

Lecture Notes for Inf-Mat 3350/4350, 2004

Tom Lyche

July 26, 2004



# Contents

<b>Preface</b>	<b>v</b>
<b>Introduction</b>	<b>vii</b>
<b>1 Triangular Factorization and Gaussian Elimination</b>	<b>1</b>
1.1 Algebraic Properties of Triangular Matrices . . . . .	1
1.2 Existence and Uniqueness of the <i>LU</i> -factorization . . . . .	3
1.3 The Symmetric Case . . . . .	5
1.4 When is Gaussian Elimination without pivoting possible? . .	6
<b>2 Positive Definite Linear Systems</b>	<b>9</b>
2.1 Definitions and Examples . . . . .	9
2.2 Triangular Factorization . . . . .	11
2.3 When is a Matrix Positive Definite? . . . . .	14
<b>3 Some Model Problems</b>	<b>17</b>
3.1 A Tridiagonal Matrix . . . . .	17
3.2 The Poisson Problem . . . . .	20
3.3 The Kronecker Product . . . . .	22
3.4 A banded Matrix . . . . .	27
3.5 Problems . . . . .	30
<b>4 Fast Direct Solution of a Large Linear System</b>	<b>31</b>
4.1 A Fast Poisson Solver based on Diagonalization and FFT . .	31
4.2 A Fast Poisson Solver based on the Discrete Sine and Fourier Transforms . . . . .	33
4.2.1 The Discrete Sine Transform (DST) . . . . .	33
4.2.2 The Discrete Fourier Transform (DFT) . . . . .	34
4.2.3 The Fast Fourier Transform (FFT) . . . . .	35
4.2.4 A Poisson Solver based on the FFT . . . . .	38
4.3 Problems . . . . .	38

<b>5</b>	<b>The Conjugate Gradient Method</b>	<b>41</b>
5.1	Derivation and Basic Properties . . . . .	41
5.2	Numerical Examples . . . . .	45
5.3	Minimization . . . . .	47
5.4	Convergence . . . . .	49
5.5	Preconditioning . . . . .	53
5.6	Preconditioning Example . . . . .	56
5.7	Problems . . . . .	59

# Preface

These lecture notes are a revised version of previous notes entitled Lecture notes in MoD200, 2002, and are written for a course in numerical linear algebra given at the advanced undergraduate level at the University of Oslo. They have been written to be used as a supplement to Leon's textbook [3].

**Acknowledgement** I would like to thank Njål Foldnes for help with typing the previous version of these notes.

Oslo, 27 July, 2004

Tom Lyche



# Introduction

The main topic in these notes concerns methods for solving linear systems of equations. In scientific computing we often encounter systems with special structures and thousands or millions of unknowns. Gaussian elimination is often not the best way to solve such problems.

We will pay special attention to methods suitable for positive definite systems. We consider Cholesky factorization and the Conjugate Gradient Method in addition to fast methods for special linear systems.

In these note we will denote matrices by capital letters  $A, B, C \dots$  and normally vectors by lower case letters  $x, y, z, \dots$ . The entry in the  $i$ th row and  $j$ th column of a matrix  $A$  will be denoted  $a_{i,j}$ ,  $a_{ij}$ ,  $A(i, j)$  or  $(A)_{i,j}$ . We write  $A \in \mathbb{R}^{m,n}$  if the entries are real numbers and  $A$  has  $m$  rows and  $n$  columns, and  $x \in \mathbb{R}^n$  for a vector with  $n$  real components. The matrix is square if  $m = n$ . A vector  $x \in \mathbb{R}^n$  can be either a column- or a row vector, but most often it will be a column vector. If  $x$  is a column vector then the transpose denoted  $x^T$  is a row vector. For matrices and vectors with complex entries we use the notation  $A \in \mathbb{C}^{m,n}$  and  $x \in \mathbb{C}^n$ .

A square matrix  $A \in \mathbb{R}^{n,n}$  is *singular* if we can find a nonzero vector  $x \in \mathbb{R}^n$  such that  $Ax = 0$ . A matrix which is not singular is said to be *nonsingular*. Thus a matrix is nonsingular if  $Ax = 0$  implies that  $x = 0$ . A matrix  $A \in \mathbb{R}^{n,n}$  is *invertible* if there is a matrix  $B$  such that  $AB = BA = I$ , where  $I \in \mathbb{R}^{n,n}$  is a diagonal matrix with ones on the diagonal, the *identity matrix of order  $n$* . We write  $B = A^{-1}$ , the *inverse* of  $A$ . A matrix  $A \in \mathbb{R}^{n,n}$  has an inverse if and only if it is nonsingular. The following elementary fact about products of matrices will be used repeatedly in these notes: If a product  $C = AB$  of two square matrices  $A, B$  is nonsingular then each factor  $A$  and  $B$  is nonsingular. In particular, if  $A, B$  are square matrices and  $AB = I$  the identity matrix, then both  $A$  and  $B$  are invertible with  $A^{-1} = B$  and  $B^{-1} = A$ .

These notes discusses methods for finding a vector  $x \in \mathbb{R}^n$  such that  $Ax = b$ , where  $b \in \mathbb{R}^n$  and  $A \in \mathbb{R}^{n,n}$  is nonsingular.





# Chapter 1

## Triangular Factorization and Gaussian Elimination

In Gaussian elimination we compute a triangular factorization of the coefficient matrix  $A$ . This factorization is known as an  $LU$ -factorization of  $A$ . In this chapter we discuss the general theory of  $LU$ -factorization of a nonsingular matrix.

### 1.1 Algebraic Properties of Triangular Matrices

We start with a result about the inverse of block-triangular matrices.

**Lemma 1.1** *Suppose*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$$

where  $A, A_{11}$  and  $A_{22}$  are square matrices. Then  $A$  is nonsingular if and only if both  $A_{11}$  and  $A_{22}$  are nonsingular. In that case

$$A^{-1} = \begin{bmatrix} A_{11}^{-1} & -A_{11}^{-1}A_{12}A_{22}^{-1} \\ 0 & A_{22}^{-1} \end{bmatrix} \quad (1.1)$$

**Proof** If  $A_{11}$  and  $A_{12}$  are nonsingular then

$$\begin{bmatrix} A_{11}^{-1} & -A_{11}^{-1}A_{12}A_{22}^{-1} \\ 0 & A_{22}^{-1} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} = I$$

and  $A$  is nonsingular with the indicated inverse. Conversely, let  $B$  be the inverse of the nonsingular matrix  $A$ . We partition  $B$  conformally with  $A$  and have

$$BA = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} = I$$

Using block-multiplication we find

$$B_{11}A_{11} = I, \quad B_{21}A_{11} = 0, \quad B_{21}A_{12} + B_{22}A_{22} = I.$$

The first equation implies that  $A_{11}$  is invertible, this in turn implies that  $B_{21} = 0$  in the second equation, and then the third equation simplifies to  $B_{22}A_{22} = I$ . We conclude that also  $A_{22}$  is invertible.  $\square$

Consider now a triangular matrix.

**Lemma 1.2** *An upper (lower) triangular matrix  $A = [a_{ij}] \in \mathbb{R}^{n,n}$  is non-singular if and only if the diagonal elements  $a_{ii}$ ,  $i = 1, \dots, n$  are nonzero. In that case the inverse is upper (lower) triangular with diagonal elements  $a_{ii}^{-1}$ ,  $i = 1, \dots, n$ .*

**Proof** We use induction on  $n$ . The result holds for  $n = 1$ : The 1-by-1 matrix  $A = (a_{11})$  is invertible if and only if  $a_{11} \neq 0$  and in that case  $A^{-1} = (a_{11}^{-1})$ . Suppose the result holds for  $n = k$  and let  $A \in \mathbb{R}^{k+1,k+1}$  be upper triangular. We partition  $A$  in the form

$$A = \begin{bmatrix} A_k & a_k \\ 0 & a_{k+1,k+1} \end{bmatrix}$$

and note that  $A_k \in \mathbb{R}^{k,k}$  is upper triangular. By Lemma 1.1  $A$  is nonsingular if and only if  $A_k$  and  $(a_{k+1,k+1})$  are nonsingular and in that case

$$A^{-1} = \begin{bmatrix} A_k^{-1} & -A_k^{-1}a_k a_{k+1,k+1}^{-1} \\ 0 & a_{k+1,k+1}^{-1} \end{bmatrix}.$$

By the induction hypothesis  $A_k$  is nonsingular if and only if the diagonal elements  $a_{11}, \dots, a_{kk}$  of  $A_k$  are nonzero and in that case  $A_k^{-1}$  is upper triangular with diagonal elements  $a_{ii}^{-1}$ ,  $i = 1, \dots, k$ . The result for  $A$  follows.  $\square$

**Lemma 1.3** *The product  $C = AB = (c_{ij})$  of two upper(lower) triangular matrices  $A = (a_{ij})$  and  $B = (b_{ij})$  is upper(lower) triangular with diagonal elements  $c_{ii} = a_{ii}b_{ii}$  for all  $i$ .*

**Proof** Exercise.  $\square$

A matrix is **unit triangular** if it is triangular with 1's on the diagonal.

**Lemma 1.4** *For a unit upper(lower) triangular matrix  $A \in \mathbb{R}^{n,n}$ :*

1.  $A$  is invertible and the inverse is unit upper(lower) triangular.
2. The product of two unit upper(lower) triangular matrices is unit upper(lower) triangular.

**Proof** 1. follows from Lemma 1.2, while Lemma 1.3 implies 2.  $\square$

## 1.2 Existence and Uniqueness of the $LU$ -factorization

We say that  $A = LU$  is an  $LU$ -factorization of  $A \in \mathbb{R}^{n,n}$  if  $L \in \mathbb{R}^{n,n}$  is lower triangular and  $U \in \mathbb{R}^{n,n}$  is upper triangular. In addition we will assume that  $L$  is unit triangular.

**Example 1.5** The equation

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1/2 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ 0 & 3/2 \end{bmatrix}$$

gives an  $LU$ -factorization of the 2-by-2 matrix  $A$ .

Not every nonsingular matrix has an  $LU$ -factorization.

**Example 1.6** An  $LU$ -factorization of  $A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$  must satisfy the equation

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ l_1 & 1 \end{bmatrix} \begin{bmatrix} u_1 & u_3 \\ 0 & u_2 \end{bmatrix} = \begin{bmatrix} u_1 & u_3 \\ l_1 u_1 & l_1 u_3 + u_2 \end{bmatrix}$$

for the unknowns  $l_1$  in  $L$  and  $u_1, u_2, u_3$  in  $U$ . Comparing  $(1,1)$ -elements we see that  $u_1 = 0$ , which makes it impossible to satisfy the condition  $1 = l_1 u_1$  for the  $(2,1)$  element. We conclude that  $A$  has no  $LU$ -factorization.

The following lemma shows that the  $LU$ -factorization in Example 1.5 is unique.

**Lemma 1.7** *The  $LU$ -factorization of a nonsingular matrix is unique whenever it exists.*

**Proof** Suppose  $A = L_1 U_1 = L_2 U_2$  are two  $LU$ -factorizations of the nonsingular matrix  $A$ . The equation  $L_1 U_1 = L_2 U_2$  can be written in the form  $L_2^{-1} L_1 = U_2 U_1^{-1}$ , where by lemmas 1.2-1.4  $L_2^{-1} L_1$  is unit lower triangular and  $U_2^{-1} U_1$  is upper triangular. But then both matrices must be diagonal with ones on the diagonal. We conclude that  $L_2^{-1} L_1 = I = U_1 U_2^{-1}$  which means that  $L_1 = L_2$  and  $U_1 = U_2$ .  $\square$

Consider next existence of the  $LU$ -factorization. For a matrix  $A = (a_{ij}) \in \mathbb{R}^{n,n}$  we let

$$A_k = \begin{bmatrix} a_{11} & \dots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kk} \end{bmatrix} \in \mathbb{R}^{k,k}$$

denote the *leading principal submatrices* of  $A$  for  $k = 1, \dots, n$ . Thus  $A_1 = (a_{11})$  is a 1-by-1 matrix, and  $A_n = A$ .

**Lemma 1.8** Suppose  $A = LU$  is the  $LU$ -factorization of  $A \in \mathbb{R}^{n,n}$ . For  $k = 1, \dots, n$  let  $A_k, L_k, U_k$  be the leading principal submatrices of  $A, L, U$ , respectively. Then  $A_k = L_k U_k$  is the  $LU$ -factorization of  $A_k$  for  $k = 1, \dots, n$ .

**Proof** We partition  $A = LU$  as follows:

$$A = \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} = \begin{bmatrix} L_k & 0 \\ M_k & N_k \end{bmatrix} \begin{bmatrix} U_k & V_k \\ 0 & W_k \end{bmatrix} = LU,$$

where  $D_k, N_k, W_k \in \mathbb{R}^{n-k, n-k}$ . Using block-multiplication we find the equations

$$A_k = L_k U_k \tag{1.2}$$

$$B_k = L_k V_k \tag{1.3}$$

$$C_k = M_k U_k \tag{1.4}$$

$$D_k = M_k V_k + N_k W_k \tag{1.5}$$

Since  $L_k$  is unit lower triangular and  $U_k$  is upper triangular we see that (1.2) gives the  $LU$ -factorization of  $A_k$ .  $\square$

**Theorem 1.9** Suppose  $A \in \mathbb{R}^{n,n}$  is nonsingular. Then  $A$  has an  $LU$ -factorization if and only if the leading principal submatrices  $A_k$  are nonsingular for  $k = 1, \dots, n - 1$ .

**Proof** Suppose  $A$  is nonsingular with the  $LU$ -factorization  $A = LU$ . Since  $A$  is nonsingular it follows that  $L$  and  $U$  are nonsingular. By (1.2) we have  $A_k = L_k U_k$ . Since  $L_k$  is unit lower triangular it is nonsingular. Moreover  $U_k$  is nonsingular since its diagonal elements are among the nonzero diagonal elements of  $U$ . But then  $A_k$  is nonsingular.

Conversely, suppose  $A = A_n$  is nonsingular and  $A_k$  is nonsingular for  $k = 1, \dots, n - 1$ . We use induction on  $n$  to show that  $A$  has a  $LU$ -factorization. The result is clearly true for  $n = 1$ , since the  $LU$ -factorization of a 1-by-1 matrix is  $(a_{11}) = (1)(a_{11})$ . Suppose that  $A_1, \dots, A_{n-1}$  are nonsingular implies that  $A_{n-1}$  has an  $LU$ -factorization, and suppose that  $A_1, \dots, A_n$  are nonsingular. To show that  $A = A_n$  has a  $LU$ -factorization we consider (1.2)-(1.5) with  $k = n - 1$ . In this case  $C_k$  and  $M_k$  are row vectors,  $B_k$  and  $V_k$  are column vectors, and  $D_k = (a_{nn})$ ,  $N_k = (1)$ , and  $W_k = (u_{nn})$  are 1-by-1 matrices, i.e. scalars. The  $LU$ -factorization of  $A_{n-1}$  is given by (1.2), and since  $A_{n-1}$  is nonsingular we see that  $L_{n-1}$  and  $U_{n-1}$  are nonsingular. But then (1.3) has a unique solution  $V_{n-1}$ , (1.4) has a unique solution  $M_{n-1}$ , and setting  $N_{n-1} = (1)$  in (1.5) we obtain  $u_{nn} = a_{nn} - M_{n-1} V_{n-1}$ . Thus we have constructed an  $LU$ -factorization of  $A$ .  $\square$

Consider Examples 1.5 and 1.6. In Example 1.5 both  $A_1 = (2)$  and  $A_2 = A$  are nonsingular and  $A$  has an unique  $LU$ -factorization. In Example 1.6

$A = A_2$  is nonsingular, but  $A_1 = (0)$  is singular. Thus  $A$  has no  $LU$ -factorization.

The  $LU$ -factorization of a singular matrix exists in certain cases, but the theory is more complicated.

### 1.3 The Symmetric Case

Suppose  $A \in \mathbb{R}^{n,n}$  is symmetric and nonsingular with an  $LU$ -factorization  $A = LU$ . We can factor  $A$  further as  $A = LDM^T$  where  $M^T = D^{-1}U$  and  $D$  is a diagonal matrix with the diagonal elements of  $U$  on the diagonal. It follows that  $M^T$  is unit upper triangular and since  $A^T = A$  we find  $A^T = (LDM^T)^T = MDL^T = LU = A$ . Now  $M(DL^T)$  and  $LU$  are two  $LU$ -factorizations of  $A$  and by the uniqueness of the  $LU$ -factorization we must have  $M = L$ . Thus  $A = LDL^T$ , where  $L$  is unit lower triangular and  $D$  is diagonal and we denote this an  $LDL^T$ -factorization of  $A$ . The discussion above shows that the  $LDL^T$ -factorization is unique if it exists.

**Example 1.10** Here is the  $LDL^T$ -factorization of a 3-by-3 matrix:

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 0 & -2/3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3/2 & 0 \\ 0 & 0 & 4/3 \end{bmatrix} \begin{bmatrix} 1 & -1/2 & 0 \\ 0 & 1 & -2/3 \\ 0 & 0 & 1 \end{bmatrix} = LDL^T.$$

The condition for the existence of an  $LDL^T$ -factorization and an  $LU$ -factorization are (apart from symmetry) the same.

**Theorem 1.11** *Suppose  $A \in \mathbb{R}^{n,n}$  is nonsingular. Then  $A$  has an  $LDL^T$ -factorization if and only if  $A = A^T$  and  $A_k$  is nonsingular for  $k = 1, \dots, n-1$ .*

**Proof** If  $A_1, \dots, A_n$  are nonsingular then  $A$  has an  $LU$ -factorization which after what we have shown reduces to an  $LDL^T$ -factorization if  $A$  is symmetric. Conversely, if  $A = LDL^T$  is an  $LDL^T$ -factorization of  $A$  then  $A$  is symmetric since  $LDL^T$  is symmetric and  $A$  has an  $LU$ -factorization with  $U = DL^T$ . By Theorem 1.9 we conclude that  $A_1, \dots, A_{n-1}$  are nonsingular.  $\square$

It can be shown that the  $LDL^T$ -factorization of an  $n$ -by- $n$  matrix can be computed in  $O(n^3/3)$  flops (additions+subtractions+multiplications+divisions). This is half of the  $= (2n^3/3)$  flops required for Gaussian elimination (where one does not take advantage of the symmetry). Once the  $LDL^T$ -factorization is known we can solve a system  $Ax = b$  in three steps:  $Lz = b$ ,  $Dy = z$ ,  $L^T x = y$ .

## 1.4 When is Gaussian Elimination without pivoting possible?

*Gaussian elimination* is a process to reduce a matrix to upper triangular form. We ask the question when this reduction can be done using only type III row-operations, i.e. by repeatedly replacing a row in a matrix by its sum with a multiple of another row. We need to take a close look at the Gaussian elimination process. Suppose  $A^{(1)} = A$  and assume for some  $i \geq 1$  that  $A^{(i)}$  is in the form:

$$A^{(i)} = \left[ \begin{array}{cccc|ccc} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1,i-1}^{(1)} & a_{1,i}^{(1)} & \cdots & a_{1,n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2,i-1}^{(2)} & a_{2,i}^{(2)} & \cdots & a_{2,n}^{(2)} \\ \vdots & \ddots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{i-1,i-1}^{(i-1)} & a_{i-1,i}^{(i-1)} & \cdots & a_{i-1,n}^{(i-1)} \\ \hline 0 & \cdots & & 0 & a_{i,i}^{(i)} & \cdots & a_{i,n}^{(i)} \\ \vdots & \ddots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & & 0 & a_{n,i}^{(i)} & \cdots & a_{n,n}^{(i)} \end{array} \right] = \begin{bmatrix} B_i & C_i \\ 0 & D_i \end{bmatrix} \quad (1.6)$$

The  $i - 1$  first columns of  $A^{(i)}$  have zeros below the diagonal. Suppose  $a_{i,i}^{(i)} \neq 0$ . We can then zero out the entries under the diagonal in column  $i$  of  $A^{(i)}$  by using row-operations of type III only. We find

$$A^{(i+1)} = \begin{bmatrix} B_{i+1} & C_{i+1} \\ 0 & D_{i+1} \end{bmatrix}$$

where

$$D_{i+1} = \begin{bmatrix} a_{i+1,i+1}^{(i+1)} & \cdots & a_{i+1,n}^{(i+1)} \\ \vdots & & \vdots \\ a_{n,i+1}^{(i+1)} & \cdots & a_{n,n}^{(i+1)} \end{bmatrix}$$

with

$$a_{kj}^{(i+1)} = a_{kj}^{(i)} - l_{ki} a_{ij}^{(i)}, \quad k, j = i + 1, \dots, n \quad (1.7)$$

$$l_{ki} = a_{ki}^{(i)} / a_{ii}^{(i)}, \quad k = i + 1, \dots, n. \quad (1.8)$$

If the pivot elements  $a_{i,i}^{(i)} \neq 0$  for  $i = 1, \dots, n - 1$  then  $U = A^{(n)}$  can be computed by this process and is upper triangular.

**Theorem 1.12** *We have  $a_{i,i}^{(i)} \neq 0$  for  $i = 1, \dots, n - 1$  if and only if the leading principal submatrices*

$$A_i = \begin{bmatrix} a_{11} & \cdots & a_{1i} \\ \vdots & & \vdots \\ a_{i1} & \cdots & a_{ii} \end{bmatrix}$$

of  $A$  are nonsingular for  $i = 1, \dots, n - 1$ .

**Proof** Observe that the matrix  $B_i$  in (1.6) is computed from  $A$  by using only entries from  $A_i$  and that only row-operations of type III are used. It follows that  $A_i$  is nonsingular if and only if  $B_i$  is nonsingular. By Lemma 1.2  $B_i$  is nonsingular if and only if  $a_{kk}^{(k)} \neq 0$ ,  $k = 1, \dots, i$ . Take  $i = n - 1$ .  $\square$

We see that the characterization in Theorem 1.12 is equivalent to the condition for existence of the  $LU$ -factorization for a nonsingular matrix in Theorem 1.9. Therefore it follows that a nonsingular matrix  $A$  has an  $LU$ -factorization if and only if  $a_{ii}^{(i)} \neq 0$  for  $i = 1, 2, \dots, n - 1$ .

**Example 1.13** Consider the 3-by-3 matrix in Example 3

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}.$$

Since  $A_k$  is nonsingular for  $k = 1, 2, 3$  we can reduce  $A$  to upper triangular form by Gaussian elimination without row interchanges.





## Chapter 2

# Positive Definite Linear Systems

This chapter contains 3 subsections. In Section 2.1 we define a positive definite matrix and give some examples. We are interested in solving linear systems  $Ax = b$  and in Section 2.2 we discuss triangular factorization of positive definite matrices. We also give algorithms for the case of a banded  $A$ . To show that a matrix is positive definite we need criteria. This is the topic of Section 2.3.

### 2.1 Definitions and Examples

Suppose  $A \in \mathbb{R}^{n,n}$  is a square matrix. The function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  given by

$$f(x) = x^T Ax = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j$$

is called a *quadratic form*. We say that  $A$  is

- (i) *positive definite* if  $A^T = A$  and  $x^T Ax > 0$  for all nonzero  $x \in \mathbb{R}^n$ .
- (ii) *positive semidefinite* if  $A^T = A$  and  $x^T Ax \geq 0$  for all  $x \in \mathbb{R}^n$ .
- (iii) *negative (semi-)definite* if  $-A$  is positive (semi-) definite.

A matrix is positive definite if it is positive semidefinite and in addition

$$x^T Ax = 0 \Rightarrow x = 0 \tag{2.1}$$

The zero-matrix is positive semidefinite, while a positive definite matrix must be nonsingular. Indeed, if  $Ax = 0$  for some  $x \in \mathbb{R}^n$  then  $x^T Ax = 0$  which by (2.1) implies that  $x = 0$ .

**Example 2.1** The  $n$ -by- $n$  tridiagonal matrix

$$J = J_n = \begin{bmatrix} 2 & -1 & 0 & & & \\ -1 & 2 & -1 & & & \\ 0 & \ddots & \ddots & \ddots & & \\ & & & & 0 & \\ & & & -1 & 2 & -1 \\ & & & 0 & -1 & 2 \end{bmatrix} = \text{tridiag}_n(-1, 2, -1) \quad (2.2)$$

is positive definite. Clearly  $J$  is symmetric. Now it can be shown that

$$x^T Jx = x_1^2 + x_n^2 + \sum_{k=1}^{n-1} (x_{k+1} - x_k)^2.$$

Thus  $x^T Jx \geq 0$  and if  $x^T Jx = 0$  then  $x_1 = x_n = 0$  and  $x_k = x_{k+1}$  for  $k = 1, \dots, n-1$  which implies that  $x = 0$ . Hence  $J$  is positive definite.

**Example 2.2** Let  $A = B^T B$ , where  $B \in \mathbb{R}^{m,n}$  and  $m, n$  are positive integers. (Note that  $B$  can be a rectangular matrix). Since  $A^T = (B^T B)^T = B^T B$  we see that  $A$  is symmetric. Moreover, for any  $x \in \mathbb{R}^n$

$$x^T Ax = x^T B^T Bx = (Bx)^T (Bx) = \|Bx\|_2^2. \quad (2.3)$$

Since the Euclidian norm  $\| \cdot \|_2$  of a vector is nonnegative this shows that  $A$  is positive semidefinite and that  $A$  is positive definite if and only if  $B$  has linearly independent columns. Note that  $A$  and  $B$  have the same null-space and hence the same rank. For if  $Bx = 0$  for some vector  $x$  then  $Ax = B^T Bx = 0$  which shows that  $N(B) \subset N(A)$ , while if  $Ax = 0$  then  $x^T Ax = 0$ , and by (2.3) we conclude that  $Bx = 0$ . Thus  $N(A) \subset N(B)$  and we have shown that  $N(A) = N(B)$ .

**Example 2.3** Suppose  $F(t) = F(t_1, \dots, t_n)$  is a real valued function of  $n$  variables which has continuous 1. and 2. order partial derivatives for  $t$  in some domain  $\Omega$ . For each  $t \in \Omega$  the *gradient* and *Hessian* of  $F$  are given by

$$\nabla F(t) = \begin{bmatrix} \frac{\partial F(t)}{\partial t_1} \\ \vdots \\ \frac{\partial F(t)}{\partial t_n} \end{bmatrix} \in \mathbb{R}^n,$$

$$H(t) = \begin{bmatrix} \frac{\partial^2 F(t)}{\partial t_1 \partial t_1} & \cdots & \frac{\partial^2 F(t)}{\partial t_1 \partial t_n} \\ \vdots & & \vdots \\ \frac{\partial^2 F(t)}{\partial t_n \partial t_1} & \cdots & \frac{\partial^2 F(t)}{\partial t_n \partial t_n} \end{bmatrix} \in \mathbb{R}^{n,n}.$$

It is shown in advanced calculus texts that under suitable conditions on the domain  $\Omega$  the matrix  $H(t)$  is symmetric for each  $t \in \Omega$ . Moreover if  $\nabla F(t^*) = 0$  and  $H(t^*)$  is positive definite then  $t^*$  is a local minimum for  $F$ . This can be shown using second-order Taylor approximation of  $F$ . Moreover,  $t^*$  is a local maximum if  $\nabla F(t^*) = 0$  and  $-H(t^*)$  is positive definite.

## 2.2 Triangular Factorization

From Theorem 1.11 it follows that if the leading principal submatrices

$$A_k = \begin{bmatrix} a_{11} & \dots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kk} \end{bmatrix}$$

of a nonsingular symmetric matrix  $A \in \mathbb{R}^{n,n}$  are nonsingular for  $k = 1, \dots, n-1$  then  $A$  has an  $LDL^T$ -factorization of the form  $A = LDL^T$  where  $L$  is unit lower triangular and  $D$  is diagonal with nonzero diagonal elements. The following theorem shows that a positive definite matrix has an  $LDL^T$ -factorization.

**Theorem 2.4** *The leading principal submatrices of a positive definite matrix are positive definite and hence nonsingular.*

**Proof** Consider a leading principal submatrix  $A_k$  of the positive definite matrix  $A \in \mathbb{R}^{n,n}$ . Clearly  $A_k$  is symmetric. Let  $x \in \mathbb{R}^k$  be nonzero, set  $y = (x^T, 0^T)^T \in \mathbb{R}^n$ , and partition  $A$  conformally with  $y$  as  $A = \begin{pmatrix} A_k & B_k \\ C_k & D_k \end{pmatrix}$ , where  $D_k \in \mathbb{R}^{n-k, n-k}$ . Then

$$0 < y^T A y = [x^T \quad 0^T] \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} \begin{bmatrix} x \\ 0 \end{bmatrix} = x^T A_k x.$$

□

For positive definite matrices there is an alternative to the  $LDL^T$ -factorization which is often useful.

**Definition 2.5** A factorization  $A = LL^T$  where  $L$  is lower triangular with positive diagonal entries is called a *Cholesky-factorization*.

**Example 2.6** The matrix  $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$  has an  $LDL^T$ - and Cholesky-factorization given by

$$\begin{aligned} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ -1/2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 3/2 \end{bmatrix} \begin{bmatrix} 1 & -1/2 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{2} & 0 \\ -1/\sqrt{2} & \sqrt{3}/2 \end{bmatrix} \begin{bmatrix} \sqrt{2} & -1/\sqrt{2} \\ 0 & \sqrt{3}/2 \end{bmatrix}. \end{aligned}$$

A matrix is positive definite if and only if it has a Cholesky-factorization.

**Theorem 2.7** *For a matrix  $A \in \mathbb{R}^{n,n}$  the following is equivalent:*

1.  $A$  is positive definite.

2.  $A$  has a factorization  $A = LDL^T$  where  $L$  is unit lower triangular and  $D$  is diagonal with positive diagonal elements.
3.  $A$  has a Cholesky-factorization.
4.  $A = B^T B$  for a nonsingular matrix  $B \in \mathbb{R}^{n,n}$ .

**Proof**  $1 \Rightarrow 2$ : It follows from Theorems 2.4 and 1.11 that  $A$  has an  $LDL^T$ -factorization  $A = LDL^T$ . We need to show that the diagonal entries of  $D$  are positive. With  $e_i$  the  $i$ th unit vector we find

$$d_{ii} = e_i^T D e_i = e_i^T L^{-1} A L^{-T} e_i = x_i^T A x_i,$$

where  $x_i = L^{-T} e_i$  is nonzero since  $L^{-T}$  is nonsingular. Since  $A$  is positive definite we see that  $d_{ii} = x_i^T A x_i > 0$  for  $i = 1, \dots, n$ .

$2 \Rightarrow 3$ : We have  $A = LDL^T = (LD^{1/2})(LD^{1/2})^T$ , and the diagonal entries of  $LD^{1/2}$  are given by  $D^{1/2} = \text{diag}(d_{11}^{1/2}, \dots, d_{nn}^{1/2})$ . Thus  $LD^{1/2}$  is lower triangular with positive diagonal elements and  $A = (LD^{1/2})(LD^{1/2})^T$  is a Cholesky-factorization of  $A$ .

$3 \Rightarrow 4$ : Take  $B^T = L$ .

$4 \Rightarrow 1$ : This follows from the discussion in Example 2.2.  $\square$

The Cholesky-factorization is unique since a positive definite matrix  $A$  has a unique  $LDL^T$ -factorization and this is equivalent to  $A$  having a unique Cholesky-factorization.

Consider next an algorithm for computing the Cholesky factorization of a matrix  $A$ . Since  $A = LL^T$  and  $L$  is lower triangular we find

$$a_{ik} = \sum_{j=1}^n l_{ij} l_{kj} = \sum_{j=1}^{\min(i,k)} l_{ij} l_{kj}, \quad i, k = 1, \dots, n. \quad (2.4)$$

We can compute  $L$  column by column or row by row. Consider a column oriented algorithm. Suppose we have computed the  $k - 1$  first columns of  $L$ . The  $k$ th column can then be computed from (2.4). Indeed, letting  $i = k$  and solving for  $l_{kk}$  we find

$$l_{kk} = \left( a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2 \right)^{1/2}, \quad (2.5)$$

and similarly for  $i > k$

$$l_{ik} = \left( a_{ik} - \sum_{j=1}^{k-1} l_{ij} l_{kj} \right) / l_{kk} \quad i = k + 1, \dots, n. \quad (2.6)$$

Consider estimating the number of flops needed to compute the Cholesky-factorization. In addition to  $n$  square roots we find

$$\sum_{k=1}^n (2k - 2 + (2k - 1)(n - k)) \approx \sum_{k=0}^n 2k(n - k) \approx \int_0^n 2x(n - x)dx = n^3/3$$

flops. This is the same number as would be needed for a symmetric version of Gaussian elimination.

In many applications the matrix  $A$  has a banded structure, and the number of flops can be reduced. We say that  $A$  has *lower bandwidth*  $p$  if  $a_{ij} = 0$  whenever  $i > j + p$ , and *upper bandwidth*  $q$  if  $a_{ij} = 0$  whenever  $j > i + q$ . A diagonal matrix has upper and lower bandwidth zero, while a matrix with upper and lower bandwidth one is tridiagonal. Moreover, if  $A$  is symmetric then  $p = q$ .

Consider now an algorithm for computing the Cholesky-factorization of a band-matrix. We first show that if  $A = LL^T$  then  $L$  has the same lower bandwidth as  $A$ .

**Lemma 2.8** *Suppose  $A$  is positive definite with Cholesky-factorization  $A = LL^T$ . If  $a_{ik} = 0$  for  $i > k + d$ , then also  $l_{ik} = 0$  for  $i > k + d$ .*

**Proof** We show that if  $L$  has lower bandwidth  $d$  in its first  $k - 1$  columns then column  $k$  also has lower bandwidth  $d$ . The proof then follows by induction on  $k$ . Now, if  $i > k + d$ , then  $a_{ik} = 0$ , and if  $L$  has lower bandwidth  $d$  in its first  $k - 1$  columns then  $l_{ij} = 0$  for  $j = 1, \dots, k - 1$ . By (2.6)  $l_{ik} = 0$ .  $\square$

Based on (2.5), (2.6) and Lemma 2.8 we obtain the following algorithm to compute the Cholesky factorization of a positive definite band-matrix  $A \in \mathbb{R}^{n,n}$  with  $a_{ij} = 0$  for  $i > j + d$ .

**Algorithm 2.9** *For  $k = 1, \dots, n$*

$$l_{kk} = \left( a_{kk} - \sum_{j=\max(1, k-d)}^{k-1} l_{kj}^2 \right)^{1/2}$$

$$l_{ik} = \left( a_{ik} - \sum_{j=\max(1, i-d)}^{k-1} l_{ij}l_{kj} \right) / l_{kk}, \quad i = k + 1, \dots, \min(k + d, n).$$

To solve  $Ax = b$  where  $A \in \mathbb{R}^{n,n}$  is positive definite with lower bandwidth  $d$  we can use Algorithm 2.9 followed by

**Algorithm 2.10**

1.  $y_i = (b_i - \sum_{j=\max(1, i-d)}^{i-1} l_{ij}y_j)/l_{ii}, \quad i = 1, \dots, n$
2.  $x_i = (y_i - \sum_{j=i+1}^{\min(n, i+d)} l_{ji}x_j)/l_{ii}, \quad i = n, n-1, \dots, 1$

The number of flops for these algorithms is  $O(2nd^2)$  for Algorithm 2.9 and  $O(4nd)$  for Algorithm 2.10. When  $d$  is small compared to  $n$  we see that these numbers are considerably smaller than the  $O(n^3/3)$  and  $O(2n^2)$  counts for the factorization of a full matrix.

There is also a banded version of the  $LDL^T$ -factorization which requires approximately the same number of flops as the Cholesky-factorization. The choice between using an  $LDL^T$ -factorization or an  $LL^T$ -factorization depends on several factors. Usually an  $LU$  or an  $LDL^T$ -factorization is preferred for matrices with small bandwidth (tridiagonal, pentadiagonal), while the  $LL^T$ -factorization is often used when the bandwidth is larger.

### 2.3 When is a Matrix Positive Definite?

Not all symmetric matrices are positive definite, and sometimes we can tell just by glancing at the matrix that it cannot be positive definite.

For example, if  $a_{ii} \leq 0$  for some  $i$  then  $e_i^T A e_i = a_{ii} \leq 0$  and  $A$  is not positive definite. Similarly, if the largest element of  $A$  is not on the diagonal then  $A$  is not positive definite. To show this suppose  $a_{ij} \geq a_{ii}$  and  $a_{ij} \geq a_{jj}$  for some  $i \neq j$ . Since  $A$  is symmetric we obtain

$$(e_i - e_j)^T A (e_i - e_j) = a_{ii} + a_{jj} - 2a_{ij} \leq 0$$

which implies that  $x^T A x \leq 0$  for some  $x \neq 0$ .

A positive definite matrix has positive eigenvalues.

**Lemma 2.11** *A matrix is positive definite if and only if it is symmetric and has positive eigenvalues.*

**Proof** If  $A$  is positive definite then by definition  $A$  is symmetric. Suppose  $Ax = \lambda x$  with  $x \neq 0$ . Multiplying both sides by  $x^T$  and solving for  $\lambda$  we find

$$\lambda = \frac{x^T A x}{x^T x} > 0.$$

Suppose conversely that  $A \in \mathbb{R}^{n,n}$  is symmetric with positive eigenvalues  $\lambda_1, \dots, \lambda_n$ . By Corollary 6.4.3 in Leon we have  $U^T A U = D$ , where  $U^T U = U U^T = I$  and  $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ . Let  $x \in \mathbb{R}^n$  be nonzero and define

$c := U^T x = (c_1, \dots, c_n)^T$ . Then  $c^T c = x^T U U^T x = x^T x$  so  $c$  is nonzero. Since  $x = Uc$  we find

$$x^T Ax = (Uc)^T AUc = c^T U^T AUc = c^T Dc = \sum_{j=1}^n \lambda_j c_j^2 > 0$$

and it follows that  $A$  is positive definite.  $\square$

In the following theorem we give three necessary and sufficient conditions for positive definiteness of a matrix.

**Theorem 2.12** *The following is equivalent for a symmetric matrix  $A \in \mathbb{R}^{n,n}$*

1.  $A$  is positive definite.
2.  $A$  has only positive eigenvalues.
- 3.

$$\det \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix} > 0 \text{ for } k = 1, \dots, n$$

4.  $A = B^T B$  for a nonsingular  $B \in \mathbb{R}^{n,n}$

**Proof** By Lemma 2.11 we know that  $1 \Rightarrow 2$ . We show that  $1 \Rightarrow 3 \Rightarrow 4 \Rightarrow 1$ .  $1 \Rightarrow 3$ : By Theorem 2.4 the leading principal submatrix  $A_k$  of  $A$  is positive definite, and hence has positive eigenvalues by Lemma 2.11. Since the determinant of a matrix equals the product of its eigenvalues we conclude that  $\det(A_k) > 0$  for  $k = 1, \dots, n$ .

$3 \Rightarrow 4$ : The condition  $\det(A_k) > 0$  implies that  $A_k$  is nonsingular for  $k = 1, \dots, n$ . By Theorem 1.11 in the lecture notes (and the discussion before)  $A$  has a unique  $LDL^T$ -factorization. Let  $L_k$  and  $D_k$  be the leading principal submatrices of order  $k$  of  $L$  and  $D$ , respectively. By partitioning  $A$ ,  $L$  and  $D$  similarly to the proof of Lemma 1.8 in the notes we see that  $A_k = L_k D_k L_k^T$  is the  $LDL^T$ -factorization of  $A_k$  for  $k = 1, \dots, n$ . Using properties of determinants we find  $\det(A_k) = \det(L_k) \det(D_k) \det(L_k^T) = \det(D_k) = d_{11} \dots d_{kk} > 0$ . Since this holds for  $k = 1, \dots, n$  we conclude that  $D$  has positive diagonal elements. We have shown that  $A$  has an  $LDL^T$ -factorization with positive diagonal elements in  $D$ . The result now follows from Theorem 2.7.

$4 \Rightarrow 1$ : This was already shown in Theorem 2.7.  $\square$





## Chapter 3

# Some Model Problems

When testing and comparing numerical algorithms it is convenient to have a problem at hand to which the different algorithms can be applied. In this section we consider several linear set of equations which we will use for testing algorithms for large linear systems.

### 3.1 A Tridiagonal Matrix

Consider the simple two point boundary value problem

$$\begin{aligned} -u''(x) &= f(x), \quad x \in [0, 1], \\ u(0) &= 0, \quad u(1) = 0, \end{aligned} \tag{3.1}$$

where  $f$  is a given continuous function on  $[0, 1]$ . In principle it is easy to solve (3.1) exactly. We just integrate  $f$  twice and determine the two integration constants so that the homogeneous boundary conditions  $u(0) = u(1) = 0$  are satisfied. For example, if  $f(x) = 1$  then  $u(x) = x(x - 1)/2$  is the solution.

We will solve (3.1) approximately using the *finite difference method*. Normally we would not use this method to solve a problem as simple as (3.1), but this method is applicable to more complicated problems. Also it could be difficult to solve (3.1) exactly if  $f$  cannot be integrated in closed form.

In the finite difference method we choose a positive integer  $m$ , define  $h := 1/(m + 1)$  and replace the interval  $[0, 1]$  by grid points  $x_j := jh$  for  $j = 0, 1, \dots, m + 1$ . We then use the following finite difference approximation to the second derivative:

$$u''(x) \approx \frac{u(x + h) - 2u(x) + u(x - h)}{h^2}.$$

We obtain approximations  $v_j$  to the exact solution  $u(jh)$  for  $j = 1, \dots, m$  by replacing the differential equation by the difference equation

$$\frac{-v_{j-1} + 2v_j - v_{j+1}}{h^2} = f(jh), \quad j = 1, \dots, m,$$



5. The LU factorization of  $J$  is given by  $J = LU$ , where

$$L = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -\frac{1}{2} & 1 & \ddots & & \vdots \\ 0 & -\frac{2}{3} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\frac{m-1}{m} & 1 \end{pmatrix}, U = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ 0 & \frac{3}{2} & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \frac{m}{m-1} & -1 \\ 0 & \cdots & \cdots & 0 & \frac{m+1}{m} \end{pmatrix}$$

6. The inverse  $J^{-1} = A = (a_{i,j})_{i,j=1}^m$  of  $J$  is nonnegative with entries given by  $a_{i,j} = a_{j,i} = (1 - ih) * j$  for  $i \geq j$ .

**Proof** We prove 1. and 2. and (3.8). We leave the rest as problems. Let  $s_j$  be given by (3.4) and consider for  $1 \leq k \leq m$  the  $k$ th components  $(Cs_j)_k$  and  $\sin(kj\pi h)$  of the vectors  $Cs_j$  and  $s_j$ . Since  $c_{k,k-1} = c_{k,k+1} = a$  and  $c_{k,k} = b$  we find

$$\begin{aligned} (Cs_j)_k &= \sum_{l=1}^m c_{k,l} \sin(lj\pi h) \\ &= a \sin((k-1)j\pi h) + b \sin(kj\pi h) + a \sin((k+1)j\pi h). \end{aligned}$$

Using the trigonometric identity  $\sin(A+B) + \sin(A-B) = 2 \sin A \cos B$  we find

$$(Cs_j)_k = (b + 2a \cos(j\pi h)) \sin(kj\pi h) = \lambda_j (s_j)_k.$$

This calculation holds for  $k = 1, \dots, m$  and (3.4) and (3.5) follow. Since  $j\pi h = j\pi/(m+1) \in (0, \pi)$  for  $j = 1, \dots, m$  and the  $\cos$  function is strictly monotone decreasing on  $(0, \pi)$  the eigenvalues are distinct and since  $C$  is symmetric the eigenvectors  $s_j$  are orthogonal (Cf. Leon Theorem 6.4.1). To finish the proof of (3.6) we compute the Euclidian norm of each  $s_j$  as follows:

$$\begin{aligned} s_j^T s_j &= \sum_{k=1}^m \sin^2(kj\pi h) = \sum_{k=0}^m \sin^2(kj\pi h) = \frac{1}{2} \sum_{k=0}^m (1 - \cos(2kj\pi h)) \\ &= \frac{m+1}{2} - \frac{1}{2} \sum_{k=0}^m \cos(2kj\pi h) = \frac{m+1}{2}, \end{aligned}$$

since the last cosine sum is zero. We show this by summing a geometric series of complex exponentials. With  $i = \sqrt{-1}$  we find

$$\sum_{k=0}^m e^{2ikj\pi h} = \frac{e^{2i(m+1)j\pi h} - 1}{e^{2ij\pi h} - 1} = \frac{e^{2\pi ij} - 1}{e^{2ij\pi h} - 1} = 0.$$

Since

$$\sum_{k=0}^m e^{2ikj\pi h} = \sum_{k=0}^m \cos(2kj\pi h) + i \sum_{k=0}^m \sin(2kj\pi h)$$

the latter sine and cosine sums are zero and (3.6) follows.

To show (3.8) we recall that  $\text{cond}_2(J) = \lambda_{max}/\lambda_{min}$ , the ratio of the largest and smallest eigenvalues of  $J$ . This follows since  $J$  is positive definite. Thus

$$\text{cond}_2(A) = \frac{2 + 2 \cos(\pi h)}{2 - 2 \cos(\pi h)} = \frac{\cos^2(\pi h/2)}{\sin^2(\pi h/2)} = \cot^2(\pi h/2).$$

Since  $\tan x > x$  for  $0 < x < \pi/2$  we have  $\cot^2 x < \frac{1}{x^2}$  for  $0 < x < \pi/2$  and the upper bound follows. The lower bound can be derived from a Taylor expansion of  $\frac{\cos^2 x}{\sin^2 x}$  which shows that  $\cot^2 x > \frac{1}{x^2} - \frac{2}{3}$ . for  $x > 0$ . □

### 3.2 The Poisson Problem

Consider the problem

$$-\Delta u = -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f \text{ on } \Omega = (0, 1)^2 \quad (3.9)$$

$$u = 0 \text{ on } \partial\Omega.$$

Here  $\Omega$  is the open unit square while  $\partial\Omega$  is the boundary of  $\Omega$ . The function  $f$  is given and continuous on  $\Omega$  and we seek a function  $u = u(x, y)$  such that (3.9) holds and which is zero on  $\partial\Omega$ .

Let  $m$  be a positive integer. We solve the problem numerically by replacing derivatives by difference quotients on a grid of points given by

$$\{(jh, kh) : j, k = 0, 1, \dots, m+1\}, \quad \text{where } h = 1/(m+1).$$

The points  $\{(jh, kh) : j, k = 1, \dots, m\}$  are the interior points, while the remaining points are the boundary points. The solution is zero at the boundary points. For an interior point we insert the approximations

$$\begin{aligned} -\frac{\partial^2 u(jh, kh)}{\partial x^2} &\approx \frac{-v_{j-1,k} + 2v_{j,k} - v_{j+1,k}}{h^2} \\ -\frac{\partial^2 u(x_j, y_k)}{\partial y^2} &\approx \frac{-v_{j,k-1} + 2v_{j,k} - v_{j,k+1}}{h^2} \end{aligned}$$

in (3.9) and multiply both sides by  $h^2$  to obtain

$$(-v_{j-1,k} + 2v_{j,k} - v_{j+1,k}) + (-v_{j,k-1} + 2v_{j,k} - v_{j,k+1}) = h^2 f_{j,k} \quad (3.10)$$

or

$$4v_{j,k} - v_{j-1,k} - v_{j+1,k} - v_{j,k-1} - v_{j,k+1} = h^2 f_{j,k} =: h^2 f(jh, kh). \quad (3.11)$$

From the boundary conditions we have in addition

$$v_{0,k} = v_{m+1,k} = v_{j,0} = v_{j,m+1} = 0, \quad j, k = 0, 1, \dots, m+1. \quad (3.12)$$

The equations (3.11) and (3.12) define a linear set of equations for the unknowns  $V = [v_{jk}] \in \mathbb{R}^{m,m}$ .

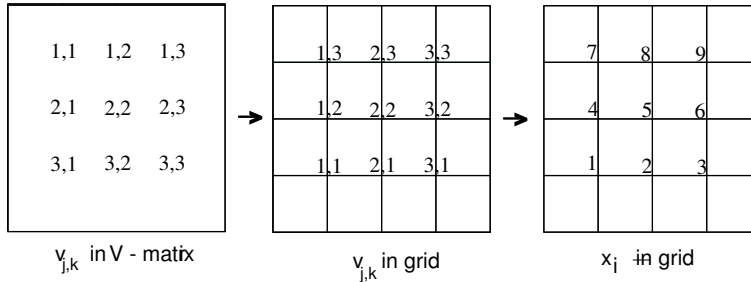
To write (3.11) and (3.12) in standard form  $Ax = b$  we need to order the unknown  $v_{j,k}$  in some way. We do this by defining the following operation of *vectorization* of a matrix.

**Definition 3.2** For any  $B \in \mathbb{R}^{m,n}$  we define the vector

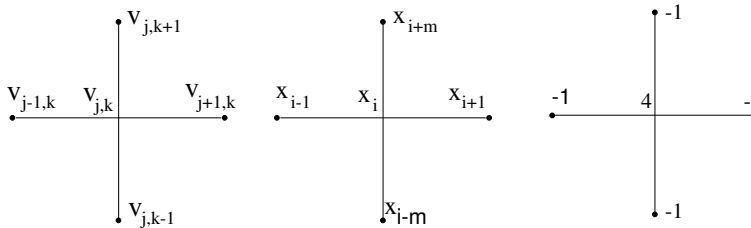
$$\text{vec}(B) := (b_{11}, \dots, b_{m1}, b_{12}, \dots, b_{m2}, \dots, b_{1n}, \dots, b_{mn})^T \in \mathbb{R}^{mn}$$

by stacking the columns of  $B$  on top of each other.

Let  $n = m^2$  and  $x := \text{vec}(V) \in \mathbb{R}^n$ . Note that forming  $x$  by stacking the columns of  $V$  on top of each other means the following ordering of the grid points. For  $m = 3$  this can be illustrated as follows:



The location of the entries in (3.11) form a 5-point stencil:



To find the matrix  $A$  we note that for values of  $j, k$  where the 5-point stencil does not touch the boundary (3.11) takes the form

$$4x_i - x_{i-1} - x_{i+1} - x_{i-m} - x_{i+m} = b_i,$$

where  $b_i = h^2 f_{jk}$ . This must be modified close to the boundary.



in block form as

$$C = \begin{bmatrix} Ab_{1,1} & Ab_{1,2} & \cdots & Ab_{1,s} \\ Ab_{2,1} & Ab_{2,2} & \cdots & Ab_{2,s} \\ \vdots & \vdots & & \vdots \\ Ab_{r,1} & Ab_{r,2} & \cdots & Ab_{r,s} \end{bmatrix}.$$

We denote the Kronecker product of  $A$  and  $B$  by  $C = A \otimes B$ .

This definition of the Kronecker product is known more precisely as the *left Kronecker product*. In the literature one also finds the *right Kronecker product* which in our notation is given by  $B \otimes A$ .

As examples of Kronecker products which are relevant for our discussion, if

$$J = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \quad \text{and} \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

then

$$J \otimes I = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & 0 & 0 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \quad \text{and} \quad I \otimes J = \begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & 2 & 0 & -1 \\ -1 & 0 & 2 & 0 \\ 0 & -1 & 0 & 2 \end{bmatrix}.$$

Also note that the Kronecker product  $u \otimes v = [u^T v_1, \dots, u^T v_r]^T$  of two vectors  $u \in \mathbb{R}^p$  and  $v \in \mathbb{R}^r$  is a vector of length  $p \times r$ .

The matrix  $A$  in (3.14) can be written as a sum of two Kronecker products. We see that

$$A = \begin{bmatrix} J & & & & \\ & J & & & \\ & & \ddots & & \\ & & & J & \\ & & & & J \end{bmatrix} + \begin{bmatrix} 2I & -I & & & \\ -I & 2I & -I & & \\ & & \ddots & \ddots & \ddots \\ & & & -I & 2I & -I \\ & & & & -I & 2I \end{bmatrix} = J \otimes I + I \otimes J.$$

**Definition 3.4** Let for positive integers  $r, s, k$ ,  $A \in \mathbb{R}^{r,r}$ ,  $B \in \mathbb{R}^{s,s}$  and  $I_k$  be the identity matrix of order  $k$ . The sum  $A \otimes I_s + I_r \otimes B$  is known as the Kronecker sum of  $A$  and  $B$ .

The following simple arithmetic rules hold for Kronecker products. For scalars  $\lambda, \mu$  and matrices  $A, A_1, A_2, B, B_1, B_2$  of dimensions such that the operations are defined we have

$$\begin{aligned} (\lambda A) \otimes (\mu B) &= \lambda \mu (A \otimes B), \\ (A_1 + A_2) \otimes B &= A_1 \otimes B + A_2 \otimes B, \\ A \otimes (B_1 + B_2) &= A \otimes B_1 + A \otimes B_2, \\ (A \otimes B)^T &= A^T \otimes B^T. \end{aligned} \tag{3.15}$$

Note however that in general we have  $A \otimes B \neq B \otimes A$ . The proofs of (3.15) are left as exercises. For showing further properties of Kronecker products and sums the following *mixed product rule* is an essential tool.

**Lemma 3.5** *Suppose  $A, B, C, D$  are rectangular matrices with dimensions so that the products  $AC$  and  $BD$  are defined. Then the product  $(A \otimes B)(C \otimes D)$  is defined and*

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD). \quad (3.16)$$

**Proof** If  $B \in \mathbb{R}^{r,t}$  and  $D \in \mathbb{R}^{t,s}$  for some integers  $r, t$  then

$$(AC) \otimes (BD) = \begin{bmatrix} (AC)(BD)_{1,1} & \cdots & (AC)(BD)_{1,s} \\ \vdots & & \vdots \\ (AC)(BD)_{r,1} & \cdots & (AC)(BD)_{r,s} \end{bmatrix}.$$

The left hand side of (3.16) is

$$\begin{bmatrix} Ab_{1,1} & \cdots & Ab_{1,t} \\ \vdots & & \vdots \\ Ab_{r,1} & \cdots & Ab_{r,t} \end{bmatrix} \begin{bmatrix} Cd_{1,1} & \cdots & Cd_{1,s} \\ \vdots & & \vdots \\ Cd_{t,1} & \cdots & Cd_{t,s} \end{bmatrix} = \begin{bmatrix} E_{1,1} & \cdots & E_{1,s} \\ \vdots & & \vdots \\ E_{r,1} & \cdots & E_{r,s} \end{bmatrix},$$

where for all  $i, j$

$$E_{i,j} = \sum_{k=1}^t b_{i,k} d_{k,j} AC = (AC)(BD)_{i,j} = ((AC) \otimes (BD))_{i,j}.$$

□

**Lemma 3.6** *Suppose  $A$  and  $B$  are square matrices. Then the eigenvalues of  $A \otimes B$  are products of eigenvalues of  $A$  and  $B$ , and the eigenvectors of  $A \otimes B$  are kronecker products of eigenvectors of  $A$  and  $B$ . More precisely, if  $A \in \mathbb{R}^{r,r}$  and  $B \in \mathbb{R}^{s,s}$  and*

$$Au_i = \lambda_i u_i, \quad i = 1, \dots, r, \quad Bv_j = \mu_j v_j, \quad j = 1, \dots, s,$$

then

$$(A \otimes B)(u_i \otimes v_j) = \lambda_i \mu_j (u_i \otimes v_j), \quad i = 1, \dots, r, \quad j = 1, \dots, s. \quad (3.17)$$

**Proof** Using (3.16) the proof is a one liner. For all  $i, j$

$$(A \otimes B)(u_i \otimes v_j) = (Au_i) \otimes (Bv_j) = (\lambda_i u_i) \otimes (\mu_j v_j) = (\lambda_i \mu_j)(u_i \otimes v_j).$$

□

Consider next a Kronecker sum



**Lemma 3.7** For positive integers  $r, s$  let  $A \in \mathbb{R}^{r,r}$   $B \in \mathbb{R}^{s,s}$ . Then the eigenvalues of the Kronecker sum  $A \otimes I_s + I_r \otimes B$  are all sums of eigenvalues of  $A$  and  $B$ , and the eigenvectors of  $A \otimes I_s + I_r \otimes B$  are all Kronecker products of eigenvectors of  $A$  and  $B$ . More precisely, if

$$Au_i = \lambda_i u_i, \quad i = 1, \dots, r, \quad Bv_j = \mu_j v_j, \quad j = 1, \dots, s,$$

then

$$(A \otimes I_s + I_r \otimes B)(u_i \otimes v_j) = (\lambda_i + \mu_j)(u_i \otimes v_j), \quad i = 1, \dots, r, \quad j = 1, \dots, s. \quad (3.18)$$

**Proof** Since  $I_s v_j = v_j$  for  $j = 1, \dots, s$  and  $I_r u_i = u_i$  for  $i = 1, \dots, r$  we obtain by Lemma 3.6 for all  $i, j$

$$(A \otimes I_s)(u_i \otimes v_j) = \lambda_i(u_i \otimes v_j), \quad \text{and} \quad (I_r \otimes B)(u_i \otimes v_j) = \mu_j(u_i \otimes v_j).$$

The result now follows by summing these relations.  $\square$

In many cases the Kronecker product and sum inherits properties of its factors.

**Lemma 3.8**

1. If  $A$  and  $B$  are nonsingular then  $A \otimes B$  and  $A \otimes I + I \otimes B$  are nonsingular. Moreover  $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$ .
2. If  $A$  and  $B$  are symmetric then  $A \otimes B$  and  $A \otimes I + I \otimes B$  are symmetric.
3. If one of  $A, B$  is positive definite and the other is positive semi-definite then  $A \otimes I + I \otimes B$  is positive definite.

**Proof** Suppose in the proof that  $A \in \mathbb{R}^{r,r}$  and  $B \in \mathbb{R}^{s,s}$ . 1. follows from the mixed product rule giving

$$(A \otimes B)(A^{-1} \otimes B^{-1}) = (AA^{-1}) \otimes (BB^{-1}) = I_r \otimes I_s = I_{rs}.$$

Thus  $(A \otimes B)$  is nonsingular with the indicated inverse. 2. and the symmetry part of 3. follow immediately from (3.15). Suppose  $A$  is positive definite and  $B$  is positive semidefinite. Then  $A$  has positive eigenvalues and  $B$  has nonnegative eigenvalues. By Lemma 3.7 the eigenvalues of  $A \otimes I + I \otimes B$  are all positive and 3. follows.  $\square$

We can apply these results to the Kronecker sum matrix

$$A = C_1 \otimes I + I \otimes C_2 = \begin{bmatrix} C_1 & & & & \\ & C_1 & & & \\ & & \ddots & & \\ & & & C_1 & \\ & & & & C_1 \end{bmatrix} + \begin{bmatrix} 0 & bI & & & \\ bI & 0 & bI & & \\ & & \ddots & \ddots & \ddots \\ & & & bI & 0 & bI \\ & & & & bI & 0 \end{bmatrix},$$

where for  $a, b, c \in \mathbb{R}$  we have  $C_1 = \text{tridiag}_m(a, c, a)$  and  $C_2 = \text{tridiag}_m(b, 0, b)$ . With  $n = m^2$  the entries of  $A$  are given by

$$\begin{aligned} a_{i,i+1} = a_{i+1,i} &= a, & i = 1, \dots, n-1, & \quad i \neq m, 2m, \dots, (m-1)m, \\ a_{i,i+m} = a_{i+m,i} &= b, & i = 1, \dots, n-m, \\ a_{i,i} &= c, & i = 1, \dots, n, \\ a_{i,j} &= 0, & \text{otherwise.} \end{aligned} \tag{3.19}$$

With  $b = a = -1$  and  $c = 4$ , we obtain the Poisson matrix given by (3.14). With  $b = a = 1/9$  and  $c = 5/9$ , we obtain a matrix which we refer to as the *averaging matrix*. We will need the  $l_2$  condition numbers of these matrices.

**Lemma 3.9** *For fixed  $m$  let  $A$  be the matrix given by (3.19) and let  $h = 1/(m+1)$ .*

1. We have  $Ax_{j,k} = \lambda_{j,k}x_{j,k}$  for  $j, k = 1, \dots, m$ , where

$$x_{j,k} = s_j \otimes s_k, \tag{3.20}$$

$$s_j = (\sin(j\pi h), \sin(2j\pi h), \dots, \sin(mj\pi h))^T, \tag{3.21}$$

$$\lambda_{j,k} = c + 2a \cos(j\pi h) + 2b \cos(k\pi h). \tag{3.22}$$

2. The eigenvectors are orthogonal

$$x_{j,k}^T x_{p,q} = \frac{1}{4h^2} \delta_{j,p} \delta_{k,q}, \quad j, k, p, q = 1, \dots, m. \tag{3.23}$$

3.  $A$  is positive definite if  $c > 0$  and  $c \geq 2|a| + 2|b|$ .
4. For the Poisson case where  $a = b = -1$  and  $c = 4$  the matrix  $A$  is positive definite and the  $l_2$  condition number is bounded below and above by

$$\frac{4}{\pi^2}(m+1)^2 - \frac{2}{3} < \text{cond}_2(A) := \|A\|_2 \|A^{-1}\|_2 < \frac{4}{\pi^2}(m+1)^2.$$

5. For the averaging matrix where  $a = b = 1/9$  and  $c = 5/9$  the matrix  $A$  is positive definite and the  $l_2$  condition number is bounded independently of  $h$ . In fact we have  $\text{cond}_2(A) \leq 9$ .

**Proof** By Lemma 3.1 the eigenvalues and eigenvectors of the matrices  $C_1 = \text{tridiag}(a, c, a)$  and  $C_2 = \text{tridiag}(b, 0, b)$  are given by

$$\begin{aligned} C_1 s_i &= (c + 2a \cos(i\pi h)) s_i, & i = 1, \dots, m, \\ C_2 s_j &= 2b \cos(j\pi h) s_j, & j = 1, \dots, m. \end{aligned}$$

1. now follows from Lemma 3.7. Using the transpose rule, the mixed product rule and (3.6) we find for  $j, k, p, q = 1, \dots, m$

$$(s_j \otimes s_k)^T (s_p \otimes s_q) = (s_j^T \otimes s_k^T) (s_p \otimes s_q) = (s_j^T s_p) \otimes (s_k^T s_q) = \frac{1}{4h^2} \delta_{j,p} \delta_{k,q}$$

and 2. follows. Since  $A$  is symmetric 3. will follow if the eigenvalues are positive. But this is true under the stated conditions. The condition in 3. is satisfied both for  $a = b = -1, c = 4$  and  $a = b = 1/5, c = 5/9$ . Thus the matrices in 4. and 5. are positive definite. From (3.22) the eigenvalues of the Poisson matrix are given by

$$\lambda_{j,k} = 4 - 2 \cos(j\pi h) - 2 \cos(k\pi h), \quad j, k = 1, \dots, m.$$

Thus

$$\text{cond}_2(A) = \frac{\max \lambda_{j,k}}{\min \lambda_{j,k}} = \frac{\lambda_{m,m}}{\lambda_{1,1}} = \frac{4 + 4 \cos(\pi h)}{4 - 4 \cos(\pi h)} = \frac{\cos^2(\frac{\pi h}{2})}{\sin^2(\frac{\pi h}{2})}.$$

Since  $\tan x > x$  for  $0 < x < \pi/2$  we have  $\frac{\cos^2 x}{\sin^2 x} < \frac{1}{x^2}$  for  $0 < x < \pi/2$  and the upper bound follows. The lower bound can be derived from a Taylor expansion of  $\frac{\cos^2 x}{\sin^2 x}$  which shows that  $\frac{\cos^2 x}{\sin^2 x} > \frac{1}{x^2} - \frac{2}{3}$ . for  $x > 0$ . This proves 4. Finally for 5. the eigenvalues of  $A$  are given by

$$\lambda_{j,k} = \frac{5}{9} + \frac{2}{9} \cos(j\pi h) + \frac{2}{9} \cos(k\pi h), \quad j, k = 1, \dots, m.$$

In this case

$$\text{cond}_2(A) = \frac{\lambda_{1,1}}{\lambda_{m,m}} = \frac{\frac{5}{9} + \frac{4}{9} \cos(\pi h)}{\frac{5}{9} - \frac{4}{9} \cos(\pi h)} \leq 9.$$

□

### 3.4 A banded Matrix

Consider the problem

$$\begin{aligned} -\frac{\partial}{\partial x} (c(x, y) \frac{\partial u}{\partial x}) - \frac{\partial}{\partial y} (c(x, y) \frac{\partial u}{\partial y}) &= f(x, y) & (x, y) \in \Omega = (0, 1)^2 \\ u(x, y) &= 0 & (x, y) \in \partial\Omega. \end{aligned} \tag{3.24}$$

Here  $\Omega$  is the open unit square while  $\partial\Omega$  is the boundary of  $\Omega$ . The functions  $f$  and  $c$  are given and we seek a function  $u = u(x, y)$  such that (3.24) holds. We assume that  $c$  and  $f$  are defined and continuous on  $\Omega$  and that  $c(x, y) > 0$  for all  $(x, y) \in \Omega$ . The problem (3.24) reduces to the Poisson problem in the special case where  $c(x, y) = 1$  for  $(x, y) \in \Omega$ .

As for the Poisson problem we solve (3.24) numerically on a grid of points

$$\{(jh, kh) : j, k = 0, 1, \dots, m+1\}, \quad \text{where } h = 1/(m+1),$$

and where  $m$  is a positive integer. Let  $(x, y)$  be one of the interior grid points. We use the finite difference approximations

$$\begin{aligned}\frac{\partial}{\partial t} \left( f(t) \frac{\partial}{\partial t} g(t) \right) &\approx \left( f(t + \frac{h}{2}) \frac{\partial}{\partial t} g(t + h/2) - f(t - \frac{h}{2}) \frac{\partial}{\partial t} g(t - \frac{h}{2}) \right) / h \\ &\approx \left( f(t + \frac{h}{2}) (g(t + h) - g(t)) - f(t - \frac{h}{2}) (g(t) - g(t - h)) \right) / h^2\end{aligned}$$

to obtain

$$\frac{\partial}{\partial x} \left( c \frac{\partial u}{\partial x} \right)_{j,k} \approx \frac{c_{j+\frac{1}{2},k} (v_{j+1,k} - v_{j,k}) - c_{j-\frac{1}{2},k} (v_{j,k} - v_{j-1,k})}{h^2}$$

and

$$\frac{\partial}{\partial y} \left( c \frac{\partial u}{\partial y} \right)_{j,k} \approx \frac{c_{j,k+\frac{1}{2}} (v_{j,k+1} - v_{j,k}) - c_{j,k-\frac{1}{2}} (v_{j,k} - v_{j,k-1})}{h^2},$$

where  $c_{p,q} = c(ph, qh)$  and  $v_{j,k} \approx u(jh, kh)$ . With these approximations the discrete analog of (3.24) turns out to be

$$\begin{aligned}-(P_h v)_{j,k} &= h^2 f_{j,k} \quad j, k = 1, \dots, m \\ v_{j,k} &= 0 \quad j = 0, m+1 \text{ all } k \text{ or } k = 0, m+1 \text{ all } j,\end{aligned} \quad (3.25)$$

where

$$\begin{aligned}-(P_h v)_{j,k} &= (c_{j,k-\frac{1}{2}} + c_{j-\frac{1}{2},k} + c_{j+\frac{1}{2},k} + c_{j,k+\frac{1}{2}}) v_{j,k} \\ &\quad - c_{j,k-\frac{1}{2}} v_{j,k-1} - c_{j-\frac{1}{2},k} v_{j-1,k} - c_{j+\frac{1}{2},k} v_{j+1,k} - c_{j,k+\frac{1}{2}} v_{j,k+1}\end{aligned} \quad (3.26)$$

and  $f_{j,k} = f(jh, kh)$ .

As before we let  $V = (v_{j,k}) \in \mathbb{R}^{m,m}$  and  $F = (f_{j,k}) \in \mathbb{R}^{m,m}$ . The corresponding linear system can be written  $Ax = b$  where  $x = \text{vec}(V)$ ,  $b = h^2 \text{vec}(F)$ , and the  $n$ -by- $n$  coefficient matrix  $A$  is given by

$$\begin{aligned}a_{i,i} &= c_{j_i, k_i - \frac{1}{2}} + c_{j_i - \frac{1}{2}, k_i} + c_{j_i + \frac{1}{2}, k_i} + c_{j_i, k_i + \frac{1}{2}}, & i = 1, 2, \dots, n \\ a_{i+1,i} &= a_{i,i+1} = -c_{j_i + \frac{1}{2}, k_i}, & i \bmod m \neq 0 \\ a_{i+m,i} &= a_{i,i+m} = -c_{j_i, k_i + \frac{1}{2}}, & i = 1, 2, \dots, n - m \\ a_{i,j} &= 0 & \text{otherwise,}\end{aligned} \quad (3.27)$$

where  $(j_i, k_i)$  with  $1 \leq j_i, k_i \leq m$  is determined uniquely from the equation  $i = j_i + (k_i - 1)m$  for  $i = 1, \dots, n$ . When  $c(x, y) = 1$  for all  $(x, y) \in \Omega$  then all  $\gamma$ 's in (3.27) are equal to one and we recover the Poisson matrix.

In general we cannot write  $A$  as a Kronecker sum. But we can show that  $A$  is positive definite as long as the function  $c$  is positive on  $\Omega$ . Recall that a matrix  $A$  is positive definite if it is symmetric and  $x^T A x > 0$  for all  $x \neq 0$ .

**Theorem 3.1** *If  $c(x, y) > 0$  for  $(x, y) \in \Omega$  then the matrix  $A$  given by (3.27) is positive definite.*

**Proof** To each  $x \in \mathbb{R}^n$  there corresponds a matrix  $V \in \mathbb{R}^{m,m}$  such that  $x = \text{vec}(V)$ . We claim that

$$x^T Ax = \sum_{j=1}^m \sum_{k=0}^m c_{j,k+\frac{1}{2}} (v_{j,k+1} - v_{j,k})^2 + \sum_{k=1}^m \sum_{j=0}^m c_{j+\frac{1}{2},k} (v_{j+1,k} - v_{j,k})^2, \quad (3.28)$$

where  $v_{0,k} = v_{m+1,k} = v_{j,0} = v_{j,m+1} = 0$  for  $j, k = 0, 1, \dots, m+1$ . Since  $c_{j+\frac{1}{2},k}$  and  $c_{j,k+\frac{1}{2}}$  correspond to values of  $c$  in  $\Omega$  for the values of  $j, k$  in the sums it follows that they are positive and from (3.28) we see that  $x^T Ax \geq 0$  for all  $x \in \mathbb{R}^n$ . Moreover if  $x^T Ax = 0$  then all quadratic factors are zero and  $v_{j,k+1} = v_{j,k}$  for  $k = 0, 1, \dots, m$  and  $j = 1, \dots, m$ . Now  $v_{j,0} = v_{j,m+1} = 0$  implies that  $V = 0$  and hence  $x = 0$ . Thus  $A$  is positive definite.

It remains to prove (3.28). From the connection between (3.26) and (3.27) we have

$$\begin{aligned} x^T Ax &= \sum_{j=1}^m \sum_{k=1}^m -(P_h v)_{j,k} v_{j,k} \\ &= \sum_{j=1}^m \sum_{k=1}^m \left( c_{j,k-\frac{1}{2}} v_{j,k}^2 + c_{j-\frac{1}{2},k} v_{j,k}^2 + c_{j+\frac{1}{2},k} v_{j,k}^2 + c_{j,k+\frac{1}{2}} v_{j,k}^2 \right. \\ &\quad - c_{j,k-\frac{1}{2}} v_{j,k-1} v_{j,k} - c_{j,k+\frac{1}{2}} v_{j,k} v_{j,k+1} \\ &\quad \left. - c_{j-\frac{1}{2},k} v_{j-1,k} v_{j,k} - c_{j+\frac{1}{2},k} v_{j,k} v_{j+1,k} \right). \end{aligned}$$

Using the homogenous boundary conditions we have

$$\begin{aligned} \sum_{j=1}^m \sum_{k=1}^m c_{j,k-\frac{1}{2}} v_{j,k}^2 &= \sum_{j=1}^m \sum_{k=0}^m c_{j,k+\frac{1}{2}} v_{j,k+1}^2, \\ \sum_{j=1}^m \sum_{k=1}^m c_{j,k-\frac{1}{2}} v_{j,k-1} v_{j,k} &= \sum_{j=1}^m \sum_{k=0}^m c_{j,k+\frac{1}{2}} v_{j,k+1} v_{j,k}, \\ \sum_{j=1}^m \sum_{k=1}^m c_{j-\frac{1}{2},k} v_{j,k}^2 &= \sum_{k=1}^m \sum_{j=0}^m c_{j+\frac{1}{2},k} v_{j+1,k}^2, \\ \sum_{j=1}^m \sum_{k=1}^m c_{j-\frac{1}{2},k} v_{j-1,k} v_{j,k} &= \sum_{k=1}^m \sum_{j=0}^m c_{j+\frac{1}{2},k} v_{j+1,k} v_{j,k}. \end{aligned}$$

It follows that

$$\begin{aligned} x^T Ax &= \sum_{j=1}^m \sum_{k=0}^m c_{j,k+\frac{1}{2}} (v_{j,k}^2 + v_{j,k+1}^2 - 2v_{j,k} v_{j,k+1}) \\ &\quad + \sum_{k=1}^m \sum_{j=0}^m c_{j+\frac{1}{2},k} (v_{j,k}^2 + v_{j+1,k}^2 - 2v_{j,k} v_{j+1,k}) \end{aligned}$$

and (3.28) follows.  $\square$

Consider solving  $Ax = b$ , where  $A$  is given by (3.27) and  $b \in \mathbb{R}^n$ . Since  $A$  is positive definite it is nonsingular and the system has a unique solution  $x \in \mathbb{R}^n$ . Moreover we can use Cholesky factorization to find  $x$ . Since  $A$  is a band matrix we can use Algorithms 2.9 and 2.10 to compute  $x$ . Since the bandwidth of  $A$  is  $m = \sqrt{n}$  this method requires  $O(2nm^2) = O(2n^2)$  for large  $n$ .

### 3.5 Problems

**3.1** Show that for  $x = (x_1, \dots, x_n)^T \in \mathbb{R}^m$  we have

$$x^T Jx = x_1^2 + x_n^2 + \sum_{k=1}^{m-1} (x_{k+1} - x_k)^2$$

and conclude again that  $J$  is positive definite.

**3.2** Show the formulae (3.7) for the  $l_1$  and  $l_\infty$  condition numbers of the matrix  $J$ .

**3.3** Show that  $C$  is positive definite if  $b > 0$  and  $b \geq 2|a|$ .

**3.4** Show the formula for the entries of the inverse of  $J$ . Hint. The entries  $x_i := a_{i,k}$  for  $i = 1, \dots, k$  in row  $k$  of the inverse is the solution of the difference equation  $-x_{i-1} + 2x_i - x_{i+1} = \delta_{i,k}$  with  $x_0 = x_{m+1} = 0$ . Solve these difference equations.

**3.5** Prove (3.15)

**3.6** Show that if  $A$  and  $B$  are positive definite then  $A \otimes B$  is positive definite.

**3.7** For  $m = 2$  the matrix (3.19) is given by

$$A = \begin{bmatrix} c & a & b & 0 \\ a & c & 0 & b \\ b & 0 & c & a \\ 0 & b & a & c \end{bmatrix}.$$

Show that  $\lambda = a + b + c$  is an eigenvalue corresponding to the eigenvector  $x = (1, 1, 1, 1)^T$ . Verify that apart from a scaling of the eigenvector this agrees with (3.22) and (3.21) for  $j = k = 1$  and  $m = 2$ .

## Chapter 4

# Fast Direct Solution of a Large Linear System

### 4.1 A Fast Poisson Solver based on Diagonalization and FFT

In this chapter we study a fast method for solving the discrete Poisson problem  $Ax = b$  given by (3.13) and (3.14). The method can be applied to other constant coefficient problems, see the problem section for some examples.

The Poisson matrix  $A$  is positive definite and banded and we could use Algorithms 2.9 and 2.10 to compute  $x$ . Since the bandwidth of  $A$  is  $m = \sqrt{n}$  this method requires  $O(2nm^2) = O(2n^2)$  for large  $n$ . The alternative algorithm we now derive will only require  $O(8n^{3/2})$  flops. In addition we only need to work with matrices of order  $m$  instead of one of order  $n = m^2$ .

To start we recall that (3.13) was derived from (3.10) and (3.12). Observe that (3.10) can be written as a matrix equation in the form

$$JV + VJ = h^2 F \quad \text{with} \quad h = 1/(m + 1), \quad (4.1)$$

where  $J = J_m = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m,m}$  is the second derivative matrix given by (3.2),  $V = (v_{jk}) \in \mathbb{R}^{m,m}$ , and  $F = (f_{jk}) = (f(jh, kh)) \in \mathbb{R}^{m,m}$ . Indeed, the  $(j, k)$  entry in  $JV + VJ$  is given by

$$\sum_{i=1}^m J_{j,i} v_{i,k} + \sum_{i=1}^m v_{j,i} J_{i,k},$$

and this is precisely the left hand side of (3.10).

Since  $J$  is symmetric the spectral theorem (cf. Leon Corollary 6.4.5) implies that there is an orthogonal matrix  $Q$  that diagonalizes  $J$ ,  $J = QDQ^T$ . The columns of  $Q$  are the orthonormal eigenvectors of  $J$  and the diagonal elements of  $D$  are the eigenvalues of  $J$ . We will show that  $Q$  is

symmetric so that  $J = QDQ$ . Inserting this factorization in (4.1) we obtain  $QDQV + VQDQ = h^2F$ . We multiply this from the left and from the right by  $Q$ . Since  $Q$  is orthogonal and symmetric we have  $Q^2 = I$  and we find  $DQVQ + QVQD = h^2QFQ$  or  $DW + WD = h^2B$ , where  $B = QFQ = (b_{j,k})$  and  $W = QVQ = (w_{j,k})$ . Since  $D$  is a diagonal matrix the equation  $DW + WD = h^2B$  can be easily solved for  $W$  and we find  $w_{j,k} = h^2b_{j,k}/(d_j + d_k)$  for  $j, k = 1, \dots, m$ . We have proved the following lemma.

**Lemma 4.1** *The solution  $V$  of  $JV + VJ = h^2F$  can be written  $V = QWQ$  where  $W = (w_{j,k})$  is given by*

$$w_{j,k} = h^2b_{j,k}/(d_j + d_k), \quad j, k = 1, \dots, m$$

and  $B = (b_{j,k}) = QFQ$ .

We need the matrices  $Q$  and  $D$  in the factorization  $J = QDQ$ .

**Lemma 4.2** *The spectral decomposition of  $J = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m,m}$  can be written*

$$J = QDQ = 2hSDS, \quad (4.2)$$

where  $h = 1/(m+1)$  and

$$Q = \sqrt{2h}S \in \mathbb{R}^{m,m}, \quad \text{with } S = \left( \sin \left( \frac{jk\pi}{m+1} \right) \right)_{j,k=1}^m, \quad (4.3)$$

and

$$D = \text{diag}(\lambda_1, \dots, \lambda_m) \in \mathbb{R}^{m,m},$$

where

$$\lambda_j = 2 - 2 \cos(j\pi h) = 4 \sin^2 \left( \frac{j\pi h}{2} \right), \quad j = 1, \dots, m. \quad (4.4)$$

Moreover  $Q$  and  $S$  are symmetric matrices with  $Q^2 = I$ .

**Proof** From Lemma 3.1 we have  $J s_j = \lambda_j s_j$ , where  $s_j$  is the  $j$ th column of  $S$  and  $\lambda_j$  is given by (4.4),  $j = 1, \dots, m$ . Since each column  $q_j$  of  $Q$  is just a scaling of  $s_j$  we immediately obtain  $J q_j = \lambda_j q_j$  for  $j = 1, \dots, m$  or  $JQ = QD$ . Clearly  $Q$  and  $S$  are symmetric matrices. Moreover  $Q$  is an orthogonal matrix since by (4.3) and (3.6) we find  $q_j^T q_k = 2h s_j^T s_k = \delta_{j,k}$  for  $j, k = 1, \dots, m$ . We conclude that  $J = QDQ = 2hSDS$  which is (4.2).  $\square$

We now have an efficient numerical method to solve the Poisson problem  $-\Delta u = f$  on  $\Omega = (0, 1)^2$ ,  $u = 0$  on  $\partial\Omega$ .



**Algorithm 4.3 (A Simple Fast Poisson Solver)**

1.  $h = 1/(m + 1); F = (f(jh, kh))_{j,k=1}^m;$   
 $S = (\sin(jk\pi h))_{j,k=1}^m; \sigma = (\sin^2((j\pi h)/2))_{j=1}^m$
2.  $G = (g_{j,k}) = SFS;$  (4.5)
3.  $Y = (y_{j,k})_{j,k=1}^m,$  where  $y_{j,k} = h^4 g_{j,k}/(\sigma_j + \sigma_k);$
4.  $V = SY S;$

The output of the algorithm is a matrix  $V \in \mathbb{R}^{m,m}$  with  $(v_{j,k})_{j,k=1}^m \approx (u(jh, kh))_{j,k=1}^m$ . For convenience we have used  $S$  instead of  $Q$  and  $\lambda_j/4$  instead of  $\lambda_j$ . Since the calculation of  $Y$  only requires  $O(m^2)$  flops the complexity of this method is for large  $m$  determined by the 4  $m$ -by- $m$  matrix multiplications and is given by  $O(4 \times 2m^3) = O(8n^{3/2})$ . (It is possible to compute  $V$  using only two matrix multiplications. This is detailed in Problem 4.4)

## 4.2 A Fast Poisson Solver based on the Discrete Sine and Fourier Transforms

In (4.3) we need to compute the product of the sine matrix  $S \in \mathbb{R}^{m,m}$  given by (4.3) and a matrix  $A \in \mathbb{R}^{m,m}$ . Since the matrices are  $m$ -by- $m$  this will normally require  $O(m^3)$  operations. In this section we show that it is possible to calculate the products  $SA$  and  $AS$  in  $O(m^2 \log_2 m)$  operations.

We need to discuss certain transforms known as the Discrete Sine Transform, the Discrete Fourier Transform and the Fast Fourier Transform. These transforms also have application to signal processing and image analysis, and is used when one is dealing with discrete samples of data on a computer.

### 4.2.1 The Discrete Sine Transform (DST)

Given  $v = (v_1, \dots, v_m) \in \mathbb{R}^m$  we say that the vector  $w = (w_1, \dots, w_m)^T$  given by

$$w_j = \sum_{k=1}^m \sin\left(\frac{jk\pi}{m+1}\right) v_k, \quad j = 1, \dots, m$$

is the *Discrete Sine Transform* (DST) of  $v$ . In matrix form we can write the DST as the matrix times vector product  $w = Sv$ , where  $S$  is the sine matrix given by (4.3). We can then identify the matrix  $B = SA$  as the DST of  $A \in \mathbb{R}^{m,n}$ , i.e. as the DST of the columns of  $A$ . The product  $B = AS$  can also be interpreted as a DST. Indeed, since  $S$  is symmetric we have  $B = (SA^T)^T$  which means that  $B$  is the transpose of the DST of the rows of  $A$ . It follows that we can compute the unknowns  $V$  in (4.3) by

carrying out Discrete Sine Transforms on 4  $m$ -by- $m$  matrices in addition to the computation of  $Y$ .

### 4.2.2 The Discrete Fourier Transform (DFT)

The fast computation of the DST is based on its relation to the Discrete Fourier Transform (DFT) and the fact that the DFT can be computed by a technique known as the Fast Fourier Transform (FFT). To define the DFT let for  $N \in \mathbb{N}$

$$\omega_N = \exp^{-2\pi i/N} = \cos\left(\frac{2\pi}{N}\right) - i \sin\left(\frac{2\pi}{N}\right), \quad (4.6)$$

where  $i = \sqrt{-1}$  is the imaginary unit. Given  $y = (y_1, \dots, y_N) \in \mathbb{R}^N$  we say that  $z = (z_1, \dots, z_N)$  given by

$$z_{j+1} = \sum_{k=0}^{N-1} \omega_N^{jk} y_{k+1}, \quad j = 0, \dots, N-1$$

is the *Discrete Fourier Transform* (DFT) of  $y$ . We can write this as a matrix times vector product  $z = F_N y$ , where the matrix  $F_N$  is given by

$$F_N = \left( \omega_N^{(j-1)(k-1)} \right)_{j,k=1}^N \in \mathbb{C}^{N,N}. \quad (4.7)$$

This matrix is known as the *Fourier matrix*. If  $A \in \mathbb{R}^{N,m}$  we say that  $B = F_N A$  is the DFT of  $A$ .

As an example, since

$$\omega_4 = \exp^{-2\pi i/4} = \cos\left(\frac{\pi}{2}\right) - i \sin\left(\frac{\pi}{2}\right) = -i$$

we find

$$F_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega_4 & \omega_4^2 & \omega_4^3 \\ 1 & \omega_4^2 & \omega_4^4 & \omega_4^6 \\ 1 & \omega_4^3 & \omega_4^6 & \omega_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}. \quad (4.8)$$

The following lemma shows how the Discrete Sine Transform of order  $m$  can be computed from the Discrete Fourier Transform of order  $2m+2$ .

**Lemma 4.4** *Given a positive integer  $m$  and a vector  $x \in \mathbb{R}^m$ . Component  $k$  of  $S_m x$  is equal to  $i/2$  times component  $k+1$  of  $F_{2m+2} z$  where*

$$z = (0, x_1, \dots, x_m, 0, -x_m, -x_{m-1}, \dots, -x_1)^T \in \mathbb{R}^{2m+2}.$$

*In symbols*

$$(S_m x)_k = \frac{i}{2} (F_{2m+2} z)_{k+1}, \quad k = 1, \dots, m.$$

**Proof** Let  $\omega = \omega_{2m+2} = e^{-2\pi i/(2m+2)} = e^{-\pi i/(m+1)}$ . Row  $k + 1$  of  $F_{2m+2}z$  is given by

$$\begin{aligned}
(F_{2m+2}z)_{k+1} &= \sum_{j=1}^m x_j \omega^{jk} - \sum_{j=m+2}^{2m+1} x_{2m+2-j} \omega^{jk} \\
&= \sum_{j=1}^m x_j \omega^{jk} - \sum_{j=1}^m x_j \omega^{(2m+2-j)k} \\
&= \sum_{j=1}^m x_j (\omega^{jk} - \omega^{-jk}) \\
&= -2i \sum_{j=1}^m x_j \sin\left(\frac{jk\pi}{m+1}\right) = -2i(S_m x)_k.
\end{aligned}$$

Dividing both sides by  $-2i$  proves the Lemma.  $\square$

It follows that we can compute the DST of length  $m$  by extracting  $m$  components from the DFT of length  $N = 2m + 2$ .

### 4.2.3 The Fast Fourier Transform (FFT)

From a linear algebra viewpoint the Fast Fourier Transform is a quick way to compute the matrix- vector product  $F_N y$ . Suppose  $N$  is even. The key to the FFT is a connection between  $F_N$  and  $F_{N/2}$  which makes it possible to compute the FFT of order  $N$  as two FFT's of order  $N/2$ . By repeating this process we can reduce the number of flops for a DFT from  $O(N^2)$  to  $O(N \log_2 N)$ .

Suppose  $N$  is even. The connection between  $F_N$  and  $F_{N/2}$  involves a permutation matrix  $P_N \in \mathbb{R}^{N,N}$  given by

$$P_N = [e_1, e_3, \dots, e_{N-1}, e_2, e_4, \dots, e_N],$$

where the  $e_k = (\delta_{j,k})$  are unit vectors. If  $A$  is a matrix with  $N$  columns  $(a_1, \dots, a_N)$  then

$$AP_N = [a_1, a_3, \dots, a_{N-1}, a_2, a_4, \dots, a_N],$$

i.e. post multiplying  $A$  by  $P_N$  permutes the columns of  $A$  so that all the odd-indexed columns are followed by all the even-indexed columns. For example we have from (4.8)

$$P_4 = [e_1 \ e_3 \ e_2 \ e_4] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad F_4 P_4 = \left[ \begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & -i & i \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & i & -i \end{array} \right],$$

where we have indicated a certain block structure of  $F_4P_4$ . These blocks can be related to the 2-by-2 matrix  $F_2$ . We define the diagonal scaling matrix  $D_2$  by

$$D_2 = \text{diag}(1, \omega_4) = \begin{bmatrix} 1 & 0 \\ 1 & -i \end{bmatrix}.$$

Since  $\omega_2 = \exp^{-2\pi i/2} = -1$  we find

$$F_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad D_2F_2 = \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix},$$

and we see that

$$F_4P_4 = \left[ \begin{array}{c|c} F_2 & D_2F_2 \\ \hline F_2 & -D_2F_2 \end{array} \right].$$

This results hold in general.

**Theorem 4.5** *If  $N = 2m$  is even then*

$$F_{2m}P_{2m} = \left[ \begin{array}{c|c} F_m & D_mF_m \\ \hline F_m & -D_mF_m \end{array} \right], \quad (4.9)$$

where

$$D_m = \text{diag}(1, \omega_N, \omega_N^2, \dots, \omega_N^{m-1}). \quad (4.10)$$

**Proof** Fix integers  $j, k$  with  $0 \leq j, k \leq m-1$  and set  $p = j+1$  and  $q = k+1$ . Since  $\omega_m^m = 1$ ,  $\omega_N^2 = \omega_m$ , and  $\omega_N^m = -1$  we find by considering elements in the four sub-blocks in turn

$$\begin{aligned} (F_{2m}P_{2m})_{p,q} &= \omega_N^{j(2k)} &= \omega_m^{jk} &= (F_m)_{p,q}, \\ (F_{2m}P_{2m})_{p+m,q} &= \omega_N^{(j+m)(2k)} &= \omega_m^{(j+m)k} &= (F_m)_{p,q}, \\ (F_{2m}P_{2m})_{p,q+m} &= \omega_N^{j(2k+1)} &= \omega_N^j \omega_m^{jk} &= (D_mF_m)_{p,q}, \\ (F_{2m}P_{2m})_{p+m,q+m} &= \omega_N^{(j+m)(2k+1)} &= -\omega_N^j \omega_m^{jk} &= (-D_mF_m)_{p,q}. \end{aligned}$$

It follows that the four  $m$ -by- $m$  blocks of  $F_{2m}P_{2m}$  have the required structure.  $\square$

Using Theorem 4.5 we can carry out the DFT as a block multiplication. Let  $y \in \mathbb{R}^{2m}$  and set  $w = P_{2m}^T y = (w_1, w_2)^T$ , where  $w_1, w_2 \in \mathbb{R}^m$ . Then

$$\begin{aligned} F_{2m}y &= F_{2m}P_{2m}P_{2m}^T y = F_{2m}P_{2m}w \\ &= \left[ \begin{array}{c|c} F_m & D_mF_m \\ \hline F_m & -D_mF_m \end{array} \right] \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} q_1 + q_2 \\ q_1 - q_2 \end{bmatrix}, \end{aligned}$$

where

$$q_1 = F_m w_1, \quad \text{and} \quad q_2 = D_m(F_m w_2).$$

In order to compute  $F_{2m}y$  we need to compute  $F_m w_1$  and  $F_m w_2$ . Note that  $w_1^T = (y_1, y_3, \dots, y_{N-1})$ , while  $w_2^T = (y_2, y_4, \dots, y_N)$ . This follows since  $w^T = (w_1^T, w_2^T) = y^T P_{2m}$  and post multiplying a vector by  $P_{2m}$  moves odd indexed components to the left of all the even indexed components.

We have seen that by combining two FFT's of order  $m$  we obtain an FFT of order  $2m$ . If  $N = 2^k$  then this process can be applied recursively as in the following Matlab function:

**Algorithm 4.6 (Recursive FFT )**

```
function z=ffttrec(y)
n=length(y);
if n==1    z=y;
else
    q1=ffttrec(y(1:2:n-1));
    q2=exp(-2*pi*i/n).^ (0:n/2-1).*ffttrec(y(2:2:n));
    z=[q1+q2 q1-q2];
end
```

Such a recursive version of FFT is useful for testing purposes, but is too slow for large problems. A challenge for FFT code writers is to develop nonrecursive versions and also to handle efficiently the case where  $N$  is not a power of two. We refer to [?] for further details.

The complexity of the FFT is given by  $\gamma N \log_2 N$  for some constant  $\gamma$  independent of  $N$ . To show this for the special case when  $N$  is a power of two let  $x_k$  be the complexity (the number of flops) when  $N = 2^k$ . Since we need two FFT's of order  $N/2 = 2^{k-1}$  and a multiplication with the diagonal matrix  $D_{N/2}$  it is reasonable to assume that  $x_k = 2x_{k-1} + \gamma 2^k$  for some constant  $\gamma$  independent of  $k$ . Since  $x_0 = 0$  we obtain by induction on  $k$  that  $x_k = \gamma k 2^k$ . This holds for  $k = 0$  and if  $x_{k-1} = \gamma(k-1)2^{k-1}$  then  $x_k = 2x_{k-1} + \gamma 2^k = 2\gamma(k-1)2^{k-1} + \gamma 2^k = \gamma k 2^k$ . Reasonable implementations of FFT typically have  $\gamma \approx 5$ , see [?].

The efficiency improvement using the FFT to compute the DFT is spectacular for large  $N$ . The direct multiplication  $F_N y$  requires  $O(8n^2)$  flops since complex arithmetic is involved. Assuming that the FFT uses  $5N \log_2 N$  flops we find for  $N = 2^{20} \approx 10^6$  the ratio

$$\frac{8N^2}{5N \log_2 N} \approx 84000.$$

Thus if the FFT takes one second of computing time and the computing time is proportional to the number of flops then the direct multiplication would take something like 84000 seconds or 23 hours.

#### 4.2.4 A Poisson Solver based on the FFT

We now have all the ingredients to compute the matrix products  $SA$  and  $AS$  using FFT's of order  $2m+2$  where  $m$  is the order of  $S$  and  $A$ . This can then be used for quick computation of the exact solution  $V$  of the discrete Poisson problem given by (4.3). We first compute  $H = SF$  using Lemma 4.4 and  $m$  FFT's, one for each of the  $m$  columns of  $F$ . We then compute  $G = HS$  by  $m$  FFT's, one for each of the rows of  $H$ . After  $Y$  is determined we compute  $Z = SY$  and  $V = ZS$  by another  $2m$  FFT's. In total the work amounts to  $4m$  FFT's of order  $2m+2$ . Since one FFT requires  $O(\gamma(2m+2)\log_2(2m+2))$  flops the  $4m$  FFT's amounts to

$$8\gamma m(m+1)\log_2(2m+2) \approx 8\gamma m^2 \log_2 m = 4\gamma n \log_2 n,$$

where  $n = m^2$  is the size of the linear system  $Ax = b$  we would be solving if Cholesky factorization was used. This should be compared to the  $O(8n^{3/2})$  flops for solving (4.3) using 4 straightforward matrix multiplications with  $S$ . What is faster will depend on the programming of the FFT and the size of the problem. We refer to [?] for other efficient ways to implement the DST.

### 4.3 Problems

**4.1** Show that the Fourier matrix  $F_4$  is symmetric, but not Hermitian.

**4.2** Verify Lemma 4.4 directly when  $m = 1$ .

**4.3** Show that the exact solution of the discrete Poisson equation (3.11) and (3.12) can be written  $V = (v_{i,j})_{i,j=1}^m$ , where

$$v_{ij} = \frac{1}{(m+1)^4} \sum_{p=1}^m \sum_{r=1}^m \sum_{k=1}^m \sum_{l=1}^m \frac{\sin\left(\frac{ip\pi}{m+1}\right) \sin\left(\frac{jr\pi}{m+1}\right) \sin\left(\frac{kp\pi}{m+1}\right) \sin\left(\frac{lr\pi}{m+1}\right)}{\left[\sin\left(\frac{p\pi}{2(m+1)}\right)\right]^2 + \left[\sin\left(\frac{r\pi}{2(m+1)}\right)\right]^2} y_{p,r}$$

**4.4** The method (4.3) involves multiplying a matrix by  $S$  four times. In this problem we show that it is enough to multiply by  $S$  two times. We achieve this by diagonalizing only the second  $J$  in (4.1). We use the notation in Lemma 4.2. In particular the diagonalization of  $J$  is written  $J = 2hSDS$ , where  $h = 1/(m+1)$ .

(a) Show that

$$JX + XD = C, \text{ where } X = VS, \text{ and } C = h^2FS.$$

(b) Show that

$$(J + \lambda_j I)x_j = c_j \quad j = 1, \dots, m, \quad (4.11)$$

where  $X = (x_1, \dots, x_m)$  and  $C = (c_1, \dots, c_m)$  and  $\lambda_j$  is given by (4.4). Thus we can find  $X$  by solving  $m$  linear systems, one for each of the columns of  $X$ . Give an algorithm to find  $X$  which only requires  $O(\delta m^2)$  flops for some constant  $\delta$  independent of  $m$ . (You do not have to determine the exact value of  $\delta$ .)

(c) Describe a method to compute  $V$  which only requires  $O(4m^3) = O(4n^{3/2})$  flops.

(d) Describe a method based on the Fast Fourier Transform which requires  $O(\gamma n \log_2 n)$  where  $\gamma$  is the same constant as mentioned at the end of the last section.

**4.5** Consider the following 9 point difference approximation to (3.9)

$$\begin{aligned} \text{(a)} \quad & -(\square_h v)_{j,k} = (\mu f)_{j,k} \quad j, k = 1, \dots, m \\ \text{(b)} \quad & v_{j,k} = 0 \quad v_{0,k} = v_{m+1,k} = v_{j,0} = v_{j,m+1} = 0, \\ & \quad \quad \quad j, k = 0, 1, \dots, m+1, \end{aligned}$$

where

$$\begin{aligned} \text{(c)} \quad & -(\square_h v)_{j,k} = [20v_{j,k} - 4v_{j-1,k} - 4v_{j,k-1} - 4v_{j+1,k} - 4v_{j,k+1} \\ & \quad \quad \quad - v_{j-1,k-1} - v_{j+1,k-1} - v_{j-1,k+1} - v_{j+1,k+1}]/(6h^2), \\ \text{(d)} \quad & (\mu f)_{j,k} = [8f_{j,k} + f_{j-1,k} + f_{j,k-1} + f_{j+1,k} + f_{j,k+1}]/12. \end{aligned} \tag{4.12}$$

a) Write down the 4-by-4 system we obtain for  $m = 2$ .

b) Find  $v_{j,k}$ ,  $j, k = 1, 2$ , if  $f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y)$  and  $m = 2$ . Answer:  $v_{j,k} = 5\pi^2/66$ .

Note that (4.12) defines an  $O(h^4)$  approximation to (3.9).

**4.6** Consider the nine point difference approximation to (3.9) given by (4.12) in Problem 4.5. Show that (4.12) is equivalent to the matrix equation

$$JV + VJ - \frac{1}{6}JVV = h^2 \mu F. \tag{4.13}$$

Here  $\mu F$  has elements  $(\mu f)_{j,k}$  given by (4.12d).

**4.7** Let  $X = QVQ = (x_{j,k})$  where  $V$  is the solution of (4.13) and  $Q$  is given by (4.3). Show that

$$x_{j,k} = \frac{h^2 g_{j,k}}{\lambda_j + \lambda_k - \lambda_j \lambda_k / 6}, \quad j, k = 1, 2, \dots, m,$$

where  $G = Q(\mu F)Q$ . Show that  $\lambda_j + \lambda_k - \lambda_j \lambda_k / 6 > 0$  for  $j, k = 1, 2, \dots, m$ . Conclude that (4.12) always has a solution  $V = QXQ$ .

**4.8** Derive an algorithm for solving (4.12) which for large  $m$  requires essentially the same number of operations as (4.3). (We assume that  $\mu F$  already has been formed).

**4.9** Consider the biharmonic equation

$$\begin{aligned} \Delta^2 u(s, t) &= \Delta(\Delta u(s, t)) = f(s, t) & (s, t) \in \Omega, \\ u(s, t) &= 0, \quad \Delta u(s, t) = 0 & (s, t) \in \partial\Omega. \end{aligned} \quad (4.14)$$

Here  $\Omega$  is the unit square. The condition  $\Delta u = 0$  is called the *Navier boundary condition*. Moreover,  $\Delta^2 u = u_{xxxx} + 2u_{xxyy} + u_{yyyy}$ .

a) Let  $v = -\Delta u$ . Then (4.14) can be written as a system

$$\begin{aligned} -\Delta v(s, t) &= f(s, t) & (s, t) \in \Omega \\ -\Delta u(s, t) &= v(s, t) & (s, t) \in \Omega \\ u(s, t) &= v(s, t) = 0 & (s, t) \in \partial\Omega. \end{aligned} \quad (4.15)$$

b) Discretizing, using (3.10), we get two matrix equations

$$JV + VJ = h^2 F, \quad JU + UJ = h^2 V.$$

Show that

$$J^2 U + 2JUJ + UJ^2 = h^4 F.$$

c) Let  $X = QUQ$  where  $Q$  is given by (4.3). Show that

$$x_{i,k} = \frac{h^4 g_{i,k}}{(\lambda_i + \lambda_k)^2},$$

where  $\lambda_k$  is given by (3.5) and  $G = QFQ$ .

**4.10** Derive an algorithm which requires only  $O(\delta n^{3/2})$  operations to find  $U$  in Problem 4.9. Here  $\delta$  is some constant independent of  $n$ .



## Chapter 5

# The Conjugate Gradient Method

The conjugate gradient method is an iterative algorithm for solving a positive definite large linear system of equations. We compute a sequence of approximations to the exact solution. Each new approximation  $x_{k+1}$  is computed from the previous  $x_k$  by a formula of the form

$$x_{k+1} = x_k + \alpha_k p_k, \quad (5.1)$$

where  $p_k$  is a vector, the **search direction**, and  $\alpha_k$  is a scalar determining the **step length**. The number of iterations to achieve a desired accuracy is essentially proportional to the square root of the 2-norm condition number of the coefficient matrix of the linear system. Thus the smaller the condition number the faster the method converges.

For problems with a large condition number the convergence can be slow. For such problems a **preconditioned conjugate gradient method** is often used, and we also consider this method here.

### 5.1 Derivation and Basic Properties

Let  $A \in \mathbb{R}^{n,n}$  be positive definite. We will use two inner products on  $\mathbb{R}^n$

1.  $(x, y) = x^T y$
2.  $\langle x, y \rangle = x^T A y$ .

The first product is the usual inner product corresponding to the Euclidean norm, while the second product is an inner product since  $A$  is positive definite. Indeed, since  $A$  is symmetric and positive definite we have

$$\langle x, y \rangle = (x, Ay) = (Ax, y) = (L^T x, L^T y),$$

where  $A = LL^T$  is the Cholesky factorization of  $A$ . In particular  $\langle x, x \rangle = \|L^T x\|_2^2 = 0$  if and only if  $x = 0$ . The associated norm

$$\|x\|_A = \sqrt{\langle x, x \rangle}$$

is called the **A-norm** or **energy norm** of  $x$ . The Euclidian norm  $\sqrt{\langle x, x \rangle}$  will be denoted by  $\|x\|$ . Two vectors  $x, y \in \mathbb{R}^n$  are **orthogonal** if  $\langle x, y \rangle = 0$  and **A-orthogonal** if  $\langle x, y \rangle = 0$ .

Suppose  $x_0 \in \mathbb{R}^n$  is an initial approximation to the solution of the linear system  $Ax = b$  and let  $r_0 := b - Ax_0$  be the corresponding residual. We consider the *Krylov subspaces*  $\mathbb{W}_k$  of  $\mathbb{R}^n$  defined by  $\mathbb{W}_0 = \{0\}$  and

$$\mathbb{W}_k = \text{span}(r_0, Ar_0, A^2 r_0, \dots, A^{k-1} r_0), \quad k = 1, 2, 3, \dots$$

This is a nested sequence of subspaces

$$\mathbb{W}_0 \subset \mathbb{W}_1 \subset \mathbb{W}_2 \subset \dots \subset \mathbb{W}_n \subset \mathbb{R}^n$$

with  $\dim(\mathbb{W}_k) \leq k$  for all  $k \geq 0$ . We also note that if  $w \in \mathbb{W}_k$  then  $Aw \in \mathbb{W}_{k+1}$ .

**Definition 5.1** Suppose  $A \in \mathbb{R}^{n,n}$  is positive definite,  $f \in \mathbb{R}^n$ , and that  $k$  is a positive integer. The  $k$ th approximation  $x_k$  in the conjugate gradient method (CG) for solving  $Ax = f$  is a vector such that  $x_k - x_0 \in \mathbb{W}_k$  and satisfying

$$\langle x_k - x, w \rangle = 0, \quad w \in \mathbb{W}_k, \quad (5.2)$$

or equivalently

$$\langle x_k, w \rangle = \langle x, w \rangle, \quad w \in \mathbb{W}_k. \quad (5.3)$$

In general  $x_k$  will belong to the translated subspace  $x_0 + \mathbb{W}_k$  of all vectors  $x_0 + w$  with  $w \in \mathbb{W}_k$  and  $x - x_0$  solves the linear system  $A(x - x_0) = r_0$ . Moreover  $x - x_0$  is A-orthogonal to  $x_0 + w$  for all  $w \in \mathbb{W}_k$ . From this we will derive the recursive formula (5.1). The derivation is valid with minor modification for a general  $x_0$ , however for simplicity of notation we will assume that  $x_0 = 0$  so that  $r_0 = f$  and  $x_k \in \mathbb{W}_k = \text{span}(f, Af, A^2 f, \dots, A^{k-1} f)$ .

We first show that the residuals are orthogonal.

**Lemma 5.2** Suppose for some  $k \geq 1$  that the residuals  $r_j := f - Ax_j$  are nonzero for  $j = 0, 1, \dots, k-1$ . Then  $\dim(\mathbb{W}_k) = k$  and  $(r_0, r_1, \dots, r_{k-1})$  is an orthogonal basis for  $\mathbb{W}_k$ .

**Proof** We obtain from (5.2) that  $(w, r_k) = w^T r_k = w^T (f - Ax_k) = w^T A(x - x_k) = \langle x - x_k, w \rangle = 0$  which means that

$$(r_k, w) = 0, \quad w \in \mathbb{W}_k. \quad (5.4)$$

Since  $r_j \in \mathbb{W}_{j+1}$  this shows that  $(r_k, r_j) = 0$  for  $j < k$  and the residuals form an orthogonal basis for  $\mathbb{W}_k$ .  $\square$

We obtain a recurrence (5.1) provided the search directions  $p_k$  are  $A$ -orthogonal.

**Lemma 5.3** *Suppose  $p_0, p_1, \dots, p_k$  is an  $A$ -orthogonal basis for  $\mathbb{W}_{k+1}$  and that  $p_j \in \mathbb{W}_{j+1}$  for  $j = 0, 1, \dots, k$ . Then*

$$x_{k+1} = x_k + \alpha_k p_k, \quad \text{with} \quad \alpha_k = \frac{\langle x, p_k \rangle}{\langle p_k, p_k \rangle}. \quad (5.5)$$

**Proof** By  $A$ -orthogonality and (5.3) we have

$$x_{k+1} = \sum_{j=0}^k \frac{\langle x_{k+1}, p_j \rangle}{\langle p_j, p_j \rangle} p_j = \sum_{j=0}^k \frac{\langle x, p_j \rangle}{\langle p_j, p_j \rangle} p_j.$$

Since  $x_k$  is equal to the sum of the first  $k$  terms on the right we obtain (5.5).  $\square$

In the next lemma we use a non normalized version of Gram-Schmidt to construct an  $A$ -orthogonal basis from the residuals.

**Lemma 5.4** *Suppose  $\dim(\mathbb{W}_j) = j$  for  $j = 1, \dots, k+2$  and define  $p_0 = r_0$  and*

$$p_{j+1} = r_{j+1} - \sum_{i=0}^j \frac{\langle r_{j+1}, p_i \rangle}{\langle p_i, p_i \rangle} p_i, \quad j = 0, \dots, k. \quad (5.6)$$

*Then  $p_0, p_1, \dots, p_k$  is an  $A$ -orthogonal basis for  $\mathbb{W}_{k+1}$  and*

$$p_{k+1} = r_{k+1} + \beta_k p_k, \quad \text{with} \quad \beta_k = -\frac{\langle r_{k+1}, p_k \rangle}{\langle p_k, p_k \rangle}. \quad (5.7)$$

*Moreover*

$$\langle p_k, w \rangle = 0 \quad \text{for} \quad w \in \mathbb{W}_k. \quad (5.8)$$

**Proof** We use induction on  $k$ . Suppose for some  $k \geq 0$  that  $p_0, p_1, \dots, p_k$  is an  $A$ -orthogonal basis for  $\mathbb{W}_{k+1}$  and let  $p_{k+1}$  be given by (5.6) with  $j = k$ . This clearly holds for  $k = 0$ . Since  $r_0, \dots, r_k$  is a basis for  $\mathbb{W}_{k+1}$  we have  $p_{k+1} = r_{k+1} + \sum_{j=0}^k \alpha_j r_j$  for some  $\alpha_0, \dots, \alpha_k$  and  $p_{k+1} \neq 0$  by the linear independence of the residuals. By assumption  $\langle p_i, p_j \rangle = 0$  for  $i \neq j$  and  $i, j \leq k$ . Linearity of the inner product then gives

$$\begin{aligned} \langle p_{k+1}, p_j \rangle &= \langle r_{k+1}, p_j \rangle - \sum_{i=0}^k \frac{\langle r_{k+1}, p_i \rangle}{\langle p_i, p_i \rangle} \langle p_i, p_j \rangle \\ &= \langle r_{k+1}, p_j \rangle - \frac{\langle r_{k+1}, p_j \rangle}{\langle p_j, p_j \rangle} \langle p_j, p_j \rangle = 0 \end{aligned}$$

for  $j = 0, \dots, k$ . Thus  $p_0, p_1, \dots, p_{k+1}$  is an  $A$ -orthogonal basis for  $\mathbb{W}_{k+2}$ . By (5.4)  $\langle r_{k+1}, p_j \rangle = 0$  for  $j = 0, \dots, k-1$  and (5.6) with  $j = k$  reduces to (5.7). Since  $p_0, \dots, p_{k-1}$  is a basis for  $\mathbb{W}_k$  we obtain (5.8)  $\square$

For computational purposes it is convenient to use slightly different formulae for the numbers  $\alpha_k$  and  $\beta_k$  in (5.5) and (5.7). We state this in the following lemma together with a recursive formula for the residuals.

**Lemma 5.5** *Suppose  $k \geq 0$  and  $r_j \neq 0$  for  $j = 0, 1, \dots, k$ . Then*

$$x_{k+1} = x_k + \alpha_k p_k, \quad \alpha_k = r_k^T r_k / p_k^T A p_k, \quad (5.9)$$

$$r_{k+1} = r_k - \alpha_k A p_k, \quad (5.10)$$

$$p_{k+1} = r_{k+1} + \beta_k p_k, \quad \beta_k = r_{k+1}^T r_{k+1} / r_k^T r_k. \quad (5.11)$$

**Proof** Since  $p_{j-1} \in \mathbb{W}_j$  we have  $(r_j, p_{j-1}) = 0$  by (5.4). This also holds for  $j = 0$  if we define  $p_{-1} = 0$ . Hence by (5.7)

$$(p_j, r_j) = (r_j + \beta_{j-1} p_{j-1}, r_j) = (r_j, r_j) = r_j^T r_j, \quad j = 0, 1, \dots, k+1. \quad (5.12)$$

Therefore, by (5.8) we find  $\langle x, p_k \rangle = \langle x - x_k, p_k \rangle = (r_k, p_k) = r_k^T r_k$  and the formula for  $\alpha_k$  follows. We obtain (5.10) if we multiply both sides of (5.9) by  $-A$  and add  $f$ .

By (5.8), (5.7), Lemma 5.2, and (5.12)

$$\begin{aligned} (p_{k+1}, r_{k+1}) &= \langle p_{k+1}, x - x_{k+1} \rangle = \langle p_{k+1}, x - x_k \rangle = (p_{k+1}, r_k) \\ &= (r_{k+1}, r_k) + \beta_k (p_k, r_k) = \beta_k (p_k, r_k) = \beta_k (r_k, r_k). \end{aligned}$$

Hence the formula for  $\beta_k$  follows.  $\square$

Lemma 5.2 implies that  $x_m$  is equal to the solution  $x$  of  $Ax = f$  for some  $m \leq n$ . For if  $x_k \neq x$  for all  $k = 0, 1, \dots, n-1$  then  $r_k \neq 0$  for  $k = 0, 1, \dots, n-1$  and hence  $\dim(\mathbb{W}_n) = n$ . But then  $x \in \mathbb{W}_n = \mathbb{R}^n$  and  $x_n = x$ .

We can show that the Euclidian norm of the error is monotonically decreasing. The proof of the following result is outlined in the exercises.

**Lemma 5.6** *Let  $x$  be the exact solution of  $Ax = b$ , define  $e_k = x - x_k$  for  $k \geq 0$  and let  $\| \cdot \|$  denote the Euclidian vector norm. If  $p_j \neq 0$  for  $j \leq k$  then  $\|e_{k+1}\| < \|e_k\|$ . More precisely,*

$$\|e_{k+1}\|^2 = \|e_k\|^2 - \frac{\|p_k\|^2}{\|p_k\|_A^2} (\|e_{k+1}\|_A^2 + \|e_k\|_A^2). \quad (5.13)$$

In many problems the solution  $x$  of  $Ax = f$  is itself an approximation to something else. For example for the Poisson problem  $x$  is an approximation to the solution of the continuous problem. In such cases there is no big loss if we stop with a sufficiently accurate approximation  $x_k$  for  $k < n$ . The convergence analysis which we give in a later section shows that  $\|x - x_k\|_A$  typically becomes small quite rapidly and we can stop the iteration with  $k$

much smaller than  $n$ . It is this rapid convergence which makes the method interesting.

The formulae in Lemma 5.5 form the basis for an algorithm for solving  $Ax = b$  with  $A$  positive definite.

**Algorithm 5.7 (Conjugate Gradient Algorithm)**

1. Choose a starting vector  $x_0$  (for example  $x_0 = 0$ )
2.  $p_0 = r_0 = f - Ax_0$
3.  $\rho_0 = r_0^T r_0$ ;  $k = 0$
4. while  $\sqrt{\rho_k/\rho_0} > \epsilon$  &  $k < kmax$ 
  - 4.1  $t_k = Ap_k$
  - 4.2  $\alpha_k = \rho_k/p_k^T t_k$
  - 4.3  $x_{k+1} = x_k + \alpha_k p_k$
  - 4.4  $r_{k+1} = r_k - \alpha_k t_k$
  - 4.5  $\rho_{k+1} = r_{k+1}^T r_{k+1}$
  - 4.6  $p_{k+1} = r_{k+1} + \frac{\rho_{k+1}}{\rho_k} p_k$
  - 4.7  $k = k + 1$

Since the Euclidian norm of the residuals are monotonically decreasing we have chosen to stop the iteration when  $\|r_k\|/\|r_0\|$  is smaller than a prescribed tolerance.

The work involved in each iteration is

1. one matrix times vector (4.1)
2. two inner products (4.2 and 4.5)
3. three vector-plus-scalar-times-vector (4.3, 4.4, and 4.6)

## 5.2 Numerical Examples

To a given integer  $m \geq 1$  and  $a, b, c \in \mathbb{R}$  we consider the linear system  $Ax = f$ , where  $A \in \mathbb{R}^{n,n}$ ,  $x, f \in \mathbb{R}^n$ , and  $n = m^2$ . Here  $f = h^2(1, 1, \dots, 1)^T$  with  $h = 1/(m + 1)$  and the entries of  $A$  are given by

$$\begin{aligned}
 a_{i,i+1} = a_{i+1,i} &= a, & i = 1, \dots, n-1, & \quad i \neq m, 2m, \dots, (m-1)m, \\
 a_{i,i+m} = a_{i+m,i} &= b, & i = 1, \dots, n-m, \\
 a_{i,i} &= c, & i = 1, \dots, n, \\
 a_{i,j} &= 0, & \text{otherwise.}
 \end{aligned}
 \tag{5.14}$$

Note that this matrix is the same as the one we studied in Chapter 3 and we have shown that it is positive definite for  $c > 0$  and  $c \geq 2|a| + 2|b|$ . We try

$n$	2500	10000	22500	40000	62500
$K$	93	187	279	369	459
$K/\sqrt{n}$	1.86	1.87	1.86	1.85	1.84

Table 5.8: The number of iterations  $K$  for CG on a  $\sqrt{n} \times \sqrt{n}$  grid for the 2-dimensional Poisson problem.

n	2500	10000	22500	40000	62500
K	18	17	17	17	16

Table 5.9: The number of iterations  $K$  for the averaging problem on a  $\sqrt{n} \times \sqrt{n}$  grid.

Algorithm 5.7 for two different choices of the parameters  $a, b, c$ . One choice with a condition number which grows linearly with  $n$  and one problem where the condition number can be bounded independently of  $n$ .

In Table 5.8 we show the results of solving  $Ax = f$ , where  $a = b = -1$ , and  $c = 4$  (corresponding to the discrete Poisson problem) for various choices of  $n = m^2$ . Using CG in the form of Algorithm 5.7 with  $\epsilon = 10^{-8}$  and  $x_0 = 0$  we list  $K$ , the required number of iterations and  $K/\sqrt{n}$ . The results show that  $K$  is much smaller than  $n$  and appears to be proportional to  $\sqrt{n}$  which is essentially the square root of the condition number of  $A$ . Indeed, in Lemma 3.9 we showed that  $\text{cond}_2(A) \approx \frac{4}{\pi^2}(m+1)^2 = O(n)$ .

We have also tried Algorithm 5.7 with  $x_0 = 0$  and  $\epsilon = 10^{-8}$  on (5.14) with  $a = b = 1/9$ , and  $c = 5/9$ , the averaging problem. In this case both the condition number and the required number of iterations as shown in Table 5.9 are independent of the size of the problem and the convergence is quite rapid.

Consider now the complexity of Algorithm 5.7 for the problem defined by (5.14). A central part of the computation is statement 4.1,  $t_k = Ap_k$ . Note that  $A$  only has  $O(5n)$  nonzero elements. Therefore, taking advantage of the sparseness of  $A$  we can compute  $t_k$  in  $O(n)$  flops. With such an implementation the total number of flops in one iteration is  $O(n)$ .

How many flops do we need to solve  $Ax = f$  by the conjugate gradient method to within a tolerance  $\epsilon$ ? Consider the discrete Poisson problem. As indicated by Table 5.8 we need  $O(n^{1/2})$  number of iterations for a prescribed tolerance, and since each iteration requires  $O(n)$  flops we arrive at a total computational effort of  $O(n^{3/2})$  flops. This number is between  $O(n^2)$  and  $O(n \log_2 n)$  which is what is needed for Cholesky factorization and the fast Fourier transform, respectively. On the other hand for the averaging problem the number of iterations seems to be independent of  $n$  and the total number of flops to solve this problem to within a prescribed tolerance is  $O(n)$ . This

is optimal since we have  $n$  unknowns in the problem.

Sometimes it is convenient to use the following 2-dimensional formulation of the system (5.14)

$$\begin{aligned} cv_{i,j} + av_{i-1,j} + av_{i+1,j} + bv_{i,j-1} + bv_{i,j+1} &= h^2, & i, j = 1, \dots, m \\ v_{i,0} = v_{i,m+1} = v_{0,j} = v_{m+1,j} &= 0, & i, j = 0, \dots, m+1, \end{aligned} \quad (5.15)$$

where  $V = (v_{i,j}) \in \mathbb{R}^{m,m}$  is the matrix which has the unknown  $x$  in  $Ax = f$  as its vectorized form,  $x = \text{vec}(V)$ . We recall that  $\text{vec}(V)$  is a one dimensional column vector obtained from  $V$  by listing the entries of  $V$  column by column starting at the left of the matrix. (5.15) can be written in matrix form as

$$cV + DV + VE = h^2F, \quad (5.16)$$

where  $D = \text{tridiag}(a, 0, a) \in \mathbb{R}^{m,m}$  and  $E = \text{tridiag}(b, 0, b) \in \mathbb{R}^{m,m}$  are tridiagonal matrices with zero on the diagonal, and  $F \in \mathbb{R}^{m,m}$  has all entries equal to one. (5.16) follows by comparing the  $(i, j)$ -entry on both sides

$$\begin{aligned} h^2 f_{i,j} &= cv_{i,j} + \sum_{k=1}^m d_{i,k} v_{k,j} + \sum_{k=1}^m v_{i,k} e_{k,j} \\ &= cv_{i,j} + av_{i-1,j} + av_{i+1,j} + bv_{i,j-1} + bv_{i,j+1}. \end{aligned}$$

This holds for  $i, j = 1, \dots, m$  and shows that (5.15) and (5.16) are equivalent.

In an implementation based on (5.16) statement 4.1 in Algorithm 5.7 can be written

$$T = cV + DV + VE. \quad (5.17)$$

If Matlab is used we can store the tridiagonal matrices  $D$  and  $E$  in sparse form. The calculation of  $T$  will then be quite fast and requires very little programming effort.

### 5.3 Minimization

The conjugate gradient method can also be used as a minimization algorithm. If  $A$  is positive definite then solving  $Ax = b$  is equivalent to minimizing

$$Q(x) := x^T Ax - 2b^T x$$

over  $\mathbb{R}^n$ . We denote the global minimum by  $x^*$  in this section. A general class of minimization algorithms for  $Q$  is given as follows:

1. Choose  $x_0 \in \mathbb{R}^n$ .
2. For  $k = 0, 1, 2, \dots$ 
  - (a) Choose a “search direction”  $d_k$ .

(b) Choose a “step length”  $\sigma_k$ .

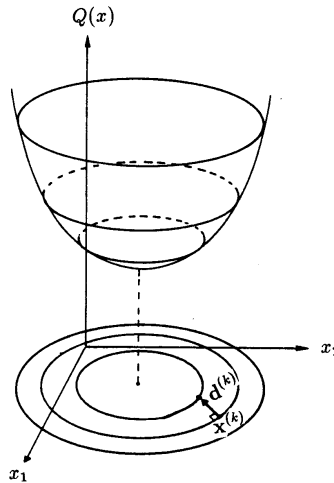
(c)  $x_{k+1} = x_k + \sigma_k d_k$ .

We would like to generate a sequence  $\{x_k\}$  of points such that  $\{x_k\}$  converges quickly to the minimum  $x$  of  $Q$ .

We can think of  $Q(x)$  as a paraboloid. To see this, let  $A = UDU^T$ , where  $U$  is orthogonal and  $D = \text{diag}(\lambda_1, \dots, \lambda_n)$  is diagonal, be the spectral decomposition of  $A$  and change variables to  $v = (v_1, \dots, v_n) := U^T x$  and  $c := U^T b = (c_1, \dots, c_n)$ . Then

$$Q(x) = x^T UDU^T x - b^T U U^T x = v^T D v - c^T v = \sum_{j=1}^n \lambda_j v_j^2 - \sum_{j=1}^n c_j v_j.$$

In particular for  $n = 2$  we have  $z := \lambda_1 v_1^2 + \lambda_2 v_2^2 - c_1 v_1 - c_2 v_2$  and since  $\lambda_1$  and  $\lambda_2$  are positive this is the equation for a paraboloid in  $(v_1, v_2, z)$  space as shown in the following figure.



For each approximation  $x_k$  to  $x^*$  we choose a search direction  $d_k$  and go from  $x_k$  along  $d_k$  a certain distance determined by  $\sigma_k$ . To see how  $\sigma_k$  and  $d_k$  should be chosen, we note that

$$Q(x_{k+1}) = Q(x_k) - 2\sigma_k(d_k, r_k) + \sigma_k^2(d_k, Ad_k), \quad (5.18)$$

where  $r_k = b - Ax_k$ . Since  $A$  is positive definite, we have  $\sigma_k d_k^T A d_k > 0$  for all nonzero  $\sigma_k$  and  $d_k$ . In order to make  $Q(x_{k+1})$  smaller than  $Q(x_k)$ , we must pick  $\sigma_k$  and  $d_k$  such that  $\sigma_k(d_k, r_k) > 0$ .

In the method of *Steepest Descent* we choose  $d_k = r_k = b - Ax_k$ .  $\sigma_k = \sigma_k^*$  is chosen such that  $Q(x_{k+1})$  is as small as possible, i.e.

$$Q(x_{k+1}) = \min_{\sigma \in \mathbb{R}} Q(x_k + \sigma d_k).$$



Differentiating with respect to  $\sigma_k$  in (5.18) and setting the right-hand side equal to zero, we find

$$\sigma_k^* = \frac{(d_k, r_k)}{(d_k, Ad_k)}. \quad (5.19)$$

This value of  $\sigma_k$  is called *optimal* with respect to  $d_k$ .

The method of steepest descent will converge very slowly if  $A$  is ill-conditioned. For then the ratio of the smallest and largest eigenvalue becomes large and the paraboloid becomes very distorted. In this case the residuals need not point in the direction of the minimum.

Consider now the conjugate gradient method. Here we choose  $A$ -orthogonal search directions  $d_k = p_k$ . Since by (??)  $x_{k+1} = x_k + \alpha_k p_k$  where  $\alpha_k = (p_k, r_k)/(p_k, Ap_k)$ , we see that the step length  $\alpha_k$  is optimal with respect to  $p_k$ . It can also be shown that

$$Q(x_{k+1}) = \min_{w \in W_{k+1}} Q(x_0 + w) \quad (5.20)$$

## 5.4 Convergence

The main result in this section is the following theorem.

**Theorem 5.10** *Suppose we apply the conjugate gradient method to a positive definite system  $Ax = f$ . Then the  $A$ -norms of the errors satisfy*

$$\frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k, \quad \text{for } k \geq 0,$$

where  $\kappa = \text{cond}_2(A) = \lambda_{\max}/\lambda_{\min}$  is the 2-norm condition number of  $A$ .

This theorem explains what we observed in the previous section. Namely that the number of iterations is linked to  $\sqrt{\kappa}$ , the square root of the condition number of  $A$ . Indeed, the following corollary gives an upper bound for the number of iterations in terms of  $\sqrt{\kappa}$ .

**Corollary 5.11** *If for some  $\epsilon > 0$  we have  $k \geq \frac{1}{2} \ln(\frac{2}{\epsilon})\sqrt{\kappa}$  then  $\|x - x_k\|_A/\|x - x_0\|_A \leq \epsilon$ .*

To prove Theorem 5.10 we first convert (5.2) into a minimization problem for algebraic polynomials. This will follow from the following best approximation characterization of the iterates in the conjugate gradient method

**Lemma 5.12** *In the conjugate gradient method we have for  $k \in \mathbb{N}$*

$$\|x - x_k\|_A = \min_{w \in W_k} \|x - x_0 - w\|_A. \quad (5.21)$$

**Proof** Suppose  $w \in \mathbb{W}_k$  and define  $y := x_k - x_0 - w$ . Then  $y \in \mathbb{W}_k$  so  $\langle x - x_k, y \rangle = 0$ . But then

$$\begin{aligned} \|x - x_0 - w\|_A^2 &= \langle x - x_k + y, x - x_k + y \rangle \\ &= \langle x - x_k, x - x_k \rangle + 2\langle x - x_k, y \rangle + \langle y, y \rangle \\ &= \|x - x_k\|_A^2 + \|y\|_A^2 \geq \|x - x_k\|_A^2. \end{aligned}$$

□

We let  $\Pi_k$  denote the class of univariate polynomials of degree  $\leq k$  with real coefficients.

**Theorem 5.13** *Suppose  $Ax = f$  where  $A \in \mathbb{R}^{n,n}$  is positive definite with eigenvalues  $\lambda_1, \dots, \lambda_n$  and corresponding orthonormal eigenvectors  $u_1, \dots, u_n$ . If  $x_k$  solves the minimization problem (??) then*

$$\|x - x_k\|_A^2 = \min_{\substack{Q \in \Pi_k \\ Q(0)=1}} \sum_{j=1}^n \frac{\sigma_j^2}{\lambda_j} Q(\lambda_j)^2, \quad (5.22)$$

where the  $\sigma_j$ 's are the coefficients when  $f$  is expanded in terms of the basis of eigenvectors of  $A$ ,  $f = \sum_{j=1}^n \sigma_j u_j$ .

**Proof** If  $w \in \mathbb{W}_k = \text{span}(f, Af, \dots, A^{k-1}f)$  then for some  $a_0, \dots, a_{k-1}$

$$w = \sum_{j=0}^{k-1} a_j A^j f = P(A)f,$$

where

$$P(A) = a_0 I + a_1 A + a_2 A^2 + \dots + a_{k-1} A^{k-1}$$

is a matrix polynomial corresponding to the ordinary polynomial  $P(t) = a_0 + a_1 t + \dots + a_{k-1} t^{k-1}$  of degree  $\leq k-1$ . Then

$$\begin{aligned} \|x - w\|_A^2 &= \langle x - w, A(x - w) \rangle \\ &= \langle A^{-1}(f - Aw), f - Aw \rangle \\ &= \langle A^{-1}(f - AP(A)f), f - AP(A)f \rangle \\ &= \langle A^{-1}Q(A)f, Q(A)f \rangle, \end{aligned} \quad (5.23)$$

where  $Q(A) = I - AP(A)$  is another matrix polynomial corresponding to the polynomial  $Q(t) = 1 - tP(t)$ . Observe that  $Q \in \Pi_k$  and  $Q(0) = 1$ . Using the eigenvector expansion for  $f$  we obtain

$$Q(A)f = \sum_{j=1}^n \sigma_j Q(A)u_j = \sum_{j=1}^n \sigma_j Q(\lambda_j)u_j. \quad (5.24)$$

To show the last equality we note that  $Au_j = \lambda_j u_j$ ,  $A^2 u_j = A(Au_j) = \lambda_j^2 u_j$ ,  $A^3 u_j = \lambda_j^3 u_j$ , etc. Therefore, if  $Q(t) = \sum_{i=0}^k \alpha_i t^i$  then

$$Q(A)u_j = \sum_{i=0}^k \alpha_i A^i u_j = \sum_{i=0}^k \alpha_i \lambda_j^i u_j = Q(\lambda_j)u_j.$$

Moreover,

$$A^{-1}Q(A)u_j = Q(\lambda_j)A^{-1}u_j = \frac{Q(\lambda_j)}{\lambda_j}u_j. \quad (5.25)$$

Combining (5.23), (5.24), and (5.25) we find

$$\begin{aligned} \|x - w\|_A^2 &= (A^{-1}Q(A)f, Q(A)f) \\ &= \left( \sum_{i=1}^n \sigma_i \frac{Q(\lambda_i)}{\lambda_i} u_i, \sum_{j=1}^n \sigma_j Q(\lambda_j) u_j \right) \\ &= \sum_{i,j} \sigma_i \sigma_j \frac{Q(\lambda_i)Q(\lambda_j)}{\lambda_i} (u_i, u_j) = \sum_{j=1}^n \sigma_j^2 \frac{Q(\lambda_j^2)}{\lambda_j}. \end{aligned}$$

Minimizing over  $w$  is the same as minimizing over all  $Q \in \Pi_k$  with  $Q(0) = 1$  and the proof is complete.  $\square$

We will use the following weaker form of Theorem 5.13 to estimate the rate of convergence.

**Corollary 5.14** *Suppose  $[a, b]$  with  $0 < a < b$  is an interval containing all the eigenvalues of  $A$ . Then for all  $Q \in \Pi_k$  with  $Q(0) = 1$  we have*

$$\frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq \max_{a \leq x \leq b} |Q(x)|.$$

**Proof** In the proof of Theorem 5.13 we showed that to each  $w \in \mathbb{W}_k$  there corresponds a polynomial  $Q \in \Pi_k$  with  $Q(0) = 1$  such that

$$\|x - w\|_A^2 = \sum_{j=1}^n \sigma_j^2 \frac{Q(\lambda_j)^2}{\lambda_j}.$$

Taking  $w = x_0$  we find  $\|x - x_0\|_A^2 = \sum_{j=1}^n \frac{\sigma_j^2}{\lambda_j}$ . Therefore, by Theorem 5.13 for any  $w \in \mathbb{W}_k$

$$\|x - x_k\|_A^2 \leq \|x - w\|_A^2 \leq \max_{a \leq x \leq b} |Q(x)|^2 \sum_{j=1}^n \frac{\sigma_j^2}{\lambda_j} = \max_{a \leq x \leq b} |Q(x)|^2 \|x - x_0\|_A^2$$

and the result follows by taking square roots.  $\square$

We will apply Corollary 5.14 with  $Q(x)$  a suitably shifted and normalized version of the Chebyshev polynomial. Recall that the Chebyshev polynomial of degree  $n$  is defined recursively by

$$T_{n+1}(t) = 2tT_n(t) - T_{n-1}(t), \quad n \geq 1$$

starting with  $T_0(t) = 1$  and  $T_1(t) = t$ . Thus  $T_2(t) = 2t^2 - 1$ ,  $T_3(t) = 4t^3 - 3t$  etc. In general  $T_n$  is a polynomial of degree  $n$ . There are some convenient closed form expressions for  $T_n$ .

**Lemma 5.15** For  $n \geq 0$

1.  $T_n(t) = \cos(n \arccos t)$  for  $t \in [-1, 1]$ ,
2.  $T_n(t) = \frac{1}{2}[(t + \sqrt{t^2 - 1})^n + (t + \sqrt{t^2 - 1})^{-n}]$  for  $|t| \geq 1$ .

**Proof 1.** With  $P_n(t) = \cos(n \arccos t)$  we have  $P_n(t) = \cos n\phi$ , where  $t = \cos \phi$ . Therefore

$$P_{n+1}(t) + P_{n-1}(t) = \cos(n+1)\phi + \cos(n-1)\phi = 2\cos\phi \cos n\phi = 2tP_n(t)$$

and it follows that  $P_n$  satisfies the same recurrence relation as  $T_n$ . Since  $P_0 = T_0$  and  $P_1 = T_1$  we have  $P_n = T_n$  for all  $n \geq 0$ .

2. Fix  $t$  with  $|t| \geq 1$  and define  $P_n(t) = \frac{1}{2}(\alpha^n + \alpha^{-n})$  with  $\alpha = t + \sqrt{t^2 - 1}$ . We find  $P_0 = T_0$  and  $P_1 = T_1$  and it is enough to show that  $P_n$  satisfies the same recurrence relation as  $T_n$ . Now

$$\begin{aligned} 2tP_n(t) - P_{n-1}(t) &= 2t \frac{1}{2}(\alpha^n + \alpha^{-n}) - \frac{1}{2}(\alpha^{n-1} + \alpha^{1-n}) \\ &= \frac{1}{2}\alpha^{n+1} \left( \frac{2t}{\alpha} - \frac{1}{\alpha^2} \right) + \frac{1}{2}\alpha^{-n-1} (2t\alpha - \alpha^2) = P_n(t), \end{aligned}$$

since each factor in parenthesis is equal to one. For this we observe that  $\alpha$  is a root of the quadratic equation  $x^2 - 2tx + 1 = 0$ .  $\square$

**Proof of Theorem 5.10.**

**Proof** Let  $\lambda_1, \dots, \lambda_n$  be the eigenvalues of  $A$  and let  $k \geq 0$ . We apply Corollary 5.14 with  $a = \min \lambda_j$ ,  $b = \max \lambda_j$ , and

$$Q(x) = T_k \left( \frac{b+a-2x}{b-a} \right) / T_k \left( \frac{b+a}{b-a} \right). \quad (5.26)$$

Note that  $Q$  is admissible since  $Q \in \Pi_k$  with  $Q(0) = 1$ . By Lemma 5.15

$$\max_{a \leq x \leq b} \left| T_k \left( \frac{b+a-2x}{b-a} \right) \right| = \max_{-1 \leq t \leq 1} |T_k(t)| = 1. \quad (5.27)$$

Moreover with  $t = (b + a)/(b - a)$  we have

$$t + \sqrt{t^2 - 1} = \frac{\sqrt{\kappa} + 1}{\sqrt{\kappa} - 1}, \quad \kappa = b/a.$$

Thus again by Lemma 5.15 we find

$$T_k \left( \frac{b + a}{b - a} \right) = \frac{1}{2} \left[ \left( \frac{\sqrt{\kappa} + 1}{\sqrt{\kappa} - 1} \right)^k + \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \right] \geq \frac{1}{2} \left( \frac{\sqrt{\kappa} + 1}{\sqrt{\kappa} - 1} \right)^k. \quad (5.28)$$

Using (5.27) and (5.28) in (5.26) completes the proof.  $\square$

### Proof of Corollay 5.11.

**Proof** The inequality

$$\frac{x - 1}{x + 1} < e^{-2/x} \quad \text{for } x > 1 \quad (5.29)$$

follows from the familiar series expansion of the exponential function. Indeed, with  $y = 1/x$  we find

$$e^{2/x} = e^{2y} = \sum_{k=0}^{\infty} \frac{(2y)^k}{k!} < 1 + 2 \sum_{k=1}^{\infty} y^k = \frac{1 + y}{1 - y} = \frac{x + 1}{x - 1}$$

and (5.29) follows. By Theorem 5.10 we then find

$$\frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k < 2e^{-2k/\sqrt{\kappa}}.$$

Solving the inequality  $2e^{-2k/\sqrt{\kappa}} < \epsilon$  leads immediatly to the result.  $\square$

## 5.5 Preconditioning

For problems  $Ax = f$  of size  $n$  where both  $n$  and  $\text{cond}_2(A)$  are large it is often possible to improve the performance of the conjugate gradient method by using a technique known as **pre-conditioning**. Instead of  $Ax = f$  we consider an equivalent system  $BAx = Bf$ , where  $B$  is nonsingular and  $\text{cond}_2(BA)$  is smaller than  $\text{cond}_2(A)$ . We cannot use CG on  $BAx = Bf$  directly since  $BA$  in general is not symmetric even if both  $A$  and  $B$  are. But if  $B$  is positive definite then we can apply CG to a symmetrized system and then transform the recurrence formulae to an iterative method for the original system  $Ax = f$ . This iterative method is known as the **pre-conditioned conjugate gradient method**. We shall see that the convergence properties of this method is determined by the eigenvalues of  $BA$ .

Suppose  $B$  is positive definite. By Theorem 2.12 there is a nonsingular matrix  $C$  such that  $B = C^T C$ . ( $C$  is only needed for the derivation and will never be computed). Now

$$BAx = Bf \Leftrightarrow C^T(CAC^T)C^{-T}x = C^T Cf \Leftrightarrow (CAC^T)y = Cf, \text{ \& } x = C^T y.$$

We have 3 linear systems

$$Ax = f \tag{5.30}$$

$$BAx = Bf \tag{5.31}$$

$$(CAC^T)y = Cf, \text{ \& } x = C^T y. \tag{5.32}$$

In addition to being positive definite the matrix  $CAC^T$  is similar to  $BA$ . Indeed,

$$C^T(CAC^T)C^{-T} = BA.$$

Thus  $CAC^T$  and  $BA$  have the same eigenvalues. Therefore if we apply the conjugate gradient method to (5.32) then the rate of convergence will be determined by the eigenvalues of  $BA$ .

By Lemma 5.5 the conjugate gradient method applied to  $(CAC^T)y = Cf$  can be written

$$\begin{aligned} y_{k+1} &= y_k + \alpha_k q_k, & \alpha_k &= z_k^T z_k / q_k^T (CAC^T) q_k, \\ z_{k+1} &= z_k - \alpha_k (CAC^T) q_k, \\ q_{k+1} &= z_{k+1} + \beta_k q_k, & \beta_k &= z_{k+1}^T z_{k+1} / z_k^T z_k. \end{aligned}$$

Here

$$z_k = Cf - CAC^T y_k$$

is the residual. With

$$x_k = C^T y_k, \quad p_k = C^T q_k, \quad s_k = C^T z_k, \quad r_k = C^{-1} z_k \tag{5.33}$$

this can be transformed into

$$x_{k+1} = x_k + \alpha_k p_k, \quad \alpha_k = s_k^T r_k / p_k^T A p_k, \tag{5.34}$$

$$r_{k+1} = r_k - \alpha_k A p_k, \tag{5.35}$$

$$s_{k+1} = s_k - \alpha_k B A p_k, \tag{5.36}$$

$$p_{k+1} = s_{k+1} + \beta_k p_k, \quad \beta_k = s_{k+1}^T r_{k+1} / s_k^T r_k. \tag{5.37}$$

Here  $x_k$  will be an approximation to the solution  $x$  of  $Ax = b$ ,  $r_k = f - Ax_k$  is the residual in the original system and  $s_k = Bf - BAx_k$  is the residual in the preconditioned system. This follows since by (5.33)

$$r_k = C^{-1} z_k = C^{-1}(Cf - CAC^T y_k) = f - Ax_k$$

and  $s_k = C^T z_k = C^T C r_k = B r_k$ . We now have the following preconditioned conjugate gradient algorithm for obtaining an approximation  $x_k$  to the solution of a positive definite system  $Ax = b$ .

**Algorithm 5.16**

1. Choose a starting vector  $x_0$  (for example  $x_0 = 0$ )
2.  $r_0 = f - Ax_0$ ,  $p_0 = s_0 = Br_0$
3.  $\rho_0 = s_0^T r_0$ ;  $k = 0$
4. while  $\sqrt{\rho_k/\rho_0} > \epsilon$  &  $k < kmax$ 
  - 4.1a  $t_k = Ap_k$
  - 4.1b  $w_k = Bt_k$
  - 4.2  $\alpha_k = \rho_k/p_k^T t_k$
  - 4.3  $x_{k+1} = x_k + \alpha_k p_k$
  - 4.4a  $r_{k+1} = r_k - \alpha_k t_k$  ( $r_k = f - Ax_k$ )
  - 4.4b  $s_{k+1} = s_k - \alpha_k w_k$  ( $s_k = Bf - BAx_k$ )
  - 4.5  $\rho_{k+1} = s_{k+1}^T r_{k+1}$
  - 4.6  $p_{k+1} = s_{k+1} + \frac{\rho_{k+1}}{\rho_k} p_k$
  - 4.7  $k = k + 1$

This algorithm is quite similar to Algorithm 5.7. The main additional work is contained in statement 4.1b. We'll discuss this further in connection with an example.

We have the following convergence result for this algorithm.

**Theorem 5.17** *Suppose we apply a positive definite preconditioner  $B$  to the positive definite system  $Ax = f$ . Then the quantities  $x_k$  computed in Algorithm 5.16 satisfy the following bound:*

$$\frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k, \quad \text{for } k \geq 0,$$

where  $\kappa = \lambda_{max}/\lambda_{min}$  is the ratio of the largest and smallest eigenvalue of  $BA$ .

**Proof** Since Algorithm 5.16 is equivalent to solving (5.32) by the conjugate gradient method Theorem 5.10 implies that

$$\frac{\|y - y_k\|_{CAC^T}}{\|y - y_0\|_{CAC^T}} \leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k, \quad \text{for } k \geq 0,$$

where  $y_k$  is the conjugate gradient approximation to the solution  $y$  of (5.32) and  $\kappa$  is the ratio of the largest and smallest eigenvalue of  $CAC^T$ . Since  $BA$

and  $CAC^T$  are similar this is the same as the  $\kappa$  in the theorem. By (5.33) we have

$$\|y-y_k\|_{CAC^T}^2 = (y-y_k, CAC^T(y-y_k)) = (C^T(y-y_k), AC^T(y-y_k)) = \|x-x_k\|_A^2$$

and the proof is complete.  $\square$

We conclude that  $B$  should satisfy the following requirements for a problem of size  $n$ :

1. The eigenvalues of  $BA$  should be located in a narrow interval. Preferably one should be able to bound the length of the interval independently of  $n$ .
2. The evaluation of  $Bx$  for a given vector  $x$  should not be expensive in storage and flops, ideally  $O(n)$  for both.

## 5.6 Preconditioning Example

Throughout this section we use the same grid and notation as in Section 2.4. Let  $h = 1/(m+1)$ .

We recall the Poisson problem

$$-\Delta u = -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f(x, y) \quad \text{for } (x, y) \in \Omega = (0, 1)^2 \quad (5.38)$$

$$u = 0 \quad \text{on } \partial\Omega,$$

where  $f$  is a given function,  $\Omega$  is the unit square in the plane, and  $\partial\Omega$  is the boundary of  $\Omega$ . For numerical solution we have the **discrete Poisson problem** which can either be written as a matrix equation

$$\begin{aligned} h^2 f_{j,k} &= 4v_{j,k} - v_{j-1,k} - v_{j+1,k} - v_{j,k-1} - v_{j,k+1}, & j, k &= 1, \dots, m \\ v_{0,k} &= v_{m+1,k} = v_{j,0} = v_{j,m+1} = 0, & j, k &= 0, 1, \dots, m+1, \end{aligned}$$

or as a system  $A_p x = b$ , where  $x = \text{vec}(v_{i,j})$ ,  $b = h^2 \text{vec}(f_{i,j})$  and the entries  $a_{i,j}$  of  $A_p$  are given by

$$\begin{aligned} a_{ii} &= 4, & i &= 1, \dots, n \\ a_{i+1,i} = a_{i,i+1} &= -1, & i &= 1, \dots, n-1, \quad i \neq m, 2m, \dots, (m-1)m \\ a_{i+m,i} = a_{i,i+m} &= -1, & i &= 1, \dots, n-m \\ a_{ij} &= 0, & & \text{otherwise.} \end{aligned}$$

We also considered a problem with variable coefficients

$$\begin{aligned} -\frac{\partial}{\partial x} (c(x, y) \frac{\partial u}{\partial x}) - \frac{\partial}{\partial y} (c(x, y) \frac{\partial u}{\partial y}) &= f(x, y) & (x, y) &\in \Omega = (0, 1)^2 \\ u(x, y) &= 0 & (x, y) &\in \partial\Omega, \end{aligned} \quad (5.39)$$



where  $c(x, y)$  is positive and continuous on  $\Omega$ .

Using finite difference approximations we obtained a discrete analog of (5.39)

$$\begin{aligned} -(P_h v)_{j,k} &= h^2 f_{j,k} & j, k = 1, \dots, m \\ v_{j,k} &= 0 & j = 0, m+1 \text{ all } k \text{ or } k = 0, m+1 \text{ all } j, \end{aligned} \quad (5.40)$$

where

$$\begin{aligned} -(P_h v)_{j,k} &= (c_{j,k-\frac{1}{2}} + c_{j-\frac{1}{2},k} + c_{j+\frac{1}{2},k} + c_{j,k+\frac{1}{2}})v_{j,k} \\ &\quad - c_{j,k-\frac{1}{2}}v_{j,k-1} - c_{j-\frac{1}{2},k}v_{j-1,k} - c_{j+\frac{1}{2},k}v_{j+1,k} - c_{j,k+\frac{1}{2}}v_{j,k+1}. \end{aligned} \quad (5.41)$$

Here  $c_{p,q} = c(ph, qh)$ ,  $f_{j,k} = f(jh, kh)$  and  $v_{j,k} \approx u(jh, kh)$ .

The corresponding linear system can be written  $Ax = f$  where the  $n$ -by- $n$  coefficient matrix is given by

$$\begin{aligned} a_{i,i} &= \gamma_{i-\frac{m}{2}} + \gamma_{i-\frac{1}{2}} + \gamma_{i+\frac{1}{2}} + \gamma_{i+\frac{m}{2}}, & i = 1, 2, \dots, n \\ a_{i+1,i} &= a_{i,i+1} = -\gamma_{i+\frac{1}{2}}, & i = 1, 2, \dots, n-1; i \bmod m \neq 0 \\ a_{i+m,i} &= a_{i,i+m} = -\gamma_{i+\frac{m}{2}}, & i = 1, 2, \dots, n-m \\ a_{i,j} &= 0 & \text{otherwise,} \end{aligned} \quad (5.42)$$

and  $\gamma_{j+(k-1)m} = c_{j,k}$ .

The matrix  $A$  is symmetric and positive definite.

If we choose  $c(x, y) \equiv 1$  in (5.39), we get the Poisson problem (5.38). With this in mind, we may think of the coefficient matrix  $A_p$  arising from the discretization of the Poisson problem as an approximation to the matrix (5.42). This suggests using  $B = A_p^{-1}$ , the inverse of the discrete Poisson matrix as a preconditioner for the system (5.40).

Consider Algorithm 5.16. With this preconditioner Statement 4.1b can be written  $A_p w_k = t_k$ .

In Section 4.1 we developed a direct method based on diagonalization for solving the discrete Poisson problem on a rectangular domain. This method can be utilized to solve  $A_p w_k = t_k$ .

Consider the specific problem where

$$c(x, y) = e^{-x+y} \text{ and } f(x, y) = 1.$$

We have used Algorithm 5.7 (conjugate gradient without preconditioning), and Algorithm 5.16 (conjugate gradient with preconditioning) to solve the problem (5.39). We used  $x_0 = 0$  and  $\epsilon = 10^{-8}$ . The results are shown in Table 5.18.

Without preconditioning the number of iterations still seems to be more or less proportional to  $\sqrt{n}$  although the convergence is slower than for the constant coefficient problem. Using preconditioning speeds up the convergence considerably. The number of iterations appears to be bounded independently of  $n$ . This illustrates that preconditioning is needed when solving nontrivial problems.

$n$	2500	10000	22500	40000	62500
$K$	222	472	728	986	1246
$K/\sqrt{n}$	4.44	4.72	4.85	4.93	4.98
$K_{pre}$	22	23	23	23	23

Table 5.18: The number of iterations  $K$  (no preconditioning) and  $K_{pre}$  (with preconditioning) for the problem (5.39) using the discrete Poisson problem as a preconditioner.

Using a preconditioner increases the work in each iteration. For the present example the number of flops in each iteration changes from  $O(n)$  without preconditioning to  $O(n^{3/2})$  or  $O(n \log_2 n)$  with preconditioning. This is not a large increase and both the number of iterations and the computing time is reduced drastically.

Let us finally show that the number  $\kappa = \lambda_{max}/\lambda_{min}$  which determines the rate of convergence for the preconditioned conjugate gradient method applied to (5.39) can be bounded independently of  $n$ .

**Theorem 5.19** *Suppose  $0 < c_0 \leq c(x, y) \leq c_1$  for all  $(x, y) \in [0, 1]^2$ . For the eigenvalues of the matrix  $BA = A_p^{-1}A$  just described we have*

$$\kappa = \frac{\lambda_{max}}{\lambda_{min}} \leq \frac{c_1}{c_0}.$$

**Proof** Suppose  $A_p^{-1}Ax = \lambda x$  for some  $x \in \mathbb{R}^n \setminus \{0\}$ . Then  $Ax = \lambda A_p x$ . Multiplying this by  $x^T$  and solving for  $\lambda$  we find

$$\lambda = \frac{x^T Ax}{x^T A_p x}.$$

We computed  $x^T Ax$  in (??) and we obtain  $x^T A_p x$  by setting all the  $c$ 's there equal to one

$$x^T A_p x = \sum_{i=1}^m \sum_{j=0}^m (v_{i,j+1} - v_{i,j})^2 + \sum_{j=1}^m \sum_{i=0}^m (v_{i+1,j} - v_{i,j})^2.$$

Thus  $x^T A_p x > 0$  and bounding all the  $c$ 's in (??) from below by  $c_0$  and above by  $c_1$  we find

$$c_0(x^T A_p x) \leq x^T Ax \leq c_1(x^T A_p x)$$

which implies that  $c_0 \leq \lambda \leq c_1$  for all eigenvalues  $\lambda$  of  $BA = A_p^{-1}A$ .  $\square$

Using  $c(x, y) = e^{-x+y}$  as above, we find  $c_0 = e^{-2}$  and  $c_1 = 1$ . Thus  $\kappa \leq e^2 \approx 7.4$ , a quite acceptable matrix condition which explains the convergence results from our numerical experiment.

## 5.7 Problems

**5.1** Do one iteration with the conjugate gradient method when  $x_0 = 0$ .  
(Answer:  $x_1 = \frac{\langle x, f \rangle}{\langle f, f \rangle} f = \frac{f^T f}{f^T A f} f$ .)

**5.2** Do two conjugate gradient iterations for the system

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

starting with  $x_0 = 0$ .

**5.3** Consider the linear system  $Ax = f$  where

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \quad \text{and} \quad f = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix}.$$

a) Determine the vectors defining the Krylow spaces for  $k \leq 3$ . Answer:

$$(f, Af, A^2 f) = \begin{pmatrix} 4 & 8 & 20 \\ 0 & -4 & -16 \\ 0 & 0 & 4 \end{pmatrix}.$$

b) Carry out three CG-iterations on  $Ax = f$ . Answer:

$$(x_0, x_1, x_2, x_3) = \begin{pmatrix} 0 & 2 & 8/3 & 3 \\ 0 & 0 & 4/3 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (r_0, r_1, r_2, r_3) = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4/3 & 0 \end{pmatrix},$$

$$(Ap_0, Ap_1, Ap_2) = \begin{pmatrix} 8 & 0 & 0 \\ -4 & 3 & 0 \\ 0 & -2 & 16/9 \end{pmatrix}, \quad (p_0, p_1, p_2, p_3) = \begin{pmatrix} 4 & 1 & 4/9 & 0 \\ 0 & 2 & 8/9 & 0 \\ 0 & 0 & 12/9 & 0 \end{pmatrix},$$

c) Show that

- $\dim(\mathbb{W}_k) = k$  for  $k = 0, 1, 2, 3$
- $x_3$  is the exact solution of  $Ax = f$
- $r_0, \dots, r_{k-1}$  is an orthogonal basis for  $\mathbb{W}_k$  for  $k = 1, 2, 3$
- $p_0, \dots, p_{k-1}$  is an  $A$ -orthogonal basis for  $\mathbb{W}_k$  for  $k = 1, 2, 3$
- $\{\|r_k\|\}$  is monotonically decreasing

**5.4** Show that  $r_k^T r_{k+1} = 0$  in the method of steepest descent and conclude that successive residuals  $r_j = b - Ax_j$  are orthogonal.

**5.5** In the conjugate gradient method show that

$$\|x - x_{k+1}\|_A \leq \|x - x_k\|_A, \quad k \geq 1.$$

**5.6** Let  $Q(x) = x^T Ax - 2b^T x$  have a minimum at  $x^* \in \mathbb{R}^n$ .

a) Show that  $Q(x) = \|x^* - x\|_A^2 - \|x^*\|_A^2$  for any  $x \in \mathbb{R}^n$ .

b) Show (5.20).

**5.7** Consider solving the least squares problem by using the conjugate gradient method on the normal equations  $A^T Ax = A^T b$ . Show that only steps 2, 4.2 and 4.4 in algorithm 5.7 have to be modified. In particular, we replace the steps 2, 4.2 and 4.4 by

$$2. p_0 = r_0 = A^T(b - Ax_0)$$

$$4.2 \alpha_k = \rho_k / (t_k, t_k)$$

$$4.4 r_{k+1} = r_k - \alpha_k A^T t_k$$

Note that the condition number of the normal equations is  $\text{cond}_2(A)^2$ .

**5.8** Study the following proof of Lemma 5.6.

Set

$$\rho_j := \|r_j\|^2 \quad \text{and} \quad \pi_j := \|p_j\|_A^2, \quad j \geq 0$$

and let  $m$  be the smallest integer such that  $\|e_m\| = 0$ . Since  $p_j \neq 0$  for  $j \leq k$  we have  $\dim \mathbb{W}_{k+1} = k + 1$  which implies that  $r_k \neq 0$  and hence  $m > k$ . For  $j < m$

$$x_{j+1} = x_j + \alpha_j p_j = x_{j-1} + \alpha_{j-1} p_{j-1} + \alpha_j p_j = \cdots = x_0 + \sum_{i=0}^j \alpha_i p_i$$

so that

$$e_j = x_m - x_j = \sum_{i=j}^{m-1} \alpha_i p_i, \quad \alpha_i = \frac{\rho_i}{\pi_i}. \quad (5.43)$$

For  $j > k$

$$(p_j, p_k) = (r_j + \beta_{j-1} p_{j-1}, p_k) = \beta_{j-1} (p_{j-1}, p_k) = \cdots = \beta_{j-1} \cdots \beta_k (p_k, p_k)$$

and since  $\beta_{j-1} \cdots \beta_k = \rho_j / \rho_k$  we obtain

$$(p_j, p_k) = \frac{\rho_j}{\rho_k} (p_k, p_k), \quad j \geq k. \quad (5.44)$$

By  $A$ -orthogonality and (5.43)

$$\|e_j\|_A^2 = \left\langle \sum_{i=j}^{m-1} \alpha_i p_i, \sum_{i=j}^{m-1} \alpha_i p_i \right\rangle = \sum_{i=j}^{m-1} \alpha_i^2 \pi_i = \sum_{i=j}^{m-1} \frac{\rho_i^2}{\pi_i}. \quad (5.45)$$

Now

$$\begin{aligned}\|e_k\|^2 &= \|e_{k+1} + x_{k+1} - x_k\|^2 = \|e_{k+1} + \alpha_k p_k\|^2 \\ &= \|e_{k+1}\|^2 + \alpha_k (2(p_k, e_{k+1}) + \alpha_k \|p_k\|^2).\end{aligned}\tag{5.46}$$

and moreover

$$\begin{aligned}& \alpha_k (2(p_k, e_{k+1}) + \alpha_k \|p_k\|^2) \\ \stackrel{(5.43)}{=} & \alpha_k \left( 2 \sum_{j=k+1}^{m-1} \alpha_j (p_j, p_k) + \alpha_k \|p_k\|^2 \right) \\ \stackrel{(5.44)}{=} & \alpha_k \left( 2 \sum_{j=k+1}^{m-1} \alpha_j \frac{\rho_j}{\rho_k} \|p_k\|^2 + \alpha_k \|p_k\|^2 \right) \\ = & \frac{\|p_k\|^2}{\pi_k} \left( \sum_{j=k}^{m-1} \frac{\rho_j^2}{\pi_j} + \sum_{j=k+1}^{m-1} \frac{\rho_j^2}{\pi_j} \right) \\ \stackrel{(5.45)}{=} & \frac{\|p_k\|^2}{\pi_k} (\|e_k\|_A^2 + \|e_{k+1}\|_A^2).\end{aligned}$$

Inserting this in (5.46) proves the lemma.  $\square$



# Bibliography

- [1] Golub, G. H., and C. F. Van Loan, *Matrix Computations*, John Hopkins University Press, Baltimore, MD, third edition, 1996.
- [2] Horn, Roger A: and Charles R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
- [3] Leon, Steven J., *Linear Algebra with Applications*, Prentice Hall, NJ, Sixth Edition, 2002.
- [4] Meyer, Carl D., *Matrix Analysis and Applied Linear Algebra* , Siam Philadelphia, 2000.
- [5] Stewart, G. G., *Matrix Algorithms Volume I: Basic Decompositions*, Siam Philadelphia, 1998.
- [6] Stewart, G. G., *Matrix Algorithms Volume II: Eigensystems*, Siam Philadelphia, 2001.
- [7] Trefethen, Lloyd N., and David Bau III, *Numerical Linear Algebra*, Siam Philadelphia, 1997.
- [8] Tveito, A., and R. Winther, *Partial Differential Equations*, Springer, Berlin.
- [9] Van Loan, Charles, *Computational Frameworks for the Fast Fourier Transform*, Siam Philadelphia, 1992.
- [10] Wilkinson, J. H., *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

# Index

- Poisson matrix, 22
- A-norm, 42
- A-orthogonal, 42
- averaging matrix, 26
- averaging problem, 46
- banded cholesky algorithm, 13
- banded LDLT factorization, 14
- bandwidth, 13
- biharmonic equation, 40
  - fast solution method, 40
- Chebyshev polynomial, 52
- cholesky-factorization, 11
- conjugate gradient method, 41
  - algorithm, 45
  - convergence, 49
  - derivation, 41
  - least squares problem, 60
  - preconditioning, 53
  - preconditioning algorithm, 55
  - preconditioning convergence, 55
- Discrete Fourier Transform, 34
- discrete Poisson Problem, 46
- Discrete Sine Transform, 33
- energy norm, 42
- Fast Fourier Transform, 35
- finite difference method, 17
- five point stencil, 21
- flops, 5
- Fourier matrix, 34
- Gaussian elimination, 6
- gradient, 10
- Gram-Schmidt orthogonalization, 43
- Hessian, 10
- identity matrix, vii
- inverse matrix, vii
- invertible matrix, vii
- Kronecker product, 22
  - eigenvalues, 24
  - eigenvectors, 24
  - inverse, 25
  - mixed product rule, 24
  - nonsingular, 25
  - positive definite, 25
  - symmetry, 25
  - transpose, 23
- Kronecker sum, 23
  - eigenvalues, 25
  - eigenvectors, 25
  - nonsingular, 25
  - positive definite, 25
  - symmetry, 25
- Krylov subspace, 42
- LDLT-factorization, 5
- leading principal submatrix, 3
- left Kronecker product, 23
- LLT factorization, 11
- lower bandwidth, 13
- LU-factorization, 3
- minimization, 47
- mixed product rule, 24
- negative (semi-)definite, 9
- nonsingular matrix, vii



paraboloid, 48  
Poisson matrix, 22  
Poisson Problem, 20  
    nine point scheme, 39  
positive definite, 9  
positive semidefinite, 9  
  
quadratic form, 9  
  
recursive FFT, 37  
right Kronecker product, 23  
  
second derivative matrix, 18  
Simple Fast Poisson Solver, 33  
singular matrix, vii  
steepest descent, 48  
stencil, 21  
  
translated subspace, 42  
tridiagonal, 13  
  
upper bandwidth, 13  
  
variable coefficient problem, 28  
variable coefficients, 27, 56  
vectorization, 21