

Лабораторная работа №1 «Численное решение СЛАУ»

ЦЕЛЬ РАБОТЫ

Изучить методы решения систем линейных алгебраических уравнений (СЛАУ)

[illegible]

Разработать соответствующие алгоритмы и процедуры.

ПРЯМЫЕ МЕТОДЫ

Метод Гаусса. СЛАУ (1.1) приводим последовательным исключением неизвестных к эквивалентной системе с верхнетреугольной матрицей:

$$\left\{ \begin{array}{l} a_{11}^{(0)} x_1 + a_{12}^{(0)} x_2 + \dots + a_{1n}^{(0)} x_n = b_1^{(0)}, \\ a_{22}^{(1)} x_2 + \dots + a_{2n}^{(1)} x_n = b_1^{(1)} \\ \dots \\ a_{nn}^{(n-1)} x_n = b_n^{(n-1)} \end{array} \right. \quad (1.2)$$

Формулы прямого хода:

$$a_{ij}^{(0)} = a_{ij}, \quad b_i^{(0)} = b_i, \quad i = 1, \dots, n, \quad j = 1, \dots, n \quad (1.3)$$

$$a_{ml}^{(k+1)} = a_{ml}^{(k)} - \frac{a_{mk}^{(k)}}{a_{kk}^{(k)}} a_{kl}^{(k)}, \quad b_m^{(k+1)} = b_m^{(k)} - \frac{a_{mk}^{(k)}}{a_{kk}^{(k)}} b_k^{(k)}, \quad m = k+1, \dots, n, l = k, \dots, n, k = 0, \dots, n-1$$

Если какой-либо диагональный элемент матрицы системы мал по величине, то используем схему метода Гаусса с выбором главного элемента (см. процедуру «Gauss1» модуля «matrlib»).

Обратный ход – из (1.2) рекуррентно находим решение:

$$x_n = b_n^{(n-1)} / a_{nn}^{(n-1)}, \quad x_i = \left(b_i^{(i-1)} - \sum_{j=i+1}^n a_{ij}^{(i-1)} x_j \right) / a_{ii}^{(i-1)}, \quad i = n-1, \dots, 1 \quad (1.4)$$

Например, решим методом Гаусса СЛАУ

$$\begin{cases} 5x_1 - 2x_2 = 3 \\ 2x_1 + 10x_2 = 12 \end{cases}$$

Прямой ход:

выбираем 1-ю строку расширенной матрицы ведущей (т.к. $a_{11} = 5 \neq 0$), умножаем ее на $-a_{21}/a_{11} = -2/5$ и результат складываем со 2-й строкой:

$$2x_1 + 10x_2 + (-2/5) \cdot (5x_1 - 2x_2) = 12 + (-2/5) \cdot 3,$$

$$(10 + 4/5)x_2 = 12 - 6/5,$$

$$54/5x_2 = 54/5, \text{ t.e. } a_{22}^{(1)} = 54/5, b_2^{(1)} = 54/5.$$

Обратный ход:

$$x_2 = \frac{(54/5)}{(54/5)} = 1,$$

$$x_1 = (3 + 2x_2)/5 = (3 + 2 \cdot 1)/5 = 1.$$

Метод прогонки. Метод прогонки предназначен для решения СЛАУ специального вида (с трехдиагональной матрицей)

$$c_j x_{j-1} + d_j x_j + e_j x_{j+1} = f_j, \quad j = 1, \dots, n \quad (1.5)$$

$$(c_1 = e_n = 0)$$

Прямой ход: рассчитываем прогоночные коэффициенты

$$\alpha_1 = -\frac{e_1}{d_1}, \quad \beta_1 = \frac{f_1}{d_1},$$

$$\alpha_j = -\frac{e_j}{d_j + c_j \alpha_{j-1}}, \quad \beta_j = \frac{f_j - c_j \beta_{j-1}}{d_j + c_j \alpha_{j-1}}, \quad j = 2, \dots, n \quad (1.6)$$

Обратный ход: находим вектор решения

$$x_n = \beta_n,$$

$$x_{j-1} = \alpha_{j-1} x_j + \beta_{j-1}, \quad j = n, n-1, \dots, 2. \quad (1.7)$$

Прогонка (1.6) - (1.7) корректна и устойчива, если матрица системы (1.5) имеет диагональное преобладание, т.е. $|d_j| \geq |c_j| + |e_j|$, $j = 1, \dots, n$.

Для рассматриваемого примера

$$c_1 = 0, d_1 = 5, e_1 = -2, f_1 = 3,$$

$$c_2 = 2, d_2 = 10, e_2 = 0, f_2 = 12.$$

Прямой ход:

$$\alpha_1 = -\frac{e_1}{d_1} = -\frac{(-2)}{5} = \frac{2}{5}, \quad \beta_1 = \frac{f_1}{d_1} = \frac{3}{5}, \quad \beta_2 = \frac{f_2 - c_2 \beta_1}{d_2 + c_2 \alpha_1} = \frac{12 - 2 \cdot 3/5}{10 + 2 \cdot 2/5} = 1.$$

Обратный ход:

$$x_2 = \beta_2 = 1,$$

$$x_1 = \alpha_1 x_2 + \beta_1 = 2/5 \cdot 1 + 3/5 = 1.$$

```
def Progon(n,c,d,e,f,x):
    """
    Метод прогонки для решения СЛАУ
        d(1)*x(1)+e(1)*x(2) =f(1), i=1
        c(i)*x(i-1)+d(i)*x(i)+e(i)*x(i+1)=f(i), 2<=i<=N-1
        c(N)*x(N-1)+d(N)*x(N)=f(N), i=N
    """
    alfa,beta = [0]*(n+1),[0]*(n+1)
    ...
```

ИТЕРАЦИОННЫЕ МЕТОДЫ

Метод простых итераций (МПИ). Тем или иным способом приведем (1.1) к эквивалентному виду

$$\begin{cases} x_1 = h_1 + g_{11}x_1 + g_{12}x_2 + \dots + g_{1n}x_n \\ x_2 = h_2 + g_{21}x_1 + g_{22}x_2 + \dots + g_{2n}x_n \\ \dots \\ x_n = h_n + g_{n1}x_1 + g_{n2}x_2 + \dots + g_{nn}x_n \end{cases} \quad (1.8)$$

или, в матричной форме, $\vec{x} = G\vec{x} + \vec{h}$. Для нахождения решения строим итерационную последовательность

$$\vec{x}^{(k+1)} = G\vec{x}^{(k)} + \vec{h}, \quad k=0,1,2,\dots \quad (1.9)$$

где k – номер итерации, $\vec{x}^{(0)}$ – произвольный вектор. Сходимость МПИ (1.9) обеспечена, если одна из канонических норм матрицы G меньше единицы, $\|G\| < 1$. В варианте *метода Якоби*, когда $a_{ii} \neq 0$ и $g_{ij} = -a_{ij} / a_{ii}$, $g_{ii} = 0$, $h_i = b_i / a_{ii}$, сходимость обеспечена, если у матрицы исходной системы (1.1) имеется диагональное преобладание:

$$|a_{ii}| > \sum_{j=1}^{i-1} |a_{ij}| + \sum_{l=i+1}^n |a_{il}|, \quad i=1,2,\dots,n. \quad (1.10)$$

Критерием окончания итерационного процесса может быть условие $\|\vec{x}^{(k+1)} - \vec{x}^{(k)}\| \leq \varepsilon_1$, где ε_1 – допустимая погрешность приближенного решения $\vec{x} \approx \vec{x}^{(k+1)}$, либо $\|A\vec{x}^{(k+1)} - \vec{b}\| \leq \varepsilon_2$, где ε_2 – допустимая невязка.

В рассматриваемом примере матрица системы имеет диагональное преобладание. Применение метода Якоби начинаем с преобразования СЛАУ:

$$\begin{cases} x_1 = (3 + 2x_2)/5 \\ x_2 = (12 - 2x_1)/10 \end{cases}$$

Тогда формула (1.9) приобретает вид:

$$\begin{cases} x_1^{(k+1)} = (3 + 2x_2^{(k)})/5 \\ x_2^{(k+1)} = (12 - 2x_1^{(k)})/10 \end{cases}$$

Пусть $\varepsilon = 0.1$, $x_1^{(0)} = 0$, $x_2^{(0)} = 0$. Тогда на 1-й итерации:

$$x_1^{(1)} = \frac{(3+2 \cdot 0)}{5} = \frac{3}{5}, \quad x_2^{(1)} = \frac{(12-2 \cdot 0)}{10} = \frac{6}{5},$$

$\Delta^{(1)} = \|\vec{x}^{(1)} - \vec{x}^{(0)}\| = \max\{|x_1^{(1)} - x_1^{(0)}|, |x_2^{(1)} - x_2^{(0)}|\} = 6/5 > \varepsilon$, т.е. сходимость не достигнута, переходим к следующей итерации.

На 2-й итерации: $x_1^{(2)} = 27/25$, $x_2^{(2)} = 54/50$, $\Delta^{(2)} = 12/25 > \varepsilon$.

На 3-й итерации: $x_1^{(3)} = 288/300$, $x_2^{(3)} = 246/250$, $\Delta^{(3)} = 36/300 > \varepsilon \dots$

```
def MPI(n, G, h, Nit, eps, x):
    """
    Метод простых итераций для решения СЛАУ x=G*x+h
    """
    for k in range(1, Nit + 1):
        xold = copy(x)
        ...
        dx = NRazn(x, xold, n)
        print(f' it={k} dx={dx} x={x[1:]}')
        if dx < eps: break
```

Метод Зейделя. Метод Зейделя представляет собой модификацию МПИ: при вычислении $(k+1)$ -го приближения неизвестной x_i учитываются уже вычисленные ранее $(k+1)$ -е приближения неизвестных $x_1, x_2, \dots, x_{i-1}$:

$$x_i^{(k+1)} = h_i + g_{i1}x_1^{(k+1)} + \dots + g_{i,i-1}x_{i-1}^{(k+1)} + g_{i,i}x_i^{(k)} + \dots + g_{in}x_n^{(k)}, \quad i=1, \dots, n, k=0, 1, 2, \dots \quad (1.11)$$

Итерационный процесс (1.11) сходится к решению со скоростью геометрической прогрессии, если выполняется условие (1.10).

Для рассматриваемого примера формула (1.11) запишется в виде:

$$\begin{cases} x_1^{(k+1)} = (3 + 2x_2^{(k)})/5 \\ x_2^{(k+1)} = (12 - 2x_1^{(k+1)})/10 \end{cases}$$

На 1-й итерации: $x_1^{(1)} = \frac{(3+2 \cdot 0)}{5} = \frac{3}{5}$, $x_2^{(1)} = \frac{(12-2 \cdot 3/5)}{10} = \frac{54}{50}$, $\Delta^{(1)} = \frac{3}{5} > \varepsilon$.

На 2-й итерации: $x_1^{(2)} = 258/250$, $x_2^{(2)} = 2484/2500$, $\Delta^{(2)} = 108/250 > \varepsilon$.

Как видно, скорость сходимости метода Зейделя выше таковой для МПИ.

ЗАДАНИЕ

С помощью системы Mathcad и модуля numpy.linalg найти число обусловленности исходной матрицы системы $cond(A) = \|A\| \cdot \|A^{-1}\|$ и сделать вывод об обусловленности СЛАУ. Проверить для A выполнение условия диагонального преобладания (1.10).

Написать функции на языке Питон, реализующие алгоритмы рассмотренных выше прямых и итерационных методов. В главной программе организовать ввод произвольной системы уравнений, для всех методов выводить на экран результат и невязку решения, для итерационных методов – также информацию о сходимости вычислительного процесса.

Решить исходную СЛАУ (см. таблицу) прямыми методами Гаусса и прогонки, итерационными методами МПИ и Зейделя при $\varepsilon_1 = 10^{-6}$. Сравнить результаты с данными, полученными в Mathcad и с помощью numpy.linalg для (1.1). Сравнить скорости сходимости итерационных методов.

СОДЕРЖАНИЕ ОТЧЕТА

1. Задание
2. Расчет $cond(A)$. Вывод об обусловленности СЛАУ. Проверка условия (1.10) для матрицы A .
3. Алгоритмы используемых численных методов.
4. Тексты функций и программ с результатами их запуска.
5. Решение (1.1) в Mathcad.
6. Сравнения точности методов (считать решение Mathcad эталонным), скорости сходимости итерационных методов. Выводы.

Вар.	СЛАУ	Вар.	СЛАУ
1	$\begin{cases} 3,21x_1 - 2,25x_2 = 5,06, \\ 7,09x_1 + 11,17x_2 - 2,23x_3 = 4,75, \\ 1,4x_2 - 2,62x_3 = -1,05; \end{cases}$	2	$\begin{cases} 0,20x_1 + 0,12x_2 = 0,10, \\ 0,12x_1 + 0,71x_2 + 0,15x_3 = 0,26, \\ 0,15x_2 + 0,63x_3 = 0,38; \end{cases}$

3	$\begin{cases} 1,42x_1 - 1,13x_2 = 6,15, \\ 1,14x_1 - 7,15x_2 + 5,11x_3 = -4,16, \\ 0,8x_2 - 1,02x_3 = -0,17; \end{cases}$	4	$\begin{cases} 0,71x_1 + 0,10x_2 = 0,29, \\ 0,10x_1 + 0,34x_2 - 0,04x_3 = 0,32, \\ 0,04x_2 + 0,1x_3 = -0,10; \end{cases}$
5	$\begin{cases} 2,5x_1 - 1,12x_2 = -7,5, \\ 0,61x_1 + 0,71x_2 - 0,05x_3 = 0,44, \\ 0,75x_2 + 0,877x_3 = -1,16; \end{cases}$	6	$\begin{cases} 0,34x_1 - 0,04x_2 = 0,33, \\ -0,04x_1 + 0,2x_2 + 0,12x_3 = -0,05, \\ 0,12x_2 + 0,71x_3 = 0,28; \end{cases}$
7	$\begin{cases} 7,09x_1 + 1,17x_2 = -4,75, \\ 0,43x_1 - 1,4x_2 - 0,62x_3 = -1,05, \\ 1,25x_2 + 2,13x_3 = 5,06; \end{cases}$	8	$\begin{cases} 0,52x_1 - 0,43x_2 = -0,17, \\ -0,07x_1 + 1,43x_2 + 0,72x_3 = 0,62, \\ 0,08x_2 - 0,25x_3 = 1,12; \end{cases}$
9	$\begin{cases} 11,4x_1 - 2,15x_2 = -4,16 \\ -0,71x_1 + 0,81x_2 - 0,02x_3 = -0,17 \\ 1,13x_2 + 7,05x_3 = 6,15 \end{cases}$	10	$\begin{cases} 1,17x_1 + 0,53x_2 = 1,15, \\ 0,64x_1 - 1,22x_2 - 0,43x_3 = 0,15, \\ 0,43x_2 - 0,93x_3 = -0,48; \end{cases}$
11	$\begin{cases} 0,61x_1 + 0,41x_2 = 0,44, \\ -1,03x_1 - 2,05x_2 + 0,87x_3 = -1,16, \\ 3,12x_2 - 5,03x_3 = -7,5; \end{cases}$	12	$\begin{cases} 1,66x_1 - 1,44x_2 = 1,83, \\ 0,48x_1 - 0,86x_2 + 0,37x_3 = -0,84, \\ 0,43x_2 + 0,64x_3 = 0,64; \end{cases}$
13	$\begin{cases} 3,11x_1 - 1,66x_2 = -0,92, \\ -1,65x_1 + 3,51x_2 - 0,78x_3 = 2,57, \\ 0,78x_2 - 1,87x_3 = 1,65; \end{cases}$	14	$\begin{cases} 0,82x_1 + 0,43x_2 = 0,48, \\ -0,35x_1 + 1,12x_2 - 0,48x_3 = 0,52, \\ 0,23x_2 + 0,37x_3 = 1,44; \end{cases}$
15	$\begin{cases} 5x_1 - 3,2x_2 = -1,4, \\ -1,2x_1 + 3,4x_2 - 0,9x_3 = 2,1, \\ 2,5x_2 + 4,3x_3 = 3,5; \end{cases}$	16	$\begin{cases} -0,5x_1 - 0,3x_2 = 1,3, \\ 0,9x_1 + 1,7x_2 - 0,2x_3 = 1,6, \\ -2,7x_2 + 3,7x_3 = -0,7; \end{cases}$
17	$\begin{cases} 4,2x_1 + 1,8x_2 = -1,6, \\ -1,9x_1 + 5,3x_2 + 2,1x_3 = -2,1, \\ -0,3x_2 - 0,9x_3 = 0,5; \end{cases}$	18	$\begin{cases} -1,7x_1 + 0,3x_2 = 0,8, \\ 2,4x_1 - 8,4x_2 - 3,2x_3 = -0,7, \\ 4,7x_2 + 11,3x_3 = -1,2; \end{cases}$
19	$\begin{cases} -5,22x_1 + 1,7x_2 = 3,55, \\ 0,95x_1 - 1,6x_2 + 0,32x_3 = -1,05, \\ -2,5x_2 + 4,76x_3 = 10,06; \end{cases}$	20	$\begin{cases} 0,91x_1 - 0,34x_2 = -1,23, \\ 1,48x_1 + 3,31x_2 - 0,81x_3 = 0,42, \\ -0,38x_2 + 0,66x_3 = 2,45; \end{cases}$
21	$\begin{cases} 4,77x_1 + 2,62x_2 = 3,53, \\ 0,65x_1 - 7,73x_2 + 4,16x_3 = -5,68, \\ 0,88x_2 - 3,19x_3 = -1,82; \end{cases}$	22	$\begin{cases} 1,23x_1 + x_2 = 0,58 \\ 2,3x_1 - 4,5x_2 + 1,1x_3 = -3,43 \\ x_2 - 3,89x_3 = 2,19 \end{cases}$
23	$\begin{cases} 3x_1 - 2x_2 = -4 \\ 2x_1 - 5x_2 + x_3 = 3 \\ 2x_2 + 6x_3 = -1 \end{cases}$		

УКАЗАНИЯ

1) Решение СЛАУ в Mathcad. Пусть $A := \begin{pmatrix} 5 & -2 \\ 2 & 10 \end{pmatrix}$ $b := \begin{pmatrix} 3 \\ 12 \end{pmatrix}$.

Тогда число обусловленности: $\text{cond}A := \text{norml}(A) \cdot \text{norml}(A^{-1})$ $\text{cond}A = 2.667 < 10^2$, т.е. система обусловлена.

Решение через обратную матрицу: $x := A^{-1} \cdot b$ $x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Решение через встроенную функцию: $x := \text{lsolve}(A, b)$ $x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

2) Пример Python-программы для решения СЛАУ:

```
import numpy as np
from copy import copy, deepcopy
from matrlib import
MatrPrint, VectPrint, Gauss1, Resid, NRazn
n = 2 # порядок СЛАУ
a = [[0]*(n+1) for i in range(n+1)] # матрица СЛАУ
a[1][1] = 5; a[1][2] = -2
a[2][1] = 2; a[2][2] = 10
b = [0, 3, 12] # вектор правых частей СЛАУ
# Решение: 1.0, 1.0 ; det(A) = 54
print("A:")
MatrPrint(a, n, n)
print("B:")
VectPrint(b, n)
# G и h для итерационных методов
G = [[0]*(n+1) for i in range(n+1)]
h = [0]*(n+1)
for i in range(1, n+1):
    for j in range(1, n+1):
        G[i][j] = -a[i][j]/a[i][i]
    G[i][i] = 0
    h[i] = b[i]/a[i][i]
print("G:")
MatrPrint(G, n, n)
print("h:")
VectPrint(h, n)
G1 = np.array(G)
normG = np.linalg.norm(G1[1:, 1:])
print("||G||=", normG)

# Метод Гаусса
x = [0]*(n+1)
a1 = deepcopy(a) # глубокие копии, чтобы
сохранить оригиналы
b1 = deepcopy(b)
print('*'*15+'Гаусс'+'*'*15)
x, det = Gauss1(a1, b1, n) # решаем СЛАУ
print("det(A) = ", det)
print("Решение:")
VectPrint(x, n)
Rmax1 = Resid(a, b, x, n)
print("Макс. невязка=", Rmax1)

# Метод прогонки
c, d, e, f, x = [0]*(n+1) for k in range(5)
c[1:] = [0.0, a[2][1]]
d[1:] = [a[1][1], a[2][2]]
e[1:] = [a[1][2], 0.0]
f[1:] = [b[1], b[2]]
Progon(n, c, d, e, f, x)
print('*'*15+'Прогонка'+'*'*15)
print("Решение:")
VectPrint(x, n)
Rmax = Resid(a, b, x, n)
print("Макс. невязка=", Rmax)

# МПИ
x = [0, 0, 0]
print('*'*15+'МПИ'+'*'*15)
MPI(n, G, h, 100, 1e-6, x)
print("Решение:")
VectPrint(x, n)
Rmax = Resid(a, b, x, n)
print("Макс. невязка=", Rmax)

# Метод Зейделя
x = [0, 0, 0]
print('*'*15+'Зейдель'+'*'*15)
Zeidel(n, G, h, 100, 1e-6, x)
print("Решение:")
VectPrint(x, n)
Rmax = Resid(a, b, x, n)
print("Макс. невязка=", Rmax)

# Numpy
a2 = np.array(a)
b2 = np.array(b)
print('*'*15+'Numpy'+'*'*15)
normA = np.linalg.norm(a2[1:, 1:])
ainv = np.linalg.inv(a2[1:, 1:])
normAinv = np.linalg.norm(ainv)
condA = normA * normAinv
print('cond(A)=', condA)
det = np.linalg.det(a2[1:, 1:])
print('det(A)=', det)
x = np.zeros(n+1)
x[1:] = np.linalg.solve(a2[1:, 1:], b2[1:])
print("Решение:")
VectPrint(x, n)
Rmax2 = np.max(np.dot(a2[1:, 1:], x[1:]) - b2[1:])
print("Макс. невязка=", Rmax2)
```

Результаты:

A:

5.00000e+00-2.00000e+00

2.00000e+00 1.00000e+01

B:

3.00000e+00 1.20000e+01

G:

0.00000e+00 4.00000e-01

-2.00000e-01 0.00000e+00

h:

6.00000e-01 1.20000e+00

$\|G\| = 0.447213595499958$

*****Гаусс*****

$\det(A) = 54.0$

Решение:

1.00000e+00 1.00000e+00

Макс. невязка= 0.0

*****Прогонка*****

Решение:

1.00000e+00 1.00000e+00

Макс. невязка= 0.0

*****МПИ*****

it=1 dx=1.2 x=[0.6, 1.2]

it=2 dx=0.4800000000000001 x=[1.08, 1.08]

it=3 dx=0.09600000000000009 x=[1.032, 0.984]

it=4 dx=0.03839999999999999 x=[0.9936, 0.9935999999999999]

it=5 dx=0.00768000000000002 x=[0.99744, 1.00128]

it=6 dx=0.0030720000000000747 x=[1.000512, 1.000512]

it=7 dx=0.000614400000000126 x=[1.0002048000000001, 0.9998975999999999]

it=8 dx=0.0002457600000000948 x=[0.99995904, 0.9999590399999999]

it=9 dx=4.9151999999996754e-05 x=[0.999983616, 1.000008192]

it=10 dx=1.966080000004311e-05 x=[1.0000032768, 1.0000032768]

it=11 dx=3.932160000053031e-06 x=[1.00000131072, 0.99999934464]

it=12 dx=1.5728640001544392e-06 x=[0.9999997378559999, 0.9999997378559999]

it=13 dx=3.1457280003088783e-07 x=[0.9999998951424, 1.0000000524288]

Решение:

1.00000e+00 1.00000e+00

Макс. невязка= 6.291456005058649e-07

*****Зейдель*****

it=1 dx=1.08 x=[0.6, 1.08]

it=2 dx=0.43200000000000005 x=[1.032, 0.9935999999999999]

it=3 dx=0.034560000000000035 x=[0.99744, 1.000512]

it=4 dx=0.0027648000000001227 x=[1.0002048000000001, 0.9999590399999999]

it=5 dx=0.00022118400000015193 x=[0.999983616, 1.0000032768]

it=6 dx=1.7694720000127617e-05 x=[1.00000131072, 0.9999997378559999]

it=7 dx=1.4155776001389953e-06 x=[0.9999998951424, 1.0000002097152]

it=8 dx=1.13246208077733e-07 x=[1.00000008388608, 0.999999983222784]

Решение:

1.00000e+00 1.00000e+00

Макс. невязка= 4.5298483408728885e-08*****Numpy*****

cond(A)= 2.462962962962963

$\det(A) = 54.00000000000001$

Решение:

1.00000e+00 1.00000e+00

Макс. невязка= 0.0