Санкт-Петербургский политехнический университет Петра Великого
Физико-механический институт
Высшая школа прикладной математики и вычислительной физики

Направление подготовки
"01.03.02. Прикладная математика и информатика"

Дисциплина "Численные методы"

Отчет по лабораторной работе №3
"Решение СЛАУ итерационными методами. Метод простых итераций"

Работу выполнил:
Иванова А.С.
Группа:
5030102/00002
Преподаватель:
Курц В.В.

Санкт-Петербург
2021

Содержание

1. Формулировка задачи 3

2. Алгоритм метода и условия его применимости 3
   2.1. Условия применимости 3
   2.2. Алгоритм метода 3

3. Предварительный анализ задачи 3

4. Проверка условий применимости метода 3

5. Тестовый пример с детальными расчетами для задачи малой размерности 4

6. Перечень контрольных тестов для иллюстрации метода 5
   6.1. Исследование точности решения при определителе, близком к нулю 5
   6.2. Сравнение решения одинаковых СЛАУ прямыми и итерационными методами 5

7. Модульная структура программы 5

8. Численный анализ решения задачи 10
   8.1. Исследование точности решения при определителе, близком к нулю 10
      8.1.1. Генерация одной матрицы и изменение ее с помощью параметра, уменьшающего определитель 10
   8.2. Сравнение решения одинаковых СЛАУ прямыми и итерационными методами 11

9. Краткие выводы 11

1. Формулировка задачи

Решить СЛАУ, используя метод простых итераций. При решении данной задачи будет использоваться одна из модификаций метода простых итераций, а именно метод Якоби. Исследовать точность решения, когда определитель матрицы близок к нулю,ввести один параметр, который уменьшает определитель. обусловленности. Сравнить решение одинаковых СЛАУ прямыми и итерационными методами при одинаковом объёме вычислений.

Уравнение в матричном виде

$Ax = b$, где A - матрица системы, b - столбец свободных членов, x - вектор-столбец неизвестных (который нужно найти)

2. Алгоритм метода и условия его применимости

2.1. Условия применимости

Матрица системы должна быть невырожденной (определитель матрицы не должен равняться 0)

Матрица системы должна обладать свойством диагонального преобладания по строкам

2.2. Алгоритм метода

Дана СЛАУ $Ax = b$

Необходимо привести ее к виду, удобному для итераций $x = Cx + g$

$x^{(k+1)} = Cx^{(k)} + g$

Метод является стационарным, следовательно матрица C и столбец g постоянны.

$Ax = b <=> -\alpha(Ax - b), \alpha \neq 0 <=> Bx = Bx - \alpha(Ax - b), \det B \neq 0$

$x^{(k+1)} = x^{(k)} - \alpha B^{-1}(Ax - b) <=> x = (E - \alpha B^{-1}A)x + \alpha B^{-1}b$

$B\frac{x^{(k+1)} - x^{(k)}}{\alpha} + Ax^{(k)} = b$

$\alpha = 1, B = D = diag(A)$

$x^{(k+1)} = x^{(k)} - D^{-1}(Ax^{(k)} - b) = (E - D^{-1}A)x^{(k)} + D^{-1}b$

Таким образом:

$E - D^{-1}A = C$

$D^{-1}b = g$

Тогда если i=j, то $c_{ij} = 0$, если нет, то $c_{ij} = -a_{ij}/a_{ii}$

Критерий остановки итерационного процесса

$||x^{(k)} - x^{(k-1)}|| < \frac{1 - ||C||}{||C||}\epsilon$

3. Предварительный анализ задачи

Матрицы генерируются в системе MATLAB таким образом, что точно выполняется условие диагонального преобладания по строкам (элементы на главной диагонали матрицы системы на несколько порядков больше остальных элементов).

4. Проверка условий применимости метода

Матрицы генерируются в системе MATLAB таким образом, что точно выполняется условие диагонального преобладания по строкам (элементы на главной диагонали матрицы системы на несколько порядков больше остальных элементов).

5. Тестовый пример с детальными расчетами для задачи малой размерности

Решим систему уравнений с точностью до 10e-4

$$\begin{cases} 10x_1 + x_2 - x_3 & = 11 \\ x_1 + 10x_2 - x_3 & = 10 \\ -x_1 + x_2 + 10x_3 & = 10 \end{cases}$$

Найдем определитель матрицы системы, чтобы убедиться, что она невырожденная

$$\det A = \begin{vmatrix} 10 & 1 & -1 \\ 1 & 10 & -1 \\ -1 & 1 & 10 \end{vmatrix} = 990$$

Наглядно видно, что данная матрица обладает свойством диагонального преобладания по строкам

Точное решение:

$$x = \begin{pmatrix} \frac{1091}{990} \\ \frac{109}{110} \\ \frac{91}{90} \end{pmatrix}$$

Данная система имеет единственное решение.

Приводим СЛАУ к удобному виду для итерации:

$$\begin{cases} x_1 & = -0,1x_2 + 0,1x_3 + 1,1 \\ x_2 & = -0,1x_1 + 0,1x_3 + 1 \\ x_3 & = 0,1x_1 - 0,1x_2 + 1 \end{cases}$$

Выберем начальное приближение

$$x^{(0)} = \begin{pmatrix} 1,1 \\ 1 \\ 1 \end{pmatrix}$$

$$\begin{cases} x_1^{(1)} & = -0,1 * 1 + 0,1 * 1 + 1,1 \\ x_2^{(1)} & = -0,1 * 1,1 + 0,1 + 1 \\ x_3^{(1)} & = 0,1 * 1,1 - 0,1 * 1 + 1 \end{cases}$$

Таким образом первое приближение:

$$x^{(1)} = \begin{pmatrix} 1,1 \\ 0,99 \\ 1,01 \end{pmatrix}$$

Далее производятся аналогичные вычисления для следующих итераций

$$x^{(2)} = \begin{pmatrix} 1,102 \\ 0,991 \\ 1,011 \end{pmatrix}$$

$$x^{(3)} = \begin{pmatrix} 1,102 \\ 0,9909 \\ 1,0111 \end{pmatrix}$$

$$x^{(4)} = \begin{pmatrix} 1,10202 \\ 0,99091 \\ 1,01111 \end{pmatrix}$$

Проверим, достиглась ли нужная точность
$||x^{(4)} - x^{(3)}||_\infty = 2 * 10^{-5}$
$||C||_\infty = 0.2$
$\frac{1-||C||}{||C||} * 0,0001 = 0.0004$
$2 * 10^{-5} < 0,0004$
Нужная точность достигнута

6. Перечень контрольных тестов для иллюстрации метода

6.1. Исследование точности решения при определителе, близком к нулю

Точное решение СЛАУ везде одинаковое (столбец единиц). Размерность матрицы фиксирована (15 на 15)

Выявляется две зависимости: относительная погрешность от определителя при фиксированном количестве итераций (в данном случае 3) и количества итераций, необходимого для достижения точности 10e-15 от определителя.

Ожидается, что при меньшем определителе погрешность будет больше и что чем определитель матрицы ближе к нулю, тем большее количество итераций требуется для нахождения корней с заданной точностью.

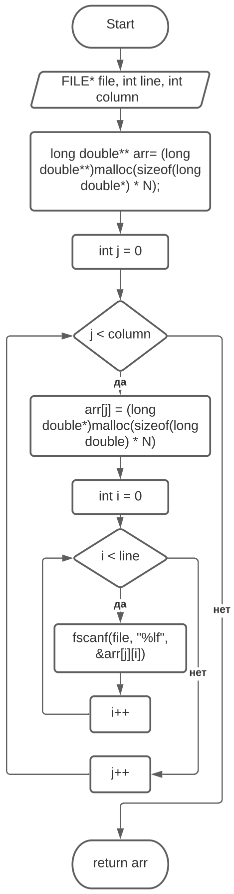
Генерируется матрица следующим образом: элементы на главной диагонали как 100*rand(), остальные элементы как rand()*0.0000001. Задается массив для изменения матрицы [1 0.5 0.1 0.05 0.01 0.005 0.001 0.0005 0.0001 0.00005 0.00001]. Все диагональные элементы умножаются на элементы данного массива, таким образом получается 11 матриц, в каждой последующей определитель на несколько порядков меньше, чем у предыдущей. Вычисляются определители этих матриц.

6.2. Сравнение решения одинаковых СЛАУ прямыми и итерационными методами

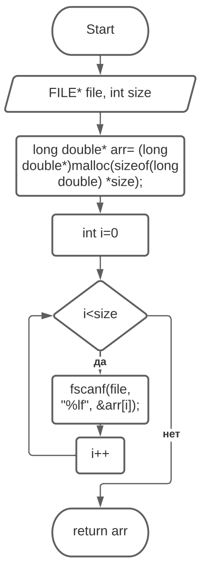
Генерируется 99 матриц с разными размерами с диагональным преобладанием. Элементы на главной диагонали генерируются как 10000*rand(), остальные элементы как rand() * 0.00001. Размер матрицы меняется в цикле от 10 до 500 с шагом 5. Сравниваются решения прямым методом вращений и методом простых итераций (модификацией методом Якоби). Ожидается, что для решения систем с большой размерностью итерационный метод якоби покажет лучшие результаты, чем прямой метод вращений. Для метода простых итераций выбрана точность 10e-15, примерно с такой точностью вычисляются корни методом вращений при хорошо обусловленной матрице.

7. Модульная структура программы

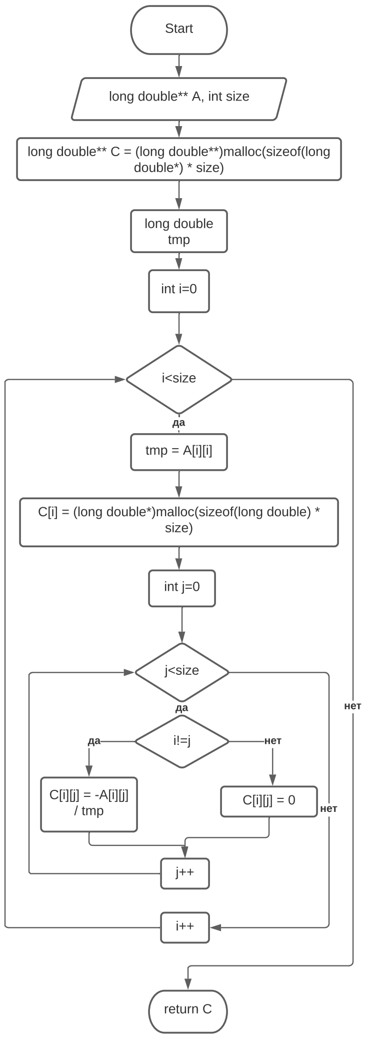
Функция long double** ArrayRead(FILE* file, int line, int column)



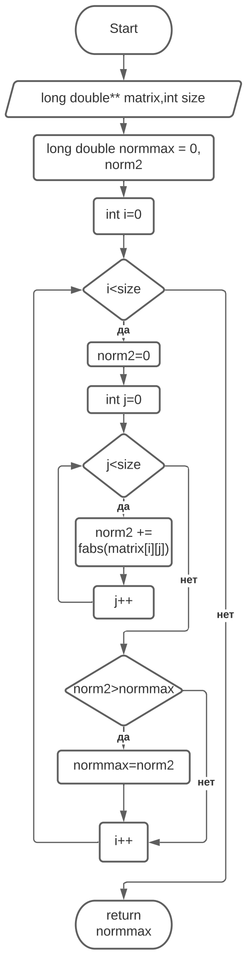
Функция long double* ColumnRead(FILE* file, int size)



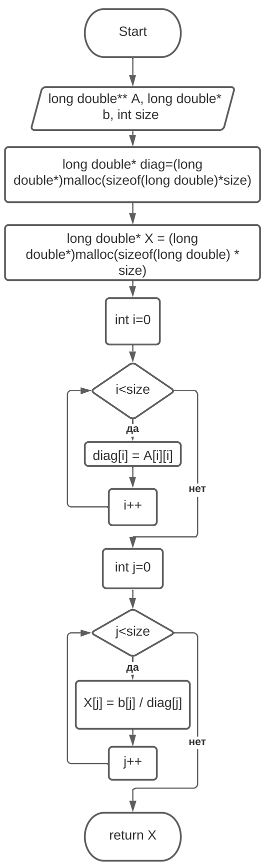
Функция long double** CreatePrecondMatrix(long double** A, int size)



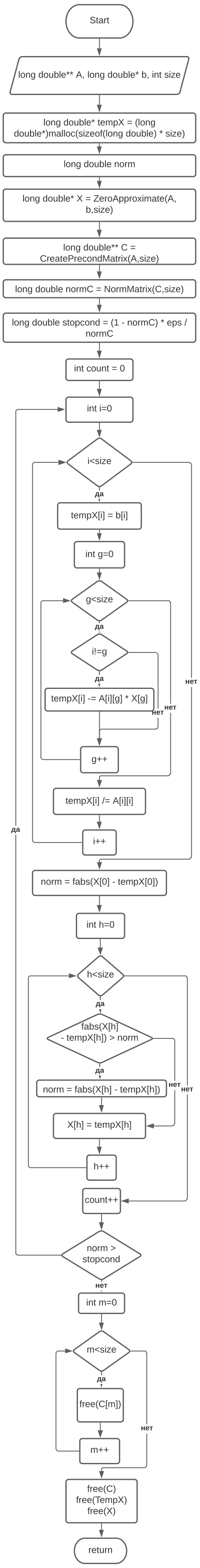
Функция long double NormMatrix(long double** matrix,int size)



Функция long double* ZeroApproximate(long double** A, long double* b, int size)

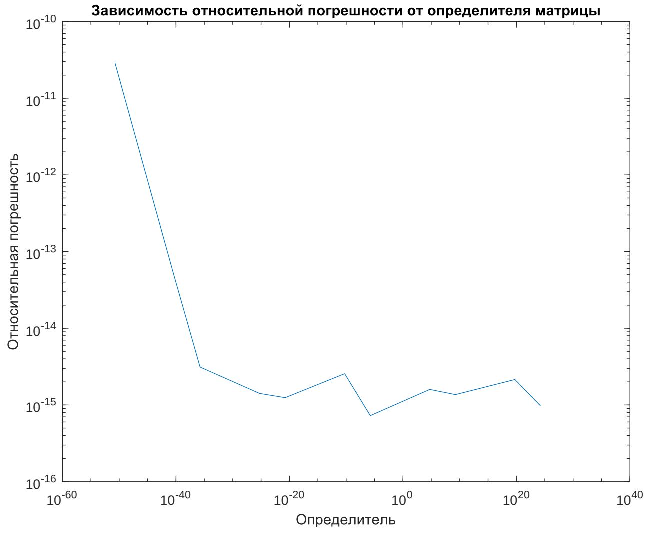


Функция void Jacobi(long double** A, long double* b, int size)

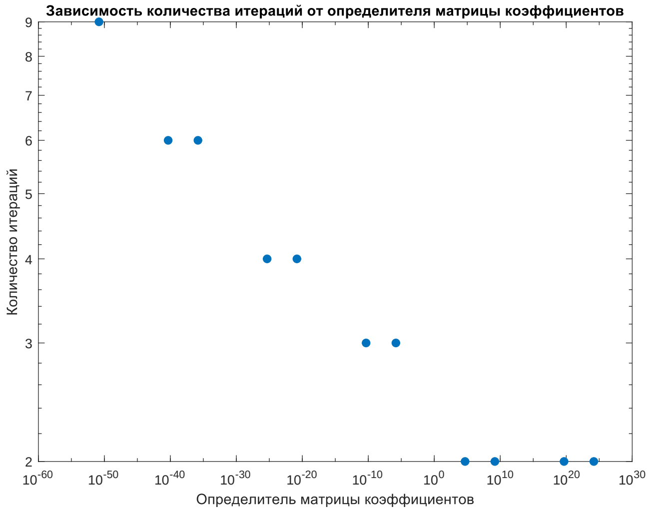


8. Численный анализ решения задачи

8.1. Исследование точности решения при определителе, близком к нулю

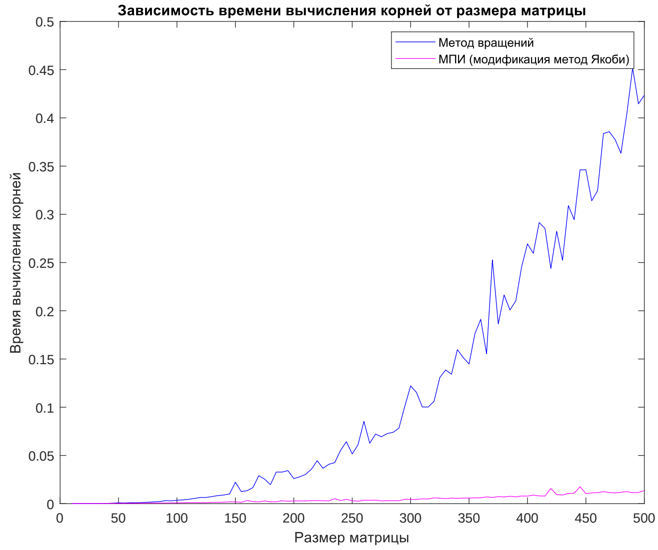


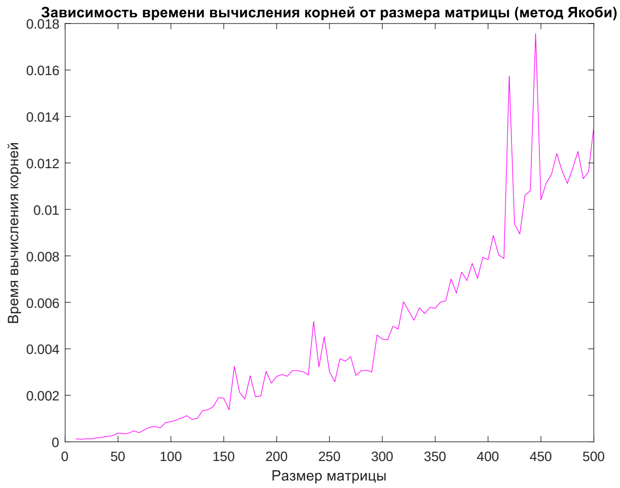
Данный график отображает зависимость относительной погрешности вычисления за три итерации от определителя матрицы. Из графика видно, что чем определитель ближе к нулю, тем больше погрешность.



Данный график отображает зависимость количества итераций, необходимого для достижения заданной точности вычислений от определителя матрицы коэффициентов. Зависимость напоминает линейную в логарифмическом масштабе, можно сделать вывод, что при приближении определителя к нулю, необходимое количество итераций, необходимое для вычисления корней увеличивается.

8.2. Сравнение решения одинаковых СЛАУ прямыми и итерационными методами





Из данных графиков видно, что с ростом размерности системы время, необходимое для вычисления корней при решении прямым методом вращений растет гораздо быстрее, чем при решении методом простых итераций. Для метода вращений зависимость близка к кубической, а для метода простых итераций близка к квадратичной.

9. Краткие выводы

Была решена задача нахождения корней СЛАУ одной из модификаций метода простых итераций - методом Якоби.

Была исследована точность решения при определителе матрицы системы, близком к нулю.

Было произведено сравнение итерационного метода Якоби и прямого метода вращений, с помощью чего было подтверждено, что для систем с большой размерностью лучше использовать итерационные методы нахождения корней.