МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Курсовая работа
по курсу «Численные методы»

Тема: «Нахождение собственных значений и собственных векторов симметричных разреженных матриц большой размерности. Метод Ланцоша».

Выполнила: Лябина М.А.
Группа: М8О-409Б-20
Преподаватель: Пивоваров Д.Е.
Дата:
Оценка:
Подпись:

Москва, 2023

Условие

Для разреженной симметричной матрицы большого размера найти собственные значения и собственные вектора методом Ланцоша.

Описание алгоритма

Алгоритм Ланцоша соединяет в себе метод Ланцоша для построения крыловского подпространства с процедурой Рэлея-Ритца. Входными данными алгоритма служат квадратная матрица A=A^T и вектор начального приближения b. Необходимо найти трехдиагональную симметричную матрицу $T_k = Q_k^T A Q_k$, собственные значения которой приближают собственные значения матрицы A. Иными словами, на k-м шаге из ортонормированных векторов Ланцоша строится матрица $Q_k = [q_1,q_2,...,q_k,]$ и в качестве приближенных собственных значений матрицы A принимаются числа Ритца.

Пусть $T_k = V \Lambda V^T$ – есть спектральное разложение матрицы T_k, столбцы матрицы $Q_k V$ рассматриваются как приближения к соответствующим собственным векторам матрицы A. Диагональные элементы обозначены как $\alpha_j = t_{jj}$, а элементы побочной диагонали $\beta_j = t_j - 1, j = t_j, j - 1$. После каждой итерации мы вычисляем α_j, β_j, из которых строится матрица T.

Алгоритм:

1) Заполняются начальные значения
   $q_1 = b/||b||,$
   $\beta_1 = 0,$
   $q_0 = 0,$
   где b - произвольный вектор, для всех j = 1..k
2) Пусть $z = Aq_j$
3) Вычисляется элемент на позиции t_{jj} матрицы Tk: $\alpha_j = q_j^T z$
4) Ортогонализация Грамма-Шмидта:
   $z = z - \sum_{i=1}^{j-1}(z^T q_i)q_i$
   $z = z - \sum_{i=1}^{j-1}(z^T q_i)q_i$
5) Обновление: $z = z - \alpha_j q_j - \beta_j q_{j-1}$
6) Вычисляются элементы на позициях $t_{j,j+1}$и $t_{j+1,j}$: $\beta_{j+1} = ||z||$
7) Если $\beta_{j+1} = 0$, то алгоритм завершается

Код программы

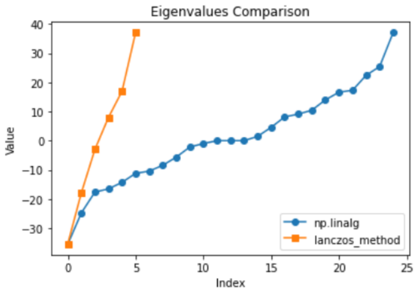
```
import numpy as np
import matplotlib.pyplot as plt

def gen_matrix():
    A = np.zeros((25, 25))
    n = np.random.randint(20, size=40)
    for k in range(len(n)):
            i = np.random.randint(25)
```

```
            j = np.random.randint(25)
            A[i][j] = n[k]
            A[j][i] = n[k]
    return A

def max_matrix_elem(A):
    i_max, j_max, max_elem = 0, 0, 0
    for i in range(A[0].size):
        for j in range(i+1, A[0].size):
            if (abs(A[i][j])>max_elem):
                max_elem = abs(A[i][j])
                i_max = i
                j_max = j
    return i_max, j_max, max_elem

def rotation_method(A, eps):
    Ak = np.copy(A)
    eigen_vectors =  np.eye(A[0].size)
    i_max, j_max, max_elem = max_matrix_elem(Ak)
    count = 0
    while (max_elem>eps):
        phi = 0.5*np.arctan(2*Ak[i_max][j_max]/(Ak[i_max][i_max]-
Ak[j_max][j_max]))
        U = np.eye(Ak.shape[0])
        U[i_max][j_max] = -np.sin(phi)
        U[j_max][i_max] = np.sin(phi)
        U[i_max][i_max] = np.cos(phi)
        U[j_max][j_max] = np.cos(phi)
        Ak = U.T @ Ak @ U
        eigen_vectors = eigen_vectors @ U
        count += 1
        i_max, j_max, max_elem = max_matrix_elem(Ak)
    eigen_values = np.array([Ak[i][i] for i in range(A[0].size)])
    return eigen_vectors, eigen_values, count

def lanczos_method(A, b, iters, EPSILON):
    Q = np.zeros((A.shape[0], iters + 1))
    alpha = np.zeros((iters))
    beta = np.zeros((iters))
    Q[:,0] = b/np.linalg.norm(b)
    for m in range(iters):
        if np.linalg.norm(b) <= EPSILON:
            break
        v = np.dot(A, Q[:,m])
        alpha[m] = np.dot(Q[:,m],v)
        if m == 0:
            v = v - alpha[m]*Q[:,m]
        else:
            v = v - alpha[m]*Q[:,m] - beta[m-1]*Q[:,m-1]
        beta[m] = np.linalg.norm(v)
        Q[:,m+1] = v/beta[m]

    T = np.dot(np.dot(Q.T,A), Q)
    Vec_T, Val, _ = rotation_method(T, 1e-16)
    Vec = Q@Vec_T
```

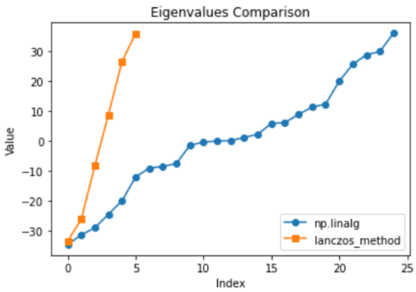
```
    return Vec, Val

A = gen_matrix()

np.set_printoptions(suppress=True)
EPSILON = 0.00000000000000001

b = np.random.rand(A.shape[0])

iters = 5
Vec, Val = lanczos_method(A, b, iters, EPSILON)
linalg_eigenvalues = np.sort(np.real(np.linalg.eigvals(A)))
kp_eigenvalues = np.sort(Val)
print("Eigenvalues with np.linalg:", linalg_eigenvalues)
print("Eigenvalues with lanczos_method:", kp_eigenvalues)

# Отображение на графике
plt.plot(linalg_eigenvalues, 'o-', label='np.linalg')
plt.plot(kp_eigenvalues, 's-', label='lanczos_method')
plt.title("Eigenvalues Comparison")
plt.xlabel("Index")
plt.ylabel("Value")
plt.legend()
plt.show()
```

Вывод программы

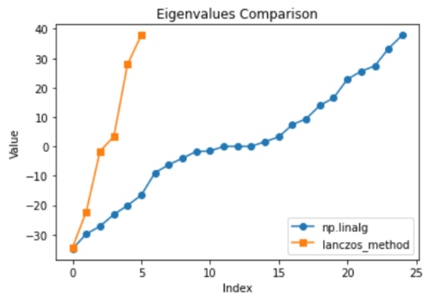
```
Eigenvalues with np.linalg: [-35.43841218 -24.73924487 -17.59639256 -16.51902752 -14.27348872
 -11.18874321 -10.44030651  -8.44100122  -5.63663479  -2.16138617
  -0.99508245  -0.00023622   0.           0.           1.47457951
   4.58082866   8.15277752   9.09985677  10.44030651  13.98259578
  16.60136893  17.25229773  22.39280616  25.38753321  37.06500564]
Eigenvalues with lanczos_method: [-35.28952593 -17.84958586  -2.96417105   7.97454519  16.95632529
  37.06381661]
```



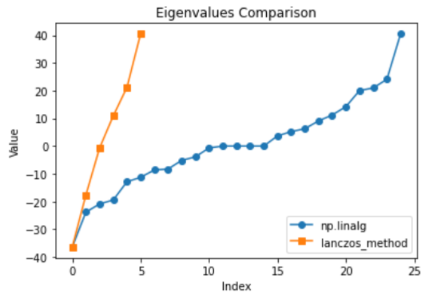

```
Eigenvalues with np.linalg: [-34.67270283 -31.45851592 -29.10309686 -24.5450239  -20.03862519
 -12.08528558  -9.10628696  -8.52625944  -7.59898415  -1.45231969
  -0.36888819   0.           0.07607801   1.2252729    2.22508494
   5.88134723   6.1511293    8.92766529  11.34572009  12.39132289
  20.02421298  25.71978915  28.84521833  30.01661831  36.12652928]
Eigenvalues with lanczos_method: [-33.55041877 -26.25340394  -8.31367208   8.80595419  26.59515522
  36.0249899 ]
```



```
Eigenvalues with np.linalg: [-34.85208145 -29.79143057 -27.16766892 -23.1858934  -20.10016514
 -16.58764802  -8.95965765  -6.24794764  -4.01869303  -1.71904451
  -1.56758552   0.           0.           0.           1.61808137
   3.21417579   7.41684681   9.39688223  13.96134232  16.51613422
  22.86906261  25.55272083  27.41799627  33.35149321  37.88308017]
Eigenvalues with lanczos_method: [-34.38930819 -22.19410626  -1.60245615   3.30193433  27.89282998
  37.86466663]
```




```
Eigenvalues with np.linalg: [-36.43358841 -23.8159298  -20.86783803 -19.43997483 -12.85885758
 -11.26948042  -8.60384659  -8.28852457  -5.14712395  -3.89776607
  -0.65693952   0.           0.           0.           0.
   3.68478741   5.21558779   6.2637151    9.07539524  11.2082299
  14.21774913  20.0394933   20.96062156  24.11435484  40.4999355 ]
Eigenvalues with lanczos_method: [-36.42214592 -17.64780484  -0.67974176  10.90581846  21.18220985
  40.49910203]
```



Вывод

В ходе данной Курсовой работы была реализована программа, алгоритм которой соответствует методу Ланцоша – нахождения собственных значений и собственных векторов симметричных разреженных матриц большой размерности. Был закреплен QR-алгоритм нахождения собственных значений матрицы, изученный ранее.