

Лабораторная работа №5

Научное программирование

Александр Дмитриев

Содержание

1	Цель работы	4
2	Теоретическое введение	5
3	Выполнение лабораторной работы	6
4	Вывод	14

Список иллюстраций

3.1	Построение на графике точек из матрицы	7
3.2	Подгонка полиномиальной кривой	8
3.3	Подгонка полиномиальной кривой	9
3.4	Подгонка полиномиальной кривой	9
3.5	Подгонка полиномиальной кривой	10
3.6	Подгонка с помощью встроенных функций	11
3.7	Построение изображения по матрице	11
3.8	Поворот изображения	12
3.9	Поворот изображения	12
3.10	Отражение изображения	13
3.11	Дилатация изображения	13

1 Цель работы

Изучить в Octave методы подгонки полиномиальной кривой, способы представления изображения в виде матрицы и действия над ним: вращение, отражение и дилатацию.

2 Теоретическое введение

Интерполяция - способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений. Интерполяция функций часто встречается при ограниченности возможностей при проведении эксперимента. В частности из-за дороговизны и трудоемкости проведения эксперимента размер соответствующей выборки может быть достаточно мал.

Аппроксимация - замена одних математических объектов другими, в том или ином смысле близкими к исходным. При интерполировании интерполирующая функция строго проходит через узловые точки таблицы вследствие того, что количество коэффициентов в интерполирующей функции равно количеству табличных значений. Аппроксимация – метод приближения, при котором для нахождения дополнительных значений, отличных от табличных данных, приближенная функция проходит не через узлы интерполяции, а между ними.

Более подробно см. в [[@Octave_1: bash](#)] и [[@Octave_2: bash](#)].

3 Выполнение лабораторной работы

Найдем параболу по методу наименьших квадратов для набора точек, заданных матрицей

$$D = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 5 \\ 4 & 4 \\ 5 & 2 \\ 6 & -3 \end{bmatrix}.$$

В матрице заданы значения x в столбце 1 и значения y в столбце 2. Введем матрицу данных в Octave и извлечем вектора x и y , затем нарисуем точки на графике (рис. fig. 3.1).

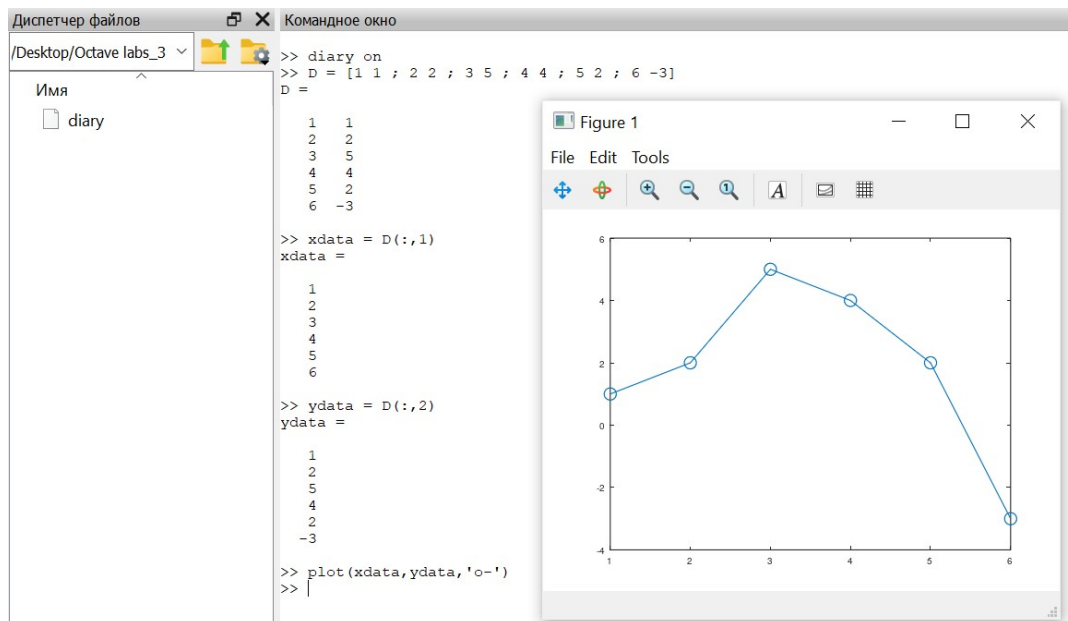


Рис. 3.1: Построение на графике точек из матрицы

Строим уравнение вида $y = ax^2 + bx + c$. Подставляя данные, получаем следующую систему линейных уравнений

$$\begin{bmatrix} 1 & 1 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \\ 16 & 4 & 1 \\ 25 & 5 & 1 \\ 36 & 6 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 5 \\ 4 \\ 2 \\ -3 \end{bmatrix}.$$

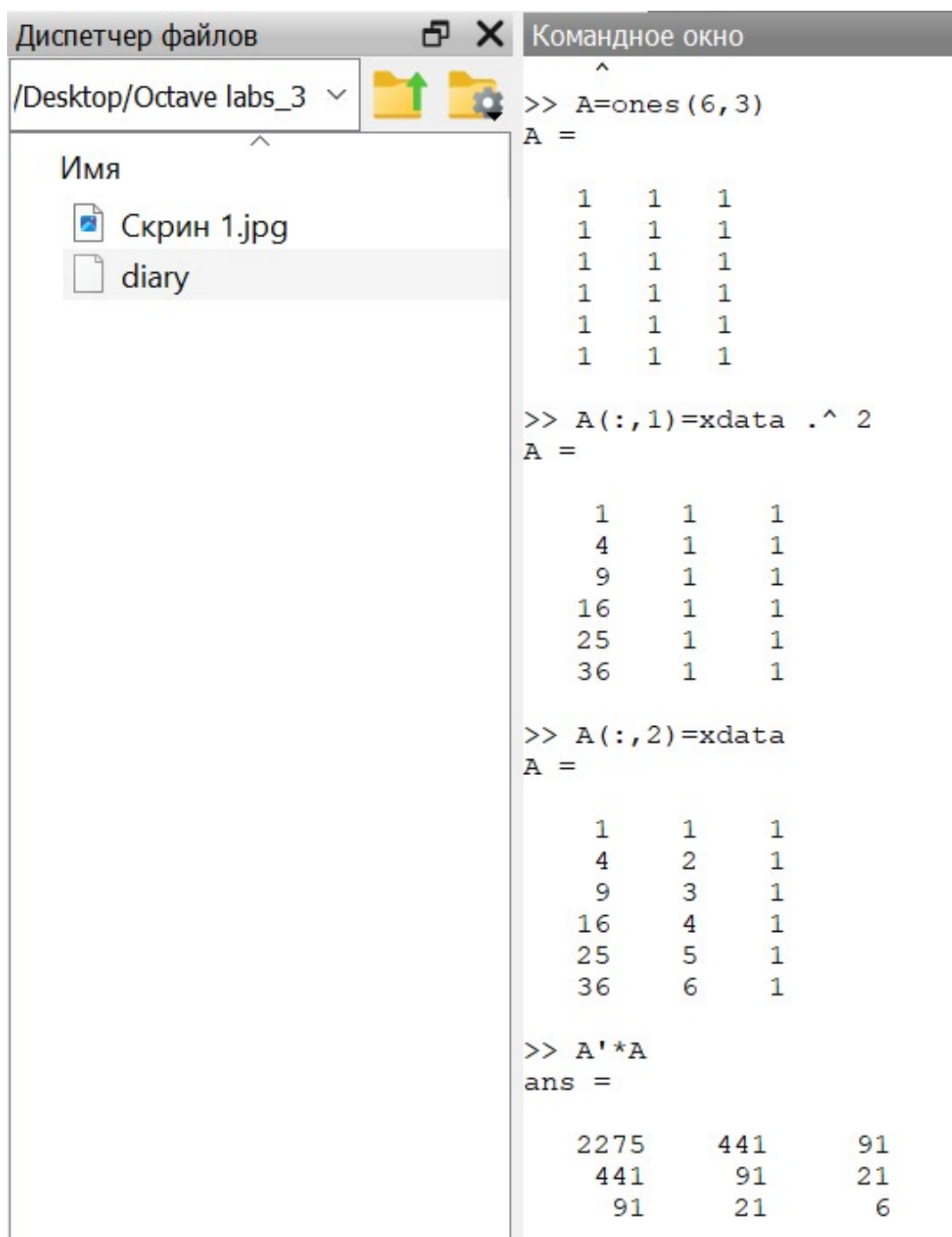
Решение по методу наименьших квадратов получается из решения уравнения $A^T A b = A^T y$, где b - вектор коэффициентов полинома. Строим соответствующие уравнения. Затем решаем задачу методом Гаусса, записав предварительно расширенную матрицу

$$B = \begin{bmatrix} 2275 & 441 & 91 & 60 \\ 441 & 91 & 21 & 28 \\ 91 & 21 & 6 & 11 \end{bmatrix}.$$

В итоге получаем искомое квадратное уравнение вида:

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

Строим соответствующий график параболы (рис. fig. 3.2) - (рис. fig. 3.5).



```
Диспетчер файлов /Desktop/Octave labs_3
Имя
Скрин 1.jpg
diary

Командное окно
^
>> 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

>> A'*A
ans =
   2275   441    91
   441    91    21
    91    21     6
```

Рис. 3.2: Подгонка полиномиальной кривой

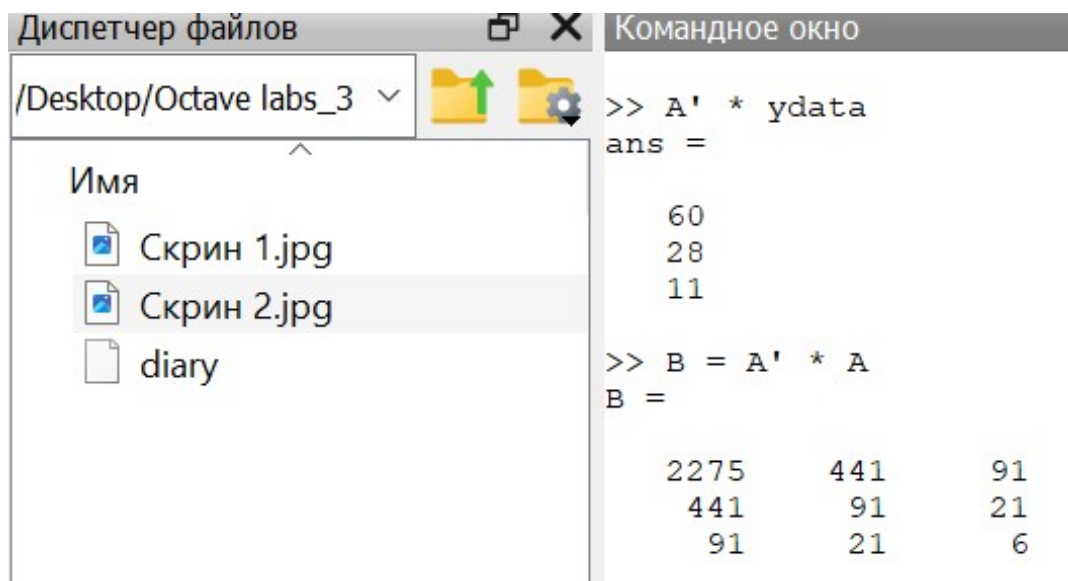


Рис. 3.3: Подгонка полиномиальной кривой

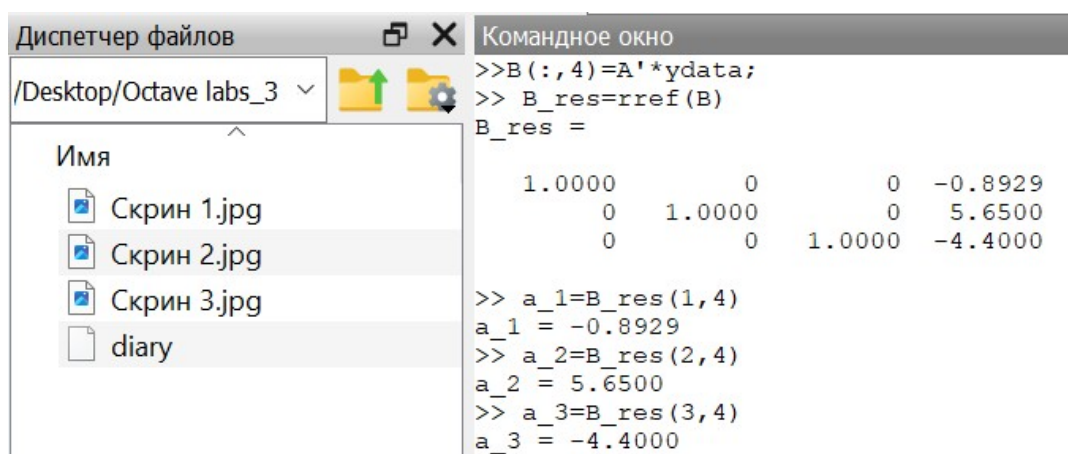


Рис. 3.4: Подгонка полиномиальной кривой

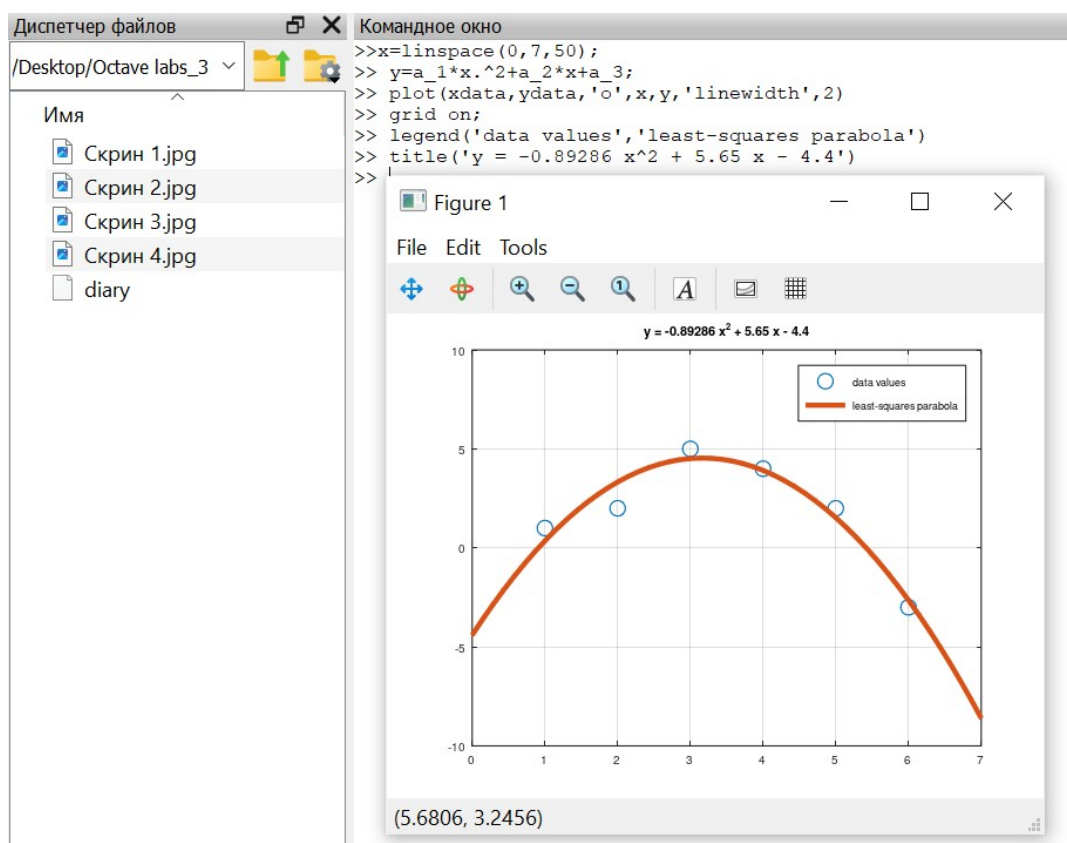


Рис. 3.5: Подгонка полиномиальной кривой

Процесс подгонки может быть автоматизирован встроенными функциями Octave. Для этого используется встроенная функция для подгонки полинома `polyfit`. Значения полинома P в точках, задаваемых вектором-строкой x можно получить с помощью функции `polyval`. Получим подгоночный полином и рассчитаем значения полинома в точках, а затем построим исходные и подгоночные данные на графике (рис. fig. 3.6).

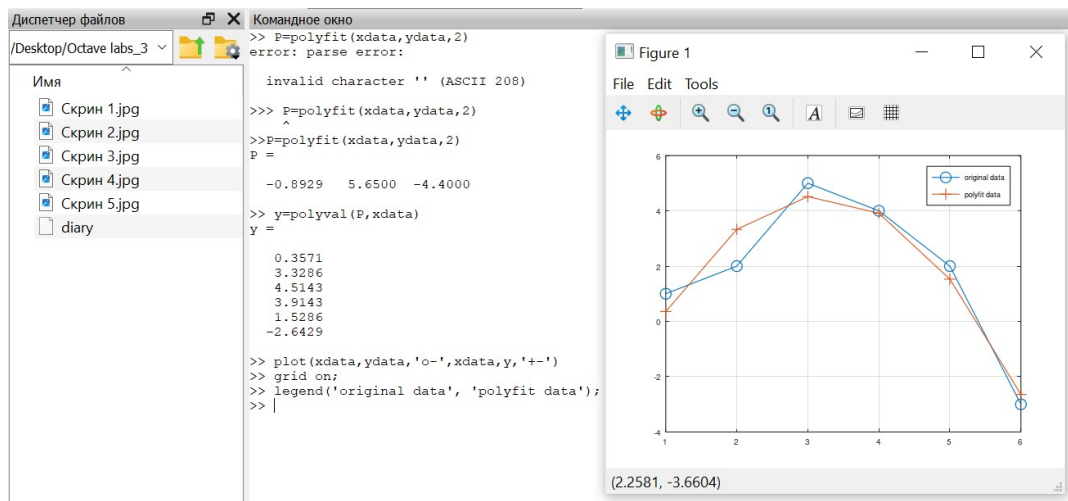


Рис. 3.6: Подгонка с помощью встроенных функций

Строим граф-домик с помощью матрицы, выбрав путь, который проходит по каждому ребру ровно один раз (цикл Эйлера) (рис. fig. 3.7):

$$D = \begin{bmatrix} 1 & 1 & 3 & 3 & 2 & 1 & 3 \\ 2 & 0 & 0 & 2 & 3 & 2 & 2 \end{bmatrix}.$$

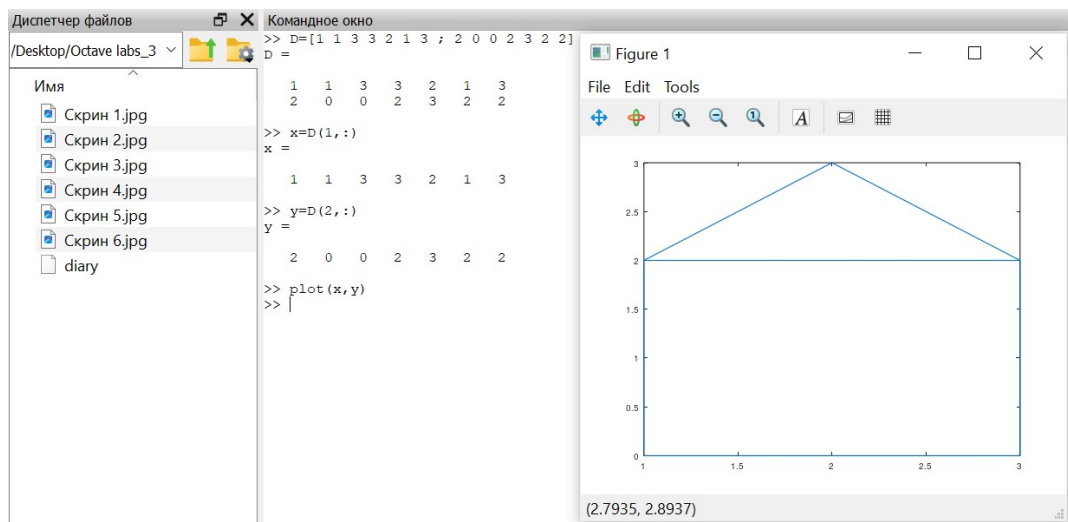


Рис. 3.7: Построение изображения по матрице

Осуществим поворот графа дома на 90 и 225 градусов, переведя углы в радианы, и построим соответствующие графики (рис. fig. 3.8) и (рис. fig. 3.9).

```

Диспетчер файлов  Командное окно
/Desktop/Octave labs_3

Имя
Скрин 1.jpg
Скрин 2.jpg
Скрин 3.jpg
Скрин 4.jpg
Скрин 5.jpg
Скрин 6.jpg
Скрин 7.jpg
diary

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

>> x1=RD1(1,:)
x1 =
   -2.0000e+00    6.1230e-17    1.8369e-16   -2.0000e+00   -3.0000e+00   -2.0000e

>> y1=RD1(2,:)
y1 =
     1     1     3     3     2     1     3

>> 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

```

Рис. 3.8: Поворот изображения

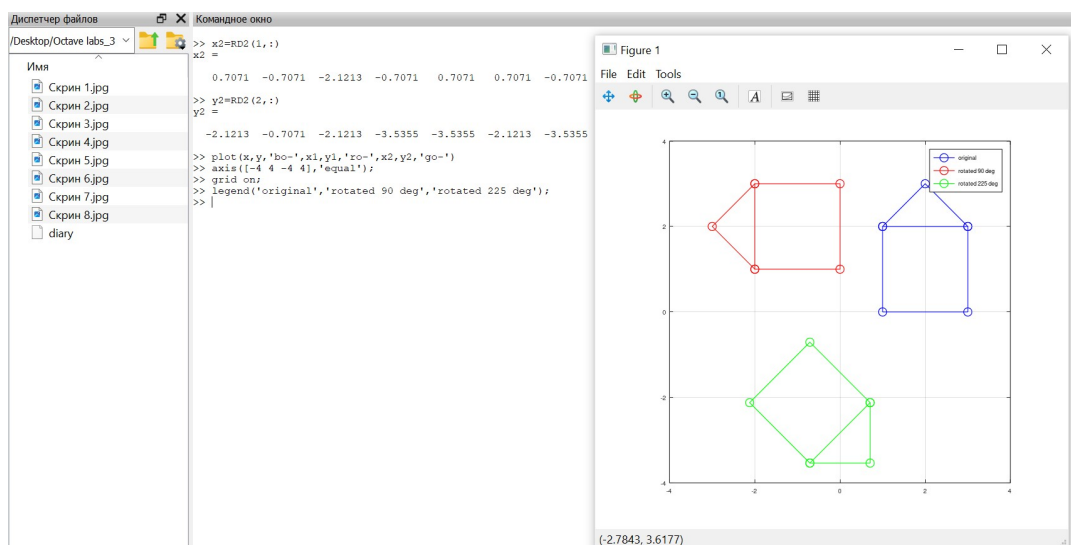


Рис. 3.9: Поворот изображения

Осуществим отражение графа дома относительно прямой $y = x$, задав матрицу отражения (рис. fig. 3.10).

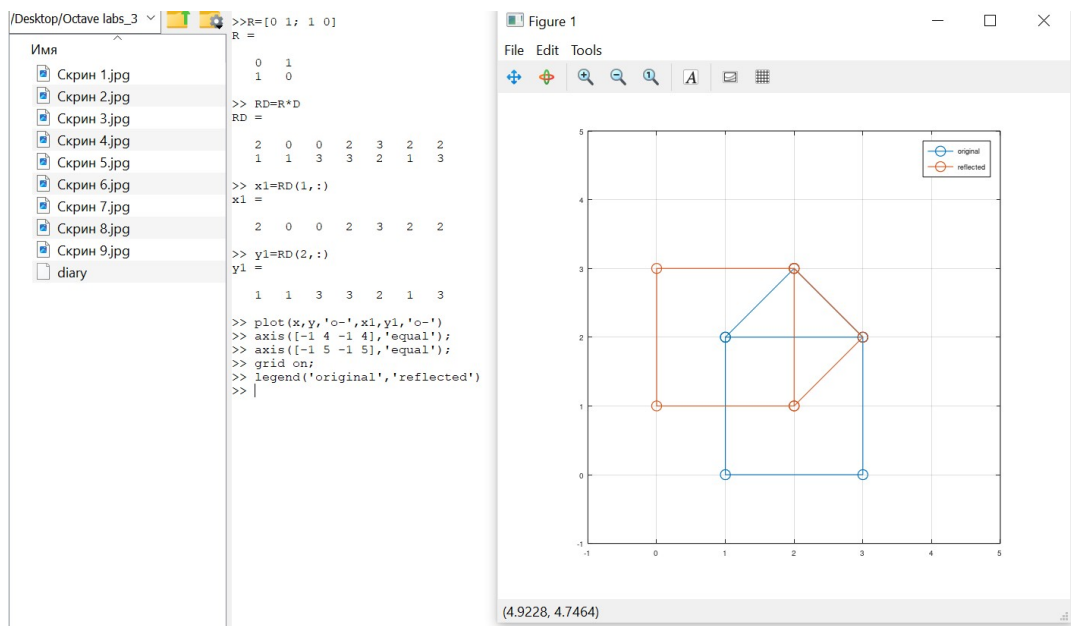


Рис. 3.10: Отражение изображения

Увеличим граф дома в 2 раза (рис. fig. 3.11).

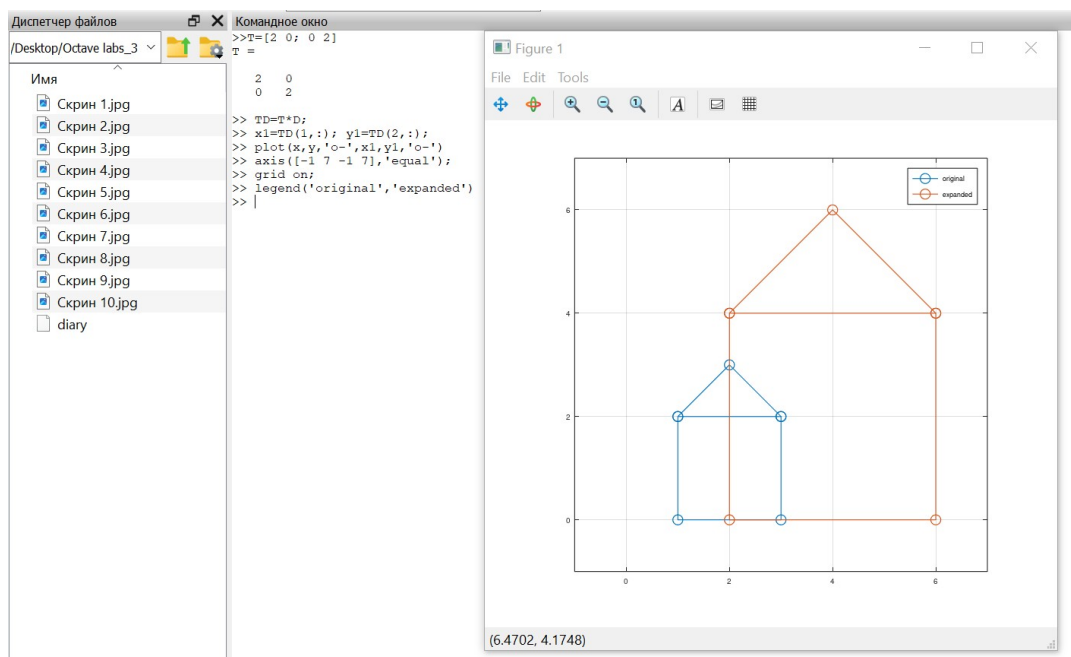


Рис. 3.11: Дилатация изображения

4 Вывод

В ходе выполнения данной лабораторной работы я изучил в Octave методы подгонки полиномиальной кривой, способы представления изображения в виде матрицы и действия над ним: вращение, отражение и дилатацию.