Лабораторная работа №4
«Метод простой итерации»

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

1) Постановка задачи

Необходимо найти решение системы линейных алгебраических уравнений вида $Ax = b$, где А- квадратная матрица n-ого порядка, x и b – столбцы размеров n×1.

$$\begin{bmatrix} a_{11}\, a_{12}\, a_{13} & \cdots & a_{1n} \mid b_1 \\ a_{21}\, a_{22} a_{23} & \dots & \mid b_2 \\ \vdots & \ddots & \vdots \\ a_{n1}\, a_{n2} a_{n3} & \cdots & a_{nn} \mid b_n \end{bmatrix}$$

Предполагается, что det A=|A|≠0. Тогда решение системы существует и оно единственно. Метод простой итерации состоит в последовательном вычислении значения вектора x^k, до тех пор, пока не выполнится условие:

$$||x^{k+1} - x^k|| \le \varepsilon$$

2)Алгоритм решения

Приведём $Ax = b$ к каноническому виду $x = Bx + g$. B и g найдём по методу Якоби, как:

$$B = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{12}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{13}}{a_{22}} & \dots & \\ & \vdots & & \ddots & \vdots \\ -\frac{a_{12}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \cdots & 0 \end{bmatrix} \qquad g = \begin{bmatrix} \frac{b_{11}}{a_{11}} \\ \frac{b_{22}}{a_{22}} \\ \vdots \\ \frac{b_{nn}}{a_{nn}} \end{bmatrix}$$

Далее будем вычислять следующее значение x, как: $x^{k+1} = Bx^k + g$, а количество итераций можно оценить по формуле

$$||x^{k+1} - x^k|| \le \frac{||B||^{k+1}}{1-||B||}||g|| \le \varepsilon \quad \Rightarrow \quad k = log_{||B||}\left(\frac{\varepsilon*(1-||B||)}{||g||}\right) - 1$$

3) Листинг программы

```
E = np.identity(n)
eps = 10**(-5)
a = np.array(A)
b = np.array(f).transpose()

diagA = np.diag(a)                  # диагональные ел-ты A
g = b/diagA                         # находим g
B = np.array(a)-E*diagA             # находим B
for i in range(n):
    for j in range(m):
        B[i][j] /= (-diagA[i])

normB = np.linalg.norm(B, 1)                          # норма B
k = log(eps*(1-normB)/np.linalg.norm(g, 1), normB)-1  # прогноз k

xk = np.array(g)
x = zeros(n)
k = 0
```

```
while True:                                   # итерационный процесс
    x = np.dot(B, xk)+g
    if abs(np.linalg.norm(x, inf)-np.linalg.norm(xk, inf)) < eps:
        break
    xk = x
    k += 1
r = np.dot(A, x)-f                            # вектор невязки
```

4) Результат и его анализ

Матрица коэффициентов A:

```
[[ 0.6897 -0.0908  0.0182  0.0363  0.1271]
 [ 0.0944  1.0799  0.     -0.0726  0.0726]
 [ 0.0545  0.      0.8676 -0.2541  0.1452]
 [-0.1089  0.2287  0.      0.8531 -0.0363]
 [ 0.4538  0.      0.1634  0.0182  1.0164]]
```

Столбец свободных членов b:

```
[ 4.2108  4.6174 -5.877   2.7842  0.2178]
```

Вектор g:

```
[ 6.10526316  4.27576627 -6.77385892  3.26362677  0.21428571]
```

Матрица B:

```
[[-0.          0.13165144 -0.02638828 -0.05263158 -0.18428302]
 [-0.0874155  -0.         -0.          0.06722845 -0.06722845]
 [-0.06281697 -0.         -0.          0.2928769  -0.16735823]
 [ 0.12765209 -0.26808112 -0.         -0.          0.0425507 ]
 [-0.44647776 -0.         -0.16076348 -0.01790634 -0.        ]]
```

Норма $||B||$:

0.625147579693

Вектор решений x:

```
[ 7.00098128  3.99994541 -6.00023074  2.99988014 -2.00059167]
```

Расчётное кол-во итераций:

32.0398110688

Кол-во итераций:

10

Вектор невязки r:

```
[ -2.00837500e-06  -5.76837007e-07  -2.16691197e-06  -1.22386185e-07
  4.04518229e-06]
```

Норма $||r||$:

8.91969244665e-06

Эпсилон ε:

1e-05

Точность решения и скорость сходимости МПИ зависят от заданного ε (1e-05) Метод сходится, т.к. выполняется достаточное условие: норма полученной матрицы B меньше единицы. Для увеличения точности решения необходимо задать эпсилон.