COMPUTATIONAL

LINEAR ALGEBRA

Les S. Jennings
les@maths.uwa.edu.au
http://maths.uwa.edu.au/~les/

Notes for third year and honours courses in
Numerical Analysis
Printed 3 May 2001

The University of Western Australia
DEPARTMENT of MATHEMATICS AND STATISTICS

CONTENTS.

APPENDICES.

1. Introduction.

This set of notes on Computational Linear Algebra contains a mathematical description of operations with matrices and vectors on a modern computer. They have been used for courses in numerical analysis at the University of Western Australia for some time. In particular, the solution of sets of linear equations is studied and an overview of eigenvalue–eigenvector computations is given. Some of the material, for example, singular value decomposition and generalized inverses is not relevant to third year students, being more advanced, while some early chapters revise second year work. Chapters 6–8 contain the mathematical description of Gauss Elimination and the Q-U factorization. Students are expected to consult other more complete references where applicable.

References to the computing library packages are also included and it is expected that these will be consulted when necessary. Note that the top quality computing libraries are written in FORTRAN but that on some computers the Pascal and C compilers have means to communicate with FORTRAN subroutines and functions. There is a slow trickle of conversions of the FORTRAN libraries into C language libraries. One problem when using both FORTRAN and Pascal or C is that arrays are stored differently; in FORTRAN by columns, in Pascal by rows, and in C by user choice but usually rows. The easiest way to overcome this problem for Pascal is to transpose the matrix in Pascal. Care has to be taken with the FORTRAN column length variable which equates with the Pascal row length. Similar interface problems exist with the C language. Users of C libraries must inform themselves of how a matrix is stored into an appropriate C data structure.

Many of the computational exercises contained herein could be computed with MATLAB, a commercially available high level system for numerical computing. In particular, MATLAB is a good interface to the NSF funded software libraries LINPACK (linear equations package) and EISPACK (eigen systems package), and now LAPACK (linear algebra package) which supercedes LINPACK and EISPACK. There is a companion library of basic linear algebra subroutines (BLAS) which can be tailored to particular machine architectures for efficiency. The NSF funded libraries are available over the network, by email, from

`netlib@ornl.gov`

and by the WEB, at

`http://www.netlib.org/index.html`

MATLAB information is available on the WEB at

`http://www.mathworks.com`

There are some freely available MATLAB look-alikes.

SCINET is available from

`http://www-rocq.inria.fr/scilab`

and OCTAVE from

`http://www.che.wisc.edu/octave/`

Appendices contain references to software and descriptions of some local software useful for teaching purposes.

2. Errors and Computer Arithmetic.

Suggested extra reading in Atkinson (1989), Dahlquist *et al.* (1974), Stewart (1973). The beginnings of this work can be found in Wilkinson (1963, 1965).

2.1 Accuracy.

Suppose x represents an exact quantity and $\overline{x}$ an approximation to it. Define the following types of accuracy with respect to some small quantity ϵ.

(i) Absolute accuracy: $\overline{x}$ is within ϵ of x absolutely if

$$|x - \overline{x}| < \epsilon.$$

This gives a certain number of decimal places accuracy (the same in both $\overline{x}$ and x) determined by ϵ.

(ii) Relative accuracy: $\overline{x}$ is within ϵ of x in a relative sense if

$$|x - \overline{x}| < \epsilon|x|, \quad x \neq 0.$$

This gives a certain number of significant digits accuracy, determined by ϵ.

(iii) Mixed test:
$$|x - \overline{x}| < \epsilon(a + |x|)$$

This test is useful if it is known that x is of magnitude a. If $|x| >> a$ it acts as a relative test while if $|x| << a$ it acts as a relative test against a. For most purposes the relative or mixed test should be used.

2.2 Precision.

Whenever a number expressed in decimal notation is stored on a computer (in binary or hexadecimal form) truncation (chopping) or rounding occurs in the last binary bit. Hence one computer number will represent a range of numbers on the real line. The relative (compared to the number) length of this range is the *precision* of the computer. The error between the number and its computer representation is known as *rounding error* and is a function of computer hardware. A suitable definition of precision of a computer (language/compiler) is the smallest positive computer number ϵ such that

$$1 + \epsilon \neq 1,$$

when executed in that computer (language/compiler).

Precision of FORTRAN variables on computers.

computer	single precision	double precision	long precision
CDC 6000 series	$2^{-47} \sim 10^{-14}$	$2^{-95} \sim 10^{-29}$	—
INTEL 80x87	$2^{-23} \sim 10^{-7}$	$2^{-52} \sim 10^{-16}$	—
SUN SPARC	$2^{-23} \sim 10^{-7}$	$2^{-52} \sim 10^{-16}$	—
HP 9000	$2^{-23} \sim 10^{-7}$	$2^{-52} \sim 10^{-16}$	—
DEC MIPS	$2^{-23} \sim 10^{-7}$	$2^{-52} \sim 10^{-16}$	—
DEC VAX	$2^{-26} \sim 10^{-8}$	$2^{-58} \sim 10^{-18}$ (D)	—
	—	$2^{-52} \sim 10^{-16}$ (G)	$2^{-102} \sim 10^{-31}$ (G)
DEC ALPHA	$2^{-26} \sim 10^{-8}$	$2^{-58} \sim 10^{-18}$ (D)	—
	—	$2^{-52} \sim 10^{-16}$ (G)	$2^{-102} \sim 10^{-31}$ (G)

Note that rounding error and precision are expressed as a relative error (significant binary digits). The precision of a computer should not be confused with the smallest positive number of a computer. There is now an IEEE standard for storage of floating point numbers and for their arithmetic. Not all manufacturers use this standard. Most numerical computation should be done at a precision smaller that 10^{-10}, which means that except in special circumstances, single precision should not be used on most machines.

2.3 Arithmetic unit errors.

Actual machine arithmetic generates further errors. Let x and y be two computer numbers and θ be one of the operations $+$, $-$, $*$ or $/$. Then the computer number stored which represents $x\theta y$ is denoted $fl(x\theta y)$ and,

$$fl(x\theta y) = (x\theta y)(1 + \epsilon(x, y)),$$

where $\epsilon(x, y)$ represents the relative error. The object of computer manufacturers is to have $|\epsilon(x, y)| \le$ machine precision for all choices of computer numbers x and y. (See Kahan (1966)) Most computers use double length accumulators for arithmetic so effectivly it can be assumed that no errors are created in the arithmetic. But when the double length result is stored back to single length in memory an error is created. This is not necessarily true for all computers, especially at double precision.

2.4 Backward error analysis.

This relates the error in a computed result back to contrived errors in the operands, that is,

$$fl(x\theta y) = (x + \delta x)\theta(y + \delta y).$$

The actual result is considered to be the exact result using perturbed data. Sometimes it is easier to put bounds on δx and δy than on $\epsilon(x, y)$. Here the operands may be matrices and θ represents a matrix operation.

2.5 Loss of precision.

(i) *Cancellation* (of significant digits)— occurs when two nearly equal numbers are subtracted, for example,

$$f'(x) = (f(x + h) - f(x))/h.$$

This is the most serious mechanism of loosing accuracy when executing millions of instructions. It could be hidden deeply within an algorithm.

(ii) *Underflow* — occurs when summing numbers which differ vastly in magnitude. For example on 32 bit hexadecimal machines

$$fl(1 + 10^{-7}) = 1.$$

This can effect statistical computation and the like in that the result of summing thousands of numbers depends on the order in which they are summed.

$$e.g. \quad fl\left(1.0 + \sum_{i=1}^{10^7} 10^{-7}\right) \ne 2 \ne fl\left(\sum_{i=1}^{10^7} 10^{-7} + 1.0\right).$$

(iii) *Scaling* problems are usually critical. Basically it means trying to bring significant figure accuracy to be the same as decimal place accuracy. If possible all numbers should be scaled to be of magnitude 1.0. For example the matrix $\begin{pmatrix} 1.111 & 1.234 \times 10^{-7} \\ 2.222 & 1.234 \times 10^{-7} \end{pmatrix}$ could be considered singular, but if the numbers are accurate to four significant figures the second column should be scaled to give the matrix $\begin{pmatrix} 1.111 & 1.234 \\ 2.222 & 1.234 \end{pmatrix}$ which is far from singular. Care must be taken to use the full significant accuracy of any given data, especially in matrix computations.

2.6 Exercises.

1. Find the error bounds both absolute and relative for storing (from a decimal number), computing and storing the result of $a + b$, $a - b$, $a * b$, and a/b in terms of a computer with precision ϵ.

2. Given two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ and a computer with precision ϵ, what is an upper bound for the error in storing, computing and storing the result of $\boldsymbol{a} + \boldsymbol{b}$ and $\boldsymbol{a}^t\boldsymbol{b}$ in terms of the norms of $\boldsymbol{a}$ and $\boldsymbol{b}$? You need to follow the detailed algorithm for these two operations on the vectors.

3. For a new computer chip using 52 bit mantissa, how long would it take to check every mantissa bit combination for one of the four basic arithmetic operations? Use a realistic time to test a pair of numbers, and, assume there are 10^7 seconds in a year.

3. Euclidean Vector Spaces.

Consult any linear algebra texts, such as, Anton and Rorres (1987), Lipschutz (1968), Strang (1988), Noble (1969), Noble and Daniel (1977) for a fuller discussion of the material in chapters 3-5.

3.1 Revision of the vector spaces $\mathbb{R}^n$.

3.1.1 Notation.

The finite dimensional vector spaces considered are the Euclidean vector spaces of dimension n, denoted $\mathbb{R}^n$ or $\mathbb{E}^n$. They have elements $\boldsymbol{x}$ written as column vectors so that

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad x_i \in \mathbb{R} = (-\infty, \infty).$$

For convenience, write $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^t$, where $\cdot^t$ represents the operation 'transpose'. Similarly $\mathbb{C}^n$ denotes the n dimensional vector space where $x_i \in \mathbb{C}$, the field of complex numbers.

(i) *The Unit Vectors:* $\boldsymbol{e}_i$ is the i–th unit vector of *the standard basis* for $\mathbb{R}^n$.

$$\boldsymbol{e}_i = (0, 0, \ldots, 1, \ldots, 0)^t$$

where the '1' is in the i–th position. Note that x_i and $\boldsymbol{x}_i$ are different, one being the i–th component of a vector, the other being the i–th vector of a sequence of vectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots\}$.

(ii) The vector $\boldsymbol{e} = (1, 1, 1, \ldots, 1)^t$ will occur occasionally.

(iii) Lower case greek letters are always scalars.

3.1.2 Inner product.

An inner product on $\mathbb{R}^n$, denoted $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ is a binary operation on two vectors, producing a scalar, with the properties, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and $\alpha, \beta \in \mathbb{R}$;

(i) $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{y}, \boldsymbol{x} \rangle$ (commutative),

(ii) $\langle \alpha \boldsymbol{x} + \beta \boldsymbol{y}, \boldsymbol{z} \rangle = \alpha \langle \boldsymbol{x}, \boldsymbol{z} \rangle + \beta \langle \boldsymbol{y}, \boldsymbol{z} \rangle$,

(iii) $\langle \boldsymbol{x}, \boldsymbol{x} \rangle > 0$, $\boldsymbol{x} \neq \boldsymbol{0}$.

The standard inner product for $\mathbb{R}^n$ is

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{n} x_i y_i = \boldsymbol{x}^t \boldsymbol{y} = \boldsymbol{y}^t \boldsymbol{x}.$$

The definition is changed for $\mathbb{C}^n$, the vector spaces over the field of complex numbers. Item (i) is changed to $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \overline{\langle \boldsymbol{y}, \boldsymbol{x} \rangle}$ where $\overline{z}$ is the complex conjugate of the complex number z. Combined with (ii) above this gives $\langle \boldsymbol{x}, \beta \boldsymbol{y} \rangle = \overline{\beta} \langle \boldsymbol{x}, \boldsymbol{y} \rangle$. The standard inner product for $\mathbb{C}^n$ is $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^t \overline{\boldsymbol{y}} = \overline{\boldsymbol{y}}^t \boldsymbol{x}$. The operation of conjugate transpose is sometimes denoted $\boldsymbol{x}^* = \overline{\boldsymbol{x}}^t$.

3.1.3 Vector norms.

The norm of a vector $\boldsymbol{x}$, is a function from $\mathbb{R}^n \to \mathbb{R}$ (or $\mathbb{C}^n \to \mathbb{R}$), denoted $\|\boldsymbol{x}\|$, if the following properties hold:

(i) $\|\boldsymbol{x}\| > 0$ for $\boldsymbol{x} \neq \boldsymbol{0}$.

(ii) $\|\alpha \boldsymbol{x}\| = |\alpha| \, \|\boldsymbol{x}\|$.

(iii) $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ (triangle inequality).

Examples:

$$\|\boldsymbol{x}\|_2 = \{\sum_{i=1}^{n} x_i^2\}^{\frac{1}{2}} = \langle \boldsymbol{x}, \boldsymbol{x}\rangle^{\frac{1}{2}} \quad : \quad \text{Euclidean norm or } 2-\text{norm}$$

$$\|\boldsymbol{x}\|_\infty = \max_i |x_i| \quad : \quad \infty-\text{norm}$$

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i| \quad : \quad 1-\text{norm}$$

$$\|\boldsymbol{x}\|_p = \left\{\sum_{i=1}^{n} |x_i|^p\right\}^{\frac{1}{p}}, 1 < p < \infty \quad : \quad p-\text{norm}$$

3.1.4 Linear independence.

Vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_r$, $(\{\boldsymbol{x}_i\}_{i=1}^r)$ are linearly independent if

$$\sum_{i=1}^{r} a_i \boldsymbol{x}_i = \boldsymbol{0} \Rightarrow a_i = 0, \ i = 1, 2, \ldots, r.$$

This means no non-zero linear combination of the $\{\boldsymbol{x}_i, \ i = 1, \ldots, r\}$ sum to zero.

3.1.5 Linear dependence.

At least one of the $\boldsymbol{x}_i$ can be expressed as a linear combination of the others,

$$\boldsymbol{x}_i = \sum_{\substack{j \neq i \\ j=1}}^{r} a_j \boldsymbol{x}_j.$$

This means there exists a non zero linear combination of the $\boldsymbol{x}_i$ which sum to zero.

3.1.6 Span.

Span $\{\boldsymbol{x}_i\}_{i=1}^r$ is the subspace formed by all linear combinations of the vectors $\boldsymbol{x}_i$.

$$\text{span}\{\boldsymbol{x}_i\}_{i=1}^r = \{\boldsymbol{y} \mid \boldsymbol{y} = \sum_{i=1}^{r} a_i \boldsymbol{x}_i, \ a_i \in \mathbb{R}\}$$

3.1.7 Basis.

A *basis* of a vector space is a set of linearly independent vectors which have a span equal to the vector space.

3.1.8 Dimension.

The *dimension* of a subspace is the number of vectors in a basis for the subspace.

3.2 Orthogonality

3.2.1 Orthogonal vectors.

Two non-zero vectors $\boldsymbol{x}, \boldsymbol{y}$ are orthogonal if their inner product $\langle \boldsymbol{x}, \boldsymbol{y}\rangle$ is zero, for example $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ are orthogonal if $i \neq j$, for the standard inner product.

3.2.2 Orthogonal subspaces.

Two subspaces X and Y are orthogonal if for all $\boldsymbol{x} \in X$ and for all $\boldsymbol{y} \in Y$, $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = 0$.

3.2.3 Orthogonal complement.

The *orthogonal complement* of a subspace X , denoted $X^\perp$ is defined as

$$X^\perp = \{\boldsymbol{y} \mid \langle \boldsymbol{x}, \boldsymbol{y}\rangle = 0 \text{ , for all } \quad \boldsymbol{x} \in X\}.$$

3.2.4 Orthogonal basis.

An *orthogonal basis* of a subspace is a basis $\{\boldsymbol{x}_i\}_{i=1}^r$ in which the basis elements are mutually orthogonal, that is

$$\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = 0 \ , \ i \neq j.$$

3.2.5 Orthonormal basis.

An *orthonormal basis* of a subspace is an orthogonal basis in which the basis elements are normalized to unit norm,

$$\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = \delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

3.3 Exercises.

1. Find out how to find an orthonormal basis using Gram-Schmidt orthogonalization.
2. Show $(X^\perp)^\perp = X$ and $X^\perp \cap X = \{\boldsymbol{0}\}$.
3. If the subspace $X \subset \mathbb{R}^n$ show $X \oplus X^\perp = \mathbb{R}^n$ where $Y \oplus Z$ is the set $\{\boldsymbol{x} \mid \boldsymbol{x} = \boldsymbol{y} + \boldsymbol{z}, \boldsymbol{y} \in Y, \boldsymbol{z} \in Z\}$.
4. Show that $\|\boldsymbol{x}\|_\infty \leq \|\boldsymbol{x}\|_2 \leq \|\boldsymbol{x}\|_1$, for all $\boldsymbol{x} \in \mathbb{R}^n$.
5. Draw the sets $\{\boldsymbol{x} | \boldsymbol{x} \in \mathbb{R}^n, \ \|\boldsymbol{x}\|_k = 1\}$, for $k = 1, 2$ and ∞, and $n = 2$ and 3.
6. Show that any inner product (not just the standard one) on $\mathbb{R}^n$ defines a natural norm, called the inner product norm, by the definition
$$\|\boldsymbol{x}\| = \langle \boldsymbol{x}, \boldsymbol{x} \rangle^{\frac{1}{2}}.$$

4. Linear Mappings, Matrices.

4.1 Linear mapping (linear transformation).

Let $\mathcal{A}$ be a mapping of $\mathbb{R}^n$ to $\mathbb{R}^m$. $\mathcal{A}$ is linear if $\mathcal{A} : \boldsymbol{x} \to \boldsymbol{w}$, $\mathcal{A} : \boldsymbol{y} \to \boldsymbol{z}$ then

$$\mathcal{A} : \alpha\boldsymbol{x} + \beta\boldsymbol{y} \to \alpha\boldsymbol{w} + \beta\boldsymbol{z}.$$

A linear mapping of $\mathbb{R}^n \to \mathbb{R}^m$ is represented in the usual basis by an $m \times n$ matrix $\boldsymbol{A}$ (say). Write

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$$

to describe $\mathcal{A} : \boldsymbol{x} \to \boldsymbol{y}$.

4.1.1 Notation.

(i) A_{ij} (or $(A)_{ij}$) represents the element in the i–th row and j–th column of the matrix $\boldsymbol{A}$.

(ii) $\rho_i(\boldsymbol{A})$ represents the i–th row of $\boldsymbol{A}$ as a row vector.

(iii) $\kappa_j(\boldsymbol{A})$ represents the j–th column of $\boldsymbol{A}$ as a column vector.

(iv) The transpose of the matrix $\boldsymbol{A}$ is the matrix $\boldsymbol{A}^t$ (or $\boldsymbol{A}'$) defined,

$$\left(A^t\right)_{ij} = A_{ji}, \ i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, m.$$

Note that A^t represents a mapping from $\mathbb{R}^m$ to $\mathbb{R}^n$.

(v) The matrix-vector product $\boldsymbol{A}\boldsymbol{x} = \sum_{i=1}^{n} x_i \kappa_i(\boldsymbol{A})$, that is, a linear combination of columns of the matrix $\boldsymbol{A}$.

4.1.2 Adjoint.

The *adjoint* mapping $\boldsymbol{A}^*$ is defined by the relation

$$\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{x}, \boldsymbol{A}^*\boldsymbol{y} \rangle \ , \forall \boldsymbol{x} \in \mathbb{R}^n \ \text{and} \ \boldsymbol{y} \in \mathbb{R}^m.$$

4.1.3 Null space (kernel).

The *null space* (or *kernel*) of a mapping denoted $\mathcal{N}(\boldsymbol{A})$ (or $\ker \boldsymbol{A}$) is

$$\mathcal{N}(\boldsymbol{A}) = \{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}\} \subset \mathbb{R}^n.$$

4.1.4 Range space.

The *range* (or *image*) of a mapping, denoted $\mathcal{R}(\boldsymbol{A})$ (or $\mathcal{I}m\boldsymbol{A}$) is

$$\mathcal{R}(\boldsymbol{A}) = \{\boldsymbol{y} \mid \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \quad \text{for some} \quad \boldsymbol{x} \in \mathbb{R}^n\} \subset \mathbb{R}^m.$$

4.1.5 Rank.

The *rank* of a matrix $\boldsymbol{A}$ is the dimension of $\mathcal{R}(\boldsymbol{A})$.

4.2 Some alternative definitions.

Let the set of vectors $\{\boldsymbol{x}_i\}_{i=1}^{r}$ be represented by the $n \times r$ matrix $\boldsymbol{X}$ where $\kappa_i(\boldsymbol{X}) = \boldsymbol{x}_i$, for $i = 1, 2, \ldots, r$. Then the following are alternative definitions.

Note :
$$\sum_{i=1}^{r} a_i \boldsymbol{x}_i = \boldsymbol{X}\boldsymbol{a} \ , \ \boldsymbol{a} = (a_1, a_2, \ldots, a_r)^t.$$

(i) Linear independence; $\nexists$ non zero $\boldsymbol{a}$ such that $\boldsymbol{X}\boldsymbol{a} = \boldsymbol{0}$, or $\mathcal{N}(\boldsymbol{X}) = \{\boldsymbol{0}\}$.

(ii) Linear dependence; $\exists$ a non zero $\boldsymbol{a}$ such that $\boldsymbol{X}\boldsymbol{a} = \boldsymbol{0}$, or $\mathcal{N}(\boldsymbol{X}) \neq \{\boldsymbol{0}\}$.

(iii) Span $(\boldsymbol{X}) = \{\boldsymbol{y} \mid \boldsymbol{y} = \boldsymbol{X}\boldsymbol{a} \ , \ \boldsymbol{a} \in \mathbb{R}^r\}, \quad (= \boldsymbol{X}(\mathbb{R}^n))$.

(iv) Orthogonal basis; $\boldsymbol{X}^t\boldsymbol{X} = \boldsymbol{D} = \text{diag}\,[d_1, d_2, \ldots, d_r], \ d_i \neq 0, \ i = 1, 2, \ldots, r$.

(v) Orthonormal basis; $\boldsymbol{X}^t\boldsymbol{X} = \boldsymbol{I}_r$, the identity matrix of order r.

4.3 Inverse mapping. $(\mathcal{A}^{-1} : \mathbb{R}^m \to \mathbb{R}^n)$

$$\mathcal{A}^{-1}\mathcal{A} = \mathcal{A}\mathcal{A}^{-1} = \text{Identity mapping}$$
$$\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I}$$

The inverse mapping $\mathcal{A}^{-1}$ exists and is represented by the inverse matrix $\boldsymbol{A}^{-1}$, in the usual basis, provided $\mathcal{A}$ is 1:1 and onto (and hence $m = n$).

$$onto \Leftrightarrow \mathcal{R}(\boldsymbol{A}) = \mathbb{R}^m$$
$$1:1 \Leftrightarrow \mathcal{N}(\boldsymbol{A}) = \{\boldsymbol{0}\}$$

4.4 Orthogonal matrices.

$\boldsymbol{Q}$ is an orthogonal matrix if
$$\boldsymbol{Q}\boldsymbol{Q}^t = \boldsymbol{Q}^t\boldsymbol{Q} = \boldsymbol{I} \ ,$$

that is $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^t$. This means the rows and columns of $\boldsymbol{Q}$ are orthonormal.

4.5 Matrix norms.

A matrix norm measures the 'magnitude' of a matrix .

(a) The function $\|\boldsymbol{A}\|$ is a matrix norm if the four properties below are true;

   (i) $\|\boldsymbol{A}\| > 0$, if $\boldsymbol{A} \neq \boldsymbol{0}$,

   (ii) $\|\alpha\boldsymbol{A}\| = |\alpha|\,\|\boldsymbol{A}\|$,

   (iii) $\|\boldsymbol{A} + \boldsymbol{B}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$,

   (iv) $\|\boldsymbol{A}\boldsymbol{B}\| \leq \|\boldsymbol{A}\|\,\|\boldsymbol{B}\|$.

(b) Operator norms (subordinate norms). Given norms for $\mathbb{R}^n$ and $\mathbb{R}^m$,

$$\|\boldsymbol{A}\| = \max_{\boldsymbol{x} \neq 0} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|} = \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{A}\boldsymbol{x}\|$$

is an operator norm or subordinate norm.

   (i) $\|\boldsymbol{A}\|_1 = \max\limits_{\|x\|_1=1} \|\boldsymbol{A}\boldsymbol{x}\|_1 = \max\limits_{j} \sum_{i=1}^{m} |A_{ij}|$. (Maximum column sum.)

   (ii) $\|\boldsymbol{A}\|_2 = \max\limits_{\|x\|_2=1} \|\boldsymbol{A}\boldsymbol{x}\|_2 = \max\limits_{i}\{\lambda_i(\boldsymbol{A}^t\boldsymbol{A})\}^{\frac{1}{2}}$.   The 2-norm of a matrix is the square root of the largest eigenvalue of $\boldsymbol{A}^t\boldsymbol{A}$.

   (iii) $\|\boldsymbol{A}\|_\infty = \max\limits_{\|x\|_\infty=1} \|\boldsymbol{A}\boldsymbol{x}\|_\infty = \max\limits_{i} \sum_{j=1}^{n} |A_{ij}|$. (Maximum row sum.)

(c) Euclidean norm of a matrix (Frobenius norm)

$$\|\boldsymbol{A}\|_E = \Big\{\sum_{i,j} A_{ij}^2\Big\}^{\frac{1}{2}}.$$

(d) If $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, then $\|\boldsymbol{y}\| \leq \|\boldsymbol{A}\|\,\|\boldsymbol{x}\|$, for appropriate 'same' norms, as in (b).

4.6 Change of basis.

Many of the factorizations of Chapter 6 have an explanation in terms of a change of basis, either of the domain or range of a mapping, or of both. Chapter 9 requires a good understanding of this section.

4.6.1 Coordinates of a vector.

All vectors are usually expressed in terms of *the unit vectors* of that space, for example

$$
\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} a + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} b + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} c = \boldsymbol{I} \begin{pmatrix} a \\ b \\ c \end{pmatrix} .
$$

Here a, b and c are the coordinates of the vector. More generally, in the standard basis

$$
\boldsymbol{x} = \sum_{i=1}^{n} \boldsymbol{e}_i x_i \quad \left(= \sum_{i=1}^{n} \langle \boldsymbol{e}_i, \boldsymbol{x} \rangle \boldsymbol{e}_i \right) \quad = \boldsymbol{I}\boldsymbol{x}, \ \boldsymbol{x} \in \mathbb{R}^n.
$$

4.6.2 Change of basis.

Choose another basis $\{\boldsymbol{b}_i\}_{i=1}^{r}$ which can be represented (in 'old' coordinates) by the $n \times r$ matrix $\boldsymbol{B}$, where $\kappa_i(\boldsymbol{B}) = \boldsymbol{b}_i$, $i = 1, 2, \ldots, r$. The equation

$$
\boldsymbol{x} = \boldsymbol{B}\boldsymbol{a}, \ \boldsymbol{a} \in \mathbb{R}^r,
$$

means that $\boldsymbol{x}$ can be represented in the basis $\boldsymbol{B}$ by the coordinate r-tuple $\boldsymbol{a}$. Note that $\boldsymbol{x}$ must be an element of $\mathcal{R}(\boldsymbol{B})$. If $\boldsymbol{B}$ has an inverse ($r = n$) then $\boldsymbol{a}$ can be found as $\boldsymbol{B}^{-1}\boldsymbol{x}$, where $\boldsymbol{x}$ is the n-tuple for the vector $\boldsymbol{x}$ in terms of the standard basis. For vectors in $\mathbb{R}^n$ no distinction between the vector and its coordinate n-tuple is usually made, only because the standard basis has been chosen to represent the vector.

4.6.3 Rotation of axes.

This can be considered as a change of basis in which another orthonormal basis has been chosen. Let $\boldsymbol{Q}$ be a matrix such that $\boldsymbol{q}_i = \kappa_i(\boldsymbol{Q})$ and $\{\boldsymbol{q}_i\}_{i=1}^{r}$ be an orthonormal basis ($\boldsymbol{Q}^t\boldsymbol{Q} = \boldsymbol{I}_r$). Then

$$
\boldsymbol{x} = \sum_{i=1}^{r} \boldsymbol{q}_i a_i = \boldsymbol{Q}\boldsymbol{a}, \ \boldsymbol{a} \in \mathbb{R}^r.
$$

The term *rotation of axes* usually refers to the case $r = n$, so that any vector $\boldsymbol{x} \in \mathbb{R}^n$ can be represented in the new basis. In this case $\boldsymbol{x} = \boldsymbol{Q}\boldsymbol{a}$, so that $\boldsymbol{Q}^t\boldsymbol{x} = \boldsymbol{a}$ as $\boldsymbol{Q}^t\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^t = \boldsymbol{I}_n$, that is, $a_i = \langle \boldsymbol{q}_i, \boldsymbol{x} \rangle$ and

$$
\boldsymbol{x} = \sum_{i=1}^{n} \langle \boldsymbol{q}_i, \boldsymbol{x} \rangle \boldsymbol{q}_i \ .
$$

Hence the use of orthogonal matrices can be thought of as a change of orthonormal basis or rotation of axes. As such, the lengths of co-ordinate vectors ($\|\boldsymbol{x}\|_2$) remains unchanged under an orthogonal change of basis. In particular any vector of errors has its Euclidean size unchanged when multiplied by an orthogonal matrix.

4.7 Determinant of a matrix

See one of the texts Anton and Rorres (1987) or Lipschutz (1968) for the definition of the determinant of a matrix. The determinant of a matrix is only defined for a square matrix and the following properties are relevant to this set of notes. Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be $n \times n$ matrices.

(i) $\det(\boldsymbol{A}\boldsymbol{B}) = \det(\boldsymbol{A})\det(\boldsymbol{B})$.

(ii) $\det(\boldsymbol{A}) \neq 0$ iff $\boldsymbol{A}$ is invertible.

(iii) $\det(\boldsymbol{A}) = \Pi_{i=1}^{n}\lambda_i(\boldsymbol{A})$, the product of all the eigenvalues of $\boldsymbol{A}$, counting repeated eigenvalues.

(iv) $\det(c\boldsymbol{A}) = c^n \det(\boldsymbol{A})$, c a scalar.

4.8 Eigenvalues and eigenvectors

See one of the texts Anton and Rorres (1987) or Lipschutz (1968) for a fuller discussion. The eigenvalue–eigenvector pair of a matrix $\boldsymbol{A}$ is defined to be the solution of the equation

$$
\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}, \quad \boldsymbol{v} \neq \boldsymbol{0},
$$

or equivalently, the eigenvalues are those values of λ which make the matrix $\boldsymbol{A} - \lambda\boldsymbol{I}$ singular, and the eigenvector $\boldsymbol{v}$ is any nonzero vector in $\mathcal{N}(\boldsymbol{A} - \lambda\boldsymbol{I})$.

4.9 Exercises.
1. Check $A_{ij} = \boldsymbol{e}_i^t \boldsymbol{A} \boldsymbol{e}_j$, $\rho_i(\boldsymbol{A}) = \boldsymbol{e}_i^t \boldsymbol{A}$, $\kappa_j(\boldsymbol{A}) = \boldsymbol{A}\boldsymbol{e}_j$.
2. Prove $\mathcal{N}(\boldsymbol{A})^\perp = \mathcal{R}(\boldsymbol{A}^t)$, and $\mathcal{N}(\boldsymbol{A}^t)^\perp = \mathcal{R}(\boldsymbol{A})$.
3. (i) Show that $\boldsymbol{A}^* = \boldsymbol{A}^t$ if the mapping, represented by the $m \times n$ matrix $\boldsymbol{A}$ is real, and the usual inner products are used in $\mathbb{R}^m$ and $\mathbb{R}^n$.
   (ii) Show that $(\boldsymbol{A}\boldsymbol{B})^t = \boldsymbol{B}^t \boldsymbol{A}^t$.
4. (i) Show that $\mathcal{A}^{-1}$ exists iff $m = n = r$ (the rank of $\mathcal{A}$).
   (ii) Show that $(\boldsymbol{A}\boldsymbol{B})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$.
5. Show that for $\boldsymbol{Q}$ orthogonal, $\|\boldsymbol{Q}\boldsymbol{x}\|_2 = \|\boldsymbol{x}\|_2$, $\|\boldsymbol{Q}\|_2 = 1$, $\|\boldsymbol{Q}\boldsymbol{A}\|_2 = \|\boldsymbol{A}\|_2$ and $\|\boldsymbol{Q}\boldsymbol{A}\|_E = \|\boldsymbol{A}\|_E$.
6. Show that $\|\boldsymbol{A}\|_2 = \|\boldsymbol{A}^t\|_2$, and hence, from 5 above, that $\|\boldsymbol{A}\boldsymbol{Q}\|_2 = \|\boldsymbol{A}\|_2$, for $\boldsymbol{Q}$ orthogonal.
7. Show that $\|\boldsymbol{A}\|_E$ is not a subordinate norm for $m, n > 1$. (Hint: use $\boldsymbol{A} = \boldsymbol{I}$).
8. Show that the $1, 2$ and ∞ norms for $n \times 1$ matrices agree with the $1, 2$ and ∞ norms for column vectors. What about row vectors?
9. Show that any finite product of orthogonal matrices is also orthogonal. (What about an infinite product?)
10. A symmetric matrix $\boldsymbol{S}$ is positive definite if $\boldsymbol{x}^t \boldsymbol{S}\boldsymbol{x} > 0$, $\forall \boldsymbol{x} \neq \boldsymbol{0}$.
    (i) Show that $\langle \boldsymbol{x}, \boldsymbol{y}\rangle_S = \boldsymbol{x}^t \boldsymbol{S}\boldsymbol{y}$ is an inner product if $\boldsymbol{S}$ is positive definite.
    (ii) Let $\boldsymbol{A}$ be an $m \times n$ real matrix and let the inner product on $\mathbb{R}^m$ be defined as $\langle \boldsymbol{y}_1, \boldsymbol{y}_2\rangle = \boldsymbol{y}_1^t \boldsymbol{S}\boldsymbol{y}_2$, $\boldsymbol{S}$ positive definite, and the innerproduct on $\mathbb{R}^n$ be defined as $\langle \boldsymbol{x}_1, \boldsymbol{x}_2\rangle = \boldsymbol{x}_1^t \boldsymbol{T}\boldsymbol{x}_2$, $\boldsymbol{T}$ positive definite. What is the adjoint mapping of $\boldsymbol{A}$?
11. From the definition of a subordinate norm show that the expressions in §4.5(b)(i), (ii) and (iii) do satisfy the four properties of a matrix norm.
12. (a) Consider the set $S_1 = \{\boldsymbol{x} | \boldsymbol{x} \in R^2, \|\boldsymbol{x}\|_1 = 1\}$ (draw it) and the set of points $\{\boldsymbol{A}\boldsymbol{x} | \boldsymbol{x} \in S_1\}$ for a 2×2 matrix $\boldsymbol{A}$. Convince yourself that the vectors $\boldsymbol{x} = \pm\boldsymbol{e}_i \in S$, $i = 1, 2$ are the only vectors to consider in the definition of $\|\boldsymbol{A}\|_1$, and hence that the 1-norm of a matrix is the maximum column sum as in §4.5(b)(i).
    (b) Repeat part (a) for the 2-norm. The relevant vectors are different, and the eigenvalues and eigenvectors of $\boldsymbol{A}^t \boldsymbol{A}$ need to be considered. A linear programming exercise is part of the proof.
    (c) Repeat part (a) for the ∞-norm. The relevant vectors are different and of course the ∞-norm is defined in terms of the maximum row sum.
13. Consider the solution set S of the set of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$.
    (a) Show that if $\boldsymbol{E}$ is a compatible matrix then the solution set of $\boldsymbol{E}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{E}\boldsymbol{y}$ contains S.
    (b) Show that if $\boldsymbol{E}$ is invertible then the solution set of $\boldsymbol{E}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{E}\boldsymbol{y}$ equals S.
14. (a) Show that if $\|\boldsymbol{A}\| < 1$, then $\|\boldsymbol{A}^n\| \to 0$ as $n \to \infty$ for any matrix norm.
    (b) Show that if $\boldsymbol{A} = \begin{pmatrix} \lambda & a \\ 0 & \lambda \end{pmatrix}$, then $\boldsymbol{A}^n = \begin{pmatrix} \lambda^n & n\lambda^{n-1}a \\ 0 & \lambda^n \end{pmatrix}$. Hence show that for all $K > 0$, there exists $\boldsymbol{A}$ such that $\|\boldsymbol{A}\| > K$ and $\|\boldsymbol{A}^n\| \to 0$ as $n \to \infty$. That is $\|\boldsymbol{A}\| > 1$ does not imply $\|\boldsymbol{A}^n\| \to \infty$.
    (c) Does there exist a non-zero matrix $\boldsymbol{A}$ such that $\boldsymbol{A}^k = \boldsymbol{0}$ for $k < n$?
15. Show that while $\|\boldsymbol{x}\|_\infty \leq \|\boldsymbol{x}\|_2 \leq \|\boldsymbol{x}\|_1$, (exercise 3.3.4), no such relation holds for matrix norms, even though the corresponding matrix norms are defined in terms of vector norms. Experiment with the matrices, $\begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$, $\begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}$, $\begin{pmatrix} 1 & a \\ 0 & 2 \end{pmatrix}$, $\begin{pmatrix} 2 & a \\ 0 & 1 \end{pmatrix}$. Maple or MATLAB would be useful to find the 2-norms.
16. Find the co-ordinates of the vector $[2, 3]^t$, in terms of the orthogonal basis
$$\left\{ \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \; \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\}.$$
17. The two diagonal matrices of order 20,
$$\boldsymbol{D}_1 = 10^{-1}\boldsymbol{I}, \quad \text{and} \quad \boldsymbol{D}_2 = \text{diag}\,(1, 1, \ldots, 1, 10^{-20}),$$
both have the same determinant $10^{-20} \approx 0$. Which of these matrices is closer to being exactly singular? Think about this in terms of the size of perturbation required to make the matrix singular. This shows that the determinant of a matrix is not a good numerical indicator of closeness to singularity.

5. Areas of Application, Types of Matrices.

5.1 Classification of problems.

There are four major areas of application or interest of computational linear algebra.

(i) Solving equations: Given $\boldsymbol{A}$ and $\boldsymbol{y}$, find $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$. The methods employed depend on the type of matrix (see §5.2) and the order of the matrix. See also Forsythe and Moler (1967), Golub and van Loan (1989), Stewart (1973), Dongarra *et al.* (1979) and Anderson *et al.* (1995).

(ii) Solving eigenvalue-eigenvector problems: Find $(\lambda, \boldsymbol{x})$ such that

$$(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}.$$

Once again, the methods used depend on the type of matrix, its order and how many eigenvalues and or eigenvectors are required. More generally, given $\boldsymbol{M}(\lambda)$ a matrix being a function of λ, find $\lambda, \boldsymbol{x}$ such that

$$\boldsymbol{M}(\lambda)\boldsymbol{x} = \boldsymbol{0}.$$

See also Wilkinson (1965), Golub and van Loan (1989), Garbow *et al.* (1977), Smith *et al.* (1976) and Anderson *et al.* (1995).

(iii) Computation of matrix expressions like projection operators (e.g. $\boldsymbol{A}(\boldsymbol{A}^t\boldsymbol{A})^{-1}\boldsymbol{A}^t$) and updating formulae (e.g. $(\boldsymbol{A}+\boldsymbol{u}\boldsymbol{v}^t)^{-1}$ in terms of $\boldsymbol{A}^{-1}$) in optimization and linear programming. Very often the algebraic expressions used to represent these operations do not provide an efficient and numerically stable method of computation. Factorizations employed in (i) above are used to simplify these expressions. See also Golub and van Loan (1989), or Trefethen and Bau (1997), or Demmel (1997).

(iv) Error Analysis: Finding expressions for error bounds for the various procedures used above, with respect to error in data and the rounding error of computers. See also Wilkinson (1965).

5.2 Types of Matrices.

5.2.1 Non-square matrices (rectangular matrices).

These have application mainly in statistics, econometrics and statistical modelling. The inverse problem (§4.3) has to be re-stated to give a meaningful result and is re-defined in chapter 9. If $\boldsymbol{A}$ is an $m \times n$ matrix then the system of equations, given $\boldsymbol{A}$ and $\boldsymbol{y}$, find $\boldsymbol{x}$,

$$\boldsymbol{A}\boldsymbol{x} : \boldsymbol{y}$$

is called over-determined if $m > n$, and under-determined if $m < n$.

5.2.2 Square matrices.

The following divisions of square matrices do not divide the set of square matrices into disjoint sets. The classifications are useful for the different uses as described in §5.1, and is not exhaustive.

(i) Orthogonal matrices:　　$\boldsymbol{Q}^t\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^t = \boldsymbol{I}$, if $\boldsymbol{Q}$ is real.

　　Unitary matrices:　　$\overline{\boldsymbol{Q}}^t\boldsymbol{Q} = \boldsymbol{Q}\overline{\boldsymbol{Q}}^t = \boldsymbol{I}$, if $\boldsymbol{Q}$ is complex.

(ii) Real symmetric matrices:　Defined by $\boldsymbol{A}^t = \boldsymbol{A}$, or $A_{ij} = A_{ji}$, while in the complex case they are called Hermitian matrices defined by $\overline{\boldsymbol{A}}^t = \boldsymbol{A}$, or $\overline{A}_{ij} = A_{ji}$.

  (a) Positive definite (symmetric) matrices:　$\boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x} > 0$, $\boldsymbol{x} \neq \boldsymbol{0}$ and real, if $\boldsymbol{A}$ is real. (And $\overline{\boldsymbol{x}}^t\boldsymbol{A}\boldsymbol{x} > 0$, $\boldsymbol{x} \neq \boldsymbol{0}$, if $\boldsymbol{A}$ is complex Hermitian.)

  (b) Positive semi-definite matrices:　$\boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x} \geq 0$, $\boldsymbol{x} \neq \boldsymbol{0}$.

  (c) Negative definite matrices:　$\boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x} < 0$, $\boldsymbol{x} \neq \boldsymbol{0}$.

(iii) Anti-symmetric matrices: defined by the property $\boldsymbol{A}^t = -\boldsymbol{A}$ or $A_{ij} = -A_{ji}$.

(iv) Full matrices: order n^2 non zero elements.

(v) Sparse matrices: order n non zero elements.

  (a) Banded matrices: $A_{ij} = 0, |i - j| > k$ give a banded matrix with band width $2k + 1$. Special cases are the bidiagonal and tridiagonal matrices. For non symmetric matrices the upper and lower band widths are usually specified as generally they would be different.

   (b) Structured sparse matrices (e.g. from finite element models).

   (c) Random sparse matrices.

(vi) Upper or lower triangular matrices: $U_{ij} = 0$, $i > j$; $L_{ij} = 0$, $i < j$.

   Strictly upper or lower triangular: $U_{ij} = 0$, $i \geq j$; $L_{ij} = 0$, $i \leq j$.

(vii) Upper or lower Hessenberg matrices $U_{ij} = 0$, $i > j + 1$; $L_{ij} = 0$, $i < j - 1$.

(viii) Stable matrix: for linear constant coefficient differential equations, the real part of all eigenvalues is negative, while for linear constant coefficient difference equations the real part of all eigenvalues must lie within $(-1, 1)$.

5.3 Exercises.

1. Describe (a) a positive definite upper triangular matrix, (b) a symmetric upper Hessenberg matrix.

2. (i) Show that the product of lower triangular matrices is lower triangular. (Similarly upper triangular matrices.)

  (ii) Is there a similar property for upper or lower Hessenberg matrices? What about the product of an upper triangular matrix and an upper Hessenberg matrix (two cases, pre and post multiplication)?

  (iii) Show that the inverse of an upper triangular matrix, if it exists, is also upper triangular.

  (iv) Let $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ be $n \times n$ upper triangular matrices. Show that if either $\boldsymbol{U}_1$ or $\boldsymbol{U}_2$ is invertible and $\boldsymbol{U}_1 \boldsymbol{A} = \boldsymbol{U}_2$, then $\boldsymbol{A}$ is upper triangular. Show by example (2×2 is enough) that if either $\boldsymbol{U}_1$ or $\boldsymbol{U}_2$ is not invertible then $\boldsymbol{A}$ may not be upper triangular.

3. Show that the product of two tridiagonal matrices is a pentadiagonal matrix. Is there a general rule for the product of two band matrices with band width $2k + 1$ and $2\ell + 1$ say?

4. Is the product of two symmetric matrices also symmetric? If not, what other additional properties are needed?

5. Under what conditions are the eigenvalues of the product of two matrices equal to the product of the eigenvalues of the matrices?

6. Show that the rank of an $n \times n$ upper triangular matrix is less than n if at least one diagonal u_{ii} is zero. Is the rank of $\boldsymbol{U}$ equal to the number of non-zero diagonal elements? Is the rank of $\boldsymbol{U}$ equal to the number of nonzero eigenvalues?

7. Show that every matrix can be written (uniquely) as the sum of a symmetric matrix and an anti-symmetric matrix.

8. Show that every square matrix can be written as the product of two symmetric matrices.

6. Elementary Operations and Elementary Matrices.

6.1 Elementary operations.

The reader is no doubt familiar with the following invertible elementary operations on a given matrix used to solve a set of equations, by the procedure called Gauss elimination.

 (i) Multiplying a row (column) by a non-zero scalar.

 (ii) Interchanging rows (columns).

(iii) Adding a multiple of a row (column) to another row (column).

(iv) An extension of (iii) is to replace a set of rows (columns) by linear combinations of that set of rows (columns).

In this chapter, the above processes are described in terms of pre (rows) or post (columns) multiplications of a given matrix by a matrix which has the same effect as an elementary operation. The matrices used to do this are called elementary matrices. The operations will be described in terms of row operations which are equivalent to pre-multiplication by an elementary matrix. The extension to column operations is straight forward, as a post multiplication by an elementary matrix. If $\boldsymbol{A}$ is the $m \times n$ matrix to be operated on then the process

$$\boldsymbol{B} = \boldsymbol{EA},$$

where $\boldsymbol{E}$ is an elementary matrix, can be written

$$\kappa_i(\boldsymbol{B}) = \boldsymbol{E}\kappa_i(\boldsymbol{A}), \ i = 1, 2, \ldots, n.$$

That is, for pre-multiplication, the matrix $\boldsymbol{A}$ is considered as a sequence of independent columns. Hence only operations of the type,

$$\boldsymbol{b} = \boldsymbol{Ea},$$

are considered, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are m-vectors. Most of the elementary operations can be expressed as the matrix $\boldsymbol{I} + \boldsymbol{uv}^t$ for some non-zero vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, and the matrix must be invertible. See exercise 4.9.13.

6.2 Multiplying a row by a non-zero scalar.

Find $\boldsymbol{D}_i$ such that

$$\boldsymbol{b} = \boldsymbol{D}_i \boldsymbol{a},$$

where $b_j = a_j, \ j \neq i, \ b_i = \alpha a_i$. Obviously $\boldsymbol{D}_i = \mathrm{diag}(1, 1, \ldots, 1, \alpha, 1, \ldots, 1)$, where α is in the i–th position. Note that $\boldsymbol{D}_i$ can also be written as $\boldsymbol{D}_i = \boldsymbol{I} + (\alpha - 1)\boldsymbol{e}_i\boldsymbol{e}_i^t$, and its inverse as $\boldsymbol{D}_i^{-1} = \boldsymbol{I} + \left(\frac{1}{\alpha} - 1\right)\boldsymbol{e}_i\boldsymbol{e}_i^t$. This operation is usually used to produce a 'nice' number like 1 in some appropriate position when working by hand. It is usually inefficient to do this in a computer program.

6.3 Interchanging rows (elementary permutation matrices).

If $\boldsymbol{P}_{ij} = \boldsymbol{I} - (\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^t$, and $\boldsymbol{b} = \boldsymbol{P}_{ij}\boldsymbol{a}$, then $b_k = a_k, \ k \neq i$ or j, and $b_i = a_j, \ b_j = a_i$. $\boldsymbol{P}_{ij}$ is an identity matrix with rows i and j interchanged. Note that $\boldsymbol{P}_{ij}$ is symmetric and orthogonal so that $\boldsymbol{P}_{ij}\boldsymbol{P}_{ij} = \boldsymbol{I}$. A product of elementary permutation matrices is called a permutation matrix. Note that interchanging two rows of a matrix takes a sizeable amount of time on a computer due to the memory accesses. Many computer programs do not interchange, rather they keep track of which rows were done in order.

6.4 Subtracting a multiple of one row from other rows.

Usually this is done to introduce zeros in various positions of a column of a matrix. That is, find a matrix $\boldsymbol{E}_i$ of the form

$$\boldsymbol{E}_i = \boldsymbol{I} + \boldsymbol{uv}^t,$$

such that if $\boldsymbol{b} = \boldsymbol{E}_i \boldsymbol{a}$, then $b_j = a_j, \ j = 1, 2, \ldots, i$, and $b_{i+1} = b_{i+2} = \ldots = b_m = 0$. (Zeroing elements below the i–th diagonal element of the i–th column as in Gauss elimination.)

$$\boldsymbol{E}_i \boldsymbol{a} = (\boldsymbol{I} + \boldsymbol{uv}^t)\boldsymbol{a}$$
$$\boldsymbol{b} = \boldsymbol{a} + (\boldsymbol{v}^t \boldsymbol{a})\boldsymbol{u}.$$

Now if the first i elements are to remain unchanged then $u_j = 0$ for $j = 1, 2, \ldots, i$, and if $b_k = 0$, $k > i$, then

$$u_k = -\frac{a_k}{(\boldsymbol{v}^t\boldsymbol{a})}, \ k > i \ .$$

If $\boldsymbol{v} = \boldsymbol{e}_i$ is chosen, then $\boldsymbol{E}_i$ corresponds to the Gauss elimination of elements in a column beneath the diagonal element (i-th), (check it!), and $\boldsymbol{E}_i$ is lower triangular and invertible. Hence

$$u_k = -a_k/a_i, \ k > i, \ a_i \neq 0, \qquad \text{and,}$$

and

$$\boldsymbol{E}_i = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \ddots & & & \bigcirc & \\ & & & 1 & & & \\ & & & -a_{i+1}/a_i & & & \\ & \bigcirc & & \vdots & & \ddots & \\ & & & -a_{m-1}/a_i & & 1 & \\ & & & -a_m/a_i & & & 1 \end{pmatrix}$$

that is, $\boldsymbol{E}_i$ is lower triangular with a spike in the i–th column below the diagonal of ones. This elementary matrix will be written as $\boldsymbol{I} - \boldsymbol{m}_i\boldsymbol{e}_i^t$, where the vector $\boldsymbol{u}$ is now denoted $-\boldsymbol{m}_i$ and referred to as the vector of multipliers. Other choices of $\boldsymbol{v}$ are possible but may not give a lower triangular matrix or allow such an efficient, in terms of operation count, application of $\boldsymbol{E}_i$ to an arbitrary vector.

6.5 Householder transformations (elementary Hermitians).

This transformation achieves the same effect as in §6.4, except that the elementary matrix is orthogonal, (see exercise 6.7.3(i))

$$\boldsymbol{H}_i = \boldsymbol{I} - 2\boldsymbol{w}\boldsymbol{w}^t, \ \|\boldsymbol{w}\|_2 = 1.$$

Consider the vector $\boldsymbol{b} = \boldsymbol{H}_i\boldsymbol{a}$. Note first that $\boldsymbol{a}$ and $\boldsymbol{b}$ must have the same 2-norm. Elements $i + 1, i + 2, \ldots, m$–th of $\boldsymbol{b}$ are required to be zero. It is impossible to have $b_j = a_j$ for all $j = 1, 2, \ldots, i$ (why?), but it is possible to leave $b_j = a_j$ for elements $j = 1, 2, \ldots, i - 1$ and change the i-th element. Using

$$\boldsymbol{b} = \boldsymbol{H}_i\boldsymbol{a} = \boldsymbol{a} - 2(\boldsymbol{w}^t\boldsymbol{a})\boldsymbol{w},$$

then $b_j = a_j$, gives $w_j = 0$, $j = 1, 2, \ldots, i - 1$. It remains to find w_k, $k = i, i + 1, \ldots, m$, and also b_i. To find b_i use $\|\boldsymbol{a}\|_2^2 = \|\boldsymbol{b}\|_2^2$,

$$\sum_{j=1}^{i-1} a_j^2 + \sum_{j=i}^{m} a_j^2 = \sum_{j=1}^{i-1} b_j^2 + b_i^2 + 0 \ .$$

Hence

$$b_i = \pm\sqrt{\sum_{j=i}^{m} a_j^2} = \alpha \quad \text{(say)}, \tag{6.5.1}$$

with the sign to be chosen. Let $\boldsymbol{w}^t\boldsymbol{a} = \gamma$, then $w_k = a_k/2\gamma$, $k = i + 1, \ldots, m$, and

$$w_i = \frac{(a_i - \alpha)}{2\gamma}. \tag{6.5.2}$$

Then

$$\gamma = \boldsymbol{w}^t\boldsymbol{a} = w_ia_i + \sum_{k=i+1}^{m} w_ka_k$$

$$= \frac{a_i^2 - \alpha a_i + \sum_{k=i+1}^{m} a_k^2}{2\gamma}$$

Rearranging,

$$\gamma = \sqrt{\frac{\alpha^2 - \alpha a_i}{2}} \tag{6.5.3}$$

Now choose the sign of α so that subtractive cancellation does not take place in (6.5.3). Hence α has opposite sign to a_i,

$$\alpha = -sign(a_i) \left(\sum_{j=i}^{m} a_j^2 \right)^{\frac{1}{2}}, \tag{6.5.4}$$

and the sequence of calculation is (i) (6.5.1) $|\alpha|$, (ii) (6.5.4) α, (iii) (6.5.3) γ, (iv) (6.5.2) w_j , $j = i, i+1, \ldots, m$.

6.6 Plane rotations (Given's rotations).

These elementary operations replace two rows by linear combinations of the two rows. Let $\boldsymbol{P}^{ij}$ denote the rotation of the i–th and j–th rows of a vector $\boldsymbol{a}$. $\boldsymbol{P}^{ij}$ is orthogonal and is usually used to zeroise one element of a vector (say a_j). Hence

$$\boldsymbol{P}^{ij} \boldsymbol{a} = \boldsymbol{b} \quad \text{say,}$$

is defined by $a_k = b_k$, $k \neq i$ or j, $b_j = 0$. $\boldsymbol{P}^{ij}$ is orthogonal so $b_i^2 = a_i^2 + a_j^2$ as these are the only elements of $\boldsymbol{a}$ to change. $\boldsymbol{P}^{ij}$ is an identity matrix with the i–th and j–th row and column intersections replaced with one of

$$
\begin{array}{cc} i & j \end{array} \\
\begin{array}{c} i \\ j \end{array} \begin{pmatrix} c & s \\ s & -c \end{pmatrix} \qquad \text{or} \qquad \begin{array}{cc} i & j \end{array} \\
\begin{array}{c} i \\ j \end{array} \begin{pmatrix} c & s \\ -s & c \end{pmatrix}
$$

$$\text{rotation + reflection} \qquad\qquad\qquad \text{pure rotation.}$$

It is best to use the symmetric one as the sign in the anti-symmetric one is bothersome when considering inverses. Only consider that part of $\boldsymbol{b} = \boldsymbol{P}^{ij}\boldsymbol{a}$ which is involved in the computation, that is,

$$\begin{pmatrix} c & s \\ s & -c \end{pmatrix} \begin{pmatrix} a_i \\ a_j \end{pmatrix} = \begin{pmatrix} b_i \\ 0 \end{pmatrix}$$

$$b_i = \sqrt{a_i^2 + a_j^2} \,, \Rightarrow c = a_i/b_i \ \text{and} \ s = a_j/b_i.$$

Note that $c^2 + s^2 = 1$, hence the notation, $c = \cos\theta$, $s = \sin\theta$, where θ is the angle of rotation of the i–th and j–th plane.

It can be shown that the plane rotation plus reflection matrix is also a Householder transformation (see exercise 6.7.4). Generally if many elements are to be zeroed the Householder transformation is more efficient than using many plane rotations by a factor of 2. The reader might like to compare the plane rotations with the elementary matrix of Gauss elimination type for zeroing one element of a column.

6.7 Exercises.

1. (i) If $\boldsymbol{u} \neq \boldsymbol{0}$ and $\alpha\boldsymbol{u} + \beta\boldsymbol{x} = \boldsymbol{0}$, show that either α and β both equal zero or $\boldsymbol{x} = \gamma\boldsymbol{u}$, $\gamma \in \mathbb{R}$.
   (ii) If $\boldsymbol{u} \neq \boldsymbol{0}$, $\boldsymbol{v} \neq \boldsymbol{0}$ and $\boldsymbol{u}\boldsymbol{v}^t + \boldsymbol{a}\boldsymbol{b}^t = \boldsymbol{0}$, using (i) show that there exists nonzero scalars α and β such that $\boldsymbol{a} = \alpha\boldsymbol{u}$ and $\boldsymbol{b} = \beta\boldsymbol{v}$.
   (iii) What is the inverse of $\boldsymbol{E} = \boldsymbol{I} + \boldsymbol{u}\boldsymbol{v}^t$ and when does it exist? Assume $\boldsymbol{E}^{-1} = \boldsymbol{I} + \boldsymbol{a}\boldsymbol{b}^t$ and use (ii).
   (iv) What are the eigenvalues and eigenvectors of $\boldsymbol{E} = \boldsymbol{I} + \boldsymbol{u}\boldsymbol{v}^t$? Use (i) on $(\boldsymbol{I} + \boldsymbol{u}\boldsymbol{v}^t)\boldsymbol{x} = \lambda\boldsymbol{x}$.
2. (i) Repeat 1(iii) and 1(iv) for $\boldsymbol{E} = \boldsymbol{E}_i = \boldsymbol{I} - \boldsymbol{m}_i\boldsymbol{e}_i^t$.
   (ii) Examine the i–th major step of Gauss elimination (which is to zero elements below the i-th diagonal) as you have done it in the past and compare it to using $\boldsymbol{E}_i$, that is, expand and interpret $\rho_j(\boldsymbol{E}_i\boldsymbol{A})$ in terms of rows. See 4.9.1.
   (iii) What effect does $\boldsymbol{E}_i$ have on $\boldsymbol{e}_j$, $j \leq i$ on both pre-multiplication $(\boldsymbol{E}_i\boldsymbol{e}_j)$ and post-multiplication $(\boldsymbol{e}_j^t\boldsymbol{E}_i)$?
   (iv) Find an expression for $\|\boldsymbol{E}_i\|_\infty$.
   (v) Find an expression for $\boldsymbol{E}_i\boldsymbol{E}_j$ and show the pattern of zero and non-zero elements in the product matrix. (Cases $i < j$, $i = j$, $i > j$.)
   (vi) Find an expression for $\det(\boldsymbol{E}_i)$.

(vii) What form does $\boldsymbol{E}_i = \boldsymbol{I} + \boldsymbol{u}\boldsymbol{v}^t$ take in Gauss-Jordan elimination (which zeros above and below the diagonal)?

3.   (i) Show that if $\boldsymbol{I} + \boldsymbol{u}\boldsymbol{v}^t$ is orthogonal then $\boldsymbol{u} = -\boldsymbol{v}$ and $\|\boldsymbol{u}\| = \sqrt{2}$.

   (ii) Where is $\|\boldsymbol{w}\|_2 = 1$ used in computing $\boldsymbol{H}_i$?

  (iii) Repeat the calculation of $\boldsymbol{H}_i$ using $\boldsymbol{H}_i = \boldsymbol{I} - \boldsymbol{w}\boldsymbol{w}^t$, $\|\boldsymbol{w}\|_2 = \sqrt{2}$.

  (iv) Can $\boldsymbol{H}_i$ ever be undefined? If so, how?

   (v) Write down $\boldsymbol{H}_i^{-1}$ in terms of $\boldsymbol{w}$.

  (vi) What are the eigenvalues and eigenvectors of $\boldsymbol{H}_i$.

 (vii) Show $\boldsymbol{H}_i\boldsymbol{e}_j = \boldsymbol{e}_j$ if $j < i$.

(viii) If α is not chosen by (6.5.4), show that it is possible for $\gamma = 0$, and hence for $\boldsymbol{H}_i$ to be undefined. But does it matter in this case?

  (ix) Find an expression for $\det(\boldsymbol{H}_i)$.

   (x) The complex case has $\boldsymbol{H}_i = \boldsymbol{I} - 2\boldsymbol{w}\overline{\boldsymbol{w}}^t$. Repeat the computation for a complex vector $\boldsymbol{a}$. Note that $\boldsymbol{H}_i$ is now unitary.

4.   (i) Show that $\begin{pmatrix} c & s \\ s & -c \end{pmatrix}$ (actually a rotation and reflection) can be written in the form $\boldsymbol{I} - 2\boldsymbol{w}\boldsymbol{w}^t$.

   (ii) Show that $\begin{pmatrix} c & s \\ -s & c \end{pmatrix}$ (a rotation) can be written in the form $\boldsymbol{J} - 2\boldsymbol{w}\boldsymbol{w}^t\boldsymbol{J}$ or $\boldsymbol{J} - 2\boldsymbol{J}\boldsymbol{w}\boldsymbol{w}^t$ where $\boldsymbol{J} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, a reflection.

  (iii) If $\theta = \pi/2$ what is the effect of the plane rotation, and, plane rotation and reflection.

5.   (i) Find the plane rotations which will zero the second and third elements of the vector $[1, 2, 2]^t$.

   (ii) Find the Householder matrix to cause the same effect.

6. Let $\boldsymbol{H}$ be a Householder matrix defined by $\boldsymbol{H}\boldsymbol{a} = \|\boldsymbol{a}\|\boldsymbol{e}_1$. Let $\boldsymbol{H}\boldsymbol{b} = \boldsymbol{c}$. Show that $c_1 = \langle \boldsymbol{a}, \boldsymbol{b} \rangle / \|\boldsymbol{a}\|$, that is, the projection of $\boldsymbol{b}$ onto a unit vector in the direction $\boldsymbol{a}$, and hence that $\boldsymbol{H}[c_1, 0, 0, \ldots, 0]^t$ is parallel to $\boldsymbol{a}$. Show that $\boldsymbol{H}[0, c_2, c_3, \ldots, c_n]^t$ is perpendicular to $\boldsymbol{a}$.

7. Show that each elementary matrix can be obtained by doing the elementary operation on an identity matrix.

8.   (i) Compare the number of arithmetic operations to compute $\boldsymbol{E}_i$ and $\boldsymbol{H}_i$.

   (ii) Compare the number of arithmetic operations to apply $\boldsymbol{E}_i$ and $\boldsymbol{H}_i$ to an arbitary vector.

  (iii) Compare the number of arithmetic operations to compute and apply a Householder transformation to zero elements below the i-th with the corresponding sequence of plane rotations which zero the same elements.

7. Decompositions, Factorizations of Matrices.

7.1 The singular value decomposition (SVD).

This factorization is very useful as a tool to gain some geometric insight into a linear mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$. It was not used in computation to a great extent, but with faster workstations it is now used more often.

Theorem 7.1.1 An arbitary $m \times n$ real matrix $\boldsymbol{A}$ can be factored to the matrix product

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^t$$

where $\boldsymbol{V}$ is $m \times m$ orthogonal, $\boldsymbol{U}$ is $n \times n$ orthogonal, and $\boldsymbol{D}$ is $m \times n$, diagonal with one of the forms,

$$
\begin{pmatrix}
d_1 & & & & \Big| & \\
 & d_2 & & & \Big| & \bigcirc \\
 & & \ddots & & \Big| & \\
 & & & d_m & \Big| &
\end{pmatrix}
\quad \text{or} \quad
\begin{pmatrix}
d_1 & & & & \\
 & d_2 & & & \\
 & & \ddots & & \\
 & & & d_n & \\
\overline{} & \overline{} & \overline{} & \overline{} & \\
 & & \bigcirc & &
\end{pmatrix}.
$$

Further, $\boldsymbol{D} = \mathrm{diag}(d_1, d_2, \ldots, d_\ell)$, $\ell = \min(m,n)$, and d_i are ordered so that $d_1 \geq d_2 \geq \cdots \geq d_r > 0$, $d_{r+1} = \cdots = d_\ell = 0$, where $r = \mathrm{rank}(\boldsymbol{A})$. The d_i are known as the singular values of $\boldsymbol{A}$. The SVD can also be written

$$\boldsymbol{A} = \sum_{i=1}^{r} d_i \boldsymbol{v}_i \boldsymbol{u}_i^t,$$

that is, the sum of r rank 1 matrices $\boldsymbol{v}_i \boldsymbol{u}_i^t$, where $\boldsymbol{v}_i = \kappa_i(\boldsymbol{V})$ and $\boldsymbol{u}_i = \kappa_i(\boldsymbol{U}) = \rho_i(\boldsymbol{U}^t)$. ∎

See Golub and Kahan (1965) for the original modern algorithm, which involves an implicit eigenvalue/eigenvector computation, and Eckart and Young (1936) for an original use in statistics.

7.2 The Q–U factorization (or Q–R factorization).

Theorem 7.2.1

(a) Every $m \times n$ matrix $\boldsymbol{A}$, $m \geq n$, with rank n (full rank) can be factored to the form

$$\boldsymbol{A} = \boldsymbol{Q} \begin{pmatrix} U \\ \boldsymbol{0} \end{pmatrix},$$

where $\boldsymbol{Q}$ is orthogonal and $\boldsymbol{U}$ is $n \times n$ upper triangular and non-singular.

(b) Every $m \times n$ matrix $\boldsymbol{A}$, $m \leq n$, with rank m (full rank) can be factored to the form

$$\boldsymbol{A} = (\,\boldsymbol{L} \quad \boldsymbol{0}\,)\,\boldsymbol{Q},$$

where $\boldsymbol{Q}$ is $n \times n$ orthogonal and L is $m \times m$ lower triangular with rank m.

(c) Every $m \times n$ matrix $\boldsymbol{A}$ of rank r can be factored to the form

$$\boldsymbol{A} = \boldsymbol{Q}_1 \begin{pmatrix} \boldsymbol{L} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \boldsymbol{Q}_2,$$

where $\boldsymbol{Q}_1$ is $m \times m$ orthogonal, $\boldsymbol{Q}_2$ is $n \times n$ orthogonal and $\boldsymbol{L}$ is $r \times r$ lower triangular of rank r.

Proof:

(a) Algorithm:

$$\boldsymbol{A}_0 = \boldsymbol{A}.$$

$$\text{For } i = 1, 2, \ldots, n$$

$$\text{do } \boldsymbol{A}_i = \boldsymbol{H}_i \boldsymbol{A}_{i-1};$$

where $\boldsymbol{H}_i$ is a Householder transformation which zeros column i of $\boldsymbol{A}_{i-1}$ below the diagonal element. Then

$$\begin{aligned} \boldsymbol{A}_n &= \boldsymbol{H}_n \boldsymbol{A}_{n-1} \\ &= \boldsymbol{H}_n \boldsymbol{H}_{n-1} \cdots \boldsymbol{H}_1 \boldsymbol{A} \\ &= \boldsymbol{Q}^t \boldsymbol{A}, \end{aligned}$$

where $\boldsymbol{Q} = \boldsymbol{H}_1 \boldsymbol{H}_2 \dots, \boldsymbol{H}_n$ is an orthogonal matrix. The matrix $\boldsymbol{A}_n$ has the form $\begin{pmatrix} U \\ 0 \end{pmatrix}$, hence

$$\boldsymbol{A} = \boldsymbol{Q} \begin{pmatrix} U \\ 0 \end{pmatrix}.$$

Note:

(i) Verify that the application of $\boldsymbol{H}_i$ to $\boldsymbol{A}_{i-1}$ does not destroy the zeros already produced in columns $1, 2, \dots, i-1$ of $\boldsymbol{A}_{i-1}$.

(ii) Is it possible that one of the $\boldsymbol{H}_i$ are undefined? (See §6.7 exercise 3(v).)

(b) Apply (a) to $\boldsymbol{A}^t$.

(c) Try this as an exercise being careful with the rank question as in (ii) above. Some column interchanges might be necessary. ∎

Note: Part (a) is a statement of Gram-Schmidt orthogonalization (see any introductory linear algebra text, for example, Anton and Rorres (1987)) but the Gram-Schmidt algorithm is numerically unstable, while the Householder algorithm is stable. Björck (1968) has an alternative stable algorithm.

7.3 The L–U factorization.

Theorem 7.3.1 Under certain conditions (stated in proof) an $n \times n$ matrix $\boldsymbol{A}$ of rank n can be factored to the form

$$\boldsymbol{A} = \boldsymbol{LU},$$

where $\boldsymbol{L}$ is lower triangular with diagonal elements unity and $\boldsymbol{U}$ is upper triangular of rank n. Note that this is an algebraic statement of the Gauss elimination algorithm with no row interchanges.

Proof:

(i) Algorithm:

$$\boldsymbol{A}_0 = \boldsymbol{A}$$
$$\text{For} \quad i = 1, 2, \dots, n-1$$
$$\text{do} \quad \boldsymbol{A}_i = \boldsymbol{E}_i \boldsymbol{A}_{i-1};$$

where $\boldsymbol{E}_i$ is the elementary matrix of type discussed in §6.4. $\boldsymbol{E}_i$ zeros the elements in the i–th column of $\boldsymbol{A}_{i-1}$ below the diagonal. Hence

$$\begin{aligned} \boldsymbol{A}_{n-1} &= \boldsymbol{E}_{n-1} \boldsymbol{A}_{n-2} \\ &= \boldsymbol{E}_{n-1} \boldsymbol{E}_{n-2} \dots \boldsymbol{E}_1 \boldsymbol{A}, \end{aligned}$$

and

$$\boldsymbol{A} = \boldsymbol{E}_1^{-1} \boldsymbol{E}_2^{-1} \dots \boldsymbol{E}_{n-1}^{-1} \boldsymbol{A}_{n-1}.$$

(ii) It remains to show that $\boldsymbol{A}_{n-1}$ is upper triangular of full rank, and the product $\boldsymbol{E}_1^{-1} \cdots \boldsymbol{E}_{n-1}^{-1}$ is lower triangular. At each step, $\boldsymbol{E}_i$ does not affect the zeros already introduced in columns $1, 2, \dots, i-1$ of $\boldsymbol{A}_{i-1}$ (exercise 2(iv) in §6.7). This shows that $\boldsymbol{A}_{n-1}$ is upper triangular. $\boldsymbol{A}_{n-1}$ has full rank because it can be shown that the product $\boldsymbol{E}_1^{-1} \cdots \boldsymbol{E}_{n-1}^{-1}$ is nonsingular.

(iii) When is $\boldsymbol{E}_i$ defined? Consider the i–th step of the algorithm. $\boldsymbol{A}_{i-1}$ has the form (let the (k, j)–th element of $\boldsymbol{A}_{i-1}$ be denoted $a_{k,j}$)

$$\boldsymbol{A}_{i-1} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1i} & \cdots & a_{1n} \\ & a_{22} & \cdots & a_{2i} & \cdots & a_{2n} \\ & & \ddots & \vdots & & \vdots \\ & \bigcirc & & a_{ii} & \cdots & a_{in} \\ & & & a_{i+1,i} & & \\ & & & \vdots & & \vdots \\ & & & a_{n,i} & \cdots & a_{nn} \end{pmatrix}$$

In order to zero the elements $a_{i+1,i}, \ldots, a_{ni}$, $\boldsymbol{E}_i$ has the form

$$\boldsymbol{E}_i = \boldsymbol{I} - \boldsymbol{m}_i \boldsymbol{e}_i^t,$$
$$\boldsymbol{m}_i = [0, 0, \ldots, 0, a_{i+1,i}, a_{i+2,i}, \ldots, a_{ni}]/a_{ii}.$$

Hence the condition for $\boldsymbol{E}_i$ to be defined is that the (i, i)–th element of $\boldsymbol{A}_{i-1}$ be non-zero, the same condition as in Gauss elimination with no row interchanges.

(iv) From exercise 2(i) of §6.7

$$\boldsymbol{E}_i^{-1} = \boldsymbol{I} + \boldsymbol{m}_i \boldsymbol{e}_i^t,$$

and hence the product

$$\boldsymbol{E}_1^{-1} \boldsymbol{E}_2^{-1} \cdots \boldsymbol{E}_{n-1}^{-1} = \boldsymbol{I} + \boldsymbol{m}_1 \boldsymbol{e}_1^t + \boldsymbol{m}_2 \boldsymbol{e}_2^t + \cdots + \boldsymbol{m}_{n-1} \boldsymbol{e}_{n-1}^t$$

which is lower triangular of the form

$$\begin{pmatrix} 1 & & & & & \\ \boldsymbol{m}_1 & 1 & & & & \\ & \boldsymbol{m}_2 & 1 & & \bigcirc & \\ & & \boldsymbol{m}_3 & \ddots & & \\ & & & \ddots & 1 & \\ & & & & \boldsymbol{m}_{n-1} & 1 \end{pmatrix}$$

that is, the non-zero parts of $\boldsymbol{m}_i$ fit underneath the 1 on the (i, i)-th diagonal element.   ∎

Pivoting.

Pivoting is the procedure of interchanging rows (and columns) in order to reduce the conditions of Theorem 7.3.1 but mainly to give numerical stability (see exercise 8.5.1). Consider an idealized situation in which the first major step of Guass elimination is done on a matrix partitioned as follows, so as to separate the first row and column,

$$\begin{pmatrix} \frac{1}{10} & | & \mathcal{O}(1) \\ & & \\ \mathcal{O}(1) & | & \mathcal{O}(1) \end{pmatrix}$$

where $\mathcal{O}(1)$ represents numbers of size 1 with p decimal places (and significant digits) accurate. Applying $\boldsymbol{E}_1$ to this matrix does row operations of the form

$$row_k \leftarrow row_k - 10 \; row_1.$$

This looses one significant decimal digit of row_k on a machine working to p significant decimal digits in the lower right $(n-1) \times (n-1)$ sub-matrix. See exercise 8.5.1 for a more dramatic example of this loss of significant digits. If this were to happen p times, the remaining sub-matrix would have no significant digits. In binary arithmetic, replace $1/10$ by $1/2$ to lose one significant binary digit per step. Single precision has about 24 significant binary digits, so doing this 24 times would lose all significant bits. This limits the size of matrix systems able to be solved on a machine with a fixed length of mantissa. It is the small diagonal elements (small with respect to the elements directly below it) causing the problem. So at each major step of Gauss elimination interchange rows so that the diagonal element is largest in magnitude of those below it. This is called row pivoting or partial pivoting and reduces the loss of significant digits. Full pivoting uses both row and column interchanges so that at each major step of Gauss elimination the largest element in the lower $(n-i) \times (n-i)$ sub-matrix is at position (i, i) of $\boldsymbol{A}_{i-1}$.

Theorem 7.3.2 (Partial pivoting) Every $n \times n$ matrix $\boldsymbol{A}$ can be factored to a form

$$\boldsymbol{A} = \boldsymbol{PLU},$$

where $\boldsymbol{P}$ is a permutation matrix (interchanges rows) and $\boldsymbol{L}$ is a lower triangular matrix and $\boldsymbol{U}$ is upper triangular. The matrix $\boldsymbol{L}$ is nonsingular ($L_{ii} = 1$) and $\boldsymbol{U}$ has the same rank as $\boldsymbol{A}$.

Proof:

(i) Algorithm:

$$\boldsymbol{A}_0 = \boldsymbol{A}$$

$$\text{For } \; i = 1, 2, \ldots, n-1$$

$$\text{do } \; \boldsymbol{A}_i = \boldsymbol{E}_i \boldsymbol{P}_i \boldsymbol{A}_{i-1}.$$

(a) $\boldsymbol{P}_i$ is an elementary permutation matrix which interchanges rows of $\boldsymbol{A}_{i-1}$ so that the (i, i)–th element of $\boldsymbol{P}_i \boldsymbol{A}_{i-1}$ is the largest (in modulus) of the elements of the i–th column below and on the diagonal, i.e.

$$|(\boldsymbol{P}_i \boldsymbol{A}_{i-1})_{ii}| \geq |(\boldsymbol{P}_i \boldsymbol{A}_{i-1})_{ji}|, \; j = i, i+1, \ldots, n.$$

(b) $\boldsymbol{E}_i$ is defined to be $\boldsymbol{I}$ if $(\boldsymbol{P}_i \boldsymbol{A}_{i-1})_{i,i}$ is zero, otherwise $\boldsymbol{E}_i$ is the usual elementary matrix for zeroing the i–th column below the diagonal

$$\boldsymbol{E}_i = \boldsymbol{I} - \boldsymbol{m}_i \boldsymbol{e}_i^t,$$

where now $\boldsymbol{m}_i$ is based on the i–th column of $\boldsymbol{P}_i \boldsymbol{A}_{i-1}$. Note that the elements of $\boldsymbol{m}_i$ are all less than or equal to 1 in magnitude. This pivoting procedure takes account of any rank deficiency in $\boldsymbol{A}$ (why?).

(ii) The algorithm produces

$$\boldsymbol{A}_{n-1} = \boldsymbol{E}_{n-1} \boldsymbol{P}_{n-1} \boldsymbol{E}_{n-2} \boldsymbol{P}_{n-2} \cdots \boldsymbol{E}_2 \boldsymbol{P}_2 \boldsymbol{E}_1 \boldsymbol{P}_1 \boldsymbol{A},$$

and

$$\boldsymbol{A}_{n-1}(= \boldsymbol{U}) = \boldsymbol{L}^{-1} \boldsymbol{P}^{-1} \boldsymbol{A},$$

is required in order to prove the theorem. This can be done provided all $\boldsymbol{P}_i$ can be shifted to the inside and leave the $\boldsymbol{E}_i$ essentially untouched in the positions of non-zero elements.

(a) Consider $\boldsymbol{P}_i \boldsymbol{E}_j \boldsymbol{P}_i$, $i > j$. $\boldsymbol{P}_i$ interchanges rows and columns of $\boldsymbol{E}_j$ using rows and columns bigger than the i–th.

$$\boldsymbol{P}_i(\boldsymbol{I} - \boldsymbol{m}_j \boldsymbol{e}_j^t)\boldsymbol{P}_i = \boldsymbol{I} - \boldsymbol{P}_i \boldsymbol{m}_j \boldsymbol{e}_j^t \text{ , as } \; \boldsymbol{e}_j^t \boldsymbol{P}_i = \boldsymbol{e}_j^t$$

$$= \boldsymbol{I} - \overline{\boldsymbol{m}}_j \boldsymbol{e}_j^t, \text{ say,}$$

where $\overline{\boldsymbol{m}}_j$ differs from $\boldsymbol{m}_j$ in that two non-zero elements have been interchanged. Hence

$$\boldsymbol{P}_i \boldsymbol{E}_j \boldsymbol{P}_i = \overline{\boldsymbol{E}}_j,$$

where $\overline{\boldsymbol{E}}_j$ has the same form as $\boldsymbol{E}_j$ in terms of zeros and non-zeros.

(b) Now write (using (a) above)

$$\boldsymbol{P}_2 \boldsymbol{E}_1 \boldsymbol{P}_1 = \boldsymbol{P}_2 \boldsymbol{E}_1 \boldsymbol{P}_2 \boldsymbol{P}_2 \boldsymbol{P}_1$$

$$= \overline{\boldsymbol{E}}_1 \boldsymbol{P}_2 \boldsymbol{P}_1,$$

and further

$$\boldsymbol{U} = \boldsymbol{E}_{n-1} \overline{\boldsymbol{E}}_{n-2} \overline{\overline{\boldsymbol{E}}}_{n-3} \cdots \overline{\overset{\vdots}{\boldsymbol{E}}}_1 \boldsymbol{P}_{n-1} \boldsymbol{P}_{n-2} \cdots \boldsymbol{P}_1 \boldsymbol{A}.$$

Hence $\boldsymbol{A} = \boldsymbol{PLU}$, where

$$\boldsymbol{P} = \boldsymbol{P}_1 \boldsymbol{P}_2 \cdots \boldsymbol{P}_{n-1},$$

$$\boldsymbol{L} = \overline{\overset{\vdots}{\boldsymbol{E}}}_1^{-1} \cdots \overline{\boldsymbol{E}}_{n-2}^{-1} \boldsymbol{E}_{n-1}^{-1},$$

and is still lower triangular.   ∎

The above is known as partial pivoting or row pivoting. The result (ii)(b) above,

$$\boldsymbol{P}^{-1} \boldsymbol{A} = \boldsymbol{LU},$$

is saying that if the row interchanges which need to be done were done first and then the algorithm of Theorem 7.3.1 applied to the new matrix $\boldsymbol{P}^{-1} \boldsymbol{A}$, there would be no difficulty with zero divisors.

The next theorem gives the result for full pivoting where at each stage, rows and columns are interchanged in order that $(\boldsymbol{A}_{i-1})_{ii}$ is the largest element of the lower right partition of $\boldsymbol{A}_{i-1}$.

Theorem 7.3.3 (Full pivoting) Every $m \times n$ matrix $\boldsymbol{A}$, of rank r, can be factored to a form,

$$\boldsymbol{A} = \boldsymbol{P}\boldsymbol{L}\boldsymbol{U}\boldsymbol{Q}$$

where $\boldsymbol{P}$ and $\boldsymbol{Q}$ are permutation matrices and $\boldsymbol{L}$ is $m \times r$ lower triangular and $\boldsymbol{U}$ is $r \times n$ upper triangular.

$$\boldsymbol{A} = \boldsymbol{P} \begin{pmatrix} 1 & & & \bigcirc \\ \vdots & \ddots & & \\ L_{r1} & & 1 & \\ \vdots & & & \vdots \\ L_{m1} & \cdots & & L_{mr} \end{pmatrix} \begin{pmatrix} U_{11} & \cdots & U_{1r} & \cdots & U_{1n} \\ & \ddots & & & \vdots \\ \bigcirc & & U_{rr} & \cdots & U_{rn} \end{pmatrix} \boldsymbol{Q}$$

Proof: — Exercise! ∎

From $\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{Q}^{-1} = \boldsymbol{L}\boldsymbol{U}$, if the appropriate permutations of rows and columns were done before Gauss elimination, then the biggest element of the remaining lower sub-matrix would be in the top left position, the pivot position.

7.4 Choleski factorization and variants.

Theorem 7.4.1 A positive definite matrix (symmetric and square) has a factorization of the form

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^t,$$

where $\boldsymbol{L}$ is lower triangular and invertible.

Proof: See for example the book by Stewart and exercise 7.7.7.

Algorithm: By solving the $n(n+1)/2$ defining equations in a particular order all elements of $\boldsymbol{L}$ are easily found.

	Defining Equations	Component of L
(i)	$A_{11} = L_{11}^2$	$L_{11} = \sqrt{A_{11}}$
(ii)	For $j = 2, 3, \ldots, n$ do	
	$A_{j1} = L_{j1}L_{11}$	$L_{j1} = A_{j1}/L_{11}$
(iii)	For $i = 2, \ldots, n$ do	
	$A_{ii} = \sum_{k=1}^{i} L_{ik}^2$	$L_{ii} = \sqrt{(A_{ii} - \sum_{k=1}^{i-1} L_{ik}^2)}$
	For $j = i+1, i+2, \ldots, n$ do	
	$A_{ji} = \sum_{k=1}^{i} L_{jk}L_{ik}$	$L_{ji} = (A_{ji} - \sum_{k=1}^{i-1} L_{jk}L_{ik})/L_{ii}$ ∎

Corollary 7.4.1 A positive definite matrix has a factorization

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^t$$

where $\boldsymbol{L}$ is lower triangular with $L_{ii} = 1$, $i = 1, 2, \ldots, n$ and $\boldsymbol{D}$ is a positive diagonal matrix.

Proof: Exercise. ∎

There is no need to consider pivoting in this algorithm because the matrix is non-singular, and the error analysis shows that pivoting is not needed to reduce error magnification. The upper triangular matrix $\boldsymbol{D}\boldsymbol{L}^t$ is equal to $\boldsymbol{U}$ of the L-U factorization, and the $\boldsymbol{L}$ matrices of both factorizations are equal.

Corollary 7.4.2 Some symmetric $n \times n$ matrices $\boldsymbol{A}$ with rank n, can be factored to

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^t,$$

where now $\boldsymbol{D}$ is diagonal but not necessarily positive definite, and $L_{ii} = 1$.

Proof: Exercise. ∎

If $\boldsymbol{D}$ has both positive and negative diagonal elements the factorization is unstable with respect to roundoff error. A negative definite matrix will have all $D_{ii} < 0$. There is no need to have alternative software for negative definite matrices $\boldsymbol{A}$, as $-\boldsymbol{A}$ is positive definite.

Corollary 7.4.3 If $\boldsymbol{A}$ is symmetric of rank n then $\boldsymbol{A}$ can be factored to

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{B}\boldsymbol{L}^t,$$

where $\boldsymbol{L}$ is as in Corollary 7.4.1 and $\boldsymbol{B}$ is block diagonal, with either a 1×1 block or a 2×2 block with negative determinant.

Proof: See Dongarra *et al.* (1979). Exercise.   ■

This factorization is stable, and is used in LAPACK.

7.5 Banded matrices.

7.5.1 The L–U factorization.

Let $\boldsymbol{B}$ be a banded matrix with ℓ bands below the diagonal and p bands above the diagonal $(\ell+p+1$ bands altogether). Provided no pivoting is required then $\boldsymbol{B}$ has an L–U factorization

$$\boldsymbol{B} = \boldsymbol{LU},$$

where $\boldsymbol{L}$ is lower triangular with a diagonal of ones and ℓ bands below it, and, $\boldsymbol{U}$ is upper triangular with p bands above the diagonal. That is, the structure of $\boldsymbol{B}$ is preserved in $\boldsymbol{L}$ and $\boldsymbol{U}$. Proof of this result follows as an application of Theorem 7.3.1.

If pivoting is used the structure of $\boldsymbol{L}$ and $\boldsymbol{U}$ is destroyed. However if only row pivoting (partial pivoting) is allowed then

$$\boldsymbol{B} = \boldsymbol{PLU},$$

where $\boldsymbol{P}$ is a permutation matrix, $\boldsymbol{L}$ as above and $\boldsymbol{U}$ is upper triangular with $p + \ell$ bands above the diagonal. The proof follows by considering Theorem 7.3.2 and the worst possible pivoting strategy in terms of "fill-in".

7.5.2 Choleski.

If $\boldsymbol{B}$ is positive definite banded (symmetric) with $2n + 1$ bands altogether, then $\boldsymbol{B}$ has a Choleski factorization,

$$\boldsymbol{B} = \boldsymbol{LL}^t,$$

where $\boldsymbol{L}$ is lower triangular with $n + 1$ bands including the diagonal.

It is possible to re-order the rows and columns of some sparse matrices to make the resulting set of equations have as small a band structure as possible. This helps reduce the number of computations involved in solving the set of equations. This is important in applications where the same matrix or same structured matrix is inverted many times, for example, iterative eigenvalue/eigenvector computations. See Duff (1977) and George and Liu (1981).

7.6 Eigenvalue — eigenvector decomposition.

Let $\boldsymbol{A}$ be an $n \times n$ matrix. The eigenvalues (characteristic values) of $\boldsymbol{A}$ are defined as the roots of the polynomial in λ

$$\det|\boldsymbol{A} - \lambda\boldsymbol{I}| = 0.$$

There exists n linear factors $(\lambda - \lambda_i)$, $i = 1, 2, \ldots, n$ of this polynomial. As $\boldsymbol{A} - \lambda_i\boldsymbol{I}$ is singular it must have a null space. Let $\{\boldsymbol{s}_i\}$ be a set of vectors spanning these spaces. Let $\boldsymbol{S}$ be the matrix of columns $\boldsymbol{s}_i$.

$$(\boldsymbol{A} - \lambda_i\boldsymbol{I})\boldsymbol{s}_i = \boldsymbol{0},$$

$$\boldsymbol{A}\,(\,\boldsymbol{s}_1 \quad \boldsymbol{s}_2 \quad \ldots \quad \boldsymbol{s}_n\,) = (\,\lambda_1\boldsymbol{s}_1 \quad \lambda_2\boldsymbol{s}_2 \quad \ldots \quad \lambda_n\boldsymbol{s}_n\,)$$

$$\boldsymbol{AS} = \boldsymbol{S\Lambda}, \qquad \boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$$

If $\boldsymbol{S}^{-1}$ exists, this is now written as

$$\boldsymbol{A} = \boldsymbol{S\Lambda S}^{-1}.$$

It is only under certain conditions that $\boldsymbol{S}^{-1}$ will exist.

The following examples show how roundoff error might affect decisions about existence of eigenvectors.

(a) $\boldsymbol{A} = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}$, $\lambda_1 = \lambda_2 = 1$, $\boldsymbol{s}_1 = \boldsymbol{s}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ if $a \neq 0$.

(b) $\boldsymbol{A} = \begin{pmatrix} 1 & 1 \\ 0 & 1+\epsilon \end{pmatrix}$, $\lambda_1 = 1$, $\lambda_2 = 1 + \epsilon$, $\boldsymbol{s}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\boldsymbol{s}_2 = \begin{pmatrix} 1 \\ \epsilon \end{pmatrix}$,

$\boldsymbol{S}^{-1} = \begin{pmatrix} 1 & -\epsilon^{-1} \\ 0 & \epsilon^{-1} \end{pmatrix}$.

Of course a further computational problem is that if $\boldsymbol{A}$ is real, λ_i may be complex.

7.7 Exercises.

1. (i) Write down all singular value decompositions of a 1×1 matrix. Hence, how many different SVD are there for an $n \times n$ matrix where the d_i are distinct.

    (ii) What is the SVD of an $m \times 1$ matrix?

    (iii) The SVD is not uniquely defined. Why and in what way?

    Use the notation of Theorem 7.1.1 in the next eleven questions.

    (iv) Let $\boldsymbol{A}$ represent the mapping $\mathcal{A}$ in the usual basis. What is the matrix representation of $\mathcal{A}$ if the basis of $\mathbb{R}^n$ is $\{\boldsymbol{u}_i\}_{i=1}^n$ and of $\mathbb{R}^m$ is $\{\boldsymbol{v}_i\}_{i=1}^m$.

    (v) Show that the number of non-zero singular values is the rank of $\boldsymbol{A}$.

    (vi) Show that $\mathcal{R}(\boldsymbol{A}) = \mathrm{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_r\}$.

    (vii) Show that $\mathcal{R}(\boldsymbol{A}^t) = \mathrm{span}\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r\}$.

    (viii) Show that $\mathcal{N}(\boldsymbol{A}) = \mathrm{span}\{\boldsymbol{u}_{r+1}, \ldots, \boldsymbol{u}_n\}$.

    (ix) Show that $\mathcal{N}(\boldsymbol{A}^t) = \mathrm{span}\{\boldsymbol{v}_{r+1}, \ldots, \boldsymbol{v}_m\}$.

    (x) Show that $\boldsymbol{A}\boldsymbol{u}_i = d_i \boldsymbol{v}_i$.

    (xi) Show that $\boldsymbol{A}^t \boldsymbol{v}_i = d_i \boldsymbol{u}_i$.

    (xii) Show that $\|\boldsymbol{A}\|_2 = \{\lambda_{max}(\boldsymbol{A}^t \boldsymbol{A})\}^{\frac{1}{2}} = d_1$. What is $\|\boldsymbol{A}^{-1}\|_2$, if $\boldsymbol{A}^{-1}$ exists?

    (xiii) Show that $\|\boldsymbol{A}\|_E = \{\sum_{i=1}^r d_i^2\}^{\frac{1}{2}}$.

    (xiv) Show that $\|\boldsymbol{A}\|_2 \le \|\boldsymbol{A}\|_E \le \sqrt{n}\|\boldsymbol{A}\|_2$.

2. Show that the result of Theorem 7.2.1(a) can be interpreted as Gram-Schmidt orthogonalization by writing it in matrix form.

3. (i) (Crout factorization) Suppose $\boldsymbol{A}(n \times n)$ can be factored to the product $\boldsymbol{LU}$ with no problems of zero divisors. Show that the elements of $\boldsymbol{L}$ and $\boldsymbol{U}$ can be found by solving the n^2 equations

$$A_{ij} = \sum_k L_{ik} U_{kj},$$

    in an appropriate order. (Let $L_{ii} = 1$, $i = 1, \ldots, n$.) Count the number of multiplications and divisions.

    (ii) Carry out the steps of Gauss elimination, the L–U factorization via elementary matrices and the Crout method on the matrix.

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 3 & 1 & 3 & 4 \\ 2 & -3 & 4 & 8 \end{pmatrix}.$$

    (Do not use pivoting in Gauss elimination, and do not divide the pivot row by the diagonal element before doing the row operation.) Can you see that these three methods compute the same numbers, but in different orders?

4. Using the notation of §7.6, prove the following:

    (i) All λ_i distinct $\Rightarrow \boldsymbol{S}^{-1}$ exists.

    (ii) $\boldsymbol{A}$ symmetric $\Rightarrow \boldsymbol{S}^{-1}$ exists.

    (iii) If $\boldsymbol{A}$ is real and λ_i is not real then $\overline{\lambda}_i$ is also an eigenvalue.

    (iv) If $\boldsymbol{A}$ is real symmetric (Hermitian) then $\boldsymbol{S}$ may be chosen and scaled so that it is orthogonal (unitary).

    (v) $\boldsymbol{A}$ is real symmetric (Hermitian) $\Rightarrow \lambda_i$ are real. $\boldsymbol{A}$ is positive definite $\Rightarrow \lambda_i$ are positive.

    (vi) What is the relation between the SVD and the eigen decomposition of, (a) a symmetric positive definite matrix, (b) a symmetric matrix.

    (vii) Show that $\boldsymbol{A}\boldsymbol{A}^t$ and $\boldsymbol{A}^t\boldsymbol{A}$ have the same non-zero eigenvalues.

(viii) Examine the relation between the SVD of A and the eigen decomposition of AA^t and A^tA.

(ix) If A is real symmetric and all eigenvalues are positive, show that A is positive definite.

(x) If A is anti-symmetric, show that iA is Hermitian, and hence show that the eigenvalues of A are purely imaginary.

(xi) Show that if A is anti-symmetric and has odd order, it is singular.

5. Define recursive relations for the elements of L and U for tridiagonal band matrices, (without pivoting) both positive definite and otherwise.

6. Define recursive relations for the elements of L and U for pentadiagonal band matrices, (without pivoting) both positive definite and otherwise.

7. Show that the Choleski factorization $A = LL^t$ exists if and only if A is positive definite. One part of this proof requires an induction arguement on the order of the matrix, and in the process gives an alternative definition of the Choleski algorithm. See question 7.7.12.

8. Describe Gauss–Jordan elimination in terms of elementary matrices (zeros above and below the diagonal), and count the number of multiplications and divisions required.

9. Expand $e_k^t E_1 A$, for $k = 1, \ldots, n$, where A is an $n \times n$ matrix and E_1 is the first elementary matrix of the L-U factorization. Describe the resulting rows of $E_1 A$ in terms of rows of A.

10. Examine the Simplex tableau algorithm for linear programming and explain it in terms of elementary matrices operating on the tableau. Note that as each major step of Gaussian elimination is done there is no account taken of pivoting requirements. Hence the basis columns of the Simplex algorithm have to be re-inverted, with pivoting, every so many basis change steps, so as to keep some accuracy.

11. Consider the following partitioned matrix equation, which can be used to describe the L-U factorization without pivoting. Show how x, y and z can be computed from A, a, b, c and the already computed factors of A, namely L and U.

$$\begin{bmatrix} A & b \\ a^t & c \end{bmatrix} = \begin{bmatrix} L & 0 \\ x^t & 1 \end{bmatrix} \begin{bmatrix} U & y \\ 0^t & z \end{bmatrix}$$

12. Consider the following partitioned matrix equation, which can be used to describe the Choleski factorization of a symmetric positive definite matrix. Show how x and z can be computed from A, a and c and the already computed factors of A, namely L.

$$\begin{bmatrix} A & a \\ a^t & c \end{bmatrix} = \begin{bmatrix} L & 0 \\ x^t & z \end{bmatrix} \begin{bmatrix} L^t & x \\ 0^t & z \end{bmatrix}$$

13. Use questions 11 and 12 to count the number of operations in L-U and Choleski factorizations, given that it takes $n(n+1)/2$ operations to compute the solution of a set of equations involving an $n \times n$ lower or upper triangular matrix.

8. The Inverse Problem.

The procedures used to solve the set of linear equations,

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}, \tag{8.0.1}$$

where $\boldsymbol{A}$ is an $n \times n$ nonsingular matrix, $\boldsymbol{y} \in \mathbb{R}^n$, a vector termed the right hand side and $\boldsymbol{x} \in \mathbb{R}^n$, the solution vector are the subjects of this chapter. Both the algorithms for finding $\boldsymbol{x}$, and the error analysis of each method are of interest. This chapter builds on the work of chapters 6 and 7. However before considering the methods, some theory is necessary to understand how close to the exact solution a computed solution can be expected to be. This analysis is called a sensitivity analysis or condition number analysis. Part of this analysis gives a clue as to how to make a numerical decision about when a matrix is close to being singular.

8.1 Sensitivity analysis.

Primarily because of error in the data ($\boldsymbol{A}$ and $\boldsymbol{y}$), and to a lesser extent roundoff error in the computer and error generated by the arithmetic unit of the computer, the solution to the perturbed equation

$$(\boldsymbol{A} + \delta\boldsymbol{A})(\boldsymbol{x} + \delta\boldsymbol{x}) = \boldsymbol{y} + \delta\boldsymbol{y}, \tag{8.1.1}$$

needs to be considered. The perturbation in $\boldsymbol{x}$, namely $\delta\boldsymbol{x}$ is expressed in terms of the perturbations in the data ($\delta\boldsymbol{A}$ and $\delta\boldsymbol{y}$). Under what conditions on $\delta\boldsymbol{A}$ does $(\boldsymbol{A} + \delta\boldsymbol{A})^{-1}$ exist?

Lemma 8.1.1

$(\boldsymbol{I} + \boldsymbol{X})^{-1}$ exists for all $\boldsymbol{X}$ such that $\|\boldsymbol{X}\| < 1$ where $\|\cdot\|$ is a subordinate norm. ($\|\boldsymbol{I}\| = 1$) Think about this statement for a 1×1 matrix.

Proof: Note: $(\boldsymbol{I} + \boldsymbol{X})\boldsymbol{x} = \boldsymbol{0}$ has only one solution, $\boldsymbol{x} = \boldsymbol{0}$, if and only if $(\boldsymbol{I} + \boldsymbol{X})^{-1}$ exists. Suppose $\boldsymbol{x} \neq \boldsymbol{0}$ satisfies $(\boldsymbol{I} + \boldsymbol{X})\boldsymbol{x} = \boldsymbol{0}$. Then $\boldsymbol{x} = -\boldsymbol{X}\boldsymbol{x}$ and

$$\|\boldsymbol{x}\| \leq \|\boldsymbol{X}\| \, \|\boldsymbol{x}\|.$$

As $\|\boldsymbol{X}\| < 1$, and $\|\boldsymbol{x}\| \neq 0$

$$\|\boldsymbol{x}\| < \|\boldsymbol{x}\|.$$

This is a contradiction, so $\boldsymbol{x} = \boldsymbol{0}$ is the only solution. Hence $(\boldsymbol{I} + \boldsymbol{X})^{-1}$ exists if $\|\boldsymbol{X}\| < 1$. ∎

Lemma 8.1.2

$(\boldsymbol{A} + \delta\boldsymbol{A})^{-1}$ exists for all $\delta\boldsymbol{A}$ such that $\|\delta\boldsymbol{A}\| \, \|\boldsymbol{A}^{-1}\| < 1$.

Proof:

$$(\boldsymbol{A} + \delta\boldsymbol{A})^{-1} = \boldsymbol{A}^{-1}(\boldsymbol{I} + \delta\boldsymbol{A}\boldsymbol{A}^{-1})^{-1}$$

By lemma 8.1.1, $(\boldsymbol{I} + \delta\boldsymbol{A}\boldsymbol{A}^{-1})^{-1}$ exists if

$$\|\delta\boldsymbol{A}\boldsymbol{A}^{-1}\| < 1,$$

which is true if $\|\delta\boldsymbol{A}\| \, \|\boldsymbol{A}^{-1}\| < 1$. Hence $(\boldsymbol{A} + \delta\boldsymbol{A})^{-1}$ exists if $\|\delta\boldsymbol{A}\| \, \|\boldsymbol{A}^{-1}\| < 1$. ∎

Lemma 8.1.3

$$\frac{1}{1 + \|\boldsymbol{X}\|} \leq \|(\boldsymbol{I} + \boldsymbol{X})^{-1}\| \leq \frac{1}{1 - \|\boldsymbol{X}\|} \ , \ \text{if} \ \ \|\boldsymbol{X}\| < 1,$$

where $\|\cdot\|$ is a subordinate norm.

Proof:

$$\boldsymbol{I} = (\boldsymbol{I} + \boldsymbol{X})^{-1}(\boldsymbol{I} + \boldsymbol{X}) \tag{8.1.2}$$
$$\|\boldsymbol{I}\| \leq \|(\boldsymbol{I} + \boldsymbol{X})^{-1}\| \, \|\boldsymbol{I} + \boldsymbol{X}\|$$
$$\leq \|(\boldsymbol{I} + \boldsymbol{X})^{-1}\|(\|\boldsymbol{I}\| + \|\boldsymbol{X}\|)$$

Rearranging,

$$\frac{1}{1 + \|\boldsymbol{X}\|} \leq \|(\boldsymbol{I} + \boldsymbol{X})^{-1}\|.$$

From (8.1.2)

$$(\boldsymbol{I} + \boldsymbol{X})^{-1} = \boldsymbol{I} - (\boldsymbol{I} + \boldsymbol{X})^{-1}\boldsymbol{X}$$

$$\therefore \quad \|(\boldsymbol{I} + \boldsymbol{X})^{-1}\| \leq 1 + \|(\boldsymbol{I} + \boldsymbol{X})^{-1}\|\,\|\boldsymbol{X}\|,$$

and rearranging,

$$\|(\boldsymbol{I} + \boldsymbol{X})^{-1}\| \leq \frac{1}{1 - \|\boldsymbol{X}\|}. \qquad \blacksquare$$

Lemma 8.1.4

$$\|(\boldsymbol{A} + \delta\boldsymbol{A})^{-1}\| \leq \frac{\|\boldsymbol{A}^{-1}\|}{1 - \|\delta\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|} \quad \text{if} \quad \|\delta\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\| < 1.$$

Proof:

$$\begin{aligned}
\|(\boldsymbol{A} + \delta\boldsymbol{A})^{-1}\| &= \|\boldsymbol{A}^{-1}(\boldsymbol{I} + \delta\boldsymbol{A}\boldsymbol{A}^{-1})^{-1}\| \\
&\leq \frac{\|\boldsymbol{A}^{-1}\|}{1 - \|\delta\boldsymbol{A}\boldsymbol{A}^{-1}\|}, \qquad \text{by Lemma 8.1.3} \\
&\leq \frac{\|\boldsymbol{A}^{-1}\|}{1 - \|\delta\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|}, \quad \text{as} \quad \|\delta\boldsymbol{A}\boldsymbol{A}^{-1}\| \leq \|\delta\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\| < 1 \qquad \blacksquare
\end{aligned}$$

Theorem 8.1.1

The relative error in the solution of (8.1.1) expressed as $\|\delta\boldsymbol{x}\|/\|\boldsymbol{x}\|$ can be related to the relative errors in $\boldsymbol{A}$ and $\boldsymbol{y}$, namely $\|\delta\boldsymbol{A}\|/\|\boldsymbol{A}\|$ and $\|\delta\boldsymbol{y}\|/\|\boldsymbol{y}\|$ by the expression

$$\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq \frac{\chi(\boldsymbol{A})}{1 - \chi(\boldsymbol{A})\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}} \left\{ \frac{\|\delta\boldsymbol{y}\|}{\|\boldsymbol{y}\|} + \frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|} \right\} \qquad (8.1.3)$$

if $\|\delta\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\| < 1$, where $\chi(\boldsymbol{A})$ is the condition number of the matrix $\boldsymbol{A}$, defined by

$$\chi(\boldsymbol{A}) = \|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|.$$

Note the norm for matrices is a subordinate norm to the vector norm.
Proof: Subtract (8.0.1) from (8.1.1) to give after rearranging

$$\begin{aligned}
(\boldsymbol{A} + \delta\boldsymbol{A})\delta\boldsymbol{x} &= \delta\boldsymbol{y} - \delta\boldsymbol{A}\boldsymbol{x}. \\
\delta\boldsymbol{x} &= (\boldsymbol{A} + \delta\boldsymbol{A})^{-1}\{\delta\boldsymbol{y} - \delta\boldsymbol{A}\boldsymbol{x}\} \\
\|\delta\boldsymbol{x}\| &\leq \|(\boldsymbol{A} + \delta\boldsymbol{A})^{-1}\|\{\|\delta\boldsymbol{y}\| + \|\delta\boldsymbol{A}\|\,\|\boldsymbol{x}\|\}
\end{aligned}$$

From Lemma 8.1.4

$$\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq \frac{\|\boldsymbol{A}^{-1}\|}{1 - \|\delta\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|} \left\{ \frac{\|\delta\boldsymbol{y}\|}{\|\boldsymbol{x}\|} + \|\delta\boldsymbol{A}\| \right\}.$$

Inserting $\|\boldsymbol{A}\|$ in appropriate places,

$$\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq \frac{\|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{A}\|}{1 - \|\delta\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|\frac{\|\boldsymbol{A}\|}{\|\boldsymbol{A}\|}} \left\{ \frac{\|\delta\boldsymbol{y}\|}{\|\boldsymbol{A}\|\,\|\boldsymbol{x}\|} + \frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|} \right\}.$$

As $\|\boldsymbol{y}\| \leq \|\boldsymbol{A}\|\,\|\boldsymbol{x}\|$,

$$\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq \frac{\chi(\boldsymbol{A})}{1 - \chi(\boldsymbol{A})\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}} \left\{ \frac{\|\delta\boldsymbol{y}\|}{\|\boldsymbol{y}\|} + \frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|} \right\}. \qquad \blacksquare$$

In practice the quantity $1 - \chi(\boldsymbol{A})\|\delta\boldsymbol{A}\|/\|\boldsymbol{A}\|$ will be of magnitude 1. The only time when this can become small is when the bound on $\|\delta\boldsymbol{A}\|$ is close to being violated. Assume

$$\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|} << \frac{1}{\chi(\boldsymbol{A})},$$

that is, the relative error in $\boldsymbol{A}$ is much smaller than the inverse of the condition number, then the quantity $\chi(\boldsymbol{A})$ gives the magnification factor between the error in the data and the error in the solution. As

$$\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I} \Rightarrow \|\boldsymbol{A}\| \, \|\boldsymbol{A}^{-1}\| = \chi(\boldsymbol{A}) \geq \|\boldsymbol{I}\| = 1,$$

the relative error bound on the solution is always greater than the relative error bounds on the data.

A common mis-understanding is that various other quantities associated with a matrix give some idea of error magnification. The two most common mis-understandings involve the determinant of $\boldsymbol{A}$ (see question 4.9.17) and the ratio of largest to smallest eigenvalue of $\boldsymbol{A}$. These do not determine condition numbers except for special types of matrices. For example the bidiagonal matrix, $A_{ii} = 1$, $A_{i,i+1} = -1$, $i = 1, 2, \ldots, n$, and all other $A_{ij} = 0$, has determinant 1, eigenvalue ratio 1 but condition number $\chi(\boldsymbol{A}) = O(n^2)$. The above lemmas and theorem are not correct for the Euclidean norm of a matrix as $\|\boldsymbol{I}\|_E = \sqrt{n}$. Subroutines may however calculate $\chi(\boldsymbol{A})$ in terms of the Euclidean norm as the 2–norm is too costly to evaluate.

8.1.1 Scaling.

It is possible to effectively reduce the condition number of a matrix by diagonal scaling. Suppose the set of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$, is replaced it by

$$\underbrace{\boldsymbol{D}_1\boldsymbol{A}\boldsymbol{D}_2}\,\underbrace{\boldsymbol{D}_2^{-1}\boldsymbol{x}} = \underbrace{\boldsymbol{D}_1\boldsymbol{y}}$$

$$\overline{\boldsymbol{A}}\,\overline{\boldsymbol{x}} = \overline{\boldsymbol{y}},$$

where $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are diagonal matrices with elements which are powers of 2. (A computer is capable of multiplying by powers of 2 with no further rounding error). $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are chosen to make $\chi(\overline{\boldsymbol{A}}) < \chi(\boldsymbol{A})$. It is not obvious what the optimal choice of $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are, however a good strategy appears to be to make all elements of $\overline{\boldsymbol{A}}$ as close as possible to the same order. For example $\begin{pmatrix} 2.345 & 1234 \\ 1234 & 5.678 \times 10^6 \end{pmatrix}$ should be replaced with $\begin{pmatrix} 2.345 & 1.234 \\ 1.234 & 5.678 \end{pmatrix}$ where $\boldsymbol{D}_1 = \boldsymbol{D}_2 = \mathrm{diag}\,(1, 10^{-3})$. Perhaps a better way of expressing the requirement is that the absolute error in matrix elements should be uniform over the elements. Note that $\boldsymbol{D}_1 = \boldsymbol{D}_2$ if $\boldsymbol{A}$ is symmetric, so as to make $\overline{\boldsymbol{A}}$ symmetric.

8.1.2 Backward error analysis.

In order to find the error bound due to the rounding error of a computer the computed solution $\boldsymbol{x} + \delta\boldsymbol{x}$ is related to contrived errors $\delta\boldsymbol{A}, \delta\boldsymbol{y}$, that is, the computed solution should be the exact solution of

$$(\boldsymbol{A} + \delta\boldsymbol{A})(\boldsymbol{x} + \delta\boldsymbol{x}) = (\boldsymbol{y} + \delta\boldsymbol{y}).$$

This procedure of finding perturbations in the data which match the actual computed solution is known as backward error analysis. (The forward error analysis is somewhat difficult.) Perturbation bounds for the L–U, Choleski and Q–U factorizations are known. Results are quoted later when discussing the solution of equations using the factorizations. Remember that usually error in the original data $\boldsymbol{A}$ and $\boldsymbol{y}$ will be larger than the roundoff error backward perturbations.

8.2 Direct methods for the solution of linear equations.

In this section the factorizations discussed in §7 are summarized and used to solve the set of linear equations

$$\boldsymbol{Ax} = \boldsymbol{y} \qquad\qquad (8.2.0)$$

where $\boldsymbol{A}$ is a square $n \times n$ matrix. As well, the problem of finding the solution $\boldsymbol{x}$ which minimizes

$$\|\boldsymbol{Ax} - \boldsymbol{y}\|_2,$$

where $\boldsymbol{A}$ is $m \times n$, $m > n$, $\text{rank}(\boldsymbol{A}) = n$, is discussed. This is the linear least squares problem.

8.2.1 Inversion costing n^2 operations.

There is of course a reason for using factorizations involving upper and lower triangular matrices. Equations in which the matrix is upper or lower triangular can be solved directly using the process known as forward or back substitution. As well the inversion of equations involving an orthogonal matrix is readily accomplished as the inverse is known with no further computation.

(a) Forward substitution. The set of equations

$$\boldsymbol{Lx} = \boldsymbol{y}$$

where $\boldsymbol{L}$ is lower triangular and invertible, has a solution $\boldsymbol{x}$ where formally

$$\boldsymbol{x} = \boldsymbol{L}^{-1}\boldsymbol{y}.$$

Components of $\boldsymbol{x}$ can be computed in the order $x_1, x_2, \ldots, x_n$. In detail

$$\begin{pmatrix} L_{11} & & & & \\ L_{21} & L_{22} & & & \\ L_{31} & L_{32} & L_{33} & & \\ \vdots & & & \ddots & \\ L_{n1} & L_{n2} & L_{n3} & \ldots & L_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix},$$

and

$$x_1 = y_1/L_{11}$$
$$x_2 = (y_2 - L_{21}x_1)/L_{22}$$
$$x_3 = (y_3 - L_{31}x_1 - L_{32}x_2)/L_{33}$$
$$\vdots$$
$$x_n = (y_n - \sum_{i=1}^{n-1} L_{ni}x_i)/L_{nn}.$$

Note that $\boldsymbol{L}^{-1}$ does not get computed. If $\boldsymbol{L}$ is written as $\boldsymbol{D} + \boldsymbol{S}$ where $\boldsymbol{D}$ is the diagonal of $\boldsymbol{L}$ and $\boldsymbol{S}$ is the strictly lower triangular part then the above computation can be written formally as

$$\boldsymbol{x} = \boldsymbol{D}^{-1}(\boldsymbol{y} - \boldsymbol{Sx}).$$

(b) Back substitution. The set of equations of the form

$$\boldsymbol{Ux} = \boldsymbol{y}$$

has a solution $\boldsymbol{x}$, where

$$\boldsymbol{x} = \boldsymbol{U}^{-1}\boldsymbol{y}.$$

Here the components of $\boldsymbol{x}$ can be computed in the order $x_n, x_{n-1}, \ldots, x_2, x_1$, where,

$$x_n = y_n/U_{nn},$$
$$x_{n-1} = (y_{n-1} - U_{n-1,n}x_n)/U_{n-1,n-1},$$
$$\vdots$$
$$x_1 = (y_1 - \sum_{i=2}^{n} U_{1i}x_i)/U_{11}.$$

Note that U^{-1} does not get computed. If U is written as $D + S$ where D is the diagonal of U and S is the strictly upper triangular part then the above computation can be written formally as

$$x = D^{-1}(y - Sx).$$

Both of these "inversions" cost $\frac{1}{2}n(n+1)$ multiplications and divisions and this cost compares favourably with the cost of a factorization which is $\mathcal{O}(n^3)$.

(c) Multiplying by orthogonal matrices. The orthogonal matrices in the factorizations are either stored in full or as a finite sequence of Householder transformations or plane rotations. Hence it is easy to "invert" such systems. Note if $Q = H_1 H_2 \cdots H_p$, then $Q^{-1} = Q^t = H_p \cdots H_2 H_1$.

8.2.2 Solution of general equations.

The detail of solving (8.2.0) has already been given in §7 and §8.2.1. As such it only remains to describe the major steps and results of the solution procedure. The factorization step is completely independent of any right hand side y and hence can be done independently of y, as was done in §7. Methods used in elementary texts, for example, Gauss elimination, are usually described using augmented matrices with row operations done on both the matrix and right hand side at the same time. Factorization descriptions simply give a picture of this process with the independent parts dissected. Included in the descriptions below are the cost (number of multiplications and divisions), the error analysis results, convenient approximations to $\chi(A)$, and the singularity test. Only the results for real matrices are given as the complex case is similar. In the error analysis, ϵ represents the machine precision and $f(n)$ represents polynomials in n, usually of order 3 (cubic) or less, corresponding to the number of arithmetic operations.

(a) Using the SVD.
$$Ax = y, \quad VDU^t x = y \quad \Rightarrow \quad x = UD^{-1}V^t y.$$

(i) Algorithm — compute in turn:
1. $A \to VDU^t$ $\mathcal{O}(n^3)$
2. $V^t y$ n^2
3. $D^{-1}(V^t y)$ n
4. $U(D^{-1}V^t y)$ n^2

(ii) Storage requirements for A, V, D, U, are $3n^2 + n$, which can be reduced to $2n^2$ by not storing V, and computing $V^t y$ as part of the algorithm.

(iii) Error analysis: If $\overline{V}, \overline{D}, \overline{U}$ are the computed V, D, U then

$$\|A - \overline{V}\,\overline{D}\,\overline{U}^t\| \le \epsilon f(n)\|A\|,$$

even though $\overline{V}, \overline{D}$ and $\overline{U}$ may not be close to V, D and U respectively, even assuming the same ordering and signs of columns of V and U.

(iv) $\chi(A) = d_1/d_n \quad (= \|A\|_2 \|A^{-1}\|_2)$.

(v) The smallest singular value close to zero is the test for close to singularity. This test would usually be done as a relative test, against the size of the largest singular value.

(b) Using the Q–U factorization.

$$Ax = y, \quad QUx = y \quad \Rightarrow \quad x = U^{-1}Q^t y.$$

(i) Algorithm — compute in turn:
1. $A \to QU$ $\frac{2}{3}n^3$
2. $Q^t y$ n^2
3. $U^{-1}(Q^t y)$ $\frac{1}{2}n^2$

(ii) Storage — requires one extra n vector to store Q and U over the matrix A.

(iii) Error analysis
$$\|\delta A\| \le \epsilon f_1(n)\|A\|$$
$$\|\delta y\| \le \epsilon f_2(n)\|y\|$$

(iv) If column pivoting is used then $|U_{11}/U_{nn}| \leq \chi(\boldsymbol{A})$ and is a good estimate, otherwise $\chi(\boldsymbol{A}) = \chi(\boldsymbol{U}) = \|\boldsymbol{U}\| \|\boldsymbol{U}^{-1}\|$ which costs $\frac{1}{6}n^3$ to compute. $\chi(\boldsymbol{A})$ can be estimated by solving with trial right hand sides to give maximum amplification as in LINPACK and LAPACK, that is, attempt to find that $\boldsymbol{x}$ which maximizes $\|\boldsymbol{Ax}\|/\|\boldsymbol{x}\|$. This procedure works well if $\boldsymbol{A}$ is ill-conditioned, that is, in the case that matters.

(v) Any U_{ii} close to zero is the test for close to singularity. This is more accurate if column pivoting is used. The relative test against the largest U_{ii} is usually used.

(c) Using the L–U factorization with pivoting.

$$\boldsymbol{Ax} = \boldsymbol{y}, \quad (\boldsymbol{PL})\boldsymbol{Ux} = y \quad \Rightarrow \quad \boldsymbol{x} = \boldsymbol{U}^{-1}(\boldsymbol{PL})^{-1}\boldsymbol{y}.$$

(i) Algorithm — compute in turn:
1. $\boldsymbol{A} \rightarrow (\boldsymbol{PL})\boldsymbol{U}$ $\frac{1}{3}n^3$
2. $(\boldsymbol{PL})^{-1}\boldsymbol{y}$ $\frac{1}{2}n^2$
3. $\boldsymbol{U}^{-1}\big((\boldsymbol{PL})^{-1}y\big)$ $\frac{1}{2}n^2$

(ii) No extra storage is needed, other than an integer array of length n to store the pivot information. $\boldsymbol{U}$ is stored over the upper triangle of $\boldsymbol{A}$, while $\boldsymbol{L}$ is stored in the strictly lower triangle of $\boldsymbol{A}$, the diagonal of 1's needing no storage.

(iii) Error analysis:
$$\|\delta\boldsymbol{A}\| \leq \epsilon f(n)(\|\boldsymbol{A}\| + \|\boldsymbol{U}\|).$$

Unfortunately $\|\boldsymbol{U}\|$ can be much larger than $\|\boldsymbol{A}\|$,

$$\boldsymbol{U} = (\boldsymbol{PL})^{-1}\boldsymbol{A},$$
$$\therefore \quad \|\boldsymbol{U}\| \leq \|(\boldsymbol{PL})^{-1}\| \|\boldsymbol{A}\|,$$

but $\|\boldsymbol{L}^{-1}\| \leq 2^{n-1}$ if pivoting is used (prove it using $\| \,.\, \|_\infty$), and this upper bound can be reached.

(iv) $\chi(\boldsymbol{A})$ could be indicated by $\max_i |U_{ii}|/\min_i |U_{ii}|$, but a better estimate is

$$\chi(\boldsymbol{A}) \leq \chi(\boldsymbol{L})\chi(\boldsymbol{U}) = \|\boldsymbol{L}\| \|\boldsymbol{L}^{-1}\| \|\boldsymbol{U}\| \|\boldsymbol{U}^{-1}\|$$

at a cost of $\frac{1}{3}n^3$ operations for the inverses. $\chi(\boldsymbol{A})$ can be estimated by solving with trial right hand sides to give maximum amplification as in LINPACK and LAPACK, that is, attempt to find that $\boldsymbol{x}$ which maximizes $\|\boldsymbol{Ax}\|/\|\boldsymbol{x}\|$. This procedure works well if $\boldsymbol{A}$ is ill-conditioned, that is, in the case that matters.

(v) Any U_{ii} close to zero, using row pivoting, is the test for closeness to singularity. A relative test against the norm of the matrix is usually required.

(d) Using the Choleski factorization.

$$\boldsymbol{Ax} = \boldsymbol{y}, \quad \boldsymbol{A} \text{ pos.def. } \boldsymbol{LL}^t\boldsymbol{x} = \boldsymbol{y} \quad \Rightarrow \quad \boldsymbol{x} = \boldsymbol{L}^{-t}\boldsymbol{L}^{-1}\boldsymbol{y}.$$

(i) Algorithm — compute in turn:
1. $\boldsymbol{A} \rightarrow \boldsymbol{LL}^t$ $\frac{1}{6}n^3$
2. $\boldsymbol{L}^{-1}\boldsymbol{y}$ $\frac{1}{2}n^2$
3. $\boldsymbol{L}^{-t}(\boldsymbol{L}^{-1}\boldsymbol{y})$ $\frac{1}{2}n^2$

(ii) No extra storage is needed, as $\boldsymbol{L}$ is stored over $\boldsymbol{A}$.

(iii) Error analysis: very stable.
$$\|\delta\boldsymbol{A}\| \leq \epsilon\|\boldsymbol{A}\|$$
$$\|\delta\boldsymbol{y}\| \leq \epsilon f(n)\|\boldsymbol{y}\|.$$

(iv) $\chi(\boldsymbol{A})$ could be indicated by $\big(\max_i L_{ii}/\min_i L_{ii}\big)^2$, or using $\frac{1}{6}n^3$ operations

$$\chi(\boldsymbol{A}) \leq (\|\boldsymbol{L}\| \|\boldsymbol{L}^{-1}\|)^2.$$

(v) Any L_{ii} close to zero, compared to the norm of the matrix, is the test for closeness to singularity.

8.3. Least squares methods.

See Lawson and Hanson (1974) for a more complete discussion and Jennings (1980) for applications in econometrics. Björk (1968) has an alternative stable algorithm for least squares. Note that Gram Schmidt orthogonalization is numerically unstable and should never be used unless orthogonalizing two or three vectors.

Write an overdetermined set of linear equations as

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{r},$$

where $\boldsymbol{A}$ is $m \times n$, $m > n$, $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y}$, $\boldsymbol{r} \in \mathbb{R}^m$. The residual vector $\boldsymbol{r}$ is defined for every $\boldsymbol{x}$ as $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}$. The least squares solution is that $\boldsymbol{x}$ which minimizes $\boldsymbol{r}^t\boldsymbol{r} = \|\boldsymbol{r}\|_2^2$. Let the function $f(\boldsymbol{x}) = \boldsymbol{r}^t\boldsymbol{r}$.

$$
\begin{aligned}
f(\boldsymbol{x}) &= (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y})^t(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}) \\
&= \boldsymbol{x}^t\boldsymbol{A}^t\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{x}^t\boldsymbol{A}^t\boldsymbol{y} + \boldsymbol{y}^t\boldsymbol{y}
\end{aligned}
$$

At a turning point $\nabla f = \boldsymbol{0}$, so that $2\boldsymbol{A}^t\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{A}^t\boldsymbol{y} = \boldsymbol{0}$, that is,

$$(\boldsymbol{A}^t\boldsymbol{A})\boldsymbol{x} = \boldsymbol{A}^t\boldsymbol{y}.$$

These are known as the normal equations. The condition for a unique minimum is that $\boldsymbol{A}^t\boldsymbol{A}$ be positive definite which is true if $\boldsymbol{A}$ has rank n.

8.3.1. Choleski algorithm for least squares.

(i)   Compute $\boldsymbol{A}^t\boldsymbol{A}$ $\qquad\qquad\qquad\qquad \frac{1}{2}n^2 m$

(ii)   Factor $\boldsymbol{A}^t\boldsymbol{A} \to \boldsymbol{L}\boldsymbol{L}^t$ $\qquad\qquad\qquad \frac{1}{6}n^3$

(iii)   Compute $\boldsymbol{L}^{-t}(\boldsymbol{L}^{-1}(\boldsymbol{A}^t\boldsymbol{y}))$ $\qquad\quad nm + n^2$

The error analysis for the Choleski algorithm shows that

$$\|\delta\boldsymbol{x}\| \leq \epsilon f(n)\chi^2(\boldsymbol{A})\|\boldsymbol{y}\| \quad \text{as} \quad \chi(\boldsymbol{A}^t\boldsymbol{A}) = \chi^2(\boldsymbol{A}).$$

Most of the accumulated error for this algorithm is in the computation of $\boldsymbol{A}^t\boldsymbol{A}$.

8.3.2. Golub algorithm for least squares.

See Golub (1966). If $\boldsymbol{A}$ is factored using the Q–U factorization, then the normal equations become

$$\boldsymbol{U}^t\boldsymbol{U}\boldsymbol{x} = (\,\boldsymbol{U}^t \quad \boldsymbol{0}\,)\,\boldsymbol{Q}^t\boldsymbol{y},$$

and writing $\boldsymbol{Q}^t\boldsymbol{y}$ as $\begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{pmatrix}$, where $\boldsymbol{y}_1 \in \mathbb{R}^n$,

$$\boldsymbol{U}^t\boldsymbol{U}\boldsymbol{x} = \boldsymbol{U}^t\boldsymbol{y}_1 \qquad \Rightarrow \qquad \boldsymbol{x} = \boldsymbol{U}^{-1}\boldsymbol{y}_1.$$

Algorithm:

(i)   Factorize $\boldsymbol{A} \to \boldsymbol{Q}\begin{pmatrix} \boldsymbol{U} \\ \boldsymbol{0} \end{pmatrix}$ $\qquad\qquad \frac{2}{3}n^2 m$

(ii)   Compute $\boldsymbol{Q}^t\boldsymbol{y}$ $\qquad\qquad\qquad 2nm - \frac{1}{2}n^2$

(iii)   Compute $\boldsymbol{U}^{-1}\boldsymbol{y}_1$ $\qquad\qquad\qquad \frac{1}{2}n^2$

Note: $\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2 \equiv \min_{\boldsymbol{x}} \left\| \begin{pmatrix} \boldsymbol{U} \\ \boldsymbol{0} \end{pmatrix} \boldsymbol{x} - \begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{pmatrix} \right\|_2 = \|\boldsymbol{y}_2\|_2 \quad \text{if } \boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}_1$. The residual vector is

$$\boldsymbol{r} = \boldsymbol{Q}\begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{y}_2 \end{pmatrix}.$$

The error analysis for the Golub algorithm shows that

$$\|\delta\boldsymbol{x}\| \leq \epsilon f_1(n)\chi(\boldsymbol{A})\|\boldsymbol{y}\| + \epsilon f_2(n)\chi^2(\boldsymbol{A})\|\boldsymbol{r}\|.$$

See Jennings and Osborne (1974) for details. The Golub algorithm is better in terms of roundoff error but involves about twice the computation than the Choleski method.

Estimates of $\chi(\boldsymbol{A})$ follow from earlier discussion for square matrices for the Choleski method, while for the Golub algorithm the condition number of $\boldsymbol{A}$ is equal to that of $\boldsymbol{U}$.

8.4 Iterative methods.

The methods for solving a set of linear equations to be discussed in this section apply mainly to systems which are large ($n > 500$) but have a sparse coefficient matrix. Usually in this case the cost of finding and storing a factorization with its resultant "fill in" is prohibitive. Further details of iterative methods can be found in Young (1971) and Varga (1962), while sparse matrix research was reviewed in Duff (1977). Blackford *et al.* (1997) discusses the LAPACK library modifications to parallel processing for all types of matrices. Dongarra *et al.* (1990) discusses the impact of parallelization on matrix computations.

8.4.1 Successive iteration (linear)

Suppose $\boldsymbol{A}$ and $\boldsymbol{y}$ are given and a guess at the solution $\boldsymbol{x}$, say $\boldsymbol{x}_1$ is known. Then if

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{y},$$

the iteration

$$\boldsymbol{x}_{i+1} = \boldsymbol{G}_i\boldsymbol{x}_i + \boldsymbol{r}_i, \quad i = 1, 2, \ldots,$$

is proposed where $\boldsymbol{G}_i$ is a matrix and $\boldsymbol{r}_i$ a vector. Convergence requires that

$$\lim_{i \to \infty} \|\boldsymbol{x}_i - \boldsymbol{x}\| = 0.$$

Consistency: The solution must satisfy the iteration

$$\boldsymbol{A}^{-1}\boldsymbol{y} = \boldsymbol{G}_i\boldsymbol{A}^{-1}\boldsymbol{y} + \boldsymbol{r}_i,$$

so that $\boldsymbol{G}_i$ and $\boldsymbol{r}_i$ are related.

$$\boldsymbol{r}_i = (\boldsymbol{I} - \boldsymbol{G}_i)\boldsymbol{A}^{-1}\boldsymbol{y}.$$

Convergence: Let $\boldsymbol{z}_i = \boldsymbol{x}_i - \boldsymbol{x} = \boldsymbol{x}_i - \boldsymbol{A}^{-1}\boldsymbol{y}$.

$$\begin{aligned}
\boldsymbol{z}_{i+1} &= \boldsymbol{x}_{i+1} - \boldsymbol{x} \\
&= \boldsymbol{G}_i\boldsymbol{x}_i + (\boldsymbol{I} - \boldsymbol{G}_i)\boldsymbol{x} - \boldsymbol{x} \\
&= \boldsymbol{G}_i(\boldsymbol{x}_i - \boldsymbol{x}) \\
&= \boldsymbol{G}_i\boldsymbol{z}_i \\
&= \boldsymbol{G}_i\boldsymbol{G}_{i-1}\cdots\boldsymbol{G}_1\boldsymbol{z}_1 \\
&= \boldsymbol{H}_i\boldsymbol{z}_1 \quad \text{say}
\end{aligned}$$

Now if $\boldsymbol{H}_i \to 0$ as $i \to \infty$ then $\boldsymbol{x}_i \to \boldsymbol{x}$. If $\boldsymbol{G}_i = \boldsymbol{G}$, (constant) then $\boldsymbol{H}_i = \boldsymbol{G}^i$ and convergence is assured if any of; $\|\boldsymbol{G}\| < 1$, for any matrix norm, or, $\max_j |\lambda_j| < 1$, where λ_j is an eigenvalue of $\boldsymbol{G}$.

Jacobi iteration.

Write the matrix $\boldsymbol{A}$ as

$$\boldsymbol{A} = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{U},$$

where $\boldsymbol{L}$ is strictly lower triangular, $\boldsymbol{D}$ the diagonal elements of $\boldsymbol{A}$, and $\boldsymbol{U}$ is strictly upper triangular. The Jacobi method is

$$\boldsymbol{D}\boldsymbol{x}_{i+1} = \boldsymbol{y} - (\boldsymbol{L} + \boldsymbol{U})\boldsymbol{x}_i$$

$$\therefore \quad \boldsymbol{G} = -\boldsymbol{D}^{-1}(\boldsymbol{L} + \boldsymbol{U}), \quad \boldsymbol{r} = \boldsymbol{D}^{-1}\boldsymbol{y}.$$

It is easily shown that if $\boldsymbol{A}$ is diagonally dominant, that is

$$\sum_{\substack{i \neq j \\ i=1}}^{n} |A_{ij}| \leq |A_{jj}| \quad \text{for all} j \quad \text{or} \quad \sum_{\substack{i \neq j \\ j=1}}^{n} |A_{ij}| \leq |A_{ii}| \quad \text{for all } i$$

then this method is convergent. (Prove it!!)

Gauss-Siedel iteration.

Here, essentially a forward substitution is computed.

$$(\boldsymbol{L} + \boldsymbol{D})\boldsymbol{x}_{i+1} = \boldsymbol{y} - \boldsymbol{U}\boldsymbol{x}_i$$
$$\boldsymbol{G} = -(\boldsymbol{L} + \boldsymbol{D})^{-1}\boldsymbol{U}, \quad \boldsymbol{r} = (\boldsymbol{L} + \boldsymbol{D})^{-1}\boldsymbol{y}$$
$$\boldsymbol{x}_{i+1} = \boldsymbol{D}^{-1}(\boldsymbol{y} - \boldsymbol{U}\boldsymbol{x}_i - \boldsymbol{L}\boldsymbol{x}_{i+1})$$

This can be thought of as "if better values for $\boldsymbol{x}$ are available in $\boldsymbol{x}_{i+1}$, why not use them". This has better convergence properties than Jacobi in that it is faster. Jacobi and Gauss Siedel iterations converge or diverge together, depending on the properties of $\boldsymbol{A}$. There are variants of this algorithm, for example, doing a back substitution instead, which corresponds to computing the new vector in reverse order. Alternating directions methods combine forward and backward methods alternately for faster convergence, as well as re-orderings of equations and variables in some problems.

Relaxation of Gauss-Siedel.

Consider $\boldsymbol{x}_{i+1} - \boldsymbol{x}_i$ as a direction of search. Then define a next iterate as

$$\boldsymbol{x}_i + w(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i).$$

If $w > 1$ this is called over-relaxation while if $w < 1$, under relaxation. Formally

$$\boldsymbol{x}_{i+1} = w\boldsymbol{D}^{-1}(\boldsymbol{y} - \boldsymbol{U}\boldsymbol{x}_i - \boldsymbol{L}\boldsymbol{x}_{i+1}) + (1 - w)\boldsymbol{x}_i$$

but this is usually best explained in terms of elements of vectors. For $j = 1, 2, \ldots, n$

$$\boldsymbol{x}_{i+1}(j) = wD_{jj}^{-1}\{y_j - \sum_{k>j} U_{jk}\boldsymbol{x}_i(k) - \sum_{k<j} L_{jk}\boldsymbol{x}_{i+1}(k)\} + (1 - w)\boldsymbol{x}_i(j).$$

Some problems, especially in the numerical solution of partial differential equations are very efficiently solved using variants of relaxation and alternating directions.

8.4.2 The conjugate gradient method.

See Hestines and Steifel (1952) for the original detail. Let the matrix $\boldsymbol{A}$ be positive definite. Then the quadratic form

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}^t\boldsymbol{x},$$

has a minimum when

$$(\nabla f)^t = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} = \boldsymbol{0}.$$

The conjugate gradient method is used to find $\boldsymbol{x}$ which gives the minimum of $f(\boldsymbol{x})$ by creating search directions from some starting point $\boldsymbol{x}_0$ say. At the i–th step form a search direction $\boldsymbol{p}_i$ and move along it a distance α_i

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i\boldsymbol{p}_i,$$

where α_i is chosen so that $\boldsymbol{x}_{i+1}$ is a minimum of $f(\boldsymbol{x})$ in this direction. In particular the gradient of $f(\boldsymbol{x})$ at $\boldsymbol{x}_{i+1}$ will be perpendicular to $\boldsymbol{p}_i$,

$$\nabla f_{i+1}\boldsymbol{p}_i = 0, \quad \text{where} \quad \nabla f_{i+1} = \nabla f(\boldsymbol{x}_{i+1}).$$

This gives

$$\alpha_i = \frac{\nabla f_i \boldsymbol{p}_i}{\boldsymbol{p}_i^t \boldsymbol{A}\boldsymbol{p}_i}.$$

Now

$$\boldsymbol{x}_n = \boldsymbol{x}_i + \sum_{j=i}^{n-1} \alpha_j \boldsymbol{p}_j$$

and

$$\boldsymbol{A}\boldsymbol{x}_n - \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_i - \boldsymbol{y} + \sum_{j=i}^{n-1} \alpha_j \boldsymbol{A}\boldsymbol{p}_j$$

or,

$$\nabla f_n^t = \nabla f_i^t + \sum_{j=i}^{n-1} \alpha_j \boldsymbol{A}\boldsymbol{p}_j.$$

In particular,

$$\boldsymbol{p}_{i-1}^t \nabla f_n^t = \boldsymbol{p}_{i-1}^t \nabla f_i^t + \sum_{j=i}^{n-1} \alpha_j \boldsymbol{p}_{i-1}^t \boldsymbol{A}\boldsymbol{p}_j, \tag{8.4.1}$$

and as $\boldsymbol{p}_{i-1}^t \nabla f_i^t = 0$, then $\boldsymbol{p}_{i-1}^t \nabla f_n^t = 0$, if $\boldsymbol{p}_{i-1}^t \boldsymbol{A}\boldsymbol{p}_j = 0$, $j = i, i+1, \ldots, n-1$. Hence $\boldsymbol{p}_j$ is chosen such that

$$\boldsymbol{p}_i^t \boldsymbol{A}\boldsymbol{p}_j = 0 \quad i \neq j.$$

This is known as $\boldsymbol{A} - conjugacy$, and the set of vectors $\{\boldsymbol{p}_j\}_{j=0}^{n-1}$ is said to be $\boldsymbol{A} - conjugate$. It is easy to show the set of non-zero vectors $\{\boldsymbol{p}_j\}_{j=0}^{n-1}$ form a basis for $\mathbb{R}^n$ and hence from (8.4.1)

$$\nabla f_n = \boldsymbol{0}$$

that is, $\boldsymbol{x}_n$ is the point giving a minimum of $f(\boldsymbol{x})$. To find the vectors $\boldsymbol{p}_j$, suppose $\boldsymbol{p}_0 = -\nabla f_0$, and let

$$\boldsymbol{p}_{i+1} = -\nabla f_{i+1}^t + \beta_i \boldsymbol{p}_i, \quad i = 0, 1, 2, \ldots, n-1.$$

Now as

$$\boldsymbol{p}_i^t \boldsymbol{A}\boldsymbol{p}_{i+1} = 0$$
$$-\boldsymbol{p}_i^t \boldsymbol{A} \nabla f_{i+1}^t + \beta_i \boldsymbol{p}_i^t \boldsymbol{A}\boldsymbol{p}_i = 0$$
$$\Rightarrow \beta_i = \frac{-\boldsymbol{p}_i^t \boldsymbol{A} \nabla f_{i+1}^t}{\boldsymbol{p}_i^t \boldsymbol{A}\boldsymbol{p}_i}$$
$$= \frac{-\boldsymbol{p}_i^t \boldsymbol{A}(\boldsymbol{A}\boldsymbol{x}_{i+1} - \boldsymbol{y})}{\boldsymbol{p}_i^t \boldsymbol{A}\boldsymbol{p}_i}$$

8.5 Exercises.

1. (i) Solve these simultaneous equations exactly. This is $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$.

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 1 \\ 2 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix}.$$

   (ii) Solve these simultaneous equations exactly. This is $(\boldsymbol{A} + \delta\boldsymbol{A})(\boldsymbol{x} + \delta\boldsymbol{x}) = \boldsymbol{y}$.

$$\begin{pmatrix} \epsilon & 1 & 1 \\ 1 & 2 & 1 \\ 2 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix}.$$

   Is there much change in solution for small ϵ? Expand your answer as a Taylor series to first order.

   (iii) Solve with no pivoting using three significant figure arithmetic (chop not round) the system in (ii) with $\epsilon = 10^{-4}$. Follow a strict Gaussian elimination algorithm where the row operations are $row_j \leftarrow row_j - m_j row_i$, $j > i$, that is, subtract a multiple of the pivot row from row_j. Does the solution change very much from the answer of (ii)?

   (iv) Is the system ill-conditioned compared to the small change to the (1,1) element? Why doesn't your solution to (iii) satisfy Theorem 8.1.1?

2. (i) Indicate how you would solve the matrix system (a rank one correction or update)

$$(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^t)\boldsymbol{x} = \boldsymbol{y},$$

   given the factors of $\boldsymbol{A} = (\boldsymbol{P}\boldsymbol{L})\boldsymbol{U}$, in order n^2 operations. You know the inverse of $\boldsymbol{I} + \boldsymbol{w}\boldsymbol{v}^t$ from exercise 6.7.1. Illustrate the algorithm by solving

$$\left(\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} + \begin{pmatrix} 0 & 10^{-4} \\ 0 & 0 \end{pmatrix} \right) \boldsymbol{x} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

(ii) Generalize to $(\boldsymbol{A} + \boldsymbol{Z}\boldsymbol{V}^t)\boldsymbol{x} = \boldsymbol{y}$ where $\boldsymbol{Z}$ and $\boldsymbol{V}$ are $n \times r$ matrices ($r < n$). This is the Sherman–Morrison–Woodbury formula for r rank one updates simultaneously. Hint: show that the inverse of $\boldsymbol{I} + \boldsymbol{W}\boldsymbol{V}^t$ is $\boldsymbol{I} - \boldsymbol{W}\boldsymbol{R}\boldsymbol{V}^t$, where $\boldsymbol{R} = (\boldsymbol{I} + \boldsymbol{V}^t\boldsymbol{W})^{-1}$, an $r \times r$ matrix.

3. Suggest an efficient method for solving a matrix equation of order 1000 which is pentadiagonal except that $A_{1,1000}$ and $A_{1000,1}$ are non-zero. Show that this can be done in about 19–25 thousand operations depending on whether row pivoting needs to be used.

4. Consider the equation

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.01 \end{pmatrix} \boldsymbol{x} = \begin{pmatrix} 1 \\ 0.1 \\ 0.01 \end{pmatrix}.$$

Show that the relative error bounds for $\|\delta\boldsymbol{x}\|/\|\boldsymbol{x}\|$ in (8.1.3) can be reached by choosing appropriate $\delta\boldsymbol{A}$ with $\delta\boldsymbol{y} = 0$ and choosing appropriate $\delta\boldsymbol{y}$ with $\delta\boldsymbol{A} = 0$, that is, first simplify the theorem by putting in turn one of $\delta\boldsymbol{y}$ and $\delta\boldsymbol{A}$ to zero. What is the biggest perturbation $\delta\boldsymbol{A}$ so that all smaller perturbations ensure $(\boldsymbol{A} + \delta\boldsymbol{A})^{-1}$ exists? Hint: use the infinity norm and choose $\delta\boldsymbol{A} = -\alpha\boldsymbol{e}_3\boldsymbol{e}_3^t$ for some suitable range of α values. This shows that Theorem 8.1.1 cannot be improved.

5. Let $\boldsymbol{A}$ be an $m \times n$ matrix $m > n$ and let

$$\boldsymbol{A} = \boldsymbol{Q}\begin{pmatrix} \boldsymbol{U} \\ \boldsymbol{0} \end{pmatrix}.$$

Consider the set of least squares equations, $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{r}$. If $\boldsymbol{x}$ is chosen to minimize $\|\boldsymbol{r}\|_2$ show that

$$\boldsymbol{r} = (\boldsymbol{I} - \boldsymbol{A}(\boldsymbol{A}^t\boldsymbol{A})^{-1}\boldsymbol{A}^t)\boldsymbol{y}$$
$$= \boldsymbol{Q}\begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{m-n} \end{pmatrix}\boldsymbol{Q}^t\boldsymbol{y}.$$

6. The linear system of equations

$$\begin{pmatrix} 1 & -a \\ -a & 1 \end{pmatrix} \boldsymbol{x} = \boldsymbol{y}$$

where a is real, can, under certain conditions be solved by the iterative method

$$\begin{pmatrix} 1 & 0 \\ -wa & 1 \end{pmatrix} \boldsymbol{x}_{i+1} = \begin{pmatrix} 1 - w & wa \\ 0 & 1 - w \end{pmatrix} \boldsymbol{x}_i + w\boldsymbol{y}.$$

for which values of a is the method convergent for $w = 1$? Use Maple to look for the best value of w in terms of a, by considering the matrix $\boldsymbol{G}$ defined in the notes. Can you generalise this to the corresponding $n \times n$ symmetric tridiagonal matrix?

7. Compute the first few iterations of the Jacobi and Gauss-Siedel iterations on the system,

$$\begin{pmatrix} 4 & 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 & 4 \end{pmatrix} \boldsymbol{x} = \begin{pmatrix} 3 \\ -2 \\ 2 \\ -2 \\ 3 \end{pmatrix},$$

starting at $\boldsymbol{x} = 0$. (Solution is $[1, -1, 1, -1, 1]^t$.) Compute the two over-relaxation iterations using $w = 1.5$. (If you write a simple 10 minute program to do this, compare the convergence rates.)

8. Write an algorithm for the conjugate gradient method. Because of the poor numerical performance of this algorithm it is usual to restart it every n iterations.

9a. Let $\boldsymbol{A}$ be an $n \times n$ matrix which is invertible.

(i) Show the inverse of the partitioned matrix $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{a} \\ \boldsymbol{b}^t & z \end{pmatrix}$ is

$$\begin{pmatrix} \boldsymbol{A}^{-1} + w\boldsymbol{A}^{-1}\boldsymbol{a}\boldsymbol{b}^t\boldsymbol{A}^{-1} & -w\boldsymbol{A}^{-1}\boldsymbol{a} \\ -w\boldsymbol{b}^t\boldsymbol{A}^{-1} & w \end{pmatrix}$$

where $w = (z - \boldsymbol{b}^t\boldsymbol{A}^{-1}\boldsymbol{a})^{-1}$.

(ii) What is the inverse (if it exists) if $\boldsymbol{A}^{-1}$ does not exist and $z \neq 0$.

(iii) Does the inverse exist if $z = 0$ and $\boldsymbol{A}^{-1}$ does not exist?

(iv) In (i) if $\boldsymbol{A}$ has been factored to $(\boldsymbol{PL})\boldsymbol{U}$ how could you solve

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{a} \\ \boldsymbol{b}^t & z \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ u \end{pmatrix} = \begin{pmatrix} \boldsymbol{y} \\ v \end{pmatrix}$$

in $\mathcal{O}(n^2)$ operations?

9b. Continuing this question let $\boldsymbol{A}$ be a symmetric positive definite matrix and $\boldsymbol{b} = \boldsymbol{a}$.

(i) What are the conditions on $\boldsymbol{a}$ and z to keep the $(n+1) \times (n+1)$ matrix positive definite?

(ii) If $\boldsymbol{A}$ has been factored to $\boldsymbol{L}\boldsymbol{L}^t$ how could you solve the order $n+1$ system in $\mathcal{O}(n^2)$ operations?

10. Find an expression for the inverse of the (not symmetric) partitioned matrix

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}$$

where $\boldsymbol{A}$ and $\boldsymbol{D}$ are square and not necessarily of the same order. You should only have two inverses in your expression. Note the conditions for existence of the computations. Compare the operation cost for solving the $2n \times 2n$ system

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{pmatrix}$$

using L–U factorization, with solving the system in partitioned form using L–U factorization on $n \times n$ systems. Note the common factors in the expressions for $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, and that $\boldsymbol{A}\boldsymbol{B}\boldsymbol{x}$ is more efficiently computed as $\boldsymbol{A}(\boldsymbol{B}\boldsymbol{x})$ and not $(\boldsymbol{A}\boldsymbol{B})\boldsymbol{x}$.

11. Given the two systems of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$ and $\boldsymbol{A}\bar{\boldsymbol{x}} = \boldsymbol{y} + \boldsymbol{z}$, show that the minimum value of $\|\boldsymbol{x} - \bar{\boldsymbol{x}}\|_2$ is $\|\boldsymbol{z}\|/d_1$ and the maximum value is $\|\boldsymbol{z}\|/d_n$. Here d_1 and d_n are the largest and smallest singular values of $\boldsymbol{A}$.

12. Show that if partial pivoting is used in the L–U factorization

$$\|\boldsymbol{U}\|_\infty \leq 2^{n-1}\|\boldsymbol{A}\|_\infty.$$

(It is easy to verify that the 1-norm gives a larger bound while it is somewhat harder to show that the 2-norm gives a larger bound.)

13. Show that the eigenvectors of the system

$$(\boldsymbol{A} - \lambda\boldsymbol{B})\boldsymbol{v} = \boldsymbol{0},$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are symmetric, are both $\boldsymbol{A}$–conjugate and $\boldsymbol{B}$–conjugate, if all the eigenvalues are distinct.

14. Consider the system

$$\begin{pmatrix} 1 & 0 & -1/4 & -1/4 \\ 0 & 1 & -1/4 & -1/4 \\ -1/4 & -1/4 & 1 & 0 \\ -1/4 & -1/4 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{pmatrix}$$

and use MATLAB to do the following computations.

(i) Starting with $\boldsymbol{x} = \boldsymbol{0}$, iterate the Jacobi method until iterates are within 10^{-10} of each other. Note the number of iterations to achieve this accuracy.

(ii) Repeat (i) using the Gauss-Seidel method.

(iii) For both methods find the matrix $\boldsymbol{G}$ of the notes and find the norm and the spectral radius (largest in magnitude eigenvalue) and hence explain the different convergence behaviour of (i) and (ii).

(iv) Is there a relationship between the norm of the distance of an iterate from the true solution and the norm of the difference of two successive iterates?

15. Show that it costs $n^3/6$ multiplications and divisions to compute the inverse matrix of an upper triangular matrix. Remember that the inverse is also upper triangular so there is no need to compute the zeros in the lower triangle.

16. What are the operation counts for computing the following most efficiently?

   (a) The product of two upper triangular matrices.

   (b) The product of an upper triangular and lower triangular matrix.

   (c) The product of the inverse of a lower triangular matrix and an upper triangular matrix, given the lower triangular matrix. Is this best done via Question 15 or by Gauss elimination in a tableau method?

   (d) The product of the inverse of a lower triangular matrix and a lower triangular matrix, given the first lower triangular matrix. Is this best done via Question 15 or by Gauss elimination in a tableau method?

9. The Generalized Inverse Problem.

The concept of a generalized inverse is examined with the view of providing an 'inverse' for any linear mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$. The geometric aspects are emphasized and used to explain the computation of solutions to any system of simultaneous linear equations in both the analytical and practical sense. Further material can be found in Ben-Israel and Greville (1974). Original work in this area goes back to Moore (1920) and Penrose (1956).

9.1 The geometry of a linear mapping.

Let a linear map from $\mathbb{R}^n$ to $\mathbb{R}^m$ be represented by the $m \times n$ matrix $\boldsymbol{A}$ in the usual orthogonal bases. The abstract distinction between the linear map $\mathcal{A}$, and the matrix $\boldsymbol{A}$ is not made, hence the linear map will be called $\boldsymbol{A}$. Let $\langle \cdot, \cdot \rangle$ be the usual inner product and $\| \cdot \|$ be the Euclidean norm for the space $\mathbb{R}^n$ or $\mathbb{R}^m$. Recall,

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^t \boldsymbol{y} = \sum_{i=1}^{n} x_i y_i \ (= \boldsymbol{y}^t \boldsymbol{x}) = \langle \boldsymbol{y}, \boldsymbol{x} \rangle,$$

and

$$\langle \boldsymbol{x}, \boldsymbol{x} \rangle^{1/2} = \|\boldsymbol{x}\|_2 = \|\boldsymbol{x}\|.$$

This inner product and norm are invariant to multiplication by an orthogonal matrix.

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{Q}\boldsymbol{x}, \boldsymbol{Q}\boldsymbol{y} \rangle$$

$$\|\boldsymbol{x}\| = \|\boldsymbol{Q}\boldsymbol{x}\|$$

From §5 this means that this inner product and norm are invariant to change of orthonormal basis

$$\boldsymbol{I}\boldsymbol{I}^t \boldsymbol{x} = \sum_{i=1}^{n} \langle \boldsymbol{e}_i, \boldsymbol{x} \rangle \boldsymbol{e}_i = \sum_{i=1}^{n} \langle \boldsymbol{q}_i, \boldsymbol{x} \rangle \boldsymbol{q}_i = \boldsymbol{Q}\boldsymbol{Q}^t \boldsymbol{x}$$

Note that $(\boldsymbol{I}\boldsymbol{x})_i = \langle \boldsymbol{e}_i, \boldsymbol{x} \rangle$ is the i–th coordinate of $\boldsymbol{x}$ with respect to the standard basis and $(\boldsymbol{Q}^t \boldsymbol{x})_i = \langle \boldsymbol{q}_i, \boldsymbol{x} \rangle$ is the i–th coordinate of $\boldsymbol{x}$ with respect to the orthogonal basis $[\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n]$. From §4 the adjoint of a real mapping is the transpose matrix as for all $\boldsymbol{x}$ and $\boldsymbol{y}$

$$\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{x}, \boldsymbol{A}^t \boldsymbol{y} \rangle.$$

The mapping $\boldsymbol{A}$ divides $\mathbb{R}^n$ and $\mathbb{R}^m$ into 'natural' orthogonal subspaces, namely

$$\mathbb{R}^n = \mathcal{N}(\boldsymbol{A}) \oplus \mathcal{R}(\boldsymbol{A}^t),$$
$$\mathbb{R}^m = \mathcal{N}(\boldsymbol{A}^t) \oplus \mathcal{R}(\boldsymbol{A}).$$

It is easy to show that

$$\mathcal{N}(\boldsymbol{A})^\perp = \mathcal{R}(\boldsymbol{A}^t),$$
$$\mathcal{N}(\boldsymbol{A}^t)^\perp = \mathcal{R}(\boldsymbol{A}).$$

Note the dimensions of $\mathcal{R}(\boldsymbol{A})$ and $\mathcal{R}(\boldsymbol{A}^t)$ are the same and is the rank of $\boldsymbol{A}$ denoted r. Hence the dimensions of $\mathcal{N}(\boldsymbol{A})$ is $n - r$ and that of $\mathcal{N}(\boldsymbol{A}^t)$ is $m - r$. It is easy to show that $\boldsymbol{A}$ maps all elements of $\mathcal{R}(\boldsymbol{A}^t)$ to $\mathcal{R}(\boldsymbol{A})$, while $\boldsymbol{A}^t$ maps all elements of $\mathcal{R}(\boldsymbol{A})$ to $\mathcal{R}(\boldsymbol{A}^t)$.

9.2 Natural orthogonal bases for a mapping.

9.2.1 The Q–U factorization.

Recall that an $m \times n$ matrix of rank r can be factored to

$$A = Q \begin{pmatrix} L & 0 \\ 0 & 0 \end{pmatrix} P^t$$

where $\boldsymbol{Q}$ is $m \times m$ orthogonal, $\boldsymbol{P}$ is $n \times n$ orthogonal and $\boldsymbol{L}$ is $r \times r$ lower triangular (of rank r). If $\boldsymbol{Q}$ and $\boldsymbol{P}$ are partitioned appropriately then

$$\boldsymbol{A} = (\,\boldsymbol{Q}_1 \quad \boldsymbol{Q}_2\,) \begin{pmatrix} \boldsymbol{L} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{P}_1^t \\ \boldsymbol{P}_2^t \end{pmatrix} = \boldsymbol{Q}_1 \boldsymbol{L} \boldsymbol{P}_1^t$$

where $\boldsymbol{Q}_1$ is $m \times r$ and $\boldsymbol{P}_1$ is $n \times r$. If $\boldsymbol{q}_i$ are columns of $\boldsymbol{Q}$ and $\boldsymbol{p}_i$ are columns of $\boldsymbol{P}$ it is easy to show that

$$\mathcal{R}(\boldsymbol{A}) = \mathrm{span}\{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_r\},$$
$$\mathcal{N}(\boldsymbol{A}^t) = \mathrm{span}\{\boldsymbol{q}_{r+1}, \boldsymbol{q}_{r+2}, \ldots, \boldsymbol{q}_m\},$$
$$\mathcal{R}(\boldsymbol{A}^t) = \mathrm{span}\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_r\},$$
$$\mathcal{N}(\boldsymbol{A}) = \mathrm{span}\{\boldsymbol{p}_{r+1}, \boldsymbol{p}_{r+2}, \ldots, \boldsymbol{p}_n\}.$$

So the two orthogonal matrices $\boldsymbol{Q}$ and $\boldsymbol{P}$ of the Q–U factorization give a natural orthonormal basis for the linear mapping $\boldsymbol{A}$. The matrix of the mapping in the new bases is of course

$$\begin{pmatrix} \boldsymbol{L} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}.$$

since

$$\boldsymbol{A}\boldsymbol{P} = \boldsymbol{Q} \begin{pmatrix} \boldsymbol{L} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}.$$

$\boldsymbol{Q}_1$ and $\boldsymbol{P}_1$ and $\boldsymbol{L}$ are unique up to multiplication by diagonal matrices of ± 1.

9.2.2 The singular value decomposition.

The $m \times n$ matrix $\boldsymbol{A}$ of rank r can be factored to

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^t$$

where $\boldsymbol{V}$ is $m \times m$ orthogonal, $\boldsymbol{U}$ is $n \times n$ orthogonal and $\boldsymbol{D}$ is $m \times n$ diagonal with $D_{ii} = d_i$, $i = 1, 2, \ldots, \min(m, n)$ and $d_1 \geq d_2 \geq \ldots \geq d_r > 0$, $d_{r+1} = d_{r+2} = \ldots = d_{\min(m,n)} = 0$. If $\boldsymbol{v}_i = k_i(\boldsymbol{V})$ and $\boldsymbol{u}_i = k_i(\boldsymbol{U})$ then

$$\boldsymbol{A} = \sum_{i=1}^{r} d_i \boldsymbol{v}_i \boldsymbol{u}_i^t,$$

that is, $\boldsymbol{A}$ is the sum of r rank one matrices which in a sense are orthogonal to each other. Partitioning according to the zeros of $\boldsymbol{D}$ produces

$$\boldsymbol{A} = (\,\boldsymbol{V}_1 \quad \boldsymbol{V}_2\,) \begin{pmatrix} \widehat{\boldsymbol{D}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{U}_1^t \\ \boldsymbol{U}_2^t \end{pmatrix} = \boldsymbol{V}_1 \widehat{\boldsymbol{D}} \boldsymbol{U}_1^t,$$

where $\boldsymbol{V}_1$ is $m \times r$, $\boldsymbol{U}_1$ is $n \times r$ and $\widehat{\boldsymbol{D}}$ is $r \times r$ diagonal. It is easy to show that

$$\mathcal{R}(\boldsymbol{A}) = \mathrm{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_r\},$$
$$\mathcal{N}(\boldsymbol{A}^t) = \mathrm{span}\{\boldsymbol{v}_{r+1}, \ldots, \boldsymbol{v}_m\},$$
$$\mathcal{R}(\boldsymbol{A}^t) = \mathrm{span}\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r\},$$
$$\mathcal{N}(\boldsymbol{A}) = \mathrm{span}\{\boldsymbol{u}_{r+1}, \ldots, \boldsymbol{u}_n\}.$$

9.3 Canonical form of a linear map.

By choosing appropriate bases in $\mathbb{R}^n$ and $\mathbb{R}^m$ every linear mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$ can be represented by a diagonal matrix. The basis for $\mathbb{R}^n$ is the columns of $\boldsymbol{U}$ and the basis for $\mathbb{R}^m$ is the columns of $\boldsymbol{V}$. Hence the $m \times n$ matrix representing the mapping is the diagonal matrix of the SVD,

$$\boldsymbol{A}\boldsymbol{U} = \boldsymbol{V}\boldsymbol{D},$$

$$\boldsymbol{A}\,(\,\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \ldots \quad \boldsymbol{u}_n\,) = (\,\boldsymbol{v}_1 \quad \boldsymbol{v}_2 \quad \ldots \quad \boldsymbol{v}_m\,)\,\boldsymbol{D},$$

$$\text{or} \quad \boldsymbol{A}\boldsymbol{u}_i = \begin{cases} d_i \boldsymbol{v}_i & i = 1, 2, \ldots, r, \\ \boldsymbol{0} & i = r+1, \ldots, n. \end{cases}$$

Likewise the mapping $\boldsymbol{A}^t : \mathbb{R}^m \to \mathbb{R}^n$ is diagonal and represented by $\boldsymbol{D}^t$ if the above bases are chosen

$$\boldsymbol{A}^t = \boldsymbol{U} \boldsymbol{D}^t \boldsymbol{V}^t$$

or

$$\boldsymbol{A}^t \boldsymbol{V} = \boldsymbol{U} \boldsymbol{D}^t$$

or

$$\boldsymbol{A}^t \boldsymbol{v}_i = \begin{cases} d_i \boldsymbol{u}_i & i = 1, 2, \ldots, r, \\ \boldsymbol{0} & i = r + 1, \ldots, m. \end{cases}$$

The projection of vectors in the usual bases to the new bases represented by $\boldsymbol{U}$ and $\boldsymbol{V}$ are,

$$\boldsymbol{x} = \boldsymbol{U}\boldsymbol{U}^t \boldsymbol{x} = \sum_{i=1}^{n} \langle \boldsymbol{u}_i, \boldsymbol{x} \rangle \boldsymbol{u}_i,$$

$$\boldsymbol{y} = \boldsymbol{V}\boldsymbol{V}^t \boldsymbol{y} = \sum \langle \boldsymbol{v}_i, \boldsymbol{y} \rangle \boldsymbol{v}_i,$$

and using $\boldsymbol{A} = \sum_{i=1}^{r} d_i \boldsymbol{v}_i \boldsymbol{u}_i^t$,

$$\boldsymbol{A}\boldsymbol{x} = \sum_{i=1}^{r} d_i \boldsymbol{v}_i \boldsymbol{u}_i^t \sum_{j=1}^{n} \langle \boldsymbol{u}_j, \boldsymbol{x} \rangle \boldsymbol{u}_j,$$

$$= \sum_{i=1}^{r} d_i \langle \boldsymbol{u}_i, \boldsymbol{x} \rangle \boldsymbol{v}_i.$$

Hence if $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, equating coefficients of $\boldsymbol{v}_i$ gives

$$\langle \boldsymbol{v}_i, \boldsymbol{y} \rangle = d_i \langle \boldsymbol{u}_i, \boldsymbol{x} \rangle, \ i = 1, 2, \ldots, r,$$
$$\langle \boldsymbol{v}_i, \boldsymbol{y} \rangle = 0, \ i = r + 1, \ldots, m.$$

So the component of $\boldsymbol{A}\boldsymbol{x}$ in the direction $\boldsymbol{v}_i$ is related only to the component of $\boldsymbol{x}$ in the direction $\boldsymbol{u}_i$. Similarly the mapping $\boldsymbol{A}^t$,

$$\boldsymbol{A}^t \boldsymbol{y} \quad (= \boldsymbol{x}) = \sum_{i=1}^{r} d_i \langle \boldsymbol{v}_i, \boldsymbol{y} \rangle \boldsymbol{u}_i,$$

$$\langle \boldsymbol{u}_i, \boldsymbol{x} \rangle = d_i \langle \boldsymbol{v}_i, \boldsymbol{y} \rangle, \quad i = 1, 2, \ldots, r$$
$$\langle \boldsymbol{u}_i, \boldsymbol{x} \rangle = 0, \quad i = r + 1, \ldots, n.$$

So the component of $\boldsymbol{A}^t \boldsymbol{y}$ in the direction $\boldsymbol{u}_i$ is related only to the component of $\boldsymbol{y}$ in the direction $\boldsymbol{v}_i$.

9.4 Generalized Inverse Mapping

9.4.1 An inverse map by restricting the domain and range.

Suppose the mapping $\boldsymbol{A}$ is restricted to $\mathcal{R}(\boldsymbol{A}^t)$ as its domain and $\mathcal{R}(\boldsymbol{A})$ as its range. As these both have the same dimension the mapping is now $1:1$ and onto, that is,

$$\boldsymbol{A}\boldsymbol{u}_i \to d_i \boldsymbol{v}_i \ , \ i = 1, 2, \ldots, r,$$

and $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r\}$ is a basis of $\mathcal{R}(\boldsymbol{A}^t)$ and $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r\}$ is a basis for $\mathcal{R}(\boldsymbol{A})$. As such an inverse map exists, say $\boldsymbol{A}^r$, which is described by

$$\boldsymbol{A}^r \boldsymbol{v}_i \to \frac{1}{d_i} \boldsymbol{u}_i \ , \ i = 1, 2, \ldots, r.$$

It is easy to check that $\boldsymbol{A}^r \boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^r = \boldsymbol{I}_r$ for $\boldsymbol{A} : \mathcal{R}(\boldsymbol{A}^t) \to \mathcal{R}(\boldsymbol{A})$.

As the inverse map is linear it can be represented by an $r \times r$ matrix in bases $\boldsymbol{U}_1$ and $\boldsymbol{V}_1$ as $\text{diag} \left[d_1^{-1}, d_2^{-1}, \ldots, d_r^{-1} \right]$,

$$\boldsymbol{A}^r \boldsymbol{V}_1 = \boldsymbol{U}_1 \begin{pmatrix} d_1^{-1} & & & \\ & d_2^{-1} & \bigcirc & \\ & \bigcirc & \ddots & \\ & & & d_r^{-1} \end{pmatrix}.$$

The definition of the inverse map can be extended so that

$$\boldsymbol{A}^r \boldsymbol{v}_i \to \boldsymbol{0} \quad, \quad i = r+1, r+2, \ldots, m,$$

and extending to the basis of $\mathbb{R}^n$

$$\boldsymbol{A}^r \begin{pmatrix} \boldsymbol{V}_1 & \boldsymbol{V}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{pmatrix} \begin{pmatrix} d_1^{-1} & & & & & \\ & d_2^{-1} & & & & \\ & & \ddots & & & \bigcirc \\ & & & d_r^{-1} & & \\ & & \bigcirc & & \bigcirc \end{pmatrix}$$

$$= \begin{pmatrix} \boldsymbol{U}_1 \boldsymbol{U}_2 \end{pmatrix} \boldsymbol{D}^r \qquad \text{(say)}.$$

Hence

$$\boldsymbol{A}^r = \boldsymbol{U} \boldsymbol{D}^r \boldsymbol{V}^t,$$

and $\boldsymbol{A}^r$ maps $\mathcal{N}(\boldsymbol{A}^t)$ to the zero vector, that is

$$\mathcal{N}(\boldsymbol{A}^t) = \mathcal{N}(\boldsymbol{A}^r),$$

and

$$\mathcal{R}(\boldsymbol{A}^t) = \mathcal{R}(\boldsymbol{A}^r).$$

The mappings $\boldsymbol{A}^r$ and $\boldsymbol{A}^t$ are similar except

$$\boldsymbol{A}^t \boldsymbol{v}_i \to d_i \boldsymbol{u}_i \qquad i = 1, 2, \ldots, r,$$

and

$$\boldsymbol{A}^r \boldsymbol{v}_i \to \frac{1}{d_i} \boldsymbol{u}_i \qquad i = 1, 2, \ldots, r.$$

9.4.2 Moore-Penrose generalized inverse.

The mapping $\boldsymbol{A}^+$ from $\mathbb{R}^m$ to $\mathbb{R}^n$ is defined to be the unique solution to the equations,

$$\begin{aligned} \boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} &= \boldsymbol{A} \\ \boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^+ &= \boldsymbol{A}^+ \\ \boldsymbol{A}\boldsymbol{A}^+ &= (\boldsymbol{A}\boldsymbol{A}^+)^t \\ \boldsymbol{A}^+\boldsymbol{A} &= (\boldsymbol{A}^+\boldsymbol{A})^t. \end{aligned} \qquad (9.4.1)$$

It is easy to show by substitution that if

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{B}\boldsymbol{P}^t,$$

where $\boldsymbol{Q}$ and $\boldsymbol{P}$ are orthogonal then

$$\boldsymbol{A}^+ = \boldsymbol{P}\boldsymbol{B}^+\boldsymbol{Q}^t,$$

that is, this inverse is invariant to change of orthogonal basis. Likewise for the partitioned $m \times n$ matrix

$$\boldsymbol{A} = \begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix}$$

where B is nonsingular $(r \times r)$,

$$\boldsymbol{A}^+ = \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix}$$

where $\boldsymbol{A}^+$ is $n \times m$. Hence for the two factorizations,

SVD : $\qquad\qquad\qquad \boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^t, \qquad \boldsymbol{A}^+ = \boldsymbol{U}\boldsymbol{D}^+\boldsymbol{V}^t,$

where $\boldsymbol{D}^+ = \boldsymbol{D}^r$, and

Q–U : $\qquad\qquad \boldsymbol{A} = \boldsymbol{Q} \begin{pmatrix} L & 0 \\ 0 & 0 \end{pmatrix} \boldsymbol{P}^t, \qquad \boldsymbol{A}^+ = \boldsymbol{P} \begin{pmatrix} L^{-1} & 0 \\ 0 & 0 \end{pmatrix} \boldsymbol{Q}^t.$

Note that $\boldsymbol{A}^r = \boldsymbol{A}^+$, that is, the Moore-Penrose generalized inverse is simply the inverse mapping restricted to the appropriate subspaces and mapping $\mathcal{N}(\boldsymbol{A}^t)$ to zero. The name pseudo-inverse is also given to this inverse. Other generalized inverses of $\boldsymbol{A}$ satisfy a subset of the four defining equations (9.4.1).

9.4.3 Minimum norm least squares solution.

Define a mapping $\boldsymbol{A}^\dagger : \mathbb{R}^m \to \mathbb{R}^n$ by the rule that $\boldsymbol{y} \in \mathbb{R}^m$ maps to $\boldsymbol{x}^\dagger$ which is that $\boldsymbol{x}$ of minimum Euclidean norm of all $\boldsymbol{x}$ which minimize $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2$. It is easy to show $\boldsymbol{x}^\dagger$ exists and is unique and that the mapping is linear. Use the SVD and change bases to $\boldsymbol{U}$ and $\boldsymbol{V}$. Let $\boldsymbol{x} = \boldsymbol{U}\overline{\boldsymbol{x}}$, $\boldsymbol{y} = \boldsymbol{V}\overline{\boldsymbol{y}}$. The $\overline{\boldsymbol{x}}$ which has minimum norm ($\|\overline{\boldsymbol{x}}\| = \|\boldsymbol{x}\|$) of all $\overline{\boldsymbol{x}}$ which minimize

$$\|\boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^t\boldsymbol{U}\overline{\boldsymbol{x}} - \boldsymbol{V}\overline{\boldsymbol{y}}\|^2 = \|\boldsymbol{D}\overline{\boldsymbol{x}} - \overline{\boldsymbol{y}}\|^2$$
$$= \sum_{i=1}^{r}(d_i\overline{x}_i - \overline{y}_i)^2 + \sum_{i=r+1}^{m}\overline{y}_i^2$$

is chosen by $\overline{x}_i = \overline{y}_i/d_i$, $i = 1, 2, \ldots, r$, and $\overline{x}_{r+1}, \overline{x}_{r+2}, \ldots, \overline{x}_n$ any value. This gives the coordinates of vectors $\overline{\boldsymbol{x}}$ minimizing the sum of squares of residuals. Now minimizing $\|\boldsymbol{x}\|$ gives that

$$\overline{x}_{r+1} = \overline{x}_{r+2} = \ldots = \overline{x}_n = 0.$$

Hence

$$\overline{\boldsymbol{x}}^\dagger = \boldsymbol{D}^r\overline{\boldsymbol{y}} = \boldsymbol{D}^\dagger\overline{\boldsymbol{y}},$$

and

$$\boldsymbol{x}^\dagger = \boldsymbol{A}^\dagger\boldsymbol{y} = \boldsymbol{A}^r\boldsymbol{y} = \boldsymbol{A}^+\boldsymbol{y}.$$

So these three different approaches produce the same unique generalized inverse.

9.5 Best rank s approximation to a matrix.

Let $\boldsymbol{A}$ be an $m \times n$ matrix with SVD, $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^t$. Then of all matrices $\boldsymbol{B}$ of rank $s \leq r$, the matrix which minimizes

$$\|\boldsymbol{A} - \boldsymbol{B}\|_E = \Big\{ \sum_{i,j}(a_{ij} - b_{ij})^2 \Big\}^{1/2}$$

is

$$\boldsymbol{A}_T = \boldsymbol{V}\boldsymbol{D}_T\boldsymbol{U}^t,$$

where $\boldsymbol{D}_T$ is a diagonal matrix equal to

$$\operatorname{diag}[d_1, d_2, \ldots, d_s, 0, \ldots, 0],$$

and assuming $d_1 \geq d_2 \geq \cdots \geq d_r > 0$. See Hoffman and Wielandt (1953) for a proof. $\boldsymbol{A}_T$ is called the truncated estimate of rank s of $\boldsymbol{A}$. Note that as the Euclidean norm is invariant to multiplication by orthogonal matrices

$$\|\boldsymbol{A} - \boldsymbol{A}_T\|_E = \|\boldsymbol{D} - \boldsymbol{D}_T\|_E = \left(\sum_{i=s+1}^{r} d_i^2 \right)^{1/2}.$$

Also $\|\boldsymbol{A} - \boldsymbol{A}_T\|_2 = d_{s+1}$. This replaces the smallest singular values by zero, and this has relevance in numerical work as seldom can singular values $d_i = 0$, $i = r + 1$, $r + 2, \ldots, \min(m, n)$ be calculated accurately.

9.6 Generalized inverse and rounding error.

In computing a generalized inverse a decision has been made of when a singular value is zero, given that rounding error effects all the factors of the SVD or Q–U factorization. In linear algebra use is made of the 'backward error analysis' approach. The actual computed factors (denoted by $\overline{\cdot}$ in this section) are assigned to be the exact factors of some perturbed original matrix. So in terms of the SVD

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^t,$$

$$\overline{\boldsymbol{A}} = \overline{\boldsymbol{V}}\,\overline{\boldsymbol{D}}\,\overline{\boldsymbol{U}}^{t},$$

and a stable algorithm for calculating the SVD will mean that

$$\|\boldsymbol{A} - \overline{\boldsymbol{A}}\| \leq p(m,n)\epsilon,$$

where ϵ is the machine precision and $p(m,n)$ is a low degree polynomial in m and n. Note this does not mean that the factors $\overline{\boldsymbol{V}}, \overline{\boldsymbol{D}}, \overline{\boldsymbol{U}}$ are necessarily close, respectively to $\boldsymbol{V}$, $\boldsymbol{D}$ and $\boldsymbol{U}$.

In particular if $\overline{\boldsymbol{D}} = \text{diag}\,(\overline{d}_1, \overline{d}_2, \ldots, \overline{d}_{\ell})$, where $\ell = \min(m,n)$, some of the singular values of $\boldsymbol{A}$ which might have been zero are probably of order the machine precision ϵ in $\overline{\boldsymbol{D}}$. In order to determine the rank of $\boldsymbol{A}$ a decision about small singular values of $\overline{\boldsymbol{D}}$ needs to be made. The neatest procedure is to replace $\overline{\boldsymbol{D}}$ with its truncated estimate $\overline{\boldsymbol{D}}_T$ so that the effective matrix is

$$\overline{\boldsymbol{A}}_T = \overline{\boldsymbol{V}}\,\overline{\boldsymbol{D}}_T\overline{\boldsymbol{U}}^{t}.$$

If $\boldsymbol{A}_T$ has rank s then (approximately as $\overline{\boldsymbol{V}}$ and $\overline{\boldsymbol{U}}$ are not orthogonal)

$$\|\boldsymbol{A} - \overline{\boldsymbol{A}}_T\| \leq p(m,n)\epsilon + \overline{d}_{s+1}$$

and $\overline{\boldsymbol{A}}_T$ will still be close to $\boldsymbol{A}$. If the non zero singular values of $\boldsymbol{A}$ are well separated from zero, relative to the machine precision, so that $d_i >> \epsilon$, $i = 1, 2, \ldots, r$, then as $|(d_i - \overline{d}_i| \leq \kappa\epsilon$ it is easy to choose an s equal to the rank of $\boldsymbol{A}$, that is,

$$\overline{d}_i >> \epsilon \quad i = 1, 2, \ldots, r$$
$$\overline{d}_i \approx \epsilon \quad i = r+1, \ldots, \ell.$$

The above ideas also apply to the Q–U factorization in that the rank of the matrix has to be decided at some stage during the factorization. Column pivoting is needed during the factorization;

$$\boldsymbol{A} \longrightarrow \boldsymbol{Q} \begin{pmatrix} & \boldsymbol{U} & \\ \boldsymbol{0} & & \boldsymbol{0} \end{pmatrix} \boldsymbol{P},$$

where $\boldsymbol{P}$ represents column interchanges. In practice before the i–th elementary Hermitian stage

$$\boldsymbol{A} \longrightarrow \overline{\boldsymbol{Q}} \begin{pmatrix} & \overline{\boldsymbol{U}} & \\ \boldsymbol{0} & & \delta\overline{\boldsymbol{U}} \end{pmatrix} \overline{\boldsymbol{P}}.$$

If $\|\delta\overline{\boldsymbol{U}}\| < \kappa\epsilon$ for some constant κ, $\delta\overline{\boldsymbol{U}}$ may be considered to be entirely round off error and an effective $\boldsymbol{A}$ is

$$\overline{\boldsymbol{A}} = \overline{\boldsymbol{Q}} \begin{pmatrix} & \overline{\boldsymbol{U}} & \\ \boldsymbol{0} & & \boldsymbol{0} \end{pmatrix} \overline{\boldsymbol{P}}.$$

Column pivoting ensures that as $\boldsymbol{U}$ gains its i–th row, $|U_{ii}|$ is larger than or equal to any succeeding $|U_{jj}|$. So $|U_{11}| \geq |U_{22}| \geq \cdots \geq |U_{\ell\ell}|$. Also $|U_{ii}| \geq |U_{ij}|$, $j = i+1, \ldots, n$. Approximately (because $\overline{\boldsymbol{Q}}$ is not exactly orthogonal)

$$\|\boldsymbol{A} - \overline{\boldsymbol{A}}\| \leq \|\delta\overline{\boldsymbol{U}}\|,$$

so this procedure can be looked upon as an approximation to the truncated estimate. The Q–U factorization is faster to compute than the SVD, but this has become less of a problem with modern workstations.

9.7 Scaling considerations.

It is possible in practical situations to have a matrix made from data which differs vastly in magnitude but which is accurate to only four significant digits (not decimal places). For example in econometrics, an $m \times n$ data matrix $\boldsymbol{A}$ has n columns each of which represent an economic variable measured over m time periods. The norm of a column depends on the units of the column, for example, population measured as a single person, 10^3 people, 10^6 people or 10^9 people, will give vastly different numbers to represent the population of China. Similarly for the rows, because of inflation, if money variables are expressed in dollars, the last row is expected to be larger than the first. These two scaling problems need to be handled differently but if a meaningful rank decision is to be made about $\boldsymbol{A}$ they must be taken into account.

Taking the row problem first. Usually a least squares solution is found to a set of equations $\boldsymbol{Ax} = \boldsymbol{y} + \boldsymbol{r}$ ($m > n$), that is, the sum of squares of residuals of each time period ($\boldsymbol{r}^t\boldsymbol{r}$) is minimized. This is a valid statistical procedure if the residuals are independent and homoscedastic, that is $\mathcal{E}(\boldsymbol{rr}^t) = \sigma^2 \boldsymbol{I}_m$, where $\boldsymbol{r} = \boldsymbol{Ax} - \boldsymbol{y}$. If $\mathcal{E}(\boldsymbol{rr}^t) = \boldsymbol{V}$ (variance-covariance of residuals) then $\boldsymbol{r}^t \boldsymbol{V}^{-1} \boldsymbol{r}$ should be minimized instead. If the residuals are independent but not homoscedastic (heteroscedastic) then $\boldsymbol{V}$ is a diagonal matrix and the rows of the equation are weighted inversely with the expected standard error of the i–th residual. Hence any row scaling is usually governed by statistical considerations in this example.

The column scalings are obviously free to be chosen to make the numerical computation as stable as possible. Consider the 3×2 matrix

$$\begin{pmatrix} 1.123 & 2.345 \times 10^9 \\ 1.123 & 2.567 \times 10^9 \\ 1.123 & 2.789 \times 10^9 \end{pmatrix}$$

representing a constant column and a variable in units of dollars say. This matrix is essentially singular as one column is less than 10^{-9} times the norm of the matrix. In fact one singular value will be of order 10^9, the other of order 1. Yet both columns have just as much signficiant data in them. If another column equal to $\kappa_1(\boldsymbol{A}) + 10^{-9}\kappa_2(\boldsymbol{A})$ is added then for the 3×3 matrix the third singular value is zero. But on a 32 bit machine one singular value of order 10^9 is computed, the other two computed are of magnitude $10 - 100$, due to roundoff error. As the choice of units is arbitrary, units to make each column unity can be chosen. This essentially means that significant figure accuracy lines up with decimal place accuracy. For our scaled 3×3 matrix two singular values of magnitude 1 and one of magnitude the machine precision are computed.

In terms of the system of equations

$$\min_{\boldsymbol{x}} \| \boldsymbol{Ax} - \boldsymbol{y} \| \tag{9.7.1}$$

$$\equiv \min_{\boldsymbol{x}} \| \boldsymbol{ADD}^{-1}\boldsymbol{x} - \boldsymbol{y} \|$$

$$\equiv \min_{\widehat{\boldsymbol{x}}} \| \widehat{\boldsymbol{A}}\widehat{\boldsymbol{x}} - \boldsymbol{y} \| \quad , \quad \widehat{\boldsymbol{x}} = \boldsymbol{D}^{-1}\boldsymbol{x} \ , \ \widehat{\boldsymbol{A}} = \boldsymbol{AD}. \tag{9.7.2}$$

If $\boldsymbol{A}$ has full rank, so that $\boldsymbol{A}^+ = (\boldsymbol{A}^t\boldsymbol{A})^{-1}\boldsymbol{A}^t = \boldsymbol{D}\widehat{\boldsymbol{A}}^+$ then the same vector $\boldsymbol{x}$ is computed for both problems (9.7.1) and (9.7.2). However if $\boldsymbol{A}$ does not have rank n and the minimum norm solution of the least squares solution is calculated then in (9.7.1) $\|\boldsymbol{x}\|$ is minimized and in (9.7.2) $\|\widehat{\boldsymbol{x}}\| = \|\boldsymbol{D}^{-1}\boldsymbol{x}\|$ is minimized. In terms of the previous 3×2 example (9.7.2) is likely to be preferred anyway as it appears to give equal relative weighting to all $\widehat{x}_i$ values.

9.8 Generalized inverses under scalings.

Let $\boldsymbol{R}$ and $\boldsymbol{S}$ be symmetric positive definite matrices of sizes m and n respectively. The inner products $\langle \boldsymbol{a}, \boldsymbol{b} \rangle_{\boldsymbol{S}} = \boldsymbol{a}^t\boldsymbol{Sb}$ and hence norms $\|\boldsymbol{a}\|_{\boldsymbol{S}} = \langle \boldsymbol{a}, \boldsymbol{a} \rangle_{\boldsymbol{S}}^{1/2} = (\boldsymbol{a}^t\boldsymbol{Sa})^{1/2}$ can be used to define generalized inverses under scalings. Define a generalized inverse $\boldsymbol{A}^+$ as the mapping of $\boldsymbol{y} \in \mathbb{R}^m$ to the unique vector $\boldsymbol{x}_G$ which is the smallest vector in terms of $\| \cdot \|_{\boldsymbol{S}}$ of all vectors $\boldsymbol{x}$ which minimize $\|\boldsymbol{Ax} - \boldsymbol{y}\|_{\boldsymbol{R}}$. If $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{Ax}$, this is finding the smallest vector $\widehat{\boldsymbol{x}}_G$ in terms of $\| \cdot \|_2$ of all vectors $\widehat{\boldsymbol{x}}$ which minimize $\boldsymbol{r}^t\boldsymbol{Rr}$ where $\widehat{\boldsymbol{x}} = \boldsymbol{S}^{1/2}\boldsymbol{x}$ and $(\boldsymbol{S}^{1/2})^t\boldsymbol{S}^{1/2} = \boldsymbol{S}$.

In short, scaling is equivalent to putting different metrics on spaces for generalised inverse computations. These different metrics can be more appropriate than a user's arbitrary choice of units. Computation within statistical or econometric packages should be done in terms of the scaled variables.

9.9 Approximations to the generalized inverse.

9.9.1 Limiting approximations.

If $\boldsymbol{A}$ is $m \times n$, $m > n$ of rank n (full rank) then

$$\boldsymbol{A}^+ = (\boldsymbol{A}^t\boldsymbol{A})^{-1}\boldsymbol{A}^t,$$

and if $m < n$, rank m

$$\boldsymbol{A}^+ = \boldsymbol{A}^t(\boldsymbol{AA}^t)^{-1}.$$

In the case of not full rank, $r < \min(m, n)$, we have

$$\boldsymbol{A}^+ = \lim_{\epsilon \to 0}(\boldsymbol{A}^t\boldsymbol{A} + \epsilon^2\boldsymbol{I})^{-1}\boldsymbol{A}^t = \lim_{\epsilon \to 0}\boldsymbol{A}^t(\boldsymbol{A}\boldsymbol{A}^t + \epsilon^2\boldsymbol{I})^{-1},$$

which is easily proved using the SVD of $\boldsymbol{A}$ and noting that

$$\lim_{\epsilon \to 0}\frac{d_i}{d_i^2 + \epsilon^2} = d_i^+ = \begin{cases} d_i^{-1}, & d_i \neq 0, \\ 0, & d_i = 0. \end{cases}$$

Of course in numerical computation $\epsilon \to 0$ is not possible as an increasingly ill-conditioned inversion needs to be computed. A similar effect to the truncated estimate generalized inverse is preferred. Note that the singular values of $(\boldsymbol{A}^t\boldsymbol{A} + \epsilon^2\boldsymbol{I})\boldsymbol{A}^t$ are

$$\frac{d_i}{d_i^2 + \epsilon^2} \approx \begin{cases} d_i^{-1} & \text{if } d_i >> \epsilon, \\ 0 & \text{if } d_i << \epsilon. \end{cases}$$

Hence by keeping ϵ to be of magnitude of the cut off for the truncated estimate (perhaps a bit smaller) a similar effect is achieved.

9.9.2 Damped least squares.

Consider the problem

$$\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \epsilon^2\|\boldsymbol{x}\|_2^2,$$

$$\equiv \min_{\boldsymbol{x}} \| \begin{pmatrix} \boldsymbol{A} \\ \epsilon\boldsymbol{I} \end{pmatrix} \boldsymbol{x} - \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix} \|_2^2.$$

As $\begin{pmatrix} \boldsymbol{A} \\ \epsilon\boldsymbol{I} \end{pmatrix}$ has full rank so the solution is given by,

$$\left(\begin{pmatrix} \boldsymbol{A}^t & \epsilon\boldsymbol{I} \end{pmatrix}\begin{pmatrix} \boldsymbol{A} \\ \epsilon\boldsymbol{I} \end{pmatrix}\right)^{-1}\begin{pmatrix} \boldsymbol{A}^t & \epsilon\boldsymbol{I} \end{pmatrix}\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix} = (\boldsymbol{A}^t\boldsymbol{A} + \epsilon^2\boldsymbol{I})^{-1}\boldsymbol{A}^t\boldsymbol{y}.$$

A way to compute this solution is to use the Q–U factorization on $\begin{pmatrix} \boldsymbol{A} \\ \epsilon\boldsymbol{I} \end{pmatrix}$.

9.9.3 Underdetermined case.

Consider the system $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{y}$, $\boldsymbol{A}$ is $m \times n$, $m < n$, and the augmented system

$$\begin{pmatrix} \boldsymbol{A} & \epsilon\boldsymbol{I} \end{pmatrix}\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{x}_r \end{pmatrix} = \boldsymbol{A}\boldsymbol{x} + \epsilon\boldsymbol{x}_r = \boldsymbol{y},$$

which has full rank, so that

$$\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{x}_r \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}^t \\ \epsilon\boldsymbol{I} \end{pmatrix}(\boldsymbol{A}\boldsymbol{A}^t + \epsilon^2\boldsymbol{I})^{-1}\boldsymbol{y},$$

or

$$\boldsymbol{x} = \boldsymbol{A}^t(\boldsymbol{A}\boldsymbol{A}^t + \epsilon^2\boldsymbol{I})^{-1}\boldsymbol{y},$$

and the residual is given by $\epsilon\boldsymbol{x}_r = \epsilon^2(\boldsymbol{A}\boldsymbol{A}^t + \epsilon^2\boldsymbol{I})^{-1}\boldsymbol{y}$.

9.9.4 Rutishauser's doubly relaxed least squares.

See Rutishauser (1968) for the details of this approximation.

$$\boldsymbol{A}^+ = \lim_{\epsilon \to 0}\{(\boldsymbol{A}^t\boldsymbol{A} + \epsilon^2\boldsymbol{I}) + \epsilon^2(\boldsymbol{A}^t\boldsymbol{A} + \epsilon^2\boldsymbol{I})^{-1}\}^{-1}\boldsymbol{A}^t.$$

9.10 Projection operators.

9.10.1 Projection onto a subspace.

Let $\mathcal{X}$ be a subspace of $\mathbb{R}^m$. $\boldsymbol{P}_\mathcal{X}$ is a projection operator from $\mathbb{R}^m$ to $\mathcal{X}$ if

$$\mathcal{R}(\boldsymbol{P}_\mathcal{X}) = \mathcal{X},$$

and, if $\boldsymbol{x} \in \mathfrak{X}$, then $\boldsymbol{P}_{\mathfrak{X}}\boldsymbol{x} = \boldsymbol{x}$. So if $\boldsymbol{y} \in \mathbb{R}^m$ and $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{z}$, $\boldsymbol{x} \in \mathfrak{X}$, $\boldsymbol{z} \in \mathfrak{X}^\perp$ then

$$\boldsymbol{P}_{\mathfrak{X}}\boldsymbol{y} = \boldsymbol{x}.$$

9.10.2 Spaces represented by matrices.

Let $\mathfrak{X}$ be the space of the span of the columns of an $m \times n$ matrix $\boldsymbol{A}$, that is, $\mathfrak{X} = \mathcal{R}(\boldsymbol{A}) \subset \mathbb{R}^m$. Then $\mathcal{R}(\boldsymbol{A})^\perp = \mathcal{N}(\boldsymbol{A}^t)$, and any vector in $\mathbb{R}^m$ can be expressed as the sum of a component in $\mathcal{R}(\boldsymbol{A})$ and a component in $\mathcal{N}(\boldsymbol{A}^t)$. $\boldsymbol{P}_A = \boldsymbol{P}_{\mathfrak{X}}$ has to choose the component in $\mathcal{R}(\boldsymbol{A})$. So

$$\boldsymbol{P}_A \kappa_j(\boldsymbol{A}) = \kappa_j(\boldsymbol{A}) \quad , \quad j = 1, 2, \ldots, n,$$

or

$$\boldsymbol{P}_A \boldsymbol{A} = \boldsymbol{A}. \tag{9.10.1}$$

This means that $\boldsymbol{P}_A$ must be represented by an $m \times m$ matrix, with each column a linear combination of columns of $\boldsymbol{A}$. Hence $\boldsymbol{P}_A = \boldsymbol{A}\boldsymbol{B}$ for some $n \times m$ matrix $\boldsymbol{B}$. So from (9.10.1)

$$\boldsymbol{A}\boldsymbol{B}\boldsymbol{A} = \boldsymbol{A},$$

and a choice for $\boldsymbol{B}$ is $\boldsymbol{A}^+$, that is

$$\boldsymbol{P}_A = \boldsymbol{A}\boldsymbol{A}^+.$$

In the full rank cases, if $\boldsymbol{A}$ has rank n then $\boldsymbol{P}_A = \boldsymbol{A}(\boldsymbol{A}^t\boldsymbol{A})^{-1}\boldsymbol{A}^t$, while if the rank is m, then $\boldsymbol{P}_A = \boldsymbol{A}\boldsymbol{A}^t(\boldsymbol{A}\boldsymbol{A}^t)^{-1} = \boldsymbol{I}$.

It is easy to show that

$$\boldsymbol{P}_{A^\perp} = \boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^+$$

is the projection onto $\mathcal{R}(\boldsymbol{A})^\perp = \mathcal{N}(\boldsymbol{A}^t)$. There is no need to find $\boldsymbol{A}^+$ in order to compute $\boldsymbol{P}_A$ or $P_{A^\perp}$, as the orthogonal factorizations give relevant expressions in terms of parts of orthogonal matrices. The actual rank of $\boldsymbol{A}$ needs to be determined.

9.10.3 Projections in terms of the SVD.

Let $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^t$, be of rank r. Then $\boldsymbol{v}_1\boldsymbol{v}_2, \ldots, \boldsymbol{v}_r$ is an orthogonal basis for $\mathcal{R}(\boldsymbol{A})$. So if $\boldsymbol{y} \in \mathbb{R}^m$,

$$\boldsymbol{y} = \sum_{i=1}^m \langle \boldsymbol{v}_i, \boldsymbol{y} \rangle \boldsymbol{v}_i,$$

$$\boldsymbol{P}_A \boldsymbol{y} = \sum_{i=1}^r \langle \boldsymbol{v}_i, \boldsymbol{y}_i \rangle \boldsymbol{v}_i,$$

and

$$\boldsymbol{P}_{A^\perp} \boldsymbol{y} = \sum_{r+1}^m \langle \boldsymbol{v}_i, \boldsymbol{y}_i \rangle \boldsymbol{v}_i.$$

Writing $\boldsymbol{V} = (\ \boldsymbol{V}_1 \quad \boldsymbol{V}_2\ )$, where $\boldsymbol{V}_1$ is $m \times r$,

$$\boldsymbol{y} = \boldsymbol{V}\overline{\boldsymbol{y}} \quad , \quad \overline{\boldsymbol{y}} = \boldsymbol{V}^t \boldsymbol{y}$$

and

$$\begin{aligned} \boldsymbol{P}_A \boldsymbol{y} &= \boldsymbol{V}_1 (\ \boldsymbol{I}_r \quad \boldsymbol{0}\ ) \overline{\boldsymbol{y}} \\ &= \boldsymbol{V}_1 (\ \boldsymbol{I}_r \quad \boldsymbol{0}\ ) \boldsymbol{V}^t \boldsymbol{y} \\ &= \boldsymbol{V}_1 \boldsymbol{V}_1^t \boldsymbol{y} \quad \text{or} \quad \boldsymbol{V} \begin{pmatrix} \boldsymbol{I}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \boldsymbol{V}^t \boldsymbol{y}, \end{aligned}$$

and similarly

$$\boldsymbol{P}_{A^\perp} \boldsymbol{y} = \boldsymbol{V}_2 \boldsymbol{V}_2^t \boldsymbol{y} = \boldsymbol{V} \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{m-r} \end{pmatrix} \boldsymbol{V}^t \boldsymbol{y}.$$

If bases are changed to the columns of V then these two projection operators are now represented by

$$\begin{pmatrix} \boldsymbol{I}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{m-r} \end{pmatrix}.$$

Substitution of the SVD of $\boldsymbol{A}$ and $\boldsymbol{A}^+$ into $\boldsymbol{A}\boldsymbol{A}^+$ and $\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^+$ will also yield the above formulae. See Jennings (1980) for an application in econometrics.

9.11 Exercises

1. Find projection operators $\boldsymbol{A}\boldsymbol{A}^+$ and $\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^+$ in terms of the Q–U factorization.

2. Prove the Rutishauser formula and compare its approximation to the truncated estimate generalized inverse.

3. Show $\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^t = \boldsymbol{A}^t$, and $(\boldsymbol{A}^+)^t = (\boldsymbol{A}^t)^+$.

4. Show that a projection matrix $\boldsymbol{P}$ is idempotent, that is, $\boldsymbol{P}^2 = \boldsymbol{P}$.

5. Using the notation of §9.8, show that

$$\boldsymbol{x}_G = \lim_{\epsilon \to 0}(\boldsymbol{A}^t\boldsymbol{R}\boldsymbol{A} + \epsilon^2\boldsymbol{S})^{-1}\boldsymbol{A}^t\boldsymbol{R}\boldsymbol{y},$$

which is the solution of the generalized damped least squares problem

$$\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_R^2 + \epsilon^2|\boldsymbol{x}\|_S.$$

10. Eigenvalue Computations – An Overview.

10.1 Introduction.

In general $\boldsymbol{M}(\lambda)$ an $n \times n$ matrix being a function of one parameter λ. The objective is to find values of λ such that

$$\det |\boldsymbol{M}(\lambda)| = 0.$$

At these points a representation of $\mathcal{N}(\boldsymbol{M}(\lambda))$ is required, that is, non-zero $\boldsymbol{v}$ such that

$$\boldsymbol{M}(\lambda)\boldsymbol{v} = \boldsymbol{0}.$$

Classical linear case.

Here $\boldsymbol{M}(\lambda) = \boldsymbol{A} - \lambda \boldsymbol{I}$, where $\boldsymbol{A}$ is an $n \times n$ matrix. The problem is stated as: find λ such that $\det |\boldsymbol{A} - \lambda \boldsymbol{I}| = 0$, and non-zero vector $\boldsymbol{v}$ such that $\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}$. The scalar λ is known as the eigenvalue and $\boldsymbol{v}$ its corresponding eigenvector. As $\det |\boldsymbol{A} - \lambda \boldsymbol{I}|$ is a polynomial in λ of degree n there exists n linear factors over $\mathbb{C}$. For each of these λ_i a non-zero $\boldsymbol{v}_i$ exists. A matrix $\boldsymbol{B}$ is similar to $\boldsymbol{A}$ if there exists a nonsingular matrix $\boldsymbol{T}$ such that

$$\boldsymbol{B} = \boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1}.$$

In this case $\boldsymbol{A}$ and $\boldsymbol{B}$ have the same eigenvalues as $\det |\boldsymbol{A} - \lambda \boldsymbol{I}| = \det |\boldsymbol{B} - \lambda \boldsymbol{I}|$. If $\boldsymbol{v}$ is an eigenvector of $\boldsymbol{A}$ then $\boldsymbol{T}\boldsymbol{v}$ is an eigenvector of $\boldsymbol{B}$.

Jordan Canonical Form.

Theorem: Every $n \times n$ matrix $\boldsymbol{A}$ is similar to a matrix of the form

$$\boldsymbol{J} = \begin{pmatrix} \boldsymbol{J}_1 & & & \\ & \boldsymbol{J}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{J}_r \end{pmatrix},$$

where each $\boldsymbol{J}_i$ has one of the forms λ_i, $\begin{pmatrix} \lambda_i & 1 \\ 0 & \lambda_i \end{pmatrix}$, $\begin{pmatrix} \lambda_i & 1 & 0 \\ 0 & \lambda_i & 1 \\ 0 & 0 & \lambda_i \end{pmatrix}$, etc. There is only one null space vector of the sub-matrix $(\boldsymbol{J}_r - \lambda_r \boldsymbol{I})$, even though the eigenvalue is a repeated root if $\boldsymbol{J}_r$ has order ≥ 2. The invertible matrix $\boldsymbol{T}$ where $\boldsymbol{A} = \boldsymbol{T}\boldsymbol{J}\boldsymbol{T}^{-1}$ still needs n linearly independent columns, even though only r of them are eigenvectors. See Lipschutz (1968) for the detail. ∎

10.2 Classes of algorithm.

10.2.1 Eigenvalues: Classes of methods.

(i) Methods which transform $\boldsymbol{A}$ to $\boldsymbol{B}$ in a finite number of elementary similarity transformations so that the polynomial

$$\det |\boldsymbol{B} - \lambda \boldsymbol{I}|$$

is readily evaluated and hence its zeros found using a zero finding technique, form the first class. These are good methods for finding one or two eigenvalues. Their efficiency is not good for finding all the eigenvalues.

(ii) The second class of methods are those which transform $\boldsymbol{A}$ to a form where the eigenvalues can be read off, for example diagonal or upper triangular.

(iii) There are some methods distinct from the above used mainly in the nonlinear eigenvalue case, but which can be used in the linear case to improve the accuracy of estimates of λ and $\boldsymbol{v}$.

10.2.2 Eigenvectors.

In most cases the method of *inverse iteration* is used to find eigenvectors. Usually some accumulation of a finite sequence of similarity transformations is necessary for efficiency.

10.3 Classes of matrices.

In order of difficulty of computation

 (i) Complex general: The eigenvalues and eigenvectors are in general complex and n linearly independent eigenvectors may not exist if eigenvalues are repeated.

 (ii) Real general: The eigenvalues are either real or in complex conjugate pairs as are the eigenvectors and n linearly independent eigenvectors may not exist if eigenvalues are repeated.

(iii) Hessenberg matrices: Matrices in the above two classes are similarity transformed to a real or complex Hessenberg matrix as part of the algorithm for finding eigenvalues and eigenvectors.

(iv) Hermitian complex: The eigenvalues are real and the eigenvectors can be complex but can be chosen and scaled so that they form an orthonormal basis for $\mathbb{C}^n$.

 (v) Hermitian real (real symmetric): The eigenvalues are real and the eigenvectors can be chosen and scaled to form an orthonormal basis for $\mathbb{R}^n$.

(vi) Symmetric banded: Special algorithms can be used to similarity transform these matrices to the next class.

(vii) Symmetric tridiagonal: These are Hessenberg matrices and are formed as part of algorithms for computing eigenvalues and eigenvectors of classes (iv), (v) and (vi).

10.4 Real symmetric matrices (Hermitian matrices).

We will only discuss the real case. A recent history and summary can be found in Parlett (1981).

10.4.1 Jacobi method.

This is an old method not in use today but it is an example of the effectiveness of elementary plane rotations. Essentially the matrix $\boldsymbol{A}$ is transformed to diagonal form by an (infinite) sequence of similarity transformations.

$$\ldots \boldsymbol{Q}_k \ldots \boldsymbol{Q}_2 \boldsymbol{Q}_1 \boldsymbol{A} \boldsymbol{Q}_1^t \boldsymbol{Q}_2^t \ldots \boldsymbol{Q}_k \ldots = \boldsymbol{\Lambda}.$$

$$\boldsymbol{Q} \boldsymbol{A} \boldsymbol{Q}^t = \boldsymbol{\Lambda}.$$

 (i) Basic algorithm.

   1. Find the largest off-diagonal element, suppose it is (i, j).

   2. Apply a similarity transformation (plane rotation) to zero the element (i, j) (and (j, i) as well since $\boldsymbol{A}$ is symmetric).

$$\overline{\boldsymbol{A}} = \boldsymbol{P} \boldsymbol{A} \boldsymbol{P}^t,$$

$$\overline{A}_{ij} = 0 \Rightarrow (\boldsymbol{P} \boldsymbol{A} \boldsymbol{P}^t)_{ij} = 0.$$

$$\rho_i(\boldsymbol{P} \boldsymbol{A}) = c\rho_i(\boldsymbol{A}) + s\rho_j(\boldsymbol{A})$$
$$\rho_j(\boldsymbol{P} \boldsymbol{A}) = s\rho_i(\boldsymbol{A}) - c\rho_j(\boldsymbol{A}).$$

$$\kappa_i(\boldsymbol{P} \boldsymbol{A} \boldsymbol{P}^t) = c\kappa_i(\boldsymbol{P} \boldsymbol{A}) + s\kappa_j(\boldsymbol{P} \boldsymbol{A})$$
$$\kappa_j(\boldsymbol{P} \boldsymbol{A} \boldsymbol{P}^t) = s\kappa_i(\boldsymbol{P} \boldsymbol{A}) - c\kappa_j(\boldsymbol{P} \boldsymbol{A}).$$

$$\therefore \qquad \overline{A}_{ii} = c^2 A_{ii} + 2sc A_{ij} + s^2 A_{jj}$$
$$\overline{A}_{jj} = c^2 A_{jj} - 2sc A_{ij} + s^2 A_{ii}$$
$$\overline{A}_{ij} = (s^2 - c^2) A_{ij} + sc(A_{ii} - A_{jj}) = 0.$$

$$\therefore \qquad \frac{2sc}{c^2 - s^2} = \frac{2A_{ij}}{A_{ii} - A_{jj}}.$$

   If $c = \cos\theta$, $s = \sin\theta$,

$$\tan 2\theta = \frac{2A_{ij}}{A_{ii} - A_{jj}}.$$

   Calculation of s and c has to be done carefully to avoid subtractive cancellation. See Boothroyd (1968).

   3. If the sum of squares of off-diagonal elements $< \epsilon$ then stop, else goto 1.

 (ii) Convergence:

Theorem: Let $\boldsymbol{A}^{(k)} = \boldsymbol{P}\boldsymbol{A}^{(k-1)}\boldsymbol{P}^t$ be the sequence of plane rotations ($\boldsymbol{A}^0 = \boldsymbol{A}$), then if E_k is the sum of squares of off-diagonal elements of $\boldsymbol{A}^{(k)}$ then $E_k \to 0$ as $k \to \infty$.

Proof:

$\|\boldsymbol{A}^{(k)}\|_E = \|\boldsymbol{A}^{(k-1)}\|_E = \|\boldsymbol{A}\|_E$. The sum of squares of the diagonal elements is shown to be monotone increasing and as it is bounded above by $\|\boldsymbol{A}\|_E$, converges to $\|\boldsymbol{A}\|_E$. This of course implies that $E_k \to 0$. Note

$$a^2 + b^2 = \frac{1}{2}\{(a+b)^2 + (a-b)^2\}. \tag{10.4.1}$$

Denote $\overline{A}_{ij} = (\boldsymbol{A}^{(k)})_{ij}, \ A_{ij} = (\boldsymbol{A}^{(k-1)})_{ij}$

$$\overline{A}_{ii} + \overline{A}_{jj} = A_{ii} + A_{jj}$$
$$\overline{A}_{ii} - \overline{A}_{jj} = (c^2 - s^2)(A_{ii} - A_{jj}) + 4csA_{ij}$$
$$= (A_{ii} - A_{jj})\sec 2\theta.$$

$$\overline{A}_{ii}^2 + \overline{A}_{jj}^2 = \frac{1}{2}\{(A_{ii} + A_{jj})^2 + (A_{ii} - A_{jj})^2 \sec^2 2\theta\} \qquad \text{from(10.4.1)}$$

$$= A_{ii}^2 + A_{jj}^2 + \frac{1}{2}(A_{ii} - A_{jj})^2(\sec^2 2\theta - 1) \qquad \text{from(10.4.1)}$$

$$= A_{ii}^2 + A_{jj}^2 + 2A_{ij}^2$$

Hence $E_k = E_{k-1} - 2(A_{ij}^{(k-1)})^2$, which implies $E_k \to 0$ as $k \to \infty$. ∎

(iii) Rate of convergence:

$$E_k = E_{k-1} - 2(A_{ij}^{(k-1)})^2$$
$$\leq \left(1 - \frac{2}{n(n-1)}\right)E_{k-1}, \ \text{as } E_{k-1} \leq n(n-1)(A_{ij}^{(k-1)})^2.$$

Let $N = n(n-1)/2$

$$E_{k+N} \leq \left(1 - \frac{1}{N}\right)^N E_k$$
$$\approx e^{-1}E_k.$$

Stronger result: if all eigenvalues are strongly separated ($|\lambda_i - \lambda_j| > \delta$) then over every N iterations quadratic convergence is achieved, so that

$$E_{k+N} \leq KE_k^2.$$

(iv) Algorithm in practice.

Sequential algorithm — zero elements in turn with no searching for the maximum off-diagonal element. One pass through the off-diagonal elements is known as a sweep.

Threshold algorithm — sweeps are performed but miss all elements less than some tolerance. Once all off diagonals are less then the tolerance decrease it and begin again.

1. $tol = 0.1$
2. Sweep until all off-diagonals are $< tol$.
3. $tol = tol^2$
   if $tol >$ machine precision goto 2
   else stop.

(v) The eigenvectors are found by storing the plane rotations on an identity matrix.

10.4.2 Reduction to tridiagonal form.

(i) Givens method – using plane rotations.
   For $i = 1$ to $n-2$ do
      For $j = i+2$ to n do
      plane rotate rows and columns $i+1$ and j to make $A_{ji} = A_{ij} = 0$;

As the zeros are introduced past zeros are not changed. This takes $\frac{8}{3}n^3$ multiplications and $O(n^2)$ square roots.

(ii) Householder method — using Householder transformations.

For $i = 1$ to $n - 2$ do

transform row and column i into their $(i + 1)$–th element, zeroing elements $i + 2$ to n.

This takes $\frac{4}{3}n^3$ operations and $n - 2$ square roots.

10.5 Tridiagonal matrices.

10.5.1 Non-symmetric matrices.

Under certain conditions a non-symmetric tridiagonal matrix can be made diagonally similar to a symmetric tridiagonal matrix.

$$
\boldsymbol{A} = \begin{pmatrix}
\alpha_1 & \beta_2 & & & \\
\gamma_2 & \alpha_2 & \beta_3 & & \\
& \ddots & \ddots & \ddots & \\
& & \ddots & \ddots & \beta_n \\
& & & \gamma_n & \alpha_n
\end{pmatrix}
$$

Choose $\boldsymbol{D} = $ diagonal such that

$$
\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{D} = \boldsymbol{T},
$$

where $\boldsymbol{T}$ is symmetric tridiagonal. The condition is that $\gamma_i \beta_i > 0, \ i = 2, 3, \ldots, n$.

10.5.2 Symmetric tridiagonal matrices.

It is an easy matter to evaluate the determinant of a symmetric tridiagonal matrix,

$$
\boldsymbol{A} = \begin{pmatrix}
\alpha_1 & \beta_2 & & & \\
\beta_2 & \alpha_2 & \beta_3 & & \\
& \ddots & \ddots & \ddots & \\
& & \ddots & \ddots & \beta_n \\
& & & \beta_n & \alpha_n
\end{pmatrix}.
$$

Assume no $\beta_i = 0$, as the matrix could be partitioned if $\beta_i = 0$ for some i. Consider $P_n(\lambda) = \det |\boldsymbol{A} - \lambda \boldsymbol{I}|$. Expand $\det |\boldsymbol{A} - \lambda \boldsymbol{I}|$ about the last row. Let $P_i(\lambda)$ be the determinant of the i–th principle minor of $(\boldsymbol{A} - \lambda \boldsymbol{I})$,

$$
P_n(\lambda) = (\alpha_n - \lambda)P_{n-1}(\lambda) - \beta_n^2 P_{n-2}(\lambda).
$$

Similarly if $P_0(\lambda) = 1$, $P_{-1}(\lambda) = 0$, define

$$
P_i(\lambda) = (\alpha_i - \lambda)P_{i-1}(\lambda) - \beta_i^2 P_{i-2}(\lambda), \ i = 1, 2, \ldots, n.
$$

Using this recursion formulae to evaluate $P_n(\lambda)$ any root finding procedure could be used to find eigenvalues.

10.5.3 Sturm sequence property

This property is used to generate a bisection method of zero finding.

Lemma: Let the quantities $P_0(x), \ldots, P_n(x)$ be evaluated at x. If $s(x)$ is the number of agreements in sign of consecutive members of the sequence, then $s(x)$ is also the number of eigenvalues of $\boldsymbol{A}$ which are strictly greater than x.

Proof: by induction, see Wilkinson (1965) p300, or Froberg (1965) p115, or Ralston (1965) p494.   ∎

Numerical stability: Wilkinson (1965) p302 – zeros found using this technique are the zeros of a matrix $\boldsymbol{A} + \boldsymbol{E}$ where $\boldsymbol{E}$ is of order the machine precision.

10.6 Inverse iteration.

This is a very effective method of finding an eigenvector given a good estimate of the eigenvalue.

10.6.1 Algorithm

Suppose λ approximates λ_i.

(i) Solve $(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{v}_1 = \boldsymbol{b}$, $\boldsymbol{b}$ is chosen as $\boldsymbol{e}$ say.

(ii) Solve $(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{v}_2 = \boldsymbol{v}_1/\|\boldsymbol{v}_1\|$.

The normalized vector $\boldsymbol{v}_2/\|\boldsymbol{v}_2\|$ is a good estimate of the eigenvector corresponding to λ_i. Note that a system which is 'singular' is solved, but of course with roundoff errors there is a low probability $(\boldsymbol{A} - \lambda\boldsymbol{I})$ is exactly singular numerically. If this happens, replace the zero element with the machine precision and continue the division.

10.6.2 Analysis of algorithm:

Suppose n linearly independent eigenvectors $\boldsymbol{x}_i$ exist. Let $\boldsymbol{X} = [\boldsymbol{x}_1 \mid \boldsymbol{x}_2 \mid \ldots \mid \boldsymbol{x}_n]$, then $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}$. Write $\boldsymbol{b} = \sum_{i=1}^{n} c_i\boldsymbol{x}_i = \boldsymbol{X}\boldsymbol{c}$. Consider

$$(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{v}_1 = \boldsymbol{b}$$
$$\boldsymbol{v}_1 = (\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1} - \lambda\boldsymbol{I})^{-1}\boldsymbol{b}$$
$$= \boldsymbol{X}(\boldsymbol{\Lambda} - \lambda\boldsymbol{I})^{-1}\boldsymbol{c}$$
$$= \sum_{i=1}^{n} \frac{c_i}{\lambda_i - \lambda}\boldsymbol{x}_i.$$

Now let λ be close to λ_1 say, and partition the sum

$$\boldsymbol{v}_1 = \frac{c_1}{\lambda_1 - \lambda}\boldsymbol{x}_1 + \sum_{i=2}^{n} \frac{c_i}{\lambda_i - \lambda}\boldsymbol{x}_i,$$

and if λ is distant (relatively) from the other λ_i and if $c_1 \neq 0$ or not small, the number $c_1/(\lambda_1 - \lambda)$ will be much larger than the rest of the $c_i/(\lambda_i - \lambda)$. (For example $\lambda_1 - \lambda \approx 10^{-13}$, $\lambda_i - \lambda \approx 1$ for $i = 2, \ldots, n$, and all $c_i \approx 1.0$) On step (ii) we have (without normalization)

$$\boldsymbol{v}_2 \propto \frac{c_1}{(\lambda_1 - \lambda)^2}\boldsymbol{x}_1 + \sum_{i=2}^{n} \frac{c_i}{(\lambda_i - \lambda)^2}\boldsymbol{x}_i,$$

so that the major component of $\boldsymbol{v}_2$ is $\boldsymbol{x}_1$. This algorithm is usually applied when $(\boldsymbol{A} - \lambda\boldsymbol{I})^{-1}\boldsymbol{b}$ is easily calculated (in $\leq \mathcal{O}(n^2)$ operations), say when $\boldsymbol{A}$ is upper Hessenberg or upper triangular. This can also be used to improve the estimate of the eigenvalue, see §10.10.4.

10.7 Reduction to upper triangular form.

10.7.1 Theorem:

Every $n \times n$ matrix $\boldsymbol{A}$ is unitary similar to an upper triangular matrix

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{T}\boldsymbol{U}^*, \ \boldsymbol{U} \text{ unitary}, \ \boldsymbol{T} \text{ upper triangular.}$$

Proof: See Wilkinson (1965).   ■

The eigenvalues of $\boldsymbol{T}$, namely T_{ii} are those of $\boldsymbol{A}$, while the eigenvectors of $\boldsymbol{T}$ are related to those of $\boldsymbol{A}$ via $\boldsymbol{U}$. The above computation is an infinite process.

10.7.2 Similarity reduction to upper Hessenberg form.

Algorithm:

For $i = 1$ to $n - 2$ do

Householder transform elements in column i below $(i + 1)$–th row into element $(i + 1, i)$.

This is the same algorithm to reduce a symmetric matrix to symmetric tridiagonal, so it could be done with plane rotations as well. It is a finite procedure.

10.7.3 The L–R algorithm

(i) $\boldsymbol{A}^{(0)} = \boldsymbol{A}$

(ii) $\boldsymbol{A}^{(k)} \to \boldsymbol{L}^{(k)} \boldsymbol{R}^{(k)}$ (L–U factorization)

(iii) $\boldsymbol{A}^{(k+1)} \leftarrow \boldsymbol{R}^{(k)} \boldsymbol{L}^{(k)}$

Note: $\boldsymbol{A}^{(k+1)} = \boldsymbol{L}^{(k)-1} \boldsymbol{A}^{(k)} \boldsymbol{L}^{(k)}$ so $\boldsymbol{A}^{(k+1)}$ is similar to $\boldsymbol{A}^{(k)}$. Under certain conditions on the separation of eigenvalues, $\boldsymbol{A}^{(k)}$ converges to an upper triangular matrix (or block upper triangular in the case of complex roots). The matrix $\boldsymbol{L}^{(k)}$ is the product of $n-1$ elementary matrices so that if $\boldsymbol{A}^{(k)}$ is upper Hessenberg each factorization can be done in $O(n^2)$ operations, and $\boldsymbol{A}^{(k+1)}$ is upper Hessenberg. See Rutishauser (1958) for the original, and Rutishauser and Schwarz (1963) for the symmetric case.

10.7.4 Q–R algorithm

This is more stable as it uses the Q–U factorization. See Francis (1961) for the original.

(i) $\boldsymbol{A}^{(0)} = \boldsymbol{A}$

(ii) $\boldsymbol{A}^{(k)} \to \boldsymbol{Q}^{(k)} \boldsymbol{R}^{(k)}$

(iii) $\boldsymbol{A}^{(k+1)} \to \boldsymbol{R}^{(k)} \boldsymbol{Q}^{(k)}. \, (\boldsymbol{A}^{(k+1)} = \boldsymbol{Q}^{(k)^t} \boldsymbol{A}^{(k)} \boldsymbol{Q}^{(k)})$

This gives slightly faster convergence to upper triangular form. See Parlett (1968) for the details. Check that if $\boldsymbol{A}^{(k)}$ is upper Hessenberg then so too is $\boldsymbol{A}^{(k+1)}$. If $\boldsymbol{A}^{(k)}$ is upper Hessenberg the matrix $\boldsymbol{Q}^{(k)}$ is the product of $n-1$ plane rotations so the factorization can be done in $O(n^2)$ operations.

10.7.5 Single shifts— give faster convergence.

(i) $\boldsymbol{A}^{(0)} = \boldsymbol{A}$

(ii) $(\boldsymbol{A}^{(k)} - \lambda_k \boldsymbol{I}) \to \boldsymbol{Q}^{(k)} \boldsymbol{R}^{(k)}$

(iii) $\boldsymbol{A}^{(k+1)} \leftarrow \boldsymbol{R}^{(k)} \boldsymbol{Q}^{(k)} + \lambda_k \boldsymbol{I}, \, (\boldsymbol{A}^{(k+1)} = \boldsymbol{Q}^{(k)t} \boldsymbol{A}^{(k)} \boldsymbol{Q}^{(k)}).$

At each iteration λ_k is chosen as an eigenvalue of the lower right 2×2 submatrix of $\boldsymbol{A}^{(k)}$. The iteration produces convergence of the (n, n) element first, which once converged, (the $(n, n-1)$ element of the upper Hessenberg is close to zero), enables the algorithm to proceed on the remaining order $n-1$ top left submatrix.

10.7.6 Double Shifts

This method gives even faster convergence and allows the implicit use of complex arithmetic for complex conjugate eigenvalues of real matrices.

(i) $\boldsymbol{A}^{(0)} = \boldsymbol{A}$

(ii) $(\boldsymbol{A}^{(k)} - \lambda_k \boldsymbol{I}) \to \boldsymbol{Q}^{(k)} \boldsymbol{R}^{(k)}$

(iii) $\boldsymbol{A}^{(k+\frac{1}{2})} \leftarrow \boldsymbol{R}^{(k)} \boldsymbol{Q}^{(k)} + \lambda_k \boldsymbol{I}$

(iv) $(\boldsymbol{A}^{(k+\frac{1}{2})} - \overline{\lambda}_k \boldsymbol{I}) \to \boldsymbol{Q}^{(k+\frac{1}{2})} \boldsymbol{R}^{(k+\frac{1}{2})}$

(v) $\boldsymbol{A}^{(k+1)} \leftarrow \boldsymbol{R}^{(k+\frac{1}{2})} \boldsymbol{Q}^{(k+\frac{1}{2})} + \overline{\lambda}_k \boldsymbol{I}$

At each iteration λ_k and $\overline{\lambda}_k$ are the two eigenvalues of the lower right 2×2 submatrix of $\boldsymbol{A}^{(k)}$.
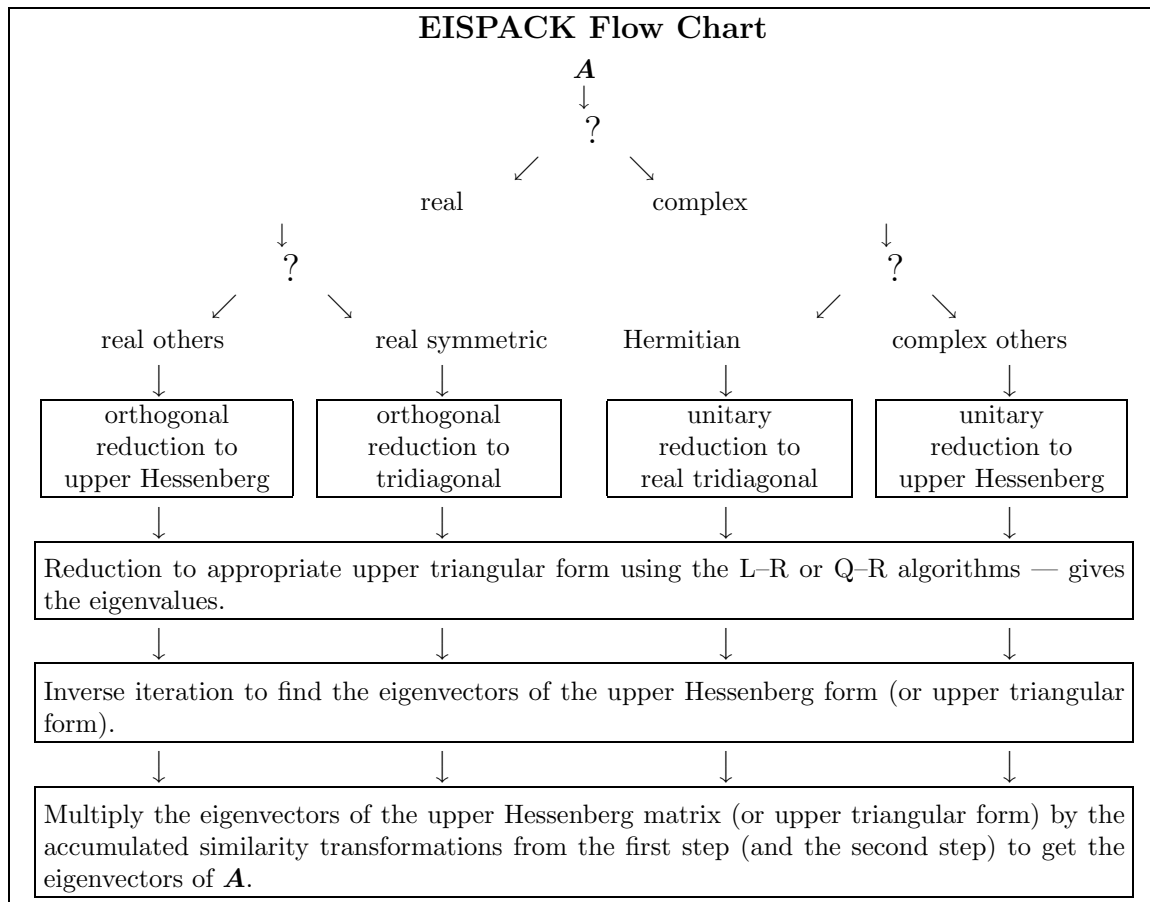
10.8 Balancing a matrix.

The idea here is to try and make the elements of the matrix as close to the same magnitude as possible using a diagonal similarity transformation.

$$\overline{\boldsymbol{A}} = \boldsymbol{D} \boldsymbol{A} \boldsymbol{D}^{-1}, \; \boldsymbol{D} = \mathrm{diag}\,[d_1, d_2, \ldots, d_n],$$

where $d_i = 2^{k_i}$ for some k_i.

EISPACK Flow Chart

$$\boldsymbol{A}$$
$$\downarrow$$
$$?$$

⟋   ⟍

real   complex

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \downarrow$$
$$? \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ?$$

⟋   ⟍   ⟋   ⟍

real others   real symmetric   Hermitian   complex others

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

orthogonal reduction to upper Hessenberg	orthogonal reduction to tridiagonal	unitary reduction to real tridiagonal	unitary reduction to upper Hessenberg

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

Reduction to appropriate upper triangular form using the L–R or Q–R algorithms — gives the eigenvalues.

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

Inverse iteration to find the eigenvectors of the upper Hessenberg form (or upper triangular form).

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

Multiply the eigenvectors of the upper Hessenberg matrix (or upper triangular form) by the accumulated similarity transformations from the first step (and the second step) to get the eigenvectors of $\boldsymbol{A}$.

10.9 EISPACK.

EISPACK was written over a number of years around 1972 at a cost of some millions of dollars. As well as the published book, Garbow *et al.* (1977), Smith *et al.* (1976), there is machine readable documentation, a short description of the main driver subroutines and a large file with a description of all subroutines. Except for routines which supply specialized output most of the package can be described by the flow chart. If only eigenvalues are required then no accumulation of transformations is necessary. The algorithms are discussed in Wilkinson and Reinsch (1971).

EISPACK has now been superceded by LAPACK, Anderson *et al.* (1995), a combination of EISPACK and LINPACK.

10.10 Other eigenvalue problems.

10.10.1 General linear problem for Hermitian systems.

$$(\boldsymbol{A} - \lambda\boldsymbol{B})\boldsymbol{v} = \boldsymbol{0}, \ \boldsymbol{B} \quad \text{positive definite}$$

$$(\boldsymbol{A} - \lambda\boldsymbol{L}\boldsymbol{L}^t)\boldsymbol{v} = \boldsymbol{0}, \ \text{where } \boldsymbol{B} = \boldsymbol{L}\boldsymbol{L}^t \quad \text{(Choleski)}$$

$$(\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-t} - \lambda\boldsymbol{I})(\boldsymbol{L}^t\boldsymbol{v}) = \boldsymbol{0}.$$

That is, find eigenvalues of the symmetric matrix $\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-t}$. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are banded then $\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-t}$ has complete fill-in so methods exist (L–Z and Q–Z algorithms) for the simultaneous reduction of $\boldsymbol{A}$ and $\boldsymbol{B}$ to upper triangular form. The L–Z and Q–Z algorithms also treat this problem more efficiently if $\boldsymbol{B}$ is ill-conditioned.

10.10.2 Singular value decomposition

The algorithm, Golub and Kahan (1965), uses an implicit Q–R algorithm, that is, it implicitly finds the square roots of the eigenvalues of, and the eigenvectors of $\boldsymbol{A}\boldsymbol{A}^t$ and $\boldsymbol{A}^t\boldsymbol{A}$.

10.10.3 Band matrices

There is a method using plane rotations to reduce matrices of this class to tridiagonal form without causing massive "fill-in". Basically an element on the edge of the band is zeroed, which introduces one non-zero outside the band. This is then "chased off" the matrix by further plane rotations before reducing another edge element to zero.

10.10.4 Nonlinear eigenvalue problems.

The ideas in this section can also be used to improve the estimates of eigenvalues. Suppose a nonlinear matrix function $\boldsymbol{M}(\lambda)$ is given, where $\boldsymbol{M}$ is $n \times n$, and $(\lambda, \boldsymbol{v})$ is required such that

$$\boldsymbol{M}(\lambda)\boldsymbol{v} = \boldsymbol{0}, \ \boldsymbol{v} \neq \boldsymbol{0} \qquad \text{or} \ \det |\boldsymbol{M}(\lambda)| = 0.$$

(i) Method (Osborne (1964)).

Consider fixed $\boldsymbol{x}$ and $\boldsymbol{s}$, then the system

$$\boldsymbol{M}(\lambda)\boldsymbol{v} = \beta(\lambda)\boldsymbol{x},$$
$$\boldsymbol{s}^t\boldsymbol{v} = 1,$$

defines

$$\beta(\lambda) = \frac{1}{\boldsymbol{s}^t \boldsymbol{M}^{-1}(\lambda)\boldsymbol{v}}.$$

Lemma: If $\det |\boldsymbol{M}(\lambda)| \to 0$ then $\beta(\lambda) \to 0$ for appropriate $\boldsymbol{s}$ and $\boldsymbol{x}$.

Proof: Compare this with inverse iteration. Let $\boldsymbol{M}(\lambda) = \boldsymbol{T}\boldsymbol{D}\boldsymbol{T}^{-1}$ and suppose that λ is close to λ^* where $\boldsymbol{M}(\lambda^*)$ has its first eigenvalue $d_1(\lambda^*)$ equal to zero. Let $\boldsymbol{D}(\lambda) = \text{diag}\,[d_1(\lambda), d_2(\lambda), \ldots, d_n(\lambda)]$.

$$\boldsymbol{M}^{-1}(\lambda) = \boldsymbol{T}\boldsymbol{D}^{-1}\boldsymbol{T}^{-1},$$

$$\beta(\lambda) = \frac{1}{(\boldsymbol{s}^t\boldsymbol{T})\boldsymbol{D}^{-1}(\boldsymbol{T}^{-1}\boldsymbol{x})}$$
$$= \frac{1}{\sum_{i=1}^{n} d_i^{-1}\langle \boldsymbol{s}, \kappa_i(\boldsymbol{T})\rangle\langle \rho_i(\boldsymbol{T}^{-1}), \boldsymbol{x}\rangle}$$
$$= \frac{d_1}{\langle \boldsymbol{s}, \kappa_1(\boldsymbol{T})\rangle\langle \rho_1(\boldsymbol{T}^{-1}), \boldsymbol{x}\rangle + \sum_{i=2}^{n} d_1/d_i\langle,\rangle\langle,\rangle}.$$

Provided $\boldsymbol{s}$ is not perpendicular to $\kappa_1(\boldsymbol{T})$ and $\boldsymbol{x}$ is not perpendicular to $\rho_1(\boldsymbol{T}^{-1})$, $\beta(\lambda) \to 0$ as $\det |\boldsymbol{M}(\lambda)| \to$ ∎
0. ∎

Applying the Newton iteration to $\beta(\lambda)$ gives,

$$\lambda_{i+1} = \lambda_i - \frac{\beta(\lambda_i)}{\beta'(\lambda_i)}$$

$$\beta'(\lambda) = -(\boldsymbol{s}^t\boldsymbol{M}^{-1}\boldsymbol{x})^{-2}\left(\boldsymbol{s}^t\frac{d\boldsymbol{M}^{-1}(\lambda)}{d\lambda}\boldsymbol{x}\right)$$
$$= +(\boldsymbol{s}^t\boldsymbol{M}^{-1}\boldsymbol{x})^{-2}\left(\boldsymbol{s}^t\boldsymbol{M}^{-1}\frac{d\boldsymbol{M}}{d\lambda}\boldsymbol{M}^{-1}\boldsymbol{x}\right)$$
$$\therefore \quad \frac{\beta(\lambda)}{\beta'(\lambda)} = \frac{\boldsymbol{s}^t\boldsymbol{M}^{-1}(\lambda)\boldsymbol{x}}{\boldsymbol{s}^t\boldsymbol{M}^{-1}\frac{d\boldsymbol{M}}{d\lambda}\boldsymbol{M}^{-1}(\lambda)\boldsymbol{x}}$$

(ii) Algorithm.

1. λ_0, $\boldsymbol{x}_0$, $i = 1$, $eps = ?$
2. $\boldsymbol{y}_i = \boldsymbol{M}^{-1}(\lambda_{i-1})\boldsymbol{x}_{i-1}$
3. $\boldsymbol{u}_i = \boldsymbol{M}^{-1}(\lambda_{i-1})\frac{d\boldsymbol{M}}{d\lambda}(\lambda_{i-1})\boldsymbol{y}_i$
4. Choose $\boldsymbol{s} = \boldsymbol{e}_j$, where $|\boldsymbol{u}_i(j)|$ is the largest element of $\boldsymbol{u}_i$.
5. $\lambda_i = \lambda_{i-1} - \boldsymbol{s}^t\boldsymbol{y}_i/\boldsymbol{s}^t\boldsymbol{u}_i$
   $= \lambda_{i-1} - \boldsymbol{y}_i(j)/\boldsymbol{u}_i(j).$

6. $\boldsymbol{x}_i = \boldsymbol{u}_i / \boldsymbol{u}_i(j)$ (renormalize)

7. If $|\beta(\lambda)| = |1/\boldsymbol{y}_i(j)| < eps$, stop
   else $i = i + 1$, goto 2.

Note: $\beta(\lambda)$ is redefined at each iteration, but in fact this improves convergence as $\boldsymbol{x}_i \rightarrow$ eigenvector. If $\boldsymbol{M}(\lambda)$ is symmetric then $\boldsymbol{s}$ can be chosen as $\boldsymbol{x}_i$ in step 4.

(iii) An example: A large $n \times n$ matrix with a banded structure and a few non-zeros in the top right and bottom left corners (periodic modelling of structures) can be written as $\boldsymbol{A} + \boldsymbol{W}\boldsymbol{W}^t$ where $\boldsymbol{A}$ is a banded matrix and $\boldsymbol{W}$ is $n \times m$, where $m << n$. The matrix $\boldsymbol{B}$ is banded, so the large linear eigenvalue problem $(\boldsymbol{A} + \boldsymbol{W}\boldsymbol{W}^t - \lambda \boldsymbol{B})\boldsymbol{y} = \boldsymbol{0}$ can be changed to a small $m \times m$ non-linear problem as follows.

$$(\boldsymbol{A} + \boldsymbol{W}\boldsymbol{W}^t - \lambda \boldsymbol{B})\boldsymbol{y} = \boldsymbol{0}.$$

$$(\boldsymbol{A} - \lambda \boldsymbol{B})\boldsymbol{y} = -\boldsymbol{W}\boldsymbol{W}^t\boldsymbol{y}$$
$$\boldsymbol{y} = -(\boldsymbol{A} - \lambda \boldsymbol{B})^{-1}\boldsymbol{W}\boldsymbol{W}^t\boldsymbol{y}$$
$$\boldsymbol{W}^t\boldsymbol{y} = -\boldsymbol{W}^t(\boldsymbol{A} - \lambda \boldsymbol{B})^{-1}\boldsymbol{W}\boldsymbol{W}^t\boldsymbol{y}$$
$$\underbrace{\left(\boldsymbol{I}_m + \boldsymbol{W}^t(\boldsymbol{A} - \lambda \boldsymbol{B})^{-1}\boldsymbol{W}\right)}_{\boldsymbol{M}(\lambda)}\boldsymbol{v} = \boldsymbol{0}, \qquad \boldsymbol{v} = \boldsymbol{W}^t\boldsymbol{y}.$$

10.10.5 Matrix pencil problems

Here the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are non-square and values of λ are required so that $\boldsymbol{A} = \lambda \boldsymbol{B}$ has less than full rank.

10.11 Exercises.

1. Show that if the matrix $\boldsymbol{A}$ in the L–R or Q–R algorithm is upper Hessenberg then all iterates $\boldsymbol{A}^{(k)}$ are upper Hessenberg.

   (i) Show that an L–U or Q–U factorization of an upper Hessenberg matrix cost $O(n^2)$ operations.

2. Show that the iterates $\boldsymbol{A}^{(k)}$ are similar in the case of double shifts.

3. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are positive definite show that the eigenvectors of $(\boldsymbol{A} - \lambda \boldsymbol{B})\boldsymbol{v} = \boldsymbol{0}$ are both A–conjugate and B–conjugate.

4. Find $\frac{d\boldsymbol{M}}{d\lambda}$ if $\boldsymbol{M}(\lambda) = \boldsymbol{I} + boldW^t(\boldsymbol{A} - \lambda \boldsymbol{B})^{-1}\boldsymbol{W}$. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are $n \times n$ tridiagonal matrices and $\boldsymbol{W}$ is a vector find the number of operations to calculate $\beta(\lambda)/\beta'(\lambda)$.

5. Let $\boldsymbol{A}$ be positive definite with eigenvalues λ_i, $i = 1, 2, \ldots, n$. Show that the eigenvalues of $\boldsymbol{A} + \boldsymbol{v}\boldsymbol{v}^t$ interlace those of $\boldsymbol{A}$, that is, between every consecutive pair of λ_i there is an eigenvalue of $\boldsymbol{A} + \boldsymbol{v}\boldsymbol{v}^t$.

6. Find the eigenvalues and eigenvectors of the symmetric tridiagonal matrix which has diagonal elements α and off-diagonal elements β. This has connections with the Discrete (Fast) Fourier Transform (see Briggs and Henson (1995)) because of the eigenvector structure.

   Hint: Consider the associated second order difference equation with appropriate homogeneous boundary conditions.

7. Show that the non-symmetric tridiagonal matrix $\boldsymbol{A}$ can be diagonally similarity transformed to a symmetric tridiagonal matrix if $\gamma_i \beta_i > 0$, $i = 2, \ldots, n$ as in §10.5.1.

References

Anderson E., Bai Z., Bischof C., Demmel J., Dongarra J., Du Croz J., Greenbaum A., Hammarling S., McKenney A., Ostrouchov S. and Sorenson D., *LAPACK Users' Guide, Second Edition*, SIAM, Philadelphia, 1995.

Anton H. and Rorres C., *Elementary Linear Algebra with Applications*, Wiley, 1987.

Atkinson K., *An Introduction to Numerical Analysis*, 2nd Ed., Wiley, 1989.

Ben-Israel A. and Greville T.N.E., *Generalised Inverses: Theory and Applications*, Wiley, 1974.

Björck A., Iterative refinement of linear least squares solutions II. *BIT*, **8**, pp8-30, 1968.

Blackford L.S., Choi J., Cleary E., D'Azevedo E., Demmel J., Dhillon I., Dongarra J., Hammarling S., Henry G., Petitet A., Stanley K., Walker D. and Whaley R.C., *ScaLAPACK User's Guide*, SIAM, Philadelphia, 1997.

J. Boothroyd., *Australian Computer Journal*, 1(1), 1968.

Briggs W.L. and Van Emden Henson, *The DFT: An Owner's Manual for the Discrete Fourier Transform*, SIAM, Philadelphia, 1995.

Dahlquist, G., Björck A. and Anderson N., *Numerical Methods*, Prentice-Hall, 1974.

Demmel J.W., *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

Dongarra J.J., Bunch J.R., Moler C.B. and Stewart G.W., *LINPACK User's Guide*, SIAM, Philadelphia, 1979.

Dongarra J.J., Duff I. and Sorenson D., *Solving Linear Systems on Vector and Shared Memory Computers*, SIAM, Philadelphia, 1990.

Duff I., A survey of sparse matrix research, *Proceedings IEEE*, **65**, pp500-535, 1977.

Forsythe G.E. and Moler C.B., *Computer Solution of Linear Algebraic Systems*, Prentice Hall Series in Automatic Computation, Englewood Cliffs, N.J., 1967.

Francis J., The Q–R transformation. A unitary analogue to the LR transformation. *Computer Journal*, **4**, pp 265–271, 1961.

Froberg C.E., *Introduction to NUmerical Analysis*, Addison Wesley, 1965.

Garbow B.S., Boyle J.M., Dongarra J.J. and Moler C.B., *Matrix Eigensystem Routines: EISPACK Guide Extension*, Lecture Notes in Computer Science, Springer-Verlag, 1977.

George A. and Liu J., *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, 1981.

Golub, G.H., Iterative Refinement of least squares solutions, *Numerische Mathematik*, **9**, 139-148, 1966.

Golub G.H. and Kahan W., Calculating the singular values and pseudo inverse of matrices. *SIAM Journal Numerical Analysis*, **2**, 205-224, 1965.

Golub G.W. and van Loan C.F., *Matrix Computations*, John Hopkins University Press, Baltimore, Edition 2, 1989.

Hestenes, M.R. and Steifel, E., Methods of Conjugate Gradients for Solving Linear Systems, *National Bureau Standards Report*, No. 1659, 1952.

Hoffman A.J. and Wielandt H.W., The variation of the spectrum of a normal matrix, *Duke Mathematical Journal*, **20**, pp 37–39, 1953.

Householder A., *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York, 1964.

Jennings L.S., Simultaneous Equation Estimation, Computational Aspects, *Journal of Econometrics*, **12**, pp 23–39, 1980.

Jennings, L.S. and Osborne, M.R., A direct error analysis for least squares, *Numerische Mathematik*, **22**, 325-332, 1974.

Kahan W., Numerical linear algebra, *Canadian Mathematics Bulletin*, **9**, pp 756–801, 1966.

Lawson C.L. and Hanson R.J., *Solving Least Squares Problems*, Prentice-Hall, 1974.

Lipschutz S., *Linear Algebra*, Schaum outline series, McGraw Hill, 1968.

Moore E.H., On the reciprocal of the general algebraic matrix, *Bulletin American Mathematical Society*, **26**, pp394–395, 1920.

Noble B., *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, N.J., 1969.

Noble B. and Daniel J., *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, N.J., 1977.

Osborne M.R., A new method for the solution of eigenvalue problems. *Computer Journal*, **7**, pp 358–362, 1964

Parlett B.N., Global convergence of the basic QR algorithm on Hessenberg matrices, *Mathematics of Computation*, **22**, pp 803–817, 1968.

Parlett B.N., *The Symmetric Eigenvalue Problem*, Prentice-Hall, 1981.

Penrose R., On best approximate solutions of linear matrix equations, *Proceedings of the Cambridge Philosophical Society*, **52**, pp 17–19, 1956.

Ralston A., *A First Course in Numerical Analysis*, McGraw-Hill, 1965.

Rutishauser H., Solution of eigenvalue problems with the LR transformation. *Applied Mathematics Series, National Bureau Standards*, **49**, pp 47-81, 1958

Rutishauser H., Once again: The least squares problem. *Linear Algebra and Its Applications*, **1**, pp 479–488, 1968.

Rutishauser H. and Schwarz H.R., Handbook Series Linear Algebra. The LR transformation method for symmetric matrices. *Numerische Mathematik*, **5**, pp 273–289, 1963.

Smith B.T., Boyle J., Garbow B., Ikebe Y., Klema V. and Moler C., Matrix Eigensystem Routines—EISPACK Guide, 2nd Ed., Lecture Notes in Computer Science, **6**, Springer Verlag, New York, 1976.

Stewart G.W., *Introduction to Matrix Computations*, Academic Press, New York, 1973.

Strang Gilbert, *Linear Algebra and its Applications*, 3rd ed., Harcourt Brace Jovanovich, 1988.

Trefethen L.N. and Bau D., *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

Varga R.S., *Matrix Iterative Analysis*, Prentice-Hall, 1962.

Wilkinson J.H., Rounding Errors in Algebraic Processes, Prentice-Hall, Englewood Clifs, N.J., 1963.

Wilkinson J.H., *The Algebraic Eigenvalue Problem*, Clarendon Press, 1965.

Wilkinson J.H. and Reinsch C., *Handbook for Automatic Computation*, Volumes I and II, Springer, 1971.

Young D., *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.