Jacob Malyshev ¶

Variant 20

In [2]:

```
import numpy as np
import matplotlib.pyplot as plt
import math
from scipy.special import comb
import sympy as sp
```

Task 1

Find an interpolation polynomial in the Lagrange form that passes through the four points whose coordinates form the columns of the matrix

$$P = \begin{bmatrix} -3 & -2 & 0 & 3 \\ 16 & -4 & 1 & 14 \end{bmatrix}$$

To solve this task we can use the following formula:

$$f(x) = L(x) = \sum_{i=0}^{n} y_i \cdot \frac{\prod_i (x - x_j)}{\prod_j (x_i - x_j)} = \frac{\sum_{i=0}^{n} y_i v(x_0, \ldots, x_i, \ldots, x_n)}{v(x_0, \ldots, x_n)}$$

In [34]:

```
P = np.array([[-3,-2,0,3],
              [16,-4,1,14]])
```

In [35]:

```
def LagrangePolinom(P):
    x = sp.symbols('x')
    n = P.shape[1]
    f = 0
    indexs = np.arange(0,n)
    for i in range(n):
        ind_i = np.delete(indexs,i)
        f += P[1,i]*(((x-P[0,ind_i[0]])*(x-P[0,ind_i[1]]) * (x-P[0,ind_i[2]]) )
            ((P[0,i]-P[0,ind_i[0]])*(P[0,i]-P[0,ind_i[1]])*(P[0,i]-P[0,ind_i[2]]
    return sp.simplify(f)
```

In [36]:

```
Lpolinom = LagrangePolinom(P)
print('Lagrange Polinom:',Lpolinom)
```

```
Lagrange Polinom: -107*x**3/90 + 14*x**2/9 + 311*x/30 + 1
```

Let's build a plot to show how this polinom makes interpolation

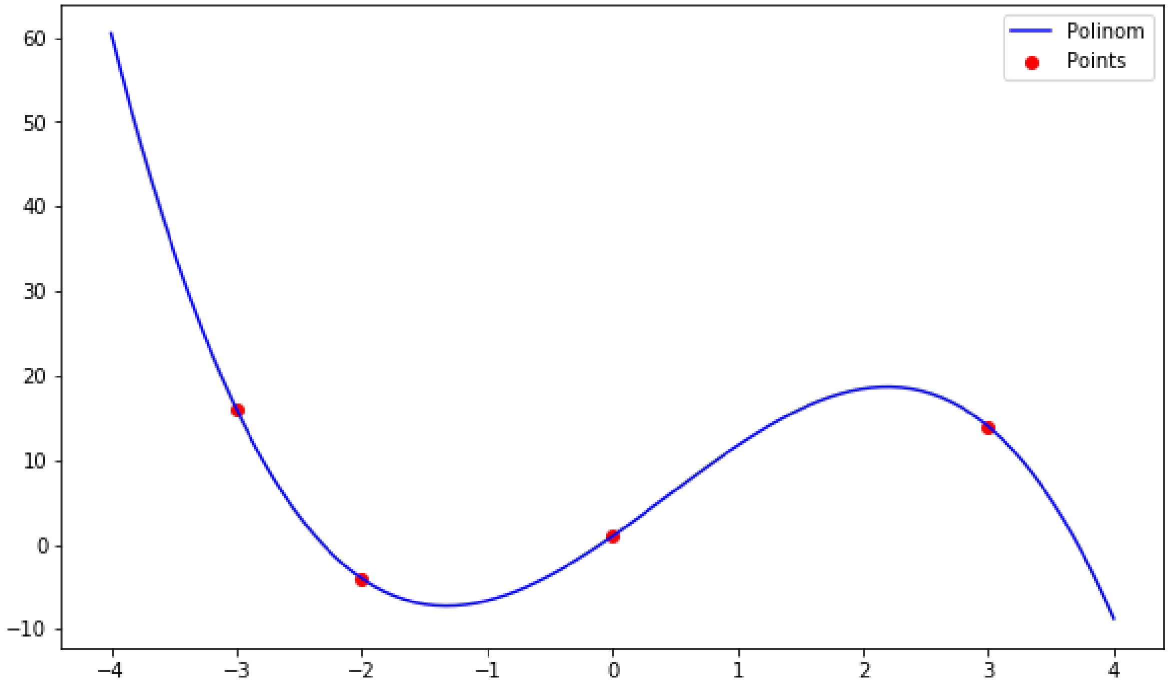
In [37]:

```
x = np.linspace(-4,4,100)
x_symb = sp.symbols('x')
y_lambdify = sp.lambdify(x_symb, LagrangePolinom(P))
y = y_lambdify(x)
```

In [38]:

```
plt.figure(figsize=(10,6))
plt.scatter(P[0],P[1],color='red',label='Points')
plt.plot(x,y,color='blue',label='Polinom')
plt.legend()
plt.show()

```



Seems pretty well

Answer:

$$f(x) = -\frac{107}{90}x^3 + \frac{14}{9}x^2 + \frac{311}{30}x + 1$$

Task 2

Find a (parametric) equation defining the Bezier curve defined by the four points whose coordinates form the columns of the matrix

$$P = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 0 & 5 & 2 & 5 \end{bmatrix}$$

Plot the points and the curve on the coordinate plane.

To solve this task we can use the following formula:

$$B(t) = \sum_{k=0}^{n} C_n^k (1-t)^{n-k} t^k P_k$$

In [41]:

```
P = np.array([[1,3,5,7],
              [0,5,2,5]])
```

In [42]:

```
def param_eq(P):
    t = sp.symbols('t')
    n = P.shape[0]  - 1
    B = 0
    for k in range(n+1):
        B += comb(n,k)*(t**k)*(1-t)**(n-k)*P[k]
    return sp.simplify(B)
```

In [50]:

```
t = np.linspace(0,1,100)
t_symb = sp.symbols('t')
x_lambdify = sp.lambdify(t_symb,param_eq(P[0]))
y_lambdufy = sp.lambdify(t_symb,param_eq(P[1]))
x = x_lambdify(t)
y = y_lambdufy(t)
```

In [46]:

```
print('x(t) =',param_eq(P[0]))
```

```
x(t) = 6.0*t + 1.0
```

In [48]:

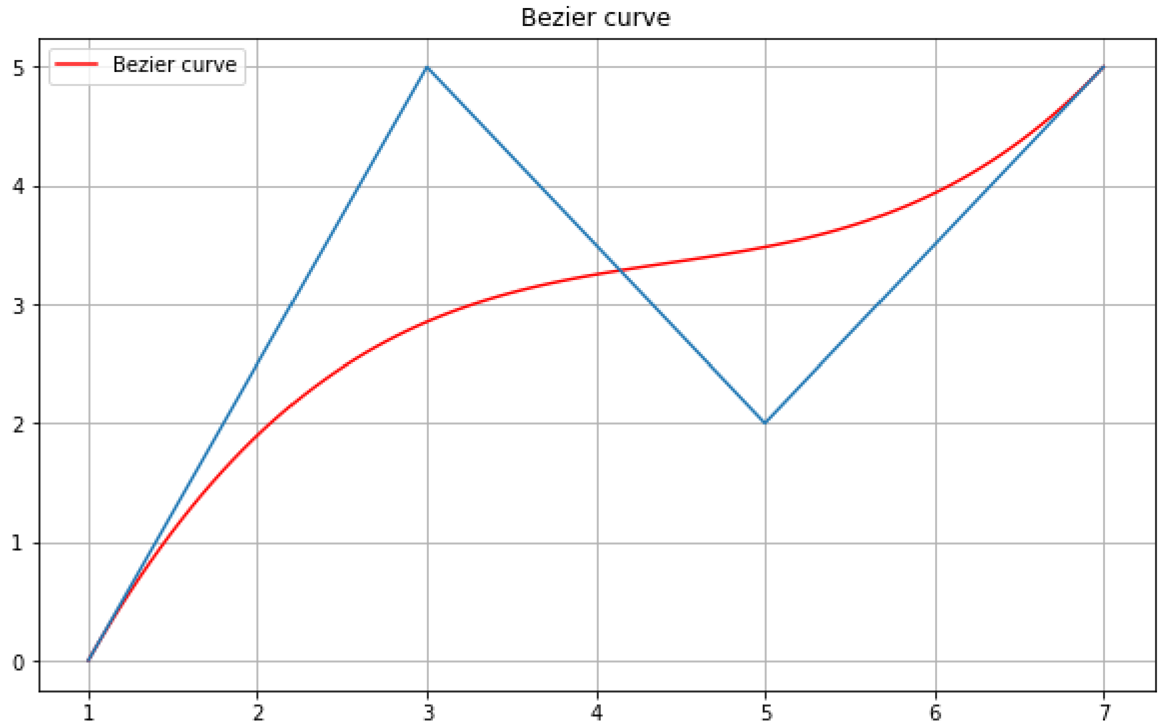
```
print('y(t) =',param_eq(P[1]))
```

```
y(t) = t*(14.0*t**2 - 24.0*t + 15.0)
```

Plot

In [55]:

```
plt.figure(figsize=(10,6))
plt.plot(x,y,color='red',label='Bezier curve')
plt.plot(P[0],P[1])
plt.title('Bezier curve')
plt.grid()
plt.legend()
plt.show()
```



Answer:

$$\begin{cases} x(t) = 6t + 1 \\ y(t) = t(14t^2 - 24t + 15) \end{cases}$$

Task 3

3. Find a full rank decomposition and the pseudoinverse of the matrix

$$A = \begin{bmatrix} 7 & 9 & 5 \\ 4 & 0 & 5 \\ 1 & -9 & 5 \\ 10 & 18 & 5 \end{bmatrix}$$

$$A = FG,$$

where F of full col rank and G of full row rank Then we can find pseudoinverse A^+:

$$A^+ = G^+ F^+$$

Firsly, we have to find rank of A:

$$A = \begin{bmatrix} 7 & 9 & 5 \\ 4 & 0 & 5 \\ 1 & -9 & 5 \\ 10 & 18 & 5 \end{bmatrix} \approx \begin{bmatrix} 3 & 9 & 0 \\ 4 & 0 & 5 \\ 1 & -9 & 5 \\ 2 & 18 & -5 \end{bmatrix} \approx \begin{bmatrix} 3 & 9 & 0 \\ 4 & 0 & 5 \\ 1 & -9 & 5 \\ -1 & 9 & -5 \end{bmatrix} \approx \begin{bmatrix} 4 & 0 & 5 \\ 4 & 0 & 5 \\ 1 & -9 & 5 \\ -1 & 9 & -5 \end{bmatrix} \approx \begin{bmatrix} 4 & 0 & 5 \\ 1 & -9 & 5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \approx$$

$$rank(A) = 2$$

And we can decompose A: A = FG:

$$F = \begin{bmatrix} 7 & 9 \\ 4 & 0 \\ 1 & -9 \\ 10 & 18 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & -\frac{5}{12} \end{bmatrix}$$

Full rank decomposition:

$$A = FG = \begin{bmatrix} 7 & 9 \\ 4 & 0 \\ 1 & -9 \\ 10 & 18 \end{bmatrix} \begin{bmatrix} 1 & 0 & \frac{5}{4} \\ 0 & 1 & -\frac{5}{12} \end{bmatrix}$$

Let's check it:

In [84]:

```
print('A matrix:\n',A)
```

```
A matrix:
 [[ 7  9  5]
 [ 4  0  5]
 [ 1 -9  5]
 [10 18  5]]
```

In [85]:

```
print('FG matrix:\n',np.dot(F,G))
```

```
FG matrix:
 [[ 7.  9.  5.]
 [ 4.  0.  5.]
 [ 1. -9.  5.]
 [10. 18.  5.]]
```

We can coclude, that our full rank decomposition of A is true

Then we use the following formulas:

$$F^+ = (F^*F)^{-1}F^*$$
$$G^+ = G^*(GG^*)^{-1}$$

In [96]:

```
A = np.array([[7,9,5],
              [4,0,5],
              [1,-9,5],
              [10,18,5]])

F = np.array([[7,9],
              [4,0],
              [1,-9],
              [10,18]])

G = np.array([[1,0,5/4],
              [0,1,-5/12]])
```

In [78]:

```
F_star = F.T
F_plus = np.linalg.inv((F_star @ F)) @ F_star
G_star = G.T
G_plus = G_star @ np.linalg.inv((G @ G_star))
```

Finally, we can use the following formula:

$$A^+ = G^+F^+$$

In [79]:

```
A_plus = G_plus @ F_plus
```

In [92]:

```
print('A matrix: \n',np.round(A_plus,decimals=3))
```

```
A matrix:
 [[ 0.02   0.025  0.03   0.015]
 [ 0.004 -0.02  -0.043  0.028]
 [ 0.024  0.04   0.056  0.008]]
```

Let's check it using the following formula:

$$A^+ A A^+ = A^+$$

In [93]:

```
A_plus @ A @ A_plus
```

Out[93]:

```
array([[ 0.02038917,  0.02529611,  0.03020305,  0.01548223],
       [ 0.00431472, -0.01954315, -0.04340102,  0.02817259],
       [ 0.02368866,  0.03976311,  0.05583756,  0.00761421]])
```

In [81]:

```
A_plus
```

Out[81]:

```
array([[ 0.02038917,  0.02529611,  0.03020305,  0.01548223],
       [ 0.00431472, -0.01954315, -0.04340102,  0.02817259],
       [ 0.02368866,  0.03976311,  0.05583756,  0.00761421]])
```

We see, that this matrices are equal.

Answer:

1) Full rank decomposition:

$$A = FG = \begin{bmatrix} 7 & 9 \\ 4 & 0 \\ 1 & -9 \\ 10 & 18 \end{bmatrix} \begin{bmatrix} 1 & 0 & \frac{5}{4} \\ 0 & 1 & -\frac{5}{12} \end{bmatrix}$$

2) Pseudoinverse of A

$$A^+ = \begin{bmatrix} 0.02 & 0.025 & 0.03 & 0.015 \\ 0.004 & -0.02 & -0.043 & 0.028 \\ 0.024 & 0.04 & 0.056 & 0.008 \end{bmatrix}$$

Task 4

Find the minimal length least squares solution of the system of linear equations

$$\begin{cases} 14x + 13y + 8z + 3t = 4 \\ 7x + 9y + 5z + 4t = 6 \\ 0x + 5y + 2z + 5t = 3 \\ 4x + 4y + 9z + 4t = 3 \end{cases}$$

We can represent this system of linear equations in the following form:

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

where

$$A = \begin{bmatrix} 14 & 13 & 8 & 3 \\ 7 & 9 & 5 & 4 \\ 0 & 5 & 2 & 5 \\ 4 & 4 & 9 & 4 \end{bmatrix}$$

$$\vec{x} = \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix}, \vec{b} = \begin{bmatrix} 4 \\ 6 \\ 3 \\ 3 \end{bmatrix}$$

We want to find $\vec{x}$, to do this we can use the following step:

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

Firstly, we have to find $rank(A)$:

$$\begin{bmatrix} 14 & 13 & 8 & 3 \\ 7 & 9 & 5 & 4 \\ 0 & 5 & 2 & 5 \\ 4 & 4 & 9 & 4 \end{bmatrix} \approx \begin{bmatrix} 0 & -5 & -2 & -5 \\ 7 & 9 & 5 & 4 \\ 0 & 5 & 2 & 5 \\ 4 & 4 & 9 & 4 \end{bmatrix} \approx \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 1 & -13 & -4 \\ 0 & 5 & 2 & 5 \\ 4 & 4 & 9 & 4 \end{bmatrix} \approx \begin{bmatrix} -1 & 1 & -13 & -4 \\ 0 & 8 & -43 & -12 \\ 0 & 5 & 2 & 5 \\ 0 & 0 & 0 & 0 \end{bmatrix} \approx$$

$$\approx \begin{bmatrix} 1 & 0 & 0 & -\frac{185}{231} \\ 0 & 1 & 0 & \frac{191}{231} \\ 0 & 0 & 1 & \frac{100}{231} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Therefore, $rank(A) = 3$ and to find $A^{+} = G^{+}F^{+}$ and to calculate it, we can use formulas from previous task.

In [104]:

```
A = np.array([[14,13,8,3],
              [7,9,5,4],
              [0,5,2,5],
              [4,4,9,4]])

b = np.array([[4,6,3,3]]).T

F = np.array([[14,13,8],
              [7,9,5],
              [0,5,2],
              [4,4,9]])

G = np.array([[1,0,0,-185/231],
              [0,1,0,191/231],
              [0,0,1,100/231]])
```

In [99]:

```
F_star = F.T
F_plus = np.linalg.inv((F_star @ F)) @ F_star
G_star = G.T
G_plus = G_star @ np.linalg.inv((G @ G_star))
```

In [100]:

```
A_plus = G_plus @ F_plus
```

In [103]:

```
print('Pseudoinverse matrix of A: \n',A_plus)
```

```
Pseudoinverse matrix of A: 
 [[ 0.06262416 -0.00878417 -0.08019249 -0.01763297]
 [ 0.01642587  0.04690689  0.07738792 -0.07368704]
 [-0.01583288 -0.03128784 -0.0467428   0.14466647]
 [-0.04342605  0.03227491  0.10797586  0.01582045]]
```

Finally, we can find least square solution of our system

In [106]:

```
x = A_plus @ b
```

In [109]:

```
print('Least square solution: \n',np.round(x,3))
```

```
Least square solution: 
 [[-0.096]
 [ 0.358]
 [ 0.043]
 [ 0.391]]
```

Answer:

$$\vec{x} = \begin{bmatrix} -0.096 \\ 0.358 \\ 0.043 \\ 0.391 \end{bmatrix}$$

Task 5

For the polynomial $x^3 + 3x^2 - 4x - 5$ find the best approximation with respect to the norm $\int_1^2 |f(x)|dx$ by a polynomial of degree 2 on a line segment [1,2]

We can reformulate this task in the following form:

$$||x^3 + 3x^2 - 4x - 5 - P_2(x)||_1 \rightarrow min$$

where $x \in [1, 2]$, norm $|f|_1 = \int_{-1}^{1} |f(x)|dx$

However, our segment is differ from $[-1, 1]$, so, firstly, we have to solve this problem. To do it, we can use the following formula:

$$x = \frac{\tilde{x} - \frac{a+b}{2}}{b-a},$$

where $a = 1, b = 2$. So, $\frac{x - \frac{3}{2}}{1} = x - \frac{3}{2}$. Using this fact and formula:

$$U_n(x) = \frac{1}{n+1} T'_{n+1}(x)$$

for $n \geqslant 0$

$\bar{U}(x) = \frac{1}{8} U_3(x - \frac{3}{2}) = \frac{1}{8}(8(x - \frac{3}{2})^3 - 4(x - \frac{3}{2})) = (x - \frac{3}{2})^3 - \frac{1}{2}(x - \frac{3}{2}) = (x - \frac{3}{2})((x - \frac{3}{2})^2 - \frac{1}{2}) = ($
$- \frac{21}{8}$

To find $P_2(x)$ we have to use this: $x^3 + 3x^2 - 4x - 5 - P_2(x) = \bar{U}_3(x)$

Then $P_2(x) = x^3 + 3x^2 - 4x - 5 - x^3 + \frac{9}{2}x^2 - \frac{25}{4}x + \frac{21}{8} = \boldsymbol{\frac{15}{2}x^2 - \frac{41}{4}x - \frac{19}{8}}$

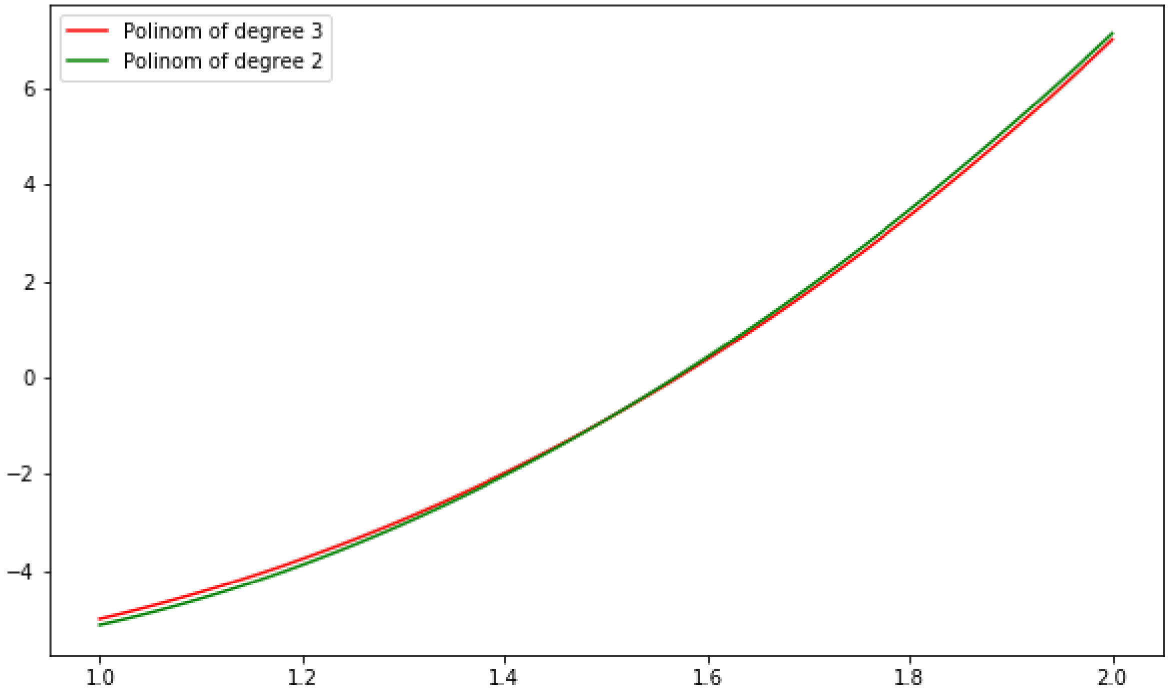
Let's build a plot to check it.

In [127]:

```
x = np.linspace(1,2,100)
y_real = x**3 + 3*x**2 - 4*x - 5
y_cheb = (15/2)*x**2 - (41/4)*x - 19/8
```

In [128]:

```
plt.figure(figsize=(10,6))
plt.plot(x,y_real,color='r',label='Polinom of degree 3')
plt.plot(x,y_cheb,color='g',label='Polinom of degree 2')
plt.legend()
plt.show()
```



Answer:

$$P_2(x) = \frac{15}{2}x^2 - \frac{41}{4}x - \frac{19}{8}$$

Task 6

Find all the values of q such that the equation $2x^2 + y^2(4q+1) + yz(-2q+4) + z^2(4q+2) = 1$ defines a unit circle with respect to some norm? Find the value of this norm fro the vector $(1,1,1)$ as a function of q.

To solve this task we have to use Minkovsky theorem. Firstly, let's build matrix of quadratic form:

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & (4q+1) & (-2q+4) \\ 0 & (-2q+4) & (4q+2) \end{bmatrix}$$

$\triangle_1 = 2 > 0$, not depends on q

$\triangle_2 = 8q + 2 > 0$, $q > -\frac{1}{4} = -0.25$

$\triangle_3 = 2(4q + 1)(4q + 2) - 2(4 - 2q)^2 > 0$, so $q \in [-\infty, -0.18] \cup [0.113, +\infty]$ (approximately)

When Q is positive defined, all conditions of Minkovsky theorem are true. To satisfy this, q must be from $(0.113, +\infty)$. Then our equation defines a unit circle.

To find the value of this norm from the vector (1,1,1) as a function of q we need to find t from the following equation:

$$2t^2 + t^2(4q + 1) + t^2(-2q + 4) + t^2(4q + 2) = 1$$
$$t^2(2 + 4q + 1 - 2q + 4 + 4q + 2) = 1$$
$$t^2(6q + 9) = 1$$
$$t^2 = \frac{1}{6q + 9}$$
$$t = \frac{1}{\sqrt{6q + 9}}$$

So, our norm $N((1, 1, 1)) = \sqrt{6q + 9}$

Answer:

1) Values of q:

$$q \in (0.113, +\infty)$$

2) Value of norm from $(1, 1, 1)$

$$N((1, 1, 1)) = \sqrt{6q + 9}$$

In [ ]:

```
1
```