SYMMETRIC INDEFINITE MATRICES: LINEAR SYSTEM SOLVERS AND MODIFIED INERTIA PROBLEMS

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Science and Engineering

January 1998

By

Sheung Hun Cheng

Department of Mathematics

Contents

List of Tables

List of Figures

Copyright

Abstract

Symmetric indefinite matrices are an important class of matrices arising in many applications. Some practically important computations associated with this class of matrices are investigated in this thesis.

First our emphasis is on examining the accuracy and stability of the two most popular methods for solving symmetric indefinite linear systems, namely the diagonal pivoting method and Aasen's method. Suitable pivoting strategies are crucial to the stability of both methods. For the diagonal pivoting method, we assess the Bunch–Kaufman and the more recent bounded Bunch–Kaufman pivoting strategies using various stability measures. We confirm that the bounded Bunch–Kaufman pivoting strategy achieves better accuracy for a set of examples. However, theoretical analyses and experimental results show that the "superior" accuracy that has been claimed is not fully justified.

For Aasen's method, a new normwise backward stability result of Higham is stated. We derive a growth factor bound which is attainable for matrices of dimension 3. Direct search methods are employed to search for large growth factors to gain insight into the behaviour of the growth factor of Aasen's method.

Our focus is then on tackling three modified inertia problems. We propose two alternative modified Cholesky algorithms based on the two previously mentioned linear solvers, and compare their performance with the two existing algorithms of Gill, Murray and Wright, and Schnabel and Eskow, both theoretically and numerically. The experimental results show that all four algorithms are competitive. Our algorithms have the advantages of ease of implementation and the existence of a priori bounds for assessing how "good" the perturbation is.

Motivated by an application in constrained optimization, we then concentrate on deriving structured perturbations for a block 2×2 matrix A, which involves

perturbing the $(1,1)$ block so that A has a particular inertia. We derive a perturbation, valid for any unitarily invariant norm, that increases the number of nonnegative eigenvalues by a given amount. An alternative approach based on a projection into the null space of the constraints is also considered. Theoretical tools developed include an extension of Ostrowski's theorem on congruence transformations and some lemmas on inertia properties of block 2×2 matrices.

Finally the generalized Hermitian eigenvalue problem is discussed. We clear some confusion on the characteristics of the eigenvalues of Hermitian matrix products. A new concept called the inner numerical radius is introduced, using which we derive an elegant solution to the nearness problem of finding the distance from an indefinite matrix pair to the nearest definite pair in the 2-norm. An alternative approach for determining the inner numerical radius of a normal pair, which exploits the characteristics of its eigenvalues, is proposed.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

- Chapter 4 is based on the technical report "A modified Cholesky algorithm based on a symmetric indefinite factorization" (with Nicholas J. Higham) [17], 1996. This work is to appear in *SIAM J. Matrix Anal. Appl.*.
- Chapter 5 is based on the technical report "Modifying the inertia of matrices arising in optimization" (with Nicholas J. Higham) [59], 1996. This work is to appear in *Linear Algebra and Appl.*.
- Section 6.5 is based on the technical report in preparation "Definite pairs and the inner numerical radius" (with Nicholas J. Higham) [18], 1997.

Acknowledgements

In writing this thesis I have been helped and influenced by many people. I am particularly indebted to Nick Higham for his unfailing patience, constant inspiration and fruitful perturbation on my thesis. I thank him.

Special thanks to Phil Jacob for making my stay in Manchester such an enjoyable experience, and Tony Cox for numerous lively and intriguing conversations.

Thanks also goes to my Mum and Dad, Betty, Amy and Carmen for their unconditional love and support.

Amongst these wonderful people whose presence made this thesis possible, it is a pleasure to thank Gigi Chao, Frances and Shun Cheung, Clare Chiu, Phil Davies, Michaela Gruber, Wendy Hall, Kathrin Happe, Angela Henderson, Setsuko Matsumoto, Lorraine Seymour, Kath Smith, Elena Takeuchi, the Pete's Eat, the Cornerhouse cinema and the Greenhouse restaurant.

I acknowledge the Committee of Vice-Chancellors and Principals of the Universities of the United Kingdom for the support of an Overseas Research Student Award, and Hulme Hall for a Postgraduate Exhibition.

Chapter 1

Introduction

1.1 Symmetric Indefinite Matrices and Numerical Analysis

A matrix is symmetric indefinite if it is symmetric and has both positive and negative eigenvalues. Symmetric indefinite matrices are an important class of matrices arising in many applications. To name a few applications, this class of matrices arises in Newton's method for the unconstrained and constrained optimization problems [20], [31], [34], [37], [44], certain interior methods for the general nonlinear programming problem [32], [33], penalty function methods for nonlinear programming [43], the augmented system of general least squares problems [8], [16], [76], some interior methods for linear and quadratic programming problem [38], [89], and in discretized incompressible Navier–Stokes equations [79].

Apart from arising intrinsically in applications, symmetric indefinite matrices are also created from definite ones because of errors in measuring or computing the matrix elements.

We introduce the basic terminology of floating point arithmetic in Section 1.2. In Section 1.3, the model of arithmetic on which our rounding error analysis is based is defined. We also describe the computational environment for all experiments. Then a brief introduction to IEEE standard arithmetic is given in Section 1.4. We summarize a few classical perturbation theory results in Section 1.5. Finally an overview of this thesis is presented in Section 1.6.

We acknowledge that the material in Sections 1.2–1.5 has been adapted from Higham [55, Chaps. 2 and 7]. Throughout this thesis, definitions and notations

are introduced when needed.

1.2 Floating Point Arithmetic

A floating point number system $F \subset \mathbb{R}$ is a subset of the real numbers whose elements have the form

$$y = \pm m \times \beta^{e-t}. \tag{1.1}$$

The system F is characterized by four integer parameters:

- the *base* $\beta > 1$ (sometimes called the *radix*),
- the *precision* t, and
- the *exponent range* $e_{\min} \leq e \leq e_{\max}$.

The *mantissa* m is an integer satisfying $0 \leq m \leq \beta^t - 1$. To ensure a unique representation for each $y \in F$ it is assumed that $m \geq \beta^{t-1}$ if $y \neq 0$, so that the system is *normalized*. The *range* of the nonzero floating point numbers in F is given by $\beta^{e_{\min}} \leq |y| \leq \beta^{e_{\max}}(1 - \beta^{-t})$.

The system F can be extended by including *subnormal numbers* (also known as *denormalized numbers*), which in the notation of (1.1) are the numbers

$$y = \pm m \times \beta^{e_{\min}-t}, \qquad 0 < m < \beta^{t-1}.$$

It is easily seen that the subnormal numbers have fewer digits of precision than the normalized numbers.

Let $G \subset \mathbb{R}$ denote all real numbers of the form (1.1) with no restriction on the exponent e. If $x \in \mathbb{R}$ then $fl(x)$ denotes an element of G nearest to x, and the transformation $x \to fl(x)$ is called *rounding*. The discrepancy $|x - fl(x)|$ induced by this transformation is termed *rounding error*.

Although we have defined fl as a mapping onto G, we are only interested in the cases where it produces a result in F. We say that $fl(x)$ *overflows* if $|fl(x)| > \max\{|y| : y \in F\}$ and *underflows* if $0 < |fl(x)| < \min\{|y| : 0 \neq y \in F\}$. We can show that every real number x lying in the range of F can be approximated by an element of F with a relative error no larger than $u = \frac{1}{2}\beta^{1-t}$. The quantity u is called the *unit roundoff*. It is the most useful quantity associated with F and is ubiquitous in the world of rounding error analysis.

1.3 Model of Arithmetic

To carry out rounding error analysis of an algorithm we first need to make some assumptions about the accuracy of the basic arithmetic operation.

Throughout this thesis, our model of floating point arithmetic is the usual model

$$fl(x \text{ op } y) = (x \text{ op } y)(1+\delta), \qquad |\delta| \leq u, \quad \text{op} = +, -, *, /, \tag{1.2}$$

where u is the unit roundoff. We introduce the constant

$$\gamma_n = \frac{nu}{1-nu},$$

which carries with it the implicit assumption that $nu < 1$.

This model is valid for most modern computers, and, in particular, holds for those implementing the IEEE standard arithmetic with guard digits. Cases in which the model is not valid can be found in [55]. Some machines do not satisfy the model because they do not use guard digits. Note that the model (1.2) ignores the possibility of underflow and overflow.

All our algorithms and experiments were carried out in MATLAB 4.2c [65] which uses IEEE standard double precision arithmetic on those machines that

Type	Size	Mantissa	Exponent	Unit roundoff	Range
Single	32 bits	23+1 bits	8 bits	$2^{-24} \approx 5.96 \times 10^{-8}$	$10^{\pm 38}$
Double	64 bits	52+1 bits	11 bits	$2^{-53} \approx 1.11 \times 10^{-16}$	$10^{\pm 308}$

Table 1.1: Floating point formats for single and double precision in IEEE arithmetic.

support it in hardware. All the results quoted were obtained on a Sun SPARCstation which uses IEEE standard floating point arithmetic. Therefore the unit roundoff u is $2^{-53} \approx 1.1 \times 10^{-16}$ throughout this thesis.

The cost of algorithms is measured in flops. A *flop* is an elementary floating point operation: $+$, $-$, $/$, $*$. We normally state only the highest order terms of flops counts. Thus, when we say that an algorithm for $n \times n$ matrices requires $n^3/3$ flops, we really mean $n^3/3 + O(n^2)$ flops as $n \to \infty$.

1.4 IEEE Arithmetic

IEEE standard 754, published in 1985 [62], defines a binary floating point arithmetic system. It was developed by a working group of a subcommittee of the IEEE Computer Society Computer Standards Committee.

The basic design principles of the standard are that it should encourage individuals to develop robust, efficient, and portable numerical programs, enable the handling of arithmetic exceptions, and provide for the development of transcendental functions and very high precision arithmetic.

The standard specifies floating point number formats, the results of the basic floating point operations and comparisons, rounding modes, floating point exceptions and their handling, and conversion between different arithmetic formats. Square root is included as a basic operation. The standard is not concerned with exponentiation or transcendental functions such as exp and cos.

Two main floating point formats, single and double precision, are defined; see Table 1.1. In both formats one bit is reserved as a sign bit. Since the floating

Exception type	Example	Default result
Invalid operation	$0/0$, $0 \times \infty$, $\sqrt{-1}$	NaN (Not a Number)
Overflow	—	$\pm\infty$
Divide by zero	Finite nonzero/0	$\pm\infty$
Underflow	—	Subnormal numbers
Inexact	Whenever $fl(x \text{ op } y) \neq x \text{ op } y$	Correctly rounded result

Table 1.2: IEEE arithmetic exceptions and default results.

point numbers are normalized, the most significant bit is always 1 and is not stored except for subnormal numbers. The *hidden bit* accounts for the +1 in Table 1.1.

The standard specifies that all arithmetic operations are to be performed as if they were first calculated to infinite precision and then rounded according to one of four modes. The default rounding mode is to round to the nearest representable number, with rounding to even (zero at the last bit of mantissa) in the case of a tie. With this default mode, the model (1.2) is obviously satisfied. Rounding to plus or minus infinity is also supported by the standard. The fourth supported mode is rounding to zero (truncation, or chopping).

IEEE arithmetic is a closed system: every arithmetic operation produces a result, whether it is mathematically expected or not, and exceptional operations raise a signal. The default results are shown in Table 1.2. The default response to an exception is to set a flag and continue, but it is also possible to take a trap (pass control to a trap handler).

A NaN is a special bit pattern that cannot be generated in the course of unexceptional operations because it has a reserved exponent field. The mantissa is arbitrary subject to being nonzero. A NaN is generated by operations such as $0/0$, $0 \times \infty$, ∞/∞, $(+\infty) + (-\infty)$, and $\sqrt{-1}$.

Another feature is that the IEEE standard provides distinct representations for $+0$ and -0, but comparison are defined so that $+0 = -0$.

The infinity symbol is represented by a zero mantissa and the same exponent

field as a NaN; the sign bit distinguishes between $\pm\infty$. The infinity symbol obeys the usual mathematical conventions regarding infinity, such as $\infty + \infty = \infty$, $(-1) \times \infty = -\infty$, and (finite)$/\infty = 0$.

The standard also allows subnormal numbers to be represented, instead of flushing them to zero as in many systems, and this feature permits *gradual underflow*.

The floating point operation op is *monotonic* if $fl(a \text{ op } b) \leq fl(c \text{ op } d)$ whenever a, b, c, and d are floating point numbers for which a op $b \leq c$ op d and neither $fl(a \text{ op } b)$ nor $fl(c \text{ op } d)$ overflows. IEEE arithmetic is monotonic, as is any correctly rounded arithmetic. Monotonic arithmetic is important in the bisection algorithm for finding the eigenvalues of a symmetric tridiagonal matrix [27].

1.5 Perturbation Theory

The effects of rounding errors in numerical algorithms are important and have been much studied. The purpose of rounding error analysis is to show the existence of an a priori bound for some appropriate measure of the effects of rounding error on an algorithm. Whether a bound exists is the most important question.

We now present some classical perturbation results for linear systems without proof. The proofs of all the theorems can be found in Higham [55] and the references therein. Our first result makes precise the intuitive feeling that if the residual is small then we have a "good" approximate solution. In all these results, $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, and $E \in \mathbb{R}^{n\times n}$ and $f \in \mathbb{R}^n$ are a matrix and vector of nonnegative tolerances.

Theorem 1.5.1 (Rigal and Gaches) *The* normwise backward error

$$\eta_{E,f}(y) := \min\{\epsilon : (A + \Delta A)y = b + \Delta b, \quad \|\Delta A\| \leq \epsilon\|E\|, \quad \|\Delta b\| \leq \epsilon\|f\|\}$$

is given by

$$\eta_{E,f}(y) = \frac{\|r\|}{\|E\|\|y\| + \|f\|},$$

where $r = b - Ay$. □

For the particular choice $E = A$ and $f = b$, $\eta_{E,f}(y)$ is called the *normwise relative backward error*.

The next result measures the sensitivity of the system.

Theorem 1.5.2 *Let* $Ax = b$ *and* $(A + \Delta A)y = b + \Delta b$, *where* $\|\Delta A\| \leq \epsilon\|E\|$ *and* $\|\Delta b\| \leq \epsilon\|f\|$, *and assume that* $\epsilon\|A^{-1}\|\|E\| < 1$. *Then*

$$\frac{\|x - y\|}{\|x\|} \leq \frac{\epsilon}{1 - \epsilon\|A^{-1}\|\|E\|}\left(\frac{\|A^{-1}\|\|f\|}{\|x\|} + \|A^{-1}\|\|E\|\right),$$

and this bound is attainable to first order in ϵ. □

For componentwise analysis, we have the following two results. Here $|A| \leq |B|$ means $|a_{ij}| \leq |b_{ij}|$ for all i, j, and $\xi/0$ is interpreted as zero if $\xi = 0$ and infinity otherwise.

Theorem 1.5.3 (Oettli and Prager) *The* componentwise backward error

$$\omega_{E,f}(y) := \min\{\epsilon : (A + \Delta A)y = b + \Delta b, \ \ |\Delta A| \leq \epsilon E, \ \ |\Delta b| \leq \epsilon f\},$$

is given by

$$\omega_{E,f}(y) = \max_i \frac{|r_i|}{(E|y| + f)_i},$$

where $r = b - Ay$. □

Here E and f are assumed to have nonnegative entries. One common choice of tolerance is $E = |A|$ and $f = |b|$, which yields the *componentwise relative backward error*.

The next result gives a forward error bound corresponding to the componentwise backward error. First recall that a norm $\|\cdot\|$ on $\mathbb{C}^n$ is said to be absolute if $\||x|\| = \|x\|$ for all $x \in \mathbb{C}^n$.

$$\begin{array}{ccc} \text{Componentwise backward stability} & \Rightarrow & \text{Componentwise forward stability} \\ \omega_{|A|,|b|}(\widehat{x}) = O(u) & & \frac{\|x-\widehat{x}\|}{\|x\|} = O(\mathrm{cond}(A,x)u) \\ \Downarrow & & \Downarrow \\ \text{Normwise backward stability} & \Rightarrow & \text{Normwise forward stability} \\ \eta_{A,b}(\widehat{x}) = O(u) & & \frac{\|x-\widehat{x}\|}{\|x\|} = O(\kappa(A)u) \end{array}$$

Figure 1.1: Backward and forward stability.

Theorem 1.5.4 *Let $Ax = b$ and $(A + \Delta A)y = b + \Delta b$, where $|\Delta A| \le \epsilon E$ and $|\Delta b| \le \epsilon f$, and assume that $\epsilon \||A^{-1}|E\| < 1$, where $\|\cdot\|$ is an absolute norm. Then*

$$\frac{\|x - y\|}{\|x\|} \le \frac{\epsilon}{1 - \epsilon \||A^{-1}|E\|} \frac{\||A^{-1}|E|x| + |A^{-1}|f\|}{\|x\|},$$

and for the ∞-norm this bound is attainable to first order in ϵ. □

A numerical method for solving a square, nonsingular linear system $Ax = b$ is *normwise backward stable* if it produces a computed solution $\widehat{x}$ such that $\eta_{A,b}(\widehat{x})$ is of order the unit roundoff. *Componentwise backward stability* is defined in a similar way: we now require the componentwise backward error $\omega_{|A|,|b|}(\widehat{x})$ to be of order u.

If a method is normwise backward stable then, by Theorem 1.5.2, the forward error $\|x - \widehat{x}\|/\|x\|$ is bounded by a multiple of $\kappa(A)u$, where $\kappa(A) = \|A^{-1}\|\|A\|$. However, a method can produce a solution whose forward error is bounded in this way without the normwise backward error $\eta_{A,b}(\widehat{x})$ being of order u [55]. Hence it is useful to define a method for which $\|x - \widehat{x}\|/\|x\| = O(\kappa(A)u)$ as *normwise forward stable*. By similar reasoning involving $\omega_{|A|,|b|}(\widehat{x})$, we say a method is *componentwise forward stable* if $\|x - \widehat{x}\|/\|x\| = O(\mathrm{cond}(A,x)u)$, where the condition number

$$\mathrm{cond}(A,x) := \frac{\||A^{-1}||A||x|\|_\infty}{\|x\|_\infty}$$

was introduced by Skeel [80]. Figure 1.1 summarizes the definitions and the relations between them.

1.6 Overview

The rest of the thesis consists of five almost self-contained chapters. We first examine the stability and accuracy of the two most popular methods for solving dense symmetric indefinite linear system $Ax = b$, $A \in \mathbb{R}^{n \times n}$, namely the diagonal pivoting method and Aasen's method.

In Chapter 2, we describe the diagonal pivoting method in which a block $\mathrm{LDL}^{\mathrm{T}}$ factorization

$$PAP^T = LDL^T$$

is computed, where P is a permutation matrix, L is unit lower triangular and D is block diagonal with diagonal blocks of dimension 1 or 2. The choice of permutation is crucial to its stability. Both state-of-the-art packages LAPACK [2] and LINPACK [29] employ the pivoting strategy of Bunch and Kaufman [12]. The diagonal pivoting method with the Bunch–Kaufman pivoting strategy is normwise backward stable [58], but the factor L is unbounded in norm. Ashcraft, Grimes and Lewis [6] comment that the solutions obtained without a bound on $\|L\|$ can be less accurate than they should be, and propose a "bounded Bunch–Kaufman" pivoting strategy that produces a bounded L. This new pivoting strategy is claimed to have "superior accuracy" to the original Bunch–Kaufman pivoting strategy. A set of test matrices for which the bounded Bunch–Kaufman pivoting strategy has achieved better accuracy is given in [6]. We assess these two closely related pivoting strategies using various stability measures and examine the significance of the Ashcraft, Grimes and Lewis examples.

In Chapter 3, we look at the stability and accuracy of Aasen's method. Aasen's method with partial pivoting computes a $\mathrm{LTL}^{\mathrm{T}}$ factorization

$$PAP^T = LTL^T,$$

where L is unit lower triangular with first column e_1, T is tridiagonal, and P is

a permutation matrix chosen such that $|l_{ij}| \leq 1$, and it is the only stable direct method with a guarantee of no more than $n^2/2$ comparisons and a bounded factor L. Despite these advantages, Aasen's method has received little attention in the literature for the last decade. Neither LAPACK [2] nor LINPACK [29] has an implementation of Aasen's method. Since 1993, Aasen's method has been included in the IMSL Fortran 90 MP Library [48], [90]. The algorithm is normwise backward stable [57] provided the tridiagonal system is solved in a numerically stable way.

Not much is known about the behaviour of the growth factor in Aasen's method. We derive a growth factor bound for Aasen's method and show that the bound is attainable for matrix of dimension 3. Direct search methods [28], [53], [86], [87] are employed to detect the large growth factor for Aasen's method. The results give useful insights into the stability of Aasen's method.

Chapters 4–6 can be viewed as examining nearness problems associated with symmetric indefinite matrices with their applications.

In Chapter 4, we look at the modified Cholesky factorization. Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ not necessarily positive definite, a modified Cholesky factorization combines a matrix factorization and a modification scheme to compute a "not-too-large" perturbation E in some suitable norm so that $P(A+E)P^T$ is positive definite, where P is a permutation matrix. We explain the two existing modified Cholesky factorizations of Gill, Murray and Wright [37] and Schnabel and Eskow [78]. Two new algorithms, based on the $\mathrm{LDL}^{\mathrm{T}}$ factorization with the bounded Bunch–Kaufman pivoting strategy and the $\mathrm{LTL}^{\mathrm{T}}$ factorization with partial pivoting, are proposed. Our algorithms have the advantages of ease of implementation and the existence of a priori bounds for assessing how "good" the perturbation is. Our experimental results show that all four algorithms are competitive from the linear algebra viewpoint.

In Chapter 5 we focus on deriving structured perturbations to a matrix $A \in \mathbb{R}^{n\times n}$ with a natural block 2×2 structure arising in optimization problems. In constrained optimization, a "second order sufficiency" condition leads to the problem of perturbating the (1,1) block of A so that A has a particular inertia. We derive a perturbation, valid for any unitary invariant norm, that increases the number of nonnegative eigenvalues by a given amount and show how it can be computed efficiently given a factorization of the original matrix. We also consider an alternative way to satisfy the optimality condition based on a projected Hessian approach. Theoretical tools developed include an extension of Ostrowski's theorem on congruence transformations and some lemmas on inertia properties of block 2×2 matrices.

In Chapter 6, the generalized Hermitian eigenvalue problem is discussed. That is, $Az = \lambda Bz$ for A, B Hermitian. For B nonsingular, it is equivalent to the standard eigenproblem $B^{-1}Az = \lambda z$. A summary of the characteristics of the eigenvalues of this matrix product is presented. Of particular interest is the case where (A, B) is a definite pair. We show that the generalized Hermitian eigenvalue problem can be reduced to a standard Hermitian eigenvalue problem in this case, and how this approach can be efficiently implemented when A and B are banded.

When (A, B) is not a definite pair, one relevant nearness problem is to compute the nearest definite pair. We derive an elegant solution, in terms of what we call the inner numerical radius, to this nearness problem in the 2-norm. We suggest an algorithm for estimating the inner numerical radius, and hence optimal 2-norm perturbations. When (A, B) is a normal pair, an alternative approach which exploits the characteristics of the eigenvalues is proposed.

Chapter 2

Accuracy and Stability of the Diagonal Pivoting Method

2.1 Introduction

The most popular method for solving a dense symmetric indefinite linear system $Ax = b$, $A \in \mathbb{R}^{n\times n}$ is the diagonal pivoting method in which we compute a block $\mathrm{LDL}^{\mathrm{T}}$ factorization

$$PAP^T = LDL^T, \tag{2.1}$$

where P is a permutation matrix, L is unit lower triangular and D is block diagonal with diagonal blocks of dimension 1 or 2. There are various ways to choose the permutations. Bunch and Parlett [14] proposed a complete pivoting strategy, which requires $O(n^3)$ comparisons. Bunch and Kaufman [12] subsequently proposed a partial pivoting strategy requiring only $O(n^2)$ comparisons, and it is this strategy that is used in LAPACK [2] and LINPACK [29].

The diagonal pivoting method with the Bunch–Kaufman pivoting strategy is normwise backward stable, but the factor L is unbounded in norm. Ashcraft, Grimes and Lewis [6] state that "the solutions obtained without a bound on $\|L\|$ can be less accurate than they should be", and they develop modifications of the Bunch–Kaufman pivoting strategy, for both dense and sparse matrices, that produce a bounded L. In particular, they propose a "bounded Bunch–Kaufman" pivoting strategy that they claim has "superior accuracy" to the original Bunch–Kaufman strategy. Both pivoting strategies passed the test certification programs in LAPACK [6]. We shall limit our discussion to these two pivoting strategies.

The purpose of this chapter is to investigate the effect of the unbounded L in the Bunch–Kaufman pivoting strategy, and to determine whether the bounded Bunch–Kaufman strategy leads to more accurate computed solutions.

The rest of the chapter is organized as follows. We describe the Bunch–Kaufman and the bounded Bunch–Kaufman pivoting strategy in Section 2.2. In Section 2.3, a growth factor bound is derived. We present the backward stability result of Higham [58] in Section 2.4 and use the result to assess whether the claimed superiority of the bounded Bunch–Kaufman pivoting strategy is justified. Section 2.5 is devoted to an investigation of the role played by the Ashcraft, Grimes and Lewis examples [6]. Concluding remarks are given in Section 2.6

2.2 Pivoting Strategies

To define the Bunch–Kaufman (BK) and bounded Bunch–Kaufman (BBK) pivoting strategies we first need to explain how the block $\mathrm{LDL}^{\mathrm{T}}$ factorization is computed. If the symmetric matrix $A \in \mathbb{R}^{n\times n}$ is nonzero, we can find a permutation Π and an integer $s = 1$ or 2 so that

$$\Pi A \Pi^T = \begin{array}{c} \\ s \\ n-s \end{array} \begin{array}{c} \begin{array}{cc} s & n-s \end{array} \\ \begin{bmatrix} E & C^T \\ C & B \end{bmatrix} \end{array},$$

with E nonsingular. Having chosen such a Π we can factorize

$$\Pi A \Pi^T = \begin{bmatrix} I_s & 0 \\ CE^{-1} & I_{n-s} \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & B - CE^{-1}C^T \end{bmatrix} \begin{bmatrix} I_s & E^{-1}C^T \\ 0 & I_{n-s} \end{bmatrix}.$$

This process is repeated recursively on the $(n-s)\times(n-s)$ Schur complement

$$S = B - CE^{-1}C^T,$$

yielding the factorization (2.1) on completion. This factorization costs $n^3/3$ operations (the same cost as Cholesky factorization of a positive definite matrix) plus the cost of determining the permutations Π.

To describe the BK pivoting strategy it suffices to describe the pivot choice for the first stage of the factorization. Here s denotes the size of the pivot block.

Algorithm BK (Bunch–Kaufman Pivoting Strategy) *This algorithm determines the pivot for the first stage of block LDL^T factorization applied to a symmetric matrix $A \in \mathbb{R}^{n\times n}$.*

$\alpha := (1+\sqrt{17})/8 \ (\approx 0.64)$

$\gamma_1 :=$ maximum magnitude of any subdiagonal entry in column 1.

If $\gamma_1 = 0$ there is nothing to do on this stage of the factorization.

if $|a_{11}| \geq \alpha\gamma_1$

(1) use a_{11} as a 1×1 pivot ($s=1$, $\Pi = I$).

else

$r :=$ row index of first (subdiagonal) entry of maximum magnitude in column 1.

$\gamma_r :=$ maximum magnitude of any off-diagonal entry in column r.

if $|a_{11}|\gamma_r \geq \alpha\gamma_1^2$

(2) use a_{11} as a 1×1 pivot ($s=1$, $\Pi = I$).

else if $|a_{rr}| \geq \alpha\gamma_r$

(3) use a_{rr} as a 1×1 pivot ($s=1$, Π swaps rows and columns 1 and r).

else

(4) use $\begin{bmatrix} a_{11} & a_{r1} \\ a_{r1} & a_{rr} \end{bmatrix}$ as a 2×2 pivot ($s=2$, Π swaps rows and columns 2 and r).

end

end

The BK pivoting strategy searches at most two columns of the Schur complement at each stage, so requires only $O(n^2)$ comparisons in total. The given choice

of α minimizes a bound on the element growth and is obtained by equating the maximal element growth over two 1×1 pivot steps to that for one 2×2 pivot step; see Section 2.3. Note that it is cases (2) and (4) of Algorithm BK in which unbounded elements in L arise, as we now explain.

- Case (1) : a_{11} is a pivot, with $|a_{11}| \geq \alpha\gamma_1$. It follows that

$$l_{i1} = a_{i1}/a_{11}, \qquad |l_{i1}| \leq \frac{1}{\alpha}.$$

- Case (2) : a_{11} is a pivot, with $|a_{11}|\gamma_r \geq \alpha\gamma_1^2$. We have

$$l_{i1} = a_{i1}/a_{11}, \qquad |l_{i1}| \leq \frac{\gamma_1}{|a_{11}|} \leq \frac{\gamma_r}{\gamma_1} \cdot \frac{1}{\alpha},$$

  where γ_r/γ_1 can be arbitrarily large.

- Case (3) : a_{rr} is a pivot, with $|a_{rr}| \geq \alpha\gamma_r$. It follows that, for $i \neq r$,

$$l_{ir} = a_{ir}/a_{rr}, \qquad |l_{ir}| \leq \frac{1}{\alpha}.$$

- Case (4) : $\begin{bmatrix} a_{11} & a_{r1} \\ a_{r1} & a_{rr} \end{bmatrix}$ is a 2×2 pivot. For $i \neq 1, r$, we have

$$\begin{aligned} l_{i1} &= \frac{a_{i1}a_{rr} - a_{r1}a_{ir}}{a_{11}a_{rr} - a_{r1}^2}, & l_{ir} &= \frac{a_{11}a_{ir} - a_{r1}a_{i1}}{a_{11}a_{rr} - a_{r1}^2}, \\ |l_{i1}| &\leq \frac{\gamma_1(\alpha\gamma_r) + \gamma_1\gamma_r}{\gamma_1^2(1-\alpha^2)} & |l_{ir}| &\leq \frac{|a_{11}|\gamma_r + \gamma_1^2}{\gamma_1^2(1-\alpha^2)} \\ &\leq \frac{\gamma_1\gamma_r(1+\alpha)}{\gamma_1^2(1-\alpha^2)} & &\leq \frac{\gamma_1^2(1+\alpha)}{\gamma_1^2(1-\alpha^2)} \\ &= \frac{\gamma_r}{\gamma_1} \cdot \frac{1}{1-\alpha}, & &= \frac{1}{1-\alpha}. \end{aligned}$$

  Here we have the same problem as in case (2); $|l_{i1}|$ is not bounded.

The BBK pivoting strategy is broadly similar to the BK strategy. The idea is to suppress case (2) and allow an iterative phase for cases (3) and (4) so that

the ratio γ_r/γ_1 is equal to 1 [6]. One immediate consequence is that every entry of L is bounded by $\max\{1/(1-\alpha), 1/\alpha\} \approx 2.78$.

Algorithm BBK (Bounded Bunch–Kaufman Pivoting Strategy) *This algorithm determines the pivot for the first stage of block LDL^T factorization applied to a symmetric matrix $A \in \mathbb{R}^{n\times n}$.*

$\alpha := (1+\sqrt{17})/8$ (≈ 0.64)

$\gamma_1 :=$ maximum magnitude of any subdiagonal entry in column 1.

If $\gamma_1 = 0$ there is nothing to do on this stage of the factorization.

if $|a_{11}| \geq \alpha\gamma_1$

use a_{11} as a 1×1 pivot ($s = 1$, $\Pi = I$).

else

$i := 1$; $\gamma_i = \gamma_1$

repeat

$r :=$ row index of first (subdiagonal) entry of maximum magnitude. in column i.

$\gamma_r :=$ maximum magnitude of any off-diagonal entry in column r.

if $|a_{rr}| \geq \alpha\gamma_r$

use a_{rr} as a 1×1 pivot ($s = 1$, Π swaps rows and columns 1 and r).

else if $\gamma_i = \gamma_r$

use $\begin{bmatrix} a_{ii} & a_{ri} \\ a_{ri} & a_{rr} \end{bmatrix}$ as a 2×2 pivot ($s = 2$, Π swaps rows and columns 1 and i, and 2 and r).

else

$i := r$, $\gamma_i := \gamma_r$.

end

until a pivot is chosen.

end

The repeat loop in Algorithm BBK searches for an off-diagonal element a_{ri} that is simultaneously the largest in magnitude in the rth and ith columns, and it uses this element to build a 2×2 pivot; the search terminates prematurely if a suitable 1×1 pivot is found.

It is readily verified [6] that any 2×2 pivot D_{ii} satisfies

$$\left| \begin{bmatrix} a_{ii} & a_{ri} \\ a_{ri} & a_{rr} \end{bmatrix}^{-1} \right| \leq \frac{1}{\gamma_r(1-\alpha^2)} \begin{bmatrix} \alpha & 1 \\ 1 & \alpha \end{bmatrix}.$$

Thus the condition number for any 2×2 pivot is bounded by

$$\kappa_2(D_{ii}) \leq \frac{1+\alpha}{1-\alpha} < 4.57. \tag{2.2}$$

Since the value of γ_i increases strictly from one pivot step to the next, the search in Algorithm BBK takes at most n steps. The cost of the searching is intermediate between the cost for the Bunch–Kaufman strategy and that for the Bunch–Parlett [14] strategy in which the whole active submatrix is searched at each step. Matrices are known [6] for which the entire remaining submatrix must be searched at each step, in which case the cost is the same as for the Bunch–Parlett strategy; see Figure 2.1 for a few examples.

However, Ashcraft, Grimes and Lewis [6] found in their experiments that, on average, less that $2.5k$ comparisons were required to find a pivot from a $k \times k$ submatrix, and they give a probabilistic analysis which shows that the expected number of comparisons is less than $ek \approx 2.718k$ for matrices with independently distributed random elements. Therefore we regard the block LDL^T factorization with the BBK pivoting strategy as being of similar cost to the Cholesky factorization, while recognizing that in certain rare cases the searching overhead may increase the operation count by about 50%.

$$\begin{bmatrix} 0 & & & 2 \\ & 4 & 4 & \\ & 4 & 0 & 3 \\ 2 & & 3 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & & & & & & 2 \\ & 6 & 6 & & & & \\ & 6 & 0 & 5 & & & \\ & & 5 & 0 & 4 & & \\ & & & 4 & 0 & 3 & \\ 2 & & & & 3 & 0 & \end{bmatrix}, \quad \begin{bmatrix} 0 & & & & & & & 2 \\ & 8 & 8 & & & & & \\ & 8 & 0 & 7 & & & & \\ & & 7 & 0 & 6 & & & \\ & & & 6 & 0 & 5 & & \\ & & & & 5 & 0 & 4 & \\ & & & & & 4 & 0 & 3 \\ 2 & & & & & & 3 & 0 \end{bmatrix}.$$

Figure 2.1: Matrices for which the entire remaining submatrix must be searched at each step of the BBK strategy.

2.3 The Growth Factor

The growth factor of the block LDL^T factorization is defined in the same way as for Gaussian elimination by

$$\rho_n = \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}, \tag{2.3}$$

where the $a_{ij}^{(k)}$ are the elements of the Schur complements arising in the course of the factorization. The normwise backward stability result in next section involves the growth factor which carries an implicit assumption that it is small. In other words, the growth factor governs the normwise backward stability of the block LDL^T factorization with various pivoting strategies.

Explicit bounds for the growth factor with various pivoting strategies have been derived [6], [11], [12]. In particular, for both the BK and BBK pivoting strategies, the best available growth factor bound is

$$\rho_n \leq (1 + 1/\alpha)^{n-1} \approx (2.57)^{n-1}, \tag{2.4}$$

where $\alpha = (1 + \sqrt{17})/8$.

In this section, we show how the growth factor bound is derived and justify

the choice of α. Recall that Algorithm BK has four pivot choices. Define $\mu^{(k)}$ by

$$\mu^{(k)} = \max_{i,j \geq k} |a_{ij}^{(k)}|.$$

- Case (1) : $a_{11}^{(k)}$ is a pivot, with $|a_{11}^{(k)}| \geq \alpha \gamma_1$. It follows that

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{i1}^{(k)} a_{1j}^{(k)}}{a_{11}^{(k)}},$$

  so that

$$|a_{ij}^{(k+1)}| \leq |a_{ij}^{(k)}| + \gamma_1 \frac{|a_{i1}^{(k)}|}{|a_{11}^{(k)}|},$$

  and hence

$$\mu^{(k+1)} \leq \mu^{(k)} + \gamma_1 \frac{\gamma_1}{|a_{11}^{(k)}|} \leq \mu^{(k)} \left(1 + \frac{1}{\alpha}\right).$$

- Case (2) : a_{11} is a pivot, with $|a_{11}| \gamma_r \geq \alpha \gamma_1^2$. We have

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{i1}^{(k)} a_{1j}^{(k)}}{a_{11}^{(k)}},$$

  so that

$$|a_{ij}^{(k+1)}| \leq |a_{ij}^{(k)}| + \frac{\gamma_1^2}{|a_{11}^{(k)}|},$$

  and hence

$$\mu^{(k+1)} \leq \mu^{(k)} + \frac{\gamma_r}{\alpha} \leq \mu^{(k)} \left(1 + \frac{1}{\alpha}\right).$$

- Case (3) : a_{rr} is a pivot, with $|a_{rr}| \geq \alpha \gamma_r$. It follows that, for $i \neq r$,

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ir}^{(k)} a_{rj}^{(k)}}{a_{rr}^{(k)}},$$

  so that

$$|a_{ij}^{(k+1)}| \leq |a_{ij}^{(k)}| + \frac{\gamma_r^2}{|a_{rr}^{(k)}|},$$

and hence

$$\mu^{(k+1)} \leq \mu^{(k)} + \gamma_r \frac{\gamma_r}{|a_{rr}^{(k)}|} \leq \mu^{(k)} \left(1 + \frac{1}{\alpha}\right).$$

- Case (4) : $\begin{bmatrix} a_{11} & a_{r1} \\ a_{r1} & a_{rr} \end{bmatrix}$ is a 2×2 pivot. We will make use of the following inequalities which arise from the conditions that the pivot satisfies:

$$|a_{11}^{(k)}\gamma_r| < \alpha\gamma_1^2, \qquad |a_{rr}^{(k)}| < \alpha\gamma_r, \qquad |a_{11}^{(k)}a_{rr}^{(k)}| < \alpha^2\gamma_1^2,$$

$$|a_{11}^{(k)}a_{rr}^{(k)} - \gamma_1^2| \geq \gamma_1^2 - |a_{11}^{(k)}a_{rr}^{(k)}| > \gamma_1^2(1-\alpha^2).$$

For $i \neq 1, r$ and $j \neq 1, r$, we have

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{11}^{(k)}a_{ir}^{(k)}a_{rj}^{(k)} - a_{r1}^{(k)}(a_{i1}^{(k)}a_{rj}^{(k)} + a_{ir}^{(k)}a_{1j}^{(k)}) + a_{rr}^{(k)}a_{i1}^{(k)}a_{1j}^{(k)}}{a_{11}^{(k)}a_{rr}^{(k)} - a_{r1}^{(k)}a_{r1}^{(k)}},$$

so that

$$|a_{ij}^{(k+1)}| < |a_{ij}^{(k)}| + \frac{|a_{11}^{(k)}|\gamma_r^2 + \gamma_1(\gamma_1\gamma_r + \gamma_r\gamma_1) + |a_{rr}^{(k)}|\gamma_1^2}{\gamma_1^2(1-\alpha^2)},$$

and hence

$$\mu^{(k+1)} \leq \mu^{(k)} \left(1 + \frac{2(1+\alpha)}{1-\alpha^2}\right) = \mu^{(k)} \left(1 + \frac{2}{1-\alpha}\right).$$

By equating the maximal element growth of two 1×1 pivot steps with that for one 2×2 pivot step, we obtain

$$\left(1 + \frac{1}{\alpha}\right)^2 = \left(1 + \frac{2}{1-\alpha}\right),$$

and $\alpha = (1+\sqrt{17})/8$ is the positive root of this quadratic equation. Hence we obtain (2.4). Whether two 1×1 pivot steps can achieve the maximal element growth is an open question.

It is easily seen that the same bounds hold for Algorithm BBK. For cases (1) and (3) no modification is required. For case (4), we have $\gamma_1 = \gamma_r$ and the same bound on element growth holds. The growth factor bound (2.4) is weak and rarely approached in general. Whether this bound is attainable remains an open question.

2.4 Error Analysis

Our model of floating point arithmetic is the usual model defined as in (1.2). The following backward stability result, valid for any pivoting strategy, is proved by Higham [58].

Theorem 2.4.1 (Higham) *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and let $\widehat{x}$ be a computed solution to the linear system $Ax = b$ produced using the diagonal pivoting method with any pivoting strategy. If all linear systems involving 2×2 pivots are solved in a componentwise backward stable way then*

$$(A + \Delta A)\widehat{x} = b, \qquad |\Delta A| \leq p(n)u\big(|A| + P^T|\widehat{L}||\widehat{D}||\widehat{L}^T|P\big) + O(u^2), \tag{2.5}$$

where p is a linear polynomial and $PAP^T \approx \widehat{L}\widehat{D}\widehat{L}^T$ is the computed block LDL^T factorization. □

The assumption in the theorem about the 2×2 pivots is satisfied provided the 2×2 systems are solved by Gaussian elimination with partial pivoting or even by use of the explicit inverse [58], so this assumption is satisfied in practice.

We now examine the implications of Theorem 2.4.1 for four different forms of stability.

2.4.1 Normwise Backward Stability

To establish the normwise backward and forward stability results using Theorem 2.4.1, the remaining task is to bound the quantity $|L||D||L^T|$ in some suitable norm. Higham [58] shows that $\| \, |L||D||L^T| \, \|_M \leq 36n\rho_n\|A\|_M$ for the BK pivoting strategy, where ρ_n is the growth factor defined as in (2.3) and

$$\|A\|_M := \max_{i,j} |a_{ij}|.$$

By inspecting the analysis it is easily seen that the same bound with a smaller constant term holds for the BBK pivoting strategy. Hence both pivoting strategies are normwise backward stable, provided there is no large element growth.

Theorem 2.4.2 (Higham) *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and let $\widehat{x}$ be a computed solution to the linear system $Ax = b$ produced using the diagonal pivoting method with either the BK or the BBK pivoting strategies. If all linear systems involving 2×2 pivots are solved in a componentwise backward stable way then*

$$(A + \Delta A)\widehat{x} = b, \qquad \|\Delta A\|_M \leq p(n)\rho_n u\|A\|_M + O(u^2), \tag{2.6}$$

where p is a quadratic polynomial and $PAP^T \approx \widehat{L}\widehat{D}\widehat{L}^T$ is the computed block LDL^T factorization. □

An immediate consequence of Theorem 2.4.2 is that in the absence of large element growth both strategies produce a forward error bounded by a multiple of $\kappa(A) = \|A\|_M\|A^{-1}\|_M$, that is, both strategies produce a normwise forward stable method.

2.4.2 Componentwise Backward Stability

For componentwise backward stability we require that

$$(A + \Delta A)\widehat{x} = b + \Delta b, \qquad |\Delta A| \leq \epsilon|A|, \quad |\Delta b| \leq \epsilon|b|,$$

where ϵ is a small multiple of the unit roundoff.

The best a priori componentwise backward error bound obtainable from Theorem 2.4.1 involves the quantity

$$\xi = \min\left\{\eta : P^T|L||D||L^T|P \leq \eta|A|\right\}. \tag{2.7}$$

Here, for simplicity, we use the exact factors instead of their computed counterparts. The bleach of correctness is harmless to the overall analysis [55, p. 177], [58].

We now show by example that the a priori componentwise backward error quantity ξ can be arbitrarily larger for the BK strategy than for the BBK strategy, and vice versa. In other words, neither method is better than the other from the point of view of an a priori componentwise backward error bound.

2.4.3 BK Beats BBK

Consider

$$A = \begin{bmatrix} 0 & \epsilon & 0 \\ \epsilon & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \qquad \epsilon > 0. \tag{2.8}$$

The BK pivoting strategy computes

$$A = LDL^T = \begin{bmatrix} 1 & & \\ 0 & 1 & \\ \epsilon^{-1} & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & \epsilon & \\ \epsilon & 0 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \epsilon^{-1} \\ & 1 & 0 \\ & & 1 \end{bmatrix}.$$

The nonnegativity of the factors tells us immediately that $|L||D||L^T| = A = |A|$, so $\xi = 1$, that is, we have perfect componentwise backward stability.

On the other hand, the BBK strategy computes

$$PAP^T = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & \epsilon \\ 0 & \epsilon & 0 \end{bmatrix}$$

and

$$PAP^T = LDL^T = \begin{bmatrix} 1 & & \\ 1 & 1 & \\ 0 & -\epsilon & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & -1 & \\ & & \epsilon^2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ & 1 & -\epsilon \\ & & 1 \end{bmatrix}.$$

We have

$$|L||D||L^T| = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & \epsilon \\ 0 & \epsilon & 2\epsilon^2 \end{bmatrix},$$

so $\xi = \infty$ for the BBK pivoting strategy since we require $2\epsilon^2 < 0$.

2.4.4 BBK Beats BK

Let

$$A = \begin{bmatrix} \epsilon^2 & \epsilon & \epsilon \\ \epsilon & 0 & 1 \\ \epsilon & 1 & 0 \end{bmatrix}, \qquad 0 < \epsilon < \alpha. \tag{2.9}$$

The BK pivoting strategy computes

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

We have

$$|L||D||L^T| = \begin{bmatrix} \epsilon^2 & \epsilon & \epsilon \\ \epsilon & 2 & 1 \\ \epsilon & 1 & 2 \end{bmatrix},$$

thus $\xi = \infty$.

The BBK strategy computes

$$PAP^T = \begin{bmatrix} 0 & 1 & \epsilon \\ 1 & 0 & \epsilon \\ \epsilon & \epsilon & \epsilon^2 \end{bmatrix} = \begin{bmatrix} 1 & & \\ 0 & 1 & \\ \epsilon & \epsilon & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & \\ 1 & 0 & \\ & & -\epsilon^2 \end{bmatrix} \begin{bmatrix} 1 & 0 & \epsilon \\ & 1 & \epsilon \\ & & 1 \end{bmatrix}.$$

So

$$|L||D||L^T| = \begin{bmatrix} 0 & 1 & \epsilon \\ 1 & 0 & \epsilon \\ \epsilon & \epsilon & 3\epsilon^2 \end{bmatrix},$$

from which we see that $\xi = 3$ for the BBK pivoting strategy.

Numerical experiments with matrices (2.8), (2.9) confirm that the actual componentwise backward errors of the BK and BBK pivoting strategies can behave

	Matrix (2.8)		Matrix (2.9)	
ϵ	BK	BBK	BK	BBK
10^{-1}	0	6e–17	1e–16	6e–17
10^{-2}	0	1e–15	2e–15	9e–17
10^{-3}	0	6e–15	9e–16	0
10^{-4}	0	2e–13	3e–14	0
10^{-5}	8e–17	2e–12	3e–12	0
10^{-6}	0	1e–11	6e–12	1e–16
10^{-7}	0	6e–11	4e–10	0

Table 2.1: Backward error for computed solution of symmetric indefinite systems of dimension 3.

as predicted by the bounds, that is, one componentwise backward error can be of order u and the other very large. We solved linear systems $Ax = b$, where $b = A[1\ 1\ \epsilon]^T$, with A defined in (2.8) and (2.9). Table 2.1 shows the componentwise relative backward error of the computed solution $\widehat{x}$,

$$\begin{aligned}\omega_{|A|,|b|}(\widehat{x}) &:= \{\epsilon : (A+\Delta A)\widehat{x} = b + \Delta b,\ \ |\Delta A| \le \epsilon|A|,\ \ |\Delta b| \le \epsilon|b|\} \\ &= \max_i \frac{|A\widehat{x} - b|_i}{(|A||\widehat{x}| + |b|)_i}\end{aligned}$$

(see [68] or [55, Thm. 7.3] for a proof of the latter equality), which would be of order u for a componentwise backward stable method. Hence we conclude that neither pivoting strategy is better than the other from the point of view of componentwise backward error.

2.4.5 Normwise and Componentwise Forward Stability

Both the Bunch–Kaufman and the bounded Bunch–Kaufman pivoting strategy have a bound for $\|x - \widehat{x}\|/\|x\|$ of order $\kappa(A)u$. Thus both strategies are normwise forward stable in the absence of a large growth factor. From Theorem 2.4.1 we

have

$$\begin{aligned} |x - \widehat{x}| &\leq |A^{-1}||\Delta A||\widehat{x}| \\ &= |A^{-1}||\Delta A||x| + O(u^2) \\ &\leq |A^{-1}|p(n)u(|A| + P^T|\widehat{L}||\widehat{D}||\widehat{L}^T|P)|x| + O(u^2), \end{aligned}$$

where $\widehat{x}$ is replaced by x in the second equality using a standard technique from [58]. Thus

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \leq p(n)u(\mathrm{cond}(A, x) + \||A^{-1}|P^T|\widehat{L}||\widehat{D}||\widehat{L}^T|P\|_\infty) + O(u^2), \qquad (2.10)$$

where

$$\mathrm{cond}(A, x) = \frac{\||A^{-1}||A||x|\|_\infty}{\|x\|_\infty}.$$

To compare the forward error bounds for different pivoting strategies we therefore need to look at the matrix

$$W = |A^{-1}|P^T|L||D||L^T|P, \qquad (2.11)$$

where we have dropped the hats. Since $PAP^T = LDL^T$, we have

$$L = PAP^TL^{-T}D^{-1},$$

thus

$$W \leq |A^{-1}|P^T \cdot P|A|P^T|L^{-T}||D^{-1}| \cdot |D||L^T|P = |A^{-1}||A|P^T|L^{-T}||D^{-1}||D||L^T|P,$$

which gives

$$\|W\|_\infty \leq \mathrm{cond}(A)\,\mathrm{cond}(|D||L^T|), \qquad (2.12)$$

where $\mathrm{cond}(A) = \||A^{-1}||A|\|_\infty$. Note that if D is diagonal then $\|W\|_\infty \leq \mathrm{cond}(A)\,\mathrm{cond}(L^T)$.

For the BK pivoting strategy, $\mathrm{cond}(|D||L^T|)$ is unbounded, as is easily shown by example. Note that $|L^{-T}||D^{-1}||D||L^T|$ is block upper triangular with diagonal blocks identical to those of $|D^{-1}||D|$. Thus $\mathrm{cond}(|D||L^T|)$ is unbounded if

cond($|D|$) is unbounded. The following example is chosen so that cond($|D|$), and hence cond($|D||L^T|$), is unbounded for the BK strategy but is bounded for the BBK strategy. Let

$$A = \begin{bmatrix} \epsilon^5 & \epsilon^2 & 0 \\ \epsilon^2 & \epsilon & 1 \\ 0 & 1 & 1 \end{bmatrix}, \qquad 0 < \epsilon \ll 1.$$

The BK strategy computes

$$A = LDL^T = \begin{bmatrix} 1 & & \\ 0 & 1 & \\ \frac{1}{\epsilon^2-\epsilon^4} & \frac{-\epsilon}{1-\epsilon^2} & 1 \end{bmatrix} \begin{bmatrix} \epsilon^5 & \epsilon^2 & \\ \epsilon^2 & \epsilon & \\ & & 1+\frac{\epsilon}{1-\epsilon^2} \end{bmatrix} \begin{bmatrix} 1 & & \frac{1}{\epsilon^2-\epsilon^4} \\ & 1 & \frac{-\epsilon}{1-\epsilon^2} \\ & & 1 \end{bmatrix}.$$

Letting D_i denote the diagonal block of D, we have

$$\text{cond}(D) = \max_i\{\text{cond}(D_i)\} = \frac{2+\epsilon+\epsilon^3}{\epsilon-\epsilon^3} \to \infty, \qquad \text{as } \epsilon \to 0.$$

For the BBK pivoting strategy cond($|D||L^T|$) is bounded explicitly. Since $\max_{i,j}|l_{ij}| < 2.79$ and it is attainable only with a 2×2 pivot for which it causes a subdiagonal element $l_{i+1,i}$ to be zero, we have

$$\|L^T\|_\infty < 1 + (n-2)2.79 = 2.79n - 4.57, \qquad \|L^{-T}\|_\infty < (3.79)^{n-1}, \qquad (2.13)$$

where the latter uses the bound in [55, Thm. 8.11, Problem 8.5].

Together with (2.2), we have

$$\text{cond}(|D||L^T|) \le \text{cond}(D)\kappa_\infty(L^T) < 4.57 \times (2.79n - 4.57)(3.79)^{n-1}.$$

This bound is very pessimistic. Typically, for the BBK strategy, $\kappa_\infty(L^T)$ is of relatively small norm.

2.5 The Role of Ashcraft, Grimes and Lewis Example

Ashcraft, Grimes and Lewis [6] have identified a set of matrices for which the BBK strategy achieves better accuracy that the BK strategy. In this section, we give some explanations for the better accuracy achieved by the BBK strategy, and assess the importance of the examples.

We know that the accuracy of the BK algorithm goes hand in hand with ill conditioned pivots and unbounded L. The examples of Ashcraft, Grimes and Lewis ensure that pivots of cases (2) and (4) of the BK pivoting strategy are chosen and that a large-normed L is formed. However the complicated requirements of pivot selection within the BK pivoting strategy guarantee cancellation between the large and small elements and hence yield the normwise backward stability result.

Experiments similar to those described in Ashcraft, Grimes and Lewis [6] were performed. Test matrices are scaled using the scheme described in Figure 2.2 and Table 2.2 in which large entries in L arise. Each set of parameters was given a different random symmetric indefinite matrix $A \in \mathbb{R}^{n\times n}$ with elements normally distributed with mean 0 and variance 1. In total 62 test matrices were generated. We chose x as $x_i = (-1)^i$ and $b := Ax$. Note that the computed b is not the exact right hand side corresponding to x due to rounding error in its formation.

We measure the ratio β_∞ between the normwise forward errors of BK and BBK

$$\beta_\infty = \frac{\|\widehat{x} - x\|_{\infty,\mathrm{BK}}}{\|\widehat{x} - x\|_{\infty,\mathrm{BBK}}},$$

and compare with $\mathrm{cond}(|D||L^T|)_{\mathrm{BK}}$, the value of $\mathrm{cond}(|D||L^T|)$ for BK. Here the subscripts BK and BBK denote the computed quantities using the BK and BBK strategies respectively. Our results, which show an increasing trend of β_∞ and

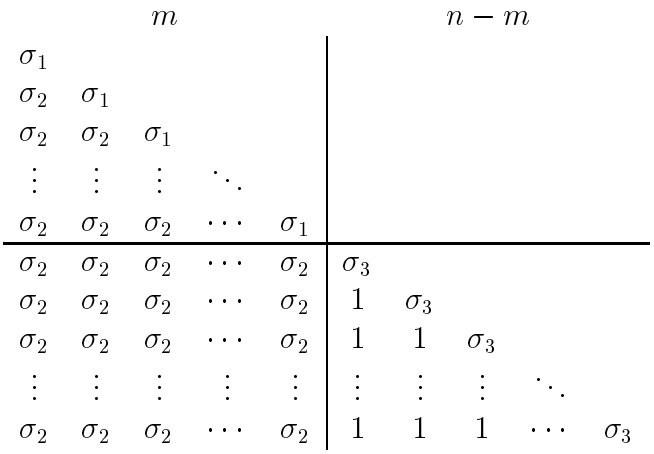

Figure 2.2: Scaling parameter for each entry of test matrix.

initial pivot	σ_1	σ_2	σ_3	t_1	t_2
1×1 pivot	10^{-t_1}	10^{-t_2}	$1/10$	$t_2+1,\ldots,2t_2$	$1,\ldots,6$
well-conditioned 2×2 block	10^{-t_1}	10^{-t_2}	10^{-t_2}	$2t_2+1,\ldots,3t_2$	$2,\ldots,6$
general 2×2 block	10^{-t_1}	10^{-t_2}	$1/10$	$2t_2+1,\ldots,3t_2$	$1,\ldots,6$

Table 2.2: Parameterization of scaled test matrices.

$\mathrm{cond}(|D||L^T|)_{\mathrm{BK}}$, agree with the test results of Ashcraft, Grimes and Lewis [6]; see Figure 2.3. We note that $\beta_\infty \approx 1$ for several entries, which shows that the scaling scheme sometimes has no effect on the accuracy of the computed solution.

Recall that if L and D are nonnegative, that is, $|L| = L$ and $|D| = D$, then $W = |A^{-1}|P^T|L||D||L^T|P = |A^{-1}||A|$, and we have a perfect stability result. The a priori componentwise forward error bound (2.10) involves the quantity $\|W\|_\infty$, which may be uninformative. The elements of the Ashcraft, Grimes and Lewis examples vary over 18 orders of magnitude, so while $\|W_{\mathrm{BK}}\|_\infty / \|W_{\mathrm{BBK}}\|_\infty$ is small (between orders 10^0 to 10^3 for these examples), we may be making large perturbations in the small elements of $|A^{-1}||A|$. Thus a componentwise measure

$$\tau = \min\{\eta : |A^{-1}|P^T|L||D||L^T|P \le \eta |A^{-1}||A|\} \tag{2.14}$$

is employed. Figure 2.4 shows an increasing trend between the ratio $\tau_{\mathrm{BBK}}/\tau_{\mathrm{BK}}$

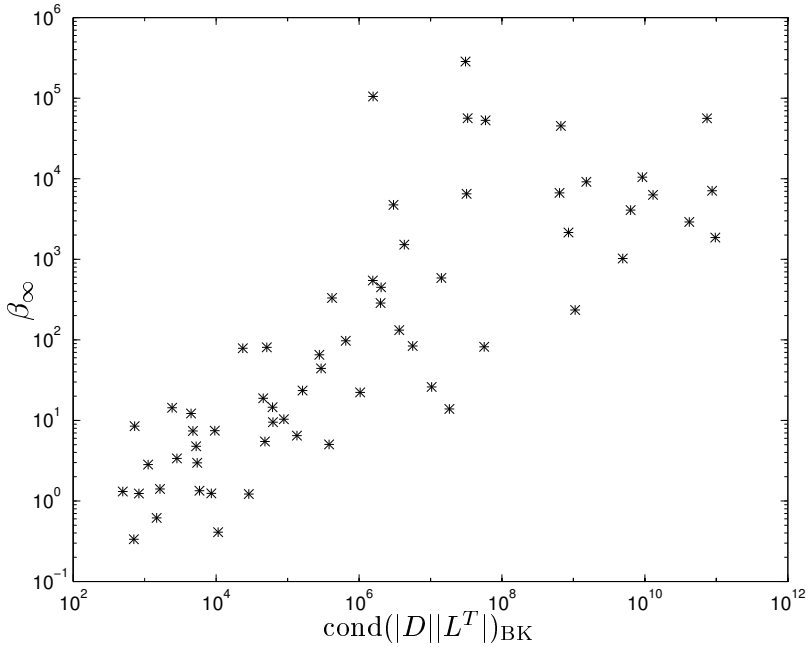

Figure 2.3: Comparison of relative normwise forward error on scaled $N(0,1)$ matrices with $m = 3$, $n = 50$.

and $\mathrm{cond}(|D||L^T|)_{\mathrm{BK}}$. Here we use the convention that $z/0 = 0$ if $z = 0$ and infinity otherwise. This brings us back to consider the componentwise backward stability, where an important measure is ξ defined as in (2.7). If ξ is small then τ is more likely to be small. Figure 2.5 shows an increasing trend between the ratio $\xi_{\mathrm{BK}}/\xi_{\mathrm{BBK}}$ and $\tau_{\mathrm{BK}}/\tau_{\mathrm{BBK}}$. Thus, we can view the Ashcraft, Grimes and Lewis examples as a special case for which large a priori componentwise backward bounds are attained for the BK strategy but not for the BBK strategy. This is best explained by the following example.

Let

$$A = \begin{bmatrix} 5.6454\text{e–}19 & 3.0242\text{e–}07 & 7.5198\text{e–}07 & 4.7523\text{e–}07 \\ & 5.4618\text{e–}19 & 5.7984\text{e–}07 & 3.9042\text{e–}07 \\ & & 9.6075\text{e–}02 & 3.5680\text{e–}01 \\ & & & 1.5935\text{e–}02 \end{bmatrix}, \tag{2.15}$$

which is scaled using the scheme described in Figure 2.2 with $m = 2$, $n = 4$,

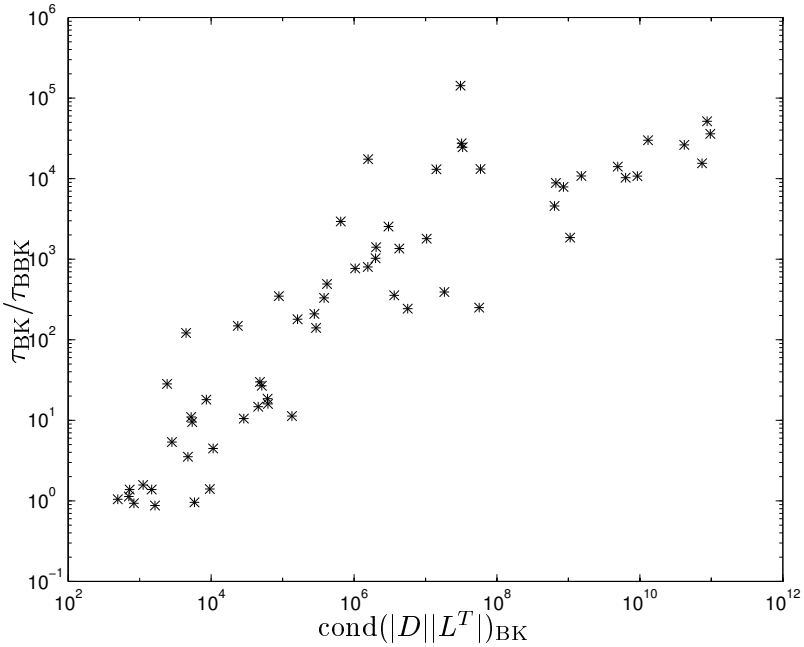

Figure 2.4: Comparison of relative componentwise forward error bound on scaled $N(0,1)$ matrices with $m = 3$, $n = 50$.

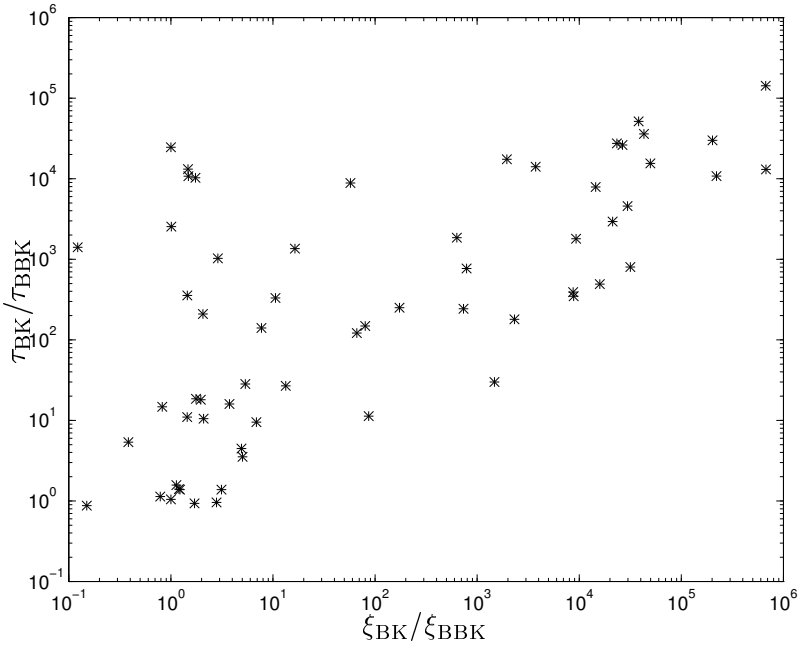

Figure 2.5: Comparison of relative componentwise forward error bound on scaled $N(0,1)$ matrices with $m = 3$, $n = 50$.

$\sigma_1 = 10^{-18}$, $\sigma_2 = 10^{-6}$ and $\sigma_3 = 10^{-1}$. For the BK pivoting strategy,

$$P^T|L||D||L^T|P = \begin{bmatrix} 5.6454\text{e–}19 & 3.0242\text{e–}07 & 7.5198\text{e–}07 & 4.7523\text{e–}07 \\ & \underline{\textit{1.3364}\text{e–}01} & \underline{\textit{7.7276}\text{e–}02} & \underline{\textit{2.8698}\text{e–}01} \\ & & 9.6075\text{e–}02 & 3.5680\text{e–}01 \\ & & & \underline{\textit{8.0927}\text{e–}01} \end{bmatrix},$$

where the italic and underlined entries are those that have changed order compared with A. Similarly, for the BBK pivoting strategy, we have

$$P^T|L||D||L^T|P = \begin{bmatrix} \underline{\textit{3.5668}\text{e–}12} & 3.0242\text{e–}07 & 7.5198\text{e–}07 & 4.7523\text{e–}07 \\ & \underline{\textit{2.2508}\text{e–}12} & 5.7984\text{e–}07 & 3.9042\text{e–}07 \\ & & 9.6075\text{e–}02 & 3.5680\text{e–}01 \\ & & & 1.5935\text{e–}02 \end{bmatrix}.$$

For matrix (2.15), $\xi_{\mathrm{BK}}/\xi_{\mathrm{BBK}} = 3.0 \times 10^{10}$ and $\tau_{\mathrm{BK}}/\tau_{\mathrm{BBK}} = 4.0 \times 10^{10}$. Thus a much larger componentwise backward error bound is obtained by the BK strategy. In this case, the BBK strategy is superior to the BK strategy.

However, from the discussion in Sections 2.4.4 and 2.4.5, we know that neither the BK strategy nor the BBK strategy is better than the other from the point of view of componentwise backward stability in general, and a large $\|W\|_\infty$ is only a necessary condition for componentwise forward instability.

2.6 Concluding Remarks

We have investigated the accuracy and stability of the diagonal pivoting method with two related pivoting strategies, namely the Bunch–Kaufman pivoting strategy and the Bounded Bunch–Kaufman pivoting strategy. Theoretical analyses and numerical examples demonstrate that the claim of superior accuracy of the BBK pivoting strategy is not fully justified.

For solving linear systems $Ax = b$, the unbounded factor L arising from the Bunch–Kaufman pivoting strategy has no effect on the backward stability or the normwise forward stability. We have confirmed that better accuracy is achieved by the BBK strategy when tested on the Ashcraft, Grimes and Lewis examples [6]. The significance of these examples is not clear, however. It may be possible that a class of numerical examples can be found where BK is more accurate than BBK. Further work is needed to produce clear statements about the relative accuracy of the BK and BBK strategies.

Chapter 3

Accuracy and Stability of Aasen's Method

3.1 Introduction

Another important direct method for solving dense symmetric indefinite linear systems is Aasen's method with partial pivoting [1] which computes an LTL$^{\mathrm{T}}$ factorization of a symmetric matrix $A \in \mathbb{R}^{n\times n}$

$$PAP^T = LTL^T, \tag{3.1}$$

where L is unit lower triangular with first column e_1 and

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix}$$

is tridiagonal. P is a permutation matrix chosen such that $|l_{ij}| \leq 1$.

To solve a linear system $Ax = b$ using the factorization $PAP^T = LTL^T$ we solve in turn

$$Lz = P^T b, \quad Ty = z, \quad L^T w = y, \quad x = Pw. \tag{3.2}$$

The symmetric indefinite tridiagonal system $Ty = z$ is usually solved in $O(n)$ flops using Gaussian elimination with partial pivoting. The disregard of symmetry at this level has little consequence since the overall process is $O(n^3)$ flops.

Aasen's method with partial pivoting is the only known stable direct method for solving symmetric indefinite linear systems with a guarantee of no more than $n^2/2$ comparisons and a bounded factor L. The operation count of Aasen's method with partial pivoting is the same, up to the highest order terms ($n^3/3$ flops), as that of the diagonal pivoting method with the Bunch–Kaufman pivoting strategy, described in Chapter 2. Despite the advantages, Aasen's method has largely been neglected for the last decade. Neither LAPACK [2] nor LINPACK [29] has an implementation of Aasen's method. Since 1993, the Visual Numerics, Inc. has included Aasen's method in their IMSL Fortran 90 MP Library [48], [90].

In the 1970s, Barwell and George [7] compared the performance of the diagonal pivoting method with the Bunch–Kaufman partial pivoting strategy with that of Aasen's method in unblocked form, on serial computers such as the IBM 360/75 and Honeywell 6050. They concluded that the difference in performance of the algorithms, in terms of execution time, is insignificant and is compiler dependent. A more recent LAPACK project report [3] compared unblocked and blocked versions of these two algorithms. The authors reported that Aasen's method with partial pivoting was faster asymptotically in the unblocked case and slower in the blocked case. However, some limitations of the report are explained in [6]:

> "Unfortunately, this report is somewhat incomplete in that no details of the blocked algorithms were given and only factorization times were considered. The test codes used are apparently lost. Further, the range of machines is limited and obsolete."

In fact the testing was only done on a Cray 2 computer with 1 processor in which floating point arithmetic does not utilize a guard digit. It is an open question which method is more computationally efficient in the context of parallel computation.

To describe Aasen's method it is convenient first to describe the Parlett and

Reid method; see Section 3.2. The methods are mathematically identical. Aasen's method is computationally more efficient because of its ingenious reordering of the tridiagonalization calculation which allows further exploitation of symmetry and structure. We present Aasen's method in Section 3.3.

Often the stability of Aasen's method is taken for granted. No backward stability result exists in the literature. In Section 3.4, we state a backward stability result of Higham [57] for which the tridiagonal system is solved by Gaussian elimination with partial pivoting.

One important practical issue concerning the stability of algorithms is the growth factor. We know very little about the behaviour of the growth factor in Aasen's method. Direct search methods [53] were employed to search for large growth factors and the results are reported in Section 3.5. In Section 3.6, we present our conclusions and identify some open problems.

3.2 The Parlett and Reid Method

We now explain how the Parlett and Reid method works. The first stage of the algorithm can be expressed as follows. If the symmetric matrix $A \in \mathbb{R}^{n\times n}$ is nonzero, we can find a permutation Π so that

$$\Pi A \Pi^T = \begin{array}{c} \\ 1 \\ 1 \\ n-2 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 1 & n-2 \end{array} \\ \begin{bmatrix} \alpha_1 & \beta_1 & y^T \\ \beta_1 & \alpha_2 & v^T \\ y & v & B \end{bmatrix} \end{array},$$

with β_1 the largest subdiagonal element in absolute value in the first column. If $\beta_1 = 0$ then no modification is needed and we proceed to the next stage. For β_1

nonzero, we can factorize

$$\Pi A \Pi^T = \begin{bmatrix} 1 & & \\ 0 & 1 & \\ 0 & w & I_{n-2} \end{bmatrix} \begin{bmatrix} \alpha_1 & \beta_1 & 0 \\ \beta_1 & \alpha_2 & v^T - \alpha_2 w^T \\ 0 & v - \alpha_2 w & C \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ & 1 & w^T \\ & & I_{n-2} \end{bmatrix},$$

where $w = y/\beta_1$ and $C = B - wv^T - vw^T + \alpha_2 ww^T$. The process is repeated recursively on the $(n-1) \times (n-1)$ submatrix

$$S = \begin{bmatrix} \alpha_2 & v^T - w^T \\ v - w & C \end{bmatrix},$$

yielding the factorization (3.1) on completion. This factorization costs $2n^3/3$ operations (twice the cost as block $\mathrm{LDL}^{\mathrm{T}}$ factorization with the Bunch–Kaufman pivoting strategy) plus $n^2/2$ comparisons. Hence Parlett and Reid's method is uncompetitive with the block $\mathrm{LDL}^{\mathrm{T}}$ factorization with the Bunch–Kaufman pivoting strategy mentioned in Chapter 2. In next section, we explain how Aasen's method exploits symmetry and hence halves the cost of the factorization.

3.3 Aasen's Method

For convenience, we assume, without loss of generality, that no interchanges are needed, which amounts to redefining $A := PAP^T$ in (3.1). To derive Aasen's method, assume that the first $i-1$ columns of T and the first i columns of L are known. We show how to compute the ith column of T and the $(i+1)$st column of L. A key role is played by the matrix

$$H = TL^T, \tag{3.3}$$

which is easily seen to be upper Hessenberg matrix. Equating the ith column in (3.3) we obtain

$$\begin{bmatrix} \underline{h_{1i}} \\ \underline{h_{2i}} \\ \vdots \\ \underline{h_{i-1,i}} \\ \underline{h_{ii}} \\ \underline{h_{i+1,i}} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = T \begin{bmatrix} l_{i1} \\ l_{i2} \\ \vdots \\ l_{i,i-1} \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha_1 l_{i1} + \beta_1 l_{i2} \\ \beta_1 l_{i1} + \alpha_2 l_{i2} + \beta_2 l_{i3} \\ \vdots \\ \beta_{i-2} l_{i,i-2} + \alpha_{i-1} l_{i,i-1} + \beta_{i-1} \\ \beta_{i-1} l_{i,i-1} + \underline{\alpha_i} \\ \underline{\beta_i} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3.4}$$

We use an underline to denote an unknown quantity to be determined.

The first $i-1$ equations in (3.4) are used to compute $h_{1,i}, \ldots, h_{i-1,i}$. The next two equations contain two unknowns each so cannot yet be used. The (i,i) and $(i+1,i)$ elements of the equation $A = LH$ give

$$a_{ii} = \sum_{j=1}^{i-1} l_{ij} h_{ji} + \underline{h_{ii}}, \tag{3.5}$$

$$a_{i+1,i} = \sum_{j=1}^{i} l_{i+1,j} h_{ji} + \underline{h_{i+1,i}}, \tag{3.6}$$

which we solve for h_{ii} and $h_{i+1,i}$. Now we can return to the last two nontrivial equations of (3.4) to obtain α_i and β_i. Finally, the ith column of the equation $A = LH$ yields

$$a_{ki} = \sum_{j=1}^{i+1} l_{kj} h_{ji}, \qquad k = i+2{:}n,$$

which yields the elements below the diagonal in the $(i+1)$st column of L:

$$l_{k,i+1} = \frac{1}{h_{i+1,i}} (a_{ki} - \sum_{j=1}^{i} l_{kj} h_{ji}), \qquad k = i+2{:}n. \tag{3.7}$$

The factorization has thereby been advanced by one step.

Clearly, equations (3.2), (3.4)–(3.6) are all $O(n^2)$ flops processes. To derive the leading order of the operation cost, we need only to consider the most expensive loop (3.7). For each $l_{k,i+1}$, it costs $2i+1$ flops. Hence the $(i+1)$st column costs $(n-i-2)(2i+1)$ flops in total. In completion of L we need

$$\sum_{i=1}^{n-2}(n-i-2)(2i+1) = 2\sum_{i=1}^{n-2}(ni-i^2) + O(n^2) = \frac{n^3}{3} + O(n^2) \text{ flops.}$$

3.4 Numerical Stability

Our model of floating point arithmetic is the usual model (1.2). The following backward stability result is proved by Higham [57].

Theorem 3.4.1 (Higham) *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and let $\widehat{x}$ be a computed solution to the linear system $Ax = b$ produced by Aasen's method with partial pivoting. Then*

$$(A+\Delta A)\widehat{x} = b, \qquad |\Delta A| \le \gamma_{3n+1}P^T|\widehat{L}||\widehat{T}||\widehat{L}^T|P + \gamma_{2n+4}P^T|\widehat{L}|\Pi^T|\widehat{M}||\widehat{U}||\widehat{L}^T|P,$$

where $\Pi\widehat{T} \approx \widehat{M}\widehat{U}$ and $PAP^T \approx \widehat{L}\widehat{T}\widehat{L}^T$ are the computed factorizations produced by LU factorization with partial pivoting and LTL^T factorization with partial pivoting respectively. Moreover

$$\|\Delta A\|_\infty \le (n-1)^2\gamma_{15n+25}\|\widehat{T}\|_\infty. \qquad \square$$

Theorem 3.4.1 shows that Aasen's method is a backward stable method for solving $Ax = b$ provided that the growth factor

$$\rho_n(A) = \frac{\max_{i,j}|t_{ij}|}{\max_{i,j}|a_{ij}|} \tag{3.8}$$

is not too large. Here, we are making the reasonable assumption that $\max_{i,j}|t_{ij}| \approx \max_{i,j}|\widehat{t}_{ij}|$ [55, p.177].

3.5 The Growth Factor

In this section, we bound the growth factor for Aasen's method with partial pivoting and investigate whether the bound is attainable using a combination of direct search methods described in [28], [53], [86], [87].

First we bound the growth factor. Using the fact that the multipliers in Aasen's method with partial pivoting are bounded by 1, it is straightforward to show that if $\max_{i,j} |a_{ij}| = 1$ then T has a bound illustrated for $n = 5$ by

$$|T| \leq \begin{bmatrix} 1 & 1 & & & \\ 1 & 1 & 2 & & \\ & 2 & 4 & 8 & \\ & & 8 & 16 & 32 \\ & & & 32 & 64 \end{bmatrix}.$$

Hence

$$\rho_n(A) \leq 4^{n-2}.$$

This upper bound is attainable for $n = 3$, as is shown by the example

$$A = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & \\ 0 & 1 & \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & \\ -1 & 1 & 2 \\ & 2 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ & 1 & -1 \\ & & 1 \end{bmatrix} = LTL^T. \quad (3.9)$$

For $n \geq 4$, we were unable to construct such an example. It is an open question whether this upper bound is attainable.

One useful approach to investigate the numerical instability of an algorithm is to rephrase the question as an optimization problem and apply a direct search method.

In our case, the growth factor is expressed as a function $f : \mathbb{R}^n \to \mathbb{R}$. To obtain an optimization problem we let $x = \text{vec}(A) \in \mathbb{R}^{(n^2+n)/2}$, where $\text{vec}(A)$ comprises the columns of the upper triangular part of A strung out into one long

vector, and we define $f(x) = \rho_n(A)$ where $\rho_n(A)$ is defined in (3.8). Then we wish to determine

$$\max_{x \in \mathbb{R}^{(n^2+n)/2}} f(x) \equiv \max_{A=A^T \in \mathbb{R}^{n \times n}} \rho_n(A). \tag{3.10}$$

Direct search methods are usually based on heuristics that do not involve assumptions about the function f. Only function values are used and no derivative estimate of f is required. The main disadvantages are that the convergence is at best linear and the nature of the point at which the methods terminate is not known since derivatives are not calculated [53]. Nevertheless it provides a convenient starting point in tackling the problem when limited information is known about f.

We have used three direct search methods implemented in the MATLAB Test Matrix Toolbox [54]. They are the alternating directions method (`adsmax.m`) [53], the multidirectional search method (`mdsmax.m`) of Dennis and Torczon [86], [87], and the Nelder-Mead simplex method (`nmsmax.m`) [28].

For our optimization problem (3.10) with $n = 3$, starting naively with initial matrix $A = I$ and default tolerance 10^{-3}, `mdsmax.m` needed only 8 iterations and 139 function evaluations to converge. It gave $\rho_n(A) = 3.9944$, where

$$A = \begin{bmatrix} -0.6370 & -0.0835 & -0.0835 \\ -0.0835 & 1.0000 & -0.9972 \\ -0.0835 & -0.9972 & 1.0000 \end{bmatrix}$$

is a different form of matrix than our example (3.9).

In our remaining experiments, we started the search with a random vector with default tolerance which is set to 10^{-3} for all three routines. When one of them converged, we restarted the search using a different method until all three of them converged to the given tolerance. Then the search was restarted with a smaller tolerance. The tolerance was reduced gradually to 10^{-15}.

For $n = 4$ and 5, the largest growth factors found so far are 7.99 and 14.61 respectively, compared with the bounds of 16 and 64. It is an open question to determine a sharp bound for the growth factor. However, unsuccessful optimizations can also provide useful information. As Miller and Spooner explain [66, p. 370],

> "Failure of the maximizer to find large values of ω (say) can be interpreted as providing evidence for stability equivalent to a large amount of practical experience with low-order matrices."

3.6 Concluding Remarks

Aasen's method is the only stable direct method with $O(n^2/2)$ comparisons and a bounded factor L, for solving symmetric indefinite linear system $Ax = b$. The diagonal pivoting method with the Bunch–Kaufman pivoting strategy and Aasen's method are competitive in terms of speed for dense matrices. It is not clear which method is more efficient for sparse matrices and on parallel architectures. More testing on a wider range of machines is desirable.

An open problem is to construct matrices for which the growth factor bound is attained for $n \geq 4$, or to derive a sharper growth factor bound for Aasen's method.

Chapter 4

Modified Cholesky Algorithms

4.1 Introduction

A standard method for solving unconstrained optimization problems is Newton's method. Given a twice continuously differentiable function $F(x) : \mathbb{R}^n \to \mathbb{R}$, let first and second derivatives of $F(x)$ be known at the iterate $x^{(k)}$. A local *quadratic model* of $F(x)$ can be obtained using the first three terms of Taylor's series at $x^{(k)}$, that is,

$$F(x^{(k)} + p) \approx F^{(k)} + g^{(k)T}p + \frac{1}{2}p^T G^{(k)} p,$$

where $F^{(k)} = F(x^{(k)})$, $g^{(k)} = \left(\frac{\partial F}{\partial x_i}\right)_{x=x^{(k)}} = \nabla F|_{x=x^{(k)}}$ is the *gradient* of F at $x^{(k)}$, $G^{(k)} = \left(\frac{\partial^2 F}{\partial x_i \partial x_j}\right)_{x=x^{(k)}}$ is the *Hessian* matrix and p is the search direction. The minimum of the quadratic model is attained when $g^{(k)T}p + \frac{1}{2}p^T G^{(k)} p$ is minimized. For a stationary point, we have $\nabla(g^{(k)T}p + \frac{1}{2}p^T G^{(k)} p) = 0$, which gives

$$G^{(k)}\widehat{p} = -g^{(k)}. \tag{4.1}$$

The search direction $\widehat{p}$ satisfying (4.1) is called the *Newton direction* and the minimization algorithm that takes a unit step in this direction at each stage is *Newton's method*. In practice, a line search is incorporated, that is, $x^{(k+1)} = x^{(k)} + \alpha_k \widehat{p}$ is used instead, where α_k is chosen so that $F(x^{(k)} + \alpha_k \widehat{p})$ is minimized.

If $G^{(k)}$ is positive definite, a descent direction $\widehat{p}$ is guaranteed since $g^{(k)T}\widehat{p} = -g^{(k)T}(G^{(k)})^{-1}g^{(k)} < 0$. In practice, we want $g^{(k)T}\widehat{p} < -\delta$, for some positive constant δ, so that F can be "sufficiently reduced" for small enough α_k, which is essential to achieve convergence [36]. In any case, if $\widehat{p}$ satisfies some modified version of (4.1), we called the algorithm a *modified Newton method*.

Given an indefinite Hessian matrix, one popular approach is to compute a "nearby" positive definite matrix to the original indefinite one. A natural approach is to combine a matrix factorization like the Cholesky (LDL^T) factorization with a modification scheme. This widely used technique is the so-called modified Cholesky factorization [37], [78]. In this chapter, we describe the two existing modified Cholesky factorizations and propose two alternative modification schemes.

Given a symmetric matrix and not necessarily positive definite matrix A, a modified Cholesky algorithm produces a symmetric perturbation E such that $A+E$ is positive definite, along with a Cholesky (or LDL^T) factorization of $A+E$. The objectives of a modified Cholesky algorithm can be stated as follows [78].

O1. If A is "sufficiently positive definite" then E should be zero.

O2. If A is indefinite, $\|E\|$ should not be much larger than

$$\min \{ \|\Delta A\| : A + \Delta A \text{ is positive definite} \},$$

for some appropriate norm.

O3. The matrix $A + E$ should be reasonably well conditioned.

O4. The cost of the algorithm should be the same as the cost of standard Cholesky factorization to highest order terms.

Two existing modified Cholesky algorithms are one of Gill, Murray and Wright (the GMW algorithm) [37, Section 4.4.2.2], which is a refinement of an earlier algorithm of Gill and Murray [36], and an algorithm of Schnabel and Eskow (the SE algorithm) [78].

We explain the GMW and SE algorithms in Sections 4.2 and 4.3 respectively. The GMW and SE algorithms both increase the diagonal entries as necessary in order to ensure that negative pivots are avoided. Hence both algorithms produce

Cholesky factors of $P(A+E)P$ with a diagonal E, where P is a permutation matrix.

In Section 4.4, we show that the "optimal" perturbation in objective (O2) is, in general, full for the Frobenius norm and can be taken to be diagonal for the 2-norm (but is generally not unique). There seems to be no particular advantage to making a diagonal perturbation to A. We propose an alternative modified Cholesky algorithm based on the block $\mathrm{LDL}^{\mathrm{T}}$ factorization with the bounded Bunch–Kaufman (BBK) pivoting strategy described in Chapter 2. Our algorithm perturbs the whole matrix, in general. However, it is suitable even for sparse matrices since our proposed modification scheme keeps the sparsity of the factors L and D.

In outline, our approach is to compute a block $\mathrm{LDL}^{\mathrm{T}}$ factorization

$$PAP^T = LDL^T, \tag{4.2}$$

where P is a permutation matrix, L is unit lower triangular and D is block diagonal with diagonal blocks of dimension 1 or 2, and to provide the factorization

$$P(A+E)P^T = L(D+F)L^T,$$

where F is chosen so that $D+F$ (and hence also $A+E$) is positive definite.

This approach is not new; it was suggested by Moré and Sorensen [67] for use with the block $\mathrm{LDL}^{\mathrm{T}}$ factorization computed with the Bunch–Kaufman [12] and Bunch–Parlett [14] pivoting strategies. However, for neither of these pivoting strategies are all the conditions (O1)–(O4) satisfied, as is recognized in [67]. The Bunch–Parlett pivoting strategy requires $O(n^3)$ comparisons for an $n \times n$ matrix, so condition (O4) does not hold. For the Bunch–Kaufman strategy, which requires only $O(n^2)$ comparisons, it is difficult to satisfy conditions (O1)–(O3), as we explain in Section 4.4.

There are two reasons why our algorithm might be preferred to those of Gill, Murray and Wright, and of Schnabel and Eskow. The first is a pragmatic one: we

can make use of any available implementation of the form (4.2), needing to add just a small amount of post-processing code to form the modified factorization. In particular, we can use the efficient implementations for both dense and sparse matrices written by Ashcraft, Grimes and Lewis [6], which make extensive use of level 2 and 3 BLAS for efficiency on high-performance machines. In contrast, in coding the GMW and SE algorithms one must either begin from scratch or make non-trivial changes to an existing Cholesky factorization code.

The second attraction of our approach is that we have a priori bounds that explain the extent to which conditions (O1)–(O3) are satisfied—essentially, if L is well conditioned then an excellent modified factorization is guaranteed. For the GMW and SE algorithms it is difficult to describe under what circumstances the algorithms can be guaranteed to perform well.

Note that the analysis in Section 4.4 works for all congruence transformations. In Section 4.5 we describe a modified Aasen algorithm based on Aasen's method. Numerical results are presented in Section 4.7. We give conclusions and directions for future work in Section 4.8.

4.2 The Gill, Murray and Wright Algorithm

In this section, we summarize the algorithm of Gill, Murray and Wright (the GMW algorithm) [37], which is designed to satisfy the four objectives stated in Section 4.1. The GWM algorithm is based on the $\mathrm{LDL}^{\mathrm{T}}$ factorization (or the Cholesky factorization) and modifies only diagonal elements, that is, given $A \in \mathbb{R}^{n\times n}$, we compute

$$PAP^T + E = L\widehat{D}L^T, \tag{4.3}$$

where P is a permutation matrix, L is unit lower triangular, $\widehat{D} = \mathrm{diag}(\widehat{d}_i)$ and $E = \mathrm{diag}(e_i)$. Here $\widehat{d}_i$ is chosen so that

$$\widehat{d}_i \geq \max\{|d^{(i)}|, \delta\} \qquad \text{and} \qquad l_{ij}\widehat{d}_j^{1/2} \leq \xi, \tag{4.4}$$

for some suitable positive constants δ and ξ, where $d^{(i)}$ is the "natural" pivot at the ith stage of the factorization. The choice of ξ and δ will be discussed later.

For any symmetric matrix $A \in \mathbb{R}^{n\times n}$, let $A^{(k)} \in \mathbb{R}^{(n-k+1)\times(n-k+1)}$ denote the Schur complement remaining at the kth stage of the factorization, for $k = 1{:}n$ ($A^{(1)} = A$). We can find a permutation Π so that

$$\Pi A^{(k)}\Pi^T = \begin{array}{c} \\ 1 \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} 1 & n-k \end{array} \\ \begin{bmatrix} d^{(k)} & c^{(k)T} \\ c^{(k)} & B^{(k)} \end{bmatrix} \end{array}, \tag{4.5}$$

with $d^{(k)}$ the maximum diagonal element in magnitude. Having chosen such a Π we can factorize

$$\Pi A^{(k)}\Pi^T + E_k = \begin{bmatrix} 1 & 0 \\ \frac{1}{\widehat{d}_k}c^{(k)} & I_{n-k} \end{bmatrix} \begin{bmatrix} \widehat{d}_k & 0 \\ 0 & A^{(k+1)} \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{\widehat{d}_k}c^T \\ 0 & I_{n-k} \end{bmatrix}, \tag{4.6}$$

where $A^{(k+1)} = B^{(k)} - \frac{1}{\widehat{d}_k}c^{(k)}c^{(k)T}$. Here $\widehat{d}_k = d^{(k)} + e_k$ is chosen to satisfy (4.4) and

$$E_k = \begin{cases} e_k, & \text{at the (1,1) entry,} \\ 0, & \text{otherwise.} \end{cases}$$

Thus we have, on squaring the second inequality in (4.4),

$$\widehat{d}_k \geq \max\{|d^{(k)}|, \delta\}, \qquad 0 \leq \frac{\|c^{(k)}\|_\infty^2}{\widehat{d}_k} \leq \xi^2. \tag{4.7}$$

In order to minimize $\widehat{d}_k$ and hence e_k, we have

$$\widehat{d}_k = \max\left\{|d^{(k)}|, \delta, \frac{\|c^{(k)}\|_\infty^2}{\xi^2}\right\}.$$

Hence

$$|e_k| = |\widehat{d}_k - d^{(k)}| \leq \max\left\{|d^{(k)}|, \delta, \frac{\|c^{(k)}\|_\infty^2}{\xi^2}\right\} + |d^{(k)}|$$
$$\leq \frac{\|c^{(k)}\|_\infty^2}{\xi^2} + 2|d^{(k)}| + \delta. \tag{4.8}$$

This process is repeated recursively on the Schur complement $A^{(k+1)}$ yielding the factorization (4.3) on completion.

Gill and Murray [36] derive an upper bound for $\|E(\xi)\|_2$, depending on ξ, using the following lemma.

Lemma 4.2.1 (Gill and Murray) *Let $A \in \mathbb{R}^{n\times n}$ and ξ be defined as in* (4.4). *Let $A^{(k)}$ denote the Schur complement at the kth stage of the GMW algorithm. We have*

$$|a_{ii}^{(k)}| \leq \alpha + (k-1)\xi^2, \qquad |a_{ij}^{(k)}| \leq \beta + (k-1)\xi^2, \quad i \neq j, \tag{4.9}$$

where $\alpha = \max_i |a_{ii}|$ and $\beta = \max_{i\neq j} |a_{ij}|$.

Proof: The proof is by induction. For $k = 1$, (4.9) is trivially true since $A^{(1)} = A$. Assume (4.9) is true for $k = m$. For $k = m + 1$, using the second inequality in (4.7), we have

$$|a_{ii}^{(m+1)}| \leq |b_{ii}^{(m)}| + |\widehat{d}_m^{-1}(c_i^{(m)})^2| \leq \alpha + m\xi^2,$$
$$|a_{ij}^{(m+1)}| \leq |b_{ij}^{(m)}| + |\widehat{d}_m^{-1}c_i^{(m)}c_j^{(m)}| \leq \beta + m\xi^2,$$

as required. □

Using Lemma 4.2.1 and (4.8), it is easily shown that

$$\|E(\xi)\|_2 = \max_i |e_i| \leq \Big(\frac{\beta}{\xi} + (n-1)\xi\Big)^2 + 2\big(\alpha + (n-1)\xi^2\big) + \delta. \tag{4.10}$$

It remains to describe the choice of ξ and δ. For ξ, note that the bound of $\|E(\xi)\|_2$ is a convex function of ξ and is minimized when $\xi^2 = \beta/\sqrt{n^2-1}$ [36].

In addition, a lower bound for ξ is required so that objective (O1) is satisfied, that is, $\|E(\xi)\|_2 = 0$ when A is sufficiently positive definite. When no modification is added, the GMW algorithm is just the standard $\mathrm{LDL}^{\mathrm{T}}$ factorization with complete pivoting and $a_{ii} = \sum_{j=1}^{i} l_{ij}^2 d_j$ with $d_j > 0$. This implies

$$0 < l_{ij}^2 d_j \leq a_{ii} \leq \max_k a_{kk} = \alpha.$$

If $\alpha \leq \xi^2$, (4.4) guarantees no modification is made. So the final choice of ξ is

$$\xi^2 = \max\{\alpha, \beta/\sqrt{n^2 - 1}, u\},$$

where u is the unit roundoff and is introduced to allow for the case when $A = 0$.

No detail of how δ is chosen is given in [37]. The default value of δ in Margaret Wright's MATLAB code for the GMW algorithm is $\delta = 2u \max\{\alpha + \beta, 1\}$.

At each stage, $\|c^{(k)}\|_\infty$ is computed to determine the amount of perturbation, hence the extra cost induced is $n^2/2$ to the highest order. The cost of the pivoting is also $n^2/2$ to the highest order. Thus objective (O4) is trivially satisfied.

4.3 The Schnabel and Eskow Algorithm

The algorithm of Schnabel and Eskow (the SE algorithm) is broadly similar to the GMW algorithm. The SE algorithm also carries out the $\mathrm{LDL}^{\mathrm{T}}$ factorization using the Gershgorin circle theorem to determine the size of the perturbation and the choice of pivot, and has a two-phase strategy. Given $A \in \mathbb{R}^{n \times n}$, the SE algorithm computes

$$PAP^T + E = L\widehat{D}L^T,$$

where P is a permutation matrix, L is unit lower triangular, $\widehat{D} = \mathrm{diag}(\widehat{d}_i)$ and $E = \mathrm{diag}(e_i)$.

The two-phase strategy is used to avoid perturbing sufficiently positive-definite matrices. In the first phase, the $\mathrm{LDL}^{\mathrm{T}}$ factorization with complete pivoting (that

is, pivoting along the diagonal) is used. We switch to the second phase of the algorithm if the minimum diagonal element in the Schur complement is smaller than δ, where $\delta = \alpha\tau$ and $\tau = u^{1/3}$ is recommended in [78].

No modification is made in the first phase and the element growth is bounded using the following lemma (cf. Schnabel and Eskow [78, Thm. 5.2.1]).

Lemma 4.3.1 *Let $A \in \mathbb{R}^{n\times n}$ and let δ be a positive constant. Let the first phase of the SE algorithm be completed at the end of the $(k-1)$st stage and switch to the second phase at the kth stage. Let $A^{(k)} \in \mathbb{R}^{(n-k+1)\times(n-k+1)}$ denote the Schur complement. If $\min_{k\le i\le n} a_{ii}^{(k)} \ge \delta$, then*

$$a_{ii}^{(k)} \le \alpha, \qquad |a_{ij}^{(k)}| \le \alpha + \beta - \delta, \quad i \ne j, \tag{4.11}$$

where $\alpha = \max_i |a_{ii}|$ and $\beta = \max_{i\ne j} |a_{ij}|$.

Proof: At the end of the $(k-1)$st stage, we have

$$\Pi A \Pi^T = \begin{array}{c} \\ k-1 \\ n-k+1 \end{array} \begin{array}{c} \begin{array}{cc} k-1 & n-k+1 \end{array} \\ \begin{bmatrix} H & C^T \\ C & B \end{bmatrix} \end{array} = \begin{bmatrix} L & 0 \\ M & I_{n-k+1} \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & A^{(k)} \end{bmatrix} \begin{bmatrix} L^T & M^T \\ 0 & I_{n-k+1} \end{bmatrix},$$

which gives

$$B = MDM^T + A^{(k)}.$$

This implies, for $i = k{:}n$,

$$\alpha \ge b_{ii} = \sum_{j=1}^{k-1} m_{ij}^2 d_j + a_{ii}^{(k)} > \delta,$$

thus $a_{ii}^{(k)} \le \alpha$ and $\sum_{j=1}^{k-1} m_{ij}^2 d_j \le \alpha - \delta$. Moreover, using the Cauchy–Schwartz inequality, we have

$$\begin{aligned} |a_{ij}^{(k)}| &\le |b_{ij}| + \Big|\sum_{r=1}^{k-1} m_{ir} d_r m_{rj}\Big| \\ &\le |b_{ij}| + \Big(\sum_{r=1}^{k-1} m_{ir}^2 d_r\Big)^{\frac{1}{2}} \Big(\sum_{r=1}^{k-1} m_{jr}^2 d_r\Big)^{\frac{1}{2}} \\ &\le \beta + \alpha - \delta, \end{aligned}$$

which completes the proof. □

If the first phase runs to completion, then the SE algorithm performs an LDL^T factorization with complete pivoting. The SE algorithm switches to the second phase at the kth stage if $\min_{k+1\le i\le n} a_{ii}^{(k+1)} < \delta$.

Once the algorithm switches to the second phase, the Gershgorin circle theorem is used to determine the amount of perturbation and the choice of pivot.

Theorem 4.3.2 (Gershgorin Circle Theorem) *Let $A \in \mathbb{C}^{n\times n}$. Then each eigenvalue λ_i of A lies in one of the disks in the complex plane*

$$G_i = \left\{ \lambda_i : |\lambda_i - a_{ii}| \le \sum_{j\ne i} |a_{ij}| \right\}, \qquad i = 1{:}n.$$

Proof: See [81]. □

For a real symmetric matrix, Theorem 4.3.2 says that all the eigenvalues lie in a union of real intervals $\{G_1 \cup \cdots \cup G_n\}$, where

$$G_i = [g_{i-}, g_{i+}] := \left[a_{ii} - \sum_{i\ne j} |a_{ij}|, a_{ii} + \sum_{i\ne j} |a_{ij}| \right]. \tag{4.12}$$

Let $A^{(k)} \in \mathbb{R}^{(n-k+1)\times(n-k+1)}$ denote the Schur complement of the kth stage of the factorization and have the Gershgorin intervals $G_i^{(k)}$, $i = k{:}n$, defined as in (4.12). Schnabel and Eskow [78] determine their choice of perturbation so that the Gershgorin intervals contract at each step of the factorization, that is, $G_i^{(k+1)} \subseteq G_i^{(k)}$, $i = k+1{:}n$. The following lemma is a modified version of [78, Lem. 5.1.1] in which we have introduced a positive tolerance δ. Recall that if

$$A^{(k)} = \begin{array}{c} \\ 1 \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} 1 & n-k \end{array} \\ \begin{bmatrix} d^{(k)} & c^{(k)T} \\ c^{(k)} & B^{(k)} \end{bmatrix} \end{array},$$

then $A^{(k+1)} = B^{(k)} - \frac{1}{\widehat{d}_k} c^{(k)} c^{(k)T}$ where $\widehat{d}_k = d^{(k)} + e_k$ is chosen to satisfy (4.13) in the following lemma.

Lemma 4.3.3 *Let $A^{(k)} \in \mathbb{R}^{(n-k+1)\times(n-k+1)}$ have the Gershgorin intervals $G_i^{(k)}$, $i = k{:}n$, defined as in (4.12). Let $d^{(k)}$ denote the pivot at the kth stage, defined as in (4.5), and let e_k denote the perturbation added to $d^{(k)}$. If*

$$\widehat{d}_k := d^{(k)} + e_k \geq \max\{d^{(k)}, \|c^{(k)}\|_1, \delta\}, \tag{4.13}$$

where $\widehat{d}_k$ is defined as in (4.6), then $G_i^{(k+1)} \subseteq G_i^{(k)}$, for $i = k+1{:}n$.

Proof: The choice of perturbation (4.13) ensures that $\widehat{d}_k > 0$. When $\widehat{d}_k > 0$, we have for $i = k+1{:}n$,

$$\begin{aligned} g_{i-}^{(k+1)} - g_{i-}^{(k)} &= a_{ii}^{(k+1)} - \sum_{\substack{j=k+1 \\ i\neq j}}^{n} |a_{ij}^{(k+1)}| - \Big[a_{ii}^{(k)} - \sum_{\substack{j=k \\ i\neq j}}^{n} |a_{ij}^{(k)}|\Big] \\ &= (a_{ii}^{(k+1)} - a_{ii}^{(k)}) + |a_{ik}^{(k)}| + \sum_{\substack{j=k+1 \\ i\neq j}}^{n} (|a_{ij}^{(k)}| - |a_{ij}^{(k+1)}|) \\ &\geq -\frac{c_i^{(k)2}}{\widehat{d}_k} + |c_i^{(k)}| - \frac{(\max\{\|c^{(k)}\|_1, \delta\} - |c_i^{(k)}|)|c_i^{(k)}|}{\widehat{d}_k} \\ &= \Big(1 - \frac{\max\{\|c^{(k)}\|_1, \delta\}}{\widehat{d}_k}\Big)|c_i^{(k)}| \geq 0, \end{aligned} \tag{4.14}$$

if $\widehat{d}_k \geq \max\{d^{(k)}, \|c^{(k)}\|_1, \delta\}$.

Similarly, we have for $i = k+1{:}n$,

$$g_{i+}^{(k+1)} - g_{i+}^{(k)} \leq \Big(-1 + \frac{\max\{\|c^{(k)}\|_1, \delta\}}{\widehat{d}_k} - \frac{2|c_i^{(k)}|}{\widehat{d}_k}\Big)|c_i^{(k)}| \leq 0.$$

Thus $G_i^{(k+1)} \subseteq G_i^{(k)}$. □

An immediate consequence is as follows.

Corollary 4.3.4 *Let $\lambda_{\min}^{(k)}$ and $\lambda_{\max}^{(k)}$ denote the minimum and maximum eigenvalues of $A^{(k)}$. Then for $k \leq i \leq n$,*

$$\max\{|\lambda_{\min}^{(i)}|, |\lambda_{\max}^{(i)}|\} \leq \alpha^{(k)} + (n-k)\beta^{(k)} + \delta,$$

where $\alpha^{(k)} = \max_i |a_{ii}^{(k)}|$, $\beta^{(k)} = \max_{i\neq j} |a_{ij}^{(k)}|$. □

The final choice of $\widehat{d}_i$ is

$$\widehat{d}_i = \max\{d^{(i)} + e_{i-1}, \|c^{(i)}\|_1, \delta\}, \tag{4.15}$$

which means that, at any stage, the perturbation e_i is as large as the perturbation on the previous stages. That is, $e_k \geq e_j$ for $k \geq j$. The reason is that the choice (4.15) does not change the value of $\|E\|_2$ and results in a larger $\widehat{d}_i$ if (4.15) is used rather than (4.13), which in turn will yield a smaller perturbation $\widehat{d}_i^{-1} c^{(i)} c^{(i)T}$. As Schnabel and Eskow [78] explain, "this reasoning does not imply that the final value of $\|E\|_2$ will be smaller using (4.15)..., but it makes this seem likely, and in practice the modification appears to be helpful in some cases and virtually never harmful."

Note that the proof of Lemma 4.3.3 is independent of the choice of pivot. For the pivoting strategy at the kth stage, the SE algorithm chooses the row for which the lower Gershgorin bound $g_{i-}^{(k)}$ is the largest. If $\max_i g_{i-}^{(k)} > 0$, then $e_k = 0$ and the Gershgorin intervals will contract.

This pivoting strategy is impractical because it assumes all the Gershgorin bounds for the remaining rows are known and it costs $O(n^3)$ operations overall. Instead, the SE algorithm approximates the lower bound of Gershgorin intervals using (4.14)

$$g_{i-}^{(k+1)} = g_{i-}^{(k)} + \Big(1 - \frac{\max\{\|c^{(k)}\|_1, \delta\}}{\widehat{d}_k}\Big)|c_i^{(k)}|.$$

The extra cost of the modification is $5n^2/2$ flops. Note that the estimate may be rather different from the exact bounds. However, the estimate is used only to determine the choice of pivot. Schnabel and Eskow [78] have shown by experiment that it does not significantly affect the performance of the SE algorithm.

At the final stage of the second phase, when only a 2×2 submatrix $A^{(n-1)}$

remains, we choose

$$\begin{aligned} e_{n-1} &= \widehat{d}_{n-1} - d^{(n-1)} \\ &= \max\left\{ e_{n-2}, -\lambda_{\min}^{(n-1)} + \max\left\{ \frac{\tau(\lambda_{\max}^{(n-1)} - \lambda_{\min}^{(n-1)})}{1-\tau}, \delta \right\} \right\} \\ &\leq \max\left\{ e_{n-2}, |\lambda_{\min}^{(n-1)}| + \frac{\tau(|\lambda_{\max}^{(n-1)}| + |\lambda_{\min}^{(n-1)}|)}{1-\tau} + \delta \right\}, \end{aligned}$$

where $\lambda_{\max}^{(n-1)}$ and $\lambda_{\min}^{(n-1)}$ denote the maximum and minimum eigenvalues of $A^{(n-1)}$ respectively.

Suppose the SE algorithm switches to the second phase at the $\widehat{k}$th stage. Using Lemma 4.3.1 and Corollary 4.3.4, we have

$$\begin{aligned} \|E\|_2 &\leq \frac{1+\tau}{1-\tau}\Big[\alpha^{(\widehat{k})} + (n-\widehat{k})\beta^{(\widehat{k})} + \delta\Big] + \delta \\ &\leq \frac{1+\tau}{1-\tau}\Big[\alpha + (n-\widehat{k})(\beta + \alpha - \delta)\Big] + \frac{2}{1-\tau}\delta \\ &\leq \frac{1+\tau}{1-\tau}[n(\alpha+\beta)] + \frac{2}{1-\tau}\delta, \end{aligned} \tag{4.16}$$

which is a smaller bound than that of the Gill, Murray and Wright algorithm defined as in (4.10) by a factor n.

4.4 The New Modified Cholesky Algorithm

We begin by defining the distance from a symmetric matrix $A \in \mathbb{R}^{n\times n}$ to the symmetric matrices with minimum eigenvalue $\lambda_{\min}$ at least δ, where $\delta \geq 0$:

$$\mu(A,\delta) = \min\left\{ \|\Delta A\| : \lambda_{\min}(A + \Delta A) \geq \delta \right\}. \tag{4.17}$$

The distance in the 2- and Frobenius norms, and perturbations that achieve them are easily evaluated (see Halmos [47], Higham [51, Thms. 2.1, 3.1]).

Theorem 4.4.1 *Let the symmetric matrix $A \in \mathbb{R}^{n\times n}$ have the spectral decomposition $A = Q\Lambda Q^T$ (Q orthogonal, $\Lambda = \operatorname{diag}(\lambda_i)$). Then, for the Frobenius norm,*

$$\mu_F(A,\delta) = \left(\sum_{\lambda_i<\delta}(\delta-\lambda_i)^2\right)^{1/2}$$

and there is a unique optimal perturbation in (4.17)*, given by*

$$\Delta A = Q\operatorname{diag}(\tau_i)Q^T, \qquad \tau_i = \begin{cases} 0, & \lambda_i \geq \delta, \\ \delta-\lambda_i, & \lambda_i < \delta. \end{cases} \tag{4.18}$$

For the 2-norm,

$$\mu_2(A,\delta) = \max(\,0, \delta - \lambda_{\min}(A)\,),$$

and an optimal perturbation is $\Delta A = \mu_2(A,\delta)I$. The Frobenius norm perturbation (4.18) *is also optimal in the 2-norm.*

Proof: Let $A + \Delta A = X$ be symmetric positive definite with $\lambda_{\min}(X) \geq \delta$, and $Y = Q^T X Q$. It is easily shown that $y_{ii} \geq \delta$. Then

$$\begin{aligned} \|A-X\|_F^2 &= \|\Lambda - Y\|_F^2 \\ &= \sum_{i\neq j} y_{ij}^2 + \sum_i (\lambda_i - y_{ii})^2 \\ &\geq \sum_{\lambda_i<\delta}(\lambda_i - y_{ii})^2 \geq \sum_{\lambda_i<\delta}(\lambda_i-\delta)^2, \end{aligned}$$

This lower bound is attained, uniquely, for the matrix $Y = \operatorname{diag}(d_i)$, where

$$d_i = \begin{cases} \lambda_i, & \lambda_i \geq \delta, \\ \delta, & \lambda_i < \delta. \end{cases}$$

The representation of ΔA follows, since $\Delta A = X - A$.

For the 2-norm perturbation, we make use of an inequality from [41, Col. 8.1.3]. We have

$$\delta = \lambda_{\min}(A+\Delta A) \leq \lambda_{\min}(A) + \lambda_{\max}(\Delta A),$$

where A, ΔA are symmetric. The equality is achieved when $\lambda_{\max}(\Delta A) = \delta - \lambda_{\min}(A)$, hence the result.

By taking the 2-norm of the perturbation, it is trivial to show that the optimal Frobenius norm perturbation is also an optimal 2-norm perturbation. □

Our modified Cholesky algorithm has a parameter $\delta \geq 0$ and it attempts to produce the perturbation (4.18).

Algorithm MC (Modified Cholesky Factorization) *Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ and a parameter $\delta \geq 0$ this algorithm computes a permutation matrix P, a unit lower triangular matrix L, and a block diagonal matrix D with diagonal blocks of dimension 1 or 2, such that*

$$P(A+E)P^T = LDL^T$$

and $A+E$ is symmetric positive definite (or symmetric positive semidefinite if $\delta = 0$). The algorithm attempts to ensure that if $\lambda_{\min} < \delta$ then $\lambda_{\min}(A+E) \approx \delta$.

1. Compute the symmetric indefinite factorization $PAP^T = L\widetilde{D}L^T$ using the BBK pivoting strategy (Algorithm BBK).

2. Let $D = \widetilde{D} + \Delta\widetilde{D}$, where $\Delta\widetilde{D}$ is the minimum Frobenius norm perturbation that achieves $\lambda_{\min}(\widetilde{D}+\Delta\widetilde{D}) \geq \delta$ (thus $\Delta\widetilde{D} = \mathrm{diag}(\Delta\widetilde{D}_{ii})$, where $\Delta\widetilde{D}_{ii}$ is the minimum Frobenius norm perturbation that achieves $\lambda_{\min}(\widetilde{D}_{ii}+\Delta\widetilde{D}_{ii}) \geq \delta$).

To what extent does Algorithm MC achieve the objectives (O1)–(O4) listed in Section 4.1? Objective (O4) is clearly satisfied, provided that the pivoting strategy does not require a large amount of searching, since the cost of step 2 is negligible. For objectives (O1)–(O3) to be satisfied we need the eigenvalues of A to be reasonably well approximated by those of $\widetilde{D}$. For the Bunch–Kaufman pivoting strategy the elements of L are unbounded and the eigenvalues of $\widetilde{D}$ can differ greatly from those of A (subject to A and $\widetilde{D}$ have the same inertia), as is

$\lambda(A)$	$\lambda(\widetilde{D}_{\mathrm{BBK}})$	$\lambda(\widetilde{D}_{\mathrm{BK}})$
−6.1e−01	−1.0e+00	−1.0e−05
1.0e−10	1.0e−10	1.0e−05
1.6e+00	1.0e+00	1.0e+00

Table 4.1: The eigenvalues of matrix (4.19) and the block diagonal matrix $\widetilde{D}$ when the BBK and BK pivoting strategies are used.

easily shown by example. Let

$$A = \begin{bmatrix} 0 & \epsilon & 0 \\ \epsilon & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \tag{4.19}$$

where $\epsilon = 10^{-5}$. Table 4.1 displays the eigenvalues of A and of the block diagonal $\widetilde{D}$ computed using the $\mathrm{LDL}^{\mathrm{T}}$ factorization with the BK and BBK strategies (denoted by $\widetilde{D}_{\mathrm{BK}}$ and $\widetilde{D}_{\mathrm{BBK}}$ respectively). The eigenvalues of $\widetilde{D}_{\mathrm{BK}}$ differ as much as an order of 5 from those of A. This is the essential reason why the Bunch–Kaufman pivoting strategy is unsuitable for use in a modified Cholesky algorithm.

To investigate objectives (O1)–(O3) we will make use of a theorem of Ostrowski [60, p. 224], [69]. Here the eigenvalues of a symmetric $n \times n$ matrix are ordered such that $\lambda_1 \leq \cdots \leq \lambda_n$.

Theorem 4.4.2 (Ostrowski) *Let $M \in \mathbb{R}^{n\times n}$ be symmetric and $S \in \mathbb{R}^{n\times n}$ nonsingular. Then for each k, $k = 1{:}n$,*

$$\lambda_k(SMS^T) = \theta_k \lambda_k(M)$$

where $\lambda_1(SS^T) \leq \theta_k \leq \lambda_n(SS^T)$. □

Note that the Sylvester law of inertia [41, Thm. 8.1.12] is a corollary of Ostrowski's theorem.

Assuming first $\lambda_{\min}(A) > 0$, and applying the theorem with $M = \widetilde{D}$ and $S = L$, we obtain

$$\lambda_{\min}(A) \leq \lambda_{\max}(LL^T)\lambda_{\min}(\widetilde{D}).$$

Now E will be zero if $\lambda_{\min}(\widetilde{D}) \geq \delta$, which is certainly true if

$$\lambda_{\min}(A) \geq \delta\lambda_{\max}(LL^T). \tag{4.20}$$

Next we assume that $\lambda_{\min}(A)$ is negative and apply Theorem 4.4.1 and Theorem 4.4.2 to obtain

$$\lambda_{\max}(\Delta\widetilde{D}) = \delta - \lambda_{\min}(\widetilde{D}) \leq \delta - \frac{\lambda_{\min}(A)}{\lambda_{\min}(LL^T)}. \tag{4.21}$$

Using Theorem 4.4.2 again, with (4.21), yields

$$\begin{aligned} \|E\|_2 = \lambda_{\max}(E) &= \lambda_{\max}(L\widetilde{D}L^T) \\ &\leq \lambda_{\max}(LL^T)\lambda_{\max}(\widetilde{D}) \\ &\leq \lambda_{\max}(LL^T)\left(\delta - \frac{\lambda_{\min}(A)}{\lambda_{\min}(LL^T)}\right), \end{aligned} \tag{4.22}$$

where $\lambda_{\min}(A) < 0$. A final invocation of Theorem 4.4.2 gives

$$\lambda_{\min}(A+E) \geq \lambda_{\min}(LL^T)\lambda_{\min}(\widetilde{D} + \Delta\widetilde{D}) \geq \lambda_{\min}(LL^T)\delta.$$

and

$$\begin{aligned} \|A+E\|_2 = \lambda_{\max}(A+E) &= \lambda_{\max}(L(\widetilde{D} + \Delta\widetilde{D})L^T) \\ &\leq \lambda_{\max}(LL^T)\lambda_{\max}(\widetilde{D} + \Delta\widetilde{D}) \\ &= \lambda_{\max}(LL^T)\max\{\delta, \lambda_{\max}(\widetilde{D})\} \\ &\leq \lambda_{\max}(LL^T)\max\left\{\delta, \frac{\lambda_{\max}(A)}{\lambda_{\min}(LL^T)}\right\}. \end{aligned}$$

Hence

$$\kappa_2(A+E) \leq \kappa_2(LL^T)\max\left\{1, \frac{\lambda_{\max}(A)}{\lambda_{\min}(LL^T)\delta}\right\}. \tag{4.23}$$

We can now assess how well objectives (O1)–(O3) are satisfied. To satisfied objective (O1) we would like E to be zero when $\lambda_{\min} \geq \delta$, and to satisfy (O2) we would like $\|E\|_2$ to be not much larger than $\delta - \lambda_{\min}$ when A is not positive definite. The sufficient condition (4.20) for E to be zero and inequality (4.22) show that these conditions do hold modulo factors $\lambda_{\max,\min}(LL^T)$. Inequality (4.23) bounds $\kappa_2(A+E)$ with the expected reciprocal dependence on δ, again with terms $\lambda_{\max,\min}(LL^T)$. The conclusion is that Algorithm MC is guaranteed to perform well if $\lambda_{\min}(LL^T)$ and $\lambda_{\max}(LL^T)$ are not too far from 1.

Note that, since L is unit lower triangular, $e_1^T(LL^T)e_1 = 1$, which implies that $\lambda_{\min}(LL^T) \leq 1$ and $\lambda_{\max}(LL^T) \geq 1$. For the BBK pivoting strategy we have $\max_{i,j} |l_{ij}| \leq 2.781$, so

$$1 \leq \lambda_{\max}(LL^T) \leq \mathrm{trace}(LL^T) = \|L\|_F^2 \leq n + \tfrac{1}{2}n(n-1)2.781^2 \leq 4n^2 - 3n. \tag{4.24}$$

Furthermore,

$$1 \leq \lambda_{\min}(LL^T)^{-1} \leq \|(LL^T)^{-1}\|_2 \leq \|L^{-1}\|_2^2 \leq (3.781)^{2n-2}, \tag{4.25}$$

using a bound from [55, Thm. 8.13 and Problem 8.5]. These upper bounds are approximately attainable, but in practice are rarely approached. In particular, the upper bound of (4.25) can be approached only in the unlikely event that most of the subdiagonal elements of L are negative and of near maximal magnitude. Note that each 2×2 pivot causes a subdiagonal element $l_{i+1,i}$ to be zero and so further reduces the likelihood of $\|L^{-1}\|_2$ being large.

The modified Cholesky algorithm in this section and the corresponding analysis are not tied exclusively to the BBK pivoting strategy. We could use instead the "fast Bunch–Parlett" pivoting strategy from [6], which appears to be more efficient than the BBK pivoting strategy when both are implemented in block form [6].

4.5 The Modified Aasen Algorithm

In this section, we assess the feasibility of using $\mathrm{LTL}^{\mathrm{T}}$ factorization with partial pivoting for computing a modified factorization. Our modified Aasen algorithm has a parameter $\delta \geq 0$ and it attempts to produce the perturbation (4.18).

Algorithm MA (Modified Aasen Factorization) *Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ and a parameter $\delta \geq 0$ this algorithm computes a permutation matrix P, a unit lower triangular matrix L, and a full matrix T such that*

$$P(A+E)P^T = LTL^T$$

and $A+E$ is symmetric positive definite (or symmetric positive semidefinite if $\delta = 0$). The algorithm attempts to ensure that if $\lambda_{\min} < \delta$ then $\lambda_{\min}(A+E) \approx \delta$.

1. Compute the factorization $PAP^T = L\widetilde{T}L^T$ using Aasen's method.
2. Let $T = \widetilde{T} + \Delta\widetilde{T}$, where $\Delta\widetilde{T}$ is the minimum Frobenius norm perturbation that achieves $\lambda_{\min}(\widetilde{T} + \Delta\widetilde{T}) \geq \delta$.

Since equalities (4.20)–(4.23) do not depend on the fact that $\widetilde{D}$ is block diagonal, it is obvious that the discussion in last section for the modified Cholesky factorization is also valid for the modified Aasen factorization. Moreover since $|l_{ij}| \leq 1$ and $L(:,1)$ is the first column of the identity matrix, we have

$$1 \leq \lambda_{\max}(LL^T) \leq \mathrm{trace}(LL^T) = \|L\|_F^2 \leq n + \tfrac{1}{2}(n-1)(n-2), \tag{4.26}$$

and

$$1 \leq \lambda_{\min}(LL^T)^{-1} \leq \|(LL^T)^{-1}\|_2 \leq \|L^{-1}\|_2^2 \leq 2^{2n-4}, \tag{4.27}$$

using a bound from [55, Thm. 8.13 and Problem 8.5]. The bounds are smaller than those in (4.24), (4.25) for the BBK pivoting strategy. We found that the factor L computed by the modified Aasen factorization was usually better conditioned than those computed by Algorithm MC when testing over random matrices.

Objectives (O1)–(O3) are duly satisfied. However objective (O4) is less trivial. The eigenproblem of a symmetric tridiagonal matrix is well studied. We give a survey of three direct methods implemented in LAPACK [2].

4.5.1 Solving Symmetric Tridiagonal Eigenproblem

Based on [26], [56], we summarize three direct methods implemented in LAPACK [2] for solving the symmetric tridiagonal eigenproblem, namely the divide and conquer algorithm, the symmetric QR algorithm and the bisection algorithm with inverse iteration.

The divide and conquer algorithm writes a symmetric tridiagonal T in the form

$$T = \begin{bmatrix} T_{11} & 0 \\ 0 & T_{22} \end{bmatrix} + \alpha v v^T,$$

where only the trailing diagonal element of T_{11} and the leading diagonal element of T_{22} differ from the corresponding elements of T. The eigensystems of T_{11} and T_{22} are found by applying the algorithm recursively, yielding $T_{11} = Q_1 \Lambda_1 Q_1^T$ and $T_{22} = Q_2 \Lambda_2 Q_2^T$. Then we have

$$\begin{aligned} T &= \begin{bmatrix} Q_1 \Lambda_1 Q_1^T & 0 \\ 0 & Q_2 \Lambda_2 Q_2^T \end{bmatrix} + \alpha v v^T \\ &= \operatorname{diag}(Q_1, Q_2)(\operatorname{diag}(\Lambda_1, \Lambda_2) + \alpha \widetilde{v}\widetilde{v}^T) \operatorname{diag}(Q_1, Q_2)^T, \end{aligned}$$

where $\widetilde{v} = \operatorname{diag}(Q_1, Q_2)^T v$. The eigensystem of a rank-one perturbed diagonal matrix $D + \rho z z^T$ can be found by solving the *secular equation* obtained by equating the characteristic polynomial to zero:

$$f(\lambda) = 1 + \rho \sum_{j=1}^{n} \frac{z_j^2}{d_{jj} - \lambda} = 0.$$

Hence by solving such an equation we can obtain the spectral decomposition

$$\operatorname{diag}(\Lambda_1, \Lambda_2) + \alpha \widetilde{v}\widetilde{v}^T = \widetilde{Q} \widetilde{\Lambda} \widetilde{Q}^T.$$

Finally, the spectral decomposition of T is given by

$$T = U\widetilde{\Lambda}U^T, \qquad U = \mathrm{diag}(Q_1, Q_2)\widetilde{Q}.$$

The formation of U is a matrix multiplication and dominates the operation count.

The divide and conquer algorithm was originally suggested by Cuppen [25], and how to solve the secular equation efficiently was shown by Bunch, Nielson and Sorensen [13], building on work of Golub [40]. Until recently, it was thought that extended precision arithmetic was needed in the solution of the secular equation to guarantee that sufficiently orthogonal eigenvectors are produced when there are close eigenvalues. However, Gu and Eisenstat [46] have found a new approach that does not require extended precision.

The divide and conquer algorithm has natural parallelism. Even on serial computers it can be many times faster than the symmetric QR algorithm, though it needs more workspace. This is currently the fastest method to find all the eigenvalues and eigenvectors for symmetric tridiagonal matrices of dimension larger than 25 [26]. In the worst case, the divide and conquer algorithm requires $O(n^3)$ flops. However, Demmel [26] found in his experiments over a large set of random test cases that, on average, $O(n^{2.3})$ flops were required.

Now we look at the symmetric QR algorithm, which finds all the eigenvalues and optionally all the eigenvectors, of a symmetric tridiagonal matrix. Given a symmetric tridiagonal matrix T, the symmetric QR algorithm compute a sequence $T^{(k)}$ of symmetric tridiagonal matrices converging to diagonal form using the QR factorization and a shift technique. At each stage, we choose a suitable real shift parameter μ_k, perform a QR factorization on $T^{(k)} - \mu_k I$, and obtain $T^{(k+1)}$ by multiplying the factors in reverse order. That is,

$$\begin{aligned} T^{(k)} - \mu_k I &=: Q^{(k)} R^{(k)}, \\ T^{(k+1)} &:= R^{(k)} Q^{(k)} + \mu_k I. \end{aligned}$$

It is easily shown that

$$T^{(k+1)} = Q^{(k)T} T^{(k)} Q^{(k)}.$$

This unitary congruence transformation preserves eigenvalues and symmetry of T and, most importantly, the tridiagonal form.

Let the diagonal entries of $T^{(k)}$ be $a_1^{(k)}, \ldots, a_n^{(k)}$ and the off-diagonal entries be $b_1^{(k)}, \ldots, b_{n-1}^{(k)}$. The shift μ_k is chosen to ensure that $b_{n-1}^{(k)} \to 0$ as $k \to \infty$. Wilkinson [91] proposed the so-called Wilkinson shift where μ_k is the eigenvalue of $\begin{bmatrix} a_{n-1}^{(k)} & b_{n-1}^{(k)} \\ b_{n-1}^{(k)} & a_n^{(k)} \end{bmatrix}$ that is closer to $a_n^{(k)}$, and is given by

$$\mu_k = a_n^{(k)} + d - \text{sign}(d)\sqrt{d^2 + b_{n-1}^{(k)2}}, \tag{4.28}$$

where $d = (a_{n-1}^{(k)} - a_n^{(k)})/2$ and

$$\text{sign}(d) = \begin{cases} +1, & \text{if } d > 0\ , \\ -1, & \text{if } d < 0\ , \\ +1, & \text{if } d = 0, \text{ and } a_n^{(k)} \geq 0, \\ -1, & \text{if } d = 0, \text{ and } a_n^{(k)} < 0. \end{cases}$$

Note that when $d = 0$, both eigenvalues have the same distance from $a_n^{(k)}$. In this case, we should choose sign(d) so that $|\mu_k|$ is minimized. Wilkinson [91] has shown that, for this choice of shift, the symmetric QR algorithm is globally, and at least quadratically, convergent, and is asymptotically cubically convergent for almost all matrices.

The symmetric QR algorithm is currently the fastest practical method to find all the eigenvalues of a symmetric tridiagonal matrix, taking $O(n^2)$ flops [26], [41]. However, for finding all the eigenvectors as well, the symmetric QR algorithm takes a little over $6n^3$ flops on average and is only the fastest algorithm for small matrices, up to about $n = 25$. This is the algorithm underlying the MATLAB command `eig`.

Eigenvalues	Eigenvectors	Size of T	Method of Choice
all	all	$n > 25$	Divide and conquer algorithm
all	all	$n \leq 25$	Symmetric QR algorithm
all	none	—	Symmetric QR algorithm
selected	selected	—	Bisection with inverse iteration

Table 4.2: Method of choice for symmetric tridiagonal matrix T.

When some but not all eigenvalues and eigenvectors of a symmetric tridiagonal matrix T are required, the bisection algorithm followed by inverse iteration is attractive. Recall that if the diagonal entries of T are $a_1, \ldots, a_n$ and the off-diagonal entries are $b_1, \ldots, b_{n-1}$ then we have the Sturm sequence recurrence

$$d_i = (a_i - \sigma)d_{i-1} - b_{i-1}^2 d_{i-2},$$

where d_i is the determinant of the leading $i \times i$ principal submatrix of $T - \sigma I$. The number of sign changes in the sequence of d_i's is the number of eigenvalues of T less than σ, denoted count(σ), and this fact is the basis for the application of the bisection method.

Although bisection is a simple and robust algorithm, it can give incorrect results if the function count(σ) is not a monotonic increasing function of σ. Demmel, Dhillon and Ren [27] give a thorough analysis of the correctness of the bisection algorithm for different implementations of the count function and under a variety of assumptions on the arithmetic. Note that the use of IEEE standard arithmetic will ensure the correctness of the bisection algorithm in this case which further confirms the importance of the standard.

Table 4.2 gives guidelines for choosing a suitable method under different circumstances. For Algorithm MA, since all eigenvalues and eigenvectors are needed, we should use either the divide and conquer algorithm or the symmetric QR algorithm, depending on the dimension of the matrix.

Another possibility is to compute an optimal 2-norm perturbation for the

symmetric tridiagonal matrix, for which only its minimum eigenvalue is required. That is, in the notation of Algorithm MA, $T := \widetilde{T} + \lambda_{\min}(\widetilde{T})I$. In this case, the bisection method is the method of choice.

4.6 Comparison of Algorithms

Here we compare the GMW and SE algorithms, Algorithm MC and Algorithm MA according to their theoretical aspects.

The bounds (4.10) and (4.16) can be compared with (4.22) for Algorithm MC and Algorithm MA. The bound (4.22) has the advantage of directly comparing the perturbation made by Algorithm MC and Algorithm MA with the optimal one, defined as in (4.17) and evaluated in Theorem 4.4.1, and it is potentially a much smaller bound than (4.10) and (4.16) if $|\lambda_{\min}(A)| \ll |\lambda_{\max}(A)|$ and $\kappa_2(LL^T)$ is not too large.

All four algorithms satisfy objective (O1) of not modifying a sufficiently positive definite matrix, though for the GMW and SE algorithms no condition analogous to (4.20) that quantifies "sufficiently" in terms of $\lambda_{\min}(A)$ is available. Bounds for $\kappa_2(A + E)$ that are exponential in n hold for the GMW and SE algorithms [78]. The same is true for Algorithm MC and MA; see (4.23)–(4.27).

To summarize, in terms of the objectives of Section 4.1 for a modified Cholesky algorithm, Algorithm MC and Algorithm MA are theoretically competitive with the GMW and SE algorithms, with the weakness that if $\kappa_2(LL^T)$ is large then the bound on $\|E\|_2$ is weak.

When applied to an indefinite matrix, the GMW and SE algorithms provide information that enables a direction of negative curvature of the matrix to be produced; these directions are required in certain algorithms for unconstrained optimization in order to move away from non-minimizing stationary points. For an indefinite matrix, Algorithm MC provides immediate access to a direction of

negative curvature from the LDL^T factorization computed in step 1 of Algorithm MC. Because $\kappa(L)$ is bounded, this direction satisfies conditions required for convergence theory [67]. For Algorithm MA, we can use the bisection with inverse iteration to compute the most negative eigenvalue and its corresponding eigenvector of the symmetric tridiagonal matrix to gain information about a direction of negative curvature.

Finally, we consider the behaviour of the algorithms in the presence of rounding error. Algorithm MC is backward stable because the underlying factorization is [58]: barring large element growth in block LDL^T factorization with the BBK pivoting strategy, the algorithm produces LDL^T factors not of $P(A+E)P^T$, but of $P(A+E+F)P^T$, where $\|F\|_2 \leq c_n u \|A+E\|_2$ with c_n a constant. We can deduce the stability of Algorithm MA using the same reasoning. Although no comments on numerical stability are given in [37] and [78], a simple argument shows that the GMW and SE algorithms are backward stable. Apply either algorithm to A, obtaining the Cholesky factorization $P(A+E)P^T = R^T R$. Now apply the same algorithm to $P(A+E)P^T$: it will not need to modify $P(A+E)P^T$, so it will return the same computed R factor. But since no modification was required, the algorithm must have carried out a standard Cholesky factorization. Since Cholesky factorization is a backward stable process, the modified Cholesky algorithm must itself be backward stable.

4.7 Numerical Experiments

We have experimented with MATLAB implementations of Algorithm MC, Algorithm MA, and the GMW and SE algorithms. The M-file for the GMW algorithm was provided by Margaret Wright and sets the tolerance $\delta = 2u \max\{\alpha + \beta, 1\}$, where $2u$ is the value of MATLAB's variable `eps`. The M-file for the SE algorithm was provided by Elizabeth Eskow and sets the tolerance $\tau = (2u)^{1/3}$. In

Algorithm MC and Algorithm MA, we set $\delta = \sqrt{u}\|A\|_\infty$.

The aims of the experiments are as follows: to see how well the Frobenius norm of the perturbation E produced by Algorithm MC and Algorithm MA approximates the distance $\mu_F(A,\delta)$ defined in (4.17), and to compare the norms of the perturbations E and the condition numbers of $A+E$ produced by the three algorithms. We measure the perturbation E by the ratios

$$\gamma_F = \frac{\|E\|_F}{\mu_F(A,\delta)}, \qquad \gamma_2 = \frac{\|E\|_2}{|\lambda_{\min}(A)|},$$

which differ only in their normalization and the choice of norm. Both Algorithm MC and Algorithm MA attempt to make γ_F close to 1. The quantity γ_2 is used by Schnabel and Eskow [78] to compare the performance of the GMW and SE algorithms; since E is diagonal for these algorithms, γ_2 compares the amount added to the diagonal with the minimum diagonal perturbation that makes the perturbed matrix positive semidefinite.

First, we note that the experiments of Schnabel and Eskow show that the SE algorithm can produce a substantially smaller value of γ_2 than the GMW algorithm. Schnabel and Eskow also identified a 4×4 matrix for which the GMW algorithm significantly outperforms the SE algorithm:

$$A = \begin{bmatrix} 1890.3 & -1705.6 & -315.8 & 3000.3 \\ & 1538.3 & 284.9 & -2706.6 \\ & & 52.5 & -501.2 \\ & & & 4760.8 \end{bmatrix}, \tag{4.29}$$

with $\lambda(A) = \{-0.39, -0.34, -0.25, 8.2 \times 10^3\}$. We give our results in Table 4.3; they show that Algorithm MC and Algorithm MA can also significantly outperform the SE algorithm.

We ran a set of tests similar to those of Schnabel and Eskow [78]. The matrices A are of the form $A = Q\Lambda Q^T$, where $\Lambda = \mathrm{diag}(\lambda_i)$ with the eigenvalues λ_i from one

	MC	MA	GMW	SE
γ_F	1.3	1.1	2.7	3.7×10^3
γ_2	1.7	1.1	2.7	2.8×10^3

Table 4.3: Measures of E for the 4×4 matrix (4.29).

n:	25	50	100
max	523	2188	8811
mean	343.9	1432.8	5998.4

Table 4.4: Number of comparisons for the BBK pivoting strategy.

of three random uniform distributions: $[-1, 10^4]$, $[-1, 1]$ and $[-10^4, -1]$. For the first range, one eigenvalue is generated for the range $[-1, 0)$ to ensure that A has at least one negative eigenvalue. The matrix Q is a random orthogonal matrix from the Haar distribution, generated using the routine `qmult` from the Test Matrix Toolbox [54], which implements an algorithm of Stewart [83]. For each eigenvalue distribution we generated 30 different matrices, each corresponding to a fresh example of Λ and of Q. We took $n = 25, 50, 100$. The ratios γ_F and γ_2 are plotted in Figures 4.1–4.3. Figures 4.4–4.6 plot the condition numbers $\kappa_2(A+E)$ for $n = 25, 50, 100$; Table 4.4 reports the number of comparisons used by the BBK pivoting strategy on these matrices for each n; the maximum number of comparisons is less than n^2 in each case.

In Figure 4.7 we report results for three nonrandom matrices from the Test Matrix Toolbox. `Clement` is a tridiagonal matrix with eigenvalues plus and minus the numbers $n-1$, $n-3$, $n-5, \ldots,$ (1 or 0). `Dingdong` is the symmetric $n \times n$ Hankel matrix with (i, j) element $0.5/(n-i-j+1.5)$, whose eigenvalues cluster around $\pi/2$ and $-\pi/2$. `Ipjfact` is the Hankel matrix with (i, j) element $1/(i+j)!$.

Our conclusions from the experiments are as follows.

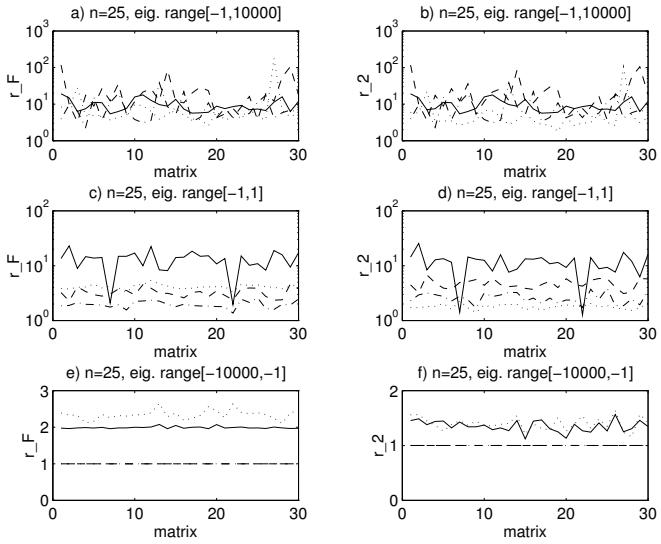

Figure 4.1: Measures of E for 30 random indefinite matrices with $n = 25$. Key: GMW —, SE $\cdots$, MA -.-., MC - - -.

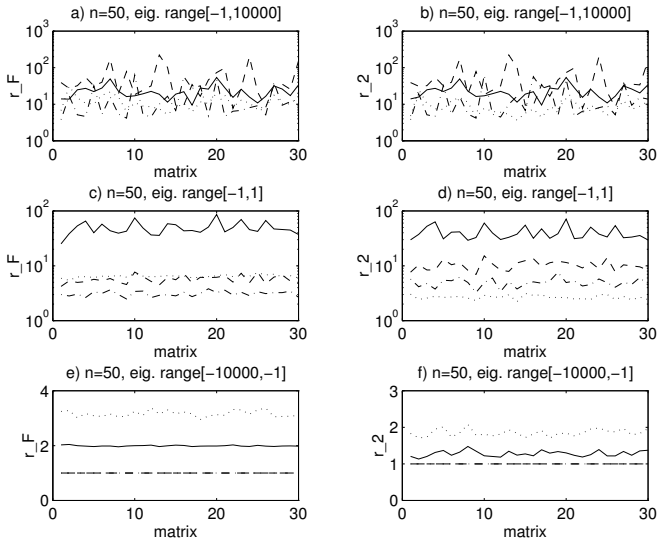

Figure 4.2: Measures of E for 30 random indefinite matrices with $n = 50$. Key: GMW —, SE $\cdots$, MA -.-., MC - - -.

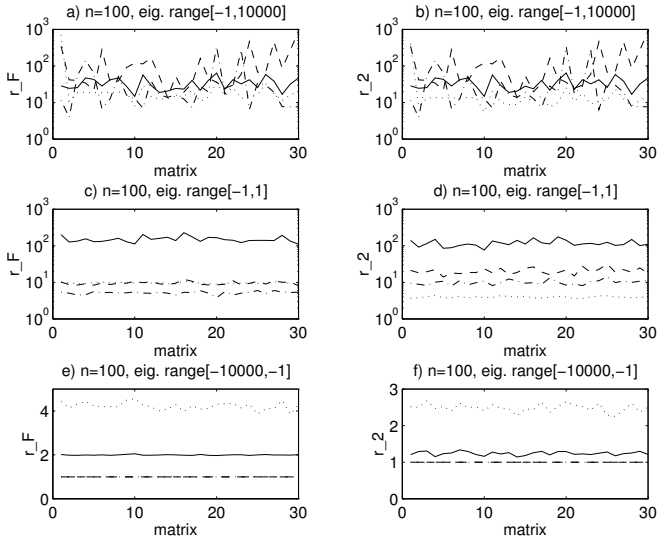

Figure 4.3: Measures of E for 30 random indefinite matrices with $n = 100$. Key: GMW —, SE $\cdots$, MA -.-., MC - - -.

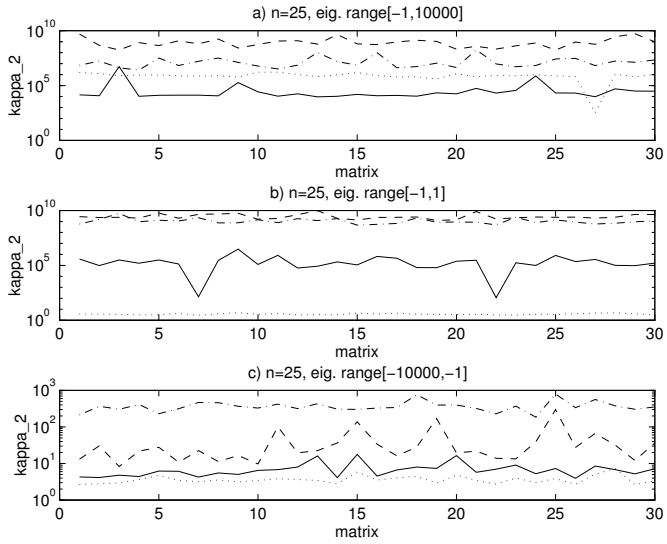

Figure 4.4: Condition numbers $\kappa_2(A+E)$ for 30 random indefinite matrices with $n = 25$. Key: GMW —, SE $\cdots$, MA -.-., MC - - -.

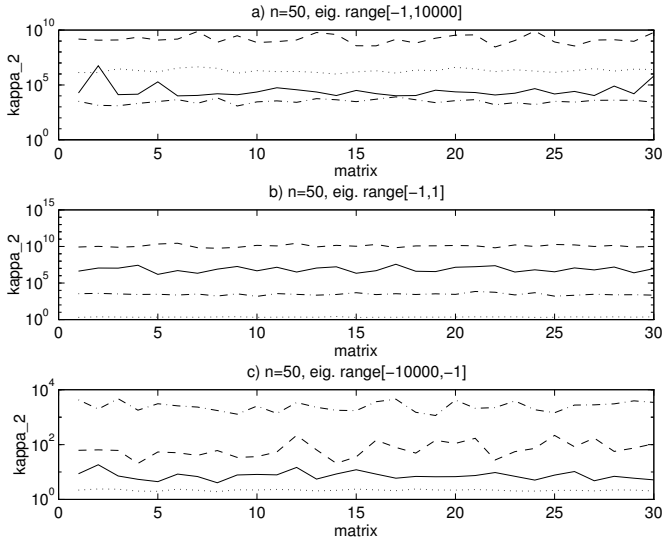

Figure 4.5: Condition numbers $\kappa_2(A+E)$ for 30 random indefinite matrices with $n = 50$. Key: GMW —, SE $\cdots$, MA -.-., MC - - -.

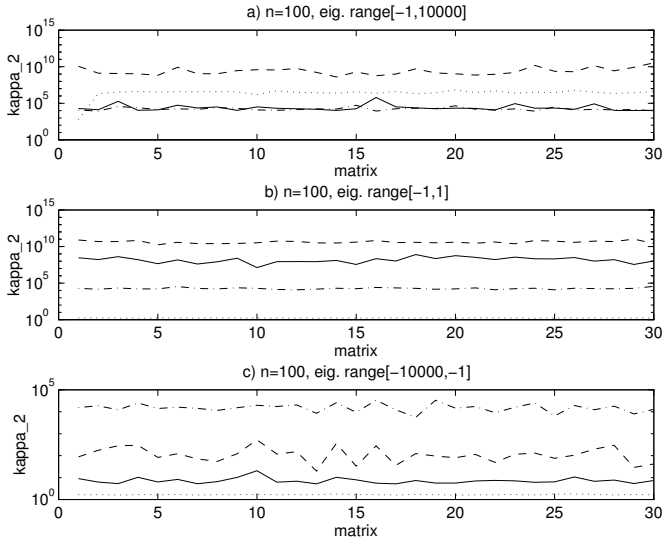

Figure 4.6: Condition numbers $\kappa_2(A + E)$ for 30 random indefinite matrices with $n = 100$. Key: GMW —, SE $\cdots$, MA -.-., MC - - -.

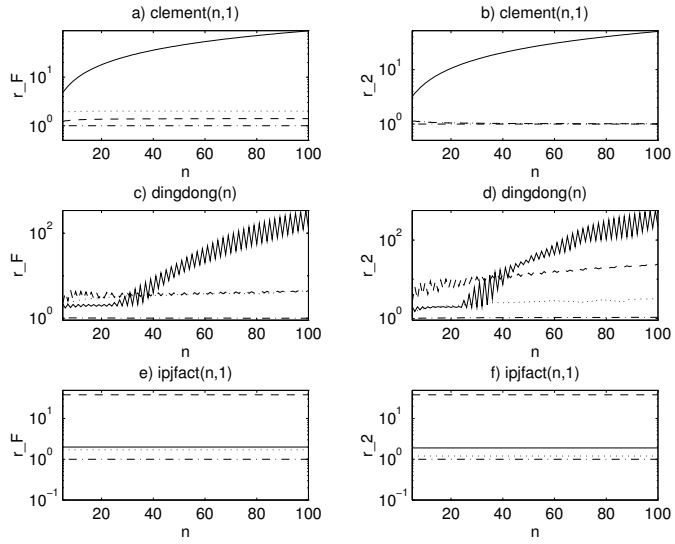

Figure 4.7: Measures for three nonrandom matrices. Key: GMW —, SE $\cdots$, MA -.-., MC - - -.

1. None of the four algorithms is uniformly better that the others in terms of producing a small perturbation E, whichever measure γ_F or γ_2 is used. All four algorithms can produce values of γ_F and γ_2 significantly greater than 1, depending on the problem.

2. Algorithm MC produced γ_F of order 10^3 for the eigenvalue distribution $[-1, 10^4]$ for each n, and the values of $\kappa_2(LL^T)$ were approximately $100\gamma_F$ in each such case. However, often γ_F was of order 1 when $\kappa_2(LL^T)$ was of order 10^2 or 10^3, so a large value of $\kappa_2(LL^T)$ is only a necessary condition, not a sufficient one, for poor performance of Algorithm MC; in other words, the bounds of Section 4.4 can be weak. Similar observations hold for Algorithm MA.

3. The condition numbers $\kappa_2(A+E)$ vary greatly among the algorithms. Our experience is that for $\delta = \sqrt{u}\|A\|_\infty$ Algorithm MC fairly consistently produces conditions of order $100/\sqrt{u}$; the condition number is, as predicted by

(4.23), much smaller for the random matrices with eigenvalues on the range $[-10^4, -1]$, because the algorithm attempts to perturb all the eigenvalues to δ and (4.23) is reduced to $\kappa_2(A+E) \leq \kappa_2(LL^T)$. Again, similar conclusions hold for Algorithm MA. The condition numbers produced by the GMW and SE algorithms vary greatly with the type of matrix.

The fact that γ_F is close to 1 for the random matrices with eigenvalues in the range $[-10^4, -1]$ for Algorithm MC and Algorithm MA is easily explained. Let A be negative definite. Then Algorithm MC computes $P(A+E)P^T = L(\delta I)L^T$. Hence

$$\begin{aligned}
\gamma_F &= \frac{\|E\|_F}{(\sum_{\lambda_i \leq \delta}(\lambda_i - \delta)^2)^{1/2}} \\
&\leq \frac{\|E\|_F}{\|A\|_F} = \frac{\|A - \delta \cdot P^T L L^T P\|_F}{\|A\|_F} \\
&\leq \frac{\|A\|_F + \delta \|LL^T\|_F}{\|A\|_F} \\
&\leq 1 + \frac{(4n^2 - 3n)\delta}{\|A\|_F},
\end{aligned}$$

using (4.24). Meanwhile, Algorithm MA computes $P(A+E)P^T = LQ(\delta I)Q^T L^T$, where $QQ^T = I$. We have

$$\begin{aligned}
\gamma_F &= \frac{\|E\|_F}{(\sum_{\lambda_i \leq \delta}(\lambda_i - \delta)^2)^{1/2}} \\
&\leq \frac{\|E\|_F}{\|A\|_F} = \frac{\|A - \delta \cdot P^T L Q Q^T L^T P\|_F}{\|A\|_F} \\
&\leq \frac{\|A\|_F + \delta \|LL^T\|_F}{\|A\|_F} \\
&\leq 1 + \frac{(n^2 - n + 2)\delta}{2\|A\|_F},
\end{aligned}$$

using (4.26). So γ_F can exceed 1 only by a tiny amount for Algorithm MC and Algorithm MA applied to a negative definite matrix, irrespective of $\kappa_2(LL^T)$. It is an open question to explain why γ_F are approximately 2 and 4 for the GMW and SE algorithms respectively for this class of matrices.

4.8 Concluding Remarks

Algorithm MC, based on a block LDL^T factorization with the bounded Bunch–Kaufman pivoting strategy, and Algorithm MA, based on Aasen's LTL^T factorization with partial pivoting, merit consideration as alternatives to the algorithms of Gill, Murray and Wright, and Schnabel and Eskow. The results in Section 4.7 suggest that the new algorithms are competitive with the GMW and SE algorithms in terms of the objectives (O1)–(O4) listed in Section 4.1. Algorithms MC and MA have the advantages that the extent to which they satisfy the objectives is neatly, although not sharply, described by the bounds of Sections 4.4 and 4.5.

Algorithms MC and MA can be implemented by augmenting existing software with just a small amount of additional code. In particular, for Algorithm MC, we can use the efficient implementations for both dense and sparse matrices written by Ashcraft, Grimes and Lewis [6], which make extensive use of level 2 and 3 BLAS for efficiency on high-performance machines. For Algorithm MA, the LTL^T factorization is implemented in IMSL Fortran 90 MP Library [48], [90].

Since all four algorithms can "fail", that is, they can produce unacceptably large perturbations, it is natural to ask how failure can be detected and what should be done about it. The GMW and SE algorithms produce their (diagonal) perturbations explicitly, so it is trivial to evaluate their norms. For Algorithm MC and Algorithm MA, the perturbations to A have the form $E = P^T L(D+F)L^T P - A$, which would require $O(n^3)$ operations to form explicitly. However, we can *estimate* $\|E\|_\infty$ using the norm estimator from [52] (which is implemented in LAPACK). The estimator require the formation of products Ex for certain vectors x, and these can be computed in $O(n^2)$ operations; the estimate produced is a lower bound that is nearly always within a factor of 3 of the true norm. For all four algorithms, then, we can inexpensively test whether the perturbation produced is acceptably small. Unfortunately, for none of the algorithms

is there an obvious way to improve a modified factorization that makes too big a perturbation; whether improvement is possible, preferably cheaply, is an open question. Of course one can always resort to computing an optimal perturbation by computing the eigensystem of A and using the formulae in Theorem 4.4.1.

We note that we have approached the problem of modified Cholesky factorization from a purely linear algebra perspective. An important test of a modified Cholesky algorithm is to evaluate it in an optimization code on representative problems, as was done by Schlick [77] for the GMW and SE algorithms. Such testing is beyond the scope of this thesis but would produce valuable information about the practical performance of the different algorithms.

Chapter 5

Modifying the Inertia of Matrices Arising in Optimization

5.1 Introduction

A block 2×2 partitioning

$$C = \begin{bmatrix} H & A \\ A^T & -M \end{bmatrix}$$

of a symmetric matrix C arises in a number of applications, including constrained optimization, least squares problems and Navier–Stokes problems, as explained in the next section. The matrix M is positive semidefinite, but H can be indefinite, depending on the application. In constrained optimization, a "second order sufficiency" condition leads to the problem of perturbing H so that C has a particular inertia. It is this problem that motivated our work. We can view this chapter as an attempt to extend the notion of modified Cholesky factorization to constrained optimization.

We present some background material on congruence transformations in Section 5.3, including an extension of Ostrowski's theorem (Theorem 4.4.2) to transformations with a rectangular matrix. In Section 5.4 we derive some useful inertia properties of the matrix C. How to make a minimal norm (full) perturbation to increase the number of nonnegative eigenvalues of a symmetric matrix by a given amount is shown in Section 5.5. The main result of this chapter is in Section 5.6, in which we derive, for any unitarily invariant norm, a perturbation of H (only) of minimal norm that increases the number of nonnegative eigenvalues of C by a given amount. For optimization applications, another way of writing the

second order sufficiency condition is based on projecting H into the null space of A. We use this approach in Section 5.7 to derive another expression for a minimal norm perturbation to H that achieves the sufficiency condition. In Section 5.8 we consider how to implement our results in the optimization application and show that directions of negative curvature are produced as a by-product of the computation. Numerical experiment results are reported in Section 5.9 and concluding remarks are given in Section 5.10.

5.2 A Symmetric Block 2×2 Matrix and its Applications

Any symmetric matrix C can be written in the form

$$C = \begin{array}{c} \\ n \\ m \end{array} \begin{array}{c} \begin{array}{cc} n & m \end{array} \\ \begin{bmatrix} H & A^T \\ A & -M \end{bmatrix} \end{array},$$

where $H \in \mathbb{R}^{n \times n}$ and $M \in \mathbb{R}^{m \times m}$ are symmetric and $A \in \mathbb{R}^{n \times m}$. The reason for using a block 2×2 partitioning and for placing a minus sign in front of the $(2,2)$ block is that C then conveniently represents some particular cases arising in applications, which we now describe in roughly decreasing order of generality.

1. When M is diagonal and positive definite, C is the "primal-dual" matrix arising in certain interior methods for the general nonlinear programming problem [32], [33]. Here, H is the Hessian of the Lagrangian function and A^T is the Jacobian of the constraint functions. The matrix C also arises in penalty function methods for nonlinear programming, with M a positive multiple of the identity matrix [43]. In these applications both $m \leq n$ and $m \geq n$ are possible.

2. When $M = 0$, C is the Karush–Kuhn–Tucker (KKT) matrix, which arises when Newton's method or a quasi-Newton method is applied to the problem

$$\min_x F(x) \text{ subject to } A^T x = b, \tag{5.1}$$

where $F : \mathbb{R}^n \to \mathbb{R}$ and $m \leq n$, [20, p. 123], [34], [44]. To be precise, Newton's method leads to the equations

$$\begin{bmatrix} H_k & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_k \end{bmatrix} = \begin{bmatrix} -g_k \\ 0 \end{bmatrix},$$

where H is the Hessian of F or an approximation to it, g is the gradient of F, p is a search direction, and λ is a Lagrange multiplier, and where a subscript k denotes evaluation at the kth iterate.

3. For $M = 0$ and $H = \text{diag}(I_p, -I_q)$ where $p + q = n$, C is the augmented system matrix arising in the indefinite least squares problem

$$\min_x (b - Ax)^T H (b - Ax),$$

where $m \geq n$ [16]. This problem reduces to the standard least squares problem when $q = 0$.

4. If H and M are positive definite, then C matches precisely the definition of a symmetric quasidefinite matrix [89]. Such matrices arise in interior methods for linear and quadratic programming and much is known about the existence and stability of their LDL^{T} factorizations [38], [89].

5. Matrices with H positive definite and $M = 0$ arise in discretized incompressible Navier–Stokes equations [79], and their spectral properties are important in the development of preconditioned iterative methods [30].

6. The matrix with $H = \delta I$ and $M = \delta I$ ($\delta > 0$) appears in the augmented system corresponding to the damped least squares problem

$$\min_x \|b - Ax\|_2^2 + \delta^2 \|x\|_2^2;$$

see Saunders [76].

7. For H positive definite and $M = 0$, C is the augmented system matrix arising in the generalized least squares problem $\min_x (b - Ax)^T H^{-1}(b - Ax)$ $(m \geq n)$ [8, Section 4.3.2]; $H = I$ gives the standard least squares problem.

In quasi-Newton methods for the linear-equality constrained problem (5.1) it is desirable that the Hessian approximation H satisfy the "second order sufficiency" condition [44]

$$p^T H p > 0 \text{ for all nonzero } p \text{ such that } A^T p = 0. \tag{5.2}$$

One equivalent condition is that the projected Hessian $Z^T H Z$ be positive definite, where the columns of Z form a basis for the null space null(A^T). Less obviously, the condition (5.2) is also equivalent to requiring the so-called KKT matrix

$$K = \begin{array}{c} \\ n \\ m \end{array} \begin{array}{c} \begin{array}{cc} n & m \end{array} \\ \begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \end{array}, \tag{5.3}$$

to have a certain inertia, as shown by Gould [42]. Recall that the inertia of a symmetric matrix is an ordered triple (i_+, i_-, i_0), where i_+ is the number of positive eigenvalues, i_- is the number of negative eigenvalues, and i_0 is the number of zero eigenvalues. We write

$$\text{inertia}(A) = (i_+(A), i_-(A), i_0(A)).$$

Theorem 5.2.1 (Gould) *Let A be of full rank m. The condition* (5.2) *holds if and only if K has the inertia $(n, m, 0)$.*

Proof: Let A have the QR factorization

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = \begin{bmatrix} Y & Z \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix},$$

where $Y \in \mathbb{R}^{n\times m}$, $Z \in \mathbb{R}^{n\times(n-m)}$ and $R \in \mathbb{R}^{m\times m}$. Then

$$K = \begin{bmatrix} H & Q\begin{bmatrix} R \\ 0 \end{bmatrix} \\ \begin{bmatrix} R^T & 0 \end{bmatrix} Q^T & 0 \end{bmatrix} = \begin{bmatrix} Q & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} Q^T H Q & \begin{bmatrix} R \\ 0 \end{bmatrix} \\ \begin{bmatrix} R^T & 0 \end{bmatrix} & 0 \end{bmatrix} \begin{bmatrix} Q & 0 \\ 0 & I \end{bmatrix}^T$$

$$\sim \begin{bmatrix} Y^T H Y & Y^T H Z & R \\ Z^T H Y & Z^T H Z & 0 \\ R^T & 0 & 0 \end{bmatrix} =: \widetilde{K},$$

where $\sim$ denotes congruence (in fact, this first transformation is an orthogonal similarity which preserves the symmetry and the eigenvalues). Now define the nonsingular matrix

$$W = \begin{bmatrix} I_m & 0 & -\frac{1}{2} Y^T H Y R^{-T} \\ 0 & I_{n-m} & -Z^T H Y R^{-T} \\ 0 & 0 & R^{-T} \end{bmatrix}.$$

It is straightforward to verify that

$$W \widetilde{K} W^T = \begin{bmatrix} 0 & 0 & I_m \\ 0 & Z^T H Z & 0 \\ I_m & 0 & 0 \end{bmatrix}.$$

By constructing eigenvectors $[e_i \;\; 0 \;\; e_i]^T$ and $[e_i \;\; 0 \;\; -e_i]^T$ where e_i is the ith row of the identity matrix I_m, it is easily seen that the eigenvalues of $W\widetilde{K}W^T$ are 1 and -1, each repeated m times, together with the $n-m$ eigenvalues of $Z^T H Z$. Since Z spans the null space of A^T, $Z^T H Z$ is positive definite if and only if (5.2) holds, which completes the proof. $\square$

The requirement (5.2) and Theorem 5.2.1 the problem give rise to the problem of perturbing H so that K achieves the desired inertia $(n, m, 0)$ [42]. The matrix A must not be perturbed because this would correspond to changing the constraints in (5.1). The same problem is relevant for the primal-dual matrix with M diagonal

and positive semidefinite [32]. We find a minimal-norm solution to a more general version of this inertia perturbation problem in Section 5.6. In Section 5.7 we consider an alternative approach to perturbing H to satisfy (5.2), based on the projected Hessian. First, we develop some necessary background theory.

5.3 Rectangular Congruence Transformations

Sylvester's inertia theorem says that the inertia of a symmetric matrix is preserved under a congruence transformation. Ostrowski's theorem (Theorem 4.4.2) [60, Thm. 4.5.9], [69], [92] goes further by explaining by how much the magnitudes of the eigenvalues can change. In the following statement of Ostrowski's theorem [60, Cor. 4.5.11] the transforming matrix X is permitted to be singular, in which case the transformation matrix X^TAX is not a congruence transformation and can change the inertia. Throughout this chapter the eigenvalues of a symmetric $n \times n$ matrix are ordered $\lambda_1 \leq \cdots \leq \lambda_n$, and $\lambda_i(A)$ denotes the ith smallest eigenvalue of A.

Theorem 5.3.1 (Ostrowski) *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and $X \in \mathbb{R}^{n\times n}$. Then for each k, $k = 1{:}n$,*

$$\lambda_k(X^TAX) = \theta_k\lambda_k(A)$$

where $\lambda_1(X^TX) \leq \theta_k \leq \lambda_n(X^TX)$. □

We now generalize Ostrowski's theorem to "rectangular congruences", in which the transformating matrix X is nonsquare. Such transformations change the dimension and hence the inertia, but for full rank X the amount by which inertia can change depends on the difference of the dimensions of X, as shown in the corollaries below. First, we consider matrices X with at least as many rows as columns.

Theorem 5.3.2 *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and let $X \in \mathbb{R}^{n\times m}$ $(n \geq m)$. Then*

$$\lambda_k(X^T AX) = \theta_k \mu_k, \quad k = 1{:}m,$$

where

$$\lambda_k(A) \leq \mu_k \leq \lambda_{k+n-m}(A), \quad k = 1{:}m,$$

and $\lambda_1(X^T X) \leq \theta_k \leq \lambda_m(X^T X)$.

Proof: Let

$$X = U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T$$

be a singular value decomposition, where $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{m\times m}$ are orthogonal and $\Sigma \in \mathbb{R}^{m\times m}$ is diagonal. Then

$$X^T AX = V[\Sigma^T \ \ 0]U^T AU \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T = V(\Sigma^T \widetilde{A}_{11} \Sigma)V^T,$$

where $\widetilde{A}_{11}$ is the leading principal submatrix of order m of $\widetilde{A} = U^T AU$. By Ostrowski's theorem,

$$\lambda_k(X^T AX) = \lambda_k(\Sigma^T \widetilde{A}_{11} \Sigma) = \lambda_k(\widetilde{A}_{11})\theta_k,$$

where

$$\lambda_1(X^T X) = \lambda_1(\Sigma^T \Sigma) \leq \theta_k \leq \lambda_m(\Sigma^T \Sigma) = \lambda_m(X^T X).$$

Cauchy's interlace theorem [71, p. 186] shows that

$$\lambda_k(A) = \lambda_k(\widetilde{A}) \leq \lambda_k(\widetilde{A}_{11}) \leq \lambda_{k+n-m}(\widetilde{A}) = \lambda_{k+n-m}(A), \quad k = 1{:}m,$$

which yields the result. □

In the case where X has orthonormal columns (so that $\theta_k \equiv 1$), Theorem 5.3.2 reduces to the Poincaré separation theorem [60, Cor. 4.3.16], [84, Cor. 4.4, p. 198].

Corollary 5.3.3 *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and let $X \in \mathbb{R}^{n\times m}$ $(n \geq m)$ be of full rank. Then*

$$\text{inertia}(A)-(n-m,n-m,n-m) \leq \text{inertia}(X^TAX) \leq \text{inertia}(A)+(0,0,n-m). \quad \square$$

The next result covers the case $n \leq m$.

Theorem 5.3.4 *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and let $X \in \mathbb{R}^{n\times m}$ $(n \leq m)$. Then X^TAX has $m-n$ zero eigenvalues, which we number $\lambda_1, \ldots, \lambda_{m-n}$; the remaining eigenvalues satisfy*

$$\lambda_{m-n+k}(X^TAX) = \theta_k\lambda_k(A), \quad k = 1{:}n,$$

where $\lambda_{m-n+1}(X^TX) \leq \theta_k \leq \lambda_m(X^TX)$.

Proof: Let

$$X^T = U\begin{bmatrix}\Sigma\\ 0\end{bmatrix}V^T$$

be a singular value decomposition, where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal and $\Sigma \in \mathbb{R}^{n\times n}$ is diagonal. Then

$$X^TAX = U[\Sigma^T \;\; 0]V^TAV\begin{bmatrix}\Sigma\\ 0\end{bmatrix}U^T = U\begin{bmatrix}\Sigma^TV^TAV\Sigma & 0\\ 0 & 0\end{bmatrix}U^T =: U\widetilde{A}U^T,$$

which shows X^TAX and $\widetilde{A}$ have the same eigenvalues under the orthogonal similarity transformation. It is easily seen that $\widetilde{A}$ has $m-n$ zero eigenvalues. Now apply Ostrowski's theorem on the $(1,1)$ block of $\widetilde{A}$, we have for $k = 1{:}n$,

$$\lambda_{m-n+k}(\Sigma^TV^TAV\Sigma) = \theta_k\lambda_k(A),$$

where $\lambda_{m-n+1}(X^TX) = \lambda_1(\Sigma^T\Sigma) \leq \theta_k \leq \lambda_n(\Sigma^T\Sigma) = \lambda_m(X^TX)$. This completes the proof. $\square$

A direct consequence is as follows.

Corollary 5.3.5 *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and let $X \in \mathbb{R}^{n\times m}$ $(n \le m)$ be of full rank. Then*

$$\text{inertia}(X^T A X) = \text{inertia}(A) + (0, 0, m-n). \quad \square$$

5.4 Inertia Properties of C

In this section we derive some inertia properties of the matrix

$$C = \begin{array}{c} \\ n \\ m \end{array} \begin{array}{c} \begin{array}{cc} n & m \end{array} \\ \begin{bmatrix} H & A^T \\ A & -M \end{bmatrix} \end{array}. \tag{5.4}$$

Assume that H is nonsingular. We have

$$\begin{bmatrix} I & 0 \\ -A^T H^{-1} & I \end{bmatrix} \begin{bmatrix} H & A \\ A^T & -M \end{bmatrix} = \begin{bmatrix} H & A \\ 0 & -M - A^T H^{-1} A \end{bmatrix}$$
$$= \begin{bmatrix} H & 0 \\ 0 & -M - A^T H^{-1} A \end{bmatrix} \begin{bmatrix} I & H^{-1}A \\ 0 & I \end{bmatrix},$$

which shows that

$$C \sim \begin{bmatrix} H & 0 \\ 0 & -M - A^T H^{-1} A \end{bmatrix}. \tag{5.5}$$

This congruence is the basis of the following lemmas, the first of which is contained in [49, Thm. 3].

Lemma 5.4.1 *If H is nonsingular, $M = 0$ and A has full rank then* $\text{inertia}(C) \ge (m, m, 0)$ *if $n \ge m$ and* $\text{inertia}(C) = (n, n, m-n)$ *if $n \le m$.*

Proof: Let $\text{inertia}(H) = (a, b, 0)$ and $\text{inertia}(-A^T H^{-1} A) = (p, q, r)$. Then from (5.5) we have

$$\text{inertia}(C) = (a + p, b + q, r).$$

First, suppose $n \geq m$. By Corollary 5.3.3 we have $p \geq b - (n - m)$, so that $a + p \geq a + b - (n - m) = m$. Similarly, $b + q \geq m$. If $n \leq m$ then Corollary 5.3.5 shows that $p = b$, $q = a$ and $r = m - n$, and the result follows. □

Lemma 5.4.2 *If H is positive definite and M is positive semidefinite then*

$$\text{inertia}(C) = (n, m - p, p),$$

where $0 \leq p \leq m$. If A has full rank or M is positive definite then $p = 0$.

Proof: The result is a direct consequence of (5.5). □

The next lemma shows the somewhat surprising property that the inertia of C is independent of H when all the blocks are square, $M = 0$ and A is nonsingular. This result is given by Haynsworth and Ostrowski [49], who attribute it to Carlson and Schneider [15].

Lemma 5.4.3 *Let $m = n$ and $M = 0$. Then C is nonsingular if and only if A is nonsingular, and in this case* $\text{inertia}(C) = (n, n, 0)$.

Proof: The nonsingularity condition follows from

$$\det(C) = (-1)^n \det\left(\begin{bmatrix} A & H \\ 0 & A^T \end{bmatrix}\right) = (-1)^n \det(A)^2.$$

The inertia is obtained as a special case of Theorem 5.2.1, since (5.2) is trivially satisfied. □

There does not seem to be any useful characterization of the eigenvalues of C. The most general matrix for which the eigenvalues are known explicitly is the matrix

$$B(\alpha, \beta) = \begin{bmatrix} \alpha I_n & A \\ A^T & -\beta I_m \end{bmatrix}, \quad A \in \mathbb{R}^{n \times m}. \tag{5.6}$$

Saunders [76] shows that if A has rank p with nonzero singular values σ_i, $i = 1{:}p$, then

$$\lambda(B(\alpha,\beta)) = \begin{cases} \frac{1}{2}(\alpha-\beta) \pm \left(\sigma_i^2 + \frac{1}{4}(\alpha+\beta)^2\right)^{1/2}, & i = 1{:}p, \\ \alpha, & n-p \text{ times}, \\ -\beta, & m-p \text{ times}. \end{cases} \tag{5.7}$$

The conclusions of Lemmas 5.4.1–5.4.3 are readily verified for this matrix.

Finally, we give inequalities that bound the eigenvalues of C away from zero, which is of interest for investigating conditioning. This lemma is a restatement of the "separation theorem" of v. Kempen [88].

Lemma 5.4.4 *If H is positive definite and M is positive semidefinite or positive definite, then the eigenvalues λ_i of C satisfy*

$$\lambda_1 \le \cdots \le \lambda_m \le -\lambda_{\min}(M) < \lambda_{\min}(H) \le \lambda_{m+1} \le \cdots \le \lambda_{m+n}, \tag{5.8}$$

Proof: Let λ be an eigenvalue of C and x a corresponding eigenvector and write $Cx = \lambda x$ as

$$\begin{bmatrix} H & A \\ A^T & -M \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \lambda \begin{bmatrix} y \\ z \end{bmatrix}.$$

Premultiplying the first equation of the pair by y^T and the second by z^T, and subtracting, yields

$$y^T H y - \lambda y^T y = -z^T M z - \lambda z^T z,$$

or

$$y^T(H - \lambda I)y + z^T(M + \lambda I)z = 0. \tag{5.9}$$

If $-\lambda_{\min}(M) < \lambda < \lambda_{\min}(H)$ then $H - \lambda I$ and $M + \lambda I$ are positive definite and (5.9) yields a contradiction since y and z are not both zero. The inequalities (5.8) now follow from Lemma 5.4.2. □

That the bounds on λ_m and λ_{m+1} in Lemma 5.4.4 are attainable is shown by (5.6) and (5.7). (For the interior eigenvalues, inequalities (5.8) can, of course, be improved by applying Cauchy's interlace theorem.)

A bound for the 2-norm condition number $\kappa_2(C) = \|C\|_2\|C^{-1}\|_2$ is immediate.

Lemma 5.4.5 *If H and M are positive definite then*

$$\kappa_2(C) \le \|C\|_2 \max\{\|H^{-1}\|_2, \|M^{-1}\|_2\}. \qquad \square$$

5.5 Modifying the Inertia: A General Perturbation

Let $A \in \mathbb{R}^{n\times n}$ be symmetric. We denoted by $\mu^{(k)}(A)$ the distance from A to the symmetric matrices with at least k more nonnegative eigenvalues than A (assuming that A has at least k negative eigenvalues):

$$\begin{aligned}\mu^{(k)}(A) = \min\{\|\Delta A\| : \Delta A = \Delta A^T,\\ i_+(A+\Delta A) + i_0(A+\Delta A) \ge i_+(A) + i_0(A) + k\}.\end{aligned} \tag{5.10}$$

The distance is characterized by the following theorem, which generalizes a result giving the distance to the nearest symmetric positive semidefinite matrix [51]. Recall that a norm $\|\cdot\|$ is a unitarily invariant norm on $\mathbb{R}^{n\times n}$ if $\|UAV\| = \|A\|$ for all orthogonal U and V. We will need the characterization that any unitarily invariant norm is a symmetric gauge function on the singular values, that is, $\|A\| = \phi(\sigma_1, \ldots, \sigma_n)$, where ϕ is an absolute vector norm that is invariant under permutations of the entries of its argument [60, Thm. 7.4.24],[84, Thm. 3.6, p. 78].

Theorem 5.5.1 *Let the symmetric matrix $A \in \mathbb{R}^{n\times n}$ have the spectral decomposition $A = Q\Lambda Q^T$, where Q is orthogonal and $\Lambda = \operatorname{diag}(\lambda_i)$ with*

$$\lambda_1 \le \cdots \le \lambda_p < 0 \le \lambda_{p+1} \le \cdots \le \lambda_n,$$

and assume that $p \geq k$. *Then for any unitarily invariant norm, an optimal perturbation in* (5.10) *is*

$$\Delta A = Q \operatorname{diag}(\tau_i) Q^T, \qquad \tau_i = \begin{cases} -\lambda_i, & i = p-k+1{:}p, \\ 0, & \text{otherwise.} \end{cases} \tag{5.11}$$

and

$$\mu^{(k)}(A) = \phi(\tau_1, \ldots, \tau_n).$$

Proof: A generalization of the Wielandt–Hoffman theorem [60, Thm. 7.4.51], [84, p. 205] says that if A and $A + \Delta A$ are symmetric then

$$\|\Delta A\| \geq \|\operatorname{diag}(\lambda_i(A + \Delta A) - \lambda_i(A))\|$$

for any unitarily invariant norm. If ΔA is a feasible perturbation in (5.10) then

$$\begin{aligned} \|\Delta A\| &\geq \|\operatorname{diag}(0, \ldots, 0, \lambda_{p-k+1}(A + \Delta A) - \lambda_{p-k+1}(A), \ldots, \\ &\qquad \lambda_p(A + \Delta A) - \lambda_p(A), 0, \ldots, 0)\| \\ &\geq \|\operatorname{diag}(0, \ldots, 0, -\lambda_{p-k+1}(A), \ldots, -\lambda_p(A), 0, \ldots, 0)\|, \end{aligned}$$

where we have used $\lambda_p(A + \Delta A) \geq \cdots \geq \lambda_{p-k+1}(A + \Delta A) \geq 0$ and the gauge function property of the norms. It is easily seen that equality is attained for the perturbation given in the statement of the theorem and that this perturbation is feasible. □

5.6 Modifying the Inertia: A Structured Perturbation

Returning to the partitioned matrix (5.4), we are interested in finding a perturbation ΔH such that

$$C + \Delta C = \begin{bmatrix} H + \Delta H & A \\ A^T & -M \end{bmatrix}$$

has a given inertia. For the analysis in this section, C can be regarded as a general block 2×2 symmetric matrix—we will not need A to have full rank or the diagonal blocks to process any definiteness properties, and m and n are arbitrary.

For the KKT matrix, practical interest is in increasing the number of positive eigenvalues (in view of Theorem 5.2.1 and Lemma 5.4.1), so we define, analogously to (5.10),

$$\begin{aligned}\psi^{(k)}(C) = \min\{\|\Delta H\| : \Delta H = \Delta H^T,\\ i_+(C+\Delta C) + i_0(C+\Delta C) \geq i_+(C) + i_0(C) + k\}.\end{aligned} \tag{5.12}$$

Clearly, an optimal ΔH in (5.12) can be taken to be positive semidefinite and of rank k, hence of the form $\Delta H = VV^T$ with $V \in \mathbb{R}^{n\times k}$ $(k \leq n)$. Our solution to this problem is based on the following lemma. The lemma is not new; essentially the same result can be found in [4, Lem. 2.1] and [5, Cor. 2.2], for example.

Lemma 5.6.1 *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and nonsingular and let $W \in \mathbb{R}^{n\times k}$. Then $i_+(A+WW^T) + i_0(A+WW^T) = i_+(A) + i_0(A) + k$ if and only if $-I_k - W^TA^{-1}W$ is positive semidefinite.*

Proof: We have the congruences

$$B = \begin{array}{c} \\ n \\ k \end{array}\!\!\begin{array}{c} \begin{array}{cc} n & k \end{array} \\ \begin{bmatrix} A & W \\ W^T & -I_k \end{bmatrix} \end{array} \sim \begin{bmatrix} A & 0 \\ 0 & -I_k - W^TA^{-1}W \end{bmatrix}$$

and, for a suitable permutation Π,

$$\Pi^T B \Pi = \begin{bmatrix} -I_k & W^T \\ W & A \end{bmatrix} \sim \begin{bmatrix} -I_k & 0 \\ 0 & A + WW^T \end{bmatrix}.$$

It follows that

$$\text{inertia}(A) + \text{inertia}(-I_k - W^TA^{-1}W) = \text{inertia}(-I_k) + \text{inertia}(A + WW^T),$$

that is,

$$\operatorname{inertia}(A + WW^T) = \operatorname{inertia}(A) + \operatorname{inertia}(-I_k - W^T A^{-1} W) - \operatorname{inertia}(-I_k).$$

The result is immediate. □

We apply Lemma 5.6.1 with A the matrix C (assumed to be nonsingular) and

$$W = \begin{array}{c} \\ n \\ m \end{array} \begin{array}{c} \overset{k}{} \\ \begin{bmatrix} V \\ 0 \end{bmatrix} \end{array}.$$

The lemma tells us that we need to minimize $\|VV^T\|$ subject to

$$\begin{bmatrix} V^T & 0^T \end{bmatrix} C^{-1} \begin{bmatrix} V \\ 0 \end{bmatrix} \in \mathbb{R}^{k \times k} \tag{5.13}$$

having all its eigenvalues less than or equal to -1. Writing $G = C^{-1}(1{:}n, 1{:}n)$, this constraint is

$$\lambda_i(V^T G V) \leq -1, \quad i = 1{:}k. \tag{5.14}$$

By Corollary 5.3.3, a matrix satisfying (5.14) exists only if G has at least k negative eigenvalues, which we assume to be the case. The following lemma and corollary show how to minimize $\|VV^T\|$ for any unitarily invariant norm subject to (5.14).

Lemma 5.6.2 *Let $A \in \mathbb{R}^{n \times n}$ be symmetric with the spectral decomposition $A = Q \operatorname{diag}(\lambda_i) Q^T$, where Q is orthogonal and*

$$\lambda_1 \leq \cdots \leq \lambda_{p-1} \leq 0 < \lambda_p \leq \cdots \leq \lambda_n.$$

Let $X \in \mathbb{R}^{n \times k}$ with $k \leq n$ and assume that $p \leq n - k + 1$. All matrices X that minimize all the singular values of X subject to satisfying the inequalities

$$\lambda_i(X^T A X) \geq 1, \quad i = 1{:}k, \tag{5.15}$$

are given by

$$X = Q(1{:}n, n-k+1{:}n)\,\mathrm{diag}(\lambda_{n-k+1}, \ldots, \lambda_n)^{-1/2} V, \tag{5.16}$$

where $V \in \mathbb{R}^{k\times k}$ *is an arbitrary orthogonal matrix.*

Proof: Let

$$X = U\Sigma V^T, \qquad \Sigma = \begin{bmatrix} S \\ 0 \end{bmatrix}, \qquad S = \mathrm{diag}(\sigma_i) \in \mathbb{R}^{k\times k}$$

be the singular value decomposition of X. Then

$$X^T A X - I = V\Sigma^T U^T A U \Sigma V^T - I = V(SBS - I)V^T,$$

where $B = (U^T A U)(1{:}k, 1{:}k)$. The constraint (5.15) is therefore equivalent to $SBS - I$ being positive semidefinite, which implies that

$$b_{ii} \geq \frac{1}{\sigma_i^2}, \qquad i = 1{:}k. \tag{5.17}$$

We wish to maximize the reciprocals σ_i^{-2}. Now the diagonal of the symmetric matrix B is largest in modulus when it contains the eigenvalues of B, that is, when B is diagonal, (5.17) is equivalent to (5.15). This fact is easily shown. Let $M \in \mathbb{R}^{n\times n}$ be symmetric. Then, for the Frobenius norm, we have

$$\|M\|_F^2 = \sum_{i\neq j} m_{ij}^2 + \sum_i m_{ii}^2 \geq \sum_i m_{ii}^2.$$

Equality holds in the inequality, and hence $\sum_i m_{ii}^2$ is maximized, when $m_{ij} = 0$ for $i \neq j$. Therefore $m_{11}, \ldots, m_{nn}$ must be the eigenvalues of M.

Hence for optimality we need to choose $U = Q(1{:}n, n-k+1{:}n)$ and then, to attain the bounds in (5.17), $\sigma_i = \lambda_{n-k+i}^{-1/2}$ (note that the σ_i are arranged in decreasing order). The matrix V is arbitrary. □

Corollary 5.6.3 *Under the condition of Lemma* 5.6.2*, the matrix* (5.16) *minimizes* $\|XX^T\|$ *subject to* (5.15) *for any unitarily invariant norm.*

Proof: The singular values of XX^T are the squares of the singular values of X, which are minimized by the matrix (5.16). The result follows from the gauge function property of unitarily invariant norms. □

We now summarize our findings in a theorem.

Theorem 5.6.4 *Let $H \in \mathbb{R}^{n\times n}$ and $M \in \mathbb{R}^{m\times m}$ be symmetric and $A \in \mathbb{R}^{n\times m}$, and let*

$$C = \begin{bmatrix} H & A \\ A^T & -M \end{bmatrix}.$$

Assume C is nonsingular and let $G = C^{-1}(1{:}n, 1{:}n)$. There exists a feasible perturbation in the definition of $\psi^{(k)}(C)$ if and only if G has at least k negative eigenvalues. Let $G = Q\,\mathrm{diag}(\gamma_i)Q^T$ be a spectral decomposition, where Q is orthogonal and $\gamma_1 \le \cdots \le \gamma_n$. Then, for any unitarily invariant norm, an optimal perturbation in (5.12) *is*

$$\Delta H = -Q\,\mathrm{diag}(\gamma_1^{-1}, \ldots, \gamma_k^{-1}, 0, \ldots, 0)Q^T \tag{5.18}$$

and, in terms of the underlying gauge function ϕ,

$$\psi^{(k)}(C) = \phi(\gamma_1^{-1}, \ldots, \gamma_k^{-1}, 0, \ldots, 0). \qquad \square \tag{5.19}$$

The perturbation (5.18) is full, in general, so may not be a suitable perturbation when H is large and sparse. It is natural, therefore, to consider diagonal perturbations. The next result shows that a perturbation consisting of a suitable multiple of the identity matrix is also optimal in the 2-norm. This result can be deduced from Theorem 5.6.4, but we give an independent proof for completeness.

Theorem 5.6.5 *Under the same condition as in Theorem* 5.6.4*, an optimal perturbation in* (5.12) *in the* 2*-norm is*

$$\Delta H = -\gamma_k^{-1} I. \tag{5.20}$$

Proof: Consider perturbations to C of the form $\Delta C = WW^T$ with

$$W = \begin{array}{c} \\ n \\ m \end{array} \overset{n}{\begin{bmatrix} \alpha I \\ 0 \end{bmatrix}}. \tag{5.21}$$

It is straightforward to prove an analogue of Lemma 5.6.1 which says that if $A \in \mathbb{R}^{n\times n}$ is symmetric and nonsingular, $W \in \mathbb{R}^{n\times k}$ and $p \leq k$, then $i_+(A + WW^T) + i_0(A + WW^T) = i_+(A) + i_0(A) + p$ if and only if $-I_k - W^T A^{-1} W$ has exactly p nonnegative eigenvalues. Applying this result to (5.21) we find that $\Delta H = WW^T$ is a feasible perturbation in (5.12) if and only if $-I_n - \alpha^2 G$ has k nonnegative eigenvalues, where $G = C^{-1}(1{:}n, 1{:}n)$. We are assuming that G has at least k nonnegative eigenvalues, so the minimal value of α^2 is $-1/\gamma_k$. This gives $\|\Delta C\|_2 = -1/\gamma_k$, which, in view of (5.19), shows that (5.21) is an optimal perturbation in the 2-norm. $\square$

Note that whereas the perturbation (5.18) increases $i_+ + i_0$ by exactly k, the perturbation (5.20) will increase it by more than k if $\gamma_k = \gamma_{k+1} = \cdots = \gamma_{k+r}$ with $r \geq 1$.

5.7 A Projected Hessian Approach

For the matrix C with $n \geq m$, there is an alternative way to find a perturbation to H of minimal norm such that the second order sufficiency condition (5.2) is satisfied. As noted earlier, the condition (5.2) is equivalent to the projected Hessian $Z^T H Z$ being positive definite, where the columns of $Z \in \mathbb{R}^{n\times(n-m)}$ form a basis for null(A^T), which we will take to be orthonormal. Therefore we are interested in solving the problem

$$\min\{\|\Delta H\| : Z^T(H + \Delta H)Z \text{ is positive semidefinite}\}. \tag{5.22}$$

From Theorem 5.5.1 we know that an optimal *arbitrary* perturbation E that makes $Z^T H Z + E$ positive semidefinite is, for any unitarily invariant norm,

$$E = U \operatorname{diag}(\max(-\mu_i, 0))U^T, \tag{5.23}$$

where $Z^T H Z = U \operatorname{diag}(\mu_i)U^T$ with $\mu_1 \leq \cdots \leq \mu_{n-m}$ is a spectral decomposition. Hence any feasible ΔH in (5.22) satisfies

$$\|E\| \leq \|Z^T \Delta H Z\| \leq \|Z^T\|_2 \|\Delta H\| \|Z\|_2 \leq \|\Delta H\|,$$

using an inequality for unitarily invariant norms from [61, p. 211]. But the perturbation (5.23) is achieved in (5.22) by setting $\Delta H = ZEZ^T$, and $\|\Delta H\| \leq \|Z\|_2 \|E\| \|Z^T\|_2 \leq \|E\|$. We conclude that

$$\Delta H = ZU \operatorname{diag}(\max(-\mu_i, 0))U^T Z^T \tag{5.24}$$

is a solution to (5.22) for any unitarily invariant norm. For the 2-norm, another solution is

$$\Delta H = \max(-\mu_1, 0) Z Z^T. \tag{5.25}$$

For the special case of the KKT matrix defined as in (5.3), for which (5.2) is equivalent to $\operatorname{inertia}(K) = (n, m, 0)$ by Theorem 5.2.1, the perturbation (5.24) is, necessarily, of the same norm as (5.18) for $k = n - i_+(K)$ in Theorem 5.6.4, although this equivalence is not obvious from the formulae.

When M is positive definite, or M is positive semidefinite and A has full rank, Lemma 5.4.2 shows that we can achieve the desired inertia $(n, m, 0)$ by choosing ΔH to make $H + \Delta H$ positive definite. Theorem 5.5.1 with $k = p$ shows that the smallest value of $\|\Delta H\|_2$ for which $H + \Delta H$ is positive semidefinite is $\max(-\lambda_{\min}(H), 0)$. By definition, this perturbation is at least as large as the optimal ones (5.18) and (5.24), and from (5.24) we have

$$\|\Delta H\|_2 \leq \max(-\lambda_{\min}(Z^T H Z), 0),$$

which can be arbitrarily smaller than $\max(-\lambda_{\min}(H), 0)$. We note, in particular, that the perturbation (5.18), (5.20), (5.24) and (5.25) all have 2-norms uniformly bounded by $\|H\|_2$, which is an important property for optimization application [44].

We give a numerical example for illustration. Consider the KKT matrix

$$K = \left[\begin{array}{cc|c} -1 & 1 & 0 \\ 1 & -100 & 1 \\ \hline 0 & 1 & 0 \end{array}\right], \quad \lambda(K) = \{-1.00 \times 10^2, -9.90 \times 10^{-1}, 1.01 \times 10^{-2}\},$$

where the eigenvalues are given to three significant figures. Hence $\text{inertia}(K) = (1, 2, 0)$, and we want to change the inertia to $(2, 1, 0)$. Since

$$K^{-1} = \left[\begin{array}{cc|c} -1 & 0 & 1 \\ 0 & 0 & 1 \\ \hline 1 & 1 & 99 \end{array}\right],$$

we find immediately from Theorem 5.6.4 with $k = 1$ that

$$\Delta H = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.26}$$

is a matrix of minimal norm, for any unitarily invariant norm, that changes the inertia of K to $(1, 1, 1)$; indeed

$$K + \Delta K = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -100 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \lambda(K + \Delta K) = \{-1.00 \times 10^2, 0, 2.00 \times 10^{-2}\}.$$

For the projected Hessian approach we have $Z = [1 \;\; 0]^T$, $Z^T H Z = -1$ and (5.24) yields the perturbation (5.26). To achieve the inertia $(2, 1, 0)$ that is required for the condition (5.2) to hold, we can replace ΔH by $(1 + \epsilon)\Delta H$ for any $\epsilon > 0$.

In order to perturb H to make it positive definite, which also produce the desired inertia, which must make a perturbation of 2-norm at least $-\lambda_{\min}(H) =$

1.00×10^2, which is two orders of magnitude larger than the minimal-norm perturbation (5.26).

5.8 Practical Algorithm

We now turn to the optimization applications. We consider the situation where a linear system $Cx = b$ must be solved, but C needs to be perturbed in its $(1,1)$ block, if necessary, to ensure that it has the inertia $(n, m, 0)$.

We assume that a block $\mathrm{LDL}^{\mathrm{T}}$ factorization of C is computed,

$$PCP^T = LDL^T,$$

where L is unit lower triangular and D is block diagonal with blocks of dimension 1 or 2; P is a permutation matrix that can be chosen according to one of various pivoting strategies mentioned in Chapter 2. Since C and D have the same inertia it is trivial to evaluate the inertia of C. If $i_+(C)$ is less than n then Theorem 5.6.4 shows that to determine the optimal perturbation (5.18) we need to compute the $k = n - i_+(C)$ most negative eigenvalues of $G = C^{-1}(1{:}n, 1{:}n)$ and their corresponding eigenvectors; for the optimal 2-norm perturbation (5.20) it suffices to determine the kth most negative eigenvalue of G. To confirm that there are k negative eigenvalue of G, we apply Cauchy's interlace theorem, which yields

$$\lambda_i(G) \leq \lambda_{i+m}(C^{-1}), \qquad i = 1{:}n.$$

Hence if C has only $i_+(C) < n$ positive eigenvalues then G has at least $n - i_+(C)$ negative eigenvalues.

Since C may be large and sparse it is undesirable to form G explicitly. Therefore we suggest that the k most negative eigenvalues of G and their corresponding eigenvectors be computed using the Lanczos algorithm, which requires only the

ability to form matrix-vector products with G. To form $y = Gx$ we note that

$$\begin{bmatrix} y \\ z \end{bmatrix} = C^{-1} \begin{bmatrix} x \\ 0 \end{bmatrix},$$

where $z \in \mathbb{R}^m$ is not of interest. Hence y is the first n components of the solution to the linear system

$$C \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix},$$

which can be solved using the block $\mathrm{LDL}^{\mathrm{T}}$ factorization.

Note that the perturbation (5.18) makes $C + \Delta C$ singular, since it perturbs k negative eigenvalues to the origin. Similarly, the perturbation (5.20) produces at least one zero eigenvalue. In practice a nonsingular $C + \Delta C$ is required, and the natural approach is to modify the perturbations so that the eigenvalues are moved to a positive tolerance δ instead of 0.

Having computed an optimal perturbation ΔH we have to refactorize $C + \Delta C$ in order to solve $(C + \Delta C)x = b$. It does not seem practical to apply updating techniques to the original factorization, since the update may not be of low rank.

However, if $M = 0$, $m \geq n$ and $A \in \mathbb{R}^{m \times n}$ has full column rank, we can update the KKT matrix K using a formula in Fletcher [31, ex. 12.12, p. 327], in which the *pseudoinverse* of A is required. Given

$$K + \Delta K = \begin{bmatrix} H + \Delta H & A \\ A^T & 0 \end{bmatrix},$$

we have

$$(K + \Delta K)^{-1} = K^{-1} - \begin{bmatrix} 0 & 0 \\ 0 & S \end{bmatrix}, \tag{5.27}$$

where $S = A^+ \Delta H (A^+)^T$ and $A^+ = (A^T A)^{-1} A^T$ is the pseudoinverse of A. When solving a linear system $(K + \Delta K)x = b$ using (5.27), the solution is given by

$$x = K^{-1} b - \begin{bmatrix} 0 & 0 \\ 0 & S \end{bmatrix} b.$$

Note that the block $\mathrm{LDL}^{\mathrm{T}}$ factorization of K is readily available when determining the inertia of K. The constraint matrix A^T, and hence the pseudoinverse A^+, remains constant within an optimization application (at least for linear equality constrained optimizations), which means that A^+ can be computed only once via a singular value decomposition or a QR factorization, so it may not be too expensive.

In the case where $M = 0$, our algorithm provides, as a by product, a direction of negative curvature, which is defined as a vector p for which (cf. (5.2)) $A^T p = 0$ and $p^T H p < 0$. Such directions are needed in nonlinear programming to achieve convergence to points that satisfy second order necessary conditions for optimality. Writing the perturbation (5.18) as $\Delta H = VV^T$, we know that the matrix (5.13), which we denote by S, is negative definite. Now

$$S = \begin{bmatrix} V^T & 0^T \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}, \quad \text{where} \quad \begin{bmatrix} H & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} V \\ 0 \end{bmatrix}.$$

Thus $HX + AY = V$ and $A^T X = 0$, which implies $X^T H X = S^T = S$. The jth column x_j of X satisfies $x_j^T H x_j = s_{jj} < 0$, since S is negative definite, and $A^T x_j = 0$. Thus, every column of X is a direction of negative curvature.

An alternative approach is to work with the projected Hessian $Z^T H Z$ and to compute an optimal perturbation ΔH from (5.24) or (5.25). Again, the Lanczos algorithm can be used, this time to compute the negative eigenvalues of $Z^T H Z$.

5.9 Numerical Experiments

Numerical experiments have been performed in MATLAB where the unit roundoff $u \approx 1.1 \times 10^{-16}$. We generated 50 random KKT matrices K with $n = 20$, $m = 5$ and elements normally distributed with mean 0 and variance 1. The perturbation is taken to be $(1+\delta)\Delta H$ where the tolerance $\delta = u\|K\|_\infty$, and the required inertia is $(n, m, 0)$.

First we focus on demonstrating that perturbations (5.18), (5.20), (5.24) and (5.25) can achieve the given inertia in practice. The inverse of K was formed explicitly by MATLAB function `inv`. Then the eigenvalues of $G = K^{-1}(1{:}n, 1{:}n)$ were computed using MATLAB function `eig`. The null space Z was computed by MATLAB function `QR`. In more than 90% of the test cases, all four perturbations yielded the specific inertia. In a few cases that our perturbations gave the inertia $(n+1, m-1, 0)$, they were easily fixed by reducing the tolerance δ.

Then we implemented our practical algorithm in which G was not formed explicitly and the eigenvalues were computed by the Lanczos method. An M-file for the Lanczos method was provided by Thierry Braconnier. Our conclusions of the experiments are as follows.

1. When computing the optimal Frobenius norm perturbation (5.18), we need to compute the k most negative eigenvalues of $G = K^{-1}(1{:}n, 1{:}n)$, using the Lanczos method. This approach is of limited practical use because the Lanczos method suffers convergence difficulties when there are clustered eigenvalues close to the origin. In our experiments, the Lanczos method often failed to converge, even if only two eigenvalues were needed. A shift and invert technique was employed to overcome the problem. Essentially, it involves working with $(G - \widehat{\lambda}I)^{-1}$ where $\widehat{\lambda}$ is a good approximation to the required eigenvalue λ. Let x denote the eigenvector corresponding to λ, and let μ be an eigenvalue of $(G - \delta I)^{-1}$. We have

$$Gx = \lambda x \Leftrightarrow (G - \delta I)x = (\lambda - \delta)x \Leftrightarrow (G - \delta I)^{-1}x = (\lambda - \delta)^{-1}x,$$

   which shows that G and $(G - \delta I)^{-1}$ share the same set of eigenvectors, and their eigenvalues are related by $\mu = (\lambda - \delta)^{-1}$. Unfortunately, G then needs to be formed explicitly if the shift and invert technique is to be used.

2. Finding the optimal 2-norm perturbation (5.20) gives the same problem as

above. To determine the kth smallest eigenvalue, we have to compute all the other $k-1$ most negative eigenvalues. No existing eigenvalue solver computes a target eigenvalue which is not an extreme eigenvalue for a large and possibly sparse matrix.

3. When the projected Hessian approach is used, the Lanczos method can be used to compute all the negative eigenvalues of $Z^T H Z$ to determine (5.24). We run into the same problem when clustered eigenvalues occur. For perturbation (5.25), we need the most negative eigenvalue of $Z^T H Z$. In this case, the Lanczos method is shown to be robust and efficient. Provided the null space Z can be computed efficiently, we recommend this approach.

5.10 Concluding Remarks

We have derived a minimal-norm perturbation, valid for any unitarily invariant norm, for perturbing only the (1,1) block of a block 2×2 matrix so that a specific inertia is achieved.

In implementing our practical algorithm, we encountered convergence difficulties of the Lanczos method in computing clustered eigenvalues, but how to deal with such difficulties is beyond the scope of this thesis.

In our experiments, the null space Z was computed by MATLAB function `QR` and is the most expensive part of our practical algorithm. How to form the null space efficiently for large and sparse matrices is again beyond the scope of thesis. We mention in passing two good references: [8], [73].

An updating formula (5.27) is given for the KKT matrix K which does not involve a refactorization of $K + \Delta K$. Unfortunately, the formula cannot be generalized for general block 2×2 matrices. To derive some efficient updating schemes, specially for low rank perturbations, is desirable.

It is an open problem to develop an efficient algorithm for computing a target eigenvalue, which is not an extreme eigenvalue, of a large and possibly sparse matrix.

Chapter 6

Generalized Hermitian Eigenvalue Problems

6.1 Introduction

A matrix pencil is a family of matrices $A - \lambda B$, parameterized by a complex number λ. The generalized eigenvalue problem is to compute the nontrivial solutions of the equation

$$Ax = \lambda Bx. \tag{6.1}$$

The natural generalization of the standard Hermitian eigenvalue problem is to pairs of Hermitian matrices, that is, A and B are Hermitian. A matrix $X \in \mathbb{C}^{n\times n}$ is said to be Hermitian if $X = X^*$ where X^* denotes the conjugate transpose of X. Throughout this chapter, we will assume A and B to be Hermitian. The generalized Hermitian eigenproblem arises in many engineering application like structual dynamics for computing damped natural modes [70], [72], and when a Sturm–Liouville problem is discretized by high order implicit difference schemes [19], [74].

For the standard Hermitian eigenvalue problem, we have a complete set of real eigenvalues and orthogonal eigenvectors. However symmetry alone is not enough to guarantee such properties for the generalized case, as the following example shows. Let

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

Then the pair (A, B) is Hermitian, and the eigenvalues of the pair are 1 (twice)

with only one eigenvector $[0\ \ 1]^T$. We call such pair of matrices a *defective pair*. This phenomenon is best explained by theorems of Frobenius [35], dating back to 1910, which explain the properties of the matrix product $B^{-1}A$. We describe Frobenius's theorems in Section 6.2. In short, the theorems say that a generalized Hermitian eigenvalue problem is equivalent to a standard eigenvalue problem that is potentially any standard eigenvalue problem. Of particular interest is the case where (A, B) is a *definite pair*, for which all the eigenvalues are real and the matrices A and B can be simultaneously diagonalized; see Section 6.3. The proof of the latter property is constructive. It suggests an algorithm to reduce a generalized eigenvalue problem to a standard Hermitian eigenvalue problem. In Section 6.4, we look at the simultaneous diagonalization approach and summarize Crawford's work which shows how to implement this reduction efficiently when A and B are banded.

A relevant nearness problem is: "Given an indefinite pair (A, B), what is the nearest definite pair?" We investigate this nearness problem in Section 6.5. We introduce the term *inner numerical radius* and show an elegant solution of this nearness problem in the 2-norm. A simple algorithm is proposed for computing the inner numerical radius and optimal perturbations. When (A, B) is a *normal pair*, we exploit the characteristics of the eigenvalues and suggest an alternative method for determining the inner numerical radius. Concluding remarks are given in Section 6.6.

6.2 Properties of Hermitian Matrix Product

In this section, we survey the properties of products of two Hermitian matrices. A well known result of Frobenius [35] states that every $n \times n$ matrix is a product of two symmetric matrices. However, not every $n \times n$ matrix is a product of two Hermitian matrices. We note that there is some confusion in the literature over

the latter distinction, for example [84, p. 281].

Based on [9], [60], [85], we state Frobenius's results in modern notation and give a clear summary of the properties of the Hermitian matrix product. First we show that every $n \times n$ real (complex) matrix is a product of two real (complex) symmetric matrices. We present the real and complex case in two theorems. This is natural in the sense that one would expect to end up with both A, B real (complex) when M is real (complex).

Theorem 6.2.1 (Frobenius) *Every $M \in \mathbb{R}^{n\times n}$ is a product of two real symmetric matrices A and B^{-1}.*

Proof: We give the proof for the case $M = B^{-1}A$. The proof is similar for $M = AB^{-1}$. For any matrix $M \in \mathbb{R}^{n\times n}$, there exists a nonsingular $X \in \mathbb{R}^{n\times n}$ such that

$$X^{-1}MX = \operatorname{diag}(C_{k_1}(a_1, b_1), \ldots, C_{k_p}(a_p, b_p), J_{k_{p+1}}(\lambda_{p+1}), \ldots, J_{k_q}(\lambda_q)), \tag{6.2}$$

where $C_k(a, b) \in \mathbb{R}^{k\times k}$ is the *real Jordan block*

$$C_k(a,b) \equiv \begin{bmatrix} C(a,b) & I & & \\ & \ddots & \ddots & \\ & & C(a,b) & I \\ & & & C(a,b) \end{bmatrix}, \qquad C(a,b) = \begin{bmatrix} a & b \\ -b & a \end{bmatrix}, \tag{6.3}$$

and $J_k(\lambda) \in \mathbb{R}^{k\times k}$ is the *Jordan block*

$$J_k(\lambda) \equiv \begin{bmatrix} \lambda & 1 & & \\ & \ddots & \ddots & \\ & & \lambda & 1 \\ & & & \lambda \end{bmatrix}.$$

Here a_k, b_k, λ_k are real; see [60, p. 151-153] for details of this real Jordan canonical form. Now let $P_k \in \mathbb{R}^{k\times k}$ be the permutation matrix

$$P_k = \begin{bmatrix} 0 & & 1 \\ & \iddots & \\ 1 & & 0 \end{bmatrix} \tag{6.4}$$

which is symmetric itself and $P_k^2 = I$. It is easily shown that $\widetilde{C}_k(a_k, b_k) = P_k C_k(a_k, b_k)$ and $\widetilde{J}_k(\lambda) = P_k J_k(\lambda)$ are symmetric. Further let

$$\begin{aligned} \widetilde{B} &= \mathrm{diag}(P_{k_1}, \ldots, P_{k_q}), \\ \widetilde{A} &= \mathrm{diag}(\widetilde{C}_{k_1}(a_1, b_1), \ldots, \widetilde{C}_{k_p}(a_p, b_p), \widetilde{J}_{k_{p+1}}(\lambda_{p+1}), \ldots, \widetilde{J}_{k_q}(\lambda_q)). \end{aligned}$$

Note that $\widetilde{A}$, $\widetilde{B}$ are symmetric with $\widetilde{B}$ nonsingular. We can rewrite (6.2) as $X^{-1}MX = \widetilde{B}\widetilde{A}$. By taking $B^{-1} = X\widetilde{B}X^T$ and $A = X^{-T}\widetilde{A}X^{-1}$, we complete the proof. □

Theorem 6.2.2 (Frobenius) *Every $M \in \mathbb{C}^{n\times n}$ is a product of two complex symmetric matrices A and B^{-1}.*

Proof: The proof is similar to Theorem 6.2.1, but uses only the complex Jordan form. For any matrix $M \in \mathbb{C}^{n\times n}$, there exists a nonsingular $X \in \mathbb{C}^{n\times n}$ such that

$$X^{-1}MX = \mathrm{diag}(J_{k_1}(\lambda_1), \ldots, J_{k_q}(\lambda_q)), \tag{6.5}$$

where $J_k(\lambda) \in \mathbb{C}^{k\times k}$ is the *Jordan block*

$$J_k(\lambda) \equiv \begin{bmatrix} \lambda & 1 & & \\ & \ddots & \ddots & \\ & & \lambda & 1 \\ & & & \lambda \end{bmatrix}. \tag{6.6}$$

Now let $P_k \in \mathbb{R}^{k\times k}$ be the permutation matrix as defined in (6.4). It is easily shown that $\widetilde{J}_k(\lambda) = P_k J_k(\lambda)$ is complex symmetric. Further let

$$\widetilde{B} = \mathrm{diag}(P_{k_1}, \ldots, P_{k_q}), \qquad \widetilde{A} = \mathrm{diag}(\widetilde{J}_{k_1}(\lambda_1), \ldots, \widetilde{J}_{k_q}(\lambda_q)).$$

Note that $\widetilde{A}$, $\widetilde{B}$ are symmetric with $\widetilde{B}$ nonsingular. We can rewrite (6.5) as $X^{-1}MX = \widetilde{B}\widetilde{A}$. By taking $B^{-1} = X\widetilde{B}X^T$ and $A = X^{-T}\widetilde{A}X^{-1}$, we obtain the result. □

Now we show by example that Theorem 6.2.1 cannot be generalized for the Hermitian case, that is, for complex M we cannot always take A and B Hermitian. Assume B is Hermitian and nonsingular, and consider

$$BM := \begin{bmatrix} a & b \\ \bar{b} & c \end{bmatrix} \begin{bmatrix} i & 1 \\ 0 & i \end{bmatrix} = \begin{bmatrix} ia & a+ib \\ i\bar{b} & \bar{b}+ic \end{bmatrix} =: A,$$

where a, c are real since B is Hermitian. For $A = A^*$, we have

$$\begin{aligned} ia = \overline{ia} &\Rightarrow a = 0, \\ i\bar{b} = \overline{a+ib} &\Rightarrow b = 0, \\ \overline{b} + ic = \overline{\bar{b}+ic} &\Rightarrow c = 0, \end{aligned}$$

thus $B = 0$ and hence we have a contradiction.

In fact, Theorem 6.2.1 holds for complex matrices if and only if M is similar to a real matrix, so that the nonreal eigenvalues come in conjugate pairs [85].

Theorem 6.2.3 (Frobenius) *A matrix $M \in \mathbb{C}^{n\times n}$ is a product of two Hermitian matrices A and B^{-1} if and only if M is similar to a real matrix.*

Proof: If M is similar to a real matrix, there exists a nonsingular matrix Y such that $M = YSY^{-1}$. Using Theorem 6.2.1, we have $S = \widetilde{B}\widetilde{A}$ where $\widetilde{A}$, $\widetilde{B}$ are symmetric. Taking $B^{-1} = Y\widetilde{B}Y^*$ and $A = Y^{-*}\widetilde{A}Y^{-1}$ gives the result.

Conversely, let $M = X\,\mathrm{diag}(J_{k_1}(\lambda_1), \ldots, J_{k_q}(\lambda_q))X^{-1} = XJX^{-1}$ be the Jordan canonical form where $k_1 + \cdots + k_q = n$ and $J_k(\lambda_k)$ is defined in (6.6). Consider $M = B^{-1}A = B^{-1}(AB^{-1})B = B^{-1}M^*B$, which implies M is similar to M^*, hence J is similar to J^*. This means, if λ is an eigenvalue of J then $\bar{\lambda}$ is also an eigenvalue with the same multiplicity and the same Jordan structure. That is, the eigenvalues are real or in complex conjugate pairs. Thus M can be written in the real Jordan canonical form (6.2),

$$M = \widehat{X}\,\mathrm{diag}(C_{k_1}(a_1, b_1), \ldots, C_{k_p}(a_p, b_p), J_{k_{p+1}}(\lambda_{p+1}), \ldots, J_{k_q}(\lambda_q))\widehat{X}^{-1} = \widehat{X}G\widehat{X}^{-1},$$

where C_k is defined in (6.3) and G is real. We have

$$M = (B^{-1}\widehat{X}^{-*})G^T(\widehat{X}^*B) = (\widehat{X}^*B)^{-1}G^T(\widehat{X}^*B),$$

as required. □

In addition, we can show that B can be positive definite if and only if M is similar to a Hermitian matrix.

Theorem 6.2.4 (Frobenius) *A matrix $M \in \mathbb{C}^{n\times n}$ is a product of two Hermitian matrices A and B^{-1} with B^{-1} positive definite if and only if M is similar to a Hermitian matrix.*

Proof: If B^{-1} is positive definite, it has a positive definite square root $B^{-\frac{1}{2}}$. Then

$$M = B^{-1}A = B^{-\frac{1}{2}}(B^{-\frac{1}{2}}AB^{-\frac{1}{2}})B^{\frac{1}{2}} = B^{-\frac{1}{2}}\widetilde{A}B^{\frac{1}{2}},$$

where $\widetilde{A}$ is Hermitian since $\widetilde{A} = \widetilde{A}^*$. Conversely, write

$$M = Y\widetilde{M}Y^{-1} = (YY^*)Y^{-*}\widetilde{M}Y^{-1} = B^{-1}A,$$

where $\widetilde{M}$, A, B^{-1} are Hermitian with B^{-1} positive definite, as required. □

We summarize our results in Table 6.1.

M	(A, B)	Similarity of M	Eigenvalues
real	real symmetric	—	real, complex conjugate
complex	complex symmetric	—	real, complex
complex	Hermitian	real	real, complex conjugate
complex	Hermitian $(B > 0)^a$	Hermitian	real

[a] $B > 0$ means B is positive definite.

Table 6.1: Properties of matrix product $M = B^{-1}A$.

6.3 Definite Pairs

In this section, we introduce the concept of the field of values [61] and that of a definite Hermitian pair from Stewart [82].

Definition 6.3.1 *The field of values of $A \in \mathbb{C}^{n\times n}$ is the set of all Rayleigh quotients:*

$$F(A) \equiv \left\{ \frac{z^*Az}{z^*z} : 0 \neq z \in \mathbb{C}^n \right\},$$

where z^* denotes the conjugate transpose of z. □

The set $F(A)$ is compact and convex [61, Thm. 1.4.2], and when A is normal ($A^*A = AA^*$) it is the convex hull of the eigenvalues. For a Hermitian matrix $F(A)$ is a segment of the real axis and for a skew-Hermitian matrix it is a segment of the imaginary axis. The following properties of $F(A)$ are easily shown.

1. $F(\alpha A + \beta I) = \alpha F(A) + \beta, \quad \alpha, \beta \in \mathbb{C}$;
2. $F(A + B) \subseteq F(A) + F(B)$.

Definition 6.3.2 (Stewart) *The Hermitian pair (A, B) is a definite pair if*

$$\gamma(A, B) := \min |F(A + iB)| \equiv \min_{\|z\|_2=1} \sqrt{(z^*Az)^2 + (z^*Bz)^2} > 0.$$

where i is the imaginary unit. □

The scalar $\gamma(A, B)$ is called the *Crawford number* of the matrix pencil $A - \lambda B$ and its association with definite pairs was first noted by Crawford [22]. It is worth noting that Crawford's original definition for real symmetric matrices is not equivalent to the definite Hermitian pair defined in Definition 6.3.2 when A, B are symmetric. His definition restricted z to be a real vector and is only valid when $n \neq 2$. The following matrix pair shows the necessity of the restriction $n \neq 2$ in Crawford's definition.

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Any $x \in \mathbb{R}^2$ with $\|x\|_2 = 1$ has the form $x = [\cos\theta \quad \sin\theta]^T$, and we have

$$\sqrt{(x^*Ax)^2 + (x^*Bx)^2} = \cos^2 2\theta + \sin^2 2\theta = 1 \quad \text{for all } \theta,$$

while $\gamma(A, B) = \min |F(A + iB)| = 0$ with the minimum attained for $z = [1 \;\; i]^T$. Stewart [82] extends Crawford's definition for the complex case and removes the restriction $n \neq 2$.

From Definition 6.3.2 we state the following lemma which give an alternative characterization of a definite pair.

Lemma 6.3.3 *(A, B) is a definite Hermitian pair if and only if $0 \notin F(A + iB)$.* □

One direct consequence is the following result in which $\text{null}(A)$ denotes the nullspace of A.

Corollary 6.3.4 *If (A, B) is a definite Hermitian pair then* $\text{null}(A) \cap \text{null}(B) = \{0\}$. □

Note that the converse of Corollary 6.3.4 is not true as the following example shows. Let

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

We have $\text{null}(A) = \text{null}(B) = \{0\}$. However $\gamma(A, B) = 0$ for $z = [1 \;\; 1]^T$.

For definite pairs, we can always assume that B is positive definite, as the next result from Stewart [82] shows; a definite Hermitian pair can be transformed by rotation into a pair in which B is positive definite.

Theorem 6.3.5 (Stewart) *Let (A, B) be a definite Hermitian pair, and for $\theta \in \mathbb{R}$ let*

$$A_\theta = A\cos\theta - B\sin\theta,$$
$$B_\theta = A\sin\theta + B\cos\theta.$$

Then there is a $\theta \in [0, 2\pi)$ such that B_θ is positive definite and

$$\gamma(A, B) = \lambda_{\min}(B_\theta),$$

where $\lambda_{\min}(B_\theta)$ is the smallest eigenvalue of B_θ.

Proof: Let the minimum of $|F(A + iB)|$ be attained at the point $h = re^{i\phi}$, that is,

$$h = re^{i\phi} = \min_z z^*(A + iB)z = z_0^*(A + iB)z_0,$$

where $z \in \mathbb{C}^n$, $\|z\|_2 = 1$ and $r > 0$. We know $0 \notin F(A + iB)$ from Lemma 6.3.3. Together with the convexity of the field of values, this implies that $F(A + iB)$ is contained in the half plane H whose boundary passes perpendicularly through h.

Let h_θ and H_θ be the corresponding quantities for the pair (A_θ, B_θ). Since $A_\theta + iB_\theta = e^{i\theta}(A + iB)$, these quantities are just the original quantities rotated through the angle θ. Choose θ so that H_θ lies in the upper half plane and h_θ lies along the imaginary axis, that is, choose θ so that $\theta + \phi = \frac{\pi}{2}$. We have

$$h_\theta = ir = re^{i(\theta+\phi)} = e^{i\theta}z_0^*(A + iB)z_0 = z_0^*(A_\theta + iB_\theta)z_0 = z_0^*A_\theta z_0 + iz_0^*B_\theta z_0.$$

Since A_θ, B_θ are Hermitian and no point lies below H_θ, we have

$$z_0^*A_\theta z_0 = 0, \qquad 0 < r = z_0^*B_\theta z_0 = \min_z |z^*B_\theta z| = \lambda_{\min}(B_\theta),$$

which proves that B_θ is positive definite. $\square$

Crawford and Moon [23], [24] present a bisection-like algorithm for computing θ such that B_θ is positive definite, for a definite pair (A, B). The main computational cost of their algorithm is a Cholesky factorization in each step to test the definiteness of B_θ for the current estimate of θ. Their algorithm can take $O(n)$ steps and therefore can require $O(n^4)$ flops.

When B_θ is positive definite, it is easily shown that A_θ and B_θ can be simultaneously diagonalized [82], [84]. Note that the following theorem can be generalized to the case when B_θ is positive semidefinite; see [41, Thm. 8.7.1].

Theorem 6.3.6 *Given a Hermitian pair (A_θ, B_θ) with B_θ positive definite, there exists a nonsingular matrix X such that $X^*A_\theta X = \Lambda_\theta$ and $X^*B_\theta X = I$, where Λ_θ is real and diagonal.*

Proof: Since B_θ is positive definite, it has a positive definite square roots $B_\theta^{1/2}$. Then the pair (A_θ, B_θ) shares the same set of eigenvalues to the pair $(B_\theta^{-1/2}A_\theta B_\theta^{-1/2}, I)$. Let $B_\theta^{-1/2}A_\theta B_\theta^{-1/2} = Q\Lambda_\theta Q^*$ be the spectral decomposition of $B_\theta^{-1/2}A_\theta B_\theta^{-1/2}$. Then $X = B_\theta^{-1/2}Q$ is easily seen to be the required matrix. $\square$

Using Theorems 6.3.5 and 6.3.6, it is easily shown from $A + iB = e^{-i\theta}(A_\theta + iB_\theta)$ that every definite pair (A, B) is simultaneously diagonalizable. Indeed, the eigenvalues λ of (A, B) and λ_θ of (A_θ, B_θ) are related by $\lambda = e^{-i\theta}\lambda_\theta$. In other words, every generalized Hermitian eigenvalue problem for a definite pair can be reduced to a standard Hermitian eigenvalue problem.

The simultaneous diagonalization approach is well known [41], [71], [84]. In practice, instead of computing the positive square root $B_\theta^{1/2}$, we compute $B_\theta = LL^*$ the Cholesky decomposition where L is lower triangular. Thus we have

$$Cy = \lambda y \quad \text{where } C = L^{-1}A_\theta L^{-*}. \tag{6.7}$$

We now investigate the simultaneous diagonalization approach.

6.4 Simultaneous Diagonalization

Throughout this section, B is assumed to be positive definite.

The simultaneous diagonalization approach (6.7) takes full advantage of the Hermitian structure of A and B, and reduces a generalized eigenvalue problem to a standard Hermitian eigenvalue problem. However it has two disadvantages. First, C is generally full even when A and B are sparse. In the case when A and B are banded matrices, Crawford [21] shows how to implement the simultaneous diagonalization efficiently. We describe Crawford's algorithm in Section 6.4.1.

The second disadvantage of this approach is that when B, and hence L, is ill conditioned, the matrix C may have very large entries and the eigenvalues of C can be severely contaminated with roundoff error, as the following example from [41] shows. If

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}, \qquad L = \begin{bmatrix} .001 & 0 & 0 \\ 1 & .001 & 0 \\ 2 & 1 & .001 \end{bmatrix}$$

and $B = LL^T$, then the two smallest eigenvalues of $A - \lambda B$ are

$$\lambda_1 = -0.619402940600584, \qquad \lambda_2 = 1.627440079051887,$$

with the condition numbers $\kappa_2(\lambda_1)$, $\kappa_2(\lambda_2)$ equal to 30.64 and 89.00 respectively. Both eigenvalues λ_1, λ_2 are well conditioned. Here we use the condition number

$$\kappa_2(\lambda) := \limsup_{\epsilon \to 0} \left\{ \frac{|\delta\lambda|}{\epsilon|\lambda|} : (A + \Delta A)(x + \Delta x) = (\lambda + \delta\lambda)(B + \Delta B)(x + \Delta x), \right.$$
$$\left. \|\Delta A\|_2 \le \epsilon \|A\|_2, \ \ \|\Delta B\|_2 \le \epsilon \|B\|_2 \right\},$$

which, for a definite pair is given by

$$\kappa_2(\lambda) = \frac{\|x\|_2^2(\|A\|_2 + |\lambda|\|B\|_2)}{|\lambda||x^*Bx|};$$

see Higham and Higham [50].

However, using `eig(L\(A/L'))` in MATLAB where the unit roundoff $u \approx 1.1 \times 10^{-16}$, we have

$$\widehat{\lambda}_1 = -0.619\underline{330319419197}, \qquad \widehat{\lambda}_2 = 1.627\underline{630594726815},$$

where the incorrect digits are in italics and underlined. The reason for obtaining only four significant digits is that $\kappa_2(B) = \|B^{-1}\|_2\|B\|_2 \approx 10^{18}$.

6.4.1 When A and B are Banded

When $A, B \in \mathbb{R}^{n\times n}$ are banded with bandwidth m for some $m \ll n$, Crawford [21] shows how to form this reduced eigenvalue problem efficiently. Recall a symmetric matrix A is said to have bandwidth m if $a_{ij} = 0$ for $|i-j| > m$.

With the assumption that B is positive definite, the Cholesky decomposition $B = LL^T$ exists and we have $C = L^{-1}AL^{-T}$. If $q = n/m$ is an integer, then we can write

$$A = \begin{bmatrix} H_1 & B_1 & & & \\ B_1^T & H_2 & B_2 & & \\ & B_2^T & \ddots & \ddots & \\ & & \ddots & \ddots & B_{q-1} \\ & & & B_{q-1}^T & H_q \end{bmatrix}, \qquad L = \begin{bmatrix} D_1 & & & & \\ M_1^T & D_2 & & & \\ & M_2^T & \ddots & & \\ & & \ddots & \ddots & \\ & & & M_{q-1}^T & D_q \end{bmatrix},$$

where A is block tridiagonal and L is block bidiagonal with all the blocks $m \times m$. Here B_i, D_i, M_i are lower triangular and H_i is symmetric. Furthermore L can be factored as follows:

$$L = L_1 L_2 \ldots L_q,$$

where $L_k = \operatorname{diag}(I_{(k-1)m}, \widetilde{L}_k, I_{(q-k-1)m})$ and

$$\widetilde{L}_k = \begin{bmatrix} D_k & \\ M_k^T & I_m \end{bmatrix} \text{ for } k = 1, \ldots, q-1, \qquad \widetilde{L}_q = D_q.$$

Note that $\widetilde{L}_k^{-1}$, and hence L_k^{-1}, is easily computed. Let $A_1 = A$ and

$$A_{k+1} = L_k^{-1} A_k L_k^{-T} \qquad \text{for } k = 1, \ldots, q.$$

We have $C = A_{q+1}$. Crawford's idea is to restore the bandwidth of A_{k+1} at each stage [21]. Let $\widetilde{A}_1 = A$ and for $k = 1, \ldots, q$,

$$\widetilde{A}_{k+1} = Q_k L_k^{-1} \widetilde{A}_k L_k^{-T} Q_k^T \tag{6.8}$$

where

$$Q_k = \mathrm{diag}(\widetilde{Q}_k, I_{(q-k)m}), \tag{6.9}$$

are orthogonal matrices chosen so that $\widetilde{A}_{k+1}$ has bandwidth m. Assuming that $\widetilde{Q}_k$ can be found, it is easily shown that for $p > k$,

$$Q_k L_p^{-1} = L_p^{-1} Q_k.$$

Thus

$$\begin{aligned}
\widetilde{C} = \widetilde{A}_{q+1} &= Q_q L_q^{-1} A_q L_q^{-T} Q_q^T \\
&= Q_q L_q^{-1} Q_{q-1} L_{q-1}^{-1} A_{q-1} L_{q-1}^{-T} Q_{q-1}^T L_q^{-T} Q_q^T \\
&= Q_q Q_{q-1} L_q^{-1} L_{q-1}^{-1} A_{q-1} L_{q-1}^{-T} L_q^{-T} Q_{q-1}^T Q_q^T \\
&= Q_q Q_{q-1} \ldots Q_1 C Q_1^T \ldots Q_{q-1}^T Q_q^T.
\end{aligned}$$

This shows that C and $\widetilde{C}$ have the same eigenvalues. It remains to show that Q_k exists. This is done by induction. For $k = 1$, consider

$$\widetilde{A}_2 = L_1^{-1} A L_1^{-T} = \begin{bmatrix} \widehat{H}_1^{(1)} & \widehat{B}_1^{(1)} & \\ \widehat{B}_1^{(1)T} & H_2^{(2)} & \ddots \\ & \ddots & \ddots \end{bmatrix}$$

where the superscript denotes the stage in the reduction and the hats indicate blocks changed by the congruence transformation. Note that $\widehat{B}_1^{(1)}$ may not be

lower triangular. We can restore the band structure by performing a QR factorization on $\widehat{B}_1^{(1)T}$, that is, choose $Q_1 = \mathrm{diag}(\widetilde{Q}_1, I_{(q-1)m})$ so that $\widetilde{Q}_1 \widehat{B}_1^{(1)} = \widehat{B}_1^{(2)}$ where $\widehat{B}_1^{(2)}$ is lower triangular. Hence $\widetilde{A}_2 = Q_1 L_1^{-1} A L_1^{-T} Q_1^T$ has bandwidth m.

Assuming that (6.8) is true for k, consider

$$\widetilde{A}_{k+1} = L_k^{-1} \widetilde{A}_k L_k^{-T} = \begin{bmatrix} H_1^{(k)} & B_1^{(k)} & & & & & \\ B_1^{(k)T} & H_2^{(k)} & B_2^{(k)} & & & & \\ & B_2^{(k)T} & \ddots & \ddots & & & \\ & & \ddots & \ddots & \widehat{B}_{k-1}^{(k)} & \widehat{F}_k^{(k)} & \\ & & & \widehat{B}_{k-1}^{(k)T} & \widehat{H}_k^{(k)} & \widehat{B}_k^{(k)} & \\ & & & \widehat{F}_k^{(k)T} & \widehat{B}_k^{(k)T} & H_k^{(k+1)} & \ddots \\ & & & & & \ddots & \ddots \end{bmatrix},$$

Now $\widehat{B}_k^{(k)}$ may not be lower triangular and $\widehat{F}_k^{(k)}$ is nonzero. These matrices are the fill-in outside the band structure. We can restore the band structure using a block version of an algorithm of Rutishauser [75] that reduces a pentadiagonal matrix to a tridiagonal one. Let $R_k^{(k)} = \mathrm{diag}(I_{(k-2)m}, X_k^{(k)}, I_{(q-k)m})$, where $X_k^{(k)}$ is chosen so that

$$X_k^{(k)} \begin{bmatrix} \widehat{F}_k^{(k)} \\ \widehat{B}_k^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ B_k^{(k+1)} \end{bmatrix},$$

and $B_k^{(k+1)}$ is lower triangular. That is, perform a QR factorization on the matrix

$$P_{2m} \begin{bmatrix} \widehat{F}_k^{(k)} \\ \widehat{B}_k^{(k)} \end{bmatrix} P_m,$$

where P_m is defined as in (6.4).

Now we have

$$R_k^{(k)} L_k^{-1} \widetilde{A}_k L_k^{-T} R_k^{(k)T} = \begin{bmatrix} H_1^{(k)} & B_1^{(k)} & & & & & \\ B_1^{(k)T} & H_2^{(k)} & B_2^{(k)} & & & & \\ & B_2^{(k)T} & \ddots & \ddots & \widehat{F}_{k-1}^{(k)} & & \\ & & \ddots & \widehat{H}_{k-1}^{(k)} & \widehat{B}_{k-1}^{(k)} & & \\ & & \widehat{F}_{k-1}^{(k)T} & \widehat{B}_{k-1}^{(k)T} & H_k^{(k+1)} & B_k^{(k+1)} & \\ & & & & B_k^{(k+1)T} & H_k^{(k+1)} & \ddots \\ & & & & & \ddots & \ddots \end{bmatrix}.$$

The nonzero block $\widehat{F}_k^{(k)}$ is shifted up one position along the diagonal. After k such transformations, we take

$$Q_k = R_1^{(k)} R_2^{(k)} \dots R_k^{(k)},$$

and

$$\widetilde{A}_{k+1} = Q_k L_k^{-1} \widetilde{A}_k L_k^{-T} Q_k^T$$

again has bandwidth m. Moreover, since each $R_k^{(p)}$ satisfies (6.9) for $p \leq k$, so does Q_k.

If initially n/m is not an integer, then A is augmented by zero rows and columns and B by rows and columns of the identity matrix so that their dimension is a multiple of m. Since we assume $m \ll n$, the extra computational cost is not significant.

However we note that the overall computational cost can be expensive. Apart from this banded case, as Golub and Van Loan [41] comment, "the simultaneous diagonalization approach is impractical for the large, sparse symmetric definite problem."

6.5 Nearest Definite Pair

We are interested in finding the nearest definite Hermitian pair to a given Hermitian pair (A, B). Throughout this section, we assume (A, B) is not definite, that is, $F(A + iB)$ contains the origin (see Lemma 6.3.3). We want to find

$$d(A, B, \delta) = \min\{\|[\Delta A \ \Delta B]\| : \gamma(A + \Delta A, B + \Delta B) = \delta\},$$

where δ is a suitable positive constant and $\gamma(\cdot\,, \cdot)$ is defined in Definition 6.3.2. Note that $\gamma(\cdot\,, \cdot)$ is invariant under rotation

$$(A, B) \to e^{-i\theta}(A, B) =: (A_\theta, B_\theta),$$

as the following lemma shows. The minus sign means that $F(A + iB)$ is rotated with an angle θ about the origin in the clockwise direction.

Lemma 6.5.1 *Let (A, B) be a Hermitian pair and for $\theta \in \mathbb{R}$ let*

$$A_\theta = A\cos\theta + B\sin\theta, \qquad B_\theta = -A\sin\theta + B\cos\theta. \tag{6.10}$$

Then $\gamma(A, B) = \gamma(A_\theta, B_\theta)$.

Proof: The proof is straightforward. By Definition 6.3.2, we have

$$\gamma(A_\theta, B_\theta) = \min|F(A_\theta + iB_\theta)| = \min|e^{-i\theta}F(A+iB)| = \min|F(A+iB)| = \gamma(A, B),$$

as required. □

It is natural to choose a norm that preserves this invariance. As the following result shows, all unitarily invariant norms have this property. Recall that for a unitarily invariant norm is one for which $\|QAU\| = \|A\|$ for all orthogonal Q, U.

Lemma 6.5.2 *Let (A, B) be a Hermitian pair, and let A_θ and B_θ be defined as in* (6.10)*. Then, for any unitarily invariant norm,*

$$\|[A \ B]\| = \|[A_\theta \ B_\theta]\|.$$

Proof: We can write

$$[A_\theta\ B_\theta] = [A\ B]\left(\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\otimes I\right) =: [A\ B]Q,$$

where $\otimes$ denotes the Kronecker product and Q is orthogonal. Then, for any unitarily invariant norm,

$$\|[A_\theta\ B_\theta]\| = \|[A\ B]Q\| = \|[A\ B]\|. \qquad \square$$

In particular, for the 2-norm and the Frobenius norm we have

$$\|[A\ B]\|_2^2 = \|A^2 + B^2\|_2, \qquad \|[A\ B]\|_F^2 = \|A\|_F^2 + \|B\|_F^2,$$

respectively. We derive optimal 2-norm perturbations and show how to determine the perturbations efficiently in Section 6.5.1. For the case where $A+iB$ is normal, we exploit the characteristics of the eigenvalues of the matrix pair and propose an alternative method to compute the perturbation in Section 6.5.2.

6.5.1 Optimal 2-norm Perturbations

We want to find

$$d_2(A, B, \delta) = \min\{\|[\Delta A\ \ \Delta B]\|_2 : \gamma(A + \Delta A, B + \Delta B) = \delta\}. \tag{6.11}$$

First we note that, in terms of the field of values, problem (6.11) is equivalent to finding ΔA and ΔB such that

$$D_\delta(0) \cap F(A + \Delta A + i(B + \Delta B)) \text{ at a single point,} \tag{6.12}$$

where $D_y(x)$ denotes a disc centred at x with radius y. In other words, we want $D_\delta(0)$ and $F(A + \Delta A + i(B + \Delta B))$ to intersect at their boundaries. Since both sets are convex and compact with $D_\delta(0)$ strictly convex, the intersection point is uniquely determined. Intuitively, one would like to associate the minimal

perturbation with the nearest distance from the origin to the boundary of $F(A + iB)$, which we call the *inner numerical radius*:

$$\zeta(A) = \min\{|w| : w \text{ is on the boundary of } F(A)\}.$$

This quantity is not to be confused with

$$r_{\min}(A) = \min\{|w| : w \in F(A)\},$$

which is indeed the Crawford number $\gamma(H, S)$ where $H = (A + A^*)/2$ and $S = (A - A^*)/2i$.

When the origin is not contained in the field of values, $\zeta(A) = r_{\min}(A)$. When the field of values does contain the origin, $r_{\min}(A) = 0$ while $\zeta(A)$ is the radius of the largest circle centred at the origin and contained within $F(A)$.

Write $C = A + iB$, we have, for any $z \in \mathbb{C}^n$,

$$w = z^*Cz = z^*Az + iz^*Bz,$$

where z^*Az and z^*Bz are real. It follows that for $w \in F(A + iB)$

$$\lambda_{\min}(A) \leq \mathrm{Re}(w) \leq \lambda_{\max}(A),$$

where $\lambda_{\min}$ and $\lambda_{\max}$ denote the minimum and maximum eigenvalues, respectively, of a Hermitian matrix. Thus the field of values lies within the vertical strip defined by the lines parallel to the imaginary axis that intersect the real axis at $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$. The bounds are attained when w is the Rayleigh quotient $z^*Cz/(z^*z)$ with z an eigenvector of A corresponding to $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$; note that this point lies on the boundary of $F(A)$. Now consider $C_\theta = e^{-i\theta}C = A_\theta + iB_\theta$. The field of values of C_θ is just that of C rotated clockwise through θ radians about the origin, so $\zeta(C_\theta) = \zeta(C)$. Applying the above argument to the rotated matrix C_θ we obtain

$$\lambda_{\min}(A_\theta) \leq \mathrm{Re}(e^{-i\theta}w) \leq \lambda_{\max}(A_\theta), \qquad w \in F(C), \tag{6.13}$$

where, again, both bounds are attained for a point on the boundary of $F(C)$.

Theorem 6.5.3 *The inner numerical radius for a Hermitian pair (A, B) satisfies*

$$\zeta(C) = |\min_{0\le\theta\le 2\pi} \lambda_{\max}(A_\theta)|, \tag{6.14}$$

where $C = A + iB$ and $C_\theta = e^{-i\theta}C = A_\theta + iB_\theta$. Let the minimum be attained at $\theta = \theta_*$. Then $0 \in F(C)$ if and only if $\lambda_{\max}(A_{\theta_*}) \ge 0$, and the point $\zeta(C)e^{i\phi}$ is on the boundary of $F(C)$ where

$$\phi = \begin{cases} \theta_*, & \text{if } 0 \in F(C), \\ \theta_* + \pi, & \text{if } 0 \notin F(C). \end{cases}$$

Proof: Consider, first, the case where $0 \in F(C)$. Then $0 \in F(C_\theta)$ for all θ, so $\lambda_{\max}(A_\theta) \ge 0$ for all θ, by (6.13). Since $F(C)$ is convex, every point w on the boundary of $F(C)$ having minimal modulus gives equality in the right-hand side inequality of (6.13), for some θ, and it follows that

$$\zeta(C) \ge |\min_{0\le\theta\le 2\pi} \lambda_{\max}(A_\theta)|. \tag{6.15}$$

If $\zeta(C)$ is attained at the point $re^{i\phi}$ on the boundary of $F(C)$, then equality is attained in (6.15) for $\theta = \phi$.

If $F(C)$ does not contain the origin then $\lambda_{\max}(A_\theta)$ takes both positive and negative values for $\theta \in [0, 2\pi]$. It is easily seen that if $\zeta(C)$ is attained at the point $re^{i\phi}$ on the boundary of $F(C)$ then (6.14) holds with the minimum being attained when $\theta = \phi - \pi$ and that $\lambda_{\max}(A_\theta)$ is negative. □

The problem (6.11) has an elegant solution, in terms of the inner numerical radius, with a positive parameter δ.

Theorem 6.5.4 *Let $A, B \in \mathbb{C}^{n\times n}$ be Hermitian, and let $C = A + iB$ and $A_\phi = A\cos\phi + B\sin\phi$. Let $\min_{0\le\phi\le 2\pi} \lambda_{\max}(A_\phi)$ be attained at the angle θ and let A_θ have the spectral decomposition*

$$A_\theta = Q\,\mathrm{diag}(\mu_i)Q^*, \qquad \mu_n \le \mu_{n-1} \le \cdots \le \mu_1.$$

If $0 \in F(C)$ *(or, equivalently,* $\mu_1 \geq 0$*) then*

$$d_2(A, B, \delta) = \delta + \mu_1 = \delta + \zeta(C).$$

If $0 \notin F(C)$ *(or, equivalently,* $\mu_1 < 0$*) then*

$$d_2(A, B, \delta) = \max(\delta + \mu_1, 0) = \max(\delta - \zeta(C), 0).$$

In both cases, two sets of optimal perturbations in (6.11) *are*

$$\begin{aligned} \Delta A &= \cos\theta \, Q \, \mathrm{diag}(\min(-\delta - \mu_i, 0)) Q^*, \\ \Delta B &= \sin\theta \, Q \, \mathrm{diag}(\min(-\delta - \mu_i, 0)) Q^* \end{aligned} \tag{6.16}$$

and

$$\Delta A = -d_2(A, B, \delta) \cos\theta \, I, \quad \Delta B = -d_2(A, B, \delta) \sin\theta \, I. \tag{6.17}$$

Proof: First, we consider the case $0 \in F(C)$. Write $\Delta C = \Delta A + i\Delta B$. Definition 6.3.2 shows that our task is to find perturbations ΔA and ΔB such that $\zeta(C + \Delta C) = r_{\min}(C + \Delta C) = \delta$ and $\|[\Delta A \;\; \Delta B]\|_2$ is minimized. If ΔC is an optimal perturbation then every point in the convex set $F(C + \Delta C)$ has modulus at least δ, with equality for at least one point, so there is a line p whose minimal distance to the origin is δ such that $F(C + \Delta C)$ lies entirely in the closed half plane H defined by p that excludes the origin. Let the line perpendicular to p passing through the origin intersect the boundary of $F(C)$ in the complement of H at $w = z^*Cz \;\; (z^*z = 1)$; if there are two such points, take the one of furthest from p. Then when C is perturbed to $C + \Delta C$ this point must move distance at least $|w| + \delta$. Hence

$$|z^* \Delta C z| = |z^*(C + \Delta C)z - z^*Cz| \geq |w| + \delta \geq \zeta(C) + \delta.$$

Now using a trick from [64],

$$\begin{aligned}
|z^*\Delta C z| &= |z^*\Delta A z + i z^*\Delta B z| \\
&= \left((z^*\Delta A z)^2 + (z^*\Delta B z)^2\right)^{1/2} \\
&= \max_\theta [z^*\Delta A z \;\; z^*\Delta B z] \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \\
&= \max_\theta z^* [\Delta A \;\; \Delta B] \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} z \\
&\le \max_\theta \left\| [\Delta A \;\; \Delta B] \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \right\|_2 \\
&\le \|[\Delta A \;\; \Delta B]\|_2 .
\end{aligned}$$

Hence

$$\|[\Delta A \;\; \Delta B]\|_2 \ge \zeta(C) + \delta. \tag{6.18}$$

With θ as specified in the statement of the theorem define $A_\theta + B_\theta = e^{-i\theta}(A+iB)$, so that $A_\theta = A\cos\theta + B\sin\theta$ and $\zeta(A_\theta + iB_\theta) = \zeta(A+iB)$. Note that $F(A_\theta + iB_\theta)$ is $F(A+iB)$ rotated θ radians clockwise about the origin. Applying Theorem 6.5.3 to C and recalling that $0 \in F(C)$, we find that $\zeta(A_\theta + iB_\theta)$ is attained at the point in the complex plane

$$(\mu_1, 0) = (q_1^* A_\theta q_1, q_1^* B_\theta q_1),$$

where $\mu_1 \ge 0$ and q_1 is the first column of Q. Let

$$\Delta A_\theta = Q\,\mathrm{diag}(\min(-\delta - \mu_i, 0))Q^*, \qquad \Delta B_\theta = 0.$$

Then all the eigenvalues of $A_\theta + \Delta A_\theta$ are less than or equal to $-\delta$ and

$$(q_1^*(A_\theta + \Delta A_\theta)q_1, q_1^*(B_\theta + \Delta B_\theta)q_1) = (-\delta, 0),$$

so it follows that

$$\gamma(A_\theta + \Delta A_\theta, B_\theta + \Delta B_\theta) = \delta.$$

Now define ΔA and ΔB by

$$A + \Delta A + i(B + \Delta B) = e^{i\theta}(A_\theta + \Delta A_\theta + i(B_\theta + \Delta B_\theta)).$$

Then, by Lemma 6.5.1, we have

$$\gamma(A + \Delta A, B + \Delta B) = \gamma(A_\theta + \Delta A_\theta, B_\theta + \Delta B_\theta) = \delta.$$

Using Lemma 6.5.2, it follows that

$$\|[\Delta A \;\; \Delta B]\|_2 = \|[\Delta A_\theta \;\; \Delta B_\theta]\|_2 = \delta + \mu_1 = \delta + \zeta(C).$$

Thus ΔA and ΔB are feasible perturbations that attain the lower bound in (6.18), and so are optimal. The perturbations (6.17) correspond to

$$\Delta A_\theta = Q\,\mathrm{diag}(-\delta - \mu_1)Q^* = -(\delta + \zeta(C))I, \qquad \Delta B_\theta = 0,$$

and are easily seen to provide another solution.

Now suppose that $0 \notin F(C)$. Note that only in this case can (A, B) already be a definite pair and hence $d_2(A, B, \delta)$ be zero. If $\zeta(C) \geq \delta$ then, trivially, $d_2(A, B, \delta) = 0$ and the distance and perturbations in the statement of the theorem are, correctly, all zero. Therefore we can assume that $\zeta(C) < \delta$. Define $A_\theta + iB_\theta$ as in the first part. Note that, by Theorem 6.5.3, $F(A_\theta + iB_\theta)$ lies in the open left half plane and $w = -\zeta(A_\theta + iB_\theta)$ is on the boundary of $F(A_\theta + iB_\theta)$. The perturbation ΔC must move w to the boundary or exterior of the circle centre at the origin with radius δ, therefore w must move a distance at least $\delta + \mu_1$. As in the first part, this leads to the bound $\|[\Delta A \;\; \Delta B]\|_2 \geq \delta + \mu_1$, and the rest of the proof is very similar to that of the first part. □

To illustrate Theorem 6.5.4, we compute the fields of values of two 5×5 random indefinite Hermitian pairs (A, B) using `fv.m` from the MATLAB Test Matrix

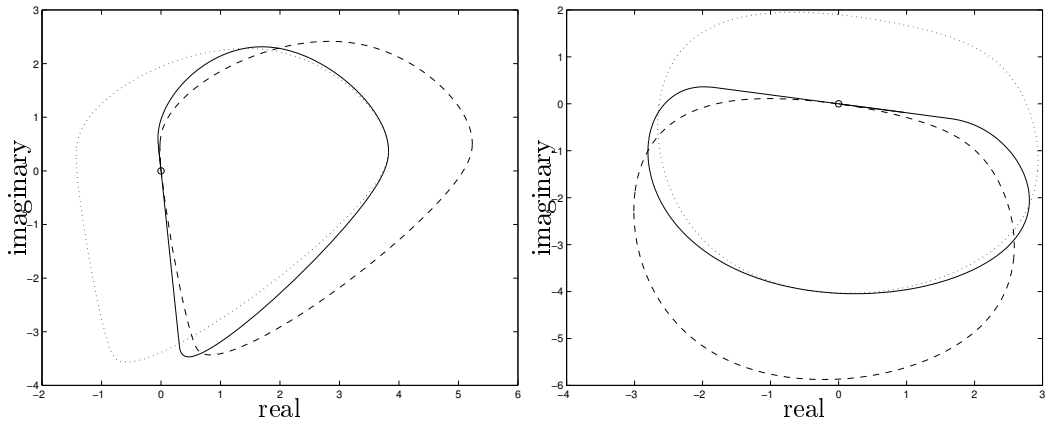

Figure 6.1: Change of boundaries of the field of values under perturbation (6.16), (6.17). Key: Original $\cdots$, (6.16) - - -, (6.17) —.

Toolbox [54] in which the eigensystems are computed by the QR algorithm. We set $\delta = u\|A+iB\|_\infty$ where $u \approx 1.1 \times 10^{-16}$ is the unit roundoff, and compute 200 points on the boundary of $F(A+iB)$. Figure 6.1 shows how the boundaries of the fields of values are perturbed, and the origin is excluded, under perturbations (6.16) and (6.17).

Theorem 6.5.3 parameterizes the problem of determining $\zeta(C)$ as a minimization problem of a function of θ. Let $\mu_{\max}(\theta)$ denote the maximum eigenvalue of $A\cos\theta + B\sin\theta$, and let the minimum in (6.14) be attained at the angle θ and $\widehat{\theta}$ denote its computed counterpart. Note that $\mu_{\max}(\theta)$ is a continuous function of θ. To determine $\widehat{\theta}$, a set of $\mu_{\max}(\theta_i)$ is computed, where $\theta_i = 2k\pi/m$ for $k = 1{:}m$. Then $\widehat{\theta} = \{\theta_i \text{ yielding } |\min_{\theta_i} \mu_{\max}(\theta_i)|\}$, and $\widehat{\theta} \to \theta$ as $m \to \infty$, using the continuity of $\mu_{\max}(\theta)$. That is, given a large enough set of $\mu_{\max}(\theta_i)$, a sufficiently good approximation to the optimal 2-norm perturbation is guaranteed.

For obtaining each $\mu_{\max}(\theta_i)$, we can use a Lanczos-based algorithm described in Braconnier and Higham [10], in which only matrix-vector products are computed. If A, B are large and sparse, so is $A\cos\theta + B\sin\theta$. Thus this algorithm is well suited for the purpose in this case. However, computing the whole boundary

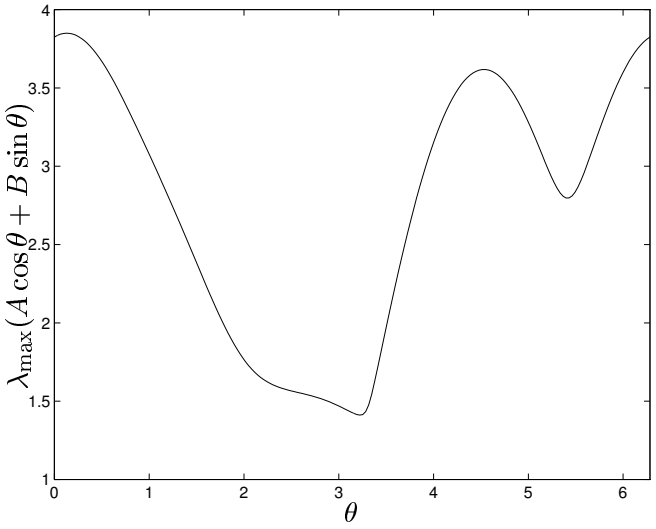

Figure 6.2: A typical graph $\lambda_{\max}(A_\theta)$ for an indefinite pair (A, B).

of $F(C)$ is expensive and most of the computed values are not needed subsequently. It is desirable to design algorithm to determine $\widehat{\theta}$ without computing the boundary of $F(C)$.

Of course, any standard minimization algorithm can be applied but may converge to a local minimum. Figure 6.2 shows a typical graph of the function $\lambda_{\max}(A\cos\theta + B\sin\theta)$ for an indefinite pair (A, B). Note that $\lambda_{\max}(A\cos\theta + B\sin\theta)$ is positive for all θ and has more than one local minimum. One useful approach is to compute a set of $\mu_{\max}(\theta_i)$ and then refine the best approximation to the inner numerical radius using an minimization algorithm. Further development of such algorithm is desirable but is beyond the scope of this thesis.

6.5.2 Normal Pairs

For the special case where $A + iB$ is normal, we apply a different approach that exploits the characteristics of the matrix pair. Much is known about the properties of normal matrices [45]. We shall call a matrix pair (A, B) for which $A + iB$ is normal a *normal pair*.

Lemma 6.5.5 *Let (A, B) be a normal pair. Then*

$$A + iB \text{ is normal} \iff A, B \text{ commute} \iff AB \text{ is Hermitian.}$$

Proof: We have

$$\begin{aligned} 0 &= (A+iB)^*(A+iB) - (A+iB)(A+iB)^* \\ &= (A-iB)(A+iB) - (A+iB)(A-iB) \\ &= (A^2 + iAB - iBA + B^2) - (A^2 - iAB + iBA + B^2) \\ &= 2i(AB - BA), \end{aligned}$$

which shows the first "$\iff$". Moreover, we have $AB - BA = AB - B^*A^* = AB - (AB)^*$, which completes the proof. □

In addition, since normal matrices are always diagonalizable [45], we have the following lemma.

Lemma 6.5.6 *If (A, B) is a normal pair then there exists a unitary Q such that both Q^*AQ and Q^*BQ are diagonal.*

Proof: Let $C = A+iB$ have the spectral decomposition $Q\Lambda Q^*$ where $QQ^* = I$ and Λ is diagonal. We have

$$\begin{aligned} A = (C + C^*)/2 &\implies Q^*AQ = (\Lambda + \Lambda^*)/2, \\ B = (C - C^*)/2i &\implies Q^*BQ = (\Lambda - \Lambda^*)/2i, \end{aligned}$$

as required. □

One immediate consequence is as follows.

Corollary 6.5.7 *Let (A, B) be a normal pair. Then z is an eigenvector of A if and only if z is an eigenvector of B.* □

Lemma 6.5.6 provides the link between the eigenvalues of a normal pair (A, B) and those of $A + iB$, as the following result shows.

Lemma 6.5.8 *Let (A, B) be a normal pair with eigenvalues λ_j for $j = 1{:}n$. Suppose $A + iB$ is nonsingular and has the spectral decomposition*

$$Q \operatorname{diag}(r_1 e^{i\theta_1}, \ldots, r_n e^{i\theta_n}) Q^*,$$

where $r_i > 0$. Then $\lambda_j = 1/\tan\theta_j$ for $j = 1{:}n$.

Proof: As in the proof of Lemma 6.5.6, we have $Q^*AQ = \Lambda_A$ and $Q^*BQ = \Lambda_B$ where $\Lambda_A = \operatorname{diag}(\xi_1, \ldots, \xi_n)$ and $\Lambda_B = \operatorname{diag}(\eta_1, \ldots, \eta_n)$. Thus

$$\operatorname{diag}(r_1 e^{i\theta_1}, \ldots, r_n e^{i\theta_n}) = Q^*(A + iB)Q = \Lambda_A + i\Lambda_B.$$

Then $A - \lambda B = Q(\Lambda_A - \lambda\Lambda_B)Q^*$ is singular, which implies

$$\lambda = \frac{\xi_j}{\eta_j} = \frac{r_j \cos\theta_j}{r_j \sin\theta_j} = \frac{1}{\tan\theta_j}.$$

for some j, as required. □

When (A, B) is a normal pair, the field of values of $A + iB$ is the convex hull of the eigenvalues. Let

$$A = B = \frac{e^{i\pi/4}}{\sqrt{2}} \operatorname{diag}(1, -1, i, -i). \tag{6.19}$$

From Figure 6.3, it is easily seen that $d_2(A, B, \delta) = \delta + h = \delta + 1/\sqrt{2}$. Note that the field of values is an union of a set of triangles and h is easily computed using a formula of Kahan [63], which is a numerically stable version of Heron's formula for computing the area of a triangle. Kahan's formula also serves as an example where computation with a guard digit is crucial to obtain a stable algorithm [39], [63].

Let a, b, c be the lengths of the sides of the triangle and arrange a, b, c so that $a \geq b \geq c$ (which is the case in Figure 6.3). Then the area A of the triangle is given by

$$A = \frac{1}{4}\sqrt{\big(a + (b + c)\big)\big(c - (a - b)\big)\big(c + (a - b)\big)\big(a + (b - c)\big)}.$$

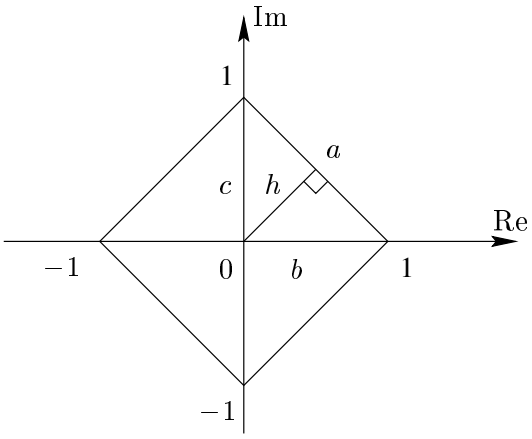

Figure 6.3: The field of values of normal pair (6.19)

.

Since A also equals to $\frac{1}{2}ah$, we have

$$h = \frac{1}{2a}\sqrt{\big(a+(b+c)\big)\big(c-(a-b)\big)\big(c+(a-b)\big)\big(a+(b-c)\big)}.$$

Note that the parentheses are essential [55]! Our algorithm for determining $d_2(A, B, \delta)$ is simple. First compute the complete set of eigenvalues λ_k of $A+iB \in \mathbb{C}^{n\times n}$, and arrange λ_k so that $0 \le \theta_p \le \theta_q < 2\pi$ for $p < q$, where θ_k denotes the angle between the positive real axis and the eigenvalue λ_k in the anti-clockwise direction. Then there exists a subsequence $\{\lambda_{k_p}\}$ comprising the extreme points of $F(A + iB)$. Calculate h_{k_p} for each triangle with vertices $(0, \lambda_{k_p}, \lambda_{k_{p+1}})$ with $\lambda_{k_{q+1}} = \lambda_{k_1}$. Then $d_2(A, B, \delta) = \delta + \min_{k_p} h_{k_p}$.

Note that if $\theta_n - \theta_1 < \pi$ and $\min_i |\lambda_i| > 0$ then (A, B) is a definite pair. If $\theta_n - \theta_1 < \pi$ and $\min_i |\lambda_i| = 0$, or $\theta_n - \theta_1 = \pi$, then the origin is on the boundary of the field of values $F(A + iB)$ and $d_2(A, B, \delta) = \delta$

6.6 Concluding Remarks

We have introduced the concept of the inner numerical radius. This important quantity is related to the distance from an indefinite matrix pair to the nearest definite pair, gives an alternative approach to determine whether a matrix is a

definite pair, and can be used to determine the angle for rotating the matrix pair so that one of them is positive definite. It is desirable to derive efficient algorithms for finding this quantity.

For a normal pair, we presented an alternative approach for determining optimal 2-norm perturbations using a formula of Kahan. This approach is numerically stable since the underlying formula is. An algorithm for determining the subsequence of extreme points on the boundary of the field of values is under development.

One open question is to generalize Theorem 6.5.4 for all unitarily invariant norms.

Bibliography

[1] Jan Ole Aasen. On the reduction of a symmetric matrix to tridiagonal form. *BIT*, 11:233–242, 1971.

[2] E. Anderson, Z. Bai, C. H. Bischof, J. W. Demmel, J. J. Dongarra, J. J. Du Croz, A. Greenbaum, A. McKenney, S. Ostrouchov, and D. C. Sorensen. *LAPACK Users' Guide, Release 2.0.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 1995.

[3] E. Anderson and J. Dongarra. Evaluating block algorithm variants in LAPACK. In J. Dongarra, P. Messina, D. Sorensen, and R. Voigt, editors, *Proceedings of the Fifth SIAM Conference on Parallel Processing for Scientific Computing*, pages 1–8, Philadelphia, PA, USA, 1990. Society for Industrial and Applied Mathematics.

[4] Peter Arbenz, Walter Gander, and Gene H. Golub. Restricted rank modification of the symmetric eigenvalue problem: Theoretical considerations. *Linear Algebra and Appl.*, 104:75–95, 1988.

[5] Peter Arbenz and Gene H. Golub. On the spectral decomposition of Hermitian matrices modified by low rank perturbations with applications. *SIAM J. Matrix Anal. Appl.*, 9(1):40–58, 1988.

[6] Cleve Ashcraft, Roger G. Grimes, and John G. Lewis. Accurate symmetric indefinite linear equation solvers. Manuscript, September 1995. To appear in SIAM J. Matrix Anal. Appl.

[7] Victor Barwell and Alan George. A comparison of algorithms for solving symmetric indefinite systems of linear equations. *ACM Trans. Math. Software*, 2(3):242–251, September 1976.

[8] Åke Björck. *Numerical Methods for Least Squares Problems.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996.

[9] A. J. Bosch. Note on the factorization of a square matrix into two Hermitian or symmetric matrices. *SIAM Review*, 29(3):463–468, September 1987.

[10] Thierry Braconnier and Nicholas J. Higham. Computing the field of values and pseudospectra using the Lanczos method with continuation. *BIT*, 36(3):422–440, 1996.

[11] James R. Bunch. Analysis of the diagonal pivoting method. *SIAM J. Numer. Anal.*, 8(4):656–680, 1971.

[12] James R. Bunch and Linda Kaufman. Some stable methods for calculating inertia and solving symmetric linear systems. *Math. Comput.*, 31:163–179, 1977.

[13] James R. Bunch, Christopher P. Nielsen, and Danny C. Sorensen. Rank-one modification of the symmetric eigenproblem. *Numer. Math.*, 31:31–48, 1978.

[14] James R. Bunch and Beresford N. Parlett. Direct methods for solving symmetric indefinite system of linear equations. *SIAM J. Numer. Anal.*, 8:639–655, 1971.

[15] David Carlson and Hans Schneider. Inertia theorems for matrices: The semidefinite case. *J. Math. Anal. and Appl.*, 6(3):430–446, 1963.

[16] S. Chandrasekaran, M. Gu, and A. H. Sayed. A stable and efficient algorithm for the indefinite linear least-square problem. To appear in SIAM J. Matrix Anal. Appl., 1997.

[17] Sheung Hun Cheng and Nicholas J. Higham. A modified Cholesky algorithm based on a symmetric indefinite factorization. Numerical Analysis Report

No. 289, Manchester Centre for Computational Mathematics, Manchester, England, April 1996. To appear in SIAM J. Matrix Anal. Appl.

[18] Sheung Hun Cheng and Nicholas J. Higham. Definite pairs and the inner numerical radius. Technical report in preparation, December 1997.

[19] Moody T. Chu and Quanlin Guo. On the least squares approximation of symmetric-definite pencils subject to generalized spectral constraints. *SIAM J. Matrix Anal. Appl.*, 19(1):1–20, January 1998.

[20] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A).* Springer Verlag, Berlin, Germany, 1992.

[21] C. R. Crawford. Reduction of a band-symmetric generalized eigenvalue problem. *Comm. ACM*, 16:41–44, 1973.

[22] C. R. Crawford. A stable generalized eigenvalue problem. *SIAM J. Numer. Anal.*, 13:854–860, 1976.

[23] C. R. Crawford. Algorithm 646 PDFIND: A routine to find a positive definite linear combination of two real symmetric matrices. *ACM Trans. Math. Software*, 12(3):278–282, September 1986.

[24] C. R. Crawford and Yiu Sang Moon. Finding a positive definite linear combination of two Hermitian matrices. *Linear Algebra and Appl.*, 51:37–48, 1983.

[25] J. J. M. Cuppen. A divide and conquer method for the symmetric tridiagonal eigenproblem. *Numer. Math.*, 36:177–195, 1981.

[26] James W. Demmel. *Applied Numerical Linear Algebra.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[27] James W. Demmel, Inderjit Dhillon, and Huan Ren. On the correctness of some bisection-like parallel eigenvalue algorithms in floating point arithmetic. *Electronic Transactions on Numerical Analysis*, 3:116–149, 1995.

[28] J. E. Dennis, Jr. and D. J. Woods. Optimization on microcomputers: The Nelder-Mead simplex algorithm. In A. Wouk, editor, *New Computing Environments: Microcomputers in Large-Scale Computing*, pages 116–122, Philadelphia, PA, USA, 1987. Society for Industrial and Applied Mathematics.

[29] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart. *LINPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1979.

[30] Bernd Fischer, Alison Ramage, David J. Silvester, and Andrew J. Wathen. Minimum residual methods for augmented systems. Mathematics Research Report 15, Department of Mathematics, University of Strathclyde, Glasgow, Scotland, June 1995.

[31] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, Chichester, England, 2nd edition, 1987.

[32] Anders Forsgren and Philip E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. Technical Report NA 96-3, Department of Mathematics, University of California, San Diego, USA, May 1996.

[33] Anders Forsgren, Philip E. Gill, and Joseph R. Shinnerl. Stability of symmetric ill-conditioned systems arising in interior methods for constrained optimization. *SIAM J. Matrix Anal. Appl.*, 17(1):187–211, 1996.

[34] Anders Forsgren and Walter Murray. Newton methods for large-scale linear equality constrained minimization. *SIAM J. Matrix Anal. Appl.*, 14(2):560–587, 1993.

[35] F. G. Frobenius. Ueber die mit einer Matrix vertauschbaren Matrizen. *Sitzungsberichte der Königlich Preusischen Akademie der Wissenschaften zu Berlin*, pages 3–15, 1910.

[36] Philip E. Gill and Walter Murray. Newton-type methods for unconstrained and linearly constrained optimization. *Math. Prog.*, 7:311–350, 1974.

[37] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization.* Academic Press, London, England, 1981.

[38] Philip E. Gill, Michael A. Saunders, and Joseph R. Shinnerl. On the stability of cholesky factorization for symmetric quasidefinite systems. *SIAM J. Matrix Anal. Appl.*, 17(1):35–46, 1996.

[39] David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1):5–48, 1991.

[40] Gene H. Golub. Some modified matrix eigenvalue problems. *SIAM Review*, 15(2):318–334, 1973.

[41] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, Maryland, USA, 2nd edition, 1989.

[42] Nicholas I. M. Gould. On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem. *Math. Prog.*, 32:90–99, 1985.

[43] Nicholas I. M. Gould. On the accurate determination of search directions for simple differentiable penalty functions. *IMA J. Numer. Anal.*, 6:357–372, 1986.

[44] Nicholas I. M. Gould. Constructing appropriate models for large-scale, linearly-constrained, nonconvex, nonlinear optimization algorithms. Report RAL-95-037, Rutherford Appleton Labortory, Didcot, Oxon, UK, August 1995.

[45] Robert Grone, Charles R. Johnson, Sá, and Henry Wolkowicz. Normal matrices. *Linear Algebra and Appl.*, 87:213–225, 1987.

[46] Ming Gu and Stanley C. Eisenstat. A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem. *SIAM J. Matrix Anal. Appl.*, 15(4):1266–1276, 1994.

[47] P. R. Halmos. Positive approximants of operators. *Indiana Univ. Math. J.*, 21:951–960, 1972.

[48] Richard J. Hanson. Aasen's method for linear systems with self-adjoint matrices. Visual Numerics, Inc., `http://www.vni.com/books/whitepapers/Aasen.html`, July 1997.

[49] Emilie V. Haynsworth and Alexander M. Ostrowski. On the inertia of some classes of partitioned matrices. *Linear Algebra and Appl.*, 1:299–316, 1968.

[50] Desmond J. Higham and Nicholas J. Higham. Structured backward error and condition of generalized eigenvalue problems. Numerical Analysis Report No. 297, Manchester Centre for Computational Mathematics, Manchester, England, November 1996.

[51] Nicholas J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and Appl.*, 103:103–118, 1988.

[52] Nicholas J. Higham. FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation (Algorithm 674). *ACM Trans. Math. Software*, 14(4):381–396, December 1988.

[53] Nicholas J. Higham. Optimization by direct search in matrix computations. *SIAM J. Matrix Anal. Appl.*, 14(4):317–333, April 1993.

[54] Nicholas J. Higham. The Test Matrix Toolbox for MATLAB (version 3.0). Numerical Analysis Report No. 276, Manchester Centre for Computational Mathematics, Manchester, England, September 1995.

[55] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996.

[56] Nicholas J. Higham. Recent developments in dense numerical linear algebra. In I. S. Duff and G. A. Watson, editors, *The State of the Art in Numerical Analysis*, volume 22, pages 1–26, Oxford, England, 1997. Oxford University Press.

[57] Nicholas J. Higham. Stability of Aasen's method. Manuscript, 1997.

[58] Nicholas J. Higham. Stability of the diagonal pivoting method with partial pivoting. *SIAM J. Matrix Anal. Appl.*, 18(1):52–65, January 1997.

[59] Nicholas J. Higham and Sheung Hun Cheng. Modifying the inertia of matrices arising in optimization. Numerical Analysis Report No. 295, Manchester Centre for Computational Mathematics, Manchester, England, September 1996. To appear in Linear Algebra and Appl.

[60] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, England, 1985.

[61] Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, England, 1991.

[62] Institute of Electrical and Electronics Engineers, New York. *IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Standard 754-1985*, 1985. Reprint in SIGPLAN Notices, 22(2):9–25, 1987.

[63] W. Kahan. How Cray's arithmetic hurts scientific computation (and what might be done about it). Manuscript prepared for the Cray User Group meeting in Toronto, June 1990.

[64] Ren-Cang Li. A perturbation bound for definite pencils. *Linear Algebra and Appl.*, 179:191–202, 1993.

[65] The MathWork, Inc., Natick, MA, USA. *MATLAB User's Guide*, 1992.

[66] W. Miller and C. Spooner. Software for roundoff analysis, II. *ACM Trans. Math. Software*, 4:369–387, 1978.

[67] Jorge J. Moré and Danny C. Sorensen. On the use of directions of negative curvature in a modified Newton's method. *Math. Prog.*, 16:1–20, 1979.

[68] W. Oettli and W. Prager. Compatibility of approximate solution of linear equations with given error bounds for coefficients. *Numer. Math.*, 6:405–409, 1964.

[69] A. M. Ostrowski. A quantitative formulation of Sylvester's law of inertia. *Proc. Nat. Acad. Sci. U.S.A.*, 45:740–744, 1959.

[70] B. N. Parlett and H. C. Chen. Use of indefinite pencils for computing damped natural modes. *Linear Algebra and Appl.*, 140:53–88, 1990.

[71] Beresford N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1980.

[72] Beresford N. Parlett. Symmetric matrix pencils. *J. Comp. and Appl. Math.*, 38:373–385, 1991.

[73] Alex Pothen. Predicting the structure of sparse orthogonal factors. *Linear Algebra and Appl.*, 192:183–203, 1993.

[74] J. D. Pryce. *Numerical Solution of Strum–Liouville Problems.* Oxford University Press, Oxford, England, 1993.

[75] H. Rutishauser. On Jacobi rotation patterns. In *Proc. of Symposia in Appl. Math.*, volume 15, pages 219–239, Providence, R.I., 1963. American Mathematical Society.

[76] Michael A. Saunders. Solution of sparse rectangular systems using LSQR and CRAIG. *BIT*, 35:588–604, 1995.

[77] Tamar Schlick. Modified Cholesky factorizations for sparse preconditioners. *SIAM J. Sci. Comput.*, 14:424–445, 1993.

[78] Robert B. Schnabel and Elizabeth Eskow. A new modified Cholesky factorization. *SIAM J. Sci. Stat. Comput.*, 11(6):1136–1158, 1990.

[79] David J. Silvester and Andrew J. Wathen. Fast and robust solvers for time-discretised incompressible Navier–Stokes equations. In D.F. Griffiths and G. A. Watson, editors, *Numerical Analysis 1995, Proceedings of the 16th Dundee Conference*, volume 344 of *Pitman Research Notes in Mathematics*, pages 154–168, Essex, England, 1996. Longman Scientific and Technical.

[80] Robert D. Skeel. Scaling for numerical stability in Gaussian elimination. *J. Assoc. Comput. Mach.*, 26(3):494–526, 1979.

[81] G. W. Stewart. *Introduction to Matrix Computations.* Academic Press, New York, 1973.

[82] G. W. Stewart. Perturbation bounds for the definite generalized eigenvalue problem. *Linear Algebra and Appl.*, 23:69–85, 1979.

[83] G. W. Stewart. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM J. Numer. Anal.*, 17:403–409, 1980.

[84] G. W. Stewart and Ji-guang Sun. *Matrix Perturbation Theory.* Academic Press, London, England, 1990.

[85] Olga Taussky. The role of symmetric matrices in the study of general matrices. *Linear Algebra and Appl.*, 5:147–154, 1972.

[86] V. J. Torczon. *Multi-directional search: A direct search algorithm for parallel machines.* PhD thesis, Rice University, Houston, Texas, USA, 1989.

[87] V. J. Torczon. On the convergence of the multidirectional search algorithm. *SIAM J. Optimization*, 1:123–145, 1991.

[88] H. P. M. v. Kempen. Variation of the eigenvalues of a special class of Hermitian matrices upon variation of some of its elements. *Linear Algebra and Appl.*, 3:263–273, 1970.

[89] Robert J. Vanderbei. Symmetric quasi-definite matrices. *SIAM J. Optimization*, 5(1):100–113, 1995.

[90] Visual Numerics, Inc., Houston, TX, USA. *IMSL Fortran 90 MP Library*, 1996. Part Number 3743, p. 9.

[91] J. H. Wilkinson. Global convergence of tridiagonal QR algorithm with origin shifts. *Linear Algebra and Appl.*, 1:409–420, 1968.

[92] Harald K. Wimmer. On Ostrowski's generalization of Sylvester's law of inertia. *Linear Algebra and Appl.*, 52/53:739–741, 1983.