

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

Подгонка полиномиальной кривой

Аминов Зулфикор Мирзокаримович

Содержание

Ход работы:	1
Матричные преобразования	9
Вращение.....	10
Отражение.....	14
Дилатация	17
Вывод.....	19

Ход работы:

В матрице заданы значения x в столбце 1 и значения y в столбце 2. Введём матрицу данных в Octave и извлечём вектора x и y .

```
>> 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
```

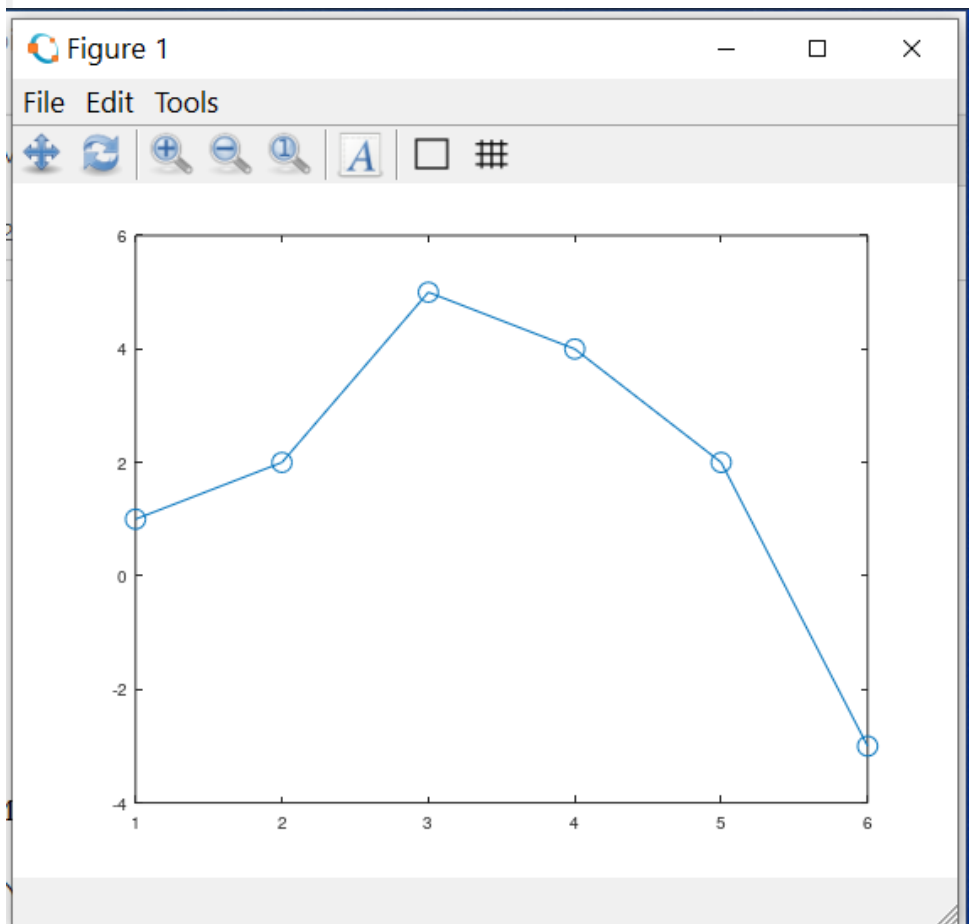
```
>> ydata = D(:, 2)
```

```
ydata =
```

```
    1  
    2  
    5  
    4  
    2  
   -3
```

Нарисуем точки на графике.

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



Построим уравнение вида $y = ax^2 + bx + c$.

```
>> 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
```

Решение по методу наименьших квадратов получается из решения уравнения $A^T A b = A^T y$, где b – вектор коэффициентов полинома. Используем Octave для построения уравнений.

```
>> A' * A
```

```
ans =
```

```
2275 441 91
441 91 21
91 21 6
```

```
>> A' * ydata
```

```
ans =
```

```
60
28
11
```

рисунка 7

Решим задачу методом Гаусса. Запишем расширенную матрицу:

```

>> B = A' * A;
>> B(:, 4) = A' * ydata;
>> 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

```

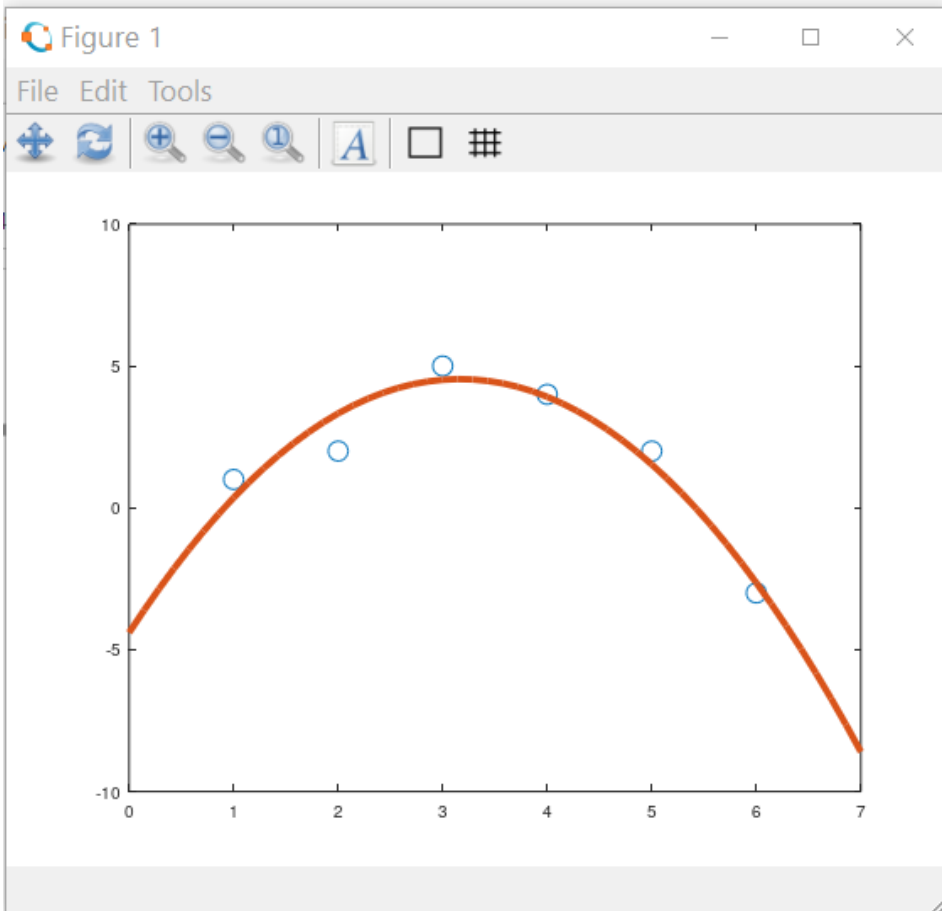
рисунка 8

Таким образом, искомое квадратное уравнение имеет вид:

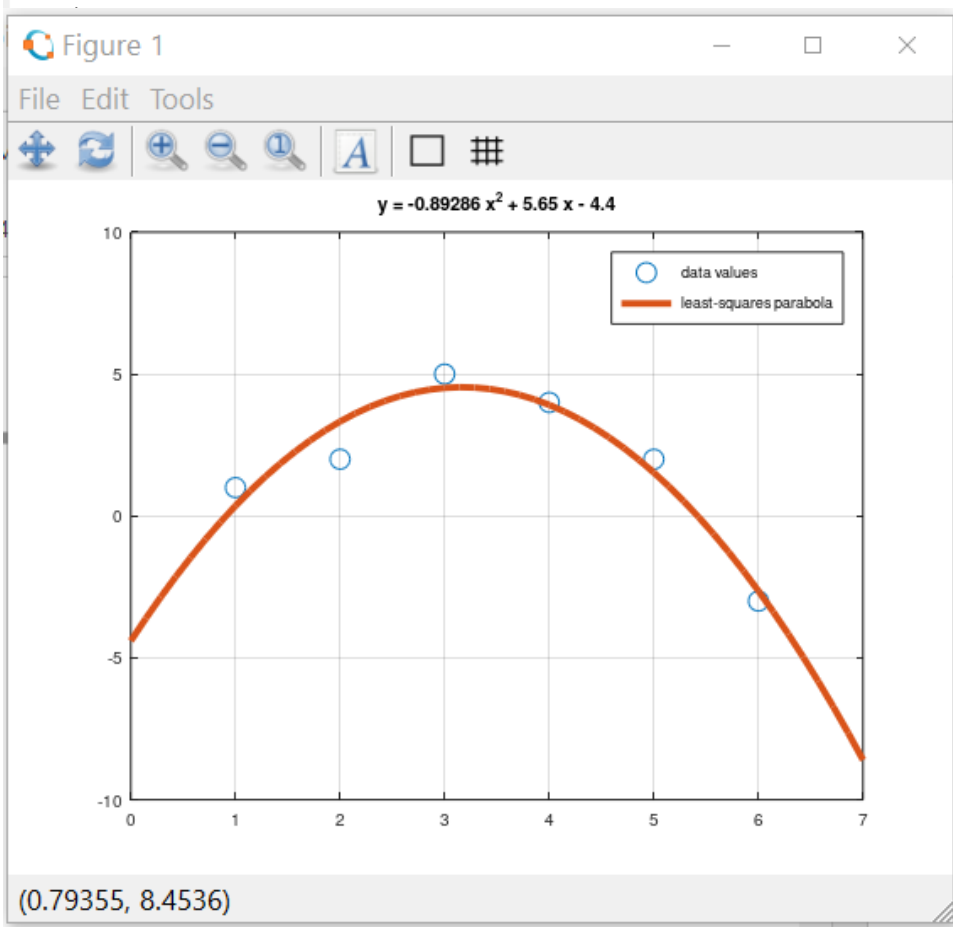
$$y = -0.89286x^2 + 5.65x - 4.4.$$

Построим соответствующий график параболы

```
>> x = linspace (0, 7, 50);  
>> y = a1 * x .^2 + a2 * x + a3;  
>>  
>> plot(xdata, udata, 'o', x, y, 'linewidth', 2)  
error: 'udata' undefined near line 1, column 1  
>> plot(xdata, ydata, 'o', x, y, 'linewidth', 2)  
>> |
```



```
>> grid on;
>> legend('data values', 'least-squares parabola')
>> title ('y = -0.89286 x^2 + 5.65 x - 4.4')
>>
```



Получим подгоночный полином.

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

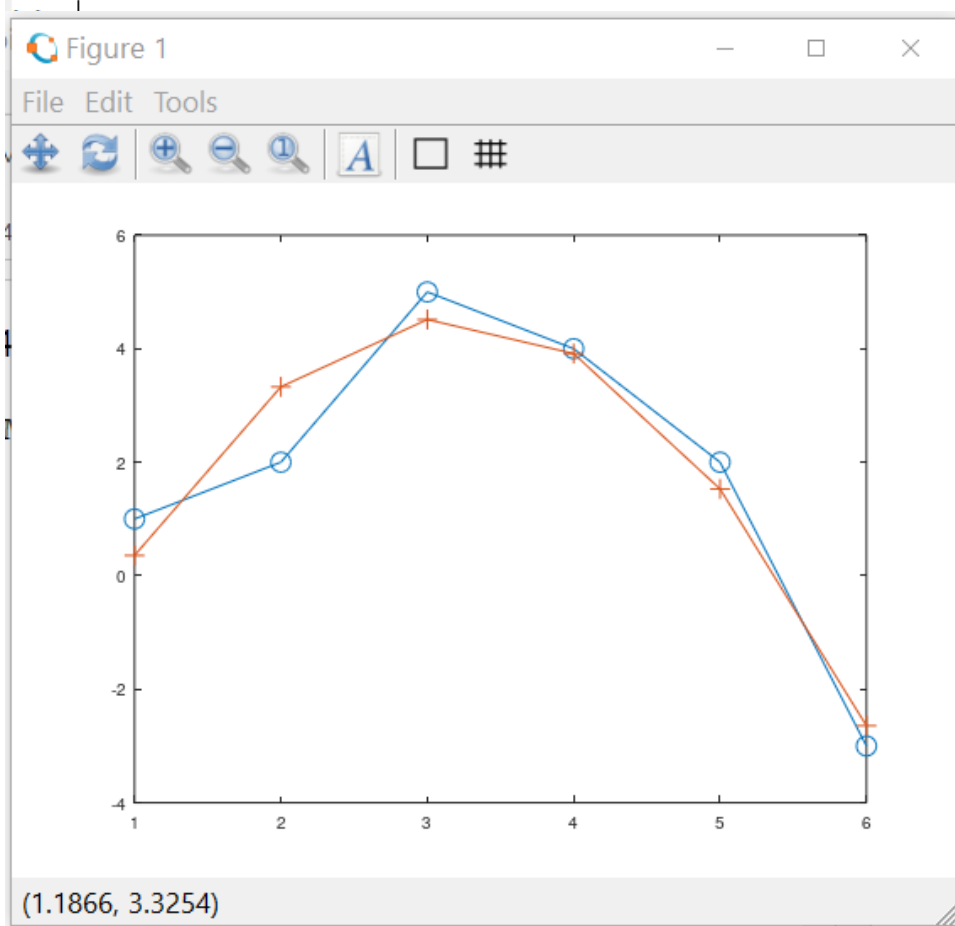
    0.3571
    3.3286
    4.5143
    3.9143
    1.5286
   -2.6429

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

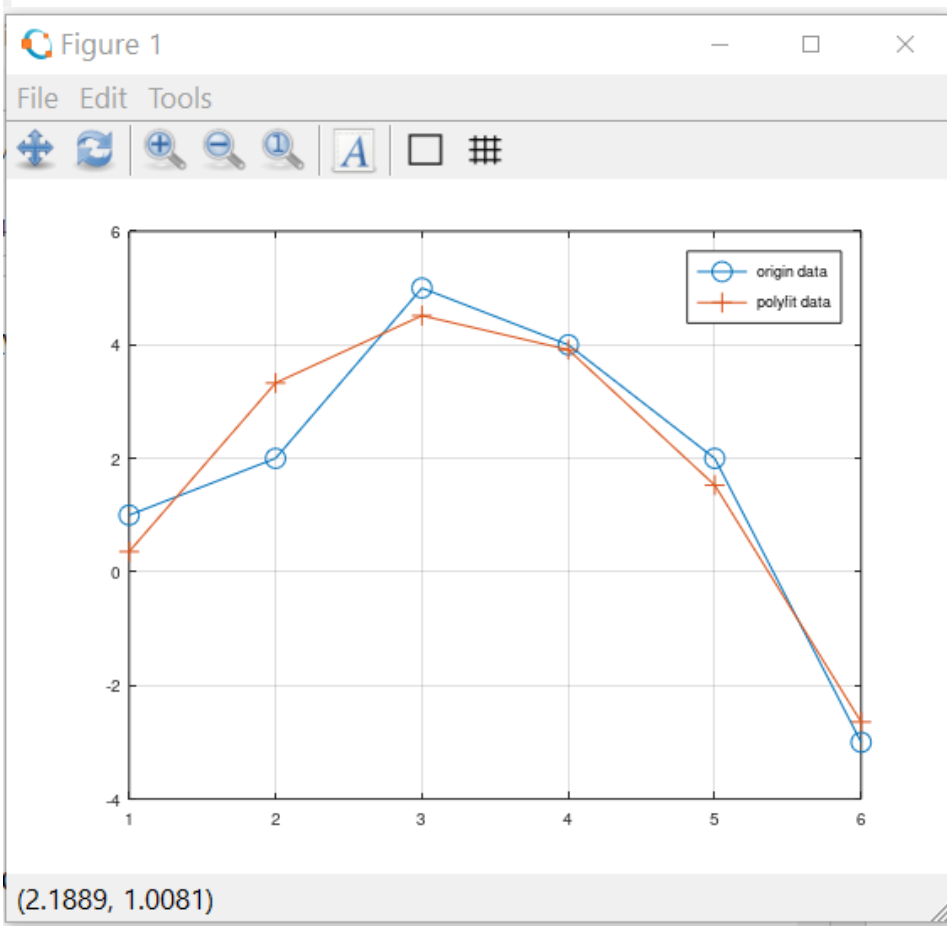
   -0.8929    5.6500   -4.4000
```

Построим исходные и подгоночные данные.

```
>> plot(xdata, ydata, 'o-', xdata, y, '+-')
```




```
>> grid on;
>> legend ('origin data', 'polyfit data');
>> |
```



Матричные преобразования

Введём матрицу данных.

```
>> 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
```

рисунка 19

Извлечём вектора x и y и нарисуем этот граф.

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

```
x =
```

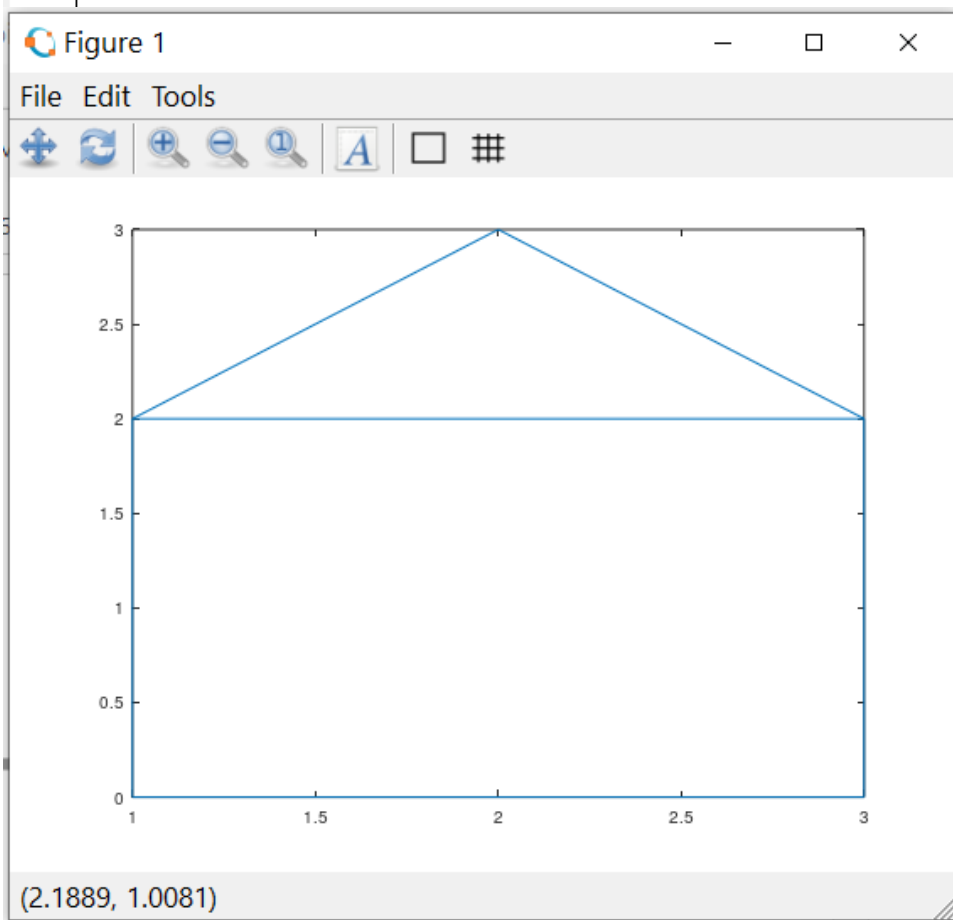
```
1 1 3 3 2 1 3
```

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

```
y =
```

```
2 0 0 2 3 2 2
```

```
>> plot(x, y)
```



Вращение

Чтобы произвести повороты матрицы данных D , нам нужно вычислить произведение матриц RD .

Повернём граф дома на 90° и 225° . Вначале переведём угол в радианы.

```

>> theta1 = 90*pi/180
theta1 = 1.5708
>> R1 = [cos(theta1) -sin(theta1); sin(theta1) cos(theta1)]
R1 =

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

>> RD1 = R1 * D
RD1 =

Columns 1 through 4:

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

Columns 5 through 7:

   -3.0000e+00   -2.0000e+00   -2.0000e+00
    2.0000e+00    1.0000e+00    3.0000e+00

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

Columns 1 through 4:

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

Columns 5 through 7:

   -3.0000e+00   -2.0000e+00   -2.0000e+00

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

Columns 1 through 6:

    1.0000    1.0000    3.0000    3.0000    2.0000    1.0000

Column 7:

    3.0000

```

```

>> 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 =

Columns 1 through 6:

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

Column 7:

   -0.7071
   -3.5355

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

Columns 1 through 6:

    0.7071   -0.7071   -2.1213   -0.7071    0.7071    0.7071

Column 7:

   -0.7071

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

Columns 1 through 6:

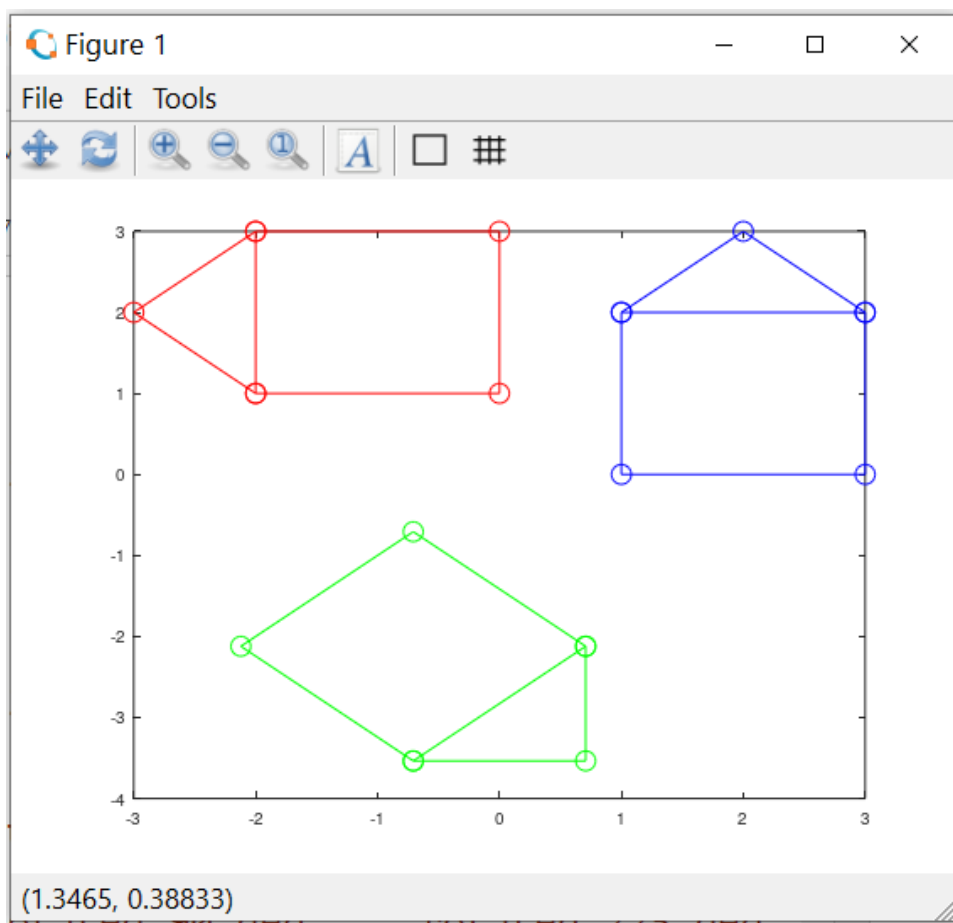
   -2.1213   -0.7071   -2.1213   -3.5355   -3.5355   -2.1213

Column 7:

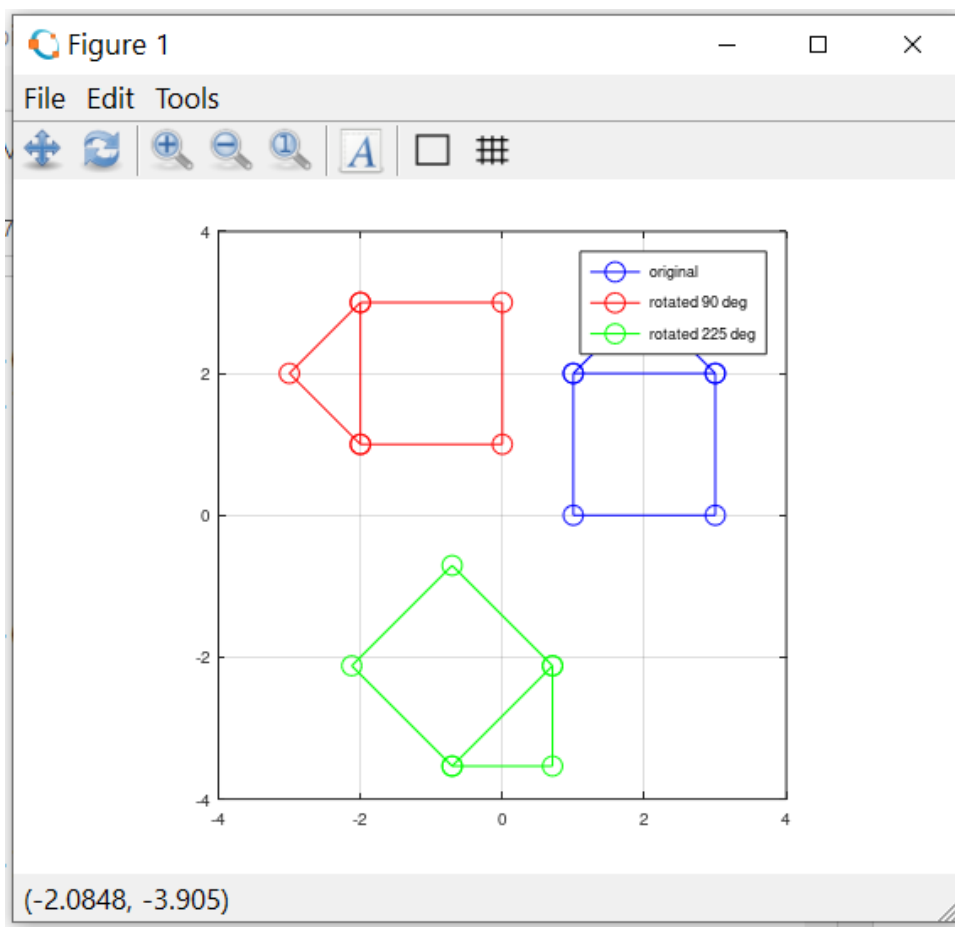
   -3.5355

>> plot (x, y, 'bo-', x1, y1, 'ro-', x2, y2, 'go-')
>> |

```



```
>> axis ([-4 4 -4 4], 'equal');  
>> grid on;  
>> legend ('original', 'rotated 90 deg', 'rotated 225 deg');  
>> |
```



Отражение

Отразим граф дома относительно прямой $y = x$. Зададим матрицу отражения (поясните, почему она такая).

```

>> 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 =

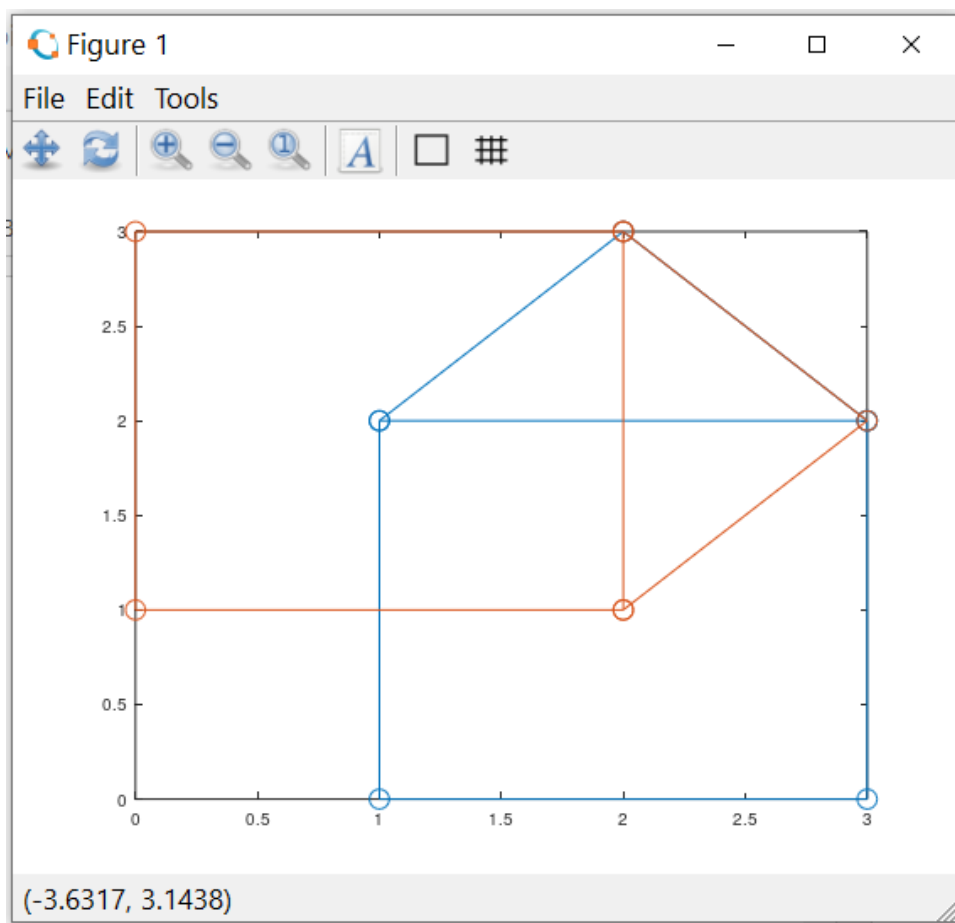
    2    0    0    2    3    2    2

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

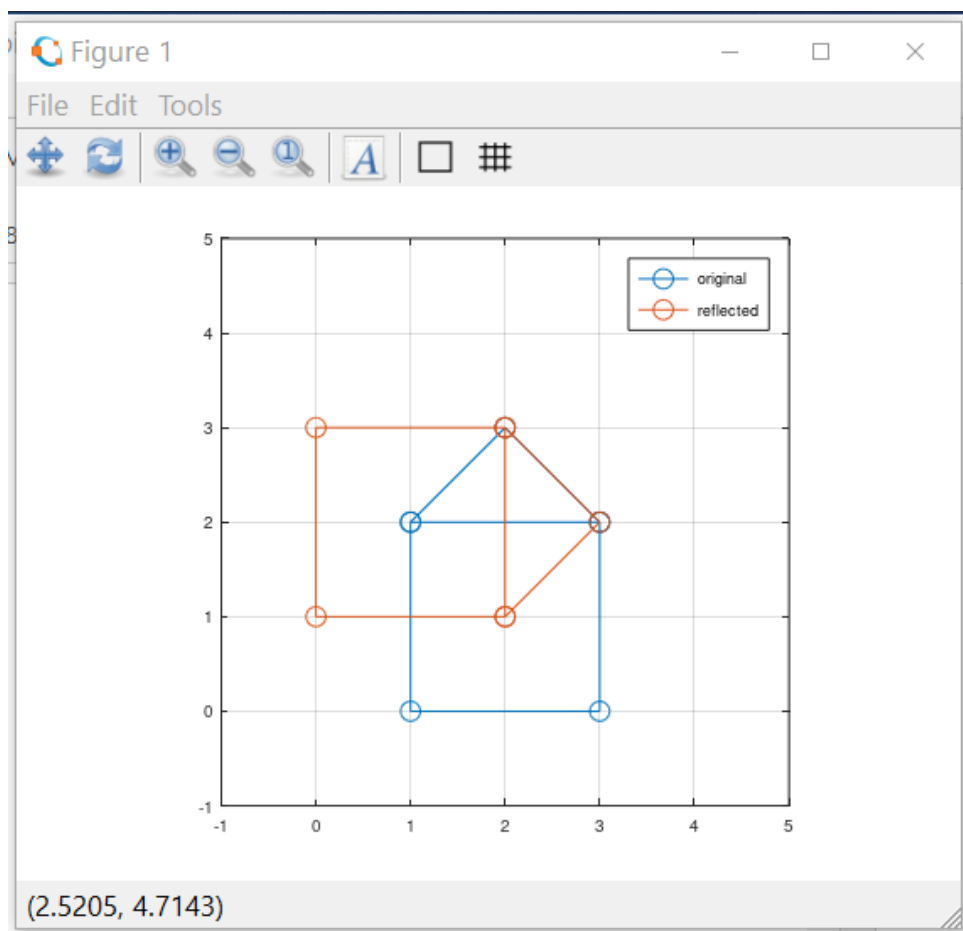
    1    1    3    3    2    1    3

>> plot(x, y, 'o-', x1, y1, 'o-')

```



```
>> axis([-1 4 -1 4], 'equal');  
>> axis([-1 5 -1 5], 'equal');  
>> grid on;  
>> legend ('original', 'reflected')  
>> |
```

Дилатация

Увеличим граф дома в 2 раза.

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

```
T =
```

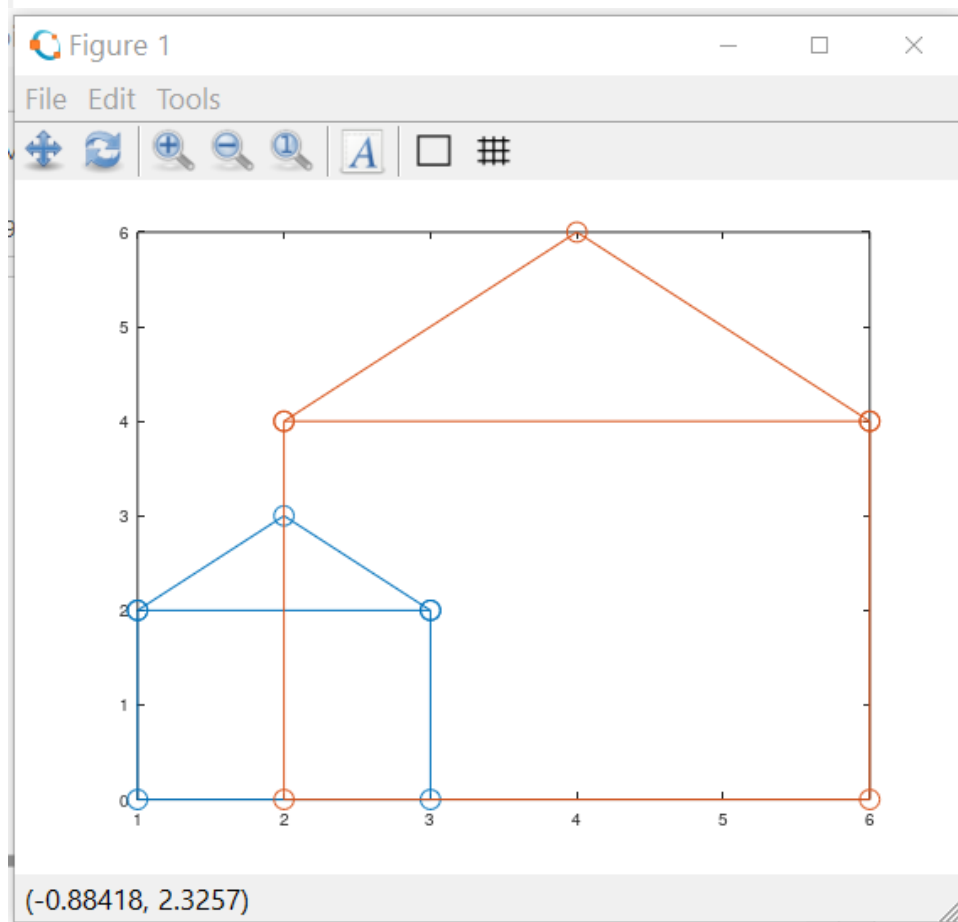
```
2    0
0    2
```

```
>> TD = T*D;
```

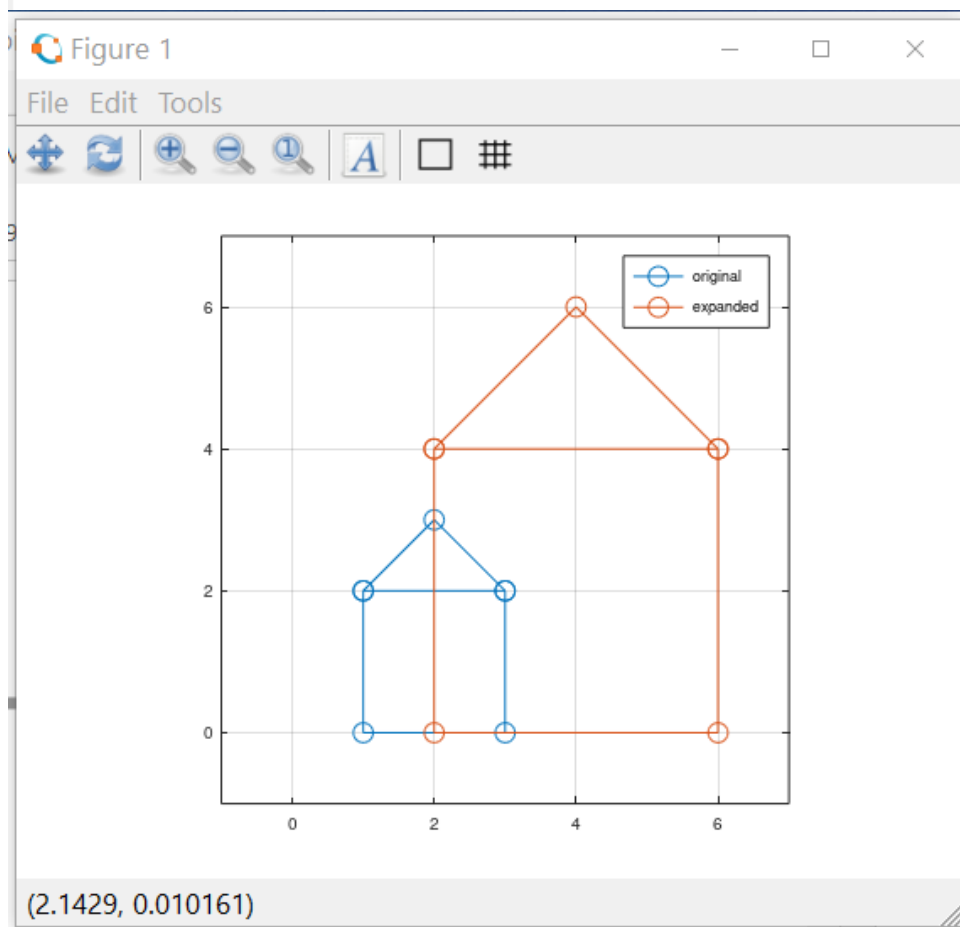
```
>> x1 = TD(1, :); y1 = TD(2, :);
```

```
>> plot (x, y, 'o-', x1, y1, 'o-')
```

```
>> |
```



```
>> axis([-1 7 -1 7], 'equal');  
>> grid on;  
>> legend('original', 'expanded')  
>> |
```



Вывод

В octave научился работать с подгонки полиномиальной кривой, с матричным преобразованием, вращений, отражений и дилатацией.