

Московский авиационный институт
(национальный исследовательский университет)

Институт №8 «Информационные технологии и прикладная
математика»

Кафедра 806 «Вычислительная математика и
программирование»

Курсовая работа по дисциплине «Численные методы»

Студент: Маринин И.С.
Группа: М8О-408Б-20
Преподаватель: Пивоваров Д.Е.

Москва, 2023

Задание: Решение систем линейных алгебраических уравнений с симметричными разреженными матрицами большой размерности. Метод сопряженных градиентов.

Тема: 1

Для решения систем линейных алгебраических уравнений с симметричными разреженными матрицами большой размерности часто используется метод сопряженных градиентов. Этот метод является эффективным способом решения таких систем уравнений. Он особенно полезен, когда матрицы большие и разреженные, что позволяет существенно сократить вычислительные затраты.

Основные преимущества метода сопряженных градиентов:

Метод сопряженных градиентов обычно работает быстрее, чем прямые методы решения систем линейных уравнений. Этот метод хорошо подходит для разреженных матриц и позволяет существенно сократить объем вычислений.

Примечание: Метод сопряженных градиентов также широко используется в различных областях, таких как машинное обучение, обработка сигналов, и оптимизация.

Упаковать

Сравнить с гаусс по времени

```
In [1]: import scipy.sparse as sc
from scipy import stats
from scipy.linalg import solve
from numpy.random import default_rng, rand
import numpy as np
from scipy.sparse.linalg import cg as lcg
from sklearn.datasets import make_sparse_spd_matrix
from numpy.linalg import solve, norm
import matplotlib.pyplot as plt
import math
from scipy.sparse import csr_matrix
import time
import copy
```

In [1]:

Создание разреженной симметрической матрицы и правой части СЛАУ

```
In [2]: def matr(size):
A = make_sparse_spd_matrix(size, alpha=0.95)
return A
```

```
In [3]: def xb(size):
b = np.random.rand(size, 1)
x = (np.zeros(size)+1.0).reshape(-1, 1)
return x, b
```

Метод сопряженных градиентов (CG)

```
In [4]: def norma(r):
res = 0
for i in r:
res += r*r
return math.sqrt(sum(res))
```

```
In [5]: def norma_s(r):
res = r.multiply(r)
res = res.sum()
return math.sqrt(res)
```

```
In [6]: def cg_s(A,b,x):
start_time = time.time()
r = b - A @ x
p = r
rsold = r.T @ r
k = 0
time2 = 0
while True:
Ap = A @ p
alpha = rsold / (p.T @ Ap)
time1 = time.time()
alpha = csr_matrix(alpha)
time2 += (time.time() - time1)
```

```

x = x + p * alpha
r = r - Ap * alpha
rsnew = r.T @ r
if norma_s(r) < 0.001:
    print("Количество шагов :", k)
    print(time.time() - start_time - time2)

    return x,k, time.time() - start_time - time2
time3 = time.time()
r_del = csr_matrix(rsnew / rsold)
time2 += (time.time() - time3)
p = r + p * r_del
rsold = rsnew
k = k + 1

```

```

In [7]: def cg(A,b,x):
        start_time = time.time()
        r = b - A @ x
        p = r
        rsold = r.T @ r
        k = 0
        while True:
            Ap = A @ p
            alpha = rsold / (p.T @ Ap)
            x = x + alpha * p
            r = r - alpha * Ap
            rsnew = r.T @ r
            if norma(r) < 0.001:
                print("Количество шагов :", k)
                print(time.time() - start_time)
                return x,k, time.time() - start_time
            p = r + (rsnew / rsold) * p
            rsold = rsnew
            k = k + 1

```

Проверка алгоритма

```

In [8]: A = np.array([[4,1],[1,3]])
        b = np.array([1,2]).reshape(-1, 1)
        x = np.array([2,1]).reshape(-1, 1)

```

```

In [9]: (np.zeros(3)+1.0).reshape(-1, 1)

```

```

Out[9]: array([[1.],
               [1.],
               [1.]])

```

```

In [10]: np.array([2,1]).reshape(-1, 1)

```

```

Out[10]: array([[2],
                [1]])

```

```

In [11]: r = b - A @ x
        r

```

```

Out[11]: array([[ -8],
                [-3]])

```

```

In [12]: cg(A,b,x)

```

```

Количество шагов : 1
0.004196882247924805

```

```
Out[12]: (array([[0.09090909],
                [0.63636364]]),
          1,
          0.004220724105834961)
```

```
In [13]: lcg(A,b,x)
```

```
Out[13]: (array([0.09090909, 0.63636364]), 0)
```

Результат

```
In [14]: def ans(size = 5):
          A = matr(size)
          x,b = xb(size)
          CG1_x, CG1_k, CG1_t = cg(A,b,x)
          CG2 = lcg(A,b,x)[0]
          A,b, x = csr_matrix(A), csr_matrix(b), csr_matrix(x)
          CG3_x, CG3_k, CG3_t = cg_s(A,b,x)
          j = 0
          summa = 0
          while j < size:
              summa += (CG2[j]-CG1_x[j])**2
              j = j + 1
          mer1 = summa/size
          cg_h = CG3_x.toarray()
          j = 0
          summa = 0
          while j < size:
              summa += (CG2[j]-cg_h[j][0])**2
              j = j + 1
          mer2 = summa/size
          return CG1_x, CG2,CG3_x.A, mer1,mer2, CG1_t, CG3_t, A.A, b.A,
```

```
In [15]: CG = ans(size = 50)
```

```
Количество шагов : 25
0.00964808464050293
Количество шагов : 23
0.1418170928955078
```

```
In [16]: A = CG[7]
          B = CG[8]
```

```
In [17]: start_time = time.time()
          solve(A,B)
          print(time.time()-start_time)

0.009508848190307617
```

```
In [18]: print("Среднеквадратичная ошибка: {}".format(CG[3][0]))
          print("Среднеквадратичная ошибка: {}".format(CG[4]))

Среднеквадратичная ошибка: 4.280656997972435e-10
Среднеквадратичная ошибка: 1.054741529553345e-08
```

```
In [19]: CG = ans(size = 75)
```

```
Количество шагов : 34
0.0354008674621582
Количество шагов : 29
0.12494468688964844
```

```
In [20]: A = CG[7]
        B = CG[8]
```

```
In [21]: start_time = time.time()
        solve(A,B)
        print(time.time()-start_time)

0.002691507339477539
```

```
In [22]: print("Среднеквадратичная ошибка: {}".format(CG[3][0]))
        print("Среднеквадратичная ошибка: {}".format(CG[4]))

Среднеквадратичная ошибка: 4.042087768754371e-10
Среднеквадратичная ошибка: 9.658966840006633e-08
```

```
In [23]: CG = ans(size = 100)

Количество шагов : 61
0.04278683662414551
Количество шагов : 53
0.26924800872802734
```

```
In [24]: A = CG[7]
        B = CG[8]
```

```
In [25]: start_time = time.time()
        solve(A,B)
        print(time.time()-start_time)

0.0037107467651367188
```

```
In [26]: print("Среднеквадратичная ошибка: {}".format(CG[3][0]))
        print("Среднеквадратичная ошибка: {}".format(CG[4]))

Среднеквадратичная ошибка: 5.355725141794007e-10
Среднеквадратичная ошибка: 1.0824983390299796e-07
```

```
In [27]: print("Ответ:\n {}".format(CG[0]))
        print("Точный ответ:\n {}".format(CG[1]))
```

Ответ:

[5.68615528e+01]
[2.85993439e+01]
[1.60278475e+01]
[1.35275894e+01]
[3.40400926e+01]
[5.99333248e+00]
[2.05370967e+01]
[8.45763554e+01]
[6.23992886e+01]
[5.47267687e+00]
[1.48761628e+01]
[1.72161540e+01]
[1.49196684e+00]
[1.02738075e+01]
[2.76368981e+01]
[2.35954945e+00]
[2.12268149e+01]
[1.44608288e+01]
[1.22366721e+01]
[1.10207213e+01]
[3.31257355e+01]
[4.12453496e+01]
[1.08137929e+01]
[3.13439019e+01]
[1.64837234e+01]
[1.18662828e+00]
[1.61945174e+01]
[3.88264059e+01]
[1.12462561e+00]
[9.49512508e+00]
[4.01008252e+01]
[3.38434164e+01]
[2.79592312e+01]
[1.87349965e+01]
[1.70644288e+00]
[2.91208232e+01]
[8.22823259e+00]
[2.18241795e+01]
[1.83831894e+01]
[2.59058973e+00]
[2.89475638e+00]
[5.12055818e+00]
[2.48060804e+01]
[3.78051133e+00]
[7.45216982e+00]
[9.91623839e-01]
[1.16925512e+01]
[7.16313881e+00]
[3.15163513e+00]
[1.73119063e+01]
[2.33229395e+01]
[2.41957864e+00]
[3.01147706e+01]
[5.87330528e+01]
[1.11253328e+01]
[2.08450607e+00]
[1.16463834e+01]
[1.39721470e+01]
[6.34156465e-01]
[1.37615268e+01]
[3.05901736e-01]
[1.82036060e+01]
[4.08937720e-02]

[1.83933419e+01]
 [1.42821281e+01]
 [4.11357794e+00]
 [8.72520288e+01]
 [1.19749444e+01]
 [2.34608710e+01]
 [2.26077976e+01]
 [2.89476752e+01]
 [8.55381172e+01]
 [7.48897217e+00]
 [2.98684985e+01]
 [1.42541573e+01]
 [8.92435182e+00]
 [2.63795297e+00]
 [3.53174586e+01]
 [8.38839489e+00]
 [7.73804257e+00]
 [1.42639727e+01]
 [6.31041412e+00]
 [2.85836972e+01]
 [1.35956296e+01]
 [1.90322997e+01]
 [6.76777198e+00]
 [5.88868941e+00]
 [4.22346617e+00]
 [5.85573544e+01]
 [4.24656868e+00]
 [1.59861693e+01]
 [4.71165670e+00]
 [2.07458719e+01]
 [2.67809774e+00]
 [5.74487217e+00]
 [1.80060051e+01]
 [2.58444895e+01]
 [2.27879359e+01]
 [1.21999673e+01]
 [4.37069816e+00]]

Точный ответ:

[5.68615308e+01 2.85993676e+01 1.60278445e+01 1.35275490e+01
 3.40400584e+01 5.99333285e+00 2.05370766e+01 8.45762872e+01
 6.23993436e+01 5.47268701e+00 1.48761747e+01 1.72161383e+01
 1.49195737e+00 1.02738361e+01 2.76369391e+01 2.35955132e+00
 2.12268220e+01 1.44608340e+01 1.22366932e+01 1.10207521e+01
 3.31257246e+01 4.12453835e+01 1.08138016e+01 3.13439386e+01
 1.64837370e+01 1.18663674e+00 1.61945057e+01 3.88264344e+01
 1.12462868e+00 9.49513987e+00 4.01008364e+01 3.38434407e+01
 2.79592709e+01 1.87349919e+01 1.70645012e+00 2.91208497e+01
 8.22823894e+00 2.18242131e+01 1.83831921e+01 2.59058003e+00
 2.89477373e+00 5.12057460e+00 2.48060853e+01 3.78051103e+00
 7.45216579e+00 9.91628574e-01 1.16925285e+01 7.16313915e+00
 3.15161588e+00 1.73118773e+01 2.33229347e+01 2.41956180e+00
 3.01147543e+01 5.87330823e+01 1.11253570e+01 2.08451871e+00
 1.16464013e+01 1.39721551e+01 6.34158939e-01 1.37615254e+01
 3.05895537e-01 1.82035584e+01 4.08852070e-02 1.83933757e+01
 1.42821111e+01 4.11355979e+00 8.72519950e+01 1.19749433e+01
 2.34608767e+01 2.26078085e+01 2.89476612e+01 8.55380774e+01
 7.48896638e+00 2.98685045e+01 1.42541561e+01 8.92434068e+00
 2.63794262e+00 3.53174209e+01 8.38840809e+00 7.73805768e+00
 1.42639800e+01 6.31043937e+00 2.85836305e+01 1.35956526e+01
 1.90322783e+01 6.76775204e+00 5.88868597e+00 4.22347978e+00
 5.85574072e+01 4.24657207e+00 1.59862189e+01 4.71162997e+00
 2.07458583e+01 2.67808469e+00 5.74486412e+00 1.80060120e+01
 2.58445162e+01 2.27879503e+01 1.21999587e+01 4.37068529e+00]

In [27]:

Сравнение времени со встроенной функцией

In [28]:

```
A = CG[7]  
B = CG[8]
```

In [29]:

```
start_time = time.time()  
solve(A,B)  
print(time.time()-start_time)  
  
0.0007696151733398438
```

In [29]:

Вывод В ходе выполнения лабораторной работы я изучил метод сопряженных градиентов как эффективный способ решения систем линейных алгебраических уравнений с симметричными разреженными матрицами большой размерности. Также я убедился, что этот метод является особенно полезным в случаях, когда матрицы разреженные, что позволяет существенно сократить вычислительные затраты. Изучение этого метода позволило лучше понять принципы решения систем линейных уравнений и его практическое применение в контексте симметричных разреженных матриц большой размерности.