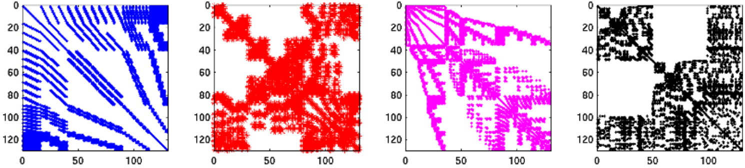


Chap. 3: Numerical Linear Algebra

§3.1 Introduction to solving linear systems

MA385 – Numerical Analysis 1

November 2019

This section of the course is concerned with solving systems of linear equations ("simultaneous equations"). All problems are of the form: *find the set of real numbers* x_1*,* x_2*, ...,* x_n *such that*

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

where the a_{ij} *and* b_i *are real numbers.*

It is natural to rephrase this using the language of linear algebra: *Find* $\boldsymbol{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ *such that*

$$A\boldsymbol{x} = \boldsymbol{b}, \qquad \text{where } A \in \mathbb{R}^{n\times n}, \text{ and } \boldsymbol{b} \in \mathbb{R}^n. \tag{1}$$

In this section, we'll try to find a clever way to solving this system. In particular,

1. we'll argue that its unnecessary and (more importantly) expensive to try to compute A^{-1};
2. we'll have a look at ***Gaussian Elimination***, but will study it in the context of LU-factorisation;
3. after a detour to talk about ***matrix norms***, we calculate the ***condition number*** of a matrix;
4. this last task will require us to be able to estimate the eigenvalues of matrices, so we will finish the module by studying how to do that.

- A vector $\boldsymbol{x} \in \mathbb{R}^n$, is an ordered n-tuple of real numbers, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$.
- A matrix $A \in \mathbb{R}^{m \times n}$ is a rectangular array of n columns of m real numbers. (But we will only deal with ***square*** matrices, i.e., ones where $m = n$).
- You already know how to multiply a matrix by a vector, and a matrix by a matrix (right??).
- You can write the **scalar** product as $(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y} = \sum_{i=1}^n x_i y_i$.
- Recall that $(A\boldsymbol{x})^T = \boldsymbol{x}^T A^T$, so $(\boldsymbol{x}, A\boldsymbol{y}) = \boldsymbol{x}^T A\boldsymbol{y} = (A^T\boldsymbol{x})^T \boldsymbol{y} = (A^T\boldsymbol{x}, \boldsymbol{y})$.
- Letting I be the identity matrix, if there is a matrix B such that $AB = BA = I$, when we call B the **inverse** of A and write $B = A^{-1}$. If there is no such matrix, we say that A is ***singular***.

Theorem 3.1

The following statements are equivalent.

1. For ***any*** $\boldsymbol{b}$, $A\boldsymbol{x} = \boldsymbol{b}$ has a solution.
2. If there is a solution to $A\boldsymbol{x} = \boldsymbol{b}$, it is unique.
3. If $A\boldsymbol{x} = 0$ then $\boldsymbol{x} = 0$.
4. The columns (or rows) of A are linearly independent.
5. There exists a matrix $A^{-1} \in \mathbb{R}^{n \times n}$ such that $AA^{-1} = I = A^{-1}A$.
6. $\det(A) \neq 0$.
7. A has rank n.

Item 5 of the above lists suggests that we could solve $A\boldsymbol{x} = \boldsymbol{b}$ as follows: find A^{-1} and then compute $\boldsymbol{x} = A^{-1}\boldsymbol{b}$. However this is not the best way to proceed. We only need x and computing A^{-1} requires quite a bit a work.

In introductory linear algebra courses, you might compute

$$A^{-1} = \frac{1}{\det(A)} A^*$$

where A^* is the adjoint matrix. But it turns out that computing $\det(A)$ directly can be ***very*** time-consuming.

Let d_n be the number of multiplications required to compute the determinant of an $n \times n$ matrix using the ***Method of Minors*** (also called "the Laplacian determinant expansion"). We know that if

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

then $\det(A) = ad - bc$. So $d_2 = 2$.

If

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & j \end{pmatrix}$$

then

$$\det(A) = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & j \end{vmatrix} = a \begin{vmatrix} e & f \\ h & j \end{vmatrix} - b \begin{vmatrix} d & f \\ g & j \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix},$$

so $d_3 \geq 3d_2$.

(See notes from class)

In general we find that it takes $(n)(n-1)\cdots(4)(3)(2)(1) = n!$ operations to compute the determinant this way. And that is a lot of operations.

Exercise 3.1

Suppose you had a computer that computer perform 2 billion operations per second. Give a lower bound for the amount of time required to compute the determinant of a 10-by-10, 20-by-20, and 30-by-30, matrix using the method of minors.

Exercise 3.2

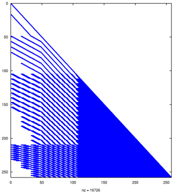
The Cauchy-Binet formula for determinants tells us that, if A and B are square matrices of the same size, then $\det(AB) = \det(A)\det(B)$. Using it, or otherwise, show that $\det(\sigma A) = \sigma^n \det(A)$ for any $A \in \mathbb{R}^{n\times n}$ and any scalar $\sigma \in \mathbb{R}$.
Note: this exercise gives us another reason to avoid trying to calculate the determinant of the coefficient matrix in order to find the inverse, and thus the solution to the problem. For example $A \in \mathbb{R}^{16\times 16}$ and the system is rescaled by $\sigma = 0.1$, then $\det(A)$ is rescaled by 10^{-16}. On standard computing platforms, like MATLAB, it would be indistinguishable from a singular matrix!

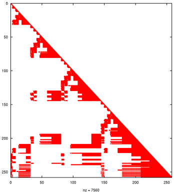
Chap. 3: Numerical Linear Algebra

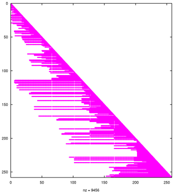
§3.2 Gaussian Elimination (and triangular matrices)

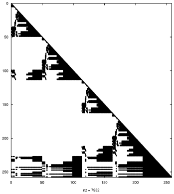
MA385/MA530 – Numerical Analysis 1

November 2019









Gaussian Elimination is an **exact** method for solving linear systems (we replace the problem with one that is easier to solve *and* has the same solution.)
This is in contrast to **approximate** methods studied earlier in the module.

There are approximate methods for solving linear systems, but they are not part of this module.

Carl Freidrich Gauß, Germany, 1777-1855.

Although he produced many very important original ideas, this wasn't one of them. The Chinese knew of "Gaussian Elimination" about 2000 years ago. His actual contributions included major discoveries in the areas of number theory, geometry, and astronomy.

Example 3.2

Consider the problem:

$$\begin{pmatrix} -1 & 3 & -1 \\ 3 & 1 & -2 \\ 2 & -2 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 0 \\ -9 \end{pmatrix}$$

We can perform a sequence of elementary row operations to yield the system:

$$\begin{pmatrix} -1 & 3 & -1 \\ 0 & 2 & -1 \\ 0 & 0 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 3 \\ -5 \end{pmatrix}.$$

Gaussian Elimination: perform elementary row operations such as

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

being replaced by

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} + \mu_{21} a_{11} & a_{22} + \mu_{21} a_{12} & a_{23} + \mu_{21} a_{13} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} =$$

$$A + \mu_{21} \begin{pmatrix} 0 & 0 & 0 \\ a_{11} & a_{12} & a_{13} \\ 0 & 0 & 0 \end{pmatrix}$$

where $\mu_{21} = -a_{21}/a_{11}$, so that $a_{21} + \mu_{21} a_{11} = 0$.

Note that

$$\begin{pmatrix} 0 & 0 & 0 \\ a_{11} & a_{12} & a_{13} \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

so we can write the row operation as $(I + \mu_{21} E^{(21)})A$, where $E^{(pq)}$ is the matrix of all zeros, except for $e_{pq} = 1$.

In general each of the row operations in Gaussian Elimination can be written as

$$(I + \mu_{pq} E^{(pq)})A \quad \text{where } 1 \leq q < p \leq n, \tag{2}$$

and $(I + \mu_{pq} E^{(pq)})$ is an example of a ***Unit Lower Triangular Matrix***.

We can conclude that each step of the process will involve multiplying A by a unit lower triangular matrix, resulting in an upper triangular matrix.

Definition 3.3 (Lower Triangular)

$L \in \mathbb{R}^{n\times n}$ is a *Lower Triangular (LT) Matrix* if the only non-zero entries are on or below the main diagonal, i.e., if $l_{ij} = 0$ for $1 \le i < j \le n$.
It is a *unit Lower Triangular matrix* if, in addition, $l_{ii} = 1$.

Examples:

Definition 3.4

Upper Triangular] $U \in \mathbb{R}^{n\times n}$ is an *Upper Triangular (UT) matrix* if $u_{ij} = 0$ for $1 \leq j < i \leq n$. It is a *Unit Upper Triangular Matrix* if $u_{ii} = 1$.

Examples:

Triangular matrices have many important properties. A very important one is: **the determinant of a triangular matrix is the product of the diagonal entries**. For a proof, see Exercise 3.5.

There are other important properties of triangular matrices, but first we need the idea of **matrix partitioning**.

Definition 3.5 (Submatrix)

X is a *submatrix* of A if it can be obtained by deleting some rows and columns of A.

Example:

Definition 3.6 (Leading Principal Submatrix)

The ***Leading Principal Submatrix*** of order k of $A \in \mathbb{R}^{n\times n}$ is $A^{(k)} \in \mathbb{R}^{k\times k}$ obtained by deleting all but the first k rows and columns of A. (Simply put, it's the $k \times k$ matrix in the top left-hand corner of A).

Example:

Matrix partitioning

To ***partition a matrix*** means to divide it into contiguous blocks that are submatrices.

Example:

Next recall that if A and V are matrices of the same size, and each are partitioned

$$A = \left(\begin{array}{c|c} B & C \\ \hline D & E \end{array}\right), \qquad \text{and} \qquad V = \left(\begin{array}{c|c} W & X \\ \hline Y & Z \end{array}\right),$$

where B is the same size as W, C is the same size as X, etc. Then

$$AV = \left(\begin{array}{c|c} BW + CY & BX + CZ \\ \hline DW + EY & DX + EZ \end{array}\right).$$

Theorem 3.7 (Properties of Lower Triangular Matrices)

For any integer $n \geq 2$:

(i) If L_1 and L_2 are $n \times n$ *Lower Triangular* (LT) Matrices that so too is their product $L_1 L_2$.

(ii) If L_1 and L_2 are $n \times n$ *Unit Lower Triangular* matrices, then so too is their product $L_1 L_2$.

(iii) L_1 is nonsingular if and only if all the $l_{ii} \neq 0$. In particular all Unit LT matrices are nonsingular.

(iv) The inverse of a LT matrix is an LT matrix. The inverse of a unit LT matrix is a unit LT matrix.

We restate Part (iv) as follows:

> *Suppose that $L \in \mathbb{R}^{n \times n}$ is a lower triangular matrix with $n \geq 2$, and that there is a matrix $L^{-1} \in \mathbb{R}^{n \times n}$ such that $L^{-1}L = I_n$. Then L^{-1} is also a lower triangular matrix.*

Theorem 3.8 (Properties of Upper Triangular Matrices)

Statements that are analogous to those concerning the properties of lower triangular matrices hold for upper triangular and unit lower triangular matrices. (For proof, see the exercises at the end of this section).

Exercise 3.3

Every step of Gaussian Elimination can be thought of as a left multiplication by a unit lower triangular matrix. That is, we obtain an upper triangular matrix U by multiplying A by k unit lower triangular matrices: $L_k L_{k-1} L_{k-2} \dots L_2 L_1 A = U$, where each $L_i = I + \mu_{pq} E^{(pq)}$, and $E^{(pq)}$ is the matrix whose only non-zero entry is $e_{pq} = 1$. Give an expression for k in terms of n.

Exercise 3.4

Let L be a lower triangular $n \times n$ matrix. Show that $\det(L) = \prod_{j=1}^{n} l_{jj}$. Hence give a necessary and sufficient condition for L to be invertible. What does that tell us about *Unit* Lower Triangular Matrices?

Exercise 3.5

Let L be a lower triangular matrix. Show that each diagonal entry of L, l_{jj} is an eigenvalue of L.

Exercise 3.6

Prove Parts (i)–(iii) of Theorem 3.7.

Exercise 3.7

Prove Theorem 3.8.

Exercise 3.8

Construct an alternative proof of the first part of 3.7 (iv) as follows: Suppose that L is a non-singular lower triangular matrix. If $\boldsymbol{b} \in \mathbb{R}^n$ is such that $b_i = 0$ for $i = 1, \dots, k \leq n$, and $\boldsymbol{y}$ solves $L\boldsymbol{y} = \boldsymbol{b}$, then $y_i = 0$ for $i = 1, \dots, k \leq n$. (Hint: partition L by the first k rows and columns.)

Now use this to give a alternative proof of the fact that the inverse of a lower triangular matrix is itself lower triangular.

Solving Linear Systems

§**3.3** LU-**factorisation**

MA385/530 – Numerical Analysis 1

November 2019

In these slides,

- *LT means "lower triangular"*
- *UT means "upper triangular"*

The goal of this section is to demonstrate that the process of Gaussian Elimination applied to a matrix A is equivalent to factoring A as the product of a unit lower triangular and upper triangular matrix.

The Section 3.2 we saw that each elementary row operation in Gaussian Elimination involves replacing A with $(I + \mu_{rs}E^{(rs)})A$.

Example: For the 3×3 case, this involved computing

$$(I + \mu_{32}E^{(32)})(I + \mu_{31}E^{(31)})(I + \mu_{21}E^{(21)})A.$$

In general we multiply A by a sequence of matrices

$$(I + \mu_{rs} E^{(rs)}),$$

all of which are unit lower triangular matrices.

When we are finished we have reduced A to an upper triangular matrix.

So we can write the whole process as

$$L_k L_{k-1} L_{k-2} \dots L_2 L_1 A = U, \qquad (3)$$

where each of the L_i is a unit LT matrix.

But from Theorem 3.2.6, we know that the product of unit LT matrices is itself a unit LT matrix. So we can write the whole process described in (3) as

$$\tilde{L}A = U. \tag{4}$$

But Theorem 3.2.6 also tells us that the inverse of a unit LT matrix exists and is a unit LT matrix. So we can write (4) as

$$A = LU$$

where L is unit lower triangular and U is upper triangular.

This is called "LU-**factorisation**".

Definition 3.9

The LU-**factorization** of the matrix is a unit lower triangular matrix L and an upper triangular matrix U such that $LU = A$.

Example 3.10

If $A = \begin{pmatrix} 3 & 2 \\ -1 & 2 \end{pmatrix}$ then:

Example 3.11

If $A = \begin{pmatrix} 3 & -1 & 1 \\ 2 & 4 & 3 \\ 0 & 2 & -4 \end{pmatrix}$ then:

We now want to work out formulae for L and U where

$$a_{i,j} = (LU)_{ij} = \sum_{k=1}^{n} l_{ik} u_{kj} \qquad 1 \leq i, j \leq n.$$

Since L and U are triangular,

$$\text{If } i \leq j \quad \text{then} \quad a_{i,j} = \sum_{k=1}^{i} l_{ik} u_{kj} \tag{5a}$$

$$\text{If } j < i \quad \text{then} \quad a_{i,j} = \sum_{k=1}^{j} l_{ik} u_{kj} \tag{5b}$$

The first of these equations can be written as

$$a_{i,j} = \sum_{k=1}^{i-1} l_{ik} u_{kj} + l_{ii} u_{ij}.$$

But $l_{ii} = 1$ so:

$$u_{i,j} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \quad \begin{cases} i = 1, \dots, j-1, \\ j = 2, \dots, n. \end{cases} \tag{6a}$$

And from the second:

$$l_{i,j} = \frac{1}{u_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right) \quad \begin{cases} i = 2, \dots, n, \\ j = 1, \dots, i-1. \end{cases} \tag{6b}$$

Example 3.12

Find the LU-factorisation of

$$A = \begin{pmatrix} -1 & 0 & 1 & 2 \\ -2 & -2 & 1 & 4 \\ -3 & -4 & -2 & 4 \\ -4 & -6 & -5 & 0 \end{pmatrix}$$

Full details of Example 3.12: First, using (6a) with $i = 1$ we have $u_{1j} = a_{1j}$:

$$U = \begin{pmatrix} -1 & 0 & 1 & 2 \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix}.$$

Then (6b) with $j = 1$ we have $l_{i1} = a_{i1}/u_{11}$:

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & l_{32} & 1 & 0 \\ 4 & l_{42} & l_{43} & 1 \end{pmatrix}.$$

Next (6a) with $i = 2$ we have $u_{2j} = a_{2j} - l_{21}u_{2j}$:

$$U = \begin{pmatrix} -1 & 0 & 1 & 2 \\ 0 & -2 & -1 & 0 \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix},$$

then (6b) with $j = 2$ we have $l_{i2} = (a_{i2} - l_{i1}u_{12})/u_{22}$:

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ 4 & 3 & l_{43} & 1 \end{pmatrix}$$

Etc....

Not every matrix has an LU-factorisation. So we need to characterise the matrices that do.

To prove the next theorem we need the Cauchy-Binet Formula: $\det(AB) = \det(A)\det(B)$.[1]

Theorem 3.13

If $n \geq 2$ and $A \in \mathbb{R}^{n\times n}$ is such that every leading principal submatrix of A is nonsingular for $1 \leq k < n$, then A has an LU-factorisation.

[1] Wikipedia disagrees with this attribution

Exercise 3.9

Many textbooks and computing systems compute the factorisation $A = LDU$ where L and U are unit lower and *unit* upper triangular matrices respectively, and D is a diagonal matrix. Show such a factorisation exists, providing that if $n \geq 2$ and $A \in \mathbb{R}^{n \times n}$, then every leading principal submatrix of A is nonsingular for $1 \leq k < n$.

Solving Linear Systems

§3.4 Solving linear systems (i.e., actually solving $Ax = b$)

MA385/MA530 – Numerical Analysis 1

November 2019

We now know

- there are sufficient conditions that guarantee we can factorise A as LU;
- How to compute L and U.

But our overarching goal is to solve: "*find* $\boldsymbol{x} \in \mathbb{R}^n$ *such that* $A\boldsymbol{x} = \boldsymbol{b}$, *for some* $\boldsymbol{b} \in \mathbb{R}^n$". We do this by first solving $L\boldsymbol{y} = \boldsymbol{b}$ for $\boldsymbol{y} \in \mathbb{R}^n$ and then $U\boldsymbol{x} = \boldsymbol{y}$. Because L and U are triangular, this is easy. The process is called **back-substitution**.

Example 3.14

Use LU-factorisation to solve

$$\begin{pmatrix} -1 & 0 & 1 & 2 \\ -2 & -2 & 1 & 4 \\ -3 & -4 & -2 & 4 \\ -4 & -6 & -5 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -2 \\ -3 \\ -1 \\ 1 \end{pmatrix}$$

Solution: In Example 4 of Section 3.3, we saw that

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ 4 & 3 & 2 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} -1 & 0 & 1 & 2 \\ 0 & -2 & -1 & 0 \\ 0 & 0 & -3 & -2 \\ 0 & 0 & 0 & -4 \end{pmatrix}$$

So then...

Example 3.15

Suppose we want to compute the LU-factorisation of

$$A = \begin{pmatrix} 0 & 2 & -4 \\ 2 & 4 & 3 \\ 3 & -1 & 1 \end{pmatrix}.$$

We can't compute l_{21} because $u_{11} = 0$. But if we swap rows 1 and 3, then we can (we did this as Example 3.4.3). This like changing the order of the linear equations we want to solve. If

$$P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad \text{then} \quad PA = \begin{pmatrix} 3 & -1 & 1 \\ 2 & 4 & 3 \\ 0 & 2 & -4 \end{pmatrix}.$$

This is called *Pivoting* and P is the permutation matrix.

Definition 3.16

$P \in \mathbb{R}^{n\times n}$ is a *Permutation Matrix* if every entry is either 0 or 1 (it is a Boolean Matrix) and if all the row and column sums are 1.

Theorem 3.17

For **any** $A \in \mathbb{R}^{n\times n}$ there exists a permutation matrix P such that $PA = LU$.

For a proof, see p53 of text book.

How efficient is the method of LU-factorization for solving $A\boldsymbol{x} = \boldsymbol{b}$? That is, how many computational steps (additions and multiplications) are required? In Section 2.6 of the textbook, you'll find a discussion that goes roughly as follows:

Suppose we want to compute $l_{i,j}$. Recall the formula from Section 3.4:

$$l_{i,j} = \frac{1}{u_{jj}}\Big(a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}\Big) \qquad i = 2, \ldots, n, \text{ and } j = 1, \ldots, i-1.$$

We see that this requires $j-2$ additions, $j-1$ multiplications, 1 subtraction and 1 division: a total of $2j-1$ operations.

In Exercise 3.13 we will show that

$$1 + 2 + \cdots + k = \frac{1}{2}k(k+1), \quad \text{and}$$

$$1^2 + 2^2 + \cdots k^2 = \frac{1}{6}k(k+1)(2k+1).$$

So the number of operations required for computing L is

$$\sum_{i=2}^{n}\sum_{j=1}^{i-1}(2j-1) = \sum_{i=2}^{n} i^2 - 2i + 1 =$$
$$\frac{1}{6}n(n+1)(2n+1) - n(n+1) + n \leq Cn^3$$

for some C.

A similar (slightly smaller) number of operations is required for computing U. (For a slightly different approach that yields cruder estimates, but requires a little less work, have a look at Lecture 10 of Stewart's *Afternotes on Numerical Analysis*).

. . . .

This doesn't tell us how long a computer program will take to run, but it does tell us how the execution time grows with n. For example, if $n = 100$ and the program takes a second to execute, then if $n = 1000$ we'd expect it to take about a thousand seconds.

Towards an error analysis

Unlike the other methods we studied so far in this course, we shouldn't have to do an error analysis, in the sense of estimating the difference between the true solution, and our numerical one. That is because the LU-approximation approach should give us exactly the true solution.

However, things are not that simple. Unlike the methods in earlier sections, the effects of (inexact) floating point computations become very pronounced. In the next section, we'll develop the ideas needed to quantify these effects.

Exercise 3.10

Suppose that A has an LDU-factorisation (see Exercises 3.9). How could this factorization be used to solve $A\boldsymbol{x} = \boldsymbol{b}$?

Exercise 3.11

Prove that

$$1 + 2 + \cdots + k = \frac{1}{2}k(k+1), \quad \text{and}$$

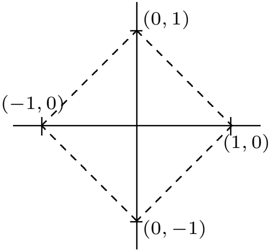
$$1^2 + 2^2 + \cdots k^2 = \frac{1}{6}k(k+1)(2k+1).$$

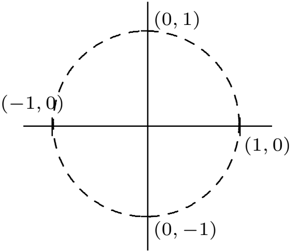
§3 Solving linear systems

§3.5 Vector and Matrix Norms

MA385 – Numerical Analysis 1

November 2019





All computer implementations of algorithms that involve floating-point numbers (roughly, finite decimal approximations of real numbers) contain errors due to round-off error.

It transpires that computer implementations of LU-factorization, and related methods, lead to these round-off errors being greatly magnified: this phenomenon is the main focus of this final section of the course.

You might remember from earlier sections of the course that we had to assume functions where well-behaved in the sense that

$$\frac{|f(x) - f(y)|}{|x - y|} \leq L,$$

for some number L, so that our numerical schemes (e.g., fixed point iteration, Euler's method, etc) would work. If a function *doesn't* satisfy a condition like this, we say it is "ill-conditioned".

One of the consequences is that a small error in the inputs gives a large error in the outputs.

We'd like to be able to express similar ideas about matrices: that $A(u - v) = Au - Av$ is not too "large" compared to $u - v$. To do this we used the notion of a "norm" to describing the relatives sizes of the vectors $\boldsymbol{u}$ and $A\boldsymbol{u}$.

When we want to consider the size of a real number, without regard to sign, we use the *absolute value*. Important properties of this function are:

1. $|x| \geq 0$ for all x.
2. $|x| = 0$ if and only if $x = 0$.
3. $|\lambda x| = |\lambda||x|$.
4. $|x + y| \leq |x| + |y|$ (triangle inequality).

This notion can be extended to vectors and matrices.

Definition 3.18

Let $\mathbb{R}^n$ be all the vectors of length n of real numbers. The function $\|\cdot\|$ is called a **norm** of $\mathbb{R}^n$ if, for all $u, v \in \mathbb{R}^n$

1. $\|v\| \geq 0$,
2. $\|v\| = 0$ if and only if $v = 0$.
3. $\|\lambda v\| = |\lambda|\|v\|$ for any $\lambda \in \mathbb{R}$,
4. $\|u + v\| \leq \|u\| + \|v\|$ (triangle inequality).

Norms on vectors in $\mathbb{R}^n$ quantify the *size* of the vector. But there are different ways of doing this...

Definition 3.19

Let $\boldsymbol{v} \in \mathbb{R}^n$: $\boldsymbol{v} = (v_1, v_2, \ldots, v_{n-1}, v_n)^T$.

(i) The 1-*norm* (a.k.a. the *Taxi cab* norm) is $\|v\|_1 = \sum_{i=1}^{n} |v_i|$.

(ii) The 2-*norm* (a.k.a. the *Euclidean norm*) $\|v\|_2 = \left(\sum_{i=1}^{n} v_i^2\right)^{1/2}$.

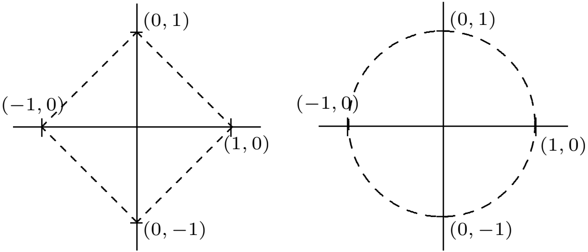
Note, if $\boldsymbol{v}$ is a vector in $\mathbb{R}^n$, then

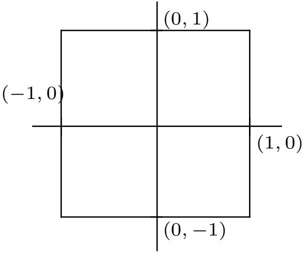
$$\boldsymbol{v}^T\boldsymbol{v} = v_1^2 + v_2^2 + \cdots + v_n^2 = \|\boldsymbol{v}\|_2^2.$$

(iii) The ∞-*norm* (a.k.a. the $\max$-*norm*) $\|v\|_\infty = \max_{i=1}^{n} |v_i|$.

Example: $\boldsymbol{v} = (-2, 4, -4)$

The unit balls in $\mathbb{R}^2$ given by $\|\cdot\|_1$ (top left), $\|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + x_2^2} = 1$ (top right), and $\|\cdot\|_\infty$.





It is easy to show that $\|\cdot\|_1$ and $\|\cdot\|_\infty$ are norms. And it is not hard to show that $\|\cdot\|_2$ satisfies conditions (1), (2) and (3) of Definition 3.18.

It takes a little bit of effort to show that $\|\cdot\|_2$ satisfies the triangle inequality; details are given in Section 3.5.9 of the notes.

Definition 3.20

Given any norm $\|\cdot\|$ on $\mathbb{R}^n$, there is a *subordinate matrix norm* on $\mathbb{R}^{n\times n}$ defined by

$$\|A\| = \max_{\boldsymbol{v}\in\mathbb{R}^n_\star} \frac{\|A\boldsymbol{v}\|}{\|\boldsymbol{v}\|}, \qquad (7)$$

where $A \in \mathbb{R}^{n\times n}$ and $\mathbb{R}^n_\star = \mathbb{R}^n/\{\mathbf{0}\}$.

You might wonder why we define a matrix norm like this. The reason is that we like to think of A as an *operator* on $\mathbb{R}^n$: if $\boldsymbol{v} \in \mathbb{R}^n$ then $\boldsymbol{A}v \in \mathbb{R}^n$. So rather than the norm giving us information about the "size" of the entries of a matrix, it tells us how much the matrix can change the size of a vector.

It is not obvious from the above definition how to calculate the norm of a given matrix. We'll see that

- The ∞-norm of a matrix is also the largest absolute-value row sum.
- The 1-norm of a matrix is also the largest absolute-value column sum.
- The 2-norm of the matrix A is the square root of the largest eigenvalue of $A^T A$.

Theorem 3.21

For any $A \in \mathbb{R}^{n\times n}$ the subordinate matrix norm associated with $\|\cdot\|_\infty$ on $\mathbb{R}^n$ can be computed by

$$\|A\|_\infty = \max_{i=1,\dots,n} \sum_{j=1}^{n} |a_{ij}|.$$

A similar result holds for the 1-norm, the proof of which is left as an exercise.

Theorem 3.22

For any $A \in \mathbb{R}^{n\times n}$ the subordinate matrix norm associated with $\|\cdot\|_\infty$ on $\mathbb{R}^n$ can be computed by

$$\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{i,j}|. \tag{8}$$

Computing the 2-norm of a matrix is a little harder that computing the 1- or ∞-norms. However, later we'll need estimates not just for $\|A\|$, but also $\|A^{-1}\|$. And, unlike the 1- and ∞-norms, we can estimate $\|A^{-1}\|_2$ without explicitly forming A^{-1}.

We begin by recalling some important facts about eigenvalues and eigenvectors.

Definition 3.23

Let $A \in \mathbb{R}^{n\times n}$. We call $\lambda \in \mathbb{C}$ an *eigenvalue* of A if there is a non-zero vector $\boldsymbol{x} \in \mathbb{C}^n$ such that

$$A\boldsymbol{x} = \lambda \boldsymbol{x}.$$

We call any such $\boldsymbol{x}$ an *eigenvector of A associated with λ*.

(i) If A is a real symmetric matrix (i.e., $A = A^T$), its eigenvalues and eigenvectors are all real-valued.

(ii) If λ is an eigenvalue of A, the $1/\lambda$ is an eigenvalue of A^{-1}.

(iii) If $\boldsymbol{x}$ is an eigenvector associated with the eigenvalue λ then so too is $\eta\boldsymbol{x}$ for any non-zero scalar η.

(iv) An eigenvector may be *normalised* as $\|\boldsymbol{x}\|_2^2 = \boldsymbol{x}^T\boldsymbol{x} = 1$.

(v) There are n eigenvectors λ_1, λ_n, ..., λ_n associated with the real symmetric matrix A. Let $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$, ..., $\boldsymbol{x}^{(n)}$ be the associated normalised eigenvectors. Then the eigenvectors are linearly independent and so form a basis for $\mathbb{R}^n$. That is, any vector $v \in \mathbb{R}^n$ can be written as a linear combination:

$$\boldsymbol{v} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}^{(i)}.$$

(vi) Furthermore, these eigenvectors are *orthogonal* and *orthonormal*:

$$\left(x^{(i)}\right)^T x^{(j)} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Here is a useful consequence of (v) and (vi), which we will use repeatedly.

The *singular values* of a matrix A are the square roots of the eigenvalues of $A^T A$. They play a very important role in matrix analysis and in areas of applied linear algebra, such as image and text processing. Our interest here is in their relationship to $\|A\|_2$.

But first we'll prove a theorem about certain matrices (so called, "normal matrices").

Theorem 3.24

For any matrix A, the eigenvalues of $A^T A$ are real and non-negative.

Part of the above proof involved showing that, if $(A^T A)\boldsymbol{x} = \lambda x$, then

$$\sqrt{\lambda} = \frac{\|A\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}.$$

This at the very least tells us that

$$\|A\|_2 := \max_{\boldsymbol{x}\in\mathbb{R}^n_\star} \frac{\|A\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} \geq \max_{i=1,\dots,n} \sqrt{\lambda_i}.$$

With a bit more work, we can show that if $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ are the the eigenvalues of $B = A^T A$, then

$$\|A\|_2 = \sqrt{\lambda_n}.$$

Theorem 3.25

Let $A \in \mathbb{R}^{n\times n}$. Let $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$, be the eigenvalues of $B = A^T A$. Then

$$\|A\|_2 = \max_{i=1,\ldots,n} \sqrt{\lambda_i} = \sqrt{\lambda_n},$$

Here is the main idea. For full details, see the text-book.

Exercise 3.12 ($\star$)

Show that, for any vector $\boldsymbol{x} \in \mathbb{R}^n$, $\|\boldsymbol{x}\|_\infty \leq \|\boldsymbol{x}\|_2$ and $\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{x}\|_1 \|\boldsymbol{x}\|_\infty$. For each of these inequalities, give an example for which the equality holds. Deduce that $\|\boldsymbol{x}\|_\infty \leq \|\boldsymbol{x}\|_2 \leq \|\boldsymbol{x}\|_1$.

Exercise 3.13

Show that if $x \in \mathbb{R}^n$, then $\|\boldsymbol{x}\|_1 \leq n\|\boldsymbol{x}\|_\infty$ and that $\|\boldsymbol{x}\|_2 \leq \sqrt{n}\|\boldsymbol{x}\|_\infty$.

Exercise 3.14

Show that, for *any* subordinate matrix norm on $\mathbb{R}^{n \times n}$, the norm of the identity matrix is 1.

Exercise 3.15 ($\star$)

Prove that

$$\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{i,j}|.$$

Hint: Suppose that

$$\sum_{i=1}^{n} |a_{ij}| \le C, \qquad j = 1, 2, \dots n,$$

show that for *any* vector $\boldsymbol{x} \in \mathbb{R}^n$

$$\sum_{i=1}^{n} |(A\boldsymbol{x})_i| \le C\|\boldsymbol{x}\|_1.$$

Now find a vector $\boldsymbol{x}$ such that $\sum_{i=1}^{n} |(A\boldsymbol{x})_i| = C\|\boldsymbol{x}\|_1$. Now deduce the result.

As mentioned on Slide 59, it takes a little effort to show that $\|\cdot\|_2$ is indeed a norm on $\mathbb{R}^2$; in particular to show that it satisfies the triangle inequality, we need the Cauchy-Schwarz inequality.

Lemma 1 (Cauchy-Schwarz)

$$|\sum_{i=1}^{n} u_i v_i| \leq \|\boldsymbol{u}\|_2 \|\boldsymbol{v}\|_2, \qquad \forall \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n.$$

The proof can be found in any text-book on analysis.

Now can now apply Cauchy-Schwatz to show that

$$\|\boldsymbol{u} + \boldsymbol{v}\|_2 \leq \|\boldsymbol{u}\|_2 + \|\boldsymbol{v}\|_2.$$

(PTO)

This is because

$$\begin{aligned} \|u+v\|_2^2 &= (u+v)^T(u+v) \\ &= u^Tu + 2u^Tv + v^Tv \\ &\leq u^Tu + 2|u^Tv| + v^Tv \quad \text{(by the triangle-inequality)} \\ &\leq u^Tu + 2\|u\|\|v\| + v^Tv \quad \text{(by Cauchy-Schwarz)} \\ &= (\|u\| + \|v\|)^2. \end{aligned}$$

It follows directly that

Corollary 2

$\|\cdot\|_2$ is a norm.

Solving linear systems of equations

§3.6 Condition Numbers

MA385 – Numerical Analysis 1

November 2019

Numerical solutions to some linear systems are adversely affected by round-off errors.

This phenomenon is related the matrices in the linear systems. Those matrices for which the issue is particularly prevalent are referred to as being ***ill-conditioned***.

For any matrix, we can assign a numerical score that gives an indication of whether it is ill-conditioned. That score is called the ***condition number***, and is the subject of these section.

The condition number is defined in terms of matrix norms.

Suppose we have a vector norm, $\|\cdot\|$ and associated subordinate matrix norm. It is not hard to see that

$$\|A\boldsymbol{u}\| \leq \|A\|\|\boldsymbol{u}\| \quad \text{for } \textit{any } \boldsymbol{u} \in \mathbb{R}^n, A \in \mathbb{R}^{n\times n}.$$

Here is why:

There is an analogous statement for the product of two matrices:

Definition 3.26 (Consistent matrix norm)

A matrix norm $\|\cdot\|$ is ***consistent*** (or "*sub-multiplicative*" if

$$\|AB\| \leq \|A\|\|B\|, \qquad \text{for all } A, B \in \mathbb{R}^{n\times n}.$$

Theorem 3.27

Any subordinate matrix norm is consistent.

The proof is left to Exercise 3.17. That exercises also demonstrates that there are matrix norms which are *not* consistent.

[Please read this slide in your own time!]

Modern computers don't store numbers in decimal (base 10), but in binary (base 2) "floating point numbers" of the form :

$$x = \pm a \times 2^{b-M}.$$

Most use *double precision*, where 8 bytes (64 bits or *binary digits*) are used to store

- the sign (1 bit),
- a, called the "significand" or "mantissa" (52 bits)
- and the exponent, $b - 1023$ (11 bits)

Note that a has roughly 16 decimal digits.

(Some older computer systems sometimes use *single precision* where a has 23 bits — giving 8 decimal digits — and b has 7; so too do many new GPU-based systems).

[OK, you can start reading again!] When we try to store a real number x on a computer, we actually store the nearest floating-point number. That is, we end up storing $x + \delta x$, where δx is the "round-off" error.

But the quantity we are mainly interested in is the **relative error:** $|\delta x|/|x|$.

Since this is not a course on computer architecture, we'll simplify a little and just take it that single and double precision systems lead to a relative error of 10^{-8} and 10^{-16} respectively.

(Sew p68–70 of Süli and Mayers for a thorough development of the concept of a condition number).

Suppose we use, say, LU-factorization and back-substitution on a computer to solve

$$A\boldsymbol{x} = \boldsymbol{b}.$$

Because of the "round-off error" we actually solve

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

Our problem now is, for a given A, if we know the (relative) error in $\boldsymbol{b}$, can we find an upper-bound on the relative error in $\boldsymbol{x}$?

Definition 3.28

The *condition number* of a matrix, with respect to a particular matrix norm $\|\cdot\|_\star$ is

$$\kappa_\star(A) = \|A\|_\star \|A^{-1}\|_\star.$$

If $\kappa_\star(A) \gg 1$ then we say A is *ill-conditioned*.

Example: Find the condition number κ_∞ of

$$A = \begin{pmatrix} 10 & 12 \\ 0.08 & 0.1 \end{pmatrix}.$$

Theorem 3.29

Suppose that $A \in \mathbb{R}^{n\times n}$ is nonsingular and that $\boldsymbol{b}, x \in \mathbb{R}^n$ are non-zero vectors. If $A\boldsymbol{x} = \boldsymbol{b}$ and $A(\boldsymbol{x} + \boldsymbol{\delta x}) = (\boldsymbol{b} + \boldsymbol{\delta b})$ then

$$\frac{\|\boldsymbol{\delta x}\|}{\|\boldsymbol{x}\|} \leq \kappa(A)\frac{\|\boldsymbol{\delta b}\|}{\|\boldsymbol{b}\|}.$$

Example 3.30

Suppose we are using a computer to solve $A\boldsymbol{x} = \boldsymbol{b}$ where

$$A = \begin{pmatrix} 10 & 12 \\ 0.08 & 0.1 \end{pmatrix} \quad \text{and } \boldsymbol{b} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

But, due to round-off error, right-hand side has a relative error (in the ∞-norm) of 10^{-6}. Give a bound for the relative error in $\boldsymbol{x}$ in the ∞-norm.

For every matrix norm we get a different condition number.

Example 3.31

Let A be the $n \times n$ matrix

$$A = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ 1 & 0 & 0 & \dots & 1 \end{pmatrix}.$$

What are $\kappa_1(A)$, and $\kappa_\infty(A)$?

First we compute $\|A\|_1$ and $\|A_\infty\|$.

For this very special example, it is easy to write down the inverse of A:

$$A^{-1} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ -1 & 0 & 0 & \dots & 1 \end{pmatrix}.$$

To compute $\kappa_1(A)$ and $\kappa_\infty(A)$, we need to know A^{-1}, which is usually not practical. However, for κ_2, we are able to *estimate* the condition number of A without knowing A^{-1}.

Recall that $\|A\|_2 = \sqrt{\lambda_n}$ where λ_n is the largest eigenvalue of $B = A^T A$.

We can also show that $\|A^{-1}\|_2 = \dfrac{1}{\sqrt{\lambda_1}}$ where λ_1 is the smallest eigenvalue of B (see Section 3.6.5 of notes). So

$$\kappa_2(A) = \Big(\lambda_n/\lambda_1\Big)^{1/2}.$$

. . . .

Motivated by this, we'll finish MA385, by studying an easy way of estimating the eigenvalues of a matrix.

Exercise 3.17

(i) Prove that, if $\|\cdot\|$ is a subordinate matrix norm, then it is *consistent*, i.e., for any pair of $n \times n$ matrices, A and B, we have $\|AB\| \leq \|A\|\|B\|$.

(ii) One might think it intuitive to define the "max" norm of a matrix as follows:

$$\|A\|_{\underset{\infty}{\sim}} = \max_{i,j} |a_{ij}|.$$

Show that this is indeed a norm on $\mathbb{R}^{n\times n}$. Show that, however, it is not consistent.

Exercise 3.18

Let A be the matrix

$$A = \begin{pmatrix} 0.1 & 0 & 0 \\ 10 & 0.1 & 10 \\ 0 & 0 & 0.1 \end{pmatrix}$$

Compute $\kappa_\infty(A)$. Suppose we wish to solve the system of equations $A\boldsymbol{x} = \boldsymbol{b}$ on *single precision* computer system (i.e., the relative error in any stored number is approximately 10^{-8}). Give an upper bound on the relative error in the computed solution $\boldsymbol{x}$.

Solving linear systems of equations

§3.7 Gerschgorin's Theorems

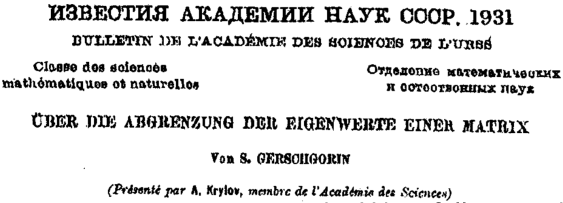
MA385 – Numerical Analysis 1

November 2018

There are some extra details posted as an "Appendix" to this section

The goal of this final section is to learn a technique for estimating eigenvalues of matrices.

The idea dates from 1931, and is a simple as it is useful. Although known to mathematicians in the USSR, the original paper was not widely read.



ИЗВЕСТИЯ АКАДЕМИИ НАУК СССР. 1931

BULLETIN DE L'ACADÉMIE DES SCIENCES DE L'URSS

Classe des sciences mathématiques et naturelles

Отделение математических и естественных наук

ÜBER DIE ABGRENZUNG DER EIGENWERTE EINER MATRIX

Von S. GERSCHGORIN

(Présenté par A. Krylov, membre de l'Académie des Sciences)

It received main-stream attention in the West following the work of Olga Taussky (*A recurring theorem on determinants*, American Mathematical Monthly, vol 56, p672–676. 1949.)

See also https://www.math.wisc.edu/hans/paper_archive/other_papers/hs057.pdf

(See Section 5.4 of Süli and Mayers).

Theorem 3.32 (Gerschgorin's First Theorem)

Given a matrix $A \in \mathbb{R}^{n\times n}$, define the n *Gerschgorin Discs*, $D_1, D_2, \dots, D_n$ as the discs in the complex plane where D_i has centre a_{ii} and radius r_i:

$$r_i = \sum_{j=1, j\neq i}^{n} |a_{ij}|.$$

So $D_i = \{z \in \mathbb{C} : |a_{ii} - z| \leq r_i\}$. All the eigenvalues of A are contained in the union of the Gerschgorin discs.

Proof.

The proof makes no assumption about A being symmetric, or the eigenvalues being real. However, if A is symmetric, then its eigenvalues are real and so the theorem can be simplified: the eigenvalues of A are contained in the union of the intervals $I_i = [a_{ii} - r_i, a_{ii} + r_i]$, for $i = 1, \ldots, n$.

Example 3.33

Let

$$A = \begin{pmatrix} 4 & -2 & 1 \\ -2 & -3 & 0 \\ 1 & 0 & 2 \end{pmatrix}$$

Theorem 3.34 (Gerschgorin's Second Theorem)

Given a matrix $A \in \mathbb{R}^{n \times n}$, let the n Gerschgorin disks be as defined in Theorem 3.32. If k of discs are disjoint (have an empty intersection) from the others, their union contains k eigenvalues.

Proof: not covered in class. If interested, see the appendix, or the textbooks.

Example 3.35

Locate the regions contains the eigenvalues of

$$A = \begin{pmatrix} -3 & 1 & 2 \\ 1 & 4 & 0 \\ 2 & 0 & -6 \end{pmatrix}$$

(The eigenvalues are approximately -7.018, -2.130 and 4.144.)

Example 3.36

Use Gerschgorin's Theorems to find an upper and lower bound for the Singular Values of the matrix

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

Hence give an upper bound for $\kappa_2(A)$.

Exercise 3.20

A real matrix $A = \{a_{i,j}\}$ is *Strictly Diagonally Dominant* if

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{i,j}| \qquad \text{for } i = 1, \ldots, n.$$

Show that all strictly diagonally dominant matrices are nonsingular.

Exercise 3.21

Let

$$A = \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & -3 \end{pmatrix}$$

Use Gerschgorin's theorems to give an upper bound for $\kappa_2(A)$.

Proof of Gerschgorin's First Theorem (Thm 3.32)

Let λ be an eigenvalues of A, so $A\boldsymbol{x} = \lambda x$ for the corresponding eigenvector $\boldsymbol{x}$. Suppose that x_i is the entry of $\boldsymbol{x}$ with largest absolute value. That is $|x_i| = \|\boldsymbol{x}\|_\infty$. Looking at the i^{th} entry of the vector $A\boldsymbol{x}$ we see that

$$(A\boldsymbol{x})_i = \lambda x_i \implies \sum_{j=1}^{n} a_{ij}x_j = \lambda x_i.$$

This can be rewritten as

$$a_{ii}x_i + \sum_{\substack{j=0 \\ j\neq i}}^{n} a_{ij}x_j = \lambda x_i,$$

which gives

$$(a_{ii} - \lambda)x_i = -\sum_{\substack{j=0 \\ j\neq i}}^{n} a_{ij}x_j$$

By the triangle inequality,

$$|a_{ii} - \lambda||x_i| = |\sum_{\substack{j=0 \\ j\neq i}}^{n} a_{ij}x_j| \leq \sum_{\substack{j=0 \\ j\neq i}}^{n} |a_{ij}||x_j| \leq |x_i| \sum_{\substack{j=0 \\ j\neq i}}^{n} |a_{ij}|,$$

since $|x_i| \geq |x_j|$ for all j. Dividing by $|x_i|$ gives

$$|a_{ii} - \lambda| \leq \sum_{\substack{j=0 \\ j\neq i}}^{n} |a_{ij}|,$$

as required.

Proof of Gerschgorin's 2nd Thm (Thm 3.34) We didn't do the proof in class, and you are not expected to know it. Here is a *sketch* of it.

Let $B(\varepsilon)$ be the matrix with entries

$$b_{ij} = \begin{cases} a_{ij} & i = j \\ \varepsilon a_{ij} & i \neq j. \end{cases}$$

So $B(1) = B$ and $B(0)$ is the diagonal matrix whose entries are the diagonal entries of A.

Each of the eigenvalues of $B(0)$ correspond to its diagonal entries and (obviously) coincide with the Gerschgorin discs of $B(0)$ – the centres of the Gerschgorin discs of A.

The eigenvalues of B are the zeros of the characteristic polynomial $\det(B(\varepsilon) - \lambda I)$ of B. Since the coefficients of this polynomial depend continuously on ε, so too do the eigenvalues.

Now as ε varies from 0 to 1, the eigenvalues of $B(\varepsilon)$ trace a path in the complex plane, and at the same time the radii of the Gerschgorin discs of A increase from 0 to the radii of the discs of A. If a particular eigenvalue was in a certain disc for $\varepsilon = 0$, the corresponding eigenvalue is in the corresponding disc for all ε.

Thus if one of the discs of A is disjoint from the others, it must contain an eigenvalue.

The same reasoning applies if k of the discs of A are disjoint from the others; their union must contain k eigenvalues.