposition A.3.4 *The power basis $\{t^i I\}$ and the Bernstein basis $\{B_i^n I\}$ are related by*

$$B_k^n(t) = \sum_{i=k}^{n} (-1)^{i-k} \binom{n}{i} \binom{i}{k} t^i I$$

and

$$t^k I = \sum_{i=k-1}^{n-1} \frac{\binom{i}{k}}{\binom{n}{k}} B_i^n(t).$$

Proof. Follows from the standard Bernstein polynomials. $\square$

Proposition A.3.5

$$\frac{d}{dt} B_k^n(t) = n(B_{k-1}^{n-1}(t) - B_k^{n-1}(t)).$$

Proof. Follows from the standard Bernstein polynomials. $\square$

A.4   A Theory of Matrix Bézier Curves

A.4.1   Matrix Polynomials and Matrix Bézier Curves

With the framework developed within, we may now proceed to examine the curves of the form

$$F(A) = \sum_{i=0}^{m} B_i^m(A)B_i, \tag{A.4}$$

where we call the B_i *matrix control points*. Any curve of the form of equation A.4 we refer to as a *Matrix Bézier curve* of degree m.

Note that the above curve is devoid of the notation $f(\underbrace{M, \ldots, M}_{m-k}, \underbrace{N, \ldots, N}_{k})$, despite our development of curves using the polar blossom f. The reason for this is that Matrix Bézier curves as defined above differ from matrix polynomial curves in certain matrix vector spaces. This fact is best illustrated by considering the following example:

Example A.4.1 *Let $F(X) = X^2$. Consider the subspace $\mathfrak{so}(3)$ of $M_{3\times3}$, the Lie algebra of $SO(3)$. This subspace consists of all skew-symmetric 3×3 matrices, i.e. those A for which:*

$$A^T + A = 0.$$

Let $X \in \mathfrak{so}(3)$. Observe that

$$F(X)^T = (X^2)^T = (X^T)^2 = (-X)^2 = X^2 = F(X).$$

Hence, for nontrivial X, $F(X) \notin \mathfrak{so}(3)$.

Hence, in the affine matrix space $\mathfrak{so}(3)$ we note that the above example shows that matrix polynomial curves and Matrix Bézier curves do not necessarily correspond. Matrix Bézier curves are affine combinations of the control points $\{B_i\}$, and thus will always lie in the space. We may say then that the class of Matrix Bézier

curves is *closed* in the affine space of definition. On the other hand, the above example shows that $F(X) = X^2$ maps elements from $\mathfrak{so}(3)$ to elements not in $\mathfrak{so}(3)$. Thus, the class of matrix polynomials is not closed in the affine space of definition. Specifically we observe that $F(X)$ may not be represented in the form of equation A.4, even though Matrix Bézier curves of degree 2 are well defined.

Thus, we see that in certain affine spaces, the notion of Matrix Bézier curves and matrix polynomial curves diverges. This is certainly not the case in the classical theory. Under addition and scalar multiplication, we observe that matrices operate exactly like points in a vector space (or and thus affine space). The difference, then, between polynomials and matrix polynomials is the matrix product. Hence, we may view the difference between this theory and the classical one as due to the richer structure of the matrix product.

Finally, we note from the preceeding sections, we observe that the definition of Matrix Bézier curves is no more powerful than that which is obtained as viewing the control matrices, the $\{B_i\}$, as vectors in a vector space and looking at classical Bézier curves over these points. Thus we see that the theory of Bézier curves obtained through the use of matrix polynomials is no more powerful than the standard theory, even though matrix polynomials produce a different set of curves than those of standard polynomials.

A.4.2   *Properties of Matrix Polynomial Curves*

Here we present some properties of Matrix Bézier curves, which may be seen to follow from results of the classical theory. We present proofs with the matrix formalism for completeness.

Proposition A.4.2 *Affine invariance*

Proof. As barycentric combinations are invariant under affine maps, then proposition A.3.3 confirms this result. □

Proposition A.4.3 *Convex Hull Property*

Proof. Observing that the matrix Bernstein polynomials are all non-negative for $t \in [0, 1]$, proposition A.3.3 shows that each point on a matrix polynomial curve with t in this range is contained in the convex hull of its control points. □

Proposition A.4.4 *Endpoint Interpolation*

Proof. This follows from definition of the matrix polynomial curve evaluated at 0 and 1. □

Proposition A.4.5 *Symmetry*

$$\sum_{i=0}^{n} B_i^n(t)B_i = \sum_{i=0}^{n} B_i^n(1-t)B_{n-i}.$$

Proof. This follows from the easily verified identity

$$B_i^n(t) = B_{n-i}^n(1-t).$$

□

Proposition A.4.6 *Invariance under Barycentric Combinations*

Proof. For $\lambda_1 + \lambda_2 = 1$ and control matrices $\{B_i\}_{i=0}^{n}$, $\{C_i\}_{i=0}^{n}$, we have the following:

$$\sum_{i=0}^{n} B_i^n(t)(\lambda_1 B_i + \lambda_2 C_i) = \lambda_1 \sum_{i=0}^{n} B_i^n(t)B_i + \lambda_2 \sum_{i=0}^{n} B_i^n(t)C_i.$$

□

Proposition A.4.7 *Linear Precision*
If the control matrices $\{B_1, B_2, \ldots, B_n\}$ all lie on some line between matrices M_1 and M_2, then the resulting matrix polynomial curve is a line.

Further, if the control matrices are evenly spaced along the line, then the parameterization is just linear interpolation of the matrices M_1 and M_2.

Proof. The first statement follows since all points on the curve are affine combinations of points on the line. Using the identity

$$\sum_{i=0}^{n} \frac{i}{n} B_i^n(t) = tI,$$

we find that if the B_i are given as

$$B_i = (1 - \frac{i}{n})M_1 + \frac{i}{n}M_2$$

then the resulting matrix polynomial curve will be given by

$$(1 - t)M_1 + tM_2.$$

Thus the second statement follows. $\square$

Proposition A.4.8 *Pseudolocal Control*

Proof. As each Bernstein matrix polynomial has it's only maxima at $t = i/n$, the greatest contribution of the control matrix B_i may be found at this point. $\square$

For the following proposition, we need a definition of derivative. Thus we have the following:

Definition A.4.9 *Let F be a matrix polynomial from some matrix vector space (viewed as an affine space) $V \to V$. Let it have polar form $f : V^m \to V$. Define the derivative $DF(A)$ as*

$$\lim_{t \to 0} \frac{F(A + tU) - F(A)}{t},$$

with

$$U = |N - M|^{-1}(N - M).$$

Proposition A.4.10

$$DF(A) = \frac{m}{|N - M|} \overrightarrow{f(\underbrace{A, \ldots, A}_{m-1}, M)f(\underbrace{A, \ldots, A}_{m-1}, N)}.$$

Proof. Following the conventions of Gallier in [10], we let $\widehat{F}$ be the homogenized version of F. As these two functions agree on V, we have

$$F(A + tU) - F(A) = \widehat{F}(A + tU) - \widehat{F}(A).$$

By definition of the polar form, we obtain

$$\widehat{F}(A + tU) = \widehat{f}(\underbrace{A + tU, \ldots, A + tU}_{m}),$$

where here $\widehat{f}$ is the homogenized version of f.

Thus, our difference becomes

$$\widehat{F}(A + tU) - \widehat{F}(A) = \widehat{F}(A + tU) = \widehat{f}(\underbrace{A + tU, \ldots, A + tU}_{m}) - \widehat{f}(\underbrace{A, \ldots, A}_{m}),$$

which reduces through the multilinearity and symmetry to

$$\widehat{F}(A+tU) - \widehat{F}(A) = \widehat{F}(A+tU) = m\,t\,\widehat{f}(\underbrace{A, \ldots, A}_{m-1}, U) + \sum_{k=2}^{k=m} \binom{m}{k} t^k\, \widehat{f}(\underbrace{A, \ldots, A}_{m-k}, \underbrace{U, \ldots, U}_{k}),$$

and hence we have

$$\lim_{t \to 0} \frac{F(A + tU) - F(A)}{t} = m\,\widehat{f}(\underbrace{A, \ldots, A}_{m-1}, U).$$

If we substitute U with its definition, we obtain

$$DF(A) = m|N - M|^{-1}\,(\widehat{f}(\underbrace{A, \ldots, A}_{m-1}, N) - \widehat{f}(\underbrace{A, \ldots, A}_{m-1}, M)).$$

As $\widehat{f}$ agrees with f on elements of V, this becomes

$$m|N - M|^{-1}(f(\underbrace{A, \ldots, A}_{m-1}, N) - (\underbrace{A, \ldots, A}_{m-1} M),$$

or viewed as a vector

$$\frac{m}{|N - M|}\overrightarrow{f(\underbrace{A, \ldots, A}_{m-1}, M)f(\underbrace{A, \ldots, A}_{m-1}, N)}.$$

$\square$

Proposition A.4.11 *Degree Elevation formula*

$$F(A) = \sum_{i=0}^{n} \frac{n+1-i}{n+1} B_i^{n+1}(A) B_i + \sum_{i=0}^{n} \frac{i+1}{n+1} B_{i+1}^{n+1}(A)$$

Proof. This follows from standard Bernstein identities. □

Thus, we observe that we can view a matrix polynomial curve of degree m as a matrix polynomial curve of degree $m + 1$. The corresponding control points are given by

$$B_i^{(1)} = \frac{i}{n+1} B_{i-1} + \left(1 - \frac{i}{n+1}\right) B_i.$$

A.5   Future Work

A.5.1   Applications

We developed this theory in the hopes that it would apply to curve design on matrix Lie groups. For this purpose, the theory in the main body of the thesis is perhaps better suited. However, the theory in this appendix gives a very nice formulation for curve design in matrix vector spaces. Indeed, in the space of all linear transforms on a finite dimensional vector space ($M_{n \times n}(\mathbb{R})$), this theory is quite satisfactory.

Path planning in the space of linear transforms may find applications in computer graphics. Indeed, instead of considering the question of rigid motion as we do through most of the thesis, it is quite feasible that one might desire to animate an object transforming smoothly under a series of linear transforms. In such a situation, the above theory becomes quite useful.

A.5.2   Further Questions

Beyond finding further application for this work, much work remains that might be done. If one considers the motivating construction of the first example, in which

a non-symmetric polar form was generated, we note that this construction differs from the final definition that was settled on for the polar form of a matrix polynomial. Our focus on defining a symmetric polar form essentially reduced the analysis to that of the classical case; instead, it may be worth pursuing the non-symmetric construction to see if a computationally tenable and more powerful theory might be developed.

A second major research question is whether a computationally efficient formalism similar to the Matrix Bézier form exists which will always agree with the matrix polynomials. This problem, of course, is not completely well defined, adding yet more avenues for research.

Bibliography

[1] C. Altafini. The de Casteljau algorithm on $SE(3)$. *Lect Note Contr Inform Sci*, 258:23–34, 2001.

[2] Andrew Baker. *Matrix Groups: An Introduction to Lie Group Theory*. Springer Undergraduate Mathematics Series. Springer, 1 edition, 2001.

[3] Alan H. Barr, Bena Currin, Steven Gabriel, and John F. Hughes. Smooth interpolation of orientations with angular velocity constraints using quaternions. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 313–320. ACM Press, 1992.

[4] C. Belta and V. Kumar. Euclidean metrics for motion generation on $SE(3)$. *Proc. Inst. Mech. Eng. Part C-J. Eng. Mech. Eng. Sci.*, 216:47–60, 2002.

[5] C. Belta and V. Kumar. An SVD-based projection method for interpolation on $SE(3)$. *IEEE Transaction on Robotics and Automation*, 18(3):334–345, 2002.

[6] Manfredo Perdigão do. Carmo. *Riemannian Geometry, trans. Francis Flaherty*. Birkhäuser, 1992.

[7] C. Chevalley. *Theory of Lie Groups*. Princeton University Press, 1946.

[8] P. Crouch, G. Kun, and F. Silva Leite. The de Casteljau algorithm on Lie groups and spheres. *Journal of Dynamical and Control Systems*, 5(3):397–429, 1995.

[9] G. Farin. *Curves and Surfaces for CAGD: A Practical Guide*. Academic Press, 5 edition, 2001.

[10] J. Gallier. *Curves and Surfaces in Geometric Modeling: Theory and Algorithms*. Morgan Kaufmann Publishers, 2000.

[11] J. Gallier. Computing exponentials of skew symmetric matrices and logarithms of orthogonal matrices. August 2002.

[12] A. Karger and J. Novak. *Space Kinematics and Lie Groups*. Gordon and Broach, 1985.

[13] F. C. Park and Bahram Ravani. Bézier curves on Riemannian manifolds and Lie groups with kinematics applications. *Transactions of the ASME*, 117:36–40, 1995.

[14] F. C. Park and Bahram Ravani. Smooth invariant interpolation of rotations. *ACM Transactions on Graphics (TOG)*, 16(3):277–295, 1997.

[15] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254. ACM Press, 1985.

[16] Michael Spivak. *A Comprehensive Introduction to Differential Geometry*, volume 1. Publish or Perish, Inc., 2 edition, 1979.

[17] J.P. Ward. *Quaternions and Cayley Numbers: Algebra and Applications*. Kluwer Academic Publishers, 1 edition, 1997.