

Лабораторная работа №5

Научное программирование

Колчева Юлия Вячеславовна

Содержание

| | | |
|---|--------------------------------|----|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 4 | Выводы | 18 |
| 5 | Список литературы | 19 |

List of Tables

List of Figures

| | | |
|------|---|----|
| 3.1 | Вывод данных | 7 |
| 3.2 | График | 8 |
| 3.3 | Матрица A | 9 |
| 3.4 | Вычисление формул | 10 |
| 3.5 | Решение системы | 10 |
| 3.6 | Код графиков | 11 |
| 3.7 | График параболы | 11 |
| 3.8 | График с помощью встроенной функции | 12 |
| 3.9 | Домик | 12 |
| 3.10 | Код для переворота | 13 |
| 3.11 | Переворот | 14 |
| 3.12 | Код для отражения | 15 |
| 3.13 | Отражение | 16 |
| 3.14 | Код для расширения | 16 |
| 3.15 | Расширение | 17 |

1 Цель работы

Изучение языка Octave, знакомство со способами работы с графиками

2 Задание

Разобраться со спецификой языка и выполнить операции.

1. Подгонка полиномиальной кривой
2. Матричные преобразования: а. Вращение б. Отражение относительно кривой в. Дилатация

3 Выполнение лабораторной работы

Для начала работы с программой включим журналирование сессии командой `diary on`. Затем приступим к выполнению первого этапа - подгонке полиномиальной кривой. Для начала задан матрицу `D` и разложим её на два вектора (рис. 3.1)

```
>> 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
```

Figure 3.1: Вывод данных

Построим получившиеся значение на графике (рис. 3.2)

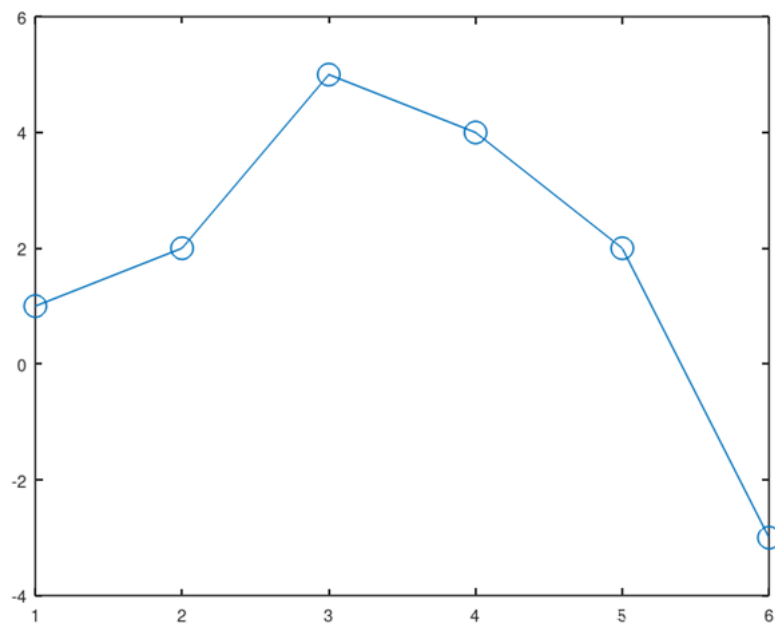


Figure 3.2: График

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

Для начала покажем как создать матрицу A (рис. 3.3)


```
Командное окно
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
```

Figure 3.3: Матрица A

И значения, которые мы будем использовать (рис. 3.4)

```

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

>> A'*ydata
ans =
    60
    28
    11

>> B = A'*A
B =
    2275    441    91
    441    91    21
    91    21    6

>> B(:,4) = A'*ydata
B =
    2275    441    91    60
    441    91    21    28
    91    21    6    11

```

Figure 3.4: Вычисление формул

Решим задачу методом Гаусса (рис. 3.5)

```

>> 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
>> x = linspace(0,7,50);
error: 'linspace' undefined near line 1, column 5
>> x = linspace(0,7,50);
>> y = a1 * x.^2 + a2 * x + a3
y =

Columns 1 through 5:
   -4.400000   -3.611079   -2.858601   -2.142566   -1.462974

```

Figure 3.5: Решение системы

Построим соответствующий график параболы. Процесс подгонки может быть автоматизирован встроенными функциями Octave. Для этого мы можем исполь-

зовать встроенную функцию для подгонки полинома `polyfit`. На скриншоте показан сначала код для параболы, а затем для встроенной функции (рис. 3.6)

```
>> 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')
>> P = polyfit(xdata, ydata, 2)
P =
-0.8929  5.6500 -4.4000

>> y = polyval(P, xdata)
y =
0.3571
3.3286
4.5143
3.9143
1.5286
-2.6429

>> plot(xdata, ydata, 'o-', xdata, y, '+-')
>> grid on ;
>> legend('original data', 'polyfit data')
>> D = [ 1 1 3 3 2 1 3 ; 2 0 0 2 3 2 2]
D =
```

Figure 3.6: Код графиков

Из предыдущего кода получились такие графики (рис. 3.7) (рис. 3.8)

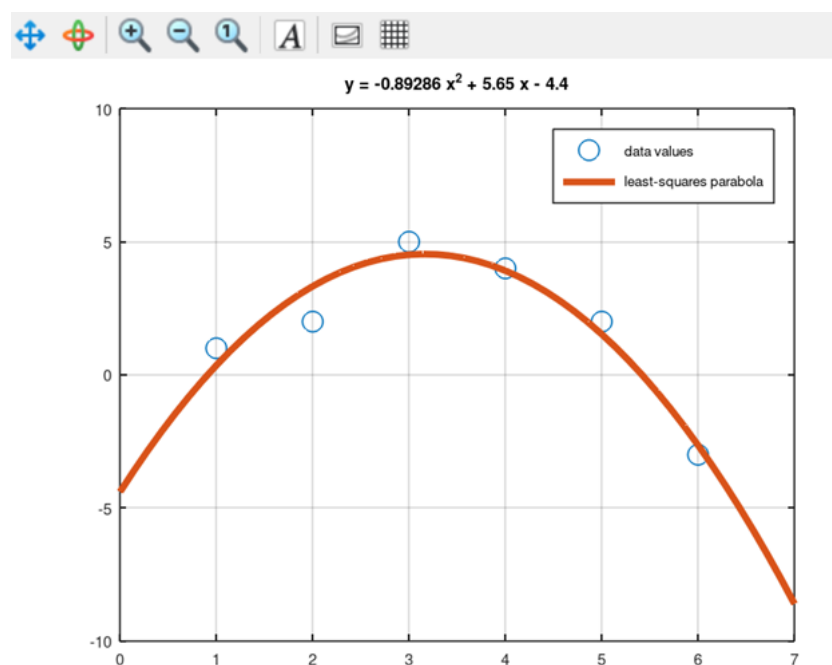


Figure 3.7: График параболы

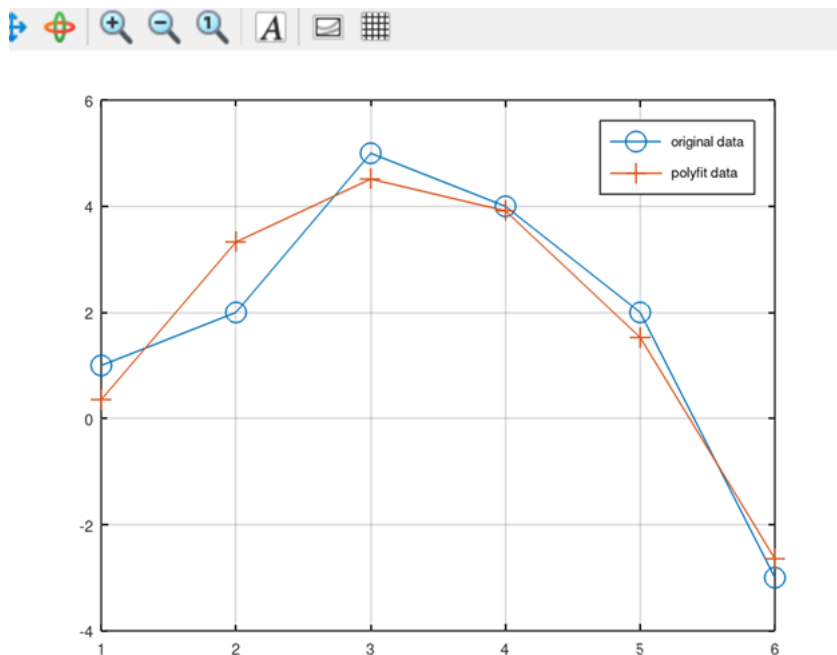


Figure 3.8: График с помощью встроенной функции

Приступим к разделу матричных преобразований. Для начала нарисуем простой домик, с которым будем работать (рис. 3.9)

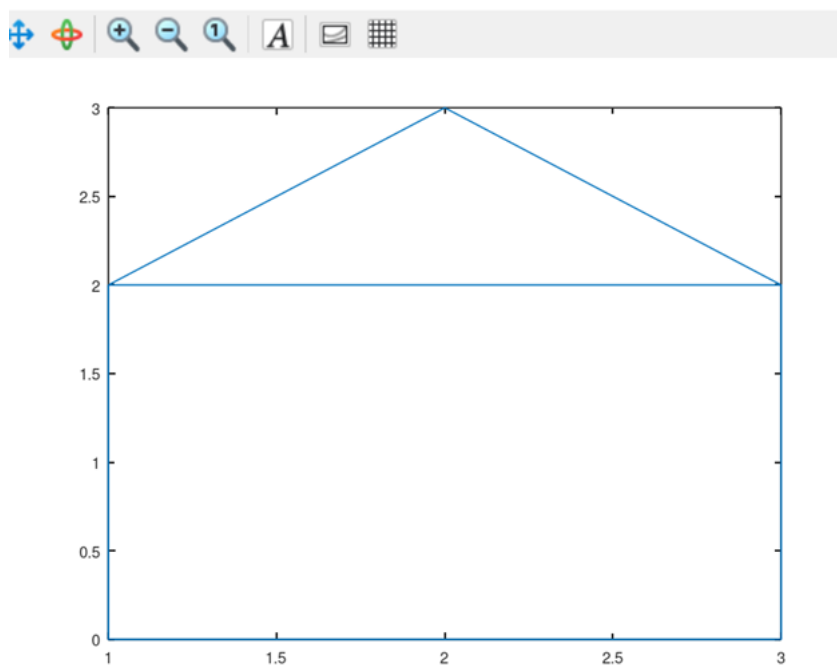


Figure 3.9: Домик

Повернём его на 90 и на 225 градусов. Вращения могут быть получены с использованием умножения на специальную матрицу. Вначале переведём угол в радианы. (рис. 3.10)

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

Figure 3.10: Код для переворота

Для второго отражения выполняем те же самые действия. В итоге получаем такой рисунок (рис. 3.11)

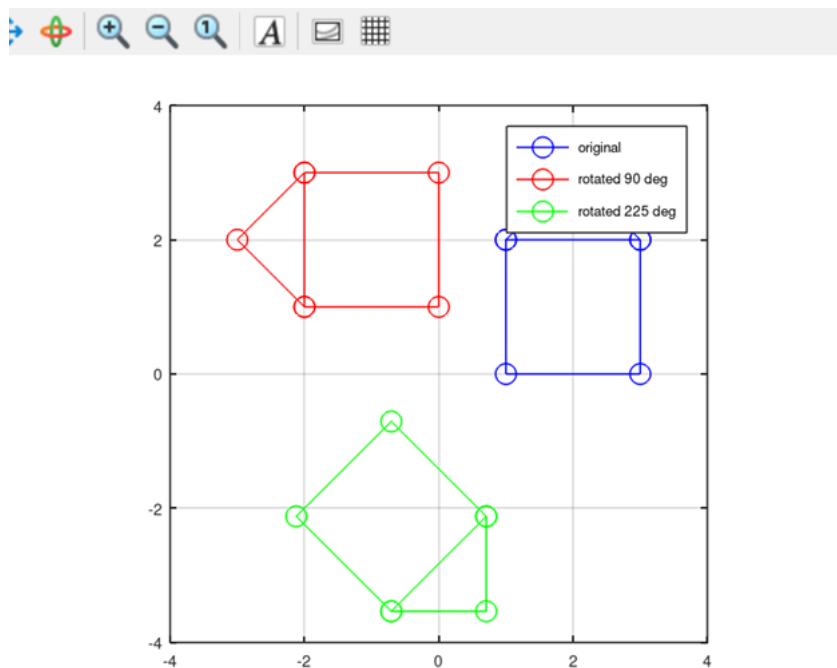


Figure 3.11: Переворот

Теперь приступим к отражению. Отразим граф дома относительно прямой $y = x$. Зададим матрицу отражения и нарисуем график. (рис. 3.12)

```

>> 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' )

```

Figure 3.12: Код для отражения

Получаем график (рис. 3.13)

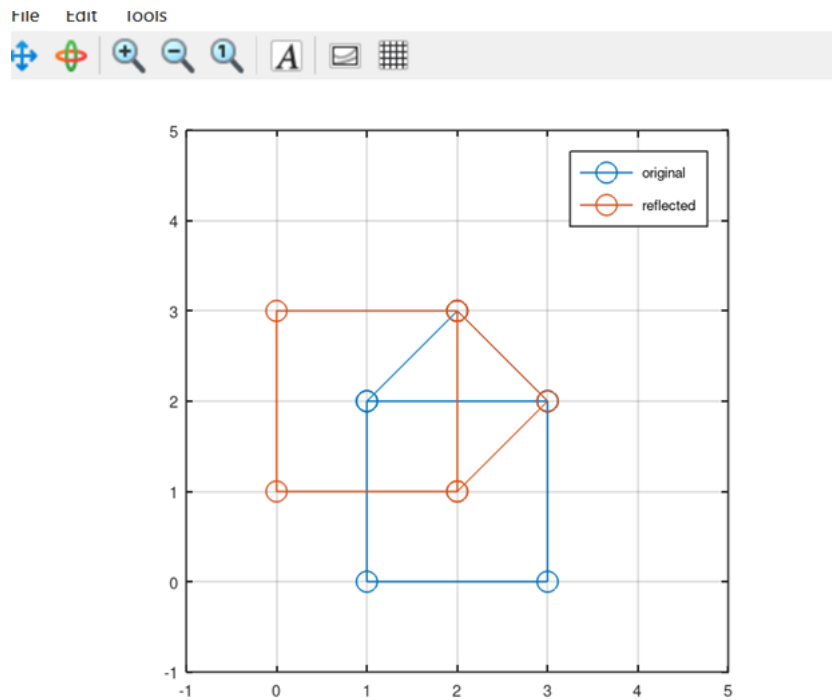


Figure 3.13: Отражение

И последнее - расширение. Увеличим граф в два раза. Заданым матрицу T и умножим на неё. (рис. 3.14)

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

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

>> TD = T*D
TD =
    2    2    6    6    4    2    6
    4    0    0    4    6    4    4

>> x1 = TD(1,:); y1 = TD(2,:);
>> plot(x,y,'o-', x1, y1,'o-')
>> axis([-1 7 -1 7], 'equal');
>> grid on;
>> legend('original', 'expanded')
```

Figure 3.14: Код для расширения

Получаем график (рис. 3.15)

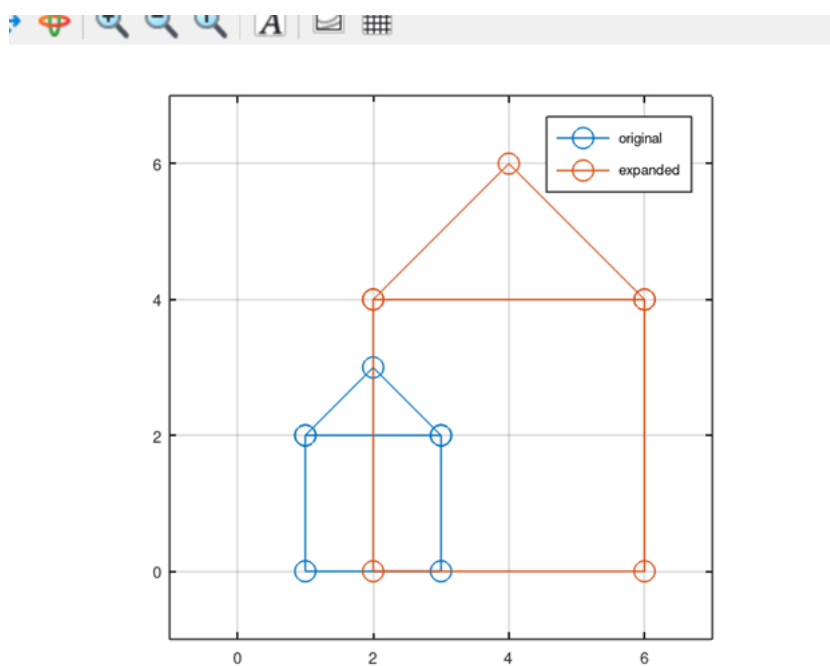


Figure 3.15: Расширение

На этом лабораторная работа закончена.

4 Выводы

Познакомилась со способами работы с графиками.

5 Список литературы

Лабораторная работа №5

Лабораторная работа № 5. Введение в работу с Octave [Электронный ресурс].

2019. URL:https://esystem.rudn.ru/pluginfile.php/2372906/mod_resource/content/2/README.pdf