Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Санкт-Петербургский политехнический университет Петра Великого»

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Лабораторная работа №2**

по дисциплине «Вычислительная математика»

**Вариант №21**

Выполнил

Студент группы 3530901/80004 Иванов К. А.

\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Цыган В. Н.

\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург 2020

**Постановка задачи:**

Составить процедуру формирования матрицы С по формулам:

Cik= d +log2(ik)+cos(ik), d – параметр.

Вычислить с помощью DECOMP и SOLVE матрицу C-1, найти матрицу R = E – C-1C и ее норму. Проанализировать решение при N = 5 и d =10;1000;10000.

**Описание решения:**

1. Создадим функцию формирования матрицы С, подавая на вход размерность матрицы n и нужное значение параметра d.
2. Используя функции библиотеки SciPy, вычислим все остальное.

Вычисления и построения функции и графиков будет выполнено при помощи двух популярных библиотек языка Python 3, SciPy и matplotlib.

Ознакомиться с тем, как работают внутренне та или иная операция можно на соответствующих сайта с документацией по библиотекам.

SciPy:[docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.ode.html#scipy-integrate-ode](https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.ode.html" \l "scipy-integrate-ode)

matplotlib:[matplotlib.org/api/\_as\_gen/matplotlib.pyplot.html?highlight=plot#module-matplotlib.pyplot](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.html?highlight=plot" \l "module-matplotlib.pyplot)

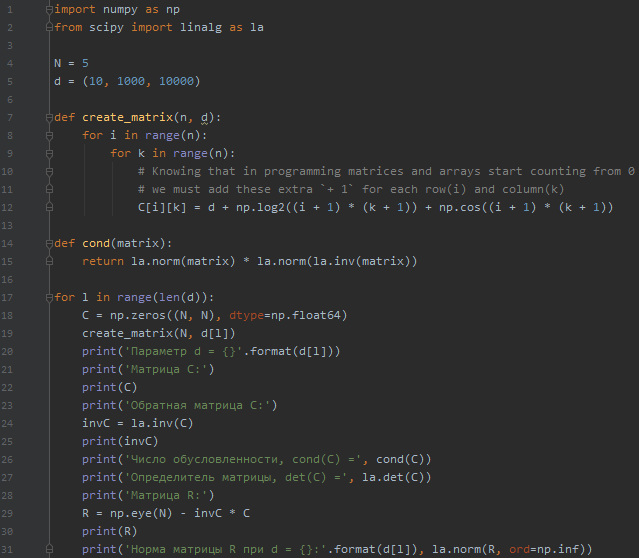
**Ход работы:**

В нашем цикле, сначала заполняем матрицу нулями, дальше на ее место начинает ставить на каждый место строчки-столбца (i, k) свой элемент, получаемый по формуле описанной в задании, дальше выводим ее на экран, считаем обратную матрицу С-1 помощью встроенной функции из библиотеки scipy.linalg, где используется методы **DECOMP и SOLVE .** Находим матрицу R и ее норму, с помощью специальной функции scipy.linalg.norm, а в ней формуле при ord = inf, которая точным образом повторяем ту, которая указана по заданию.

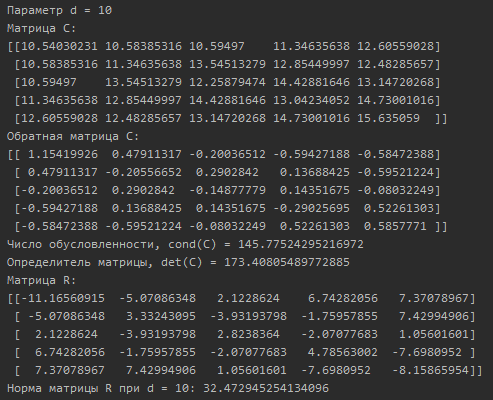


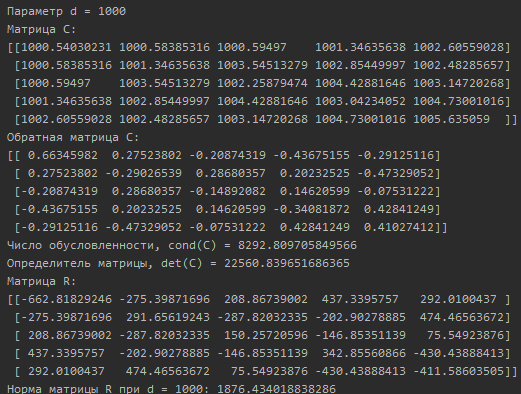
Также при работе с матрицей, хотелось бы иметь возможность смотреть на число обусловленности матрицы. При помощи числа **cond** можем проводить анализ возникающих ситуации.

Также в учебнике, нашего лектора (Устинов С.М.) сказано то, что на практике плохая обусловленность часто сопровождается малой величиной определителя матрицы **det(C)**, так как работа подразумевает некоторое личное исследование, хочу дополнительно высчитать определитель и подтвердить на практике данное утверждение из учебника.

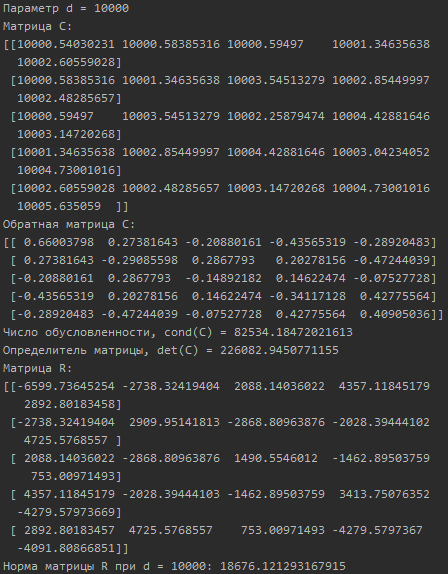


Результаты работы программы для всех трех случаев:





(Матрица при d = 10000 немного поехала, но только визуально)



**Выводы:**

Посмотрев на то, как отработала программа и делая анализ этих данных можно объединить данные в следующую таблицу:

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр d |  | cond(C) | det(C) |
| 10 | 32.5 | 146 | 173 |
| 1000 | 1876 | 8293 | 22561 |
| 10000 | 18676 | 82534 | 226083 |

Находя матрицу R, по данной формуле, очевидно становится то, что так как мы не получили ошибки при работе программы или же просто R = 0 можно сказать, что обратная матрица С-1 не устойчива, тем самым, мы имеем получаем ненулевую матрицу R.

Из полученных нами числах обусловленности можно точно сказать, что матрицы во всех трех случаях являются плохо обусловленными, что обозначает неустойчивость матрицы, а это как мы знаем говорит о том, что малым изменениям элементов матрицы C отвечают большие изменения матрицы C-1.

Исходя из моего дополнительного исследования, я не заметил, что для моих матриц плохая обусловленность также сопровождается малой величиной определителя матрицы. Для конкретно наших исследуемых матриц это явно не так.

Весь исходный код и файлы можно найти в моем профиле:

[github.com/b0r1ngx/ComputationalMath](https://github.com/b0r1ngx/ComputationalMath)