МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ФАКУЛЬТЕТ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Дисциплина: «Вычислительная математика»

Схема Халецкого

Выполнил:

студент группы N3247

Василев Васил Николаев

Проверил:

Гришенцев Алексей Юрьевич

Санкт-Петербург, 2022

1. Техническое задание

Разработать алгоритм и написать программу реализующую: решение системы линейных уравнений, на множестве вещественных чисел, методом схема Халецкого. Предусмотреть возможность решения уравнений различного порядка без перекомпиляции программы. Исходные данные (матрицу коэффициентов уравнения, вектор правых частей), вводить в программу из консоли или из файла. Оценить вычислительную сложность решения задачи.

2. Теория

Схема Халецкого или LU-разложение — это представление матрицы A в виде A=L•U, где L — нижнетреугольная матрица с единичной диагональю, а U — верхнетреугольная матрица. LU-разложение является модификации её метода Гаусса. Основные применения данного алгоритма — решение систем алгебраических уравнений, вычисление определителя, вычисление обратной матрицы и др.

2. Теория

$$Ax = B$$

где A = [a_{ij}] – квадратная матрица порядка n и

$$x = \begin{bmatrix} x_1 \\ . \\ . \\ . \\ x_n \end{bmatrix}, \qquad B = \begin{bmatrix} b_1 \\ . \\ b_n \end{bmatrix}$$

- векторы-столбцы. Предтавим матрицу A в виде произведения нижней треугольной матрицы L = [l_{ij}] и верхней треугольной матрицы U = [u_{ij}] с единичной диагональю , т.е.

$$A = LU,$$

2. Теория

где

$$L = \begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix} \quad \text{и} \quad U = \begin{bmatrix} u_{11} & 0 & \dots & 0 \\ u_{21} & u_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ u_{n1} & u_{n2} & \dots & u_{nn} \end{bmatrix}$$

Тогда элементы l_{ij} и u_{ij} определяются по формулам

$$l_{i1} = b_{i1}, \qquad l_{ij} = b_{ij} - \sum_{k=1}^{j-1} l_{\mathrm{ik}} u_{kj}$$

и

$$u_{1j} = \frac{b_{1j}}{l_{11}}, \qquad u_{ij} = \frac{1}{l_{ii}}\left(a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}\right)(1 < i < j) \cdot$$

Отсюда искомый вектор x может быть вычислен из цепи уравнений

Ly=B, Ux=y (5)

2. Теория

Так как матрицы B и C – треугольные, то системы (5) легко решаются, а именно:

$$y_1 = \frac{b_1}{l_{11}}, \qquad y_i = \frac{1}{l_{ii}}\left(b_1 - \sum_{k=1}^{i-1} l_{ik} y_k\right) \quad (i > 1) \qquad (6)$$

и

$$x_n = y_n, \qquad x_i = y_i - \sum_{k=i+1}^{n} n_{ik} x_k \quad (i < n)\cdot$$

Из формул (6) видно, что числа y_i выгодно вычислять вместе с коэффициентами u_{ij}. Этот метод получил название схемы Халецкого. В схеме применяется обычный контроль с помощью сумм.

Заметим, что если матрица A – симметрическая, т.е. $a_{ij} = a_{ji}$, то

$$u_{ij} = \frac{l_{ji}}{l_{ii}} \quad (i < j).$$

Схема Халецкого удобна для работы на вычислительных машинах, так как в этом случае операции «накопления» (3) и (4) можно проводить без записи промежуточных результатов.

Сложность алгоритма равна: $T(n) = O(n^3) + O(n^2)$

$O(n^3)$ – треугольное разложение матрицы СЛАУ

$O(n^2)$ – решение двух треугольных систем

3. Листинг

```
#include<iostream>
using namespace std;
int main() {
int n = 0, i = 0, j = 0, k = 0, z, q, v, ranka, rankab, var;
cout << "Enter size of square matrix : "<<endl;
cin >> n;
cout << "Enter number of variables : "<<endl;
cin >> var;
float a[n][n],a2[n][n], b[n], /
ab[n][n+1], l[n][n], u[n][n], x[n], y[n], temp, p;
cout<<"\nEnter matrix A values: "<<endl;
for (i = 0; i < n; i++)
for (j = 0; j < n; j++)
cin >> a[i][j];
cout<<"\nEnter B values: "<<endl;
for (i = 0; i < n; i++)
cin >> b[i];


cout << "\nYour matrix is: "<<endl;
for (i = 0; i < n; i++) {
  for (j = 0; j < n; j++) {
     cout<<a[i][j]<<" ";
     ab[i][j] = a[i][j];
     a2[i][j] = a[i][j];
  }
  cout << b[i] << endl;
  ab[i][n] = b[i];
}
```

```
//ранг матрицы a
int R = n;
int C = n;
if(a[0][0]==0){
  for(j=0;j<C;j++){
    z=-1;
    for(i=1;i<R;i++){if(a[i][j]!=0){z=i; break;}}
    if(z!=-1){
      for( v=0;v<C;v++){
        float t=a[0][v];
        a[0][v]=a[i][v];
        a[i][v]=t;
      }
      break;
    }
  }
}
for(k=0;k<R-1;k++){
  for(i=k+1;i<R;i++){
    p=a[k][k];
    q=a[i][k];
    for(j=0;j<C;j++){
      a[i][j]=p*a[i][j] - q*a[k][j];
    }
  }
}
z=0;
for(i=0;i<R;i++){
  int c=0;
  for(j=0;j<C;j++){
    if(a[i][j]==0){
      c++;
    }
  }
  if(c==C){z++;}
}
ranka = R-z;
cout<<"\nRank a = "<< ranka <<endl;
cout<<"\nVariables = "<< var <<endl;
```

3. Листинг

```
//ранг матрицы ab
  R = n;
  C = n+1;
  if(ab[0][0]==0){
    for(j=0;j<C;j++){
      z=-1;
      for(i=1;i<R;i++){
        if(ab[i][j]!=0){z=i; break;}
      }
      if(z!=-1){
        for( v=0;v<C;v++){
          float t=ab[0][v];
          ab[0][v]=ab[i][v];
          ab[i][v]=t;
        }
        break;
      }
    }
  }
  for(k=0;k<R-1;k++){
    for(i=k+1;i<R;i++){
      p=ab[k][k];
      q=ab[i][k];
      for(j=0;j<C;j++){ab[i][j]=p*ab[i][j] - q*ab[k][j];}
    }
  }
  z=0;
  for(i=0;i<R;i++){
    int c=0;
    for(j=0;j<C;j++){
      if(ab[i][j]==0){c++;}
    }
    if(c==C){z++;}
  }
  rankab = R-z;
  cout<<"\nRank ab = "<< rankab <<endl;
```

```
//проверка
if (ranka > rankab && ranka != var) {
  cout << "\nСистема - неопределенная" << '\n';
  return 0;
}
if (ranka < rankab){
  cout << "\nСистема - не совместна" << '\n';
  return 0;
}
if (ranka == rankab && ranka != var){
  cout << "\nСистема - не совместна" << '\n';
  return 0;
}
```

3. Листинг

```
//LU разложение
 for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++) {
       if (j < i)
       l[j][i] = 0;
       else {
          l[j][i] = a2[j][i];
          for (k = 0; k < i; k++) {
             l[j][i] = l[j][i] - l[j][k] * u[k][i];
            }
       }
    }
    for (j = 0; j < n; j++) {
       if (j < i)
       u[i][j] = 0;
       else if (j == i)
       u[i][j] = 1;
       else {
          u[i][j] = a2[i][j] / l[i][i];
          for (k = 0; k < i; k++) {
            u[i][j] = u[i][j] - ((l[i][k] * u[k][j]) / l[i][i]);
            }
       }
    }
 }
 cout << "\nL Decomposition is as follows..."<<endl;
 for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++) {cout<<l[i][j]<<" ";}
    cout << endl;
 }
 cout << "\nU Decomposition is as follows..."<<endl;
 for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++) {cout<<u[i][j]<<" ";}
    cout << endl;
 }
```

```
//ly=b
y[0] = b[0]/l[0][0];

for (i = 1; i < n; i++) {
  temp = 0;
  for (k = 0; k < i; k++) {
    temp = temp + l[i][k]*y[k];
  }
  y[i]=1/l[i][i]*(b[i]-temp);
}
  cout << '\n';
for (i = 0; i < n; i++) {
  cout << "y["<< i <<"] = " << y[i] << '\n';
}
cout << endl;

//ux=y
x[n-1] = y[n-1];

for (i = n-1; i > -1; i--) {
  temp = 0;
  for (k = i+1; k < n ; k++) {
    temp = temp + u[i][k]*x[k];
  }
  x[i]=y[i]-temp;
}

cout << '\n';
for (i = 0; i < n; i++) {
cout << "x["<< i <<"] = " << x[i] << '\n';
}
 return 0;
}
```

4. Заключение

Enter size of square matrix :

3

Enter number of variables :

3

Enter matrix A values:

1 1 -1

1 -2 3

2 3 1

Enter B values:

4 -6 7

Your matrix is:

1 1 -1 4

1 -2 3 -6

2 3 1 7

Rank a = 3

Variables = 3

Rank ab = 3

L Decomposition is as follows...

1 0 0

1 -3 0

2 1 4.33333

U Decomposition is as follows...

1 1 -1

0 1 -1.33333

0 0 1

y[0] = 4

y[1] = 3.33333

y[2] = -1

x[0] = 1

x[1] = 2

x[2] = -1

Проверка:

1(1) + 1(2) -1(-1) = 4

1(1) - 2(2) + 3(-1) = -6

2(1) + 3(2) + 1(-1) = 7

4. Заключение

Enter size of square matrix :
4

Enter number of variables :
4

Enter matrix A values:
3 1 -1 2
-5 1 3 -4
2 0 1 -1
1 -5 3 -3

Enter B values:
6 -12 1 3

Your matrix is:
3 1 -1 2 6
-5 1 3 -4 -12
2 0 1 -1 1
1 -5 3 -3 3

Rank a = 4

Variables = 4

Rank ab = 4

L Decomposition is as follows...
3 0 0 0
-5 2.66667 0 0
2 -0.666667 2 0
1 -5.33333 6 2.5

U Decomposition is as follows...
1 0.333333 -0.333333 0.666667
0 1 0.5 -0.25
0 0 1 -1.25
0 0 0 1

y[0] = 2
y[1] = -0.75
y[2] = -1.75
y[3] = 3

x[0] = 1
x[1] = -1
x[2] = 2
x[3] = 3

Проверка:

3(1) + 1(-1) – 1(2) + 2(3) = 6
-5(1) + 1(-1) + 3(2) - 4(3) = -12
2(1) + 0(-1) + 1(2) – 1(3) = 1
1(1) - 5(-1) + 3(2) - 3(3) = 3

4. Заключение

Enter size of square matrix :
4

Enter number of variables :
4

Enter matrix A values:
4 -3 2 -1
3 -2 1 -3
5 -3 1 -8
0 0 0 0

Enter B values:
8 7 1 0

Your matrix is:
4 -3 2 -1 8
3 -2 1 -3 7
5 -3 1 -8 1
0 0 0 0 0

Rank a = 2

Variables = 4

Rank ab = 3

Система - не совместна

$$\left(\begin{array}{cccc|c} 4 & -3 & 2 & -1 & 8 \\ 3 & -2 & 1 & -3 & 7 \\ 5 & -3 & 1 & -8 & 1 \end{array}\right) \xrightarrow{(1)} \left(\begin{array}{cccc|c} -1 & 0 & 1 & 7 & 7 \\ 3 & -2 & 1 & -3 & 7 \\ 5 & -3 & 1 & -8 & 1 \end{array}\right) \xrightarrow{(2)} \left(\begin{array}{cccc|c} -1 & 0 & 1 & 7 & 7 \\ 0 & -2 & 4 & 18 & 28 \\ 0 & -3 & 6 & 27 & 36 \end{array}\right) \xrightarrow{(3)}$$

$$\rightarrow \left(\begin{array}{cccc|c} -1 & 0 & 1 & 7 & 7 \\ 0 & -1 & 2 & 9 & 14 \\ 0 & 1 & -2 & -9 & -12 \end{array}\right) \xrightarrow{(4)} \left(\begin{array}{cccc|c} -1 & 0 & 1 & 7 & 7 \\ 0 & -1 & 2 & 9 & 14 \\ 0 & 0 & 0 & 0 & 2 \end{array}\right)$$

5. Список литературы

– Демидович Б. П., Марон И. А. Основы вычислительной математики: Учеб. Пособие. – 3-е изд., – М: Изд-во «Наука», 1966. – 664 с.

Спасибо за внимание!