

Отчёт по лабораторной работе №5

Графики и операции над ними

Виктория Михайловна Шутенко, НФИбд-03-19

Содержание

Цель работы	3
Выполнение лабораторной работы	4
Подгонка полиномиальной кривой	4
Матричные преобразования	11
Вращение	13
Отражение	16
Дилатация	18
Выводы	21

Цель работы

Приобрести практические навыки работы с графиками и преобразованиями над ними в Octave.

Выполнение лабораторной работы

Подгонка полиномиальной кривой

1. Для начала я задала матрицу D.

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

- Далее я извлекла векторы x и y (Рис. 01):

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

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

- Потом нарисова точки на графике:

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

- Получился графийк 1 (рис. 02).

```

Last login: Thu May 20 17:54:54 on ttys000
shutenkovika@MacBook-Air-Viktoria ~ % cd work/2020-2021/"введение в научное программирование"
/laboratory/lab5
shutenkovika@MacBook-Air-Viktoria lab5 % ls
academic-laboratory-report-template  academic-presentation-markdown-template
shutenkovika@MacBook-Air-Viktoria lab5 % octave
GNU Octave, version 6.2.0
Copyright (C) 2021 The Octave Project Developers.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.

Octave was configured for "x86_64-apple-darwin20.3.0".

Additional information about Octave is available at https://www.octave.org.

Please contribute if you find this software useful.
For more information, visit https://www.octave.org/get-involved.html

Read https://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

[octave:1> diary on
[octave: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

[octave:3> xdata = D(:,1)
xdata =

    1
    2
    3
    4
    5
    6

[octave:4> ydata = D(:,2)
ydata =

    1
    2
    5
    4
    2
   -3

[octave:5> plot(xdata,ydata,'o-')
[octave:6> FALLBACK (log once): Fallback to SW vertex for line stipple

```

Рис. 0.1: задание матрицы D; извлечение векторов x и y; построение графика

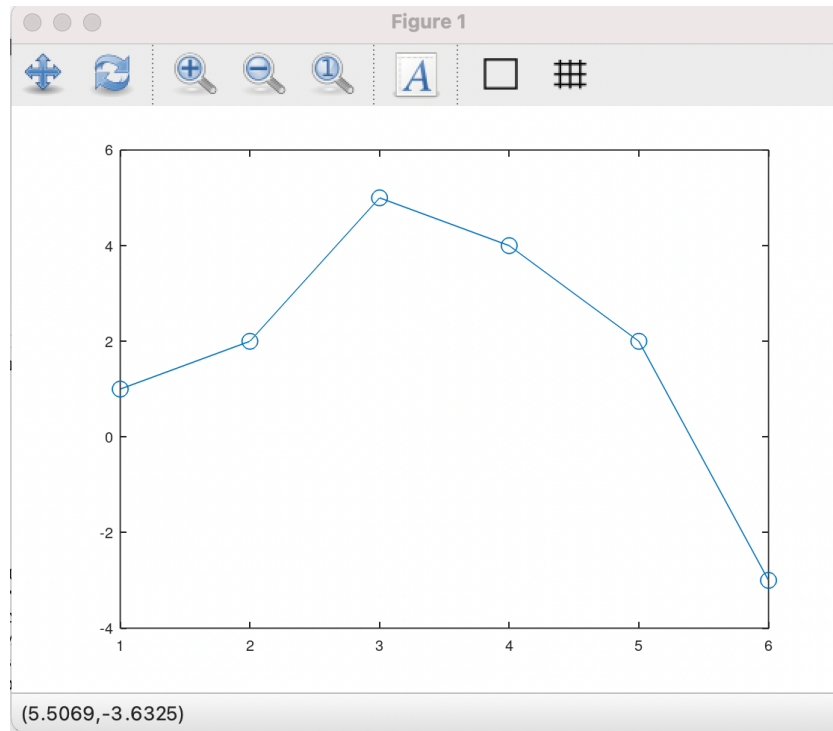


Рис. 0.2: график 1

- Далее я построила уравнение вида:

$$>> y = ax^2 + bx + c$$

- Данная матрица имеет особый вид: первый столбец содержит квадраты значений x , второй столбец - значения x , а третий - все единицы. Для того чтобы построить матрицу коэффициентов, я использовала команду `ones`. Так я задала матрицу единиц соответствующего размера, а затем переписала первый и второй столбцы соответствующими данными. (Рис. 03)

$$>> A = \text{ones}(6, 3)$$

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

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

```

[octave:6> print graph1.png -dpng
[octave:7> print('graph1.pdf', '-dpdf')
[octave:8> A = ones(6,3)
A =

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

[octave:9> A(:,1) = xdata .^ 2
A =

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

[octave:10> A(:,2) = xdata
A =

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

[octave:11> 

```

Рис. 0.3: Задание матрицы единиц

- Потом я использовала Octave для решения по методу наименьших квадратов, через решение уравне

$$>> A^T A b = A^T y$$

- где y - вектор коэффициентов полинома. (Рис. 04)

$$>> A' * A$$

$$>> A' * ydata$$

- Далее задача решалась методом Гаусса. Для начала я задала расширенную матрицу B.

$$>> \hat{A} = A' * A$$

$$>> \hat{A}(:,4) A' * ydata$$

$$>> B_{res} = rref(B)$$

```
>> a1 = B - res(1,4)
```

```
>> a2 = B - res(2,4)
```

```
>> a3 = B - res(3,4)
```

- Квадратное уравнение приняло вид:

```
>> y = -0,89286x2 + 5,65x - 4,4
```

```
[octave:11> A'*A
ans =

    2275    441    91
    441     91    21
     91     21     6

[octave:12> A' * ydata
ans =

    60
    28
    11

[octave:13> B = A' * A;
[octave:14> B(:,4) = A' * ydata;
[octave:15> B_res = rref (B)
B_res =

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

[octave:16> a1=B_res(1,4)
a1 = -0.8929
[octave:17> a2=B_res(2,4)
a2 = 5.6500
[octave:18> a3=B_res(3,4)
a3 = -4.4000
[octave:19> █
```

Рис. 0.4: решение по методу наименьших квадратов.

- Далее, я выполнила построение графика параболы введя (Рис. 05):

```
>> x = linspace(0,7,50;)
```

```
>> y = a1 * x.^2 + a2 * x + a3
```

```
>> plot(xdata,ydata,'o',x,y,'linewidth',2)
```

```
>> gridon
```



```
>> legend('data values','least-squares parabola')
```

```
>> title('y = -0,89286x^2 + 5,65x - 4,4')
```

- Получился график 2. (Рис. 06)

```
octave:19> x = linspace (0,7,50);
octave:20> y = a1 * x .^ 2 + a2 * x + a3;
octave:21> plot (xdata,ydata, 'o' ,x,y, 'linewidth', 2)
octave:22> grid on;
octave:23> legend ('data values', 'least-squares parabola')
octave:24> title ('y = -0.89286 x^2 + 5.65 x - 4.4')
octave:25> 
```

Рис. 0.5: Построение графика параболы

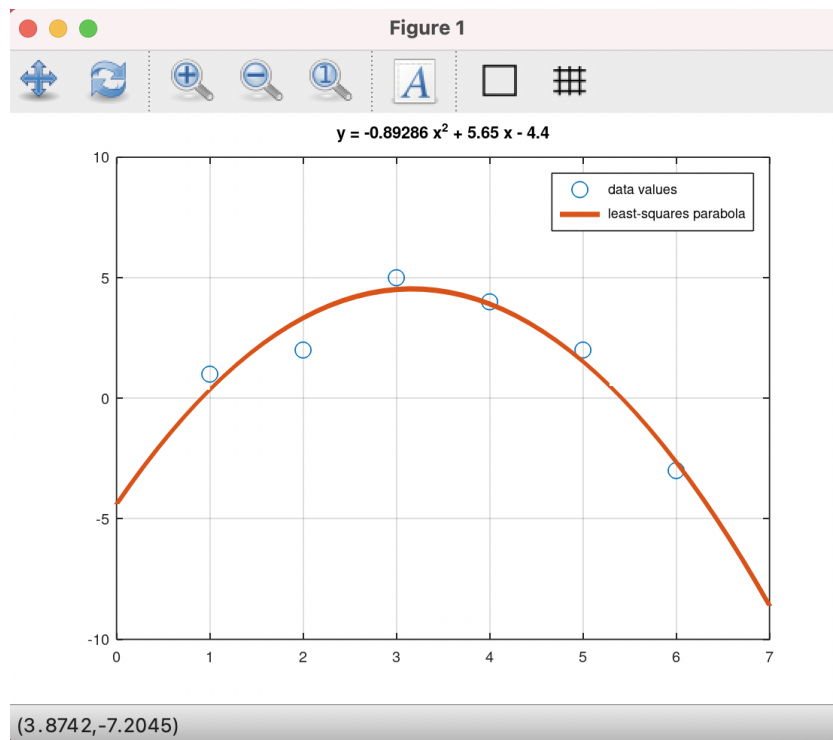


Рис. 0.6: График параболы

- Далее выполнил подгонку, используя команду polyfit (Рис. 07):

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

- Расчитала значение полинома в точках:

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

- Построила исходные и подгоночные данные и получила график 3. (Рис. 08)

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

```
>> gridon
```

```
>> legend('originaldata', 'polyfitdata')
```

```
[octave:27> P = polyfit (xdata, ydata, 2)
P =

    -0.8929    5.6500   -4.4000

[octave:28> y = polyval (P,xdata)
y =

    0.3571
    3.3286
    4.5143
    3.9143
    1.5286
   -2.6429

[octave:29> plot(xdata,ydata,'o-',xdata,y,'+-')
[octave:30> grid on ;
[octave:31> legend ('original data' , 'polyfit data' ) ;
[octave:32> ]
```

Рис. 0.7: Выполнение подгонки и построение графика

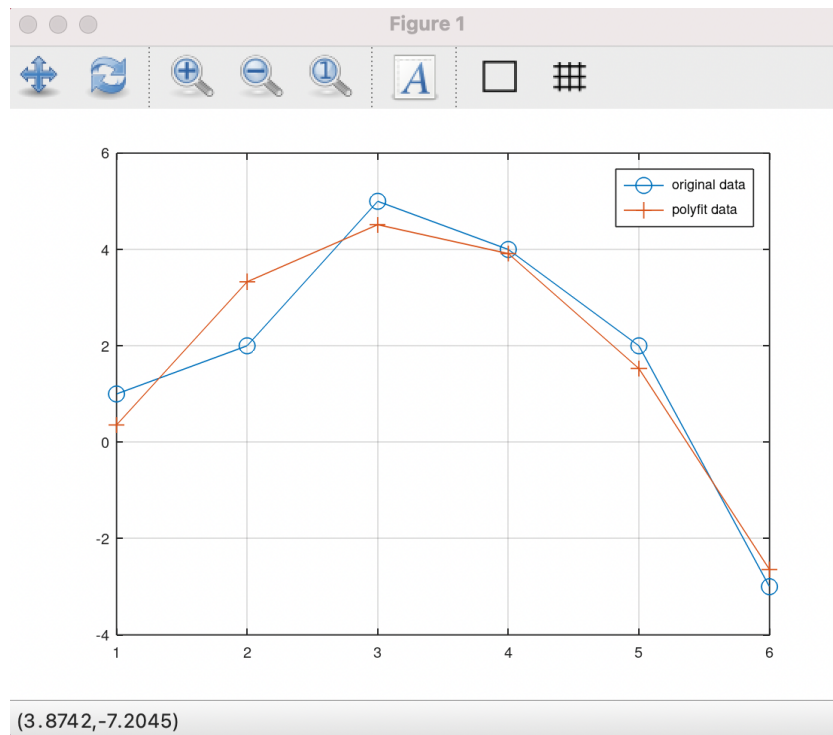


Рис. 0.8: График 3

Матричные преобразования

1. Далее я работала с построением графа матрицы. В методичке рассматривается метод перечисления ряда вершин, соединенных последовательно. Я строила граф-домик по методу Эйлера.
 - Сначала задала матрицу D.(Рис.9)

```
>> D = [1133213; 2002322]
```

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

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

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

- Получился граф-домик. (Рис. 10) - Выделила из расширенной матрицы B матрицу A и вектор b:

$$>> A = B(:, 1 : 3)$$

$$>> b = B(:, 4)$$

```
[octave:34> clear
[octave:35> clf
[octave:36> 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

[octave:37> x = D(1,:)
x =

     1     1     3     3     2     1     3

[octave:38> y = D(2,:)
y =

     2     0     0     2     3     2     2

[octave:39> plot (x,y)
[octave:40> ]
```

Рис. 0.9: Построение графа-домика

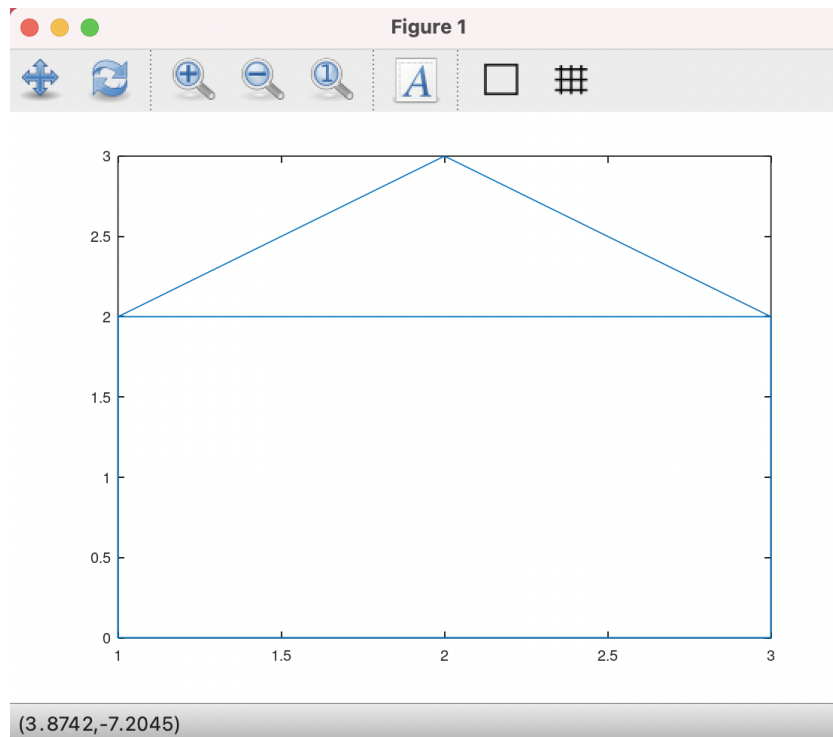


Рис. 0.10: График 4: Граф-домик

Вращение

1. Потом я перешла к вращениям. Вращения получаюся с использованием умножения на специальную матрицу R. Выполнила повороты матрицы D, используя произведение матриц RD. Я должна была выполнить поворот графа-домика на 90 и 225. Для этого я выполнила следующие действия:

- Перевела угол в радианы для угла 90:

```
>> theta1 = 90 * pi/180
```

```
>> R1 = [cos(theta1) - sin(theta1); sin(theta1) cos(theta1)]
```

```
>> RD1 = R1 * D
```

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

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

- Затем я перевела угол в радианы для угла 225:

```
>> theta2 = 225 * pi/180
```

```
>> R2 = [cos(theta2) - sin(theta2); sin(theta2) cos(theta2)]
```

```
>> RD2 = R2 * D
```

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

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

- Далее я построила новый график по полученным данным:

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

```
>> axis([-44 - 44], 'equal');
```

```
>> gridon
```

```
>> legend('original', 'rotated90deg', 'rotated225deg');
```

```

[octave:67> theta1 = 90*pi/180
theta1 = 1.5708
[octave:68> R1 = [cos(theta1) -sin(theta1); sin(theta1) cos(theta1)]
R1 =

    6.1232e-17   -1.0000e+00
    1.0000e+00    6.1232e-17

[octave:69> RD1 = R1*D
RD1 =

   -2.0000e+00    6.1232e-17    1.8370e-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

[octave:70> x1 = RD1(1,:)
x1 =

   -2.0000e+00    6.1232e-17    1.8370e-16   -2.0000e+00   -3.0000e+00   -2.0000e+00   -2.0000e+00

[octave:71> y1 = RD1(2,:)
y1 =

    1.0000    1.0000    3.0000    3.0000    2.0000    1.0000    3.0000

[octave:72> theta2 = 225*pi/180
theta2 = 3.9270
[octave:73> R2 = [cos(theta2) -sin(theta2); sin(theta2) cos(theta2)]
R2 =

   -0.7071    0.7071
   -0.7071   -0.7071

[octave:74> 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

[octave:75> x2 = RD2(1,:)
x2 =

    0.7071   -0.7071   -2.1213   -0.7071    0.7071    0.7071   -0.7071

[octave:76> y2 = RD2(2,:)
y2 =

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

[octave:77> plot (x,y, 'bo-', x1 , y1 , 'ro-' , x2 , y2 , 'go-' )
[octave:78> axis ([-4 4 -4 4] , 'equal' ) ;
[octave:79> grid on ;
[octave:80> legend ('original' , 'rotated 90 deg' , 'rotated 225 deg' ) ;
[octave:81>

```

Рис. 0.11: Перевод градусов в радианы и построение графика 5

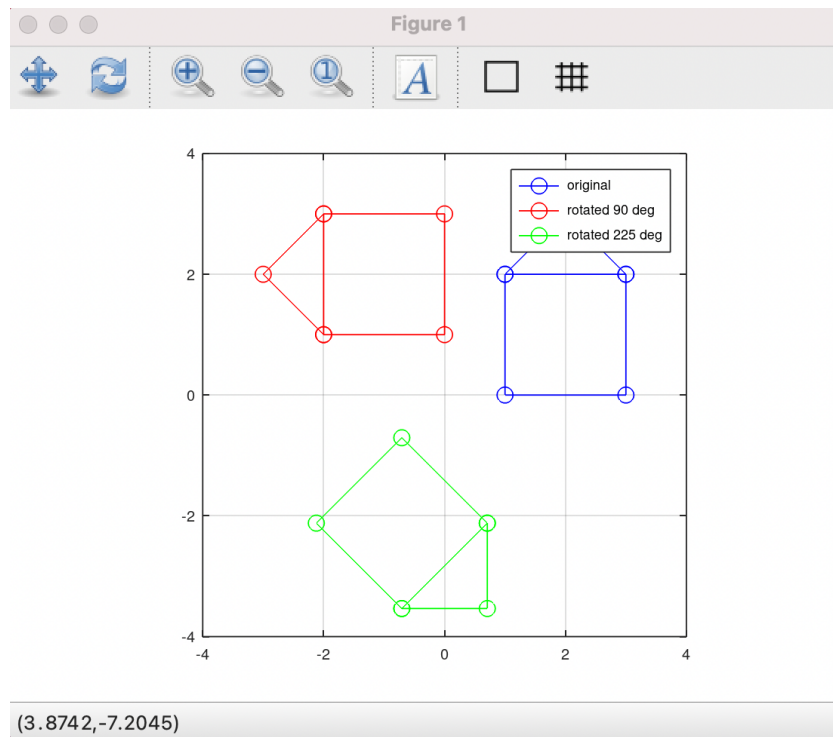


Рис. 0.12: График 5: вращение 90 и 225 градусов

Отражение

1. В этом пункте я отражала граф-дом относительно прямой $y = x$:

- Сначала я задала матрицу отражения:

$$>> R = [01; 10]$$

$$>> RD = R * D$$

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

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

- Потом построила график по полученным данным:

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

```
>> axis([-44 - 44], 'equal');
```

```
>> axis([-15 - 15], 'equal');
```

```
>> grid on
```

```
>> legend('original', 'reflected');
```

```
[octave:81> print graph5.png -dpng
[octave:82> print('graph5.pdf', '-dpdf')
[octave:83> R = [0 1; 1 0]
R =

    0    1
    1    0

[octave:84> RD = R * D
RD =

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

[octave:85> x1 = RD(1,:);
x1 =

    2    0    0    2    3    2    2

[octave:86> y1 = RD(2,:);
y1 =

    1    1    3    3    2    1    3

[octave:87> plot (x,y,'o-',x1,y1,'o-')
[octave:88> axis([-1 4 -1 4], 'equal');
[octave:89> axis([-1 5 -1 5], 'equal');
[octave:90> grid on ;
[octave:91> legend ( 'original' , 'reflected' )
[octave:92> ]
```

Рис. 0.13: Задание отражения через матрицу отражения

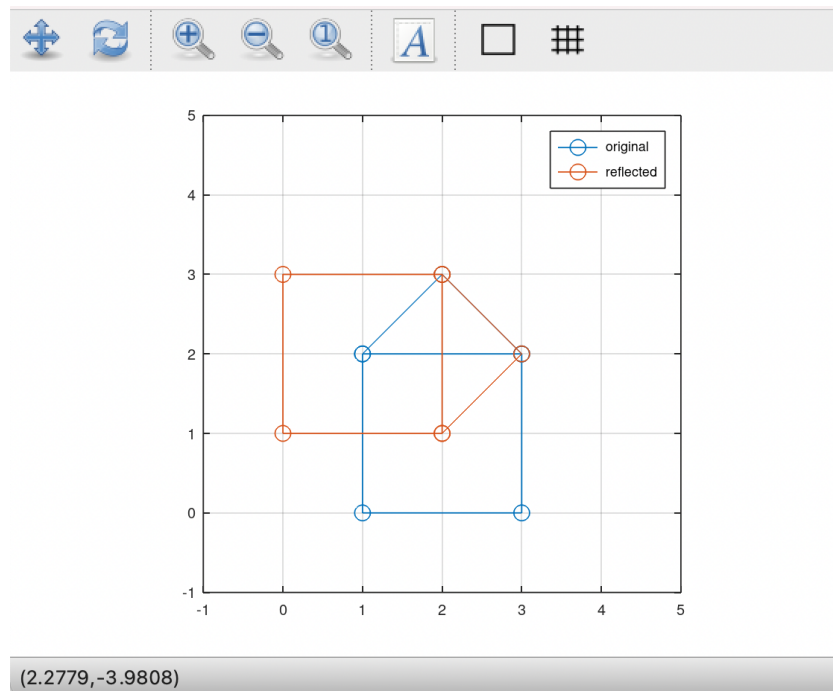


Рис. 0.14: График6: отражения

Дилатация

1. Дилатация - это расширение или сжатие, которое тоже выполняется перемножением матриц. Я должна была увеличить грааф-домик в 2 раза:

- Задала матрицу дилатации:

$$>> T = [20; 02]$$

$$>> TD = T * D$$

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

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

- Наконец, я построила график по полученным данным:

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

```
>> axis([-17 -17], 'equal');
```

```
>> grid on
```

```
>> legend('original', 'expanded');
```

```
[octave:94> T = [2 0; 0 2]
T =

     2     0
     0     2

[octave:95> TD = T*D;
[octave:96> x1 = TD(1,:); y1 = TD(2,:);
[octave:97> plot (x, y, 'o-', x1, y1, 'o-')
[octave:98> axis ([-1 7 -1 7], 'equal');
[octave:99> grid on;
[octave:100> legend ('original', 'expanded')
[octave:101> print graph7.png -dpng
[octave:102> print('graph7.pdf', '-dpdf')
[octave:103> diary off
[octave:104> █
```

Рис. 0.15: Задание дилатации через матрицу дилатации

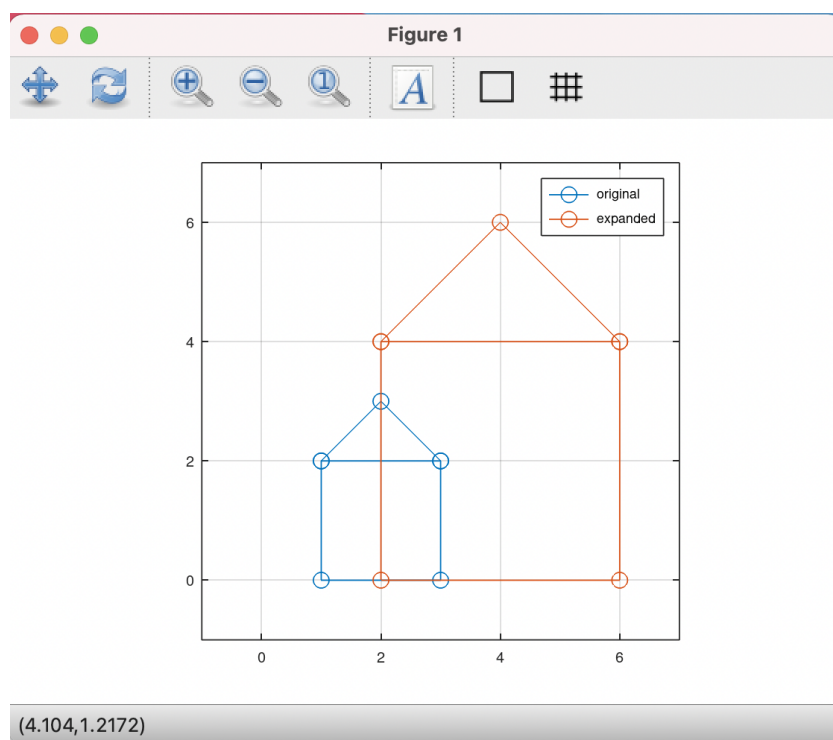


Рис. 0.16: График7: дилатация

Выводы

В ходе выполнения лабораторной работы я приобрела практические навыки работы с графиками и их преобразования в Octave.