Lecture notes on Numerical Analysis

Robert M. Gower

September 19, 2020

Abstract

Theses are my notes for my lectures for the MDI210 Optimization and Numerical Analysis course. Theses notes are a work in progress, and will probably contain several small mistakes (let me know?). If you are following my lectures you may find them useful to recall what we covered in class. Otherwise, I recommend you read the excellent book by Golub and Van Loan [1]. All topics covered in these notes and the lectures are covered in [1]. Furthermore these notes are mostly based on [1].

Contents

1 Introduction to Numerical Linear Algebra

1.1 Notation

Numerical linear algebra is a set of numerical problems at the heart of which lies a matrix

$$A = (a_{ij}) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{d1} & a_{d2} & a_{d3} & \dots & a_{dn} \end{bmatrix}.$$

Numerical linear algebra problems are in turn at the heart of most optimization and engineering problems. Thus their importance. We will learn to *decompose* a matrix into simpler matrices (triangular or diagonal) to describe a matrix through a set of fundamental vectors and numbers (eigenvalues and eigenvectors), and to see how sensitive a problem involving a matrix is or if it is well posed (Conditioning).

There exist several classes of matrices, of particular interest in this course are

- Normal matrices $AA^\top = A^\top A$
- Symmetric matrices $(a_{ij}) = A = A^\top = (a_{ji})$
- Orthogonal matrices $AA^\top = A^\top A = I$,

where $I = (\delta_{ij})$ denotes the identity matrix.

1.2 Norms

First we generalize the notation of distance by defining a norm

Definition 1 *Let E be a vector space defined over the reals $\mathbb{R}$. We say that the function $\|\cdot\| : x \in E \to R_+$ is a norm if it is*

Point separating: $\|x\| = 0 \Leftrightarrow x = 0, \forall x \in E.$

Subadditive: $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in E$

Homogeneous: $\|ax\| = |a|\|x\|, \forall x \in E, a \in \mathbb{R}.$

If a multiplication operator is defined between vectors (think matrices) then we say that the norm is submultiplicative if

Submultiplicative: $\|xy\| \leq \|x\|\|y\|, \forall x, y \in E.$

Exercise 2 *Show that $\|Vy\|_2 = \|y\|_2$ for every $y \in \mathbb{R}^n$ and orthogonal matrix $V \in \mathbb{R}^{n \times n}$.*

We can define an *induced* norm over matrices by using vector norms. Let $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}_+$ be a norm. Then we can extend the norm to operate over matrices by overloading its definition with

$$\|A\| \overset{\text{def}}{=} \sup_{x \in \mathbb{R}^n, x \neq 0} \frac{\|Ax\|}{\|x\|}.$$

Lemma 3 *All induced norms satisfy*

$$\|Ax\| \leq \|A\| \|x\|, \forall x \in \mathbb{R}^n,$$

and are submultiplicative. Furthermore the L2 induced norm satisfies

$$\|A\|_2 = \sigma_{\max}(A).$$

Proof: Homework (For the last part use the SVD decomposition).

1.3 Condition number and sensitivity

Now that we have established a notion of distance through norms, we turn our attention to answering how far can an approximate solution of a linear system be from the true solution. That is, consider the problem of determining $x \in \mathbb{R}^n$ such that

$$Ax = b,$$

where $b \in \mathbb{R}^n$. But imagine we have perturbed the vector b by adding on an error δb and end up solving

$$A(x + \delta x) = b + \delta b. \tag{1}$$

How big can $\|\delta x\|$ be? It turns out that δx can be very far from x and how far depends on the *condition number* of A.

For instance, consider the linear system given by

$$\begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix} \quad \text{with solution} \quad x = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Let us say we find a solution that is *close* in the sense that the right hand side is close, that is let $x' \in \mathbb{R}^4$ such that

$$\begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 32.1 \\ 22.9 \\ 33.1 \\ 30.9 \end{bmatrix} \tag{2}$$

Note that we have introduced an error on right hand side of the order of 1/300 relative to the original right hand side. Surprisingly, this small change on the right hand side introduces a significant error in the solution x' since, as we can check, the solution to (2) is given by

$$x' = \begin{bmatrix} 9.2 \\ -12.6 \\ 4.5 \\ -1.1 \end{bmatrix}$$

Which is almost times the magnitude of the original solution $x = (1\ 1\ 1\ 1)^\top$. This large error in the solution is due to the large *condition number* in A, as we explain next.

Turning back to (1) let us calculate this error algebraically. Since we have that $Ax = b$ we have from (1) that

$$A\delta x = \delta b.$$

Assuming A is invertible and left multiplying A^{-1} on both sides we get

$$\delta x = A^{-1}\delta b.$$

Taking norms we have

$$\|\delta x\| \leq \|A^{-1}\|\|\delta b\|.$$

Furthermore $\|b\| = \|Ax\| \leq \|A\|\|x\|$ and thus

$$\frac{1}{\|x\|} \leq \|A\|\frac{1}{\|b\|}.$$

Putting the two above equations together gives

$$\frac{\|\delta x\|}{\|x\|} \leq \underbrace{\|A^{-1}\|\|A\|}_{\overset{\text{def}}{=}\kappa(A)}\frac{\|\delta b\|}{\|b\|}.$$

This last quantity is what we define as the condition number, and it governs how much the relative error in x gets amplified.

2 Linear Systems

The work horse of numerical linear algebra is the solution of linear systems

$$Ax = b,$$

where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ are given, and $x \in \mathbb{R}^n$ is the unknown.

2.1 Triangular systems

There are two efficient algorithms for solving triangular linear systems. Either the forward substitution or backward substitution. For instance, to deduce the backwards substitution method, consider the upper triangular system given by

$$\sum_{j=i}^{n} a_{ij}x_j = b_i, \quad \text{for } i = 1, \ldots, n. \tag{3}$$

In particular for $i = n$ the above reduces to $a_{nn}x_n = b_n$. Assuming that $a_{nn} \neq 0$ (otherwise there is no solution) we have that $x_n = b_n/a_{nn}$. Now for the remaining components of x, for a given i, separating out the x_i term in (3) we have

$$\sum_{j=i+1}^{n} a_{ij}x_j + a_{ii}x_i = b_i. \tag{4}$$

Assuming $a_{ii} \neq 0$ and isolating x_i gives

$$x_i = \frac{b_i - \displaystyle\sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}. \tag{5}$$

This suggests an algorithm, by calculating first $x_n = \frac{b_n}{a_{nn}}$ then progressing backwards.

Algorithm 1 Backward substitution

```
for $k = n, \dots, 1$ do
    $x_i = \dfrac{b_i - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}.$
```

Exercise 4 *What can we do if we find $a_{ii} = 0$? What does it say about this triangular system if $a_{ii} = 0$?*

Exercise 5 *How many arithmetic operations does Algorithm 1 apply? In other words, how many scalar multiplications, subtractions, summations and divisions are applied by Algorithm 1? Yet in other words, what is the total complexity of this algorithm?*

Because of the ease in which we can solve triangular linear systems, it is convenient to decompose all linear systems into triangles. For instance, suppose we can find a decomposition of $A = LU$ where $L \in \mathbb{R}^{n\times n}$ is lower triangular and $U = \mathbb{R}^{n\times n}$ is upper triangular. We can then solve the linear system $Ax = b$ using two triangular solves. First we solve a lower triangular system

$$Ly = b \quad \Leftrightarrow L\underbrace{Ux}_{y} = b.$$

Then we solve the upper triangular system

$$Ux = y.$$

The resulting x is our desired solution to $Ax = b$.

2.2 Gaussian elimination and LU decomposition

An efficient direct method for decomposing a matrix into a lower and upper triangular form is Gaussian elimination. Despite its modern name, the method has been re-discovered several times (by Joseph Lagrange, Newton, Gauss) and dates back at least to ancient China over 2000 years ago (Jiuzhang Suanshu, Nine Chapters of the Mathematical Art). The main idea behind Gaussian elimination is that we will left multiply the system $Ax = b$ by and invertible matrix $P \in \mathbb{R}^{n\times n}$ so that the resulting linear system $PAx = Pb$ is easier to solve than the original. In particular, Gaussian elimination performs varies row transformations on A until we reach an upper triangular matrix. In particular, note that

$$\{x \,|\, PAx = Pb\} \quad = \quad \{x \,|\, Ax = b\}.$$

If we can construct P so that PA is triangular, then our work is done. We will do this iteratively. Let $A^0 = A$ and A^k denote the matrix after all elements $a^k_{ij} = 0$ for $1 \leq j \leq k$ and $i \geq j + 1$. To generate A^{k+1} from A^k we need to perform a *row operation.*

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \vdots & 0 & 0 \\ \vdots & & 1 & 0 & 0 & \vdots \\ \vdots & \vdots & -\,a^k_{k+1k}\big/\,a^k_{kk} & 1 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & -\,a^k_{nk}\big/\,a^k_{kk} & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} a^k_{11} & a^k_{12} & a^k_{13} & \dots & a^k_{1n} \\ 0 & \ddots & \vdots & \vdots & a^k_{2n} \\ \vdots & 0 & a^k_{kk} & \vdots & \vdots \\ \vdots & 0 & a^k_{k+1k} & \dots & a^k_{k+1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a^k_{nk} & \dots & a^k_{nn} \end{bmatrix} = \begin{bmatrix} a^k_{11} & a^k_{12} & a^k_{13} & \dots & a^k_{1(k+1)} & \dots & a^k_{1n} \\ 0 & \ddots & \vdots & \dots & a^k_{2(k+1)} & \dots & a^k_{2n} \\ \vdots & 0 & a^k_{kk} & \vdots & \vdots & \dots & \vdots \\ \vdots & 0 & 0 & \vdots & a^{k+1}_{(k+1)(k+1)} & \dots & a^{k+1}_{(k+1)n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \\ 0 & 0 & 0 & \dots & a^{k+1}_{n(k+1)} & \dots & a^{k+1}_{nn} \end{bmatrix}$$

This operation can be represented in a much more compact way using algebra. Let

$$P_k = I - v_k e_k^\top, \tag{6}$$

where $e_k = (0, \cdots, \underset{kth}{1}, 0, \cdots, 0) \in \mathbb{R}^n$ is the kth unit coordinate vector and $v_k = (0, \ldots, 0, \underset{(k+1)th}{\frac{a^k_{k+1k}}{a^k_{kk}}}, \ldots, \frac{a^k_{nk}}{a^k_{kk}})$. We refer to (6) as the kth row operation. With this notation we can write (2.2) as

$$P_k A^k = A^{k+1}.$$

Before moving on we need the following lemma.

Lemma 6 *Let P_k be the kth row operation. It follows*

1. $P_k^{-1} = I + v_k e_k^\top$.

2. $P_{k-1}^{-1} P_k^{-1} = I + v_k e_k^\top + v_{k-1} e_{k-1}^\top$

Proof:

1. By direct computation we have

$$(I + v_k e_k^\top)(I - v_k e_k^\top) = I + v_k e_k^\top - v_k e_k^\top - v_k e_k^\top v_k e_k^\top = I - v_k e_k^\top v_k e_k^\top.$$

Since the support of v_k does not intersect with the support of e_k we have that $e_k^\top v_k = 0$.

2. Again by computation

$$P_{k-1}^{-1}P_k^{-1} = (I + v_{k-1}e_{k-1}^\top)(I + v_k e_k^\top) = I + v_{k-1}e_{k-1}^\top + v_k e_k^\top + v_{k-1}(e_{k-1}^\top v_k)e_k^\top.$$

Once again we have an inner product $e_{k-1}^\top v_k$ between two vector with disjoint support, thus $e_{k-1}^\top v_k = 0$ and the result follows. ▮

Gaussian elimination applies n row operations until the matrix is upper triangular

$$P_n P_{n-1} \cdots P_1 A = U. \tag{7}$$

The cost of applying P_k is $(n-k-1)n$ consequently the cost of performing (7) is

$$\sum_{k=1}^{n}(n-k-1)n = O(n^3).$$

Since by Lemma 6 P_k is invertible for every k we have that the product of row operations in (7) is also invertible with

$$(P_n P_{n-1} \cdots P_1)^{-1} = P_1^{-1} \cdots P_{n-1}^{-1} P_n^{-1} \overset{\text{def}}{=} L. \tag{8}$$

Again by Lemma 6 and straight forward induction we have that the matrix L in (8) is lower triangular. Left multiplying (7) by L we have

$$A = LU. \tag{9}$$

This is known as the LU decomposition

2.3 Cholesky Decomposition

When A is a positive definite matrix, we can efficiently compute a triangular decomposition. We say a matrix is positive definite if it is symmetric and if

$$v^\top A v > 0, \quad \forall v \neq 0. \tag{10}$$

If A is positive definite then there exists a lower triangular matrix $B \in \mathbb{R}^{n\times n}$ such that

$$A = BB^\top = \begin{bmatrix} b_{11} & 0 & \dots & 0 \\ b_{21} & b_{22} & 0 & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} \begin{bmatrix} b_{11} & b_{21} & \dots & b_{n1} \\ 0 & b_{22} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_{nn} \end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix} - & b_{1:}^\top & - \\ - & b_{2:}^\top & - \\ & \vdots & \\ - & b_{n:}^\top & - \end{bmatrix} \begin{bmatrix} | & | & \dots & | \\ b_{1:} & b_{2:} & \dots & b_{n:} \\ | & | & \dots & | \end{bmatrix}$$

as we will show by construction. We can use the above to directly calculate the elements of B. For example, the first column on either side of the above equation is given by

$$a_{:1} = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} = b_{11} \begin{bmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{n1} \end{bmatrix} = b_{11} b_{:1}.$$

The first line of the above says that $b_{11}^2 = a_{11}$ thus $b_{11} = \sqrt{a_{11}}$. We can continue in a similar fashion to calculate the remaining elements of B, by observing that $a_{ij} = \langle b_i, b_j \rangle$ and consequently

$$a_{:j} = \sum_{i=1}^{n} \langle b_{j:}, b_{i:} \rangle \, e_i = \sum_{i=1}^{n} \sum_{k=1}^{j} b_{jk} b_{ik} e_i = \sum_{k=1}^{j} b_{jk} b_{:k}.$$

We can build a recurrence in k from the above by first separating out the jth term in the summation to give

$$b_{jj} b_{:j} = a_{:j} - \sum_{k=1}^{j-1} b_{jk} b_{:k} \overset{\text{def}}{=} v. \tag{11}$$

Now suppose we have already calculated the columns from the 1st to $(j-1)$th column of B. Using (11) we can then calculate the jth column by first noting that $b_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} b_{jk}^2} = \sqrt{v_j}$ then setting

$$b_{:j} = \frac{v}{\sqrt{v_j}} = \frac{a_{:j} - \sum_{k=1}^{j-1} b_{jk} b_{:k}}{\sqrt{b_{jj}}}.$$

This provides the following algorithm

Algorithm 2 (B) =Cholesky Decomposition(A)

```
for  $j = 1, \ldots, n$ do
    Calculate $v = a_{:j} - \sum_{k=1}^{j-1} b_{jk} b_{:k}$
    Set $b_{:j} = v / \sqrt{v_j}$
```

Exe: Show that the number of flops of the Cholesky algorithm is proportional to $O(n^3)$.
Sol: The summation is where most of the effort goes. Since there are k elements in $b_{:k}$ it costs k to add on $b_{jk} b_{:k}$.

$$\sum_{j=1}^{n} \sum_{k=1}^{j-1} k = \sum_{j=1}^{n} \frac{(j-1)j}{2} \leq \sum_{j=1}^{n} \frac{j^2}{2} \leq \frac{1}{2} \int_{x=0}^{n} x^2 dx = \left. \frac{x^3}{6} \right|_n - \left. \frac{x^3}{6} \right|_0 = \frac{n^3}{6}.$$

Using the Cholesky decomposition, we can uncover many properties of positive definite matrices.

Lemma 7 *Let A be a positive definite matrix. It follows that*

1. *The Cholesky decomposition $B^\top B = A$ always exists. We can prove this by construction. That is, using induction we can show that Algorithm 2 works. This boils down to showing that $v_j \neq 0$ does not occur.*

2. *$det(A) = (b_1 \cdots b_n)^2$. Indeed, using properties of the determinant we have that*

$$det(A) = det(B^\top B) = det(B^\top) det(B) = det(B)^2 = (b_{11} \cdots b_{nn})^2.$$

3 Eigenvalues

Eigenvalues are important. To get a feel for this importance, watch the video `https://www.youtube.com/watch?v=XggxeuFDaDU` on the collapse of Tacoma Narrows Bridge as it resonates in the wind. This resonance is related to the smallest eigenvalue of the structural equations.

We can calculate the eigenvalues of a matrix $A \in \mathbb{R}^{n \times n}$ by finding the roots of its *characteristic polynomial.* That is, let (λ, x) be an eigenpair of A thus

$$Ax = \lambda x \Leftrightarrow (A - \lambda I)x = 0.$$

Since $x \neq 0$ this shows that $A - \lambda I$ is not invertible and consequently

$$\det(A - \lambda I) = 0. \tag{12}$$

If we can find all the solutions of (12) in λ, then we will have all the eigenvalues. But solving (12) requires finding all the roots of the polynomial (12), and this is difficult. Indeed, according to the AbelRuffini theorem there is no general, explicit and exact algebraic formula for the roots of a polynomial with degree 5 or more. Thus we turn to approximate methods for finding eigenvalues.

3.1 Eigenvalue and the similarity transform

One of the problems with writing a matrix down as a square of numbers is that we must choose a coordinate basis to do so. The choice of this coordinate basis is somewhat arbitrary, and the most important properties of the matrix are independent of this choice. The eigenvalues and eigenvectors of a matrix give us some insight into these intrinsic properties of the matrix that are independent of the coordinate basis we used to represent the matrix.

Definition 8 *Let $A \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$ and $\lambda \in \mathbb{C}$. We say that x is an eigenvector and λ an eigenvalue of A if $x \neq 0$ and*

$$Ax \quad = \quad \lambda x.$$

We also refer to (x, λ) as an eigenpair of A. We say $\lambda(A) \subset \mathbb{C}$ is the spectrum of A if $\lambda(A)$ contains all the eigenvalues of A, that is

$$\lambda(A) \overset{def}{=} \{\lambda \,|\, \exists x \in \mathbb{R}^n \textit{ such that } x \neq 0,\ Ax = \lambda x\}.$$

We say that A is invertible if $0 \notin \lambda(A)$.

A matrix can be entirely described by its eigenvalues and eigenvectors. Accordingly, we say that two matrices are similar if they share the same spectrum. Or, said in another way:

Definition 9 *We say that $A \in \mathbb{R}^{n \times n}$ is similar to $B \in \mathbb{R}^{n \times n}$ if there exists $P \in \mathbb{R}^{n \times n}$ invertible such that*

$$A = P^{-1}BP.$$

We say that A is diagonalizable when A is similar to a diagonal matrix.

Similar matrices define the same linear operator upto coordinate changes defined by P. Consequently they also have the same spectrum.

Lemma 10 *If $A, B \in \mathbb{R}^{n\times n}$ are similar matrices then*

$$\lambda(A) = \lambda(B).$$

Proof: Consider $\lambda \in \lambda(A)$. Then there exists $x \in \mathbb{R}^n$ such that $Ax = \lambda x$. By the similarity of A and B we have that

$$P^{-1}BPx = \lambda x.$$

Left multiplying by P shows that $\lambda \in \lambda(B)$ with associated eigenvector Px. ▮

Lemma 11 *If $O \in \mathbb{R}^{n\times n}$ is an orthogonal matrix then every $\lambda \in \lambda(O)$ is such that $|\lambda| = 1$.*

Proof: Let (x, λ) be such that $Ox = \lambda x$. If follows that

$$\langle x, x\rangle = \left\langle x, O^\top Ox \right\rangle = \langle Ox, Ox\rangle = \|Ox\|_2^2 = |\lambda|^2 \langle x, x\rangle .$$

Dividing by $\langle x, x\rangle$ on both sides gives the result.

We know by Lemma 10 that we are allowed to transform A through similarity transforms using an invertible matrix P without changing the spectrum. The next most natural question is, what should P be so that it is easy to compute the spectrum of $B = PAP^{-1}$? If B is a diagonal matrix with $B = \mathrm{diag}\,(B_{11}, \ldots, B_{nn})$ then clearly $\lambda(B) = (B_{11}, \ldots, B_{nn})$. Thus we should try and transform A into a diagonal matrix using similarity transforms. Is this possible? It is for symmetric matrix matrices.

Theorem 12 (Spectral Theorem for symmetric matrices) *Symmetric matrices are diagonalizable. That is, let $A \in \mathbb{R}^{n\times n}$ with $A = A^\top$. Then there exists an orthogonal matrix $V \in \mathbb{R}^{n\times n}$ and $\Lambda = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^{n\times n}$ such that*

$$A = VDV^\top.$$

Proof: See Theorem 8.1.1 and proof in [1].

3.2 Jacobi method

Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix. We know from the Spectral Theorem that symmetric matrices are diagonalizable, that is, there exists an orthogonal matrix $V \in \mathbb{R}^{n\times n}$ and diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ such that $A = V\Lambda V^\top$. This also shows that A is similar to Λ and thus the eigenvalues of A are $\lambda_1, \ldots, \lambda_n$. Furthermore, by right multiplying by V it follows that

$$AV = V\Lambda,$$

consequently the columns of V are the eigenvectors of A. Thus if we could calculate the decomposition $A = V\Lambda V^\top$ we would know the eigenvalues and eigenvectors of A. The objective of this section is to detail the *Jacobi method* for calculating this decomposition.

The idea behind the Jacobi method is to iteratively minimize the off-diagonal elements of A until we have a diagonal matrix. That is, we want to minimize the offset

$$\operatorname{off}(A) = \sum_{i=1}^{n} \sum_{j \neq i} a_{ij}^2 = \|A\|_F^2 - \sum_{i=1}^{n} a_{ii}^2. \tag{13}$$

The method proceeds by scanning through the matrix A and finding the element that has the largest absolute value, that is $a_{pq} = \max_{1 \leq i < j \leq n} |a_{ij}|$, then replace this element by a zero by using similarity transformations. Our tool for eliminating large off diagonal elements is the Givens/Jacobi Transform defined by

$$J(p,q,\theta) = \begin{array}{c c} \begin{array}{ccccccc} & & p & & q & & \end{array} & \\ \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \dots & c & \dots & s & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & -s & \dots & c & \dots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix} & \begin{array}{c} \\ \\ p \\ \\ q \\ \\ \\ \end{array} \end{array}$$

Where $c = \cos(\theta)$ and $s = \sin(\theta)$. We can define this matrix more succinctly as

$$J(p,q,\theta) = I + (c-1)e_p e_p^\top + (c-1)e_q e_q^\top + s e_p e_q^\top - s e_q e_p^\top = I + \begin{bmatrix} e_p & e_q \end{bmatrix} \begin{bmatrix} c-1 & s \\ -s & c-1 \end{bmatrix} \begin{bmatrix} e_p^\top \\ e_q^\top \end{bmatrix}$$

By carefully choosing θ we can use the following transform

$$B = J(p,q,\theta) A J(p,q,\theta)^\top, \tag{14}$$

to eliminate a_{pq} (and a_{qp} because of symmetry). Note that B is a similar matrix to A. To see how to do this, by examining the pth and qth row and column of (14) we see that the following holds

$$\begin{bmatrix} b_{pp} & b_{pq} \\ b_{qp} & b_{qq} \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^\top \begin{bmatrix} a_{pp} & a_{pq} \\ a_{qp} & a_{qq} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}. \tag{15}$$

This in turn shows that

$$b_{pq} = cs(a_{pp} - a_{qq}) + (c^2 - s^2) a_{pq}.$$

Now we choose θ so that $b_{pq} = 0$. Setting the above to zero and dividing through by $c^2 a_{pq}$ we have

$$-t^2 + 2Kt + 1 = 0, \tag{16}$$

where $t = \tan(\theta) = c/s$ and $K = \frac{a_{pp} - a_{qq}}{2a_{pq}}$. The solutions to (16) are given by

$$t = K \pm \sqrt{K^2 + 1}.$$

In the standard Jacobi method we choose the smallest of the two above roots

$$t = \min\{K + \sqrt{K^2+1}, K - \sqrt{K^2+1}\}.$$

We can then recover c and s using that

$$c = \frac{1}{\sqrt{1+t^2}}, \quad s = ct.$$

This gives us the following method for calculating c and s in Algorithm 3.

Algorithm 3 (c, s) =Calculate Jacobi Transform(p, q, A)

```
$K = \frac{a_{pp} - a_{qq}}{2a_{pq}}$
$t = \min\{K + \sqrt{K^2+1}, K - \sqrt{K^2+1}\}$.
$c = \frac{1}{\sqrt{1+t^2}}$
$s = ct$
```

With the means to calculate a single Jacobi transform, we can now iteratively apply many transforms to try to minimize the off diagonal elements of A, see Algorithm 4. Next we prove that Algorithm 4 converges and does what we intended it to do.

Algorithm 4 (c, s) =Calculate Jacobi Transform$((p, q, A))$

```
Initialize: $k = 0$ and $A^0 = A$.
while off$(A^{k+1}) < \epsilon$ do
    Choose $(p, q)$ so that $a_{pq} = \max_{i \neq j} |a_{pq}|$
    $(c, s)$ =Calculate Jacobi Transform$((p, q, A^k))$
    $A^{k+1} = J(p, q, \theta)^\top A^k J(p, q, \theta)$.
```

3.3 Convergence of Jacobi

We now need the following lemma, which will be given as an exercise in class

Lemma 13 *1. Let*

$$J = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}.$$

Show that $J^\top J = JJ^\top = I$, that is, J is an orthogonal matrix.

2. Prove that $Tr(AB) = Tr(BA)$ for compatible matrices.

3. Let $\|A\|_F^2 = Tr\left(A^\top A\right)$ and let J be an orthogonal matrix. Prove that $\|J^\top AJ\|_F^2 = \|A\|_F^2$.

4. Consider (14) and show that $b_{ii} = a_{ii}$ for $i = \{1, \ldots, n\} \setminus \{p, q\}$.

5. Show that $J(p, q, \theta)$ is an orthogonal matrix.

Proof:

1. Direct computation.

2. We have that

$$\operatorname{Tr}(AB) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ji} = \sum_{j=1}^{n}\sum_{i=1}^{n} b_{ji}a_{ij} = \operatorname{Tr}(BA).$$

3. From the previous property it follows that

$$\|J^\top AJ\|_F^2 = \operatorname{Tr}\left(J^\top A^\top JJ^\top AJ\right) = \operatorname{Tr}\left(J^\top A^\top AJ\right) = \operatorname{Tr}\left(A^\top AJJ^\top\right) = \operatorname{Tr}\left(A^\top A\right) = \|A\|_F^2.$$

4. Simply note that for $i \neq p, q$ we have that $e_i^\top J(p,q,\theta) = e_i^\top$.

5. Computing

$$\begin{aligned}
J(p,q,\theta)^\top J(p,q,\theta) &= \left(I + \begin{bmatrix} e_p & e_q \end{bmatrix}\begin{bmatrix} c-1 & -s \\ s & c-1 \end{bmatrix}\begin{bmatrix} e_p^\top \\ e_q^\top \end{bmatrix}\right)\left(I + \begin{bmatrix} e_p & e_q \end{bmatrix}\begin{bmatrix} c-1 & s \\ -s & c-1 \end{bmatrix}\begin{bmatrix} e_p^\top \\ e_q^\top \end{bmatrix}\right) \\
&= I + \begin{bmatrix} e_p & e_q \end{bmatrix}\begin{bmatrix} c-1 & -s \\ s & c-1 \end{bmatrix}\begin{bmatrix} c-1 & s \\ -s & c-1 \end{bmatrix}\begin{bmatrix} e_p^\top \\ e_q^\top \end{bmatrix} \\
&\quad + \begin{bmatrix} e_p & e_q \end{bmatrix}\begin{bmatrix} c-1 & -s \\ s & c-1 \end{bmatrix}\begin{bmatrix} e_p^\top \\ e_q^\top \end{bmatrix} + \begin{bmatrix} e_p & e_q \end{bmatrix}\begin{bmatrix} c-1 & s \\ -s & c-1 \end{bmatrix}\begin{bmatrix} e_p^\top \\ e_q^\top \end{bmatrix} \\
&= I + \begin{bmatrix} e_p & e_q \end{bmatrix}\begin{bmatrix} 2(1-c) & 0 \\ 0 & 2(1-c) \end{bmatrix}\begin{bmatrix} e_p^\top \\ e_q^\top \end{bmatrix} + \begin{bmatrix} e_p & e_q \end{bmatrix}\begin{bmatrix} 2(c-1) & 0 \\ 0 & 2(c-1) \end{bmatrix}\begin{bmatrix} e_p^\top \\ e_q^\top \end{bmatrix} \\
&= I. \qquad (17)
\end{aligned}$$

Using the previous lemma, given that $J(p,q,\theta)$ is an orthogonal matrix, we have from (15) that

$$\|A\|_F^2 = \|B\|_F^2.$$

What is more, the Frobenius norm of both sides of (15) are also the same thus

$$a_{pp}^2 + a_{qq}^2 + 2a_{pq}^2 = b_{pp}^2 + b_{qq}^2 + 2b_{pq}^2 = b_{pp}^2 + b_{qq}^2. \qquad (18)$$

Consequently since $b_{pq} = 0$ we have that

$$\begin{aligned}
\operatorname{off}(B) &= \|B\|_F^2 - \sum_{i=1}^{n} b_{ii}^2 \\
&= \|A\|_F^2 - \sum_{i=1, i\neq p,q}^{n} b_{ii}^2 - b_{pp}^2 - b_{qq}^2 \\
&= \|A\|_F^2 - \sum_{i=1, i\neq p,q}^{n} a_{ii}^2 - b_{pp}^2 - b_{qq}^2 \\
&= \|A\|_F^2 - \sum_{i=1}^{n} a_{ii}^2 + a_{pp}^2 + a_{qq}^2 - b_{pp}^2 - b_{qq}^2 \\
&\overset{(18)}{=} \operatorname{off}(A) - 2a_{pq}^2.
\end{aligned}$$

This shows that the off diagonal terms are decreasing. Furthermore, since a_{pq} is chosen as the largest off diagonal term in absolute value, we have that

$$a_{pq}^2 \geq \frac{\text{off}(A)}{n(n-1)}.$$

Thus finally

$$\text{off}(B) \leq \text{off}(A) - \frac{2}{n(n-1)}\text{off}(A) = \left(1 - \frac{2}{n(n-1)}\right)\text{off}(A).$$

That is, applying k steps of Algorithm 4 we have that

$$\text{off}(A^k) \leq \left(1 - \frac{2}{n(n-1)}\right)^k \text{off}(A).$$

4 The SVD decomposition

Symmetric matrices have a delightfully simple spectral theory. The same does not hold for asymmetric matrices. Though we can borrow the spectra theory of symmetric matrices to give insight into any asymmetric matrix through their singular values.

Definition 14 *We refer to $\sigma(A) \overset{def}{=} \lambda(A^\top A)$ as the set of singular values of A.*

Exercise 15 *Show that $A^\top A$ is similar to $AA^\top$.*

Proof: First we show that $\lambda(A^\top A) \subset \lambda(AA^\top)$. Let $\lambda \in \lambda(A^\top A)$ thus there exists $x \neq 0 \in \mathbb{R}^n$ such that

$$A^\top Ax = \lambda x.$$

Left multiplying by A gives

$$AA^\top(Ax) = \lambda(Ax).$$

If $Ax = 0$, then from the two preceding equalities we have that $0 \in \lambda(A^\top A)$ and $0 \in \lambda(AA^\top)$. If $Ax \neq 0$, then Ax is an eigenvector of $AA^\top$ with associated eigenvalue λ thus $\lambda \in \lambda(AA^\top)$. The opposite inclusion $\lambda(AA^\top) \subset \lambda(A^\top A)$ can be derived verbatim by re-labeling $\bar{A} = A^\top$ and $\bar{A}^\top = A$.

Though $AA^\top$ and $A^\top A$ have the same eigenvalues, they may have different eigenvectors. We will use their eigenvectors and common eigenvalues to construct a decomposition for A

Theorem 16 *Let $A \in \mathbb{R}^{m\times n}$, $\Sigma = diag(\sigma_1, \ldots, \sigma_n)$ be the singular values of A. Then there exits orthogonal matrices $V \in \mathbb{R}^{n\times n}$ and $U \in \mathbb{R}^{m\times m}$ such that*

$$A = U\Sigma V^\top.$$

Proof: See Chapter 2.5 in [1].

Exe: Show that the columns of U and V are the eigenvectors of $AA^\top$ and $A^\top A$, respectively.