# BRUNEL UNIVERSITY

## DEPARTMENT OF MATHEMATICAL SCIENCES

SYLLABUS

**Module Code:** MA5151A

**Module Title:** Nonlinear equations and computing.

**Semester:** Autumn

**Level:** 5

**Lecturer:** Drs. Michael Warby and Simon Shaw

**Credit value:** 15

**Prerequisites:** Mathematics or other numerate degree.

**Method of delivery:** **C programming:** *twelve hours of PC lab sessions with three one-hour follow up seminars.* **LaTeX 2ε:** *three hours of PC lab sessions.* **Linear and nonlinear equations:** *fifteen one-hour lectures.*

**Formative feedback:** Participation in the lectures, seminars and PC lab sessions and a marked programming assignment.

**Method of assessment:** **50% in-module assessment:** C programming project (36%) plus LaTeX 2ε project report(14%).
**50% final exam** on linear and nonlinear equations (1.5 hour, closed book, unseen, written).

## AIMS:

- To develop C programming skills for numerical problems.
- To develop mathematical typesetting skills using LaTeX 2ε.
- To introduce and analyse the established techniques for solving linear and nonlinear equations.

# OBJECTIVES:

On successful completion of this module students should:

- be able to implement numerical algorithms in the C programming language.
- be able to use the LaTeX 2ε typesetting package.
- be able to solve linear equations using direct or iterative methods.
- be able to use Newton-type methods for solving nonlinear equations.

# MODULE CONTENT:

### C programming.

(i) The unix environment, basic operators, input and output.

(ii) Program flow: `if`, `while`, `switch`, `for` etc.

(iii) Arrays and pointers, dynamic memory allocation, pointers to pointers.

(iv) Functions: built-in and user defined, prototyping, call-by value/reference, linking separate files, pointers to functions.

### LaTeX 2ε.

(i) Typing mathematics and the `\label` and `\ref` commands.

(ii) Processing and printing LaTeX 2ε documents.

### Linear equations by direct methods.

(i) Basic Gauss elimination and pivoting.

(ii) The $LU$, $QR$ and the Cholesky $LL^T$ factorizations.

(iii) Vector and matrix norms, the sensitivity of linear equations and the condition number.

### Nonlinear equations

(i) The contraction mapping theorem in $\mathbb{R}$ and $\mathbb{R}^n$.

(ii) Order of convergence, inverse interpolation, bracketing techniques and multiple roots in one-dimension.

(iii) Basic Newton's method for systems and minimization. Continuation techniques. Damping, approximate Jacobians and the line search.

### Linear equations by the conjugate gradient method

(i) The basic iteration.

(ii) Convergence and preconditioning.

# READING LIST

## Core Texts:

- N. Kantaris, *Learning to program in C*, Bernard Babani Books (BP 258) 1997.

## Supplementary Texts:

- B. N. Datta (1995): Numerical Linear Algebra and Applications, Brooks/Cole Publishing Company. 1995. QA184.D37.
- J. E. Dennis and R. B. Schnable, Numerical Methods for Unconstrained Optimization, Prentice-Hall, 1983, Re-published by SIAM in the Classics in Applied Mathematics series, SIAM, Philadelphia, 1997. QA218.D46.
- H. Kopka and P. W. Daly, A guide to LaTeX, Addison Wesley, Third Edition, 1999.
- W. H. Press and , S. A. Teukolsky and , W. T. Vetterling and B. P. Flannery, Numerical Recipes in C, Cambridge University Press, 1995. QA297.N882
- B. W. Kernighan and D. M. Ritchie, The C programming language. Prentice Hall, Second Edition 1988. QA76.73.C15K47

---

I will put these printed notes for the nonlinear equations part of this module (as a gzipped postscript file) in my home page area which has the URL:

`http://www.brunel.ac.uk/~icstmkw/home.html`

Please let me know if you find mistakes and would like to suggest improvements.

# MA5151A 2002–2003 Teaching Schedule

## Calender

```
                                 2002

        July                  August               September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6                1  2  3    1  2  3  4  5  6  7
 7  8  9 10 11 12 13    4  5  6  7  8  9 10    8  9 10 11 12 13 14
14 15 16 17 18 19 20   11 12 13 14 15 16 17   15 16 17 18 19 20 21
21 22 23 24 25 26 27   18 19 20 21 22 23 24   22 23 24 25 26 27 28
28 29 30 31            25 26 27 28 29 30 31   29 30

      October               November              December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
       1  2  3  4  5                   1  2    1  2  3  4  5  6  7
 6  7  8  9 10 11 12    3  4  5  6  7  8  9    8  9 10 11 12 13 14
13 14 15 16 17 18 19   10 11 12 13 14 15 16   15 16 17 18 19 20 21
20 21 22 23 24 25 26   17 18 19 20 21 22 23   22 23 24 25 26 27 28
27 28 29 30 31         24 25 26 27 28 29 30   29 30 31


                                 2003

      January               February                March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                      1                      1
 5  6  7  8  9 10 11    2  3  4  5  6  7  8    2  3  4  5  6  7  8
12 13 14 15 16 17 18    9 10 11 12 13 14 15    9 10 11 12 13 14 15
19 20 21 22 23 24 25   16 17 18 19 20 21 22   16 17 18 19 20 21 22
26 27 28 29 30 31      23 24 25 26 27 28      23 24 25 26 27 28 29
                                              30 31
```

## C programming part: 12 hours in September/October

| Lecture number (approx) | Topics |
|---|---|
| 1–3 | Introduction to C programming: the unix environment, basic operators, input and output. |
| 4–6 | Program flow: `if`, `while`, `switch`, `for` etc. |
| 7–9 | Arrays and pointers, dynamic memory allocation, pointers to pointers. |
| 10–12 | Functions: built-in and user defined, prototyping, call-by value/reference, linking separate files, pointers to functions. |

There will be **3** follow-up C "surgeries" which are currently planned as follows:

Tuesday 19 November, 1100–1200 in room `M123` (week 8).

Tuesday 3 December, 1100–1200 in room `M123` (week 10).

Tuesday 17 December, 1100–1200 in room `M123` (week 12).

### LaTeX part: 3 hours on Tuesday 5 November (week 6)

All the sessions will be at the usual times (1100–1200, 1400–1600) in room `M123`.
This will involve an introduction to $\LaTeX\,2_\varepsilon$ including typing mathematics and the `\label` and `\ref` commands and processing and printing $\LaTeX\,2_\varepsilon$ documents.

### Nonlinear equations part: 18 hours in November–December

The current plan is that these 18 hours will be as follows.

```
Week
6    Wed   6  Nov  1100--1200   LC067
6    Wed   6  Nov  1200--1300   LC106
7    Tue  12  Nov  1100--1200   LC111
7    Tue  12  Nov  1400--1500   LC111
7    Tue  12  Nov  1500--1600   LC263
7    Wed  13  Nov  1100--1200   LC067
8    Tue  19  Nov  1400--1500   LC111
8    Tue  19  Nov  1500--1600   LC263
9    Tue  26  Nov  1100--1200   LC111
9    Tue  26  Nov  1400--1500   LC111
9    Tue  26  Nov  1500--1600   LC263
10   Tue   3  Dec  1400--1500   LC111
10   Tue   3  Dec  1500--1600   LC263
11   Tue  10  Dec  1100--1200   LC111
11   Tue  10  Dec  1400--1500   LC111
11   Tue  10  Dec  1500--1600   LC263
12   Tue  17  Dec  1400--1500   LC111
12   Tue  17  Dec  1500--1600   LC263
```

Thus in week 6 there will be 2 lectures on the Wednesday as well as the 3 hours of LaTeX on the Tuesday and in week 7 there will be in total 4 lectures. These special arrangements are because there were no labs or lectures in `MA5151A` in week 1. Weeks 8–12 will then follow the usual pattern of either 2 lectures in nonlinear equations and 1 lab session in C or 3 lectures in nonlinear equations.

## Possible teaching timetable

| Lecture number (approx) | Topics |
|---|---|
| 1–2 | Introduction to the linear and nonlinear problems of the module. Mathematical preliminaries: Taylor's series in one and $n$ dimensions. Basic Newton's method for systems and minimization. A revision of basic linear algebra. The space $\mathbb{R}^n$. Vector and matrix norms. |
| 3 | Basic Gauss elimination and the $LU$ factorization. |
| 4 | Pivoting, tri-diagonal systems and the Cholesky factorization $LL^T$ (and possibly the $QR$ factorization). |
| 5 | The sensitivity of linear equations and the condition number. |
| 6 | Iteration functions and the contraction mapping theorem. |
| 7–8 | One-dimensional aspects: Order of convergence, inverse interpolation, bracketing techniques and multiple roots. |
| 9 | The convergence of Newton's method for systems. |
| 10–11 | Continuation techniques. The damped Newton method. The line search. Scaling. |
| 12–15 | The basic conjugate gradient iteration, the convergence of the iteration and preconditioning. |
| 16–18 | Revision of the module. |

### Extra dates

Revision week: Monday 13 January 2003–Friday 17 January 2003.

Examination period: Monday 20 January 2003–Friday 31 January 2003.

# 1 Introduction

## 1.1 Some notation relating to $\mathbb{R}^n$

In these notes the following notation is used.

$\mathbb{R}$ = the set of real numbers.

$\mathbb{R}^n$ = the set of real vectors of length $n$.

$\mathbb{R}^{m,n}$ = the set of real matrices with $m$ rows and $n$ columns.

To denote the column vector of length $n$ we use the notation

$$\underline{x} = (x_i) = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} .$$

## 1.2 The problems considered in this module

This module is concerned with various methods and theory relating to the following problems.

(i) Given an $n \times n$ matrix $A$ and a $n \times 1$ vector $\underline{b}$, solve $A\underline{x} = \underline{b}$.

(ii) For $f : \mathbb{R} \to \mathbb{R}$ solve $f(x) = 0$.

(iii) For $\underline{f} : \mathbb{R}^n \to \mathbb{R}^n$ solve $\underline{f}(\underline{x}) = \underline{0}$.

(iv) For $f : \mathbb{R}^n \to \mathbb{R}$ minimise $f(\underline{x})$.

If in the above $f$ or $\underline{f}$ are not defined everywhere then we would have $f : \mathcal{X} \to \mathbb{R}$ or $\underline{f} : \mathcal{X} \to \mathbb{R}^n$ where $\mathcal{X}$ is suitable subset on which $f$ or $\underline{f}$ is defined.

In this module we will start with a revision of basic linear algebra and introduce vector and matrix norms. We then consider solving problem (i) by Gauss elimination and matrix factorization methods. As the material may already be familiar to you the pace of coverage of the material will be quite fast. Next, we consider Newton type iterative methods for attempting to solve (ii), (iii) and (iv) and we will then return to problem (i) to consider a specific iterative method known as the conjugate gradient method (CGM) in the case that the matrix $A$ is symmetric and positive definite.

The four problems listed above are connected in a number of ways. Trivially we can note that problem (ii) is a special case of problem (iii) corresponding to $n = 1$. To solve problem (iii) by Newton type methods involves an iteration and this in turn requires the solution of a linear system (problem (i)) at each step. Problems (iii) and (iv) are related in that we can attempt to solve problem (iii) by attempting to minimise the scalar function $g(\underline{x}) = \underline{f}(\underline{x})^T \underline{f}(\underline{x})$ (which has a minimum when $\underline{f}(\underline{x}) = \underline{0}$). Conversely, we can attempt to solve problem (iv) by attempting to solve the nonlinear equations

$$\nabla f(\underline{x}) = \underline{0}$$

where $\nabla$ is the gradient operator (see later). This connection between solving a system of equations and minimising a function is used again when the CGM is considered as this is a method for minimising the quadratic

$$\phi(\underline{x}) = \frac{1}{2} \underline{x}^T A \underline{x} - \underline{x}^T \underline{b} .$$

The unique point $\underline{x}$ which minimises this quadratic $\phi$ is also the unique solution of $A\underline{x} = \underline{b}$ when $A$ is symmetric and positive definite.

## 1.3 Some terminology

When we consider the nonlinear problems in $\mathbb{R}^n$ there are a number of terms that will occur throughout and hence we introduce these here.

For a scalar function $f$ we have the **gradient vector** $\nabla f$ and **Hessian matrix** $H_f$ given by

$$\nabla f := \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} \quad \text{and} \quad H_f := \left( \frac{\partial^2 f}{\partial x_i \partial x_j} \right) . \tag{1.1}$$

For the vector function $\underline{f}$ we have the **Jacobian matrix**

$$J_f = \left( \frac{\partial f_i}{\partial x_j} \right) = \left( \frac{\partial \underline{f}}{\partial x_1}, \cdots, \frac{\partial \underline{f}}{\partial x_n} \right) = \begin{pmatrix} (\nabla f_1)^T \\ \vdots \\ (\nabla f_n)^T \end{pmatrix} . \tag{1.2}$$

Note that if $f$ is a scalar then the Jacobian matrix of the vector $\nabla f$ is the Hessian matrix $H_f$. The Hessian matrix is symmetric (provided $f$ is twice continuously differentiable) but for a general $\underline{f}$ the Jacobian matrix $J_f$ will usually be non-symmetric.

In the problem (iv), as we will later show that at a (strong) local minimum of $f(\underline{x})$ the Hessian matrix will be positive semi-definite.

As a final remark about terminology, we will show that the gradient vector and the Hessian matrix of the quadratic

$$\phi(\underline{x}) = \frac{1}{2}\underline{x}^T A \underline{x} - \underline{x}^T \underline{b}$$

are respectively

$$J_\phi = \nabla \phi = A\underline{x} - \underline{b} \quad \text{and} \quad H_\phi = A .$$

## 1.4 Taylor's series for $f(x)$, $f(\underline{x})$ and $\underline{f}(\underline{x})$

To develop the methods used in this module it is necessary to understand the expansions of $f(x+h)$, $f(\underline{x}+\underline{h})$ and $\underline{f}(\underline{x}+\underline{h})$. These are as follows.

The simplest is

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \cdots + \frac{h^r}{r!}f^{(r)}(x) + \cdots \tag{1.3}$$

or, when a remainder is used,

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \cdots + \frac{h^r}{r!}f^{(r)}(x+\theta h) \tag{1.4}$$

for some $\theta = \theta(h) \in (0,1)$. To obtain the $n$-dimensional versions we convert them to the 1D-case by considering, for example,

$$g(t) = f(\underline{x} + t\underline{h}) . \tag{1.5}$$

The expansion of $g$ about $t = 0$ is simply

$$g(t) = g(0) + tg'(0) + \frac{t^2}{2}g''(0) + \cdots . \tag{1.6}$$

By the chain rule of partial differentiation

$$g'(t) = \sum_1^n \frac{\partial f}{\partial x_i}(\underline{x} + t\underline{h})h_i = (\nabla f(\underline{x} + t\underline{h}))^T \underline{h} \tag{1.7}$$

and

$$g''(t) = \sum_1^n \sum_1^n \frac{\partial^2 f}{\partial x_i \partial x_j}(\underline{x} + t\underline{h})h_i h_j = \underline{h}^T H_f(\underline{x} + t\underline{h})\underline{h} \ . \tag{1.8}$$

Thus

$$f(\underline{x} + \underline{h}) = f(\underline{x}) + (\nabla f(\underline{x}))^T \underline{h} + \frac{1}{2}\underline{h}^T H_f(\underline{x})\underline{h} + \mathcal{O}(\|\underline{h}\|^3) \tag{1.9}$$

or with remainder

$$f(\underline{x} + \underline{h}) = f(\underline{x}) + (\nabla f(\underline{x}))^T \underline{h} + \frac{1}{2}\underline{h}^T H_f(\underline{x} + \theta\underline{h})\underline{h} \tag{1.10}$$

for some $\theta \in (0,1)$. To obtain an expression for $\underline{f}(\underline{x} + \underline{h})$ we have for each component

$$f_i(\underline{x} + \underline{h}) = f_i(\underline{x}) + (\nabla f_i(\underline{x}))^T \underline{h} + \mathcal{O}(\|\underline{h}\|^2) \tag{1.11}$$

and thus for the whole vector we have

$$\underline{f}(\underline{x} + \underline{h}) = \begin{pmatrix} f_1(\underline{x} + \underline{h}) \\ \vdots \\ f_n(\underline{x} + \underline{h}) \end{pmatrix} = \underline{f}(\underline{x}) + J_f(\underline{x})\underline{h} + \mathcal{O}(\|\underline{h}\|^2) \ . \tag{1.12}$$

In this case we cannot write a remainder in quite the same way as (1.4) or (1.10) as we would have a different intermediate point for each index $i$. The best that can be done is to note that from

$$g(1) - g(0) = \int_0^1 g'(t)dt \tag{1.13}$$

we have

$$f_i(\underline{x} + \underline{h}) - f_i(\underline{x}) = \int_0^1 (\nabla f_i(\underline{x} + t\underline{h}))^T \, dt \, \underline{h} \tag{1.14}$$

and hence

$$\underline{f}(\underline{x} + \underline{h}) - \underline{f}(\underline{x}) = \left( \int_0^1 J_f(\underline{x} + t\underline{h}) \, dt \right) \underline{h} \ . \tag{1.15}$$

That is we have the 'averaged' Jacobian matrix

$$\int_0^1 J_f(\underline{x} + t\underline{h}) \, dt \tag{1.16}$$

instead of the Jacobian matrix evaluated at a point near $\underline{x}$.

## 1.5 Descent directions and the minimization characterization

The function

$$g(t) = f(\underline{x} + t\underline{d}) \tag{1.17}$$

decreases at $t = 0$ if

$$g'(0) = (\nabla f(\underline{x}))^T \underline{d} < 0 \ . \tag{1.18}$$

In such a case $\underline{d}$ is said to be a descent direction of $f$ at $\underline{x}$. If $\underline{x}^*$ is a **strong local minimum** of $f$ then

$$f(\underline{x}^* + \underline{h}) > f(\underline{x}^*) \quad \text{for all } \underline{h} \neq \underline{0}. \tag{1.19}$$

From the Taylor expansion

$$f(\underline{x} + \underline{h}) = f(\underline{x}) + (\nabla f(\underline{x}))^T \underline{h} + \frac{1}{2} \underline{h}^T H_f(\underline{x}) \underline{h} + \mathcal{O}(\|\underline{h}\|^3) \tag{1.20}$$

this implies that

$$\nabla f(\underline{x}^*) = \underline{0} \tag{1.21}$$

and

$$\underline{h}^T H_f(\underline{x}^*) \underline{h} \geq 0 \quad \text{for all } \underline{h} \neq \underline{0}. \tag{1.22}$$

## 1.6 The basic Newton method for nonlinear problems

From the Taylor expansions given earlier we are in a position to give the unmodified form of Newton's method for the nonlinear problems to be considered in this module. Assuming that the scalar $x_k$ or the vector $\underline{x}_k$ are the 'best' current estimates of $x^*$ or $\underline{x}^*$, as appropriate, then local linear models of $f$ or $\underline{f}$ about $x_k$ or $\underline{x}_k$ are respectively

$$M_k(x) = f(x_k) + f'(x_k)(x - x_k) \tag{1.23}$$

and

$$\underline{M}_k(\underline{x}) = \underline{f}(\underline{x}_k) + J_f(\underline{x}_k)(\underline{x} - \underline{x}_k) \ . \tag{1.24}$$

The scalar $x_{k+1}$ or the vector $\underline{x}_{k+1}$ are then determined so that

$$M_k(x_{k+1}) = 0$$

or

$$\underline{M}_k(\underline{x}_{k+1}) = \underline{0}$$

as appropriate giving, in the scalar case,

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{1.25}$$

and in the vector case

$$\underline{x}_{k+1} = \underline{x}_k - J_f(\underline{x}_k)^{-1} \underline{f}(\underline{x}_k) \ . \tag{1.26}$$

In the case of minimising $f(\underline{x})$ we similarly get

$$\underline{x}_{k+1} = \underline{x}_k - H_f(\underline{x}_k)^{-1} \nabla f(\underline{x}_k). \tag{1.27}$$

Another way of viewing (1.27) is by considering the local quadratic model

$$M_k(x) = f(\underline{x}_k) + (\nabla f(x_k))^T (\underline{x} - \underline{x}_k) + \frac{1}{2} (\underline{x} - \underline{x}_k)^T H_f(x_k)(\underline{x} - \underline{x}_k). \tag{1.28}$$

If $H_f(\underline{x}_k)$ is positive definite then $\underline{x}_{k+1}$ in (1.27) minimises $M_k$ in (1.28). The advantage of viewing the method in this way is that it is more evident whether an iteration is converging to a maximum or a minimum.

# 2 A revision of some basic linear algebra

For **matrices** the notation $A = (a_{ij}) \in \mathbb{R}^{n\times m}$ is shorthand for a real matrix with $n$ rows and $m$ columns with the $ij$ entry $a_{ij}$, i.e.

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nm} \end{pmatrix} . \tag{2.1}$$

In this module we will frequently use the following.

The **zero vector** $\underline{0}$ .

The $n \times n$ **identity matrix**

$$I = (\delta_{ij}) = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 1 \end{pmatrix} \in \mathbb{R}^{n\times n} . \tag{2.2}$$

where

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

The identity matrix is a special case of a **diagonal matrix**, i.e. a matrix with all non-diagonal entries equal to zero.

The **standard base vectors** for $\mathbb{R}^n$ will be denoted by $\underline{e}_1, \cdots, \underline{e}_n$ where

$$\underline{e}_i = i\text{th column of } I . \tag{2.4}$$

## 2.1 Basic matrix operations

### 2.1.1 Addition and scalar multiplications

If $A = (a_{ij})$ and $B = (b_{ij})$ are of the same size and $\alpha \in \mathbb{R}$ then

$$A + B = (a_{ij} + b_{ij}), \quad \alpha A = (\alpha a_{ij}) . \tag{2.5}$$

### 2.1.2 Matrix multiplication

For the matrix product $AB$ to be defined we require that

$$\text{number of columns of } A = \text{number of rows of } B .$$

If $A = (a_{ij}) \in \mathbb{R}^{m\times p}$ and $B = (b_{ij}) \in \mathbb{R}^{p\times n}$ then $C = (c_{ij}) = AB \in \mathbb{R}^{m\times n}$ with

$$c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj} = (a_{i1}, \cdots, a_{ip}) \begin{pmatrix} b_{1j} \\ \vdots \\ b_{pj} \end{pmatrix} = (i\text{th row of } A) \times (j\text{th column of } B) . \tag{2.6}$$

Matrix multiplication is **associative**, i.e. $A(BC) = (AB)C$.

**Matrix multiplication is not in general commutative**, i.e. $AB \neq BA$ in general.

There are special situations in which matrices do commute, e.g. diagonal matrices and in situations of the form $A^2A^3 = A^3A^2 = A^5$.

### 2.1.3 Special cases of matrix multiplication

1. If $\underline{x}, \underline{y} \in \mathbb{R}^n$ then we can form the product

$$\underline{x}^T\underline{y} = \underline{y}^T\underline{x}, \tag{2.7}$$

which is a scalar. This is the **inner product** of the vectors $\underline{x}$ and $\underline{y}$. It is the same as what is referred to as the dot product or scalar product which is the jargon more commonly used in modules in Vector Calculus or Mechanics.

2. If $\underline{x}, \underline{y} \in \mathbb{R}^n$ then we can form the product

$$A = \underline{x}\,\underline{y}^T = (x_iy_j) = \begin{pmatrix} x_1y_1 & \cdots & x_1y_n \\ \vdots & \cdots & \vdots \\ x_ny_1 & \cdots & x_ny_n \end{pmatrix} \in \mathbb{R}^{n\times n} \ . \tag{2.8}$$

This is the **outer product** of the vectors $\underline{x}$ and $\underline{y}$. A matrix of the form $A = \underline{x}\,\underline{y}^T$ with $\underline{x} \neq \underline{0}, \underline{y} \neq \underline{0}$ is a **rank 1 matrix** and it is a type of matrix which appears a number of times in this module. It is often used in situations of the form

$$A\underline{u} = (\underline{x}\,\underline{y}^T)\underline{u} = \underline{x}(\underline{y}^T\underline{u}) = (\underline{y}^T\underline{u})\underline{x}, \tag{2.9}$$

where the last 2 equals are respectively as a consequence of the associative property of matrix multiplication and the definition of a scalar times a vector. The important point here is that whatever the vectors $\underline{y}$ and $\underline{u}$ are, $\underline{y}^T\underline{u} \in \mathbb{R}$ is just a scalar and hence for all vectors $\underline{u}$ the vector $A\underline{u}$ will just be a multiple of the vector $\underline{x}$. Another way of saying this is that $A\underline{u}$ only has **one direction** of more abstractly this can be written as

$$\mathrm{span}\{A\underline{u} : \ \underline{u} \in \mathbb{R}^n\} = \mathrm{span}\{\underline{x}\} \ . \tag{2.10}$$

$A\underline{u}$ only having one direction is the reason for the name rank one matrix. As a final remark concerning rank 1 matrices, it may be tempting when you first meet a matrix of the form $\underline{x}\underline{y}^T$ to expand it out to display it in matrix form. It is however usually neater and simpler to leave the matrix in the form $\underline{x}\underline{y}^T$ for the type of manipulation that is done in this module.

### 2.1.4 Euclidean length and orthogonality of vectors

If $\underline{x} = (x_i) \in \mathbb{R}^n$ then $(x_1^2 + x_2^2 + \cdots + x_n^2)^{1/2} = (\underline{x}^T\underline{x})^{1/2}$ is the **Euclidean length** of the vector $\underline{x}$. The Euclidean length is also called the 2-norm — to be covered later in the module.

A vector $\underline{x}$ is said to be **normalised** if it has **Euclidean length=1**, i.e. $\underline{x}^T\underline{x} = 1$.

Vectors $\underline{x}$ and $\underline{y}$ are **orthogonal** if

$$\underline{x}^T\underline{y} = \underline{y}^T\underline{x} = \sum_1^n x_iy_i = 0. \tag{2.11}$$

### 2.1.5 The size of $\underline{x}^T \underline{y}$ — the Cauchy-Schwarz inequality

$$(\underline{x}^T \underline{y})^2 \le (\underline{x}^T \underline{x})(\underline{y}^T \underline{y}) \tag{2.12}$$

Proof

The proof involves the following trick. Since all vectors have non-negative length it follows that for all real numbers $t$

$$0 \le (\underline{x} + t\underline{y})^T (\underline{x} + t\underline{y}) = (\underline{x}^T \underline{x}) + 2t(\underline{x}^T \underline{y}) + t^2 (\underline{y}^T \underline{y}) = c + bt + at^2, \tag{2.13}$$

where $c = \underline{x}^T \underline{x}$, $b = 2\underline{x}^T \underline{y}$ and $a = \underline{y}^T \underline{y}$. As the quadratic cannot be negative it can have at most 1 real root which will be a double root. Thus the quadratic has 1 real double root or 2 complex conjugate root which corresponds to the condition

$$0 \ge b^2 - 4ac = (2\underline{x}^T \underline{y})^2 - 4(\underline{x}^T \underline{x})(\underline{y}^T \underline{y}), \quad \text{i.e. } (\underline{x}^T \underline{y})^2 \le (\underline{x}^T \underline{x})(\underline{y}^T \underline{y}). \tag{2.14}$$

## 2.2 Special types of matrix

### 2.2.1 Diagonal matrix

$$D = \mathrm{diag}\{d_1, \cdots, d_n\} = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{pmatrix}. \tag{2.15}$$

### 2.2.2 Lower triangular matrix

$$A = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & & 0 \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}. \tag{2.16}$$

That is $A = (a_{ij})$ with $a_{ij} = 0$ for $j > i$. A matrix $A$ is said to be **strictly lower triangular** if it is lower triangular and all diagonal entries are zero. A matrix $A$ is said to be **unit lower triangular** if it is lower triangular and all diagonal entries are equal to 1.

### 2.2.3 Upper triangular matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{pmatrix}. \tag{2.17}$$

That is $A = (a_{ij})$ with $a_{ij} = 0$ for $i > j$. A matrix $A$ is said to be **strictly upper triangular** if it is upper triangular and all diagonal entries are zero. A matrix $A$ is said to be **unit upper triangular** if it is upper triangular and all diagonal entries are equal to 1.

### 2.2.4 Tri-diagonal matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & & & 0 \\ a_{21} & a_{22} & a_{23} & & & \vdots \\ \vdots & a_{32} & a_{33} & a_{34} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & a_{n-1,n} \\ 0 & \cdots & & & a_{n,n-1} & a_{nn} \end{pmatrix} . \tag{2.18}$$

That is $A = (a_{ij})$ with $a_{ij} = 0$ for $|i-j| > 1$. The entries $a_{i,i+1}$ are the **super-diagonal** entries. The entries $a_{i+1,i}$ are the **sub-diagonal** entries.

### 2.2.5 Banded matrix

This can be similarly defined.

### 2.2.6 Symmetric matrix

A matrix $A$ is **symmetric** if $A^T = A$.

### 2.2.7 Positive definite matrix

A matrix $A$ is **positive definite** if $\underline{x}^T A \underline{x} > 0$ for all vectors $\underline{x} \neq \underline{0}$.

(Observe that in components $(A\underline{x})_i = \sum_{j=1}^n a_{ij} x_j$ so that

$$\underline{x}^T A \underline{x} = \sum_{i=1}^n x_i (A\underline{x})_i = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j$$

a **quadratic form** in $\mathbb{R}^n$)

### 2.2.8 Positive semi-definite matrix

A matrix $A$ is **positive semi-definite** if $\underline{x}^T A \underline{x} \geq 0$ for all vectors $\underline{x}$.

**Negative definite** and **negative semi-definite** can be similarly defined.

### 2.2.9 Sparse and dense matrices

These are not precisely defined. Roughly speaking a matrix is **sparse** if "most" of its entries are zero and a matrix is **dense** if "most" of its entries are non-zero.

### 2.2.10 Orthogonal matrix

A real matrix $Q$ is **orthogonal** if

$$Q^T Q = Q Q^T = I, \quad \text{i.e. } Q^{-1} = Q^T . \tag{2.19}$$

The condition $Q^T Q = I$ means that the columns of $Q$ are normalised and mutually orthogonal. The columns of $Q$ give an orthonormal basis for $\mathbb{R}^n$. That is, if we write $Q$ in the partitioned form $Q = (\underline{q}_1, \cdots, \underline{q}_n)$ then the vectors $\underline{q}_1, \cdots, \underline{q}_n$ are an orthonormal basis for $\mathbb{R}^n$ in the same way as is the standard basis $\underline{e}_1, \cdots, \underline{e}_n$ which are the columns of the identity matrix $I$.

## 2.3 Singular and non-singular (square) matrices

There are many equivalent ways of stating that a matrix $A$ is singular. A condition which will be used several times in this module is as follows.

$A$ is **singular** if there exists $\underline{x} \neq \underline{0}$ such that

$$A\underline{x} = \underline{0}, \tag{2.20}$$

i.e. the homogeneous problem $A\underline{x} = \underline{0}$ has a non-trivial solution.

$A$ is **non-singular** if the only solution of

$$A\underline{x} = \underline{0} \tag{2.21}$$

is $\underline{x} = \underline{0}$.

We will also use the condition that $A$ is singular when $\det A = 0$ and non-singular when $\det A \neq 0$. Another characterization is that $A$ is singular if it has an eigenvalue equal to 0 and non-singular if all eigenvalues are non-zero.

When $A$ is singular we cannot solve uniquely the linear equations $A\underline{x} = \underline{b}$. When inexact floating point arithmetic is used (as in usual computer calculations) there is difficulty in solving $A\underline{x} = \underline{b}$ when $A$ is "nearly singular". Such a matrix is often described as "ill-conditioned" or "badly-conditioned". We will consider this aspect more precisely in this module.

## 2.4 Eigenvalues and Eigenvectors

A vector $\underline{v} \neq \underline{0}$ is an **eigenvector** of a matrix $A$ with **eigenvalue** $\lambda$ if

$$A\underline{v} = \lambda\underline{v} \ . \tag{2.22}$$

Thus the homogeneous problem

$$(\lambda I - A)\underline{v} = \underline{0} \tag{2.23}$$

has a non-trivial solution and the matrix $\lambda I - A$ is singular. In this context this condition is usually expressed as

$$\det(\lambda I - A) = 0, \tag{2.24}$$

which is the **characteristic equation**.

$$p_A(\lambda) = \det(\lambda I - A) \tag{2.25}$$

is known as the **characteristic polynomial**. For an $n \times n$ matrix $A$ the polynomial $p_A(\lambda)$ has degree $n$ and by the fundamental theorem of algebra[1] we know that it has exactly $n$ (complex) roots provided multiple roots are counted appropriately. Hence an $n \times n$ matrix has exactly $n$ eigenvalues which may be complex. In the case of a real matrix the polynomial $P_A(\lambda)$ has real coefficients which implies that the non-real eigenvalues must occur in complex conjugate pairs.

[1] The fundamental theorem of algebra is a result of complex analysis.

## 2.5 A basis and a basis consisting of eigenvectors

For the $n$-dimensional space $\mathbb{R}^n$ (or $\mathbb{C}^n$) there is the standard orthonormal basis $\underline{e}_1, \cdots, \underline{e}_n$. Thus when we write $\underline{x} = (x_i) \in \mathbb{R}^n$ we mean

$$\underline{x} = \sum_1^n x_i \underline{e}_i \; . \tag{2.26}$$

For analysing many methods involving a matrix $A$ it is often convenient to use a basis consisting of eigenvectors of the matrix when this is possible. If $\lambda_1, \cdots, \lambda_n$ are the eigenvalues of $A$ and $\underline{v}_1, \cdots, \underline{v}_n$ are associated eigenvectors (which may be complex) then the question is when can $\underline{v}_1, \cdots, \underline{v}_n$ be chosen so that it is a basis? There is a precise (technical) answer to this question which is that this is possible when the geometric multiplicity of each distinct eigenvalue is the same as the algebraic multiplicity. To be much less precise the eigenvectors $\underline{v}_1, \cdots, \underline{v}_n$ are a basis for $\mathbb{C}^n$ in "nearly all cases" in the sense that $A$ must have "special properties" for it not to be a basis. The most satisfactory situation is when we restrict attention to the important case of real symmetric matrices as in this case the eigenvalues are all real and the eigenvectors can be chosen to be orthonormal. As these results are important to this module we state them as follows.

## 2.6 Eigenvalues and eigenvectors of real symmetric matrices

1. **All the eigenvalues are real**
   This can be proved as follows. If $\underline{v}$ is an eigenvector and $\lambda$ is the corresponding eigenvalue, which we must suppose at the outset can be complex, then

   $$A\underline{v} = \lambda \underline{v} \quad \text{and} \quad (\bar{\underline{v}}^T A \underline{v}) = \lambda(\bar{\underline{v}}^T \underline{v}), \tag{2.27}$$

   where $\bar{\underline{v}}$ denotes the complex conjugate of $\underline{v}$, i.e. $\bar{\underline{v}} = (\bar{v}_i)$. Now

   $$\bar{\underline{v}}^T \underline{v} = \sum_1^n |v_i|^2 > 0 \tag{2.28}$$

   and thus in particular this quantity is real. Also

   $$\bar{\underline{v}}^T A \underline{v} = \sum_{i=1}^n \sum_{j=1}^n a_{ij} \bar{v}_i v_j = \sum_{i=1}^n a_{ii} |v_i|^2 + \sum_{i=1}^n \sum_{j=1}^{i-1} a_{ij} (\bar{v}_i v_j + v_i \bar{v}_j) \tag{2.29}$$

   since $A$ is symmetric, i.e. $a_{ij} = a_{ji}$. Thus since $(\bar{v}_i v_j + v_i \bar{v}_j) = 2\text{Real}(\bar{v}_i v_j)$ is real it follows that

   $$\lambda = \frac{(\bar{\underline{v}}^T A \underline{v})}{(\bar{\underline{v}}^T \underline{v})} \tag{2.30}$$

   is real. As $\lambda$ is real the eigenvector $\underline{v}$ is also real.

2. **A real symmetric $n \times n$ matrix has $n$ orthonormal eigenvectors**
   The full proof is more appropriate in a Linear Algebra module and will not be given here but we will now show that it is relatively straight forward to show that eigenvectors corresponding to distinct eigenvalues must be orthogonal. If we have

   $$A\underline{v}_i = \lambda_i \underline{v}_i, \quad A\underline{v}_j = \lambda_j \underline{v}_j, \quad \text{with } \lambda_i \neq \lambda_j$$

then by the symmetry of $A$ we obtain

$$\underline{v}_j^T A \underline{v}_i = (\underline{v}_j^T A \underline{v}_i)^T = \underline{v}_i^T A^T \underline{v}_j = \underline{v}_i^T A \underline{v}_j \ .$$

The eigenvector/eigenvalue relations then imply that

$$\lambda_i \underline{v}_j^T \underline{v}_i = \underline{v}_j^T A \underline{v}_i = \underline{v}_i^T A \underline{v}_j = \lambda_j \underline{v}_i^T \underline{v}_j = \lambda_j \underline{v}_j^T \underline{v}_i \ .$$

Since $\lambda_i \neq \lambda_j$ we must have $\underline{v}_i^T \underline{v}_j = 0$.

We now consider some consequences of this orthonormal property of the eigenvectors.

If $\underline{v}_1, \cdots, \underline{v}_n$ denotes an orthonormal basis consisting of the eigenvectors corresponding respectively to the eigenvalues $\lambda_1, \cdots, \lambda_n$ then

$$A\underline{v}_i = \lambda_i \underline{v}_i, \quad i = 1, \cdots, n, \tag{2.31}$$

or in partitioned matrix form

$$\begin{aligned} A(\underline{v}_1, \underline{v}_2, \cdots, \underline{v}_n) &= (A\underline{v}_1, A\underline{v}_2, \cdots, A\underline{v}_n) = (\lambda_1 \underline{v}_1, \lambda_2 \underline{v}_2, \cdots, \lambda_n \underline{v}_n) && (2.32) \\ &= (\underline{v}_1, \underline{v}_2, \cdots, \underline{v}_n) \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix} . && (2.33) \end{aligned}$$

When the above is met for the first time it is natural to struggle with verifying the last equality (which we give again below in reverse form)

$$(\underline{v}_1, \underline{v}_2, \cdots, \underline{v}_n) \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix} = (\lambda_1 \underline{v}_1, \lambda_2 \underline{v}_2, \cdots, \lambda_n \underline{v}_n) \ . \tag{2.34}$$

However, by just doing the matrix multiplication of the left hand side will verify the result.

Now if we let

$$Q = (\underline{v}_1, \underline{v}_2, \cdots, \underline{v}_n) \quad \text{and} \quad D = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix} \tag{2.35}$$

then the above can be compactly written in the matrix form as

$$AQ = QD \ . \tag{2.36}$$

$\underline{v}_1, \underline{v}_2, \cdots, \underline{v}_n$ being an orthonormal basis implies that $Q$ is an orthogonal matrix and thus pre-multiplying $AQ = QD$ by $Q^T$ gives

$$Q^T A Q = D \tag{2.37}$$

which is an orthogonal similarity transformation. This can also be re-written as

$$A = QDQ^T = (\underline{v}_1, \cdots, \underline{v}_n) D \begin{pmatrix} \underline{v}_1^T \\ \vdots \\ \underline{v}_n^T \end{pmatrix} = (\lambda_1 \underline{v}_1, \lambda_2 \underline{v}_2, \cdots, \lambda_n \underline{v}_n) \begin{pmatrix} \underline{v}_1^T \\ \vdots \\ \underline{v}_n^T \end{pmatrix} \tag{2.38}$$

$$= \sum_1^n \lambda_i \underline{v}_i \, \underline{v}_i^T \, . \tag{2.39}$$

Thus $A$ can be expressed as a linear combination of the symmetric rank 1 matrices $\underline{v}_i \, \underline{v}_i^T$. There is a corresponding more complicated representation for general non-symmetric (and even non-square) matrices known as the singular value decomposition which will not be covered in this module.

## 2.7 Remarks about orthonormal vectors

If $\underline{v}_1, \underline{v}_2, \cdots, \underline{v}_n$ is a basis for $\mathbb{R}^n$ then any vector $\underline{x} \in \mathbb{R}^n$ can be written in the form

$$\underline{x} = \sum_1^n c_i \underline{v}_i, \tag{2.40}$$

for some real coefficients $c_1, \cdots, c_n$. For a general basis we have to solve a linear system to actually determine $c_1, \cdots, c_n$. However when $\underline{v}_1, \cdots, \underline{v}_n$ are orthonormal things simplify considerably since

$$\underline{x}^T \underline{v}_j = \underline{v}_j^T \underline{x} = \underline{v}_j^T \left( \sum_1^n c_i \underline{v}_i \right) = \sum_1^n c_i \underline{v}_j^T \underline{v}_i = c_j \, . \tag{2.41}$$

Orthonormal basis, or more generally orthogonal basis, are used in many areas of mathematics, compare for example Fourier series.

# 3 Vector and matrix norms

In this module there will be many occasions when we will use norms to describe the size or length of vectors and occasionally matrices. We use the usual notation $\|\underline{x}\|$ for the norm of a vector $\underline{x}$. Formally, a norm on $\mathbb{R}^n$ is a mapping $\|.\| : \mathbb{R}^n \to \mathbb{R}$ satisfying the following axioms.

(i) $\|\underline{x}\| \geq 0$ for all vectors $\underline{x} \in \mathbb{R}^n$ with $\|\underline{x}\| = 0$ if and only if $\underline{x} = \underline{0}$.

(ii) $\|\alpha \underline{x}\| = |\alpha| \|\underline{x}\|$ for all $\alpha \in \mathbb{R}$ and for all vectors $\underline{x} \in \mathbb{R}^n$.

(iii) $\|\underline{x} + \underline{y}\| \leq \|\underline{x}\| + \|\underline{y}\|$ for all vectors $\underline{x}$ and $\underline{y} \in \mathbb{R}^n$.

The most commonly used vector norm is the Euclidean norm or 2-norm

$$\|\underline{x}\|_2 = (x_1^2 + x_2^2 + \cdots + x_n^2)^{1/2} = \left( \sum_1^n x_i^2 \right)^{1/2} = \left( \underline{x}^T \underline{x} \right)^{1/2} . \tag{3.1}$$

We also have the 1-norm and $\infty$-norm defined by

$$\|\underline{x}\|_1 = \sum_1^n |x_i| \quad \text{and} \quad \|\underline{x}\|_\infty = \max_{1 \leq i \leq n} |x_i| . \tag{3.2}$$

Given any vector norm we can define a corresponding matrix norm which is said to be subordinate to it or to be induced by it. For $A \in \mathbb{R}^{n,n}$ we have

$$\|A\| = \max_{\|\underline{x}\|=1} \|A\underline{x}\| = \max_{\|\underline{y}\| \neq 0} \frac{\|A\underline{y}\|}{\|\underline{y}\|} . \tag{3.3}$$

By defining the norm in this way we have

$$\|A\underline{x}\| \leq \|A\| \|\underline{x}\| \tag{3.4}$$

for all vectors $\underline{x}$. From this it also follows that $A$ and $B$ are both $n \times n$ matrices then

$$\|AB\underline{x}\| \leq \|A\| \, \|B\underline{x}\| \leq \|A\| \, \|B\| \, \|\underline{x}\| \tag{3.5}$$

and we have

$$\|AB\| \leq \|A\| \, \|B\| \tag{3.6}$$

which verifies one of the axioms for a (submultiplicative) matrix norm. For the complete set axioms in this case we have the following:

A function $\|.\| : \mathbb{R}^{n \times n} \to \mathbb{R}$ is a matrix norm provided the following conditions are satisfied.

(i) $\|A\| \geq 0$ for all $A \in \mathbb{R}^{n \times n}$ with $\|A\| = 0$ if and only if $A = 0$. ($A = 0$ means $A$=zero matrix.) The non-negative condition.

(ii) $\|A + B\| \leq \|A\| + \|B\|$ for all $A, B \in \mathbb{R}^{n \times n}$. The triangle inequality condition.

(iii) $\|\alpha A\| = |\alpha| \, \|A\|$ for all scalars $\alpha \in \mathbb{R}$ and all $A \in \mathbb{R}^{n \times n}$. The linearity condition.

(iv) $\|AB\| \leq \|A\| \, \|B\|$ for all $A, B \in \mathbb{R}^{n \times n}$. The **sub-multiplicative condition**.

There are certain results relating norms with the eigenvalues of the matrix. We give first an inequality and then representations of particular matrix norms.

From the eigenvalue-eigenvector statement $A\underline{v} = \lambda\underline{v}$, $\underline{v} \neq \underline{0}$ we have

$$|\lambda| \|\underline{v}\| = \|A\underline{v}\| \leq \|A\| \|\underline{v}\|, \quad \text{i.e. } |\lambda| \leq \|A\| \; . \tag{3.7}$$

Thus for all subordinate norms,

$$\begin{aligned} \rho(A) &= \text{spectral radius of } A && (3.8)\\ &:= \text{the largest eigenvalue of } A \text{ in magnitude} \leq \|A\| \; . && (3.9) \end{aligned}$$

Next we derive explicit representation in the case of the norms $\|A\|_2$ and $\|A\|_\infty$ and we state the result in the case of $\|A\|_1$.

In the case of the matrix 2-norm we have for the subordinate matrix norm

$$\|A\|_2^2 = \max_{\|\underline{x}\|_2=1} \|A\underline{x}\|_2^2 = \max_{\|\underline{x}\|_2=1} (A\underline{x})^T(A\underline{x}) = \max_{\|\underline{x}\|_2=1} \underline{x}^T A^T A \underline{x} \; . \tag{3.10}$$

To obtain the result you need to note that $A^TA$ is a symmetric positive semi-definite matrix having orthonormal eigenvectors $\underline{v}_1, \cdots, \underline{v}_1$ with corresponding non-negative eigenvalues $\lambda_1, \cdots, \lambda_n$, i.e. $A^TA\underline{v}_i = \lambda_i \underline{v}_i$ and $\underline{v}_i^T \underline{v}_i = 1$. Let $\underline{x}$ be a vector such that $\underline{x}^T\underline{x} = 1$. We have

$$\begin{aligned} \underline{x} &= \sum_1^n (\underline{x}^T \underline{v}_i)\underline{v}_i, && (3.11)\\ A\underline{x} &= \sum_1^n (\underline{x}^T \underline{v}_i)\lambda_i\underline{v}_i, && (3.12)\\ 1 = \underline{x}^T\underline{x} &= \sum_1^n (\underline{x}^T \underline{v}_i)^2, && (3.13)\\ \underline{x}^T A^T A \underline{x} &= \sum_1^n (\underline{x}^T \underline{v}_i)^2 \lambda_i \leq \max_i \lambda_i \; . && (3.14) \end{aligned}$$

The maximum is attained with the 'dominant' eigenvector, i.e. the eigenvector corresponding to the largest eigenvalue. Hence the result is

$$\|A\|_2^2 = \max_i \lambda_i = \rho(A^TA) = \text{spectral radius of } A^TA. \tag{3.15}$$

If the matrix $A$ is symmetric and positive definite then this simplifies to

$$\|A\|_2 = \rho(A) = \text{the largest eigenvalue of } A, \tag{3.16}$$

i.e. the largest eigenvalue is the 2-norm in this case.

In the case of $\|A\|_\infty$ we have

$$\|A\|_\infty = \max_{\|\underline{x}\|_\infty=1} \|A\underline{x}\|_\infty, \quad \|A\underline{x}\|_\infty = \max_i |(A\underline{x})_i| \quad \text{and} \quad (A\underline{x})_i = \sum_{j=1}^n a_{ij}x_j \; . \tag{3.17}$$

Let $\underline{x}$ be a vector with $\|\underline{x}\|_\infty = 1$ and suppose that the index $m$ is such that $|x_m| = \|\underline{x}\|_\infty$. By the triangle inequality and that $|x_j| \leq 1$ we have

$$|(A\underline{x})_i| = \left| \sum_{j=1}^n a_{ij}x_j \right| \leq \sum_{j=1}^n |a_{ij}| \leq \max_i \sum_{j=1}^n |a_{ij}| \; . \tag{3.18}$$

If the maximum row sum of absolute values occurs on row $p$ then

$$(A\underline{x})_p = \sum_{j=1}^{n} a_{pj} x_j = \sum_{j=1}^{n} |a_{pj}| \tag{3.19}$$

if we take the vector $\underline{x}$ so that

$$x_j = \begin{cases} |a_{pj}|/a_{pj}, & \text{if } a_{pj} \neq 0, \\ 1, & \text{otherwise,} \end{cases} \tag{3.20}$$

then this vector $\underline{x}$ is such that $\|\underline{x}\|_\infty = 1$ and with this vector we attain the bound. Hence we have shown that

$$\|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}| = \text{max. row sum of absolute values.} \tag{3.21}$$

In the case of the 1-norm a similar argument shows that

$$\|A\|_1 = \text{max column sum of absolute values} = \max_{1\le j\le n} \sum_{i=1}^{n} |a_{ij}| \,. \tag{3.22}$$

There is also a commonly used matrix norm which is not subordinate to any vector norm known as the Euclidean matrix norm of Frobenius norm which is given by

$$\|A\|_E = \left( \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^2 \right)^{1/2} \,. \tag{3.23}$$

This norm is consistent with the matrix 2-norm.

# 4 Direct methods for solving linear systems

In this section we consider a class of methods for attempting to solve the system of linear equations

$$A\underline{x} = \underline{b}, \tag{4.1}$$

where $A$ is an $n \times n$ matrix and $\underline{b} \neq \underline{0}$ is an $n \times 1$ column vector. This part of the module is concerned with the class of methods known as **direct methods** which generally means Gauss elimination in some form. Another type of methods to be considered later is known as **iterative methods** . Roughly speaking direct methods are methods which give you the exact solution $\underline{x}$ (when it exists) in a finite number of steps when exact arithmetic is used. In practice however computer floating point arithmetic is used and thus only an approximation (usually an accurate approximation) is obtained. Iterative methods are concerned with generating a sequence of vectors $\underline{x}^{(1)}$, $\underline{x}^{(2)}$, $\cdots$ which hopefully get closer and closer to $\underline{x}$. With most iterative methods the exact answer is not achieved in a finite number of steps but it is sometimes possible to get a "sufficiently accurate" approximation to $\underline{x}$ faster than with direct methods. Iterative methods are usually used for sparse systems and often require considerably less storage than do direct methods when implemented on a computer.

## 4.1 Basic idea of direct methods

The basic idea of most direct methods is to systematically transform the system $A\underline{x} = \underline{b}$ into an equivalent system (i.e. a system with the same solution) $S\underline{x} = \hat{\underline{b}}$ which is "easy" to solve. In practice this means changing the system to an equivalent triangular system, i.e. with $S$ as an upper or a lower triangular matrix. Thus we begin by considering how to solve a triangular system.

## 4.2 Triangular systems of equations

Let $U = (u_{ij})$ be an upper triangular matrix and consider the system

$$U\underline{x} = \underline{b} \tag{4.2}$$

or in full

$$\begin{array}{ccccccccc} u_{11}x_1 & + & u_{12}x_2 & + & \cdots & + & u_{1n}x_n & = & b_1 \\ & + & u_{22}x_2 & + & \cdots & + & u_{2n}x_n & = & b_2 \\ & & \ddots & & & & & & \vdots \\ & & & & u_{n-1,n-1}x_{n-1} & + & u_{n-1,n}x_n & = & b_{n-1} \\ & & & & & & u_{nn}x_n & = & b_n \ . \end{array} \tag{4.3}$$

This is solved by **back substitution** , i.e. we get $x_n$ then $x_{n-1}$ then $\cdots$ and finally $x_1$ by doing the following.

$$x_n = b_n/u_{nn} \tag{4.4}$$

$$x_i = \left( b_i - \sum_{k=i+1}^{n} u_{ik}x_k \right) / u_{ii}, \quad i = n-1, \cdots, 1 \ . \tag{4.5}$$

For this to work we require that the diagonal entries $u_{ii} \neq 0$ for $i = 1, 2, \cdots, n$. To calculate $x_i$ involves $(n-i)$ multiplications, 1 division and $(n-i)$ additions/subtractions. If you do not distinguish between the operations then the total number of such operations is

$$\sum_{i=1}^{n}(1+2(n-i)) = (2n-1)+(2n-3)+\cdots+5+3+1 \quad (4.6)$$

$$= \frac{1}{2}(2n+2n+\cdots+2n) = n^2 \ . \quad (4.7)$$

On a computer such operations are known as floating point operations (flops) and thus back substitution involves $n^2$ flops. As a comparison, multiplying a full matrix $A$ by a vector $\underline{x}$ involves $2n^2$ flops.

Lower triangular systems are similarly solved by forward substitution and have the same operation count.

## 4.3 Reduction to triangular form by Gauss elimination

The basic operation in Gauss elimination is the subtraction of a multiple of one equation from another in order to eliminate certain unknowns. In its basic form (assuming no pivoting is necessary) it involves using the first equation to eliminate the first unknown $x_1$ from the other $n-1$ equations. The new second equation is then used to eliminate $x_2$ from equations 3 to $n$ and continuing in this way the $k$th step involves using the current $k$th equation to eliminate $x_k$ from equations $k+1$ to $n$. To distinguish between the different equivalent systems we will use the superscript $(k)$ for the $k$th system, i.e. at the end of the $k$th step we have

$$A^{(k)}\underline{x} = \underline{b}^{(k)} \quad (4.8)$$

and at the start $A^{(0)} = A$, $\underline{b}^{(0)} = \underline{b}$. In full the initial system is

$$\begin{array}{ccccccc} a_{11}^{(0)}x_1 & + & \cdots & + & a_{1n}^{(0)}x_n & = & b_1^{(0)} \\ a_{21}^{(0)}x_1 & + & \cdots & + & a_{2n}^{(0)}x_n & = & b_2^{(0)} \\ \vdots & & & & & & \\ a_{n1}^{(0)}x_1 & + & \cdots & + & a_{nn}^{(0)}x_n & = & b_n^{(0)} \ . \end{array} \quad (4.9)$$

After step 1 we have

$$\begin{array}{ccccccccc} a_{11}^{(0)}x_1 & + & a_{12}^{(0)}x_2 & + & \cdots & + & a_{1n}^{(0)}x_n & = & b_1^{(0)} \qquad \text{(no change in equation 1)} \\ & & a_{22}^{(1)}x_2 & + & \cdots & + & a_{2n}^{(1)}x_n & = & b_2^{(1)} \\ & & a_{32}^{(1)}x_2 & + & \cdots & + & a_{3n}^{(1)}x_n & = & b_3^{(1)} \\ & & \vdots & & & & & & \\ & & a_{n2}^{(1)}x_2 & + & \cdots & + & a_{nn}^{(1)}x_n & = & b_n^{(1)} \ . \end{array} \quad (4.10)$$

After $k-1$ steps we have

$$\begin{array}{ccccccccc} a_{11}^{(0)}x_1 & + & a_{12}^{(0)}x_2 & + & \cdots & + & a_{1n}^{(0)}x_n & = & b_1^{(0)} \\ & \ddots & & & & & & & \\ & & a_{kk}^{(k-1)}x_k & + & \cdots & + & a_{kn}^{(k-1)}x_n & = & b_k^{(k-1)} \\ & & a_{k+1,k}^{(k-1)}x_k & + & \cdots & + & a_{k+1,n}^{(k-1)}x_n & = & b_{k+1}^{(k-1)} \\ & & \vdots & & & & & & \\ & & a_{nk}^{(k-1)}x_k & + & \cdots & + & a_{nn}^{(k-1)}x_n & = & b_n^{(k-1)} \ . \end{array} \quad (4.11)$$

We must describe the $k$th step. We use the current $k$th equation to eliminate $x_k$ from the current equations $k+1$ to $n$. The $k$th equation is known as the **pivotal equation**, $a_{kk}^{(k-1)}$ is the **pivot element** and

$$m_{ik} = \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}, \quad i = k+1, \cdots, n \tag{4.12}$$

are the **multipliers**. We have, in terms of equations,

$$(\text{Equation } i)^{(k)} = (\text{Equation } i)^{(k-1)} - m_{ik}(\text{Equation } k)^{(k-1)}, \quad i = k+1, \cdots, n\ . \tag{4.13}$$

In components

$$\begin{aligned} a_{ij}^{(k)} &= a_{ij}^{(k-1)} - m_{ik} a_{kj}^{(k-1)}, & j &= k+1, \cdots, n, \\ & & i &= k+1, \cdots, n, \\ b_i^{(k)} &= b_i^{(k-1)} - m_{ik} b_k^{(k-1)}, & i &= k+1, \cdots, n. \end{aligned} \tag{4.14}$$

The elimination is complete at the end of step $n-1$ as $A^{(n-1)}$ is upper triangular. If you count the number of floating point operations (flops) you find that this requires the order of $2n^3/3$ flops which we write as $\mathcal{O}(2n^3/3)$. The $n^3$ term arises since we have 3 nested looping operations to perform (the outer $k$ loop and the inner $i$ and $j$ loops). The subsequent back substitution to actually obtain $\underline{x}$ involves $n^2$ flops. Hence for large $n$ the reduction part involves by far the most computation.

## 4.4 Pivoting and stability

As given the basic algorithm fails if any of the pivot elements $a_{kk}^{(k-1)} = 0$, e.g. it fails for the perfectly conditioned[2] system

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} . \tag{4.15}$$

The method is numerically unstable if any of the pivot elements $a_{kk}^{(k-1)}$ are very small, e.g. a system of the form

$$\begin{pmatrix} \epsilon & 1 & 1 \\ 1 & \epsilon & 1 \\ 1 & 1 & \epsilon \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2+\epsilon \\ 2+\epsilon \\ 2+\epsilon \end{pmatrix}, \tag{4.16}$$

where $\epsilon$ is "very small" which has solution $\underline{x} = (1,1,1)^T$. After 1 step of basic Gauss elimination we have the equivalent system

$$\begin{pmatrix} \epsilon & 1 & 1 \\ 0 & \epsilon - \frac{1}{\epsilon} & 1 - \frac{1}{\epsilon} \\ 0 & 1 - \frac{1}{\epsilon} & \epsilon - \frac{1}{\epsilon} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2+\epsilon \\ 2+\epsilon - \dfrac{(2+\epsilon)}{\epsilon} \\ 2+\epsilon - \dfrac{(2+\epsilon)}{\epsilon} \end{pmatrix} . \tag{4.17}$$

If $\epsilon$ is very small in magnitude then the $2 \times 2$ system for $x_2$ and $x_3$ is very close to the consistent but singular system

$$-\frac{1}{\epsilon} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} = -\frac{2}{\epsilon} \begin{pmatrix} 1 \\ 1 \end{pmatrix} . \tag{4.18}$$

Latter steps of the Gauss elimination procedure would then either fail (due to division by zero) or it would proceed but lead to the wrong final answer depending the precise rounding error that takes place during the computations.

[2] The condition of a matrix is defined in another section of this module. Roughly speaking a perfectly conditioned system is one which is "as far as possible from a singular system".

The failure in the above example can be overcome by using a technique known as pivoting. Roughly this involves re-ordering the equations in order to put as large as possible an entry in the pivot position. If this is precisely what is done then this is known as **partial pivoting**. A variant of this is **row scaled partial pivoting** which is the technique usually used. Another technique of this type is **complete pivoting** which involves possibly re-ordering both the equations and the unknowns. This is the most stable pivoting algorithm but as it involves a significant increase in the computation, which is rarely necessary, it is not usually used in practice.

Pivoting is important for constructing an algorithm which is "stable in practice" for a general system but it is not something which will be considered in any more detail in these notes. For certain special systems to be considered later (e.g. involving symmetric positive definite matrices) we will show that pivoting is not necessary.

## 4.5 A matrix description of basic Gauss elimination

The $k$th step of Gauss elimination involves subtracting a multiple of the $k$th equation from equations $k+1, \cdots, n$ in the system

$$A^{(k-1)}\underline{x} = \underline{b}^{(k-1)} . \tag{4.19}$$

Now

$$\underline{e}_k{}^T A^{(k-1)} = k\text{th row of } A^{(k-1)} \tag{4.20}$$

and

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ m_{k+1,k} \\ \vdots \\ m_{nk} \end{pmatrix} \underline{e}_k{}^T A^{(k-1)} = \text{matrix with each row a multiple of the } k\text{th row of } A^{(k-1)}. \tag{4.21}$$

Define

$$\underline{m}_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ m_{k+1,k} \\ \vdots \\ m_{nk} \end{pmatrix} . \tag{4.22}$$

Then

$$\begin{aligned} A^{(k)} &= A^{(k-1)} - \underline{m}_k\, \underline{e}_k{}^T A^{(k-1)} = (I - \underline{m}_k\, \underline{e}_k{}^T)A^{(k-1)} && (4.23) \\ &= M_k A^{(k-1)}, && (4.24) \end{aligned}$$

where

$$M_k = I - \underline{m}_k\, \underline{e}_k{}^T \tag{4.25}$$

is a **Gauss transformation matrix**. Thus if we let $U = A^{(n-1)}$ be the upper triangular matrix at the end of the Gauss elimination procedure then we have

$$U = A^{(n-1)} = M_{n-1}M_{n-2}\cdots M_1 A \tag{4.26}$$

which rearranges to

$$A = (M_{n-1}\cdots M_1)^{-1}U = M_1^{-1}\cdots M_{n-1}^{-1}U = LU \tag{4.27}$$

where

$$L = M_1^{-1}\cdots M_{n-1}^{-1}\ . \tag{4.28}$$

## 4.6 Some properties of $M_k$ and $L$

1.

$$M_k = I - \underline{m}_k\underline{e}_k{}^T = \begin{pmatrix} 1 & \cdots & & & 0 \\ \vdots & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ 0 & & & & 1 \end{pmatrix} - \begin{pmatrix} 0 & \cdots & & & 0 \\ \vdots & \ddots & & & \\ & & 0 & & \\ & & * & \ddots & \\ 0 & & * & & 0 \end{pmatrix} \tag{4.29}$$

where $*$ indicates a possible non-zero entry. Thus $M_k$ is **unit lower triangular**.

2.

$$M_k^{-1} = (I - \underline{m}_k\underline{e}_k{}^T)^{-1} = I + \underline{m}_k\underline{e}_k{}^T\ . \tag{4.30}$$

This follows by considering the product

$$(I - \underline{m}_k\underline{e}_k{}^T)(I + \underline{m}_k\underline{e}_k{}^T) = I - \underline{m}_k\underline{e}_k{}^T\underline{m}_k\underline{e}_k{}^T = I \tag{4.31}$$

since $\underline{e}_k{}^T\underline{m}_k = (k\text{th component of } \underline{m}_k) = 0$.

3.

$$L = M_1^{-1}\cdots M_{n-1}^{-1} = I + \underline{m}_1\underline{e}_1{}^T + \cdots + \underline{m}_{n-1}\underline{e}_{n-1}{}^T \tag{4.32}$$

is **unit lower triangular**. This can be proved by induction by showing that

$$M_1^{-1}\cdots M_{r-1}^{-1} = I + \underline{m}_1\underline{e}_1{}^T + \cdots + \underline{m}_{r-1}\underline{e}_{r-1}{}^T\ . \tag{4.33}$$

It is true when $r = 2$ by part (2). Suppose it is true for $r = k$, i.e.

$$M_1^{-1}\cdots M_{k-1}^{-1} = I + \underline{m}_1\underline{e}_1{}^T + \cdots + \underline{m}_{k-1}\underline{e}_{k-1}{}^T\ . \tag{4.34}$$

Then

$$\begin{aligned} M_1^{-1}\cdots M_k^{-1} &= \left(M_1^{-1}\cdots M_{k-1}^{-1}\right)M_k^{-1} && (4.35)\\ &= \left(I + \underline{m}_1\underline{e}_1{}^T + \cdots + \underline{m}_{k-1}\underline{e}_{k-1}{}^T\right)\left(I + \underline{m}_k\underline{e}_k{}^T\right) && (4.36)\\ &= I + \underline{m}_1\underline{e}_1{}^T + \cdots + \underline{m}_k\underline{e}_k{}^T && (4.37)\\ &\qquad + \left(\underline{m}_1\underline{e}_1{}^T + \cdots + \underline{m}_{k-1}\underline{e}_{k-1}{}^T\right)\underline{m}_k\underline{e}_k{}^T\ . && (4.38) \end{aligned}$$

The last term vanishes since

$$\underline{e}_1{}^T\underline{m}_k = \cdots = \underline{e}_{k-1}{}^T\underline{m}_k = 0 \tag{4.39}$$

as the first $k$ components of $\underline{m}_k$ are zero.

In the above it has been shown that Gauss elimination is equivalent to an $LU$ factorization of the matrix $A$ with $L$ as unit lower triangular and $U$ as upper triangular. The multipliers used in Gauss elimination are precisely the strictly lower triangular entries of $L$. Once the factors $L$ and $U$ have been computed so that $A = LU$ we have

$$A\underline{x} = LU\underline{x} = \underline{b} \ . \tag{4.40}$$

To solve for $\underline{x}$ we have

$$L\underline{y} = \underline{b} \qquad \text{(solve for } \underline{y} \text{ by forward substitution)} \tag{4.41}$$

and

$$U\underline{x} = \underline{y} \qquad \text{(solve for } \underline{x} \text{ by back substitution).} \tag{4.42}$$

A point to note is that to solve $A\underline{x} = \underline{b}$ (the solution being $\underline{x} = A^{-1}\underline{b}$) we do not need to compute $A^{-1}$. Indeed $A^{-1}$ is usually computed by first computing $A = LU$ and then solving $A\underline{x} = \underline{e}_i$ for $i = 1, 2, \cdots, n$ by forward and back substitution.

We now consider such factorizations. First we consider the existence and uniqueness and for this we require the following Lemma.

**Lemma.**

(1) If $U$ is upper triangular and non-singular then $U^{-1}$ is also upper triangular. Similarly for lower triangular matrices.

(2) If $L$ is unit lower triangular then $L^{-1}$ is also unit lower triangular. Similarly for unit upper triangular matrices.

(3) The product of 2 upper triangular matrices is upper triangular. Similarly for lower triangular matrices. The product of 2 unit lower triangular matrices is unit lower triangular. Similarly for unit upper triangular matrices.

**Proof.**

(1) The solution of $U\underline{x} = \underline{e}_i$ is $\underline{x} = U^{-1}\underline{e}_i$ $=i$th column of $U^{-1}$. In full this linear system is

$$\begin{array}{ccccccccc} u_{11}x_1 & + & u_{12}x_2 & + & \cdots & + & u_{1n}x_n & = & 0 \\ & & \cdots & & \cdots & & \vdots & & \vdots \\ & & u_{ii}x_i & + & \cdots & + & u_{in}x_n & = & 1 \\ & & & & \ddots & & & & \vdots \\ & & & & & & u_{nn}x_n & = & 0 \end{array} \tag{4.43}$$

By considering equations $n$, $n-1$, $\cdots$, $i+1$, back substitution gives $x_n = 0$, $x_{n-1} = 0$, $\cdots$, $x_{i+1} = 0$. Then

$$x_i = 1/u_{ii}, \tag{4.44}$$

i.e. the $i$th column of $U^{-1}$ (which is the vector $U^{-1}\underline{e}_i$) is zero below the diagonal.

(2) The proof of part (2) is similar to the proof of part (1).

(3) If $A$ and $B$ are upper triangular then

$$(AB)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} = \sum_{k=i}^{j} a_{ik} b_{kj} = 0 \quad \text{if } i > j. \tag{4.45}$$

Thus $AB$ is upper triangular. All other possibilities follow similarly. □

## 4.7 Uniqueness of the $LU$ factorization

**Theorem.** If $A$ is non-singular and

$$A = L_1 U_1 = L_2 U_2 \tag{4.46}$$

with $L_1$ and $L_2$ as unit lower triangular matrices and $U_1$ and $U_2$ as upper triangular matrices then

$$L_1 = L_2 \quad \text{and} \quad U_1 = U_2 \ . \tag{4.47}$$

**Proof.** The first thing to note is that $L_1$, $U_1$, $L_2$ and $U_2$ are all non-singular. This follows from the relations

$$\det L_1 = 1, \quad \det L_2 = 1, \tag{4.48}$$

and

$$0 \neq \det A = (\det L_1)(\det U_1) = \det U_1, \quad 0 \neq \det A = (\det L_2)(\det U_2) = \det U_2 \ . \tag{4.49}$$

Thus rearranging $L_1 U_1 = L_2 U_2$ gives

$$L_2^{-1} L_1 = U_2 U_1^{-1} \ . \tag{4.50}$$

By the Lemma the left hand side is unit lower triangular and the right hand side is upper triangular. The only possibility for both of these to be true is when

$$L_2^{-1} L_1 = U_2 U_1^{-1} = I, \quad \text{i.e. } L_2 = L_1, \quad U_2 = U_1 \ .\square \tag{4.51}$$

## 4.8 Existence of the $LU$ factorization

Since the $LU$ factorization of $A$ is equivalent to the Gauss elimination procedure the factorization exists provided zero pivots are not encountered during the elimination. As we will show a zero pivot corresponds to a principal sub-matrix of $A$ being singular.

**Definition.** The $k \times k$ principal sub-matrix of $A$ is the $k \times k$ matrix

$$\begin{pmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kk} \end{pmatrix} . \tag{4.52}$$

If $A = LU$ is an $LU$ factorization then for $1 \leq i, j \leq n$

$$a_{ij} = \sum_{r=1}^{\min(i,j)} l_{ir} u_{rj} \ . \tag{4.53}$$

In particular if we restrict $i, j$ so that $1 \le i, j \le k$ then

$$(A_k)_{ij} = a_{ij} = \sum_{r=1}^{\min(i,j)} l_{ir} u_{rj} = (L_k U_k)_{ij} \tag{4.54}$$

where $L_k$ and $U_k$ are the $k \times k$ leading principal sub-matrices of $L$ and $U$ respectively. This shows that

$$A_k = L_k U_k, \quad k = 1, 2, \cdots, n \ . \tag{4.55}$$

Observe that

$$\det A_k = (\det L_k)(\det U_k) = \det U_k = u_{11} u_{22} \cdots u_{kk} \tag{4.56}$$

and recall that the diagonal entries of the upper triangular matrix generated by the elimination process are precisely the pivots used. Thus if all pivots encountered are non-zero then

$$\det A_k \ne 0, \quad k = 1, \cdots, n-1 \ . \tag{4.57}$$

If the $k$th pivot is zero (i.e. $u_{kk} = 0$) then the $k$th principal sub-matrix is singular and the elimination process breaks down at this point.

Observe that to have a solution to the system we only require that $A$ is non-singular, i.e. $\det A = \det A_n \ne 0$ but for the basic Gauss elimination procedure to work we also require that equation (4.57) is true. These extra conditions are removed when a pivoting procedure is used.

## 4.9 Obtaining the $LU$ factorization directly

If $A = LU$ with $L = (l_{ij})$ unit lower triangular and $U = (u_{ij})$ upper triangular then this requires that

$$a_{ij} = (i\text{th row of } L) \times (j\text{th column of } U) \tag{4.58}$$

$$= \sum_{r=1}^{\min(i,j)} l_{ir} u_{rj} = \begin{cases} \left( \sum_{1}^{i-1} l_{ir} u_{rj} \right) + u_{ij}, & \text{if } i \le j, \\ \left( \sum_{1}^{j-1} l_{ir} u_{rj} \right) + l_{ij} u_{jj}, & \text{if } j < i. \end{cases} \tag{4.59}$$

Thus rearranging

$$u_{ij} = a_{ij} - \sum_{r=1}^{i-1} l_{ir} u_{rj}, \quad \text{for } i \le j, \tag{4.60}$$

$$l_{ij} = \left( a_{ij} - \sum_{r=1}^{j-1} l_{ir} u_{rj} \right) / u_{jj} \quad \text{for } j < i \ . \tag{4.61}$$

Since the terms are the same as in the Gauss elimination procedure we use the knowledge of the Gauss elimination algorithm to determine the order in which these are to be computed. In Gauss elimination row $k$ of $U$ is complete when $A^{(k-1)}$ is determined. Then column $k$ of $L$ (the multipliers) is computed as one of the steps in computing $A^{(k)}$. The $k$th row of $U$ and the $k$th column of $L$ are

$$u_{kj} = a_{kj} - \sum_{r=1}^{k-1} l_{kr} u_{rj}, \quad \text{for } j = k, \cdots, n, \tag{4.62}$$

$$l_{ik} = \left( a_{ik} - \sum_{r=1}^{j-1} l_{ir} u_{rk} \right) / u_{kk}, \quad \text{for } i = k+1, \cdots, n \ . \tag{4.63}$$

## 4.10 Remarks on the computer implementation

The $LU$ factorization algorithm involves doing the above for $k = 1, 2, \cdots, n$. This involves the same operations as Gauss elimination but with a different order. Gauss elimination involved many operations of the form

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - m_{ik} a_{kj}^{(k-1)}, \quad (m_{ik} = l_{ik}), \tag{4.64}$$

i.e. part of the computation of $u_{kj}$ and $l_{ik}$ is done at each of the elimination steps prior to step $k$. The $LU$ factorization idea is to wait until the computation can be done in one go. This does not involve any less flops but it is the preferred way of doing the operation on a computer because it can better exploit the way a computer actually performs its floating point operations. A computation of the form

$$\sum_{1}^{m} y_i z_i = y_1 z_1 + \cdots + y_m z_m \tag{4.65}$$

(known technically as a **BLAS 1 operation**[3]) can be done efficiently and accurately by using the **double accumulation of inner products**. Roughly this involves the following. If the mantissa of numbers stored on a computer involve $t$ binary digits[4] then the mantissa of a product $y_i z_i$ involves $2t$ binary digits for an exact representation. This is done internally in the computers registers but then this is usually truncated back to $t$ digits which gives a floating point error. However if the $2t$ digits can be kept in the register then we can successively compute

$$y_1 z_1, \quad y_1 z_1 + y_2 z_2, \quad \cdots, \quad y_1 z_1 + \cdots + y_m z_m \tag{4.66}$$

to $2t$ binary digit accuracy with the truncation to $t$ digits done only once at the end. This is thus faster and more accurate.

## 4.11 Remarks on pivoting

Mathematically partial pivoting corresponds to an $LU$ factorization of a matrix of the form

$$PA = LU, \tag{4.67}$$

where $P$ is a permutation matrix (the columns of $P$ are a re-arrangement of the columns of the identity matrix or equivalently the rows of $P$ are a re-arrangement of the rows of the identity matrix ). The matrix $PA$ is the matrix $A$ with the rows re-ordered which in the linear equations case $A\underline{x} = \underline{b}$ corresponds to rearranging the equations. Complete pivoting corresponds to a factorization

$$PAQ = LU, \tag{4.68}$$

where $P$ and $Q$ are both permutation matrices. When Gauss elimination or the equivalent $LU$ factorization is implemented with pivoting the re-arrangements described by $P$ (and $Q$ in the case of complete pivoting) are determined during the process.

## 4.12 Special linear systems, e.g. symmetric positive definite systems

With general linear systems it is essential that some sort of pivoting is used in order to be confident of generating a reasonably accurate approximation to systems of equations that are not badly

[3] BLAS stands for Basic Linear Algebra Software

[4] A number has a sign (+ or −) a mantissa and an exponent. In decimal form the FORTRAN number `-0.571E+07` has the sign −, `0.571` corresponds to the mantissa part and `E+07` corresponds to the exponent part.

conditioned[5]. However, in many applications the systems which arise often have some special structure which enables the $LU$ factorization to be performed in a stable manner without pivoting. In addition the systems sometimes also have a sparse banded structure which implies that $L$ and $U$ also have the sparse banded form provided pivoting is not done. (If pivoting is done for such a matrix then the bandwidths of the resulting $L$ and $U$ are greater than that of the original matrix $A$.)

We consider next the special case of a certain sparse matrix which has a sparse factorization by considering specifically the solution of a tri-diagonal system assuming that pivoting is not required. We then consider the case of a linear system with a real symmetric and positive definite matrix which can always be factorized without pivoting in a "symmetric way" in what is called as the Cholesky factorization.

### 4.12.1 A Tri-diagonal system

A tri-diagonal system involves a tri-diagonal matrix which is a matrix for which all entries are 0 except on the diagonal and on the sub- and super- diagonals. It is customarily in such situations to adopt a one-subscript notation ($a_i$, $d_i$ and $b_i$) for the entries instead of the double subscript notation $a_{ij}$. Thus we will write the system as

$$\begin{aligned} d_1 x_1 + b_2 x_2 &= r_1 && (4.69)\\ a_i x_{i-1} + d_i x_i + b_{i+1} x_{i+1} &= r_i, \quad 2 \le i \le n-1, && (4.70)\\ a_n x_{n-1} + d_n x_n &= r_n && (4.71)\\ & && (4.72) \end{aligned}$$

or as

$$\begin{pmatrix} d_1 & b_2 & \cdots & & & 0 \\ a_2 & d_2 & b_3 & & & \vdots \\ \vdots & a_3 & d_3 & b_4 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & b_n \\ 0 & \cdots & & & a_n & d_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ \vdots \\ r_{n-1} \\ r_n \end{pmatrix} . \tag{4.73}$$

The first step in Gauss elimination involves eliminating below the diagonal in column 1 and thus in this case we need only subtract a multiple of equation 1 from equation 2 to produce the modified equation

$$d_2' x_2 + b_3 x_3 = r_2', \quad l_2 = a_2/d_1,\ d_2' = d_2 - l_2 b_2,\ r_2' = r_2 - l_2 r_1 . \tag{4.74}$$

One step of Gauss elimination (to create $d_2'$ and $r_2'$ in this case) hence just involves 1 division, 2 multiplications and 2 subtractions. Continuing this process to eliminate below the diagonal in column 2, then in column 3, then in $\cdots$ and finally in column $n-1$ similarly gives

$$\begin{array}{lll} l_3 = a_3/d_2', & d_3' = d_3 - l_3 b_3, & r_3' = r_3 - l_3 r_2, \\ l_4 = a_4/d_3', & d_4' = d_4 - l_4 b_4, & r_4' = r_4 - l_4 r_3. \\ \cdots & \cdots & \cdots \end{array}$$

With $d_1' = d_1$ and $r_1' = r_1$ we hence have

$$l_k = a_k/d_{k-1}', \quad d_k' = d_k - l_k b_k, \quad r_k' = r_k - l_k r_{k-1}' \quad k = 2, 3, \cdots, n \tag{4.75}$$

[5]The term "badly conditioned" is defined in another section of this module. Roughly speaking a system is badly conditioned if the corresponding matrix is "nearly singular". When a system is nearly singular it is difficult to solve accurately.

In matrix form we have created the upper triangular system

$$\begin{pmatrix} d_1' & b_2 & \cdots & & & 0 \\ & d_2' & b_3 & & & \vdots \\ & & d_3' & b_4 & & \\ & & & \ddots & \ddots & \\ & & & & \ddots & b_n \\ 0 & \cdots & & & & d_n' \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} r_1' \\ r_2' \\ \vdots \\ \vdots \\ r_{n-1}' \\ r_n' \end{pmatrix} \tag{4.76}$$

which can then be solved by back substitution. We simply have

$$x_n = r_n'/d_n' \tag{4.77}$$

$$x_i = (r_i' - b_{i+1}x_{i+1})/d_i' \quad \text{for } i = n-1, n-2, \cdots, 1. \tag{4.78}$$

The point to note is that the storage is $\mathcal{O}(n)$ and the number of operations to obtain the solution is also $\mathcal{O}(n)$ in contrast to a general system which has respectively $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$.

Another way of expressing what has just been done is in terms of the $LU$ factorization of the tri-diagonal matrix. Since Gauss elimination without pivoting is the same as an $LU$ factorization we have that

$$\begin{pmatrix} d_1 & b_2 & \cdots & & & 0 \\ a_2 & d_2 & b_3 & & & \vdots \\ \vdots & a_3 & d_3 & b_4 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & b_n \\ 0 & \cdots & & & a_n & d_n \end{pmatrix} = \begin{pmatrix} 1 & & & & & 0 \\ l_2 & 1 & & & & \\ \vdots & l_3 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & & \\ 0 & \cdots & & & l_n & 1 \end{pmatrix} \begin{pmatrix} d_1' & b_2 & \cdots & & & 0 \\ \vdots & d_2' & b_3 & & & \vdots \\ & & d_3' & b_4 & & \\ & & & \ddots & \ddots & \\ & & & & \ddots & b_n \\ 0 & \cdots & & & & d_n' \end{pmatrix}. \tag{4.79}$$

A procedure which efficiently stores and solves tri-diagonal systems is already implemented in MATLAB. If `a`, `d`, `b` and `r` in MATLAB correspond respectively to the vectors $\underline{a}$, $\underline{d}$, $\underline{b}$ and $\underline{r}$ and `n` has already been set then in MATLAB the system can be set up and solved by the 2 commands

```
T=spdiags([a d b],-1:1,n,n);
x=T\r;
```

The `spdiags` command creates the matrix and stores it in a sparse way. The MATLAB interpreter recognises that `T` is sparse and performs the solution procedure implied by `T\r` in an efficient way as indicated above.

### 4.12.2 Real symmetric and positive definite systems

A matrix $A$ is a real symmetric and positive definite matrix if its entries are real and

$$A = A^T \qquad \text{and} \qquad \underline{x}^T A \underline{x} > 0 \quad \text{for all } \underline{x} \neq \underline{0}. \tag{4.80}$$

There are not too many theorems in this module but we now present one that guarantees a special type of lower triangular/ upper triangular factorization of such a matrix.

**Theorem.** (Cholesky factorization) If $A$ is an $n \times n$ real symmetric positive definite matrix then there exists a unique lower triangular $n \times n$ real matrix $L$ with positive diagonal entries such that

$$A = LL^T \ . \tag{4.81}$$

(Note that here $L$ denotes a lower triangular matrix and not a unit lower triangular matrix as in the previous pages of these notes.)

**Proof.** We use induction on the size $n$ of the matrix.

The case $n = 1$.
In this case $\underline{x}^T A \underline{x} = a_{11} x_1^2 > 0$ for all $\underline{x} \neq \underline{0}$ implies that $a_{11} > 0$. Hence take $l_{11} = \sqrt{a_{11}} > 0$ and we have $A = LL^T$.

*Induction hypothesis:* Assume the result holds for all $n < m$.

Let $A_k$ be the $k \times k$ leading principal sub-matrix of $A$. We consider $A_m$ in partitioned form.

$$A_m = \begin{pmatrix} A_{m-1} & \underline{b} \\ \underline{b}^T & a_{mm} \end{pmatrix}, \quad \underline{b} \in \mathbb{R}^{m-1} \ . \tag{4.82}$$

Since $A$ is positive definite all its principal sub-matrices are also positive definite. This follows since if

$$\underline{x}^T = (x_1, \cdots, x_k, 0, \cdots, 0) \in \mathbb{R}^m \quad \text{and} \quad \underline{y}^T = (x_1, \cdots, x_k) \in \mathbb{R}^k \tag{4.83}$$

then

$$\underline{x}^T A \underline{x} = \underline{y}^T A_k \underline{y} > 0 \quad \text{for all } \underline{y} \neq \underline{0}. \tag{4.84}$$

Thus $A_{m-1}$ satisfies the positive definite condition and by the induction hypothesis $A_{m-1}$ has a unique Cholesky factorization

$$A_{m-1} = L_{m-1} L_{m-1}^T \ . \tag{4.85}$$

We now construct a factorization of $A_m$. For this consider the product

$$\begin{pmatrix} L_{m-1} & \underline{0} \\ \underline{l}^T & l_{mm} \end{pmatrix} \begin{pmatrix} L_{m-1}^T & \underline{l} \\ \underline{0}^T & l_{mm} \end{pmatrix} = \begin{pmatrix} L_{m-1} L_{m-1}^T & L_{m-1} \underline{l} \\ \underline{l}^T L_{m-1}^T & \underline{l}^T \underline{l} + l_{mm}^2 \ . \end{pmatrix} \tag{4.86}$$

This is equal to $A_m$ provided

$$L_{m-1} \underline{l} = \underline{b} \quad \text{and} \quad \underline{l}^T \underline{l} + l_{mm}^2 = a_{mm} \ . \tag{4.87}$$

This uniquely determines $\underline{l}$ (the unique solution of $L_{m-1} \underline{l} = \underline{b}$) and $l_{mm}^2 = a_{mm} - \underline{l}^T \underline{l}$. To complete the proof we must show that

$$a_{mm} - \underline{l}^T \underline{l} > 0 \tag{4.88}$$

so that $l_{mm}$ is real. To prove this the trick is to use results about determinants and a result about the eigenvalues of positive definite symmetric matrices. We have

$$\det A_m = \det(L_m L_m^T) = (\det L_m)(\det L_m^T) \tag{4.89}$$
$$= (l_{11} \cdots l_{mm})^2 \tag{4.90}$$

since $L_m$ is lower triangular. Now since a real symmetric positive definite matrix has real positive eigenvalues and

$$\det A = \text{product of the eigenvalues of } A \tag{4.91}$$

it follows that $\det A_m > 0$ which in turn implies that $l_{mm}^2 > 0$ and thus we can choose $l_{mm} > 0$. □

*Stability of the Cholesky factorization*

With $A = (a_{ij})$ and $L = (l_{ij})$ the factorization $A = LL^T$ is in component form

$$a_{ij} = \sum_{k=1}^{\min(i,j)} l_{ik} l_{jk} \ . \tag{4.92}$$

For $i = j$ (i.e. the diagonal entry)

$$a_{ii} = \sum_{k=1}^{i} l_{ik}^2 = l_{i1}^2 + l_{i2}^2 + \cdots + l_{ii}^2 \ . \tag{4.93}$$

From this last equation we have that in particular none of the non-negative numbers $l_{i1}^2, \cdots, l_{ii}^2$ can exceed $a_{ii}$, i.e.

$$|l_{ij}| \le \sqrt{a_{ii}} \quad \text{for all } i \text{ and } j. \tag{4.94}$$

This result which bounds the size of the entries of $L$ in terms of the size of the entries of $A$ (the diagonal entries of $A$ in this particular case) is called a stability result.

*The Cholesky algorithm*

From the factorization

$$A = LL^T \tag{4.95}$$

we have in component form

$$a_{ij} = \sum_{k=1}^{\min(i,j)} l_{ik} l_{jk} = \begin{cases} \left( \sum_{k=1}^{j-1} l_{ik} l_{jk} \right) + l_{ij} l_{jj}, & j < i, \\ \\ l_{i1}^2 + \cdots + l_{ii}^2, & j = i. \end{cases} \tag{4.96}$$

This leads to the following algorithm for computing the entries of $L$.

$$y_1 = a_{11} \tag{4.97}$$

$$l_{11} = \sqrt{y_1} \tag{4.98}$$

For $i = 2, \cdots, n$ do the following:

$$\tag{4.99}$$

$$l_{ij} = \left( a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right) / l_{jj}, \quad j = 1, \cdots, i-1, \tag{4.100}$$

$$y_i = a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2, \tag{4.101}$$

$$l_{ii} = \sqrt{y_i} \ . \tag{4.102}$$

*Remark*

The above algorithm requires that $y_i > 0$ for $i = 1, \cdots, n$ which is ensured by the positive definite condition of $A$. If the algorithm is used on an arbitrary real symmetric matrix then it will fail with $y_i \le 0$ where $A_i$ (which is the $i \times i$ principal sub-matrix of $A$) is the first leading principal sub-matrix which is not positive definite. This is because if it had not failed we would have constructed a factorization

$$A_i = L_i L_i^T \tag{4.103}$$

which in turn implies that for $\underline{x}_i \in \mathbb{R}^i$, $\underline{x}_i \neq \underline{0}$

$$\begin{aligned} \underline{x}_i^T A_i \underline{x}_i &= \underline{x}_i^T L_i L_i^T \underline{x}_i = \left(L_i^T \underline{x}_i\right)^T \left(L_i^T \underline{x}_i\right) && (4.104) \\ &= (\text{sum of the squares of the components of the vector } L_i^T \underline{x}_i) > 0, && (4.105) \end{aligned}$$

i.e. $A_i$ is positive definite. Hence the Cholesky algorithm is a way of testing whether or not a matrix is positive definite.

*Example*

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 5 & -1 \\ -1 & -1 & 3 \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{pmatrix} \tag{4.106}$$

By equating entries in the order $1,1$, then $2,1$, then $2,2$, then $3,1$, then $3,2$ and finally $3,3$ we get

$$\begin{array}{rcllrcl} 1 &=& l_{11}^2 & \text{gives} & l_{11} &=& 1, \\ 2 &=& l_{21} l_{11} & \text{gives} & l_{21} &=& 2, \\ 5 &=& l_{21}^2 + l_{22}^2 = 4 + l_{22}^2, & \text{gives} & l_{22} &=& 1, \\ -1 &=& l_{31} l_{11} & \text{gives} & l_{31} &=& -1, \\ -1 &=& l_{31} l_{21} + l_{32} l_{22} = -2 + l_{32}, & \text{gives} & l_{32} &=& 1, \\ 3 &=& l_{31}^2 + l_{32}^2 + l_{33}^2 = 1 + 1 + l_{33}^2, & \text{gives} & l_{33} &=& 1. \end{array} \tag{4.107}$$

Hence

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix} \quad \text{and} \quad LL^T = A\ . \tag{4.108}$$

To do this with MATLAB you can type the following.

```
A=[1  2 -1
   2  5 -1
  -1 -1  3];
R=chol(A);
disp(R);
```

The output produced is

```
     1     2    -1
     0     1     1
     0     0     1
```

i.e. `R` is upper triangular and corresponds to $L^T$.

# 5 The $QR$ factorization of a matrix

In the Gauss elimination process/$LU$ factorization procedure the reduction of a matrix $A$ to upper triangular form (sometimes also called right triangular form, which is the reason for using the letter $R$ in this context) was achieved by pre-multiplying by a finite sequence of Gauss transformation matrices

$$M_k = I - \underline{m}_k\, \underline{e}_k^T, \quad \underline{m}_k = \begin{pmatrix} 0 \\ \vdots \\ m_{k+1,k} \\ \vdots \\ m_{nk} \end{pmatrix} . \tag{5.1}$$

The objective now is to show that the reduction to upper triangular form can also be achieved by a sequence of certain "special" orthogonal matrices.

## 5.1 Householder matrices

(Householder matrices are named after Alston Householder who died on July 4, 1993 at the age of 89.)

$$P = P(\underline{v}) = I - 2\underline{v}\,\underline{v}^T, \quad \underline{v}^T\underline{v} = 1 . \tag{5.2}$$

If $\underline{v}$ is an $n \times 1$ vector then the normalization $\underline{v}^T\underline{v} = 1$ means that there are $n-1$ independent components.

## 5.2 Givens' matrices

$$G(i,j,\theta) = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & c & s & & \\ & & -s & c & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix}, \quad \text{where } c = \cos(\theta),\ s = \sin(\theta). \tag{5.3}$$

$G(i,j,\theta)$ differs from $I$ only in the 4 entries shown. For given $i$ and $j$ there is only 1 independent component, i.e. $\theta$.

## 5.3 Properties of $P$ and $G$

(1)

$$P^2 = (I - 2\underline{v}\,\underline{v}^T)(I - 2\underline{v}\,\underline{v}^T) = I - 4\underline{v}\,\underline{v}^T + 4\underline{v}(\underline{v}^T\underline{v})\underline{v}^T = I, \tag{5.4}$$

since $\underline{v}^T\underline{v} = 1$. Hence $P = P^T = P^{-1}$ is symmetric and orthogonal. $P$ has a geometric interpretation as a reflection, i.e. $\underline{x}$ and $P\underline{x}$ are mirror images in a certain hyperplane (which is defined by the vector $\underline{v}$) in $n$-dimensional space.

(2)

$$(G(i,j,\theta))^T G(i,j,\theta) = I, \quad \text{i.e. } G(i,j,\theta) \text{ is orthogonal.} \tag{5.5}$$

This follows since the columns of $G(i,j,\theta)$ are mutually orthonormal, i.e. they all have Euclidean length of 1 and they are all mutually orthogonal. $G$ has a geometrical interpretation as a rotation of 2 of components in $n$-dimensional space.

## 5.4 Choice of $\underline{v}$ for reduction

If $\underline{x} = \underline{0}$ then the vector is already "reduced". Thus we will suppose that $\underline{x} \neq \underline{0}$. In this case the aim is to choose $\underline{v}$ so that $P(\underline{v})\underline{x}$ only has 1 non-zero component. As $P$ is orthogonal the vectors $P\underline{x}$ and $\underline{x}$ have the same 2-norm and so suppose that

$$P\underline{x} = (I - 2\underline{v}\,\underline{v}^T)\underline{x} = \beta \|\underline{x}\|_2\, \underline{e}_1, \quad \text{where } \beta = \pm 1 \;. \tag{5.6}$$

Thus

$$\underline{x} - 2(\underline{v}^T\underline{x})\underline{v} = \beta \|\underline{x}\|_2\, \underline{e}_1 \tag{5.7}$$

and hence

$$2(\underline{v}^T\underline{x})\underline{v} = \underline{x} - \beta \|\underline{x}\|_2\, \underline{e}_1 \;. \tag{5.8}$$

In words we have that a scalar multiplied by $\underline{v}$ is equal to the vector on the right hand side of the above equation, that is the direction of $\underline{v}$ is determined provided the right hand side is not the zero vector. As we have a choice of whether $\beta = 1$ or $\beta = -1$ we can select to choose it so that the vector on the right hand side is non-zero by selecting from the 2 choices the one which gives the greatest 2-norm. This choice is desirable computationally as it avoids any cancellation error in the construction of $\underline{v}$. If we denote the resulting vector by $\underline{y}$ and we take

$$\underline{v} = \frac{1}{\|\underline{y}\|_2}\underline{y} \tag{5.9}$$

then we have the desired effect that

$$P(\underline{v})\underline{x} = \beta \|\underline{x}\|_2\, \underline{e}_1 \;. \tag{5.10}$$

We have just shown how to "reduce" a vector by a single Householder matrix. The next task is to indicate how a sequence of such matrices can be used to reduce a matrix. The key point is to choose the matrices so as not to undo previous reduction steps. If

$$\underline{v}_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ v_{kk} \\ \vdots \\ v_{nk} \end{pmatrix} = \begin{pmatrix} \underline{0} \\ \tilde{\underline{v}}_k \end{pmatrix} \quad \text{where} \quad \tilde{\underline{v}}_k = \begin{pmatrix} v_{kk} \\ \vdots \\ v_{nk} \end{pmatrix} \tag{5.11}$$

then the first $k-1$ rows and columns of $\underline{v}_k\, \underline{v}_k^T$ are zero. Thus $P(\underline{v}_k)$ has the partitioned form

$$P(\underline{v}_k) = \begin{pmatrix} I_{k-1} & O \\ O & P(\tilde{\underline{v}}_k) \end{pmatrix}, \tag{5.12}$$

where $I_{k-1}$ is the $(k-1)\times(k-1)$ identity matrix and where $P(\tilde{\underline{v}}_k)$ is a smaller Householder matrix. If a matrix $A^{(k-1)}$ has zero entries in columns 1 to $k-1$ below the diagonal then the product $P(\underline{v}_k)A^{(k-1)}$ has the same entries in these columns. By choosing $\tilde{\underline{v}}_k$ to eliminate the entries below the diagonal in column $k$ it is hence possible to generate a sequence of matrices $A^{(0)} = A$, $A^{(1)}$, $\cdots$, $A^{(n-1)}$ by

$$A^{(k)} = P(\underline{v}_k)A^{(k-1)}, \quad k = 1, 2, \cdots, n-1, \tag{5.13}$$

where $A^{(k)}$ has been reduced in columns 1 to $k$ below the diagonal. The vector $\underline{\tilde{v}}_k$, which is part of $\underline{v}_k$, which achieves the reduction depends on the entries

$$\begin{pmatrix} a_{kk}^{(k-1)} \\ \vdots \\ a_{nk}^{(k-1)} \end{pmatrix} . \tag{5.14}$$

In the above it is assumed that if a vector is already appropriately reduced then the step is skipped. The result of the reduction is that we have an upper triangular matrix $R$ where

$$R = A^{(n-1)} = P_{n-1} \cdots P_1 A, \quad \text{where } P_k = P(\underline{v}_k) \ . \tag{5.15}$$

Thus

$$A = (P_{n-1} \cdots P_1)^{-1} R = (P_1 \cdots P_{n-1}) R = QR, \tag{5.16}$$

where

$$Q = P_1 \cdots P_{n-1} \tag{5.17}$$

is an orthogonal matrix.

### 5.4.1 Remarks

(1) The above reduction can always be done, i.e. there is no restriction to $A$ being non-singular or even being a square matrix. If $A$ has at least as many rows as columns then $R$ having zero entries below the diagonal means that it has the partitioned form

$$R = \begin{pmatrix} R_1 \\ O \end{pmatrix}, \tag{5.18}$$

where $R_1$ is a square upper triangular matrix.

(2) The computation of the product $(I - 2\underline{v}_k \underline{v}_k^T) A^{(k-1)}$ involves about twice as many flops as does the computation of the product $(I - 2\underline{m}_k \underline{e}_k^T) A^{(k-1)}$ in the $LU$ factorization. Hence computing a $QR$ factorization involves about twice as many flops as computing a $LU$ factorization.

(3) The reduction can also be done using Givens' matrices but this involves more flops that the Householder approach. Givens' matrices are used when only a few entries are to be reduced.

(4) In MATLAB the command

```
[Q R]=qr(A);
```

takes the input matrix `A` and returns the orthogonal matrix `Q` and an upper triangular matrix `R` such that, in MATLAB notation, `A=Q*R`.

Unlike in the case of Gauss elimination, it is nearly always messy to attempt to illustrate the construction of a $QR$ factorization by hand calculation. The reason for this is that the process of normalising a vector requires square roots and thus you cannot easily set up examples in which the numbers are manageable by hand calculations. $QR$ calculations are intended for computers only as part of some algorithm.

(5) By the Householder approach it was not possible to guarantee that the diagonal entries of $R$ are all positive. However if a Householder step gives a negative diagonal entry then the use of the appropriate Givens' matrix with $\theta = \pi$ will make the diagonal entry positive. Thus it is always possible to factorize $A$ as $QR$ with $R$ having non-negative diagonal entries. In the case of a square matrix and the factorization $A = QR$ we have

$$\det A = \det Q \det R = \pm \det R \quad \text{since } Q^T Q = I \text{ implies } 1 = (\det Q)^2. \tag{5.19}$$

As $R$ is upper triangular its determinant is the product of its diagonal entries and hence we can easily compute $|\det A| = |\det R|$. In the case of a non-singular square matrix $A$ the $QR$ factorization with $R$ having positive diagonal entries is unique. This follows by considering

$$A = Q_1 R_1 = Q_2 R_2 \quad \text{which implies that} \quad Q_2^T Q_1 = R_2 R_1^{-1} = \tilde{R}\ . \tag{5.20}$$

From a previous result the product of two upper triangular matrices is also upper triangular (i.e. $\tilde{R}$ is upper triangular) and by the above line it is also orthogonal (as it is equal to $Q_2^T Q_1$). Hence $\tilde{R}^T \tilde{R} = I$ which is Cholesky factorization of $I$ and this in turn implies that $\tilde{R} = I$ from which we obtain $Q_1 = Q_2$ and $R_1 = R_2$.

## 5.5 Some properties and applications

(1) **Stability**

$$\|A\|_2^2 = \rho(A^T A) = \rho(R^T Q^T Q R) = \rho(R^T R) = \rho(R_1^T R_1) = \|R_1\|_2^2 \tag{5.21}$$

where $R_1$ is the square upper triangular part of $R$.

(2) **Solving linear equations**
In the case of a square matrix $A$ and the system

$$A\underline{x} = \underline{b} \quad \text{and} \quad A = QR \tag{5.22}$$

we have

$$Q\underline{y} = \underline{b}, \quad \underline{y} = Q^T \underline{b} \quad \text{(matrix multiplication to get } \underline{y}\text{)} \tag{5.23}$$

followed by

$$R\underline{x} = \underline{y} \quad \text{(solve for } \underline{x} \text{ by back substitution).} \tag{5.24}$$

# 6 Perturbations of matrices and the sensitivity of linear equations

One application of norms is in quantifying the effect of perturbations. For example if $A$ is a non-singular matrix then how "small" must the matrix $\Delta A$ be to ensure that $A+\Delta A$ is also non-singular? Related to this question is by how much do the solutions $\underline{x}$ and $\underline{x}+\Delta\underline{x}$ of the linear systems

$$A\underline{x} = \underline{b} \tag{6.1}$$

and

$$(A+\Delta A)(\underline{x}+\Delta\underline{x}) = \underline{b}+\Delta\underline{b} \tag{6.2}$$

differ? This is an important question to ask when such systems are solved on a computer using the usual floating point arithmetic as initially entering $A$ and $\underline{b}$ on a computer may involve perturbing them respectively by $\epsilon A$ and $\epsilon\underline{b}$ where $\epsilon$ is the computer's machine precision (typically about $10^{-16}$ when FORTRAN double precision, double in C or MATLAB are used).

In the following we will often use

$$A+\Delta A = A(I+A^{-1}\Delta A), \quad (A+\Delta A)^{-1} = (I+A^{-1}\Delta A)^{-1}A^{-1} \tag{6.3}$$

and thus we begin by considering perturbations of the identity matrix $I$. Also, when a matrix norm is used we will always assume that it is a norm which is subordinate to a vector norm.

**Theorem.** If $E$ is an $n\times n$ matrix and $\|E\|<1$ then the matrix $I-E$ is non-singular and

$$(I-E)^{-1} = \sum_{k=0}^{\infty} E^k = I+E+E^2+\cdots \tag{6.4}$$

with

$$\|(I-E)^{-1}\| \le \frac{1}{1-\|E\|} \quad \text{and} \quad \|(I-E)^{-1}-I\| \le \frac{\|E\|}{1-\|E\|}\ . \tag{6.5}$$

**Proof.** To prove that $I-E$ is non-singular we can use either of the following:

(i) We consider the eigenvalues of $I-E$. $\lambda$ is an eigenvalue of $E$ if and only of $1-\lambda$ is an eigenvalue of $I-E$. Since $\rho(E)\le\|E\|<1$ it follows that no eigenvalue of $I-E$ can be 0 abd thus $I-E$ is non-singular.

(ii) We use a proof by contradiction, i.e. we suppose that $I-E$ is singular and show that this leads to a contradiction. $I-E$ being singular means that there exist $\underline{x}\ne\underline{0}$ with

$$(I-E)\underline{x} = \underline{0}, \quad \text{i.e. } \underline{x} = E\underline{x}\ . \tag{6.6}$$

But by the matrix norm properties

$$\|\underline{x}\| = \|E\underline{x}\| \le \|E\|\,\|\underline{x}\| < \|\underline{x}\| \tag{6.7}$$

since $\|E\|<1$ and $\|\underline{x}\|>0$. This is a contradiction. Thus the original assumption was incorrect and $I-E$ is non-singular.

We now consider the representation of the inverse and the inequality. For this observe that, as in the case of scalars,

$$(I-E)(I+E+E^2\cdots+E^{m-1}) = I-E^m \ . \tag{6.8}$$

Thus

$$I+E+\cdots+E^{m-1} = (I-E)^{-1}(I-E^m) \ . \tag{6.9}$$

Now observe that by repeatedly using the sub-multiplicative property we have

$$\|E^m\| \le \|E\|^m \to 0 \quad \text{as } m\to\infty \ . \tag{6.10}$$

Hence $E^m \to 0$ as $m\to\infty$ and

$$(I-E)^{-1} = I+E+E^2+\cdots = \sum_{k=0}^{\infty} E^k \quad \text{and} \quad (I-E)^{-1}-I = E+E^2+\cdots = \sum_{k=1}^{\infty} E^k \ . \tag{6.11}$$

By the triangle inequality

$$\|(I-E)^{-1}(I-E^m)\| = \|I+E+\cdots+E^{m-1}\| \le \|I\|+\|E\|+\|E\|^2+\cdots+\|E\|^{m-1} \tag{6.12}$$

$$\le \sum_{k=0}^{\infty}\|E\|^k = \frac{1}{1-\|E\|} \ . \tag{6.13}$$

Similarly

$$\|(I-E)^{-1}(I-E^m)-I\| \le \sum_{k=1}^{\infty}\|E\|^k = \frac{\|E\|}{1-\|E\|} \ . \tag{6.14}$$

Since these are true for all $m$ we have

$$\|(I-E)^{-1}\| \le \frac{1}{1-\|E\|} \quad \text{and} \quad \|(I-E)^{-1}-I\| \le \frac{\|E\|}{1-\|E\|} \ .\square \tag{6.15}$$

In the following analysis of the linear equations problem $A\underline{x}=\underline{b}$ the **condition number** of $A$ is an important quantity.

**Definition.** If $A$ is a non-singular matrix then the **condition number** with respect to a matrix norm $\|.\|$ is given by

$$\kappa(A) = \|A\|\,\|A^{-1}\| \ . \tag{6.16}$$

When the matrix 2-norm (sometimes called the spectral norm) is used the condition number is called the **spectral condition number** .

Since for all matrix norms $1\le\|I\|$ we have

$$1 \le \|I\| = \|AA^{-1}\| \le \|A\|\,\|A^{-1}\| = \kappa(A) \tag{6.17}$$

with $1=\|I\|$ for matrix norms which are subordinate to vector norms. For such norms we say that a matrix is perfectly conditioned when $\kappa(A)=1$, well conditioned when $\kappa(A)$ is "near" 1 and badly-conditioned or ill-conditioned when $\kappa(A)$ is "large".

In the following the aim is to consider the problems $A\underline{x}=\underline{b}$ and $(A+\Delta A)(\underline{x}+\Delta\underline{x}) = \underline{b}+\Delta\underline{b}$. This is done by first considering simpler problems corresponding to only changing the right hand side $\underline{b}$ or to only changing the matrix $A$.

**Lemma.** (only changing $\underline{b}$) If the matrix $A$ is non-singular, the vector $\underline{b} \neq \underline{0}$ and vectors $\underline{x}$ and $\underline{x} + \Delta\underline{x}$ satisfy

$$A\underline{x} = \underline{b} \quad \text{and} \quad A(\underline{x} + \Delta\underline{x}) = \underline{b} + \Delta\underline{b} \tag{6.18}$$

then

$$\frac{\|\Delta\underline{x}\|}{\|\underline{x}\|} \leq \kappa(A)\frac{\|\Delta\underline{b}\|}{\|\underline{b}\|} \ . \tag{6.19}$$

**Proof.**

First observe that $A\underline{x} = \underline{b}$ and $A(\underline{x} + \Delta\underline{x}) = \underline{b} + \Delta\underline{b}$ implies that

$$A\Delta\underline{x} = \Delta\underline{b} \quad \text{and} \quad \Delta\underline{x} = A^{-1}\Delta\underline{b} \ . \tag{6.20}$$

Therefore

$$\|\Delta\underline{x}\| = \|A^{-1}\Delta\underline{b}\| \leq \|A^{-1}\| \, \|\Delta\underline{b}\| \ . \tag{6.21}$$

Since we wish to obtain an upper bound for $\|\Delta\underline{x}\|/\|\underline{x}\|$ we require a lower bound for $\|\underline{x}\|$. We can obtain this as follows. From $\underline{b} = A\underline{x}$ we have

$$\|\underline{b}\| = \|A\underline{x}\| \leq \|A\| \, \|\underline{x}\|, \quad \text{and hence } \|\underline{x}\| \geq \frac{\|\underline{b}\|}{\|A\|} \ . \tag{6.22}$$

Thus

$$\frac{\|\Delta\underline{x}\|}{\|\underline{x}\|} \leq \|A\| \, \|A^{-1}\| \, \frac{\|\Delta\underline{b}\|}{\|\underline{b}\|} = \kappa(A)\frac{\|\Delta\underline{b}\|}{\|\underline{b}\|} \ .\square \tag{6.23}$$

**Lemma.** (only changing $A$) If the matrix $A$ is non-singular, the vector $\underline{b} \neq \underline{0}$, $\Delta A$ is a matrix such that $\|A^{-1}\Delta A\| < 1$ and $\underline{x}$ and $\Delta\underline{x}$ are vectors satisfying

$$A\underline{x} = \underline{b} \quad \text{and} \quad (A + \Delta A)(\underline{x} + \Delta\underline{x}) = \underline{b} \tag{6.24}$$

then $A + \Delta A$ is non-singular and

$$\frac{\|\Delta\underline{x}\|}{\|\underline{x}\|} \leq \frac{r}{1 - r}, \quad \text{where} \quad r = \kappa(A)\frac{\|\Delta A\|}{\|A\|} \ . \tag{6.25}$$

**Proof.**

Observe that $A + \Delta A = A(I + A^{-1}\Delta A)$ is non-singular since $A$ is non-singular and $\|A^{-1}\Delta A\| < 1$ implies that $I + A^{-1}\Delta A$ is non-singular. (The product of two non-singular matrices is non-singular.)[6] Rearranging we have

$$\underline{x} + \Delta\underline{x} = (A + \Delta A)^{-1}\underline{b} = (I + A^{-1}\Delta A)^{-1}A^{-1}\underline{b} = (I + A^{-1}\Delta A)^{-1}\underline{x} \tag{6.26}$$

and thus

$$\Delta\underline{x} = \left((I + A^{-1}\Delta A)^{-1} - I\right)\underline{x} \ . \tag{6.27}$$

Taking norms and using the result of the theorem on page 6.1 we have

$$\frac{\|\Delta\underline{x}\|}{\|\underline{x}\|} \leq \|(I + A^{-1}\Delta A)^{-1} - I\| \leq \frac{\|A^{-1}\Delta A\|}{1 - \|A^{-1}\Delta A\|} \leq \frac{r}{1 - r} \tag{6.28}$$

[6] From the result $\det(AB) = (\det A)(\det B)$, $\det A \neq 0$ and $\det B \neq 0$ implies that $\det(AB) \neq 0$ and as a consequence $AB$ is non-singular.

since

$$\|A^{-1}\Delta A\| \le \|A^{-1}\|\,\|\Delta A\| = \kappa(A)\frac{\|\Delta A\|}{\|A\|} = r\ .\Box \tag{6.29}$$

**Theorem.** If a matrix $A$ is non-singular, a vector $\underline{b} \neq \underline{0}$, a matrix $\Delta A$ satisfies $\|A^{-1}\Delta A\| < 1$ and vectors $\underline{x}$ and $\underline{x} + \Delta\underline{x}$ satisfy

$$A\underline{x} = \underline{b} \quad \text{and} \quad (A + \Delta A)(\underline{x} + \Delta\underline{x}) = (\underline{b} + \Delta\underline{b}) \tag{6.30}$$

then

$$\frac{\|\Delta\underline{x}\|}{\|\underline{x}\|} \le \frac{\kappa(A)}{1 - \kappa(A)\dfrac{\|\Delta A\|}{\|A\|}}\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta\underline{b}\|}{\|\underline{b}\|}\right)\ . \tag{6.31}$$

**Proof.**

The proof involves the two given systems and the intermediate system

$$A\underline{y} = \underline{b} + \Delta\underline{b}\ . \tag{6.32}$$

We can use the previous two lemmas to compare $\underline{y}$ with $\underline{x}$ and $\underline{x} + \Delta\underline{x}$. From the lemmas we have

$$\frac{\|\underline{x} - \underline{y}\|}{\|\underline{x}\|} \le \kappa(A)\frac{\|\Delta\underline{b}\|}{\|\underline{b}\|}, \qquad \frac{\|\underline{x} + \Delta\underline{x} - \underline{y}\|}{\|\underline{y}\|} \le \frac{r}{1-r}, \tag{6.33}$$

where as before

$$r = \frac{\kappa(A)\|\Delta A\|}{\|A\|}\ . \tag{6.34}$$

Observe that

$$\Delta\underline{x} = (\underline{x} + \Delta\underline{x} - \underline{y}) - (\underline{x} - \underline{y}) \tag{6.35}$$

and thus

$$\|\Delta\underline{x}\| \le \|\underline{x} + \Delta\underline{x} - \underline{y}\| + \|\underline{x} - \underline{y}\| \tag{6.36}$$

$$\le \frac{r}{1-r}\|\underline{y}\| + \kappa(A)\frac{\|\Delta\underline{b}\|}{\|\underline{b}\|}\|\underline{x}\|\ . \tag{6.37}$$

To complete the proof we must relate $\underline{y}$ with $\underline{x}$ in order to bound $\|\underline{y}\|$ in terms of $\|\underline{x}\|$. We have

$$\underline{y} = \underline{x} - (\underline{x} - \underline{y}) \tag{6.38}$$

and hence

$$\|\underline{y}\| = \|\underline{x} - (\underline{x} - \underline{y})\| \le \|\underline{x}\| + \|\underline{x} - \underline{y}\|\ . \tag{6.39}$$

Thus

$$\|\underline{y}\| \le \|\underline{x}\| + \kappa(A)\frac{\|\Delta\underline{b}\|}{\|\underline{b}\|}\|\underline{x}\| = \left(1 + \kappa(A)\frac{\|\Delta\underline{b}\|}{\|\underline{b}\|}\right)\|\underline{x}\| \tag{6.40}$$

Combining the results gives

$$(1-r)\|\Delta\underline{x}\| \le r\left(1 + \kappa(A)\frac{\|\Delta\underline{b}\|}{\|\underline{b}\|}\right)\|\underline{x}\| + (1-r)\kappa(A)\frac{\|\Delta\underline{b}\|}{\|\underline{b}\|}\|\underline{x}\| \tag{6.41}$$

$$(1-r)\frac{\|\Delta\underline{x}\|}{\|\underline{x}\|} \le r + \kappa(A)\frac{\|\Delta\underline{b}\|}{\|\underline{b}\|} = \kappa(A)\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta\underline{b}\|}{\|\underline{b}\|}\right)\ .\Box \tag{6.42}$$

## 6.1 Remarks

The above results show that when $\kappa(A)$ is large we cannot expect to solve a linear system $A\underline{x} = \underline{b}$ as accurately as when $\kappa(A)$ is near 1. With usual computer floating point arithmetic with a machine precision of about $\epsilon = 10^{-16}$ we cannot expect to solve a linear system $A\underline{x} = \underline{b}$ on such a computer by <u>any</u> method when $\kappa(A)$ is of the order of $1/\epsilon = 10^{16}$ or when $\kappa(A)$ is larger than this.

To compute the condition number accurately for a given norm involves the computation of $A^{-1}$ which requires more computation than is involved in solving $A\underline{x} = \underline{b}$. Thus in some respects the condition number is of more theoretical importance than it is a practical number to calculate. However methods for solving $A\underline{x} = \underline{b}$ can often be adapted to also give an estimate for $\kappa(A)$ without too much additional computation and in practice you only need to know roughly the size of $\kappa(A)$.

In MATLAB you can compute a condition number by typing `cond(A)`, which gives the spectral condition number of `A` when `A` is a square matrix.

## 6.2 Some properties of condition numbers

(1) For a real symmetric and positive definite matrix $A$ with eigenvalues $\lambda_1, \cdots, \lambda_n$ with

$$0 < \lambda_n \le \lambda_{n-1} \le \cdots \le \lambda_1 \tag{6.43}$$

the eigenvalues of $A^{-1}$ are

$$0 < \frac{1}{\lambda_1} \le \frac{1}{\lambda_2} \le \cdots \le \frac{1}{\lambda_n} = \rho\left(A^{-1}\right) \ . \tag{6.44}$$

We have already shown that the 2-norm of a real symmetric positive definite matrix is the largest eigenvalue and thus

$$\|A\|_2 = \lambda_1, \quad \|A^{-1}\|_2 = \frac{1}{\lambda_n} \quad \text{and} \quad \kappa_2(A) = \frac{\lambda_1}{\lambda_n}. \tag{6.45}$$

Thus for a real symmetric positive definite matrix the spectral condition number is the ratio of the largest to the smallest eigenvalue.

(2) If $Q$ is an orthogonal matrix then the result

$$\|A\|_2 = \|AQ\|_2 = \|QA\|_2 \tag{6.46}$$

implies

$$\kappa_2(A) = \kappa_2(AQ) = \kappa_2(QA) \ . \tag{6.47}$$

This is a **stability** property of orthogonal matrices.

## 6.3 The condition number of certain tridiagonal matrices

We end this section by showing, by means of an example, how to estimate a condition number and how to compute explicitly a condition number in the case of certain tridiagonal matrices.

Let $A$ denote the $n \times n$ tridiagonal matrix

$$A = \begin{pmatrix} 2+d & -1 & & & \\ -1 & 2+d & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2+d & -1 \\ & & & -1 & 2+d \end{pmatrix}, \quad d \ge 0 \quad \text{and let} \quad F = \begin{pmatrix} 0 & 1 & & & \\ 1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 0 & 1 \\ & & & 1 & 0 \end{pmatrix}. \tag{6.48}$$

We can write this as

$$A = (2+d)(I-E) \quad \text{where} \quad (2+d)E = -F\ . \tag{6.49}$$

From the representation of the $\infty$-matrix norm in terms of row sums we have

$$\|A\|_\infty = 4+d, \quad \|F\|_\infty = 2 \quad \text{and } \|E\|_\infty = \frac{2}{2+d} \le 1. \tag{6.50}$$

Observe that

$$A^{-1} = \frac{1}{2+d}(I-E)^{-1}\ . \tag{6.51}$$

Thus when $d > 0$, so that $\|E\|_\infty < 1$, we have

$$\|A^{-1}\|_\infty = \frac{1}{2+d}\|(I-E)^{-1}\|_\infty \le \left(\frac{1}{2+d}\right)\frac{1}{1-\|E\|_\infty} = \frac{1}{(2+d)-2} = \frac{1}{d}. \tag{6.52}$$

Hence

$$\kappa_\infty(A) = \|A\|_\infty\|A^{-1}\|_\infty \le \frac{4+d}{d}\ . \tag{6.53}$$

Notice that this bound is independent of the size $n$ of the matrix.

We can in fact actually give all the eigenvalues and eigenvectors of $A$. Consider the vector $\underline{v}_k$ where

$$(\underline{v}_k)_j = \sin\left(\frac{jk\pi}{n+1}\right). \tag{6.54}$$

For the matrix $F$ we have

$$(F\underline{v}_k)_j = (\underline{v}_k)_{j-1} + (\underline{v}_k)_{j+1} = \sin\left(\frac{(j-1)k\pi}{n+1}\right) + \sin\left(\frac{(j+1)k\pi}{n+1}\right) = 2\cos\left(\frac{k\pi}{n+1}\right)\sin\left(\frac{jk\pi}{n+1}\right), \tag{6.55}$$

for $j = 2, 3, \cdots, n-1$. In fact with a closer inspection the relation reveals that it is also true when $j = 1$ and $j = n$. Thus the vector $\underline{v}_k$ is an eigenvector of $F$ for $k = 1, 2, \cdots, n$. The vectors $\underline{v}_1, \underline{v}_2, \cdots, \underline{v}_n$ are also eigenvectors of the matrix $A$ with the corresponding eigenvalues being

$$\lambda_k = (2+d) - 2\cos\left(\frac{k\pi}{n+1}\right) = d + 4\sin^2\left(\frac{k\pi}{2(n+1)}\right). \tag{6.56}$$

In the case $d = 0$ the smallest and largest eigenvalue are respectively

$$\lambda_1 = 4\sin^2\left(\frac{\pi}{2(n+1)}\right) \approx 4\left(\frac{\pi}{2(n+1)}\right)^2 \quad \text{and} \quad \lambda_n = 4\sin^2\left(\frac{n\pi}{2(n+1)}\right) \approx 4 \tag{6.57}$$

for large $n$. The spectral condition number is thus

$$\kappa(A) = \frac{\lambda_n}{\lambda_1} \approx \frac{4(n+1)^2}{\pi^2} \tag{6.58}$$

which increases as the size $n$ of the system increases.

# 7 Iterations in general – order, fixed points and the contraction mapping theorem

## 7.1 Convergence and order

**Definition.** A sequence of vectors $(\underline{x}_k)$ in $\mathbb{R}^n$ converges to $\underline{x}^* \in \mathbb{R}^n$ if

$$\lim_{k\to\infty} \|\underline{x}_k - \underline{x}^*\| = 0 \tag{7.1}$$

for some vector norm $\|.\|$.

There is a result of functional analysis which shows that all norms in a finite dimensional space are equivalent which implies that if the iteration converges in any vector norm of $\mathbb{R}^n$ then it must converge in all vector norms of $\mathbb{R}^n$.

From a practical point of view it is desirable to have an iteration which converges rapidly which leads to the consideration of the 'order of convergence'. The **order is at least** $p \ge 1$ if there exists a constant $C > 0$ such that

$$\|\underline{x}_{k+1} - \underline{x}^*\| \le C\|\underline{x}_k - \underline{x}^*\|^p \tag{7.2}$$

for all sufficiently large $k$. The **order is** $p$ if the iteration converges in such a way that

$$\lim_{k\to\infty} \frac{\|\underline{x}_{k+1} - \underline{x}^*\|}{\|\underline{x}_k - \underline{x}^*\|^p} \tag{7.3}$$

exists and is non-zero. If it does exist then

$$C = \lim_{k\to\infty} \frac{\|\underline{x}_{k+1} - \underline{x}^*\|}{\|\underline{x}_k - \underline{x}^*\|^p} \tag{7.4}$$

is called the **asymptotic error constant**. The value of $C$ will depend on which norm is used. If $p = 1$ then convergence is said to be **linear**. In this case we require that $0 < C < 1$ in (7.4) and $0 \le C < 1$ in (7.2). If $p > 1$ then convergence is said to be **superlinear**. There is no restriction on the magnitude of $C$ although for practical reasons the smaller is $C$ the better. If $p = 2$ then convergence is said to be **quadratic** and if $p = 3$ then convergence is is said to be **cubic**.

## 7.2 Iteration functions and fixed points

In methods to be described in these notes we usually consider sequences which are obtained by one or more starting vectors and an iteration function. Specifically most of the methods to be considered are of the form

$$\underline{x}_{k+1} = \underline{\phi}(\underline{x}_k) \tag{7.5}$$

or

$$\underline{x}_{k+1} = \underline{\phi}(\underline{x}_k, \cdots, \underline{x}_{k-m}), \tag{7.6}$$

where $\underline{\phi}$ is known as the **iteration function**. In (7.5) we have a 1-point method and in (7.6) we have a multi-point method if $m \ge 1$. Assuming that $\underline{\phi}$ is continuous and the iteration converges to $\underline{x}^*$ then in (7.5)

$$\underline{x}^* = \lim \underline{x}_{k+1} = \lim \underline{\phi}(\underline{x}_k) = \underline{\phi}(\underline{x}^*) \tag{7.7}$$

and similarly in (7.6)

$$\underline{x}^* = \underline{\phi}(\underline{x}^*, \cdots, \underline{x}^*) \ . \tag{7.8}$$

$\underline{x}^*$ is a **fixed point** of $\underline{\phi}$.

Newton's methods introduced earlier in these notes are all 1-point methods with

$$\phi(x) = x - \frac{f(x)}{f'(x)} \quad \text{or} \quad \underline{\phi}(\underline{x}) = \underline{x} - (J_f(\underline{x}))^{-1}\underline{f}(\underline{x}) \tag{7.9}$$

as appropriate. The secant method (to be considered later) is an example of a 2-point method.

## 7.3 The Contraction Mapping Theorem in $\mathbb{R}^n$

In the context of a 1-point iteration $\underline{x}_{k+1} = \underline{\phi}(\underline{x}_k)$ the following is a fundamental theorem relating to the existence of a fixed point and to the convergence of the iteration.

**Theorem.** Let $\mathcal{M}$ be a closed bounded region such that the following conditions hold.

(i) $\underline{\phi} : \mathcal{M} \to \mathcal{M}$ .

(ii) There exists a positive constant $L < 1$ such that

$$\|\underline{\phi}(\underline{x}) - \underline{\phi}(\underline{y})\| \le L\|\underline{x} - \underline{y}\| \quad \forall \underline{x}, \underline{y} \in \mathcal{M} .$$

Then there exists a unique fixed point $\underline{x}^* \in \mathcal{M}$ and given any $\underline{x}_0 \in \mathcal{M}$ the iteration generated by $\underline{x}_{k+1} = \underline{\phi}(\underline{x}_k)$ converges to $\underline{x}^*$.

**Proof.** The bulk of the proof is concerned with showing that the sequence $(\underline{x}_k)$ is a Cauchy sequence. This then establishes both the convergence of the sequence and the existence of a fixed point $\underline{x}^*$ since a sequence converges if and only if it is a Cauchy sequence. Recall that $(\underline{x}_k)$ is a Cauchy sequence if given any $\epsilon > 0$ there exists $N = N(\epsilon)$ such that

$$\|\underline{x}_n - \underline{x}_m\| < \epsilon \quad \text{for all } n, m > N. \tag{7.10}$$

Now by repeatedly applying the iteration function and the contraction property we have

$$\begin{aligned}
\|\underline{x}_{i+1} - \underline{x}_i\| &= \|\underline{\phi}(\underline{x}_i) - \underline{\phi}(\underline{x}_{i-1})\| \le L\|\underline{x}_i - \underline{x}_{i-1}\| && (7.11)\\
&= L\|\underline{\phi}(\underline{x}_{i-1}) - \underline{\phi}(\underline{x}_{i-2})\| \le L^2\|\underline{x}_{i-1} - \underline{x}_{i-2}\| && (7.12)\\
&\le \cdots \le L^i\|\underline{x}_1 - \underline{x}_0\| . && (7.13)
\end{aligned}$$

Thus for any $m > 0$ and $n = m + j$ we have by the triangle inequality that

$$\|\underline{x}_n - \underline{x}_m\| = \|\underline{x}_{m+j} - \underline{x}_m\| \le \|\underline{x}_{m+j} - \underline{x}_{m+j-1}\| + \cdots + \|\underline{x}_{m+1} - \underline{x}_m\| . \tag{7.14}$$

Combining this with the above gives

$$\begin{aligned}
\|\underline{x}_n - \underline{x}_m\| &\le (L^{m+j-1} + L^{m+j-2} + \cdots + L^m)\|\underline{x}_1 - \underline{x}_0\| && (7.15)\\
&= (L^{j-1} + L^{j-2} + \cdots + L + 1)L^m\|\underline{x}_1 - \underline{x}_0\| = \frac{1 - L^j}{1 - L}L^m\|\underline{x}_1 - \underline{x}_0\| && (7.16)\\
&\le \frac{L^m}{1 - L}\|\underline{x}_1 - \underline{x}_0\| && (7.17)
\end{aligned}$$

since $0 < L < 1$. The bound tends to 0 as $m \to \infty$ and thus given any $\epsilon > 0$ there exists an integer $N = N(\epsilon)$ as required in the Cauchy sequence definition. Hence the iteration converges and the continuity of $\underline{\phi}$ (which is implied by the contraction property) proves that the limit point is a fixed point of $\underline{\phi}$.

The only thing still to prove is that the limit point is unique. If there are two distinct limit points $\underline{x}_1^*$ and $\underline{x}_2^*$ then

$$0 < \|\underline{x}_1^* - \underline{x}_2^*\| = \|\underline{\phi}(\underline{x}_1^*) - \underline{\phi}(\underline{x}_2^*)\| \le L\|\underline{x}_1^* - \underline{x}_2^*\| < \|\underline{x}_1^* - \underline{x}_2^*\| \tag{7.18}$$

which is a contradiction.

### 7.3.1 The Lipschitz constant

In the statement of the contraction mapping theorem we have a condition of the form

$$\|\underline{\phi}(\underline{x}) - \underline{\phi}(\underline{y})\| \le L\|\underline{x} - \underline{y}\| \quad \text{for all } \underline{x}, \underline{y} \in \mathcal{M}\ . \tag{7.19}$$

A function $\underline{\phi}$ satisfying a condition of this type for any size of $L > 0$ is said to be **Lipschitz continuous** on $\mathcal{M}$ and the constant $L$ is called the **Lipschitz constant**. (Lipschitz continuity is a more stringent condition than continuity but it is less stringent than differentiability.) The contraction mapping theorem hence applies to functions which are Lipschitz continuous with in addition a Lipschitz constant $L < 1$.

### 7.3.2 Discussion about the contraction mapping theorem when the iteration function is differentiable

We note the following.

1. If in one-dimension $\phi : [a, b] \to [a, b]$ then the existence of a fixed point follows by considering the intermediate value theorem applied to the continuous function $g : [a, b] \to \mathbb{R}$ given by $g(x) := \phi(x) - x$. As $\phi(a) \in [a, b]$ and $\phi(b) \in [a, b]$ it follows that $g(a) \ge 0$ and $g(b) \le 0$. The intermediate value theorem implies that there exists $x^* \in [a, b]$ with $g(x^*) = 0$.

2. If the iteration function $\phi : \mathcal{I} \to \mathcal{I}$, $\mathcal{I} \subset \mathbb{R}$ is differentiable then the Mean Value Theorem gives
$$\phi(x) - \phi(y) = \phi'(\xi)(x - y) \quad \text{for some } \xi \text{ between } x \text{ and } y. \tag{7.20}$$
Thus a suitable $L$ is
$$L = \sup_{x \in \mathcal{I}} |\phi'(x)|\ . \tag{7.21}$$
In the case that $\mathcal{I} = [a, b]$ is a closed bounded interval we have
$$L = \max_{x \in \mathcal{I}} |\phi'(x)|\ . \tag{7.22}$$

3. In the $n$-dimensional case the Mean Value Theorem is a bit more involved. For $\underline{\phi} : \mathcal{M} \to \mathcal{M}$ we have
$$\underline{\phi}(\underline{x}) - \underline{\phi}(\underline{y}) = \left( \int_0^1 J_\phi(\underline{y} + t\underline{h})\, dt \right) (\underline{x} - \underline{y}), \quad \underline{h} = \underline{x} - \underline{y} \tag{7.23}$$
and thus for any subordinate matrix norm we have
$$\begin{aligned} \|\underline{\phi}(\underline{x}) - \underline{\phi}(\underline{y})\| &\le \|\left( \int_0^1 J_\phi(\underline{y} + t(\underline{x} - \underline{y}))\, dt \right)\|\, \|\underline{x} - \underline{y}\| && (7.24) \\ &\le \int_0^1 \|J_\phi(\underline{y} + t(\underline{x} - \underline{y}))\|\, dt\, \|\underline{x} - \underline{y}\| && (7.25) \\ &\le \max_{0 \le t \le 1} \|J_\phi(\underline{y} + t(\underline{x} - \underline{y}))\|\, \|\underline{x} - \underline{y}\|\ . && (7.26) \end{aligned}$$
Hence for all possible $\underline{x}, \underline{y} \in \mathcal{M}$ we can take for $L$ the value
$$L = \sup_{\underline{x} \in \mathcal{M}} \|J_\phi(\underline{x})\|\ . \tag{7.27}$$
The size of $L$ defined in (7.27) depends on which norm is used. As a technical point we note the matrix norm-eigenvalue inequality
$$\|J_\phi(\underline{x})\| \ge \rho(J_\phi(\underline{x})) = \text{Largest eigenvalue in magnitude of } J_\phi(\underline{x}). \tag{7.28}$$

4. The contraction mapping theorem gives the existence of a unique fixed point and the convergence of the iteration if the conditions of the theorem are satisfied. In some cases however it is not necessary to use the full force of the theorem. For example if the existence of the fixed point $\underline{x}^*$ is already established and there is an $L < 1$ in some ball $B(\underline{x}^*, r) = \{\underline{x} : \ \|\underline{x} - \underline{x}^*\| < r\}$ then if $\underline{x}_0 \in B(\underline{x}^*, r)$ we have

$$\|\underline{x}_k - \underline{x}^*\| = \|\underline{\phi}(\underline{x}_{k-1}) - \underline{\phi}(\underline{x}^*)\| \le L\|\underline{x}_{k-1} - \underline{x}^*\| \le \cdots \le L^k\|\underline{x}_0 - \underline{x}^*\| \to 0 \tag{7.29}$$

as $k \to \infty$. If at the fixed point $\|J_\phi(\underline{x}^*)\| < 1$ then by continuity $\|J_\phi(\underline{x})\| < 1$ for $\underline{x}$ in a neighbourhood of $\underline{x}^*$ which guarantees the existence of the ball and the local convergence. That is the iteration converges if $\underline{x}_0$ is sufficiently close to $\underline{x}^*$. In the 1D case this remark simply means that if $|\phi'(x^*)| < 1$ then the iteration will converge if $x_0$ is sufficiently close to $x^*$.

# 8 1D problems

In the case of convergent fixed point iterations of the form $x_{k+1} = \phi(x_k)$ in 1 dimension the order of convergence depends on the derivatives of $\phi$ at the fixed point $x^*$. As has already been shown, if $|\phi'(x^*)| < 1$ then we have convergence when $x_0$ is sufficiently close to $x^*$ but it is only linear convergence unless $\phi'(x^*) = 0$. In the latter case we have the following result.

**Theorem.** Assume that the iteration $x_{k+1} = \phi(x_k)$ converges to $x^*$ and that $\phi$ is $p$-times continuously differentiable. Then if

$$\phi^{(j)}(x^*) = 0, \quad j = 1, 2, \cdots, p-1 \quad \text{with } \phi^{(p)}(x^*) \neq 0 \tag{8.1}$$

then the order of convergence is the integer $p$ and the asymptotic error constant is

$$C = \frac{1}{p!} \left| \phi^{(p)}(x^*) \right| \ . \tag{8.2}$$

**Proof.** The result follows by considering the Taylor expansion of $\phi$ about $x^*$. With the Lagrange form of remainder we have

$$e_{k+1} = x_{k+1} - x^* = \phi(x_k) - \phi(x^*) = \sum_{j=1}^{p-1} (x_k - x^*)^j \frac{\phi^{(j)}(x^*)}{j!} + (x_k - x^*)^p \frac{\phi^{(p)}(\xi_k)}{p!} \tag{8.3}$$

for some $\xi_k$ between $x_k$ and $x^*$. Thus in our case

$$e_{k+1} = e_k^p \frac{\phi^{(p)}(\xi_k)}{p!} \tag{8.4}$$

and

$$\lim_{k\to\infty} \left| \frac{e_{k+1}}{e_k^p} \right| = \frac{1}{p!} \left| \phi^{(p)}(x^*) \right| \ . \tag{8.5}$$

## 8.1 Generation of the iteration function in 1D

We now consider two related approaches to constructing iteration functions $\phi$ in both one-point and multi-point cases which have 'good' local convergence properties. The approaches involve respectively the interpolation of $f$ and the interpolation of the inverse function $g := f^{(-1)}$ and such methods are sometimes called **direct interpolation methods** and **inverse interpolation methods** respectively. In the case of direct interpolation this involves approximating $y = f(x)$ at an iterate $x_k$ by a function $M_k(x)$ which interpolates $f(x)$ in some way and then determining $x_{k+1}$ as a root of $M_k(x)$, i.e.

$$M_k(x_{k+1}) = 0 \ . \tag{8.6}$$

For this to be practical $M_k(x)$ needs to be a function for which the root or roots can be found easily, e.g. linears, quadratics or even rational functions with a numerator which is a linear or a quadratic polynomial. In the case of inverse interpolation the approach is based on the observation that if

$$f(x^*) = 0 \tag{8.7}$$

then

$$x^* = g(0) = f^{(-1)}(0). \tag{8.8}$$

The approach is thus to interpolate $x = g(y)$ at $y_k = f(x_k)$ by a function $N_k(y)$ in some way and then simply to take

$$x_{k+1} = N_k(0) \; . \tag{8.9}$$

As this just involves evaluating the function $N_k(y)$, which is locally modelling $g(y)$, there is much less restriction on the types of functions $N_k(y)$ that can be used as compared with direct interpolation. Thus, as will be shown, it is possible to create methods of any integer order in the case of simple roots.

**Remarks**

1. Although it is possible to create methods of higher and higher order this does not imply that they are better methods as generally they involve more computation per iteration and the evaluation of higher order derivatives of $f$. Also you need to be sufficiently close to the root to take advantage of the high orders of convergence. In many practical situations most of the total computation time is in first getting close to $x^*$.

2. In the case of simple choices of $M_k$ and $N_k$, i.e. linears and even bilinears, the two approaches are identical. For this reason we consider only the inverse interpolation approach.

## 8.2 Some common interpolation based methods

In the following we will need the derivatives of $g(y) = f^{(-1)}(y)$. We have

$$y = f(x), \quad x = g(y) \tag{8.10}$$

and

$$\frac{dx}{dy} = 1/\left(\frac{dy}{dx}\right) = \frac{1}{f'(x)}, \quad \frac{d^2x}{dy^2} = \frac{dx}{dy}\frac{d}{dx}\left(\frac{dx}{dy}\right) = -\frac{f''(x)}{f'(x)^3} \; . \tag{8.11}$$

That is

$$g' = \frac{1}{f'} \quad \text{and} \quad g'' = -\frac{f''}{f'^3} \; . \tag{8.12}$$

### 8.2.1 Linear interpolation at $x_k$ using $f$ and $f'$ – Newton's method

$$N_k(y) = g(y_k) + (y - y_k)g'(y_k) = x_k + (y - f(x_k))\frac{1}{f'(x_k)} \; . \tag{8.13}$$

Thus

$$x_{k+1} = N_k(0) = x_k - \frac{f(x_k)}{f'(x_k)} \; . \tag{8.14}$$

By using theorem on page 8.1 this gives quadratic convergence if $f'(x^*) \neq 0$ and $f''(x^*) \neq 0$. In the case of a multiple root (i.e. $f'(x^*) = 0$) the convergence can be shown to be only linear.

### 8.2.2 Linear interpolation at $x_k$ and $x_{k-1}$ – secant method

$$\begin{aligned} N_k(y) &= g(y_k) + (y - y_k)\left(\frac{g(y_k) - g(y_{k-1})}{y_k - y_{k-1}}\right) && (8.15) \\ &= x_k + (y - f(x_k))\left(\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}\right) \; . && (8.16) \end{aligned}$$

Thus

$$x_{k+1} = N_k(0) = x_k - f(x_k)\left(\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}\right) . \tag{8.17}$$

Observe that this is a 2-point method, i.e. $x_{k+1} = \phi(x_k, x_{k-1})$, and thus the order of convergence cannot be deduced from the theorem on page 8.1. It can be shown that the order of convergence is

$$p = \frac{1+\sqrt{5}}{2} = 1.618 . \tag{8.18}$$

Establishing this 'rigorously' is in fact a little involved and it is not done in 'elementary' books on Numerical Analysis. Without too much difficulty you can show that if there is a $p$ then it is the number given and this is done in many books. However showing that the sequence converges in such a way that $e_{k+1}/e_k^p$ has a limit is more involved. We will not consider this further in these notes.

### 8.2.3 Inverse quadratic interpolation at $x_k$ using $f$, $f'$ and $f''$ – Chebyshev's method

This uses the truncated Taylor's series

$$N_k(y) = g(y_k) + (y - y_k)g'(y_k) + \frac{1}{2}(y - y_k)^2 g''(y_k) . \tag{8.19}$$

Thus

$$\begin{aligned} x_{k+1} &= N_k(0) = g(y_k) - y_k g'(y_k) + \frac{1}{2}y_k^2 g''(y_k) && (8.20)\\ &= x_k - \frac{f(x_k)}{f'(x_k)} + \frac{1}{2}f(x_k)^2\left(\frac{-f''(x_k)}{f'(x_k)^3}\right) . && (8.21)\end{aligned}$$

It can be shown that this is cubically convergent in general.

### 8.2.4 Bilinear interpolation at $x_k$ using $f$, $f'$ and $f''$ – Halley's method

Motivated by Newton's method we note that for any constant $b_0$ we satisfy

$$N_k(y_k) = g(y_k) = x_k \quad \text{and} \quad N_k'(y_k) = g'(y_k) = \frac{1}{f'(x_k)} \tag{8.22}$$

by the bilinear

$$\begin{aligned} N_k(y) &= x_k + \frac{(y - f(x_k))/f'(x_k)}{1 - b_0(y - f(x_k))} && (8.23)\\ &= x_k + \frac{y - f(x_k)}{f'(x_k)}\left(1 + b_0(y - f(x_k)) + \mathcal{O}((y - f(x_k))^2)\right) && (8.24)\\ &= x_k + \frac{1}{f'(x_k)}(y - f(x_k)) + \frac{b_0}{f'(x_k)^2}(y - f(x_k))^2 + \mathcal{O}((y - f(x_k))^3) . && (8.25)\end{aligned}$$

For the second derivative we have

$$g_k''(y_k) = -\frac{f''(x_k)}{f'(x_k)^3} = N_k''(y_k) = \frac{2b_0}{f'(x_k)}, \quad \text{provided} \quad b_0 = -\frac{1}{2}\frac{f''(x_k)}{f'(x_k)^2} . \tag{8.26}$$

Thus

$$x_{k+1} = N_k(0) = x_k - \frac{f(x_k)}{f'(x_k)(1 + b_0 f(x_k))}, \quad b_0 = -\frac{1}{2}\frac{f''(x_k)}{f'(x_k)^2} . \tag{8.27}$$

It can be shown that this is cubically convergent in general.

## 8.3 Obtaining the order of convergence using the interpolation error

For the 1-point methods, e.g. Newton's, Halley's and Chebyshev's methods, we can obtain the order of convergence by using theorem on page 8.1. This is straightforward but, for the higher order methods, it is a bit messy algebraically to construct $\phi'(x^*)$, $\phi''(x^*)$ and $\phi'''(x^*)$ etc. . A better approach is to use the properties of the interpolant. Typically we have the interpolation condition

$$(g - N_k)^{(j)}(y_k) = 0, \quad j = 0, 1, \cdots, p-1 \tag{8.28}$$

so that by Taylor's series

$$(g - N_k)(y) = \frac{1}{p!}(y - y_k)^p (g - N_k)^{(p)}(\xi_k) \tag{8.29}$$

for some $\xi_k$ between $y$ and $y_k$. In the case that $x^*$ is a simple zero of $f$ so that $g(y) = f^{(-1)}(y)$ is well defined in the vicinity of $y = 0$ and assuming that $g$ is differentiable as many times as we need we have

$$-e_{k+1} = x^* - x_{k+1} = g(0) - N_k(0) = \frac{(-1)^p}{p!} y_k^p (g - N_k)^{(p)}(\xi_k^{(0)}) \tag{8.30}$$

where $\xi_k^{(0)}$ is the $\xi_k$ value when $y = 0$. Now as we have a simple zero

$$y_k = f(x_k) = f(x_k) - f(x^*) = (x_k - x^*) f'(\eta_k) = e_k f'(\eta_k) \ . \tag{8.31}$$

Thus

$$\frac{e_{k+1}}{e_k^p} = \frac{(-1)^{p+1}}{p!} (f'(\eta_k))^p (g - N_k)^{(p)}(\xi_k^{(0)}) \ . \tag{8.32}$$

When $N_k$ is a polynomial of degree $\le p-1$ we have $N_k^{(p)}(y) = 0$ and thus

$$\lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k^p|} = C := \left| \frac{f'(x^*)^p}{p!} g^{(p)}(0) \right| \ . \tag{8.33}$$

For Newton's method we have $p = 2$ and

$$C := \left| \frac{f'(x^*)^2}{2} \frac{f''(x^*)}{f'(x^*)^3} \right| = \frac{1}{2} \left| \frac{f''(x^*)}{f'(x^*)} \right| \ . \tag{8.34}$$

For Chebyshev's method we have $p = 3$ and

$$C = \left| \frac{f'(x^*)^3}{6} g^{(3)}(0) \right| \ . \tag{8.35}$$

For Halley's method we also have $p = 3$ although it is noted here that to rigorously show that the iteration converges cubically you need to show that $\lim_{k\to\infty}(g - N_k)^{(3)}(\xi_k^{(0)})$ is bounded. As $N_k$ is a rational function we have $N_k^{(3)} \ne 0$. We do not consider this further.

The order of convergence for multi-point methods based on polynomial interpolation at more than one point can also be established by considering the error in the polynomial interpolant. The value $p = (1 + \sqrt{5})/2$ for the secant method is obtained in this way. We do not consider this here.

### 8.3.1 Some efficiency remarks

1. As has already been remarked, it does not follow that a higher order method is a 'better method' than a lower order method in the sense of obtaining a desired accuracy more quickly as this also depends on the amount of work per iteration. This is in turn will be dependent on the computational effort required to evaluate the function $f$ at the latest iterate $x_k$ and with some of these methods it will also depend on the cost of evaluating $f'$ and $f''$ at $x_k$. It is often the case that additionally computing $f'(x_k)$ or even $f''(x_k)$ is less work than is involved in first computing $f(x_k)$. For example if $f$ is of the form

$$f(x) = \mathrm{e}^x(A\cos x + B\sin x) + C$$

then $f'(x)$ and $f''(x)$ are also of this form and you can implement the methods so that when $f(x)$, $f'(x)$ and $f''(x)$ are computed you only compute $\mathrm{e}^x$, $\cos x$ and $\sin x$ once. Thus Newton's method compares favourably with the other methods in such cases.

2. The efficiency index *EI* of a method is defined as

$$EI = p^{1/\theta} \tag{8.36}$$

where $p$ is the order and where $\theta$ is the amount of work per iteration. This number can be regarded as a measure of the computation required to get another significant digit of accuracy when the function $f$ under consideration gives an error constant of 1. If $\theta_N$ and $\theta_S$ are the amounts of work per iteration for respectively Newton's method and the secant method then we have

$$2^{1/\theta_N} \quad \text{and} \quad (1.618)^{1/\theta_S} \tag{8.37}$$

for the efficiency indices. The secant method has a larger efficiency index if

$$(1.618)^{1/\theta_S} > 2^{1/\theta_N}, \quad \text{i.e. } \frac{\log(1.618)}{\theta_S} > \frac{\log 2}{\theta_N}, \quad \frac{\theta_S}{\theta_N} < \frac{\log(1.618)}{\log 2}\ . \tag{8.38}$$

Both methods involve evaluating $f$ at the latest $x_k$. The secant method already has the value $f(x_{k-1})$ and thus this does not need to be recomputed. With Newton's method you additionally need to determine $f'(x_k)$. If computing $f$ is 1 unit of work and computing $f'$ is $\alpha$ units of work and all other computations are negligible then

$$\frac{\theta_S}{\theta_N} \approx \frac{1}{1+\alpha}\ . \tag{8.39}$$

Hence if

$$\frac{1}{1+\alpha} < \frac{\log 1.618}{\log 2}, \quad \text{i.e. if } 0 \le \alpha < \frac{\log 2 - \log 1.618}{\log 1.618} = 0.44 \tag{8.40}$$

then Newton's method may be considered to be more efficient but if

$$\alpha > 0.44 \tag{8.41}$$

then the secant method may be considered to be more efficient.

The following example indicates typically what you obtain with Newton's method and Secant's method. For the function $f(x) = \mathrm{e}^x - 2 - x^2$ we note that $f(0) = -1$ and $f(2) = \mathrm{e}^2 - 6 > 0$ indicating that $f$ has a root in $(0, 2)$. Starting Newton's method with $x_0 = 0$ and Secant's method with $x_0 = 0$, $x_1 = 2$ we get the following.

| $n$ | Newton's $x_n$ | $f(x_n)$ | Secant's $x_n$ | $f(x_n)$ |
|---|---|---|---|---|
| 0 | 0.00000000000000E+00 | -0.1000E+01 | 0.00000000000000E+00 | -0.1000E+01 |
| 1 | 0.10000000000000E+01 | -0.2817E+00 | 0.20000000000000E+01 | +0.1389E+01 |
| 2 | 0.13922111911773E+01 | +0.8549E-01 | 0.83715070604462E+00 | -0.3910E+00 |
| 3 | 0.13232331920681E+01 | +0.4598E-02 | 0.10926004398155E+01 | -0.2118E+00 |
| 4 | 0.13190873294921E+01 | +0.1504E-04 | 0.13943126382146E+01 | +0.8809E-01 |
| 5 | 0.13190736770045E+01 | +0.1622E-09 | 0.13056716631345E+01 | -0.1461E-01 |
| 6 | 0.13190736768574E+01 | +0.2220E-15 | 0.13182823563273E+01 | -0.8713E-03 |
| 7 | 0.13190736768574E+01 | -0.6661E-15 | 0.13190820588616E+01 | +0.9235E-05 |
| 8 | | | 0.13190736716198E+01 | -0.5771E-08 |
| 9 | | | 0.13190736768573E+01 | -0.3753E-13 |
| 10 | | | 0.13190736768574E+01 | -0.6661E-15 |

Both methods converge rapidly with, as this example shows, Secant's method taking 2 or 3 more iterations to obtain 12 digit accuracy. The iterations were stopped in each case when the difference between two consecutive iterates was less than $10^{-12}$. In this example, there are 11 function evaluations in Secant's method and there are 8 function evaluations and 7 derivative evaluations in Newton's method.

### 8.3.2 Other remarks

1. The secant method can be regarded as just a modification of Newton's method with $1/f'(x_k)$ replaced by the difference quotient

$$\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} . \tag{8.42}$$

   When this is implemented on a computer and the iteration is 'near' to convergence we can expect a loss of significance in evaluating this quotient due to the subtraction of nearly equal numbers in the numerator and denominator. For example if the precision of the computer is about 16 decimal digits, $f(x_0)$ is $\mathcal{O}(1)$ in magnitude with the latest iterate also of $\mathcal{O}(1)$ and $x_k - x_{k-1}$ and $f(x_k) - f(x_{k-1})$ are both $\mathcal{O}(10^{-8})$ then we can only expect about 8 significant figures of accuracy in computing the quotient. In practice this does not however matter since $f(x_k)$ should also be sufficiently small at this stage that we can compute the change

$$f(x_k)\left(\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}\right) \tag{8.43}$$

   to sufficient accuracy for our needs. The only care that needs to be taken is to stop the iteration when the difference $|x_k - x_{k-1}|$ is sufficiently small when the method can be judged to have converged.

2. Newton's method can also be modified by replacing $1/f'(x_k)$ by the difference quotient

$$\frac{h}{f(x_k + h) - f(x_k)} \tag{8.44}$$

   which involves evaluating $f$ at 2 different points per iteration. The iteration function is now

$$\phi(x) = x - \frac{hf(x)}{f(x+h) - f(x)} . \tag{8.45}$$

For fixed $h \neq 0$ convergence is linear with

$$\phi'(x^*) = 1 - \frac{hf'(x^*)}{f(x^*+h)} = \frac{f(x^*+h) - hf'(x^*)}{f(x^*+h)} \tag{8.46}$$

and when $h$ is small we have

$$\phi'(x^*) \approx \frac{(h^2/2)f''(x^*)}{hf'(x^*)} = \frac{h}{2}\frac{f''(x^*)}{f'(x^*)} \tag{8.47}$$

and the error constant is thus also small in general when $h$ is small.

This is often a useful modification of Newton's method to implement at the development stage of implementing the full Newton method as a check that the derivativa and the function specified in a computer program are consistent.

3. A further modification of Newton's method results if we do not re-compute the derivative at each stage but instead take

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_j)}, \quad j \le k\ . \tag{8.48}$$

If $j$ remains fixed as $k \to \infty$ then we have the iteration function

$$\phi(x) = x - \frac{f(x)}{f'(x_j)}, \quad \phi'(x^*) = 1 - \frac{f'(x^*)}{f'(x_j)}\ . \tag{8.49}$$

Thus provided $|\phi'(x^*)| < 1$ we have linear convergence for sufficiently close starting iterates. If $x_j$ is close to $x^*$ then $|\phi'(x^*)|$ is small and convergence will be rapid.

## 8.4 Bracketing methods

A drawback of the methods so far considered is that they will only converge if the starting point $x_0$ is sufficiently close to a root of $x^*$. We now consider a class of methods which will always converge provided the function $f$ is continuous on an interval $(a, b)$ and we know that

$$f(a)f(b) < 0, \tag{8.50}$$

i.e. $f$ changes sign in $(a, b)$ and consequently it must have at least one root in $(a, b)$. In all these methods we generate a sequence of smaller and smaller nested intervals

$$[a_{k+1}, b_{k+1}] \subset [a_k, b_k] \tag{8.51}$$

in which we know that $f$ has a root. Each of the intervals computed hence brackets a root.

### 8.4.1 The bisection method

The simplest method to implement, which is often given as an early programming exercise, is to repeatedly half the interval by setting

$$c_k = \frac{1}{2}(a_k + b_k), \tag{8.52}$$

evaluating $f(c_k)$ and then setting

$$[a_{k+1}, b_{k+1}] = \begin{cases} [a_k, c_k], & f(a_k)f(c_k) < 0, \\ [c_k, b_k], & \text{otherwise.} \end{cases} \tag{8.53}$$

If $[a_0, b_0] = [a, b]$ then we know that after $k$ steps that

$$b_k - a_k = 2^{-k}(b - a) \ . \tag{8.54}$$

Convergence is thus guaranteed but the rate of convergence is slow. In addition the method takes no account of the actual values $f(a_k)$, $f(c_k)$ and $f(b_k)$ other than their signs.

### 8.4.2 The method of False Position

This is a method in which the intermediate point $c_k \in (a_k, b_k)$ is chosen as the root of the linear interpolant

$$M_k(x) = f(a_k) + (x - a_k)\left(\frac{f(b_k) - f(a_k)}{b_k - a_k}\right) \ . \tag{8.55}$$

Thus

$$c_k = a_k - f(a_k)\left(\frac{b_k - a_k}{f(b_k) - f(a_k)}\right) = a_k + \frac{|f(a_k)|}{|f(a_k)| + |f(b_k)|}(b_k - a_k) \ . \tag{8.56}$$

The new interval $[a_{k+1}, b_{k+1}]$ is then selected as in the bisection method.

The idea of using the linear interpolant is good but unfortunately if the method is implemented precisely as given then it is a poor method with in general $b_k \to x^*$ or $a_k \to x^*$ but with $b_k - a_k \not\to 0$, i.e. the convergence is usually only from one side of the root. The convergence is only linear and can even be slower than in the bisection method. This is described in many text books and the analysis, which is quite short and straightforward, will not be considered here. However to show what typically happens consider again the example involving $f(x) = e^x - 2 - x^2$ considered in the discussion of Newton's method and Secant's method. The method of false position generates the following.

| $n$ | $a_n$ | $f(a_n)$ | $b_n$ | $f(b_n)$ |
|---|---|---|---|---|
| 0 | 0.00000000000000E+00 | -0.1000E+01 | 0.20000000000000E+01 | 0.1389E+01 |
| 1 | 0.83715070604462E+00 | -0.3910E+00 | 0.20000000000000E+01 | 0.1389E+01 |
| 2 | 0.10926004398155E+01 | -0.2118E+00 | 0.20000000000000E+01 | 0.1389E+01 |
| 3 | 0.12126321517279E+01 | -0.1082E+00 | 0.20000000000000E+01 | 0.1389E+01 |
| 4 | 0.12695090537394E+01 | -0.5255E-01 | 0.20000000000000E+01 | 0.1389E+01 |
| 5 | 0.12961364359018E+01 | -0.2482E-01 | 0.20000000000000E+01 | 0.1389E+01 |
| 6 | 0.13084935430039E+01 | -0.1156E-01 | 0.20000000000000E+01 | 0.1389E+01 |
| 7 | 0.13142012064029E+01 | -0.5348E-02 | 0.20000000000000E+01 | 0.1389E+01 |
| 8 | 0.13168314441500E+01 | -0.2466E-02 | 0.20000000000000E+01 | 0.1389E+01 |
| 9 | 0.13180421983538E+01 | -0.1136E-02 | 0.20000000000000E+01 | 0.1389E+01 |
| 10 | 0.13185992498397E+01 | -0.5225E-03 | 0.20000000000000E+01 | 0.1389E+01 |

In this case we get $a_n \to x^*$ but $b_n = 2$ for all $n$.

### 8.4.3 The Illinois and Pegasus methods

By making relatively small changes to the method of false position we can create methods with vastly better performance. The changes are made in order to 'encourage' both end points to change which in turn requires a small amount of housekeeping operations. To describe the modified methods let $L_0 = f(a_0)$, $R_0 = f(b_0)$ and let $c_0$ and $c_1$ be as in the method of false position, i.e.

$$c_0 = a_0 + \frac{|L_0|}{|L_0| + |R_0|}(b_0 - a_0) \tag{8.57}$$

the next interval is

$$[a_1, b_1] = \begin{cases} [c_0, b_0], & L_1 = f(c_0),\ R_1 = R_0 \quad \text{if } f(c_0)f(b_0) < 0, \\ [a_0, c_0], & L_1 = L_0,\ R_1 = f(c_0) \quad \text{otherwise} \end{cases} \tag{8.58}$$

and

$$c_1 = a_1 + \frac{|L_1|}{|L_1| + |R_1|}(b_1 - a_1)\ . \tag{8.59}$$

If $c_0$ and $c_1$ are both on the same side of the root then we proceed in a different way to the method of false position. We define

$$[a_2, b_2] = \begin{cases} [c_1, b_1], & L_2 = f(c_1),\ R_2 = \alpha_1 R_1 \quad \text{if } f(c_1)f(b_1) < 0, \\ [a_1, c_1], & L_2 = \alpha_1 L_1,\ R_2 = f(c_1) \quad \text{otherwise,} \end{cases} \tag{8.60}$$

where $0 < \alpha_1 < 1$ and compute $c_2$ from

$$c_2 = a_2 + \frac{|L_2|}{|L_2| + |R_2|}(b_2 - a_2)\ . \tag{8.61}$$

That is we replace the function value in the formula at the end which is not changing by something which has the same sign but which is smaller in magnitude. The procedure then continues in this way.

If $c_{k-1}$ and $c_k$ are on opposite sides of the root then we do one step of the false position, i.e.

$$[a_{k+1}, b_{k+1}] = \begin{cases} [c_k, b_k], & L_{k+1} = f(c_k),\ R_{k+1} = R_k \quad \text{if } f(c_k)f(b_k) < 0, \\ [a_k, c_k], & L_{k+1} = L_k,\ R_{k+1} = f(c_k) \quad \text{otherwise.} \end{cases} \tag{8.62}$$

If $c_{k-1}$ and $c_k$ are on the same side of the root then instead we take

$$[a_{k+1}, b_{k+1}] = \begin{cases} [c_k, b_k], & L_{k+1} = f(c_k),\ R_{k+1} = \alpha_k R_k \quad \text{if } f(c_k)f(b_k) < 0, \\ [a_k, c_k], & L_{k+1} = \alpha_k L_k,\ R_{k+1} = f(c_k) \quad \text{otherwise.} \end{cases} \tag{8.63}$$

At each stage $L_k$ and $R_k$ have opposite sign with one of $L_k$ and $R_k$ being a 'true' function value whilst the other may be a function value multiplied by a positive number which is less than one. In the **Illinios method** $\alpha_k = 1/2$ whilst in the **Pegasus method** we take

$$\alpha_k = \begin{cases} \dfrac{|L_k|}{|L_k| + |L_{k+1}|} & \text{if the right end point is 'diminished'.} \\ \dfrac{|R_k|}{|R_k| + |R_{k+1}|} & \text{if the left end point is 'diminished'.} \end{cases} \tag{8.64}$$

In both cases it can be shown that $b_k - a_k \to 0$ with an 'average' order of convergence of respectively

$$3^{1/3} = 1.442 \quad \text{and} \quad (7.275)^{1/4} = 1.642\ . \tag{8.65}$$

If we consider again the problem of finding a root of $f(x) = \mathrm{e}^x - 2 - x^2$ the Illinois method gives

| $n$ | $a_n$ | $f(a_n)$ | $b_n$ | $f(b_n)$ |
|---|---|---|---|---|
| 0 | 0.00000000000000E+00 | -0.1000E+01 | 0.20000000000000E+01 | 0.1389E+01 |
| 1 | 0.83715070604462E+00 | -0.3910E+00 | 0.20000000000000E+01 | 0.1389E+01 |
| 2 | 0.10926004398155E+01 | -0.2118E+00 | 0.20000000000000E+01 | 0.1389E+01 |
| 3 | 0.13046179743318E+01 | -0.1575E-01 | 0.20000000000000E+01 | 0.1389E+01 |
| 4 | 0.13046179743318E+01 | -0.1575E-01 | 0.13347837397916E+01 | 0.1753E-01 |
| 5 | 0.13188943930168E+01 | -0.1975E-03 | 0.13347837397916E+01 | 0.1753E-01 |
| 6 | 0.13190714556289E+01 | -0.2447E-05 | 0.13347837397916E+01 | 0.1753E-01 |
| 7 | 0.13190714556289E+01 | -0.2447E-05 | 0.13190758424345E+01 | 0.2386E-05 |
| 8 | 0.13190736768536E+01 | -0.4185E-11 | 0.13190758424345E+01 | 0.2386E-05 |
| 9 | 0.13190736768574E+01 | 0.0000E+00 | 0.13190758424345E+01 | 0.2386E-05 |

The method, in this example, compares well with Secant's method and by chance, in my computer implementation, $f(a_9)$ evaluates to 0 on the computer used which is why the iteration was stopped at this stage. You need to do an additional calculation to produce a $b_n$ value which differs from the last $a_n$ used by less than $10^{-12}$.

## 8.5 The Aitken $\delta^2$ process

The Aitken $\delta^2$ process is a way of speeding up the convergence of a **linearly convergent sequence** $(x_k)$. In case of a linearly convergent iteration of the form $x_{k+1} = \phi(x_k)$ we now show that it leads to a quadratically convergent iteration.

For the analysis let

$$C = \phi'(x^*) \quad \text{and} \quad C_2 = \frac{1}{2}\phi''(x^*) \; . \tag{8.66}$$

Then the errors $e_{k+1} = x_{k+1} - x^*$ and $e_{k+2} = x_{k+2} - x^*$ satisfy

$$\begin{aligned} e_{k+1} &= \phi(x_k) - \phi(x^*) = Ce_k + C_2 e_k^2 + \mathcal{O}(e_k^3) && (8.67)\\ e_{k+2} &= \phi(x_{k+1}) - \phi(x^*) = Ce_{k+1} + C_2 e_{k+1}^2 + \mathcal{O}(e_{k+1}^3) && (8.68) \end{aligned}$$

giving

$$\begin{aligned} e_{k+1}^2 - e_k e_{k+2} &= C_2 e_k e_{k+1}(e_k - e_{k+1}) + \mathcal{O}(e_k^4) && (8.69)\\ &= C_2 C(1-C) e_k^3 + \mathcal{O}(e_k^4) \; . && (8.70) \end{aligned}$$

If the right hand side is 0 then we have an equation for $x^*$. Based on this observation we take $\hat{x}_{k+2}$ such that

$$(x_k - \hat{x}_{k+2})^2 - (x_k - \hat{x}_{k+2})(x_{k+2} - \hat{x}_{k+2}) = 0, \tag{8.71}$$

i.e.

$$\hat{x}_{k+2} = \frac{x_k x_{k+2} - x_{k+1}^2}{x_k - 2x_{k+1} + x_{k+2}} \; . \tag{8.72}$$

$\hat{x}_{k+2}$ is our improved estimate of $x^*$ and note that the change from $x_{k+2}$ to $\hat{x}_{k+2}$ is

$$\hat{x}_{k+2} - x_{k+2} = -\frac{(x_{k+1} - x_{k+2})^2}{x_k - 2x_{k+1} + x_{k+2}} \; . \tag{8.73}$$

To determine by how much $\hat{x}_{k+2}$ is likely to be a better estimate observe that if $\hat{e}_{k+2} = \hat{x}_{k+2} - x^*$ then

$$\hat{e}_{k+2} - e_{k+2} = \hat{x}_{k+2} - x_{k+2} = -\frac{(e_{k+1} - e_{k+2})^2}{e_k - 2e_{k+1} + e_{k+2}} \; . \tag{8.74}$$

Thus

$$\hat{e}_{k+2} = \frac{e_{k+2}(e_k - 2e_{k+1} + e_{k+2}) - (e_{k+1} - e_{k+2})^2}{e_k - 2e_{k+1} + e_{k+2}} = \frac{e_k e_{k+2} - e_{k+1}^2}{e_k - 2e_{k+1} + e_{k+2}} \tag{8.75}$$

$$= \frac{C_2 C(C-1)e_k^3 + \mathcal{O}(e_k^4)}{(C-1)^2 e_k + \mathcal{O}(e_k^2)} = \left(\frac{C_2 C}{C-1}\right) e_k^2 + \mathcal{O}(e_k^3)\ . \tag{8.76}$$

This is a quadratic convergence result. If $\hat{x}_0 = x_0$, $x_1 = \phi(x_0)$, $x_2 = \phi(x_1)$, $\hat{x}_2$ is computed from the above, $x_3 = \phi(\hat{x}_2)$, $x_4 = \phi(x_3)$, $\hat{x}_4$ is computed from the above, etc. then the subsequence $\hat{x}_0$, $\hat{x}_2$, $\hat{x}_4, \cdots$ is quadratically convergent.

## 8.6 Multiple roots

If $x^*$ is a multiple root of $f$ with multiplicity $q$, i.e.

$$f^{(r)}(x^*) = 0, \quad r = 0, 1, \cdots, q-1 \tag{8.77}$$

which implies that

$$f(x) = (x - x^*)^q f_0(x), \quad f_0(x^*) \neq 0 \tag{8.78}$$

then the standard Newton's method will converge to $x^*$ if $x_0$ is a sufficiently good starting point but it will only converge linearly. The following modification of Newton's method, sometimes known as Schröder's method, will however converge quadratically

$$x_{k+1} = x_k - q\frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \cdots\ . \tag{8.79}$$

To show this let

$$\phi(x) = x - q\frac{f(x)}{f'(x)}\ . \tag{8.80}$$

Then

$$\phi' = 1 - q\left(\frac{f'^2 - f\,f''}{f'^2}\right) = (1-q) + q\frac{f\,f''}{f'^2}\ . \tag{8.81}$$

From (8.78) we have

$$f(x) = (x - x^*)^q f_0(x), \tag{8.82}$$

$$f'(x) = q(x - x^*)^{q-1} f_0(x) + \mathcal{O}((x - x^*)^q), \tag{8.83}$$

$$f''(x) = q(q-1)(x - x^*)^{q-2} f_0(x) + \mathcal{O}((x - x^*)^{q-1}), \tag{8.84}$$

As $x \to x^*$ we have

$$q\frac{f(x)f''(x)}{(f'(x))^2} \to q\frac{q(q-1)}{q^2} = q - 1 \tag{8.85}$$

and hence

$$\phi'(x) \to 0 \quad \text{as } x \to x^* \tag{8.86}$$

indicating quadratic convergence in general.

# 9 Newton's method for systems

As was introduced briefly in section 1.6, Newton's method for systems is based on approximating $\underline{f}(\underline{x})$ about $\underline{x} = \underline{x}_k$ by a linear function. That is, for the $i$th component

$$f_i(\underline{x}_k + \underline{h}) = f_i(\underline{x}_k) + (\nabla f_i(\underline{x}_k))^T \underline{h} + \mathcal{O}(\|\underline{h}\|^2) \tag{9.1}$$

and for the vector $\underline{f}$ we have

$$\underline{f}(\underline{x}) = \underline{f}(\underline{x}_k) + J_f(\underline{x}_k)(\underline{x} - \underline{x}_k) + \mathcal{O}(\|\underline{x} - \underline{x}_k\|^2) \ . \tag{9.2}$$

The linear function is

$$\underline{M}_k(\underline{x}) = \underline{f}(\underline{x}_k) + J_f(\underline{x}_k)(\underline{x} - \underline{x}_k) \ . \tag{9.3}$$

$\underline{x}_{k+1}$ is defined by $\underline{M}_k(\underline{x}_{k+1}) = \underline{0}$ giving

$$\underline{x}_{k+1} = \underline{x}_k - (J_f(\underline{x}_k))^{-1} \underline{f}(\underline{x}_k) \ . \tag{9.4}$$

This is usually implemented by first solving

$$J_f(\underline{x}_k)\underline{s}_k = -\underline{f}(\underline{x}_k) \tag{9.5}$$

and then letting

$$\underline{x}_{k+1} = \underline{x}_k + \underline{s}_k \ . \tag{9.6}$$

## 9.1 The quadratic convergence of Newton's method

In the one-dimensional case sufficient conditions for Newton's method to converge quadratically (or possibly faster) to $x^*$ are that $f$ is 2 times continuously differentiable, $f'(x^*) \neq 0$ and $x_0$ is sufficiently close to $x^*$. A corresponding result in $\mathbb{R}^n$ requires that $J_f(\underline{x}^*)$ is non-singular (i.e. invertible), $\underline{x}_0$ is a sufficiently close starting point and that $J_f$ is Lipschitz continuous in the vicinity of $\underline{x}^*$. More specifically let

$$\mathcal{N}(\underline{x}^*, r) = \{\underline{x} : \quad \|\underline{x} - \underline{x}^*\| < r\} \tag{9.7}$$

denote a neighbourhood of $\underline{x}^*$ in which there exists $L \geq 0$ such that

$$\|J_f(\underline{x}) - J_f(\underline{y})\| \leq L\|\underline{x} - \underline{y}\| \quad \forall \underline{x}, \underline{y} \in \mathcal{N}(\underline{x}^*, r) \tag{9.8}$$

and let

$$\beta = \|J_f(\underline{x}^*)^{-1}\| < \infty \ . \tag{9.9}$$

How close $\underline{x}_0$ needs to be to $\underline{x}^*$ to guarantee convergence depends on both $L$ and $\beta$. The Lipschitz constant $L$ indicates the degree of non-linearity of $\underline{f}$ since if $\underline{f}$ were linear then $J_f(\underline{x})$ would be constant (i.e. independent of $\underline{x}$) and we could take $L = 0$. The constant $\beta$ indicates how close $J_f(\underline{x})$ is to a singular matrix. The best situation, as regards the convergence of the iteration, is when $L$ and $\beta$ are both small which we now make precise by some analysis. We prove the quadratic convergence result by a number of steps.

(i) The interpolation error:

We have already given the error in the form of $\mathcal{O}(\|\underline{h}\|^2)$ in (1.12). For a bound note that

$$\begin{aligned}
\underline{f}(\underline{x}) - \underline{M}_k(\underline{x}) &= (\underline{f}(\underline{x}) - \underline{f}(\underline{x}_k)) - J_f(\underline{x}_k)(\underline{x} - \underline{x}_k) && (9.10)\\
&= \int_0^1 J_f(\underline{x}_k + t(\underline{x} - \underline{x}_k))\, dt\, (\underline{x} - \underline{x}_k) - J_f(\underline{x}_k)(\underline{x} - \underline{x}_k) && (9.11)\\
&= \int_0^1 (J_f(\underline{x}_k + t(\underline{x} - \underline{x}_k)) - J_f(\underline{x}_k))\, dt\, (\underline{x} - \underline{x}_k) \ . && (9.12)
\end{aligned}$$

This is an expression for the interpolation error. Then taking norms and using the Lipschitz continuity gives

$$\|\underline{f}(\underline{x}) - \underline{M}_k(\underline{x})\| \le L\|\underline{x} - \underline{x}_k\|^2 \int_0^1 t\,dt = \frac{L}{2}\|\underline{x} - \underline{x}_k\|^2 \ . \qquad (9.13)$$

(ii) The non-singularity of $J_f(\underline{x})$:

The idea here is to show that $J_f(\underline{x}^*)$ being non-singular and $J_f(\underline{x})$ being Lipschitz continuous guarantees that $J_f(\underline{x})$ is non-singular in a neighbourhood of $\underline{x}^*$. This can be done as follows. From

$$J_f(\underline{x}) = J_f(\underline{x}^*) + (J_f(\underline{x}) - J_f(\underline{x}^*)) = J_f(\underline{x}^*)\left(I + J_f(\underline{x}^*)^{-1}(J_f(\underline{x}) - J_f(\underline{x}^*))\right) \qquad (9.14)$$

we have

$$J_f(\underline{x}) = J_f(\underline{x}^*)(I - E), \quad E = -J_f(\underline{x}^*)^{-1}(J_f(\underline{x}) - J_f(\underline{x}^*)). \qquad (9.15)$$

Provided $I - E$ is invertible we have

$$J_f(\underline{x})^{-1} = (I - E)^{-1} J_f(\underline{x}^*)^{-1} \ . \qquad (9.16)$$

We showed in section 6 that when $E$ is such that $\|E\| < 1$, the matrix $I - E$ is non-singular with

$$(I - E)^{-1} = I + E + E^2 + \cdots \quad \text{and} \quad \|(I - E)^{-1}\| \le \sum_{k=0}^{\infty} \|E\|^k = \frac{1}{1 - \|E\|} \ . \qquad (9.17)$$

Hence if $\underline{x} \in \mathcal{N}(\underline{x}^*, r)$ then $\|E\| \le \beta L r$ and provided $\beta L r < 1$ we have

$$\|J_f(\underline{x})^{-1}\| \le \frac{\beta}{1 - \|E\|} \le \frac{\beta}{1 - \beta L r} \ . \qquad (9.18)$$

If we take $r = 1/(2\beta L)$ then in the neighbourhood $\mathcal{N}(\underline{x}^*, r)$ we have $\|J_f(\underline{x})^{-1}\| \le 2\beta$.

(iii) The convergence:

From (ii) we assume that $\underline{x}_0 \in \mathcal{N}(\underline{x}^*, r)$ with $r = 1/(2\beta L)$. We need now to relate $\|\underline{x}_1 - \underline{x}^*\|$ to $\|\underline{x}_0 - \underline{x}^*\|$. We have

$$\begin{aligned} \underline{x}_1 - \underline{x}^* &= \underline{x}_0 - (J_f(\underline{x}_0))^{-1} \underline{f}(\underline{x}_0) - \underline{x}^* && (9.19) \\ &= (J_f(\underline{x}_0))^{-1} \left(J_f(\underline{x}_0)(\underline{x}_0 - \underline{x}^*) - \underline{f}(\underline{x}_0)\right) && (9.20) \end{aligned}$$

and since $\underline{f}(\underline{x}^*) = \underline{0}$ we have

$$\begin{aligned} \underline{x}_1 - \underline{x}^* &= (J_f(\underline{x}_0))^{-1} \left(J_f(\underline{x}_0)(\underline{x}_0 - \underline{x}^*) + \underline{f}(\underline{x}^*) - \underline{f}(\underline{x}_0)\right) && (9.21) \\ &= (J_f(\underline{x}_0))^{-1} \left(\underline{f}(\underline{x}^*) - \underline{M}_0(\underline{x}^*)\right) \ . && (9.22) \end{aligned}$$

Using the bound on $\|\underline{f} - \underline{M}_0\|$ we have, since $\|\underline{x}_0 - \underline{x}^*\| \le r = 1/(2\beta L)$,

$$\begin{aligned} \|\underline{x}_1 - \underline{x}^*\| &\le 2\beta \frac{L}{2} \|\underline{x}_0 - \underline{x}^*\|^2 = \beta L \|\underline{x}_0 - \underline{x}^*\|^2 && (9.23) \\ &\le \beta L r \|\underline{x}_0 - \underline{x}^*\| = \frac{1}{2}\|\underline{x}_0 - \underline{x}^*\| \ . && (9.24) \end{aligned}$$

Hence the point $\underline{x}_1$ is nearer to $\underline{x}^*$ than is $\underline{x}_0$. This implies that $\underline{x}_1$ is well defined and similarly for $\underline{x}_2$, $\underline{x}_3, \cdots$ and it also shows that the points converge to $\underline{x}^*$ with

$$\|\underline{x}_{k+1} - \underline{x}^*\| \le \beta L \|\underline{x}_k - \underline{x}^*\|^2 \le \frac{1}{2}\|\underline{x}_k - \underline{x}^*\| \le \cdots \le \frac{1}{2^{k+1}}\|\underline{x}_0 - \underline{x}^*\| \ . \qquad (9.25)$$

Thus the error always decreases by at least 1/2 and is eventually at least quadratic.

# 10 Globally convergent approaches and modifications of Newton's method

Newton's method works well when you are sufficiently close to the desired solution but it may not converge otherwise. Modifications of the basic Newton scheme are described in these notes which increase the likelihood of convergence of Newton's method from starting points which are not so close to the solution. First however we consider a strategy for generating sufficiently good starting points.

## 10.1 A continuation method

For the nonlinear equations problem

$$\underline{f}(\underline{x}) = \underline{0} \tag{10.1}$$

the idea of a one-parameter continuation technique is to construct a one-parameter family of functions $\underline{f}^{(\theta)}(\underline{x})$, which varies continuously with $\theta$ for $0 \le \theta \le 1$, such that the solution of

$$\underline{f}^{(0)}(\underline{x}) = \underline{0} \tag{10.2}$$

is known and such that

$$\underline{f}^{(1)}(\underline{x}) = \underline{f}(\underline{x}) \ . \tag{10.3}$$

Provided each nonlinear system

$$\underline{f}^{(\theta)}(\underline{x}(\theta)) = \underline{0} \tag{10.4}$$

has a solution then the continuity of $\underline{f}^{(\theta)}$ with respect to $\theta$ gives a continuous path $\underline{x}(\theta)$, $0 \le \theta \le 1$ from the known solution when $\theta = 0$ to the solution of the desired problem when $\theta = 1$. Provided the Jacobian of each function $\underline{f}^{(\theta)}$ is nonsingular at its solution point $\underline{x}(\theta)$ then with

$$0 = \theta_0 < \theta_1 < \cdots < \theta_M = 1 \tag{10.5}$$

the vector $\underline{x}(\theta_i)$ will be sufficiently close to $\underline{x}(\theta_{i+1})$ for Newton's method to converge provided $\theta_{i+1} - \theta_i$ is sufficiently small. Thus, if all the above conditions are satisfied, Newton's method and a continuation technique will lead to the required solution.

As examples of one-parameter parametrizations we have

$$\underline{f}^{(\theta)}(\underline{x}) = \underline{f}(\underline{x}) - \underline{f}(\underline{x}_0)(1-\theta), \quad 0 \le \theta \le 1 \tag{10.6}$$

and

$$\underline{f}^{(\theta)}(\underline{x}) = (1-\theta)(\underline{x} - \underline{x}_0) + \theta \underline{f}(\underline{x}), \quad 0 \le \theta \le 1 \tag{10.7}$$

with in each case $\underline{x}(0) = \underline{x}_0$.

As a final note concerning implementation, it is frequently not necessary to solve each intermediate system very accurately. All that is needed is a sufficiently good starting value for the next value of $\theta$.

## 10.2 The descent idea

As has already been indicated in this module, the solving of $\underline{f}(\underline{x}) = \underline{0}$ corresponds to the minimisation of $\|\underline{f}\|_2^2 = \underline{f}^T \underline{f}$ and we can regard making progress towards the solution if at each stage $\|\underline{f}(\underline{x}_{k+1})\|_2 < \|\underline{f}(\underline{x}_k)\|_2$. Likewise for the problem of minimising $f(\underline{x})$ (i.e. $f : \mathbb{R}^n \to \mathbb{R}$) we would consider that we are making progress towards the solution if $f(\underline{x}_{k+1}) < f(\underline{x}_k)$. This leads naturally to the consideration of descent directions at $\underline{x}_k$.

For a problem involving $f(\underline{x})$ recall that we have the Taylor expansion

$$f(\underline{x}_k + t\underline{p}) = f(\underline{x}_k) + (\nabla f(\underline{x}_k))^T t\underline{p} + \mathcal{O}\left(t^2\right), \tag{10.8}$$

assuming that $t$ is small in magnitude, with $\underline{p}$ being a **descent direction** if

$$(\nabla f(\underline{x}_k))^T \underline{p} < 0 . \tag{10.9}$$

If $\underline{p}$ is such a direction then for sufficiently small $t > 0$ we have $f(\underline{x}_k + t\underline{p}) < f(\underline{x}_k)$. Hence about half of all the directions are descent directions. We now consider specific descent directions.

### 10.2.1 Steepest descent

If we assume that $\underline{p}$ is a unit vector (in the Euclidean or 2-norm) then the direction of **steepest descent** is the direction which minimises $(\nabla f(\underline{x}_k))^T \underline{p}$. Since by the Cauchy Schwarz inequality we have

$$-\|\nabla f(\underline{x}_k)\| \le (\nabla f(\underline{x}_k))^T \underline{p} \le \|\nabla f(\underline{x}_k)\| \tag{10.10}$$

we see that the steepest descent direction is

$$\underline{p} = -\frac{1}{\|\nabla f\|}\nabla f, \quad \text{i.e. parallel to } -\nabla f. \tag{10.11}$$

### 10.2.2 Other descent directions

If $M$ is any symmetric positive definite matrix then

$$\underline{p} = -M\nabla f \tag{10.12}$$

is a descent direction since

$$(\nabla f)^T \underline{p} = -(\nabla f)^T M \nabla f < 0 \tag{10.13}$$

provided $\nabla f \neq \underline{0}$. $M = H_f^{-1}$ corresponds to Newton's direction and $M = I$, the identity matrix, gives the steepest descent direction.

## 10.3 Descent directions in the nonlinear system context

A vector $\underline{x}^*$ satisfying $\underline{f}(\underline{x}^*) = \underline{0}$ is also a vector which minimises

$$g(\underline{x}) = \frac{1}{2}\|\underline{f}(\underline{x})\|_2^2 = \frac{1}{2}\underline{f}(\underline{x})^T \underline{f}(\underline{x}) . \tag{10.14}$$

Now

$$\begin{aligned}
\frac{\partial g}{\partial x_i} &= \frac{1}{2}\left(\left(\frac{\partial \underline{f}}{\partial x_i}\right)^T \underline{f} + \underline{f}^T\left(\frac{\partial \underline{f}}{\partial x_i}\right)\right) && (10.15)\\
&= \left(\frac{\partial \underline{f}}{\partial x_i}\right)^T \underline{f} && (10.16)\\
&= (i\text{th column of } J_f)^T \underline{f} && (10.17)\\
&= (i\text{th row of } J_f^T)\underline{f} && (10.18)
\end{aligned}$$

thus as a vector identity

$$\nabla g = J_f^T \underline{f} \; . \tag{10.19}$$

The steepest direction is thus

$$\underline{p} = -\nabla g = -J_f^T \underline{f} \; . \tag{10.20}$$

The standard Newton direction is

$$\underline{p} = -J_f^{-1} \underline{f} \;\; = \;\; -\left(J_f^{-1} J_f^{-T}\right) J_f^T \underline{f} \tag{10.21}$$

$$= \;\; -\left(J_f^T J_f\right)^{-1} J_f^T \underline{f} \; . \tag{10.22}$$

Provided $J_f$ is nonsingular (as is required in Newton's method) the matrix $J_f^T J_f$ is symmetric and positive definite and thus $\underline{p}$ is a descent direction from the previous result. More directly this can be shown by considering

$$(\nabla g)^T \underline{p} = -\underline{f}^T J_f J_f^{-1} \underline{f} = -\underline{f}^T \underline{f} = -\|\underline{f}\|_2^2 < 0 \tag{10.23}$$

at a point $\underline{x}$ for which $\underline{f}(\underline{x}) \neq \underline{0}$.

As a final note about Newton's method for solving $\underline{f}(\underline{x}) = \underline{0}$ and the problem of minimising $g(\underline{x})$ we have that Newton's method was derived by taking the linear model

$$\underline{M}_k(\underline{x}) = \underline{f}(\underline{x}_k) + J_f(\underline{x}_k)(\underline{x} - \underline{x}_k) \tag{10.24}$$

with the point $\underline{x}_{k+1}$ satisfying $\underline{M}_k(\underline{x}_{k+1}) = \underline{0}$. This point also minimises the quadratic

$$\frac{1}{2}\underline{M}_k(\underline{x})^T \underline{M}_k(\underline{x}) = \frac{1}{2}\left(\underline{f}^T \underline{f} + 2(\underline{x} - \underline{x}_k)^T J_f^T \underline{f} + (\underline{x} - \underline{x}_k)^T J_f^T J_f (\underline{x} - \underline{x}_k)\right), \tag{10.25}$$

where in the above $\underline{f} = \underline{f}(\underline{x}_k)$, $J_f = J_f(\underline{x}_k)$. You should note that this quadratic model is not the same as the quadratic model of $g$ obtained using the gradient and the Hessian of $g$. That quadratic model is

$$\frac{1}{2}\left(\underline{f}^T \underline{f} + 2(\underline{x} - \underline{x}_k)^T J_f^T \underline{f} + (\underline{x} - \underline{x}_k)^T H_g (\underline{x} - \underline{x}_k)\right) \tag{10.26}$$

and the Hessian $H_g$ will involve the second partial derivatives of the component functions $f_i$.

To complete the discussion on descent directions for the nonlinear system problems we note that the choice of minimising function

$$g(\underline{x}) = \frac{1}{2}\underline{f}(\underline{x})^T \underline{f}(\underline{x}) \tag{10.27}$$

is not unique. We could have taken

$$g(\underline{x}) = \frac{1}{2}\underline{f}(\underline{x})^T M \underline{f}(\underline{x}) \tag{10.28}$$

for any symmetric positive definite matrix $M$. With this choice for $g$ we have

$$\frac{\partial g}{\partial x_j} = \frac{1}{2}\left(\frac{\partial \underline{f}}{\partial x_j}^T M \underline{f} + \underline{f}^T M \frac{\partial \underline{f}}{\partial x_j}\right) = \frac{\partial \underline{f}}{\partial x_j}^T M \underline{f} \tag{10.29}$$

thus

$$\nabla g = J_f^T M \underline{f} \; . \tag{10.30}$$

The Newton direction $\underline{p} = -J_f^{-1} \underline{f}$ is a descent direction for this $\underline{g}$ since

$$(\nabla g)^T \underline{p} = -\underline{f}^T M J_f J_f^{-1} \underline{f} = -\underline{f}^T M \underline{f} < 0 \; . \tag{10.31}$$

## 10.4 The damped Newton method for systems

In section 10.3 we showed that the Newton direction

$$\underline{p}_k = -J_f^{-1}(\underline{x}_k)\underline{f}(\underline{x}_k) \tag{10.32}$$

is a descent direction of $g = \frac{1}{2}\|\underline{f}\|_2^2$. Based on this observation an outline of the damped Newton method is as follows.

$\underline{x}_0 =$ starting vector

For $k = 0, 1, 2, \cdots$

Compute $\underline{p}_k = -J_f^{-1}(\underline{x}_k)\underline{f}(\underline{x}_k)$

Determine a step $\alpha_k \in (0, 1]$ such that

$$\|\underline{f}(\underline{x}_k + \alpha_k \underline{p}_k)\|_2 < \|\underline{f}(\underline{x}_k)\|_2$$

Set $\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{p}_k$.

For the step parameter $\alpha_k$ you would in a typical implementation first try $\alpha_k = 1$ and use this if it is acceptable. If not then you would successively try $\alpha_k = 1/2,\ 1/4,\ 1/8, \cdots$ until the descent test is satisfied. It is in this sense that the method is termed 'damped'. The possible use of the damping factor ensures that $\|\underline{f}(\underline{x}_k)\|_2$ decreases with $k$.

## 10.5 Remarks about scaling

If $A$ and $B$ are nonsingular matrices and

$$\underline{g}(\underline{y}) = A\underline{f}(B\underline{y}) \tag{10.33}$$

then we could consider using Newton's method to solve $\underline{g}(\underline{y}) = \underline{0}$ instead of using Newton's method to solve $\underline{f}(\underline{x}) = \underline{0}$. However, as we will now show, the Newton iterates $\underline{y}_0$, $\underline{y}_1$, $\underline{y}_2, \cdots$ that are obtained when the method is applied to $\underline{g}$ correspond to the iterates $\underline{x}_0$, $\underline{x}_1$, $\underline{x}_2, \cdots$ that are obtained when the method is applied to $\underline{f}$ provided

$$\underline{x}_0 = B\underline{y}_0\ . \tag{10.34}$$

For the Jacobian matrix $J_g$ we have the following:

$$\frac{\partial \underline{g}}{\partial y_j} = A\frac{\partial \underline{f}(B\underline{y})}{\partial y_j} \tag{10.35}$$

and

$$\frac{\partial f_i}{\partial y_j}(B\underline{y}) = \sum_{k=1}^{n} \frac{\partial f_i(\underline{x})}{\partial x_k} b_{kj} = (J_f(\underline{x})B)_{ij}, \quad \underline{x} = B\underline{y}\ . \tag{10.36}$$

Thus

$$J_g = AJ_fB, \quad J_g^{-1} = B^{-1}J_f^{-1}A^{-1}, \quad \text{and} \quad J_g^{-1}\underline{g} = B^{-1}J_f^{-1}\underline{f}\ . \tag{10.37}$$

Hence

$$\begin{aligned} \underline{y}_1 &= \underline{y}_0 - J_g^{-1}(\underline{y}_0)\underline{g}(\underline{y}_0) = B^{-1}\left(\underline{x}_0 - J_f^{-1}(\underline{x}_0)\underline{f}(\underline{x}_0)\right) && (10.38)\\ &= B^{-1}\underline{x}_1 \ . && (10.39)\end{aligned}$$

That is $\underline{x}_0 = B\underline{y}_0$ implies that $\underline{x}_1 = B\underline{y}_1$ and by induction $\underline{x}_k = B\underline{y}_k$. This shows that Newton's method is invariant to linear changes to the independent and to the dependent variables

The practical implementation of Newton's method does however depend on the scale of both the independent and dependent variables in the following ways. Stopping criterion of the form

$$\text{Is } \|\underline{f}(\underline{x}_k)\| < \epsilon \quad \text{or is} \quad \|\underline{x}_k - \underline{x}_{k-1}\| < \text{tol ?} \tag{10.40}$$

depend on the scale of both $\underline{f}$ and $\underline{x}_k$. Also the damping requirement that

$$\|\underline{f}(\underline{x}_k + \alpha_k \underline{p}_k)\|_2 < \|\underline{f}(\underline{x}_k)\|_2 \tag{10.41}$$

is different from the damping requirement that

$$\|\underline{g}(\underline{y}_k + \alpha_k B^{-1}\underline{p}_k)\|_2 < \|\underline{g}(\underline{y}_k)\|_2 \ . \tag{10.42}$$

In packages Newton's method is often implemented using a diagonal matrix to re-scale the independent variables and another diagonal matrix to re-scale the dependent variables so that all quantities have a similar order of magnitude.

## 10.6 Remarks about the line search

The previous sections concerned descent directions. For the problem of minimising $f(\underline{x})$ we wish, when we are at a point $\underline{x}_k$, to always move in a descent direction $\underline{p}_k$. When we are near to the minimising point $\underline{x}^*$ and we are using Newton's method, we wish to be taking the full Newton step but when we are far from $\underline{x}^*$ we just need a step which yields an 'acceptable' next point $\underline{x}_{k+1}$. That is at iteration $k$ we do the following.

Calculate a descent direction $\underline{p}_k$.

Set $\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{p}_k$ for some $\alpha_k > 0$ that makes $\underline{x}_{k+1}$ an acceptable next iterate.

The term line search refers to the choice of $\alpha_k$. A line search is said to be **exact** if $\alpha_k$ minimises the one dimensional function

$$f(\underline{x}_k + \alpha \underline{p}_k) \tag{10.43}$$

with respect to $\alpha$. This involves solving

$$\frac{d}{d\alpha} f(\underline{x}_k + \alpha \underline{p}_k) = \underline{p}_k^T \nabla f(\underline{x}_k + \alpha \underline{p}_k) = 0 \ . \tag{10.44}$$

If $\alpha = \alpha_k$ solves this equation and $\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{p}_k$ then observe that

$$\underline{p}_k^T \nabla f(\underline{x}_{k+1}) = 0 \ . \tag{10.45}$$

When $f$ is a quadratic the exact line search can be done easily (the complete minimisation only involves solving one linear system) but it is generally considered wasteful to try to do one-dimensional minimization accurately when $\underline{x}_k$ is far from $\underline{x}^*$. All that is needed is an acceptable value which will generally mean close to the exact line search value. The obvious question is what is acceptable.

An obvious requirement of the line search is that it gives a point $\underline{x}_{k+1}$ such that

$$f(\underline{x}_{k+1}) < f(\underline{x}_k) \ . \tag{10.46}$$

If $f$ is bounded below then the sequence $(f(\underline{x}_k))$ is a monotonically decreasing sequence which converges to a limit. Unfortunately this may not correspond to a minimum of $f$. For example if $f(x) = x^2$ (which has a minimum at $x = 0$) then the descent condition is satisfied when

$$x_k = 1 + \frac{1}{k} \downarrow 1 \quad \text{and when} \quad x_k = (-1)^k \left(1 + \frac{1}{k}\right) \ . \tag{10.47}$$

In both cases $f(x_k) \downarrow 1$ which is clearly not the minimum of $f$. In the first case the points are moving in the correct direction but by too small an amount. In the second case the decrease in $f$ is also too little and in addition the sequence of points $(x_k)$ does not converge. In the second case we have that $x_{2k+1} - x_{2k}$ and $x_{2k} - x_{2k-1}$ both tend to a direction which is orthogonal to the steepest descent direction.

Thus to summarize, for the minimization algorithm to succeed we need both an acceptable step length $\alpha_k$ and an acceptable direction $\underline{p}_k$. There is not time in this module to go into details about what is 'acceptable' and then to prove convergence when the 'acceptable' conditions are satisfied. It suffices to say however that we require the direction $\underline{p}_k$ to be uniformly bounded away from being orthogonal to $\nabla f(\underline{x}_k)$ and we require $\alpha_k$ to be 'somewhere near' the exact line search value.

The 'somewhere near' condition is satisfied, for example, by a condition of the form

$$|\underline{p}_k^T \nabla f(\underline{x}_k + \alpha_k \underline{p}_k)| \le \beta |\underline{p}_k^T \nabla f(\underline{x}_k)| \tag{10.48}$$

for some fixed $\beta \in [0, 1)$. Observe that $\beta = 0$ corresponds to an exact line search.

Regarding the direction requirement, we note that the Newton directions

$$\underline{p}_k = -H_f(\underline{x}_k)^{-1} \nabla f(\underline{x}_k), \quad k = 1, 2, \cdots \tag{10.49}$$

are uniformly bounded away from being orthogonal to the gradient direction $-\nabla f(\underline{x}_k)$ provided the matrices $(H_f(\underline{x}_k))$ are uniformly bounded away from a singular matrix. We have

$$\nabla f(\underline{x}_k)^T \underline{p}_k = -\nabla f(\underline{x}_k)^T H_f(\underline{x}_k)^{-1} \nabla f(\underline{x}_k) \ . \tag{10.50}$$

If $0 < \mu_n \le \cdots \le \mu_1$ are the eigenvalues of the positive definite matrix $H_f(\underline{x}_k)$, then

$$0 < \frac{1}{\mu_1} \le \cdots \le \frac{1}{\mu_n} \tag{10.51}$$

are the eigenvalues of $H_f(\underline{x}_k)^{-1}$ and the spectral condition number of $H_f(\underline{x}_k)$ is

$$\kappa(H_f(\underline{x}_k)) = \|H_f(\underline{x}_k)\|_2 \|(H_f(\underline{x}_k))^{-1}\|_2 = \frac{\mu_1}{\mu_n} \ . \tag{10.52}$$

Now from (10.49) we have

$$\|\underline{p}_k\|_2 \le \|H_f(\underline{x}_k)^{-1}\|_2 \|\nabla f(\underline{x}_k)\|_2 \le \frac{1}{\mu_n} \|\nabla f(\underline{x}_k)\|_2 \tag{10.53}$$

and

$$H_f(\underline{x}_k) \underline{p}_k = -\underline{f}(\underline{x}_k) \quad \text{gives} \quad \|\underline{f}(\underline{x}_k)\| \le \|H_f(\underline{x}_k)\| \|\underline{p}_k\| = \mu_1 \|\underline{p}_k\| \tag{10.54}$$

Thus

$$\frac{1}{\mu_1}\|\nabla f(\underline{x}_k)\|_2 \le \|\underline{p}_k\| \tag{10.55}$$

and from (10.50) and using Rayleigh quotient bounds we have

$$\frac{1}{\mu_1}\|\nabla f(\underline{x}_k)\|_2^2 \le -\nabla f(\underline{x}_k)^T \underline{p}_k = \nabla f(\underline{x}_k)^T H_f(\underline{x}_k)^{-1}\nabla f(\underline{x}_k) \le \frac{1}{\mu_n}\|\nabla f(\underline{x}_k)\|_2^2 \,. \tag{10.56}$$

Hence by taking the 'smallest' numerator and 'largest denominator' we have

$$\frac{-\underline{p}_k^T \nabla f(\underline{x}_k)}{\|\nabla f(\underline{x}_k)\|_2 \|\underline{p}_k\|_2} \ge \frac{\mu_n}{\mu_1} = \frac{1}{\kappa(H_f(\underline{x}_k))} \,. \tag{10.57}$$

$\kappa(H_f(\underline{x}_k))$ being bounded as $k \to \infty$ thus ensures that $\underline{p}_k$ is bounded away from being orthogonal to $\nabla f(\underline{x}_k)$.

# 11 The conjugate gradient method (CGM)

## 11.1 Introductory remarks

Earlier in this module direct methods of the Gauss elimination type were considered for solving systems of linear equations $A\underline{x} = \underline{b}$. For a full matrix $A \in \mathbb{R}^{n\times n}$ this involves $\mathcal{O}(n^2)$ storage and $\mathcal{O}(n^3)$ operations to obtain $\underline{x}$. It was also noted that in the case of tridiagonal systems (i.e. with a tridiagonal matrix $A$) that the storage and operation count could be reduced to $\mathcal{O}(n)$. In numerical analysis, tridiagonal matrices typically arise in methods for solving one-dimensional problems, e.g. in the finite difference method for a two-point boundary value problem for a second order ordinary differential equation. In such cases $n$ itself may not be too large for "sufficient accuracy" and the solution procedure itself is very efficient. For matrices $A$ which arise in methods for solving problems in higher dimensions the size $n$ of the system is generally much larger and the matrices obtained, although sparse, generally have a large band width. If $q \le n$ is the bandwidth then a direct solver such as Gauss elimination procedure needs $\mathcal{O}(nq)$ storage and $\mathcal{O}(nq^2)$ operations. When finite difference or finite element methods are used to approximately solve a second order partial differential equation (pde) in two space dimensions we typically have that $q$ will be no smaller than $\sqrt{n}$ giving $\mathcal{O}(n^{3/2})$ storage and $\mathcal{O}(n^2)$ operations. Although this is much less than $\mathcal{O}(n^2)$ storage and $\mathcal{O}(n^3)$ operations respectively it may still be excessive if $n$ is very large. Note that in the pde example just given we typically only have $\mathcal{O}(n)$ non-zero entries (perhaps about 5 or 6 non-zero entries on each row). To cope with the storage problem when $n$ is large we need methods which only need to use the non-zero entries of the matrix which is why iterative methods are used. For efficiency we would also like to have an iterative method which can achieve a given accuracy with as little as possible computation. A popular method in the case of systems with symmetric positive definite matrices which goes some way to meeting these goals is the CGM. It is efficient in storage as the only reference to the matrix $A$ is in the computation of matrix vector products $A\underline{p}_i$ at various stages for various vectors $\underline{p}_i$. Efficiency in the sense of minimizing the total amount of computation is concerned with the amount of work per iteration and the number of iterations required to reach a desired accuracy. It turns out that this is related to something known as preconditioning which is concerned with possibly modifying the given problem to create an equivalent problem which has faster convergence. The choice of the best preconditioner is problem dependent. We will only briefly comment on preconditioning in this module.

To summarize, the CGM is concerned with constructing a sequence of vectors $\underline{x}_0$, $\underline{x}_1$, $\underline{x}_2, \cdots$ which converge to the solution of

$$A\underline{x} = \underline{b} \tag{11.1}$$

when $A$ is a symmetric positive definite matrix, i.e. a matrix for which

$$A^T = A \quad \text{and} \quad \underline{x}^T A\underline{x} > 0 \quad \text{for all } \underline{x} \ne \underline{0}. \tag{11.2}$$

It is typically used when the size $n$ of the system is large and the matrix $A$ is sparse. Before the method is described we briefly review a few of the properties of positive definite symmetric matrices.

(i) Since $A$ is real and symmetric, the eigenvalues and eigenvectors are real. Further, there is an orthonormal basis $\underline{v}_1, \cdots, \underline{v}_n$ of the eigenvectors. If

$$A\underline{v}_i = \lambda_i \underline{v}_i \tag{11.3}$$

then by pre-multiplying by $\underline{v}_i^T$ we obtain

$$\lambda_i = \frac{\underline{v}_i^T A\underline{v}_i}{\underline{v}_i^T \underline{v}_i} > 0 \tag{11.4}$$

since the numerator and denominator are both positive. The eigenvalues of a positive definite matrix are hence all positive. We will need to refer to the eigenvalues when we consider the convergence of the iteration.

(ii) A symmetric positive definite matrix $A$ has a Cholesky factorization

$$A = L\,L^T \tag{11.5}$$

where $L$ is a lower triangular matrix with positive diagonal entries. Note that by using the property that $L$ is lower triangular,

$$a_{ij} = (A)_{ij} = (L\,L^T)_{ij} = \sum_{r=1}^{n} l_{ir} l_{jr} = \sum_{r=1}^{\min\{i,j\}} l_{ir} l_{jr}\ . \tag{11.6}$$

Hence if $A_k$ and $L_k$ are the $k \times k$ principal submatrices of $A$ and $L$ respectively then for $1 \le i, j \le k$,

$$a_{ij} = (A_k)_{ij} = (A)_{ij} = \sum_{r=1}^{\min\{i,j\}} l_{ir} l_{jr} = (L_k\, L_k^T)_{ij}, \tag{11.7}$$

i.e. for the principal submatrices we have

$$A_k = L_k\, L_k^T, \quad k = 1, 2, \cdots, n. \tag{11.8}$$

We do not need these specifically to describe the CGM but there is a preconditioner for the CGM which involves using what is known as an incomplete Cholesky factorization which, loosely speaking, involves approximately constructing this factorization.

## 11.2 The CGM: the linear equations, minimization, the error and the residual

The CGM is concerned with minimising the quadratic $f : \mathbb{R}^n \to \mathbb{R}$ defined by

$$f(\underline{x}) = \frac{1}{2}\underline{x}^T A \underline{x} - \underline{x}^T \underline{b} \tag{11.9}$$

which has gradient vector $\nabla f$ and Hessian matrix $H_f$ given respectively by

$$\nabla f = A\underline{x} - \underline{b} \quad \text{and} \quad H_f = A\ . \tag{11.10}$$

If $\underline{x}^*$ is the vector satisfying $A\underline{x}^* = \underline{b}$, i.e. $\underline{x}^* = A^{-1}\underline{b}$ then $\underline{x}^*$ uniquely minimises $f$. We can actually verify the minimization property directly by considering

$$\begin{aligned} f(\underline{x}^* + \underline{d}) - f(\underline{x}^*) &= \frac{1}{2}\left((\underline{x}^* + \underline{d})^T A(\underline{x}^* + \underline{d}) - \underline{x}^{*T} A\underline{x}^*\right) - \left(\underline{b}^T(\underline{x}^* + \underline{d}) - \underline{b}^T\underline{x}^*\right) \\ &= \frac{1}{2}\left(2\underline{d}^T A\underline{x}^* + \underline{d}^T A\underline{d}\right) - \underline{d}^T\underline{b} = \frac{1}{2}\underline{d}^T A\underline{d} > 0 \end{aligned}$$

for all $\underline{d} \ne \underline{0}$. In the CGM a sequence of vectors $\underline{x}_0, \underline{x}_1, \underline{x}_2, \cdots$ are generated which converge to $\underline{x}^*$ in $n$ or less steps (assuming exact arithmetic) with $f(\underline{x}_0) > f(\underline{x}_1) > f(\underline{x}_2) > \cdots$ until convergence. We define the **error** in the approximation $\underline{x}_k$ as

$$\underline{e}_k = \underline{x}^* - \underline{x}_k \tag{11.11}$$

and we define the **residual** $\underline{r}_k = \underline{r}(\underline{x}_k)$ as

$$\underline{r}_k = \underline{b} - A\underline{x}_k \tag{11.12}$$

and we note immediately that

$$\underline{r}_k = A\underline{e}_k \quad \text{and} \quad \underline{r}(\underline{x}) = -\nabla f(\underline{x}) \ . \tag{11.13}$$

The residual $\underline{r}_k$ is something that we can compute and it measures how closely the vector $\underline{x}_k$ satisfies the linear equations. The error $\underline{e}_k$ and the difference $f(\underline{x}_k) - f(\underline{x}^*) = \underline{e}_k^T A \underline{e}_k/2$ both cannot be computed in general as we do not in general already know the exact solution. Also, it needs to be noted that the sequence $(\underline{e}_k^T A \underline{e}_k)$ is guaranteed to decrease but that the sequences $(\underline{e}_k^T \underline{e}_k)$ and $(\underline{r}_k^T \underline{r}_k)$ need not be strictly decreasing although they do both converge to 0.

## 11.3 Outline of an algorithm with general descent directions

We now consider an outline of a general descent algorithm for minimising $f$. Suppose we start with $\underline{x}_0$ and after $k-1$ iterations we have computed $\underline{x}_{k-1}$ with $\underline{r}_{k-1}$ denoting the residual at this stage. Given a direction $\underline{p}_k$ in which $f(\underline{x})$ decreases at $\underline{x}_{k-1}$, i.e.

$$\underline{p}_k^T \nabla f(\underline{x}_{k-1}) = -\underline{p}_k^T \underline{r}_{k-1} < 0 \tag{11.14}$$

we minimise $f(\underline{x}_{k-1} + t\underline{p}_k)$ with respect to $t$ to get the next point $\underline{x}_k$. By the chain rule

$$\frac{d}{dt} f(\underline{x}_{k-1} + t\underline{p}_k) = \underline{p}_k^T \nabla f(\underline{x}_{k-1} + t\underline{p}_k) = \underline{p}_k^T (A(\underline{x}_{k-1} + t\underline{p}_k) - \underline{b}) = t\underline{p}_k^T A \underline{p}_k - \underline{p}_k^T \underline{r}_{k-1} \tag{11.15}$$

and thus at the minimum $t = \alpha_k$ where

$$\alpha_k = \frac{\underline{p}_k^T \underline{r}_{k-1}}{\underline{p}_k^T A \underline{p}_k} \quad \text{and we define} \quad \underline{x}_k = \underline{x}_{k-1} + \alpha_k \underline{p}_k \ . \tag{11.16}$$

As a consequence, for the residual $\underline{r}_k$ at $\underline{x}_k$ we have

$$\underline{r}_k = \underline{b} - A\underline{x}_k = \underline{b} - A(\underline{x}_{k-1} + \alpha_k \underline{p}_k) = \underline{r}_{k-1} - \alpha_k A \underline{p}_k \ . \tag{11.17}$$

If we start with $\underline{x}_0$ and the first direction is the steepest descent direction $\underline{p}_1 = \underline{r}_0$ then an outline of an algorithm is as follows.

For $k = 1, 2, \cdots$

$$\alpha_k = \frac{\underline{p}_k^T \underline{r}_{k-1}}{\underline{p}_k^T A \underline{p}_k}$$
$$\underline{x}_k = \underline{x}_{k-1} + \alpha_k \underline{p}_k,$$
$$\underline{r}_k = \underline{r}_{k-1} - \alpha_k A \underline{p}_k \ .$$

Select the next descent direction $\underline{p}_{k+1}$.

If we take $\underline{p}_{k+1} = \underline{r}_k$ then we always search in the steepest descent direction and we get the method of steepest descent. This can be a very poor method with very slow convergence when the condition number of $A$ is very large. The CGM chooses different directions which leads to faster convergence. We will consider these directions in subsequent subsections.

It is notationally convenient in what follows to assume that $\underline{x}_0 = \underline{0}$ giving $\underline{p}_1 = \underline{r}_0 = \underline{b}$. In this way we can write

$$\underline{x}_k = \alpha_1 \underline{p}_1 + \cdots + \alpha_k \underline{p}_k \in \text{span}\{\underline{p}_1, \cdots, \underline{p}_k\} \ . \tag{11.18}$$

## 11.4 Using $A$-conjugate directions

In the CGM the non-zero vectors $\underline{p}_1, \underline{p}_2, \cdots$ that are used have a $A$-conjugacy property that

$$\underline{p}_i^T A \underline{p}_j = 0 \quad \text{for all } i \neq j. \tag{11.19}$$

We will postpone for the moment how we construct $\underline{p}_{k+1}$ to be $A$-conjugate to all the previous vectors $\underline{p}_1, \cdots, \underline{p}_k$ and consider instead some consequences of having this property.

The first point to note is that such directions $\underline{p}_1, \underline{p}_2, \cdots$ are linearly independent. This follows immediately by considering a general linear combination.

$$c_1 \underline{p}_1 + c_2 \underline{p}_2 + \cdots + c_n \underline{p}_n = \underline{0}\ . \tag{11.20}$$

Pre-multiplying by $\underline{p}_i^T A$ gives, by the $A$-conjugacy, that

$$\underline{p}_i^T A (c_1 \underline{p}_1 + c_2 \underline{p}_2 + \cdots + c_n \underline{p}_n) = c_i \underline{p}_i^T A \underline{p}_i = 0 \tag{11.21}$$

and since $\underline{p}_i^T A \underline{p}_i > 0$, because $A$ is positive definite, $c_i = 0$. As this is true for $i = 1, 2, \cdots, n$ we have shown that the vectors are linearly independent. $n$ linearly independent vectors in $\mathbb{R}^n$ is a basis for $\mathbb{R}^n$ and hence the solution $\underline{x}^*$ can be expressed in terms of them, i.e.

$$\underline{x}^* = \sum_{j=1}^{n} \tilde{\alpha}_j \underline{p}_j\ . \tag{11.22}$$

Pre-multiplying this relation by $\underline{p}_k^T A$ gives

$$\underline{p}_k^T A \underline{x}^* = \tilde{\alpha}_k \underline{p}_k^T A \underline{p}_k \tag{11.23}$$

and hence

$$\tilde{\alpha}_k = \frac{\underline{p}_k^T A \underline{x}^*}{\underline{p}_k^T A \underline{p}_k} = \frac{\underline{p}_k^T \underline{b}}{\underline{p}_k^T A \underline{p}_k}\ . \tag{11.24}$$

As $\underline{p}_k$ is $A$-conjugate to $\mathrm{span}\{\underline{p}_1, \cdots, \underline{p}_{k-1}\}$ it follows that since $\underline{x}_{k-1} \in \mathrm{span}\{\underline{p}_1, \cdots, \underline{p}_{k-1}\}$ we have

$$\underline{p}_k^T \underline{b} = \underline{p}_k^T (\underline{b} - A\underline{x}_{k-1}) = \underline{p}_k^T \underline{r}_{k-1} \tag{11.25}$$

and

$$\tilde{\alpha}_k = \frac{\underline{p}_k^T \underline{r}_{k-1}}{\underline{p}_k^T A \underline{p}_k} = \alpha_k\ .$$

That is with $A$-conjugate directions the exact solution is given by

$$\underline{x}^* = \sum_{j=1}^{n} \alpha_j \underline{p}_j \tag{11.26}$$

and the $k$th CGM vector $\underline{x}_k$ is given by

$$\underline{x}_k = \sum_{j=1}^{k} \alpha_j \underline{p}_j \quad \text{for } k = 1, 2, \cdots\ . \tag{11.27}$$

This explains why the method converges in $n$ or less steps.

Other consequences are that

$$\underline{e}_k = \underline{x}^* - \underline{x}_k = \sum_{j=k+1}^{n} \alpha_j \underline{p}_j \quad \text{and} \quad \underline{r}_k = A\underline{e}_k = \sum_{j=k+1}^{n} \alpha_j A\underline{p}_j \tag{11.28}$$

which imply that $\underline{e}_k$ is $A$-conjugate to span$\{\underline{p}_1, \cdots, \underline{p}_k\}$ and $\underline{r}_k$ is orthogonal to span$\{\underline{p}_1, \cdots, \underline{p}_k\}$. It also follows that if we consider any other vector

$$\underline{y} = \sum_{i=1}^{k} c_i \underline{p}_i \in \text{span}\{\underline{p}_1, \cdots, \underline{p}_k\} \tag{11.29}$$

and we define

$$E(\underline{y}) := (\underline{y} - \underline{x}^*)^T A (\underline{y} - \underline{x}^*) \tag{11.30}$$

then $\underline{x}_k$ minimises $E(\underline{y})$ over the space span$\{\underline{p}_1, \cdots, \underline{p}_k\}$. This is verified by writing

$$\underline{x}^* - \underline{y} = \sum_{i=1}^{k} (\alpha_i - c_i)\underline{p}_i + \sum_{i=k+1}^{n} \alpha_i \underline{p}_i \tag{11.31}$$

and noting that the $A$-conjugate property implies that

$$E(\underline{y}) = \sum_{i=1}^{k} (\alpha_i - c_i)^2 \underline{p}_i^T A \underline{p}_i + \sum_{i=k+1}^{n} \alpha_i^2 \underline{p}_i^T A \underline{p}_i \geq \sum_{i=k+1}^{n} \alpha_i^2 \underline{p}_i^T A \underline{p}_i = E(\underline{x}_k) \ . \tag{11.32}$$

Hence $\underline{x}_k$ is the best approximation to $\underline{x}^*$, in this sense, from all the directions $\underline{p}_1, \cdots, \underline{p}_k$ so far considered. Another way of stating this is that

$$f(\underline{y}) = \frac{1}{2}\underline{y}^T A \underline{y} - \underline{y}^T \underline{b} = f(\underline{x}^*) + E(\underline{y})/2 \tag{11.33}$$

has not only been minimised in one direction at each step but it has also been minimised in the space spanned by all the directions used up to that point.

## 11.5 Construction of the $A$-conjugate direction $\underline{p}_{k+1}$

We now consider how to construct the $A$-conjugate directions in an economical way. In the case of the construction of $\underline{p}_{k+1}$ it involves finding a non-zero vector which satisfies

$$\underline{p}_j^T A \underline{p}_{k+1} = 0, \quad j = 1, 2, \cdots, k \ . \tag{11.34}$$

As span$\{\underline{p}_1, \cdots, \underline{p}_k\}$ is a $k$-dimensional subspace of $\mathbb{R}^n$, there is a $n-k$ dimensional subspace of $\mathbb{R}^n$ for which we can search for such vectors. If $\underline{r}_k = \underline{0}$ then $\underline{x}_k = \underline{x}^*$ is the exact solution and we do not need to continue with the iteration. Otherwise, as we have already shown, $\underline{r}_k$ is orthogonal to span$\{\underline{p}_1, \cdots, \underline{p}_k\}$ and hence $\underline{r}_k$ is in the $n-k$ dimensional subspace to search. In the CGM we take

$$\underline{p}_{k+1} \in \text{span}\{\underline{p}_1, \cdots, \underline{p}_k, \underline{r}_k\} \ . \tag{11.35}$$

Specifically we take

$$\underline{p}_{k+1} = \underline{r}_k - \sum_{j=1}^{k} \gamma_{k+1,j}\, \underline{p}_j \tag{11.36}$$

where $\gamma_{k+1,1}, \cdots, \gamma_{k+1,k}$ are to be determined to ensure the $A$-conjugate property. To do this we pre-multiply by $\underline{p}_i^T A$, $i \le k$, to give

$$0 = \underline{p}_i^T A \underline{p}_{k+1} = \underline{p}_i^T A \underline{r}_k - \gamma_{k+1,i}\, \underline{p}_i^T A \underline{p}_i \tag{11.37}$$

giving

$$\gamma_{k+1,i} = \frac{\underline{p}_i^T A \underline{r}_k}{\underline{p}_i^T A \underline{p}_i}, \quad i = 1, 2, \cdots, k\ . \tag{11.38}$$

This gives the vector $\underline{p}_{k+1}$ that is used but it appears at present not to be very efficient due to the computation of the $k$ numbers $\gamma_{k+1,i}$, $i = 1, 2, \cdots, k$. In fact, as we show in a moment, $\gamma_{k+1,i} = 0$ for $i = 1, 2, \cdots, k-1$ so that we only actually have to compute $\beta_k := -\gamma_{k+1,k}$. We establish this property after considering another way of describing the space $\mathrm{span}\{\underline{p}_1, \cdots, \underline{p}_k\}$ in the next subsection.

## 11.6 The Krylov subspaces

When we start the CGM with $\underline{x}_0 = 0$ we get $\underline{p}_1 = \underline{r}_0 = \underline{b}$. Then by repeated use of the relations

$$\begin{aligned} \underline{x}_k &= \underline{x}_{k-1} + \alpha_k \underline{p}_k && (11.39)\\ \underline{r}_k &= \underline{r}_{k-1} - \alpha_k A \underline{p}_k && (11.40)\\ \underline{p}_{k+1} &\in \mathrm{span}\{\underline{p}_1, \cdots, \underline{p}_k, \underline{r}_k\} && (11.41) \end{aligned}$$

for $k = 1, 2, \cdots$ we obtain

$$\begin{aligned} \underline{r}_0, \underline{p}_1, \underline{x}_1 &\in \mathrm{span}\{\underline{b}\}\\ \underline{r}_1, \underline{p}_2, \underline{x}_2 &\in \mathrm{span}\{\underline{b}, A\underline{b}\}\\ \underline{r}_2, \underline{p}_3, \underline{x}_3 &\in \mathrm{span}\{\underline{b}, A\underline{b}, A^2\underline{b}\}\\ \cdots & \quad \cdots\\ \underline{r}_k, \underline{p}_{k+1}, \underline{x}_{k+1} &\in \mathrm{span}\{\underline{b}, A\underline{b}, A^2, \cdots, A^k\underline{b}\}\ . \end{aligned}$$

As the search directions are linearly independent and we have

$$\underline{p}_1, \cdots, \underline{p}_{k+1} \in \mathrm{span}\{\underline{b}, A\underline{b}, \cdots, A^k\underline{b}\}$$

it follows that we must have

$$\mathrm{span}\{\underline{p}_1, \cdots, \underline{p}_{k+1}\} = \mathrm{span}\{\underline{b}, A\underline{b}, \cdots, A^k\underline{b}\}\ . \tag{11.42}$$

### 11.6.1 The construction of $\underline{p}_{k+1}$ continued

We return to the consideration of the terms

$$\gamma_{k+1,i} = \frac{\underline{p}_i^T A \underline{r}_k}{\underline{p}_i^T A \underline{p}_i}, \quad i = 1, 2, \cdots, k\ . \tag{11.43}$$

Now

$$\begin{aligned} \underline{p}_i &\in \mathrm{span}\{\underline{b}, A\underline{b}, \cdots, A^{i-1}\underline{b}\}, && (11.44)\\ A\underline{p}_i &\in \mathrm{span}\{\underline{b}, A\underline{b}, \cdots, A^i\underline{b}\} = \mathrm{span}\{\underline{p}_1, \cdots, \underline{p}_{i+1}\}\ . && (11.45) \end{aligned}$$

We have already shown that $\underline{r}_k$ is orthogonal to span$\{\underline{p}_1, \cdots, \underline{p}_k\}$ and hence

$$\underline{p}_i^T A \underline{r}_k = 0 \quad \text{for } i+1 \le k, \text{ i.e. } i \le k-1 \tag{11.46}$$

and this shows that $\gamma_{k+1,i} = 0$ for $i = 1, 2, \cdots, k-1$. Hence with $\beta_k := -\gamma_{k+1,k}$ we have

$$\underline{p}_{k+1} = \underline{r}_k + \beta_k \underline{p}_k \quad \text{where } \beta_k := -\frac{\underline{p}_k^T A \underline{r}_k}{\underline{p}_k^T A \underline{p}_k} \ . \tag{11.47}$$

## 11.7 The basic CGM algorithm

We are now in a position to describe the basic CGM algorithm.

$\underline{x}_0 = 0$, $\underline{p}_1 = \underline{r}_0 = \underline{b}$.

For $k = 1, 2, \cdots$

$$\alpha_k = \frac{\underline{p}_k^T \underline{r}_{k-1}}{\underline{p}_k^T A \underline{p}_k}$$

$$\underline{x}_k = \underline{x}_{k-1} + \alpha_k \underline{p}_k$$

$$\underline{r}_k = \underline{r}_{k-1} - \alpha_k A \underline{p}_k$$

$$\beta_k = -\frac{\underline{p}_k^T A \underline{r}_k}{\underline{p}_k^T A \underline{p}_k}$$

$$\underline{p}_{k+1} = \underline{r}_k + \beta_k \underline{p}_k$$

In an implementation you only need to compute $\underline{z}_k = A\underline{p}_k$ once for each $k$ and also $\underline{p}_k^T A \underline{p}_k = \underline{z}_k^T \underline{p}_k$ only has to be done once.

There are some additional simplifications regarding the parameters $\alpha_k$ and $\beta_k$. Since

$$\underline{p}_k = \underline{r}_{k-1} + \beta_{k-1}\underline{p}_{k-1} \quad \text{and} \quad \underline{r}_{k-1} \text{ is orthogonal to span}\{\underline{p}_1, \cdots, \underline{p}_{k-1}\} \tag{11.48}$$

we have

$$\underline{p}_k^T \underline{r}_{k-1} = \underline{r}_{k-1}^T \underline{r}_{k-1} \quad \text{and} \quad \alpha_k = \frac{\underline{r}_{k-1}^T \underline{r}_{k-1}}{\underline{p}_k^T A \underline{p}_k} \ . \tag{11.49}$$

Similarly

$$\underline{r}_k = \underline{r}_{k-1} - \alpha_k A \underline{p}_k \quad \text{gives} \quad A\underline{p}_k = (\underline{r}_{k-1} - \underline{r}_k)/\alpha_k \quad \text{and} \quad \underline{r}_k^T A \underline{p}_k = -\underline{r}_k^T \underline{r}_k / \alpha_k \ . \tag{11.50}$$

We have just shown that

$$\underline{p}_k^T A \underline{p}_k = \underline{r}_{k-1}^T \underline{r}_{k-1} / \alpha_k$$

and thus

$$\beta_k = -\frac{\underline{p}_k^T A \underline{r}_k}{\underline{p}_k^T A \underline{p}_k} = \frac{\underline{r}_k^T \underline{r}_k}{\underline{r}_{k-1}^T \underline{r}_{k-1}} \ . \tag{11.51}$$

**Summary of the basic CGM algorithm**

With these relations for $\alpha_k$ and $\beta_k$ the basic CGM is as follows.

$\underline{x}_0 = 0$, $\underline{p}_1 = \underline{r}_0 = \underline{b}$.

For $k = 1, 2, \cdots$

$$\alpha_k = \frac{\underline{r}_{k-1}^T \underline{r}_{k-1}}{\underline{p}_k^T A \underline{p}_k}$$
$$\underline{x}_k = \underline{x}_{k-1} + \alpha_k \underline{p}_k$$
$$\underline{r}_k = \underline{r}_{k-1} - \alpha_k A \underline{p}_k$$
$$\beta_k = \frac{\underline{r}_k^T \underline{r}_k}{\underline{r}_{k-1}^T \underline{r}_{k-1}}$$
$$\underline{p}_{k+1} = \underline{r}_k + \beta_k \underline{p}_k$$

Note the simplicity of the method. The only reference to the matrix $A$ is in forming the product $A\underline{p}_k$ and to form the product $A\underline{p}_k$ we only need the non-zero entries of $A$.

## 11.8 The convergence of the iteration

The convergence analysis of the CGM is based on the result that

$$E(\underline{y}) := (\underline{y} - \underline{x}^*)^T A (\underline{y} - \underline{x}^*) = (\underline{x}^* - \underline{y})^T A (\underline{x}^* - \underline{y}) \tag{11.52}$$

is minimised by the vector $\underline{x}_k$ over the $k$-dimensional space

$$\operatorname{span}\{\underline{p}_1, \cdots, \underline{p}_k\} = \operatorname{span}\{\underline{b}, A\underline{b}, \cdots, A^{k-1}\underline{b}\} \ . \tag{11.53}$$

Since $\underline{b} = A\underline{x}^*$ we have

$$\operatorname{span}\{\underline{b}, A\underline{b}, \cdots, A^{k-1}\underline{b}\} = \operatorname{span}\{A\underline{x}^*, A^2\underline{x}^*, \cdots, A^k\underline{x}^*\} \ . \tag{11.54}$$

and thus any $\underline{y}$ in this space is of the form

$$\underline{y} = c_1 A\underline{x}^* + c_2 A^2 \underline{x}^* + \cdots + c_k A^k \underline{x}^* \tag{11.55}$$

giving

$$\underline{x}^* - \underline{y} = q_k(A)\underline{x}^* \tag{11.56}$$

where $q_k$ is the polynomial of degree less than or equal to $k$ satisfying $q_k(0) = 1$ given by

$$q_k(t) = 1 - (c_1 t + c_2 t^2 + \cdots + c_k t^k) \ . \tag{11.57}$$

Thus we need to minimise

$$E(\underline{y}) = \underline{x}^{*T} q_k(A)^2 A \underline{x}^* \tag{11.58}$$

over all polynomials of degree less than or equal to $k$ satisfying $q_k(0) = 1$.

The next step in the analysis is to express $\underline{x}^*$ in terms of the orthonormal eigenvectors $\underline{v}_1, \cdots, \underline{v}_n$ of $A$. Let $\lambda_1, \cdots, \lambda_n$ denote the corresponding eigenvalues which we assume are ordered so that $\lambda_n \le \lambda_{n-1} \le \cdots \le \lambda_1$. We have, because of the orthonormality, that

$$\underline{x}^* = \sum_{i=1}^n \gamma_i \underline{v}_i, \quad \gamma_i = \underline{x}^{*T} \underline{v}_i \tag{11.59}$$

which gives

$$
\begin{aligned}
A\underline{x}^* &= \sum_{i=1}^n \gamma_i\lambda_i\underline{v}_i, && (11.60)\\
\underline{x}^{*T}A\underline{x}^* &= \sum_{i=1}^n \gamma_i^2\lambda_i, && (11.61)\\
E(\underline{y}) = (\underline{x}^* - \underline{y})^T q_k(A)^2 A(\underline{x}^* - \underline{y}) &= \sum_{i=1}^n \gamma_i^2 q_k(\lambda_i)^2\lambda_i, && (11.62)\\
&\le \max_j q_k(\lambda_j)^2 \left(\sum_{i=1}^n \gamma_i^2\lambda_i\right) && (11.63)\\
&= \max_j q_k(\lambda_j)^2 (\underline{x}^{*T}A\underline{x}^*) && (11.64)\\
&\le \max_{t\in[\lambda_n,\lambda_1]} q_k(t)^2 (\underline{x}^{*T}A\underline{x}^*)\ . && (11.65)
\end{aligned}
$$

From these representations, minimising $E(\underline{y})$, or at least bounding $E(\underline{y})$, over the $k$-dimensional Krylov space corresponds to minimising or bounding the above expression over all polynomials of degree less than or equal to $k$ normalised by $q_k(0) = 1$. We have the following immediate remarks.

(i) If $A$ only has $m \le n$ distinct eigenvalues which we label as $s_1, \cdots, s_m \in \{\lambda_1, \cdots, \lambda_n\}$ then by taking the polynomial

$$q_m(t) = \frac{(s_1 - t)(s_2 - t)\cdots(s_m - t)}{s_1 s_2 \cdots s_m}, \qquad (11.66)$$

which is of degree $m$ with $q_m(0) = 1$, we have $q_m(\lambda_i) = 0$ for $i = 1, 2, \cdots, n$ and $E(\underline{x}_m) = 0$. The exact solution is obtained after $m$ iterations.

(ii) Following on from the above remark, it follows that if the $n$ eigenvalues are clustered around $m$ distinct points then we can expect a very accurate approximation once $m$ iterations have been completed.

To obtain a bound on the rate of convergence we can take a translated Chebyshev polynomial of degree $k$. The Chebyshev polynomial $T_k(t)$ of degree $k$ is defined via the properties

$$
\begin{aligned}
T_k(\cos s) &= \cos ks \qquad \text{which with } t = \cos s \text{ defines } T_k(t) \text{ in } [-1, 1], && (11.67)\\
T_k(\cosh s) &= \cosh ks \qquad \text{which with } t = \cosh s \text{ defines } T_k(t) \text{ in } t > 1, && (11.68)\\
T_k(-\cosh s) &= (-1)^k \cosh ks \qquad \text{which with } t = -\cosh s \text{ defines } T_k(t) \text{ in } t < -1. && (11.69)
\end{aligned}
$$

By defining the linear mapping $l : [\lambda_n, \lambda_1] \to [-1, 1]$ by

$$l(t) = \frac{t - (\lambda_1 + \lambda_n)/2}{(\lambda_1 - \lambda_n)/2} \qquad (11.70)$$

we have

$$|T_k(l(t))| \le 1 \quad \text{for } t \in [\lambda_n, \lambda_1]. \qquad (11.71)$$

Hence if we define

$$q_k(t) = \frac{T_k(l(t))}{T_k(l(0))} \qquad (11.72)$$

then we have a polynomial of degree $k$, $q_k(0) = 1$ and

$$\max_{t\in[\lambda_n,\lambda_1]} |q_k(t)| = \frac{1}{|T_k(l(0))|} \,. \tag{11.73}$$

As $T_k$ is an odd or even function

$$T_k(\pm l(0))^2 = \cosh^2 kt \quad \text{where } t \text{ is such that} \quad \cosh t = \frac{\lambda_1+\lambda_n}{\lambda_1-\lambda_n} = \frac{\lambda_1/\lambda_n+1}{\lambda_1/\lambda_n-1} = \frac{\kappa+1}{\kappa-1} \tag{11.74}$$

where $\kappa = \lambda_1/\lambda_n$ is the spectral condition number of $A$. Now

$$\mathrm{e}^t = \cosh t + \sinh t \tag{11.75}$$

and

$$\sinh^2 t = \cosh^2 t - 1 = \left(\frac{\kappa+1}{\kappa-1}\right)^2 - 1 = \frac{4k}{(\kappa-1)^2} \,. \tag{11.76}$$

Thus

$$\begin{aligned} \mathrm{e}^t &= \cosh t + \sinh t && (11.77)\\ &= \frac{\kappa + 2\sqrt{\kappa}+1}{\kappa-1} && (11.78)\\ &= \frac{(\sqrt{\kappa}+1)^2}{(\sqrt{\kappa}-1)(\sqrt{\kappa}+1)} && (11.79)\\ &= \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \,. && (11.80)\end{aligned}$$

Thus

$$\cosh kt = \frac{\mathrm{e}^{kt}+\mathrm{e}^{-kt}}{2} > \frac{\mathrm{e}^{kt}}{2} \tag{11.81}$$

and

$$\frac{1}{T_k(l(0))} = \frac{1}{\cosh kt} \le 2\mathrm{e}^{-kt} = 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \,. \tag{11.82}$$

This gives us the bound

$$E(\underline{x}_k) \le 4r^{2k}\left(\underline{x}^{*T} A \underline{x}^*\right) \quad \text{where} \quad r = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \,. \tag{11.83}$$

If the condition number $\kappa = \lambda_1/\lambda_n$ is large then $r$ is close to 1 which suggests slow convergence and when $\kappa \ge 1$ is not too large we are guaranteed fast convergence.

## 11.9 Remarks about preconditioning

Preconditioning is concerned with taking some symmetric positive definite matrix $M = C^2$, with $C$ also being symmetric and positive definite, which is easily invertible and replacing the problem

$$A\underline{x} = \underline{b} \tag{11.84}$$

by the equivalent problem

$$M^{-1}A\underline{x} = M^{-1}\underline{b} \quad \text{or equivalently} \quad (C^{-1}AC^{-1})C\underline{x} = C^{-1}\underline{b} \,. \tag{11.85}$$

As $A$ is symmetric and positive definite the matrix $\tilde{A} = C^{-1}AC^{-1}$ is also symmetric and positive definite and the basic CGM can be applied to the problem

$$\tilde{A}\underline{y} = \underline{\tilde{b}} \tag{11.86}$$

where $\underline{\tilde{b}} = C^{-1}\underline{b}$. The solution $\underline{y}^*$ to this problem is related to the solution of the original problem by $\underline{y}^* = C\underline{x}^*$ and likewise for each approximate vector $\underline{y}_k$ for this transformed problem we can define the vector $\underline{x}_k = C^{-1}\underline{y}_k$ as an approximation to $\underline{x}^*$. We will not consider here the implementation of the algorithm but it will just be noted that to implement the algorithm only involves solving at each stage a linear system of the form

$$M\underline{z}_k = \underline{r}_k \tag{11.87}$$

giving the residual $\underline{z}_k$ of the problem $M^{-1}A\underline{x} = \underline{b}$. We do not actually need $C$ at any stage in the implementation.

The convergence of the preconditioned problem depends on the eigenvalues of the transformed problem which in turn depends on the eigenvalues of $M^{-1}A = C^{-2}A$ which is similar to $\tilde{A} = C^{-1}AC^{-1}$. If the condition number of this matrix is much smaller than the condition number of $A$ and the solution of $M\underline{z}_k = \underline{r}_k$ is very efficient, then the preconditioned CGM will be much more efficient than the basic CGM. For many problems $\kappa(A)$ can be very large and preconditioning is essential to get a usable method.

Constructing a good preconditioner is highly problem dependent. Some general ones include taking the diagonal matrix

$$M = \mathrm{diag}(A) \tag{11.88}$$

or computing a lower triangular matrix $L$ which is as sparse as $A$ which satisfies

$$a_{ij} = (A)_{ij} = (LL^T)_{ij} \quad \text{for all } i, j \text{ for which } a_{ij} \neq 0. \tag{11.89}$$

The preconditioner $M = L\,L^T$ constructed in this way is known as an incomplete Cholesky preconditioner.

# 12 Exercises

## 12.1 Some numerical linear algebra related questions

1. The following was part of question 3 of the January 2002 examination paper.

   Consider the following Cholesky factorization of the real $n \times n$ matrix

$$A = \begin{pmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix} = \begin{pmatrix} l_1 & & & & \\ c_2 & l_2 & & & \\ & \ddots & \ddots & & \\ & & c_{n-1} & l_{n-1} & \\ & & & c_n & l_n \end{pmatrix} \begin{pmatrix} l_1 & c_2 & & & \\ & l_2 & c_3 & & \\ & & \ddots & \ddots & \\ & & & l_{n-1} & c_n \\ & & & & l_n \end{pmatrix},$$

   Give an algorithm to compute the coefficients $l_1, l_2, \cdots, l_n$ and $c_2, \cdots, c_n$.

   **[5 MARKS]**

   Show that we can write

$$l_k = g_1(l_{k-1}), \quad k = 2, 3, \cdots, n \quad \text{and} \quad c_k = g_2(c_{k-1}), \quad k = 3, \cdots, n$$

   giving the functions $g_1$ and $g_2$ in each case and find the fixed points of $g_1$ and $g_2$.

   **[4 MARKS]**

---

2. The following was part of question 3 of the January 2001 examination paper.

   For $k = 1, 2, \cdots, n$ let $A_k$ denote the $k \times k$ real symmetric and positive definite tri-diagonal matrices shown below which have the given Cholesky factorization $A_k = L_k L_k^T$.

$$A_k = \begin{pmatrix} a_1 & b_2 & & & \\ b_2 & a_2 & b_3 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{k-1} & a_{k-1} & b_k \\ & & & b_k & a_k \end{pmatrix} = \begin{pmatrix} l_1 & & & & \\ c_2 & l_2 & & & \\ & \ddots & \ddots & & \\ & & c_{k-1} & l_{k-1} & \\ & & & c_k & l_k \end{pmatrix} \begin{pmatrix} l_1 & c_2 & & & \\ & l_2 & c_3 & & \\ & & \ddots & \ddots & \\ & & & l_{k-1} & c_k \\ & & & & l_k \end{pmatrix},$$

   Give an algorithm to compute the coefficients $l_1, l_2, \cdots, l_n$ and $c_2, \cdots, c_n$. Also show that

$$l_k^2 = \frac{\det A_k}{\det A_{k-1}} .$$

   **[5 MARKS]**

---

3. The following was part of question 3 of the January 2000 examination paper.

   In the following $A = (a_{ij})$ denotes a symmetric positive definite $n \times n$ matrix with real eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$ and corresponding orthonormal eigenvectors $\underline{v}_1, \underline{v}_2, \cdots, \underline{v}_n$ respectively.

   (i) Show that $\lambda_i > 0$ for $i = 1, 2, \cdots, n$.

   (ii) Let $A = LL^T$ denote the Cholesky factorization of $A$ with $L$ being a lower triangular matrix with positive diagonal entries. Show the following.

(a)
$$A_k = L_k L_k^T, \quad k = 1, 2, \cdots ,$$
where $A_k$ and $L_k$ denote respectively the $k \times k$ principal submatrices of $A$ and $L$ respectively.

(b) If $L = (l_{ij})$ then
$$l_{ij}^2 \le a_{ii}, \quad j = 1, 2, \cdots , i \ .$$

---

4. The $n \times n$ tri-diagonal matrix

$$T_n = \begin{pmatrix} d_1 & b_2 & & & \\ a_2 & d_2 & b_3 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & d_{n-1} & b_n \\ & & & a_n & d_n \end{pmatrix}$$

has an $LU$ decomposition of the form

$$L_n U_n = \begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & \ddots & \ddots & & \\ & & l_{n-1} & 1 & \\ & & & l_n & 1 \end{pmatrix} \begin{pmatrix} u_1 & b_2 & & & \\ & u_2 & b_3 & & \\ & & \ddots & \ddots & \\ & & & u_{n-1} & b_n \\ & & & & u_n \end{pmatrix},$$

where it is assumed that no pivoting is necessary. Describe how you would compute the unknown coefficients of $L_n$ and $U_n$ and determine the total number of arithmetic operations involved in the process.

---

5. Let $A$ be an $n \times n$ matrix which has the factorization

$$A = LU$$

where $L$ is unit lower triangular and $U$ is upper triangular. Show that if $A_k$, $L_k$ and $U_k$ are respectively the $k \times k$ principal sub-matrices of $A$, $L$ and $U$ respectively then

$$A_k = L_k U_k$$

for $k = 1, 2, \cdots , n$. Also give an expression for the determinant of each matrix $A_k$ in terms of the entries of $U_k$.

---

6. The $(n+1) \times (n+1)$ matrix $B$ has the partitioned form

$$B = \begin{pmatrix} A & \underline{c} \\ \underline{r}^T & b_{n+1,n+1} \end{pmatrix},$$

where $A$ is an $n \times n$ matrix, $\underline{c}$ and $\underline{r}$ are $n \times 1$ vectors and $b_{n+1,n+1}$ is a scalar. If $A$ is non-singular and has the $LU$ decomposition $A = LU$, which has already been computed, describe how you would obtain the $LU$ decomposition of $B$.

---

7. Let $A$ be the symmetric $n \times n$ tri-diagonal matrix

$$A = \begin{pmatrix} 2+d & 1 & & & \\ 1 & 2+d & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2+d & 1 \\ & & & 1 & 2+d \end{pmatrix}$$

with $d > 0$. If $\underline{x}$ satisfies $A\underline{x} = \underline{b}$ then show that

$$\frac{\|\underline{b}\|_\infty}{4+d} \le \|\underline{x}\|_\infty \le \frac{\|\underline{b}\|_\infty}{d} \ .$$

In your answer you may assume the following. If $E$ is a square matrix and $\|E\| < 1$ then $I - E$ is non-singular with

$$\frac{1}{1+\|E\|} \le \|(I-E)^{-1}\| \le \frac{1}{1-\|E\|} \ .$$

The matrix norm $\|A\|_\infty$ has the representation

$$\|A\|_\infty = \max_i \left\{ \sum_{j=1}^{n} |a_{ij}| \right\} \ .$$

---

## 12.2 Some questions relating to solving nonlinear equations

1. This was question 1 of the January 2002 paper.

   Throughout this question let $\phi : \mathbb{R} \to \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$ denote sufficiently continuously differentiable functions with $\phi$ having a fixed point at $x^*$ and $f$ having a simple root at $x^*$.

   (i) Let $x_0 \in \mathbb{R}$ and $x_{k+1} = \phi(x_k)$, $k = 0, 1, 2, \cdots$.

   (a) In the case that $x_k \to x^*$ as $k \to \infty$, define what it means for the iteration to converge with order $p$ and show that we have order $p$ convergence when $\phi^{(r)}(x^*) = 0$, $r = 1, 2, \cdots, p-1$ and $\phi^{(p)}(x^*) \neq 0$ and give the error constant.

   **[4 MARKS]**

   Use this to find the order of convergence in the case when

   $$\phi(x) = x - \frac{f(x)}{f'(x)} \quad \text{and in the case when} \quad \phi(x) = x - \frac{hf(x)}{f(x+h) - f(x)},$$

   where $h \neq 0$ and assuming that $f''(x^*) \neq 0$, $f(x^* + h) \neq 0$ and $f(x^* + h) \neq hf'(x^*)$.

   **[6 MARKS]**

   (b) If $x_0 > x^*$ and $\phi$ is such that $\phi'(x^*) = 0$ and there are constants $c$, $L_1$ and $L_2$ such that for $x^* < x \le x_0$ we have

   $$0 < \phi'(x) \le L_1 < 1, \quad 0 < c < \phi''(x) \le L_2$$

   then explain why each $x_k > x^*$ and

   $$x_{k+1} - x^* \le \min\{L_1(x_k - x^*), L_2(x_k - x^*)^2/2\}.$$

   Hence show that $x_k \to x^*$ as $k \to \infty$.

   **[4 MARKS]**

   (ii) Describe briefly what are meant by inverse interpolation methods and direct interpolation methods for generating a sequence of approximations $(x_k)$ in a method for attempting to determine $x^*$.

   **[4 MARKS]**

   Let $g$ denote the inverse function to $f$. If the interpolating function to $g$ is the truncated Taylor series polynomial of degree $p-1$ about the point $y_k = f(x_k)$ then show that when $p = 3$ we get Chebyshev's method

   $$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} - \frac{f''(x_k)f(x_k)^2}{2f'(x_k)^3} .$$

   **[7 MARKS]**

2. The following was question 1 of the January 2001 examination paper.

Throughout this question let $\phi : \mathbb{R} \to \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$ denote sufficiently continuously differentiable functions with $\phi$ having a fixed point at $x^*$ and $f$ having a simple root at $x^*$.

(i) Let $\mathcal{I} = [x^* - h, x^* + h]$, for some $h > 0$ and suppose that

$$M = \max_{x \in \mathcal{I}} |\phi'(x)| < 1 .$$

Show the following:

(a) There is only one fixed point of $\phi$ in $\mathcal{I}$.

(b) If $x_0 \in \mathcal{I}$ then $|x_k - x^*| \le M^k |x_0 - x^*|$.

(c) Define what it means for the sequence $(x_k)$ to converge to $x^*$ with order $p$. Assuming that $x_k \to x^*$ as $k \to \infty$ with $\phi'(x^*) = 0$ and $\phi''(x^*) \neq 0$ show that the iteration converges with order two. Also explain why if $\phi''(x) > 0$ for all $x \in \mathcal{I}$ then given any $x_0 \in \mathcal{I}$,

$$x_1 \ge x_2 \ge \cdots \ge x_k \ge x_{k+1} \ge \cdots \ge x^*,$$

i.e. the sequence decreases to $x^*$.

**[10 MARKS]**

(ii) Assuming that $x_0$ has already been specified consider the following fixed point iterations

$$x_{k+1} = \phi(x_k), \quad \phi(x) = x - \frac{f(x)}{\alpha(x)}, \quad k = 0, 1, 2, \cdots$$

corresponding to the different choices of $\alpha$ given below.

$$\text{(a)} \quad \alpha(x) = f'(x_0), \qquad \text{(b)} \quad \alpha(x) = f'(x^*), \qquad \text{(c)} \quad \alpha(x) = f'(x) .$$

State a condition on $\phi$ for the iteration to converge with order of convergence of at least two. Assuming that in each case $x_k \to x^*$ as $k \to \infty$, show that the order of convergence in (a) is at least one and that the order of convergence in (b) and (c) is at least two.

**[5 MARKS]**

(iii) Describe briefly what is meant by an inverse interpolation method for generating a sequence of approximations $(x_k)$ in a method for attempting to determine $x^*$.

(a) Give the linear interpolants that are used in respectively the Secant method and in Newton's method.

(b) If the interpolating function to the inverse is the truncated Taylor series polynomial of degree 2 about the point $y_k = f(x_k)$ then show that $x_{k+1}$ is given by

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} - \frac{f''(x_k) f(x_k)^2}{2 f'(x_k)^3} .$$

**[10 MARKS]**

3. The following was question 2 of the January 2002 examination paper.

   Let $\underline{f} : D \to \mathbb{R}^n$ be twice continuously differentiable on an open connected set $D \subset \mathbb{R}^n$ which is such that the Jacobian matrix $J_f(\underline{x})$ is non-singular for all $\underline{x} \in D$. Also let $\underline{x}^* \in D$ be such that $\underline{f}(\underline{x}^*) = \underline{0}$.

   (i) Let $A$ denote an $n \times n$ real symmetric and positive definite matrix and let $g : D \to \mathbb{R}$ be defined by

   $$g(\underline{x}) = \frac{1}{2}\underline{f}(\underline{x})^T A \underline{f}(\underline{x}) \ .$$

   (a) Explain what is meant by a descent direction of $g$ at a point $\underline{x}_k$ and show that $\underline{p}$ is such a direction if $\underline{p}^T \nabla g(\underline{x}_k) < 0$. What is the steepest descent direction?

   **[3 MARKS]**

   (b) Show that the direction $-J_f(x)^{-1}\underline{f}(x)$ is a descent direction of $g$ at all points at which $\underline{f}(\underline{x}) \neq \underline{0}$ for all such matrices $A$ and give a matrix $A$ such that this direction is the steepest descent direction of $g$.

   **[4 MARKS]**

   (c) In the case that the matrix $A$ is the identity matrix, explain how this descent property is used in the damped Newton method and describe the damped Newton method.

   **[5 MARKS]**

   (d) In the case that $n = 1$ and we have $g(x) = f_1(x)^2/2$ give the Newton iteration for solving $f_1(x) = 0$ and give the Newton iteration for minimising $g(x)$ and indicate why they are generally different.

   **[3 MARKS]**

   (ii) Show that in the basic Newton method for systems

   $$\underline{x}_{k+1} - \underline{x}^* = -J_f(\underline{x}_k)^{-1}\underline{M}_k(\underline{x}^*) \quad \text{where} \quad \underline{M}_k(\underline{x}) := \underline{f}(\underline{x}_k) + J_f(\underline{x}_k)(\underline{x} - \underline{x}_k) \ .$$

   **[3 MARKS]**

   If for all points $\underline{x}_k \in D$ and for all $\underline{x} \in D$ there exists constants $\beta > 0$ and $L > 0$ such that

   $$\|J_f(x)^{-1}\| < \beta \quad \text{and} \quad \|\underline{f}(\underline{x}) - \underline{M}_k(\underline{x})\| \leq \frac{L}{2}\|\underline{x} - \underline{x}_k\|^2,$$

   where $\|.\|$ denotes a vector norm or the corresponding subordinate matrix norm as appropriate, then show that the basic Newton method will converge to $\underline{x}^*$ for all starting points $\underline{x}_0$ satisfying

   $$\|\underline{x}_0 - \underline{x}^*\| \leq \frac{\gamma}{L\beta}$$

   when $\gamma < 2$.

   **[7 MARKS]**

4. The following was question 2 of the January 2001 examination paper.

   Let $\underline{f} : D \to \mathbb{R}^n$ be twice continuously differentiable on an open connected set $D \subset \mathbb{R}^n$ which is such that the Jacobian matrix $J_f(\underline{x})$ is non-singular for all $\underline{x} \in D$. Also let $\underline{x}^* \in D$ be such that $\underline{f}(\underline{x}^*) = \underline{0}$.

   (i) Give the vector valued linear function $\underline{M}_k(\underline{x})$ which interpolates $\underline{f}$ and $J_f$ at a point $\underline{x} = \underline{x}_k$ and hence describe Newton's method for systems in the form

   $$\underline{x}_{k+1} = \underline{\phi}(\underline{x}_k), \quad k = 0, 1, 2, \cdots .$$

   If the Jacobian matrix satisfies the Lipschitz condition

   $$\|J_f(\underline{x}) - J_f(\underline{y})\| \le L\|\underline{x} - \underline{y}\| \quad \forall \underline{x}, \underline{y} \in D$$

   for some constant $L$, where $\|.\|$ denotes the vector norm or the corresponding subordinate matrix norm as appropriate, then show that

   $$\|\underline{f}(\underline{x}) - \underline{M}_k(\underline{x})\| \le \frac{L}{2}\|\underline{x} - \underline{x}_k\|^2 .$$ **[7 MARKS]**

   (In your answer you may assume the result

   $$\underline{f}(\underline{x}) - \underline{f}(\underline{y}) = \int_0^1 J_f(\underline{y} + t(\underline{x} - \underline{y}))\, dt \ \ (\underline{x} - \underline{y}) \quad \text{for all } \underline{x}, \underline{y} \in D.)$$

   (ii) Let $g : D \to \mathbb{R}$ be defined by

   $$g(\underline{x}) = \frac{1}{2}\underline{f}(\underline{x})^T \underline{f}(\underline{x}) . \qquad (*)$$

   Explain what is meant by a descent direction of $g$ at $\underline{x} = \underline{x}_k$ and show that the Newton direction $-J_f(\underline{x}_k)^{-1}\underline{f}(\underline{x}_k)$ is a descent direction of $g$ at $\underline{x}_k$ provided $\underline{f}(\underline{x}_k) \ne \underline{0}$. Indicate how this information is used in the damped Newton method.

   **[6 MARKS]**

   (iii) Describe briefly Newton's method for minimising $g$ in equation $(*)$ and explain briefly why this method is generally different from Newton's method for solving $\underline{f}(\underline{x}) = \underline{0}$.

   **[4 MARKS]**

   (iv) Let $A$ and $B$ be $n \times n$ diagonal matrices with positive diagonal entries and let $\underline{f}_1 : D \to \mathbb{R}^n$ be defined by

   $$\underline{f}_1(\underline{y}) = A\underline{f}(B\underline{y}) .$$

   Also let $\underline{x}_0, \underline{x}_1, \underline{x}_2, \cdots$ be points generated in Newton's method applied to $\underline{f}$ and let $\underline{y}_0, \underline{y}_1, \underline{y}_2, \cdots$ be points generated by Newton's method applied to $\underline{f}_1$. Show that if $\underline{x}_0 = B\underline{y}_0$ then $\underline{x}_k = B\underline{y}_k$ for $k = 1, 2, \cdots$.

   **[6 MARKS]**

   Explain why if the damped Newton method is used instead of the 'basic' Newton method then it need not be the case that $\underline{x}_k = B\underline{y}_k$ for $k = 1, 2, \cdots$.

   **[2 MARKS]**

5. The following was most of question 1 of the January 2000 examination paper.

   Suppose that a real function $f$ is $p$-times continuously differentiable and has a simple root at $x^*$ and let $g$ denote the inverse function to $f$.

   Describe briefly what is meant by an inverse interpolation method for generating a sequence of approximations $(x_k)$ in a method for attempting to determine $x^*$.

   If the interpolating function to $g$ is the truncated Taylor series polynomial of degree $p-1$ about the point $y_k = f(x_k)$ then show that when $p = 2$ we get Newton's method

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

   and when $p = 3$ we get Chebyshev's method

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} - \frac{f''(x_k)f(x_k)^2}{2f'(x_k)^3} \ .$$

   Also show that for a general degree $p \ge 2$ the order of convergence is at least $p$ and that when the order is $p$ the asymptotic error constant is

$$\frac{1}{p!} \left|f'(x^*)\right|^p \ \left|g^{(p)}(0)\right| \ .$$

---

6. The following were parts of a question from the January 1997 MA551 paper.

   Let $\underline{f} : D \to \mathbb{R}^n$ be a continuously differentiable function defined on an open connected set $D \subset \mathbb{R}^n$ and suppose that $\underline{f}(\underline{x}^*) = \underline{0}$ for a point $\underline{x}^* \in D$.

   The Newton iterative method involves the formula

$$\underline{x}_{k+1} = \underline{x}_k - \left(J_f(\underline{x}_k)\right)^{-1} \underline{f}(\underline{x}_k), \qquad (12.1)$$

   where $J_f(\underline{x}_k)$ is the Jacobian matrix of $\underline{f}$ at $\underline{x}_k$.

   (i) Show that if the $i$th component function $f_i(\underline{x})$ is linear in $\underline{x}$ then given any $\underline{x}_0 \in D$ the Newton iterates $\underline{x}_1, \underline{x}_2, \cdots$ will give $f_i(\underline{x}_k) = 0$, $k = 1, 2, \cdots$ .

   (ii) If $A$ and $B$ are non-singular constant matrices and $\underline{g}$ is defined by

$$\underline{g}(\underline{y}) = A\,\underline{f}(B\,\underline{y})$$

   and if $\underline{x}_0$ and $\underline{y}_0$ are related by $\underline{x}_0 = B\,\underline{y}_0$ then show that Newton's method applied to $\underline{g}$ generates points $\underline{y}_1, \underline{y}_2, \cdots$ which satisfy

$$\underline{x}_k = B\,\underline{y}_k \ .$$

---

7. The following were parts of a question from the January 1998 ST5051 paper.

(i) Let $\phi : \mathbb{R} \to \mathbb{R}$ be $p$-times continuously differentiable, $p \geq 1$, with a fixed point at $x^*$ and consider the fixed point iteration $x_{k+1} = \phi(x_k)$ given some $x_0 \in \mathbb{R}$. Also let $\mathcal{I} = [x^* - h, x^* + h]$, for some $h > 0$, and suppose that

$$M = \max_{x \in \mathcal{I}} |\phi'(x)| < 1 \ .$$

Show the following.

(a) If $x_0 \in \mathcal{I}$ and $\phi'(x^*) = 0$ with $\phi''(x) > 0$ for all $x \in \mathcal{I}$ then convergence to $x^*$ is from one side with specifically $x^* \leq x_{k+1} \leq x_k$ for $k = 1, 2, \cdots$.

(b) If $f(x^*) = 0$ and $f'(x^*) \neq 0$ then in each of the following cases convergence is at least quadratic.

$$\phi(x) = x - \frac{f(x)}{f'(x^*)} \quad \text{or } \phi(x) = x - \frac{f(x)}{f'(x)} \quad \text{or } \phi(x) = x - \frac{f(x)}{f'(x)(1 + \theta(x) f(x))},$$

where $\theta(x)$ is any bounded differentiable function and where $1 + \theta(x) f(x) \neq 0$ for all $x \in \mathcal{I}$.

(ii) Let $\underline{f} : \mathbb{R}^n \to \mathbb{R}^n$, $\underline{f} = (f_k)$, be twice continuously differentiable, let

$$g(\underline{x}) = \frac{1}{2} \underline{f}(\underline{x})^T \underline{f}(\underline{x})$$

and let $J_f(\underline{x})$ and $H_g(\underline{x})$ denote respectively the Jacobian matrix of $\underline{f}$ and the Hessian matrix of $g$.

Briefly explain why the Newton directions

$$-J_f(\underline{x})^{-1} \underline{f}(\underline{x}) \quad \text{and} \quad -H_g(\underline{x})^{-1} \nabla g(\underline{x})$$

for respectively solving $\underline{f}(\underline{x}) = \underline{0}$ and minimising $g(\underline{x})$ are generally different.

## 12.3 Some questions relating to the CGM

1. The following was part of question 3 of the January 2001 paper.

   In the following let $A$ denote an $n \times n$ real symmetric and positive definite matrix, let $\underline{b} \in \mathbb{R}^n$ and let $\underline{x}^* \in \mathbb{R}^n$ satisfy $A\underline{x}^* = \underline{b}$. Also let the quadratic $f : \mathbb{R}^n \to \mathbb{R}$ and the residual $\underline{r} : \mathbb{R}^n \to \mathbb{R}^n$ be given by

$$f(\underline{x}) = \frac{1}{2}\underline{x}^T A\underline{x} - \underline{x}^T \underline{b} \quad \text{and} \quad \underline{r}(\underline{x}) = \underline{b} - A\underline{x} \ .$$

   (a) Give the gradient $\nabla f$, the Hessian matrix $H_f$ and show that for any vector $\underline{d} \in \mathbb{R}^n$ we have

$$f(\underline{x}^* + \underline{d}) - f(\underline{x}^*) = \frac{1}{2}\underline{d}^T A\underline{d} \ .$$

   **[3 MARKS]**

   (b) Show that if $\underline{p} \neq 0$ and $\underline{x}$ are fixed vectors in $\mathbb{R}^n$ then $g(\alpha) = f(\underline{x} + \alpha\underline{p})$ has a minimum at

$$\alpha = \frac{\underline{p}^T \underline{r}(\underline{x})}{\underline{p}^T A\underline{p}} \ .$$

   **[2 MARKS]**

   (c) Let $\underline{p}_1, \underline{p}_2, \cdots, \underline{p}_n$ be non-zero vectors in $\mathbb{R}^n$.

   State what it means for $\underline{p}_1, \underline{p}_2, \cdots, \underline{p}_n$ to be $A$-conjugate and show that such vectors are linearly independent.

   **[3 MARKS]**

   Show that the vector $\underline{x}^*$ has the representation

$$\underline{x}^* = \sum_{i=1}^{n} \alpha_i \underline{p}_i \quad \text{where } \alpha_i = \frac{\underline{p}_i^T \underline{b}}{\underline{p}_i^T A\underline{p}_i}, \quad i = 1, 2, \cdots, n \ .$$

   **[2 MARKS]**

   For $k \le n$ show that the vector

$$\underline{x}_k = \sum_{i=1}^{k} \alpha_i \underline{p}_i$$

   minimises $f$ over the $k$-dimensional subspace span$\{\underline{p}_1, \cdots, \underline{p}_k\}$ and that $\alpha_i$ can be expressed in the form

$$\alpha_i = \frac{\underline{p}_i^T \underline{r}(\underline{x}_{i-1})}{\underline{p}_i^T A\underline{p}_i}.$$

   **[8 MARKS]**

   Also show that the residual $\underline{r}(\underline{x}_k)$ is orthogonal to the space span$\{\underline{p}_1, \cdots, \underline{p}_k\}$.

   **[2 MARKS]**

---

2. The following was part of question 3 of the January 2000 paper.

   In the following $A = (a_{ij})$ denotes a symmetric positive definite $n \times n$ matrix with real eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$ and corresponding orthonormal eigenvectors $\underline{v}_1, \underline{v}_2, \cdots, \underline{v}_n$ respectively.

   Let $\underline{p}_1, \underline{p}_2, \cdots, \underline{p}_n$ denote non-zero vectors in $\mathbb{R}^n$ which are $A$-conjugate.

   (a) Explain what the term $A$-conjugate means.

   (b) Show that $\underline{p}_1, \underline{p}_2, \cdots, \underline{p}_n$ are linearly independent.

   (c) Show that if $\underline{b} \in \mathbb{R}^n$ and $\underline{x} \in \mathbb{R}^n$ satisfies $A\underline{x} = \underline{b}$ then

   $$\underline{x} = \sum_{i=1}^{n} \alpha_i \underline{p}_i \quad \text{where} \quad \alpha_i = \frac{\underline{p}_i^T \underline{b}}{\underline{p}_i^T A \underline{p}_i}.$$

   (d) Let the $\alpha_i$ be as in part (c) and let the vectors $\underline{x}_k$, $\underline{e}_k$ and $\underline{r}_k$, $k = 1, 2, \cdots$ be defined by

   $$\underline{x}_k := \sum_{i=1}^{k} \alpha_i \underline{p}_i, \quad \underline{e}_k := \underline{x} - \underline{x}_k \quad \text{and} \quad \underline{r}_k := \underline{b} - A\underline{x}_k \ .$$

   Also let

   $$E(\underline{y}) := (\underline{y} - \underline{x})^T A (\underline{y} - \underline{x}) \ .$$

   Show that

   $$\underline{p}_i^T \underline{b} = \underline{p}_i^T \underline{r}_{i-1}, \quad i = 1, 2, \cdots, n$$

   and that

   $$E(\underline{x}_k) \le E(\underline{y}) \quad \text{for all } \underline{y} \in \operatorname{span}\{\underline{p}_1, \underline{p}_2, \cdots, \underline{p}_k\}.$$

---

3. This was the final part of question 3 of the January 2002 paper.

   Given that in the conjugate gradient method the residual $\underline{r}_k = \underline{b} - A\underline{x}_k \neq \underline{0}$ is $A$-conjugate to $\operatorname{span}\{\underline{p}_1, \cdots, \underline{p}_{k-1}\}$ determine $\beta_k$ in

   $$\underline{p}_{k+1} = \underline{r}_k + \beta_k \underline{p}_k$$

   such that $\underline{p}_{k+1}$ is $A$-conjugate to $\operatorname{span}\{\underline{p}_1, \cdots, \underline{p}_k\}$.

   **[4 MARKS]**

---

4. If $\underline{p}_1, \cdots, \underline{p}_n$ are mutually orthogonal, i.e. $\underline{p}_i^T \underline{p}_j = 0, i \neq j$, then show the following:

   (i) $\underline{p}_1, \cdots, \underline{p}_n$ are linearly independent.

   (ii) For any vector $\underline{v}$,

   $$\underline{v} = \sum_{1}^{n} c_i \underline{p}_i, \quad \text{with} \quad c_i = \frac{\underline{p}_i^T \underline{v}}{\underline{p}_i^T \underline{p}_i}, \ i = 1, \cdots, n$$

   and that

   $$\underline{v}^T \underline{v} = \sum_{1}^{n} c_i^2 \underline{p}_i^T \underline{p}_i \ .$$

   (iii) If $\underline{v}_k = \sum_1^k c_i \underline{p}_i$ where $c_i$ is as defined in (ii) then

   $$(\underline{v} - \underline{v}_k)^T (\underline{v} - \underline{v}_k) \le (\underline{v} - \underline{y})^T (\underline{v} - \underline{y}) \quad \forall \underline{y} \in \operatorname{span}\{\underline{p}_1, \cdots, \underline{p}_k\} \ .$$

---

In the following set of questions it is assumed that $\underline{p}_1, \underline{p}_2, \cdots, \underline{p}_n$ are non zero vectors in $n$-dimensional space $\mathbb{R}^n$ and that $A$ is an $n \times n$ symmetric positive definite matrix. We will also assume that $\underline{x}^*$ is the unique vector satisfying

$$A\underline{x}^* = \underline{b}$$

where $\underline{b}$ is a vector of length $n$, and that the quadratic $f$, the error $\underline{e}$ and the residual $\underline{r}$ are defined respectively by

$$f(\underline{x}) = \frac{1}{2}\underline{x}^T A\underline{x} - \underline{x}^T\underline{b}, \quad \underline{e}(\underline{x}) = \underline{x}^* - \underline{x} \quad \text{and} \quad \underline{r}(\underline{x}) = \underline{b} - A\underline{x}\,.$$

---

5. The following question is similar to question 4 and is more closely related to the conjugate gradient method. If $\underline{p}_1, \cdots, \underline{p}_n$ are mutually $A$-conjugate, i.e. $\underline{p}_i^T A\underline{p}_j = 0, i \neq j$, then show the following:

   (i) $\underline{p}_1, \cdots, \underline{p}_n$ are linearly independent.

   (ii) For any vector $\underline{v}$,

   $$\underline{v} = \sum_1^n c_i \underline{p}_i, \quad \text{with} \quad c_i = \frac{\underline{p}_i^T A\underline{v}}{\underline{p}_i^T A\underline{p}_i}, \; i = 1, \cdots, n$$

   and that

   $$\underline{v}^T A\underline{v} = \sum_1^n c_i^2 \underline{p}_i^T A\underline{p}_i$$

   (iii) If $\underline{v}_k = \sum_1^k c_i \underline{p}_i$ where $c_i$ is as defined in (ii) then

   $$(\underline{v} - \underline{v}_k)^T A(\underline{v} - \underline{v}_k) \leq (\underline{v} - \underline{y})^T A(\underline{v} - \underline{y}) \quad \forall \underline{y} \in \text{span}\{\underline{p}_1, \cdots, \underline{p}_k\}\ .$$

---

6. Show that if $\underline{x}$ and $\underline{p} \neq \underline{0}$ are given then the quadratic

   $$g(\alpha) = f(\underline{x} + \alpha\underline{p})$$

   is minimized when

   $$\alpha = \frac{\underline{p}^T \underline{r}(\underline{x})}{\underline{p}^T A\underline{p}}\ .$$

---

7. Show that the vector $\underline{x}^*$ satisfying the linear equations is given by

   $$\underline{x}^* = \underline{x}_0 + \sum_1^n \alpha_i \underline{p}_i$$

   where $\alpha_i = (\underline{p}_i^T \underline{r}_0)/(\underline{p}_i^T A\underline{p}_i),\ \underline{r}_0 = \underline{r}(\underline{x}_0),\ i = 1, \cdots, n$ . Further show that if

   $$\underline{x}_k = \underline{x}_0 + \sum_1^k \alpha_i \underline{p}_i$$

   and $\underline{r}_k = \underline{r}(\underline{x}_k)$ then

   $$\underline{p}_k^T \underline{r}_0 = \underline{p}_k^T \underline{r}_{k-1}\ .$$

---

8. The following is concerned with the finite termination criterion of the conjugate gradient method.

   It can be shown that the choice of the $A$-conjugate search directions in the conjugate gradient method implies that

   $$\begin{aligned}\mathrm{span}\{\underline{p}_1, \cdots, \underline{p}_k\} &= \mathrm{span}\{\underline{r}_0, A\underline{r}_0, A^2\underline{r}_0, \cdots, A^{k-1}\underline{r}_0\} \\ &= \mathrm{span}\{A\underline{e}_0, A^2\underline{e}_0, A^3\underline{e}_0, \cdots, A^k\underline{e}_0\}\end{aligned}$$

   where $\underline{e}_0 = \underline{x}^* - \underline{x}_0$. As a consequence the error like term

   $$\|\underline{x}^* - \underline{x}_k\|_A^2 := (\underline{x}^* - \underline{x}_k)^T A(\underline{x}^* - \underline{x}_k) = (\underline{e}_0 - \underline{w}_k)^T A(\underline{e}_0 - \underline{w}_k)$$

   where $\underline{w}_k$ has the representation

   $$\underline{w}_k = \left(\sum_1^k c_i A^i\right) \underline{e}_0 \ .$$

   Observe that $\|\underline{x}^* - \underline{x}_k\|_A = 0$ when $\underline{w}_k = \underline{e}_0 = A^{-1}\underline{r}_0$. The following is concerned with when this occurs by considering representations of the inverse $A^{-1}$ by polynomials in $A$.

   The Cayley-Hamilton theorem states that every polynomial satisfies its own characteristic equation, i.e. if

   $$\det(tI - A) = (t - \lambda_1)(t - \lambda_2) \cdots (t - \lambda_n) = 0$$

   is the characteristic equation with $\lambda_1, \cdots, \lambda_n$ as the eigenvalues of $A$ then we have the matrix identity

   $$(A - \lambda_1 I)(A - \lambda_2 I) \cdots (A - \lambda_n I) = 0,$$

   where $I$ denotes the $n \times n$ identity matrix. In the terminology of linear algebra it is said that the characteristic polynomial annihilates $A$. If in the case of a symmetric positive definite matrix $A$ there are only $m$ distinct eigenvalues $\mu_1, \cdots, \mu_m$ then show that for all vectors $\underline{x}$ we have

   $$(A - \mu_1 I)(A - \mu_2 I) \cdots (A - \mu_m I)\underline{x} = \underline{0}$$

   and hence prove the matrix identity

   $$(A - \mu_1 I)(A - \mu_2 I) \cdots (A - \mu_m I) = 0,$$

   i.e. a polynomial of degree $m$ annihilates $A$. Thus prove that in this case $A^{-1}$ has the representation

   $$A^{-1} = \gamma_0 I + \gamma_1 A + \cdots + \gamma_{m-1} A^{m-1}$$

   and explain why this proves that $\|\underline{x}^* - \underline{x}_m\|_A = 0$ and $A\underline{w}_m = \underline{r}_0$.

---