|  |
| --- |
| БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ |
| Факультет прикладной математики и информатики |
| Кафедра информационных систем управления |
| Лукьянович Александр Сергеевич  2 курс, 10 группа |
| Лабораторная работа |
| **Метод прогонки для решения СЛАУ**    **Преподаватель**  *Полевиков Виктор Кузьмич* |
|  |
| Дата сдачи:  13.10.2016 |

|  |
| --- |
| Минск, 2016 |

Оглавление

[ **Постановка задачи** 3](#_Toc430887395)

[ **Краткая теория** 5](#_Toc430887396)

[ **Листинг программы** 6](#_Toc430887397)

[ **Результаты** 8](#_Toc430887398)

* **Постановка задачи**

Рассматриваем следующую систему: A\*x = f, где A – трёхдиагональная матрица коэффициентов, x – вектор неизвестных, f – вектор свободных значений.

A=

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  |  |  |  |  |
| 7.918 | 3.345 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| -0.962 | 5.123 | -3.113 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 0.000 | 0.133 | 4.156 | -1.823 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 0.000 | 0.000 | 4.948 | 9.298 | -0.594 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 0.000 | 0.000 | 0.000 | 1.573 | -7.253 | -3.620 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 0.000 | 0.000 | 0.000 | 0.000 | -2.053 | -9.181 | 1.704 | 0.000 | 0.000 | 0.000 | 0.000 |
| 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | -2.102 | 7.235 | 3.596 | 0.000 | 0.000 | 0.000 |
| 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1.087 | -6.180 | 0.116 | 0.000 | 0.000 |
| 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | -1.959 | 9.103 | 3.125 | 0.000 |
| 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | -4.660 | 8.880 | 1.608 |
| 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | -3.091 | -3.282 |

Достаточные условия леммы о корректности и устойчивости прогонки соблюдены (см. в теории)

|  |
| --- |
| -4 |
| -8 |
| 8 |
| -1 |
| -5 |
| 7 |
| 2 |
| 9 |
| 1 |
| 5 |
| 8 |

=

Перемножив матрицу и вектор , получили следующий вектор :

=

|  |
| --- |
| -58.4320000000000022 |
| -62.0400000000000063 |
| 34.0069999999999979 |
| 33.2560000000000002 |
| 9.3520000000000003 |
| -50.5939999999999941 |
| 32.1200000000000045 |
| -53.3299999999999983 |
| 7.0969999999999995 |
| 52.6040000000000134 |
| -41.7109999999999985 |

1. Построить стандартную программу для решения системы по методу прогонки для числа уравнений n ≥ 1.
2. Решить систему с помощью этой программы и получить приближенное значение .
3. Сравнить полученное приближённое значение с исходным вектором

* **Краткая теория**

Имеем систему уравнений:

Решая систему по методу прогонки, получаем следующими коэффициенты:

Вектор неизвестных находится следующим образом:

**Лемма** о корректности метода прогонки:

Метод прогонки применим, если выполняются следующие условия:

Достаточное условие корректности и устойчивости прогонки: метод прогонки применим и устойчив, если одно из условий {(2), (3)} выполняется строго.

* **Листинг программы**

Программа выполнена на языке программирования C++, в среде Microsoft Visual Studio Community 2015.

#define \_SCL\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <fstream>

#include <exception>

#include <iomanip>

#include <boost\numeric\ublas\matrix.hpp>

#include <boost\numeric\ublas\vector.hpp>

std::ofstream fout("output.csv");

std::ifstream fin("input.txt");

const int N = 10; //умное число

const int size = N + 1;

const char\* separator = ";";

void readFromFile(boost::numeric::ublas::matrix<double>&, boost::numeric::ublas::vector<double>&);

void printMatrix(boost::numeric::ublas::matrix<double>);

void printVector(boost::numeric::ublas::vector<double>);

void printfVector(boost::numeric::ublas::vector<double>);

boost::numeric::ublas::vector<double> pryamoyAndObratnyiyHod(boost::numeric::ublas::matrix<double>, boost::numeric::ublas::vector<double>&);

int main(int argc, char argv[]) {

setlocale(LC\_ALL, ".1251");

try {

if (!fout)

throw std::exception("выходной файл не создан.");

else if (!fin)

throw std::exception("входной файл не отрыт.");

boost::numeric::ublas::matrix<double> A(size, size);

boost::numeric::ublas::vector<double> x(size), f(size);

readFromFile(A, x);

fout << "матрица А:\n";

printMatrix(A);

fout << "\nвектор х(транспонированный):\n";

printVector(x);

fout << "\n\nвектор f(транспонированный):\n";

printfVector(f = boost::numeric::ublas::prod(A, x));

fout << "\n\nприближённое значение x1:\n";

printfVector(pryamoyAndObratnyiyHod(A, f));

}

catch (std::exception& e) { std::cerr << e.what() << std::endl; }

return 0;

}

void readFromFile(boost::numeric::ublas::matrix<double>& A, boost::numeric::ublas::vector<double>& x) {

for (int i(0); i < size; ++i)

for (int j(0); j < size; ++j)

fin >> A(i, j);

for (int i(0); i < size; ++i)

fin >> x(i);

//закроем файл тут, он глобальный для удобства. ну и чтобы закрытие не маячило в main'е

fin.close();

}

void printMatrix(boost::numeric::ublas::matrix<double> A) {

for (int i(0); i < size; ++i) {

for (int j(0); j < size; ++j)

fout << std::fixed << std::setprecision(3) << A(i, j) << separator;

fout << std::endl;

}

}

void printVector(boost::numeric::ublas::vector<double> v) {

fout.unsetf(std::ios::fixed);

for (int i(0); i < size; ++i)

fout << v(i) << "\n";

}

void printfVector(boost::numeric::ublas::vector<double> v) {

for (int i(0); i < size; ++i)

fout << std::fixed << std::setprecision(16) << v(i) << "\n";

}

boost::numeric::ublas::vector<double> pryamoyAndObratnyiyHod (boost::numeric::ublas::matrix<double> A, boost::numeric::ublas::vector<double>& f) {

boost::numeric::ublas::vector<double> a(size), b(size), c(size), alpha(N), betta(N);

b(0) = -A(0, 1);

a(N) = -A(N, N - 1);

c(0) = A(0, 0);

for (int i(1); i < N; ++i) {

a(i) = -A(i, i - 1);

b(i) = -A(i, i + 1);

c(i) = A(i, i);

}

//прямой ход

alpha(0) = b(0) / c(0);

betta(0) = f(0) / c(0);

for (int i(1); i < N; ++i) {

alpha(i) = b(i) / (c(i) - a(i)\*alpha(i - 1));

betta(i) = (f(i) + a(i)\*betta(i - 1)) / (c(i) - a(i)\*alpha(i-1));

}

//обратный ход

boost::numeric::ublas::vector<double> x(size);

x(N) = (f(N) - a(N)\*betta(N - 1)) / (c(N) + alpha(N - 1)\*a(N));

for (int i(N - 1); i >= 0; --i)

x(i) = alpha(i)\*x(i + 1) + betta(i);

return x;

}

* **Результаты**

Применив обратный ход метода прогонки, получили следующее решение :

|  |
| --- |
| -3.9999999999999996 |
| -8.0000000000000000 |
| 8.0000000000000000 |
| -1.0000000000000007 |
| -5.0000000000000000 |
| 7.0000000000000000 |
| 1.9999999999999991 |
| 9.0000000000000000 |
| 1.0000000000000000 |
| 5.0000000000000009 |
| 7.9999999999999973 |

|  |
| --- |
| -4 |
| -8 |
| 8 |
| -1 |
| -5 |
| 7 |
| 2 |
| 9 |
| 1 |
| 5 |
| 8 |