Федеральное государственное автономное

образовательное учреждение высшего образования

“Национальный исследовательский университет ИТМО”

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1

по вычислительной математике

Вариант №2

Выполнил:

Голиков Андрей

Группа: P32092

Принял: Рыбаков

Степан Дмитриевич

???

?



Санкт-Петербург, 2023

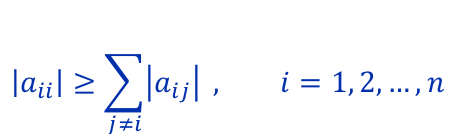
Цель: ознакомиться с методами решения СЛАУ

Метод: метод простой итерации

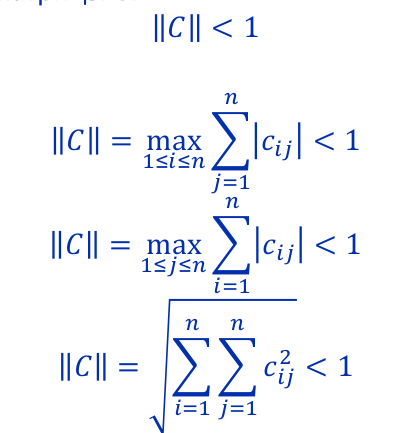
Описание: Приводим СЛАУ к виду x = Cx + D, подставляя в правую часть предыдущие значения x(на первой итерации можно взять D или 0), получаем все лучшее приближение решения

Условие сходимости:

1. Необходимый



1. Достаточный



Листинг:

public ResolveResult resolve(OutputFormatter outputFormatter, boolean isGeneratedAutomatically) {

if (!isGeneratedAutomatically) {

double determinant = calculateDeterminant(matrix);

if (Double.*isNaN*(determinant) || determinant == 0) return ResolveResult.*DEGENERATE\_MATRIX*;

getPredominanceOfDiagonalElements();

if (!isPredominanceOfDiagonalElements())

return ResolveResult.*PREDOMINANCE\_OF\_DIAGONAL\_ELEMENTS\_CONDITION*;

getXValues();

}

if (!isConvergenceCondition()) return ResolveResult.*COVERAGE\_CONDITION\_FAILED*;

Double[] freeElements = getFreeElements();

Double[] previousIterationElements = getFreeElements();

int iterationNumber = 1;

while (calculateAnswer(matrix.getMatrixValues(), previousIterationElements, freeElements, matrix.getEpsilon(),

iterationNumber, outputFormatter)) {

iterationNumber += 1;

}

return ResolveResult.*SUCCESSFUL*;

}

private double calculateDeterminant(Matrix srcMatrix) {

Matrix matrixCopy = srcMatrix.getCopy();

double determinant = 1.;

setToTriangleView(matrixCopy);

for (int i = 0; i < matrixCopy.getColumnSize(); i++) {

determinant \*= matrixCopy.getMatrixValues()[i][i];

}

return determinant;

}

private void setToTriangleView(Matrix matrixCopy) {

for (int i = 0; i < matrixCopy.getColumnSize(); i++) {

double hostElement = matrixCopy.getMatrixValues()[i][i];

for (int j = i + 1; j < matrixCopy.getColumnSize(); j++) {

double currentElement = matrixCopy.getMatrixValues()[j][i];

double k = currentElement / hostElement;

double[] newHostRow = multipleRow(matrixCopy, i, k);

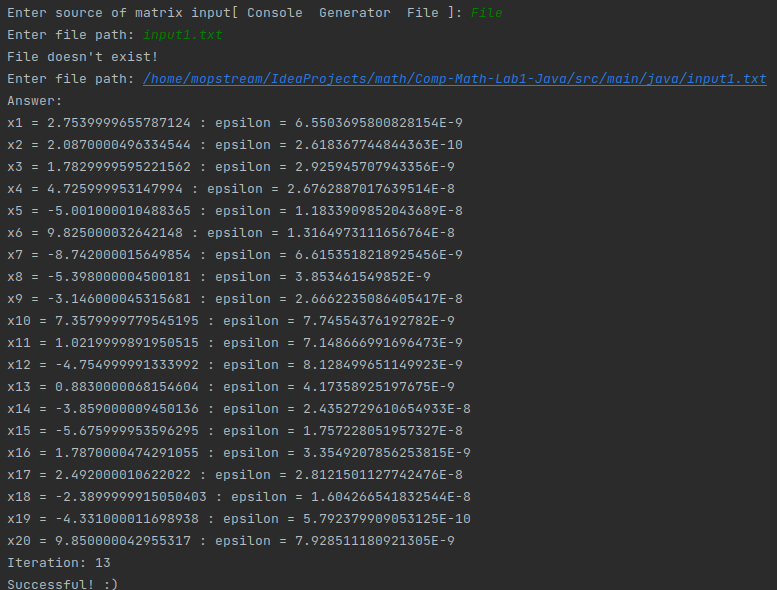
subRows(matrixCopy, newHostRow, j);

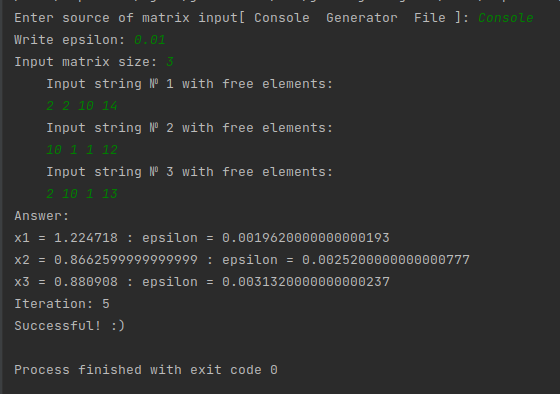
}

}

}

Примеры:





Вывод: Реализовал на языке програмирования Java метод простых итераций для решения системы линейных алгебраических уравнений. Выявил преимущества метода в его универсальности и простоте реализации и недостатки в трудоемкости и медленной скорости сходимости.