

# **Отчёт по лабораторной работе №5**

**Дисциплина: Научное программирование**

Полиенко Анастасия Николаевна, НПМмд-02-23

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Подгонка полиномиальной прямой . . . . .	6
3.2	Матричные преобразования . . . . .	13
3.2.1	Вращение . . . . .	15
3.2.2	Отражение . . . . .	16
3.2.3	Дилатация . . . . .	18
<b>4</b>	<b>Выводы</b>	<b>21</b>

## Список иллюстраций

3.1	Исходные данные. Матрица и векторы . . . . .	7
3.2	Исходные данные. График . . . . .	8
3.3	Матрица A . . . . .	9
3.4	Коэффициенты . . . . .	10
3.5	Векторы x и y . . . . .	11
3.6	График $y = -0.8929x^2 + 5.65x - 4.4$ . . . . .	11
3.7	Метод Polyfit . . . . .	12
3.8	График Polyfit . . . . .	13
3.9	Исходные данные. Матрица и векторы . . . . .	14
3.10	Граф “домик” . . . . .	14
3.11	Данные для поворота на 90 градусов . . . . .	15
3.12	Данные для поворота на 225 градусов . . . . .	15
3.13	Результат поворота . . . . .	16
3.14	Данные для отражения . . . . .	17
3.15	Результат отражения . . . . .	18
3.16	Данные для расширения . . . . .	19
3.17	Результат расширения . . . . .	20

# 1 Цель работы

Изучить матричные преобразования для графиков.

## 2 Задание

1. Изучить подгонку полиномиальной кривой с помощью МНК
2. Изучить построение графов
3. Изучить повороты изображения
4. Изучить отображения изображения
5. Изучить дилатацию изображения

## **3 Выполнение лабораторной работы**

### **3.1 Подгонка полиномиальной прямой**

1. Создадим матрицу данных и отдельные вектора  $x$  и  $y$  (рис. 3.1), которые в графическом представлении имеют вид (рис. 3.2).

```

>> D = [1 1; 2 2; 3 5; 4 4; 5 2; 6 -3]
D =

     1     1
     2     2
     3     5
     4     4
     5     2
     6    -3

>> xdata = D(:,1)
xdata =

     1
     2
     3
     4
     5
     6

>> ydata = D(:,2)
ydata =

     1
     2
     5
     4
     2
    -3

>> plot(xdata,ydata,'o-')

```

Рис. 3.1: Исходные данные. Матрица и векторы

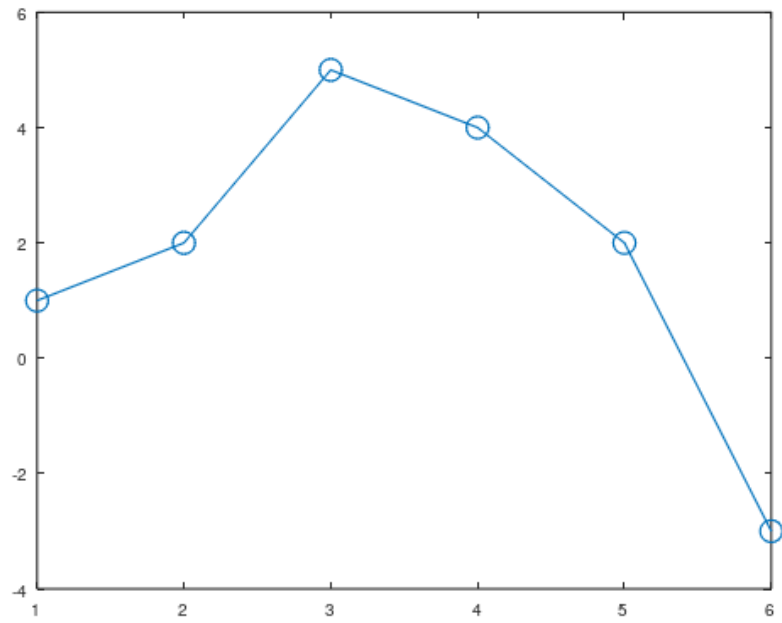


Рис. 3.2: Исходные данные. График

2. Мы хотим подогнать наши данные под кривую  $y = a_1 x^2 + a_2 x + a_3$ . Для этого создадим матрицу  $A$  (рис. 3.3).



```

>> A = ones(6,3)
A =

     1     1     1
     1     1     1
     1     1     1
     1     1     1
     1     1     1
     1     1     1

>> A(:,1) = xdata.^2
A =

     1     1     1
     4     1     1
     9     1     1
    16     1     1
    25     1     1
    36     1     1

>> A(:,2) = xdata
A =

     1     1     1
     4     2     1
     9     3     1
    16     4     1
    25     5     1
    36     6     1

```

Рис. 3.3: Матрица A

3. Найдём коэффициенты (рис. 3.4).

```
>> B_res = rref(B)
B_res =

    1.0000         0         0   -0.8929
         0    1.0000         0    5.6500
         0         0    1.0000   -4.4000

>> a1 = B_res(1,4)
a1 = -0.8929
>> a2 = B_res(2,4)
a2 = 5.6500
>> a3 = B_res(3,4)
a3 = -4.4000
```

Рис. 3.4: Коэффициенты

4. Для построения полиномиальной кривой создадим векторы  $x$  и  $y$  (рис. 3.5) и построим график (рис. 3.6).

```

>> x = linspace(0,7,50)
x =
Columns 1 through 11:
    0    0.1429    0.2857    0.4286    0.5714    0.7143    0.8571    1.0000    1.1429    1.2857    1.4286
Columns 12 through 22:
    1.5714    1.7143    1.8571    2.0000    2.1429    2.2857    2.4286    2.5714    2.7143    2.8571    3.0000
Columns 23 through 33:
    3.1429    3.2857    3.4286    3.5714    3.7143    3.8571    4.0000    4.1429    4.2857    4.4286    4.5714
Columns 34 through 44:
    4.7143    4.8571    5.0000    5.1429    5.2857    5.4286    5.5714    5.7143    5.8571    6.0000    6.1429
Columns 45 through 50:
    6.2857    6.4286    6.5714    6.7143    6.8571    7.0000
>> y = a1 * x.^2 + a2 * x + a3
y =
Columns 1 through 9:
   -4.400000   -3.611079   -2.858601   -2.142566   -1.462974   -0.819825   -0.219120    0.357143    0.890962
Columns 10 through 18:
    1.388338    1.849271    2.279761    2.661808    3.013411    3.328571    3.607289    3.849563    4.055394
Columns 19 through 27:
    4.224781    4.357726    4.454227    4.514286    4.537901    4.525073    4.475802    4.390087    4.267930
Columns 28 through 36:
    4.109329    3.914286    3.682799    3.414869    3.110496    2.769679    2.392420    1.978717    1.528571
Columns 37 through 45:
    1.041983    0.518950   -0.040525   -0.636443   -1.268805   -1.937609   -2.642857   -3.384548   -4.162682
Columns 46 through 50:
   -4.977259   -5.828280   -6.715743   -7.639650   -8.600000

```

Рис. 3.5: Векторы  $x$  и  $y$

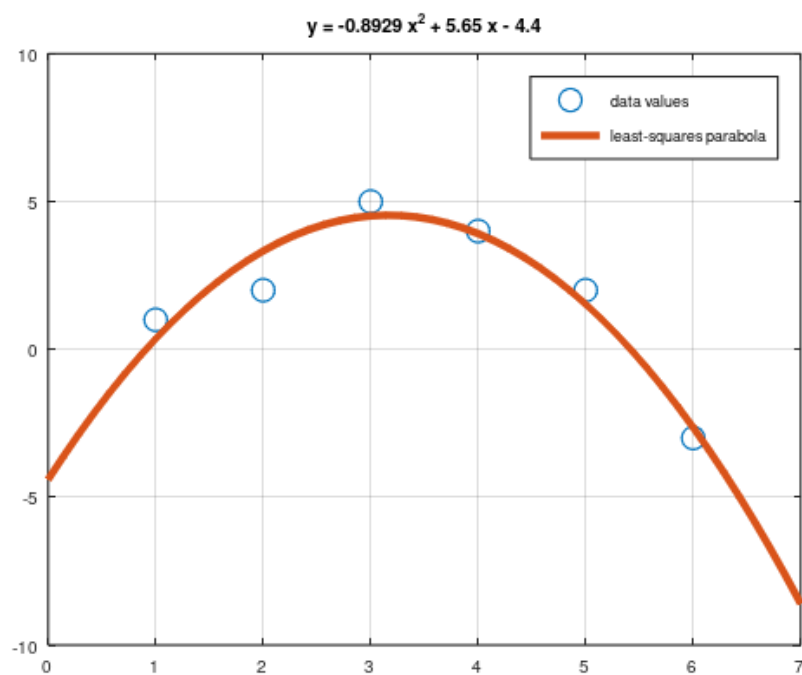


Рис. 3.6: График  $y = -0.8929x^2 + 5.65x - 4.4$

5. Для построения полиномиальной кривой можно использовать встроенный метод *polyfit* (рис. 3.7), В результате получаем такой график (рис. 3.8).

```
>> P = polyfit(xdata, ydata, 2)
P =

    -0.8929    5.6500   -4.4000

>> y = polyval(P, xdata)
y =

    0.3571
    3.3286
    4.5143
    3.9143
    1.5286
   -2.6429
```

Рис. 3.7: Метод Polyfit

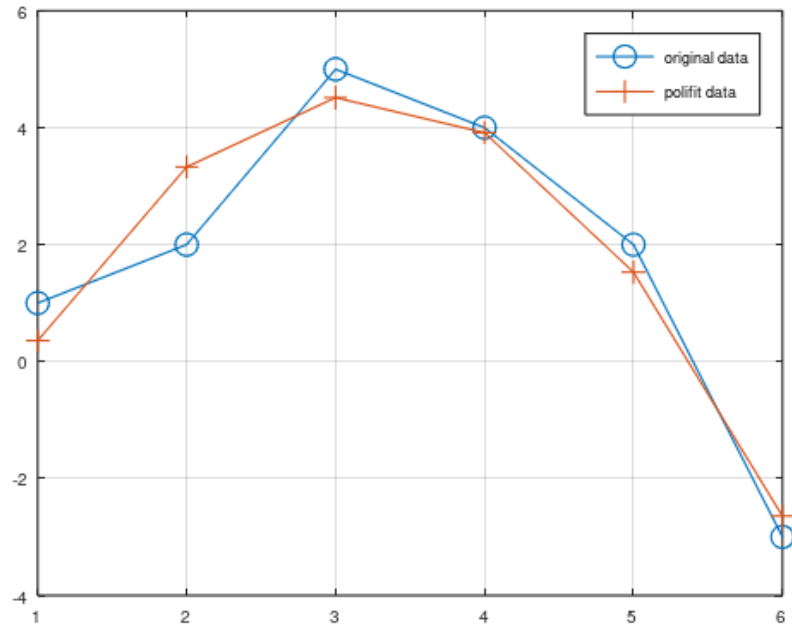


Рис. 3.8: График Polyfit

## 3.2 Матричные преобразования

Создадим матрицу данных и отдельные вектора  $x$  и  $y$  (рис. 3.9), которые в графическом представлении имеют вид (рис. 3.10).

```

-----
>> D = [1 1 3 3 2 1 3; 2 0 0 2 3 2 2]
D =

     1     1     3     3     2     1     3
     2     0     0     2     3     2     2

>> x = D(1, :)
x =

     1     1     3     3     2     1     3

>> y = D(2, :)
y =

     2     0     0     2     3     2     2

>> plot(x, y)

```

Рис. 3.9: Исходные данные. Матрица и векторы

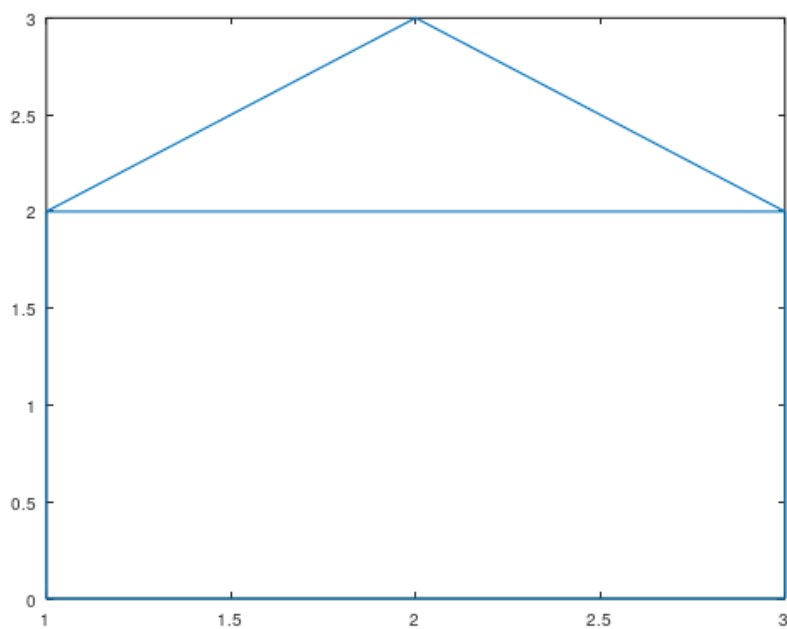


Рис. 3.10: Граф “домик”

### 3.2.1 Вращение

Изучим, как осуществляется вращение изображения.

Зададим угол поворота и матрицу вращения, посчитаем новые координаты для угла 90 градусов (рис. 3.11) и угла 225 градусов (рис. 3.12). В результате получаем такую картинку (рис. 3.13).

```
>> thetal = 90*pi/180
thetal = 1.5708
>> R1 = [cos(thetal) -sin(thetal); sin(thetal) cos(thetal)]
R1 =

    6.1230e-17   -1.0000e+00
    1.0000e+00    6.1230e-17

>> RD1 = R1*D
RD1 =

   -2.0000e+00    6.1230e-17    1.8369e-16   -2.0000e+00   -3.0000e+00   -2.0000e+00   -2.0000e+00
    1.0000e+00    1.0000e+00    3.0000e+00    3.0000e+00    2.0000e+00    1.0000e+00    3.0000e+00

>> x1 = RD1(1, :)
x1 =

   -2.0000e+00    6.1230e-17    1.8369e-16   -2.0000e+00   -3.0000e+00   -2.0000e+00   -2.0000e+00

>> y1 = RD1(2, :)
y1 =

    1    1    3    3    2    1    3
```

Рис. 3.11: Данные для поворота на 90 градусов

```
>> theta2 = 225*pi/180
theta2 = 3.9270
>> R2 = [cos(theta2) -sin(theta2); sin(theta2) cos(theta2)]
R2 =

   -0.7071    0.7071
   -0.7071   -0.7071

>> RD2 = R2*D
RD2 =

    0.7071   -0.7071   -2.1213   -0.7071    0.7071    0.7071   -0.7071
   -2.1213   -0.7071   -2.1213   -3.5355   -3.5355   -2.1213   -3.5355

>> x2 = RD2(1, :)
x2 =

    0.7071   -0.7071   -2.1213   -0.7071    0.7071    0.7071   -0.7071

>> y2 = RD2(2, :)
y2 =

   -2.1213   -0.7071   -2.1213   -3.5355   -3.5355   -2.1213   -3.5355
```

Рис. 3.12: Данные для поворота на 225 градусов

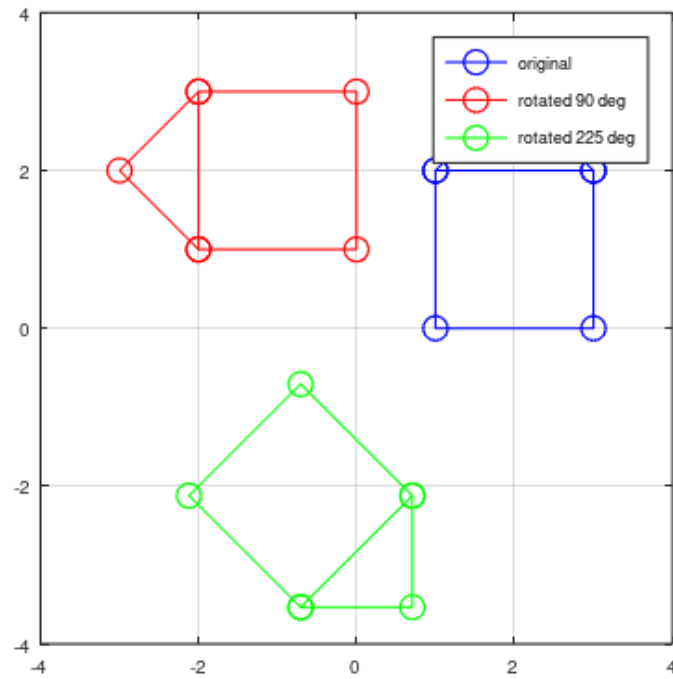


Рис. 3.13: Результат поворота

### 3.2.2 Отражение

Изучим, как осуществляется отражение изображения относительно прямой.

Зададим матрицу отражения относительно прямой  $x = y$ , посчитаем новые координаты (рис. 3.14). В результате получаем такую картинку (рис. 3.15).



```

>> R = [0 1; 1 0]
R =

     0     1
     1     0

>> RD = R*D
RD =

     2     0     0     2     3     2     2
     1     1     3     3     2     1     3

>> x1 = RD(1,:)
x1 =

     2     0     0     2     3     2     2

>> y1 = RD(2,:)
y1 =

     1     1     3     3     2     1     3

```

Рис. 3.14: Данные для отражения

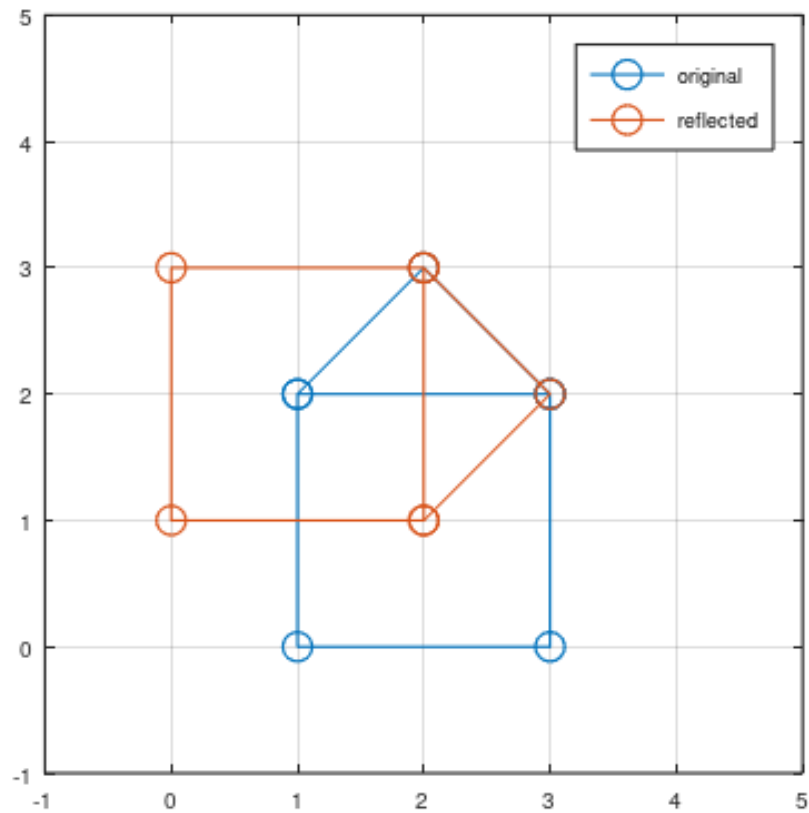


Рис. 3.15: Результат отражения

### 3.2.3 Дилатация

Изучим, как осуществляется дилатация (расширение или сжатие) изображения.

Зададим матрицу расширения в 2 раза, посчитаем новые координаты (рис. 3.16). В результате получаем такую картинку (рис. 3.17).

```

>> T = [2 0; 0 2]
T =

    2    0
    0    2

>> TD = T*D
TD =

    2    2    6    6    4    2    6
    4    0    0    4    6    4    4

>> x1 = TD(1,:)
x1 =

    2    2    6    6    4    2    6

>> y1 = TD(2,:)
y1 =

    4    0    0    4    6    4    4

```

Рис. 3.16: Данные для расширения

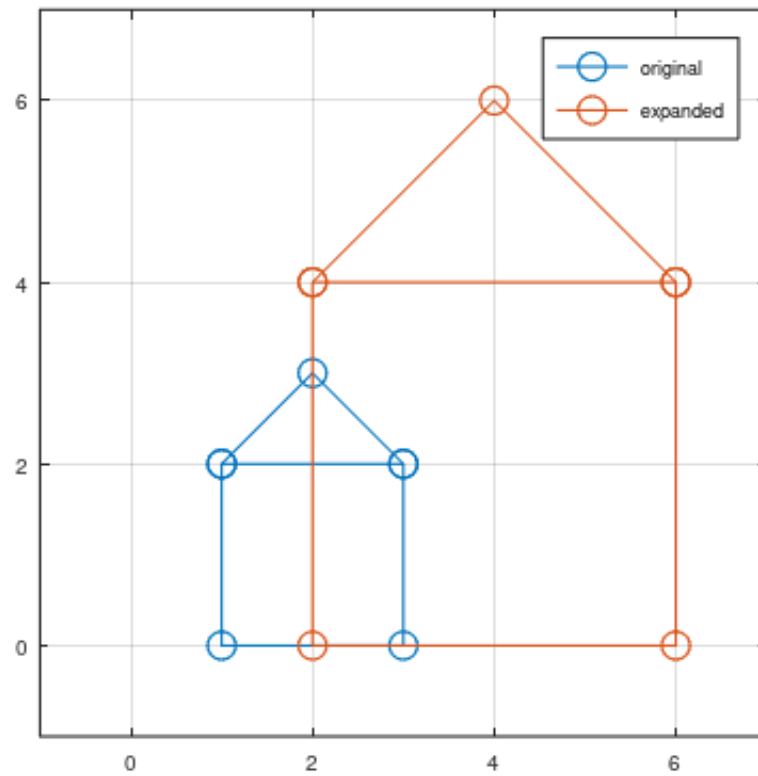


Рис. 3.17: Результат расширения

## 4 Выводы

Изучила подгонку полиномиальной прямой и матричные преобразования в Octave.