Лабораторная работа №5

Научное программирование

Минов Кирилл Вячеславович | НПМмд-02-23

Содержание

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 [1] и [2].

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.

```
>> 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
```

Рис. 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. 2) - (рис. fig. 5).

```
>> A'*A
ans =

   2275    441     91
    441     91     21
     91     21      6

>> A' * ydata
ans =

   60
   28
   11

>> 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
>> |
```

Рис. 2: Подгонка полиномиальной кривой

```
>> 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: Подгонка полиномиальной кривой

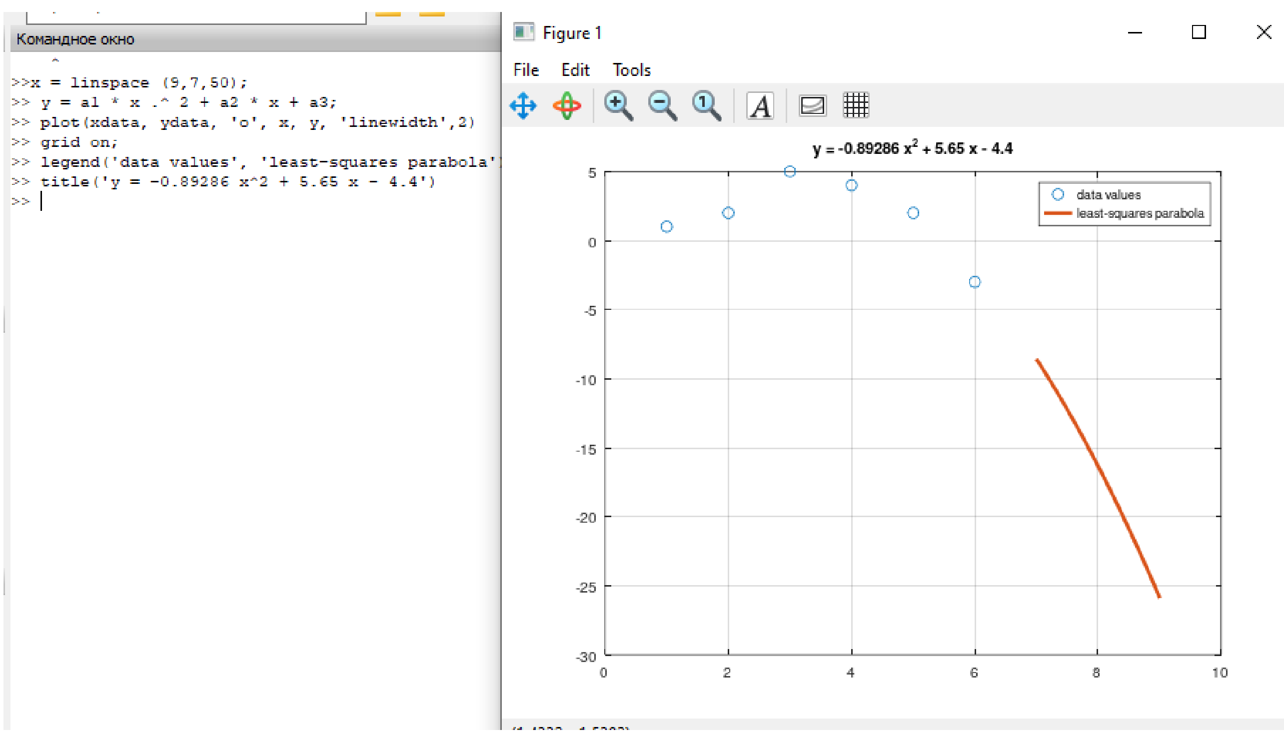

Рис. 4: Подгонка полиномиальной кривой

Строим граф-домик с помощью матрицы, выбрав путь, который проходит по каждому ребру ровно один раз (цикл Эйлера) (рис. fig. 7):

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

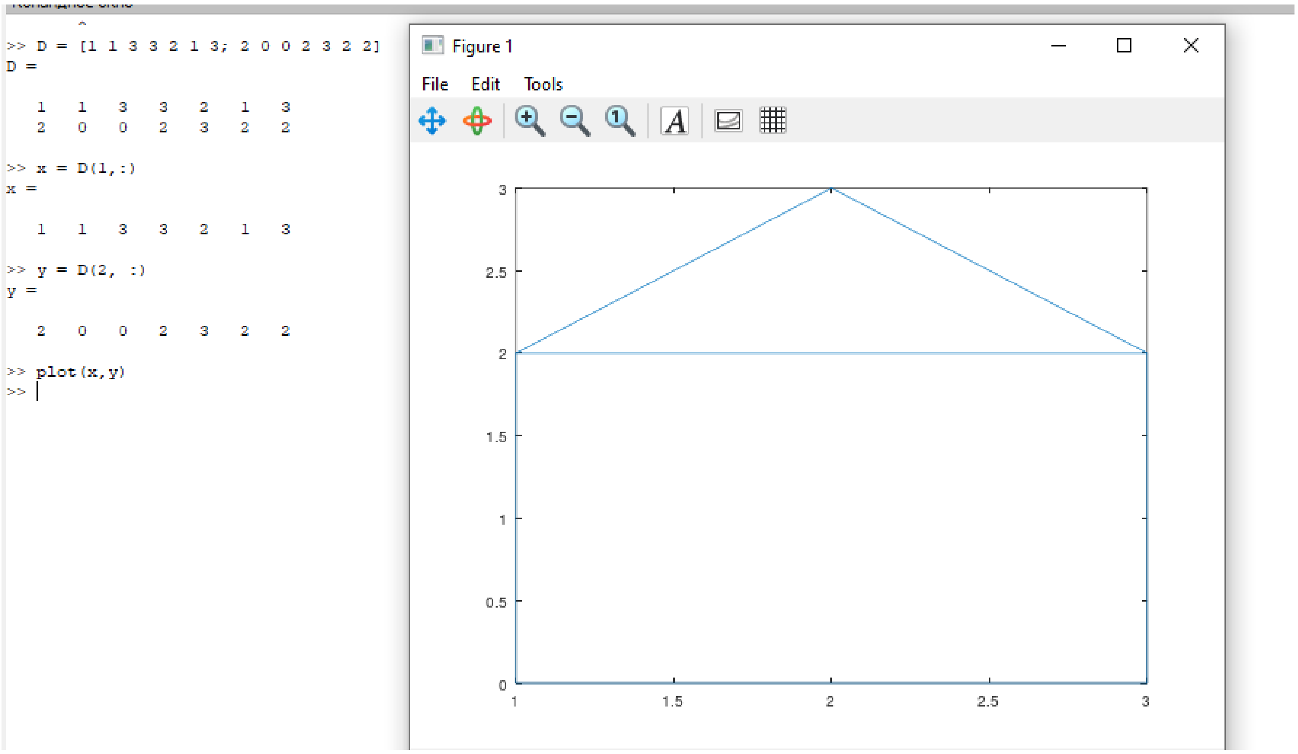

Рис. 5: Построение изображения по матрице

Осуществим поворот графа дома на 90 и 225 градусов, переведя углы в радианы, и построим соответствующие графики (рис. fig. 8) и (рис. fig. 9).

```
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+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.6: Поворот изображения

```
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

>> plot(x,y,'bo-',x1,y1, 'ro-',x2,y2,'go-')
>> axis([-4 4 -4 4], 'equal');
>> legend('original', 'rotated 90 deg', 'rotated 225 deg');
>> |
```

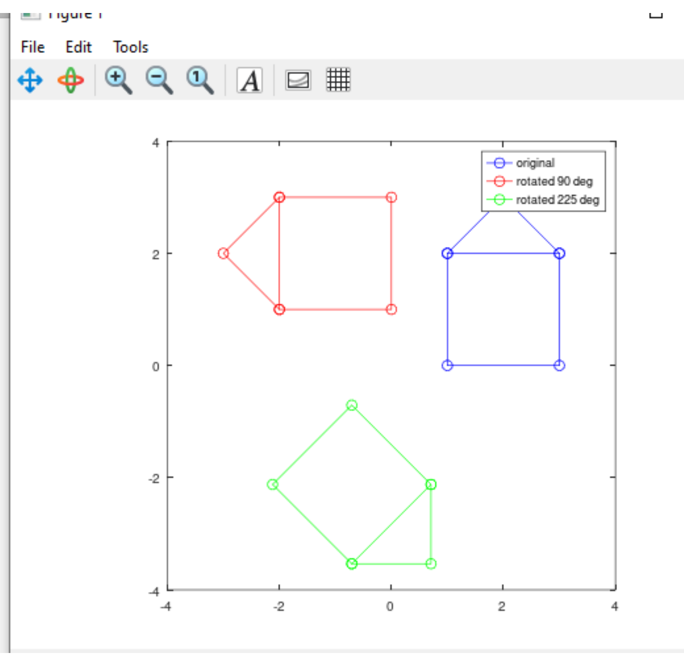

Рис. 7: Поворот изображения

Осуществим отражение графа дома относительно прямой $y = x$, задав матрицу отражения.

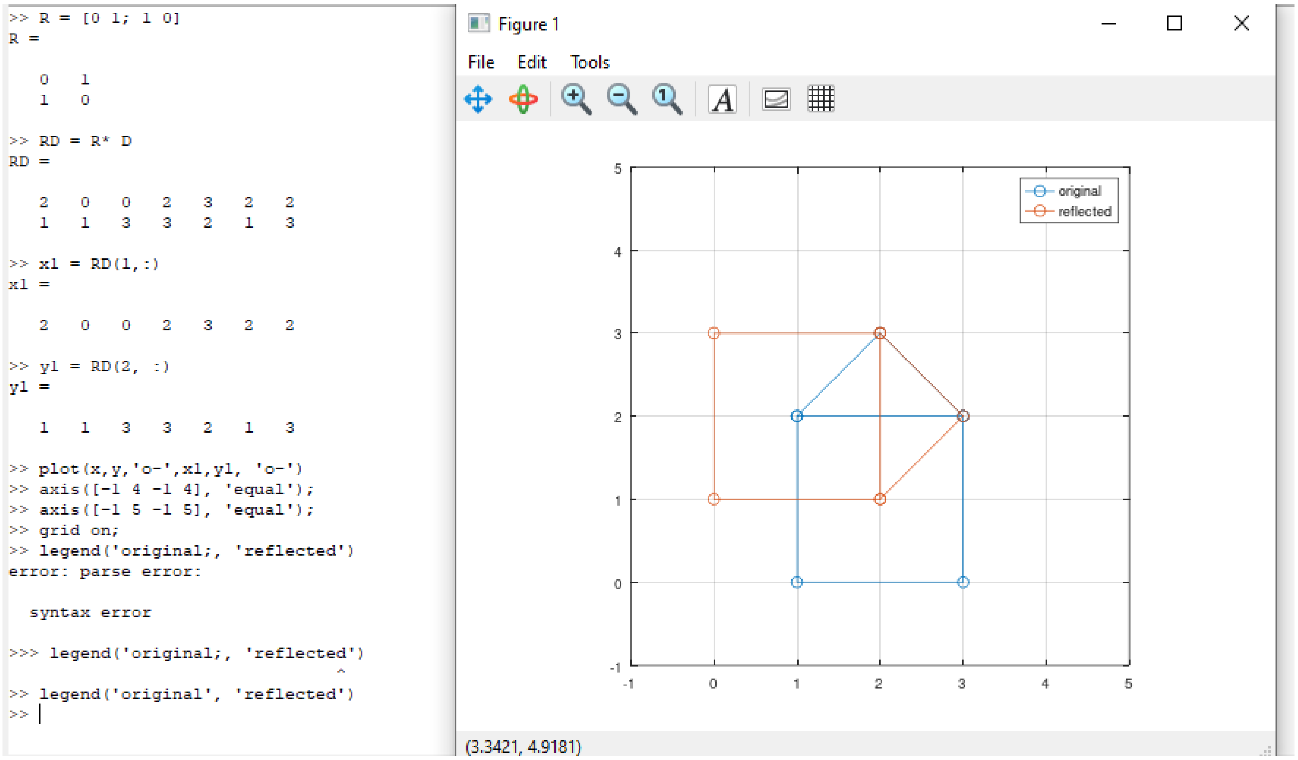

Рис. 8: Отражение изображения

Увеличим граф дома в 2 раза

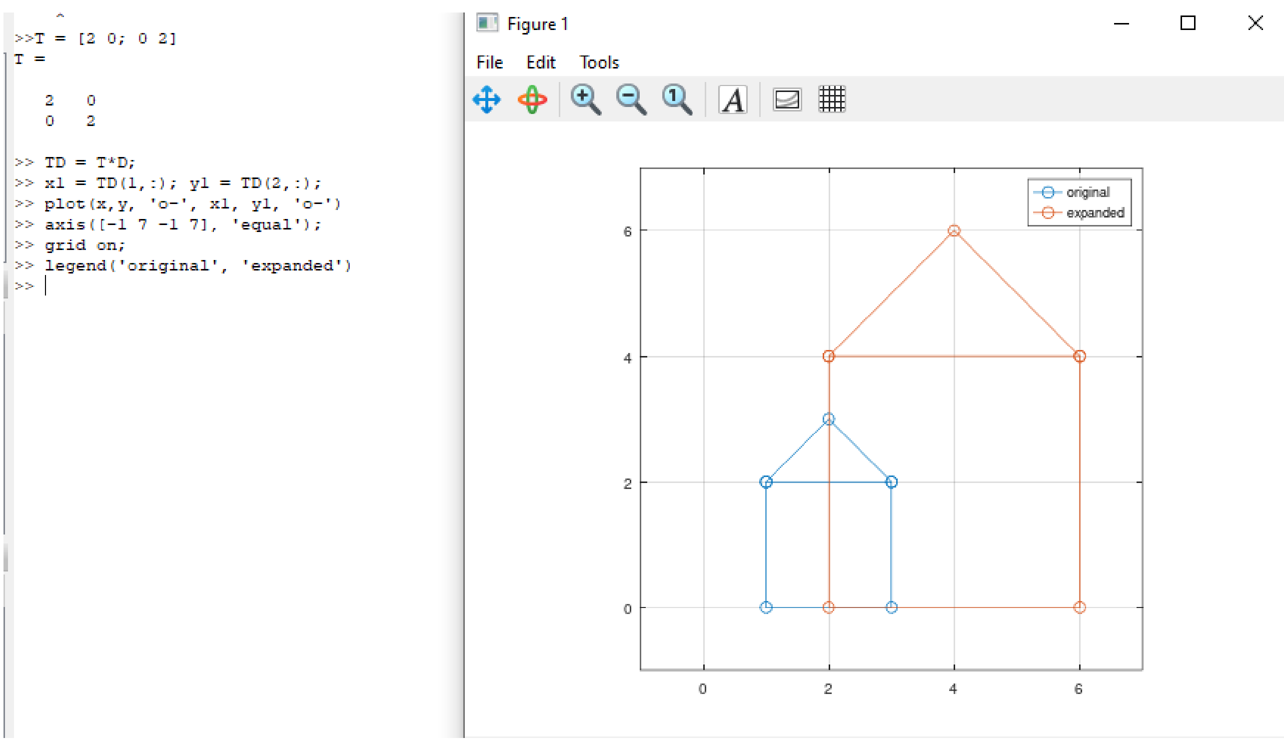

Рис. 9: Дилатация изображения

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.