**Министерство образования Республики Беларусь**

**Белорусский государственный университет**

Факультет прикладной математики и информатики

Лабораторная работа №3

По курсу «Вычислительные методы алгебры»

**Метод Зейделя**

Вариант №5

Работу выполнил:

студент 3 курса 7 группы

**Шатерник Артём**

Преподаватель:

**Будник А. М**.

**Минск 2023**

1. **Постановка задачи.**

Найти решение системы линейных алгебраических уравнений c расширенной матрицей вида

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0.5757 | -0.0758 | 0.0152 | 0.0303 | 0.1061 |  | 3.5148 |
|  |  |  |  |  |  |  |
| 0.0788 | 0.9014 | 0.0000 | -0.0606 | 0.0606 |  | 3.8542 |
|  |  |  |  |  |  |  |
| 0.0455 | 0.0000 | 0.7242 | -0.2121 | 0.1212 |  | -4.9056 |
|  |  |  |  |  |  |  |
| -0.0909 | 0.1909 | 0.0000 | 0.7121 | -0.0303 |  | 2.3240 |
|  |  |  |  |  |  |  |
| 0.3788 | 0.0000 | 0.1364 | 0.0152 | 0.8484 |  | 0.1818 |

применяя метод Зейделя. Вычислить невязки и сравнить с методом Гаусса и отражений по точности и экономичности.

1. **Алгоритм решения.**

**Предварительные преобразования.**

Для использования метода Зейделя систему уравнений нужно привести к каноническому виду .

Чтобы построить матрицу для начала приведём матрицу к симметричному виду. Воспользуемся трансформацией Гаусса:

* Вычислим .
* Умножим систему слева на : .
* Полученная матрица будет симметричной, а система будет иметь вид

Матрицу будем строим по формуле , при этом .

При таком построении собственные значения матрицы будут меньше единицы и метод Зейделя будет сходиться по необходимому и достаточному условию.

**Метод Зейделя.**

Критерий остановки процесса:  , где .

1. **Листинг программы.**

int main() {

setlocale(LC\_ALL, "Russian");

// Ввод данных

int size = 5;

std::vector <std::vector <long double>> x\_result(size, std::vector <long double>(1));

std::vector <std::vector <long double>> a\_matrix(size, std::vector <long double>(size));

std::vector <std::vector <long double>> b\_vector(size, std::vector <long double>(1));

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

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

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

input >> a\_matrix[i][j];

}

}

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

input >> b\_vector[i][0];

}

// Приводим матрицу к симметричному виду

// Симметричный вид

auto b\_vector\_sim = matrix\_product(transpose(a\_matrix), b\_vector);

auto a\_matrix\_sim = matrix\_product(transpose(a\_matrix), a\_matrix);

long double c = 1 / first\_matrix\_norm(a\_matrix\_sim);

// B = E - c \* A\_sim

std::vector <std::vector <long double>> b\_matrix(size, std::vector <long double>(size));

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

b\_matrix[i] = -c \* a\_matrix\_sim[i];

b\_matrix[i][i] = 1 + b\_matrix[i][i];

}

// g = c \* b\_sim

std::vector <std::vector <long double>> g\_vector(size, std::vector <long double>(1));

g\_vector = c \* b\_vector\_sim;

// Точность

long double e = 1e-5;

// Метод Зейделя

x\_result = b\_vector;

int i = 0;

auto x\_result\_new = x\_result;

while(true) {

i++;

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

long double sum = 0;

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

sum += b\_matrix[i][j] \* x\_result\_new[j][0];

}

x\_result\_new[i][0] = sum + g\_vector[i][0];

}

if (first\_matrix\_norm(x\_result - x\_result\_new) <= e) {

break;

} else {

x\_result = x\_result\_new;

}

}

// Вывод данных

std::cout << "x = (";

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

std::cout << std::setw(8) << round(x\_result\_new[i][0] \* 10000) / 10000 << std::setw(8);

}

std::cout << std::setw(1) << ")" << std::endl;

std::cout << "Колличество итераций: " << i << std::endl;

std::vector <std::vector <long double>> r = matrix\_product(a\_matrix, x\_result\_new) - b\_vector;

std::cout << std::endl << "Невязка r = Ax - b:" << std::endl << "( " << std::setw(5);

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

std::cout << std::setw(14) << r[i][0] << std::setw(14);

}

std::cout << std::setw(5) << ")" << std::endl;

std::cout << std::endl << "Норма невязки r = " << first\_matrix\_norm(r) << std::endl;

return 0;

}

1. **Результат и его анализ.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x = (7.0011 | 3.9999 | -6.0003 | 2.9998 | -2.0007) |

Количество итераций: 65

Невязка r = Ax - b:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (-2.45592e-05 | -1.66302e-06 | -6.71944e-06 | -2.94597e-06 | 9.22789e-06) |

Норма невязки r = 2.455922e-05 (первая норма)

Экономичность:

Сходится со скоростью геометрической прогрессии. При данном методе построения матрицы её норма получилась больше единицы, поэтому чтобы определить знаменатель этой прогрессии потребуются формулы использующие собственные значения, которые не приводились на лекциях.

Точность:

Метод является итерационным и даёт наперёд заданную точность. Ценой большей точности является большее количество итераций и соответственно большее количество операций. Однако, как и при методах Гаусса и отражений мы не можем получить точность выше без изменения типа хранимых данных.