БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Лабораторная работа №3

Решение систем линейных уравнений с трёхдиагональной матрицей методом правой прогонки

Выполнил:
Студент 2 курса 5 группы ФПМИ
Карасик Семён

Руководитель:
Радкевич Елена Владимировна

Минск, 2017 г.

Оглавление

Постановка задачи

Решить систему линейных алгебраических уравнений с трёхдиагональной матрицей методом правой прогонки.

Описание метода прогонки для решения СЛАУ с трёхдиагональной матрицей

Метод прогонки используется для решения СЛАУ с трёхдиагональной матрицей. По своей сути он является реализация метода Гаусса, учитывающей специфику исходной системы, что позволяет упростить решение.

$$\begin{cases} c_0 y_0 - b_0 y_1 = f_0 \\ -a_{i-1} y_{i-1} + c_i y_i - b_{i+1} y_{i+1} = f_i, i = \overline{1, n-1} \\ -a_n y_{n-1} + c_n y_n = f_n \end{cases}$$

Поделим первое уравнение на c_0 и определим $\alpha_1 = b_0 / c_0, \beta_1 = f_0 / c_0$, тогда уравнение примет вид:

$y_0 - \alpha_1 y_1 = \beta_1$

Исключим переменную y_0 из второго уравнения, отняв от него первое уравнение, домноженное на a_0 . На этом этапе мы имеем $n-1$ уравнение с $n-1$ неизвестной.

Рассмотрим i-ое и (i+1)-ое уранение. Поделим i-ое уравнение на $c_i - \alpha_i a_i$ и определим $\alpha_{i+1} = \frac{b_i}{c_i - \alpha_i a_i} i = \overline{1, n-1}, \beta_{i+1} = \frac{f_i + a_i \beta_i}{c_i - \alpha_i a_i} i = \overline{1, n}$. Исключим переменную y_i из (i+1)-го уравнения.

Процесс вычисления коэффициентов α_i и β_i называется правой прогонкой.

Повторив данные рассуждения для всех уравнений системы, получим

$y_n = \beta_{n+1}$ и $y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}, i = \overline{n-1, 0}$

Процесс вычисления неизвестных называется обратным ходом метода правой прогонки.

Если бы начали наши рассуждения с аналогичных действий над последним уравнением системы, получили бы метод левой прогонки.

Также в случае, если нужно найти значение только одной неизвестной $y_k, k = \overline{0, n}$, можно применить метод встречных прогонок: вычислить прогоночные коэффициенты правой и левой прогонки, а затем, приняв во внимание тот факт, что два метода вычисляют одни и те же неизвестные, получаем следующую формулу:

$y_k = \frac{\eta_k + \xi_k \beta_k}{1 - \xi_k \alpha_k}$.

Очевидно, при необходимости вычислить только одну переменную или группу подряд идущих переменных, данный метод значительно сокращает объем вычислений.

Листинг программы

```
//lab3, v20. Simon Karasik, course 2, group 5.
#include <fstream>
#include <iomanip>
#include <cmath>
#include <vector>

using namespace std;

typedef std::vector<double> Vector;

void printVector(ostream & os, const Vector & vector,
int from, int to)
{
	for (int i = from; i < to; i++)
		os << setprecision(4) << vector[i] <<
'\t';
	os << endl;
}

void printVector(ostream & os, const Vector & vector)
{
	printVector(os, vector, 0, vector.size());
}

Vector solutionError(
	const Vector & a,
	const Vector & b,
	const Vector & c,
	const Vector & f,
	const Vector & y)
{
	const int n = c.size();
	Vector error(n), f1(n, 0);
	f1[0] = c[0]*y[0] - b[0]*y[1] ;
	f1[n - 1] = c[n-1]*y[n-1] - a[n-1]*y[n-2];
	for (int i = 1; i < n - 1; i++)
		f1[i] = -a[i]*y[i - 1] + c[i]*y[i] -
b[i]*y[i+1];
	for (int i = 0; i < n; i++)
		error[i] = abs(f[i] - f1[i]);
	return error;
}

int main()
{
	ifstream fin("input.txt");
	ofstream fout("output.txt");


	/*
	c0 b0
	a0 c1 b1
	a1
	*/
	int n;
	fin >> n;
	Vector a(n, 0), b(n, 0), c(n, 0), f(n, 0);
	for (int i = 1; i < n; i++)
	{
		fin >> a[i];
		a[i] *= -1;
	}
	for (int i = 0; i < n - 1; i++)
	{
		fin >> b[i];
		b[i] *= -1;
	}
	for (int i = 0; i < n; i++)
		fin >> c[i];
	for (int i = 0; i < n; i++)
		fin >> f[i];

	Vector alpha(n + 1), beta(n + 2);

	alpha[0] = b[0] / c[0];
	beta[0] = f[0] / c[0];
	for (int i = 0; i < n; i++)
	{
		double denom = c[i] - a[i] * alpha[i];
		if (i < n - 1)
			alpha[i + 1] = b[i] / denom;
		beta[i + 1] = (f[i] + a[i] * beta[i]) /
denom;
	}

	Vector y(n);
	y[n - 1] = beta[n];
	for (int i = n - 2; i >= 0; i--)
		y[i] = alpha[i + 1] * y[i + 1] + beta[i +
1];

	Vector error = solutionError(a, b, c, f, y);

	fout << "-a:" << endl;
	printVector(fout, a, 0, n - 1);
	fout << "-b:" << endl;
	printVector(fout, b, 1, n);
	fout << "c:" << endl;
	printVector(fout, c);
	fout << "alpha:" << endl;
	printVector(fout, alpha, 0, n);
	fout << "beta:" << endl;
	printVector(fout, beta, 0, n + 1);
	fout << "y:" << endl;
	printVector(fout, y);
	fout << "y error:" << endl;
	printVector(fout, error);

	return 0;
}
```

Входные данные

input.txt:

```
5
-0.0305 -0.0914 -0.0741 -0.0122
0.0000 0.0000 0.0122 0.0000
0.4974 0.3284 0.5887 0.5887 0.4263
1.5875 -1.7590 1.4139 1.7702 -2.0767
```

Выходные данные

```
output.txt
-a:
0     0.0305      0.0914      0.0741
-b:
-0    -0.0122     -0    0
c:
0.4974      0.3284      0.5887      0.5887      0.4263
alpha:
-0    -0    -0    -0.02072    -0
beta:
3.192 3.192 -5.06 1.616 3.202 -4.78
y:
3.192 -5.06 1.55  3.202 -4.78
y error:
0     2.22e-16    2.22e-16    0     0
```