

Лабораторная работа №5

Работа с матрицами

Демидова Екатерина Алексеевна

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	График точек, заданных матрицей D	8
4.2	Построение матрицы коэффициентов	9
4.3	Нахождение коэффициентов	10
4.4	Построение графика параболы	10
4.5	Построение графика исходных и подгоночных данных	11
4.6	Граф-домик	11
4.7	Построение повёрнутого графика дома	12
4.8	График домика, отраженный относительно прямой $y = x$	12
4.9	Увеличинный в 2 раза график домика	13

1 Цель работы

Научиться подгонять полиномиальные кривые и выполнять различные матричные преобразования с помощью системы для математических вычислений Octave.

2 Задание

- Выполнить подгонку полиномиальной кривой с помощью Octave.
- Представить изображение с помощью матрицы.
- Перевернуть изображение на определённый угол.
- Отразить изображение относительно прямой.
- Выполнить преобразование делитации.

3 Теоретическое введение

Подгонка кривой — это процесс построения кривой или математической функции, которая наилучшим образом соответствует ряду точек данных, возможно, с учетом ограничений[1]. Подгонка кривой может включать либо интерполяцию, где требуется точная подгонка к данным, либо сглаживание, при котором строится «гладкая» функция, которая приблизительно соответствует данным.

Если l — прямая, проходящая через начало координат, то **отражение** точки (x, y) относительно прямой l определяется как [2]

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

Дилатация (то есть расширение или сжатие) также может быть выполнено путём умножения матриц. Матричное произведение TD будет преобразованием дилатации D с коэффициентом k , где

$$\begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix}$$

4 Выполнение лабораторной работы

Решим более общую проблему подгонки полинома к множеству точек. Пусть нам нужно найти параболу по методу наименьших квадратов для набора точек, заданных матрицей

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

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 . Нарисуем точки на графике.(рис. [4.1])

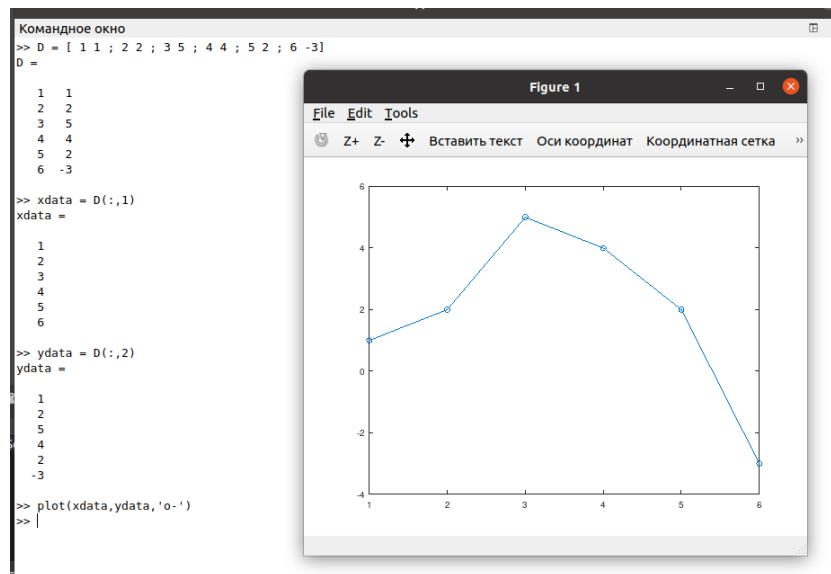


Рис. 4.1: График точек, заданных матрицей D

Построим уравнение вида $y = ax^2 + bx + c$. Для построения матрицы коэффициентов используем команду `ones` для создания матрицы единиц соответствующего размера, а затем перезапишем первый и второй столбцы необходимыми данными. (рис. [4.2])


```

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. 4.2: Построение матрицы коэффициентов

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. Решим задачу методом Гаусса. Запишем расширенную матрицу B . Таким образом, искомое квадратное уравнение имеет вид

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

(рис. [4.3])

```

>> A' * ydata
ans =
    60
    28
    11
>> B = A' * A;
>> B (:,4) = A' * ydata;
>> B
B =
    2275    441    91    60
    441    91    21    28
    91    21    6    11
>> B_res = rref (B)
B_res =
    1.00000    0.00000    0.00000   -0.89286
    0.00000    1.00000    0.00000    5.65000
    0.00000    0.00000    1.00000   -4.40000
>> a1=B_res(1,4)
a1 = -0.89286
>> a2=B_res(2,4)
a2 = 5.6500
>> a3=B_res(3,4)
a3 = -4.4000
>> |

```

Рис. 4.3: Нахождение коэффициентов

Построим соответствующий график параболы (рис. [4.4])

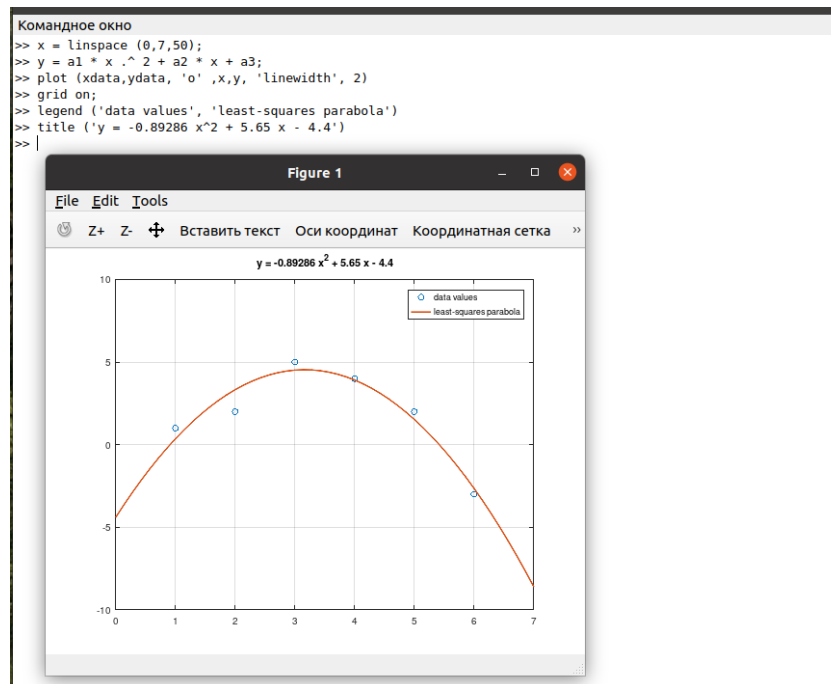


Рис. 4.4: Построение графика параболы

Процесс подгонки может быть автоматизирован встроенными функциями Octave. Для этого мы можем использовать встроенную функцию для подгонки полинома `polyfit`. Синтаксис: `polyfit (x, y, order)`, где `order` – это степень

полинома. Значения полинома P в точках, задаваемых вектором-строкой x можно получить с помощью функции `polyval`. Синтаксис: `polyval (P, x)`. Получим подгоночный полином. Рассчитаем значения полинома в точках, построим исходные и подгоночные данные (рис. [4.5])

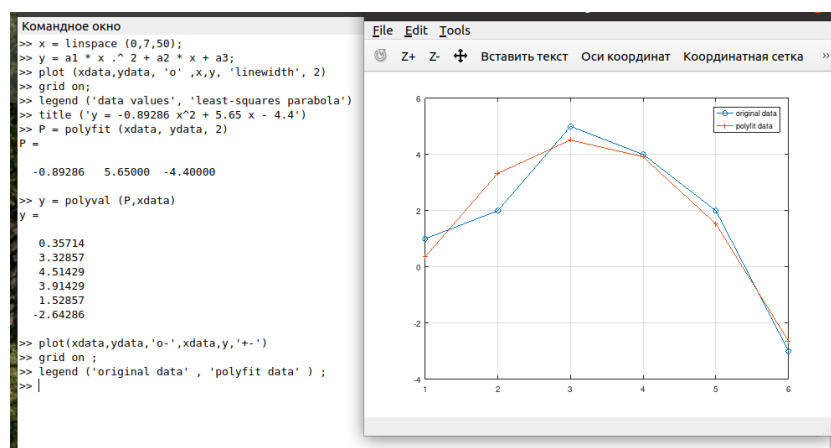


Рис. 4.5: Построение графика исходных и подгоночных данных

Попробуем закодировать граф-домик. Есть много способов закодировать это как матрицу. Эффективный метод состоит в том, чтобы выбрать путь, который проходит по каждому ребру ровно один раз (цикл Эйлера). (рис. [4.6])

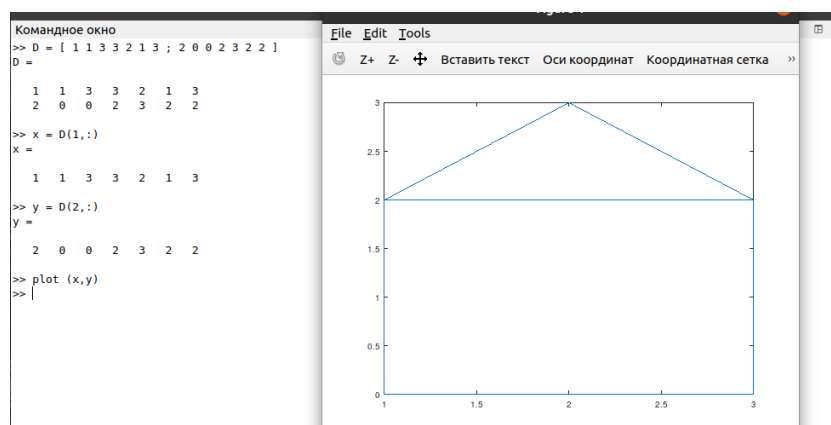


Рис. 4.6: Граф-домик

Повернём граф дома на 90 и 225 градусов. Вначале переведём угол в радианы, а затем воспользовавшись матрицей поворота повернём домик. (рис. [4.7])

```

>> plot(x,y)
>> theta1 = 90*pi/180
theta1 = 1.5708
>> R1 = [cos(theta1) -sin(theta1); sin(theta1) cos(theta1)]
R1 =
    6.1232e-17 -1.0000e+00
    1.0000e+00  6.1232e-17
>> 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
>> 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
>> y1 = RD1(2,:)
y1 =
    1.0000    1.0000    3.0000    3.0000    2.0000    1.0000    3.0000
>> theta2 = 225*pi/180
theta2 = 3.9270
>> R2 = [cos(theta2) -sin(theta2); sin(theta2) cos(theta2)]
R2 =
    0.70711    0.70711
   -0.70711   -0.70711
>> RD2 = R2*D
RD2 =
    0.70711   -0.70711   -2.12132   -0.70711    0.70711    0.70711   -0.70711
   -2.12132   -0.70711   -2.12132   -3.53553   -3.53553   -2.12132   -3.53553
>> x2 = RD2(1,:)
x2 =
    0.70711   -0.70711   -2.12132   -0.70711    0.70711    0.70711   -0.70711
>> y2 = RD2(2,:)
y2 =
   -2.12132   -0.70711   -2.12132   -3.53553   -3.53553   -2.12132   -3.53553
>> plot(x,y,'b-',x1,y1,'r-',x2,y2,'g-')
>> axis([-4 4 -4 4], 'equal')
>> grid on
>> legend('original', 'rotated 90 deg', 'rotated 225 deg')
>>

```

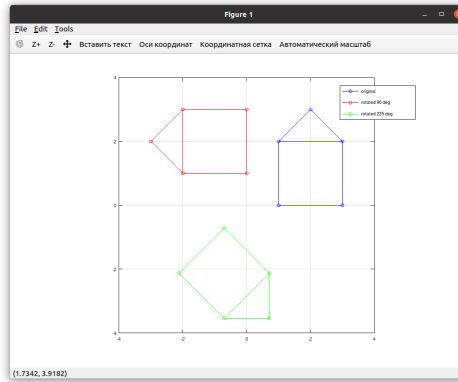


Рис. 4.7: Построение повёрнутого графика дома

Отразим граф дома относительно прямой $y = x$. Зададим матрицу отражения, подставив угол 45 градусов, так как именно под таким углом относительно оси абсцисс проходит прямая $y = x$. (рис. [4.8])

```

>> axis([-4 4 -4 4], 'equal') ;
>> grid on ;
>> legend('original', 'rotated 90 deg', 'rotated 225 deg')
>> 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')
error: 'legend' undefined near line 1 column 1
>> legend('original', 'reflected')
>>

```

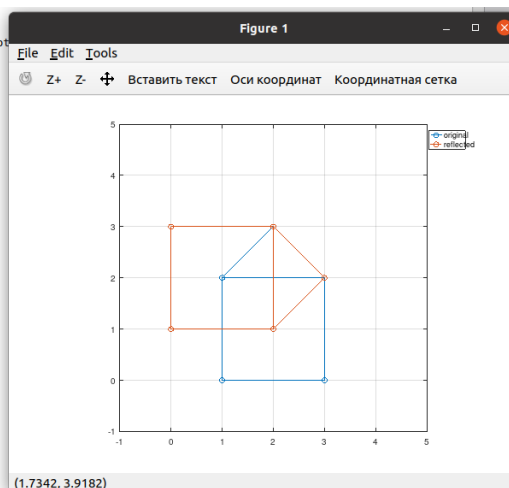


Рис. 4.8: График домика, отраженный относительно прямой $y = x$

Увеличим граф дома в 2 раза, используя матрицу для делитации (рис. [4.9])

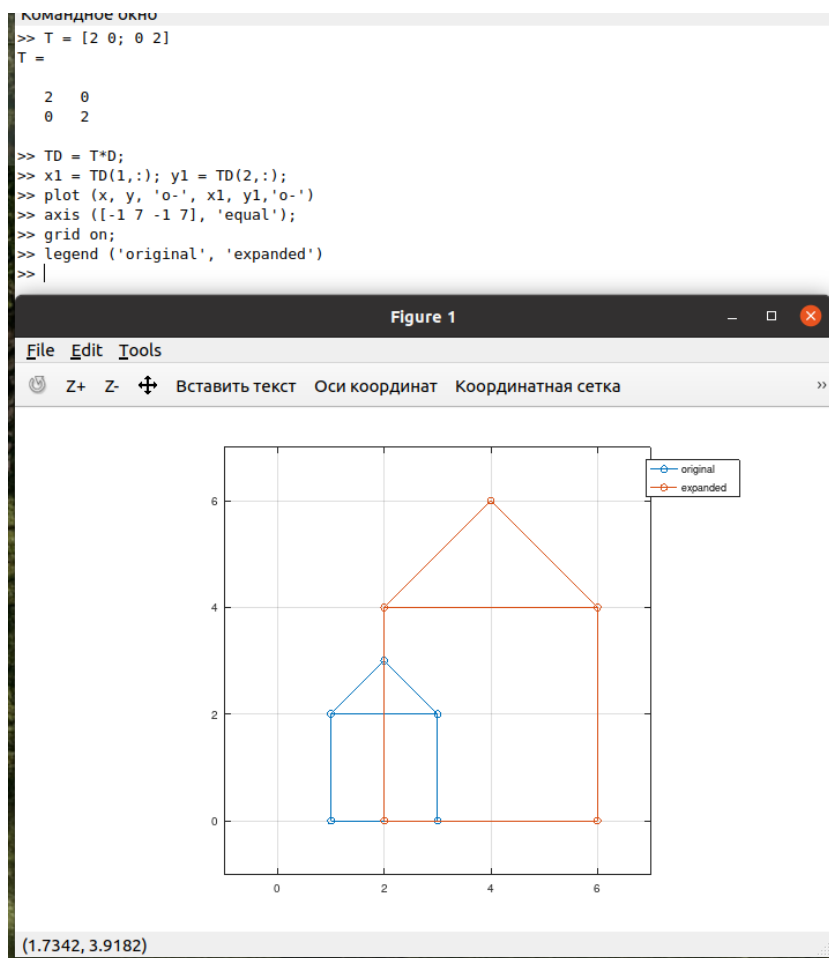


Рис. 4.9: Увеличинный в 2 раза график домика

5 Выводы

В результате выполнения работы научились подгонять полиномиальные кривые и выполнять различные матричные преобразования с помощью системы для математических вычислений Octave.

Список литературы

1. Подгонка кривой [Электронный ресурс]. Wikimedia Foundation, Inc., 2023.
URL: https://wikipedia.net/ru/Model_fitting#cite_note-3.
2. Умно́в А.Е. АНАЛИТИЧЕСКАЯ ГЕОМЕТРИЯ И ЛИНЕЙНАЯ АЛГЕБРА. МФТИ, 2011. 544 с.