Report on Relaxed Jordan Canonical Form for Computer Animation and Visualization

Allen Van Gelder
University of California, Santa Cruz
http://www.cse.ucsc.edu/∼avg

Jan. 30, 2009

Abstract

Relaxed Jordan canonical form (RJCF) is introduced for three-dimensional linear transformations, and applied to various problems in computer animation and visualization, especially problems involving matrix exponentials and matrix logarithms. In all cases our procedures use closed-form expressions, so they are not iterative, whereas the corresponding procedures that have been published in the graphics literature are iterative. New numerically robust procedures are presented for Alexa's "linear" interpolation between linear transformations, based on a matrix Lie product, for polar decomposition, and other techniques that require matrix exponentials and matrix logarithms. Extensions to cases not covered by the published procedures are also discussed, particularly for matrices with less than full rank. There are also applications to analysis of vector fields and tensor fields in three dimensions. Besides providing a basis for numerical procedures, the geometric interpretations of RJCF often provide useful analytical insights.

RJCF is based on new numerically stable procedures to derive a canonical coordinate system for a two or three-dimensional linear transformation. The coordinate transformation has unit determinant and is composed of meaningful building blocks: shear, rotation, nonuniform scale. The derivation uses the new concept of "outer root" of a cubic polynomial, and properties of the characteristic and minimal polynomials. New closed-form procedures to compute eigenvectors are developed, which use fewer operations than eigenvector routines found in texts. Accuracy and stability are examined empirically. Tests indicate that the new method is an order of magnitude faster than general methods, as well as being more accurate.

The three-dimensional problem is separated into independent one-dimensional and two-dimensional problems. The outer root's eigenspace is one-dimensional, collinear with the outer eigenvector. A simple calculation determines the two-dimensional invariant subspace of the other eigenvalues, which need not be orthogonal to the outer eigenvector and may be complex.

Keywords and Phrases:

Three-dimensional linear transformations, relaxed Jordan canonical form, outer roots, eigenvalues, eigenvectors, generalized eigenvectors, invariant subspaces, characteristic polynomials, minimal polynomials, Cayley-Hamilton theorem, matrix animation, matrix Lie product, quaternions, polar decomposition, slerp, matrix exponential, matrix logarithm, streamline topology, critical points, linear differential equations, tensor fields, numerical methods, Cardano equations, interpolation, numerical methods.

1 Introduction

In computer graphics animation it is often desirable to interpolate in the space of linear transformations, particularly three-dimensional linear transformations. The common 4×4 matrix representation of affine 3-D transformations is usually treated as a 3×3 matrix for the linear part and a 3-vector for the translational part, which are handled independently [SD92]. Naive methods that are not invariant under orthonormal changes in the coordinate system were quickly found to be unsatisfactory, such as element-by-element interpolation in matrices and component-by-component interpolation of Euler angles. Several methods have been proposed over the years for general 3-D matrix

interpolation, with *polar decomposition* perhaps having the best theoretical foundation among older methods [SD92]. Recently, methods based on matrix logarithms and matrix exponentials have received increased attention [MLS94, KKS95, PR97, Gra98, BM98, Ale02].

One of the technical problems with most of the methods cited is that one or more key matrix operations rely on an iterative procedure whose behavior has not been analyzed and comes with no guarantees of cost or accuracy (at least, no analyses have been published). One of the main contributions of this paper is to provide closed-form procedures for these key matrix operations. The user of operations that previously relied on an iterative procedure may benefit from our methodology in three ways: (1) our procedure is 17–20 times faster than an off-the-shelf eigen-decomposition procedure; (2) our procedure is more accurate than an off-the-shelf eigen-decomposition procedure; (3) the user's previously iterative procedure can now be performed in closed form, avoiding the hassles of nonconvergence and choosing tolerance parameters.

We introduce *relaxed Jordan canonical form* (RJCF) for three-dimensional linear transformations and give a closed-form procedure to find it for any 3×3 matrix. RJCF is of interest for computer graphics because it can be computed efficiently and accurately and it can be used to compute matrix exponentials and matrix logarithms. We show how it can be used for various interpolation schemes, including Alexa's "linear combination" method, which is based on a matrix Lie product [Ale02], the polar decomposition proposed by Shoemake and Duff [SD92], as well as singular-value decomposition. Previous literature offered only iterative procedures, and in most cases the procedures depended on nonzero determinants. RJCF is also applicable to the problem of tracing parallel vectors, and may be useful wherever spectral analysis (eigenvalues and eigenvectors) is needed.

Besides providing closed-form procedures, another contribution is to show how RJCF serves as a basis for extending several of the above-mentioned techniques. Although the techniques of matrix Lie product and polar decomposition were "defined" for boundary cases that have zero determinants, the published procedures do not handle these cases, and do not address behavior as a system becomes ill-conditioned. RJCF also provides a seamless way to handle such boundary cases because it never relies on having a nonzero determinant. Having the canonical form in hand permits other extensions to be considered easily.

Three-dimensional linear transformations play a pivotal role in physics and other fields, as well as computer graphics. Thus they frequently play a role in scientific visualization. The roles of eigenvalues and eigenvectors in analysis of physical systems are well known. RJCF assists in these tasks by providing a *natural* coordinate transformation into the "relaxed" canonical system. This transformation is real-valued and has unit determinant, meaning that it preserves volume. By default, it stays as close as possible to the original coordinate system, there are choices to allow it to stay close to a previously calculated coordinate system, instead. When the matrix has a pair of complex eigenvalues, as is often the case, only one column of the RJCF transformation is an eigenvector, but the other two columns have a natural interpretation as "principal axes" of a related ellipse.

More generally, properties of RJCF may be useful to analyze the behavior of other well known operations in 3-D, or to provide computational procedures with known accuracy for them. Several examples are given in the paper as applications of RJCF.

The analysis of n-dimensional linear transformations in terms of eigenvalues, eigenvectors, and generalized eigenvectors has been well studied for both theory and numerical methods, and many linear-algebra packages are available. The thesis of this paper is that further gains can be made by using the special properties of 3×3 matrices. The springboard for most of the improvements is the observation that the characteristic polynomial for a 3×3 matrix, a cubic polynomial, has a distinguished real root, called the *outer root*, which can be calculated accurately and quickly, and which can be exploited to simplify the analysis of the transformation represented by the matrix.

1.1 Relaxed vs. Strict Canonical Form

We use the adjective "relaxed" because true (strict) Jordan canonical form has the property that two matrices are related by a similarity transformation (see Section 1.4) if and only if they have *exactly* the same Jordan canonical form. The set of matrices that are related by a similarity transformation form an equivalence class. Strictly speaking, a canonizing procedure should transform *all* elements of an equivalence class into the same element, which is the canonical form for that equivalence class. Thus the test to see if two elements are in the same equivalence class is to see if their canonical forms are identical. This test is not important in any of the applications that we consider or know about.

A modified version of the original Jordan canonical form is called *real Jordan form* [HS74]. The original Jordan canonical form of a real matrix might be complex. However, real Jordan form is restricted to real matrices, while

maintaining the property that two real matrices are related by a similarity transformation if and only if they have the same real Jordan form. The RJCF procedure produces a matrix that differs from the real Jordan form at most by a permutation of coordinate indices in the most common practical cases. However, when the underlying transformation involves a shear component, the differences are more extensive.

It is known that the Jordan canonical form of a matrix is not a continuous function of its elements, although the eigenvalues of a matrix are continuous functions [GVL96, problems 7.1.5–6]. The same holds for the strict real Jordan form. However, RJCF offers a choice of forms for a given matrix, so that there is some choice that is continuous locally in most, but not all, regions of matrix space. See Section 2.6 for more discussion.

1.2 Organization of the Paper

The remainder of this section presents our notation and reviews some relevant linear algebra. The main geometric and algebraic ideas behind RJCF are presented in Section 2, while technical details and numerical procedures are deferred to the appendices. The analytical power of RJCF is demonstrated in Section 3 by using it to characterize the solutions of the problem of detecting parallel vectors in two affine vector fields. Application of RJCF to the calculation of matrix exponentials and matrix logarithms is covered in Section 4, and the more specific application to Alexa's matrix Lie product operator is explained in Section 5. The role of RJCF in computing polar decomposition is discussed in Section 6. Besides presenting new computational procedures, a theorem about polar decomposition of matrices with rank deficiency of one is given, and a new method of blending is proposed. Unlike most of the paper, the results in this section generalize to higher dimensions, but may be impractical for much higher dimensions. Section 7.1 applies RJCF to calculation of the exponential map, with emphasis on handling ill-conditioned situations accurately. Section 7 applies RJCF to some problems involving rotations. Section 8 briefly compares the polar decomposition and the matrix Lie product methods for interpolation with attention to behavior on rotations and on ill-conditioned matrices. Applications to linear differential equations are also briefly discussed in Section 9. Performance issues are examined in Section 10, with attention to both accuracy and speed. The paper concludes in Section 11, but many technical issues are elaborated upon in the appendices.

1.3 Notation

Matrices and linear transformations are denoted by boldface capital letters. Vectors are denoted by boldface lowercase Roman and Greek letters. Normally matrices are square and vectors are column vectors. The dimensions are inferred by context, usually 3×3 and 3×1. Specific row vectors are denoted by their component values in the form $[v_0, v_1, v_2]$. Superscript T denotes transpose of a vector or matrix.

A special notation for *similarity transformations* uses this symbol: $\overset{\mathbf{T}}{\rightsquigarrow}$. See Definition 1.1 and Table 1 for the definition and usage.

Matrices and vectors are indexed beginning at 0 and index expressions are always understood to be modulus the matrix or vector dimension, usually 3. $\mathbf{M}_j$ denotes the j-th *column* of matrix $\mathbf{M}$. $\mathbf{M}^{(i)}$ and $\mathbf{x}^{(i)}$ denote the i-th *row* of $\mathbf{M}$ and the i-th component of vector $\mathbf{x}$. However M_{ij} denotes the element in row i, column j of $\mathbf{M}$. It is sometimes convenient to consider a matrix as a "row vector" of its columns, or a "column vector" of its rows.

$$\mathbf{M} \;=\; \left[\mathbf{M}_0, \mathbf{M}_1, \mathbf{M}_2\right] \;=\; \begin{bmatrix} \mathbf{M}^{(0)} \\ \mathbf{M}^{(1)} \\ \mathbf{M}^{(2)} \end{bmatrix} \tag{1}$$

The standard basis vectors $\mathbf{e}_i$ have 1 in the i-th component and 0 in the remaining components. $\mathbf{I}$ is the identity matrix; $\mathbf{0}$ is the zero matrix.

The cross product is denoted by $\mathbf{p} \times \mathbf{q}$ and the dot product is $\mathbf{p} \cdot \mathbf{q} = \mathbf{p}^T \mathbf{q} = \mathbf{q}^T \mathbf{p}$. The determinant of $\mathbf{M}$ is denoted by $\det(\mathbf{M})$. The *Euclidean length* of $\mathbf{v}$ and *Frobenius norm* of $\mathbf{M}$ are denoted by $|\mathbf{v}|$ and $|\mathbf{M}|$, respectively:

$$|\mathbf{v}| \;=\; \sqrt{\mathbf{v} \cdot \mathbf{v}} \tag{2}$$

$$|\mathbf{M}| \;=\; \sqrt{|\mathbf{M}_0|^2 + |\mathbf{M}_1|^2 + |\mathbf{M}_2|^2} \;=\; \sqrt{\textstyle\sum_{i,j} \mathsf{M}_{ij}^2} \tag{3}$$

Exponentiation is denoted by both $e^{\mathbf{A}}$ and $\exp(\mathbf{A})$, with the latter being favored for complicated exponent expressions.

Two vectors $\mathbf{p}$ and $\mathbf{q}$ are *orthogonal* if $\mathbf{p} \cdot \mathbf{q} = 0$ and are *orthonormal* if, in addition, $|\mathbf{p}| = |\mathbf{q}| = 1$. A set of vectors is *orthonormal* if each pair in the set is. A square matrix is *orthonormal* if its columns form an orthonormal set of vectors; in this case its rows are also orthonormal. Some writings use the term *orthogonal* for such sets and matrices; we use *orthonormal* to avoid any ambiguity.

1.4 Miscellaneous Facts of Linear Algebra

This subsection reviews some basics of linear algebra that are used in the paper. The facts in this subsection may be found in, or easily derived from, intermediate texts on linear algebra [HS74, GVL96]. Although many of the statements in this section hold for n dimensions, many do not, so we confine attention to three and fewer dimensions when they are simpler.

Let $\mathbf{M}$ be a 3×3 matrix. The *determinant* of $\mathbf{M}$ is the *triple scalar product*:

$$\det(\mathbf{M}) \;=\; \mathbf{M}_0 \cdot \mathbf{M}_1 \times \mathbf{M}_2 \;=\; \mathbf{M}_0 \times \mathbf{M}_1 \cdot \mathbf{M}_2. \tag{4}$$

The *trace* of $\mathbf{M}$ is the sum of its diagonal elements. The i-th *principal minor* of $\mathbf{M}$ is the matrix that results by deleting row i and column i. $\mathbf{M}$ is called *singular* if $\det(\mathbf{M}) = 0$, otherwise it is *nonsingular* and has an inverse, which is given by a special case of Cramer's rule:

$$\mathbf{M}^{-1} \;=\; \frac{1}{\det(\mathbf{M})} \begin{bmatrix} (\mathbf{M}_1 \times \mathbf{M}_2)^T \\ (\mathbf{M}_2 \times \mathbf{M}_0)^T \\ (\mathbf{M}_0 \times \mathbf{M}_1)^T \end{bmatrix}; \tag{5}$$

If $\mathbf{M}\boldsymbol{\xi} = \lambda\boldsymbol{\xi}$ for some scalar λ and nonzero vector $\boldsymbol{\xi}$, both of which might be complex, then λ is an *eigenvalue* and $\boldsymbol{\xi}$ is an *eigenvector* of $\mathbf{M}$. All 3×3 matrices have at least one real eigenvalue and an associated real eigenvector; the other two eigenvalues may be real also or may be a complex conjugate pair. In the case of complex eigenvalues, there are complex eigenvectors for each complex eigenvalue. In the case of three real eigenvalues, there are several possibilities for eigenvectors and *generalized* eigenvectors (see Appendix C).

The *rank* of a matrix is the number of linearly independent columns or rows. The *nullity* of a square matrix $\mathbf{M}$ is the difference between its dimension and its rank. The subspace that is mapped to 0 by $\mathbf{M}$ called the *kernel* of $\mathbf{M}$ and its dimension equals the nullity of $\mathbf{M}$. For two square matrices $\mathbf{M}_1$ and $\mathbf{M}_2$:

$$\mathit{nullity}(\mathbf{M}_1\mathbf{M}_2) \;\leq\; \mathit{nullity}(\mathbf{M}_1) + \mathit{nullity}(\mathbf{M}_2). \tag{6}$$

The only matrix of rank 0 is $\mathbf{0}$. Thus, if $\mathbf{M}$ is 3×3 and $\mathbf{M}^2 = \mathbf{0}$, then the rank of $\mathbf{M}$ is 0 or 1. This turns out to be very useful in 3-D, in conjunction with the following paragraph, for determination of eigenvectors and generalized eigenvectors.

Rank-one matrices have a special form using the tensor, or outer, product. Let M_{ij} be a nonzero element in the rank-one matrix $\mathbf{M}$. Then

$$\mathbf{M} \;=\; \frac{\mathbf{M}_j \mathbf{M}^{(i)}}{\mathsf{M}_{ij}} \tag{7}$$

The equation holds for any nonzero element, but choosing M_{ij} to be an element of maximum magnitude ensures that $|\mathbf{M}_j|$ is maximal over all columns and $|\mathbf{M}^{(i)}|$ is maximal over all rows.

The *characteristic polynomial* of $\mathbf{M}$ is the scalar polynomial

$$f(x) \;=\; \det(\mathbf{I}x - \mathbf{M}). \tag{8}$$

If $p(x)$ is any polynomial in one scalar variable, then p may be evaluated over 3×3 matrices, e.g., $p(\mathbf{M})$, yielding a 3×3 matrix. The classical *Cayley-Hamilton theorem* states that $f(\mathbf{M}) = \mathbf{0}$, where $f(x)$ is the characteristic polynomial of $\mathbf{M}$. It follows that the roots of $f(x)$ are precisely the eigenvalues of $\mathbf{M}$.

Linear transformations from the Euclidean space $\mathbf{E}^n$ to $\mathbf{E}^n$ can be represented by $n \times n$ matrices. The matrix representation depends on the coordinate system, or the set of basis vectors.

$\mathbf{A} \overset{\mathbf{T}}{\rightsquigarrow}$	is read as "the similarity transformation of **A** based on **T**; i.e., $\mathbf{T}^{-1}\mathbf{A}\mathbf{T}$."
$\mathbf{A} \overset{\mathbf{T}}{\rightsquigarrow} \mathbf{B}$	is read as "**B** is similar to **A** based on transformation by **T**; i.e., $\mathbf{B} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}$."
$\mathbf{A} \rightsquigarrow \mathbf{B}$	is read as "**A** and **B** are similar, since the relationship is symmetric."

Table 1: Notation for similarity transformations in which $\det(\mathbf{T}) = 1$; see Definition 1.1.

Diagonal	Spiral	Simple Shear	Compound Shear
$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & -2 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 3 \\ 0 & 2 & 0 \\ -3 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 4 & 3 \end{bmatrix}$	$\begin{bmatrix} 3 & 0 & -1 \\ 0 & 3 & 0 \\ 0 & 4 & 3 \end{bmatrix}$

Table 2: Examples of Relaxed Jordan Canonical Forms for 3×3 matrices.

Definition 1.1: For an $n \times n$ matrix **A**, if **T** is any nonsingular real $n \times n$ matrix, then $\mathbf{T}^{-1}\mathbf{A}\mathbf{T}$ is a *similarity transformation* on **A**. If, in addition, $\det(\mathbf{T}) = 1$, the notation $\mathbf{A} \overset{\mathbf{T}}{\rightsquigarrow}$ is used for $\mathbf{T}^{-1}\mathbf{A}\mathbf{T}$. **T** is called the *transforming matrix* and may be a matrix expression as well as a single symbol. Two matrices, **A** and **B**, are *similar* if **B** is produced by some similarity transformation on **A** (and vice versa). The notation $\mathbf{A} \overset{\mathbf{T}}{\rightsquigarrow} \mathbf{B}$ is synonymous with $\mathbf{B} = \mathbf{A} \overset{\mathbf{T}}{\rightsquigarrow}$. See Table 1 for suggested phrasing. □

Although similarity transformations are defined for all nonsingular **T**, the notation of Definition 1.1 with an explicit transforming matrix implies that the determinant of the transforming matrix is one. However, if two matrices are similar, then it is always possible to use a transforming matrix whose determinant is one, so the use of the unadorned "$\rightsquigarrow$" is consistent and this operator is reflexive.

A *similarity transformation* on a matrix corresponds to a "change of coordinates". A 3-D coordinate system consists of three linearly independent *basis vectors*, which need not be orthogonal. If two matrices are related by a similarity transformation, then they represent the same linear transformation in different coordinate systems and have the same eigenvalues.

Lemma 1.1: The characteristic polynomial, trace, and determinant are invariant under similarity transformations. ▮

The topological classification of a linear transformation is invariant under similarity transformations [Lad73, KR73], as are many other interesting properties. It is beyond the scope of this paper to describe the topology under consideration, other than to mention that it is related to continuous mappings between families of solutions of linear differential equations. Readers may consult the cited works for complete explanations.

2 Relaxed Jordan Canonical Form

A *relaxed Jordan canonical form* (RJCF) of the 3×3 or 2×2 matrix **A** is a matrix **J** such that $\mathbf{A} \overset{\mathbf{T}}{\rightsquigarrow} \mathbf{J}$ and **J** satisfies certain conditions. (Recall that the notation implies $\det(\mathbf{T}) = 1$.) Since $\mathbf{J} \overset{\mathbf{T}^{-1}}{\rightsquigarrow} \mathbf{A}$, the transforming matrix **T** is often considered part of the relaxed Jordan canonical form for **A**. Certain cases when **A** has three equal eigenvalues are deferred to Appendix C because they are trivial (uniform scale) or very messy and unlikely to occur in practice (compound shear). The 3×3 cases are summarized in Table 2, with examples.

If **A** has a RJCF of the *Diagonal* form, then **A** is called *diagonalizable* and its eigenvalues are simply the diagonal elements of **J**. This form has real eigenvectors that span the space.

If **A** has a RJCF of the *Spiral* form, then the real eigenvalue is the entry of **J** that has no off-diagonal element in its row or column, 2 in the example in Table 2. The other two eigenvalues are a complex conjugate pair, whose real

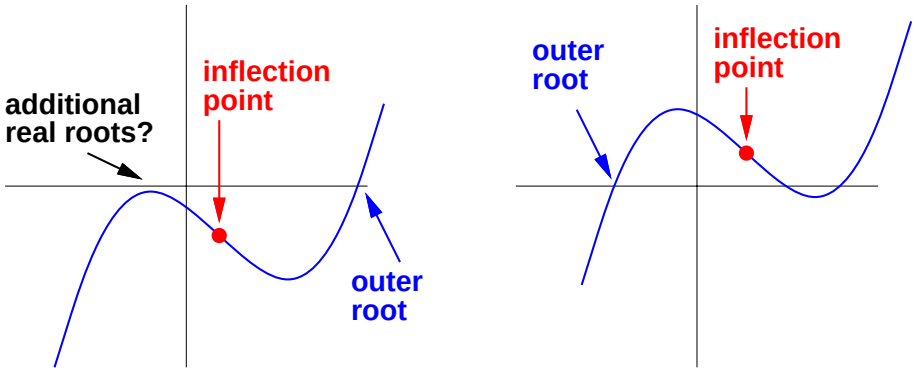

Figure 1: The location of the outer root relative to the inflection point of $f(x)$. Its location does not require knowing how many real roots $f(x)$ has.

part appears on the diagonal and whose imaginary part appears off the diagonal, $-1 \pm 3i$ in the same example. The complex eigenvalues have complex eigenvectors, but RJCF is able to avoid using complex values throughout.

If **A** has a RJCF of the *Simple Shear* form, then its eigenvalues are again simply the diagonal elements, at least two of which are equal, but there are only two independent eigenvectors. This form is actually the composition of a shear and a nonuniform scale. The *Compound Shear* form is discussed in Appendix C.2, and has only one independent eigenvector. It is recognized by the property that after a suitable permutation of the indices, the matrix is upper triangular and both super-diagonal elements are nonzero (e.g., flip 1 and 2 in the example in Table 2). The 2×2 cases are similar, except that a 2×2 compound shear does not exist.

In all cases $\mathbf{A} \overset{\mathbf{T}}{\rightsquigarrow} \mathbf{J}$ represents a similarity transformation on **A** that exposes the eigenvalues in **J** and the eigenvectors and generalized eigenvectors in **T**. (Complex eigenvalues and eigenvectors are easy to calculate from the real entries in **J** and **T**, but RJCF does not require them.)

2.1 Outer Root, Eigenvalue, and Eigenvector

Two key concepts for understanding the procedure to find RJCF are the *outer eigenvalue* and the *invariant subspaces* of a 3×3 matrix **A**. A detailed description of the computational procedures is deferred to Appendix B. We present the main geometric and algebraic ideas here. So that readers may be assured that their animation code will not fall into a black hole by using RJCF, we note that the complete calculation consists of at most 10 square-root calls, 2 calls to trigonometric functions, and 202 miscellaneous floating multiply-adds.

The eigenvalues are the roots of the *characteristic polynomial* of **A**:

$$f(x) \quad = \quad \det(\mathbf{I}x - \mathbf{A}) \; = \; x^3 - tx^2 + sx - d. \tag{9}$$

In this equation, t is the *trace*, d is the *determinant*, and s is the sum of the determinants of the *principal minors*.

Definition 2.1: The *outer root* of a cubic polynomial is the real root that is most distinct in the following senses. When there is only one real root, it is the outer root. When there are three real roots, the one that is furthest from the median root is the outer root; break a tie in favor of the root with maximum absolute value. (The specific case in which there is still a tie: $-\lambda$, 0, λ, is unlikely to occur in computer graphics, but choose the positive root as "outer" in this case.) See Figure 1.

The *outer eigenvalue* is the outer root of the characteristic polynomial. An *outer eigenvector* is an eigenvector of the outer eigenvalue. □

Our interest in identifying the outer root is that its calculation presents the fewest numerical problems. Also, it is often useful to have an eigenvalue of multiplicity one. The numerical analysis community advises generally against using the characteristic polynomial to compute eigenvalues [PFTV88]. However, by computing the outer root first, we have found that we obtain more accurate results for 3×3 matrices than general-purpose eigenvalue routines.

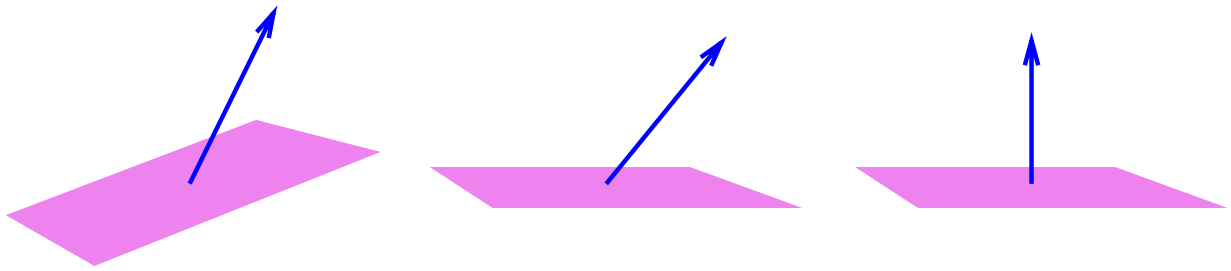

Figure 2: The outer eigenvector and the two-dimensional invariant subspace are not necessarily orthogonal in the original coordinate system (left). A rotation $\mathbf{R}^{-1}$ places the invariant subspace in a coordinate plane (center), then a shear $\mathbf{S}^{-1}$ aligns the outer eigenvector with the third axis (right). Progressing from right to left illustrates the application of $\mathbf{S}$ then $\mathbf{R}$.

It is useful that the outer eigenvalue can be found without knowing whether there are one or three real eigenvalues. Referring to Eq. 9 and Figure 1, the inflection point is $\widetilde{x} = t/3$, i.e., $f''(\widetilde{x}) = 0$. If $f(\widetilde{x})$ and $f'(\widetilde{x})$ have opposite signs the outer root is the minimum root, otherwise it is the maximum root.

The outer root can be found with a specialized form of Cardano's equations for roots of cubic polynomials (see Appendix A.2). The outer root also can be found with a monotonically convergent Newton-Raphson procedure by starting at $\pm(1+|t|+|s|+|d|)$, depending on whether the maximum or minimum root is wanted (see Appendix A.1). Both methods are an order of magnitude faster than using a general (n-D) iterative eigenvalue procedure, such as found in BLAS or matlab.

Definition 2.2: An *invariant subspace* for a linear transformation $\mathbf{A}$ is a set of vectors S that is closed under $\mathbf{A}$; i.e., if $\mathbf{v} \in S$, then $\mathbf{A}\mathbf{v} \in S$. □

For our purposes the important invariant subspace is the 2-D space spanned by real vectors that are linear combinations of the two eigenvectors *other than* the outer eigenvector (the precise characterization is given in Lemma 2.2). These two eigenvectors might be a complex-conjugate pair. For the simple shear, the 2-D space is spanned by an eigenvector and a *generalized eigenvector*. With some abuse of terminology we call this invariant subspace an *eigenspace* or *generalized eigenspace*. (Generalized eigenspaces are needed only in the presence of repeated eigenvalues, and not necessarily even then; they are defined for 2-D in Section 2.5 and for 3-D in Appendix C.)

A key innovation of RJCF is that the transformation into "canonical form" is derived geometrically (see Figure 2). By applying a similarity transformation to $\mathbf{A}$, we map this 2-D invariant subspace into a coordinate plane, and make the outer eigenvector collinear with the normal vector to this plane. This separates the transformation into the direct product of a 1-D transformation (aligned with the outer eigenvector, out of the plane) and a 2-D transformation in the plane. The 2-D transformation can then be analyzed for RJCF in isolation. The remainder of this subsection outlines the calculation of the outer eigenvector and the invariant subspace normal vector. The following subsection outlines how to carry out this separation. Full details and discussion are presented in Appendix B.

Suppose the outer eigenvalue λ_0 (the outer root of $f(x)$) has been found as described in the discussion of Eq. 9. The multiplicity (both algebraic and geometric) of λ_0 is one. Define the quadratic polynomial

$$q(x) \;=\; f(x) \;/\; (x-\lambda_0). \tag{10}$$

Recall that polynomials can be evaluated over square matrices, as well as scalars. Indeed, the Cayley-Hamilton theorem states that

$$f(\mathbf{A}) \;=\; \mathbf{A}^3 - tA^2 + sA - d\mathbf{I} \;=\; \mathbf{0} \tag{11}$$

where the scalars t, s, and d are the same as in Eq. 9; the details of their computation are given by Eq. 86. Using Eq. 10,

$$f(\mathbf{A}) \;=\; (\mathbf{A} - \lambda_0 \mathbf{I})\, q(\mathbf{A}), \tag{12}$$

The analysis of $q(\mathbf{A})$ provides key insights into the spectral structure of $\mathbf{A}$.

Lemma 2.1: Assume **A** does not have three equal eigenvalues. The transformation $q(\mathbf{A})$ has rank one. Indeed, every column of the matrix of $q(\mathbf{A})$ is a scalar multiple of ξ_0, the outer eigenvector of **A**.

Proof: Because λ_0 has multiplicity one, its only eigenvectors are multiples of ξ_0. Suppose the *i*-th column of $q(\mathbf{A})$ were *not* such a multiple. Then the standard basis vector $\mathbf{e}_i$ would not be mapped to an eigenvector of λ_0 by $q(\mathbf{A})$. It would follow from Eq. 12 that

$$f(\mathbf{A})\,\mathbf{e}_i \;=\; (\mathbf{A}-\lambda_0 I)\,q(\mathbf{A})\,\mathbf{e}_i \;\neq\; \mathbf{0},$$

violating the Cayley-Hamilton theorem Eq. 11. ∎

Since $q(\mathbf{A})$ is a rank-one matrix, it has the form of a scalar multiple γ times an outer product of two 3-vectors:

$$q(\mathbf{A}) \;=\; \gamma\xi_0\eta^T, \tag{13}$$

where ξ_0 is an outer eigenvector. The significance of η is established next. (Note that ξ_0 and η are not necessarily unit length; in fact, for numerical purposes, it is best to choose the column and row of maximum magnitude in $q(\mathbf{A})$ for these vectors.)

Let $q(\mathbf{A})_{ij}$ be an element of maximum magnitude; call it q_{max}. Break ties arbitrarily. For numerical purposes, we choose ξ_0 to be the *j*-th column of $q(\mathbf{A})$.

1. If the *i*-th row of $q(\mathbf{A})$ satisfies $q(\mathbf{A})^{(i)}\,\xi_0 > 0$, then we choose $\eta^T = q(\mathbf{A})^{(i)}$ and $\gamma = 1/q_{max}$;
2. otherwise, $\eta^T = -q(\mathbf{A})^{(i)}$ and $\gamma = -1/q_{max}$.

Note that ξ_0 is a column of maximum magnitude and η^T is a (possibly negated) row of maximum magnitude. Also, all rows of $q(\mathbf{A})$ are scalar multiples of η^T. Thus Eq. 13 is satisfied.

Next we show that η is orthogonal to the two-dimensional *invariant subspace* of the remaining eigenvalues, say λ_1 and λ_2 (see Definition 2.2). A corollary is that ξ_0 is *not* orthogonal to η, under the condition that **A** does not have three equal eigenvalues.

Lemma 2.2: Assume **A** does not have three equal eigenvalues. Let η be defined as in Eq. 13. Let λ_1 and λ_2 be the two eigenvalues of **A** other than the outer eigenvalue. Let $\perp\eta$ denote the 2-D subspace orthogonal to η.

1. $\perp\eta$ is invariant under **A**; i.e., if $\mathbf{v} \in \perp\eta$, then $\mathbf{Av} \in \perp\eta$.
2. $\perp\eta$ contains the real and imaginary parts of all vectors that are linear combinations of eigenvectors of λ_1 and λ_2.

Proof: (1) Invariance is immediate by Eq. 13. (2) For containment, let ξ_1 and ξ_2 be eigenvectors of λ_1 and λ_2, But λ_1 and λ_2 are the roots of $q(x)$. Therefore $q(\mathbf{A})\,\xi_1 = \mathbf{0}$ and $q(\mathbf{A})\,\xi_2 = \mathbf{0}$. Since $q(\mathbf{A}) = \xi_0\eta^T$ and $\xi_0 \neq \mathbf{0}$, it follows that $\eta^T\xi_1 = 0$ and $\eta^T\xi_2 = 0$. ∎

If eigenvectors of λ_1 and λ_2 span a 2-D space, this space coincides with $\perp\eta$ in Lemma 2.2. This is also the case if a generalized eigenvector is required, but the details are deferred to Section 2.5.

For matrix **A** below, Example I.1 in Appendix I shows the calculation of the outer eigenvalue, $\lambda_0 = 1$, the outer eigenvector ξ_0, and the normal vector η. The two vectors make an angle of 45°.

$$\mathbf{A} \;=\; \begin{bmatrix} -15 & -16 & -20 \\ 44 & 45 & 19 \\ 16 & 16 & 21 \end{bmatrix} \qquad \xi_0 \;=\; \begin{bmatrix} 576 \\ -576 \\ 0 \end{bmatrix} \qquad \eta \;=\; \begin{bmatrix} 576 \\ 0 \\ 576 \end{bmatrix}. \tag{14}$$

2.2 Separation into 1-D and 2-D Problems

We are now prepared to separate the three-dimensional problem into a one-dimensional problem and a two-dimensional problem. The quantities ξ_0, η, γ, *i* and *j* have been computed as described in the paragraph before Lemma 2.2. A composition of a rotation and a shear aligns η with the nearest coordinate axis and aligns ξ_0 with η. The sequence is suggested by Figure 2.

The idea is first to rotate the coordinate frame so that $\boldsymbol{\eta}$ aligns with an axis. Specifically, $\mathbf{R}^{-1}\boldsymbol{\eta}$ is aligned with the nearest coordinate axis. (To maintain continuity of a sequence of transformations a different axis may be chosen.) The invariant subspace for λ_1 and λ_2 is now contained in the coordinate plane orthogonal to $\boldsymbol{\eta}$.

Next, apply a shear transformation "parallel" to this coordinate plane to align $\boldsymbol{\xi}_0$ with the same axis as $\boldsymbol{\eta}$, while leaving the coordinate plane invariant. Specifically, $\mathbf{S}^{-1}\mathbf{R}^{-1}\boldsymbol{\xi}_0$ is aligned with this axis. This axis is the eigenvector direction for λ_0 in the canonical coordinate system, so the separation is accomplished. The rotation and shear transformations, **R** and **S**, respectively, each have determinant one. The transforming matrix that separates the problem is **RS**, and the separated form of the matrix is $\mathbf{A} \overset{\mathbf{RS}}{\rightsquigarrow}$. The equations are given with optimizations in Appendix B.1, Eq. 87–92.

Continuing with matrix **A** and vectors $\boldsymbol{\xi}_0$ and $\boldsymbol{\eta}$ in Eq. 14, Example I.2 in Appendix I shows the calculation of the separating matrix **RS**, shown below. The outer-root eigenvalue ($\lambda_0 = 1$) has been separated from the rest of the transformation.

$$(\mathbf{RS}) = \begin{bmatrix} \sqrt{2} & 0 & -1/\sqrt{2} \\ -\sqrt{2} & 1 & 0 \\ 0 & 0 & 1/\sqrt{2} \end{bmatrix} \qquad \mathbf{A} \overset{\mathbf{RS}}{\rightsquigarrow} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 45 & -25/\sqrt{2} \\ 0 & 32/\sqrt{2} & 5 \end{bmatrix} \tag{15}$$

With the separation accomplished, the coordinate plane containing the eigenspace for λ_1 and λ_2 is transformed into canonical form by the methods described in Section 2.3.

The partitioned form of the matrix is $\mathbf{A} \overset{\mathbf{RS}}{\rightsquigarrow}$, with λ_0, the outer eigenvalue, in the (j, j)-element. We write matrix diagrams as though $j = 0$ for simplicity, but wherever these diagrams show indices 0, 1, 2, the code uses, j, $j+1$, and $j+2$ (both mod 3, as usual).

$$\mathbf{A} \overset{\mathbf{RS}}{\rightsquigarrow} = \begin{bmatrix} \lambda_0 & 0 & 0 \\ 0 & & \\ 0 & & \mathbf{B} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \tag{16}$$

The rows and columns with indices $j+1$ and $j+2$ comprise the 2x2 matrix **B** that remains to be canonized. Row j and column j are final.

There are several ways to put **B** into RJCF, i.e., find $\mathbf{J}_B$ and $\mathbf{U}_B$ such that $\mathbf{B} \overset{\mathbf{U_B}}{\rightsquigarrow} \mathbf{J}_B$. The procedure derived in Section 2.3 again uses a geometrical approach. Then $\mathbf{J}_B$ and $\mathbf{U}_B$ are extended to 3-D by

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & & \\ 0 & & \mathbf{U}_B \end{bmatrix} \qquad \mathbf{J} = \begin{bmatrix} \lambda_0 & 0 & 0 \\ 0 & & \\ 0 & & \mathbf{J}_B \end{bmatrix} \tag{17}$$

Define $\mathbf{T} = \mathbf{RSU}$ and it follows that

$$\mathbf{A} \overset{\mathbf{T}}{\rightsquigarrow} \mathbf{J} \tag{18}$$

is the RJCF for **A**. Explicit expressions for **J** and the matrix inverses are worked out from Eq. 98–105 in Section B.2. Example I.5 in Appendix I shows a complete example.

Although the foregoing procedure might seem protracted, it is at least an order of magnitude faster than than a general "eigen-package" such as BLAS (matlab provides an interface to BLAS for the eigen-package). It is also more accurate. Section 10 gives experimental results to support these claims. The roles of the key matrices **J** and **T** in applications are discussed in Section 4 and following sections.

2.3 Two-Dimensional Canonical Coordinates

This section shows how to transform a 2×2 matrix into a canonical coordinate system with a short sequence of familiar area-preserving transformations: rotations and nonuniform scales. The defective case, in which there is only one eigenvector, and nearly defective cases, in which the eigenvectors are nearly parallel, are treated. The transforming matrix and RJCF are continuous functions of the matrix elements in most of the 4-D space. We believe this is the first 2-D canonizing procedure with this degree of continuity. The equations and details are deferred to Appendix B.2, which includes a brief description of the subspace where discontinuity occurs.

$\begin{bmatrix} J_{11} & 0 \\ 0 & J_{22} \end{bmatrix}$	$\begin{bmatrix} J_{11} & J_{12} \\ -J_{12} & J_{11} \end{bmatrix}$	$\begin{bmatrix} J_{11} & 0 \\ J_{21} & J_{11} \end{bmatrix}$	$\begin{bmatrix} J_{11} & J_{12} \\ 0 & J_{11} \end{bmatrix}$
Diagonal	Spiral	Shear	Shear

Table 3: Relaxed Jordan Canonical Forms for 2×2 matrices.

We use the following notation for the elements of a 2×2 matrix $\mathbf{B}$:

$$\mathbf{B} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \tag{19}$$

Recall that B_{11} in this section corresponds to $B_{j+1,j+1}$ in the 3-D context, etc. In a relaxed Jordan canonical coordinate system the matrix has one of a few special forms shown in Table 3. They correspond fairly obviously to the 3-D forms in Table 2. Thus the problem is, given a matrix $\mathbf{B}$, find a transformation $\mathbf{U}$ such that $\mathbf{B} \overset{\mathbf{U}}{\rightsquigarrow}$ is in one of these forms. In general, $\mathbf{U}$ is not unique.

2.4 Area-Preserving Similarity Transformations

For coordinate transformation $\mathbf{T}$, the similarity transformation $\mathbf{T}^{-1}\mathbf{B}\mathbf{T}$ *preserves area* if the determinant of $\mathbf{T}$ is 1. Recall that the notation $\mathbf{B} \overset{\mathbf{T}}{\rightsquigarrow}$ is used in this case. A canonical coordinate system can be derived with a short sequence of area-preserving transformations of two familiar types, two-dimensional rotations ($\mathbf{R}_\theta$) and nonuniform scales ($\mathbf{S}_w$):

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad \mathbf{R}_\theta^{-1} = \mathbf{R}_\theta^T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{20}$$

$$\mathbf{S}_w = \begin{bmatrix} w & 0 \\ 0 & 1/w \end{bmatrix} \qquad \mathbf{S}_w^{-1} = \begin{bmatrix} 1/w & 0 \\ 0 & w \end{bmatrix} \tag{21}$$

First, we state some important properties of these transformations. The next subsection exploits these properties. Matrices with equal diagonal elements play a central role.

Lemma 2.3: Let $\mathbf{R}$ be a 2-D rotation and let $\mathbf{S}$ be an area-preserving scale, as above.

1. The *difference between* the off-diagonal elements of $\mathbf{B}$ is preserved by $\mathbf{B} \overset{\mathbf{R}}{\rightsquigarrow}$.
2. The *values of* the diagonal elements of $\mathbf{B}$ are preserved by $\mathbf{B} \overset{\mathbf{S}}{\rightsquigarrow}$.

Proof: Direct linear algebra computation. ∎

Lemma 2.4: Let the 2×2 matrix $\mathbf{M}$ be defined by $\mathbf{B} \overset{\mathbf{R}}{\rightsquigarrow} \mathbf{M}$, and let $\mathbf{M}$ have two real, independent eigenvectors.

1. The eigenvectors of $\mathbf{M}$ are bisected by the coordinate axes if and only if the diagonal elements of $\mathbf{M}$ are equal.
2. Suppose the eigenvectors are bisected by the coordinate axes. An area-preserving scale $\mathbf{S}^{-1}$ causes the scaled eigenvectors to be orthogonal and remain bisected by the coordinate axes if and only if the off-diagonal elements of $\mathbf{M} \overset{\mathbf{S}}{\rightsquigarrow}$ are equal.

Proof: Figure 3 shows the idea of the proof, where we use the notation:

$$\mathbf{M} = \begin{bmatrix} a & b \\ e & a \end{bmatrix}, \qquad \mathbf{v} = \begin{bmatrix} X \\ Y \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} X \\ -Y \end{bmatrix}.$$

Note that $\mathbf{M}$ has equal diagonal elements. Let $\mathbf{v}$ be an eigenvector of $\mathbf{M}$. First, we need to show that $\mathbf{u}$ is also an eigenvector of $\mathbf{M}$.

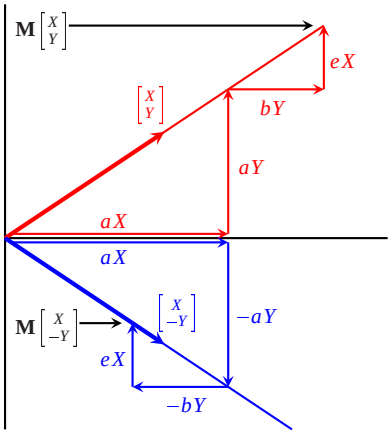

Figure 3: Diagram for proof of Lemma 2.4. The thick red vector denotes $\mathbf{v} = \begin{bmatrix} X \\ Y \end{bmatrix}$, a given eigenvector of $\mathbf{B}$. The thick blue vector denotes $\mathbf{u} = \begin{bmatrix} X \\ -Y \end{bmatrix}$.

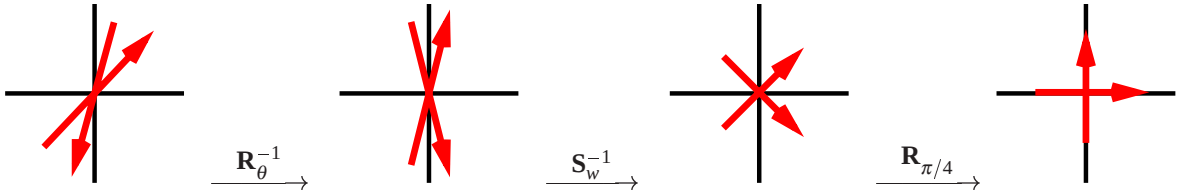

Figure 4: Steps to transform into a 2-D canonical coordinate system. The arrows are equal-length eigenvectors in their initial orientation (left), progressing to alignment with the coordinate axes in the canonical coordinate system (right).

The horizontal and vertical arrows show the construction of $\mathbf{Mv}$ (red) and $\mathbf{Mu}$ (blue). $\mathbf{Mv}$ must be collinear with $\mathbf{v}$ by hypothesis. By congruent triangles, it follows that $\mathbf{Mu}$ must be collinear with $\mathbf{u}$, so $\mathbf{u}$ is also an eigenvector. The diagram shows a, b, and e all positive, but clearly the construction works with $a \leq 0$ and/or b and e both negative.

On the other hand, if $\mathbf{M}_{22}$ is altered to $f > a$ (and $\mathbf{v}$ is still an eigenvector), then the vertical arrows are altered to length fY in the constructions, and e must be reduced to maintain the collinearity of $\mathbf{Mv}$. But then the blue arrows that construct $\mathbf{Mu}$ will end up too low for collinearity with $\mathbf{u}$, so $\mathbf{u}$ is *not* an eigenvector. A similar argument holds for $f < a$, concluding the proof of part (1).

For part (2), if S_w is a nonuniform area-preserving scale, then $\mathbf{S}_w^{-1}\mathbf{v}$ and $\mathbf{S}_w^{-1}\mathbf{u}$ make equal and opposite-sign angles with the X-axis. They can only be orthogonal if that angle is 45°. Also, if $\mathbf{v}$ and $\mathbf{u}$ are eigenvectors of $\mathbf{M}$, then $\mathbf{S}_w^{-1}\mathbf{v}$ and $\mathbf{S}_w^{-1}\mathbf{u}$ are eigenvectors of $\mathbf{M} \overset{\mathbf{S}_w}{\rightsquigarrow}$. Interpreting Figure 3 to show $\mathbf{M} \overset{\mathbf{S}_w}{\rightsquigarrow}$ instead of $\mathbf{M}$ (i.e., b, e, X, Y are quantities after transforming by $\mathbf{S}_w$), we see that the off-diagonal elements of $\mathbf{M} \overset{\mathbf{S}_w}{\rightsquigarrow}$ must be equal. ∎

2.5 Geometric Procedure for Two Dimensions

At a high level the geometric procedure to construct the transforming matrix $\mathbf{U}$ is shown in Figure 4. The equations and details are deferred to Appendix B.2. The figure shows the effect on two real eigenvectors of the successive stages of $\mathbf{U}^{-1}$). However, the equations work in all cases.

If $\mathbf{B} = \lambda\mathbf{I}$ is a scaled identity, it is in RJCF and all vectors are eigenvectors. Assuming this is not the case, the procedure to transform $\mathbf{B}$ into RJCF is:

1. ***Rotate*** $\mathbf{B}$ to $\mathbf{B} \overset{\mathbf{R}_\theta}{\rightsquigarrow}$, where θ in Eq. 20 is chosen to equalize the diagonal elements in the result, per Eq. 98–103 in Appendix B.2.

2. ***Scale*** $\mathbf{B} \overset{\mathbf{R}_\theta}{\rightsquigarrow}$ to $\mathbf{B} \overset{\mathbf{R}_\theta \mathbf{S}_w}{\rightsquigarrow}$, where w in Eq. 21 is chosen to equalize the *magnitudes* of the off-diagonal elements in the result, per Eq. 105. This step "fails" if one of the off-diagonal elements of $\mathbf{B} \overset{\mathbf{R}_\theta}{\rightsquigarrow}$ is zero (or orders of magnitude smaller than the other off-diagonal element), in which case $\mathbf{S}_w$ has the *pro forma* value $\mathbf{I}$. This case is discussed in Section 2.6.

3. ***Rotate*** $\mathbf{B} \overset{\mathbf{R}_\theta \mathbf{S}_w}{\rightsquigarrow}$ to $\mathbf{B} \overset{\mathbf{R}_\theta \mathbf{S}_w \mathbf{R}_\phi}{\rightsquigarrow}$, where the angle ϕ for rotation $\mathbf{R}_\phi$ is chosen based on the off-diagonal elements of $\mathbf{B} \overset{\mathbf{R}_\theta \mathbf{S}_w}{\rightsquigarrow}$ as follows:

   (a) After the rotation and scale, if the off-diagonal elements of the resulting matrix have the *same sign* and step 2 did not "fail" (i.e., the matrix is symmetric, but not diagonal), then $\phi = -\pi/4$. That is, rotate by $-45°$.

   (b) If the off-diagonal elements have *opposite signs* and step 2 did not "fail," then $\phi = 0$.

   (c) If step 2 "failed," because one of the off-diagonal elements is zero (or near-zero), $\phi = 0$.

The final transforming matrix is $\mathbf{U} = \mathbf{R}_\theta \mathbf{S}_w \mathbf{R}_\phi$ and the RJCF is $\mathbf{J}_\mathbf{B} = \mathbf{B} \overset{\mathbf{U}}{\rightsquigarrow}$. If step 2 "fails," simply $\mathbf{U} = \mathbf{R}_\theta$. Example I.4 in Appendix I shows an example.

The relationship of $\mathbf{U}$ to eigenvectors of $\mathbf{B}$ depends on which case occurred in step 3 above. In case 3a the columns of $\mathbf{U}$ comprise two unique real eigenvectors. As a special subcase, if $\mathbf{B}$ is symmetric, $\mathbf{U}$ is just a rotation and is called a Jacobi rotation [GVL96].

In case 3b the columns of $\mathbf{U}$ are called *principal axes*. They are orthogonal, but in general, they are not equal length, and the rows of $\mathbf{U}$ are not orthogonal. Complex eigenvectors can be constructed by taking one principal axis as the real part and the other as the imaginary part. The name "principal axis" comes from the role of an ellipse having these principal axes in the solution of $d\mathbf{x}/dt = \mathbf{B}\mathbf{x}$ [HS74].

In case 3c $\mathbf{U}$ is a rotation and $\mathbf{B}$ has two equal eigenvalues and only one independent eigenvector. If column j of $\mathbf{J}_\mathbf{B}$ contains the zero, then column j of $\mathbf{U}$ is the eigenvector, and the other column is a *generalized eigenvector* [BD96]. Let λ_1 be the eigenvalue and let $\boldsymbol{\xi}$ be the eigenvector. Then for 2-D, a generalized eigenvector is $\boldsymbol{\eta}$ such that

$$\begin{aligned} \boldsymbol{\eta}^T \boldsymbol{\xi} &= 0, \\ (\mathbf{B} - \lambda_1 \mathbf{I})\boldsymbol{\eta} &= \boldsymbol{\xi}. \end{aligned} \tag{22}$$

Of course, nonzero scalar multiples of $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are eigenvectors and generalized eigenvectors, too. Additional discussion may be found in Appendix C.

2.6 Numerically Unstable Case

The only problematical case occurs when the matrix is asymmetric after the diagonal elements are equalized in step 1 of the geometric procedure in Section 2.5, and one of the off-diagonal elements is "nearly zero," i.e., orders of magnitude smaller than the other off-diagonal element. For example,

$$\mathbf{B} \overset{\mathbf{R}_\theta}{\rightsquigarrow} \equiv \mathbf{M} = \begin{bmatrix} \mathsf{M}_{11} & \pm\varepsilon \\ \mathsf{M}_{21} & \mathsf{M}_{11} \end{bmatrix}$$

The eigenvectors are nearly parallel. An extremely uneven nonuniform scale would be needed in step 2 of the geometric procedure. Depending on whether the sign of ε is the same as that of M_{21} or opposite, the eigenvalues are real or complex. This condition might be determined by rounding accidents of numerical operations. An innocent-looking example with all moderate numbers is matrix (d) in Section 10, whose computed eigenvector matrix is shown in Eq. 69.

If ε is exactly zero, the matrix is already in a canonical form, and no transformation is appropriate. We recommend a practical treatment as follows: When $|\varepsilon| \ll |\mathsf{M}_{21}|$, do not rescale, and treat ε as 0.

A suitable threshold ratio depends on the accuracy of the data. For example, if there are about four significant digits of accuracy, then the criterion might be $|\varepsilon| < 10^{-4}|\mathsf{M}_{21}|$. This cut-off would mean that the scale value w in step 2 of the geometric procedure would never exceed 10 or fall below 1/10 (see Eq. 105).

This case applies to the separated matrix in Eq. 15. Example I.3 in Appendix I discusses numerical issues, but the ideal 2-D RJCF is:

$$\mathbf{B} = \begin{bmatrix} 45 & -\frac{25}{\sqrt{2}} \\ \frac{32}{\sqrt{2}} & 5 \end{bmatrix} = \mathbf{U}\mathbf{J}_\mathbf{B}\mathbf{U}^{-1} \qquad \text{where } \mathbf{U} = \begin{bmatrix} \sqrt{\frac{32}{57}} & \sqrt{\frac{25}{57}} \\ -\sqrt{\frac{25}{57}} & \sqrt{\frac{32}{57}} \end{bmatrix} \text{ and } \mathbf{J}_\mathbf{B} = \begin{bmatrix} 25 & 0 \\ \frac{57}{\sqrt{2}} & 25 \end{bmatrix} \tag{23}$$

Note that $\mathbf{U}$ is the $\mathbf{R}_\theta$ that equalizes the diagonal elements of $\mathbf{B}$.

Now suppose the 0 element in $\mathbf{J}$ above (we drop the subscript $\mathbf{B}$ for this discussion) becomes a very small nonzero value, $\pm\varepsilon$. The real canonical form matrix changes from $\mathbf{J}$ to a diagonal matrix, if the 0 becomes $+\varepsilon$, or a spiral matrix with small off-diagonal elements, if the 0 becomes $-\varepsilon$. The change is discontinuous in both cases.

This concludes the high-level description of RJCF. The appendices provide detailed equations and computational procedures. as well as experimental results concerning accuracy and speed. Subsequent sections illustrate how RJCF can be applied to provide new or improved solutions to various problems in computer animation, modeling, and scientific visualization.

3 Parallel Vectors within Vector Fields

In certain scientific visualization applications, especially those involving fluid flows, the question arises whether vectors in two different vector fields are parallel. For example, several concepts of "vortex core" can be formulated as the set of points where the vectors drawn from two 3-D fields are parallel [PR99]. For two smoothly varying 3-D vector fields, say $\mathbf{v}$ and $\mathbf{w}$, Roth observed that the set of points where $\mathbf{v}\|\mathbf{w}$, i.e., $\mathbf{v}$ is parallel to $\mathbf{w}$, normally forms some number of smooth curves. To date, the only methods published for identifying those curves are heuristic [Rot00, TSW+05]. Little is known about the structure of parallel-vector curves. Moreover, all published methods are computationally very intensive, and have no mathematical guarantees about accuracy or completeness.

In many vector-field datasets, the field is defined at grid vertices in 3-D, and is modeled as an affine function within each tetrahedron that partitions the 3-D space, with grid vertices as its corners. For this model, RJCF analysis yields the complete structure of parallel-vector curves, as well as closed-form formulas for such curves. That is the topic of this section. For this section, we assume that $\mathbf{v}$ and $\mathbf{w}$ are affine vector fields, and the goal is to find the curves along which $\mathbf{v}\|\mathbf{w}$.

We assume that a 3-D coordinate system has been chosen such that the field $\mathbf{w}$ is $\mathbf{0}$ at the origin. A general 3-D point is denoted as $\mathbf{p}$. We use the notation

$$\mathbf{v} = \mathbf{A}\mathbf{p} + \mathbf{v}_0 \tag{24}$$

$$\mathbf{w} = \mathbf{B}\mathbf{p} \tag{25}$$

We further assume that the rank of $\mathbf{B}$ is at least that of $\mathbf{A}$ (otherwise, reverse the roles of $\mathbf{v}$ and $\mathbf{w}$).

We adopt the convention that $\mathbf{0}$ is considered parallel to every vector. Besides $\mathbf{p} = \mathbf{0}$, the vectors are parallel exactly when there is a scalar quantity t such that

$$\mathbf{A}\mathbf{p} + \mathbf{v}_0 = t\,\mathbf{B}\mathbf{p}. \tag{26}$$

If $\mathbf{v}_0 = \mathbf{0}$, this is simply a *generalized eigenvalue problem* [PFTV88, GVL96]. (Roth observed that, when $\mathbf{B}$ is nonsingular and $\mathbf{v}_0 = \mathbf{0}$, premultiplying by $\mathbf{B}^{-1}$ yields a standard eigenvalue problem.) For the remainder of this section, we address the case where $\mathbf{B}$ is nonsingular and $\mathbf{v}_0 \neq \mathbf{0}$. Analytical results are also available when both matrices are singular. For example, if both matrices are rank 2, the problem can be reduced to a 2-D version of the same problem as Eq. 26, and solved by a similar analysis. Since these cases are of limited practical interest, details are omitted.

Under the hypotheses stated above, $\mathbf{0}$ is the unique critical point of field $\mathbf{w}$ (i.e., point where $\mathbf{w} = \mathbf{0}$). Thus we seek solutions in $(\mathbf{p}, t)$ of

$$\mathbf{B}^{-1}\mathbf{A}\,\mathbf{p} = t\,\mathbf{p} - \mathbf{B}^{-1}\mathbf{v}_0. \tag{27}$$

Consider the RJCF of $(\mathbf{B}^{-1}\mathbf{A})$. Let $(\mathbf{B}^{-1}\mathbf{A}) \overset{\mathbf{T}}{\rightsquigarrow} \mathbf{J}$. Define $\mathbf{q}$ to be a general point in the canonical coordinate system: $\mathbf{q} = \mathbf{T}^{-1}\mathbf{p}$. We also abbreviate $\mathbf{T}^{-1}\mathbf{B}^{-1}\mathbf{v}_0 = \mathbf{r}$. Now we seek solutions in $(\mathbf{q}, t)$ of

$$\mathbf{J}\,\mathbf{q} = t\,\mathbf{q} - \mathbf{r} \qquad \text{where } \mathbf{r} = \mathbf{T}^{-1}\mathbf{B}^{-1}\mathbf{v}_0. \tag{28}$$

If we were working in a tetrahedron with vertices $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, we now are concerned with the tetrahedron given by $\mathbf{q}_i = \mathbf{T}^{-1}\mathbf{p}_i$.

For convenience of notation, we assume the outer eigenvalue corresponds to J_{00} and its eigenvector corresponds to $\mathbf{T}_0$; other cases are covered by the appropriate cyclic permutation of indices. As usual, we assume for the main presentation that $(\mathbf{B}^{-1}\mathbf{A})$ does not have three equal eigenvalues; these cases can be handled by minor variations of the main methods. Due to the partitioned nature of $\mathbf{J}$, we immediately obtain the solution for $\mathbf{q}^{(0)}$:

$$\mathbf{q}^{(0)} \;=\; \mathbf{r}^{(0)}/(t-\mathsf{J}_{00}) \tag{29}$$

Thus all solutions lie in the plane $\mathbf{q}^{(0)} = 0$ if $\mathbf{r}^{(0)} = 0$; this special case can be treated by variations of the methods for the general case, and we assume henceforth that $\mathbf{r}^{(0)} \neq 0$. Then, Eq. 29 gives a 1-1 mapping that allows t to be eliminated from the full solutions:

$$t \;=\; \mathsf{J}_{00} + \mathbf{r}^{(0)}/\mathbf{q}^{(0)}. \tag{30}$$

We also note that as $\mathbf{q}^{(0)}$ passes through 0, t "wraps around" from $+\infty$ to $-\infty$, or vice versa.

3.1 Diagonal RJCF

We now complete the analysis for the case that $\mathbf{J}$ has the diagonal form. To be specific, assume $\mathsf{J}_{00} < \mathsf{J}_{11} < \mathsf{J}_{22}$. We have:

$$\mathbf{q}^{(j)} \;=\; \mathbf{r}^{(j)}/(t-\mathsf{J}_{jj}) \qquad\qquad \text{for } j = 0,1,2. \tag{31}$$

If any $\mathbf{r}^{(j)} = 0$, then all solutions lie in the plane $\mathbf{q}^{(j)} = 0$. Suppose all components of $\mathbf{r}$ are nonzero. Then there are three distinct nonintersecting curves parameterized by t for these ranges:

Curve 1: $\mathsf{J}_{00} < t < \mathsf{J}_{11}$; this curve recedes to $\pm\infty$ in the directions of $\mathbf{q}^{(0)}$ and $\mathbf{q}^{(1)}$;

Curve 2: $\mathsf{J}_{11} < t < \mathsf{J}_{22}$; this curve recedes to $\pm\infty$ in the directions of $\mathbf{q}^{(1)}$ and $\mathbf{q}^{(2)}$;

Curve 3: $\mathsf{J}_{22} < t$ or $t < \mathsf{J}_{00}$, together with the limit point $\mathbf{0}$; this curve recedes to $\pm\infty$ in the directions of $\mathbf{q}^{(2)}$ and $\mathbf{q}^{(0)}$.

If $\mathsf{J}_{11} > \mathsf{J}_{22}$, just interchange their roles in the range definitions. If J_{00} is the maximum eigenvalue, rather than the minimum, just reverse all the inequalities in the range definitions. Various special cases are minor variations of the above structure, as long as $\mathbf{J}$ is a diagonal matrix:

1. If $\mathsf{J}_{11} = \mathsf{J}_{22}$, then Curve 2 does not exist.
2. If all three eigenvalues are equal, then Curve 1 and Curve 2 do not exist.
3. If exactly one component of $\mathbf{r}$ is zero, two curves merge. For example, if $\mathbf{r}^{(1)} = 0$ and the other components are nonzero, then Curve 1 and Curve 2 merge (assuming Curve 2 exists, else Curve 1 and Curve 3 merge). The merged curve does not recede to $\pm\infty$ in the direction of $\mathbf{q}^{(1)}$; instead it remains in the plane $\mathbf{q}^{(1)} = 0$ and includes the limit point $[\mathbf{r}^{(0)}/(\mathsf{J}_{11}-\mathsf{J}_{00}), 0, \mathbf{r}^{(2)}/(\mathsf{J}_{11}-\mathsf{J}_{22})]^T$.
4. If two components of $\mathbf{r}$ are zero, the three curves merge and lie on the third axis.

Thus the same basic structure is present in all cases that $\mathbf{J}$ is a diagonal matrix.

Now we consider finding the intersection of the above curves with the tetrahedron whose vertices are $\mathbf{q}_i$, $i = 0$, 1, 2, 3. Suppose the plane equation for the face $\mathbf{q}_0$, $\mathbf{q}_1$, $\mathbf{q}_2$ is $\mathbf{n}^T\mathbf{q} + d = 0$, with $\mathbf{n}$ oriented so that $\mathbf{n}^T\mathbf{q}_3 > 0$. Then solution points from Eq. 31 are in the same half-space as $\mathbf{q}_3$ when

$$\frac{\mathbf{n}^{(0)}\,\mathbf{r}^{(0)}}{(t-\mathsf{J}_{00})} + \frac{\mathbf{n}^{(1)}\,\mathbf{r}^{(1)}}{(t-\mathsf{J}_{11})} + \frac{\mathbf{n}^{(2)}\,\mathbf{r}^{(2)}}{(t-\mathsf{J}_{22})} + d \;\geq\; 0 \tag{32}$$

The intervals of t that satisfy the inequality have the roots of a cubic equation as their end-points (see Appendix A for solution methods). Intersect the intervals given for each face of the tetrahedron to find the intervals of t that define points where $\mathbf{u}||\mathbf{v}$ within the tetrahedron.

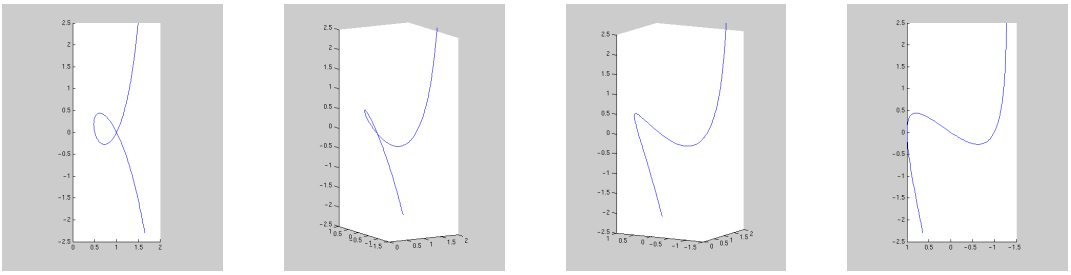

Figure 5: Four views of the parallel-vector curve of the linear vector fields given in Eq. 35.

3.2 Spiral RJCF

Next, we complete the analysis for the case that **J** has the *spiral* form. In this form the possibly nonzero elements are J_{00}, J_{11}, $\mathsf{J}_{12} \neq 0$, $\mathsf{J}_{21} = -\mathsf{J}_{12}$, and $\mathsf{J}_{22} = \mathsf{J}_{11}$. To be specific, assume $\mathsf{J}_{00} < \mathsf{J}_{11}$. To solve Eq. 28, we observe that:

$$(\mathbf{J} - \mathbf{I}t)^{-1} = \begin{bmatrix} 1/(\mathsf{J}_{00} - t) & 0 & 0 \\ 0 & (\mathsf{J}_{11} - t)/((\mathsf{J}_{11} - t)^2 + \mathsf{J}_{12}^2) & -\mathsf{J}_{12}/((\mathsf{J}_{11} - t)^2 + \mathsf{J}_{12}^2) \\ 0 & \mathsf{J}_{12}/((\mathsf{J}_{11} - t)^2 + \mathsf{J}_{12}^2) & (\mathsf{J}_{11} - t)/((\mathsf{J}_{11} - t)^2 + \mathsf{J}_{12}^2) \end{bmatrix} \tag{33}$$

This leads to the solution:

$$\begin{aligned} \mathbf{q}^{(0)} &= \mathbf{r}^{(0)}/(t - \mathsf{J}_{00}) \\ \mathbf{q}^{(1)} &= \frac{\mathbf{r}^{(1)}(t - \mathsf{J}_{11}) + \mathbf{r}^{(2)}\mathsf{J}_{12}}{(t - \mathsf{J}_{11})^2 + \mathsf{J}_{12}^2} \\ \mathbf{q}^{(2)} &= \frac{\mathbf{r}^{(2)}(t - \mathsf{J}_{11}) - \mathbf{r}^{(1)}\mathsf{J}_{12}}{(t - \mathsf{J}_{11})^2 + \mathsf{J}_{12}^2} \end{aligned} \tag{34}$$

This is one curve whose projection onto the $\mathbf{q}^{(0)}$-plane is a circle that is centered at $[0, r^{(2)}/2\mathsf{J}_{12}, -\mathbf{r}^{(1)}/2\mathsf{J}_{12}]^T$ and passes through the origin. If $\mathbf{r}^{(0)} = 0$, this circle is the actual solution. If $\mathbf{r}^{(0)} \neq 0$, the solution curve recedes to ∞ in the $\pm\mathbf{q}^{(0)}$ directions and completes one revolution around the circle in $(\mathbf{q}^{(1)}, \mathbf{q}^{(2)})$ as $\mathbf{q}^{(0)}$ ranges from $-\infty$ to ∞.

To find the intersection of the solution curve with a tetrahedron the technique is the same as for the case when **J** is diagonal: substitute Eq. 34 into the plane equation of each face, yielding a cubic equation to solve for the values of t at which the solution curve intersects the face (see Appendix A for solution methods).

In the canonical coordinate system, the solution curve is monotonic in $\mathbf{q}^{(0)}$ (i.e., it can be solved as $\mathbf{q}^{(1)}$ and $\mathbf{q}^{(2)}$ being rational functions of $\mathbf{q}^{(0)}$). However, the curve may well *not* be monotonic in any original spatial coordinate. An example of this phenomenon is shown in Figure 5. This figure shows four views of the parallel-vector curve for the pair of fields:

$$\mathbf{v} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} -2 \\ 0 \\ 0 \end{bmatrix} \tag{35}$$

The reason is that the transforming matrix **T** in Eq. 28 may have a shear component, as it does in this case. This shows the advantage of performing analysis in the canonical coordinate system, where the underlying structure is usually clearer.

3.3 Shear RJCF

The analysis for the simple shear and compound shear forms of **J** use the same ideas as the above cases. The inverse of $(\mathbf{J} - \mathbf{I}t)$ can be written explicitly as rational functions of t with denominators $(t - \mathsf{J}_{00})$ and $(t - \mathsf{J}_{11})^2$. In the normal

case there are two distinct curves. Substitution of the solution into the plane equation for each face of the tetrahedron produces a cubic equation whose roots specify where the curves intersect that face. The details are omitted.

4 Matrix Exponentials and Logarithms

Matrix exponentials and matrix logarithms are finding increasing applications in computer graphics and related fields [MLS94, KKS95, PR97, Gra98, BM98, Ale02], so it is important to have efficient and reliable procedures for their calculation. Alexa uses the matrix logarithm in interpolation [Ale02] and provides complicated procedures to compute it. This section derives closed forms. Once the matrix logarithm is found, the n-th root of the original matrix can be obtained simply by dividing the log matrix by n and exponentiating the result. This section shows that RJCF is the most advantageous way to do this. Again, the compound shear cases that are less likely to be useful for computer graphics are deferred to Appendix C.2. All quantities are real.

For any square matrix $\mathbf{B}$, the matrix exponential of $\mathbf{B}$ is defined by:

$$\mathbf{A} \;=\; e^{\mathbf{B}} \;=\; \sum_{k=0}^{\infty} \mathbf{B}^k/k! \;=\; \mathbf{I}+\mathbf{B}+\mathbf{B}^2/2!+\mathbf{B}^3/3!+\ldots \tag{36}$$

and any matrix $\mathbf{B}$ that satisfies Eq. 36 is a logarithm of $\mathbf{A}$.

Caution must be used with this expression because $e^{\mathbf{B}}e^{\mathbf{C}}$ does not equal $e^{\mathbf{B}+\mathbf{C}}$ unless the matrices $\mathbf{B}$ and $\mathbf{C}$ commute for multiplication. Also the matrix exponential operator is not 1-1, as the real scalar operator is. Possibly, multiple real-valued $\mathbf{B}$'s satisfy Eq. 36 and possibly no real-valued $\mathbf{B}$ satisfies the equation.

Some well known properties of the matrix exponential are [HS74]:

$$\exp(\mathbf{B}\overset{\mathbf{T}}{\rightsquigarrow}) \;=\; \left(e^{\mathbf{B}}\right)\overset{\mathbf{T}}{\rightsquigarrow} \tag{37}$$

$$\exp\left(\begin{bmatrix} \mathbf{C} & 0 \\ 0 & \mathbf{D} \end{bmatrix}\right) \;=\; \begin{bmatrix} e^{\mathbf{C}} & 0 \\ 0 & e^{\mathbf{D}} \end{bmatrix} \qquad \text{for } \mathbf{C} \text{ and } \mathbf{D} \text{ square matrices.} \tag{38}$$

For computations involving Eq. 36, where $\mathbf{B}$ is a matrix logarithm of $\mathbf{A}$, the key observation is that the eigenvectors of $\mathbf{B}$ are also those for $\mathbf{A}$; i.e., they are preserved by the exponential operator. (In the simple shear case, the generalized eigenvector is also preserved; both properties are proved in Appendix D.) Therefore, by Eq. 37, it suffices to consider $\mathbf{A}$ and $\mathbf{B}$ in RJCF when considering how to compute the matrix logarithm or exponential. We will see in the next few paragraphs that if $\mathbf{B}$ is in RJCF, then so is $\mathbf{A}$. The problem is further simplified by Eq. 38: the problem separates into a 1-D problem and a 2-D problem.

We now describe the relationships between matrices and their logs in RJCF. Proofs are given in Appendix D. In the following analysis $\mathbf{A}$ and $\mathbf{B}$ might refer to 1×1 or 2×2 matrices, due to this separation.

Diagonal Case:

If $\mathbf{B}$ is a diagonal matrix (which includes the 1-D case), then $\mathbf{A}$ is a positive diagonal matrix. Working back, if $\mathbf{A}$ is a positive diagonal matrix (a scale matrix), then $\log \mathbf{A} = \mathbf{B}$, where $\mathbf{B}$ is diagonal and

$$\mathsf{B}_{jj} \;=\; \log(\mathsf{A}_{jj}). \tag{39}$$

Spiral Case:

Next we consider spiral and rotation matrices. Let a and b be arbitrary scalars (usually b is nonzero).

$$\text{If} \quad \mathbf{B} \;=\; \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \quad \text{then} \quad \mathbf{A} \;=\; e^{\mathbf{B}} \;=\; e^{a}\begin{bmatrix} \cos b & -\sin b \\ \sin b & \cos b \end{bmatrix}. \tag{40}$$

This equation is well known in differential equation theory [HS74]. If $a = 0$ it yields a rotation matrix. It is the only case for which we can tolerate negative entries on the diagonal of $\mathbf{A}$. Working back for spiral and rotation matrices, Let λ and μ be arbitrary, except that $\lambda^2+\mu^2 > 0$.

$$\text{If } \mathbf{A} = \begin{bmatrix} \lambda & -\mu \\ \mu & \lambda \end{bmatrix} \quad \text{then } \mathbf{B} = \log \mathbf{A} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \qquad \text{where } a = \tfrac{1}{2}\log(\lambda^2+\mu^2),\ b = \texttt{atan2}(\mu,\lambda). \tag{41}$$

Here atan2 is the two-parameter arc-tangent function, which satisfies $\tan b = \mu/\lambda$. Any integer multiple of 2π may be added to b to produce an alternate solution.

It is important to note that Eq. 41 includes the case in which **A** has two equal negative real eigenvalues, so such matrices have real matrix logarithms despite having negative eigenvalues.

Simple Shear Case:

Turning to simple shear RJCFs, let a and b be arbitrary (usually b is nonzero).

$$\text{If} \quad \mathbf{B} = \begin{bmatrix} a & b \\ 0 & a \end{bmatrix} \quad \text{then} \quad \mathbf{A} = e^{\mathbf{B}} = e^a \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix}. \tag{42}$$

Working back for simple shear matrices, Let $\lambda > 0$ and let μ be arbitrary.

$$\text{If} \quad \mathbf{A} = \begin{bmatrix} \lambda & \mu \\ 0 & \lambda \end{bmatrix} \quad \text{then} \quad \mathbf{B} = \log \mathbf{A} = \begin{bmatrix} \log\lambda & \mu/\lambda \\ 0 & \log\lambda \end{bmatrix} \tag{43}$$

Of course, if μ or b is below the diagonal, simply use transposes.

4.1 Fractional Powers of Matrices

We can apply the equations for matrix logarithms to the problem of computing $\mathbf{A}^\alpha$ where α is fractional. First find the RJCF $\mathbf{J} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}$ as in Section 2. Then find $\log(\mathbf{J})$ by one of the above formulas. Multiply this by α. Finally, re-exponentiate by one of the above formulas, Eq. 39–43.

$$\mathbf{A}^\alpha = \mathbf{T} e^{\alpha \log \mathbf{J}} \mathbf{T}^{-1} \tag{44}$$

Observe that α is *not* applied to **T**.

Examples I.5 and I.6 show the calculations to put **A** into RJCF, $\mathbf{A} \overset{\mathbf{T}}{\rightsquigarrow} \mathbf{J}$, and then find $\log \mathbf{J}$, where:

$$\mathbf{A} = \begin{bmatrix} 1.60 & -1.20 & 1.60 \\ 0.54 & 1.12 & -0.16 \\ -0.32 & -0.96 & 2.28 \end{bmatrix} \qquad \mathbf{J} = \mathbf{A} \overset{\mathbf{T}}{\rightsquigarrow} = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix} \qquad \log(\mathbf{J}) = \begin{bmatrix} 0.8047 & 0 & 0.4636 \\ 0 & 0 & 0 \\ -0.4636 & 0 & 0.8047 \end{bmatrix}$$

Note that $\alpha \log(\mathbf{J})$ has the same separated form as **J** for any scalar α. Using Eq. 44 and 40, and the value of **T** found in Example I.5, the examples show that

$$\mathbf{A}^{\frac{1}{2}} = \mathbf{T} \begin{bmatrix} 5^{1/4}\cos(.2318) & 0 & 5^{1/4}\sin(.2318) \\ 0 & 1 & 0 \\ -5^{1/4}\sin(.2318) & 0 & 5^{1/4}\cos(.2318) \end{bmatrix} \mathbf{T}^{-1} = \begin{bmatrix} 1.3179 & -0.4123 & 0.5497 \\ 0.2213 & 1.0815 & -0.1086 \\ -0.0831 & -0.3835 & 1.5113 \end{bmatrix}$$

4.2 Logarithms of Matrices with Deficient Rank

If **A** has only rank 2, the **J** has a zero row and a zero column. Although **J** does not have a finite logarithm, its remaining 2x2 submatrix may well have: the other eigenvalues need to be positive or a complex-conjugate pair. (Equal negative eigenvalues can be treated as complex with zero imaginary parts.) The zero eigenvalue has a perfectly good eigenvector, so the matrix **T** is well defined, and a typical computer graphics application using the logarithm (e.g., to find $\mathbf{A}^\alpha$ for fractional α) will get sensible results. An example is discussed in Section 5.1.

4.3 Summary for Matrix Logarithms

To summarize, real matrix logarithms, when they exist, can be computed by transforming to RJCF. The matrix logarithm of the canonical form is then transformed back to the original coordinate system to provide the logarithm of the original matrix, or kept in the factored form for further operations. There are multiple real solutions when the RJCF has the spiral form.

The existence of real matrix logarithms in *n*-D is completely characterized by Hirsch and Smale in problems 9 and 10 of Section 6.4, in terms of Jordan blocks of negative eigenvalues [HS74, p. 132–3]. In 2-D, the eigenvalues must be positive or a complex conjugate pair. As a limiting case, two equal negative eigenvalues when the RJCF is the *diagonal* form can be treated as a complex conjugate pair with imaginary parts zero. However, if the two equal negative eigenvalues occur when the RJCF has a *simple shear* form, the RJCF comprises a single 2×2 Jordan block, and there is no real matrix logarithm.

Alexa also identified these cases as having real matrix logarithms [Ale02], except that his statements about two equal negative eigenvalues were imprecise; he did not distinguish between these two cases:

$$\begin{bmatrix} -2 & 0 \\ 1 & -2 \end{bmatrix} \qquad \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix}$$

The left matrix has no real logarithm, although it has two eigenvalues of –2. Also Alexa did not supply closed-form expressions, and provided a procedure with nested iterations and unclear convergence behavior. RJCF considerations provide a direct test for existence and supply closed-form expressions for all the cases that do exist (see Appendix D).

5 Interpolation with Matrix Lie Product

We have already remarked that $e^{\log\mathbf{B}+\log\mathbf{C}}$ does not equal $\mathbf{BC}$, in general. Alexa overcame this problem by boldly defining it to be a new operation: $\mathbf{B} \oplus \mathbf{C} = \mathbf{C} \oplus \mathbf{B} = e^{\log\mathbf{B}+\log\mathbf{C}}$ [Ale02]. Together with the "scalar multiplication" operation $\alpha \odot \mathbf{B} = \mathbf{B}^\alpha = e^{\alpha \log\mathbf{B}}$, this provides a system for interpolating between linear transformations.

$$L(\mathbf{B},\mathbf{C},\alpha) \quad = \quad \exp((1-\alpha)\log\mathbf{B} + \alpha\log\mathbf{C}) \tag{45}$$

As usual, the issue that the log may not exist or may not be unique has to be addressed, as was discussed in Section 4.3.

We have already seen that the exponentiation can be carried out in closed form by finding the RJCF for the right-hand side of Eq. 45, then re-exponentiating according to Section 4. In general, the RJCF of the blend using α is not related to the RJCFs for the individual matrices, because $\mathbf{B}$ and $\mathbf{C}$ do not have the same eigenvectors. In other contexts the closed form has proven faster and more accurate than iterative techniques, such as that proposed by Alexa. Closed forms have the advantages that there is no decision to make on when to stop iterating, and the accuracy can be analyzed using well-developed techniques of numerical analysis [GVL96]. The same observations apply to the polar decomposition discussed in Section 6.

5.1 Matrix Lie Product Interpolation with Singular Matrices

Although Alexa seems to imply that the "$\oplus$" operator works when a matrix has a determinant of 0, this might just be imprecise wording in one part of his paper [Ale02], because in another part he requires positive determinant and the procedure he gives uses matrix inverse, so it cannot handle such matrices. We now show how to extend the definition of this operator so that it can blend a matrix $\mathbf{B}$ of rank 2 with a matrix $\mathbf{C}$ of rank 3. Perhaps the main interest in this topic is the light it sheds on how this style of interpolation behaves on *ill-conditioned* matrices, i.e., those for which the eigenvalue ratio $\lambda_{min}/\lambda_{max}$ is much less than one.

Assume the RJCF for $\mathbf{B}$ is $\mathbf{B} = \mathbf{VJV}^{-1}$ with row 0 and column 0 of $\mathbf{J}$ being zeros. Define $\mathbf{D} = \mathbf{V}^{-1}\mathbf{CV}$. Then we have

$$\begin{aligned} \mathbf{L} \quad &= \quad \alpha\log\mathbf{J} + (1-\alpha)\log\mathbf{D} \\ \alpha\log\mathbf{B} + (1-\alpha)\log\mathbf{C} \quad &= \quad \mathbf{VLV}^{-1} \end{aligned}$$

Because $(\log\mathbf{J})_{00} = -\infty$, it dominates row 0 and column 0 of $\mathbf{L}$ for any $\alpha > 0$. It is shown in Appendix D that we get correct results by simply setting the off-diagonal elements of row 0 and column 0 of $\mathbf{L}$ to zero. Therefore we work with the four lower right elements of $\mathbf{L}$ and find a further transformation $\mathbf{U}$ to put them into RJCF ($\mathsf{U}_{00} = 1$). This produces $\mathbf{H}$ in RJCF, with $\mathsf{M}_{00} = -\infty$, such that: $\mathbf{VLV}^{-1} = \mathbf{VUHU}^{-1}\mathbf{V}^{-1}$. This matrix needs to be exponentiated, to give the final blend, and that is accomplished by exponentiating $\mathbf{H}$. But $e^{\mathbf{H}}$ has all zeros in its 0-th row and 0-th column. This gives the result that any blend with $\alpha > 0$ has rank 2, but when $\alpha = 0$, we have $\mathbf{C}$, which has rank 3.

Carrying out the steps just described, Example I.7 shows that matrices **B** and **C** blend into $L(\mathbf{B},\mathbf{C},0.9) = 0.1 \odot \mathbf{B} \oplus 0.9 \odot \mathbf{C}$, as follows:

$$\mathbf{B} = \begin{bmatrix} 1.60 & -1.20 & 1.60 \\ 0.86 & 0.48 & -0.64 \\ -0.08 & -1.44 & 1.92 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad L(\mathbf{B},\mathbf{C},0.9) = \begin{bmatrix} 1.0062 & -0.3550 & 0.4734 \\ 0.6729 & 0.2749 & -0.3665 \\ -0.2264 & -0.6032 & 0.8042 \end{bmatrix}.$$

B has rank 2 and its only real eigenvalue is 0. **C** has rank 3 and its only real eigenvalue is 1. It is noteworthy that although $L(\mathbf{B},\mathbf{C},0.9)$ supposedly has only "0.1 of **B**", its only real eigenvalue is 0. We consider how polar decomposition treats these matrices in Section 8.

This proposed extension might seem unsatisfactory, because the animator probably wants a sequence that flattens (or unflattens) progressively, not instantly. But it brings to light a fact about the *matrix Lie product* method of blending two matrices where one is ill-conditioned. The same analysis carries through with $-\infty$ replaced by a very large positive or negative number. The determinant of the result will be extremely large or small through most of the range of α.

6 Polar Decomposition and Singular-Value Decomposition

Shoemake and Duff proposed *polar decomposition* for a framework to interpolate between transformations [SD92]. They argue that it is more suitable than *singular-value decomposition* for computer animation. Part of their argument is that SVD is quite complicated to compute and not unique (see also *Numerical Recipes in C* [PFTV88], which considers SVD too complicated to explain in the book). Apparently the only published procedures to compute either decomposition are iterative [Hig86, HS90, HP94, ZZ95, GVL96], and some require the matrix to have full rank; however, they are not limited or specialized to the 3×3 case. SVD and polar decomposition are defined even for nonsquare matrices.

We present a new theorem about the structure of polar decomposition and describe a new computational procedure, based on RJCF. To keep the presentation focussed on the main ideas, this section considers only square matrices. Extensions to nonsquare matrices are straightforward (see Appendix H.2). Unlike most topics in this paper, $n \times n$ matrices are also considered.

Definition 6.1: The *polar decomposition* of an $n \times n$ matrix **M** is defined to be two matrices **Q** and **S** such that

1. **Q** is orthonormal; i.e., $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, and $\det(\mathbf{Q}) = \pm 1$; ($\det(\mathbf{Q})$ the same sign as $\det(\mathbf{M})$ when the latter is nonzero);
2. **S** is symmetric and positive definite or positive semidefinite;
3. $\mathbf{M} = \mathbf{QS}$.

□

Definition 6.2: The *singular-value decomposition* (SVD) of a $n \times n$ matrix **M** is defined to be three matrices **U**, $\mathbf{\Lambda}$, **V**, such that

1. **U** and **V** are orthonormal;
2. $\mathbf{\Lambda}$ is diagonal, and positive definite or positive semidefinite;
3. $\mathbf{M} = \mathbf{U\Lambda V}^T$.

The diagonal elements of $\mathbf{\Lambda}$ are called the *singular values* of **M**. □

It is known that the polar decomposition is unique for nonsingular matrices, whereas the singular-value decomposition is not. The polar decomposition is not unique for singular matrices. However, RJCF enables us to prove the following theorem, which we have not seen stated in the literature.

Theorem 6.1: An $n \times n$ matrix with a 0 eigenvalue of multiplicity 1 has two polar decompositions, one whose orthonormal component has determinant $+1$ and one whose orthonormal component has determinant -1 .

Proof: See Appendix H.1.

This section gives closed form solutions for the 3×3 cases, and provides a procedure that works for all ranks. Further, the procedure can be used for $n \times n$, for $n > 3$ by using the Jacobi method to get eigenvalues and eigenvectors of a real symmetric matrix [PFTV88, GVL96].

Shoemake and Duff give an iterative procedure to compute $\mathbf{Q}$ in Definition 6.1, attributed to Higham, but since it requires $\mathbf{M}^{-1}$, it only applies when $\mathbf{M}$ is nonsingular (also square), hence $\mathbf{S}$ is positive definite [SD92, Hig86]. We now describe a method based on RJCF that is closed form and works in the semidefinite case, as well.

If $\det(\mathbf{M}) < 0$, our method succeeds, but the orthonormal component has $\det(\mathbf{Q}) < 0$. Such a $\mathbf{Q}$ is the composition of a rotation and a reflection through some arbitrary plane (hyperplane for $n > 3$), and its interpretation for purposes of computer graphics or mechanics is unclear.

Shoemake and Duff suggest handling the case of $\det(\mathbf{M}) < 0$ essentially by finding the polar decomposition of $-\mathbf{M}$ instead. Readers should be aware that the rotation found this way has no straightforward relationship to the orthonormal component (with negative determinant) of the actual polar decomposition. See Appendix H for more details.

Since $\mathbf{M}^T\mathbf{M} = \mathbf{S}^2$ it suffices to find the RJCF of $\mathbf{A} = \mathbf{M}^T\mathbf{M}$, which is positive semidefinite by construction. It is known that the RJCF takes the form $\mathbf{R}\mathbf{J}\mathbf{R}^T$, where $\mathbf{R}$ is a rotation matrix and $\mathbf{J}$ is diagonal with nonnegative elements.

If $\mathbf{M}$ is $n \times n$ for $n > 3$, the Jordan canonical form of $\mathbf{A}$ has the same properties: a diagonal $\mathbf{J}$ with nonnegative elements and orthogonal matrix $\mathbf{R}$ with $\det(\mathbf{R}) = 1$ whose columns are eigenvectors. The Jacobi method is accurate and robust for computing $\mathbf{J}$ and $\mathbf{R}$, but inefficiency may become a problem for $n > 10$ [PFTV88].

Let $\mathbf{\Lambda} = \sqrt{\mathbf{J}}$ (this is the positive square root). Then $\mathbf{S} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^T$, in theory, and if $\mathbf{S}$ is nonsingular, $\mathbf{Q} = \mathbf{M}\mathbf{S}^{-1}$ in theory. (Note that $\mathbf{M}$, $\mathbf{A}$, and $\mathbf{S}$ all have the same rank.) But Eq. 50 shows a better numerical alternative for $\mathbf{S}$ and the following paragraphs describe a better procedure to compute $\mathbf{Q}$, a procedure that also applies when $\mathbf{S}$ is singular.

It is possible to complete the decomposition without inverting any matrix, exploiting the fact that $\mathbf{R}$ always has full rank. This allows us to obtain results when $\mathbf{S}$ has rank 2 or 1, and should be more stable than using $\mathbf{S}^{-1}$ if $\mathbf{\Lambda}$ is *ill-conditioned*, i.e., the ratio $\lambda_{min}/\lambda_{max}$ is much less than 1.0.

We need to find $\mathbf{Q}$ (known to be orthonormal) such that $\mathbf{M} = \mathbf{Q}\mathbf{R}\mathbf{\Lambda}\mathbf{R}^T$. To this end we define intermediate matrices to be computed:

$$\mathbf{C} = \mathbf{Q}\mathbf{R} \tag{46}$$

$$\mathbf{G} = \mathbf{M}\mathbf{R} = \mathbf{C}\mathbf{\Lambda}. \tag{47}$$

Note that $\mathbf{C}$ is orthonormal and the columns of $\mathbf{G}$ must be orthogonal, but not necessarily orthonormal, because

$$\mathbf{R}^T\mathbf{M}^T\mathbf{M}\mathbf{R} = \mathbf{\Lambda}^2. \tag{48}$$

Using Eq. 47, the first expression for $\mathbf{G}$ consists of known values, while the second expression for $\mathbf{G}$ implies that

$$\mathbf{C}_j = \mathbf{G}_j/\lambda_j, \tag{49}$$

provided λ_j/λ_{max} is not dangerously small. (Recall that the single subscript denotes the column of the matrix.)

For best numerical accuracy, the conclusion of the computation differs between $n = 3$ and $n > 3$. We consider $n = 3$ first. If $\mathbf{\Lambda}$ is well conditioned, $\mathbf{C}$ as computed by Eq. 49 and $\mathbf{Q} = \mathbf{C}\mathbf{R}^T$, for the orthonormal part of the decomposition.

Still considering $n = 3$, suppose that $\mathbf{\Lambda}$ is ill conditioned and assume $\Lambda_{22} = 0$ (or is dangerously small). Other cases are handled by cyclically permuting the indices. We already know that $\mathbf{C}_0$ and $\mathbf{C}_1$ are orthonormal by Eq. 48. Since $\mathbf{C}$ needs to be orthonormal, set $\mathbf{C}_2 = \mathbf{C}_0 \times \mathbf{C}_1$. This also ensures that $\det(\mathbf{C}) = +1$. As in the main case, $\mathbf{Q} = \mathbf{C}\mathbf{R}^T$. By Theorem 6.1, this is the unique polar decomposition for which $\det(\mathbf{Q}) = +1$. If $\mathbf{M}$ has rank one, see Appendix H.3 for this special case.

Now we consider the more general case, $n > 3$, but restrict consideration to the case in which at most one eigenvalue is zero (or dangerously small). If $\mathbf{\Lambda}$ is well conditioned, it is tempting to accept $\mathbf{C}$ as computed by Eq. 49, but this produces mediocre accuracy in our experience. Assume the eigenvectors and eigenvalues, as returned by the Jacobi subroutine, are permuted so that the eigenvalues appear in increasing order; as a result, we do not assume that $\mathbf{R}$ has positive determinant. (This is the case for the `eig` function in *matlab*.) Thus $\mathbf{C}_0$ is the only problematical column in Eq. 49. Although Eq. 48 implies that $\mathbf{C}_1, \ldots, \mathbf{C}_{n-1}$ are orthonormal in theory, as a precaution, the Gram-Schmidt procedure (see Appendix F) can be used to ensure that this is the case.

There are several ways to find $\mathbf{C}_0$ that is orthogonal to $\mathbf{C}_1, \ldots, \mathbf{C}_{n-1}$. A straightforward procedure is:

1. Initialize $\mathbf{C}_0 = \mathbf{0}$;
2. Choose a "reasonable" row i of $\mathbf{C}$ with magnitude well under 1.0, e.g., the minimum row;
3. Initialize $\mathbf{C}_0 = \mathbf{e}_i$;
4. If $\det(\mathbf{C})$ has the opposite sign from $\det(\mathbf{R})$, reinitialize $\mathbf{C}_0 = -\mathbf{e}_i$;
5. Use the Gram-Schmidt procedure (see Appendix F) to make $\mathbf{C}_0$ orthonormal to $\mathbf{C}_1$, ..., $\mathbf{C}_{n-1}$. (This does not change the sign of the determinant.)

The final result does not depend on which "reasonable" $\mathbf{e}_i$ is the starting value. By Theorem 6.1, the resulting $\mathbf{Q} = \mathbf{C}\mathbf{R}^T$ is the unique polar decomposition with $\det(\mathbf{Q}) = +1$.

The procedure can be modified to produce $\det(\mathbf{Q}) = -1$ when $\det(\mathbf{M}) \leq 0$. Simply choose $\mathbf{e}_i$ or $-\mathbf{e}_i$ in steps (3) and (4) so that $\det(\mathbf{C})$ has the *opposite* sign from $\det(\mathbf{R})$. See Appendix H.1 for more details.

The procedure can be generalized further to accommodate $d > 1$ eigenvalues that are zero (or dangerously small). Initialize d leftmost columns of $\mathbf{C}$ to $\mathbf{0}$, corresponding to the columns of 0 eigenvalues. Then, one by one, make them orthonormal to the columns corresponding to nonzero eigenvalues, as just described, (step 4 is skipped until the last column). There are infinite families of solutions and infinitely many eigenvectors of 0. Users will need to analyze their applications to decide which polar decomposition, if any, is appropriate.

Once $\mathbf{Q}$ is computed, for both $n = 3$ and $n > 3$ our computational experience indicates that $\mathbf{S}$ can be computed most accurately by

$$\mathbf{S} = ((\mathbf{Q}^{-1}\mathbf{M}) + (\mathbf{Q}^{-1}\mathbf{M})^T)/2. \tag{50}$$

Higham recommends the same formula, except using $\mathbf{Q}^T$ in place of $\mathbf{Q}^{-1}$ [HP94], and in theory they are equal. We obtain slightly more accuracy with $\mathbf{Q}^{-1}$.

6.1 From Polar to SVD

The singular-value decomposition becomes a corollary. Just set $\mathbf{V} = \mathbf{R}$ and $\mathbf{U} = \mathbf{C}$ and $\mathbf{\Lambda}$ is the same as for the polar decomposition. SVD uses three matrices and this gives extra flexibility, compared to polar decomposition. Consequently, this solution is not unique, even when $\mathbf{M}$ is nonsingular [PFTV88].

6.2 Using Polar Decomposition for Interpolation

As for application of the polar decomposition, the RJCF form of $\mathbf{S}$ (i.e., $\mathbf{R}\mathbf{\Lambda}\mathbf{R}^T$) is likely to be useful. For two matrices $\mathbf{B}$ and $\mathbf{C}$ Shoemake and Duff proposed an interpolation scheme $L_{PSD}(\mathbf{B}, \mathbf{C}, \alpha)$ based on polar decompositions as follows (the subscript *PSD* stands for "Polar, Shoemake and Duff"):

$$\begin{aligned} \mathbf{B} &= \mathbf{Q}_\mathbf{B}\mathbf{S}_\mathbf{B} \\ \mathbf{C} &= \mathbf{Q}_\mathbf{C}\mathbf{S}_\mathbf{C} \\ L_{PSD}(\mathbf{B}, \mathbf{C}, \alpha) &= \texttt{slerp}(\mathbf{Q}_\mathbf{B}, \mathbf{Q}_\mathbf{C}, \alpha)\,((1-\alpha) \qquad (51) \\ & \quad S_\mathbf{B} + \alpha\mathbf{S}_\mathbf{C}) \qquad (52) \end{aligned}$$

where `slerp` interpolates rotations in quaternion space [Sho85] (see also Appendix E).

The RJCF provides an interesting alternative: In the spirit of Alexa (see Eq. 45 in Section 5), as well as Shoemake and Duff, we propose to use

$$L_{mix}(\mathbf{B}, \mathbf{C}, \alpha) = \texttt{slerp}(\mathbf{Q}_\mathbf{B}, \mathbf{Q}_\mathbf{C}, \alpha)\, L(\mathbf{S}_\mathbf{B}, \mathbf{S}_\mathbf{C}, \alpha) \tag{53}$$

This scheme is invariant under rotations of the coordinate system. When $\mathbf{S}_\mathbf{B}$ and $\mathbf{S}_\mathbf{C}$ both have full rank, the blend is continuous in α, but when $\mathbf{S}_\mathbf{B}$ is only positive semi-definite, there is a discontinuity at $\alpha = 1$, as with the pure matrix Lie blend. Shoemake and Duff's proposal is continuous in this case. We offer L_{mix} as an alternative when exponential interpolation of scale is desired, rather than the linear interpolation of scale proposed by Shoemake and Duff. RJCF makes the computation practical.

7 Applications Involving Rotations

This section applies the techniques developed in earlier sections to analysis of the cross-product matrix and its matrix exponential; the combination is called the exponential map of a vector in R^3. The inverse problem is also addressed: how to find the axis of rotation and the angle of rotation from the 3×3 matrix. Expressions for matrix exponential and matrix logarithm developed in Section 4, exploiting RJCF, provide a technique to compensate for numerical errors in the matrix when computing the axis of rotation and the angle of rotation, but this topic is deferred to Appendix D.4; for example, columns might not be exactly orthogonal or exactly unit magnitude.

The 3×3 rotation matrices comprise a group, called the *special orthogonal group on* R^3 (SO(3)). The properties that put a matrix in this group are that the determinant is 1, all columns have unit Euclidean length and the columns are mutually orthogonal.

Many additional properties follow from these defining properties. One such is that the columns of $\mathbf{A} \in$ SO(3) define a right-handed coordinate frame such that $\mathbf{A}_j = \mathbf{A}_{j+1} \times \mathbf{A}_{j+2}$ (with subscripts interpreted mod 3, as usual). Another is $\mathbf{A}^{-1} = \mathbf{A}^T$.

7.1 Exponential Map and Cross-Product Matrix

Every vector ω in R^3 can be interpreted as the *exponential coordinates* of a rotation transformation, and every rotation can be represented by exponential coordinates. The term *exponential coordinates* is adopted from Murray, Li, and Sastry [MLS94] and is defined in this section. The mapping from exponential coordinates in R^3 to 3×3 rotation matrices is called the *exponential map*.

Compared to popular alternatives, such as 3×3 matrices and quaternions, the exponential map has the advantage that there are no side conditions on the components of the representation vector; i.e., every vector in R^3 represents some rotation. Exponential coordinates are a special case of six-dimensional *twists* found in the robotics literature [MLS94], also called *fixed-pole generalized velocity* in the mechanics literature [BB98]. Arbitrary rigid-body transformations, which include both rotation and translation, have a twist representations. Exponential maps are starting to become known in the graphics community, often in combination with quaternions, following papers by Kim [KKS95], Grassia [Gra98], Bregler and Malik [BM98], and others. The matrix exponential has recently been used for interpolation between deforming linear transformations by Alexa [Ale02], who provides a good bibliography of its recent use in graphics.

We rederive the exponential map to illustrate the techniques developed in earlier sections, especially outer eigenvectors and relaxed Jordan canonical forms. The exponential map is based on the cross-product matrix and the matrix exponential operator.

Definition 7.1: The cross-product matrix associated with any vector $\omega \in R^3$ is defined as

$$\chi(\omega) \;=\; \begin{bmatrix} 0 & -\omega_2 & \omega_1 \\ \omega_2 & 0 & -\omega_0 \\ -\omega_1 & \omega_0 & 0 \end{bmatrix} \tag{54}$$

That is, $\chi(\omega)\,\mathbf{v} = \omega \times \mathbf{v}$ for all vectors $\mathbf{v} \in R^3$. □

Some useful facts are that

$$(\chi(\omega))^2 \;=\; \omega\omega^T - |\omega|^2\,\mathbf{I}; \tag{55}$$

$$(\chi(\omega))^k \;=\; -|\omega|^2\,(\chi(\omega))^{k-2} \qquad k > 2. \tag{56}$$

Eq. 56 follows from Eq. 55 and the fact that $\chi(\omega)\omega = 0$.

The characteristic polynomial of $\chi(\omega)$ is $f(x) = x^3 + |\omega|^2 x$. Thus the outer eigenvalue is 0 and there are two imaginary roots, $\pm i|\omega|$. Following Section 2.2, we have

$$q(\chi(\omega)) \;=\; (\chi(\omega))^2 + |\omega|^2\,\mathbf{I} \;=\; \omega\omega^T \tag{57}$$

Thus ω is the outer eigenvector, ξ_0, and it is also the normal vector η defined in Eq. 13, and used in Eqs. 87–92 to compute the RJCF. Assuming index 0 contains the maximum-magnitude component of ω, let

$$\mathbf{q} \;=\; |\omega|\,\mathbf{e}_0 \;=\; [|\omega|, 0, 0]^T. \tag{58}$$

We have the relaxed Jordan canonical form

$$\chi(\omega) \;=\; \mathbf{T}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -|\omega| \\ 0 & |\omega| & 0 \end{bmatrix}\mathbf{T}^{-1} \;=\; \mathbf{T}\chi(\mathbf{q})\mathbf{T}^{-1}, \tag{59}$$

where $\mathbf{T}$ may be any orthonormal matrix with $\omega/|\omega|$ as column 0. Similar expressions are obtained by cyclic permutation of rows and columns when a different component has maximum magnitude.

The definition and some well known properties of the matrix exponential operator were given in Eq. 36–38. Recall from that section that $e^{\mathbf{B}}e^{\mathbf{C}}$ does not equal $e^{\mathbf{B}+\mathbf{C}}$ unless the matrices $\mathbf{B}$ and $\mathbf{C}$ commute for multiplication. Cross-product matrices $\chi(u)$ and $\chi(v)$ commute for multiplication if and only if u and v are parallel, i.e., $u \times v = 0$.

Now we substitute $\chi(\mathbf{q})$ from Eq. 59 for $\mathbf{B}$ in Eq. 37 and use Eqs. (55) and (56) to obtain:

$$e^{\chi(\mathbf{q})} \;=\; \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos|\omega| & -\sin|\omega| \\ 0 & \sin|\omega| & \cos|\omega| \end{bmatrix} \tag{60}$$

That is, $e^{\chi(\mathbf{q})}$ represents a rotation of $|\omega|$ radians around the x-axis. From now on we define $\theta = |\omega|$.

Definition 7.2: For general $\omega \in R^3$, $e^{\chi(\omega)}$ is called the *exponential map* of ω [MLS94]. □

Since $e^{\chi(\omega)} = \mathbf{T}e^{\chi(\mathbf{q})}\mathbf{T}^{-1}$, and column 0 of $\mathbf{T}$ is $\omega/|\omega|$, we have rederived the fact that $\mathbf{Q} = e^{\chi(\omega)}$ represents a rotation of $|\omega|$ radians about the axis ω. Again, using Eq. 55, we have the general formula

$$\mathbf{Q} \;=\; e^{\chi(\omega)} \;=\; \mathbf{I} + \chi(\omega)\frac{\sin\theta}{\theta} + (\chi(\omega))^2\frac{(1-\cos\theta)}{\theta^2} \qquad \text{where } \theta = |\omega|. \tag{61}$$

This is known as Rodrigues' formula [MLS94]. In the next subsection, we use our techniques to find the vector ω, given a rotation matrix, $\mathbf{Q}$.

7.2 Finding the Axis of Rotation and Exponential Coordinates

It is known that all matrices in SO(3) can be interpreted as a rotation about some axis $\mathbf{u}$ by some angle θ. That is, the canonical form of such a matrix is

$$\mathbf{J} \;=\; \mathbf{T}^{-1}\mathbf{Q}\mathbf{T} \;=\; \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{62}$$

Since the trace is preserved by similarity transformations, we easily find $\cos\theta = (\mathrm{Q}_{00} + \mathrm{Q}_{11} + \mathrm{Q}_{22} - 1)/2$. The problem we address in this section is the numerically accurate calculation of the axis, $\mathbf{u}$ (of unit length) and an exponential-coordinate vector, $\omega = \mathbf{u}\theta$, that generates $\mathbf{Q}$.

Shoemake used *unit quaternions* to represent rotations [Sho85]. With $\mathbf{u}$ and θ as just defined, the corresponding quaternion is the 4-D row vector:

$$q \;=\; [\mathtt{normv}(\omega)\mathbf{u}^T \sin\theta/2, \cos\theta/2] \tag{63}$$

The last coordinate is called the *scalar part* of q, and ensures unit magnitude. Note that $-q$ represents a rotation of $2\pi - \theta$ around the axis $-\mathbf{u}$, which is geometrically the same rotation as $\mathbf{Q}$. We will return to quaternions later.

Observe that Eq. 61 splits the expression for $\mathbf{Q}$ into symmetric and skew-symmetric parts. Let $\theta = |\omega|$ and let $\mathbf{u} = \omega/\theta$. Thus $\mathbf{Q} - \mathbf{Q}^T = 2\chi(\omega)\sin(\theta)/\theta = 2\chi(\mathbf{u})\sin(\theta)$. For small θ, $\mathbf{u}$ can be calculated accurately from the off-diagonal elements of this matrix difference because the off-diagonal symmetric terms are order θ^2. See Appendix J about normalizing vectors with very small components. We have

$$\begin{aligned} \sin\theta \;&=\; \tfrac{1}{2}\sqrt{(\mathrm{Q}_{01}-\mathrm{Q}_{10})^2 + (\mathrm{Q}_{02}-\mathrm{Q}_{20})^2 + (\mathrm{Q}_{21}-\mathrm{Q}_{12})^2} \\ \mathbf{u} \;&=\; \mathtt{normv}(\left[(\mathrm{Q}_{21}-\mathrm{Q}_{12}), (\mathrm{Q}_{02}-\mathrm{Q}_{20}), (\mathrm{Q}_{10}-\mathrm{Q}_{01})\right]^T). \end{aligned} \tag{64}$$

The singularities are where $\sin\theta = 0$. If $\theta = 0$ (or an integer multiple of 2π), then $\mathbf{Q} = \mathbf{I}$, and all axes are equally valid.

For θ near π, the above calculation can become inaccurate because the off-diagonal symmetric terms can be order 1 while the skew-symmetric terms are order $(\pi - \theta)$; i.e., as θ approaches π (or any odd integer multiple of π), $\mathbf{Q}$ approaches symmetry and in general has nonzero off-diagonal elements.

If $\text{trace}(\mathbf{Q}) < 1$, an alternate formula is used. Instead of Eq. 64, we work with

$$\mathbf{Q} + \mathbf{Q}^T = 2\left((1-\cos\theta)\,\mathbf{u}\mathbf{u}^T + \cos\theta\,\mathbf{I}\right). \tag{65}$$

Using the fact (if $\mathbf{Q}$ is really in SO(3)!) that $\cos\theta = (\text{trace}(\mathbf{Q}) - 1)/2 < 0$, we can solve for $\mathbf{u}\mathbf{u}^T$:

$$\mathbf{u}\mathbf{u}^T = \frac{\mathbf{Q} + \mathbf{Q}^T + (1 - \text{trace}(\mathbf{Q}))\,\mathbf{I}}{(3 - \text{trace}(\mathbf{Q}))} \tag{66}$$

By Eq. 61 we see that the algebraically largest diagonal element of $\mathbf{Q}$ corresponds to the largest-magnitude element of $\mathbf{u}$; assume this is $\mathbf{u}^{(j)}$. For θ to be less than π, $\mathbf{u}^{(j)}$ must have the opposite sign from $(\mathsf{Q}_{j+1,j+2} - \mathsf{Q}_{j+2,j+1})$. Let $\boldsymbol{\xi}$ be the j-th column of the numerator of the right-hand side of Eq. 66 (or j-th row).

$$\begin{aligned}
\xi_j &= 1 + \mathsf{Q}_{j,j} - \mathsf{Q}_{j+1,j+1} - \mathsf{Q}_{j+2,j+2}, \\
\xi_{j+1} &= \mathsf{Q}_{j,j+1} + \mathsf{Q}_{j+1,j}, \\
\xi_{j+2} &= \mathsf{Q}_{j,j+2} + \mathsf{Q}_{j+2,j}; \\
\mathbf{u} &= \text{sgn}(\mathsf{Q}_{j+2,j+1} - \mathsf{Q}_{j+1,j+2})\,\texttt{normv}(\boldsymbol{\xi}); \\
\sin(\tfrac{1}{2}(\pi - \theta)) &= \tfrac{1}{2}\sqrt{(1 + \text{trace}(\mathbf{Q}))},
\end{aligned} \tag{67}$$

where indices are mod 3, as usual. We address the problem of $\mathbf{Q}$ having small errors that cause it not to be a member of SO(3) in Appendix D.4.

Curiously, we did not find an explicit formula for the axis of rotation, given the matrix, in any major graphics texts. A formula similar to Eq. 64 may be inferred from equations to convert rotation matrices to quaternions in some sources [Sho85, appendix I.2], [Wat00, page 489]. However, this formula has singularities for all rotation matrices that represent a 180° rotation, as well as at $\mathbf{A} = \mathbf{I}$. A procedure with better numerical properties is nicely summarized by Nielson [Nie04, appendix A]. The derivation in this section seems to be the first in the graphics community that does not rely on quaternion theory.

8 Comparison of Matrix Lie Product and Polar Decomposition Blending

This section briefly compares some aspects of Alexa's matrix Lie product interpolation and Shoemake and Duff's interpolation based on polar decomposition. The latter seems to have received little attention. Although Alexa cites Shoemake and Duff, there are no comparisons of results. For the comparisons in this section, the object rendered in all cases starts as a unit cube in the positive octant with a corner at the origin; then the current transformation is applied to it. The XZ-faces are red; the YZ-faces are green; the XY-faces are blue. The emphasis is on interpolation between two transformations, not on interpolation between the identity and one transformation.

The first observation is that the two techniques disagree on interpolation of rotations. This is surprising, given the close connection of both methods to exponential coordinates. Interpolating rotations is a basic task in computer animation. It seems that the 10-th root of a 20° rotation should be a 2° rotation, so one might easily believe that the interpolations would be the same for rotations in both techniques. A direct computation on an example shows this to be untrue. The initial rotation, $\mathbf{Q}_0$, is 30° around the Z-axis, and the final rotation, $\mathbf{Q}_1$, is 180° around $[0.5, 0.5, -\sqrt{0.5}]^T$. The latter axis makes an angle of $-45°$ with the XY-plane. Therefore, the axis of rotation changes by 135° over the course of the interpolations. The initial and final matrices, and the two interpolated versions at $\alpha = 0.5$ are (numbers

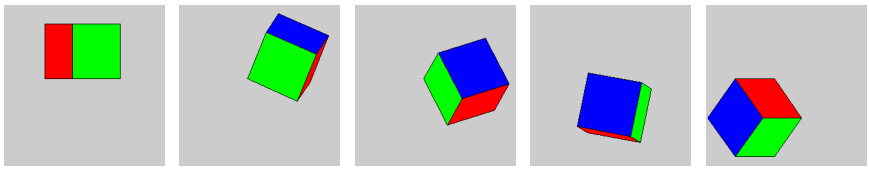

Figure 6: Blend of rotations using Alexa's matrix Lie product. $\alpha = 0, .25, .50, .75, 1.00$.

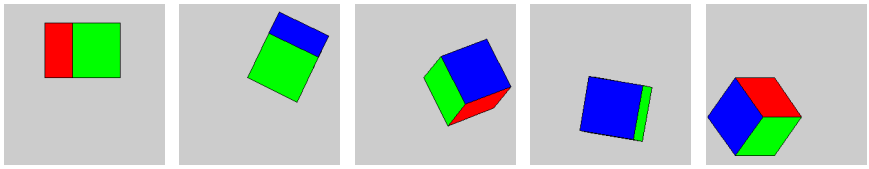

Figure 7: Blend of rotations using Shoemake's `slerp`. $\alpha = 0, .25, .50, .75, 1.00$.

with four decimals are approximate):

$$\begin{bmatrix} \frac{1}{2}\sqrt{3} & -\frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2}\sqrt{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & -\sqrt{\frac{1}{2}} \\ \frac{1}{2} & -\frac{1}{2} & -\sqrt{\frac{1}{2}} \\ -\sqrt{\frac{1}{2}} & -\sqrt{\frac{1}{2}} & 0 \end{bmatrix}$$

$\mathbf{Q}_0$ $\mathbf{Q}_1$

$$\begin{bmatrix} 0.4480 & 0.8360 & 0.3168 \\ -0.2240 & 0.4480 & -0.8655 \\ -0.8655 & 0.3168 & 0.3880 \end{bmatrix} \qquad \begin{bmatrix} 0.4333 & 0.8596 & 0.2709 \\ -0.3368 & 0.4333 & -0.8359 \\ -0.8359 & 0.2709 & 0.4773 \end{bmatrix}$$

`slerp` matrix Lie product

It is evident from the numbers that the matrices are substantially different at $\alpha = 0.5$, and calculation shows about an 7° difference. Also, the differences are apparent visually in Figure 6 and 7. The view is from the positive X direction with Z up. Notice that both sequences show a regular progression of the angles of the blue-green edge, which is parallel to the Y-axis, but the other edge angles vary nonmonotonically.

A further examination shows that the two techniques follow different trajectories; the differences are not due to varying speeds along a common trajectory. We know that `slerp` follows a great-circle arc on the 4-D unit sphere at a uniform speed. On the other hand, from Eq. 61 we know that the matrix-Lie-product follows a straight-line trajectory in the space of exponential coordinates at a uniform speed. In terms of physical justification, `slerp` has the better pedigree because its interpolation corresponds to constant angular velocity in 3-D, not just in quaternion space. Shoemake makes this claim somewhat indirectly [Sho85] and cites a reference, but it seems quite difficult to justify from his `slerp` formula. However, Appendix E gives an improved `slerp` formula using orthonormal basis vectors, which makes the proof feasible; in any case, the fact seems to be accepted in the graphics community.

Another qualitative difference is that `slerp` is invariant with respect to the starting rotation, while matrix Lie product is not. To explain what we mean, consider interpolating between $\mathbf{I}$ and $\mathbf{Q}_1\mathbf{Q}_0^{-1}$. Because $\mathbf{I}$ is the starting point, `slerp` and matrix Lie product *do agree* on this interpolation. In fact, with L representing matrix Lie product interpolation (see Eq. 45), we have

$$L(\mathbf{I}, \Delta, \alpha) \quad = \quad \texttt{slerp}(\mathbf{I}, \Delta, \alpha) \tag{68}$$

for any rotation Δ. For our example, $\Delta = \mathbf{Q}_1\mathbf{Q}_0^{-1}$.

However, `slerp` enjoys the following compositional property:

$$\texttt{slerp}(\mathbf{Q}_0, \Delta\mathbf{Q}_0, \alpha) \quad = \quad \texttt{slerp}(\mathbf{I}, \Delta, \alpha)\,\mathbf{Q}_0.$$

In other words, with `slerp` it is sufficient to know the desired "relative rotation," Δ, and the interpolation can be composed with the starting rotation, $\mathbf{Q}_0$, to achieve the same result as interpolating directly from $\mathbf{Q}_0$ to $\Delta\mathbf{Q}_0$.

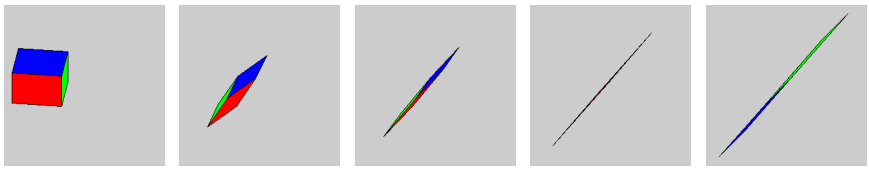

Figure 8: Blend using Alexa's matrix Lie product. $\alpha = 0,\ 0.25,\ 0.50,\ 0.75,\ 1.00$.

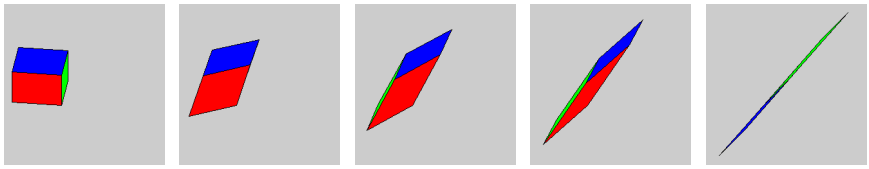

Figure 9: Blend using polar decomposition, `slerp`, and true linear interpolation. $\alpha = 0,\ 0.25,\ 0.50,\ 0.75,\ 1.00$.

In contrast, with L representing matrix Lie product interpolation (see Eq. 45),

$$L(\mathbf{Q}_0, \Delta\mathbf{Q}_0, \alpha) \quad \neq \quad L(\mathbf{I}, \Delta, \alpha)\,\mathbf{Q}_0.$$

unless $\mathbf{Q}_0$ and Δ are rotations around a common axis.

For the current example, the inequality is demonstrated by letting $\mathbf{Q}_1 = \Delta\mathbf{Q}_0$ and observing that the right-hand side agrees with Figure 7, in view of Eq. 68, while the left-hand side agrees with Figure 6. Alexa's paper acknowledges that this is the case, but takes the position that $L(\mathbf{I}, \Delta, \alpha)\,\mathbf{Q}_0 \neq \mathbf{Q}_0\,L(\mathbf{I}, \Delta, \alpha)$, so it is not a desirable interpolant, anyway.

The next comparison, shown in Figure 8 and 9, blends from an orthogonal orientation to one that is ill-conditioned and not orthogonal, using the matrices **C** and **B**, described in Section 5.1, except that the middle element of **B** was increased by 1 percent to remove the singularity. The sequences show the progression from **C** to **B**. The view is from azimuth $-37.5°$, elevation $30°$, where the positive X-axis is azimuth 0 and the XY-plane is elevation 0. The Z-axis is up.

Here we see a noticeable difference between the techniques. By the time $\alpha = 0.50$, the matrix Lie product is almost completely flattened, as predicted in Section 5.1, while the polar decomposition is more believable as a midpoint between the two transformations.

Our proposed L_{mix} (in Eq. 53) is not illustrated because for the first example it is exactly the same as `slerp`, and for the second example it is very close to Figure 8.

One might object that the chosen examples are biased against Alexa's method, in favor of polar decomposition, but we are not aware of any examples that favor Alexa's method, relative to polar decomposition, and his paper did not include any. Practical experience is needed to determine how well any of these schemes work for interpolating in computer animation. This paper's contribution is primarily to provide analytical insights and computational tools for both methods, based on RJCF analysis.

9 Streamline Characteristics and Relaxed Jordan Canonical Forms

The streamline characteristics for two-dimensional linear differential equations are discussed in detail in many sources [HS74, HK91, BD96]. Nielson and Jung describe the three-dimensional linear and affine cases with constant coefficients in detail [NJ99], but without distinguishing all topologically different cases. We include a brief review to identify the topologically distinct linear cases and to consider numerical issues. We describe the unique three-dimensional topological case that does *not* separate into a two-dimensional case and a one-dimensional case.

Streamlines are solutions of $d\mathbf{p}/dt = \mathbf{A}\mathbf{p}$ with an initial condition $\mathbf{p}(0) = \mathbf{p}_0$. (If **A** does not have full rank, then nonhomogeneous differential equations require separate treatment [NJ99]; these are equations with an additive

constant **c** on the right-hand side. Whenever **A** has full rank, a translation of coordinates can remove such an additive constant.)

Solving the differential equation in canonical coordinates provides a uniform procedure with good numerical behavior. Let $\mathbf{J} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}$ be the relaxed Jordan canonical form of **A** and $\mathbf{q}_0 = \mathbf{T}\mathbf{p}_0$. Then $\mathbf{q}(t) = e^{\mathbf{J}t}\mathbf{q}_0$ and $\mathbf{p}(t) = \mathbf{T}^{-1}\mathbf{q}(t)$. The matrix exponential formulas for relaxed Jordan canonical form were given in Section 4.

Two matrices **A** and **B** are topologically equivalent if there is a continuous 1-1 mapping between the streamlines of **A** and the streamlines of **B** [Lad73]. This is the case if and only if **A** and **B** are related through a similarity transformation [KR73] (recall Definition 1.1 in Section 1.4). Thus it suffices to consider the relaxed Jordan canonical forms.

First, we examine how streamlines can be partitioned when the outer eigenvalue, λ_0, has multiplicity 1, i.e., almost all cases. Recall from Section 2.2, Eq. 13, that $\boldsymbol{\eta}$ is orthogonal to the invariant subspace for the other two eigenvalues, λ_1 and λ_2. It is useful in many applications to know that no streamlines can cross the plane that defines this invariant subspace, i.e., the plane that passes through the critical point and is orthogonal to $\boldsymbol{\eta}$. As shown in the left part of Figure 2, the outer eigenvector is *not* orthogonal to the invariant subspace, in general. Planes orthogonal to this eigenvector have no special significance, as far as we know.

When there are three independent, real, eigenvectors, then each pair of linearly independent eigenvectors, anchored at the critical point, defines a (possible) canonical coordinate plane and streamlines cannot cross any such plane. If an eigenvalue has multiplicity greater than one, there might be infinitely many linearly independent eigenvectors, hence the word "possible" in the prior sentence.

The 2-D classifications of streamlines are well covered in the cited sources and are based on the canonical forms described in Section 2.5. If the off-diagonal elements have opposite signs in the the result of step 2 in that section, then the streamlines are spirals or circles. If the off-diagonal elements have the same sign, the streamlines follow a polynomial curve, such as $(y/y_0)^{\lambda_1} = (x/x_0)^{\lambda_2}$. The degrees may be fractional, of course. If both off-diagonal elements are 0, then $\lambda_2 = \lambda_1$ and all straight lines through the origin are streamlines. With exactly one off-diagonal element equal to 0, again $\lambda_2 = \lambda_1$, but now the streamlines of the associated vector field behave like "defective" spirals, which are topologically distinct from the straight lines of the previous case. There is only one distinct eigenvector and all streamlines tend to become parallel to it.

In most cases, 3-D transformations separate into a 1-D system, corresponding to the outer eigenvalue, and a 2-D system. If the outer eigenvector is the *z*-axis in the 1-D system, the streamline has $z = z_0 e^{\lambda_0 t}$, so t can be eliminated with $t = (\log z - \log z_0)/\lambda_0$. Note that t is single-valued in z, but might *not* be single-valued in x or y.

The only 3-D case that does not separate is when the relaxed canonical form is a triangular matrix, as defined in Definition C.2 of Appendix C.2, with two or three nonzero off-diagonal elements. Necessarily, the diagonal elements are all equal to λ_0, the eigenvalue. If A_{ij} is the corner element, then $\mathbf{e}_i$ is the unique eigenvector. This is the only case with three equal eigenvalues for which the streamlines can be nonplanar.

Other cases with three equal eigenvalues have either two independent eigenvectors (planar, but curved, streamlines) or three independent eigenvectors (all straight lines through the origin are streamlines and *every* nonzero vector is an eigenvector). The various cases are not topologically equivalent. Appendix C describes how to distinguish among these cases.

Some texts suggest that the special cases that result from repeated eigenvalues can be avoided by perturbing **A** slightly [HS74]. Cases in which all eigenvalues are distinct are called *hyperbolic*. Although perturbation might simplify the mathematical treatment, numerically it can be a disaster. When eigenvalues are close to each other, some of the formulas given by Nielson and Jung become numerically unstable, such as their equation 2.8 and several following cases. Moreover, perturbation might be imposed accidentally, as a result of numerical inaccuracies. Example I.8 in Appendix I gives one dramatic example in which eigenvalues with imprecision on the order of 10^{-15} give rise to errors on the order of 10^{15}.

10 Experimental Results

The procedures described here have been implemented in ANSI C as `findCanon3D` and `findCanon2D` using double precision floating point. The code is available from the author for noncommercial use.

The general problem of finding eigenvectors for nonsymmetric matrices is considered too advanced for *Numerical Recipes in C*, whose authors suggest using an "eigen-package" [PFTV88]. We used the C extensions of *matlab*

$$\begin{bmatrix} -15 & -16 & -20 \\ 44 & 45 & 19 \\ 16 & 16 & 21 \end{bmatrix}$$ Example I.1 (a)

$$\begin{bmatrix} -32 & -16 & -20 \\ 44 & 28 & 19 \\ 16 & 16 & 5 \end{bmatrix}$$ (b)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 45 & -25/\sqrt{2} \\ 0 & 32/\sqrt{2} & 5 \end{bmatrix}$$ Eq. 146 (c)

$$\begin{bmatrix} 3 & 5 & -3 \\ 1 & 7 & -3 \\ 2 & 10 & -4 \end{bmatrix}$$ Example C.1 (d)

$$\begin{bmatrix} -3 & 2 & -2 \\ -8 & 4 & -3 \\ 15 & -5 & 8 \end{bmatrix}$$ Example C.3 (e)

$$\begin{bmatrix} 10 & 0 & 0 \\ 0 & -1 & 2 \\ 0 & 1 & 3 \end{bmatrix}$$ Example I.4 (f)

$$\begin{bmatrix} 5 & -3 & 6 \\ -3 & 5 & -6 \\ 0 & 0 & 2 \end{bmatrix}$$ (g)

Figure 10: Matrices studied emprically and where they are discussed in the paper, where applicable.

Matrix source	Eigenvalues	(# distinct eigenvectors)	Max abs. eigenvalue error `findCanon3D`	`mlfEig`
(a)	1, 25, 25	(2)	0.0	$5 \cdot 10^{-7}$
(b)	-16, 8, 8	(2)	0.0	$5 \cdot 10^{-7}$
(c)	1, 25, 25	(2)	0.0	$3 \cdot 10^{-7}$
(d)	2, 2, 2	(2)	0.0	$1 \cdot 10^{-15}$
(e)	3, 3, 3	(1)	0.0	$3 \cdot 10^{-5}$
(f)	$10, 1+\sqrt{6}, 1-\sqrt{6}$	(3)	0.0	$9 \cdot 10^{-16}$
(g)	8, 2, 2	(3)	$4 \cdot 10^{-16}$	$9 \cdot 10^{-16}$

Table 4: Accuracy of eigenvalue calculations.

[Mol88] for a comparison implementation. Their function is named `mlfEig` and also uses double precision. This function invokes routines from BLAS and/or LAPACK [DDCDH90].

To gather timing data, a set of ten matrices, including the examples in this paper, was solved 100,000 times, for a total of 1,000,000 procedure calls. An SGI workstation with an R12000 MIPS processor, 360 MHz, was used.

Our `findCanon3D` required 7.8 microseconds per procedure call. The `mlfEig` procedure required 184 microseconds per procedure call in its default mode. With the "nobalance" flag, the time dropped to 155 microseconds. To check whether the `mlf` interface alone incurred a lot of overhead, we conducted a similar test, just getting the 1-norm of the matrix, instead of the eigen-decomposition. This required only 9 microseconds per call, indicating that the interface alone is not consuming much time. We observed similar results calling the CBLAS library [DDCDH90] directly from our C code.

Of course, `mlfEig` is a general procedure for all $n \times n$ complex matrices, so these results do not imply any deficiency in it or the underlying BLAS code. However, they do show that the new technique is computationally much simpler for 3×3 real matrices. Apparently our method is roughly 20 times faster.

In another comparison on a newer 2.6GHz Linux/AMD platform, with the same test data, using `matlab` interactively, their built-in function `eig` (equivalent to `mlfEig` in the other environment) required 17.43 microseconds per procedure call with the "nobalance" flag. This gain is consistent with the increased speed of the computer, and a "do-nothing" function took 1.2 microseconds per call. It also shows, together with several other tests, that using `matlab` interactively does not by itself incur much overhead. On this platform our C code required 0.94 microseconds per procedure call. This is still 17 times faster than the general procedure.

We performed a quick `matlab` implementation of `findCanon3D` to run interactively. Using Cardano equations to find the outer root (the fasterchoice), its time jumped to 752 microseconds per call, 800 times slower than the C code. This confirms that low-level `matlab` code incurs a large overhead, compared to C code; it is much better when the code can use the high-level operations offered in `matlab`. Our implementation was mostly line by line conversion from C, although we used obvious matrix and vector operations where possible.

Now we address the question of accuracy, which was the primary motivation for developing a new specialized method. Figure 10 shows the matrices studied, which are from examples, or are closely related to examples. First we consider how accurately the eigenvalues themselves were calculated (see Table 4).

The general method incurs errors as early as the sixth significant digit, even in cases where the characteristic polynomial has exact roots. Machine accuracy is at least 16 decimal digits. One ULP (unit in the last place, also called *machine epsilon*) equals $1.1 \cdot 10^{-16}$ for numbers between 0.5 and 1.0. We observe that the general method has greatest difficulties when the eigenvectors do not span the space. This tendency has been documented [PFTV88].

For the numbers reported in the tables our method uses Newton-Raphson iteration to find the outer eigenvalue (Appendix A.1). It achieved machine accuracy in most cases, and 2 ULPs error in one case. In particular, when

the eigenvectors do not span the space, our new method maintains its accuracy. Tests were also run using Cardano's equations to find the outer eigenvalue (Appendix A.2), and they exhibited similar times and accuracy.

The accuracy of eigenvectors is perhaps even more important than that of eigenvalues. Matrix (d) provides an extreme example of how sensitive traditional methods can be. Recall the discussion of discontinuity in Section 2.6. Because matrix (d) is a simple shear case, we may expect trouble with the eigenvectors. Our specialized method (both C and `matlab` versions) finds two orthogonal eigenvectors, whose approximate values are $[-0.897, 0.345, 0.276]$ and $[0.408, 0.408, 0.816]$. In addition, the generalized eigenvector $[0.169, 0.845, -0.507]$ is found. The general method, `eig` (without the `nobalance` flag—it is less accurate with that flag in this case), finds three distinct eigenvectors near $[0.408, 0.408, 0.816]$, which differ in the 15-th digit. They define the following transforming matrix

$$\mathbf{T} = \begin{bmatrix} -0.40824829046386352 & 0.40824829046386335 & -0.40824829046386213 \\ -0.40824829046386285 & 0.40824829046386296 & -0.40824829046386335 \\ -0.81649658092772592 & 0.81649658092772615 & -0.81649658092772637 \end{bmatrix} \tag{69}$$

This matrix is too badly conditioned for an accurate inverse to be computed. The approximate inverse is

$$\mathbf{T}^{-1} \approx \begin{bmatrix} -3.603 \cdot 10^{15} & -17.294 \cdot 10^{15} & 10.448 \cdot 10^{15} \\ -3.603 \cdot 10^{15} & -20.897 \cdot 10^{15} & 12.250 \cdot 10^{15} \\ 0 & -3.603 \cdot 10^{15} & 1.801 \cdot 10^{15} \end{bmatrix} \tag{70}$$

Matrix (d) is also discussed in Example I.8.

11 Conclusion

Relaxed Jordan canonical form has been introduced as a tool for manipulating linear transformations, particularly for interpolation purposes, and analyzing related problems. Procedures were described to compute a unit-determinant transformation that changes the coordinate system into a (relaxed) canonical system, which exposes the structure of the eigenvectors and eigenvalues. The key ideas were the definition of the outer root of a cubic polynomial and the exploitation of the Cayley-Hamilton theorem as specialized for three dimensions. Preliminary tests have shown it to be about 17–20 times faster than off-the-shelf "eigen-packages," as well as being more accurate. However, it is much slower if implemented in `matlab`, rather than C.

Efficient closed forms have been developed for several matrix problems that have had only iterative solutions published previously. The themes for most of the improvements were the exploitation of the structure imparted by the outer eigenvalue and outer eigenvector. Matrix logarithms and matrix exponentials were shown to have explicit formulas for these relaxed canonical forms. These results led to closed forms for Alexa's matrix Lie product operator and for polar decomposition without requiring matrix inverse. Thus, RJCF provides closed-form procedures for several matrix operations that are needed by various matrix animation techniques. It also makes some extensions of these techniques possible, primarily because it tolerates matrices with zero or negative eigenvalues.

Besides numerical procedures, RJCF provides an analytical tool. New solutions were developed for the problem of finding parallel vectors in two vector fields. New insights were gained into the structure of polar decomposition of matrices with rank deficiency of one.

Experimental results, though not exhaustive, indicate greater accuracy and significant speed-ups for the new techniques. The techniques are reasonably simple to implement, so we anticipate that they will become standard in future three-dimensional graphics packages. (Some code is available from the author for noncommercial research use.) However, the techniques are largely specialized to three-dimensional space and do not suggest extensions to higher dimensions.

Acknowledgments

Work was partially supported by the UCSC/Los Alamos Institute for Scalable Scientific Data Management (ISSDM).

References

[Ale02] M. Alexa. Linear combination of transformations. In *ACM Transactions on Graphics (Siggraph Proceedings)*, pages 380–387, 2002.

[BB98] C. L. Bottasso and M. Borri. Interating finite rotations. *Computer Methods in Applied Mechanics and Engineering*, 164:307–331, 1998.

[BD96] W. E. Boyce and R. C. DiPrima. *Elementary Differential Equations*. Wiley, New York, 6th edition, 1996.

[BM98] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *IEEE Conf. on Computer Vision and Pattern Recognition*, Santa Barbara, CA, 1998. IEEE Comput. Soc.

[DDCDH90] J. Dongarra, J. Du Croz, I. Duff, and S. Hammarling. A set of level-3 basic linear algebra subprograms. *ACM Transactions on Mathematical Software*, 16:1–17, 1990. (See also `http://www.netlib.org/cgi-bin/checkout/blast/blast.pl`).

[DH94] T. Delmarcelle and L. Hesselink. The topology of symmetric, second-order tensor fields. In *IEEE Visualization*, pages 140 –147, 1994.

[FvDFH90] James D. Foley, Andries van Dam, Steven Feiner, and John Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley Publishing Company, Reading, Mass., 2nd edition, 1990.

[Gan59] F. R. Gantmacher. *Matrix Theory*. Chelsea, New York, 1959.

[Gra98] F. S. Grassia. Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools*, 3:29–48, 1998.

[GVL96] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

[Hig86] N. J. Higham. Computing the polar decomposition – with applications. *SIAM Journal on Scientific and Statistical Computing*, 7:1160–1174, 1986.

[HK91] Jack K. Hale and Huseyin Koçak. *Dynamics and Bifurcations*. Springer-Verlag, New York, 1991.

[Hod41] C. D. Hodgman. *Mathematical Tables from Handbook of Chemistry and Physics*. Chemical Rubber Publishing, Cleveland, OH, 7th edition, 1941.

[HP94] N. J. Higham and P. Papadimitriou. A parallel algorithm for computing the polar decomposition. *Parallel Computing*, 20:1161–1173, 1994.

[HS74] M.W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems, and Linear Algebra*. Academic Press, New York, 1974.

[HS90] N. J. Higham and R. S. Schreiber. Fast polar decomposition of an arbitrary matrix. *SIAM Journal on Scientific and Statistical Computing*, 11:648–655, 1990.

[KKS95] M.J. Kim, M.S. Kim, and S.Y. Shin. A general construction scheme for unit quaternion curves with simple high-order derivatives. *Computer Graphics (ACM SIGGRAPH Proceedings)*, 29(369–376), August 1995.

[KR73] N. H. Kuiper and J. W. Robbin. Topological classification of linear endomorphisms. *Inventiones mathematicae*, 19:83–106, 1973.

[Lad73] N. N. Ladis. Topological equivalence of linear flows. *Differential Equations*, 9:938–947, 1973. Translated from Russian.

[MLS94] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC, Boa Raton, FL, 1994.

[Mol88] C. Moler. MATLAB—a mathematical visualization laboratory. In *Thirty-Third IEEE Computer Society Int'l Conf.*, San Francisco, 1988. (See `http://www.mathworks.com/products/matlab`.

[Nie04] Gregory M. Nielson. v-quaternion splines for the smooth interpolation of orientations. *IEEE Transactions on Visualization and Computer Graphics*, 10:224–229, 2004.

[NJ99] Gregory M. Nielson and Il-Hong Jung. Tools for computing tangent curves for linearly varying vector fields over tetrahedral domains. *IEEE Transactions on Visualization and Computer Graphics*, 5:360–372, 1999.

[PFTV88] W. Press, B. Flannery, S. Teukolsky, and W. Vettering. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.

[PR97] F. C. Park and B. Ravani. Smooth invariant interpolations of rotations. *Transactions on Graphics*, 16(3):277–295, July 1997.

[PR99] R. Peikert and M. Roth. The parallel vector operator – a vector field visualization primitive. In D. Ebert, M. Gross, and B. Hamann, editors, *Proceedings IEEE Visualization '99*, pages 263–270. IEEE Computer Society Press, 1999.

[Rot00] Martin Roth. *Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization*. PhD thesis, ETH Zurich, Institute of Scientific Computing, 2000. Diss. ETH No. 13673.

[SD92] K. Shoemake and T. Duff. Matrix animation and polar decomposition. In *Graphics Interface*, pages 258–264, Vancouver, 1992. CHCCS.

[Sho85] Ken Shoemake. Animating rotation with quaternion curves. *SIGGRAPH '85 Conference Proceedings*, 19(3):245–254, July, 1985.

[TSW+05] Holger Theisel, Jan Sahner, Tino Weinkauf, Hans-Christian Hege, and Hans-Peter Seidel. Extraction of parallel vector surfaces in 3D fields and application to vortex core line tracking. In *Proc. IEEE Visualization 2005*, Minneapolis, USA, October 2005. accepted for publication.

[Wat00] A. H. Watt. *3D Computer Graphics*. Addison-Wesley, Reading, MA, 3rd edition, 2000.

[Wei99] E. W. Weisstein. Cubic equation – from mathworld, 1999. (URL: `http://mathworld.wolfram.com/CubicEquation.html`).

[ZZ95] P. Zieliński and K. Ziętak. The polar decomposition — properties, applications and algorithms. *Applied Mathematics, Annals of the Polish Mathematical Society*, 38:23–49, 1995.

Appendix A Outer Root Finding Procedures for Cubic Polynomials

This section describes two procedures to find the outer root of a cubic polynomial,

$$f(x) = x^3 - tx^2 + sx - d. \tag{71}$$

The first is based on Newton-Raphson iteration. The second is based on the classical Cardano equations for roots of a cubic polynomial. The Cardano method requires inverse trigonometric functions to be evaluated; its robustness depends on the robustness of those functions, which are usually imported from a library. The first method does not depend on any transcendental library functions.

As described in Section 2.1 an *outer root* of f is a real root that is "most distinct" from the other real roots. If λ is the *only* real root of f, of course it is the outer root. Otherwise, let the three real roots be λ_0, λ_1, λ_2. In this case λ_0 is an outer root if $|\lambda_0 - \lambda_1| \geq |\lambda_2 - \lambda_1|$ and $|\lambda_0 - \lambda_2| \geq |\lambda_1 - \lambda_2|$. This simple definition is applicable only to cubic polynomials.

Let $\widetilde{x}$ be the point where $d^2 f/dx^2 = 0$. (See Figure 1 in Section 2.1.) The two-dimensional point $(\widetilde{x}, f(\widetilde{x}))$ is called an *inflection point* on the graph of $f(x)$. We have:

$$\widetilde{x} = t/3 \tag{72}$$

$$f(\widetilde{x}) = -2t^3/27 + st/3 - d \tag{73}$$

As shown in Figure 1, the sign of $f(\widetilde{x})$ determines the relative location of the outer root λ: if $f(\widetilde{x}) < 0$, then $\lambda > \widetilde{x}$; if $f(\widetilde{x}) > 0$, then $\lambda < \widetilde{x}$. If $f(\widetilde{x})$ is exactly zero, there are two candidate outer roots, and we select the one with larger magnitude. If the magnitudes are equal, we arbitrarily select the positive one. If the sign of $f(\widetilde{x})$ is ambiguous numerically, it does not matter if the wrong root is chosen as the outer root; what is important is that it can be computed robustly.

Appendix A.1 Outer Root via Newton-Raphson

The key to the Newton-Raphson iteration is to find a starting value that will converge monotonically and quickly. Recall Eq. 73. Also note that $f'(\widetilde{x}) = -t^2/3 + s$. Let λ denote the outer root being sought. If $f(\widetilde{x}) < 0$, then $\lambda > \widetilde{x}$; if $f(\widetilde{x}) > 0$, then $\lambda < \widetilde{x}$. Our immediate goal is to find a value x_0 such that λ is guaranteed to be between $\widetilde{x}$ and x_0.

We derive a safe x_0 for the case $f(\widetilde{x}) < 0$; the other case is similar, with appropriate signs reversed. First we determine $\Delta x > 0$ such that $f(\widetilde{x} + \Delta x) = f(\widetilde{x})$. Direct calculation shows that $\Delta x = \sqrt{-f'(\widetilde{x})}$. For all $x > \widetilde{x} + \Delta x$, both $f'(x)$ and $f''(x)$ are positive, so we are guaranteed that $f(x_0) > 0$ for

$$x_0 = \widetilde{x} + \Delta x + \sqrt[3]{|f(\widetilde{x})|} \qquad f(\widetilde{x}) < 0. \tag{74}$$

(A simpler safe choice that might require more iterations is $x_0 = (1 + |t| + |s| + |d|)$.) Similarly,

$$x_0 = \widetilde{x} - \Delta x - \sqrt[3]{|f(\widetilde{x})|} \qquad f(\widetilde{x}) > 0. \tag{75}$$

(A simpler safe choice that might require more iterations is $x_0 = -(1 + |t| + |s| + |d|)$.) Moreover, it suffices to use an upper bound on $\sqrt[3]{|f(\widetilde{x})|}$ to calculate the starting value x_0.

Now the Newton-Raphson iteration is

$$x_n \;=\; x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} \;=\; \frac{x_{n-1} f'(x_{n-1}) - f(x_{n-1})}{f'(x_{n-1})} \qquad n \geq 1. \tag{76}$$

In theory this converges monotonically to λ, because $f(x)$ and $f''(x)$ have the same sign throughout an interval that includes λ and x_0.

In practice, if the value calculated for $f'(x_{n-1})$ is too small due to numerical errors, x_n can overshoot λ, then x_{n+1} will overshoot in the other direction, and be further from λ than x_{n-1}. For this reason, if x_n has the opposite sign from x_{n-1} we finish the calculation by Newton's *bisection* method [PFTV88], starting with x_{n-1} and x_n. This method does not calculate f'. We terminate when the interval surrounding λ has shrunk to 1 ULP.

If all three roots of $f(x)$ are equal, or nearly equal, poor numerical behavior can result. The derivative $f'(x_{n-1})$ approaches zero as $f(x_{n-1})$ approaches the root. Aside from numerical cancellation errors, the convergence to the root becomes "linear" rather than the usual "quadratic" [PFTV88]. In practice, three nearly equal roots occur rarely, except for uniform scale matrices, which are easy to recognize. However, if the possibility exists in the applications under consideration, the Cardano method (Section A.2) is comparably efficient and appears to be more robust for this situation.

Appendix A.2 Outer Root via Cardano's Equations

Cubic equations can be solved in closed form using a system known as Cardano's equations, but actually discovered about 1500 by dal Ferro at University of Bologna, and improved subsequently [Hod41, PFTV88, Wei99]. We present a simplified version of Cardano's equations that is specialized toward finding the outer root of a cubic polynomial, $f(x) = x^3 - tx^2 + sx - d$. Numerical accuracy is good even if all three roots are equal or nearly equal. The first step in the derivation is to substitute $x = y + t/3$ in $f(x)$, giving:

$$g(y) \;=\; y^3 + py - q \qquad \text{where } p \;=\; s - \frac{t^2}{3} \quad \text{and} \quad q \;=\; d + \frac{(2t^2 - 9s)\,t}{27} \tag{77}$$

We now find the values of y that solve $g(y) = 0$. If $p = 0$, there are three equal real roots, $y = \sqrt[3]{q}$.

If $q = 0$, one root is $y = 0$ and the others are $\pm\sqrt{-p}$, which may be real or imaginary. If $q = 0$ and $p < 0$, the outer root is $y = -\sqrt{-p}$ when $t < 0$ and is $y = \sqrt{-p}$ when $t \geq 0$. If $q = 0$ and $p > 0$, then $y = 0$ is the only real root. We assume hereafter that none of these simple cases occurred, i.e., that neither p nor q is zero.

The second step is to compute a discriminant D. The standard equation for D can be rewritten to eliminate several higher powers of t that would otherwise cancel numerically, with possible loss of accuracy:

$$D = \left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2 = \frac{4dt^3 - s^2t^2 - 18sdt + 4s^3 + 27d^2}{108}. \tag{78}$$

Horner's rule can be used to evaluate Eq. 78 with fewer operations. If $D > 0$, there is one real root,

$$y = \sqrt[3]{\frac{q}{2} + \sqrt{D}} + \sqrt[3]{\frac{q}{2} - \sqrt{D}} \quad \text{for } D > 0. \tag{79}$$

Fast and accurate cube root functions are provided in the standard C math library as `cbrt`, in *matlab* as `nthroot`, and in Appendix J if these sources are not available. (For completeness, although they are not needed for RJCF purposes, the complex roots are $-y/2 \pm \frac{1}{2}\sqrt{3}\left(\sqrt[3]{\frac{1}{2}q + \sqrt{D}} - \sqrt[3]{\frac{1}{2}q - \sqrt{D}}\right) i$, where y is the real root.)

If $D \le 0$, there are three real roots. They depend on parameters r and θ_0, whose equations are given below. They are of the form $r\cos\theta$, where θ is either θ_0, $\theta_0 + 2\pi/3$, or $\theta_0 - 2\pi/3$. Thus the roots are the x-ordinates of three equi-spaced points on a circle of radius r, centered at the origin of an xy-plane, which we call the "Cardano wheel."

The correct choice for the *outer* root is the one that maximizes $|\cos\theta|$. Note that $p < 0$, since $D \le 0$.

$$r = 2\sqrt{\frac{-p}{3}}, \qquad \cos(3\theta_0) = \frac{3q}{-pr} \qquad 0 \le 3\theta_0 \le \pi. \tag{80}$$

For $q < 0$, $\theta = \theta_0 + 2\pi/3$ gives the outer root. For $q > 0$, $\theta = \theta_0$ gives the outer root.

The equations below select the correct outer root and use the inverse sine in the ranges where the inverse cosine becomes inaccurate.

$$r = 2\sqrt{-p/3} \tag{81}$$

$$\theta = \frac{1}{3}\cos^{-1}\left(\frac{3|q|}{-pr}\right) \qquad \text{for } |q| < \frac{-pr}{6} \tag{82}$$

$$\theta = \frac{1}{3}\sin^{-1}\left(\frac{6\sqrt{-D}}{-pr}\right) \qquad \text{for } |q| \ge \frac{-pr}{6} \tag{83}$$

$$y = \mathrm{sgn}(q)\, r\cos\theta \tag{84}$$

When $q < 0$, the absolute value in Eq. 82 and the sign function in Eq. 84 are justified by the identities $\cos(3\theta_0) = -\cos(\pi - 3\theta_0)$ and $\cos(\theta_0 + 2\pi/3) = -\cos(\pi/3 - \theta_0)$.

Numerical inaccuracies might make the sign of D uncertain, resulting in y being computed by Eq. 79 in one case and Eq. 84 in the other. However, the value of y varies continuously through this switch in equations, provided that p and q are accurate. When all three roots are nearly equal, p and q are "close" to zero. Specifically, $|p| \ll |t|^2$ and $|q| \ll |t|^3$. In this case p and q might be inaccurate. However, the computed value of y will be similarly small, in comparison to $|t|$, so the outer root found for $f(x)$ will be reasonably accurate.

Appendix B Three-Dimensional Canonical Coordinates

This section provides the detailed mathematical exposition of the relaxed Jordan canonical form (RJCF). It should be read in conjunction with Section 2, which provides the overview and continuity.

Our method for the analysis of a 3×3 matrix $\mathbf{A}$ begins with its *characteristic polynomial*:

$$f(x) = \det(x\mathbf{I} - \mathbf{A}) = x^3 - tx^2 + sx - d. \tag{85}$$

where x is a scalar variable. The coefficients are given by

$$\begin{aligned} t &= \mathsf{A}_{00} + \mathsf{A}_{11} + \mathsf{A}_{22} \\ s &= (\mathsf{A}_{00}\mathsf{A}_{11} - \mathsf{A}_{01}\mathsf{A}_{10}) + (\mathsf{A}_{11}\mathsf{A}_{22} - \mathsf{A}_{12}\mathsf{A}_{21}) \\ &\quad + (\mathsf{A}_{22}\mathsf{A}_{00} - \mathsf{A}_{20}\mathsf{A}_{02}) \\ d &= \det(\mathbf{A}) = (\mathbf{A}_0 \times \mathbf{A}_1) \cdot \mathbf{A}_2. \end{aligned} \tag{86}$$

Note that t is the *trace* of $\mathbf{A}$, d is the determinant of $\mathbf{A}$, and s is the sum of the determinants of the principal minors of $\mathbf{A}$. All three quantities are invariant under similarity transformations.

The roots of $f(x) = 0$ are the eigenvalues of $\mathbf{A}$. Being a cubic polynomial, f has one or three real roots. Let us assume for this section that at least two roots are distinct, and postpone consideration of the case that all three roots are equal to Appendix C.

As defined in Definition 2.1 in Section 2.1, an *outer root* of f is a real root that is "most distinct" from the other real roots, and is the *outer eigenvalue* of the matrix being transformed into RJCF.

The outer root can be efficiently computed by a numerical procedure, or by careful application of Cardano's solution for cubic equations. Both methods are described in Appendix A, with optimizations for the fact that the cubic polynomial corresponds to a characteristic polynomial. The role of the outer eigenvalue in computing a transformation into RJCF is discussed in Section 2.

Appendix B.1 Separation of Eigenspaces

This section provides the computational details for the separating matrix $\mathbf{RS}$ that separates the three-dimensional RJCF problem into a one-dimensional problem and a two-dimensional problem. See Section 2.2 for an overview. The quantities ξ_0, η, γ, i and j have been computed as described in the paragraph before Lemma 2.2. A composition of a rotation and a shear aligns η with the nearest coordinate axis and aligns ξ_0 with η, as suggested by Figure 2.

The first task is to find a rotation $\mathbf{R}$ such that $\mathbf{R}^{-1}\eta$ is aligned with the nearest coordinate axis. The second task is to find a shear transformation $\mathbf{S}$ such that $\mathbf{S}^{-1}\mathbf{R}^{-1}\xi_0$ is aligned with this same axis. This axis is the eigenvector direction for λ_0 in the canonical coordinate system, so the separation is accomplished. The rotation and shear transformations, $\mathbf{R}$ and $\mathbf{S}$, respectively, each have determinant one. The methods to compute $\mathbf{R}$ and $\mathbf{S}$ are applications of standard techniques in linear algebra.

We now give the details for the rotation transformation, $\mathbf{R}$. The components of vectors and matrices are indexed with 0, 1, 2, and arithmetic on such indexes is always understood to be mod 3.

Recall that j is the index of the column that became ξ_0. The columns of the rotation matrix $\mathbf{R}$ are the unit-length vectors $\mathbf{R}_0$, $\mathbf{R}_1$, $\mathbf{R}_2$, defined as:

$$\mathbf{R}_j = \frac{\eta}{|\eta|} \qquad \mathbf{R}_{j+1} = \frac{\mathbf{e}_{j+2} \times \eta}{|\mathbf{e}_{j+2} \times \eta|} \qquad \mathbf{R}_{j+2} = \frac{\mathbf{R}_j \times \mathbf{R}_{j+1}}{|\mathbf{R}_j \times \mathbf{R}_{j+1}|} \tag{87}$$

Transposing $\mathbf{R}$ produces $\mathbf{R}^{-1}$. Because a positive component of η has maximum magnitude, the angle of rotation associated with $\mathbf{R}$ is in the range from $-\pi/2$ to $\pi/2$.

With $\mathbf{R}$ thus defined, $\mathbf{R}\,\mathbf{e}_j$ is in the direction of η. We define the rotated eigenvector

$$\chi \;=\; \mathbf{R}^{-1}\xi_0. \tag{88}$$

Assuming $\mathbf{R}$ was defined to align η with $\mathbf{e}_j$, the next task is to define a shear transformation $\mathbf{S}$, with determinant 1, that maps $\mathbf{e}_j$ into the direction of χ. Let $\chi^{(j)}$ be the j-th component of χ. Since the j-th row of $\mathbf{R}^{-1}$ is $\eta^T/|\eta|$,

$$\chi^{(j)} \;=\; \eta \cdot \xi_0/|\eta|. \tag{89}$$

This must be nonzero because λ_0 has multiplicity 1. Then $\mathbf{S}$ is the matrix whose columns are $\mathbf{S}_0$, $\mathbf{S}_1$, $\mathbf{S}_2$, where

$$\mathbf{S}_j = \frac{\chi}{\chi^{(j)}} \qquad \mathbf{S}_{j+1} = \mathbf{e}_{j+1} \qquad \mathbf{S}_{j+2} = \mathbf{e}_{j+2} \tag{90}$$

Negating the off-diagonal elements in column j produces $\mathbf{S}^{-1}$. Algebraically, $(\mathbf{S}^{-1})_j = (2\,\mathbf{e}_j - \mathbf{S}_j)$.

The transforming matrix that separates the problem is $\mathbf{RS}$, and $\mathbf{A} \overset{\mathbf{RS}}{\rightsquigarrow}$ is the separated form of the matrix. It is unnecessary to compute $\mathbf{R}$ and $\mathbf{S}$ and multiply them. Using their definitions, we find

$$(\mathbf{RS})_j = \frac{|\eta|}{\eta \cdot \xi_0}\xi_0 \qquad (\mathbf{RS})_{j+1} = \mathbf{R}_{j+1} \qquad (\mathbf{RS})_{j+2} = \mathbf{R}_{j+2} \tag{91}$$

where $\mathbf{R}$ was defined by Eq. 87 and indexes are interpreted mod 3. The following equation for $(\mathbf{RS})^{-1}$ uses an outer product and can be verified by multiplying it by $(\mathbf{RS})$ on the right and using Eq. 138 in Appendix G:

$$(\mathbf{RS})^{-1} \quad = \quad (\mathbf{e}_j - \mathbf{S}_j)(\mathbf{R}_j)^T + \mathbf{R}^T. \tag{92}$$

See Example I.1 for a worked-out example.

Appendix B.2 Details for Two-Dimensional RJCF

This section shows how to transform a 2×2 matrix into a canonical coordinate system with a short sequence of familiar area-preserving transformations: rotations and nonuniform scales. The defective case, in which there is only one eigenvector, and nearly defective cases, in which the eigenvectors are nearly parallel, are treated. The transforming matrix and RJCF are continuous functions of the matrix elements in most of the 4-D space. We believe this is the first 2-D canonizing procedure with this degree of continuity.

We use the following notation for the elements of a 2×2 matrix $\mathbf{B}$:

$$\mathbf{B} \quad = \quad \begin{bmatrix} \mathsf{B}_{11} & \mathsf{B}_{12} \\ \mathsf{B}_{21} & \mathsf{B}_{22} \end{bmatrix} \tag{93}$$

Recall that B_{11} in this section corresponds to $\mathsf{B}_{j+1,j+1}$ in the 3-D context, etc. In a relaxed Jordan canonical coordinate system the matrix has one of a few special forms shown in Table 3. Thus the problem is, given a matrix $\mathbf{B}$, find a transformation $\mathbf{T}$ such that $\mathbf{B} \overset{\mathbf{T}}{\rightsquigarrow}$ is in one of these forms. In general, $\mathbf{T}$ is not unique.

For coordinate transformation $\mathbf{T}$, recall that the similarity transformation $\mathbf{T}^{-1}\mathbf{B}\mathbf{T}$ is denoted by $\mathbf{B} \overset{\mathbf{T}}{\rightsquigarrow}$. The transformation $\overset{\mathbf{T}}{\rightsquigarrow}$ preserves area if the determinant of $\mathbf{T}$ is 1. A canonical coordinate system can be derived with a short sequence of area-preserving transformations of two familiar types, two-dimensional rotations ($\mathbf{R}_\theta$) and nonuniform scales ($\mathbf{S}_w$):

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad \mathbf{R}_\theta^{-1} = \mathbf{R}_\theta^T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{94}$$

$$\mathbf{S}_w = \begin{bmatrix} w & 0 \\ 0 & 1/w \end{bmatrix} \qquad \mathbf{S}_w^{-1} = \begin{bmatrix} 1/w & 0 \\ 0 & w \end{bmatrix} \tag{95}$$

The geometric procedure to construct the transforming matrix $\mathbf{T}$ was described at a high level in Section 2.5. Also recall the invariants of similarity transformations discussed in Section 2.4. Recall that $\mathrm{trace}(\mathbf{B}) = \mathsf{B}_{11} + \mathsf{B}_{22}$ is invariant under any similarity transformation.

Let the *rotation matrix* $\mathbf{R}_\theta$ be given by Eq. 94. Then

$$\begin{aligned} \mathbf{B} \overset{\mathbf{R}_\theta}{\rightsquigarrow} & \begin{bmatrix} \mathsf{B}_{11}c^2 + \mathsf{B}_{22}s^2 + (\mathsf{B}_{12}+\mathsf{B}_{21})cs & \mathsf{B}_{12}c^2 - \mathsf{B}_{21}s^2 - (\mathsf{B}_{11}-\mathsf{B}_{22})cs \\ \mathsf{B}_{21}c^2 - \mathsf{B}_{12}s^2 - (\mathsf{B}_{11}-\mathsf{B}_{22})cs & \mathsf{B}_{11}s^2 + \mathsf{B}_{22}c^2 - (\mathsf{B}_{12}+\mathsf{B}_{21})cs \end{bmatrix} \\ = & \begin{bmatrix} \left(\frac{\mathsf{B}_{11}+\mathsf{B}_{22}}{2}\right) + \left(\frac{\mathsf{B}_{11}-\mathsf{B}_{22}}{2}\right)c_2 + \left(\frac{\mathsf{B}_{12}+\mathsf{B}_{21}}{2}\right)s_2 & \left(\frac{\mathsf{B}_{12}-\mathsf{B}_{21}}{2}\right) + \left(\frac{\mathsf{B}_{12}+\mathsf{B}_{21}}{2}\right)c_2 - \left(\frac{\mathsf{B}_{11}-\mathsf{B}_{22}}{2}\right)s_2 \\ -\left(\frac{\mathsf{B}_{12}-\mathsf{B}_{21}}{2}\right) + \left(\frac{\mathsf{B}_{12}+\mathsf{B}_{21}}{2}\right)c_2 - \left(\frac{\mathsf{B}_{11}-\mathsf{B}_{22}}{2}\right)s_2 & \left(\frac{\mathsf{B}_{11}+\mathsf{B}_{22}}{2}\right) - \left(\frac{\mathsf{B}_{11}-\mathsf{B}_{22}}{2}\right)c_2 - \left(\frac{\mathsf{B}_{12}+\mathsf{B}_{21}}{2}\right)s_2 \end{bmatrix}, \end{aligned} \tag{96}$$

where

$$c = \cos\theta \qquad s = \sin\theta \qquad c_2 = \cos 2\theta \qquad s_2 = \sin 2\theta. \tag{97}$$

The goal of the first rotation (by θ) is to equalize the diagonal elements, which are B_{11} and B_{22} in $\mathbf{B}$. If $\mathsf{B}_{11} = \mathsf{B}_{22}$ already, the equations give $\theta = 0$ (if $\mathsf{B}_{12} + \mathsf{B}_{21} \geq 0$) or $\theta = \pi/2$ (if $\mathsf{B}_{12} + \mathsf{B}_{21} < 0$). By Eq. 96 it is necessary and sufficient that $\tan 2\theta = -(\mathsf{B}_{11} - \mathsf{B}_{22})/(\mathsf{B}_{12} + \mathsf{B}_{21})$. There are always four solutions in the range $-\pi < \theta \leq \pi$. Consistent choices are needed to avoid unnecessary discontinuities.

With the notation of Eq. 97 and $\sqrt{\ }$ denoting the positive square root, let $\mathbf{B} \overset{\mathbf{R}}{\rightsquigarrow} \mathbf{M}$. We define these quantities:

$$D = (\mathsf{B}_{22} - \mathsf{B}_{11})^2 + (\mathsf{B}_{12} + \mathsf{B}_{21})^2 \tag{98}$$

$$\sigma = \begin{cases} 1 & \text{if } \mathsf{B}_{12} + \mathsf{B}_{21} > 0 \\ -1 & \text{if } \mathsf{B}_{12} + \mathsf{B}_{21} \le 0 \end{cases} \tag{99}$$

$$c = \sqrt{\tfrac{1}{2} + \frac{|\mathsf{B}_{12} + \mathsf{B}_{21}|}{2\sqrt{D}}} \tag{100}$$

$$s = -\operatorname{sgn}(\mathsf{B}_{11} - \mathsf{B}_{22})\, \sigma \sqrt{\tfrac{1}{2} - \frac{|\mathsf{B}_{12} + \mathsf{B}_{21}|}{2\sqrt{D}}} \tag{101}$$

$$\theta = -\operatorname{sgn}(\mathsf{B}_{11} - \mathsf{B}_{22})\, \sigma \tan^{-1}(|s/c|) \tag{102}$$

Here $\operatorname{sgn}(x)$ denotes the *sign* function, which is 1 for $x > 0$, 0 for $x = 0$, and -1 for $x < 0$. Thus $s = 0$ and $\theta = 0$ when $\mathsf{B}_{11} = \mathsf{B}_{22}$. Note that σ is never zero. The $\mathbf{R}$ defined by these values of c and s produces the following value for M:

$$\mathbf{M} = \begin{bmatrix} \frac{1}{2}(\mathsf{B}_{11} + \mathsf{B}_{22}) & \frac{1}{2}(\sigma\sqrt{D} + (\mathsf{B}_{12} - \mathsf{B}_{21})) \\ \frac{1}{2}(\sigma\sqrt{D} + (\mathsf{B}_{21} - \mathsf{B}_{12})) & \frac{1}{2}(\mathsf{B}_{11} + \mathsf{B}_{22}) \end{bmatrix} \tag{103}$$

Thus the diagonal elements have been equalized. The angle of rotation is between $-\pi/4$ and $\pi/4$.

The reason for choosing the signs of square roots as specified in Equations 100 and 101 is to ensure that $|\theta| \le \pi/2$. Whatever range is chosen, there will be some discontinuities in $\mathbf{M}$ and $\mathbf{R}$ as θ jumps from one end of its range to the other. However, almost all regions of the 2×2 matrix space have a locally continuous transformation to RJCF for some choice of signs for the square roots; in fact, there is normally only one *bad* choice. The exception is described in the next paragraph.

It is known that the eigenvector field is discontinuous at the one-dimensional subspace (line) $\mathsf{B}_{11} = \mathsf{B}_{22}$, $\mathsf{B}_{12} = 0$, $\mathsf{B}_{21} = 0$ [DH94]. This can also be seen from Eq. 98–102 by taking a sequence of $\mathbf{B}$'s for which $\mathsf{B}_{12} = \mathsf{B}_{21}$, $(\mathsf{B}_{11} + \mathsf{B}_{22}) = 4$, and $(\mathsf{B}_{11} - \mathsf{B}_{22})/(\mathsf{B}_{12} + \mathsf{B}_{21})$ alternates between 1 and 2 as both the numerator and denominator approach 0. Because the $\mathbf{B}$'s are symmetric, $\mathbf{S}_w = \mathbf{I}$ in all cases. Thus $\mathbf{T}$ alternates between distinct rotation matrices without converging, and its columns are the eigenvectors. Of course, $\mathbf{J}_B$ converges to $2\mathbf{I}$, but this is of little value in practice unless the transforming matrix also converges.

Let $\mathbf{S}_w$ be given by Eq. 95.

$$\mathbf{B} \overset{\mathbf{S}_w}{\rightsquigarrow} \begin{bmatrix} \mathsf{B}_{11} & \mathsf{B}_{12}/w^2 \\ \mathsf{B}_{21}w^2 & \mathsf{B}_{22} \end{bmatrix} \tag{104}$$

The goal of the next step is to choose the nonuniform scale $\mathbf{S}_w$ that equalizes the magnitudes of the off-diagonal elements of $\mathbf{M}$. By Eq. 104

$$\begin{bmatrix} \mathsf{M}_{11} & \mathsf{M}_{12} \\ \mathsf{M}_{21} & \mathsf{M}_{22} \end{bmatrix} \overset{\mathbf{S}_w}{\rightsquigarrow} \begin{bmatrix} \mathsf{M}_{11} & \mathsf{M}_{12}/w^2 \\ \mathsf{M}_{21}w^2 & \mathsf{M}_{22} \end{bmatrix} \tag{105}$$

so $w = \sqrt[4]{|\mathsf{M}_{12}/\mathsf{M}_{21}|}$ is the desired scale factor, provided that neither M_{12} nor M_{21} is zero or "too close to zero." This is discussed further in Section 2.6. See Example I.4 for a worked-out example.

Appendix C Three Equal Eigenvalues

There are three distinct topological categories of transformation when all three eigenvalues are equal to λ. They are distinguished by the *minimal polynomial* of $\mathbf{A}$ [Gan59]. The characteristic polynomial is $f(x) = (x - \lambda)^3$ in all three cases, but the degree of minimal polynomial ranges from 1 to 3.

Definition C.1: Let $\mathbf{A}$ have three equal eigenvalues of λ. The *minimal polynomial* of $\mathbf{A}$ is $(x-\lambda)^d$, where d is the minimum value such that

$$(\mathbf{A}-\lambda\,\mathbf{I})^d \;=\; \mathbf{0} \tag{106}$$

The *geometric multiplicity* of λ is the dimension of the subspace spanned by eigenvectors of λ, and is known to equal $4-d$. The *algebraic multiplicity* of λ is its multiplicity as a root of the characteristic polynomial, which is 3 by hypothesis. □

Note that the case $d=1$ occurs if and only if $\mathbf{A}=\lambda\,\mathbf{I}$. The other two cases are not so easy to identify.

Because λ has algebraic multiplicity three, the subspace not spanned by eigenvectors is spanned by *generalized eigenvectors* [BD96]. A vector η is a generalized eigenvector[1] for λ if there is *some* ξ that is an eigenvector for λ and *some* positive integer k such that

$$\begin{aligned} \eta^T\xi &= 0, \\ (\mathbf{A}-\lambda\,\mathbf{I})^k\eta &= \xi. \end{aligned} \tag{107}$$

The *generalized eigenspace* of an eigenvalue λ_1 is the space spanned by its eigenvectors and generalized eigenvectors together, and its dimension equals the algebraic multiplicity of λ_1 [HS74]. In the case that all eigenvalues are equal, the generalized eigenspace is the entire space.

For the subsequent discussion, define:

$$\mathbf{M}_1 \;=\; \mathbf{A}-\lambda\,\mathbf{I} \qquad\qquad \mathbf{M}_2 \;=\; (\mathbf{A}-\lambda\,\mathbf{I})^2 \tag{108}$$

We shall see that one of these matrices has rank one.

Appendix C.1 Geometric Multiplicity 2

Suppose the eigenvalue of $\mathbf{A}$ is λ and it has geometric multiplicity 2; i.e., $d=2$ in Definition C.1. This means that $\mathbf{M}_1\neq\mathbf{0}$, but $\mathbf{M}_2=\mathbf{0}$. It follows that $\mathbf{M}_1$ has rank one. For numerical purposes suppose that $\gamma=(\mathbf{M}_1)_{ij}$ is an element of $\mathbf{M}_1$ that has maximum magnitude (γ may be negative). By Eq. 7,

$$\mathbf{M}_1 \;=\; \frac{\xi_1\,\eta^T}{\gamma} \tag{109}$$

where η^T is row i of $\mathbf{M}_1$ and ξ_1 is column j of $\mathbf{M}_1$.

Any vector orthogonal to a nonzero row of $\mathbf{M}_1$ is an eigenvector, confirming that the eigenvectors span a two-dimensional subspace. We use η as the normal vector for this subspace.

We now claim that η is a generalized eigenvector, as defined in Eq. 107. To establish this claim, note that any nonzero *column* of $\mathbf{M}_1$ (in particular, ξ_1) is an eigenvector for λ. That is, suppose ξ_1 were not an eigenvector. Then the standard basis vector $\mathbf{e}_j$ would not be mapped to 0 by $(\mathbf{A}-\lambda\,\mathbf{I})^2$, contradicting the assumption that $\mathbf{M}_2=\mathbf{0}$.

It follows from Eq. 109 that

$$(\mathbf{A}-\lambda\,\mathbf{I})\eta \;=\; \left(\frac{|\eta|^2}{\gamma}\right)\xi_1. \tag{110}$$

Since the right-hand side of Eq. 110 is an eigenvector for λ, the claim that η is a generalized eigenvector is established.

The canonical coordinate system is obtained by defining $\xi_0=\xi_1\times\eta$ to form a right-handed system of basis vectors. Then the transforming matrix $\mathbf{T}$ is defined by

$$\mathbf{T}_0=\frac{\xi_0}{|\xi_0|} \qquad\qquad \mathbf{T}_1=\frac{\xi_1}{|\xi_1|} \qquad\qquad \mathbf{T}_2=\frac{\eta}{|\eta|} \tag{111}$$

which is a rotation matrix. In fact, ξ_0 can be treated as the eigenvector of an outer root, separating the problem into a one-dimensional subspace spanned by ξ_0 and a two-dimensional subspace spanned by ξ_1 and η.

[1] Actually, our definition is slightly "relaxed" compared to the standard, to take advantage of our relaxed Jordan forms.

$\begin{bmatrix} 3 & 0 & -1 \\ 0 & 3 & 0 \\ 0 & 4 & 3 \end{bmatrix}$
Compound Shear

Table 5: The RJCF for a Compound Shear is a triangular matrix with a corner element, according to Definition C.2, and with three diagonal elements equal.

Example C.1: Let

$$\mathbf{A} \;=\; \begin{bmatrix} 3 & 5 & -3 \\ 1 & 7 & -3 \\ 2 & 10 & -4 \end{bmatrix} \tag{112}$$

All three eigenvalues are 2.

$$\mathbf{M}_1 \;=\; (\mathbf{A}-2\mathbf{I}) \;=\; \begin{bmatrix} 1 & 5 & -3 \\ 1 & 5 & -3 \\ 2 & 10 & -6 \end{bmatrix} \tag{113}$$

The matrix $\mathbf{M}_1 \neq \mathbf{0}$, but $\mathbf{M}_1^2 = \mathbf{0}$, so the minimal polynomial is $(x-2)^2$. Therefore the eigenvalue 2 has geometric multiplicity 2. The eigenvectors span a two-dimensional subspace that is orthogonal to $\boldsymbol{\eta} = [2, 10, -6]^T$. In particular, $\boldsymbol{\xi}_1 = [5, 5, 10]^T$ is an eigenvector and

$$(\mathbf{A}-2\mathbf{I})\,\boldsymbol{\eta} \;=\; 14\boldsymbol{\xi}_1, \tag{114}$$

confirming that $\boldsymbol{\eta}$ is a generalized eigenvector. Define

$$\boldsymbol{\xi}_0 \;=\; \boldsymbol{\xi}_1 \times \boldsymbol{\eta} \;=\; [-130, 50, 40]^T \tag{115}$$

Then by Eq. 111 the transforming matrix is

$$\mathbf{T} \;=\; \begin{bmatrix} -13/\sqrt{210} & 1/\sqrt{6} & 1/\sqrt{35} \\ 5/\sqrt{210} & 1/\sqrt{6} & 5/\sqrt{35} \\ 4/\sqrt{210} & 2/\sqrt{6} & -3/\sqrt{35} \end{bmatrix} \qquad \mathbf{T}^{-1} \;=\; \mathbf{T}^T$$

Then

$$\mathbf{J} \;=\; \mathbf{T}^{-1}\mathbf{A}\mathbf{T} \;=\; \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & \sqrt{210} \\ 0 & 0 & 2 \end{bmatrix} \tag{116}$$

□

Appendix C.2 Compound Shear Relaxed Jordan Canonical Form

Finally, suppose the eigenvalue of $\mathbf{A}$ is λ and it has geometric multiplicity 1; i.e., $d = 3$ in Definition C.1. Some terminology helps to specify Compound Shear RJCFs. Indices *i*, *j*, and *k* are understood to be in the range 0, 1, 2.

Informally, a 3×3 matrix is *triangular with a corner element* if some permutation of the indices puts it into upper triangular form (i.e., all entries below the main diagonal are zero), and after being put into upper triangular form the $(0,1)$ and $(1,2)$ elements are both nonzero. The $(0,2)$ element may or may not be zero. See Table 5. However, we do not want to try permutations to determine if this property holds, so a more technical definition is given.

Definition C.2: A matrix is called *triangular* if for every non-diagonal nonzero element, say A_{ij}, either all non-diagonal elements in column i are zero or all non-diagonal elements in row j are zero, or both. (Note that A_{ij} is in row i and column j, but we are talking about the row and column of the (j,i) element.)

Element A_{ij} is called a *corner element* of $\mathbf{A}$ if $\mathbf{A}$ is triangular, $\mathsf{A}_{ji} = 0$, $\mathsf{A}_{ik} \neq 0$ and $\mathsf{A}_{kj} \neq 0$, where i, j, and k are distinct indices. □

The corner element itself may be zero or nonzero, but the other non-diagonal elements in its row and column must be nonzero. A corner element might not exist for a triangular matrix, but it is unique if it does exist.

Lemma C.1: Let $\mathbf{A}$ be a 3×3 matrix.

1. A 3×3 matrix $\mathbf{A}$ is *triangular* if and only if there is some permutation of the indices that puts it into upper triangular form, say $\mathbf{B}$. In this case, $\mathbf{A}$ has a corner element if and only if B_{02} is a corner element, i.e., $\mathsf{B}_{01} \neq 0$ and $\mathsf{B}_{12} \neq 0$.
2. $\mathbf{A}$ is *triangular* if and only if there is some permutation of the indices that puts it into lower triangular form, say $\mathbf{C}$. In this case, $\mathbf{A}$ has a corner element if and only if C_{20} is a corner element.

Proof: For part (1) enumerate the six index permutations that map $\mathbf{B}$ into various possibilities for $\mathbf{A}$ and apply the definitions in each case. Part (2) is similar, using $\mathbf{C}$. ∎

If $\mathbf{A}$ is triangular, then for any distinct indices i, j, at least one of A_{ij} and A_{ji} is zero. However, the converse is not true; see Example C.2, following.

Definition C.3: A matrix $\mathbf{J}$ is a *compound-shear* relaxed Jordan canonical form if it a triangular matrix with a corner element and all three diagonal elements are equal. □

Example C.2: Several matrices are shown below.

$$\mathbf{A}_1 = \begin{bmatrix} 2 & 1.5 & 0.5 \\ 0 & 2 & 0 \\ 0 & -1 & 2 \end{bmatrix} \qquad \mathbf{A}_2 = \begin{bmatrix} 2 & 0 & 0.5 \\ 0 & 2 & 0 \\ 0 & -1 & 2 \end{bmatrix} \qquad \mathbf{A}_3 = \begin{bmatrix} 2 & 1.5 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 2 \end{bmatrix} \qquad \mathbf{A}_4 = \begin{bmatrix} 2 & 0 & 0.5 \\ 1.5 & 2 & 0 \\ 0 & -1 & 2 \end{bmatrix}$$

Matrix $\mathbf{A}_1$ is triangular with corner element $(1,2)$, as is $\mathbf{A}_2$. Since all the diagonal elements are equal, both are Compound Shear RJCFs. Although $\mathbf{A}_3$ is triangular and all the diagonal elements are equal, it has no corner element, so it is *not* in RJCF. Its RJCF takes the Simple Shear form. Finally, $\mathbf{A}_4$ is *not* triangular, although for all distinct (i,j) either $\mathsf{A}_{ij} = 0$ or $\mathsf{A}_{ji} = 0$. Its RJCF takes the Spiral form. □

Suppose the only eigenvalue of $\mathbf{A}$ is λ and it has geometric multiplicity 1; i.e., $d = 3$ in Definition C.1. Define $q(x) = (x-\lambda)^2$. Recall the notation of Eq. 108. As in the case where there is a distinct outer root, $\mathbf{M}_1$ has rank two. Therefore, $\mathbf{M}_2 = q(\mathbf{A})$ has rank one. For numerical purposes suppose that $\gamma = (\mathsf{M}_2)_{ij}$ is an element of $\mathbf{M}_2$ that has maximum magnitude (γ may be negative). By Eq. 7,

$$\mathbf{M}_2 = \frac{\xi_0 \eta_2^T}{\gamma} \tag{117}$$

where η_2^T is row i of $\mathbf{M}_2$ and ξ_0 is column j of $\mathbf{M}_2$.

Any nonzero column of $\mathbf{M}_2$ is an eigenvector (or else $f(\mathbf{A})$ would be nonzero, violating Eq. 11). We use ξ_0 as the eigenvector.

The generalized eigenvectors must satisfy

$$\begin{aligned} \eta_1^T \xi &= 0, \\ \eta_2^T \eta_1 = \eta_2^T \xi &= 0, \\ (\mathbf{A} - \lambda \mathbf{I}) \eta_1 &= \alpha \xi_0 \qquad (118) \\ (\mathbf{A} - \lambda \mathbf{I})^2 \eta_2 &= \beta \xi_0 \qquad (119) \end{aligned}$$

where α and β are nonzero scalars [BD96]. By arguments similar to previous cases (see Section C.1), the η_2 defined by Eq. 117 satisfies Eq. 119 with $\beta = |\eta_2|^2/\gamma$. Finally, we define $\eta_1 = \eta_2 \times \xi_0$. We need to show that η_1 satisfies Eq. 118.

First, we show that η_2 is orthogonal to ξ_0, so that ξ_0, η_1, η_2 form a right-handed orthogonal coordinate system. Recalling Eq. 117, we have $\mathbf{M}_2 = \xi_0 \eta_2^T/\gamma$. Note that $(\mathbf{M}_1)^3 = \mathbf{0}$ and $\mathbf{M}_1\xi_0 = 0$. Thus

$$\frac{\xi_0}{\gamma}\left(\eta_2^T \xi_0\right) \;=\; \mathbf{M}_2\xi_0 \;=\; \mathbf{M}_1(\mathbf{M}_1\xi_0) \;=\; 0$$

Since $\xi_0/\gamma \neq 0$, the claim is proved.

Similarly, η_2 is orthogonal to $\mathbf{M}_1\eta_2$ because

$$\frac{\xi_0}{\gamma}\left(\eta_2^T \mathbf{M}_1\eta_2\right) \;=\; \mathbf{M}_1^3\eta_2 \;=\; 0$$

Now we are prepared to argue that η_1 satisfies Eq. 118. Since $\beta \neq 0$ in Eq. 119, $\mathbf{M}_1\eta_2$ is not collinear with ξ_0. We showed that ξ_0 and $\mathbf{M}_1\eta_2$ are orthogonal to η_2, and η_1 is orthogonal to η_2 by construction. Therefore, η_1 is some linear combination of $\mathbf{M}_1\eta_2$ and ξ_0. It follows that $\mathbf{M}_1\eta_1$ satisfies Eq. 118 for some scalar α. The transforming matrix is the rotation matrix:

$$\mathbf{T}_j = \frac{\xi_0}{|\xi_0|} \qquad \mathbf{T}_{j+1} = \frac{\eta_1}{|\eta_1|} \qquad \mathbf{T}_{j+2} = \frac{\eta_2}{|\eta_2|} \tag{120}$$

where indexes are interpreted mod 3, as usual.

Example C.3: Let

$$\mathbf{A} \;=\; \begin{bmatrix} -3 & 2 & -2 \\ -8 & 4 & -3 \\ 15 & -5 & 8 \end{bmatrix} \tag{121}$$

All three eigenvalues are 3.

$$\mathbf{M}_1 \;=\; (\mathbf{A}-3\mathbf{I}) \;=\; \begin{bmatrix} -6 & 2 & -2 \\ -8 & 1 & -3 \\ 15 & -5 & 5 \end{bmatrix} \qquad \mathbf{M}_2 \;=\; (\mathbf{A}-3\mathbf{I})^2 \;=\; \begin{bmatrix} -10 & 0 & -4 \\ -5 & 0 & -2 \\ 25 & 0 & 10 \end{bmatrix} \tag{122}$$

The matrix $(\mathbf{A}-3\mathbf{I})^2 \neq \mathbf{0}$, so the minimal polynomial is the characteristic polynomial, $(x-3)^3$. Therefore the eigenvalue 3 has geometric multiplicity 1. The eigenvectors span a one-dimensional subspace containing $[-10,-5,25]^T$.

The row of maximum magnitude in $\mathbf{M}_2$ provides one generalized eigenvector, $\eta_2^T = [25,0,10]$. Note that $\gamma = 25$. Then $\eta_1 = \eta_2 \times \xi_0 = [-50,-725,-125]^T$ is an independent generalized eigenvector.

To complete the computation of $\mathbf{T}$, with $j = 0$, we get the exact and approximate expressions:

$$\mathbf{T}_0 \;=\; \frac{1}{\sqrt{30}}\begin{bmatrix} -2 \\ -1 \\ 5 \end{bmatrix} \quad \mathbf{T}_1 = \frac{1}{\sqrt{870}}\begin{bmatrix} 2 \\ -29 \\ -5 \end{bmatrix} \quad \mathbf{T}_2 = \frac{1}{\sqrt{29}}\begin{bmatrix} 5 \\ 0 \\ 2 \end{bmatrix}$$

$$\mathbf{T} \;\approx\; \begin{bmatrix} -0.365 & 0.068 & 0.928 \\ -0.183 & -0.983 & 0 \\ 0.913 & 0.170 & 0.371 \end{bmatrix} \tag{123}$$

The relaxed Jordan canonical form is

$$\mathbf{J} \;=\; \mathbf{T}^{-1}\mathbf{A}\mathbf{T} \;=\; \begin{bmatrix} 3 & 30/\sqrt{29} & 539/\sqrt{870} \\ 0 & 3 & 29/\sqrt{30} \\ 0 & 0 & 3 \end{bmatrix} \;\approx\; \begin{bmatrix} 3 & 5.571 & 18.274 \\ 0 & 3 & 5.295 \\ 0 & 0 & 3 \end{bmatrix} \tag{124}$$

The traditional Jordan canonical form can be obtained by a further similarity transformation, but this serves no purpose, computationally. □

Compound Shear Case:

For completeness, we address the case in which the final relaxed Jordan canonical form does not separate, but remains as a 3-D problem. This case is a 3-D compound shear in which only one line is invariant; it occurs rarely. Upper-triangular matrices are used for diagramming simplicity; other configurations can be achieved by applying the same permutation to both columns and rows. In general, $\mathbf{A}$ and $\mathbf{B}$ can be any triangular matrices and c and τ are their corner elements, as defined in Definition C.2. Then other nonzero off-diagonal elements are identified by sharing a row or a column with the corner element.

In this family of RJCFs, the three diagonal entries, a, must be equal (and might be zero); the super-diagonal entries, b and d, must be nonzero; but the corner entry, c, might be zero. If a, b, c, and d meet these conditions, and

$$\mathbf{B} = \begin{bmatrix} a & b & c \\ 0 & a & d \\ 0 & 0 & a \end{bmatrix} \quad \text{then} \quad \mathbf{A} = e^{\mathbf{B}} = \begin{bmatrix} e^a & e^a b & e^a(c+bd/2) \\ 0 & e^a & e^a d \\ 0 & 0 & e^a \end{bmatrix} \tag{125}$$

This form is proved in Lemma D.5.

Working back for 3-D compound shears, if $\lambda > 0$, τ is arbitrary, μ and ρ are nonzero, and

$$\mathbf{A} = \begin{bmatrix} \lambda & \mu & \tau \\ 0 & \lambda & \rho \\ 0 & 0 & \lambda \end{bmatrix} \text{ then } \mathbf{B} = \log \mathbf{A} = \begin{bmatrix} \log\lambda & \mu/\lambda & (\tau - \mu\rho/2)/\lambda \\ 0 & \log\lambda & \rho/\lambda \\ 0 & 0 & \log\lambda \end{bmatrix} \tag{126}$$

The computations are complicated by the requirement to know the location of the corner element. However, the code can be simplified by recognizing that the same formula works for all off-diagonal locations. The input matrix, whether $\mathbf{A}$ or $\mathbf{B}$ must be a compound-shear RJCF. As above, let $a = \mathbf{B}_{ii}$ and $\lambda = \mathbf{A}_{ii} > 0$. Let i, j and k be distinct indices.

$$\begin{aligned} \mathbf{A}_{ij} &= (\mathbf{B}_{ij} + \mathbf{B}_{ik} * \mathbf{B}_{kj}/2)e^a \\ \mathbf{B}_{ij} &= (\mathbf{A}_{ij} - \mathbf{A}_{ik} * \mathbf{A}_{kj}/2)/\lambda \end{aligned}$$

This works because the terms with "$*$" are zero, except for the corner element.

As degenerate as this family is — there is only one eigenvector — the matrix $\mathbf{T}$ to transform to and from the RJCF coordinate system is a pure rotation matrix (see Eq. 120).

Appendix D Derivations for Matrix Logarithms

The inverse of the matrix exponential operator is the matrix logarithm: $\mathbf{B} = \log \mathbf{A}$ if $e^{\mathbf{B}} = \mathbf{A}$. Such a $\mathbf{B}$ does not always exist. Alexa uses the matrix logarithm in interpolation [Ale02] and provides complicated procedures to compute it. This section derives the closed forms presented in Section 4. We also discuss the use of the matrix logarithm when $\mathbf{A}$ is supposed to be a rotation matrix, but has numerical errors.

Suppose $\mathbf{A} = e^{\mathbf{B}}$. The key observation is that the eigenvectors and generalized eigenvectors (see Eq. 107 in Appendix C) of $\mathbf{B}$ are also those for $\mathbf{A}$; i.e., they are preserved by the exponential operator. Therefore, by Eq. 37, it suffices to consider $\mathbf{A}$ and $\mathbf{B}$ in relaxed Jordan canonical form when considering how to compute the matrix logarithm. We will see that if $\mathbf{B}$ is in relaxed Jordan canonical form (Section 2), then so is $\mathbf{A}$. The problem is further simplified by Eq. 38: In most cases the problem separates into a 1-D problem and a 2-D problem. In the following analysis $\mathbf{A}$ and $\mathbf{B}$ might refer to 1×1 or 2×2 matrices, due to this separation.

Because the results in this section rely heavily on the fact that eigenvectors and generalized eigenvectors are preserved by the matrix exponential operator, we prove this claim.

Lemma D.1: Consider the following conditions:

(A) ξ is an eigenvector of $\mathbf{B}$ for λ, i.e., $\mathbf{B}\xi = \lambda\xi$.

(B) η is not parallel to ξ and satisfies $\mathbf{B}\eta = \lambda\eta + \alpha\xi$.

(C) ρ is not parallel to ξ or η and satisfies $\mathbf{B}\rho = \lambda\rho + \beta\eta + \gamma\xi$.

1. If (A) holds, then ξ is an eigenvector of $e^{\mathbf{B}}$ for e^{λ}.
2. If (A) and (B) hold, then η is a generalized eigenvector of $e^{\mathbf{B}}$ for e^{λ}.
3. If (A) and (B) and (C) hold, then ρ is a generalized eigenvector of $e^{\mathbf{B}}$ for e^{λ}.

Proof: For part (1)

$$e^{\mathbf{B}}\xi \;=\; \sum_{k=0}^{\infty}\mathbf{B}^k\xi/k! \;=\; \left(\sum_{k=0}^{\infty}\lambda^k/k!\right)\xi \;=\; e^{\lambda}\xi;$$

For part (2)

$$e^{\mathbf{B}}\eta \;=\; \sum_{k=0}^{\infty}\mathbf{B}^k\eta/k! \;=\; \sum_{k=0}^{\infty}\left(\lambda^k\eta + k\lambda^{k-1}\alpha\xi\right)/k! \;=\; e^{\lambda}\eta + \alpha e^{\lambda}\xi;$$

For part (3)

$$\begin{aligned} e^{\mathbf{B}}\rho \;&=\; \sum_{k=0}^{\infty}\mathbf{B}^k\rho/k! \\ &=\; \sum_{k=0}^{\infty}\left(\lambda^k\rho + k\lambda^{k-1}(\beta\eta+\gamma\xi) + \tfrac{1}{2}k(k-1)\alpha\beta\xi\right)/k! \\ &=\; e^{\lambda}\rho + \beta e^{\lambda}\eta + (\gamma+\tfrac{1}{2}\alpha\beta)e^{\lambda}\xi. \end{aligned}$$

∎

Appendix D.1 Exponentials of Common Cases

We now derive the common exponential closed forms needed for RJCF.

Lemma D.2: [HS74] If **B** is a diagonal matrix (which includes the 1-D case), then **A** is the positive diagonal matrix $\mathsf{A}_{ii} = e^{\mathsf{B}_{ii}}$.

Proof: An easy induction shows that $(\mathbf{B}^k)_{ii} = \mathsf{B}_{ii}^k$ for all integers $k \geq 0$. Substitute into Eq. 36. ∎

Lemma D.3: If $\mathbf{B} = \begin{bmatrix} a & b \\ 0 & a \end{bmatrix}$, where a is arbitrary and b is nonzero, then

$$\mathbf{A} \;=\; e^{\mathbf{B}} \;=\; e^{a}\begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix}. \tag{127}$$

Proof: An easy induction shows that

$$\mathbf{B}^k \;=\; \begin{bmatrix} a^k & b\,k\,a^{k-1} \\ 0 & a^k \end{bmatrix} \tag{128}$$

for all integers $k \geq 0$. Substitute into Eq. 36. ∎

Lemma D.4: [HS74] If $\mathbf{B} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}$, where a is arbitrary and b is nonzero, then

$$\mathbf{A} \;=\; e^{\mathbf{B}} \;=\; e^{a}\begin{bmatrix} \cos b & -\sin b \\ \sin b & \cos b \end{bmatrix} \tag{129}$$

Proof: (This equation is well known in differential equation theory [HS74, pp. 84–85], but a direct proof is hard to find.) Define the complex numbers $z = a + ib$, $\tilde{z} = a - ib$, so that $a = \frac{1}{2}(z+\tilde{z})$ and $b = -\frac{1}{2}i(z-\tilde{z})$. Then an easy induction shows that

$$\mathbf{B}^k \;=\; \tfrac{1}{2}\begin{bmatrix} z^k + \tilde{z}^k & i(z^k - \tilde{z}^k) \\ -i(z^k - \tilde{z}^k) & z^k + \tilde{z}^k \end{bmatrix} \tag{130}$$

for all integers $k \geq 0$. Substitute into Eq. 36 and use the identity $e^{ib} = \cos b + i\sin b$. ∎

Appendix D.2 Exponential of Compound Shear Matrix

Next, we address the case in which the final relaxed Jordan canonical form does not separate, but remains as a 3-D problem. This case is a compound shear in which only one line is invariant; it occurs rarely.

Lemma D.5: Let

$$\mathbf{B} \;=\; \begin{bmatrix} a & b & c \\ 0 & a & d \\ 0 & 0 & a \end{bmatrix}, \tag{131}$$

where a and c are arbitrary, b and d are nonzero. Then

$$\mathbf{A} \;=\; e^{\mathbf{B}} \;=\; e^{a} \begin{bmatrix} 1 & b & (c+bd/2) \\ 0 & 1 & d \\ 0 & 0 & 1 \end{bmatrix}. \tag{132}$$

Proof: A straightforward induction shows that

$$\mathbf{B}^k \;=\; \begin{bmatrix} a^k & b\,k\,a^{k-1} & c\,k\,a^{k-1} + \frac{1}{2}\,b\,d\,\binom{k}{2}\,a^{k-2} \\ 0 & a^k & d\,k\,a^{k-1} \\ 0 & 0 & a^k \end{bmatrix} \tag{133}$$

for all integers $k \geq 0$. Substitute into Eq. 36. ∎

Appendix D.3 Logarithm of a Singular Matrix

A technical argument follows to justify that we can simply change the off-diagonal elements in the row and column with $-\infty$ to zero in a matrix $\mathbf{L}$ that represents a logarithm. To make sense of this situation, we imagine that $(\mathbf{L})_{11} = -M$ for some large $M > 0$ and consider the limiting RJCF as M approaches ∞. The outer eigenvalue converges to $-M$ and the outer eigenvector approaches $[0,1,0]^T$. After separating the 2-D invariant subspace, the matrix must have zeros in off-diagonal entries in the row and column containing the outer eigenvalue, as in Eq. 16. Working through the equations of Appendix B.1, the normal to the 2-D invariant subspace is $\boldsymbol{\eta} \approx [-0.9679/M, 1, 0.0291/M]^T$, while the outer eigenvector is $\boldsymbol{\xi}_0 \approx [0.3798/M, 1, -0.6679]^T$. The transforming matrix and separated matrix are, ignoring $O(1/M^2)$ terms:

$$\mathbf{T} \;=\; \begin{bmatrix} 1.0000 & 0.3798/M & 0.0000 \\ 0.9679/M & 1.0000 & -0.0291/M \\ 0.0000 & -0.6679/M & 1.0000 \end{bmatrix}. \qquad \mathbf{T}^{-1}\mathbf{L}\mathbf{T} \;=\; \begin{bmatrix} 0.2347 & 0 & 0.3052 \\ 0 & -M & 0 \\ -0.4149 & 0 & 0.7003 \end{bmatrix}.$$

This matrix can be exponentiated by dealing with rows and columns 0 and 2 separately from row and column 1, as described in Section 4. As $M \to \infty$, $\mathbf{T} \to \mathbf{I}$, and the resulting $\mathbf{A}$ has rank 2.

It is necessary to put the 2×2 submatrix into RJCF using Eq. 98–105 in Section B.2.

$$\mathbf{L} \;=\; \mathbf{U}\mathbf{H}\mathbf{U}^{-1} \;=\; \begin{bmatrix} 0.9263 & 0 & -0.5254 \\ 0 & 1 & 0 \\ 1.1699 & 0 & 0.4160 \end{bmatrix} \begin{bmatrix} 0.4675 & 0 & 0.2691 \\ 0 & -\infty & 0 \\ -0.2691 & 0 & 0.4675 \end{bmatrix} \begin{bmatrix} 0.4160 & 0 & 0.5254 \\ 0 & 1 & 0 \\ -1.1699 & 0 & 0.9263 \end{bmatrix}.$$

The middle matrix $\mathbf{H}$ is exponentiated as follows:

$$e^{\mathbf{H}} \;=\; \exp\begin{pmatrix} 0.4675 & 0 & 0.2691 \\ 0 & -\infty & 0 \\ -0.2691 & 0 & 0.4675 \end{pmatrix} \;=\; e^{0.4675} \begin{bmatrix} \cos(-0.2691) & 0 & \sin(0.2691) \\ 0 & 0 & 0 \\ -\sin(0.2691) & 0 & \cos(-0.2691) \end{bmatrix}$$

$$\;=\; \begin{bmatrix} 1.5386 & 0 & 0.4243 \\ 0 & 0 & 0 \\ -0.4243 & 0 & 1.5386 \end{bmatrix}$$

The final blend, called $\mathbf{A}^*$, is:

$$\begin{aligned}\mathbf{A}^* \;=\; 0.1\odot\mathbf{B}\oplus 0.9\odot\mathbf{C} \;&=\; \mathbf{V}\mathbf{A}\mathbf{V}^{-1} \;=\; \mathbf{V}\mathbf{U}e^{\mathbf{H}}\mathbf{U}^{-1}\mathbf{V}^{-1}\\ &=\; \begin{bmatrix} 1.0343 & -0.4889 & 0.6519\\ 0.6508 & 0.5789 & -0.7719\\ -0.1782 & -1.0980 & 1.4640\end{bmatrix}.\end{aligned}$$

It is noteworthy that although $\mathbf{A}^*$ supposed has only 0.1 of $\mathbf{B}$, it is closer to $\mathbf{B}$ than to $\mathbf{C}$, as measured by Frobenius norms, $|\mathbf{B}-\mathbf{A}^*|^2 = 2.130$ and $|\mathbf{C}-\mathbf{A}^*|^2 = 3.035$.

Appendix D.4 Analysis of Approximate Rotation Matrices (Numerical Issues)

If some 3-D rotation matrix $\mathbf{R}$ is similarity-transformed into its RJCF, say $\mathbf{J} = \mathbf{R}\overset{\mathbf{T}}{\rightsquigarrow}$, then the real eigenvector of $\mathbf{R}$, say $\mathbf{u}$, appears as the column of $\mathbf{T}$ corresponding to the eigenvalue 1. When scaled to unit length, it is the axis of rotation. That is, we have something like

$$\mathbf{J} \;=\; \begin{bmatrix} 1 & 0\\ 0 & \mathbf{A}\end{bmatrix} \tag{134}$$

where $\mathbf{A}$ is a 2-D rotation matrix. If $\mathbf{A}$ is a 2-D rotation matrix, we see in Eq. 41 that $(\log\mathbf{A})_{00} = 0$ and $(\log\mathbf{A})_{10} = \theta$, the angle of rotation.

If $\mathbf{R}$ is *expected* to be a rotation matrix, but might have small numerical errors, a robust procedure to find the axis of rotation and the angle of rotation is to find the outer eigenvector, $\mathbf{u}$, then transform into RJCF, which should look like $\mathbf{J}$ in Eq. 134, except that J_{00} might not be exactly 1. Take the matrix logarithm of $\mathbf{A}$. Use $(\log\mathbf{A})_{10}$ as the angle of rotation θ, and force $(\log\mathbf{A})_{00}$ and $(\log\mathbf{A})_{11}$ which should be quite small if $\mathbf{R}$ is almost a rotation matrix, to be 0. The outer eigenvector $\mathbf{u}$, scaled to unit-length, can be used as the axis of rotation. The value of θ depends only on the ratio of $\mathbf{A}_{10}$ and $\mathbf{A}_{00}$, so it is not sensitive to errors in magnitude of rows or columns. As exponential coordinates for $\mathbf{R}$, simply use $\theta\,\mathtt{normv}(\mathbf{u})$ (see Appendix J); for $\log(\mathbf{R})$ use $\chi(\theta\,\mathtt{normv}(\mathbf{u}))$. Adjusting the RJCF back into SO(3) provides a logical way to "tune up" a rotation matrix that has accumulated errors through a sequence of matrix multiplications. We performed one numerical test on a matrix that originally represented a rotation of about 178.6°, but had been forced out of SO(3) by rounding to two significant digits. The RJCF (adjusted as just described) reduced the error in the angle of rotation by a factor of 10, compared to Eq. 67, from 1.2° to 0.125°.

Another approach to tuning a rotation matrix that is slightly out of SO(3) is to perform the polar decomposition (see Section 6) and use the orthogonal part of this decomposition as the corrected matrix. Higham proposes this solution and gives an iterative procedure that converges quadratically when the errors are small [Hig86].

Appendix E Great-Circle Interpolation on the n-D Sphere

This section describes a numerically robust procedure for great-circle interpolation on the n-D unit sphere between two n-vectors $\mathbf{u}$ and $\mathbf{v}$. This permits interpolation of quaternions when $n = 4$, popularized by Shoemake with the `slerp` function [Sho85]. Another benefit is that the relationship between movement on the great circle and rotation in 3-D is practical to verify. A problem with published methods is that they lose accuracy when the angle θ between $\mathbf{u}$ and $\mathbf{v}$ is near 0 or near 180°; in particular, the usual interpolation formula has $\sin\theta$ in the denominator and nearly equal quantities are subtracted in the numerator.

In n-D a *great circle* may be defined as the intersection of a 2-D linear subspace with the unit n-D sphere. A basis for a 2-D linear subspace consists of two linearly independent vectors. If the basis is orthonormal, generating points on the great circle is very simple. Our procedure is based on the observation that if $\mathbf{u}$ and $\mathbf{v}$ are two unit-length n-vectors that determine the great circle, then the pair $(\mathbf{v}+\mathbf{u})$, $(\mathbf{v}-\mathbf{u})$ is an orthogonal (but not unit-length) basis for the same 2-D subspace. There may be numerical inaccuracies if either $(\mathbf{v}+\mathbf{u})$ or $(\mathbf{v}-\mathbf{u})$ is very small, but making the computation more complicated will not improve matters!

Therefore our procedure is to compute two orthonormal n-vectors $\mathbf{p}$ and $\mathbf{q}$ such that the interpolated points all have the form $\mathbf{p}\cos\beta\theta + \mathbf{q}\sin\beta\theta$. For interpolation, $\beta = -\frac{1}{2}$ corresponds to $\mathbf{u}$ and $\beta = \frac{1}{2}$ corresponds to $\mathbf{v}$. Extrapolation

may be performed by using larger values of β. The quantities $\mathbf{p}$, $\mathbf{q}$ and θ are given by

$$\begin{aligned} \mathbf{p} &= \texttt{normv}(\mathbf{v}+\mathbf{u}), \\ \mathbf{q} &= \texttt{normv}(\mathbf{v}-\mathbf{u}), \\ \theta &= 2\,\text{atan2}(|\mathbf{v}-\mathbf{u}|,|\mathbf{v}+\mathbf{u}|) \text{if } |\mathbf{v}-\mathbf{u}| \le |\mathbf{v}+\mathbf{u}|, \\ \theta &= \pi - 2\,\text{atan2}(|\mathbf{u}+\mathbf{v}|,|\mathbf{v}-\mathbf{u}|) \text{if } |\mathbf{v}+\mathbf{u}| < |\mathbf{v}-\mathbf{u}|, \end{aligned} \tag{135}$$

where the function "normv" accurately normalizes a vector as described in Appendix J. The slerp function is produced by

$$\texttt{slerp}(\mathbf{u},\mathbf{v},\alpha) = \mathbf{p}\cos(\alpha-\tfrac{1}{2})\theta + \mathbf{q}\sin(\alpha-\tfrac{1}{2})\theta \tag{136}$$

in conjunction with Eq. 135.

It is apparently accepted as folk lore that varying α at a constant rate in slerp corresponds to rotating in 3-D around a fixed axis at a constant rate; i.e., the orientation maintains a constant angular velocity. This is perhaps the major physical justification for quaternion interpolation. Shoemake states this briefly with a reference, but it seems quite difficult to prove with published versions of slerp [Sho85]. However, the representation in Eq. 136, together with equations from Shoemake for quaternion multiplication and inverse, makes it straightforward to verify this important property.

We briefly sketch this technical result, assuming familiarity with the background, including "vector" and "scalar" components of quaternions and their multiplication rule [Sho85]. (Also see Eq. 63 in Section 7.2.) The main point is that because $\mathbf{p}$ and $\mathbf{q}$ are orthogonal by construction, $(\mathbf{p}^{-1}\mathbf{q})$ has zero "scalar" component, and its "vector" component is the needed axis in 3-D. Using the quaternion multiplication equation and writing the the quaternion as a row vector with the "scalar" component last:

$$\mathbf{p}^{-1}\,\texttt{slerp}(\mathbf{u},\mathbf{v},\alpha) = \left[\,(\mathbf{p}^{-1}\mathbf{q})_{vector}\,\sin(\alpha-\tfrac{1}{2})\theta,\ \cos(\alpha-\tfrac{1}{2})\theta\,\right] \tag{137}$$

This corresponds in 3-D to a rotation around $(\mathbf{p}^{-1}\mathbf{q})_{vector}$ (which has unit magnitude) by an angle of $(\alpha-\frac{1}{2})\theta/2$.

Appendix F Gram-Schmidt Procedure

The Gram-Schmidt procedure produces a vector that is orthonormal to a given set of orthonormal vectors, given some vector that is linearly independent of the set of orthonormal vectors [HS74, GVL96]. Let $\mathbf{C}_j$, $k < j < n$, be columns of a matrix that are already orthonormal. Let $\mathbf{C}'_k$ be the independent column.

$$\begin{aligned} \mathbf{v} &= \mathbf{C}'_k - \sum k < j < n \mathbf{C}_j\,(\mathbf{C}_j^T\,\mathbf{C}_k) \\ \mathbf{C}_k &= \texttt{normv}(\mathbf{v}). \end{aligned}$$

That is, the parallel components are subtracted off and the remainder is normalized (see Appendix J about numerical issues). For most applications in this paper, the more complicated *modified* Gram-Schmidt procedure [GVL96] is not needed, because most of the starting vectors are nearly orthogonal.

Appendix G Invariance under Permutations and Negations

Let $\mathbf{C} = \mathbf{A}\mathbf{B}^T$. Then $\mathbf{C}$ can be written as a sum of rank-one matrices [GVL96], recalling that single subscripts on matrices denote columns.

$$\mathbf{C} = \sum_j \mathbf{A}_j\,(\mathbf{B}_j)^T \tag{138}$$

Therefore if the same permutation of columns is applied to $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{C}$ does not change. Furthermore, if the same set of columns are negated in $\mathbf{A}$ and in $\mathbf{B}$, $\mathbf{C}$ does not change.

Appendix H Polar Decomposition Details

Appendix H.1 Semi-Uniqueness with One Eigenvalue of Zero

In this subsection we prove Theorem 6.1 in Section 6, which states that an $n \times n$ matrix with a 0 eigenvalue of multiplicity 1 has *two* polar decompositions, one whose orthonormal component has determinant $+1$ and one whose orthonormal component has determinant -1 . We have not seen this fact stated in the literature.

To recap from Section 6, a polar decomposition of $\mathbf{M}$ consists of an orthonormal matrix $\mathbf{Q}$ and a symmetric positive semidefinite matrix $\mathbf{S}$ such that $\mathbf{M} = \mathbf{QS}$. Since $\mathbf{M}^T\mathbf{M} = \mathbf{S}^2$, $\mathbf{S}$ is unique.

Let $\mathbf{S} = \mathbf{R}\Lambda\mathbf{R}^T$, where Λ is diagonal and $\mathbf{R}$ is orthonormal. Define $\lambda_j = \Lambda_{jj}$. By Appendix G, $\mathbf{S}$ is invariant if the same permutation is applied to the rows and columns of Λ and to the columns of $\mathbf{R}$. Also, $\mathbf{S}$ is invariant if any set of columns of $\mathbf{R}$ is negated. For the moment, let us fix one choice of column order and signs and define j_0 to be the column associated with eigenvalue 0, i.e., $\lambda_{j_0} = 0$. Note that $\mathbf{R}_{j_0}$ is the unique unit-length eigenvector of 0 up to multiplication by ± 1, for $\mathbf{S}$ and hence for $\mathbf{M}$.

We now express constraints on $\mathbf{Q}$ to comprise the orthonormal component of the polar decomposition. Define

$$\begin{aligned} \mathbf{G} &= \mathbf{MR} = \mathbf{QR}\Lambda \\ \mathbf{C}_j &= \mathbf{G}_j/\lambda_j = \mathbf{MR}_j/\lambda_j \qquad \text{for } j \neq j_0 \\ \mathbf{C}_{j_0} &= \text{(see next paragraph)} \\ \mathbf{Q}_+ &= \mathbf{CR}^T \end{aligned} \tag{139}$$

All columns of $\mathbf{C}$ other than j_0 are orthonormal (examine expressions for $(\mathbf{C}_j)^T\mathbf{C}_k$ for $j,k \neq j_0$ and use the fact that $\mathbf{M}^T\mathbf{M} = \mathbf{R}\Lambda^2\mathbf{R}^T$). Let $\mathbf{C}_{j_0}$ be the unique unit-length vector orthogonal to all other columns of $\mathbf{C}$ such that $\det(\mathbf{C}) = \det(\mathbf{R})$. Thus, for a given $\mathbf{R}$, $\mathbf{Q}_+$ is the only solution of Eq. 139 with determinant +1. By construction, $\mathbf{Q}_+\mathbf{S} = \mathbf{M}$.

Next, we use an explicit representation for $\mathbf{Q}_+$, based on Eq. 138 and Eq. 139:

$$\mathbf{Q}_+ = \mathbf{M}\sum_{j\neq j_0} \mathbf{R}_j\,(\mathbf{R}_j)^T/\lambda_j + \mathbf{C}_{j_0}\,(\mathbf{R}_{j_0})^T \tag{140}$$

Now suppose the columns of $\mathbf{R}$ are permuted and λ_j are subjected to the same permutation. It is evident from Eq. 140 that $\mathbf{Q}_+$ is not affected. The columns of $\mathbf{C}$ undergo the same permutation, since $(\mathbf{MR})_j = \mathbf{M}(\mathbf{R}_j)$, etc. If the permutation causes $\det(\mathbf{R})$ to change sign, $\det(\mathbf{C})$ also changes sign. The same applies if some columns of $\mathbf{R}$ other than j_0 are negated. If column j_0 is negated, that reverses the determinant of $\mathbf{R}$ and the computation rule requires that the sign of column $\mathbf{C}_{j_0}$ also be reversed, and again $\mathbf{Q}_+$ is not affected.

Suppose some λ_j has multiplicity $k > 1$. Without loss of generality we can assume it occurs in columns 0 through $k-1$, and $j_0 > k-1$. Now if $\mathbf{T}$ is any orthonormal $k \times k$ matrix and $\mathbf{B}$ denotes columns 0 through $k-1$ of $\mathbf{R}$, then those columns can be replaced by the new basis $\mathbf{B}_* = \mathbf{BT}$; that is, the columns of $\mathbf{B}_*$ constitute another orthonormal set of eigenvectors for λ_0, which occurs precisely at indexes 0 through $k-1$, for a possible diagonalization of $\mathbf{S}$. However,

$$\sum_{0\leq j<k} \mathbf{R}_j\,(\mathbf{R}_j)^T/\lambda_0 = \mathbf{BB}^T/\lambda_0 = \mathbf{B}_*\mathbf{B}_*^T/\lambda_0$$

so this substitution also leaves $\mathbf{Q}_+$ invariant.

In summary, proceeding from a particular diagonalization of $\mathbf{S}$, we have considered all transformations leading to other possible diagonalizations of $\mathbf{S}$, and have shown that $\mathbf{Q}_+$ is invariant, provided the sign of $\mathbf{C}_{j_0}$ is chosen to make $\det(\mathbf{C}) = \det(\mathbf{R})$. This choice is necessary for the determinant of $\mathbf{Q}_+$ to be positive.

The same analysis can be repeated choosing the sign of $\mathbf{C}_{j_0}$ to make $\det(\mathbf{C})$ the negative of $\det(\mathbf{R})$, leading to a matrix $\mathbf{Q}_-$ with negative determinant, instead of $\mathbf{Q}_+$. Using Eq. 140, we have $\mathbf{Q}_- = \mathbf{Q}_+ - 2\mathbf{C}_{j_0}\,(\mathbf{R}_{j_0})^T$, where the sign of $\mathbf{C}_{j_0}$ is chosen as required for $\mathbf{Q}_+$. That is, the two solutions differ in the orthonormal component by a rank-one matrix.

Appendix H.2 Nonsquare Matrices

Suppose $\mathbf{M}$ is $m \times n$ with $m > n$. The steps leading up to Eq. 139 can still be carried out. First, suppose that $\mathbf{M}$ has full rank. Then $\mathbf{S}$ is nonsingular, there is no column j_0, and all columns of $\mathbf{C}$ are determined normally. Conclude by computing $\mathbf{Q} = \mathbf{C}\mathbf{R}^T$. The determinant is not defined. As argued immediately after Eq. 139, all columns of $\mathbf{Q}$ are orthonormal and $\mathbf{M} = \mathbf{Q}\mathbf{S}$. Also, the analog of Eq. 140 simply omits the special case for j_0, and shows that $\mathbf{Q}$ is invariant under varying diagonalizations of $\mathbf{S}$. This procedure is equivalent to computing $\mathbf{M}\mathbf{S}^{-1}$ but the inverse is not materialized.

Now suppose $\mathbf{M}$ has rank $n-1$, so $\mathbf{S}$ has a 0 eigenvalue of multiplicity 1. Proceed as before up to the point where $\mathbf{C}_{j_0}$ needs to be computed. Now $\mathbf{C}_{j_0}$ is an m-vector that needs to be orthonormal to a set of $n-1$ other m-vectors. The solution is never unique (with $m > n$), but solutions are easy to come by with the Gram-Schmidt procedure (see Appendix F) applied to an initial guess for the column.

Appendix H.3 Rank-One Matrices

Although the need for polar decomposition of a 3×3 rank-1 matrix is unlikely to arise in computer graphics, the analysis is included for completeness. The matrix has the form $\mathbf{M} = \mathbf{u}\mathbf{v}^T$. The symmetric part of the decomposition is $(|\mathbf{u}|/|\mathbf{v}|)\,\mathbf{v}\mathbf{v}^T$. The rotation $\mathbf{R}$ may be any rotation that maps $\mathbf{v}$ to $(|\mathbf{v}|/|\mathbf{u}|)\,\mathbf{u}$. If $\mathbf{u}$ and $\mathbf{v}$ are collinear, $\mathbf{R} = \mathbf{I}$; otherwise, any rotation whose axis is in the plane that bisects $\mathbf{u}$ and $\mathbf{v}$ suffices, but perhaps the most natural axis is $\mathbf{v} \times \mathbf{u}$. Using this axis, the amount of rotation is θ, where $\cos\theta = (\mathbf{u}^T\mathbf{v})/(|\mathbf{u}||\mathbf{v}|)$. The matrix to accomplish any such rotation can be found in any standard text [FvDFH90].

The approach generalizes $n > 3$, where $\mathbf{M}$ has rank one. The hyperplane that bisects $\mathbf{u}$ and $\mathbf{v}$ is orthogonal to $(\mathbf{u}/|\mathbf{u}|) - (\mathbf{v}/|\mathbf{v}|)$. There are $n-3$ degrees of freedom for choosing the axis direction even after requiring it to be orthogonal to $\mathbf{u}$ and $\mathbf{v}$.

Appendix I Examples

Example I.1: Let

$$\mathbf{A} = \begin{bmatrix} -15 & -16 & -20 \\ 44 & 45 & 19 \\ 16 & 16 & 21 \end{bmatrix} \tag{141}$$

The characteristic polynomial is

$$f(x) = x^3 - 51x^2 + 675x - 625 \tag{142}$$

We find that $\widetilde{x} = 17$ and $f(\widetilde{x}) = 1024 > 0$. So the outer root, λ_0, is less than 17. Both root-finding methods described in Appendix A calculate $\lambda_0 = 1.0$, exactly. By way of comparison, *eig*, the general eigenvalue routine in *matlab*, computed this eigenvalue with an error of 4 ULPs (*U*nits in the *L*ast *P*lace).

Now $q(x) = x^2 - 50x + 625$, and

$$q(\mathbf{A}) = \mathbf{A}^2 - 50\mathbf{A} + 625\mathbf{I} = \begin{bmatrix} 576 & 0 & 576 \\ -576 & 0 & -576 \\ 0 & 0 & 0 \end{bmatrix} \tag{143}$$

Therefore $\gamma = 1/576$ and the outer eigenvector and normal vector are

$$\boldsymbol{\xi}_0 = \begin{bmatrix} 576 \\ -576 \\ 0 \end{bmatrix} \qquad \boldsymbol{\eta} = \begin{bmatrix} 576 \\ 0 \\ 576 \end{bmatrix}, \tag{144}$$

respectively. This example is continued in Example I.2. □

Example I.2: Continuing with Example I.1, we were given $\mathbf{A}$, and had calculated $\boldsymbol{\xi}_0$, and $\boldsymbol{\eta}$. In this case, $j = 0$ corresponds to the component of $\boldsymbol{\eta}$ that is maximum in magnitude. So the columns of the required rotation matrix are in the directions of $\boldsymbol{\eta}$, $\mathbf{e}_2 \times \boldsymbol{\eta}$, and $\boldsymbol{\eta} \times (\mathbf{e}_2 \times \boldsymbol{\eta})$, which are:

$$\begin{bmatrix} 576 \\ 0 \\ 576 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ 576 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} -576^2 \\ 0 \\ 576^2 \end{bmatrix}$$

These columns need to be normalized. Using Eqs. 91 and 92, the results are

$$(\mathbf{RS}) = \begin{bmatrix} \sqrt{2} & 0 & -1/\sqrt{2} \\ -\sqrt{2} & 1 & 0 \\ 0 & 0 & 1/\sqrt{2} \end{bmatrix} \qquad (\mathbf{RS})^{-1} = \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 1 & 1 & 1 \\ 0 & 0 & \sqrt{2} \end{bmatrix} \tag{145}$$

We also have

$$\mathbf{A} \overset{\mathbf{RS}}{\rightsquigarrow} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 45 & -25/\sqrt{2} \\ 0 & 32/\sqrt{2} & 5 \end{bmatrix} \tag{146}$$

Therefore, the outer-root eigenvalue ($\lambda_0 = 1$) has been separated from the rest of the transformation.

Of course, the exact expressions with square roots given above are only approximately represented numerically. Although the outer-root eigenvalue ($\lambda_0 = 1$) was computed exactly by our numerical procedures in Appendix A, the numerical representation of the remaining 2×2 matrix in the lower right corner of Eq. 146 is not exact when computed with double precision. This example is continued in Example I.3. □

Example I.3: Continuing with Example I.2, we had the 2×2 matrix in the lower right corner of Eq. 146:

$$\mathbf{B} = \begin{bmatrix} 45 & -25/\sqrt{2} \\ 32/\sqrt{2} & 5 \end{bmatrix} \tag{147}$$

Applying Eqs. 98–103

$$D = 1624.5 \tag{148}$$

$$c = \sqrt{\frac{1}{2} + \frac{7}{2\sqrt{3249}}} = \sqrt{\frac{32}{57}} \tag{149}$$

$$s = -\sqrt{\frac{1}{2} - \frac{7}{2\sqrt{3249}}} = -\sqrt{\frac{25}{57}} \tag{150}$$

$$M = \begin{bmatrix} 25 & 0 \\ 57/\sqrt{2} & 25 \end{bmatrix} \tag{151}$$

$$\tag{152}$$

The element M_{12} is calculated in double precision as

$$M_{12} = \tfrac{1}{2}\left(\sqrt{1624.5} + \left(\frac{-25}{\sqrt{2}} - \frac{32}{\sqrt{2}}\right)\right) \tag{153}$$

which happens to come out exactly 0, but with the operations performed with a different associativity it comes out as $\approx 1.78 \cdot 10^{-15}$. If we proceed with the nonuniform scale using this value, then $w \approx 0.0000815$, $1/w \approx 12273$, and the off-diagonal elements are equalized at $\approx 2.68 \cdot 10^{-7}$. After diagonalizing the matrix with a final 45° rotation, the diagonal elements are $25 \pm 2.68 \cdot 10^{-7}$. However, the transforming matrix is given by:

$$\mathbf{T} = \begin{bmatrix} 5747.44724025183 & 5747.44715391519 \\ 6502.49418195522 & 6502.49425826675 \end{bmatrix} \qquad \mathbf{T}^{-1} = \begin{bmatrix} 6502.49425826675 & -5747.44715391519 \\ -6502.49418195522 & 5747.44724025183 \end{bmatrix} \tag{154}$$

Although the determinant of $\mathbf{T}$ is 1 in theory, its calculated value is $1 - 8.8 \cdot 10^{-9}$. It is clear that the value of $\mathbf{T}$ in Eq. 154 is very undesirable to use for further calculations, and will lead to inaccuracies.

If we had stopped after the first rotation and declared M_{12} to be zero, then

$$\mathbf{T} = \begin{bmatrix} 0.74926864926536 & 0.66226617853252 \\ -0.66226617853252 & 0.74926864926536 \end{bmatrix} \tag{155}$$

□

Example I.4: Suppose we start with the matrix

$$\mathbf{B} = \begin{bmatrix} -1 & 2 \\ 1 & 3 \end{bmatrix}$$

Note that $\det(\mathbf{B}) = -5$. We have $(\mathsf{B}_{11} - \mathsf{B}_{22}) = -4$, $(\mathsf{B}_{12} + \mathsf{B}_{21}) = 3$, $D = 25$, $\sqrt{D} = 5$. By Equations 100 and 101, $c = \sqrt{0.8}$ and $s = \sqrt{0.2}$. These values determine θ by Eq. 102, which in turn determines the matrix $\mathbf{R}_\theta$ by Eq. 20. After the similarity transformation by $\mathbf{R}_\theta$, according to Eq. 103:

$$\mathbf{B} \overset{\mathbf{R}_\theta}{\rightsquigarrow} \begin{bmatrix} 1 & 3 \\ 2 & 1 \end{bmatrix}$$

For the next step, choose $w = \sqrt[4]{3/2}$ and apply $\overset{\mathbf{S}_w}{\rightsquigarrow}$ (see Equations 21 and 105):

$$\mathbf{B}_{RS_w} = \begin{bmatrix} 1 & \sqrt{6} \\ \sqrt{6} & 1 \end{bmatrix}$$

The final step, since the matrix is symmetric, is to apply a rotation of $-\pi/4$ as a similarity transformation. Thus the composite coordinate transformation is

$$\begin{aligned} \mathbf{T} = \mathbf{R}\mathbf{S}_w\mathbf{R}_{-\pi/4} &= \begin{bmatrix} \sqrt{\frac{4}{5}} & -\sqrt{\frac{1}{5}} \\ \sqrt{\frac{1}{5}} & \sqrt{\frac{4}{5}} \end{bmatrix} \begin{bmatrix} \sqrt[4]{\frac{3}{2}} & 0 \\ 0 & \sqrt[4]{\frac{2}{3}} \end{bmatrix} \begin{bmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix} \\ &= \frac{\sqrt[4]{6}}{\sqrt{60}} \begin{bmatrix} 2\sqrt{3}+\sqrt{2} & 2\sqrt{3}-\sqrt{2} \\ \sqrt{3}-2\sqrt{2} & \sqrt{3}+2\sqrt{2} \end{bmatrix} \approx \begin{bmatrix} 0.986 & 0.414 \\ -0.222 & 0.921 \end{bmatrix} \end{aligned}$$

giving

$$\mathbf{B} \overset{\mathbf{T}}{\rightsquigarrow} \begin{bmatrix} 1-\sqrt{6} & 0 \\ 0 & 1+\sqrt{6} \end{bmatrix}$$

We easily verify that $\det(\mathbf{B} \overset{\mathbf{T}}{\rightsquigarrow}) = -5$, as it should for any similarity transformation. More importantly, $\det(\mathbf{T}) = 1$.

Since the canonical matrix is diagonal, the eigenvalues and eigenvectors of $\mathbf{B}$ are real. The diagonal elements of $\mathbf{B} \overset{\mathbf{T}}{\rightsquigarrow}$ define the eigenvalues. The columns of $\mathbf{T}$ define the eigenvectors. Since $\mathbf{B}$ is asymmetric, the eigenvectors are not orthogonal. Also, they are not of unit length because we desired $\det(\mathbf{T}) = 1$. The eigenvectors map into the axes of the canonical coordinate system. □

Example I.5: Consider the matrix $\mathbf{A}$ and its characteristic polynomial:

$$\mathbf{A} = \begin{bmatrix} 1.60 & -1.20 & 1.60 \\ 0.54 & 1.12 & -0.16 \\ -0.32 & -0.96 & 2.28 \end{bmatrix}, \qquad f(x) = x^3 - 5x^2 + 9x - 5.$$

The outer root is 1. We find that $q(x) = x^2 - 4x + 5$. Thus

$$q(\mathbf{A}) = \begin{bmatrix} 0 & 0 & 0 \\ -0.64 & 1.28 & 0.96 \\ -0.48 & 0.96 & 0.72 \end{bmatrix}, \qquad \xi_0 = \begin{bmatrix} 0 \\ 1.28 \\ 0.96 \end{bmatrix}, \qquad \eta = \begin{bmatrix} -0.64 \\ 1.28 \\ 0.96 \end{bmatrix},$$

because 1.28 has the maximum magnitude in $q(\mathbf{A})$. The transforming matrix and partitioned matrix are found using Eq. 87–92 in Appendix B.1:

$$\mathbf{RS} = \begin{bmatrix} 0.9285 & 0 & 0 \\ 0.2971 & 0.8616 & -0.60 \\ 0.2228 & 0.6462 & 0.80 \end{bmatrix}, \qquad \mathbf{A} \overset{\mathbf{RS}}{\rightsquigarrow} = \begin{bmatrix} 1.60 & 0 & 2.1541 \\ 0 & 1 & 0 \\ -0.5385 & 0 & 2.40 \end{bmatrix} = \begin{bmatrix} b_{22} & 0 & b_{21} \\ 0 & 1 & 0 \\ b_{12} & 0 & b_{11} \end{bmatrix}$$

Numbers with four decimal places are approximate. Using Eq. 98–105 in Section B.2,

$$\mathbf{U} = \begin{bmatrix} 1.4598 & 0 & -0.1520 \\ 0 & 1 & 0 \\ 0.3416 & 0 & 0.6495 \end{bmatrix}$$

Combining $\mathbf{T} = \mathbf{RSU}$ gives the final RJCF:

$$\mathbf{T} = \begin{bmatrix} 1.3554 & 0 & -0.1411 \\ 0.2288 & 0.8616 & -0.4348 \\ 0.5986 & 0.6462 & 0.4857 \end{bmatrix}, \qquad \mathbf{J} = \mathbf{A} \overset{\mathbf{T}}{\rightsquigarrow} = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix}$$

In Example I.6 we will see how to obtain $\log(\mathbf{A})$ from these matrices. □

Example I.6: Recall $\mathbf{A}$, its RJCF $\mathbf{J}$ and transforming matrix $\mathbf{T}$ from Example I.5. By Eq. 41, $a = \frac{1}{2}\log(5) = 0.8047$ and $b = \text{atan2}(1,2) = 0.4636$. (All values with four decimal places are approximate.) Then

$$\log(\mathbf{J}) = \begin{bmatrix} \frac{1}{2}\log(5) & 0 & \text{atan2}(1,2) \\ 0 & 0 & 0 \\ -\text{atan2}(1,2) & 0 & \frac{1}{2}\log(5) \end{bmatrix} = \begin{bmatrix} 0.8047 & 0 & 0.4636 \\ 0 & 0 & 0 \\ -0.4636 & 0 & 0.8047 \end{bmatrix}$$

$$\log(\mathbf{A}) = \mathbf{T}\log(\mathbf{J})\mathbf{T}^{-1} = \begin{bmatrix} 0.6193 & -0.5564 & 0.7419 \\ 0.3595 & 0.1784 & -0.2378 \\ -0.0665 & -0.6088 & 0.8118 \end{bmatrix}$$

Although we could multiply this by $\frac{1}{2}$, it is not clear how to recover $\mathbf{A}^{\frac{1}{2}}$ from this expression. Instead, re-exponentiate $\frac{1}{2}\log(\mathbf{J})$ with Eq. 40.

$$\mathbf{A}^{\frac{1}{2}} = \mathbf{T}\begin{bmatrix} 5^{1/4}\cos(.2318) & 0 & 5^{1/4}\sin(.2318) \\ 0 & 1 & 0 \\ -5^{1/4}\sin(.2318) & 0 & 5^{1/4}\cos(.2318) \end{bmatrix}\mathbf{T}^{-1} = \begin{bmatrix} 1.3179 & -0.4123 & 0.5497 \\ 0.2213 & 1.0815 & -0.1086 \\ -0.0831 & -0.3835 & 1.5113 \end{bmatrix}$$

Other fractions also work. □

Example I.7: This example shows the calculations for a matrix Lie product style interpolation where one matrix has deficient rank. Consider the matrices $\mathbf{B}$ and $\mathbf{C}$:

$$\mathbf{B} = \begin{bmatrix} 1.60 & -1.20 & 1.60 \\ 0.86 & 0.48 & -0.64 \\ -0.08 & -1.44 & 1.92 \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

$\mathbf{B}$ has rank 2 and its outer eigenvalue is 0. Its RJCF is

$$\mathbf{B} \overset{\mathbf{V}}{\rightsquigarrow} \mathbf{J} = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 2 \end{bmatrix} \qquad \text{where } \mathbf{V} = \begin{bmatrix} 1.3519 & 0 & 0.1714 \\ 0.3218 & 0.8616 & -0.3713 \\ 0.4722 & 0.6462 & 0.6093 \end{bmatrix}.$$

Numbers with four decimal places are approximate. By Eq. 41:

$$\log \mathbf{J} = \begin{bmatrix} 0.8047 & 0 & 0.4636 \\ 0 & -\infty & 0 \\ -0.4636 & 0 & 0.8047 \end{bmatrix}$$

A		Nielson-Jung Method with Imprecise Eigenvalues				Relaxed Canonical Form Method or Nielson-Jung with Exact Eigenvalues			
	t	0	0.0001	0.1	1	0	0.0001	0.1	1
3 5 –3	$\mathbf{p}(t)$	0.125	$-.0563\cdot 10^{12}$	$-62.3\cdot 10^{12}$	$-1798.\cdot 10^{12}$	0.1	0.100050008	0.15878	2.95562
1 7 –3		0.125	$-.0563\cdot 10^{12}$	$-62.3\cdot 10^{12}$	$-1798.\cdot 10^{12}$	0.1	0.100050008	0.15878	2.95562
2 10 –4		0.125	$-.1126\cdot 10^{12}$	$-124.6\cdot 10^{12}$	$-3596.\cdot 10^{12}$	0.1	0.100080014	0.19542	5.17234

Table 6: An extreme example of numerical instability, discussed in Example I.8

Since **C** is already in RJCF, we have

$$\log \mathbf{D} \;=\; (\log \mathbf{C})\overset{\mathbf{V}}{\rightsquigarrow} \;=\; \begin{bmatrix} \frac{1}{2}\log 2 & -\pi/4 & 0 \\ \pi/4 & \frac{1}{2}\log 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \overset{\mathbf{V}}{\rightsquigarrow} \;=\; \begin{bmatrix} 0.2950 & -0.4846 & 0.2692 \\ 0.7914 & 0.4731 & -0.1259 \\ -1.0679 & -0.1263 & -0.0750 \end{bmatrix}.$$

Now let us blend with $\alpha = 0.1$. Let **A** be defined indirectly by:

$$\log \mathbf{A} \;=\; \mathbf{L} \;=\; 0.1\log \mathbf{J} + 0.9\log \mathbf{D} \;=\; \begin{bmatrix} 0.3460 & -0.4361 & 0.2886 \\ 0.7122 & -\infty & -0.1133 \\ -1.0075 & -0.1137 & 0.0129 \end{bmatrix}.$$

The final result will be

$$0.1 \odot \mathbf{B} \oplus 0.9 \odot \mathbf{C} \;=\; \mathbf{A}\overset{\mathbf{V}^{-1}}{\rightsquigarrow},$$

and it is still several steps to derive it.

First, we set L_{01}, L_{21}, L_{10}, and L_{12} to zero. Then it is necessary to put the 2×2 submatrix that omits row 1 and column 1 into RJCF. We find **U** and **H** using Eq. 98–105 in Section B.2.

$$\mathbf{L} \;=\; \mathbf{UHU}^{-1} \;=\; \begin{bmatrix} 0.7345 & 0 & -0.1526 \\ 0 & 1 & 0 \\ -0.1526 & 0 & 1.3932 \end{bmatrix} \begin{bmatrix} 0.1795 & 0 & 0.5129 \\ 0 & -\infty & 0 \\ -0.5129 & 0 & 0.1795 \end{bmatrix} \begin{bmatrix} 1.3932 & 0 & 0.1526 \\ 0 & 1 & 0 \\ 0.1526 & 0 & 0.7345 \end{bmatrix}.$$

The middle matrix **H** is exponentiated as follows:

$$e^{\mathbf{H}} = \exp\begin{pmatrix} 0.1795 & 0 & 0.5129 \\ 0 & -\infty & 0 \\ -0.5129 & 0 & 0.1795 \end{pmatrix} = e^{0.1795}\begin{bmatrix} \cos(-0.5129) & 0 & \sin(0.5129) \\ 0 & 0 & 0 \\ -\sin(0.5129) & 0 & \cos(-0.5129) \end{bmatrix} = \begin{bmatrix} 1.0426 & 0 & 0.5871 \\ 0 & 0 & 0 \\ -0.5871 & 0 & 1.0426 \end{bmatrix}$$

The final blend, called $L(\mathbf{B},\mathbf{C},0.9)$, is:

$$\begin{aligned} L(\mathbf{B},\mathbf{C},0.9) \;=\; 0.1\odot\mathbf{B}\oplus 0.9\odot\mathbf{C} \;&=\; \mathbf{A}\overset{\mathbf{V}^{-1}}{\rightsquigarrow} \;=\; (e^{\mathbf{H}})\overset{\mathbf{U}^{-1}}{\rightsquigarrow}\overset{\mathbf{V}^{-1}}{\rightsquigarrow} \\ &=\; \begin{bmatrix} 1.0062 & -0.3550 & 0.4734 \\ 0.6729 & 0.2749 & -0.3665 \\ -0.2264 & -0.6032 & 0.8042 \end{bmatrix}. \end{aligned}$$

It is noteworthy that although $L(\mathbf{B},\mathbf{C},0.9)$ supposedly has only "0.1 of **B**", its only real eigenvalue is zero. □

Example I.8: Recall the matrix analyzed in Example C.1 (repeated on the left side of Table 6) and also discussed in Section 10 as matrix (d); see Table 4 in that section.

We applied the Nielson-Jung equations [NJ99] to the solution of $d\mathbf{p}/dt = \mathbf{Ap}$ with the initial condition $\mathbf{p}_0 = [0.1, 0.1, 0.1]^T$ (see discussion in Section 9). Due to the use of numerical eigenvalue procedures, the computed eigenvalues varied slightly from their exact values of 2, by about 10^{-15}. All calculations were performed in matlab, which uses double precision. Some values of $\mathbf{p}(t)$ are shown in Table 6.

The Nielson-Jung numbers, shown in the left part of the table, occurred because the procedure detected two distinct eigenvalues, very close to each other. A wide variety of initial conditions produced similar behavior. The fact that even $\mathbf{p}(0)$ has a 25% error shows how bad the numerical problems were, and by $t = 1$ the error was on the order of 10^{15}. The discussion in Section 10 mentioned that the transforming matrix, $\mathbf{T}$, is very badly conditioned. Although Nielson and Jung do not compute the transforming matrix explicitly, we believe the left part of the table is a similar phenomenon.

The results obtained using the methods of this paper are shown in the right part of the table and are very accurate. Essentially identical results are computed by the Nielson-Jung procedures when they treat all eigenvalues as equal, thereby using equations for a non-hyperbolic case.

□

Appendix J Scaling for Floating-Point Accuracy: Vector Norm and Cube Root

Scaling for floating-point accuracy is occasionally helpful to extend the range of values for which a procedure is accurate. A typical case is the function `normv` that normalizes a vector, say $[x,y,z]$. If all components are very small in magnitude, say less than 2^{-63}, we can safely multiply them by 2^{126}. No accuracy is lost because the mantissa does not change. We are now assured that the largest component can be squared without underflowing to zero, and that no components will overflow the largest representable floating point value. Similarly, scaling down in the presence of a very large component can prevent floating point overflow. Since the final result is unit-length, it is not necessary to undo the scaling.

A more advanced example is the cube root procedure, which is a special case of the n-th root procedure, where n is a positive integer. This function is available in *matlab* as `nthroot` with a different implementation from that described here. The standard C math library has `cbrt`. In the description below, it is simple to adapt the procedure to compute the n-th root instead. (Of course, if n is even, x must be positive.)

The cube root procedure uses the IEEE library functions `lrexp` and `ldexp` to achieve scaling by powers of 2^3. Logically, the identity is

$$\sqrt[3]{x} \;=\; 2^k \sqrt[3]{x/2^{3k}} \tag{156}$$

The integer k is chosen so that $2^{-2} \le |x/2^{3k}| < 2$. (Note that `lrexp` returns p and $x/2^p$ such that $\frac{1}{2} \le |x/2^p| < 1$.) Let $y = |x/2^{3k}|$ and apply the Newton-Raphson procedure, as follows:

$$r_m \;=\; \frac{2r_{m-1} + (y/r_{m-1}^2)}{3} \qquad\qquad r_0 = 1.0. \tag{157}$$

The sequence r_1, r_2, ... monotonically decreases to $\sqrt[3]{y}$. It can be cut off early to obtain an upper bound that is reasonably close. The final result can be "assembled" with $\texttt{ldexp}(r_m, k)$. This function is much faster than $e^{\log(|x|)/3.0}$ and is also more accurate.

Appendix K Transforming from Affine to Linear

Suppose we are given an n-dimensional coordinate system $\mathbf{p}_{raw}$ and a nonsingular affine transformation

$$\mathbf{A}\mathbf{p}_{raw} + \mathbf{c} \qquad \mathbf{p}_{raw} = \begin{bmatrix} p^{(0)} \\ \vdots \\ p^{(n-1)} \end{bmatrix}_{raw} \tag{158}$$

where $\mathbf{A}$ is $n \times n$ and $\mathbf{c}$ and $\mathbf{p}_{raw}$ are $n \times 1$. This expression also defines an n-dimensional vector field. When $\mathbf{A}$ is nonsingular, the unique critical point, where the vector field = 0, is given by

$$\mathbf{p}_{crit} \;=\; -\mathbf{A}^{-1}\mathbf{c} \tag{159}$$

If we make the translation $\mathbf{p} = \mathbf{p}_{raw} - \mathbf{p}_{crit}$, then in the coordinates of $\mathbf{p}$ the critical point is at zero and the transformation is simply $\mathbf{A}$, applied to $\mathbf{p}$. We assume that this translation has been done as a preprocessing step and the transformation under consideration is $\mathbf{A}$.

In the case that $\mathbf{A}$ is singular, there are infinitely many points for which $\mathbf{A}\mathbf{p} = 0$. Yet there is not necessarily *any* critical point, $\mathbf{p}_{crit}$, that satisfies $\mathbf{A}\mathbf{p}_{crit} + \mathbf{c} = 0$. In these cases the nonhomogeneous term $\mathbf{c}$ has to be carried along, and transformed into the canonical coordinate system.

On the other hand, if $\mathbf{A}$ is singular and $\mathbf{A}\mathbf{p}_{crit} + \mathbf{c} = 0$ has solutions for $\mathbf{p}_{crit}$, then it has infinitely many solutions. A convenient one can be chosen to carry out the translation $\mathbf{p} = \mathbf{p}_{raw} - \mathbf{p}_{crit}$. For example, the critical point closest to a specified point of interest, $\mathbf{p}_{central}$, can be found using the pseudo-inverse [PFTV88]. Possibly with rearrangement of rows and columns, let $\mathbf{A}_{ul}$, the upper left partition of $\mathbf{A}$, be nonsingular and have the same rank as $\mathbf{A}$. Let the subscript u denote the corresponding upper partition of vectors. That is,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{ul} & \mathbf{A}_{ur} \\ \mathbf{A}_{ll} & \mathbf{A}_{lr} \end{bmatrix} \qquad \mathbf{c} = \begin{bmatrix} \mathbf{c}_u \\ \mathbf{c}_l \end{bmatrix} \tag{160}$$

Then

$$\begin{aligned} \mathbf{p}_{crit} &= \mathbf{p}_{central} \\ &- \begin{bmatrix} \mathbf{A}_{ul}^T \\ \mathbf{A}_{ll}^T \end{bmatrix} \left(\mathbf{A}_{ul}\mathbf{A}_{ul}^T + \mathbf{A}_{ll}\mathbf{A}_{ll}^T\right)^{-1} \left(\mathbf{c} - \mathbf{p}_{central}\right)_u \end{aligned} \tag{161}$$

In general, our procedures do not assume $\mathbf{A}$ is nonsingular.