Научное програмирование

Отчет по лабораторной работе № 5

Кейела Патачона НПМмд-02-21

Содержание

[Цель работы 1](#_Toc90043599)

[Подгонка полиномиальной кривой 1](#_Toc90043600)

[Матричные преобразования 8](#_Toc90043601)

[Вращение 9](#_Toc90043602)

[Отражение 11](#_Toc90043603)

[Дилатация 13](#_Toc90043604)

[Вывод 15](#_Toc90043605)

# Цель работы

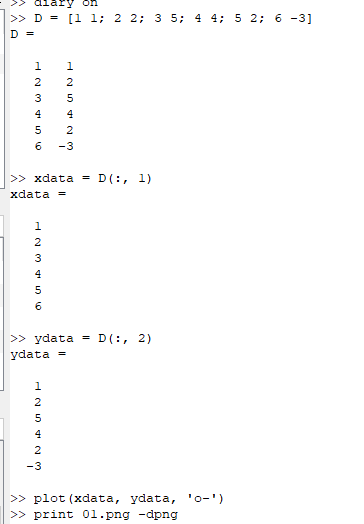
Решение проблемы подгонки полинома к множеству точек и изучение матричные преобразования.

# Подгонка полиномиальной кривой

В статистике часто рассматривается проблема подгонки прямой линии к набору данных. Решим более 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 в Octave и извлечём вектора и .



Ввод данных

Нарисуем точки на графике.

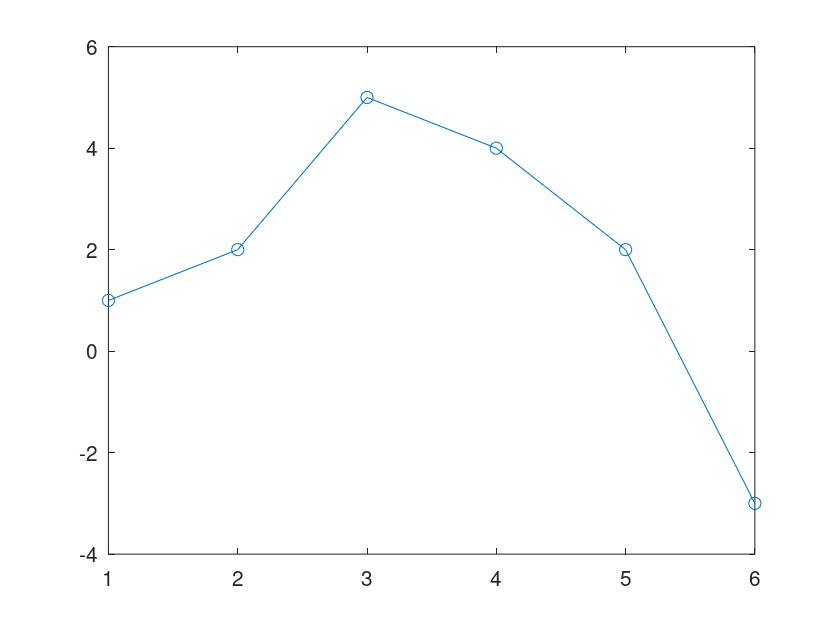
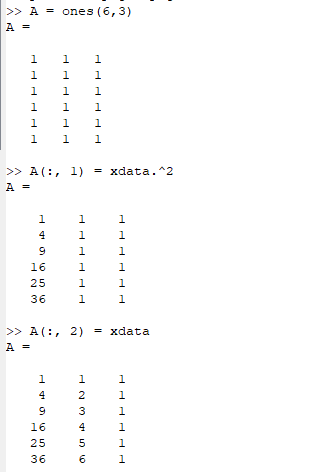


график исходных данных

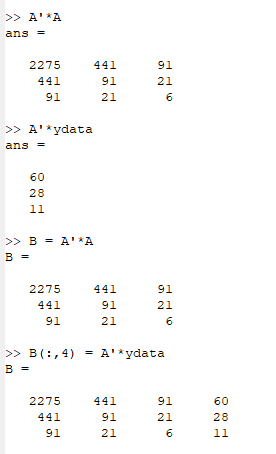
Построим уравнение вида Подставляя данные, получаем следующую систему линейных уравнений

Обращаем 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 для создания матрицы единиц соответствующего размера, а затем перезаписать первый и второй столбцы необходимыми данными.



Заполнение матрицы коэффициентов

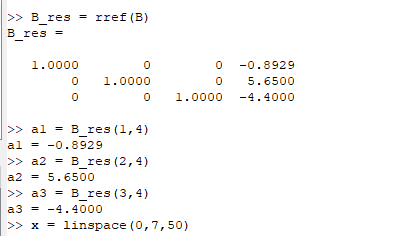
Решение по методу наименьших квадратов получается из решения уравнения , где – вектор коэффициентов полинома. Используем Octave для построения уравнений. Запишем расширенную матрицу для решения задачи методом Гаусса.



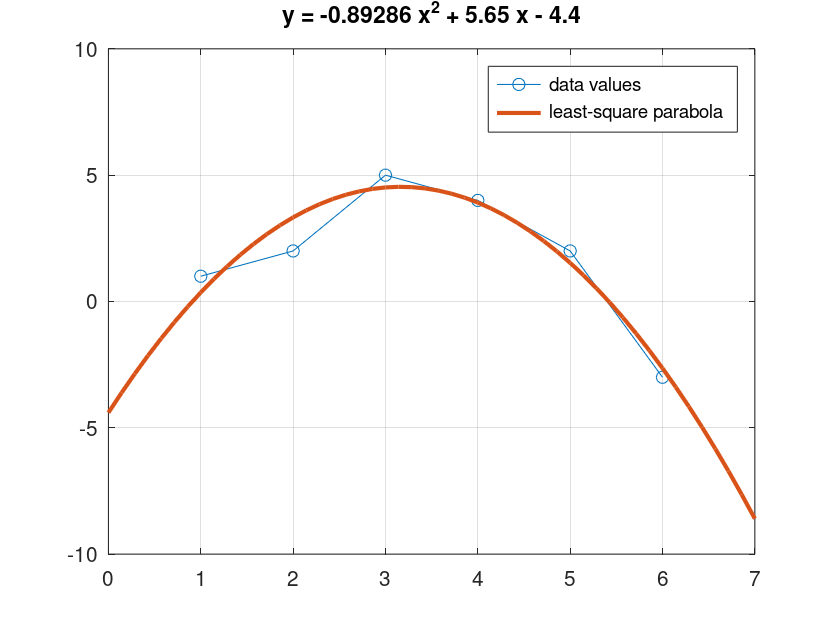
Уравнения

Таким образом, искомое квадратное уравнение имеет вид

. Построим соответствующий график параболы.

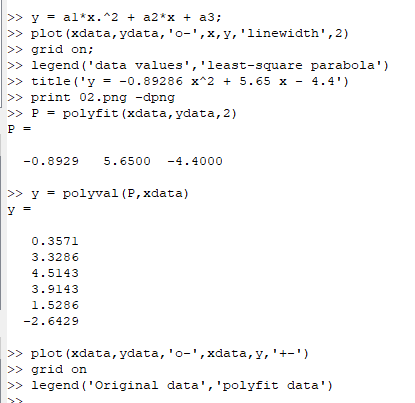


Решение

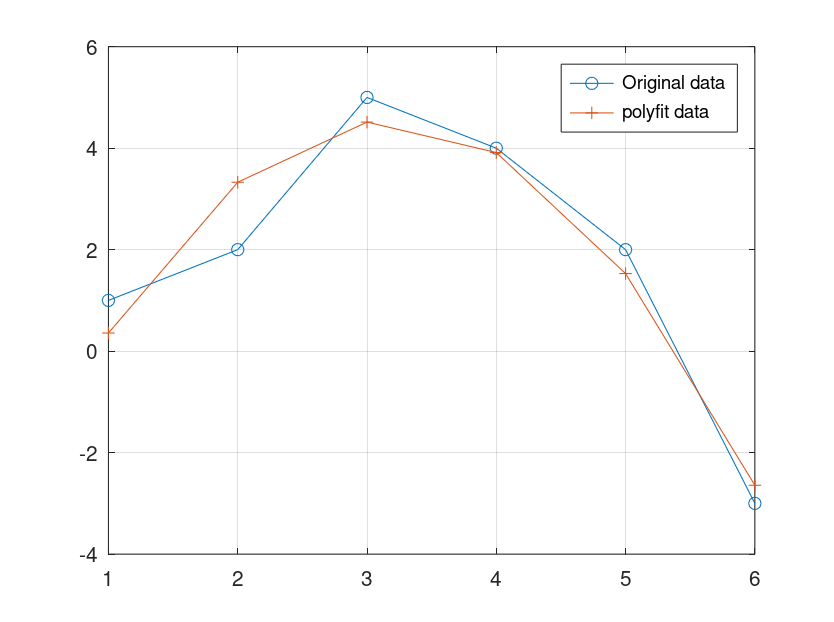


Результат подгонки

Процесс подгонки может быть автоматизирован встроенными функциями Octave. Для этого мы можем использовать встроенную функцию для подгонки полинома . Синтаксис: , где – это степень полинома. Значения полинома в точках, задаваемых вектором-строкой можно получить с помощью функции . Синтаксиса: . Получим подгоночный полином.



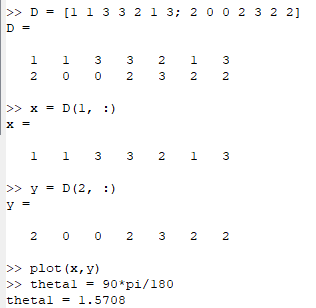
Процесс построения подгонки



Результат подгонки

# 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 , 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).

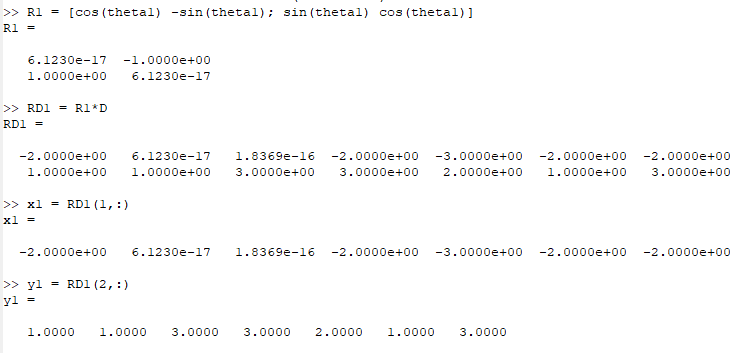


Кодировка домика

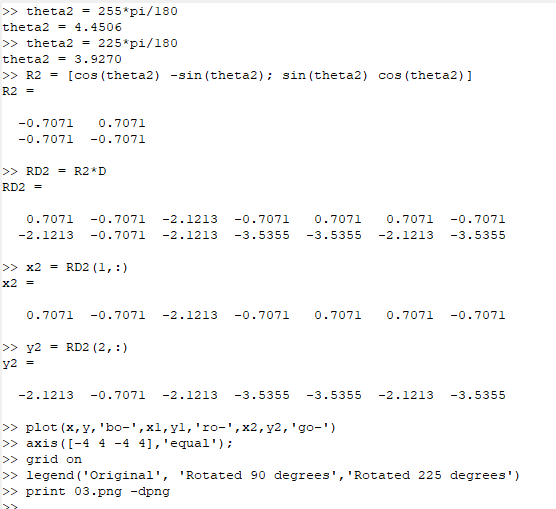
## Вращение

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

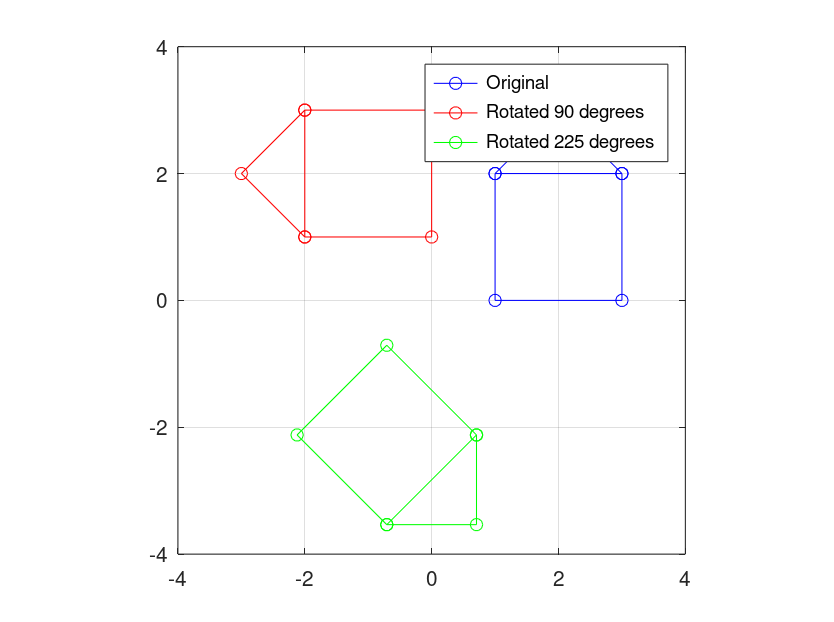
где



Вращение домика 1



Вращение домика 2



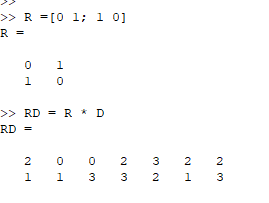
Результат вращения

## Отражение

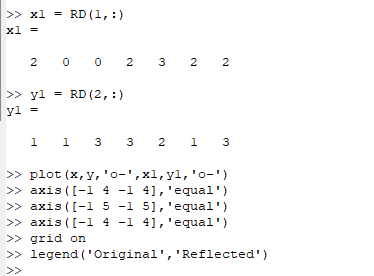
Если – прямая, проходящая через начало координат, то отражение точки относительно прямой определяется как

где

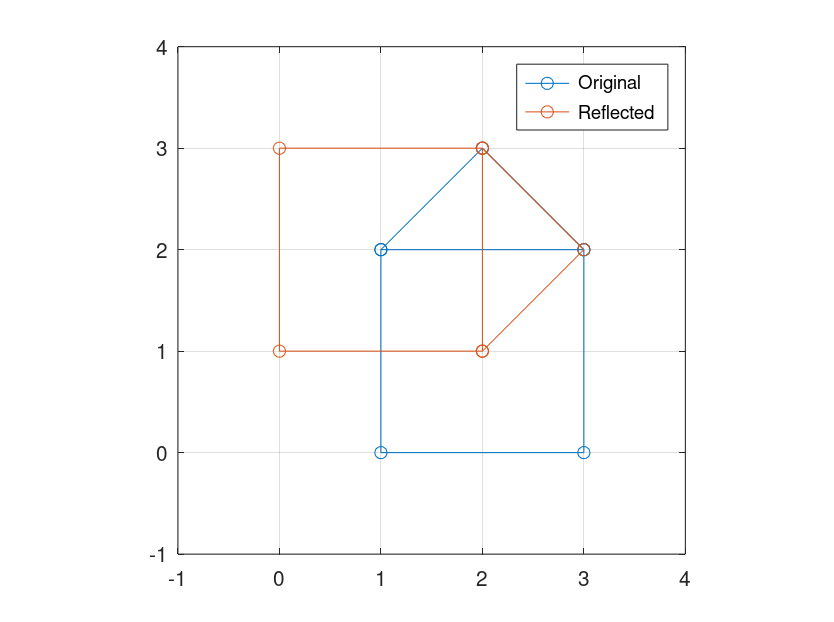
– угол между прямой и осью абсцисс (измеренный против часовой стрелки). Отразим граф дома относительно прямой . Зададим матрицу отражения (поясните, почему она такая).



Отражение домика 1



Отражение домика 2

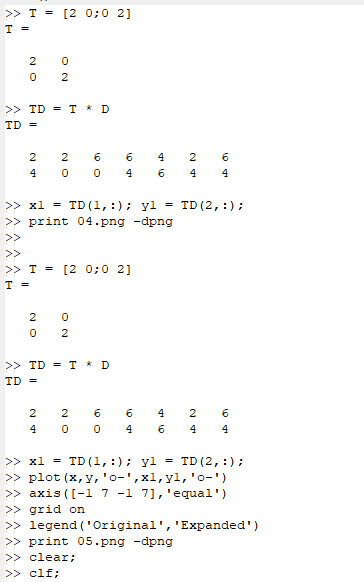


Результат отражения

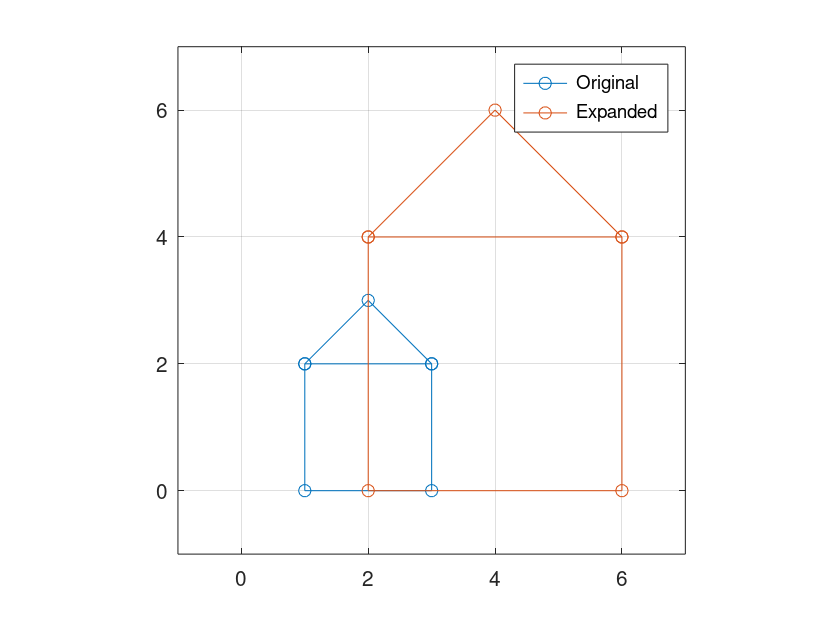
## Дилатация

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

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



Дилатация домика



Результат дилатации

# Вывод

В ходе выполнения данной работы я научился решить задачу подгонки и работать с матричными преобразованиями.