Федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

Лабораторная работа №1

Вариант: Метод Гаусса с выбором главного элемента

Выполнил:
Курносова Ирина
P3231

Преподаватель:
Перл Ольга Вячеславовна

Санкт-Петербург, 2022

Описание метода

Решение СЛАУ методом Гаусса с выбором главного элемента состоит из двух этапов: прямого и обратного хода. Прямой ход заключается в приведении матрицы к треугольному виду. Для начала выбирается максимальный по модулю элемент (по строке, по столбцу или во всей матрице) и происходит перестановка строк/столбцов так, чтобы этот максимальный по модулю элемент оказался на главной диагонали. Затем путём элементарных преобразований обнуляются ненулевые элементы матрицы, лежащие под главной диагональю: для каждой строки вычисляется соответственный коэффициент, затем из текущей строки вычитается вышележащая строка, умноженная на найденный коэффициент.

После приведения матрицы к треугольному виду начинается обратный ход: выражается решение системы в численном виде. Эта процедура начинается с последнего уравнения (в котором осталась только одна неизвестная), и путём подстановки найденных неизвестный в вышележащие уравнения находятся остальные неизвестные.

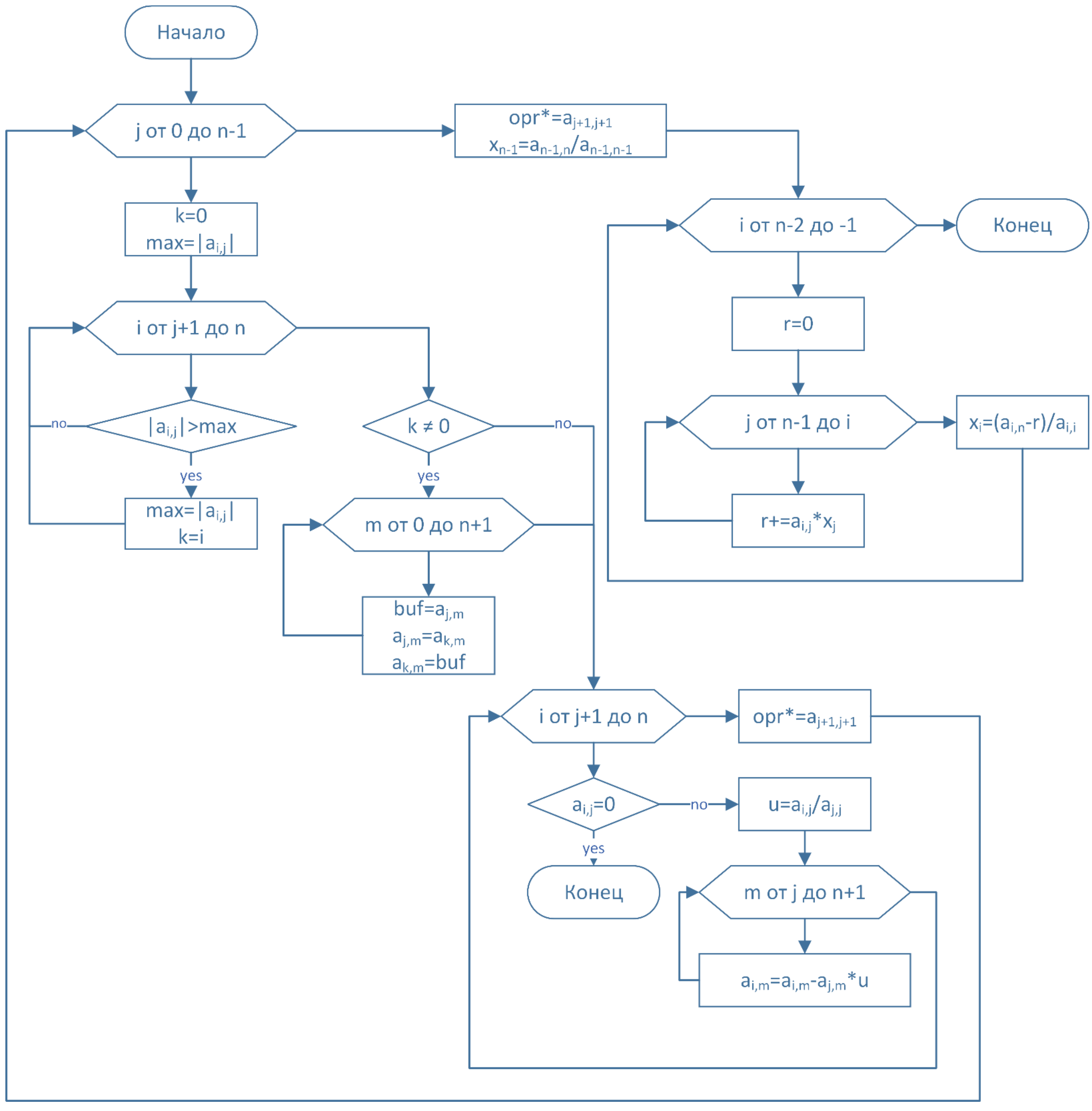
Расчётные формулы метода

Преобразованная система:

$$\begin{cases} a_{1j_1}x_{j_1} + a_{1j_2}x_{j_2} + \ldots + a_{1j_r}x_{j_r} + \ldots + a_{1j_n}x_{j_n} = b_1 \\ a_{2j_2}x_{j_2} + \ldots + a_{2j_r}x_{j_r} + \ldots + a_{2j_n}x_{j_n} = b_2 \\ \ldots \\ a_{rj_r}x_{j_r} + \ldots + a_{rj_n}x_{j_n} = b_r \\ 0 = b_{r+1} \\ \ldots \\ 0 = b_m \end{cases}$$

$$a_{1j_1}, \ldots, a_{1j_1} \neq 0$$

Блок-схема метода



Функция приведения матрицы к треугольному виду

```
for j in range(n-1):
    k=0
    max = abs(a[j][j])
    for i in range(j+1, n, 1):     # Поиск максимального элемента столбца
        if abs(a[i][j])>max:
            max=abs(a[i][j])
            k=i                    # Номер строки, содержащей
                                   # этот максимальный элемент
    if k!=0:
        for m in range(n+1):       # Меняем строки местами,
            buf=a[j][m]            # чтобы максимальный элемент оказался
            a[j][m]=a[k][m]        # на главной диагонали
            a[k][m]=buf

    for i in range(j+1, n, 1):     # Обнуление чисел под гл. диагональю
        if a[j][j]==0:
            print("Матрицу нельзя привести к треугольному виду")
            exit(0)
        u=a[i][j]/a[j][j]
        for m in range(j, n+1, 1):
            a[i][m]=a[i][m]-a[j][m]*u

    opr*=a[j][j]                   # Счётчик определителя
```

Функция для поиска неизвестных

```
x[n-1] = a[n-1][n]/a[n-1][n-1]
for i in range(n-2, -1, -1):
    r=0
    for j in range(n-1, i, -1):
        r+=a[i][j]*x[j]
    x[i]=(a[i][n]-r) / a[i][i]
```

Функция расчёта и вывода невязки

```
for i in range(n):
    r=0
    for j in range(n):
        r+=b[i][j]*x[j]
    b[i][n] = abs(abs(b[i][n]) - abs(r))
    print(i+1, ") ", round(b[i][n], 20), sep='')
```

Примеры

```
Размер матрицы (max - 20): 3

Расширенная матрица:
   -4.728145  -11.334673   78.486841 |   31.799472
   73.585908   63.243633   25.338378 |   68.819848
  -27.480851   48.015575  -10.939891 |  -76.747564

Определитель матрицы:  421516.0703

Треугольная матрица:
   73.585908   63.243633   25.338378 |   68.819848
    0.000000   71.634076   -1.477207 |  -51.046611
    0.000000    0.000000   79.964977 |   31.040011

Результат:
X(1) = 1.4071
X(2) = -0.7046
X(3) = 0.3882

Невязка:
1) 3.55271e-15
2) 1.421085e-14
3) 0.0

Время выполнения:  0:00:00.000965
```

```
Размер матрицы (max - 20): 5

Расширенная матрица:
   81.213668  -87.188200    4.890068   90.905270  -29.340595 |  -18.857629
  -41.951992   17.559462  -19.978719  -36.457011  -86.559061 |   87.701233
   48.355574   68.685587   97.209468  -47.924591  -74.018808 |   60.648897
   29.137235  -66.669774  -24.522286  -35.358160  -30.693623 |   64.986534
  -69.602939  -36.686041  -15.803519   30.900620   78.678808 |   79.045439

Определитель матрицы:  7957733274.1049

Треугольная матрица:
   81.213668  -87.188200    4.890068   90.905270  -29.340595 |  -18.857629
    0.000000  120.598468   94.297864 -102.050659  -56.549072 |   71.876951
    0.000000    0.000000   75.500195   14.534780    1.292631 |  129.284019
    0.000000    0.000000    0.000000  -98.187220  -36.784952 |   90.456104
    0.000000    0.000000    0.000000    0.000000 -109.601229 |   74.968245

Результат:
X(1) = -1.7098
X(2) = -1.7356
X(3) = 1.8521
X(4) = -0.665
X(5) = -0.684

Невязка:
1) 3.55271e-15
2) 0.0
3) 7.10543e-15
4) 0.0
5) 4.263256e-14

Время выполнения:  0:00:00.000999
```

```
Размер матрицы (max - 20): 7

Расширенная матрица:
  -68.077226   54.789082   59.819303   37.053205   18.385881  -54.536218   84.954912 |  -91.202699
  -99.105458  -39.091192  -76.969725   55.543287  -10.760068  -45.414636  -71.708981 |   92.544693
   80.152089    5.538938  -12.050151   65.161527  -55.713056  -37.920962  -27.002529 |  -41.259423
   93.652133   54.271781  -45.904216   83.961225  -68.161065   82.180926  -48.456765 |   83.244998
  -90.449722  -29.465842   40.226073   66.445715   86.583807  -42.346606   36.192167 |  -59.690942
   19.309553   17.524263   19.839796   64.343131  -80.561414  -13.946340    0.370480 |  -82.091092
   -7.152287   33.169311   82.908094   91.170920  -78.839410   32.346466  -46.573402 |    5.067882

Определитель матрицы:  -115078718780305.08

Треугольная матрица:
  -99.105458  -39.091192  -76.969725   55.543287  -10.760068  -45.414636  -71.708981 |   92.544693
    0.000000   81.641487  112.691117   -1.100424   25.777155  -23.340132  134.213032 | -154.773224
    0.000000    0.000000 -142.561745  136.681824  -83.801265   44.220108 -144.711893 |  203.554049
    0.000000    0.000000    0.000000  124.832532 -112.224912   57.943430 -139.933737 |  121.996502
    0.000000    0.000000    0.000000    0.000000  136.611540  -20.214348  115.259383 |  -97.837051
    0.000000    0.000000    0.000000    0.000000    0.000000 -123.143473   51.619864 | -118.995331
    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000   47.506931 |  -83.045551

Результат:
X(1) = -0.29
X(2) = 1.354
X(3) = -0.4092
X(4) = -0.3775
X(5) = 0.7932
X(6) = 0.2335
X(7) = -1.7481

Невязка:
1) 0.0
2) 1.421085e-14
3) 1.421085e-14
4) 1.421085e-14
5) 2.842171e-14
6) 1.421085e-14
7) 3.996803e-14

Время выполнения:  0:00:00.003999
```

Вывод

Прямые методы используют конечные формулы для вычисления неизвестных. Они дают достаточно точное решение за конечное число арифметических операций. Эти методы сравнительно просты и более универсальны.

К недостаткам прямых методов можно отнести следующие моменты: при вычислениях могут образовываться достаточно большие погрешности. Это происходит ввиду ограниченности разрядной сетки компьютера, и при обратном ходе погрешность на каждом шаге увеличивается. Также такие методы (в отличие от итерационных) подразумевают хранение всей матрицы в памяти компьютера, что очень неприятно может влиять на оперативную память и производительность компьютера.

Однако если сравнивать две вариации метода Гаусса (просто метод Гаусса и метод Гаусса с выбором главного элемента), то мой метод является более точным, т.к. путём выбора наибольшего элемента главной диагонали мы пытаемся избегать слишком маленьких коэффициентов (близким к 0), за счёт чего в методе Гаусса и появляется бОльшая погрешность.