БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Лабораторная работа №5

Вычисление собственных значений и собственных векторов матрицы итерационным степенным методом

Вариант 1

Выполнил:

Черепенников Роман

2 курс 8 группа

Преподаватель:

Радкевич Елена Владимировна

Содержание

Постановка задачи

1. Итерационным степенным методом найти максимальное по модулю собственное значение матрицы $A*A^T$ и соответствующий собственный вектор с точность e = 0.1

Алгоритм решения

Для матрицы наибольшее собственное значение вещественное и простое. Поэтому алгоритм будет выглядеть следующим образом:

1) Выбрать произвольный начальный ненулевой вектор $y^{(0)}$. Положить $k = 0$ (номер итерации).

2) Вычислить $y^{(k+1)} = Ay^{(k)}$. Найти $\lambda_i^{(k+1)} = y_i^{(k+1)}/y_i^{(k)}$ $i = 1,2\ldots n$, где $y_i^{(k+1)}$ и $y_i^{(k)}$, соответствующие координаты векторов $y^{(k+1)}$ и $y^{(k)}$.

3) $\Delta^{(k+1)} = \max|\lambda_i^{(k+1)} - \lambda_i^{(k)}| \leq \varepsilon$ ($\lambda_i^{(0)}$ можно задавать произвольно), процесс завершить. За приближенное собственное значение λ_1 можно взять значение $\lambda_i^{(k+1)}$при любом i n =1, 2… n , или среднее арифметическое этих значений. Вектор $y^{(k+1)}$ приближенно представляет собой собственный вектор x_1 , соответствующий собственному значению λ_1 . Если (1) $\Delta^{(k+1)} > \varepsilon$, положить $k= k+1$ и перейти к п. 2.

При неудачном выборе $y^{(0)}$ предел отношения $y_i^{(k+1)}/y_i^{(k)}$ может не существовать. В этом случае следует задать другое начальное приближение.

Листинг программы

```
def stepennoy(A, y0, eps):
    number_of_iteratons = 0
    print('Исходная матрица A:')
    print(A)
    print('Матрица после умножения A^T справа:')
    A = np.dot(A, A.transpose())
    print(A)
    error = eps
    l0 = 0
    while error >= eps:
        y0 = y0 / np.linalg.norm(y0)
        y = np.dot(A, y0)
        l = y[0] / y0[0]
        y0 = y
        error = fabs(l - l0)
        l0 = l
        number_of_iteratons += 1
    print('Максимальное по модулю собственное значение:')
    print(l0)
    y0 = y0 / np.linalg.norm(y0)
    print('Собственный вектор соответсвующий максимальному по модулю
собственному значению: ')
    print(y0)
    print('Число итераций: ', number_of_iteratons)
    r = np.dot(A, y0) - l0 * y0
    print('Невязка: ', r)


A = np.array([
    [0.6444, 0.0000, -0.1683, 0.1184, 0.1973],
    [-0.0395, 0.4208, 0.0000, -0.0802, 0.0263],
    [0.0132, -0.1184, 0.7627, 0.0145, 0.0460],
    [0.0395, 0.0000, -0.0960, 0.7627, 0.0000],
    [0.0263, -0.0395, 0.1907, -0.0158, 0.5523]
])

stepennoy(A,np.array([[1],[1],[1],[1],[1]]),0.1)
```

Входные данные

A

0,6444	0,0000	-0,1683	0,1184	0,1973
-0,0395	0,4208	0,0000	-0,0802	0,0263
0,0132	-0,1184	0,7627	0,0145	0,0460
0,0395	0,0000	-0,0960	0,7627	0,0000
0,0263	-0,0395	0,1907	-0,0158	0,5523

Вывод программы

```
Исходная матрица A:
[[ 0.6444  0.     -0.1683  0.1184  0.1973]
 [-0.0395  0.4208  0.     -0.0802  0.0263]
 [ 0.0132 -0.1184  0.7627  0.0145  0.046 ]
 [ 0.0395  0.     -0.096   0.7627  0.    ]
 [ 0.0263 -0.0395  0.1907 -0.0158  0.5523]]
Матрица после умножения A^T справа:
[[ 0.4965221  -0.02976049 -0.10906373  0.13191428  0.09195098]
 [-0.02976049  0.18575662 -0.05029722 -0.06272879 -0.0018678 ]
 [-0.10906373 -0.05029722  0.59823034 -0.06163865  0.17564755]
 [ 0.13191428 -0.06272879 -0.06163865  0.59248754 -0.02931901]
 [ 0.09195098 -0.0018678   0.17564755 -0.02931901  0.34390336]]
Максимальное по модулю собственное значение:
[0.61194182]
Собственный вектор соответсвующий максимальному по модулю собственному значению:
[[ 0.51155128]
 [-0.10689339]
 [ 0.47723124]
 [ 0.5191756 ]
 [ 0.47916191]]
Число итераций:  2
Невязка:
[[ 0.00463556]
 [-0.02713328]
 [-0.00479646]
 [ 0.02062158]
 [-0.01259374]]
```

Вывод

Идея итерационного степенного метода заключается в построении такой итерационной последовательности векторов, чтобы в ней доминировала одна составляющая в разложении по базису из собственных векторов. Тем самым достигается сходимость по направлению к выделенному собственному вектору.

Этот метод может быть обобщён для нахождения нескольких вещественных или комплексных собственных значений. Однако следует учесть, что при этом собственные значения определяются путём решения алгебраических уравнений. Если эти собственные значения одного знака и близки по модулю друг к другу, то потеря точности при их вычислении неизбежна вследствие неустойчивости процесса. Эта неустойчивость связана не только с характерной чувствительностью многочлена к малым возмущениям коэффициентов, но также и с возможной плохой обусловленностью уравнений, по которым определяются сами коэффициенты. Поэтому для одновременного вычисления нескольких близких собственных значений необходимо применять более устойчивые итерационные методы.