NLA Homework 3

Group O

Dafu Zhu, Jiawei Ni

Question 1

The LU decomposition with row pivoting of the matrix

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

is given by $PA = LU$, where

$$P = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}; \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -0.5 & 0.25 & 1 & 0 \\ 0.5 & 0.25 & 0 & 1 \end{pmatrix};$$
$$U = \begin{pmatrix} 4 & -1 & 0 & 1 \\ 0 & -2 & 0 & 1 \\ 0 & 0 & 1 & -0.75 \\ 0 & 0 & 0 & 0.25 \end{pmatrix}$$

(i) Solve $Ax = b$, where

$$b = \begin{pmatrix} 3 \\ -1 \\ 0 \\ 2 \end{pmatrix}$$

Solution

In [1]:
```python
import numpy as np
import pandas as pd
import copy
import sys
import os
sys.path.append(os.path.abspath(os.path.join(os.getcwd(), "..")))
from nla import func # my own package based on previous homeworks
np.set_printoptions(suppress=True, precision=6)

def lu_row_pivoting(A): #@save
    """
    LU decomposition with row pivoting
    input: A(np.array)
    output: P, L, U
    """
    AA = copy.deepcopy(A).astype(float)  # precision issue
                                         # debugged for a long time
    n = AA.shape[0]
    # initialize
    P, L = np.eye(n), np.eye(n)
    U = np.eye(n)
    for i in range(0, n-1):
        i_max = np.argmax(np.abs(AA[i:n, i])) + i
        # switch rows i and i_max of A
        vv = copy.deepcopy(AA[i, i:n])
        AA[i, i:n] = AA[i_max, i:n]
        AA[i_max, i:n] = vv
        # update matrix P
        cc = copy.deepcopy(P[i])
        P[i] = P[i_max]
        P[i_max] = cc
        if i > 0:
            ww = copy.deepcopy(L[i, 0:i])
            L[i, 0:i] = L[i_max, 0:i]
            L[i_max, 0:i] = ww
        for j in range(i, n):
            L[j, i] = AA[j, i] / AA[i, i]
            U[i, j] = AA[i, j]
        for j in range(i+1, n):
            for k in range(i+1, n):
                AA[j, k] = AA[j, k] - (L[j, i] * U[i, k])
    L[n-1, n-1] = 1
    U[n-1, n-1] = AA[n-1, n-1]
    return P, L, U
```

In [3]:
```python
def linear_solve_lu_row_pivoting(A, b): #@save
    """
    Linear solver using LU decomposition with row pivoting
    input: A(np.array), b(np.array)
    output: x
    """
    P, L, U = lu_row_pivoting(A)
    y = func.forward_subset(L, P@b)
    x = func.backward_subset(U, y)
    return x
```

In [5]:
```python
A = np.array([
    [2,-1,0,1],
    [-2,0,1,-1],
    [4,-1,0,1],
    [4,-3,0,2]
])
b = np.array([3,-1,0,2])
x = linear_solve_lu_row_pivoting(A, b)
x
```

Out[5]:
```
[-1.5, 4.0, 6.0, 10.0]
```

$$\therefore \quad x = \begin{pmatrix} -1.5 \\ 4 \\ 6 \\ 10 \end{pmatrix}$$

(ii) Find A^{-1}, the inverse matrix of A.

Solution

In [7]:
```python
def system_solve_lu_row_pivoting(A, B): #@save
    """
    input:
    - A(np.array): nonsingular square matrix
    - B(np.array): col vectors of size n, [b_1, b_2, ..., b_p]
    output:
    - X: solution to Ax_i=b_i, [x_1, ..., x_p]
    """
    P, L, U = lu_row_pivoting(A)
    p = B.shape[1]
    X = np.zeros((B.shape[0], B.shape[1]))
    for i in range(p):
        b_i = B[:, i]
        y = func.forward_subset(L, P@b_i)
        x_i = func.backward_subset(U, y)
        X[:, i] = x_i
    return X
```

Solving A^{-1} is equivalent to finding X, where $AX = B, B = I$. Write I, X as $I = \text{col}(e_k)_{k=1:p}$, $X = \text{col}(x_k)_{k=1:p}$

In [9]:
```python
B = np.eye(A.shape[0])
X = system_solve_lu_row_pivoting(A, B)
X
```

Out[9]:
```
array([[-0.5,  0. ,  0.5,  0. ],
       [ 2. , -0. , -0. , -1. ],
       [ 3. ,  1. ,  0. , -1. ],
       [ 4. ,  0. , -1. , -1. ]])
```

$$\therefore \quad A^{-1} = \begin{pmatrix} -0.5 & 0 & 0.5 & 0 \\ 2 & 0 & 0 & -1 \\ 3 & 1 & 0 & -1 \\ 4 & 0 & -1 & -1 \end{pmatrix}$$

Question 2

The following discount factors are obtained by fitting market data:

Date	Discount Factor
2 months	0.9980
5 months	0.9935
11 months	0.9820
15 months	0.9775

The overnight rate is 1%.

(i) What is the linear system that has to be solved for the cubic spline interpolation of the zero rate curve?

Solution

Find out the zero rates by discount factors

$$\text{Disc}(t) = \exp(-t \cdot r(0,t))$$

Therefore

$$r(0,t) = -\frac{1}{t}\ln(\text{Disc}(t))$$

In [34]:
```python
disc = np.array([0.998,0.9935,0.982,0.9775])
t = np.array([2/12,5/12,11/12,15/12])
r = -1/t * np.log(disc)
r
```

Out[34]:
```
array([0.012012, 0.015651, 0.019815, 0.018206])
```

Together with the overnight rate of 0.01, i.e. $r(0,0) = 0.01$, we have

$$v = [0.01, 0.012012, 0.015651, 0.019815, 0.018206]$$

In [37]:
```python
x = np.array([0, 2/12, 5/12, 11/12, 15/12])
v = np.array([0.01, 0.012012, 0.015651, 0.019815, 0.018206])
```

$f(x) = f_i(x) = a_i + b_i x + c_i x^2 + d_i x^3$ for $i = 1:n$

Constraints:
$$\begin{cases} f_i(x_{i-1}) = v_{i-1} & i=1:n \\ f_i(x_i) = v_i & i=1:n \\ f_i'(x_i) = f_{i+1}'(x_i) & i=1:n-1 \\ f_i''(x_i) = f_{i+1}''(x_i) & i=1:n-1 \\ f_1''(x_0) = 0 & \\ f_n''(x_n) = 0 & \end{cases}$$

Implement the above constraints in Python code.

In [39]:
```python
def cubic_spline_interpolate(x, v): #@save
    """
    input:
    - x: interpolation nodes, i=0:n
    - v: interpolation values, i=0:n
    output:
    - b_bar, M_bar: linear system
    - coef: list([[a1,b1,c1,d1],[a2,b2,...],...])
    """
    n = len(x) - 1
    coef = []
    # compute vector b_bar
    b_bar = np.zeros(4*n)
    b_bar[0], b_bar[4*n-1] = 0, 0
    for i in range(1, n+1):
        b_bar[4*i-3] = v[i-1]
        b_bar[4*i-2] = v[i]
```

```python
    for i in range(1, n):
        b_bar[4*i-1] = 0
        b_bar[4*i] = 0
    # compute matrix M_bar
    M_bar = np.zeros((4*n, 4*n))
    M_bar[0,2], M_bar[0,3] = 2, 6*x[0]
    M_bar[4*n-1,4*n-2], M_bar[4*n-1,4*n-1] = 2, 6*x[-1]
    for i in range(1, n+1):
        # f(x)
        M_bar[4*i-3,4*i-4] = 1
        M_bar[4*i-3,4*i-3] = x[i-1]
        M_bar[4*i-3,4*i-2] = x[i-1] ** 2
        M_bar[4*i-3,4*i-1] = x[i-1] ** 3
        M_bar[4*i-2,4*i-4] = 1
        M_bar[4*i-2,4*i-3] = x[i]
        M_bar[4*i-2,4*i-2] = x[i] ** 2
        M_bar[4*i-2,4*i-1] = x[i] ** 3
    for i in range(1, n):
        # f'(x)
        M_bar[4*i-1,4*i-3] = 1
        M_bar[4*i-1,4*i-2] = 2 * x[i]
        M_bar[4*i-1,4*i-1] = 3 * (x[i] ** 2)
        M_bar[4*i-1,4*i+1] = -1
        M_bar[4*i-1,4*i+2] = -2 * x[i]
        M_bar[4*i-1,4*i+3] = -3 * (x[i] ** 2)
        # f''(x)
        M_bar[4*i,4*i-2] = 2
        M_bar[4*i,4*i-1] = 6 * x[i]
        M_bar[4*i,4*i+2] = -2
        M_bar[4*i,4*i+3] = -6 * x[i]
    x_bar = linear_solve_lu_row_pivoting(M_bar, b_bar)
    for i in range(1,n+1):
        coef.append(x_bar[4*i-4:4*i])
    return b_bar, M_bar, coef
```

In [41]:
```python
b_bar, M_bar, coef = cubic_spline_interpolate(x, v)
```

In [43]:
```python
pd.DataFrame(M_bar)
```

Out[43]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0.0	0.000000	2.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000
1	1.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000
2	1.0	0.166667	0.027778	0.004630	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000
3	0.0	1.000000	0.333333	0.083333	0.0	-1.000000	-0.333333	-0.083333	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000
4	0.0	0.000000	2.000000	1.000000	0.0	0.000000	-2.000000	-1.000000	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000
5	0.0	0.000000	0.000000	0.000000	1.0	0.166667	0.027778	0.004630	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000
6	0.0	0.000000	0.000000	0.000000	1.0	0.416667	0.173611	0.072338	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000
7	0.0	0.000000	0.000000	0.000000	0.0	1.000000	0.833333	0.520833	0.0	-1.000000	-0.833333	-0.520833	0.0	0.000000	0.000000	0.000000
8	0.0	0.000000	0.000000	0.000000	0.0	0.000000	2.000000	2.500000	0.0	0.000000	-2.000000	-2.500000	0.0	0.000000	0.000000	0.000000
9	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	1.0	0.416667	0.173611	0.072338	0.0	0.000000	0.000000	0.000000
10	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	1.0	0.916667	0.840278	0.770255	0.0	0.000000	0.000000	0.000000
11	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.0	1.000000	1.833333	2.520833	0.0	-1.000000	-1.833333	-2.520833
12	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.0	0.000000	2.000000	5.500000	0.0	0.000000	-2.000000	-5.500000
13	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	1.0	0.916667	0.840278	0.770255
14	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	1.0	1.250000	1.562500	1.953125
15	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.0	0.000000	2.000000	7.500000

In [45]:
```python
pd.DataFrame(b_bar)
```

Out[45]:

	0
0	0.000000
1	0.010000
2	0.012012
3	0.000000
4	0.000000
5	0.012012
6	0.015651
7	0.000000
8	0.000000
9	0.015651
10	0.019815
11	0.000000
12	0.000000
13	0.019815
14	0.018206
15	0.000000

$$\overline{M} =$$

$$\begin{pmatrix}
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0.166667 & 0.027778 & 0.00463 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0.333333 & 0.083333 & 0 & -1 & -0.333333 & -0.083333 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 1 & 0 & 0 & -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0.166667 & 0.027778 & 0.00463 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0.416667 & 0.173611 & 0.072338 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0.833333 & 0.520833 & 0 & -1 & -0.833333 & -0.520833 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 2.5 & 0 & 0 & -2 & -2.5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.416667 & 0.173611 & 0.072338 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.916667 & 0.840278 & 0.770255 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1.833333 & 2.520833 & 0 & -1 & -1.833333 & -2.520833 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5.5 & 0 & 0 & -2 & -5.5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.916667 & 0.840278 & 0.770255 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1.25 & 1.5625 & 1.953125 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 7.5
\end{pmatrix}$$

$$\bar{b} = \begin{pmatrix} 0 \\ 0.01 \\ 0.012012 \\ 0 \\ 0 \\ 0.012012 \\ 0.015651 \\ 0 \\ 0 \\ 0.015651 \\ 0.019815 \\ 0 \\ 0 \\ 0.019815 \\ 0.018206 \\ 0 \end{pmatrix}, \quad \bar{x} = \begin{pmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \\ a_2 \\ b_2 \\ c_2 \\ d_2 \\ a_3 \\ b_3 \\ c_3 \\ d_3 \\ a_4 \\ b_4 \\ c_4 \\ d_4 \end{pmatrix}$$

The linear system to solve is

$$\overline{\overline{M}}\bar{x} = \bar{b}$$

(ii) Use cubic spline interpolation to find a zero rate curve for all times less than 15 months matching the discount factors above.

Solution

In [47]:
```
coef
```

Out[47]:
```
[[0.01, 0.011456, 0.0, 0.022173],
 [0.010215, 0.007585, 0.023228, -0.024283],
 [0.009153, 0.015227, 0.004886, -0.00961],
 [-0.01484, 0.09375, -0.080775, 0.02154]]
```

$$r(0,t) = \begin{cases} 0.01 + 0.011456t + 0.022173t^3 & \text{if } 0 \le t \le \frac{2}{12} \\ 0.010215 + 0.007585t + 0.023228t^2 + -0.024283t^3 & \text{if } \frac{2}{12} \le t \le \frac{5}{12} \\ 0.009153 + 0.015227t + 0.004886t^2 + -0.00961t^3 & \text{if } \frac{5}{12} \le t \le \frac{11}{12} \\ -0.01484 + 0.09375t + -0.080775t^2 + 0.02154t^3 & \text{if } \frac{11}{12} \le t \le \frac{15}{12} \end{cases}$$

(iii) Find the value of a 13 months quarterly bond with 2.5% coupon rate.

Note: A quarterly coupon bond with face value $\$100$, coupon rate C, and maturity T pays the holder of the bond a coupon payment equal to $\frac{C}{4} \cdot 100$ every three months, except at maturity. The final payment at maturity T is equal to the face value of the bond plus one coupon payment, i.e., $100 + \frac{C}{4}100$.

Solution

Coupon = $\frac{1}{4} \times 100 \times 2.5\% = 0.625$

Cash flow

$0.625 in 1 month

$0.625 in 4 month

$0.625 in 7 month

$0.625 in 10 month

$100.625 in 13 month

Bond price should be

$$B = 0.625 \times Disc(\tfrac{1}{12}) + 0.625 \times Disc(\tfrac{4}{12}) + 0.625 \times Disc(\tfrac{7}{12}) + 0.625 \times Disc(\tfrac{10}{12}) + 100.625 \times Disc(\tfrac{13}{12})$$

Calculate Disc(·) by the zero rate curve, with

$$Disc(t) = \exp(-t \cdot r(0,t))$$

In [49]:
```
def zero_rate_curve(t, coef, x):
    """
    input:
    - t(float): time
    - coef(list): the result of cubic_spline_interpolate
    - x(list): given time nodes
    Note: len(x) should be 1 larger than len(coef)
    output:
    - corresponding zero rate at time t
    """
    assert len(coef) == len(x) - 1, "size not match"
    def _zero_rate(i, t, coef):
        i_coef = np.array(coef[i])
        tt = [1,t,t**2,t**3]
        return np.sum(i_coef*tt)
    for i in range(len(coef)):
        if x[i] <= t <= x[i+1]:
            return _zero_rate(i, t, coef)
```

In [51]:
```
coupon = [0.625,0.625,0.625,0.625,100.625]
dates = [1/12, 4/12, 7/12, 10/12, 13/12]
B = 0
for i in range(len(coupon)):
    t = dates[i]
    disc_t = np.exp(-t*zero_rate_curve(t, coef, x))
    B += coupon[i] * disc_t
B.round(6)
```

Out[51]:
```
101.021659
```

$\therefore$ the value of a 13 months quarterly bond with 2.5% coupon rate is 101.021659

Question 3

The values of the following coupon bonds with face value $100 are given:

Bond Type	Coupon Rate	Bond Price
10 months semiannual	3\%	$101.30
16 months semiannual	4\%	$102.95
22 months annual	6\%	$107.35
22 months semiannual	5\%	$105.45

(i) List the cash flows and cash flow dates for each bond.

Bond Type	Cash flow and dates
10 mos semi	$1.5 in 4 mos $101.5 in 10 mos
16 mos semi	$2 in 4 mos $2 in 10 mos $102 in 16 mos
22 mos annual	$6 in 10 mos $106 in 22 mos
22 mos semi	$2.5 in 4 mos $2.5 in 10 mos $2.5 in 16 mos $102.5 in 22 mos

(ii) Identify the matrix and the right hand side vector corresponding to the linear system whose solution are the 4 months, 10 months, 16 months, and 22 months discount factors.

Denote the discount factors of 4 mos, 10 mos, 16 mos and 22 mos as d_1, d_2, d_3, d_4, respectively.

$$101.3 = 1.5d_1 + 101.5d_2$$
$$102.95 = 2d_1 + 2d_2 + 102d_3$$
$$107.35 = 6d_2 + 106d_4$$
$$105.45 = 2.5d_1 + 2.5d_2 + 2.5d_3 + 102.5d_4$$

Matrix
$$\begin{pmatrix} 1.5 & 101.5 & 0 & 0 \\ 2 & 2 & 102 & 0 \\ 0 & 6 & 0 & 106 \\ 2.5 & 2.5 & 2.5 & 102.5 \end{pmatrix}$$

RHS vector
$$\begin{pmatrix} 101.3 \\ 102.95 \\ 107.35 \\ 105.45 \end{pmatrix}$$

(iii) Find the 4 months, 10 months, 16 months, and 22 months discount factors.

In [53]:
```
M = np.array([
    [1.5,101.5,0,0],
    [2,2,102,0],
    [0,6,0,106],
    [2.5,2.5,2.5,102.5]
])
b = np.array([101.3,102.95,107.35,105.45])
x = linear_solve_lu_row_pivoting(M, b)
x
```

Out[53]:
```
[0.98604, 0.983458, 0.970696, 0.957068]
```

$$\therefore \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix} = \begin{pmatrix} 0.98604 \\ 0.983458 \\ 0.970696 \\ 0.957068 \end{pmatrix}$$

Question 4

Consider three assets with the following expected rates of return, standard deviations of their rates of return, and correlations of their rates of return:

$$\mu_1 = 0.1; \quad \sigma_1 = 0.15; \quad \rho_{1,2} = -0.25$$
$$\mu_2 = 0.15; \quad \sigma_2 = 0.3; \quad \rho_{2,3} = 0.2;$$
$$\mu_3 = 0.2; \quad \sigma_3 = 0.35; \quad \rho_{1,3} = 0.3$$

(i) Find the covariance matrix M of the rates of return of the three assets.

$$\sigma_{ij} = \sigma_i \sigma_j \rho_{ij}$$

$$\sigma_{12} = \sigma_1 \sigma_2 \rho_{12} = 0.15 \times 0.3 \times (-0.25) = -0.01125$$

$$\sigma_{23} = \sigma_2 \sigma_3 \rho_{23} = 0.3 \times 0.35 \times 0.2 = 0.021$$

$$\sigma_{13} = \sigma_1 \sigma_3 \rho_{13} = 0.15 \times 0.35 \times 0.3 = 0.01575$$

$$\sigma_{11} = \sigma_1^2 = 0.0225, \quad \sigma_{22} = \sigma_2^2 = 0.09, \quad \sigma_{33} = \sigma_3^2 = 0.1225$$

$$M = \begin{pmatrix} 0.0225 & -0.01125 & 0.01575 \\ -0.01125 & 0.09 & 0.021 \\ 0.01575 & 0.021 & 0.1225 \end{pmatrix}$$

(ii) A minimum variance portfolio with 16% expected rate of return can be set up by investing a percentage w_i of the total value of the portfolio in asset i, with $i = 1:3$, where w_i can be found by solving the following linear system:

$$\begin{pmatrix} 2M & \mathbf{1} & \mu \\ \mathbf{1}^t & 0 & 0 \\ \mu^t & 0 & 0 \end{pmatrix} \begin{pmatrix} w \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ \mu_P \end{pmatrix} \tag{1}$$

where $\mu_P = 0.16$,

$$\mu = \begin{pmatrix} 0.1 \\ 0.15 \\ 0.2 \end{pmatrix} \quad \text{and} \quad \mathbf{1} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

The matrices from the LU decomposition with row pivoting of the matrix on the left hand side of (1) are

$$P = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -0.0225 & 1 & 0 & 0 & 0 \\ 0.0315 & 0.051852 & 1 & 0 & 0 \\ 0.045 & -0.333333 & 0.038067 & 1 & 0 \\ 0.1 & 0.246914 & 0.400056 & -0.482738 & 1 \end{pmatrix};$$

$$U = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0.2025 & 0.0645 & 1 & 0.15 \\ 0 & 0 & 0.210555 & 0.948148 & 0.192222 \\ 0 & 0 & 0 & 1.297240 & 0.142683 \\ 0 & 0 & 0 & 0 & -0.045059 \end{pmatrix}$$

Find the weights of each asset in this minimum variance portfolio.

Solution:

$$2M = \begin{pmatrix} 0.045 & -0.0225 & 0.0315 \\ -0.0225 & 0.18 & 0.042 \\ 0.0315 & 0.042 & 0.245 \end{pmatrix}$$

The matrix on the left hand side of (1) is

$$\begin{pmatrix} 0.045 & -0.0225 & 0.0315 & 1 & 0.1 \\ -0.0225 & 0.18 & 0.042 & 1 & 0.15 \\ 0.0315 & 0.042 & 0.245 & 1 & 0.2 \\ 1 & 1 & 1 & 0 & 0 \\ 0.1 & 0.15 & 0.2 & 0 & 0 \end{pmatrix}$$

Now solve the linear system using `linear_solve_lu_row_pivoting`

In [55]:
```
A = np.array([
    [0.045 , -0.0225 ,  0.0315 , 1 ,0.1],
    [-0.0225 ,  0.18  ,  0.042 , 1 , 0.15],
    [0.0315 ,  0.042 , 0.245 , 1 , 0.2],
    [1,1,1,0,0],
    [0.1,0.15,0.2,0,0]
])
b = np.array([0,0,0,1,0.16])
x = linear_solve_lu_row_pivoting(A, b)
x
```

Out[55]:
```
[0.235075, 0.329849, 0.435076, 0.09413, -1.109912]
```

$$\therefore \quad \omega_1 = 0.235075, \omega_2 = 0.329849, \omega_3 = 0.435076$$

(iii) Compute the standard deviation of the returns of the following portfolios with 16% expected rate of return:

1. 30% invested in asset 1, 20% invested in asset 2, 50% invested in asset 3
2. 50% invested in asset 1, 70% invested in asset 3, and short an amount equal to 20% of the value of the portfolio of asset 2

Solution:

Denote the returns of asset 1, 2 and 3 as r_1, r_2, r_3, respectively

The return of the portfolio can be written as

$$r_p = \omega_1 r_1 + \omega_2 r_2 + \omega_3 r_3$$

The variance of portfolio's return is

$$Var(r_p) = \omega^T M \omega$$

In [57]:
```python
def pf_std(w, Sigma):
    """
    Calculate the standard deviation of portfolio return
    input:
    - w(np.array): weight vector
    - Sigma(np.array): covariance matrix
    """
    var = w.T @ Sigma @ w
    return np.sqrt(var).round(6)
```

In [59]:
```python
M = np.array([
    [0.0225, -0.01125, 0.01575],
    [-0.01125, 0.09, 0.021],
    [0.01575, 0.021, 0.1225]
])
w_1 = np.array([0.3,0.2,0.5])
w_2 = np.array([0.5,-0.2,0.7])
```

In [61]:
```python
pf_std(w_1, M)
```

Out[61]:
```
0.209344
```

In [63]:
```python
pf_std(w_2, M)
```

Out[63]:
```
0.276848
```

1. $\sigma_{p_1} = 0.209344$
2. $\sigma_{p_2} = 0.276848$

Eigenvalues and eigenvectors

Question 1

Let A and B be square matrices of the same size. Show that if v is an eigenvector of both A and B, then v is also an eigenvector of the matrix

$$M = c_1 A + c_2 B$$

where c_1 and c_2 are constants. What is the eigenvalue of M corresponding to the eigenvector v ?

Denote λ_1, λ_2 as the eigenvalues of A and B, corresponding to v.

$Av = \lambda_1 v, \quad Bv = \lambda_2 v$

$Mv = c_1 A + c_2 B = (c_1\lambda_1 + c_2\lambda_2) v$

$\therefore$ v is also an eigenvector of M.

with eigenvalue $c_1\lambda_1 + c_2\lambda_2$

Question 2

Let A be a square matrix such that $A^2 = A$. Show that any eigenvalue of A is either 0 or 1.

Note: A matrix A with the property that $A^2 = A$ is called an idempotent matrix.

Assume $\exists$ eigenvector x and eigenvalue λ. s.t.

$Ax = \lambda x$

Since $A^2 = A$

$A^2 x = A \cdot (Ax) = A \cdot (\lambda x) = \lambda \cdot (Ax) = \lambda^2 x$

$Ax = \lambda^2 x$

$\therefore \lambda = \lambda^2 \Rightarrow \lambda(\lambda - 1) = 0 \quad \therefore \lambda_1 = 0, \lambda_2 = 1 \quad \square.$

Question 3

Let A be a square matrix with the property that there exists a positive integer n such that $A^n = 0$. Show that any eigenvalue of A must be equal to 0 .

Note: A matrix A with the property that $A^n = 0$ for a positive integer n is called a nilpotent matrix.

Consider eigenvector x and eigenvalue λ

$Ax = \lambda x \quad$ where $x \neq 0$

$A^n x = A^{n-1}(Ax) = \lambda A^{n-1} x$

Repeat it ...

$A^n x = \lambda^n x = 0$

$\because x \neq 0 \quad \therefore \lambda^n = 0, \ \lambda = 0$

Question 4

Let v be a column vector of size n, and let $A = vv^t$ be an $n \times n$ matrix.

(i) How many non-zero eigenvalues does the matrix A have?

(ii) What are the eigenvalues of A, and what are the corresponding eigenvectors?

1. Since A is constructed by the outer product of v.

$\mathrm{rank}(A) = 1$.

So A has only 1 nonzero eigenvalue if v is nonzero.

If v is zero vector, A has no nonzero eigenvalues

2. If $v = 0$, eigenvalues of A are all 0.

If $v \neq 0$, the only nonzero eigenvalue of A is

$$\mathrm{tr}(A) = \sum_{i=1}^{n} A_{ii} = \sum_{i=1}^{n} (vv^T)_{ii} = \sum_{i=1}^{n} v_i^2 = \|v\|^2$$

while all other eigenvalues $= 0$.

Question 5

Find the eigenvalues and the eigenvectors of the $n \times n$ matrix

$$A = \begin{pmatrix} d & 1 & \dots & 1 \\ 1 & d & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & d \end{pmatrix}$$

where $d \in \mathbb{R}$ is a constant.

Express $A = (d-1)I + J$, where J is matrix of all ones,

Since J is a rank-1 matrix, it has a nonzero eigenvalue n,

and $(n-1)$ zero eigenvalues

Eigenvalues of I are all ones.

So A has eigenvalue $(d-1+n)$ with multiplicity 1,

and $(d-1)$ with multiplicity $n-1$.

The eigenvector correspond to $d+n-1$ is $\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$

$Ax = \lambda x$

$$x_1 \begin{pmatrix} d \\ 1 \\ \vdots \\ 1 \end{pmatrix} + \dots + x_n \begin{pmatrix} 1 \\ \vdots \\ 1 \\ d \end{pmatrix} = \sum_{i=1}^{n} x_i \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} + (d-1) \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

$$\sum_{i=1}^{n} x_i \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = (\lambda - d + 1) \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

$\therefore \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ is a multiple of $\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$

The eigenvectors correspond to $d-1$ are vectors orthogonal to $\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$

$$(\underbrace{A - (d-1)I}_{\text{all one matrix}})\, x = 0$$

Question 6

Let λ and v be an eigenvalue and the corresponding eigenvector of the square matrix A of size n. Let S be a nonsingular matrix of size n. Show that λ is also an eigenvalue of the matrix $S^{-1}AS$. What is the corresponding eigenvector?

$Av = \lambda v$

$S^{-1}Av = \lambda S^{-1} v$

Let $w = S^{-1} v$, such that $v = Sw$

$S^{-1}ASw = \lambda S^{-1} S w = \lambda w$

$\therefore$ eigenvector is $S^{-1}v$

Question 7

Let A be a square matrix with real entries. If $\lambda = a + ib$ is a complex eigenvalue of A (i.e., with $b \neq 0$), show that $\bar{\lambda} = a - ib$, the complex conjugate of λ, is also an eigenvalue of A.

Since A is a matrix with all real entries

then $A = \bar{A}$

$Ax = \lambda x$, take conjugate on both sides.

$\bar{A}\bar{x} = \bar{\lambda}\bar{x} \Rightarrow A\bar{x} = \bar{\lambda}\bar{x}$

So $\bar{\lambda}$ is also an eigenvalue of A, with eigenvector $\bar{x}$

Question 8

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

(i) Compute the eigenvalues and the eigenvectors of the matrix A.

(ii) What is the diagonal form of A ?

(iii) Compute A^{12}.

(i) $\det(A - tI) = \begin{vmatrix} -1-t & 2 \\ 2 & 2-t \end{vmatrix} = (1+t)(t-2) - 4$
$= t^2 - t - 6 = 0$

$\therefore (t-3)(t+2) = 0$, $t_1 = 3, t_2 = -2$

For eigenvalue $= 3$,

$\begin{pmatrix} -4 & 2 \\ 2 & -1 \end{pmatrix} x_1 = 0 \Rightarrow x_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

For eigenvalue $= -2$

$\begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix} x_2 = 0 \Rightarrow x_2 = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$

So eigenvalues and eigenvectors are

$\lambda_1 = 3$, $x_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

$\lambda_2 = -2$, $x_2 = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$

(ii) $A = V\Lambda V^{-1} = \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 0 & -2 \end{pmatrix} \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix}^{-1}$

$V^{-1} = \frac{1}{1-(-4)} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} = \begin{pmatrix} 1/5 & 2/5 \\ -2/5 & 1/5 \end{pmatrix}$

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

(iii) $A^{12} = V\Lambda^{12}V^{-1}$

$= \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 0 & -2 \end{pmatrix}^{12} \begin{pmatrix} 1/5 & 2/5 \\ -2/5 & 1/5 \end{pmatrix}$

$= \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 3^{12} & 0 \\ 0 & 2^{12} \end{pmatrix} \begin{pmatrix} 1/5 & 2/5 \\ -2/5 & 1/5 \end{pmatrix}$

$= \begin{pmatrix} 109565 & 210938 \\ 210938 & 425972 \end{pmatrix}$

Question 9

Let $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ be a 2×2 matrix, and let

$$P_A(t) = t^2 - (a+d)t + (ad - bc)$$

be the characteristic polynomial associated to A.

Show that $P_A(A) = 0$, i.e., show that

$$A^2 - (a+d)A + (ad - bc)I = 0$$

Note: This is the 2×2 case of the Cayley-Hamilton theorem which states that $P_A(A) = 0$ for any square matrix A.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^2 = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a^2+bc & ab+bd \\ ac+cd & bc+d^2 \end{pmatrix}$$

$$A^2 - (a+d)A + (ad-bc)I$$

$$= \begin{pmatrix} a^2+bc & ab+bd \\ ac+cd & bc+d^2 \end{pmatrix} - \begin{pmatrix} a^2+ad & ab+bd \\ ac+cd & ad+d^2 \end{pmatrix} + \begin{pmatrix} ad-bc & 0 \\ 0 & ad-bc \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} = 0$$

Question 10

Let A be an $n \times n$ matrix given by

$$\begin{aligned} A(i,i) &= 2, \forall i = 1:n \\ A(i,i-1) &= 1, \forall i = 2:n \\ A(j,k) &= 0, \text{ otherwise.} \end{aligned}$$

Find the eigenvalues and the eigenvectors of A.

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

$$\det(A-\lambda I) = \begin{vmatrix} 2-\lambda & & & & \\ 1 & 2-\lambda & & & \\ & 1 & \ddots & & \\ & & \ddots & 2-\lambda & \\ & & & 1 & 2-\lambda \end{vmatrix}_n$$ This notates that it's a $n \times n$ matrix's determinant

Denote $$f(n) = \begin{vmatrix} 2-\lambda & & & \\ 1 & 2-\lambda & & \\ & 1 & \ddots & \\ & & 1 & 2-\lambda \end{vmatrix}_n$$

Then $f(n) = (2-\lambda) f(n-1)$, since determinant with a row of all zeros is 0.

$f(1) = (2-\lambda)$

$f(n) = (2-\lambda)^{n-1} \cdot (2-\lambda) = (2-\lambda)^n = 0$

$\therefore$ Eigenvalue of A is 2 with multiplicity of n.

$$(A-2I)v = 0 \Rightarrow \begin{pmatrix} 0 & & & \\ 1 & 0 & & \\ & 1 & \ddots & \\ & & 1 & 0 \end{pmatrix} v = 0 \Rightarrow v = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

Question 11

(i) Show that the eigenvalues of the matrix

$$A = \begin{pmatrix} -16 & 6 & -6 & 0 \\ -30 & 11 & -12 & 0 \\ 15 & -6 & 5 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

are -1 , with multiplicity 3 , and 2 , with multiplicity 1 .

Show that $\begin{pmatrix} 0 \\ 1 \\ 1 \\ -2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \\ -3 \\ 3 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$ are three linearly independent eigenvectors corresponding to the eigenvalue -1 , and show that there exists only one linearly independent eigenvector of the matrix A corresponding to the eigenvalue 2 , e.g., $\begin{pmatrix} 1 \\ 2 \\ -1 \\ 0 \end{pmatrix}$.

(i)

$$\det(A-\lambda I) = \begin{vmatrix} -16-\lambda & 6 & -6 & 0 \\ -30 & 11-\lambda & -12 & 0 \\ 15 & -6 & 5-\lambda & 0 \\ 0 & 0 & 0 & -1-\lambda \end{vmatrix}$$

$$= (-1-\lambda) \begin{vmatrix} -16-\lambda & 6 & -6 \\ -30 & 11-\lambda & -12 \\ 15 & -6 & 5-\lambda \end{vmatrix}$$

$$\begin{vmatrix} -16-\lambda & 6 & -6 \\ -30 & 11-\lambda & -12 \\ 15 & -6 & 5-\lambda \end{vmatrix} = (-16-\lambda)\begin{vmatrix} 11-\lambda & -12 \\ -6 & 5-\lambda \end{vmatrix} - 6 \begin{vmatrix} -30 & -12 \\ 15 & 5-\lambda \end{vmatrix} - 6 \begin{vmatrix} -30 & 11-\lambda \\ 15 & -6 \end{vmatrix}$$

$$= (-16-\lambda)(\lambda^2-16\lambda-17) - 6(30\lambda+30) - 6(15\lambda+15)$$

$$= -(16+\lambda)(\lambda-17)(\lambda+1) - 180(\lambda+1) - 90(\lambda+1)$$

$$= (\lambda+1)\left[-(\lambda+16)(\lambda-17) - 180 - 90\right]$$

$$= -(\lambda+1)^2(\lambda-2)$$

$\therefore \det(A-\lambda I) = (\lambda+1)^3(\lambda-2)$

$\therefore$ A's eigenvalues are -1 with multiplicity 3, 2 with multiplicity 1

When $\lambda = -1$

$$(A-\lambda I)x = \begin{pmatrix} -15 & 6 & -6 & 0 \\ -30 & 12 & -12 & 0 \\ 15 & -6 & 6 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 0$$

This system has a null space of dimension 3, given

$$\begin{pmatrix} 0 \\ 1 \\ 1 \\ -2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \\ -3 \\ 3 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

When $\lambda = 2$

$$\begin{pmatrix} -18 & 6 & -6 & 0 \\ -30 & 9 & -12 & 0 \\ 15 & -6 & 3 & 0 \\ 0 & 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 0 \Rightarrow x = \begin{pmatrix} 1 \\ 2 \\ -1 \\ 0 \end{pmatrix}$$

(ii) Show that the eigenvalues of the matrix

$$B = \begin{pmatrix} 10 & -20 & -32 & -26 \\ 18 & -41 & -68 & -54 \\ -14 & 19 & 26 & 23 \\ 7 & 1 & 9 & 4 \end{pmatrix}$$

are -1 , with multiplicity 3 , and 2 , with multiplicity 1 .

Show that there exists only one linearly independent eigenvector of the matrix B corresponding to the eigenvalue -1 , e.g., $\begin{pmatrix} 0 \\ -1 \\ -1 \\ 2 \end{pmatrix}$, and show that $\begin{pmatrix} 1 \\ 2 \\ -1 \\ 0 \end{pmatrix}$ is an eigenvector corresponding to the eigenvalue 2 .

(ii) $\det(B-\lambda I) = \begin{vmatrix} 10-\lambda & -20 & -32 & -26 \\ 18 & -41-\lambda & -68 & -54 \\ -14 & 19 & 26-\lambda & 23 \\ 7 & 1 & 9 & 4-\lambda \end{vmatrix}$

$= (\lambda+1)^3(\lambda-2) = 0$ after full expansion.

Hence $\lambda_1 = -1$ (multiplicity 3), $\lambda_2 = 2$ (multiplicity 1)

For $\lambda = -1$

$(B+I)x = 0$

Perform Gaussian elimination

$$B+I = \begin{pmatrix} 11 & -20 & -32 & -26 \\ 18 & -40 & -68 & -54 \\ -14 & 19 & 27 & 23 \\ 7 & 1 & 9 & 5 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Only one free variable, so $x = \begin{pmatrix} 0 \\ -1 \\ -1 \\ 2 \end{pmatrix}$

For $\lambda = 2$

$$B-2I = \begin{pmatrix} 8 & -20 & -32 & -26 \\ 18 & -43 & -68 & -54 \\ -14 & 19 & 24 & 23 \\ 7 & 1 & 9 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$\therefore x = \begin{pmatrix} 1 \\ 2 \\ -1 \\ 0 \end{pmatrix}$