Numerical Methods Notes

Hunter Smith

Contents

1 Introduction **2**

1.1 Significant Digits, Rounding and Chopping 2

1.2 Floating Point Representation 7

2 Systems of Linear Equations **7**

2.1 Matrix Factorization 7

3 Homework **12**

Definitions **21**

1 Introduction

1.1 Significant Digits, Rounding and Chopping

Definition 1.1. *Significant digits* are digits beginning with the leftmost nonzero digit and ending with the rightmost correct digit

Significant digits are a representation of the number of digits of precision that a measurement has.

Definition 1.2. *Normalized Scientific Notation* for the decimal system is given by

$$x = \pm 0.d_1 d_2 d_3 \cdots \times 10^n$$

where $d_1 \neq 0$and n is an integer.

Example 1.1. The following is in normalized scientific notation for decimals

$$0.12345 \times 10^5$$

The following are not proper representations

$$0.012345 \times 10^6$$

$$1.2345 \times 10^4$$

Definition 1.3. Normalized Scientific Notation for the binary system is given by

$$x = \pm 0.1 d_1 d_2 d_3 \cdots \times 2^{\pm n}$$

where $d_i = 0$ or $d_i = 1$ and n is an integer.

Example 1.2. The following is in normalized scientific notation for binary

$$0.101 \times 2^{-2}$$

The following are not proper representations

$$0.0101 \times 2^{-1}$$

$$1.01 \times 2^{-3}$$

Scientific notation may have other representations, but they are equivalent to the above definitions.

Definition 1.4. Suppose α and β are two numbers where one is an approximation of the other. The error of β as an approximation to α is $\alpha - \beta$. The *absolute error* is the absolute value $|\alpha - \beta|$. The *relative error* of β as an approximation to α is $\frac{|\alpha-\beta}{\alpha}$.

The distinction between the two errors is that absolute error allows α and β to be interchanged while relative error requires one of the values to be specified as correct.

Example 1.3. Consider x = 0.00347 rounded to x' = 0.0035 and y = 30.158 rounded to 0.3×10^{-4}
The $x' = 0.35 \times 10^{-2}$ has two significant digits with absolute error of 0.3×10^{-4}, relative error of 0.865×10^{-2} and $y' = 0.3015 \times 10^{2}$ has four significant digits with absolute error of 0.2×10^{-2} and relative error of 0.66×10^{-4}

In Python, the following code will return the amount of significant digits in a number

```
1 # Returns counts of significant digits for input
2 def numberOfSignificantDigits(n):
3
4     # Convert n to float
5     n = repr(float(n))
6
7     # Split n at decimal point
8     nSplit = n.split('.')
```

```
    whole = nSplit[0].lstrip('0')

    # Ensure n has only 1 decimal point
    if len(nSplit) == 2:
        decimal = nSplit[1].rstrip('0')
        return len(whole) + len(decimal)

    return len(whole)
```

Definition 1.5. A number is *chopped* to n digits if all the digits following the n'th are removed from the end of the number.

Example 1.4. The following numbers rounded to two decimal places

$$0.217 \approx 0.22,\ 0.475 \approx 0.48,\ 0.592 \approx 0.59$$

Chopping them to two significant digits gives

$$0.217 \approx 0.21,\ 0.475 \approx 0.47,\ 0.592 \approx 0.59$$

Chopping and rounding numbers is very straightforward in Python:

```
import math
def Round(n, roundVal):
    return round(n, roundVal)

def Chop(n, chopVal):
    decimals = len(str(n).split('.')[1])
    if decimals <= chopVal:
        return n
    step = 10.0 ** chopVal
    return math.trunc(step * n) / step
```

Example 1.5. Where n denotes the number of significant digits,

$$0.031,\ n = 2$$

$$031,\ n = 2$$

$$3.1400,\ n = 5$$

$$100.,\ n = 3$$

$$100.0,\ n = 4$$

$$100,\ n = 1$$

$$1230,\ n = 3$$

Example 1.6. Solve the following linear system by carrying out three significant digits of rounding precision and then four.

$$\begin{cases} 0.1036x + 0.2122y = 0.7381 \\ 0.2081x + 0.4247y = 0.9327 \end{cases}$$

When solving to 3 significant digits, we begin with the linear system

$$\begin{cases} 0.104x + 0.212y = 0.738 \\ 0.208x + 0.425y = 0.933 \end{cases}$$

Then we take choose multiple of the first equation such that the coefficient of x becomes 0. In this case, $\frac{0.208}{0.104}$

$$\begin{cases} 0.104x + & 0.212y = 0.738 \\ & 0.425y - \frac{0.208}{0.104}0.212y = 0.933 - \frac{0.208}{0.104}0.738 \end{cases}$$

from which we get

$$0.425y - 0.424y = 0.933 - 1.48$$

$$0.001y = -0.547$$

$$y = -547$$

With four significant digits we have

$$\begin{cases} 0.1036x + 0.2122y = 0.7381 \\ 0.2081x + 0.4247y = 0.9327 \end{cases}$$

Then we choose a multiple of the first equation such that the coefficient of x becomes 0. In this case, $\frac{0.2081}{0.1036}$

$$\begin{cases} 0.1036x + & 0.2122y = 0.7381 \\ & 0.4247y - \frac{0.2081}{0.1036}0.2122y = 0.9327 - \frac{0.2081}{0.1036}0.7381 \end{cases}$$

and solving for y yields from which we get

$$0.4247y - 2.009 \times 0.2122y = 0.9327 - 2.009 \times 0.7381$$

$$-0.001600y = -0.5503$$

$$y = 343.9$$

From the above example, we should observe that accurate data should have its calculations carried out to full precision, otherwise the result will be inaccurate.

1.2 Floating Point Representation

A note must be made about the limits of computing large values. The following definition gives limits to the size of values that computers can handle precisely.

Definition 1.6. The *maximum* and *minimum* value for single-precision floating point values are given by

$$\max(\text{singleFloat}) = 3.4 \times 10^{38}$$

$$\min(\text{singleFloat}) = 1.2 \times 10^{-38}$$

The maximum and minimum value for double-precision floating point values are given by

$$\max(\text{singleFloat}) = 1.8 \times 10^{308}$$

$$\min(\text{singleFloat}) = 2.2 \times 10^{-308}$$

Example 1.7. Using double-precision floating point values, the first example causes an underflow

$$1 + 1 \times 10^{-308} = 1 + 0 = 1$$

$$1 + 3 \times 10^{-308} \neq 1$$

2 Systems of Linear Equations

2.1 Matrix Factorization

A system of linear equations can be written in matrix form

$$A\vec{x} = \vec{b}$$

The main objective is to show that Gaussian Elimination applied to A yields a factorization of A into a product of a lower triangular and an upper triangular matrix. Recall that a lower triangular matrix is a matrix whose entries consist of all 1's along the diagonal and 0's in the upper half, an upper triangular matrix has all non-zero entries on or above the main diagonal.

Example 2.1. Consider the system of equations represented in matrix form

$$\begin{pmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 16 \\ 26 \\ -19 \\ -34 \end{pmatrix}$$

After the first step of Gaussian Elimination, we get the system

$$\begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & -12 & 8 & 1 \\ 0 & 2 & 3 & -14 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ -27 \\ -18 \end{pmatrix}$$

Notice that this form can be obtained by multiplying A with a lower triangular matrix M_1 as follows,

$$M_1 A\vec{x} = M_1\vec{b}$$

where

$$M_1 A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{pmatrix} = \begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & -12 & 8 & 1 \\ 0 & 2 & 3 & -14 \end{pmatrix}$$

After performing two more steps of Gaussian elimination, we are left with

the system

$$M_3M_2M_1A\vec{x} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{pmatrix} = U$$

Finally, we can obtain a representation of A in terms of a lower-diagonal matrix L and upper diagonal U

$$A = M^{-1}U = M_1^{-1}M_2^{-1}M_3^{-1}U = LU$$

where

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 3 & 1 & 0 \\ 0 & -\frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{pmatrix}$$

Theorem 2.1. *Let $A = (a_{ij})$ be an $n \times n$ matrix, assume the forward elimination phase of the naive Gaussian algorithm is applied to A without encountering any 0 divisiors. Let the matrix resulting from this forward elimination phase be denoted by $\tilde{A} = \tilde{a_{ij}}$. If L is a lower diagonal matrix whose non-zero entries correspond to $\tilde{a_{ij}}$ and U is an upper diagonal matrix whose non-zero entries correspond to $\tilde{a_{ij}}$, then $A = LU$*

For a symmetrix matrix that has an ordinary LU factorization, there is another way to factorize the matrix. We can obtain a LDL^T matrix where D is a diagonal matrix. Notice that for a symmetric matrix

$$LU = A = A^T = (LU)^T = U^TL^T = LDL^T$$

Using Python, one can easily find the LDL^T factorizaiton of a matrix:

```
1 #LDL^T Factorization of a matrix
2 from scipy.linalg import ldl
```

```
3 import numpy as np
4
5 def llTranspose(A):
6     lu, d, perm = ldl(A, lower=0)
7     return(lu, d, lu.T)
```

Example 2.2. To find the LDL^T factorization for the matrix

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

we first find the LU factorization which is given by

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

Then, we extract the diagonal elements of U and put them in D to obtain

$$U = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & \frac{3}{4} & 0 & 0 \\ 0 & 0 & \frac{2}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 1 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} = DL^T$$

When a matrix is symmetric *and* positive definite (a matrix is positive definite if $\vec{x}^T A\vec{x} > 0$ for every non zero $\vec{x}$), we can further simplify the factorization into a form called the *Cholesky factorization*

Theorem 2.2. *If A is a real, symmetric positive definite matrix, then it has a unique factorization $A = LL^T$ where L is lower triangular and has a positive diagonal.*

Example 2.3. The matrix

$$\begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

has a Cholesky factorization of the form

$$A = LDL^T = (LD^{\frac{1}{2}})(D^{\frac{1}{2}}L^T) = \tilde{L}\tilde{L^T}$$

where

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{3}{4} & 1 & 0 & 0 \\ \frac{1}{2} & \frac{2}{3} & 1 & 0 \\ \frac{1}{4} & \frac{1}{3} & \frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & \frac{1}{2}\sqrt{3} & 0 & 0 \\ 0 & 0 & \sqrt{\frac{2}{3}} & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & 0 \\ \frac{3}{2} & \frac{1}{2}\sqrt{3} & 0 & 0 \\ 1 & \frac{1}{3}\sqrt{3} & \sqrt{\frac{2}{3}} & 0 \\ \frac{1}{2} & \frac{1}{6}\sqrt{3} & \frac{1}{2}\sqrt{\frac{2}{3}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

Theorem 2.3. *If a matrix is symmetric and has all positive eigenvalues if and only if it is positive definite.*

Proof. Let A be a positive definite matrix. Let zero be an eigenvalue, then there exists an eigenvector x such that $Ax = 0$. But then $x^TAx = 0$ so A is not positive definite. Let λ be an eigenvalue such that $\lambda < 0$, then there is a vector x such that $Ax = \lambda x$ where $x^TAx = \lambda|x^2|$. But $\lambda|x^2|$ is negative since λ is negative and $|x^2| > 0$ so A is not positive definite. By contradiction, if a matrix is positive definite, it has all positive eigenvalues.
Let A be a symmetric matrix with all positve eigenvalues λ. Then there exists a matrix factorization such that $A = QDQ^T$ where D is a diagonal matrix. Then we have

$$x^TAx = x^T(Q^TDQ)x = (x^TQ^T)D(Qx) = (Qx)^TD(Qx) = \sum_{i=1}^{n} \lambda_i(q_i x_i^2) > 0$$

Thus A is a positive symmetric matrix. □

We can implement this theorem in Python to check if a matrix is symmetric, and then positive definite. If so, it can be factorized in Cholesky form.

```
import numpy as np

# Return true if matrix is symmetric
def symmetricCheck(A):
    return (A==A.T).all()

# Return true if all eigenvals are positive
def isSymmetricPositiveDefinite(A):
    if(symmetricCheck(A)):
        return np.all(np.linalg.eigvals(A) > 0)
```

3 Homework

Homework 1

Problem 3.1. Solve the following linear system with five digits of rounding precision

$$\begin{cases} 0.1036x + 0.2122y = 0.7381 \\ 0.2081x + 0.4247y = 0.9327 \end{cases}$$

Solution 3.1. Begin with the system

$$\begin{cases} 0.1036x + 0.2122y = 0.7381 \\ 0.2081x + 0.4247y = 0.9327 \end{cases}$$

Seeking an equation of 1 variable, we multiply the second equation by a multiple of the first to eliminate x. In this case, the multiple is $\frac{0.2081}{0.1036}$, which

gives us

$$\begin{cases} 0.1036x + & 0.2122y = 0.7381 \\ & 0.4247y - \frac{0.2081}{0.1036}0.2122y = 0.9327 - \frac{0.2081}{0.1036}0.7381 \end{cases}$$

from which we solve the first equation for y and substitute it into the second to obtain

$$0.4247y - 2.0087 \times 0.2122y = 0.9327 - 2.0087 \times 0.7381$$

$$-0.0015461y = -0.54992$$

$$y = 355.68$$

Problem 3.2. Solve the linear system by Gaussian Elimination without pivoting, using the matrix form

$$\begin{cases} x_1 + 5x_2 + 2x_3 = 1 \\ 3x_1 + 6x_2 + 2x_3 = -1 \\ -2x_1 + 2x_2 + x_3 = 3 \end{cases}$$

Solution 3.2. We begin by finding the lower-upper decomposition of the matrix

$$A\vec{x} = \begin{pmatrix} 1 & 5 & 2 \\ 3 & 6 & 2 \\ -2 & 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

First, we take $R_2 = R_2 - 3R_1$ to obtain

$$\begin{pmatrix} 1 & 5 & 2 \\ 0 & -9 & -4 \\ -2 & 2 & 1 \end{pmatrix}$$

Then we take $R_3 = R_3 + 2R_1$ to obtain

$$\begin{pmatrix} 1 & 5 & 2 \\ 0 & -9 & -4 \\ 0 & 12 & 5 \end{pmatrix}$$

Next, we have $R_3 = R_3 + \frac{4R_2}{3}$ which gives us

$$U = \begin{pmatrix} 1 & 5 & 2 \\ 0 & -9 & -4 \\ 0 & 0 & -\frac{1}{3} \end{pmatrix}$$

The L matrix is given by the coeffiecients of A corresponding to the zero-entries of U

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

To solve the system, we need to solve

$$LU\vec{x} = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -2 & -\frac{4}{3} & 1 \end{pmatrix} \begin{pmatrix} 1 & 5 & 2 \\ 0 & -9 & -4 \\ 0 & 0 & -\frac{1}{3} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix}$$

Let

$$\begin{pmatrix} 1 & 5 & 2 \\ 0 & -9 & -4 \\ 0 & 0 & -\frac{1}{3} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

So that

$$\begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -2 & -\frac{4}{3} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix}$$

From this we conclude that

$$y_1 = 1 \implies y_2 = -1 - 3(1) = -4 \implies y_3 = 3 + 2(1) + (\frac{4}{3})(-4) = -\frac{1}{3}$$

Now let

$$\begin{pmatrix} 1 & 5 & 2 \\ 0 & -9 & -4 \\ 0 & 0 & -\frac{1}{3} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -4 \\ -\frac{1}{3} \end{pmatrix}$$

So that

$$x_3 = -1 \implies -9x_2 - 4(1) = -4 \implies x_2 = 0$$

$$\implies x_1 = 1 + 2(-1) = -1 \implies x_1 = -1$$

The values $x_1 = -1, x_2 = 0, x_3 = 1$ correspond to the solutions to the original system.

Problem 3.3. Solve the linear system by Gaussian Elimination with partial pivoting, using the matrix form

$$\begin{cases} x_1 + 5x_2 + 2x_3 = 1 \\ 3x_1 + 6x_2 + 2x_3 = -1 \\ -2x_1 + 2x_2 + x_3 = 3 \end{cases}$$

Solution 3.3. We begin with the matrix

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

First, we swap R_1 and R_2 to obtain

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

This is reflected in our permutation matrix P by

$$P_1 A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 5 & 2 \\ 3 & 6 & 2 \\ -2 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 6 & 2 \\ 1 & 5 & 2 \\ -2 & 2 & 1 \end{pmatrix}$$

Next, we have $R_2 = -\frac{1}{3}R_1 + R_2$ and $R_3 = -\frac{2}{3}R_1 + R_3$ from which the product becomes

$$P_2 A = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{2}{3} & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 & 6 & 2 \\ 1 & 5 & 2 \\ -2 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 5 & 2 \\ 0 & 3 & \frac{4}{3} \\ 0 & 6 & \frac{7}{3} \end{pmatrix}$$

Now, $R_3 = 2R_2 - R_3$ gives

$$U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 & 5 & 2 \\ 0 & 6 & \frac{7}{3} \\ 0 & 0 & \frac{1}{6} \end{pmatrix} = \begin{pmatrix} 3 & 6 & 2 \\ 0 & 3 & \frac{4}{3} \\ 0 & 0 & \frac{1}{3} \end{pmatrix}$$

Thus we get the relation

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

As found previously, this has solutions $x_1 = -1, x_2 = 0, x_3 = 1$. Alternatively, the problem can be solved in Python

```
# Adapted from https://docs.scipy.org/
from scipy.linalg import lu_factor, lu_solve
import numpy as np

A = np.array([1, 5, 2, 3, 6, 2, -2, 2, 1])
b = np.array([1,-1,3])
m = 3
```

```
n = 3

def linSysLU(A, b, m, n):
    A = A.reshape(m, n)
    lu, piv = lu_factor(A)

    # Solution to system
    x = lu_solve((lu, piv), b)
    return x
```

Homework 2

Problem 3.4. Prove that the matrix $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ is symmetric and positive definite.

Solution 3.4. Let $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$. By swapping the rows and columns of A, we obtain

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = A^t.$$

Since the transpose of A is A, it is a symmetric matrix. Now, observe that subtracting λ from the diagonals of A, the eigenvalues are given by $\lambda_1 = 3$ and $\lambda_2 = 1$. Since A is symmetric and has all positive eigenvalues, A is a positive definite matrix.

Problem 3.5.

Find the Cholesky factorization of $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$.

Solution 3.5. Consider the lower diagonal matrix given by

$$L = \begin{pmatrix} \sqrt{2} & 0 \\ \frac{1}{\sqrt{2}} & \sqrt{\frac{3}{2}} \end{pmatrix},$$

whose diagonal is positive and whose transpose is given by

$$L^T = \begin{pmatrix} \sqrt{2} & \frac{1}{\sqrt{2}} \\ 0 & \sqrt{\frac{3}{2}} \end{pmatrix}.$$

We have

$$LL^T = \begin{pmatrix} \sqrt{2} & 0 \\ \frac{1}{\sqrt{2}} & \sqrt{\frac{3}{2}} \end{pmatrix} \begin{pmatrix} \sqrt{2} & \frac{1}{\sqrt{2}} \\ 0 & \sqrt{\frac{3}{2}} \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = A,$$

which is in the form of a Cholesky factorization. This can also be solved in Python:

```
# Cholesky factorization in Python
from scipy.linalg import cholesky
import numpy as np

A = np.array([[2, 1], [1, 2]])
def CholeskyFactorization(A):
    L = cholesky(A, lower = True)
    Lt = L.T.conj()
    return(L,Lt)
```

Problem 3.6. For the matrix

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

find all the eigenvalues and their corresponding eigenspaces. Additionally, find the algebraic and geometric multiplicity of each eigenvalue.

Solution 3.6. We begin by finding the eigenvalues. Seeking the character-

istic polynomial, we compute $A - \lambda I$. Now,

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

Now we need to find λ such that $\det(A - \lambda I) = \vec{0}$,

$$\det(A-\lambda I) = (0)(-1)^4 \begin{vmatrix} -4 & 3-\lambda \\ 1 & 0 \end{vmatrix} + 0(-1)^5 \begin{vmatrix} -\lambda - 1 & 1 \\ 1 & 0 \end{vmatrix} + (2-\lambda)(-1)^6 \begin{vmatrix} -\lambda - 1 & 1 \\ -4 & 3-\lambda \end{vmatrix}$$

$$= (2-\lambda) \begin{vmatrix} -\lambda - 1 & 1 \\ -4 & 3-\lambda \end{vmatrix} = (-\lambda-1)(3-\lambda)-1(-4) = \lambda^2-2\lambda+1 = -(\lambda-2)(\lambda-1)^2.$$

The roots of the characteristic polynomial yield the eigenvalues for A so $\lambda_1 = 2, \lambda_2 = 1$. The term $(\lambda-1)^2$ indicates that the eigenvalue λ_2 has an algebraic multiplicity of 2, by similar argument, λ_1 has an algebraic multiplicity of 1. Substituting our eigenvalues into $A - \lambda I$, we get

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

which has a null space of $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ and for the eigenvalue 1,

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

which has a null space of $\begin{pmatrix} -1 \\ -2 \\ 1 \end{pmatrix}$. Since these vectors consist of the null space of $A - \lambda I$, these are the eigenvectors of their associated eigenvalues. Since both null spaces are 1-dimensional, the geometric multiplicity of both λ_1 and λ_2 is 1.

This can be done in Python with the following:

```
# eigenvals in Python
from numpy import linalg as LA
A = np.array([[-1, 1, 0], [-4, 3, 0], [1, 0, 2]])
# eigenvals are repeated according to multiplicity
def Eigenvalues(A):
    eigenvals, eigenvecs = LA.eig(A)
    return eigenvals, eigenvecs
```

Definitions

absolute error, 3

Cholesky factorization, 10
chopped, 4

Floating-Point Limit Values, 7

Normalized Scientific Notation, 2

relative error, 3

Significant Digits, 2