БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Лабораторная работа
по вычислительным методам алгебры на тему:

Решение систем линейных алгебраических уравнений с помощью LU-разложения

Выполнил:
Архангельский И.А.

Проверил:
Кондратюк А.П.

Минск, 2012

Входные и выходные данные.

Входные данные

Входной файл содержит матрицу $(A|f)$, где A - квадратная матрица коэффициентов СЛУ, f - вектор-столбец свободных членов.

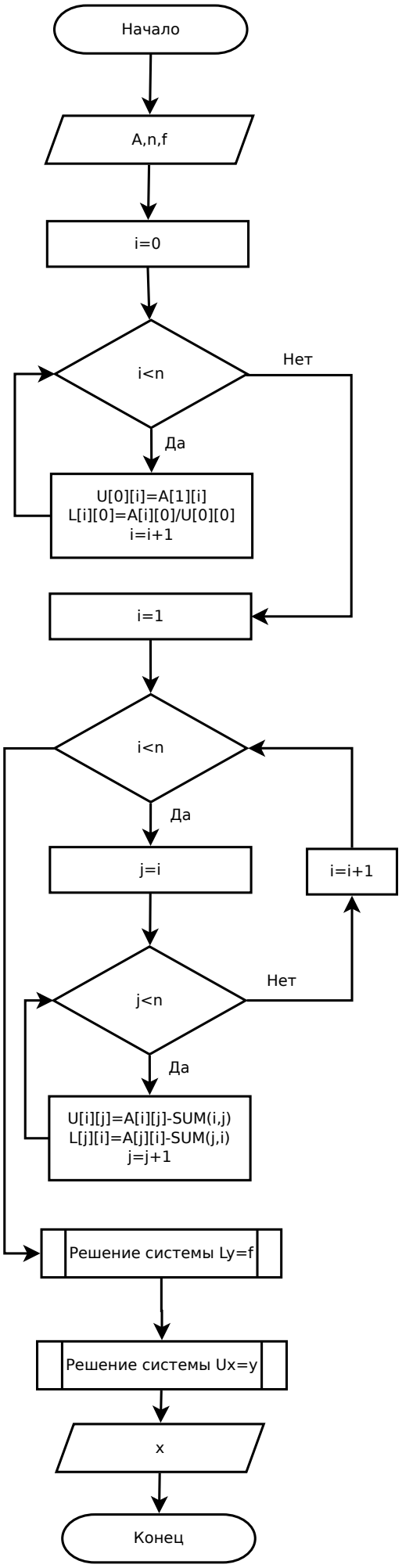
Выходные данные

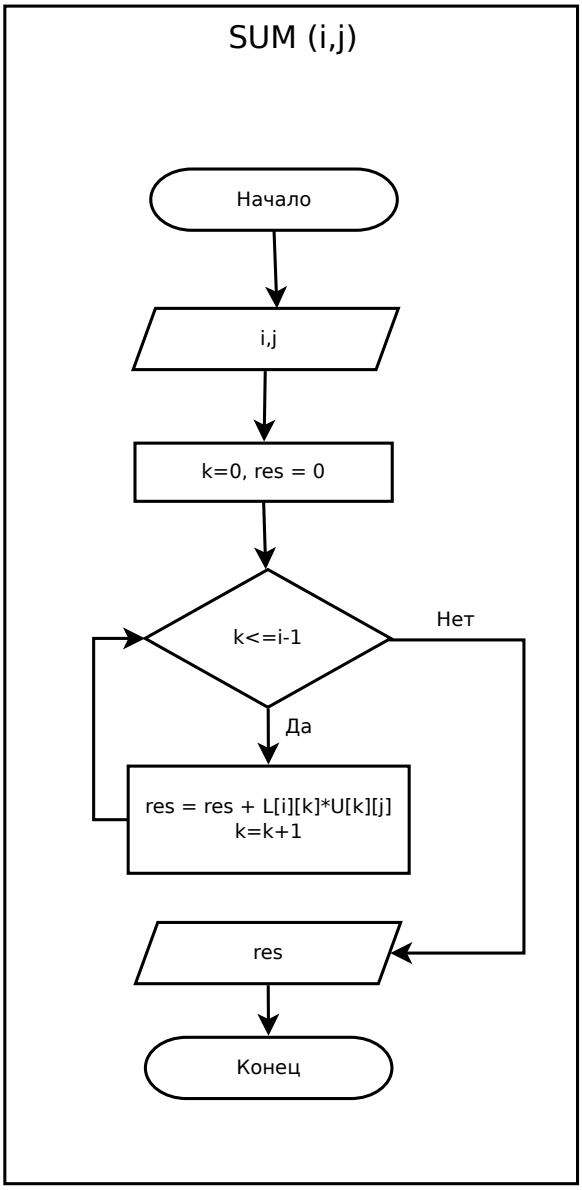
Выходной файл содержит решение СЛУ.

Исключения

- NilMatrixExcpetion - на вход подана пустая матрица (например, входной файл пуст)
- WrongMatrixDimentionsException - на вход подана матрица размеры которой не соотвествуют предполагаемым
- NoArgsException - не передано имя входного файла
- DivByZeroException - во время получения LU-разложения произошло деление на 0, т.е. на диагонали матрицы U стоит 0

Блок-схема


Начало
A,n,f
i=0
i<n
Нет
Да
U[0][i]=A[1][i]
L[i][0]=A[i][0]/U[0][0]
i=i+1
i=1
i<n
Да
j=i
i=i+1
j<n
Нет
Да
U[i][j]=A[i][j]-SUM(i,j)
L[j][i]=A[j][i]-SUM(j,i)
j=j+1
Решение системы Ly=f
Решение системы Ux=y
x
Конец


SUM (i,j)
Начало
i,j
k=0, res = 0
k<=i-1
Нет
Да
res = res + L[i][k]*U[k][j]
k=k+1
res
Конец

Реализация

```ruby
#!/usr/bin/env ruby
class NilMatrixExcpetion < StandardError; end
class WrongMatrixDimentionsException < StandardError; end
class NoArgsException < StandardError; end
class DivByZeroException < StandardError; end

class LU
  def initialize (stream)
    @matrix = Array.new
    stream.each { |line| tmp = Array.new; line.scan(/-?\d+/) {|match| tmp.push(match.to_f)}; @matrix.push(
    raise NilMatrixExcpetion unless @matrix.length!=0
    raise WrongMatrixDimentionsException unless allok?
  end

  def allok?
    return @matrix.find_all { |obj| obj.length == @matrix.length+1}.length==@matrix.length
  end

  def solve
    l = Array.new(@matrix.length).map{Array.new}
    u = Array.new(@matrix.length).map{Array.new}
    @matrix.each_index { |i| u[0].push(@matrix[0][i])}
    raise DivByZeroException unless u[0][0]!=0
    @matrix.each_index {|i| l[i].push(@matrix[i][0]/u[0][0])}
    @matrix.each_index {|i|
      if i>0
        @matrix.each_index { |j|
          if j>=i
            tmp = 0
            for k in 0 .. i-1
              tmp +=l[i][k]*u[k][j]
            end
            u[i][j]=@matrix[i][j]-tmp;
          end
        }
        @matrix.each_index { |j|
          if j>=i+1
            tmp = 0
            for k in 0..i-1
              tmp+=l[j][k]*u[k][i]
            end
            l[j][i] =(@matrix[j][i]-tmp)/u[i][i];
            raise DivByZeroException unless u[i][i]!=0
          end
        }
        l[i][i]=1.0;
      end
    }
    y = Array.new(@matrix.length);
    y.each_index {|i|
      tmp = 0
      l[i].each_index { |j| tmp+=l[i][j]*y[j] if j<i }
      y[i]=@matrix[i].last-tmp
    }
    x = Array.new(@matrix.length)
    x.each_index { |i|
      tmp = 0 ;
      u[u.length-1-i].each_index { |j| tmp+=u[u.length-1-i][j]*x[j] if j>u.length-1-i }
      x[x.length-1-i] = (y[x.length-1-i]-tmp)/u[u.length-1-i][u.length-1-i];
    }
   return x;
  end
end

raise NoArgsException unless ARGV.first!=nil
lu = LU.new(File.open(ARGV.first, "r"))
fout = File.open(ARGV.first.gsub(/\.in/, ".out"),"w")
lu.solve.each { |e| fout<<sprintf("%3.3f",e)<<" "}
```

Тестовые данные

test01.in	test01.out
1 1 3 2 4 10	1.000 2.000

test02.in	test02.out
3 8 3 −1 4 2 3 4 1 −4 1 −3 −2 −2 3 5 −8 4 2 −8	2.000 1.000 −3.000 1.000

test03.in	test03.out
1 4 5 2 20 1 2 3 1 11 7 9 10 2 40 2 9 8 3 27	16.667 −1.333 −14.667 41.000

test04.in	test04.out
0 2 3 1 2 2 4 2 2 3 3 3	DivByZeroException

test05.in	test05.out
2 1 4 3 10 5 3 5 3 16 5 1 8 6 20 7 3 5 3 18	1.000 1.000 1.000 1.000