Университет ИТМО

Кафедра Вычислительной Техники

**Лабораторная работа №1**

**по дисциплине “Вычислительная математика”**

Группа Р3201

Метод простых итераций

Комаров Егор Андреевич

Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Принял: Калёнова Ольга Вячеславовна

Санкт-Петербург, 2018

**Описание метода:**Метод простых итераций – один из приближенных численных методов.

Пусть дана линейная система

Рассмотрим матрицы:

Тогда систему можно записать в виде уравнения

Разрешим первое уравнение системы относительно x1, второе – относительно x2 и тд:

Где

Получаются матрицы

Тогда систему можно записать в виде уравнения

Будем решать систему методом последовательных приближений. За нулевое приближение можно взять произвольные числа

И так далее

Если последовательность приближений имеет предел, то этот предел является решением системы

При этом процесс итерации сводится к единственному решению этой системы только при выполнении по меньшей мере одного из условий

Или

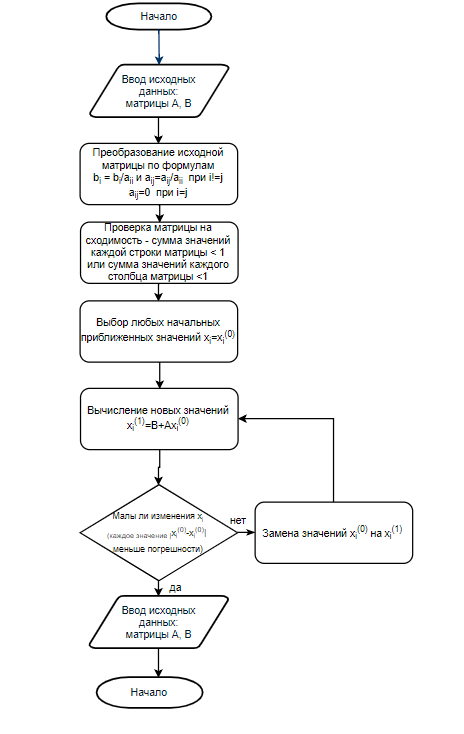
IterMethod.h

|  |
| --- |
| #pragma once  #include"Matrix.h"  class IterMethod  {  public:  IterMethod();  bool calculate(Matrix matrixA, Matrix matrixB, float precision);  void convertInitMatrix(Matrix& matrixA, Matrix& matrixB);  bool checkPrecision(const Matrix& matrix);  bool checkConvergence(Matrix& matrix);  //getters  Matrix& getResult();  Matrix& getUnknown();  Matrix& getErrors();  int getItersTime();  private:  int m\_iterNums;  double m\_precision;  Matrix m\_unknown;  Matrix m\_errors;  Matrix m\_result;  }; |

IterMethod.cpp

|  |
| --- |
| #include "IterMethtod.h"  #include <cmath>  IterMethod::IterMethod()  {  m\_iterNums = 0;  }  bool IterMethod::calculate(Matrix matrixA, Matrix matrixB, float precision)  {  // check correct size  if (matrixA.getWidth() != matrixA.getHeight() || matrixB.getWidth() != 1 || matrixB.getHeight() != matrixA.getHeight())  return false;  // prepare start values  m\_iterNums = 0;  m\_precision = precision;  m\_errors.resize(0, matrixA.getWidth());  m\_unknown.resize(0, matrixA.getWidth());  // transform matrix for expressing X  convertInitMatrix(matrixA, matrixB)  // check convergence first  if (!checkConvergence(matrixA))  return false;  m\_result = matrixB;  Matrix oldResult;  // iterations  do  {  oldResult = m\_result;  m\_result = matrixB + matrixA \* oldResult;  m\_unknown.addRow(m\_result.transponate());  m\_errors.addRow((m\_result - oldResult).transponate());  m\_iterNums++;  }  while(!checkPrecision(m\_result - oldResult));  return true;  }  void IterMethod::convertInitMatrix(Matrix& matrixA, Matrix& matrixB)  {  for (int i = 0; i < matrixB.getHeight();i++)  matrixB.set(0, i, matrixB.get(0, i) / matrixA.get(i, i));  for (int i = 0; i < matrixA.getHeight();i++)  for(int j = 0;j<matrixA.getWidth();j++)  {  if (i != j)  matrixA.set(j, i, -matrixA.get(j, i) / matrixA.get(i, i));  }  for (int i = 0; i < matrixA.getHeight();i++)  matrixA.set(i, i,0);  }  bool IterMethod::checkPrecision(const Matrix& matrix)  {  for (int i = 0;i < matrix.getHeight();i++)  if (abs(matrix.get(0, i)) > m\_precision)  return false;  return true;  }  bool IterMethod::checkConvergence(Matrix& matrix)  {  // check first condition  bool b = true;  for(int i =0;i<matrix.getWidth();i++)  {  double sum = 0;  for (int j = 0;j < matrix.getWidth();j++)  sum += abs(matrix.get(j,i));  if (sum >= 1)  {  b = false;  break;  }  }  if (b)  return true;  // chek second condition  for (int i = 0;i<matrix.getWidth();i++)  {  double sum = 0;  for (int j = 0;j < matrix.getWidth();j++)  sum += abs(matrix.get(i, j));  if (sum >= 1)  return false;  }  return true;  }  // getters  Matrix& IterMethod::getResult()  {  return m\_result;  }  Matrix& IterMethod::getUnknown()  {  return m\_unknown;  }  Matrix& IterMethod::getErrors()  {  return m\_errors;  }  int IterMethod::getItersTime()  {  return m\_iterNums;  } |

Блок схема:



Пример:

Точность – 0.0001

Результат:

0.425718

0.535542

-0.268325

0.180662

Количество итераций:

26

Столбец неизвестных:

0.279381 0.351259 -0.429461 0.0139629

0.544107 0.650053 -0.141935 0.289755

0.349513 0.443645 -0.352075 0.0962941

0.485199 0.595096 -0.204548 0.237012

0.38624 0.48928 -0.311534 0.138004

0.455724 0.566325 -0.236055 0.209666

0.405344 0.512151 -0.290561 0.159025

0.440893 0.551383 -0.251969 0.195544

0.415226 0.523672 -0.279753 0.169657

0.433402 0.543663 -0.26003 0.188276

0.420317 0.529497 -0.274198 0.175048

0.429606 0.539688 -0.264122 0.184544

0.422934 0.532449 -0.271349 0.177787

0.427679 0.537647 -0.266203 0.182631

0.424276 0.533949 -0.26989 0.17918

0.4267 0.5366 -0.267262 0.181651

0.424963 0.534712 -0.269144 0.179889

0.426201 0.536064 -0.267802 0.18115

0.425315 0.5351 -0.268762 0.180251

0.425947 0.53579 -0.268077 0.180894

0.425495 0.535298 -0.268567 0.180435

0.425817 0.53565 -0.268217 0.180763

0.425587 0.535399 -0.268467 0.180529

0.425751 0.535578 -0.268288 0.180697

0.425634 0.53545 -0.268416 0.180577

0.425718 0.535542 -0.268325 0.180662

Столбец погрешностей:

-0.377852 -0.391056 -0.435423 -0.38896

0.264726 0.298794 0.287526 0.275792

-0.194593 -0.206407 -0.21014 -0.193461

0.135686 0.151451 0.147527 0.140718

-0.09896 -0.105816 -0.106986 -0.0990081

0.0694844 0.0770448 0.0754786 0.0716626

-0.0503803 -0.0541734 -0.0545058 -0.050641

0.0355498 0.0392318 0.0385922 0.0365189

-0.0256678 -0.0277111 -0.0277842 -0.0258871

0.0181762 0.0199909 0.0197229 0.0186187

-0.0130843 -0.0141664 -0.0141685 -0.0132278

0.00928891 0.0101917 0.0100761 0.0094958

-0.00667243 -0.00723907 -0.00722721 -0.00675715

0.00474552 0.0051977 0.00514651 0.00484417

-0.00340358 -0.00369804 -0.00368728 -0.00345103

0.00242381 0.00265148 0.00262819 0.00247164

-0.0017365 -0.00188871 -0.00188151 -0.00176226

0.00123777 0.00135284 0.00134199 0.00126126

-0.000886087 -0.00096448 -0.000960172 -0.000899793

0.000632019 0.000690333 0.000685175 0.000643668

-0.000452191 -0.000492461 -0.000490033 -0.000459391

0.000322687 0.0003523 0.000349806 0.00032851

-0.00023078 -0.000251429 -0.000250106 -0.000234531

0.000164743 0.000179803 0.00017858 0.00016767

-0.000117787 -0.000128361 -0.000127655 -0.000119729

0.000084103 0.0000917701 0.0000911644 0.0000855807

Точность – 0.0000000001

Результат:

0.0000615337

0.00085381

0.000661722

0.00126921

0.000436343

Количество итераций:

3

Столбец неизвестных:

6.15316e-05 0.000853806 0.00066172 0.00126921 0.00043634

6.15337e-05 0.00085381 0.000661722 0.00126921 0.000436343

6.15337e-05 0.00085381 0.000661722 0.00126921 0.000436343

Столбец погрешностей:

-9.68394e-07 -1.70735e-06 -1.05862e-06 -1.97999e-06 -1.79042e-06

2.11578e-09 3.97186e-09 2.57182e-09 4.30041e-09 3.20646e-09

-4.35532e-12 -8.30804e-12 -5.64343e-12 -9.63671e-12 -7.21206e-12

Вывод:

Метод простых итераций подходит для систем любого размера, в том числе при n>200, так как количество вычислений относительно невелико для каждой итерации. Также методом достигается низкая погрешность за счет итераций, по сравнению с прямыми методами, из-за большого количества последовательных вычислений и ограниченности разрядной сетки в которых, итоговый результат получается неточным. Однако для выполнения метода должны выполняться строгие условия сходимости, поэтому не для всех систем данный метод будет работать. Также преимуществом метода можно считать его понятность и простоту реализации.