Appendix 4

Matrix Properties and Decompositions

In this appendix we discuss matrices with particular forms that occur throughout the book, and also various matrix decompositions.

A4.1 Orthogonal matrices

A square matrix $\mathtt{U}$ is known as *orthogonal* if its transpose is its inverse – symbolically $\mathtt{U}^\mathsf{T}\mathtt{U} = \mathtt{I}$, where $\mathtt{I}$ is the identity matrix. This means that the column vectors of $\mathtt{U}$ are all of unit norm and are orthogonal. This may be written $\mathbf{u}_i^\mathsf{T}\mathbf{u}_j = \delta_{ij}$. From the condition $\mathtt{U}^\mathsf{T}\mathtt{U} = \mathtt{I}$ one easily deduces that $\mathtt{U}\mathtt{U}^\mathsf{T} = \mathtt{I}$. Hence the row vectors of $\mathtt{U}$ are also of unit norm and are orthogonal. Consider once more the equation $\mathtt{U}^\mathsf{T}\mathtt{U} = \mathtt{I}$. Taking determinants leads to the equation $(\det \mathtt{U})^2 = 1$, since $\det \mathtt{U} = \det \mathtt{U}^\mathsf{T}$. Thus if $\mathtt{U}$ is orthogonal, then $\det \mathtt{U} = \pm 1$.

One easily verifies that the orthogonal matrices of a given fixed dimension form a group, denoted O_n, since if $\mathtt{U}$ and $\mathtt{V}$ are orthogonal, then $(\mathtt{U}\mathtt{V})^\mathsf{T}\mathtt{U}\mathtt{V} = \mathtt{V}^\mathsf{T}\mathtt{U}^\mathsf{T}\mathtt{U}\mathtt{V} = \mathtt{I}$. Furthermore, the orthogonal matrices of dimension n with positive determinant form a group, called SO_n. An element of SO_n is called an n-dimensional rotation.

Norm-preserving properties of orthogonal matrices. Given a vector $\mathbf{x}$, the notation $\|\mathbf{x}\|$ represents its Euclidean length. This can be written as $\|\mathbf{x}\| = (\mathbf{x}^\mathsf{T}\mathbf{x})^{1/2}$. An important property of orthogonal matrices is that multiplying a vector by an orthogonal matrix preserves its norm. This is easily seen by computing

$$(\mathtt{U}\mathbf{x})^\mathsf{T}(\mathtt{U}\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathtt{U}^\mathsf{T}\mathtt{U}\mathbf{x} = \mathbf{x}^\mathsf{T}\mathbf{x}.$$

By the QR decomposition of a matrix is usually meant the decomposition of the matrix $\mathtt{A}$ into a product $\mathtt{A} = \mathtt{Q}\mathtt{R}$, where $\mathtt{Q}$ is orthogonal, and $\mathtt{R}$ is an upper-triangular matrix. The letter $\mathtt{R}$ stands for *Right*, meaning upper-triangular. Similar to the QR decomposition, there are also QL, LQ and RQ decompositions, where $\mathtt{L}$ denotes a *Left* or lower-triangular matrix. In fact, the RQ decomposition of a matrix is the one that will be of most use in this book, and which will therefore be discussed here. The most important case is the decomposition of a 3×3 matrix and we will concentrate on this in the following section.

A4.1.1 Givens rotations and RQ decomposition

A 3-dimensional Givens rotation is a rotation about one of the three coordinate axes. The three Givens rotations are

$$\mathtt{Q}_x = \begin{bmatrix} 1 & & \\ & c & -s \\ & s & c \end{bmatrix} \quad \mathtt{Q}_y = \begin{bmatrix} c & & s \\ & 1 & \\ -s & & c \end{bmatrix} \quad \mathtt{Q}_z = \begin{bmatrix} c & -s & \\ s & c & \\ & & 1 \end{bmatrix} \tag{A4.1}$$

where $c = \cos(\theta)$ and $s = \sin(\theta)$ for some angle θ and blank entries represent zeros.

Multiplying a 3×3 matrix $\mathtt{A}$ on the right by (for instance) $\mathtt{Q}_z$ has the effect of leaving the last column of $\mathtt{A}$ unchanged, and replacing the first two columns by linear combinations of the original two columns. The angle θ may be chosen so that any given entry in the first two columns becomes zero.

For instance, to set the entry $\mathtt{A}_{21}$ to zero, we need to solve the equation $ca_{21} + sa_{22} = 0$. The solution to this is $c = -a_{22}/(a_{22}^2 + a_{21}^2)^{1/2}$ and $s = a_{21}/(a_{22}^2 + a_{21}^2)^{1/2}$. It is required that $c^2 + s^2 = 1$ since $c = \cos(\theta)$ and $s = \sin(\theta)$, and the values of c and s given here satisfy that requirement.

The strategy of the RQ algorithm is to clear out the lower half of the matrix one entry at a time by multiplication by Givens rotations. Consider the decomposition of a 3×3 matrix $\mathtt{A}$ as $\mathtt{A} = \mathtt{RQ}$ where $\mathtt{R}$ is upper-triangular and $\mathtt{Q}$ is a rotation matrix. This may take place in three steps. Each step consists of multiplication on the right by a Givens rotation to set a chosen entry of the matrix $\mathtt{A}$ to zero. The sequence of multiplications must be chosen in such a way as not to disturb the entries that have already been set to zero. An implementation of the RQ decomposition is given in algorithm A4.1.

Objective

Carry out the RQ decomposition of a 3×3 matrix $\mathtt{A}$ using Givens rotations.

Algorithm

(i) Multiply by $\mathtt{Q}_x$ so as to set $\mathtt{A}_{32}$ to zero.
(ii) Multiply by $\mathtt{Q}_y$ so as to set $\mathtt{A}_{31}$ to zero. This multiplication does not change the second column of $\mathtt{A}$, hence $\mathtt{A}_{32}$ remains zero.
(iii) Multiply by $\mathtt{Q}_z$ so as to set $\mathtt{A}_{21}$ to zero. The first two columns are replaced by linear combinations of themselves. Thus, $\mathtt{A}_{31}$ and $\mathtt{A}_{32}$ remain zero.

Algorithm A4.1. *RQ decomposition of a* 3×3 *matrix.*

Other sequences of Givens rotations may be chosen to give the same result. As a result of these operations, we find that $\mathtt{AQ}_x\mathtt{Q}_y\mathtt{Q}_z = \mathtt{R}$ where $\mathtt{R}$ is upper-triangular. Consequently, $\mathtt{A} = \mathtt{RQ}_z^\mathsf{T}\mathtt{Q}_y^\mathsf{T}\mathtt{Q}_x^\mathsf{T}$, and so $\mathtt{A} = \mathtt{RQ}$ where $\mathtt{Q} = \mathtt{Q}_z^\mathsf{T}\mathtt{Q}_y^\mathsf{T}\mathtt{Q}_x^\mathsf{T}$ is a rotation. In addition, the angles θ_x, θ_y and θ_z associated with the three Givens rotations provide a parametrization of the rotation by three Euler angles, otherwise known as roll, pitch and yaw angles.

It should be clear from this description of the decomposition algorithm how similar QR, QL and LQ factorizations may be carried out. Furthermore, the algorithm is easily generalized to higher dimensions.

A4.1.2 Householder matrices and QR decomposition

For matrices of larger dimension, the QR decomposition is more efficiently carried out using Householder matrices. The symmetric matrix

$$\mathtt{H}_{\mathbf{v}} = \mathtt{I} - 2\mathbf{v}\mathbf{v}^{\mathsf{T}}/\mathbf{v}^{\mathsf{T}}\mathbf{v} \tag{A4.2}$$

has the property that $\mathtt{H}_{\mathbf{v}}^{\mathsf{T}}\mathtt{H}_{\mathbf{v}} = \mathtt{I}$, and so $\mathtt{H}_{\mathbf{v}}$ is orthogonal.

Let $\mathbf{e}_1$ be the vector $(1, 0, \ldots, 0)^{\mathsf{T}}$, and let $\mathbf{x}$ be any vector. Let $\mathbf{v} = \mathbf{x} \pm \|\mathbf{x}\|\mathbf{e}_1$. One easily verifies that $\mathtt{H}_{\mathbf{v}}\mathbf{x} = \mp\|\mathbf{x}\|\mathbf{e}_1$; thus $\mathtt{H}_{\mathbf{v}}$ is an orthogonal matrix that transforms the vector $\mathbf{x}$ to a multiple of $\mathbf{e}_1$. Geometrically $\mathtt{H}_{\mathbf{v}}$ is a reflection in the plane perpendicular to $\mathbf{v}$, and $\mathbf{v} = \mathbf{x} \pm \|\mathbf{x}\|\mathbf{e}_1$ is a vector that bisects $\mathbf{x}$ and $\pm\|\mathbf{x}\|\mathbf{e}_1$. Thus reflection in the $\mathbf{v}$ direction takes $\mathbf{x}$ to $\mp\|\mathbf{x}\|\mathbf{e}_1$. For reasons of stability, the sign ambiguity in defining $\mathbf{v}$ should be resolved by setting

$$\mathbf{v} = \mathbf{x} + \operatorname{sign}(x_1)\|\mathbf{x}\|\mathbf{e}_1. \tag{A4.3}$$

If $\mathtt{A}$ is a matrix, $\mathbf{x}$ is the first column of $\mathtt{A}$, and $\mathbf{v}$ is defined by (A4.3), then forming the product $\mathtt{H}_{\mathbf{v}}\mathtt{A}$ will clear out the first column of the matrix, replacing the first column by $(\|\mathbf{x}\|, 0, 0, \ldots, 0)^{\mathsf{T}}$. One continues left multiplication by orthogonal Householder matrices to clear out the below-diagonal part of the matrix $\mathtt{A}$. In this way, one finds that eventually $\mathtt{Q}\mathtt{A} = \mathtt{R}$, where $\mathtt{Q}$ is a product of orthogonal matrices and $\mathtt{R}$ is an upper-triangular matrix. Therefore, one has $\mathtt{A} = \mathtt{Q}^{\mathsf{T}}\mathtt{R}$. This is the QR decomposition of the matrix $\mathtt{A}$.

When multiplying by Householder matrices it is inefficient to form the Householder matrix explicitly. Multiplication by a vector $\mathbf{a}$ may be carried out most efficiently as

$$\mathtt{H}_{\mathbf{v}}\mathbf{a} = (\mathtt{I} - 2\mathbf{v}\mathbf{v}^{\mathsf{T}}/\mathbf{v}^{\mathsf{T}}\mathbf{v})\mathbf{a} = \mathbf{a} - 2\mathbf{v}(\mathbf{v}^{\mathsf{T}}\mathbf{a})/\mathbf{v}\mathbf{v}^{\mathsf{T}} \tag{A4.4}$$

and the same holds for multiplication by a matrix $\mathtt{A}$. For more about Householder matrices and the QR decomposition, the reader is referred to [Golub-89].

Note. In the QR or RQ decomposition, R refers to an upper-triangular matrix and Q refers to an orthogonal matrix. In the notation used elsewhere in this book, $\mathtt{R}$ refers usually to a rotation (hence orthogonal) matrix.

A4.2 Symmetric and skew-symmetric matrices

Symmetric and skew-symmetric matrices play an important role in this book. A matrix is called *symmetric* if $\mathtt{A}^{\mathsf{T}} = \mathtt{A}$ and *skew-symmetric* if $\mathtt{A}^{\mathsf{T}} = -\mathtt{A}$. The eigenvalue decompositions of these matrices are summarized in the following result.

Result A4.1. Eigenvalue decomposition.

(i) *If* $\mathtt{A}$ *is a real symmetric matrix, then* $\mathtt{A}$ *can be decomposed as* $\mathtt{A} = \mathtt{U}\mathtt{D}\mathtt{U}^{\mathsf{T}}$, *where* $\mathtt{U}$ *is an orthogonal matrix and* $\mathtt{D}$ *is a real diagonal matrix. Thus, a real symmetric matrix has real eigenvalues, and the eigenvectors are orthogonal.*

(ii) *If* $\mathtt{S}$ *is real and skew-symmetric, then* $\mathtt{S} = \mathtt{U}\mathtt{B}\mathtt{U}^{\mathsf{T}}$ *where* $\mathtt{B}$ *is a block-diagonal*

matrix of the form $\mathrm{diag}(a_1\mathtt{Z}, a_2\mathtt{Z}, \ldots, a_m\mathtt{Z}, 0, \ldots, 0)$, *where* $\mathtt{Z} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. *The eigenvectors of* $\mathtt{S}$ *are all purely imaginary, and a skew-symmetric matrix of odd order is singular.*

A proof of this result is given in [Golub-89].

Jacobi's method. In general, eigenvalue extraction from arbitrary matrices is a difficult numerical problem. For real symmetric matrices however, a very stable method exists: Jacobi's method. An implementation of this algorithm is given in [Press-88].

Cross products

Of particular interest are 3×3 skew-symmetric matrices. If $\mathtt{a} = (a_1, a_2, a_3)^\mathsf{T}$ is a 3-vector, then one defines a corresponding skew-symmetric matrix as follows:

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{A4.5}$$

Note that any skew-symmetric 3×3 matrix may be written in the form $[\mathbf{a}]_\times$ for a suitable vector $\mathbf{a}$. Matrix $[\mathbf{a}]_\times$ is singular, and $\mathbf{a}$ is its null-vector (right or left). Hence, a 3×3 skew-symmetric matrix is defined up to scale by its null-vector.

The cross product (or vector product) of two 3-vectors $\mathbf{a} \times \mathbf{b}$ (sometimes written $\mathbf{a} \wedge \mathbf{b}$) is the vector $(a_2b_3 - a_3b_2, a_3b_1 - a_1b_3, a_1b_2 - a_2b_1)^\mathsf{T}$. The cross product is related to skew-symmetric matrices according to

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b} = \left(\mathbf{a}^\mathsf{T}[\mathbf{b}]_\times\right)^\mathsf{T}. \tag{A4.6}$$

Cofactor and adjoint matrices. Let $\mathtt{M}$ be a square matrix. By $\mathtt{M}^*$ is meant the matrix of cofactors of $\mathtt{M}$. That is, the (ij)-th entry of the matrix $\mathtt{M}^*$ is equal to $(-1)^{i+j} \det \hat{\mathtt{M}}_{ij}$, where $\hat{\mathtt{M}}_{ij}$ is the matrix obtained from $\mathtt{M}$ by striking out the i-th row and j-th column. The transpose of the cofactor matrix $\mathtt{M}^*$ is known as the *adjoint* of $\mathtt{M}$, and denoted $\mathrm{adj}(\mathtt{M})$.

If $\mathtt{M}$ is invertible, then it is well known that

$$\mathtt{M}^* = \det(\mathtt{M})\, \mathtt{M}^{-\mathsf{T}} \tag{A4.7}$$

where $\mathtt{M}^{-\mathsf{T}}$ is the inverse transpose of $\mathtt{M}$. This formula does not hold for non-invertible matrices, but $\mathrm{adj}(\mathtt{M})\, \mathtt{M} = \mathtt{M}\, \mathrm{adj}(\mathtt{M}) = \det(\mathtt{M})\, \mathtt{I}$ is always valid.

The cofactor matrix is related to the way matrices distribute with respect to the cross product.

Lemma A4.2. *If* $\mathtt{M}$ *is any* 3×3 *matrix (invertible or not), and* $\mathbf{x}$ *and* $\mathbf{y}$ *are column vectors, then*

$$(\mathtt{M}\mathbf{x}) \times (\mathtt{M}\mathbf{y}) = \mathtt{M}^*(\mathbf{x} \times \mathbf{y}). \tag{A4.8}$$

This equation may be written as $[\mathtt{Mx}]_\times \mathtt{M} = \mathtt{M}^* [\mathbf{x}]_\times$, dropping $\mathbf{y}$ which is inessential. Now, putting $\mathbf{t} = \mathtt{Mx}$ and assuming $\mathtt{M}$ is invertible, one obtains a rule for commuting a skew-symmetric matrix $[\mathbf{t}]_\times$ with any non-singular matrix $\mathtt{M}$. One may write (A4.8) as follows.

Result A4.3. *For any vector* $\mathbf{t}$ *and non-singular matrix* $\mathtt{M}$ *one has*

$$[\mathbf{t}]_\times \mathtt{M} = \mathtt{M}^* [\mathtt{M}^{-1}\mathbf{t}]_\times = \mathtt{M}^{-\mathsf{T}} [\mathtt{M}^{-1}\mathbf{t}]_\times \textit{ (up to scale).}$$

Note (see result 9.9(*p*254)) that $[\mathbf{t}]_\times \mathtt{M}$ is the form of the fundamental matrix for a pair of cameras $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$ and $\mathtt{P}' = [\mathtt{M}|\mathbf{t}]$. The formula of result A4.3 is used in deriving alternative forms (9.2–*p*244) for the fundamental matrix.

A curious property of 3×3 skew-symmetric matrices is that up to scale, $[\mathbf{a}]_\times = [\mathbf{a}]_\times [\mathbf{a}]_\times [\mathbf{a}]_\times$ (including scale $[\mathbf{a}]_\times^3 = -\|\mathbf{a}\|^2 [\mathbf{a}]_\times$). This is easily verified, since the right hand side is clearly skew-symmetric and its null-space generator is $\mathbf{a}$. The next result follows immediately:

Result A4.4. *If* $\mathtt{F} = [\mathbf{e}']_\times \mathtt{M}$ *is a fundamental matrix (a* 3×3 *singular matrix), then* $[\mathbf{e}']_\times [\mathbf{e}']_\times \mathtt{F} = \mathtt{F}$ *(up to scale). Hence one may decompose* $\mathtt{F}$ *as* $\mathtt{F} = [\mathbf{e}']_\times \mathtt{M}$*, where* $\mathtt{M} = [\mathbf{e}']_\times \mathtt{F}$.

A4.2.1 Positive-definite symmetric matrices

The special class of real symmetric matrices that have positive real eigenvalues are called *positive-definite* symmetric matrices. We list some of the important properties of a positive-definite symmetric real matrix.

Result A4.5. Positive-definite symmetric real matrix.

(i) *A symmetric matrix* $\mathtt{A}$ *is positive-definite if and only if* $\mathbf{x}^\mathsf{T}\mathtt{A}\mathbf{x} > 0$ *for any non-zero vector* $\mathbf{x}$.

(ii) *A positive-definite symmetric matrix* $\mathtt{A}$ *may be uniquely decomposed as* $\mathtt{A} = \mathtt{K}\mathtt{K}^\mathsf{T}$ *where* $\mathtt{K}$ *is an upper-triangular real matrix with positive diagonal entries.*

Proof. The first part of this result follows almost immediately from the decomposition $\mathtt{A} = \mathtt{U}\mathtt{D}\mathtt{U}^\mathsf{T}$. As for the second part, since $\mathtt{A}$ is symmetric and positive-definite, it may be written as $\mathtt{A} = \mathtt{U}\mathtt{D}\mathtt{U}^\mathsf{T}$ where $\mathtt{D}$ is diagonal, real and positive and $\mathtt{U}$ is orthogonal. We may take the square root of $\mathtt{D}$, writing $\mathtt{D} = \mathtt{E}\mathtt{E}^\mathsf{T}$ where $\mathtt{E}$ is diagonal. Then $\mathtt{A} = \mathtt{V}\mathtt{V}^\mathsf{T}$ where $\mathtt{V} = \mathtt{U}\mathtt{E}$. The matrix $\mathtt{V}$ is not upper-triangular. However, we may apply the RQ-decomposition (section A4.1.1) to write $\mathtt{V} = \mathtt{K}\mathtt{Q}$ where $\mathtt{K}$ is upper-triangular and $\mathtt{Q}$ is orthogonal. Then $\mathtt{A} = \mathtt{V}\mathtt{V}^\mathsf{T} = \mathtt{K}\mathtt{Q}\mathtt{Q}^\mathsf{T}\mathtt{K}^\mathsf{T} = \mathtt{K}\mathtt{K}^\mathsf{T}$. This is the Cholesky factorization of $\mathtt{A}$. One may ensure that the diagonal entries of $\mathtt{K}$ are all positive by multiplying $\mathtt{K}$ on the right by a diagonal matrix with diagonal entries equal to ± 1. This will not change the product $\mathtt{K}\mathtt{K}^\mathsf{T}$.

Now, we prove uniqueness of the factorization. Specifically, if $\mathtt{K}_1$ and $\mathtt{K}_2$ are two upper-triangular matrices satisfying $\mathtt{K}_1\mathtt{K}_1^{\mathsf{T}} = \mathtt{K}_2\mathtt{K}_2^{\mathsf{T}}$ then $\mathtt{K}_2^{-1}\mathtt{K}_1 = \mathtt{K}_2^{\mathsf{T}}\mathtt{K}_1^{-\mathsf{T}}$. Since the left side of this equation is upper-triangular, and the right side is lower-triangular, they must both in fact be diagonal. Thus $\mathtt{D} = \mathtt{K}_2^{-1}\mathtt{K}_1 = \mathtt{K}_2^{\mathsf{T}}\mathtt{K}_1^{-\mathsf{T}}$. However, $\mathtt{K}_2^{-1}\mathtt{K}_1$ is the inverse transpose of $\mathtt{K}_2^{\mathsf{T}}\mathtt{K}_1^{-\mathsf{T}}$, and so $\mathtt{D}$ is equal to its own inverse transpose, and hence is a diagonal matrix with diagonal entries equal to ± 1. If both $\mathtt{K}_1$ and $\mathtt{K}_2$ have positive diagonal entries then $\mathtt{D} = \mathtt{I}$, and $\mathtt{K}_1 = \mathtt{K}_2$. □

The above proof gives a constructive method for computing the Cholesky factorization. There is, however, a very simple and more efficient direct method for computing the Cholesky factorization. See [Press-88] for an implementation.

A4.3 Representations of rotation matrices

A4.3.1 Rotations in n-dimensions

Given a matrix $\mathtt{T}$, we define $e^{\mathtt{T}}$ to be the sum of the series

$$e^{\mathtt{T}} = \mathtt{I} + \mathtt{T} + \mathtt{T}^2/2! + \ldots + \mathtt{T}^k/k! + \ldots$$

This series converges absolutely for all values of $\mathtt{T}$. Now, we consider powers of a skew-symmetric matrix. According to result A4.1 a skew-symmetric matrix can be written as $\mathtt{S} = \mathtt{U}\mathtt{B}\mathtt{U}^{\mathsf{T}}$ where $\mathtt{B}$ is block diagonal of the form $\mathtt{B} = \mathrm{diag}(a_1\mathtt{Z}, a_2\mathtt{Z}, \ldots, a_m\mathtt{Z}, 0, \ldots, 0)$, matrix $\mathtt{U}$ is orthogonal, and $\mathtt{Z}^2 = -\mathtt{I}_{2\times 2}$. We observe that the powers of $\mathtt{Z}$ are

$$\mathtt{Z}^2 = -\mathtt{I}\ ;\ \mathtt{Z}^3 = -\mathtt{Z}\ ;\ \mathtt{Z}^4 = \mathtt{I}$$

and so on. Thus,

$$e^{\mathtt{Z}} = \mathtt{I} + \mathtt{Z} - \mathtt{I}/2! - \mathtt{Z}/3! + \ldots \quad = \quad \cos(1)\mathtt{I} + \sin(1)\mathtt{Z} \quad = \quad \mathtt{R}_{2\times 2}(1)$$

Where $\mathtt{R}_{2\times 2}(1)$ means the 2×2 matrix representing a rotation through 1 radian. More generally,

$$e^{a\mathtt{Z}} = \cos(a)\mathtt{I} + \sin(a)\mathtt{Z} = \mathtt{R}_{2\times 2}(a)\ .$$

With $\mathtt{S} = \mathtt{U}\mathtt{B}\mathtt{U}^{\mathsf{T}}$ as above, it now follows that

$$e^{\mathtt{S}} \quad = \quad \mathtt{U}e^{\mathtt{B}}\mathtt{U}^{\mathsf{T}} \quad = \quad \mathtt{U}\,\mathrm{diag}(\mathtt{R}(a_1), \mathtt{R}(a_2), \ldots, \mathtt{R}(a_m), 1, \ldots, 1)\,\mathtt{U}^{\mathsf{T}}$$

Thus $e^{\mathtt{S}}$ is a rotation matrix. On the other hand, any rotation matrix may be written in the block-diagonal form $\mathtt{U}\,\mathrm{diag}(\mathtt{R}(a_1), \mathtt{R}(a_2), \ldots, \mathtt{R}(a_m), 1, \ldots, 1)\,\mathtt{U}^{\mathsf{T}}$, and it follows that the matrices $e^{\mathtt{S}}$ where $\mathtt{S}$ is an $n \times n$ skew-symmetric are exactly the set of n-dimensional rotation matrices.

A4.3.2 Rotations in 3-dimensions

If $\mathbf{t}$ is a 3-vector, then $[\mathbf{t}]_\times$ is a skew-symmetric matrix, and any 3×3 skew-symmetric matrix is of this form. Consequently, any 3-dimensional rotation can be written as $e^{[\mathbf{t}]_\times}$. We seek to describe the rotation $e^{[\mathbf{t}]_\times}$.

Let $[\mathbf{t}]_\times = \mathtt{U}\,\mathrm{diag}(a\mathtt{Z}, 0)\,\mathtt{U}^\mathsf{T}$. Then by matching the Frobenius norms of the matrices on both sides, we see that $a = \|\mathbf{t}\|$. Thus,

$$e^{[\mathbf{t}]_\times} = \mathtt{U}\,\mathrm{diag}(\mathtt{R}(\|\mathbf{t}\|), 1)\,\mathtt{U}^\mathsf{T} \quad .$$

Thus, $e^{[\mathbf{t}]_\times}$ represents a rotation through an angle $\|\mathbf{t}\|$. It is easily verified that $\mathbf{u}_3$, the 3-rd column of $\mathtt{U}$ is the eigenvector of $\mathtt{U}\,\mathrm{diag}(\mathtt{R}, 1)\,\mathtt{U}^\mathsf{T}$ with unit eigenvector, hence the axis of rotation. However, $[\mathbf{t}]_\times \mathbf{u}_3 = \mathtt{U}\mathrm{diag}(a\mathtt{Z}, 0)\mathtt{U}^\mathsf{T}\mathbf{u}_3 = \mathtt{U}\,\mathrm{diag}(a\mathtt{Z}, 0)(0, 0, 1)^\mathsf{T} = 0$. Since $\mathbf{u}_3$ is the generator of the null space of $[\mathbf{t}]_\times$, it must be that $\mathbf{u}_3$ is a unit vector in the direction of $\mathbf{t}$. We have shown

Result A4.6. *The matrix $e^{[\mathbf{t}]_\times}$ is a rotation matrix representing a rotation through an angle $\|\mathbf{t}\|$ about the axis represented by the vector $\mathbf{t}$.*

This representation of a rotation is called the *angle-axis* representation.

We may write a specific formula for the rotation matrix corresponding to $e^{[\mathbf{t}]_\times}$. We observe that $[\mathbf{t}]_\times^3 = -\|\mathbf{t}\|^2\,[\mathbf{t}]_\times = -\|\mathbf{t}\|^3\,[\hat{\mathbf{t}}]_\times$, where $\hat{\mathbf{t}}$ represents a unit vector in the direction $\mathbf{t}$. Then, with $\mathrm{sinc}(\theta)$ representing $\sin(\theta)/\theta$, we have

$$\begin{aligned}
e^{[\mathbf{t}]_\times} &= \mathtt{I} + [\mathbf{t}]_\times + [\mathbf{t}]_\times^2/2! + [\mathbf{t}]_\times^3/3! + [\mathbf{t}]_\times^4/4! + \ldots \\
&= \mathtt{I} + \|\mathbf{t}\|\,[\hat{\mathbf{t}}]_\times + \|\mathbf{t}\|^2\,[\hat{\mathbf{t}}]_\times^2/2! - \|\mathbf{t}\|^3\,[\hat{\mathbf{t}}]_\times/3! - \|\mathbf{t}\|^4\,[\hat{\mathbf{t}}]_\times^2/4! + \ldots \\
&= \mathtt{I} + \sin\|\mathbf{t}\|\,[\hat{\mathbf{t}}]_\times + (1 - \cos\|\mathbf{t}\|)\,[\hat{\mathbf{t}}]_\times^2 \\
&= \mathtt{I} + \mathrm{sinc}\|\mathbf{t}\|\,[\mathbf{t}]_\times + \frac{1 - \cos\|\mathbf{t}\|}{\|\mathbf{t}\|^2}\,[\mathbf{t}]_\times^2 \qquad \text{(A4.9)} \\
&= \cos\|\mathbf{t}\|\mathtt{I} + \mathrm{sinc}\|\mathbf{t}\|\,[\mathbf{t}]_\times + \frac{1 - \cos\|\mathbf{t}\|}{\|\mathbf{t}\|^2}\,\mathbf{t}\mathbf{t}^\mathsf{T}
\end{aligned}$$

where the last line follows from the identity $[\mathbf{t}]_\times^2 = \mathbf{t}\mathbf{t}^\mathsf{T} - \|\mathbf{t}\|^2\mathtt{I}$.

Some properties of these representations:

(i) Extraction of the axis and rotation angle from a rotation matrix $\mathtt{R}$ is just a little tricky. The unit rotation axis $\mathbf{v}$ can be found as the eigenvector corresponding to the unit eigenvalue – that is by solving $(\mathtt{R} - \mathtt{I})\mathbf{v} = \mathbf{0}$. Next, it is easily seen from (A4.9) that the rotation angle ϕ satisfies

$$\begin{aligned}
2\cos(\phi) &= (\mathrm{trace}(\mathtt{R}) - 1) \\
2\sin(\phi)\mathbf{v} &= (\mathtt{R}_{32} - \mathtt{R}_{23}, \mathtt{R}_{13} - \mathtt{R}_{31}, \mathtt{R}_{21} - \mathtt{R}_{12})^\mathsf{T}. \qquad \text{(A4.10)}
\end{aligned}$$

Writing this second equation as $2\sin(\phi)\mathbf{v} = \hat{\mathbf{v}}$, we can then compute $2\sin(\phi) = \mathbf{v}^\mathsf{T}\hat{\mathbf{v}}$. Now, the angle ϕ can be computed from $\sin(\phi)$ and $\cos(\phi)$ using a two-argument $\arctan$ function (such as the C-language function atan2(y, x)).

It has often been written that ϕ can be computed directly from (A4.10) using $\arccos$ or $\arcsin$. However, this method is not numerically accurate, and fails to find the axis when $\phi = \pi$.

(ii) To apply a rotation $\mathtt{R}(\mathbf{t})$ to some vector $\mathbf{x}$, it is not necessary to construct the

matrix representation of $\mathbf{t}$. In fact

$$\begin{aligned} \mathtt{R}(\mathbf{t})\mathbf{x} &= \left(\mathtt{I} + \operatorname{sinc}\|\mathbf{t}\|[\mathbf{t}]_\times + \frac{1-\cos\|\mathbf{t}\|}{\|\mathbf{t}\|^2}[\mathbf{t}]_\times^2\right)\mathbf{x} \\ &= \mathbf{x} + \operatorname{sinc}\|\mathbf{t}\|\mathbf{t}\times\mathbf{x} + \frac{1-\cos\|\mathbf{t}\|}{\|\mathbf{t}\|^2}\mathbf{t}\times(\mathbf{t}\times\mathbf{x}) \end{aligned} \quad \text{(A4.11)}$$

(iii) If $\mathbf{t}$ is written as $\mathbf{t} = \theta\hat{\mathbf{t}}$, where $\hat{\mathbf{t}}$ is a unit vector in the direction of the axis, and $\theta = \|\mathbf{t}\|$ is the angle of rotation, then (A4.9) is equivalent to the Rodrigues formula for a rotation matrix:

$$\mathtt{R}(\theta,\hat{\mathbf{t}}) = \mathtt{I} + \sin\theta[\hat{\mathbf{t}}]_\times + (1-\cos\theta)[\hat{\mathbf{t}}]_\times^2 \quad \text{(A4.12)}$$

A4.3.3 Quaternions

Three-dimensional rotations may also be represented by unit quaternions. A unit quaternion is a 4-vector and may be written in the form $\mathbf{q} = (\mathbf{v}\sin(\theta/2), \cos(\theta/2))^\mathsf{T}$, as may indeed any unit 4-vector. Such a quaternion represents a rotation about the vector $\mathbf{v}$ through the angle θ. This is a 2-to-1 representation in that both $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation. To check this, note that $-\mathbf{q} = (\mathbf{v}\sin(\theta/2+\pi), \cos(\theta/2+\pi))^\mathsf{T}$, which represents a rotation through $\theta + 2\pi = \theta$.

The relationship between the angle-axis representation of a rotation and the quaternion representation is easily determined. Given a vector $\mathbf{t}$, the angle-axis representation of a rotation, the corresponding quaternion is easily seen to be

$$\mathbf{t} \leftrightarrow \mathbf{q} = (\operatorname{sinc}(\|\mathbf{t}\|/2)\mathbf{t},\ \ \cos(\|\mathbf{t}\|/2))^\mathsf{T}$$

A4.4 Singular value decomposition

The singular value decomposition (SVD) is one of the most useful matrix decompositions, particularly for numerical computations. Its most common application is in the solution of over-determined systems of equations.

Given a square matrix $\mathtt{A}$, the SVD is a factorization of $\mathtt{A}$ as $\mathtt{A} = \mathtt{U}\mathtt{D}\mathtt{V}^\mathsf{T}$, where $\mathtt{U}$ and $\mathtt{V}$ are orthogonal matrices, and $\mathtt{D}$ is a diagonal matrix with non-negative entries. Note that it is conventional to write $\mathtt{V}^\mathsf{T}$ instead of $\mathtt{V}$ in this decomposition. The decomposition may be carried out in such a way that the diagonal entries of $\mathtt{D}$ are in descending order, and we will assume that this is always done. Thus a circumlocutory phrase such as "the column of $\mathtt{V}$ corresponding to the smallest singular value" is replaced by "the last column of $\mathtt{V}$."

The SVD also exists for non-square matrices $\mathtt{A}$. Of most interest is the case where $\mathtt{A}$ has more rows than columns. Specifically, let $\mathtt{A}$ be an $m \times n$ matrix with $m \geq n$. In this case $\mathtt{A}$ may be factored as $\mathtt{A} = \mathtt{U}\mathtt{D}\mathtt{V}^\mathsf{T}$ where $\mathtt{U}$ is an $m \times n$ matrix with orthogonal columns, $\mathtt{D}$ is an $n \times n$ diagonal matrix and $\mathtt{V}$ is an $n \times n$ orthogonal matrix. The fact that $\mathtt{U}$ has orthogonal columns means that $\mathtt{U}^\mathsf{T}\mathtt{U} = \mathtt{I}_{n\times n}$. Furthermore $\mathtt{U}$ has the norm-preserving property that $\|\mathtt{U}\mathbf{x}\| = \|\mathbf{x}\|$ for any vector $\mathbf{x}$, as one readily verifies. On the other hand, $\mathtt{U}\mathtt{U}^\mathsf{T}$ is in general not the identity unless $m = n$.

Not surprisingly, one can also define a singular value decomposition for matrices

with more columns than rows, but generally this will not be of interest to us. Instead on the occasional instances in this book where we need to take the singular value decomposition of a matrix A with $m < n$, it is appropriate to extend A by adding rows of zeros to obtain a square matrix, and then take the SVD of this resulting matrix. Usually this will be done without special remark.

Common implementations of the SVD, such as the one in [Press-88], assume that $m \geq n$. Since in this case the matrix U has the same dimension $m \times n$ as the input, matrix A may be overwritten by the output matrix U.

Implementation of the SVD. A description of the singular value decomposition algorithm, or a proof of its existence, is not given in this book. For a description of how the algorithm works, the reader is referred to [Golub-89]. A practical implementation of the SVD is given in [Press-88]. However, the implementation of the SVD given in the first edition of "Numerical Recipes in C" can sometimes give incorrect results. The version of the algorithm given in the second edition [Press-88] of "Numerical Recipes in C" corrects mistakes in the earlier version.

Singular values and eigenvalues. The diagonal entries of matrix D in the SVD are non-negative. These entries are known as the singular values of the matrix A. They are not the same thing as eigenvalues. To see the connection of the singular values of A with eigenvalues, we start with $\mathtt{A} = \mathtt{UDV}^\mathsf{T}$. From this it follows that $\mathtt{A}^\mathsf{T}\mathtt{A} = \mathtt{VDU}^\mathsf{T}\mathtt{UDV}^\mathsf{T} = \mathtt{VD}^2\mathtt{V}^\mathsf{T}$. Since V is orthogonal, $\mathtt{V}^\mathsf{T} = \mathtt{V}^{-1}$, and so $\mathtt{A}^\mathsf{T}\mathtt{A} = \mathtt{VD}^2\mathtt{V}^{-1}$. This is the defining equation for eigenvalues, indicating that the entries of $\mathtt{D}^2$ are the eigenvalues of $\mathtt{A}^\mathsf{T}\mathtt{A}$ and the columns of V are the eigenvectors of $\mathtt{A}^\mathsf{T}\mathtt{A}$. In short, the singular values of A are the square-roots of the eigenvalues of $\mathtt{A}^\mathsf{T}\mathtt{A}$.

Note, $\mathtt{A}^\mathsf{T}\mathtt{A}$ is symmetric and positive-semi-definite (see section A4.2.1 above), so its eigenvalues are real and non-negative. Consequently, singular values are real and non-negative.

Computational complexity of the SVD

The computational complexity of the SVD depends on how much information needs to be returned. For instance in algorithm A5.4 to be considered later, the solution to the problem is the last column of the matrix V in the SVD. The matrix U is not used, and does not need to be computed. On the other hand, algorithm A5.1 of section A5.1 requires the complete SVD to be computed. For systems of equations with many more rows than columns, the extra effort required to compute the matrix U is substantial.

Approximate numbers of floating-point operations (flops) required to compute the SVD of an $m \times n$ matrix are given in [Golub-89]. To find matrices U, V and D, a total of $4m^2n + 8mn^2 + 9n^3$ flops are needed. However, if only the matrices V and D are required, then only $4mn^2 + 8n^3$ flops are required. This is an important distinction, since this latter expression does not contain any term in m^2. Specifically, the number of operations required to compute U varies as the square of m, the number of rows. On the other hand, the complexity of computing D and V is linear in m. For cases where there are many more rows than columns, therefore, it is important (supposing

computation time is an issue) to avoid computing the matrix $\mathtt{U}$ unless it is needed. To illustrate this point we consider the DLT algorithm for camera resection, described in chapter 7. In this algorithm a 3×4 camera matrix is computed from a set of n 3D to 2D point matches. The solution involves using algorithm A5.4, to solve a set of equations $\mathtt{A}\mathbf{p} = \mathbf{0}$ where $\mathtt{A}$ is a $2n \times 12$ matrix. The solution vector $\mathbf{p}$ is the last column of the matrix $\mathtt{V}$ in an SVD, $\mathtt{A} = \mathtt{UDV}^\mathsf{T}$. Thus the matrix $\mathtt{U}$ is not required. Table A4.1 gives the total number of flops for carrying out the SVD for six (the minimum number), 100 or 1000 point correspondences.

# points	# equations per point	# equations (m)	#unknowns (n)	# operations not computing U	# operations computing U
6	2	12	12	20,736	36,288
100	2	200	12	129,024	2,165,952
1000	2	2000	12	1,165,824	194,319,552

Table A4.1. *Comparison of the number of flops required to compute the SVD of a matrix of size $m \times n$, for varying values of m and for $n = 12$. Note that the computational complexity increases sub-linearly in the number of equations when* U *is not computed. On the other hand, the extra computational burden of computing* U *is very large, especially for large numbers of equations.*

Further reading. Two invaluable text books for this area are [Golub-89] and [Lutkepohl-96].