Лабораторная работа №9

«Степенной метод»

выполнил Пажитных Иван, 2-й курс, 1-я группа

1) Постановка задачи

Необходимо найти максимальное собственное значение и соответствующий ему собственный вектор матрицы А:

$$A - \lambda E_n = \begin{bmatrix} a_{11}-\lambda & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22}-\lambda & a_{23} & \cdots & \\ \vdots & & & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn}-\lambda \end{bmatrix}, \quad \det(A - \lambda E_n) = (-1)^n \, \mathrm{Pn}(\lambda)$$

С помощью построения итерационной последовательности λ^k

2)Алгоритм решения

Степенной метод является итерационным методом решения полной (теоретически, а на практике частичной) проблемы собственных значений.

Суть метода заключается в последовательном приближении y^k к собственному вектору соответствующему максимальному собственному значению λ. За λ^k берётся отношение соответствующий произвольных координат векторов y^{k+1} и y^k. Итерационный процесс останавливается, когда $|\lambda^{k+1} - \lambda^k| \le \varepsilon$.

Возьмём начальное приближение $y^0 = (1,1,\ldots,1)$, а последующее будем вычислять как: $y^{k+1} = Ay^k$. $\lambda^k \approx \frac{y_0^{k+1}}{y_0^k}$, за x можно принять $x \approx y^{k+1}$.

3) Листинг программы

```
eps = 10 ** (-15)
a = np.array(A)
At = a.transpose()                # находим At
a = np.dot(At, a)                 # перемножаем A на At, теперь A – симметрическая

yk = np.ones(5)                   # начальное приближение y^0
y = np.dot(a, yk)                 # y^1
l = y[0] / yk[0]                  # начальное λ^0
k = 1

while (True):                     # итерационный процесс
    yk = np.dot(a, y)
    lk = yk[0] / y[0]
    yk /= max(yk)                 # нормируем
    if abs(lk - l) <= eps:
        break
    y = yk
    l = lk
    k += 1

p = [4.58801522, -7.82119475, 6.11344651, -2.15665219, 0.2685558]       # P_n из Данилевского
r = np.dot(a, yk) - lk * yk                                            # находим вектор невязки
rnorm = np.linalg.norm(r, 1)                                           # находим норму невязки
p.insert(0, -1)                   # считаем невязку собственного многочлена P_n(λ^k)
r1 = sum(-(lk ** (n - i)) * p[i] for i in range(n + 1))
```

4) Результат и его анализ

Симметрическая AA^T:

[[0.70536135 0.01441237 0.13398766 -0.08030921 0.5676231]
 [0.01441237 1.22673234 -0.00165256 0.11340719 0.05855825]
 [0.13398766 -0.00165256 0.77976056 -0.21682262 0.2943685]
 [-0.08030921 0.11340719 -0.21682262 0.79926611 -0.0500214]
 [0.5676231   0.05855825 0.2943685 -0.0500214   1.07689486]]

Коэффициенты собственного многочлена $P(\lambda)$:

[4.58801522 -7.82119475 6.11344651 -2.15665219 0.2685558]

Максимальное собственное λ_{max}:

1.64297992228

Собственный вектор матрицы A — $x(\lambda_{max})$:

[0.7008202   0.09634149 0.51098681 -0.24436177 1.   ]

Количество итераций k:

105

Вектор невязки r:

[-4.44089210e-16 -2.77528001e-13   3.03090886e-14 -8.11573031e-14
 -1.33226763e-14]

Норма $||r||$:

4.02761157758e-13

Невязка $P_n(\lambda^k)$:

1.89213228419e-08

Эпсилон ε:

1e-15

С помощью степенного метода мы нашли максимальное по модулю собственное значение и соответствующий ему собственный с точностью порядка 10^{-13} за 105 итераций для эпсилона порядка 10^{-15}. Невязка собственного многочлена также довльно близка к нулю (порядка 10^{-8}) что означает, что собственное значение также найдено правильно. Собственное значение и собственный вектор также совпадают с получеными ранее методами Крылова и Данилевского. Чтобы СМ сходился необходимо и достаточно, чтобы у матрицы были доминирующее собственное значение.