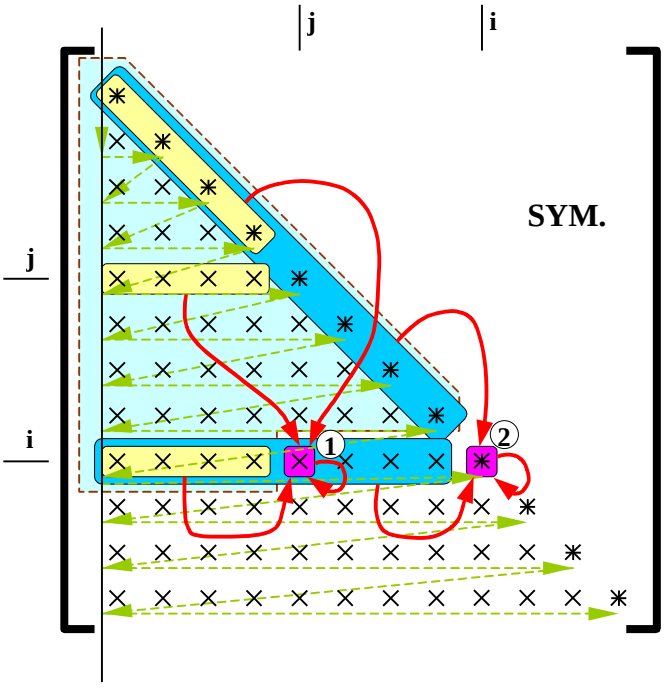
Matrix Decompositions With Implementation Remarks

For use in Nonlinear Optimization Software

Igor Grešovnik
May 2012

Electronic Book
Revision 4.2
(revision 1: August 2008)



This report is public.

Contents:

1 INTRODUCTION

This document introduces the basics of matrix computations and especially matrix decompositions, which are also used in the Investigative Optimization Library (IOptLib).

The document was created by putting together a number of notes in MS Word created by the myself for memorizing a number of results from linear algebra that are relevant for optimization and other fields of numerical analysis.

I have created a large portion of these notes during my undergraduate study when I have practicing numerical software development for training, hobby and to earn some additional money that allowed me to cover more than just the basic needs, e.g. a bicycle or a one week trip to Paris. Another package of the notes was created when finishing the graduate degree on physics, when I have also put some hand written notes form the "Mathematics 1" subject into electronic form.

The collection was further supplemented during my Ph.D. study at the University of Swansea in the U.K. Then, one day when looking at all those scattered documents that contained short notes on individual topics, I realized that it would be much easier for myself to have all the material in a single document. At that time, the MS Word has also advanced enough to allow creation of longer documents with lots of formulas and images[1]. So I have reserved a weekend and put all the documents in a single files, arranged titles and references and formatted the document in such a way that it acquired a relatively readable form.

After that, I was still adding notes from time to time when I needed something to memorize, and this was much easier to do in a settled document than in a group of scattered files. However, many portions of the script are not in the form I could recommend for studying. It must be understood that some of these notes emerged from yellowed old notes which represented just a very condensed hints on individual topics, and there are many portions that I didn't touch (except for formatting) since they were put in electronic forms. But when I need to look back at something and I notice that this is very awkwardly written, I usually take some time to correct that. In this way I hope that the script will one day become a useful and easily digestible source of reference information for people like myself, who are not truly experts in linear algebra but use it in everyday life.

Author of this script, *Igor Grešovnik*

[1] For a long time, at least over the decade, this was the major obstacle for writing scientific and engineering documents in MS Word, and this is probably the main reason that Tech remained the desktop publishing program of choice in many scientific and not so rare engineering environments. Because for people who did not have problems with learning programming languages and other computer related skills, it was far easier to work with a program where you couldn't see the result of your work instantly than with a program that crashed for nothing every couple of minutes.

2 BASICS OF MATRIX THEORY

2.1 Basic terms

2.1.1.1 Inner product

The standard inner product in $\mathbb{R}^n$ is defined by

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} \overline{x_i}\, y_i \ . \tag{1}$$

2.1.1.2 Matrix basic terms

A *matrix* is a rectangular table of numbers or other quantities that can be added or multiplied. Most common is use of matrices that are defined over the real (real matrices) or complex (complex matrices) field. We will usually denote matrices by large bold letters, where dimensions (the number of rows and columns) will sometimes be specified in subscript, e.g. $\mathbf{A}_{m\times n}$ denotes a matrix with *m* rows and *n* columns. A matrix where one of the dimensions equals 1 is often called a vector, and is interpreted as an element of a coordinate space. A $1\times n$ matrix is called a row vector and a $n\times 1$ matrix is called a column vector. Matrix components will usually be denoted by the corresponding letter, but not written in bold, with indices in the subscript, e.g. A_{ij} will mean the element in row *i* and column *j* of the matrix **A**. Sometimes components will be denoted by small letters, i.e. a_{ij}. A *square matrix* is those that has the same number of rows and columns.

Matrix is *normal* if it commutes with its conjugate transpose:

$$\mathbf{A}^* \mathbf{A} = \mathbf{A}\, \mathbf{A}^* \ . \tag{2}$$

Adjoint matrix of a *n*-by-*m* matrix **A** is its conjugate transpose:

$$\left(\mathbf{A}^*\right)_{i,j} = \overline{\mathbf{A}_{j,i}} \tag{3}$$

where subscripts denote the *i,j*-th component of the matrix, with $1 \le i \le n$ and $1 \le j \le m$. This is expressed as

$$\mathbf{A}^* = \bar{\mathbf{A}}^T \ . \tag{4}$$

The *conjugate transpose of a matrix or its adjoint matrix* is also denoted by $\mathbf{A}^H$ (H coming from *Hermitian conjugate*)

2.1.1.3 Hermitian and unitary matrices

Hermitian or *self-adjoint matrix* is a square matrix that is equal to its conjugate transpose, i.e.

$$\mathbf{A} = \mathbf{A}^* . \tag{5}$$

Hermitian matrices are *normal*, and the finite dimensional spectral theorem applies, which means that every Hermitian matrix can be diagonalized by a unitary matrix. Eigenvalues of every Hermitian matrix are real and eigenvectors with distinct eigenvalues are orthogonal.

The sum of two Hermitian matrices is Hermitian, and inverse of an invertible Hermitian matrix is also Hermitian. The product of two Hermitian matrices is Hermitian only if they commute.

Anti-Hermitian or skew Hermitian matrix is a matrix for which $\mathbf{A}^* = -\mathbf{A}$. Entries in the main diagonal are pure imaginary. The same is true for eigenvalues.

Unitary matrix is a square matrix that satisfies the condition

$$\mathbf{U}^* \mathbf{U} = \mathbf{U}\mathbf{U}^* = \mathbf{I} , \tag{6}$$

where **I** is the identity matrix. A matrix is unitary if it has an inverse which is equal to its conjugate transpose. Unitary matrix preserves the standard inner product on $\mathbb{C}^n$,

$$\langle \mathbf{U}\mathbf{x}, \mathbf{U}\mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle \tag{7}$$

If **A** is a square matrix then the following conditions are equivalent:

1. **A** is unitary
2. $\mathbf{A}^*$ is unitary
3. The columns of **A** form an *orthonormal basis* of $\mathbb{C}^n$ with respect to the standard inner product on $\mathbb{C}^n$.

4. **A** is an *isometry* (i.e. distance preserving isomorphism[1]) with respect to the norm for this inner product.

A unitary matrices are calles special if its determinant is 1.

Unitary matrices are *normal* , therefore the spectral theorem applies to them. Every unitary matrix **U** has decomposition of the form

$$\mathbf{U} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^{*}, \tag{8}$$

where **V** is unitary and $\mathbf{\Sigma}$ is diagonal and unitary.

Real unitary matrices are orthogonal matrices and real hermitian matrices are symmetric matrices.

2.2 Polar Decomposition

Polar decomposition is a matrix decomposition of the form

$$\mathbf{A} = \mathbf{U}\mathbf{P}\ , \tag{9}$$

where **U** is an unitary matrix and **P** is a positive-semidefinite Hermitean matrix. The decomposition always exists, and if **A** is invertible then the decomposition is unique and **P** positive definite. The matrix **P** is given by

$$\mathbf{P} = \sqrt{\mathbf{A}^{*}\mathbf{A}}\,, \tag{10}$$

where $\mathbf{A}^{*}$ is a conjugate transpose of **A**. A positive definite Hermitean matrix has a unique positive square root. The matrix U is then given as

$$\mathbf{U} = \mathbf{A}\mathbf{P}^{-1}\ . \tag{11}$$

In terms of the singular value decomposition, we have

$$\begin{aligned} \mathbf{P} &= \mathbf{V}\mathbf{\Sigma}\mathbf{V}^{*} \\ \mathbf{U} &= \mathbf{W}\mathbf{V}^{*} \end{aligned}\ , \tag{12}$$

[1] Isomorphism is a bijective map *f* such that both *f* and f^{1} are *homomorphisms* (i.e structure preserving maps - $\phi(u \bullet v) = \phi(u) \circ \phi(v)$)

which confirms that **P** is positive-semidefinite and **U** unitary.

Determinant of **A** can be expressed, according to (9), as

$$\det \mathbf{A} = \det \mathbf{P} \det \mathbf{U} = r e^{i\theta} , \tag{13}$$

which gives the corresponding polar decomposition of the determinant of **A**, since $\det \mathbf{P} = r = \left|\det \mathbf{A}\right|$ and $\det \mathbf{U} = e^{i\theta}$.

A matrix can also be decomposed as

$$\mathbf{A} = \mathbf{P}' \mathbf{U} , \tag{14}$$

where **U** is the same as before and

$$\mathbf{P}' = \mathbf{U}\, \mathbf{P}\, \mathbf{U}^{-1} = \sqrt{\mathbf{A}\mathbf{A}^*} = \mathbf{W}\boldsymbol{\Sigma}\mathbf{W}^* \tag{15}$$

The matrix **A** is normal if and only if $\mathbf{P}' = \mathbf{P}$. Then $\mathbf{U}\boldsymbol{\Sigma} = \boldsymbol{\Sigma}\mathbf{U}$ and it is possible to diagonalize **U** with a unitary similarity matrix **S** that commutes with $\boldsymbol{\Sigma}$, giving $\mathbf{S}\mathbf{U}\mathbf{S}^* = \boldsymbol{\Phi}^{-1}$, where $\boldsymbol{\Phi}$ is a diagonal unitary matrix of phase $e^{i\phi}$. Putting, $\mathbf{Q} = \mathbf{V}\mathbf{S}^*$, the polar decomposition can be re-written as

$$\mathbf{A} = \left(\mathbf{Q}\boldsymbol{\Phi}\mathbf{Q}^*\right)\left(\mathbf{Q}\boldsymbol{\Sigma}\mathbf{Q}^*\right) , \tag{16}$$

so **A** also has a spectral decomposition

$$\mathbf{A} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^* \tag{17}$$

with complex eigenvalues so that $\boldsymbol{\Lambda}\boldsymbol{\Lambda}^* = \boldsymbol{\Sigma}^2$ and a unitary matrix of complex eigenvectors **Q**.

2.3 Eigenvalues and eigenvectors

Let $\mathbf{A} \in \mathbb{C}^{n\times n}$. $\lambda \in \mathbb{C}$ is called an eigenvalue of **A** if there exists a non-zero (non-null) vector $x \in \mathbb{C}^n$ such that $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$. **x** is an *eigenvector* associated with λ. The set of eigenvalues of **A** is called the *spectrum* of **A** denoted by $\sigma(\mathbf{A})$. **x** and **y** are the *left and right eigenvectors* of **A** associated with λ, respectively, if

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}, \quad \mathbf{y}^H\mathbf{A} = \lambda\mathbf{y}^H \ . \tag{18}$$

The eigenvalue corresponding to the eigenvector **x** can be determined by computing the *Raileigh quotient* $\lambda = \mathbf{x}^H\mathbf{A}\mathbf{x}/\left(\mathbf{x}^H\mathbf{x}\right)$. Eigenvalue λ is solution of a *characteristic equation*

$$p_A(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I}) = 0 \ , \tag{19}$$

where $p_A(\lambda)$ is the *characteristic polynomial*. Because this is a polynomial of degree *n*, there exist *n* eigenvalues of **A** that are not necessarily distinct. The following is true:

$$\begin{aligned} &\det(\mathbf{A}) = \prod_{i=1}^{n}\lambda_i \ , \ tr(\mathbf{A}) = \sum_{i=1}^{n}\lambda_i \\ &\sigma(\mathbf{A}) = \sigma\left(\mathbf{A}^T\right), \ \sigma\left(\mathbf{A}^H\right) = \sigma\left(\bar{\mathbf{A}}\right) \end{aligned} \tag{20}$$

From the first relation we see that a matrix is *singular* if it has at least one zero eigenvalue, since $p_A(0) = \det\mathbf{A} = \prod_{i=1}^{n}\lambda_i$.

If **A** has *real entries*, then coefficients of $p_A(\lambda)$ are real and therefore *complex eigenvalues occur in complex conjugate pairs*.

The maximum module (absolute value) of the eigenvalues of **A** is called the *spectral radius* of **A** and denoted by

$$\rho(\mathbf{A}) = \max_{\lambda\in\sigma(\mathbf{A})}|\lambda| \tag{21}$$

λ is an eigenvalue of **A** iff $\bar{\lambda}$ is an eigenvalue of $\mathbf{A}^H$. Therefore $\rho(\mathbf{A}) = \rho\left(\mathbf{A}^H\right)$. Also $\rho(\alpha\mathbf{A}) = |\alpha|\rho(\mathbf{A})$ and $\rho\left(\mathbf{A}^k\right) = \left(\rho(\mathbf{A})\right)^k \ \forall k \in \square$

Let **A** be a block triangular matrix

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1k} \\ \mathbf{0} & \mathbf{A}_{22} & \cdots & \mathbf{A}_{2k} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{kk} \end{bmatrix}. \tag{22}$$

Because $p_A(\lambda) = p_{A1}(\lambda) + p_{A22}(\lambda) + ... + p_{Akk}(\lambda)$, the spectrum of **A** is union of the spectra of each diagonal block. As a consequence, if **A** is triangular then its eigenvalues are its diagonal elements.

For each eigenvalue of matrix **A** the set of eigenvectors associated with λ, together with the null vector, identifies a subsubspace in $\mathbb{R}^n$ which is called the *eignespace* associated with λ and corresponds by definition to $\ker(\mathbf{A}-\lambda\mathbf{I})$. The dimension of the eigenspace is

$$\dim\left(\ker\left(\mathbf{A}-\lambda\mathbf{I}\right)\right)=n-rank\left(\mathbf{A}-\lambda\mathbf{I}\right), \tag{23}$$

and is called *geometric multiplicity* of the eigenvalue λ. It can not be greater than the *algebraic multiplicity* of λ, which is the multiplicity of λ as a root of the characteristic polynomial. Eignevalues that have geometric multiplicity strictly less than algebraic multiplicity are called *defective*.

The eigenspace associated with an eigenvalue of the matrix **A** is invariant with respect to **A**in the sense of the following definition:

A subspace $S\in\mathbb{R}^n$ is called invariant with respect to a square matrix **A** if $\mathbf{A}S\subseteq S$, where **A***S* is *S* transformed through **A**.

2.4 Spectral theorem

2.5 Similarity transformations, Schur Decomposition & Spectral Decomposition

Similarity transformation is transformation of the form

$$\mathbf{A}\rightarrow\mathbf{C}^{-1}\mathbf{A}\mathbf{C} \tag{24}$$

where **C** is a square nonsingular matrix having the same order as **A**. We say that matrixes **A** and $\mathbf{C}^{-1}\mathbf{A}\mathbf{C}$ are similar matrices. If **C** is unitary then matrices are unitary similar. Two similar matrices have the same spectrum and the same characteristic polynomial.

We can easily check that if $(\lambda,\mathbf{x})$ are eigenvalue-eigenvector pair for **A** then $\left(\lambda,\mathbf{C}^{-1}\mathbf{x}\right)$ are eigenvalue-eigenvector pair for $\mathbf{C}^{-1}\mathbf{A}\mathbf{C}$ since $\left(\mathbf{C}^{-1}\mathbf{A}\mathbf{C}\right)\mathbf{C}^{-1}\mathbf{x}=\mathbf{C}^{-1}\mathbf{A}\mathbf{x}=\lambda\mathbf{C}^{-1}\mathbf{x}$

Matrices $\mathbf{AB}$ and $\mathbf{BA}$, $\mathbf{A} \in \mathbb{C}^{n \times m}$ and $\mathbf{B} \in \mathbb{C}^{m \times n}$ are not similar, but satisfy the following property:

$$\sigma(\mathbf{AB}) \setminus \{0\} = \sigma(\mathbf{BA}) \setminus \{\mathbf{0}\}, \tag{25}$$

i.e. $\mathbf{AB}$ and $\mathbf{BA}$ share the same spectrum apart from null eigenvalues, so that $\rho(\mathbf{AB}) = \rho(\mathbf{BA})$.

2.5.1 Schur Decomposition theorem

Given $\mathbf{A} \in \mathbb{C}^{n \times n}$, there exists an *unitary* matrix $\mathbf{U}$ such that

$$\mathbf{U}^{-1}\mathbf{A}\mathbf{U} = \mathbf{U}^{H}\mathbf{A}\mathbf{U} = \begin{bmatrix} \lambda_1 & b_{12} & \cdots & b_{1n} \\ 0 & \lambda_2 & & b_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} = \mathbf{T} \tag{26}$$

where λ_i are eigenvalues of $\mathbf{A}$.

It follows that every square matrix $\mathbf{A}$ is unitary smilar to an upper triantular matrix. Matrices $\mathbf{T}$ and $\mathbf{U}$ are not necessarily unique.

Among the others, the Schur decomposition theorem gives rise to the following results:

Every Hermitian matrix is unitary similar to a diagonal real matrix. Every Schur decomposition of a Hermitan matrix is diagonal:

$$\mathbf{A}^{H} = \mathbf{A} \Rightarrow \ \mathbf{U}^{-1}\mathbf{A}\mathbf{U} = \Lambda = diag(\lambda_1, ..., \lambda_n) \tag{27}$$

It turns that $\mathbf{AU} = \mathbf{U}\Lambda$, i.e. $\mathbf{A}\mathbf{u}_i = \lambda_i \mathbf{u}_i$ for $i = 1, \ldots, n$, i.e. column vectors of $\mathbf{U}$ are eigenvectors of $\mathbf{A}$. Since eigenvectors are mutually orthogonal, a hermitean matrix has a set of orthogonal vectors that generate the whole space $\mathbb{C}^n$. Furthermore, it can be shown that a matrix $\mathbf{A}$ of order n is similar to a diagonal matrix $\mathbf{D}$ iff the eigenvectors of $\mathbf{A}$ form a basis in $\mathbb{C}^n$.

Spectral decomposition (or eigendecompozition) of a normal matrix:

A matrix $\mathbf{A} \in \mathbb{C}^{n \times n}$ is normal iff it is unitary similar to a diagonal matrix. As a consequence, a normal matrix admits the following *spectral decomposition*:

$$\mathbf{A} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^H = \sum_{i=1}^{n} \lambda_i \, \mathbf{u}_i \, \mathbf{u}_i^H \,, \tag{28}$$

with **U** being unitary and $\boldsymbol{\Lambda}$ diagonal.

Let **A** and **B** be two normal and commutative matrices. Then the generic eigenvalue μ_i of $\mathbf{A}+\mathbf{B}$ is given by the sum $\mu_i = \lambda_i + \xi_i$, where λ_i and μ_i are eigenvalues of **A** and **B** associated with the same eigenvector.

There are nonsymmetric matrices that are unitary similar to diagonal matrices, but they are not unitary similar.

2.5.1.1 Canonical Jordan form

Schur decomposition cnan be improved as follows.

Let **A** be any square matrix. Then, there exists a nonsingular matrix **X** which transforms **A** into a block diagonal matrix **J** such that

$$\mathbf{X}^{-1} \mathbf{A} \mathbf{X} = \mathbf{J} = diag\left(\mathbf{J}_{k_1}(\lambda_1), \mathbf{J}_{k_2}(\lambda_2), ..., \mathbf{J}_{k_n}(\lambda_n) \right), \tag{29}$$

which is called a *Jordan canonical form*. λ_j are eigenvalues of **A** and $J_k(\lambda) \in \mathbb{C}^{k \times k}$ is a Jordan block of the form $J_1(\lambda) = \lambda$ for *k*=1 and

$$J_k(\lambda) = \begin{bmatrix} \lambda & 1 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & \vdots & \vdots \\ \vdots & \ddots & \ddots & 1 & 0 \\ \vdots & & \ddots & \lambda & 1 \\ 0 & \cdots & \cdots & 0 & \lambda \end{bmatrix}. \tag{30}$$

If an eigenvalue is defective, the size of the corresponding Jordan block is greater than one. Therefore, a matrix can be diagonalized by a similarity transform iff it is nondefective. Nondeffective matrices are therefore called *diagonalizable*. Normal matrices are diagonalizable.

2.6 Singular Value Decomposition

The Singular value decomposition (SVD) can be seen as generalization of the spectral theorem to arbitrary matrices (not necessarily square). For a *m*-by-*n* matrix over the field *K* of real or complex numbers **A** there exists a factorization of the form

$$\mathbf{A}_{(m\times n)} = \mathbf{U}_{(m\times m)}\,\mathbf{\Sigma}_{(m\times n)}\,\mathbf{V}^{T}_{(n\times n)}$$
$$\mathbf{U}^T\,\mathbf{U} = \mathbf{I}\,;\,\mathbf{V}^T\,\mathbf{V} = \mathbf{I}\,;\,\Sigma_{ii} = \sigma_i\,;\;\Sigma_{ij} = 0\,;\,i \neq j \qquad (31)$$

where **U** is a *m*-by-*m* unitary matrix over *K*, $\mathbf{\Sigma}$ is *m*-by-*n* trapezoid matrix with nonnegative numbers on the diagonal and zeros off the diagonal and **V** is a *n*-by-*n* unitary matrix over *K*. Such decomposition is called a *singular-value decomposition* of **M**. A common convention is to order the values $\sigma_{ii} = \mathbf{\Sigma}_{ii}$ in non-increasing fashion. In this case, the diagonal matrix is uniquely determined by **M** (but the matrices **U** and **V** are not).

The matrix **V** contains a set of orthonormal "input" vectors for "analysis" basis vector directions for **A**, the matrix **U** contains a set of orthonormal "output" basis vector directions for **A**, the matrix $\mathbf{\Sigma}$ contains the singular values, which can be understood as scalar "gain controls" by which each corresponding input is multiplied to give a corresponding output.

A non-negative number σ is a *singular value* of **A** only if there exists unit-length vectors **u** in K^m and **v** in K^n such that

$$\mathbf{A}\,\mathbf{v} = \sigma\,\mathbf{u} \;\; and \;\; \mathbf{A}^{*}\mathbf{u} = \sigma\,\mathbf{v}\;. \qquad (32)$$

The vectors **u** and **v** are called **left-singular** and **right-singular vectors** for σ, respectively. In any singular value decomposition, the diagonal entries of $\mathbf{\Sigma}$ are equal to the singular values of **A**. The columns of **U** and **V** are left and right singular vectors for the corresponding singular values. The theorem states that

- An $m\times n$ matrix **A** has at most *p*=min(*m*,*n*) distinct singular values.
  - It is always possible to find a unitary basis for K^m consisting of left-singular vectors of **A**.

- It is always possible to find a unitary basis for k^n consisting of right-singular vectors fo **A**.

A singular value for which we can find two left (or right) singular vectors that are not linearly dependent is called *degenerate*. Non-degenerate singular values always have unique left and right singular vectors, up to multiplication by a unit phase factor $e^{i\phi}$ (for the real case, up to sign). Consequently, if all singular values of *M* are non-degenerate and non-zero, then its singular value decomposition is unique, up to multiplication of a column of **U** by a unit phase factor and simultaneous multiplication of the corresponding column of **V** by the same unit phase factor.

Degenerate singular values have non-unique singular vectors (by definition). If $\mathbf{u}_1$ and $\mathbf{u}_2$ are two left-singular vectors which both correspond to the singular value σ, then any normalized linear combination of the two vectors is also a left singular vector corresponding to σ (and similar for the right singular vectors). Singular decomposition is then not unique.

When **A** is a Hermitian matrix which is positive semi-definite (all eigenvalues are real and non-negative), then the singular values and singular vectors of **A** coincide with the eigenvalues and eiganvectors of **M**,

$$\mathbf{A} = \mathbf{V}\,\mathbf{\Lambda}\,\mathbf{V}^* \ . \tag{33}$$

More generally, given a SVD of **M**, the following two relations hold:

$$\begin{aligned} \mathbf{A}^*\mathbf{A} &= \mathbf{V}\,\mathbf{\Sigma}^*\,\mathbf{U}^*\,\mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^* = \mathbf{V}\left(\mathbf{\Sigma}^*\,\mathbf{\Sigma}\right)\mathbf{V}^* \\ \mathbf{A}\,\mathbf{A}^* &= \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^*\,\mathbf{V}\,\mathbf{\Sigma}^*\,\mathbf{U}^* = \mathbf{U}\left(\mathbf{\Sigma}\,\mathbf{\Sigma}^*\right)\mathbf{U}^* \end{aligned} \tag{34}$$

The right hand sides of these relations describe the eigenvalue decompositions of the left hand sides. Consequently, the squares of the non-zero singular values of **A** are equal to the non-zero eigenvalues of $\mathbf{A}^*\mathbf{A}$. The columns of **U** (left singular vectors) are eigenvectors for $\mathbf{A}\,\mathbf{A}^*$ and the columns of **V** (right singular vectors) are eigenvectors of $\mathbf{A}^*\mathbf{A}$.

2.7 Examples in physics

2.7.1 Finite deformation tensor

Finite deformation tensors are used when the deformation of a body is sufficiently large that the assumptions in small strain theory are not valid.

We denote **X** the position vector of a particle in the initial (undeformed) state of a body relative to some coordinate basis. The position of the particle in the deformed state is denoted **x**. If $d\mathbf{X}$ is a line segment that joins two nearby particles in the undeformed state and $d\mathbf{x}$ is the line segment joining the same two particles in the deformed state, then the linear transformation between the two segments is given by

$$d\,\mathbf{x} = \mathbf{F}\,d\,\mathbf{X} \ . \tag{35}$$

The quantity **F** is called the *deformation gradient*. and is given by:

$$\mathbf{F} = \nabla_X\,\mathbf{x} = \frac{\partial\,\mathbf{x}}{\partial\,\mathbf{X}} \ , \tag{36}$$

or:

$$F_{ij} = \frac{\partial\,x_i}{\partial\,X_j} \ . \tag{37}$$

F is a second order tensor and contains information about the stretch and rotation of the body.

Polar decoposition:

$$\mathbf{F} = \mathbf{R}\,\mathbf{U} = \mathbf{V}\,\mathbf{R} \ , \tag{38}$$

where **R** is a proper orthogonal tensor representing rotation, and **U** and **V** are positive definite symmetric tensors that represent stretches. **U** is called the right stretch and **V** is called the left stretch tensor.

Spectral decompositions of **U** and **V** are

$$\mathbf{U} = \sum_{i=1}^{3} \lambda_i\,\mathbf{N}_i \otimes \mathbf{N}_j \tag{39}$$

and

$$\mathbf{V} = \sum_{i=1}^{3} \lambda_i\,\mathbf{n}_i \otimes \mathbf{n}_j \ , \tag{40}$$

where λ_i are the principal stretches, and $\mathbf{N}_i$ and $\mathbf{n}_i$ are the directions of the principal stretches (principal directions). The principal directions are related by

$$\mathbf{n}_i = \mathbf{R}\,\mathbf{N}_i \ . \tag{41}$$

Rotation independent deformation measures are introduced because rotation should not induce any stress in a deformable body. Rotation is excluded by multiplying **R** by its transpose. In this way we obtain the **Right Cauchy-Green Tensor**

$$\begin{aligned} &\mathbf{C}=\mathbf{F}^T\mathbf{F} = \mathbf{U}^T\mathbf{U} \\ &C_{ij} = \sum_{k=1}^{3} \frac{\partial x_k}{\partial X_i}\frac{\partial x_k}{\partial X_j} \end{aligned} \tag{42}$$

and the left Cauchy-Green tensor

$$\begin{aligned} &\mathbf{B} = \mathbf{F}\mathbf{F}^T = \mathbf{V}\mathbf{V}^T = \mathbf{V}^2 \\ &B_{ij} = \sum_{i=1}^{3} \frac{\partial x_i}{\partial X_k}\frac{\partial x_j}{\partial X_k} \quad . \end{aligned} \tag{43}$$

The spectral decompositions are

$$\begin{aligned} &\mathbf{C} = \sum_{i=1}^{3} \lambda_i^2\, \mathbf{N}_i \otimes \mathbf{N}_j \\ &\mathbf{B} = \sum_{i=1}^{3} \lambda_i^2\, \mathbf{n}_i \otimes \mathbf{n}_j \end{aligned} \tag{44}$$

3 LU DECOMPOSITION

L – lover triangular

U – upper triangular with 1 on diagonal

$$L_{ij} = \begin{Bmatrix} L_{ij}; & i > j \\ 1; & i = j \\ 0; & i < j \end{Bmatrix} \qquad (3.0.$$

$$U_{ij} = \begin{Bmatrix} U_{ij}; & i \le j \\ 0; & i > j \end{Bmatrix} \qquad (3.0.$$

$$\left[\mathbf{LU}\right]_{ij} = \sum_{k=1}^{n} L_{ik} U_{ki}$$

$$\begin{Bmatrix} i < j: & \underbrace{\sum_{k=1}^{i-1} L_{ik} U_{kj}}_{k<i<j} + \underbrace{L_{ii}}_{=1} U_{ij} + \underbrace{\sum_{k=i+1}^{j-1} \underbrace{L_{ik}}_{=0} U_{kj}}_{i<k<j} + L_{ij} \underbrace{U_{jj}}_{=0} + \underbrace{\sum_{k=j+1}^{n} \underbrace{L_{ik}}_{=0} \underbrace{U_{kj}}_{=0}}_{i<j<k} \\ i = j: & \underbrace{\sum_{k=1}^{i-1} L_{ik} U_{kj}}_{k<i=j} + L_{ii} \underbrace{U_{ii}}_{=0} + \underbrace{\sum_{k=i+1}^{n} \underbrace{L_{ik}}_{=0} \underbrace{U_{kj}}_{=0}}_{i=j<k} \\ i > j: & \underbrace{\sum_{k=1}^{j-1} L_{ik} U_{kj}}_{k<j<i} + L_{ij} \underbrace{U_{jj}}_{=0} + \underbrace{\sum_{k=j+1}^{i-1} L_{ik} \underbrace{U_{kj}}_{=0}}_{j<k<i} + L_{ii} \underbrace{U_{ij}}_{=0} + \underbrace{\sum_{k=i+1}^{n} \underbrace{L_{ik}}_{=0} \underbrace{U_{kj}}_{=0}}_{j<i<k} \end{Bmatrix}$$

$$\left[\mathbf{LU}\right]_{ij} = \begin{Bmatrix} i < j: & \sum_{k=1}^{i} L_{ik} U_{kj} \\ i = j: & \sum_{k=1}^{i-1} L_{ik} U_{kj} \\ i > j: & \sum_{k=1}^{j-1} L_{ik} U_{kj} \end{Bmatrix} \qquad (3.45)$$

4 LDL^T DECOMPOSITION

4.1 Introduction: General on Matrix Products

$$\left[\mathbf{LU}\right]_{ij} = \sum_{k=1}^{n} L_{ik} U_{ki}$$

$$\left\{\begin{array}{ll}
i < j: & \underbrace{\sum_{k=1}^{i-1} L_{ik} U_{kj}}_{k<i<j} + L_{ii} U_{ij} + \underbrace{\sum_{k=i+1}^{j-1} \underbrace{L_{ik}}_{=0} U_{kj}}_{i<k<j} + L_{ij} U_{jj} + \underbrace{\sum_{k=j+1}^{n} \underbrace{L_{ik}}_{=0} \underbrace{U_{kj}}_{=0}}_{i<j<k} \\
i = j: & \underbrace{\sum_{k=1}^{i-1} L_{ik} U_{kj}}_{k<i=j} + L_{ii} U_{ii} + \underbrace{\sum_{k=i+1}^{n} \underbrace{L_{ik}}_{=0} \underbrace{U_{kj}}_{=0}}_{i=j<k} \\
i > j: & \underbrace{\sum_{k=1}^{j-1} L_{ik} U_{kj}}_{k<j<i} + L_{ij} U_{jj} + \underbrace{\sum_{k=j+1}^{i-1} L_{ik} U_{kj}}_{j<k<i} + L_{ii} \underbrace{U_{ij}}_{=0} + \underbrace{\sum_{k=i+1}^{n} \underbrace{L_{ik}}_{=0} \underbrace{U_{kj}}_{=0}}_{j<i<k}
\end{array}\right\}$$

$$\left[\mathbf{LU}\right]_{ij} = \sum_{k=1}^{\min(i,j)} L_{ik} U_{kj} \qquad (4.0.$$

$$a_{ij} = \left[\mathbf{LDL}^T\right]_{ij} = \sum_{k=1}^{n} l_{ik}\left[\mathbf{DL}^T\right]_{kj} = \sum_{k=1}^{n} l_{ik}\left[\mathbf{L}^T\right]_{kj} d_{kk} = \sum_{k=1}^{n} l_{ik} l_{jk} d_{kk} =$$

$$= \sum_{k=1}^{\min(i,j)} l_{ik} l_{jk} d_{kk} = \begin{cases} i < j: & \sum_{k=1}^{i} l_{ik} l_{jk} d_{kk} \\ i = j: & \sum_{k=1}^{i-1} l_{ik} l_{ik} d_{kk} + \underbrace{l_{ii}}_{=1}^{2} d_{ii} \\ i > j: & \sum_{k=1}^{j} l_{ik} l_{jk} d_{kk} \end{cases} = \begin{cases} i < j: & \sum_{k=1}^{i-1} l_{ik} l_{jk} d_{kk} + \underbrace{l_{ii}}_{=1} l_{ji} d_{ii} \\ i = j: & \sum_{k=1}^{i-1} l_{ik}^2 d_{kk} + \underbrace{l_{ii}}_{=1}^{2} d_{ii} \\ i > j: & \sum_{k=1}^{j-1} l_{ik} l_{jk} d_{kk} + l_{ij} \underbrace{l_{jj}}_{=1} d_{jj} \end{cases}$$

$$i < j:\ a_{ij} = \sum_{k=1}^{i-1} l_{ik} l_{jk} d_{kk} + \underbrace{l_{ii}}_{=1} l_{ji} d_{ii} \qquad a_{ji} = \sum_{k=1}^{i-1} l_{jk} l_{ik} d_{kk} + l_{ji} \underbrace{l_{ii}}_{=1} d_{ii}$$

$$\Rightarrow a_{ij} = a_{ji} \Rightarrow \mathbf{A} = \mathbf{A}^T$$

- LDL^T decomposition is possible **only for symmetric matrices**.

Let **A** be any square matrix and **D** any diagonal matrix. Then

$$\left[\mathbf{AD}\right]_{ij} = A_{ij} D_{jj}, \tag{4.0.}$$

$$\left[DA\right]_{ij} = A_{ij} D_{ii} \tag{4.0.}$$

and

$$\left[\mathbf{D}^2\right]_{ij} = \left[\mathbf{DD}^T\right]_{ij} = \left[\mathbf{D}^T\mathbf{D}\right]_{ij} = d_{ij}^{\ 2}$$

$$\left[\mathbf{AA}^T\right]_{ij} = \sum_{i=1}^{n} a_{ik}\left[\mathbf{A}^T\right]_{kj} = \sum_{k=1}^{n} a_{ik} a_{jk}$$

$$\left[\mathbf{A}^T\mathbf{A}\right]_{ij} = \sum_{i=1}^{n}\left[\mathbf{A}^T\right]_{ik} a_{kj} = \sum_{k=1}^{n} a_{ki} a_{kj}$$

$$\left(\mathbf{AA}^T\right)^T = \left(\mathbf{A}^T\right)^T \mathbf{A}^T = \mathbf{AA}^T$$

from which follows that $\mathbf{AA}^T$ is symmetric.

4.1.1 Eigenvalues and Eigenvectors

Let's have a square matrix **A** and let's there exist some non-zero vector **v** so that

$$\mathbf{Av} = \lambda\mathbf{v}, \qquad (4.0.$$

where λ is a scalar. Then **v** is an eigenvector and λ an eigenvalue of matrix a. We also see that if **v** is an eigenvector of a, then also $k\mathbf{v}$ is eigenvector of **A**, where *k* is any scalar different than zero. An invertable square matrix has *n* eigenvectors where *n* is matrix dimension (i.e. number of rows or columns):

$$\mathbf{Av}_i = \lambda_i\mathbf{v}_i, \ i = 1,...,n. \qquad (4.0.$$

Positive definitenes:

Matrix **A** is positive-definite, if

$$\mathbf{u}^T\mathbf{Au} > 0 \ \forall \ \mathbf{u} \neq 0. \qquad (4.0.$$

Matrix is positive-semidefinite, if

$$\mathbf{u}^T\mathbf{Au} \geq 0 \ \forall \ \mathbf{u} \neq 0, \qquad (4.0.$$

negative-definite, if

$$\mathbf{u}^T\mathbf{Au} < 0 \ \forall \ \mathbf{u} \neq 0, \qquad (4.0.$$

negative-semidefinite if the relationship includes allows the equality sign, and indefinite otherwise.

Matrix is positive-definite, if and only if all its eigenvalues are greater than zero (which can be seen if the matrix components are written in its eigensystem, i.e. a cooordinate system the basis of which form matrix eigenvectors.

If a square matrix **A** is invertable (i.e. has a full rank), then matrices $\mathbf{AA}^T$, $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}^2$ are all positive definite.

Let's denote such matrix **B**. The relation (4.0 can be verified if we write vector **u** as a linear combination of eigenvectors of **A**.

4.1.2 General (asymmetric) matrices

4.1.3 Symmetric Real Matrices

For symmetric matrices,

$$\mathbf{A} = \mathbf{A}^T \qquad (4.0.$$

We can perform the LDLT decomposition of such a matrix to a product of a lower triangular matrix **L**, diagonal matrix **D** and transpose of **L**:

$$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^T \qquad (4.0.$$

where **L** is a general lower triangular matrix with unit diagonal elements,

$$\mathbf{L}_{ij} = \begin{Bmatrix} l_{ij}; & i \geq j \\ 0; & i < j \end{Bmatrix}, \qquad (4.0.$$

and **D** is diagonal:

$$\mathbf{D}_{ij} = \begin{Bmatrix} d_{ij}; & i = j \\ 0; & i \neq j \end{Bmatrix} \qquad (4.0.$$

$$a_{ij} = \left[\mathbf{LDL}^T\right]_{ij} = \sum_{k=1}^{\min(i,j)} l_{ik}\left[\mathbf{DL}^T\right]_{kj} = \sum_{k=1}^{\min(i,j)} l_{ik}\left[\mathbf{L}^T\right]_{kj} d_{kk} = \sum_{k=1}^{\min(i,j)} l_{ik} l_{jk} d_{kk} =$$

$$= \begin{Bmatrix} i < j: & \sum_{k=1}^{i-1} l_{ik} l_{jk} d_{kk} + l_{ii} l_{ji} d_{ii} \\ i = j: & \sum_{k=1}^{i-1} l_{ik}{}^2 d_{kk} + l_{ii}{}^2 d_{ii} \\ i > j: & \sum_{k=1}^{j-1} l_{ik} l_{jk} d_{kk} + l_{ij} l_{jj} d_{jj} \end{Bmatrix}$$

$$i < j:\ a_{ij} = \sum_{k=1}^{i} l_{ik} l_{jk} d_{kk} \qquad a_{ji} = \sum_{k=1}^{i} l_{l_{jk} l_{ik} d_{kk}}$$

$$\Rightarrow a_{ij} = a_{ji} \Rightarrow \mathbf{A} = \mathbf{A}^T$$

- LDL^T decomposition is possible **only for symmetric matrices**.

$$a_{11} = d_{11}$$

$$a_{12} = l_{21} d_{11} + l_{21} d_{11}$$

$$a_{21} = l_{21} d_{11} + l_{21} d_{11}$$

4.2 LDLT Decomposition

$$\mathbf{A} = \mathbf{A}^T \qquad (4.0.$$

We can decompose symmetric matrices in the form

$$\mathbf{A} = \mathbf{LDL}^T \qquad (4.0.$$

where **L** is a lower triangular matrix with unit diagonal elements,

$$\mathbf{L} = \begin{Bmatrix} L_{ij}; & i > j \\ 1; & i = j \\ 0; & i < j \end{Bmatrix}, \tag{4.0.}$$

and **D** is diagonal:

$$\mathbf{D}_{ij} = \begin{Bmatrix} D_{ij}; & i = j \\ 0; & i \neq j \end{Bmatrix} \tag{4.0.}$$

$$a_{ij} = \left[\mathbf{L}\mathbf{D}\mathbf{L}^T\right]_{ij} = \sum_{k=1}^{\min(i,j)} l_{ik}\left[\mathbf{D}\mathbf{L}^T\right]_{kj} = \sum_{k=1}^{\min(i,j)} l_{ik}\left[\mathbf{L}^T\right]_{kj} d_{kk} = \sum_{k=1}^{\min(i,j)} l_{ik} l_{jk} d_{kk} =$$

$$= \begin{Bmatrix} i \geq j: & \sum_{k=1}^{j} l_{ik} l_{jk} d_{kk} \\ i < j: & a_{ji} \end{Bmatrix}$$

$$l_{ii} = 1 \ \forall i$$

$$i < j: \ a_{ij} = \sum_{k=1}^{i} l_{ik} l_{jk} d_{kk} \qquad a_{ji} = \sum_{k=1}^{i} l_{l_{jk} l_{ik} d_{kk}}$$

$$\Rightarrow a_{ij} = a_{ji} \Rightarrow \mathbf{A} = \mathbf{A}^T$$

$$a_{11} = \underbrace{l_{11} l_{11}}_{=1} \underline{\underline{d_{11}}}$$

$$a_{21} = \underline{\underline{l_{21}}} \underbrace{l_{11}}_{=1} d_{11}$$

$$a_{22} = l_{21} l_{21} d_{11} + \underbrace{l_{22} l_{22}}_{=1} \underline{\underline{d_{22}}}$$

$$a_{31} = \underline{\underline{l_{31}}} \underbrace{l_{11}}_{=1} d_{11}$$

$$a_{32} = l_{31} l_{21} d_{11} + \underline{\underline{l_{32}}} \underbrace{l_{22}}_{=1} d_{22}$$

$$a_{33} = l_{31} l_{31} d_{11} + l_{32} l_{32} d_{22} + \underbrace{l_{33} l_{33}}_{=1} \underline{\underline{d_{33}}}$$

$$a_{41} = \underline{\underline{l_{41}}} \underbrace{l_{11}}_{=1} d_{11}$$

$$a_{42} = l_{41} l_{21} d_{11} + \underline{\underline{l_{42}}} \underbrace{l_{22}}_{=1} d_{22}$$

$$a_{43} = l_{41} l_{31} d_{11} + l_{42} l_{32} d_{22} + \underline{\underline{l_{43}}} \underbrace{l_{33}}_{=1} d_{33}$$

$$a_{44} = l_{41} l_{41} d_{11} + l_{42} l_{42} d_{22} + l_{43} l_{43} d_{33} + \underbrace{l_{44} l_{44}}_{=1} \underline{\underline{d_{44}}}$$

4.2.1 LDLT Decomposition Using Lower Triangle

A general formula is then

$$
\begin{aligned}
&i = 1, \ldots, n: \\
&l_{ij} = \frac{1}{d_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} d_{kk} \right), \quad j = 1, \ldots, i-1 \\
&d_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik}{}^2 d_{kk}
\end{aligned}
\tag{4.0.}
$$

Instead by lines we can also evaluate factors by columns:

$$
\begin{aligned}
&i = 1, \ldots, n: \\
&d_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik}{}^2 d_{kk} \\
&l_{ji} = \frac{1}{d_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk} d_{kk} \right), \quad j = i+1, i+2, \ldots, n
\end{aligned}
\tag{4.0.}
$$

(note that $a_{ij} = a_{ji}$ since **A** is symmetric).

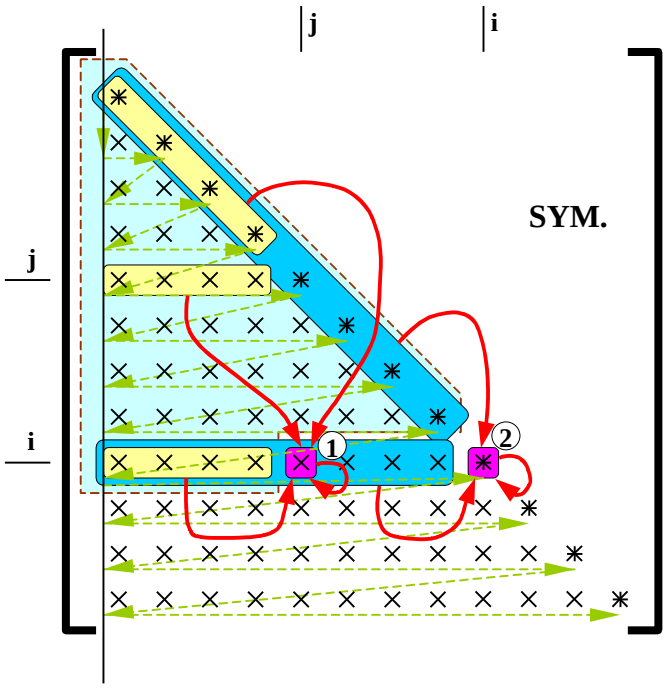


Figure 4.1: Evaluation of elements of **D** and **L** (i.e. the lower triangle) of the LDLT decomposition from lower triangle of the original matrix according to equation (4.0.

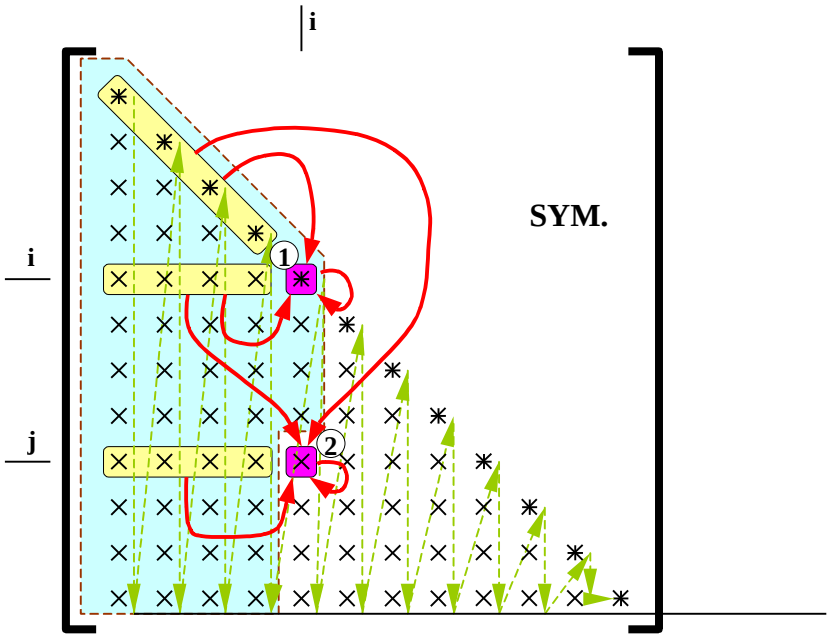


Figure 4.2: Evaluation of elements of **D** and **L** (i.e. the lower triangle) of the LDLT decomposition from lower triangle of the original matrix according to equation (4.0.

4.2.2 LDLT Decomposition Using Only Upper Triangle:

Equations can be obtained from (4.0 and (4.0 taking into account relations $a_{ij} = a_{ji}$ and $u_{ij} = l_{ji}$, where $\mathbf{U} = \mathbf{L}^T$ is transpose of the lower triangular factor (i..e the upper triangular factor) of the decomposition. Formulas are then (derived from (4.0:

$$
\begin{aligned}
&i = 1, \ldots, n: \\
&u_{ji} = \frac{1}{d_{jj}} \left(a_{ji} - \sum_{k=1}^{j-1} u_{ki} u_{kj} d_{kk} \right), \quad j = 1, \ldots, i-1 \\
&d_{ii} = a_{ii} - \sum_{k=1}^{i-1} u_{ki}{}^2 d_{kk}
\end{aligned}
\tag{4.0.}
$$

or (derived from (4.0):

$$
\begin{aligned}
&i = 1, \ldots, n: \\
&d_{ii} = a_{ii} - \sum_{k=1}^{i-1} u_{ki}{}^2 d_{kk} \\
&u_{ij} = \frac{1}{d_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj} d_{kk} \right), \quad j = i+1, i+2, \ldots, n
\end{aligned}
\tag{4.0.}
$$

Evaluation order for terms in both equations is shown graphically in Figure 4.4 and Figure 6.1.

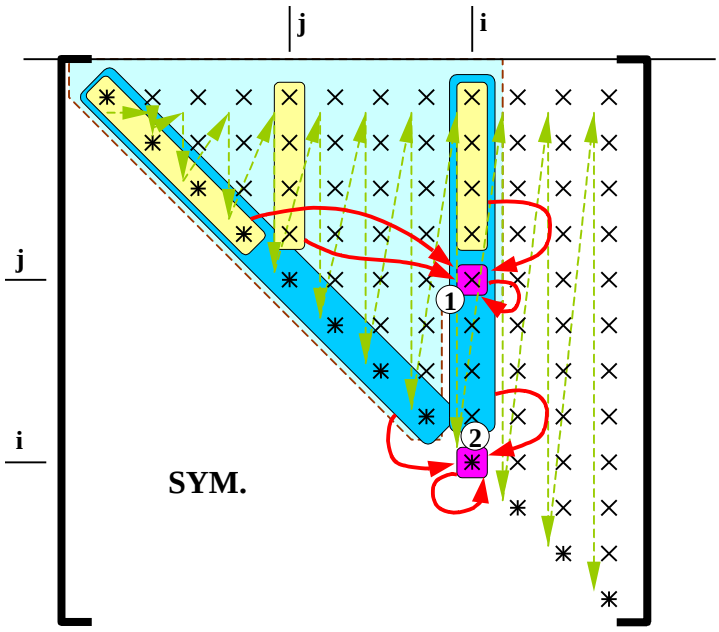


Figure 4.3: Evaluation of elements of **D** and $\mathbf{L}^T$ (i.e. the upper triangle) of the LDLT decomposition from upper triangle of the original matrix according to equation (4.0).

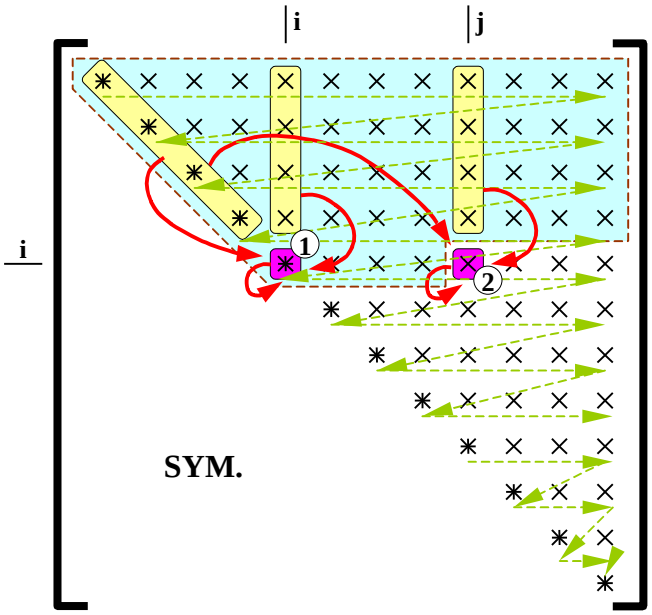


Figure 4.4: Evaluation of elements of **D** and $\mathbf{L}^T$ (i.e. the upper triangle) of the LDLT decomposition from upper triangle of the original matrix according to equation (4.0).

5 CHOLESKY DECOMPOSITION

This decomposition is of form

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T, \tag{5.0.}$$

where **L** is a lower triangular matrix

$$\mathbf{L} = \begin{Bmatrix} l_{ij}; & i \geq j \\ 0; & i < j \end{Bmatrix}. \tag{5.0.}$$

Note that diagonal elements of **L** are in general different than 1, unlike in the LDL^T decomposition. **A** must be a symmetric positive definite matrix.

$$a_{ij} = \left[\mathbf{L}\mathbf{L}^T\right]_{ij} = \sum_{k=1}^{n} l_{ik}\left[\mathbf{L}^T\right]_{kj} = \sum_{k=1}^{i} l_{ik}\left[\mathbf{L}^T\right]_{kj} = \sum_{k=1}^{i} l_{ik}l_{jk} = \sum_{k=1}^{\min(i,j)} l_{ik}l_{jk} =$$

$$= \sum_{k=1}^{\min(i,j)} l_{ik}l_{jk} =$$

$$= \begin{Bmatrix} i \geq j: & \sum_{k=1}^{j} l_{ik}l_{jk} \\ i < j: & a_{ji} \end{Bmatrix}$$

$$a_{11} = \underline{\underline{l_{11} l_{11}}}$$
$$a_{21} = \underline{l_{21}} l_{11}$$
$$a_{22} = l_{21} l_{21} + \underline{\underline{l_{22} l_{22}}}$$
$$a_{31} = \underline{l_{31}} l_{11}$$
$$a_{32} = l_{31} l_{21} + \underline{l_{32}} l_{22}$$
$$a_{33} = l_{31} l_{31} + l_{32} l_{32} + \underline{\underline{l_{33} l_{33}}}$$
$$a_{41} = \underline{l_{41}} l_{11}$$
$$a_{42} = l_{41} l_{21} + \underline{l_{42}} l_{22}$$
$$a_{43} = l_{41} l_{31} + l_{42} l_{32} + \underline{l_{43}} l_{33}$$
$$a_{44} = l_{41} l_{41} + l_{42} l_{42} + l_{43} l_{43} + \underline{\underline{l_{44} l_{44}}}$$

Computing the LL^T factors:

$$l_{11} = \sqrt{a_{11}}$$
$$l_{21} = \frac{1}{l_{11}} a_{21}$$
$$l_{22} = \sqrt{a_{22} - l_{21} l_{21}}$$
$$l_{31} = \frac{1}{l_{11}} a_{31}$$
$$l_{32} = \frac{1}{l_{22}} \left(a_{32} - l_{31} l_{21} \right)$$
$$l_{33} = \sqrt{\left(a_{33} - l_{31} l_{31} - l_{32} l_{32} \right)}$$
$$l_{41} = \frac{1}{l_{11}} a_{41}$$
$$l_{42} = \frac{1}{l_{22}} \left(a_{42} - l_{41} l_{21} \right)$$
$$l_{43} = \frac{1}{l_{33}} \left(a_{43} - l_{41} l_{31} - l_{42} l_{32} \right)$$
$$l_{44} = \sqrt{a_{44} - l_{41} l_{41} - l_{42} l_{42} - l_{43} l_{43}}$$

General formula:

$$
\begin{aligned}
&i = 1, \ldots, n: \\
&l_{ij} = \frac{1}{l_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right), \quad j = 1, \ldots, i-1 \\
&l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^{\,2}}
\end{aligned}
\tag{5.0.}
$$

In this case we evaluate elements of **L** by rows, i.e. in the following order (for a 5×5 matrix):

$$
\begin{bmatrix}
1: l_{11} & 0 & 0 & 0 & 0 \\
2: l_{21} & 3: l_{22} & 0 & 0 & 0 \\
4: l_{31} & 5: l_{32} & 6: l_{33} & 0 & 0 \\
7: l_{41} & 8: l_{42} & 9: l_{43} & 10: l_{44} & 0 \\
11: l_{51} & 12: l_{52} & 13: l_{53} & 14: l_{54} & 15: l_{55}
\end{bmatrix}
$$

Instead of by rows, we can also evaluate factors by columns:

$$
\begin{aligned}
&i = 1, \ldots, n: \\
&l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^{\,2}} \\
&l_{ji} = \frac{1}{l_{ii}}\left(a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk} \right), \quad j = i+1, i+2, \ldots, n
\end{aligned}
\tag{5.46}
$$

Evaluation order is then the following for a 5×5 matrix):

$$
\begin{bmatrix}
1: l_{11} & 0 & 0 & 0 & 0 \\
2: l_{21} & 6: l_{22} & 0 & 0 & 0 \\
3: l_{31} & 7: l_{32} & 10: l_{33} & 0 & 0 \\
4: l_{41} & 8: l_{42} & 11: l_{43} & 13: l_{44} & 0 \\
5: l_{51} & 9: l_{52} & 12: l_{53} & 14: l_{54} & 15: l_{55}
\end{bmatrix}
$$

Outline of an algorithm for computing the LL^T factors:

A nice property of the algorithm is that we don't neeed to store the original matrix throuout the algorithm When a specific element of **L** is computed, it can immedially replace the corresponding element of **A** since this is not needed any more. The algorithm assumes that diagonal

and below diagonal elements of **A** are stored in the input matrix, while elements beyond diagonal do not need to be stored because **A** is symmetric.

6 GRAM-SCHMIDT (GS) ORTHOGONALIZATION

In this Section, a procedure of obtaining a set of mutually orthogonal (or orthonormal) vectors from an arbitrary set of linearly independent vectors by the process called *Gram-Schmidt orthogonalization* is described. This process can be used for orthogonal decomposition of a matrix, which includes the known QR factorization described in Section 6.2, where a matrix is factorized as a product of an orthogonal and upper triangular matrix. This can be used for calculating eigenvalues and eigenvectors of the matrix (by application of an iterative procedure known as the QR iteration) or for solution of systems of equations, which is a method of choice when the system matrix is ill conditioned.

After the QR factorization, a non-standard form of factorization (67) is described in Section 6.3, where one obtains an orthogonal matrix (whose columns are orthogonalized columns of the original matrix) expressed as a product of the original and upper triangular matrix. The factorization will not be used in practice for solution of equations (although this is possible, as described in Section 6.3.2), but is described for instructive purposes. The reader can skip this section without harm.

We have vectors $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_m$, where $m \leq n$ (n is the dimension of the vector space). We want to construct m mutually orthogonal vectors $\mathbf{q}_i$ which are obtained from the original set of vectors, i.e. are linear combinations of vectors $\mathbf{v}_i$.

6.1.1 Basic Idea

Let us have vectors $\mathbf{q}_1$ and $\mathbf{v}_2$. We want to construct a vector $\mathbf{q}_2$ which will be a linear combination of these two vectors and will be orthogonal to the vector $\mathbf{q}_1$. The most obvious way is to subtract from $\mathbf{v}_2$ its orthogonal projection on $\mathbf{q}_1$ (see Figure 6.1). This projection has direction of $\mathbf{q}_1$ and size

$$\left\| \mathbf{v}_{2\mathbf{q}_1} \right\|_2 = \left\| \mathbf{v}_2 \right\|_2 \cos\phi = \frac{< \mathbf{q}_1, \mathbf{v}_2 >}{\left\| \mathbf{q}_1 \right\|_2}, \qquad (6.0.$$

since

$$< \mathbf{q}_1, \mathbf{v}_2 >= \|\mathbf{q}_1\|_2 \|\mathbf{v}_2\|_2 \cos\phi .$$

The projection is then

$$\mathbf{v}_{2\mathbf{q}_1} = \|\mathbf{v}_{2\mathbf{q}_1}\|_2 \frac{\mathbf{q}_1}{\|\mathbf{q}_1\|_2} = \frac{< \mathbf{q}_1, \mathbf{v}_2 >}{\|\mathbf{q}_1\|_2^2} \mathbf{q}_1 = \frac{< \mathbf{q}_1, \mathbf{v}_2 >}{< \mathbf{q}_1, \mathbf{q}_1 >} \mathbf{q}_1 \qquad (6.0.$$

and the constructed orthogonal vector $\mathbf{q}_2$ is

$$\underline{\mathbf{q}_2 = \mathbf{v}_2 - \mathbf{v}_{2\mathbf{q}_1} = \mathbf{v}_2 - \frac{< \mathbf{v}_2, \mathbf{q}_1 >}{< \mathbf{q}_1, \mathbf{q}_1 >} \mathbf{q}_1} \qquad (6.0.$$

It can be easily shown that this vector is orthogonal to $\mathbf{q}_1$: dot product of the above equation with $\mathbf{q}_1$ yields

$$< \mathbf{q}_2, \mathbf{q}_1 >= \left\langle \mathbf{v}_2 - \frac{< \mathbf{v}_2, \mathbf{q}_1 >}{< \mathbf{q}_1, \mathbf{q}_1 >} \mathbf{q}_1, \mathbf{q}_1 \right\rangle = < \mathbf{v}_2, \mathbf{q}_1 > - \frac{< \mathbf{v}_2, \mathbf{q}_1 >}{< \mathbf{q}_1, \mathbf{q}_1 >} < \mathbf{q}_1, \mathbf{q}_1 >= 0 .$$

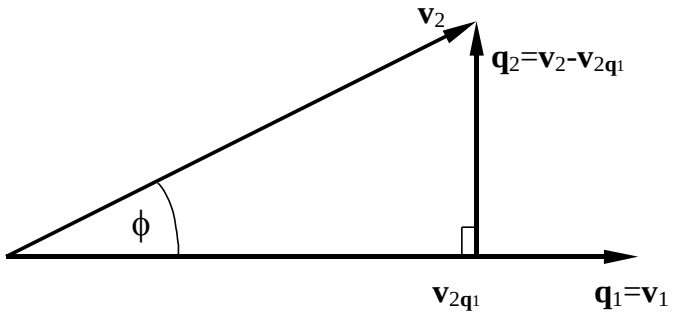


Figure 6.1: Construction of the next orthogonal vector.

The construction described above can be easily extended for a set of *n* vectors. We set

$$\underline{\underline{\mathbf{q}_1 = \mathbf{v}_1}} \qquad (6.0.$$

and the subsequent orthogonal vectors are formed as

$$\underline{\underline{\mathbf{q}_j = \mathbf{v}_j - \sum_{l=1}^{j-1} \frac{< \mathbf{v}_j, \mathbf{q}_l >}{< \mathbf{q}_l, \mathbf{q}_l >} \mathbf{q}_l, \quad j = 2, 3, ..., n .}} \qquad (6.0.$$

We can see by total induction that each $\mathbf{q}_j$ generated in this way is a linear combination of vectors $\mathbf{v}_k, k \le j$ and that it is orthogonal to all $\mathbf{q}_k, k < j$. At the end of the procedure we therefore obtain a set of *m* mutually orthogonal vectors $\mathbf{q}_i$, which are all linear combinations of original vectors and are non-zero if the original vectors are linearly independent.

6.2 Modified Gram-Schmidt Orthogonalization and QR Factorization

The Gram-Schmidt orthogonalization procedure described by (6.0 is numerically less stable because the generated vectors tend to loose their linear independency because of the numerical errors. A modified procedure is therefore used in practice, which is quite similar to the original one, but the order is a bit different.

We will use a bit different notation and will denote the original vectors by $\mathbf{a}_i$, $i = 1, ..., n$. We want to obtain an orthonormal basis of vectors $\mathbf{q}_1$ from $\mathbf{a}_1$ such that

$$\begin{aligned} \langle \mathbf{q}_i, \mathbf{q}_i \rangle &= 1 \; \forall i = 1, ..., n \\ \langle \mathbf{q}_i, \mathbf{q}_j \rangle &= 0 \; \forall i \neq n \end{aligned} . \qquad (6.0.$$

The original procedure is performed by the following steps:

$$\begin{aligned} \mathbf{q}_1 &= \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|} \\ k &= 2, ..., n: \\ \tilde{\mathbf{q}}_k &= \mathbf{a}_k - \sum_{i=1}^{k-1} \langle \mathbf{q}_i, \mathbf{a}_k \rangle \mathbf{q}_i \;, \\ \mathbf{q}_k &= \frac{\tilde{\mathbf{q}}_k}{\|\tilde{\mathbf{q}}_k\|_2} \end{aligned} . \qquad (6.0.$$

In the first part of each step, an new vector is produced from subtracting form the original vector all orthogonal projections of this vector to already computed orthogonal and normal vectors. In this way a vector is produced that is normal to all previously calculated vector, and in the second step this vector is normalized with respect to the Euclidean norm. If vectors were not normalized, then we should perform division by $\langle \mathbf{q}_i, \mathbf{q}_i \rangle$ in order to obtain the correct projection.

The **modified Gram-Schmidt procedure** is performed in such a way that projections are gradually subtracted from the original vector and such reduced vectors are projected on already obtained ortonormal vectors:

$$
\begin{aligned}
&\mathbf{q}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|};\\
&j = 2, ..., n:\\
&\quad \mathbf{q}_j^{(0)} = \mathbf{a}_j\\
&\quad k = 1, ..., j-1\\
&\qquad \mathbf{q}_j^{(k)} = \mathbf{q}_j^{(k-1)} - \left\langle \mathbf{q}_j^{(k-1)}, \mathbf{q}_k \right\rangle \mathbf{q}_k\\
&\quad \mathbf{q}_j = \frac{\mathbf{q}_j^{(j-1)}}{\left\|\mathbf{q}_j^{(j-1)}\right\|_2}
\end{aligned}
\qquad . \qquad (6.47)
$$

Vectors obtained by this process are the same since, taking into account orthonormality of vectors $\mathbf{q}_i$, we have

$$
\mathbf{q}_k^{k-1} = \mathbf{a}_k - \left\langle \mathbf{q}_1, \mathbf{a}_k \right\rangle \mathbf{q}_1 - \left\langle \mathbf{q}_2, \left\langle \mathbf{q}_1, \mathbf{a}_k \right\rangle \mathbf{q}_1 \right\rangle \mathbf{q}_2 - ... =
$$
$$
\mathbf{a}_k - \sum_{j=1}^{k-1} \left\langle \mathbf{q}_j, \mathbf{a}_k \right\rangle \mathbf{q}_j \qquad .
$$

We can use the Gram-Schmidt procedure for computing the QR factorization of a matrix. Matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ ($m \geq n$) admits a QR decomposition if there exist an orthogonal matrix (columns orthogonal and normalized) $\mathbf{Q} \in \mathbb{R}^{m \times m}$ and an upper trapezoidal matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$, such that $\mathbf{A} = \mathbf{Q}\mathbf{R}$.

Reduced QR factorization:

If $\mathbf{A}_{m \times n}$ is of rank *n* (i.e. ahs full rank) for which a QR factorization is known, then there exists a unique factorization of **A** of the form

$$
\mathbf{A}_{m \times n} = \tilde{\mathbf{Q}}_{m \times n} \tilde{\mathbf{R}}_{n \times n} \qquad (48)
$$

where $\tilde{\mathbf{Q}}$ is orthogonal and $\tilde{\mathbf{R}}$ upper triangular, and these are submatrices of **Q** and **R** (left upper corner). $\tilde{\mathbf{Q}}$ has orthonormal vectors of columns and coincides with the *Cholesky factor* **H** of the symmetric positive definite matrix $\mathbf{A}^T\mathbf{A}$, i.e.

$$
\mathbf{A}^T\mathbf{A} = \tilde{\mathbf{R}}^T\tilde{\mathbf{R}} \ . \qquad (49)
$$

This matrix is positive definite because every matrix of the form $\mathbf{A}^T\mathbf{A}$ is positive semidefinite, and if **A**has full rank then it is positive definite.

If **A** has full rank *n* then the column vectors of **A** form an orthonormal basis for the vector space $range(\mathbf{A})$. The QR factrorization is therefore a procedure of generating an orthonormal basis for a given set of vectors. If **A** has rank *r*<*n*, then the QR factorization does not necessarily yield an orthonormal basis for $range(\mathbf{A})$. However, it is possible to obtain a factorization of the form

$$\mathbf{Q}^T\mathbf{A}\,\mathbf{P} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where **Q** is orthogonal, **P** is a permutation matrix and $\mathbf{R}_{11}$ is a nonsingular upper triangular matrix of order *r*.

$$\mathbf{A}_{(m\times n)} = \mathbf{Q}_{(m\times n)}\,\mathbf{R}^{upper}_{(n\times n)}$$

Figure 6: The reduced QR factorization.

If the QR decomposition is performed by the modified Gram-Schmidt procedure, then the columns of **A** are vectors $\mathbf{a}_i$ and columns of **Q** are vectors $\mathbf{q}_i$ of the equation (47). Matrix **R** is obtained by left multiplication of **A** by $\mathbf{Q}^T$ (from equation n(48), since **Q** is orthogonal):

$$\underline{\underline{\mathbf{R} = \mathbf{Q}^T\,\mathbf{A}}}. \qquad (50)$$

6.2.1.1 Case *m*=*n*

$$\mathbf{A}_{(n\times n)} = \mathbf{Q}_{(n\times n)}\,\mathbf{R}^{upper}_{(n\times n)} \qquad (51)$$

If **A** is non-singular and **R** has positive diagonal elements (by agreement), then the factorization (51) is unique.

6.2.2 Remark: Extended QR decomposition

In fact, some QR algorithms produce the expanded trapezoid matrix $\mathbf{R}_{(m\times n)}$. In this way we have

$$\mathbf{A}_{m\times n} = \mathbf{Q}_{(m\times m)}\,\mathbf{R}_{(m\times n)}\ , \qquad (52)$$

where **R** is upper trapezoidal,

$$r_{ij} = \begin{cases} r_{ij}\ ;\ i \le n \\ 0; i > n \end{cases}\ , \tag{53}$$

and **A** is upper triangular, i.e. $a_{ij} = 0; i > j$. The orthogonal matrix **Q** is orthogonal in the whole space $\mathbb{R}^{n\times n}$, therefore its orthonormal basis could not obtain only from columns of **A**, but was expanded. We will therefore call the procedure **expandeded QR decomposition with Gram-Schmidt orthogonalization**.

While **R** is simply supplemented by zeros below the *m*-th row, there are different possibilities of how to expand **Q**. The most obvious possibility is to successively take trial vectors of some basis of $\mathbb{R}^n$, orthogonalize them with respect to already calculated columns of **Q**, and if linear dependence on already obtained orthonormal vectors occurs then reject the current trial vector and skip to the next one (since at least *m-k* vectors of the basis will be linearly independent on all *k* vectors that are already there). Orthonormal basis vectors $\mathbf{e}_i = [0, ..., 1, 0, ...]$ can be most conveniently taken[1].

For a measure of linear dependency, we can simply take the norm of what remains after subtraction of projections on already calculated $\mathbf{q}_k$.[2]. Some lower bound must be set on this norm, e.g. 0.1. The norm must be such that there for sure exists such basis vector that after subtraction of all projections on any set of lower dimensional basis vectors, the norm of the remaining vector is greater than this lower bound. Besides, the norm should not be too small because this would increase numerical errors (because we would allow acceptance of vectors that are almost linearly dependent on the currently available basis vectors).

If we would also need to extend **A** with additional columns such that it would have a full rank in $\mathbb{R}^n$, then we can proceed as follows: We first extend **Q** as described above. Then we extend **R** to the dimension $m \times m$ by diagonal elements taking the value 1 (or maybe an average absolute value of already calculated diagonal elements obtained by the reduced QR factoritzation, in order to improve scaling) and out of diagonal elements taking the value 0. Then we calculate the extended **A** by simply calculating the product $\mathbf{QR}$ by extended matrices. The extended **A** would in this way have the full rank *m* if the original **A** had full rank *n* (because both **Q** and **R** would have a full rank). Besides, the first *n* columns of **A** would be the same as with the original because of the zeros in **R** below the upper-left $n \times n$ block.

6.2.3 Solution of Systems of Equations with QR decomposition

We are solving the system

[1] In special cases when there is a reason to suspect that columns of **A** are close to the first *m* of these vectors, we can reverse the order in which these vectors are taken, in order to minimize the possibility of rejecting vectors.
[2] In fact, the ratio between this norm and norm of $\mathbf{e}_i$ should be taken, but norm of $\mathbf{e}_i$ is 1.

$$\mathbf{A}_{(m\times n)}\,\mathbf{x}_{(n\times 1)} = \mathbf{b}_{(m\times 1)} \tag{54}$$

We assume we have factored matrix **A** as a product of an orthogonal matrix **Q** and an upper trapezoid matrix **R**:

$$\mathbf{A}_{(m\times n)} = \mathbf{Q}_{(m\times m)}\mathbf{R}_{(m\times n)} \tag{55}$$

where **Q** has a full rank *m*, and

$$i > j \Rightarrow r_{ij} = 0\ . \tag{56}$$

If the *m*>*n* then the system

$$\mathbf{A}_{(m\times n)}\,\mathbf{x}_{(n\times 1)} = \mathbf{b}_{(m\times 1)} \tag{57}$$

is **overdetermined** (i.e. there are moer equations) and can therefore be solved in the least square sense.

6.2.3.1 Case *m*=*n*

We first limit ourselves on the case *m*=*n*, i.e. the number of equations is the same as the number of variables. We solve the system by first setting (since $\mathbf{Q}^{-1} = \mathbf{Q}^{T}$)

$$\mathbf{y} = \mathbf{Q}^{T}\,\mathbf{b}\ . \tag{58}$$

and then solving the system

$$\mathbf{R}\,\mathbf{x} = \mathbf{y} \tag{59}$$

Equation (59) is a system with an upper triangular matrix whose solution is described in Section 8.4.1.

6.2.3.2 Case *m*>*n*

In this case we have an overdetermined system with more equations than unknowns. Solution of such systems is described in Section 7.

6.3 Non-standard factorization by using the GS

In this chapter a non-standard form of a factorization obtained derived from the Gramm-Schmidt orthogonalization is used. A standard form of factorization known as QR decomposition is described in Section 6.2. The Section 6.1.1 explains the idea of Gram-Schmidt orthogonalization that is used for derivation of both forms, while the procedure described in this Section is specific for non-standard form.

In standard form, we perform the QR factorization in such a way that the original matrix is a product of an orthogonal and upper triangular matrix produced by the factorization. The orthogonal matrix is calculated first ant the upper triangular matrix is obtained by left multiplying the original matrix by its transpose (since the inverse of the orthogonal matrix equals its transpose).

In the non-standard factorization described here, we express theorthogonal matrix that is obtained from the original one, as a product of the original and upper triangular matrix. Again, columns of the orthogonal matrix are orthogonalized vectors obtained from columns of the original matrix. The orthogonal matrix is obtained by a similar Gram-Schmidt process as in QR decomposition, while the upper triangular matrix is obtained simultaneously with the orthogonal matrix by performing equivalent operations as on the orthogonal matrix, on the matrix that is first set to identity matrix (and these operations transform it to the upper triangular form such that its left product with the original matrix equals the orthogonal matrix). The method will not be often used in practice, but it is described here because it is instructive.

After the Gram-Schmidt orthogonalization, each orthogonal vectors $\mathbf{q}_j$ is a linear combination of the original vectors. We can therefore write

$$\underline{\underline{\mathbf{Q}_{(m\times n)} = \mathbf{V}_{(m\times n)}\mathbf{A}_{(n\times n)}}}\ , \qquad ,(6.60)$$

where **A** is a matrix of coefficients of these linear combinations[1], **V** is the matrix whose columns are the original vectors $\mathbf{v}_i$ and **Q** is the matrix whose columns are orthogonal vectors $\mathbf{q}_j$. The above equation can be written by columns, which gives orthogonal vectors expressed as linear combination of the original ones:

Remark: the following derivation should be checked. The final procedure is correct, however, because the functions described in Section 6.3.3 were all verified.

$$\mathbf{q}_j = \sum_{k=1}^{n} a_{kj}\mathbf{v}_k, \quad j = 1, 2, ..., m\,. \qquad (6.0.$$

[1] Beware of different notation with respect to the usual notation. Here **V** denotes the original matrix and **A** denotes an upper triangular matrix, while in most commonly, **A** is used for the original matrix and **R** is used for the upper triangular matrix.

This comes from

$$\left(\mathbf{q}_j\right)_i = q_{ij} = \sum_{k=1}^{n} v_{ik} a_{kj} = \sum_{k=1}^{n} a_{kj} v_{ik} \ , \ \mathbf{q}_j = \sum a_{kj} \left(\mathbf{V}^T\right)_k \ .$$

It is seen from (6.0 and (6.0 that matrix **A** is upper triangular with unit diagonal elements:

$$\mathbf{A} = \begin{bmatrix} 1 & * & ... & * \\ 0 & 1 & ... & * \\ ... & ... & ... & ... \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (6.0.$$

Evaluation of orthogonal vectors $\mathbf{q}_j$ can be accompanied by simultaneous evaluation of coefficients of **A**. The complete procedure is then the following:

$j = 1$:

set

$$\mathbf{q}_1 = \mathbf{v}_1 \qquad (6.0.$$

and

$$a_{11} = 1, \ a_{k1} = 0, \ k = 2,3,...,m \qquad (6.0.$$

for $j = 2,3,...m$:

set

$$\mathbf{q}_j = \mathbf{v}_j + \sum_{l=1}^{j-1} - \frac{< \mathbf{v}_j, \mathbf{q}_l >}{< \mathbf{q}_l, \mathbf{q}_l >} \mathbf{q}_l \ , \qquad (6.0.$$

and

$$a_{jj} = 1, \ a_{kj} = \sum_{l=k}^{j-1} - \frac{< \mathbf{v}_j, \mathbf{q}_l >}{< \mathbf{q}_l, \mathbf{q}_l >} a_{kl} \ , k = 1,2,..., j-1 \ . \qquad (6.0.$$

Equation (6.0 follows from (6.0 and (6.0, which give

$$\mathbf{q}_j = \mathbf{v}_j + \sum_{l=1}^{j-1} -\frac{<\mathbf{v}_j, \mathbf{q}_l>}{<\mathbf{q}_l, \mathbf{q}_l>}\mathbf{q}_l = \mathbf{v}_j + \sum_{l=1}^{j-1} -\frac{<\mathbf{v}_j, \mathbf{q}_l>}{<\mathbf{q}_l, \mathbf{q}_l>}\sum_{k=1}^{l} a_{kl}\mathbf{v}_k = \sum_{k=1}^{j} a_{kj}\mathbf{v}_k \ , \qquad (6.0.$$

where (6.0 was taken into account twice. Equating terms at $\mathbf{v}_k$ gives (6. 0, where it is taken into account that $a_{kl} = 0$ for $l < k$.

In the algorithm the coefficient $-\frac{<\mathbf{v}_j, \mathbf{q}_l>}{<\mathbf{q}_l, \mathbf{q}_l>}\mathbf{q}_l$ should be evaluated only once for each (j,l), therefore individual contributions to all a_{kj} in (6.0 are evaluated when specific coefficient is available. The algorithm step ((6.0, (6.0) is therefore actually the following:

$\underline{\underline{j = 1:}}$

set

$$\underline{\mathbf{q}_1 = \mathbf{v}_1} \qquad (6.0.$$

and

$$\underline{a_{11} = 1,\ a_{k1} = 0, k = 2,3,...,m} \qquad (6.0.$$

for $j = 2,3,...m$:

set

$$\underline{a_{jj} = 1,\ a_{kj} = 0, k = 1,2,..., j-1} \qquad (6.0.$$

and

$$\underline{\mathbf{q}_j = \mathbf{v}_j\,,} \qquad (6.0.$$

for $\underline{\underline{l = 2,3,...j-1}}$

$$\underline{\mathbf{q}_j = \mathbf{q}_j + -\frac{<\mathbf{v}_j, \mathbf{q}_l>}{<\mathbf{q}_l, \mathbf{q}_l>}\mathbf{q}_l\,,} \qquad (6.0.$$

and

$$a_{kj} = a_{kj} - \frac{< \mathbf{v}_j, \mathbf{q}_l >}{< \mathbf{q}_l, \mathbf{q}_l >} a_{kl}, \;\; k = 1, 2, ..., l \; . \qquad \text{(6.0.}$$

6.3.1 Orthogonalization with normalization

This operation generates vectors which are mutually orthogonal and whose Euclidian norm equals 1. The procedure is practically equivalent except for the normalization factor that is applied at the end of calculation of each column of **Q**. The procedure follows those described in (6.0 to (6.0, except that orthogonal vectors are normalised immediately after they are constructed and because of that some of the coefficient in the above equations become 1. Orthogonalisation with normalisation is approximately as fast as without normalisation, but is more convenient for solution of systems of equations. Matrix **A** does not have units in diagonal any more:

$$\mathbf{A} = \begin{bmatrix} * & * & ... & * \\ 0 & * & ... & * \\ ... & ... & ... & ... \\ 0 & 0 & 0 & * \end{bmatrix} \qquad \text{(6.0.}$$

The procedure is the following:

$j = 1$:

set

$$\mathbf{q}_1 = \frac{\mathbf{v}_1}{< \mathbf{v}_1, \mathbf{v}_1 >} \qquad \text{(6.0.}$$

and

$$a_{11} = 1, \;\; a_{k1} = 0, k = 2, 3, ..., m \qquad \text{(6.0.}$$

for $j = 2, 3, ...m$:

set

$$a_{jj} = 1, \;\; a_{kj} = 0, k = 1, 2, ..., j-1 \qquad \text{(6.0.}$$

and

$$\mathbf{q}_j = \mathbf{v}_j . \tag{6.0.}$$

for $l = 2,3,...j-1$

set

$$\mathbf{q}_j = \mathbf{q}_j - < \mathbf{v}_j, \mathbf{q}_l > \mathbf{q}_l , \tag{6.0.}$$

and

$$a_{kj} = a_{kj} - < \mathbf{v}_j, \mathbf{q}_l > a_{kl}, \ k = 1,2,...,l . \tag{6.0.}$$

for $k = 1,2,..., j$

set

$$a_{kj} = \frac{a_{kj}}{\sqrt{< \mathbf{q}_j, \mathbf{q}_j >}} . \tag{6.0.}$$

set

$$\mathbf{q}_j = \frac{\mathbf{q}_j}{\sqrt{< \mathbf{q}_j, \mathbf{q}_j >}} \tag{6.0.}$$

6.3.2 Solution of Systems of Equations with Gram-Schmidt Orthogonalisation (non-standard form)

In Gram-Schmidt Orthogonalisation we construct from matrix of original vectors (columns) **V** an orthogonal matrix **Q** and upper trapezoid matrix of coefficients **A** so that

$$\mathbf{Q}_{(n\times n)} = \mathbf{V}_{(n\times m)}\mathbf{A}_{(m\times m)} \ ; m \le n . \tag{6.0.}$$

For solution of system of equations only square matrices will be considered here, i.e. *m*=*n*.

We can solve the system

$$\underline{\underline{\mathbf{V}\mathbf{x} = \mathbf{b}}} \qquad (6.0.$$

by introducing a new variable $\mathbf{y} = \mathbf{A}^{-1}\mathbf{x}$. The above system then falls to one orthogonal system and matrix multiplication:

$$\underline{\underline{\mathbf{Q}\mathbf{y} = \mathbf{b}}} \qquad (6.0.$$

and

$$\underline{\underline{\mathbf{x} = \mathbf{A}\mathbf{y}}} \ . \qquad (6.0.$$

Solution of (6.0 is described in Section 8.2 while (6.0 is just a simple matrix multiplication.

6.3.3 Implementation of GS (non-standard form)

In IOptLib the following functions are used:

```
void GSortplain(matrix V,matrix Q,matrix A)
```

- From original matrix $\mathbf{V}_{(m\times n)}$ ($m \geq n$), a matrix with mutually orthogonal (but not normal) columns $\mathbf{Q}_{m\times n}$ and upper triangular $\mathbf{A}_{(n\times n)}$ are calculated such that $\mathbf{Q} = \mathbf{V}\,\mathbf{A}$. All matrices must be allocated and of appropriate dimensions.

```
void GSortnormplain(matrix v,matrix q,matrix a)
```

- Similar as `GSortplain`, only that $\mathbf{Q}$ is orthogonal (i.e. columns are not only mutually orthogonal but also normalized).

```
void GSort0(matrix V,matrix *Q,matrix *A)
```

and

```
int GSortnorm0(matrix v,matrix *q,matrix *a)
```

are comfortable forms of `GSortplain` and `GSortnormplain`. Matrices of results are allocated or reallocated with the appropriate dimensions if necessary.

```
vector solvGS0(matrix M,vector b,vector *x)
```

Solves a system of equations $\mathbf{M}\mathbf{x} = \mathbf{b}$ in such a way that the Gram-Schmidt orthogonalization is first performed by `GSortnormplain` and then the system is solved according to (0 and (0. The intermediate storage necessary for operations is allocated and then released within the function. Because of this, function is not efficient and should not be used for important computations.

The systems can instead be solved by three calls – first to orthogonalization (`GSortnormplain0(M,&Q,&A)`) and then by `solvortnorm0(Q,b,&x)` to solve the orthogonal system (0 followed by `matprodvec0(A,x,&x)` to perform matrix multiplication (0. Solution of the orthogonal system can also be performed by `mattranspprodvec(Q,b,&x)`.

```
matrix solvmatGS0(matrix M,matrix B,matrix *X)
```

- This function is similar to `solvGS0`, but it solves several systems of equations at once, where **B** contains right-hand sides as its columns and solutions are stored in columns of **X**.

7 OVERDETERMINED SYSTEMS

In this Section we describe solution of **over determined systems** of equations where we have more equations than unknowns:

$$\mathbf{A}_{(m\times n)}\,\mathbf{x}_{(n\times 1)} = \mathbf{b}_{(m\times 1)}\,,\ m > n\ . \tag{61}$$

We assume that rank of **A** is *n*. In general this means than we can not simultaneously satisfy all the equations. We therefore search for the solution in a **least squares sense**, i.e. we are searching for such $\mathbf{x}_0$ for which the sum of squares of differences between left-hand and right-hand sides is minimal:

$$\mathbf{x}_0 = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\| \mathbf{A}\,x - \mathbf{b} \right\|_2^2\ . \tag{62}$$

The function to be minimized is expressed component wise as

$$\left\| \mathbf{A}\,x - \mathbf{b} \right\|_2^2 = \sum_{i=1}^{m}\left(\sum_{j=1}^{n}\left(a_{ij}x_j\right) - b_i \right)^2\ .$$

A unique solution $\mathbf{x}_0$ exists (under assumption $rank\,\mathbf{A} = n$). This is exactly the solution of equation

$$\underline{\underline{\mathbf{A}^T\,\mathbf{A}\,\mathbf{x} = \mathbf{A}^T\,\mathbf{b}}}\ , \tag{63}$$

which is obtained by left multiplying the equation by $\mathbf{A}^T$. This system is called the *normal system* of equations.

We can verify that the solution of (63) is also the solution of (62). We write

$$\mathbf{B} = \mathbf{A}^T\mathbf{A},\quad \mathbf{c} = \mathbf{A}^T\mathbf{b}\,. \tag{64}$$

Then we have

$$\| \mathbf{A}x-\mathbf{b} \|_2^2 = \langle \mathbf{A}\mathbf{x}-\mathbf{b},\, \mathbf{A}\mathbf{b}-\mathbf{b} \rangle = (\mathbf{B}\mathbf{x}-\mathbf{c})^T \mathbf{B}^{-1}(\mathbf{B}\mathbf{x}-\mathbf{c}) - \mathbf{c}^T\mathbf{B}^{-1}\mathbf{c} + \mathbf{b}^T\mathbf{b} \ .$$

Matrix **B** is positive definite: $\forall \mathbf{x},\ \mathbf{x}^T\mathbf{B}\mathbf{x} = \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = \| \mathbf{A}\mathbf{x} \|_2^2 \geq 0$. This is 0 only when $\mathbf{A}\mathbf{x} = \mathbf{0}$, which is possible only when $\mathbf{x} = \mathbf{0}$ because matrix **A** has a full rank *n*.

If **B** is positive definite then also $\mathbf{B}^{-1}$ is positive definite. The smallest value of the quadratic form therefore equals 0 and this is when equation (63) holds, while other terms are independent of **x** and represent the sum of squares of differences in the solution. Because $\mathbf{A}^T\mathbf{A} = \mathbf{B}$ is not singular, the solution is uniquely defined.

System (63) can be calculated by the **Cholesky decomposition** of **B**,

$$\mathbf{A}^T\mathbf{A} = \mathbf{B} = \mathbf{V}^T\mathbf{V} \tag{65}$$

where **V** is upper triangular, which is followed by solution the lower triangular system

$$\mathbf{V}^T\mathbf{y} = \mathbf{c} \quad \left(= \mathbf{A}^T\mathbf{b} \right) \tag{66}$$

and the upper triangular system

$$\mathbf{V}\mathbf{x}_0 = \mathbf{y} \ . \tag{67}$$

The described classical way is not recommendable because sensitivity of the system can be strongly increased at explicit calculation ob **B** (i.e. the matrix can become very ill conditioned). Let us suppose that eigenvalues of **B** arranged in decreasing order are σ_i^2. Spectral sensitivity of **A** is then σ_1 / σ_n, while spectral sensitivity of **B** is $\left(\sigma_1 / \sigma_n \right)^2$. More recommendable is solution through orthogonal transformations.

7.1 Orthogonal methods

The basis of the orthogonal methods is the QR factorization of the matrix **A**, to a product of an **orthogonal matrix Q** and **upper trapezoid matrix U**:

$$\mathbf{A}_{(m\times n)} = \mathbf{Q}_{(m\times m)}\,\mathbf{U}_{(m\times n)} \ , \tag{68}$$

where

$$\begin{aligned} &\mathbf{Q}^T\,\mathbf{Q} = \mathbf{I}_m \\ &i > j \Rightarrow u_{ij} = 0 \end{aligned} \tag{69}$$

We calculate the solution of the orthogonal system

$$\underline{\underline{\mathbf{z} = \mathbf{Q}^T\mathbf{b}}}\ , \tag{70}$$

and we write **U** and **z** in block form,

$$\underline{\underline{\mathbf{U} = \begin{bmatrix} \mathbf{V}_{(n\times n)} \\ \mathbf{0}_{((m-n)\times n)} \end{bmatrix},\ \mathbf{z} = \begin{bmatrix} \mathbf{y}_n \\ \mathbf{w}_{m-n} \end{bmatrix}}}\ . \tag{71}$$

Taking into account the decomposition (68) and block form (71), we have

$$\mathbf{B} = \mathbf{A}^T\mathbf{A} = \left(\mathbf{Q}\,\mathbf{U}\right)^T\left(\mathbf{Q}\,\mathbf{U}\right) = \mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V}\ . \tag{72}$$

This means that upper triangular matrix **V** is the Cholesky factor of the normal matrix $\mathbf{A}^T\mathbf{A}$. This means that with the orthogonal methods, calculation of the normal matrix $\mathbf{A}^T\mathbf{A}$ and its factorization is avoided. Matrices **B** in (71) and (65) are the same if we take care in the orthogonal decomposition that all diagonal elements of **V** are positive.

We can further calculate

$$\mathbf{c} = \mathbf{A}^T\mathbf{b} = \left(\mathbf{Q}\,\mathbf{U}\right)^T\mathbf{b} = \mathbf{U}^T\mathbf{Q}^T\mathbf{b} = \mathbf{U}^T\mathbf{z} = \mathbf{V}^T\mathbf{y}\ . \tag{73}$$

This means that **y**, which is the upper part of the transformed vector **z**, solution of the lower triangular system (66).

The least squares solution is obtained by solution of the upper triangular system

$$\underline{\underline{\mathbf{V}\,\mathbf{x}_0 = \mathbf{y}}}\ . \tag{74}$$

Vector **w**, which is the lower part of **z**, also has its meaning – square of its Euclidean norm is the sum of squares of differences:

$$\left\|\,\mathbf{A}\,x - \mathbf{b}\,\right\|_2^2 = \left\|\,\mathbf{w}\,\right\|_2^2 . \tag{75}$$

This is derived as follows:

$$\| \mathbf{A}x - \mathbf{b} \|_2^2 = \| \mathbf{Q}\mathbf{U}\mathbf{x}_0 - \mathbf{Q}\mathbf{z} \|_2^2 = \| \mathbf{U}\mathbf{x}_0 - \mathbf{z} \|_2^2 = \| \mathbf{V}\mathbf{x}_0 - \mathbf{y} \|_2^2 + \| \mathbf{w} \|_2^2 = \| \mathbf{w} \|_2^2 \ .$$

Remark on QR decomposition:
The QR decomposition is often calculated in the **reduced form**, such that

$$\mathbf{A}_{(m\times n)} = \mathbf{Q}_{(m\times n)} \mathbf{U}_{(n\times n)} \ , \tag{76}$$

In this case **Q** must be supplemented by orthogonal columns up to full rank *m*.

8 SPECIAL SYSTEMS OF EQUATIONS

8.1 Orthogonal systems

By the term orthogonal system, we will refer to a systems with ***orthogonal matrix***, i.e. a matrix with **orthonormal columns** (and consequently rows). A more general class of systems with matrices whose columns are mutually orthogonal but not normalized is described in Section 8.2.

For orthogonal real matrices[1] the following equation is valid:

$$\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}; \ \mathbf{Q}^{-1} = \mathbf{Q}^T \ . \tag{77}$$

Therefore, the system of equations with an orthogonal matrix

$$\mathbf{Q}\mathbf{x} = \mathbf{b} \tag{78}$$

is solved simply by multiplying the vector of right hand sides by the transpose of the system matrix:

$$\mathbf{x} = \mathbf{Q}^T\mathbf{b} \tag{79}$$

[1] Equivalent for matrices defined over complex field is unitary.

8.2 Systems with Matrices that have Orthogonal Columns

Let's have a system

$$\underline{\underline{\mathbf{Q}\mathbf{x}=\mathbf{b}}}, \tag{8.0.}$$

where **Q** is orthogonal, i.e. such that its columns are mutually orthogonal (dot product zero). The above system can be written in the form

$$\mathbf{b}=x_1\mathbf{Q}_1+x_2\mathbf{Q}_2+\ldots+x_n\mathbf{Q}_n, \tag{8.0.}$$

where $\mathbf{Q}_i$ denotes the *i*-th column of **Q** and x_i denotes the *i*-th component of vector **x**. If the equation is dot-multiplied by $\mathbf{Q}_i$, we obtain

$$<\mathbf{b},\mathbf{Q}_i>=x_i<\mathbf{Q}_i,\mathbf{Q}_i>, \tag{8.0.}$$

which follows from orthogonality of $\mathbf{Q}_j$ and $\mathbf{Q}_i$ for each $j\neq i$. It follows that

$$\underline{\underline{x_i=\frac{<\mathbf{b},\mathbf{Q}_i>}{<\mathbf{Q}_i\mathbf{Q}_i>}}}. \tag{8.0.}$$

If columns of **Q** are also <u>normed</u> in the Euclidian norm, then we simply have

$$\mathbf{x}_i=<\mathbf{b},\mathbf{Q}_i>, \tag{8.0.}$$

which simplifies to

$$\underline{\underline{\mathbf{x}=\mathbf{Q}^T\mathbf{b}}}. \tag{80}$$

If columns of **Q** are orthonormal then **Q** is an orthogonal matrix for which $\mathbf{Q}^{-1}=\mathbf{Q}^T$.

8.2.1 Inverse of a Matrix with Orthogonal Columns

We search for an unknown matrix **X** such that

$$\mathbf{Q}\mathbf{X}=\mathbf{I}. \tag{8.0.}$$

For the *i*-th column of **X** we have

$$\mathbf{Q}\mathbf{X}_i = \mathbf{e}_i, \tag{8.0.}$$

where $\mathbf{e}_i$ is the *i*-th unit vector (i.e. component *i* is 1 and all others are 0). It follows from (8.0 that

$$x_{ji} = [\mathbf{X}_i]_j = \frac{<\mathbf{e}_i, \mathbf{Q}_j>}{<\mathbf{Q}_j, \mathbf{Q}_j>} = \frac{\mathbf{q}_{ij}}{<\mathbf{Q}_j, \mathbf{Q}_j>} \tag{8.0.}$$

or finally

$$\underline{\underline{[\mathbf{Q}^{-1}]_{ij} = \frac{\mathbf{q}_{ji}}{<\mathbf{Q}_i, \mathbf{Q}_i>}}} \tag{8.0.}$$

If clolumns of **Q** are also <u>normalised</u> with respect to the Euclidian norm, then we have simply

$$\underline{\underline{[\mathbf{Q}^{-1}]_{ij} = \mathbf{q}_{ji}}}. \tag{8.81}$$

8.2.2 Component-wise verification

We verify some of the equations from Section 8.2 in component-wise notation. Let's have a system

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{8.82}$$

$$\mathbf{A}_{(m\times n)}\mathbf{x}_{(n\times 1)} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \\ \ldots. \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \end{bmatrix} = x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \ldots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \ldots \\ a_{m2} \end{bmatrix} + \ldots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \ldots \\ a_{mn} \end{bmatrix} \tag{8.83}$$

$$= x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \ldots x_n\mathbf{a}_n$$

With $\mathbf{a}_i$ we denote the *i*-th column of **A**.

For product of two matrices it holds

$$\left[\mathbf{A}_{(m\times n)}\mathbf{X}_{n\times p}\right]_j = \begin{bmatrix} a_{1k}\, x_{kj} \\ a_{2k}\, x_{kj} \\ \dots \\ a_{mk}\, x_{kj} \end{bmatrix} = \mathbf{A}\mathbf{x}_j\,, \tag{8.84}$$

i.e. the *j*-th column of the product equals product of the left matrix with the *j*-th column of the right matrix. In the above equation, the Einstein summation rule was applied, i.e. we have summation over indices that are doubled.

Let's have a matrix **Q** whose **columns are mutually orthogonal**:

$$\mathbf{q}_i\mathbf{q}_j = \sum_{k=1}^{n} q_{ki}q_{kj} = \delta_{ij}\|\mathbf{q}_i\|^2\,, \tag{8.85}$$

where $\delta_{ij} = \begin{Bmatrix} 1; i = j \\ 0; i \neq j \end{Bmatrix}$.

Product $\mathbf{Q}^T\mathbf{Q} = \mathbf{D}$ is a diagonal matrix because

$$\left[\mathbf{Q}^T\mathbf{Q}\right]_{ij} = <\mathbf{q}_i, \mathbf{q}_j> = \delta_{ij}\|\mathbf{q}_i\|^2\,, \tag{8.86}$$

where $\mathbf{q}_i$ is the *i*-th column vector of matrix **Q.**

A special case are **orthogonal matrices** which in addition to (8.0 have normed columns, such that

$$<\mathbf{q}_i, \mathbf{q}_j> = \delta_{ij} \tag{8.87}$$

For such a matrix the following is true:

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}\,. \tag{8.88}$$

8.3 Systems of Equations with Matrix with Orthogonal Columns

Let's have a system

$$\underline{\underline{\mathbf{Q}\mathbf{x}=\mathbf{b}}}, \tag{8.89}$$

where **Q** is orthogonal matrix. It follows that

$$\mathbf{b}=x_1\mathbf{q}_1+x_2\mathbf{q}_2+\ldots+x_n\mathbf{q}_n=\sum_{k=1}^{n}x_k\mathbf{q}_k\ , \tag{8.90}$$

where $\mathbf{q}_I$ is the *i*-th column of **Q**. After multiplication of the above equation with $\mathbf{q}_i^T$ from left we obtain

$$<\mathbf{b},\mathbf{q}_i>=x_i<\mathbf{q}_i,\mathbf{q}_i> \tag{8.91}$$

and so

$$\underline{\underline{x_i=\frac{<\mathbf{b},\mathbf{q}_i>}{\left\|\mathbf{q}_i\right\|^2}}}\ . \tag{8.92}$$

Solution of the system can is obtained very simply. This can be used for formulae for inverse of an orthogonal matrix:

$$\mathbf{Q}\mathbf{X}=\mathbf{I},\ =\left[\mathbf{Q}\mathbf{x}\right]_i=\mathbf{Q}\mathbf{x}_i=\mathbf{e}_i\ , \tag{8.93}$$

using (8.0 we have

$$\left[\mathbf{x}_i\right]_j=x_{ji}=\frac{<\mathbf{e}_i,\mathbf{q}_j>}{\left\|\mathbf{q}_j\right\|^2}=\frac{q_{ij}}{\left\|\mathbf{q}_j\right\|^2} \tag{8.94}$$

or

$$\underline{\underline{\left[\mathbf{Q}^{-1}\right]_{ij}=\frac{q_{ji}}{\left\|\mathbf{q}_i\right\|^2}=\frac{q_{ji}}{\sum_{k=1}^{n}q_{ki}q_{ki}}}}\ . \tag{8.95}$$

If columns of **Q** are also normed with respect to the Euclidian norm, then the denominator of the above equation equals 1.

8.4 Triangular Systems

8.4.1 Upper Triangular Systems

We solve the system

$$\mathbf{Ax} = \mathbf{b}, \tag{8.96}$$

where **A** is upper triangular, i.e. elements below diagonal are zero, see (3.0. The system looks like this:

$$\begin{matrix} & \mathbf{A} & & & & \mathbf{x} & & \mathbf{b} \end{matrix}$$
$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} \tag{8.97}$$

The above equations written in reversed order are:

$$\begin{array}{ccccc} & & & + a_{n\,n} x_n & = b_n \\ & & + a_{n-1\,n-1} x_{n-1} & + a_{n-1\,n} x_n & = b_{n-1} \\ & + a_{n-2\,n-2} x_{n-2} & + a_{n-2\,n-1} x_{n-1} & + a_{n-2\,n} x_n & = b_{n-2} \\ \dots & \dots & \dots & \dots & \dots \\ a_{11} x_1 & + \dots & + a_{1\,n-1} x_{n-1} & + a_{1\,n} x_n & = b_1 \end{array} \tag{8.98}$$

The solution is evaluated backwards:

$$\underline{\underline{x_n = \frac{b_n}{a_{nn}}}}, \tag{8.99}$$

$$x_i = \left(b_i - \sum_{k=i+1}^{n} a_{ik} x_k \right) / a_{ii}, \quad i = n-1, n-2, ..., 1 \tag{8.100}$$

8.4.2 Lower Triangular Systems

We solve the system

$$\mathbf{Ax} = \mathbf{b}, \tag{8.101}$$

where **A** is lower triangular, i.e. elements below diagonal are zero, see (3.0. The system looks like this:

$$\begin{matrix} & \mathbf{A} & & & & \mathbf{x} & \mathbf{b} \end{matrix}$$
$$\begin{bmatrix} a_{11} & 0 & ... & 0 \\ a_{21} & a_{22} & ... & ... \\ ... & ... & ... & 0 \\ a_{n1} & a_{n2} & ... & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ ... \\ b_n \end{bmatrix} \tag{8.102}$$

The solution is evaluated forwards:

$$x_1 = \frac{b_1}{a_{11}}, \tag{8.103}$$

$$x_i = \left(b_i - \sum_{k=1}^{i-1} a_{ik} x_k \right) / a_{ii}, \quad i = 2, 3, ..., n \tag{8.104}$$

9 NUMBER OF OPERATINOS FOR STANDARD MATRIX OPERATIONS

9.1 Basic Operations

9.1.1 Matrix and vector multiplications:

Matrix multiplications:

$\mathbf{C}_{m\times n} = \mathbf{A}_{m\times p}\mathbf{B}_{p\times n}$: $N = 2mnp$ (vsak element približno p množenj in p seštevanj)

$\underline{\mathbf{C}_{n\times n}\mathbf{A}_{n\times n}\mathbf{B}_{n\times n}}$: $\underline{N = 2n^3}$

Matrix times vector:

$\mathbf{c}_{m\times 1} = \mathbf{A}_{m\times n}\mathbf{b}_{n\times 1}$: $N = 2mn$ (vsak element približno n množenj in n seštevanj)

$\underline{\mathbf{c}_{n\times 1} = \mathbf{A}_{n\times n}\mathbf{b}_{n\times 1}}$: $\underline{N = 2n^2}$ (vsak element približno n množenj in n seštevanj)

Scalar product:

$c = \mathbf{a}_{n\times 1}{}^T\mathbf{b}_n$: $N = 2n$ (n množenj in n seštevanj)

9.2 Special systems of equations

9.2.1.1 Lower triangular system

$$\mathbf{L}_{n\times n}\,\mathbf{y}_{n\times 1}=\mathbf{b}_{n\times 1}\,,i<j\Rightarrow l_{ij}=0,\,l_{ii}=1\,\underline{:}\,N=\frac{1}{2}n^2-\frac{1}{2}n$$

this follows from $N=\sum_{r=1}^{n-1}\left(n-r\right)=\frac{1}{2}n^2-\frac{1}{2}n$

9.2.1.2 Upper triangular system

$$\mathbf{U}_{n\times n}\,\mathbf{y}_{n\times 1}=\mathbf{b}_{n\times 1}\,,i>j\Rightarrow u_{ij}=0\,\underline{:}\,N=\frac{1}{2}n^2+\frac{1}{2}n$$

this follows from $N=\sum_{i=1}^{n}\left(n-i+1\right)=\frac{1}{2}n^2+\frac{1}{2}n$; The difference with respect to lower triangular system is that diagonal elements are different than 1, which adds *n* divisions.

The total number of operations requires $\frac{1}{2}n^2+\frac{1}{2}n$ multiplications or divisions and $\frac{1}{2}n^2-\frac{1}{2}n$ summations or subtractions. This is four times as much multiplications as Gaussian elimination and twice as much summations, beside the $n\left(n-1\right)/2$ square roots.

9.2.1.3 Orthogonal system

$$\mathbf{Q}_{n\times n}\,\mathbf{y}_{n\times 1}=\mathbf{b}_{n\times 1}\,,\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}\ :\ N=3n^2-3n\,,$$

i.e. $2n^2-2n$ multiplications and n^2-n summations

9.3 Factorizations

9.3.1.1 LU factorization

$\mathbf{A}_{n\times n} = \mathbf{L}_{n\times n}\,\mathbf{U}_{n\times n}$: $N = \frac{1}{3}n^3 - \frac{1}{3}n$

Solution of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$: $N = \frac{1}{3}n^3 + n^2 - \frac{1}{3}n$

(factorization + lower triangular + upper triangular system)

Solution of *m* systems of equations: $N = \frac{1}{3}n^3 + mn^2 - \frac{1}{3}n$

Determinant of a matrix $\det \mathbf{A}_{n\times n}$: $\frac{1}{3}n^3 + \frac{2}{3}n - 1$

(factorization + *n* multiplications of diagonal terms of **U**)

Inverse matrix: n^3 operations

(by solution of $\mathbf{A}\mathbf{X} = \mathbf{I}$, solution with the *s*-th unit vector on the right is the *s*-th column of **X**) Decomposition: $N = \frac{1}{3}n^3 - \frac{1}{3}n$ operations, lower triangular with $\mathbf{e}_i$ as right hand sides (less operations): $\frac{1}{6}n^3 - \frac{1}{2}n^2 + \frac{1}{3}n$ upper triangular: $n^2/2 + n^2/2$, toal: n^3

9.3.1.2 QR Factorization

9.3.1.2.1 Reduced factorization with the modified Gram-Schmidt method:

$\mathbf{A}_{m\times n} = \tilde{\mathbf{Q}}_{m\times n}\,\tilde{\mathbf{R}}_{n\times n}$; $\tilde{\mathbf{Q}}^T\mathbf{Q} = \mathbf{I}$, $i > j \Rightarrow r_{ij} = 0$: $N = 2mn^2$.

Solution of a system of equations:

$N = 2mn^2$ for factorization, $3m^2 - 3m$ for solution of the orthogonal system, and n^2 operations for upper triangular system, together.

9.3.1.2.2 QR factorization with Givens method

$\mathbf{A}_{n\times n} = \mathbf{Q}_{n\times n}\,\mathbf{R}_{n\times n}$; $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, $i > j \Rightarrow r_{ij} = 0$: $N = \left(\frac{1}{2}n^2 - \frac{1}{2}n\right) + \left(\frac{4}{3}n^3 - \frac{4}{3}n\right) + \frac{2}{3}n^3 - \frac{1}{2}n^2 - \frac{1}{6}n$

$N = 2n^3 - 2n$

$\frac{1}{2}n^2-\frac{1}{2}n$ square roots, $\frac{4}{3}n^3-\frac{4}{3}n$ divisions or multiplications and $\frac{2}{3}n^3-\frac{1}{2}n^2-\frac{1}{6}n$ summations

Solution of a system of equations:

$N=2n^3-2n$ for factorization, $3n^2-3n$ for solution of the orthogonal system, and n^2 operations for upper triangular system, together $2n^2+n^2-5n$ operations.

9.3.1.2.3 QR factorization with the Housholder method

$\mathbf{A}_{n\times n}=\mathbf{Q}_{n\times n}\mathbf{R}_{n\times n};\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}, i>j\Rightarrow r_{ij}=0\ :\ N=(n-1)+\left(\frac{2}{3}n^3+O\left(n^2\right)\right)+\left(\frac{2}{3}n^3+O\left(n^2\right)\right)$ - twice less multiplications as the Givens method and twice as much as Gauss elimination. Total number of operations is for 1/3 less than at Givens method and twice greater than with Gauss elimination.

$n-1$ square roots, $\frac{2}{3}n^3+O\left(n^2\right)$ divisions or multiplications and $\frac{2}{3}n^3+O\left(n^2\right)$ summations.

Solution of a system of equations:

$\frac{4}{3}n^3+O\left(n^2\right)$ for factorization, $3n^2-3n$ for solution of the orthogonal system, and n^2 operations for upper triangular system.

9.3.1.3 LDMT factorization

$$\mathbf{A}_{n\times n}=\mathbf{L}_{n\times n}\mathbf{D}_{n\times n}\mathbf{M}^T_{n\times n};\ i<j\Rightarrow l_{ij}=0\wedge m_{ij}\neq 0,\ i\neq j\Rightarrow d_{ij}=0:\ N=\frac{1}{3}n^3$$

9.3.1.4 LDLT factorization

$$\mathbf{A}_{n\times n}=\mathbf{L}_{n\times n}\mathbf{D}_{n\times n}\mathbf{L}_{n\times n};\ i<j\Rightarrow l_{ij}=0,\ i\neq j\Rightarrow d_{ij}=0:\ N=\frac{1}{3}n^3$$

9.3.1.5 Cholesky factorization

$$N = \frac{1}{6}n^3 + O\left(n^2\right)$$

References:

[1] Alfio Quarteroni, Riccardo Sacco, Fausto Saleri, *Numerical Mathematics*. Texts in Applied Mathematics 37, Springer-Verlag, New York, 2000.

[2] Richard L. Burden, J. Douglas Faires, *Numerical Analysis*, 6th Edition, Brooks/Cole Publishing Company, 1997.

[3] James W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[4] Lloyd N. Trefethen, David Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[5] I. N. Bronstein, K. A. Smendljajew, G. Musiol, H. Mühlig, *Taschenbuch des Mathematik (second edition - in German)*, Verlag Harri Deutsch, Frankfurt am Main, 1995.

[6] I. Kuščer, A. Kodre, H. Neunzert, *Mathematik in Physik und Technik (in German)*, Springer - Verlag, Heidelberg, 1993.

[7] I. Kuščer, A. Kodre, *Matematika v fiziki in tehniki (in Slovene)*, Društvo matematikov, fizikov in astronomov Slovenije, Ljubljana, 1994.

[8] W.H. Press, S.S. Teukolsky, V.T. Vetterling, B.P. Flannery, *Numerical Recipies in C – the Art of Scientific Computing*, Cambridge University Press, Cambridge, 1992.

[9] K.J. Bathe, *Finite Element Procedures*, p.p. 697-745, Prentice Hall, New Yersey, 1996.

[10] Z. Bohte, *Numerične metode* (in Slovene), Društvo matematikov, fizikov in astronomov SRS, Ljubljana, 1987.

[11] Zvonimir Bohte, *Numerično reševanje sistemov linearnih enačb* (in Slovene), Društvo fizikov, matematikov in astronomov Slovenije, Ljubljana, 1994.

[12] Zvonimir Bohte, *Numerično reševanje nelinearnih enačb* (in Slovene), i Društvo fizikov, matematikov in astronomov Slovenije, Ljubljana, 1993.

[13] Egon Zakrajšek, *Matematično modeliranje* (in Slovene), DMFA – založništvo, 2004.

[14] Jože Petrišič, *Reševanje enačb* (in Slovene), Univerza v Ljubljani – Fakulteta za strojništvo, 2006.

[15] Jože Petrišič, *Interpolacija in osnove računalniške grafike* (in Slovene), Univerza v Ljubljani – Fakulteta za strojništvo, 1999.

[16] Juan Restrepo: Numerical Linear Algebra, electronic document at http://www.physics.arizona.edu/~restrepo/475A/Notes/sourcea/node60.html

[17] Igor Grešovnik: Optimization shell Inverse, http://www2.arnes.si/~ljc3m2/inverse/

[18] Igor Grešovnik: Investigative Optimization Library, http://www2.arnes.si/~ljc3m2/igor/ioptlib/

[19] Igor Grešovnik: Investigative Generic Library, http://www2.arnes.si/~ljc3m2/igor/iglib/

[20] Igor Grešovnik: The Use of Moving Least Squares for a Smooth Approximation of Sampled Data, Journal of Mechanical Engineering 53 (2007) 9, 582-598 (UDK 517.518.8:519.243, 2007), http://www2.arnes.si/~ljc3m2/igor/doc/papers/07_09_Gresovnik_Mech_Eng_MLS_Sampled_Data_Optimization.pdf