Содержание

1 Задание

Для матрицы A с $N = 5$ столбцами выполните следующее.

0. Постройте тестовый пример(вычислите вектор правых частей b), точным решением которого является вектор, состоящий из единиц. Изменяя первый элемент вектора b, постройте основную задачу, точного решения НЕ имеющую.

1. Решите переопределенную СЛАУ $Ax = b$ в смысле наименьших квадратов с помощью QR разложения с выбором главного столбца, используя отражения Хаусхолдера (не формируя матрицу Q в явном виде).

2. Укажите норму невязки полученного решения для основной задачи.

3. Продемонстрируйте правильность работы метода на тестовом примере (с точным решением).

4. Вычислите в явном виде матрицу Q, не выполняя перемножений и не формируя в явном виде матрицы промежуточных отражений.

5. Сформируйте в явном виде матрицу перестановок Π. Запишите полученное разложение в виде матричного равенства $A\Pi = QR$. Проверьте его непосредственным перемножением (представьте матрицу $A\Pi - QR$).

6. Найдите SVD разложение матрицы A, используя одноименную функцию библиотеки *numpy.linalg*. Решите исходную задачу с помощью найденного разложения.

7. Найдите псевдообратную к A матрицу A^+, используя полученное SVD разложение. Сравните друг с другом и единичной матрицей произведения A^+A и AA^+. Решите исходную задачу с помощью псевдообратной матрицы.

8. Выполните п.п. 1 и 6 для основной задачи большего порядка.

Исходные данные

Матрица A имеет N столбцов и $N+1$ строку. Ее элементы задаются для четных вариантов (по списку в журнале группы) $A_{ij} = (i-j)^2$; для нечетных вариантов (по списку в журнале группы) $A_{ij} = i+j-2ij$; $(1 \le i \le N+1, 1 \le j \le N-1)$. В первые шесть элементов N - го столбца (для всех вариантов) впишите дату рождения Вашего тезки-математика в формате "mm.yyyy".

2 Жордан Мари Энмон Камиль

Дата рождения 01.1838, Лион.

Французский математик, известный благодаря своим фундаментальным работам в теории групп и «Курсу анализа». Он родился в Лионе и учился в Политехнической школе. По образованию Жордан был инженером; позже он преподавал в Политехнической школе и Коллеж де Франс.

3 Информация о выборе главного столбца

Шаг 1.

Норма 0 столбца = 7.416198
Норма 1 столбца = 5.567764
Норма 2 столбца = 18.41195
Норма 3 столбца = 31.28898
Норма 4 столбца = 11.78983

	0	1	2	3	4
0	0.0	1.0	2.0	3.0	0.0
1	1.0	0.0	-1.0	-2.0	1.0
2	2.0	-1.0	-4.0	-7.0	1.0
3	3.0	-2.0	-7.0	-12.0	8.0
4	4.0	-3.0	-10.0	-17.0	3.0
5	5.0	-4.0	-13.0	-22.0	8.0

Промежуточная матрица с зануленным столбцом

	0	1	2	3	4
0	-31.288976	-5.529104	-18.409040	7.350832	10.610766
1	0.000000	0.380828	0.190414	0.571242	0.381098
2	0.000000	0.332899	0.166449	0.499348	-1.166159
3	0.000000	0.284969	0.142485	0.427454	4.286585
4	0.000000	0.237039	0.118520	0.355559	-2.260671
5	0.000000	0.189110	0.094555	0.283665	1.192073

Шаг 2.

Норма 1 подстолбца = 0.6549879
Норма 2 подстолбца = 0.327494
Норма 3 подстолбца = 0.9824819
Норма 4 подстолбца = 5.139226

	0	1	2	3	4
0	-31.288976	-5.529104	-18.409040	7.350832	10.610766
1	0.000000	0.380828	0.190414	0.571242	0.381098
2	0.000000	0.332899	0.166449	0.499348	-1.166159
3	0.000000	0.284969	0.142485	0.427454	4.286585
4	0.000000	0.237039	0.118520	0.355559	-2.260671
5	0.000000	0.189110	0.094555	0.283665	1.192073

Промежуточная матрица с зануленным столбцом

	0	1	2	3	4
0	-31.288976	10.610766	-18.409040	7.350832	-5.529104
1	0.000000	-5.139226	-0.064993	-0.194979	-0.129986
2	0.000000	0.000000	0.220404	0.661211	0.440807
3	0.000000	0.000000	-0.055842	-0.167525	-0.111683
4	0.000000	0.000000	0.223114	0.669341	0.446227
5	0.000000	0.000000	0.039402	0.118205	0.078803

Шаг 3.

Норма 2 подстолбца = 0.3209801
Норма 3 подстолбца = 0.9629402
Норма 4 подстолбца = 0.6419601

	0	1	2	3	4
0	-31.288976	10.610766	-18.409040	7.350832	-5.529104
1	0.000000	-5.139226	-0.064993	-0.194979	-0.129986
2	0.000000	0.000000	0.220404	0.661211	0.440807
3	0.000000	0.000000	-0.055842	-0.167525	-0.111683
4	0.000000	0.000000	0.223114	0.669341	0.446227
5	0.000000	0.000000	0.039402	0.118205	0.078803

Промежуточная матрица с зануленным столбцом

	0	1	2	3	4
0	-31.288976	10.610766	7.350832	-18.409040	-5.529104
1	0.000000	-5.139226	-0.194979	-0.064993	-0.129986
2	0.000000	0.000000	-0.962940	-0.320980	-0.641960
3	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000000	0.000000

Полученная треугольная система

$$R = \begin{pmatrix} -31.288976 & 10.610766 & 7.350832 & -18.409040 & -5.529104 \\ 0.000000 & -5.139226 & -0.194979 & -0.064993 & -0.129986 \\ 0.000000 & 0.000000 & -0.962940 & -0.320980 & -0.641960 \\ 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 \\ 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 \\ 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 \end{pmatrix} \tag{1}$$

$$\tilde{b} = \begin{pmatrix} -37.26552161 \\ -5.52918474 \\ -1.92588031 \\ 0. \\ 0. \\ 0. \end{pmatrix} \tag{2}$$

Найденное решение

$$x = \begin{pmatrix} 0. \\ 2. \\ 0. \\ 2. \\ 1. \end{pmatrix} \tag{3}$$

Невязка решения $\rho = 7.10542 \cdot 10^{-28}$

4 Тестовый пример

$$A = \begin{pmatrix} 0.0 & 1.0 & 2.0 & 3.0 & 0.0 \\ 1.0 & 0.0 & -1.0 & -2.0 & 1.0 \\ 2.0 & -1.0 & -4.0 & -7.0 & 1.0 \\ 3.0 & -2.0 & -7.0 & -12.0 & 8.0 \\ 4.0 & -3.0 & -10.0 & -17.0 & 3.0 \\ 5.0 & -4.0 & -13.0 & -22.0 & 8.0 \end{pmatrix} b = \begin{pmatrix} 6. \\ -1. \\ -9. \\ -10. \\ -23. \\ -26. \end{pmatrix} \tag{4}$$

Точным решением является вектор из единиц.

Полученная система

$$R = \begin{pmatrix} -31.288976 & 10.610766 & 7.350832 & -18.409040 & -5.529104 \\ 0.000000 & -5.139226 & -0.194979 & -0.064993 & -0.129986 \\ 0.000000 & 0.000000 & -0.962940 & -0.320980 & -0.641960 \\ 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 \\ 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 \\ 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 \end{pmatrix} \tag{5}$$

$$\tilde{b} = \begin{pmatrix} -37.26552161 \\ -5.52918474 \\ -1.92588031 \\ 0. \\ 0. \\ 0. \end{pmatrix} \tag{6}$$

Полученное решение системы приведенной выше

$$x = \begin{pmatrix} 2. \\ 1. \\ 2. \\ 0. \\ 0. \end{pmatrix} \tag{7}$$

Тестовая задача точного решения не имеющая

$$A = \begin{pmatrix} 0.0 & 1.0 & 2.0 & 3.0 & 0.0 \\ 1.0 & 0.0 & -1.0 & -2.0 & 1.0 \\ 2.0 & -1.0 & -4.0 & -7.0 & 1.0 \\ 3.0 & -2.0 & -7.0 & -12.0 & 8.0 \\ 4.0 & -3.0 & -10.0 & -17.0 & 3.0 \\ 5.0 & -4.0 & -13.0 & -22.0 & 8.0 \end{pmatrix} b = \begin{pmatrix} \boxed{7.} \\ -1. \\ -9. \\ -10. \\ -23. \\ -26. \end{pmatrix} \tag{8}$$

$rank(A) = 3$, $rank(A|b) = 4$ система точного решения не имеет.

5 Матричные разложения

Матрица Q

	0	1	2	3	4	5
0	-0.095880	-0.197961	-0.691842	0.280312	0.250449	0.575911
1	0.063920	-0.062608	-0.537858	-0.767396	0.093471	-0.324144
2	0.223721	0.267326	-0.423274	0.275405	-0.769061	-0.178946
3	0.383522	-0.764812	-0.032893	0.300761	0.067973	-0.414505
4	0.543322	0.538032	-0.115307	0.240289	0.545107	-0.217132
5	0.703123	-0.104943	0.196275	-0.329371	-0.187940	0.558815

Матрица перестановок Π

	0	1	2	3	4
0	0.0	0.0	1.0	0.0	0.0
1	0.0	0.0	0.0	0.0	1.0
2	0.0	0.0	0.0	1.0	0.0
3	1.0	0.0	0.0	0.0	0.0
4	0.0	1.0	0.0	0.0	0.0

Проверим равенство $A\Pi = QR$:

	0	1	2	3	4
0	6.661338e-15	-2.886580e-15	-1.110223e-15	5.551115e-15	1.443290e-15
1	-6.661338e-16	-6.661338e-16	0.000000e+00	6.661338e-16	1.110223e-16
2	-3.552714e-15	1.110223e-15	8.881784e-16	-4.440892e-16	2.220446e-16
3	-5.329071e-15	0.000000e+00	4.440892e-16	-2.664535e-15	-4.440892e-16
4	-1.065814e-14	2.220446e-15	1.776357e-15	-5.329071e-15	-8.881784e-16
5	-1.065814e-14	1.776357e-15	1.776357e-15	-7.105427e-15	-8.881784e-16

Таблица 1: AΠ - QR

6 SVD

	0	1	2	3	4	5
0	0.089725	-0.212609	-0.688312	-0.502896	0.359235	-0.301693
1	-0.065307	-0.069055	-0.536902	0.829588	0.119578	0.013719
2	-0.213226	0.268281	-0.428060	-0.236973	-0.323654	0.733768
3	-0.410932	-0.750861	-0.021238	-0.033963	-0.507228	-0.091970
4	-0.523288	0.555389	-0.125240	-0.025027	-0.289069	-0.563773
5	-0.706770	-0.076188	0.196445	-0.030729	0.641138	0.209949

Таблица 2: U

	0
0	3.894933e+01
1	4.951476e+00
2	1.196871e+00
3	8.183017e-16
4	3.183863e-16

Таблица 3: Singular values

	0	1	2	3	4
0	-0.188747	0.141768	0.472282	0.802797	-2.770269e-01
1	0.011218	-0.068785	-0.148787	-0.228790	-9.595104e-01
2	-0.815015	-0.524567	-0.234119	0.056328	5.094924e-02
3	0.293785	-0.047160	-0.787035	0.540410	-7.155734e-17
4	0.462266	-0.835330	0.283860	0.089203	5.551115e-17

Таблица 4: V^T

Полученное решение при помощи SVD-разложения

	0
0	1.0
1	1.0
2	1.0
3	1.0
4	1.0

Проверка ортогональности матриц V и U

	0	1	2	3	4	5
0	1.000e+00	-2.775e-17	8.326e-17	8.326e-17	-1.110e-16	5.551e-17
1	-2.775e-17	1.000e+00	-1.422e-16	-3.252e-17	2.081e-16	-2.671e-16
2	8.326e-17	-1.422e-16	1.000e+00	-1.812e-16	-5.551e-17	-4.857e-17
3	8.326e-17	-3.252e-17	-1.812e-16	1.000e+00	-5.204e-17	-8.239e-17
4	-1.110e-16	2.081e-16	-5.551e-17	-5.204e-17	1.000e+00	-1.110e-16
5	5.551e-17	-2.671e-16	-4.857e-17	-8.239e-17	-1.110e-16	1.000e+00

Таблица 5: U^TU

	0	1	2	3	4	5
0	1.000e+00	2.315e-16	2.498e-16	-2.636e-16	5.551e-17	1.526e-16
1	2.315e-16	1.000e+00	1.578e-16	-3.144e-17	1.101e-16	-8.803e-17
2	2.498e-16	1.578e-16	1.000e+00	2.498e-16	5.551e-17	-1.665e-16
3	-2.636e-16	-3.144e-17	2.498e-16	1.000e+00	-4.857e-17	-1.283e-16
4	5.551e-17	1.101e-16	5.551e-17	-4.857e-17	1.000e+00	1.804e-16
5	1.526e-16	-8.803e-17	-1.665e-16	-1.283e-16	1.804e-16	1.000e+00

Таблица 6: UU^T

	0	1	2	3	4
0	1.000000e+00	1.665335e-16	-2.775558e-17	-2.081668e-17	-6.475048e-17
1	1.665335e-16	1.000000e+00	1.110223e-16	-1.942890e-16	-4.831531e-18
2	-2.775558e-17	1.110223e-16	1.000000e+00	1.595946e-16	-1.742552e-16
3	-2.081668e-17	-1.942890e-16	1.595946e-16	1.000000e+00	8.782203e-18
4	-6.475048e-17	-4.831531e-18	-1.742552e-16	8.782203e-18	1.000000e+00

Таблица 7: VV^T

	0	1	2	3	4
0	1.000000e+00	5.551115e-17	9.020562e-17	-9.119900e-17	1.372776e-16
1	5.551115e-17	1.000000e+00	-9.020562e-17	-8.399565e-17	3.000320e-17
2	9.020562e-17	-9.020562e-17	1.000000e+00	-7.997363e-17	-1.290107e-16
3	-9.119900e-17	-8.399565e-17	-7.997363e-17	1.000000e+00	1.942890e-16
4	1.372776e-16	3.000320e-17	-1.290107e-16	1.942890e-16	1.000000e+00

Таблица 8: V^TV

	0	1	2	3	4
0	38.949331	0.000000	0.000000	0.0	0.0
1	0.000000	4.951476	0.000000	0.0	0.0
2	0.000000	0.000000	1.196871	0.0	0.0
3	0.000000	0.000000	0.000000	0.0	0.0
4	0.000000	0.000000	0.000000	0.0	0.0
5	0.000000	0.000000	0.000000	0.0	0.0

Таблица 9: Σ

Проверка матричного равенства $U^TAV = \Sigma$

Псевдообратная матрица

	0	1	2	3	4
0	2.131628e-14	0.000000e+00	-1.665335e-15	-2.780609e-15	-1.548767e-16
1	-1.332268e-15	8.881784e-16	2.775558e-16	-2.151431e-16	-4.302666e-16
2	-2.508410e-15	1.734723e-16	1.110223e-15	-7.537641e-16	-5.725431e-16
3	-1.013208e-15	-1.333499e-16	1.510774e-16	-1.308964e-18	-2.318506e-17
4	-3.298696e-16	-8.472348e-16	-3.166872e-16	1.304668e-16	2.052876e-16
5	8.426199e-16	1.250281e-17	4.588944e-16	3.547494e-16	-3.331899e-17

Таблица 10: $U^T AV - \Sigma$

Если $A = U\Sigma V^T$ - сингулярное разложение матрицы A, тогда $A^+ = V\Sigma^+ U^T$ Из алгоритма решение СЛАУ с помощью SVD разложения:

$$c = U^T b,\ y = \Sigma^+ c,\ x = Vy \Rightarrow x = V\Sigma^+ U^T b = A^+ b$$

	0	1	2	3	4	5
0	0.467793	0.365766	0.293131	0.014752	0.089077	-0.130518
1	0.304955	0.236036	0.183108	0.018243	0.045270	-0.087613
2	0.142117	0.106306	0.073086	0.021734	0.001464	-0.044707
3	-0.020721	-0.023423	-0.036937	0.025225	-0.042342	-0.001802
4	0.011261	-0.009009	-0.068694	0.147523	-0.109234	0.028153

Таблица 11: A^+

Решение с помощью псевдообратной матрицы

	0
0	1.0
1	1.0
2	1.0
3	1.0
4	1.0

Сравнение AA^+ и A^+A

Судя по результатам перемножения псевдообратной матрицы на изначальную, их произведения не равны между собой и не равны единичной.

	0	1	2	3	4	5
0	0.527027	0.378378	0.218468	0.137387	-0.078829	-0.182432
1	0.378378	0.297297	0.225225	0.090090	0.063063	-0.054054
2	0.218468	0.225225	0.300676	-0.104730	0.314189	0.046171
3	0.137387	0.090090	-0.104730	0.733108	-0.199324	0.343468
4	-0.078829	0.063063	0.314189	-0.199324	0.597973	0.302928
5	-0.182432	-0.054054	0.046171	0.343468	0.302928	0.543919

Таблица 12: AA^+

	0	1	2	3	4
0	7.000000e-01	4.000000e-01	1.000000e-01	-2.000000e-01	-2.220446e-16
1	4.000000e-01	3.000000e-01	2.000000e-01	1.000000e-01	-2.220446e-16
2	1.000000e-01	2.000000e-01	3.000000e-01	4.000000e-01	-3.330669e-16
3	-2.000000e-01	1.000000e-01	4.000000e-01	7.000000e-01	5.551115e-17
4	1.387779e-16	-9.714451e-17	-2.775558e-16	-5.551115e-16	1.000000e+00

Таблица 13: A^+A

7 Большая система

N	***SVD***		***Hausholder***	
	Точность	***Время***	***Точность***	***Время***
5	1.49144e-30	0.0009965	1.2154e-28	0.001996
10	2.35048e-26	0.001991	3.1756e-26	0.004164
100	1.4927e-17	0.009149	2.2106e-19	2.1235
500	8.41267e-13	0.5747	2.7845e-14	1579.5157
1000	1.9031e-12	4.1839	–	>часа

8 Вывод

В ходе лабораторной работы был реализован метод наименьших квадратов для решения переопределенной СЛАУ с помощью QR разложения с выбором главного столбца, используя отражения Хаусхолдера. Также были сформированы матрицы Q и П в явном виде, проверена истинность матричного равенства $A\Pi - QR$. Была создана подпрограмма для решения исходной задачи с использованием SVD разложения и псевдообратной матрицы. Выполнены вычислительные эксперименты для задач большего порядка, из которых видно, что порядок точности найденного решения обоих методов примерно равен, а временная составляющая уступает у отражений Хаусхолдера.

9 Приложение

```
import numpy as np
import pandas as pd
import time

# Число столбцов в матрице A
N = 500
# Вектор перестановок
trans = np.linspace(1, N, N)
#  Ввод матрицы A
A = np.zeros((N + 1, N))
for i in range(N + 1):
    for j in range(N):
        A[i][j] = i + j - 2 * i * j
#  MM.ГГГГ
happy_birthday = [0, 1, 1, 8, 3, 8]
for i in range(6):
    A[i][N - 1] = happy_birthday[i]
# копия матрицы A
Atr = A.copy()
# Тестовый пример
# Вектор точного решения
solution = np.zeros(N)
for i in range(N):
    solution[i] = 1
# Вектор правой части
b = np.dot(A, solution)
# Представление в виде столбца N+1 x 1
b = b.reshape((N + 1, 1))
# Копирование вектора правой части
btr = np.copy(b)
btr = btr.reshape(N + 1)

# Инициализация матрицы Q
Q = np.eye(N + 1)
# Матрица перестановок
Matr_of_swap = np.eye(N)


# функция перестановки столбцов в матрице
def swap(A, n1, n2):
```

```python
    if n1 != n2:
        # Перестановка в матрице A
        temp1 = A[:, n1].copy()
        temp2 = A[:, n2].copy()
        A[:, n2] = temp1[:]
        A[:, n1] = temp2[:]
        # Перестановка, для восстановления вектора решения
        temp3 = trans[n1 - 1]
        trans[n1 - 1] = trans[n2 - 1]
        trans[n2 - 1] = temp3


# Отражения Хаусхолдера
def householder_mod(A, b):
    # допуск к изменению локальной переменной, как к глобальной
    global Q
    for k in range(N):
        print(k + 1, 'Шаг Хаусхолдера')
        # номер ведущего столбца
        h = -1
        # норма ведущего столбца
        sr = -1
        # выбор ведущего столбца
        for j in range(k, N):
            print("Норма", j - k, "столбца = ", format(np.linalg.norm(A[k:, j]), '.7'), "\\\ ")
            if np.linalg.norm(A[k:, j]) > sr:
                h = j
                sr = np.linalg.norm(A[k:, j])
        # Условие на проверку равенства нулю всех столбцов на k-шаге алгоритма
        if sr <= 10 ** (-8):
            break
        # Перестановка ведущего столбца на первое место на k-шаге в матрице A
        swap(A, k, h)
        # Перестановка столбцов в матрице перестановок
        swap(Matr_of_swap, k, h)
        # ведущий столбец
        main_column = A[k:, k]
        # норма ведущего столбца
        norm_main_column = sr
```

```
        # единичный базисный вектор
        e_1 = np.zeros(len(main_column))
        e_1[0] = 1
        # вектор w в отражениях Хаусхолдера
        w = np.zeros((1, N + 1 - k))
        # вычисление w, исходя из условия на выбор знака
        if main_column[0] >= 0:
            w[0] = main_column + norm_main_column * e_1
        else:
            w[0] = main_column - norm_main_column * e_1

        # Матрица Хаусхолдера
        mainE = np.eye(N + 1)
        mainE[k:N + 1, k:N + 1] = np.eye(N + 1 - k) - 2 / (np.linalg.norm(w[0]) ** 2) * np.dot(np.transpose(w), w)
        # вычисление матрицы Q
        Q = np.dot(Q, mainE)

        # A^T * w
        b_help = np.dot(np.transpose(A[k:N + 1, k:N + 1]), np.transpose(w))
        # (w, b)
        bb_help = np.dot(w, b[k:N + 1])
        # 2/(w,w)
        beta_help = 2 / np.dot(w, np.transpose(w))[0][0]
        # Поэлементно вычитаем из вектора правой части и матрицы
        for i in range(k, N + 1):
            b[i] -= beta_help * np.dot(bb_help, w)[0][i - k]
            for j in range(k, N):
                A[i][j] -= beta_help * np.dot(np.transpose(w), np.transpose(b_help))[i - k][j - k]
    return A, b

# обратный ход метода Гаусса
def gauss(A, B):
    sol = np.zeros(len(A[0]))
    for i in range(len(A[0]) - 1, -1, -1):
        s = 0
        if i == len(A[0]) - 1:
            sol[i] = B[i] / A[i][i]
        else:
```

```
            for j in range(i + 1, len(A[0]), 1):
                s += A[i][j] * sol[j]
            sol[i] = (B[i] - s) / A[i][i]
    return sol

# вычисление ранга треугольной матрицы
def rank_matrix(A):
    rank = 0
    for i in range(N):
        if abs(A[i][i]) <= 10 ** (-8) :
            break
        else:
            rank += 1
    return rank

# функция перестановки компонент вектора решения
def trsol(pst, sol):
    true_sol = np.zeros(len(pst))
    for i in range(len(pst)):
        true_sol[pst[i] - 1] = sol[i]
    return true_sol

# Решение методом SVD разложения
def svd_solve(A, B):
    # Инициализация вектора решения
    solution = np.zeros(len(A[0]))
    # SVD разложение
    U, S, V_t = np.linalg.svd(Atr)
    # ранг матрицы
    r = 0
    # вспомогательный вектор y
    y = np.zeros(len(A[0]))
    # Вычисление ранга по сингулярным числам
    for i in range(len(S)):
        if S[i] > 10 ** (-8):
            r += 1
    # Вспомогательный вектор c = U^T * b
    c = np.dot(np.transpose(U), B)
    # вычисление вектора y
    for i in range(r):
        y[i] = c[i] / S[i]
    # дополнение нулями компонент с r + 1 до n
    y.resize(N)
```

```
    # Псевдообратная матрица к матрице \Sigma^+
    sigm = np.zeros((N + 1, N))
    for i in range(r):
        sigm[i][i] = S[i]
    # вычисление решения
    solution = np.dot(np.transpose(V_t), y)
    # вычисление точности
    acurrancy = (np.linalg.norm(np.dot(sigm, y).reshape((N + 1,
     ↪  1)) - c)) ** 2
    return solution, acurrancy

# Получение псевдообратной матрицы
def Generalized_inverse(A):
    # SVD разложение
    U, S, V_t = np.linalg.svd(A)
    # вычисление псевдообратной матрицы \Sigma^+
    sigm_plus = np.zeros((N, N + 1))
    for i in range(3):
        sigm_plus[i][i] = 1 / S[i]
    return np.dot(np.dot(np.transpose(V_t), sigm_plus),
     ↪  np.transpose(U))

```