### РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

### Факультет физико-математических и естественных наук

### Кафедра прикладной информатики и теории вероятностей

### *дисциплина*: Научное программирование

# Лабораторная работа №5

### Выполнил: Маслов Александр

### Группа: НФИмд-02-20

### С/б: 1032202156

#### Москва

#### 2020

### Цель работы:

Рассмотреть с помощью Octave подгонку полиномиальной кривой, решить проблему подгонки полинома к множеству точек и рассмотреть такие матричные преобразования, как:

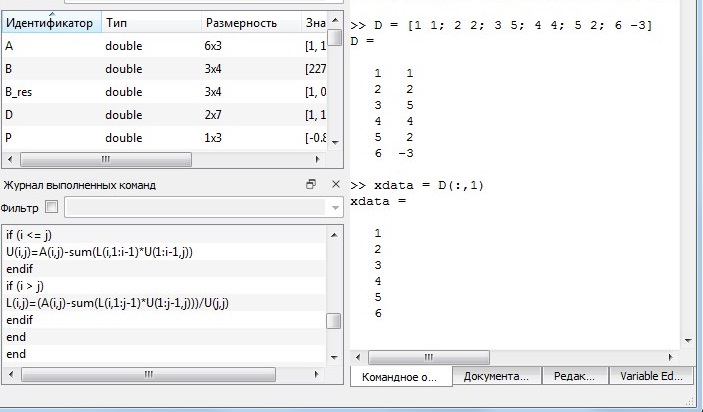
* вращение
* отражение
* дилатация

### Ход работы:

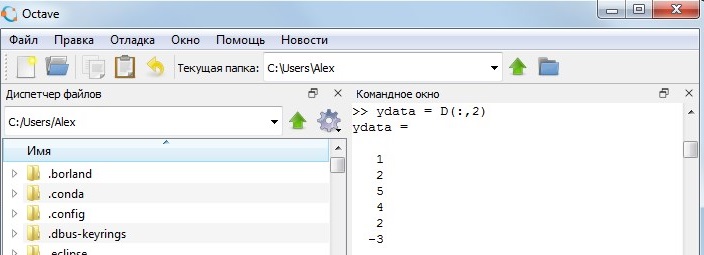
#### Подгонка полиномиальной кривой

В статистике часто рассматривается проблема подгонки прямой линии к набору данных. Решим общую проблему подгонки полинома к множеству точек. Пусть нам нужно найти параболу по методу наименьших квадратов для набора точек, заданных матрицей:

В матрице заданы значения в столбце 1 и значения в столбце 2. Введём матрицу данных и извлечём вектора и :



[Рисунок 1. Матрица D и вектор x](screen1.jpg)

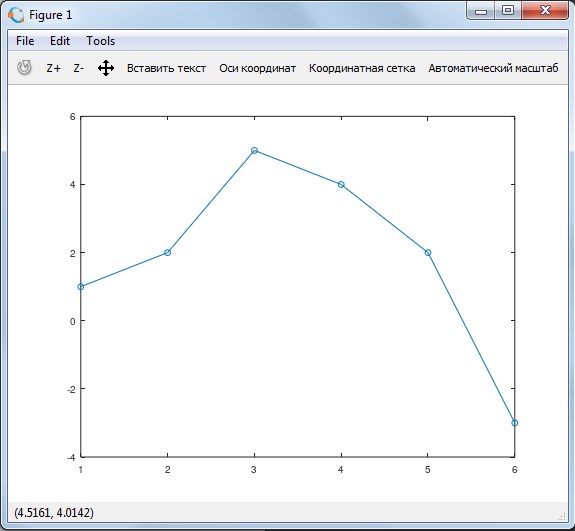


[Рисунок 2. Вектор y](screen2.jpg)

Нарисуем точки на графике:



[Рисунок 3. Команда построения графика](screen3.jpg)



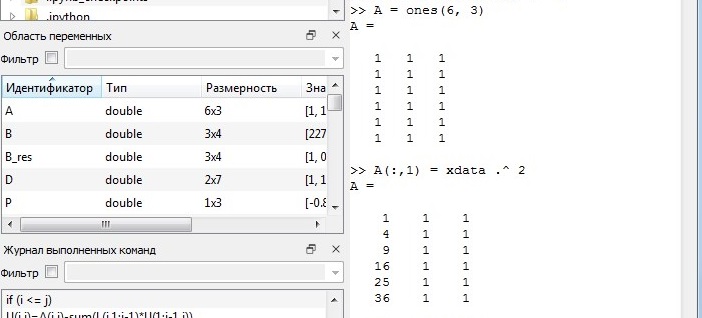
[Рисунок 4. График 1](graph1.jpg)

Построим уравнение вида . Подставляя данные, получаем следующую систему линейных уравнений.

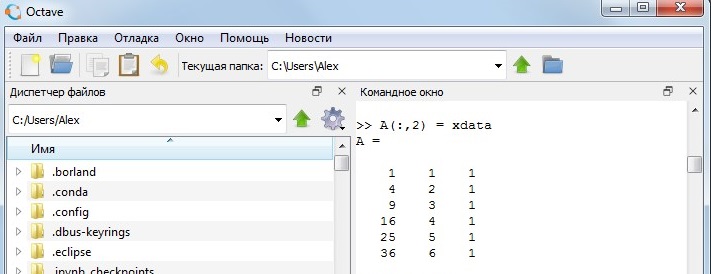
$

$

Обратите внимание на форму матрицы коэффициентов . Третий столбец – все единицы, второй столбец – значения , а первый столбец – квадрат значений . Правый вектор – это значения . Есть несколько способов построить матрицу коэффициентов в Octave. Один из подходов состоит в том, чтобы использовать команду ones для создания матрицы единиц соответствующего размера, а затем перезаписать первый и второй столбцы необходимыми данными.

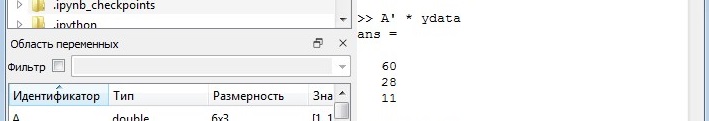


[Рисунок 5. Формируем матрицу A](screen4.jpg)



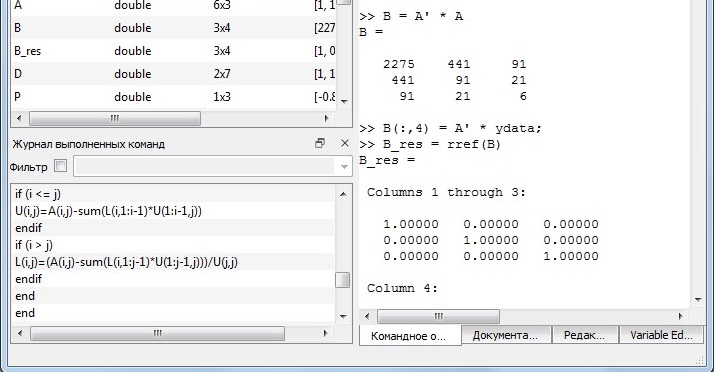
[Рисунок 6. Полученная матрица A](screen5.jpg)

Решение по методу наименьших квадратов получается из решения уравнения $ A^T Ab = A^T y$, где - вектор коэффициентов полинома.

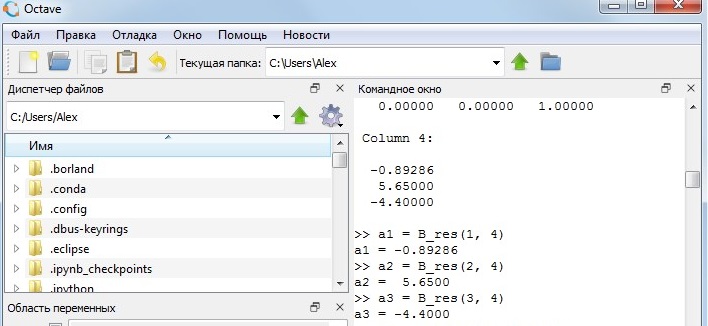


[Рисунок 7. Транспонированная матрица А, умноженная на вектор y](screen6.jpg)

Решим задачу методом Гаусса. Запишем расширенную матрицу:



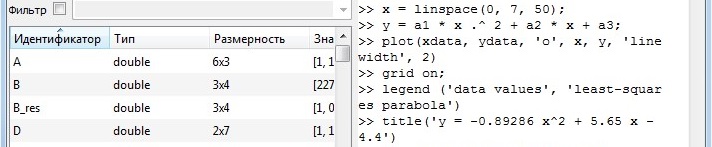
[Рисунок 8. Полученная матрица B](screen7.jpg)



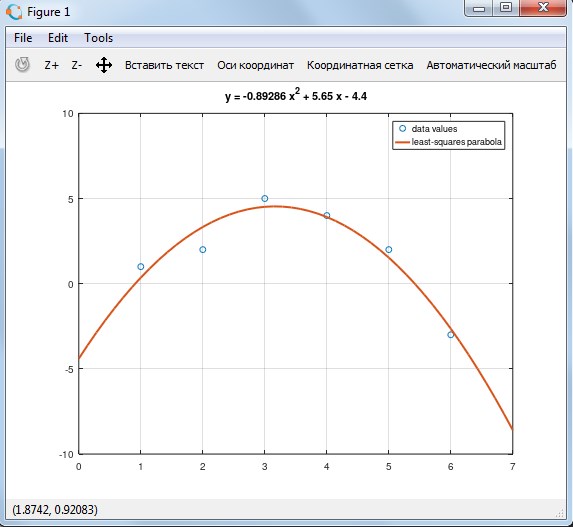
[Рисунок 9. Коэффициенты уравнения](screen8.jpg)

Таким образом, искомое квадратное уравнение имеет вид

Построим соответствующий график параболы.

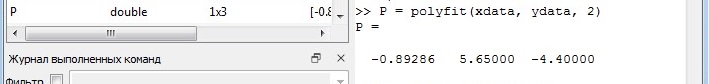


[Рисунок 10. Построение графика параболы](screen9.jpg)



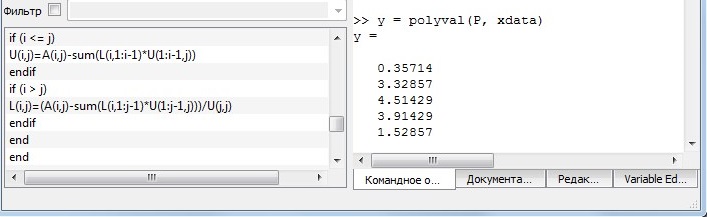
[Рисунок 11. График параболы](graph2.jpg)

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. Синтаксиса: polyval (P, x). Получим подгоночный полином.



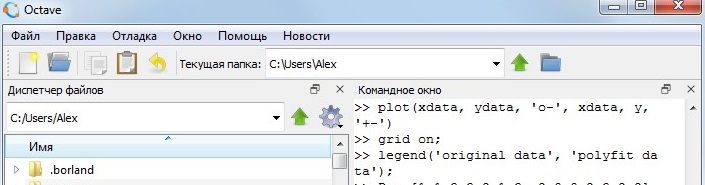
[Рисунок 12. Подгоночный полином](screen10.jpg)

Рассчитаем значения полинома в точках.

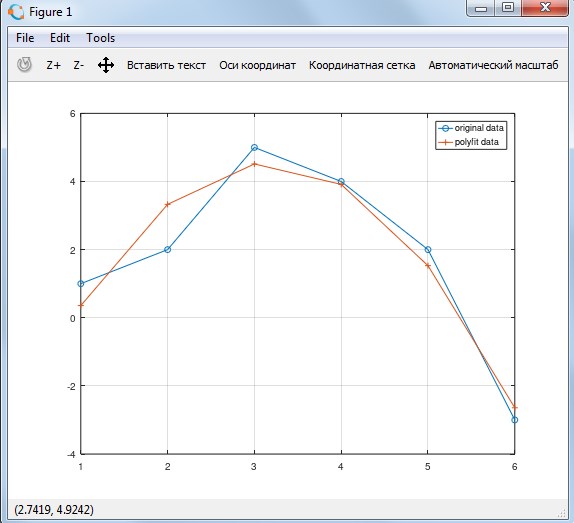


[Рисунок 13. Значения полинома в точках](screen11.jpg)

Построим исходные и подгоночные данные.



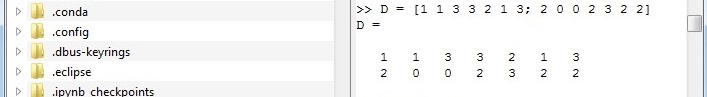
[Рисунок 14. Исходные и подгоночные данные](screen12.jpg)



[Рисунок 15. График исходных и подгоночных данных](graph3.jpg)

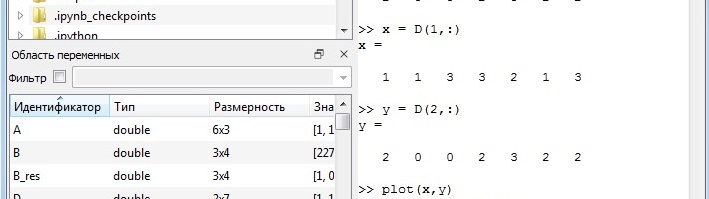
#### Матричные преобразования

Матрицы и матричные преобразования играют ключевую роль в компьютерной графике. Существует несколько способов представления изображения в виде матрицы. Подход, который мы здесь используем, состоит в том, чтобы перечислить ряд вершин, которые соединены последовательно, чтобы получить ребра простого графа. Мы записываем это как матрицу $ 2 n$, где каждый столбец представляет точку на рисунке. В качестве простого примера, давайте 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).

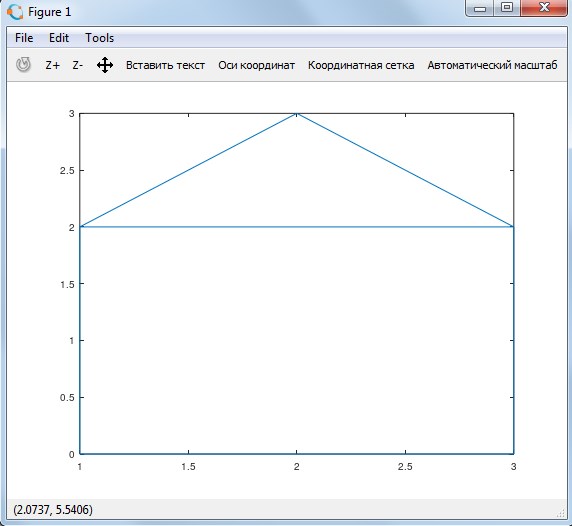


[Рисунок 16. Матрица D](screen13.jpg)

Нарисуем этот граф



[Рисунок 17. Построение графа](screen14.jpg)



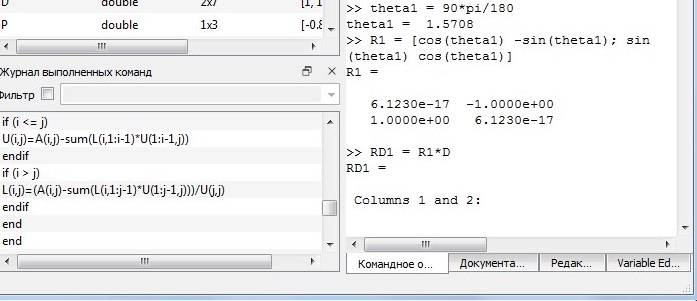
[Рисунок 18. Граф](graph4.jpg)

**Вращение**

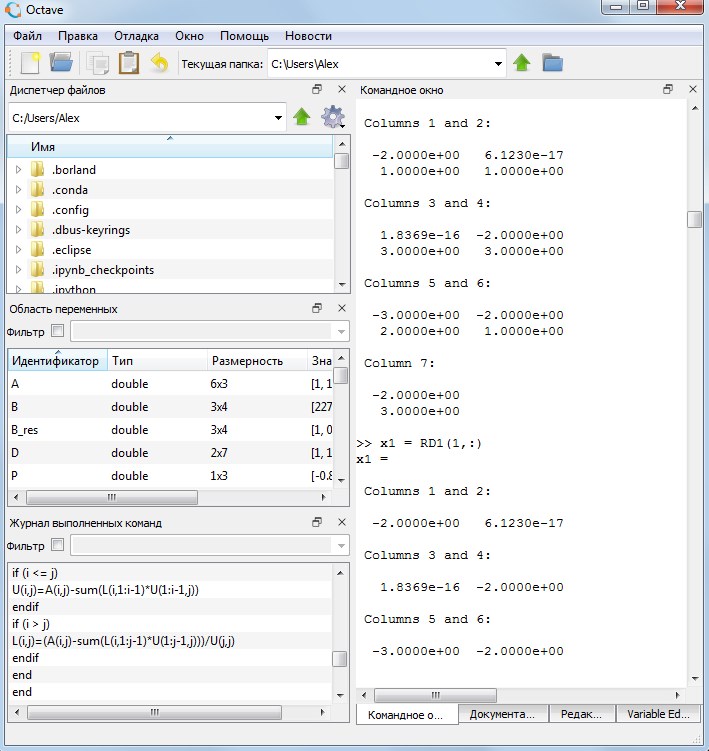
Рассмотрим различные способы преобразования изображения. Вращения могут быть получены с использованием умножения на специальную матрицу. Вращение точки относительно начала координат определяется как

где

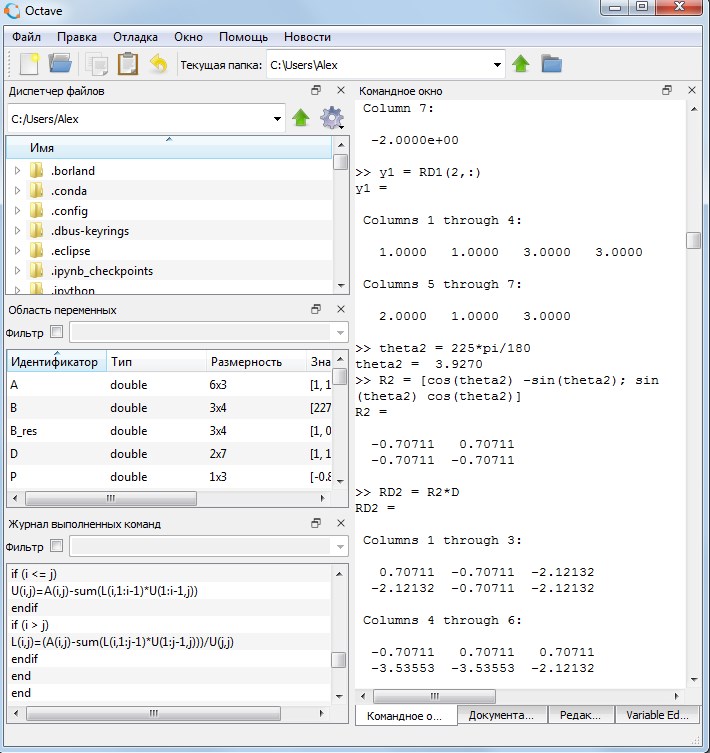
- угол поворота (измеренный против часовой стрелки). Теперь, чтобы произвести повороты матрицы данных *D*, нам нужно вычислить произведение матриц *RD*. Повернём граф дома на и . Вначале переведём угол в радианы.



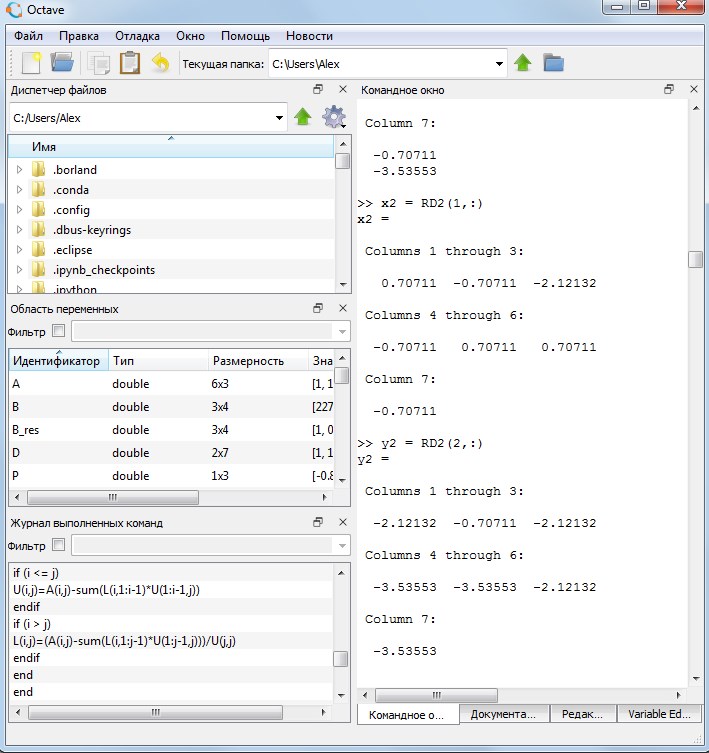
[Рисунок 19. Перевод градусов в радианы и начало алгоритма поворота графа на](screen15.jpg)



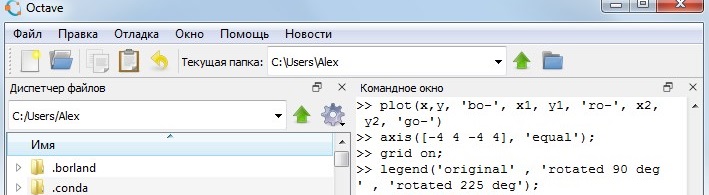
[Рисунок 20. Алгоритм поворота графа на](screen16.jpg)



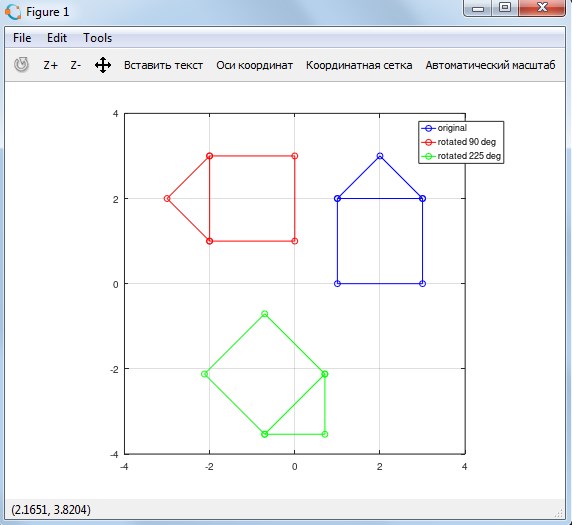
[Рисунок 21. Перевод градусов в радианы и начало алгоритма поворота графа на](screen17.jpg)



[Рисунок 22. Алгоритм поворота графа на](screen18.jpg)



[Рисунок 23. Построение графа](screen19.jpg)



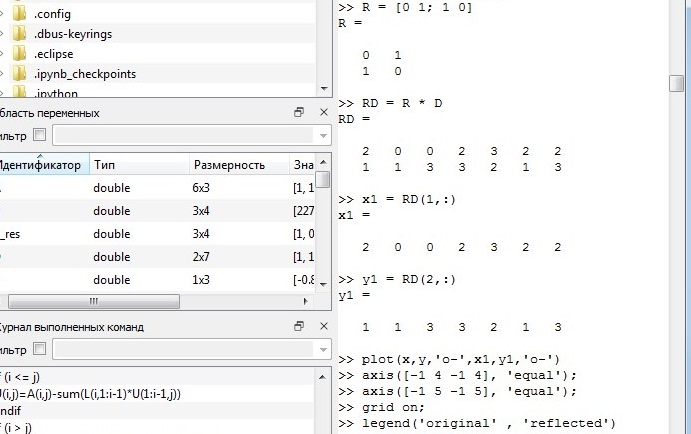
[Рисунок 24. Граф](graph5.jpg)

**Отражение**

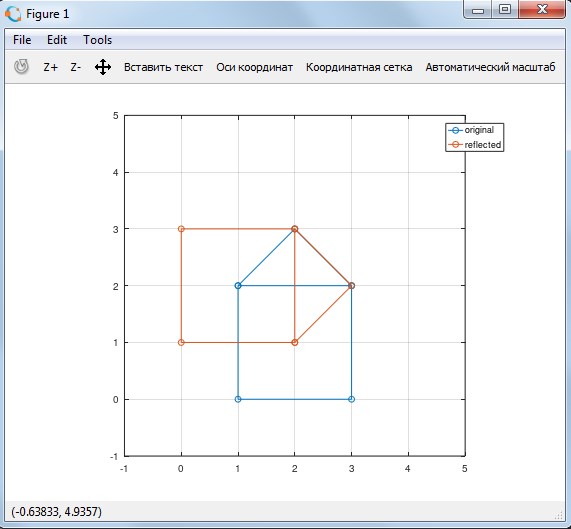
Если – прямая, проходящая через начало координат, то отражение точки относительно прямой определяется как

где

- угол между прямой и осью абсцисс (измеренный против часовой стрелки). Отразим граф дома относительно прямой . Зададим матрицу отражения.



[Рисунок 25. Алгоритм отражения и построение графа](screen20.jpg)

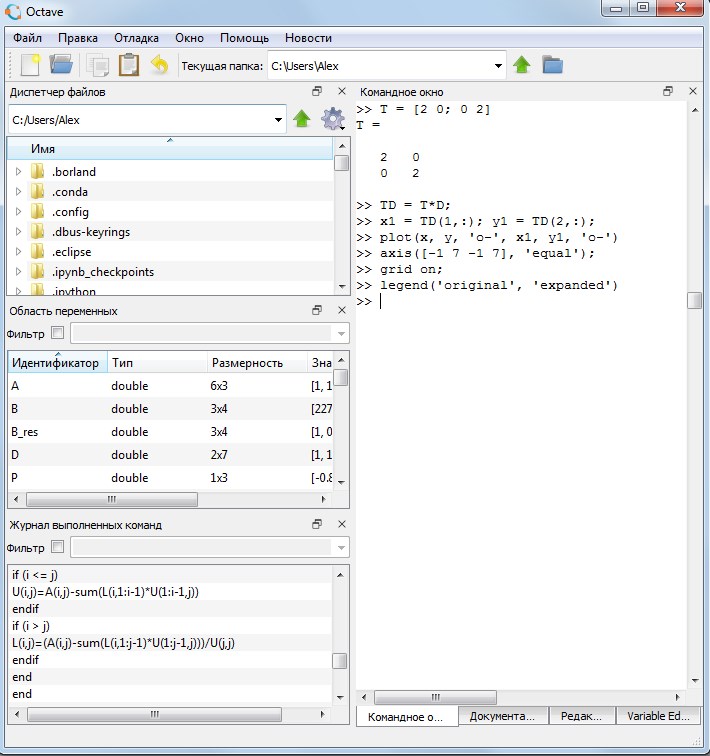


[Рисунок 26. Граф](graph6.jpg)

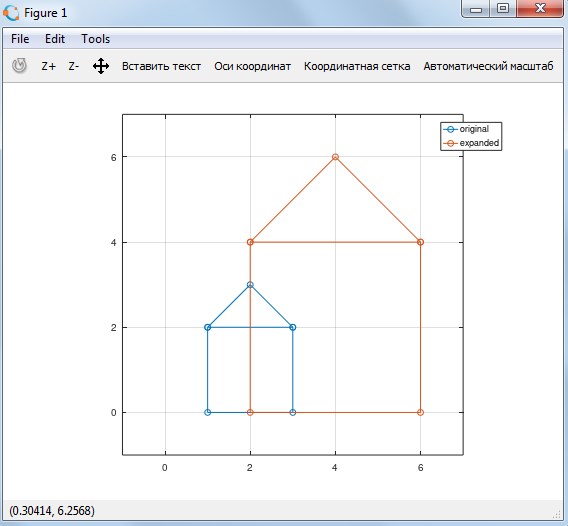
**Дилатация**

Дилатация (то есть расширение или сжатие) также может быть выполнено путём умножения матриц. Пусть

Тогда матричное произведение *TD* будет преобразованием дилатации *D* с коэффициентом . Увеличим граф дома в 2 раза.



[Рисунок 27. Алгоритм дилатации и построение графа](screen21.jpg)



[Рисунок 28. Граф](graph7.jpg)

**Вывод:** в процессе выполнения лабораторной работы с помощью Octave мы решили общую проблему подгонки полинома к множеству точек, а также рассмотрели такие матричные преобразования, как:

* вращение
* отражение
* дилатация