БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Лабораторная работа №1

**Метод Гаусса решения СЛАУ**

**Выполнилa:**

Осипчик Анна

2 курс 8 группа

**Преподаватель:**

Горбачева Ю.Н.

Минск, 2021

**Содержание**

Постановка задачи…………………………………………………………………....3

Алгоритм решения…………………………………………………………………....3

Листинг программы…………………………………………………………………..4

Результаты……………………………………………………………..........…………6

Вывод…………………………………………………………………………………..10

**Постановка задачи**

Методом Гаусса найти решение системы линейных алгебраических уравнений (СЛАУ);

найти обратную к ней матрицу и сделать проверку; вычислить вектор невязки.

**Алгоритм решения**

Первоначально имеем уравнение вида  Ax=f;

Решение этого уравнения найдем с помощью модификации метода Гаусса выбора

главного элемента в столбце. Для решения поставленной задачи заведем счетчик i от 0 до

n-1, где n - порядок матрицы-системы.

1. Выбираем ведущий элемент а11 (максимальный элемент i-ого столбца)

2. Поделив первое уравнение на а11 , получаем

x1+a(1)12x2+a(1)13x3+a(1)14x4=g1

3. Исключаем переменную x1 из всех последующих уравнений, начиная со второго, путём вычитания уравнения 2, умноженного на коэффициент, стоящий при x1 в соответствующем уравнении. Получаем:

a(1)11x1+a(1)12x2+a(1)13x3+a(1)1nxn=f(1)1

0 a(1)22x2+a(1)23x3+a(1)2nxn=f(1)2

0 a(1)32x2+a(1)33x3+a(1)3nxn=f(1)3

0 a(1)n2x2+a(1)n3x3+a(1)nnxn=f(1)4

Где а(1)i j= a(0)i j- a(0)1j \* a(0)i1 ,i-1=1,…,9, j=1,…,9

4. Выбираем ведущий элемент во втором уравнении a(1)22 ,не равный нулю

и так далее.

В общем случае, если ни на каком шаге не встретится 0, матрица А примет треугольный вид.

Из системы 3 отыскиваем неизвестные следующим образом

xn=f(n)n

xn-1=f(n-1)n-1-a(n-1)n-1nxn

xi=f(i) - Σ a(i)ijxj, j = i+1, n

Процесс приведения матрицы к треугольному виду называется прямым ходом метода Гаусса, а нахождение корней по формулам - обратным ходом метода Гаусса.

**Листинг программы**

from random import randint  
import numpy as np  
from copy import deepcopy  
  
  
def gen(dim: int):  
 return list(range(1, dim + 1))  
  
  
def generation(dim):  
 matrix = [[randint(1, 10) for \_ in range(dim)] for \_ in range(dim)]  
 f = list()  
 x = gen(dim)  
 for equation in matrix: # генерация вектора свободных членов  
 el = sum(a \* b for a, b in zip(equation, x))  
 f.append(el)  
 return matrix, f  
  
  
def find\_max\_element(l\_1: list, f: list, idx: int):  
 list\_of\_column = [column[idx] for column in l\_1] # составление столбца  
 max\_el = max(list\_of\_column, key=abs) # поиск ведущего элемента столбца  
 index = list\_of\_column.index(max\_el)  
 l\_1[index], l\_1[idx] = l\_1[idx], l\_1[index]  
 f[index], f[idx] = f[idx], f[index] # замена строчек местами  
  
  
def gauss(matrix: list, f: list):  
  
 idx\_of\_main\_element = 0  
  
 for i in range(len(matrix) - 1):  
  
 find\_max\_element(matrix, f, idx\_of\_main\_element)  
 el = matrix[i][idx\_of\_main\_element] # ведущий элемент  
 matrix[i] = list(map(lambda x: x / el, matrix[i])) # деление строчки на ведущий элемент  
 f[i] /= el  
  
 for j in range(i + 1, len(matrix)):  
 first\_el = matrix[j][idx\_of\_main\_element]  
 matrix[j] = list(map(lambda x1, x2: x2 - x1 \* first\_el, matrix[i], matrix[j]))  
 # зануление всего столбца!!!  
 f[j] -= f[i] \* first\_el  
  
 idx\_of\_main\_element += 1  
  
  
 matrix[-1][-1], f[-1] = 1, f[-1] / matrix[-1][-1]  
  
 answer = [0]  
 counter = True  
  
 for row, y in zip(reversed(matrix), reversed(f)): # нахождение вектора неизвестных  
 x = y - sum(a \* b for a, b in zip(reversed(row), answer))  
 if counter:  
 answer.pop(0)  
 counter = False  
 answer.append(x)  
 answer.reverse()  
 return answer  
  
  
def inaccuracy(x: list, answer: list):  
 answ = max([a - b for a, b in zip(x, answer)]) / max(x)  
 return answ  
  
  
def steps(matrix, e\_matrix, idx\_of\_main\_element=0):  
  
 for i in range(len(matrix) - 1):  
  
 find\_max\_element(matrix, e\_matrix, idx\_of\_main\_element)  
 el = matrix[i][idx\_of\_main\_element] # ведущий элемент  
 matrix[i] = list(map(lambda x: x / el, matrix[i])) # деление строчки на ведущий элемент  
 e\_matrix[i] = list(map(lambda x: x / el, e\_matrix[i]))  
 # у меня делится вся строчка!!!  
  
 for j in range(i + 1, len(matrix)):  
 first\_el = matrix[j][idx\_of\_main\_element]  
 matrix[j] = list(map(lambda x1, x2: x2 - x1 \* first\_el, matrix[i], matrix[j]))  
 # зануление всего столбца!!!  
 e\_matrix[j] = list(map(lambda x1, x2: x2 - x1 \* first\_el, e\_matrix[i], e\_matrix[j]))  
 # зануление всего столбца!!!  
  
 idx\_of\_main\_element += 1  
  
  
def inverse\_matrix(matrix: list, dim: int):  
  
 e\_matrix = [[1 if i == j else 0 for i in range(dim)] for j in range(dim)]  
 copy\_of\_matrix = deepcopy(matrix)  
  
 steps(matrix, e\_matrix)  
  
 matrix[-1][-1], e\_matrix[-1] = 1, list(map(lambda x: x / matrix[-1][-1], e\_matrix[-1]))  
  
 idx\_of\_main\_element = len(matrix) - 1  
  
 for i in range(len(matrix) - 1, 0, -1):  
  
 el = matrix[i][idx\_of\_main\_element] # ведущий элемент  
 matrix[i] = list(map(lambda x: x / el, matrix[i])) # деление строчки на ведущий элемент  
 e\_matrix[i] = list(map(lambda x: x / el, e\_matrix[i]))  
 # у меня делится вся строчка!!!  
  
 for j in range(i - 1, -1, -1):  
 first\_el = matrix[j][idx\_of\_main\_element]  
 matrix[j] = list(map(lambda x1, x2: x2 - x1 \* first\_el, matrix[i], matrix[j]))  
 # зануление всего столбца!!!  
 e\_matrix[j] = list(map(lambda x1, x2: x2 - x1 \* first\_el, e\_matrix[i], e\_matrix[j]))  
 # зануление всего столбца!!!  
  
 idx\_of\_main\_element -= 1  
  
 A = np.array(copy\_of\_matrix)  
 B = np.array(e\_matrix)  
 C = A.dot(B)  
 return C, B  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 dim = 10  
 matrix, f = generation(dim)  
 print('\nПорядок матрицы:', dim)  
 print('\nСгенерированная матрица:')  
  
 for el in matrix:  
 print(el)  
  
 e, inverse = inverse\_matrix(matrix[:], dim)  
 answ = gauss(matrix, f)  
 print('\nПервоначальный столбец неизвестных:', gen(dim))  
 print('\nПодсчитанный столбец неизвестных:', answ)  
 print("\nОтносительная погрешность: ", inaccuracy(gen(dim), answ))  
 print("\nОбратная матрица:\n", inverse)  
 print("\nРезультат перемножения матрицы и обратной к ней:\n", e)

from main import generation,gauss,inaccuracy, gen  
  
with open("Results\_of\_gauss.txt", "w", encoding="utf-8") as file1:  
 print("Порядок матрицы Относительная погрешность", file=file1)  
 for k in range(3, 104):  
 matrix, f = generation(k)  
 normal\_answer = gen(k)  
 answ = gauss(matrix, f)  
 print(" "\*7, k, " "\*15, inaccuracy(normal\_answer, answ), file=file1)

**Результаты**

#Текстовый файл с погрешностями вычислений решений матриц разных порядков (3 пункт)

Порядок матрицы Относительная погрешность  
 3 7.401486830834377e-17  
 4 1.1102230246251565e-16  
 5 3.552713678800501e-16  
 6 2.9605947323337506e-16  
 7 7.61295788314393e-16  
 8 1.7763568394002505e-15  
 9 1.1842378929335002e-15  
 10 1.0658141036401502e-15  
 11 6.459479416000911e-16  
 12 1.2582527612418441e-15  
 13 4.099285014000578e-15  
 14 5.075305255429287e-16  
 15 4.736951571734001e-16  
 16 3.552713678800501e-15  
 17 1.9853399969767506e-15  
 18 3.157967714489334e-15  
 19 2.6177890264845795e-15  
 20 8.437694987151189e-16  
 21 1.8271098919545435e-14  
 22 1.453382868600205e-15  
 23 4.4795085515310666e-15  
 24 4.440892098500626e-15  
 25 1.3500311979441904e-15  
 26 2.596213842200366e-15  
 27 3.684295666904223e-15  
 28 1.7763568394002505e-15  
 29 5.390324202318001e-15  
 30 1.1605531350748303e-14  
 31 1.7706266560473463e-14  
 32 7.105427357601002e-15  
 33 8.827955201867911e-15  
 34 2.507797890918001e-15  
 35 3.045183153257572e-15  
 36 1.5493779099213294e-14  
 37 3.696742611724846e-15  
 38 2.0942312211876636e-14  
 39 3.042580432613762e-14  
 40 3.4638958368304883e-15  
 41 5.545699401054441e-15  
 42 6.767073673905716e-15  
 43 2.1812009562868193e-14  
 44 1.8248029350202574e-14  
 45 4.421154800285068e-15  
 46 9.885811975792698e-15  
 47 3.3259447205791924e-15  
 48 3.4046839421838135e-15  
 49 1.0629139292084764e-13  
 50 2.1515234038815834e-13  
 51 1.1563734719233003e-14  
 52 1.0931426704001542e-14  
 53 6.435104021978266e-15  
 54 1.5263510620031783e-14  
 55 6.201100239360874e-15  
 56 2.20775778611174e-14  
 57 3.3657287483373164e-14  
 58 2.958552942863176e-14  
 59 1.5504283356229712e-12  
 60 1.7266188478970435e-13  
 61 6.75598011050587e-15  
 62 9.741311699936857e-15  
 63 2.0526790144180672e-14  
 64 5.972555783273492e-12  
 65 7.597341559281071e-15  
 66 4.736951571734001e-15  
 67 4.666250802006628e-15  
 68 6.687461042448002e-15  
 69 8.650085478818612e-15  
 70 1.3474935453164757e-14  
 71 2.0255471791245674e-13  
 72 2.3586071367592216e-14  
 73 1.2458831531136003e-14  
 74 1.2674546097342328e-14  
 75 2.1789977229976407e-14  
 76 4.547473508864641e-13  
 77 1.919388169325985e-14  
 78 2.7328566760003854e-14  
 79 4.3172216856309886e-15  
 80 2.2595258997171186e-13  
 81 2.5790069668329563e-14  
 82 3.570043989428796e-14  
 83 6.574660494744059e-14  
 84 1.6240976817373718e-14  
 85 7.523393672754002e-14  
 86 1.8176674635723494e-14  
 87 1.3720825242264004e-14  
 88 2.115479508740298e-14  
 89 6.818016812799164e-14  
 90 6.552783007565369e-15  
 91 1.9988894544459963e-14  
 92 1.5909978648541375e-14  
 93 1.2224391152861939e-14  
 94 3.114293692905971e-14  
 95 1.9147256879430068e-14  
 96 1.569115208136888e-14  
 97 3.0619264283270295e-14  
 98 1.1890715169862902e-14  
 99 1.3206046806046306e-14  
 100 4.82458517581108e-14  
 101 1.0172720751575296e-13  
 102 4.307839488176921e-13  
 103 1.3796946325438839e-14

#Результаты 1 и 2 пункта

Порядок матрицы: 10

Сгенерированная матрица:

[10, 6, 2, 4, 8, 7, 6, 9, 6, 5]

[5, 4, 2, 2, 9, 7, 1, 9, 2, 2]

[6, 8, 2, 10, 4, 3, 7, 4, 3, 3]

[9, 7, 2, 10, 4, 7, 2, 6, 3, 5]

[2, 10, 6, 1, 8, 10, 6, 8, 9, 7]

[7, 3, 5, 3, 5, 7, 1, 8, 5, 10]

[3, 2, 1, 5, 1, 4, 5, 7, 5, 8]

[8, 6, 8, 6, 9, 3, 2, 8, 2, 10]

[7, 7, 8, 3, 3, 9, 4, 6, 3, 3]

[4, 1, 8, 2, 2, 8, 8, 5, 4, 7]

Первоначальный столбец неизвестных: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Подсчитанный столбец неизвестных: [1.0, 1.9999999999999645, 3.000000000000025, 4.000000000000028, 5.000000000000007, 5.999999999999989, 6.999999999999983, 8.0, 9.000000000000046, 9.999999999999972]

Относительная погрешность: 3.552713678800501e-15

Обратная матрица:

[[ 1.88499746e-01 -1.25159214e-01 -1.86587291e-01 1.28295333e-01

-2.19195902e-02 -1.55516924e-01 3.87907456e-04 7.94621247e-02

2.32670801e-02 9.59793384e-03]

[ 3.93341931e-01 -3.71612749e-01 -1.18313971e+00 9.12903275e-01

3.13096901e-01 -1.46088373e+00 4.27049446e-01 6.14929958e-01

-2.71412777e-02 9.91866049e-02]

[-4.96916611e-01 3.52939579e-01 1.36269036e+00 -1.11283783e+00

-3.11761759e-01 1.61476355e+00 -4.53923667e-01 -6.15561643e-01

2.22866686e-01 -2.27439359e-01]

[-4.58221744e-01 3.44407345e-01 1.17005427e+00 -7.97206029e-01

-2.56527438e-01 1.33176779e+00 -3.66197609e-01 -5.57836310e-01

5.70333260e-02 -1.58140064e-01]

[-4.58723482e-02 1.04300412e-01 1.97811877e-01 -9.73851786e-02

3.08389022e-03 2.11883769e-01 -1.97225606e-01 -5.94368303e-02

-1.44636734e-01 5.42728015e-02]

[ 2.03750579e-01 -1.52856810e-01 -8.13144477e-01 7.38768972e-01

2.03710941e-01 -9.09535692e-01 1.77578084e-01 3.04049820e-01

-1.43426767e-01 2.38726723e-01]

[ 3.23027539e-01 -2.18966928e-01 -7.70007715e-01 5.86512096e-01

1.67100121e-01 -1.04691425e+00 2.85287516e-01 4.01654115e-01

-1.05019778e-01 2.16559792e-01]

[-1.60251782e-01 2.05650444e-01 3.92170461e-01 -4.44634987e-01

-1.53993207e-01 4.28390779e-01 8.99943569e-02 -1.77394939e-01

2.06244670e-01 -1.90581891e-01]

[-7.24117900e-01 5.53901756e-01 2.37868173e+00 -1.87374365e+00

-4.74301520e-01 2.91339650e+00 -7.97584943e-01 -1.21607812e+00

2.23894546e-01 -4.56945472e-01]

[ 4.74240818e-01 -4.45421698e-01 -1.50258971e+00 1.19738592e+00

3.53749464e-01 -1.72377150e+00 4.81011126e-01 7.61702379e-01

-2.30411841e-01 2.89724326e-01]]

Результат перемножения матрицы и обратной к ней:

[[ 1.00000000e+00 3.88578059e-16 -4.44089210e-16 1.11022302e-15

-3.33066907e-16 2.22044605e-16 2.22044605e-16 6.66133815e-16

-2.22044605e-16 1.11022302e-16]

[ 7.77156117e-16 1.00000000e+00 -2.66453526e-15 8.88178420e-16

0.00000000e+00 -2.22044605e-15 4.44089210e-16 1.33226763e-15

-5.55111512e-17 5.55111512e-16]

[ 5.55111512e-17 4.99600361e-16 1.00000000e+00 -2.88657986e-15

-1.11022302e-16 3.33066907e-15 0.00000000e+00 -1.11022302e-15

1.11022302e-16 -3.33066907e-16]

[ 8.32667268e-16 -4.99600361e-16 4.44089210e-16 1.00000000e+00

5.55111512e-16 2.22044605e-16 6.66133815e-16 1.55431223e-15

-2.22044605e-16 -1.11022302e-16]

[ 2.77555756e-16 7.21644966e-16 1.33226763e-15 -1.99840144e-15

1.00000000e+00 -1.11022302e-15 -6.66133815e-16 -2.88657986e-15

4.44089210e-16 -5.55111512e-16]

[-1.11022302e-16 7.77156117e-16 2.66453526e-15 -1.33226763e-15

-6.66133815e-16 1.00000000e+00 -4.44089210e-16 -4.44089210e-16

0.00000000e+00 -2.22044605e-16]

[ 4.44089210e-16 0.00000000e+00 -1.77635684e-15 0.00000000e+00

0.00000000e+00 1.77635684e-15 1.00000000e+00 0.00000000e+00

-4.44089210e-16 0.00000000e+00]

[-9.99200722e-16 7.77156117e-16 8.88178420e-16 -1.33226763e-15

-6.66133815e-16 4.44089210e-16 -4.44089210e-16 1.00000000e+00

0.00000000e+00 2.22044605e-16]

[-1.27675648e-15 9.43689571e-16 1.33226763e-15 6.66133815e-16

-3.33066907e-16 2.44249065e-15 2.22044605e-16 -1.11022302e-15

1.00000000e+00 -2.22044605e-16]

[-1.66533454e-16 7.21644966e-16 1.33226763e-15 1.55431223e-15

-1.11022302e-16 6.66133815e-16 -2.22044605e-16 -2.22044605e-16

0.00000000e+00 1.00000000e+00]]

Process finished with exit code 0

**Выводы**

В ходе выполнения лабораторной работы я на практике реализовала и поверила работу

алгоритма Гаусса, а вернее его модификации с выбором главного элемента по столбцу.

Все результаты представлены выше. Анализируя их, можно выявить, что данный способ

решения СЛАУ не гарантирует точных вычислений. Это видно по относительной погрешнос

ти, посчитанной в пункте 2. Также я выявила закономерность в уменьшении точности вычис

лений в зависимости от порядка рассматриваеой матрицы-системы: относительная погрешно

сть при вычислении матриц сотых порядков ухудшилась в 1000 раз, что говорит о том,

что с увеличением порядка матрицы точность вычислений значительно снижается.