

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ     
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

**ДАЛЬНЕВОСТОЧНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**

**Институт математики и компьютерных технологий**

**Департамент математического и компьютерного моделирования**

**ОТЧЕТ**

**по лабораторной работе** **№ 4**

«QR разложение Грэма-Шмидта»

Выполнил: студент гр. Б9120-02.03.01сцт

Пограничный К. О.

Проверил: преподаватель

Кузнецов К. С.

Владивосток 2023

**Постановка задания**

Дана система линейных алгебраических уравнений в виде Ax=b.

Требуется решить ее при помощи QR разложения:

Изображение выглядит как диаграмма, линия, дизайн, типография

Автоматически созданное описание

**Метод решения**

Задача была выполнена на языке программирования Python с использованием средств библиотеки NumPy. Из условия были созданы матрица A и вектор b.

Был реализован алгоритм QR – разложения, после чего произведены вычисления для уравнений , которые в свою очередь позволили нам получить ответ на поставленную задачу.

**Алгоритм решения**

1. Матрицу A представим в следующем виде:

где первый столбец матрицы, а – последний столбец матрицы A.

1. Вводится промежуточная матрица U, элементы которой определяются при помощи формулы:
2. Далее находим матрицу C, элементы которой определяются при помощи формулы:
3. Матрицы Q и R должны быть представлены в следующем виде:
4. Найдем матрицы U, Q, R
5. Решаем СЛАУ в два этапа:

**Подробный анализ кода.**

1. Создание исходной матрицы A и вектора b из условия с помощью

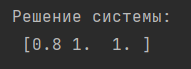
numpy.array()

1. Создание функции QR – разложения “qr\_decomposition”
2. С помощью “dim” указал размерность массива.
3. Создал нулевые матрицы R, U, C.
4. Заполнил их в соответствии с формулами выше.
5. Q = C.T – матрица Q является транспонированной копией матрицы C.
6. Функция “proj” описывает оператор проекции вектора x на вектор y, что и является отличительной особенностью реализации метода QR – разложения Грэма-Шмидта.
7. Функция “get\_solution” решает СЛАУ

**Использованные функции**

1. np.dot() – вычисляет скалярное произведение массивов.
2. np.zeros() – создает нулевой массив.
3. .transpose() – функция транспонирования.
4. np.linalg.norm() – возвращает норму вектора.
5. np.linalg.tensorsolve() - используется для разрешения тензорного уравнения Ax=b относительно x.

**Результаты тестов**



**Вывод**

Разложив матрицу на QR, можно просто решить систему уравнений с данной матрицей, вычислить ее определитель, найти обратную матрицу.

**Приложение**

Листинг программы:

import numpy as np  
  
  
def proj(x, y):  
 return y \* (x.dot(y) / y.dot(y))  
  
  
def qr\_decomposition(A):  
 dim = A.shape  
 R = np.zeros((dim[1], dim[1]))  
 U = np.zeros(dim)  
 C = np.zeros(dim)  
  
 for k in range(dim[0]):  
 sum = 0  
  
 if k > 0:  
 sum = 0  
 for j in range(k-1):  
 sum += proj(A.transpose()[k], U[j])  
  
 U[k] = A.transpose()[k] - sum  
 C[k] = U[k] / np.linalg.norm(U[k])  
  
 for i in range(dim[1]):  
 for j in range(dim[1]):  
 if i <= j:  
 R[i][j] = C[i].dot(A.transpose()[j])  
  
 Q = C.T  
  
 return Q, R  
  
  
def get\_solution(A, b):  
 Q, R = qr\_decomposition(A)  
 y = np.linalg.tensorsolve(Q, b)  
  
 return np.linalg.tensorsolve(R, y)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 A = np.array([  
 [2, 0, 1],  
 [0, 1, -1],  
 [1, 1, 1]  
 ])  
 b = np.array([3, 0, 3])  
  
 Q, R = qr\_decomposition(A)  
 print("Решение системы:\n", get\_solution(A, b))  
 print("\nТочное решение:\n", np.linalg.tensorsolve(A, b))