

Дисциплина: Численные методы  
Лабораторное задание №1

Отчет

Тема: Решение систем линейных уравнений с разреженными матрицами  
специального вида

Выполнили:  
студентки 3 курса 62 группы  
Пахомова П.В., Киселёва О.А.

Проверила:  
старший преподаватель Фролова О.А.

## 1. Постановка задачи

Решение системы линейных уравнений с разреженными матрицами специального вида

Задача № 1 состоит в разработке алгоритма для решения системы уравнений с матрицей специального вида. Всего имеется 11 вариантов задания, и каждый из них имеет уникальный портрет матрицы системы. Однако все матрицы систем имеют общую структуру, которая представляет собой матрицу с тремя диагоналями и "испорченными" элементами: двумя столбцами, двумя строками или одним столбцом и одной строкой.

Во всех вариантах задания исходные системы уравнений определяются шестью векторами:  $a$ ,  $b$ ,  $c$ ,  $f$ ,  $p$ ,  $q$ . Векторы  $a$ ,  $b$ ,  $c$  содержат значения трех диагоналей матрицы, вектор  $f$  - вектор правой части системы, а векторы  $p$  и  $q$  - векторы для строк или столбцов, которые "портят" матрицы системы.

Цель данной задачи также включает оптимизацию использования памяти и улучшение эффективности решения системы с разреженными матрицами специального вида.

Для варианта 13 требуется написать алгоритм решения системы с матрицей специального вида. В данном варианте первый и последний столбцы "испорчены", главная диагональ, диагонали над главной и под главной заполнены числами, а остальная область заполнена нулями. Также необходимо вычислить среднюю относительную погрешность системы и среднее значение оценки точности.

При записи погрешностей используются 2–3 значащие цифры, не более.

[illegible]

**b,a,c** - главная диагональ, диагональ над ней и под ней.

**p1, pn** - первый и последние столбцы (испорченные)

## 2. Теоретическая часть

Основой метода решения системы линейных уравнений с матрицей специального вида, является прямой ход метода Гаусса.

Подход состоит из следующих шагов:

1. Прямой ход: В данном шаге вы преобразуете исходную систему уравнений в эквивалентную систему, при которой все элементы под главной диагональю матрицы обнуляются. Для этого вы применяете элементарные преобразования строк, такие как сложение строк или замена строк.

Обмен строк: Если на диагонали матрицы встречается ноль, вы меняете строки местами. Это может потребовать дополнительных проверок и обменов строк во время прямого хода.

\*После завершения прямого хода, система уравнений принимает треугольную форму. Вы можете использовать метод обратного хода для вычисления решения системы. Однако, в моём случае я не выполняю обратный ход, поэтому решение системы основано на матрице после выполнения прямого хода.

2. Вычисление решений через матрицу: После прямого хода, полученная матрица представляет систему уравнений в виде упрощенной треугольной матрицы. Вычисление решений системы и формулы расписаны в алгоритме (ШАГ 5), где  $x_1, \dots, x_n$  вычисляются с конца.

Подход является специфичным для матриц специального вида и может не применяться к общим случаям систем линейных уравнений.

### 3. Алгоритм

#### Условия для работы алгоритма:

- первый элемент не может быть 0
- если на главной диагонали 0, то справа и над ним число отлично от 0

Из-за специфики моей задачи не использую обратный ход.

Данный алгоритм является методом решения системы линейных уравнений.

#### Прямой ход:

Каждый следующий столбец для расчётов храниться в  $p1$

Поочередно для каждого диагонального элемента матрицы(начинаем с  $j=1$ ), начиная со следующей строки ( $i=j$ ), выполняются следующие шаги:

#### Шаг 1

т.к.  $b$  и  $c$  меняются при решении,

- Присваивается  $p1[j-1]$  значение  $b[j-1]$ .
- Присваивается  $p1[j]$  значение  $c[j-1]$ .

#### Шаг 2

- Устанавливаются флаги  $ab$  и  $ac$ , указывающие на то, что  $b$  и  $c$  ещё не менялись из-за  $a$ .
- Для каждого элемента  $j$ , начиная с  $i$ -й строки и до последней строки матрицы, выполняются следующие шаги:
  - Если  $p1[j-1]$ (является элементом главной диагонали) равно нулю, происходит обмен строками  $j-1$  и  $j$  для  $p1$ ,  $pn$ ,  $a$  и  $b$ ,

Например:

				$c$	$b  p1$	$a$	$x$			$pn$	$rhs$
$j-1:$	0	0	0	0	0	*	0	0	0	*	*
$j:$	0	0	0	0	*	*	*	0	0	*	*

$p1[j - 1]$  меняется с  $p1[j]$

$rhs[j - 1]$  меняется с  $rhs[j]$

$a[j-1]$  меняется с  $b[j]$

И если возможно то

$x$  сохраняет  $a[j]$  и  $a[j] = 0$

а затем продолжается следующая итерация цикла, потому что элемент ниже ноль.

Это делается, чтобы гарантировать, что элемент на главной диагонали не равен нулю.

- Вычисляется множитель:  $multiplier = p1[i] / p1[j-1]$ .
- Рассчитывается следующий элемент строки  $p1[i] = (-1) * (a[j-1] * multiplier)$ .
- Если  $b$  ещё не затрагивался, обновляется  $b[i] -= a[j-1] * multiplier$ , а затем помечаем  $ab$ .
- В противном случае, если  $c$  ещё не затрагивался, обновляется  $c[i-1] -= a[j-1] * multiplier$ , а затем помечаем  $ac$ .
- Рассчитывается следующий элемент строки  $pn[i] -= pn[j-1] * multiplier$  и  $rhs[i] -= rhs[j-1] * multiplier$ .

### Шаг 3

Обновления для последних значений:

- Устанавливается  $p1[SIZE-1]$  равным  $pn[SIZE-1]$ .
- Устанавливается  $a[SIZE-2]$  равным  $pn[SIZE-2]$ .

### Шаг 4

Происходит проверка возможности решения (**isCorrectToBeSolved**).

Если это возможно, выполняются следующие шаги иначе решение невозможно

### Шаг 5 (Вычисление решения $x_1, \dots, x_n$ )

- Вычисляется  $x_n$  на основе значения последнего элемента диагонали (если правая и левая части равны нулю,  $x$  присваивается любое значение, по умолчанию -999, если пользователь не задал иное).

- Из-за того что у нас есть  $x_n$ , можно исключить последний столбец.

Для каждой строки  $i$  столбца после  $=$ , вычитается новое значение  $rhs[i] -= pn[i] * x[SIZE - 1]$ .

- Вычисляется  $x_{n-1}$  на основе значения предпоследнего элемента диагонали (если правая и левая части равны нулю,  $x$  присваивается любое значение, по умолчанию -999, если пользователь не задал иное).

- Начиная с предпоследней строки и до нулевой строки,  $x_i$  вычисляются как  
Если  $x_i$  хранит коэффициент из Шага 2, то учитываем его для вычислений  
 $rhs[i] -= x[i] * x[i + 2]$ ;

Далее находим  $x[i]$

$a[i] *= x[i+1]$

$x[i] = (rhs[i] - a[i]) / p1[i]$ .

## Шаг 6

Устанавливается флаг `solved` и проверяется корректность полученного решения (`checkSolution`).

Проверка возможности решения (`isCorrectToBeSolved`) из Шага 4

Данная функция проверяет, возможно ли найти решение системы линейных уравнений.

- Проверка условия, что если  $|rhs[SIZE - 1]|$  превышает  $E$  и  $p1[SIZE - 1]$  равно нулю,

Это означает, что если последний элемент вектора правой части не равен нулю, но соответствующий элемент на главной диагонали матрицы равен нулю, то система не имеет решения.

Проверяется корректность полученного решения из **Шага 6**  
(**checkSolution**)

В данной проверке выполняются следующие **шаги**:

- Создание временного массива testS размером SIZE равного testRhs.
- Создается переменная result.
- Итерация по элементам массива testS от 0 до SIZE-1:
  - Для каждой строки  $i$  выполняется проверка:
  - все  $x$  умножаются на соответствующие коэффициенты начальных массивов  $p1, c, b, a, p_n$  складываются между собой и вычитаются из  $rhs[i]$ .
- Если на каком-то этапе разность оказалось выше  $E$ , алгоритм завершает свою работу и говорит о неверности решения.
- Если все итерации успешно прошли алгоритм завершает свою работу и говорит о верности решения.

\*Для расчёта средней оценки точности (**E1**) и средней относительной погрешности системы (**E2**) используется массивы  $e1Rhs$ ,  $e2Rhs$  и  $e2X$

До начала решения генерируется вектор правой части системы  $e1Rhs$  для решения  $(1, \dots, 1)$

и вектор правой части системы  $e2Rhs$  для решения  $e2X$  сгенерированного в заданном диапазоне.

Параллельно с основным решением изначальной системы производиться такое же решение для  $e1Rhs$ ,  $e2Rhs$ , решения представляют собой массивы  $eSolution1X$  и  $eSolution2X$

Для расчёта погрешности перебираем все решения:

$q$  – неотрицательное число близкое к нулю, по умолчанию  $10^{(-10)}$

Для  $eSolution1X[i]$ :

a. Вычисление разницы между  $eSolution1X[i]$  и 1:  $\delta1 = eSolution1X[i] - 1$

b. Если  $\delta1$  отрицательна, то погрешность  $e1$  вычисляется следующим

образом:  $e1 = (-1 * \delta1) + 1$  c. Если  $\delta1$  неотрицательна, то погрешность  $e1$  вычисляется следующим образом:  $e1 = \delta1 - 1$



Для eSolution2X[i]:

$$\delta_x = \max_i \delta x_i,$$

$$\delta x_i = \begin{cases} \left| \frac{x_i - x_i^*}{x_i^*} \right|, & \text{если } |x_i^*| > q \\ |x_i - x_i^*|, & \text{если } |x_i^*| \leq q, \end{cases}$$

а. Вычисление разницы между eSolution2X[i] и e2X[i]:

delta2 = (eSolution2X[i] - e2X[i]) // обозначение

б. Вычисление погрешности e2:

Если delta2 отрицательна: e2 = (-1 \* delta2)

Иначе: e2 = delta2

Если  $|e2X[i]| \leq q$ , то: e2

Если  $|e2X[i]| > q$ , то:  $e2 = e2 / |e2X[i]|$

Для E1 и E2 находим максимальную по модулю погрешности соответственно.

#### 4. Тестирование

\*средние значения рассчитываются по результатам 10 удачных решений

Результаты тестирования в виде таблицы:

№ теста	Размерность системы	Диапазон значе- ний элементов матрицы	Средняя относитель- ная погрешность системы	Среднее значе- ние оценки точ- ности
1	10	[-10:10]	$8.78e-14$	$1.96e-14$
2	10	[-100:100]	$4.25e-14$	$2.2e-14$
3	10	[-1000:1000]	$2.95e-13$	$7.9e-14$
4	100	[-10:10]	$2.58e-11$	$2.1e-12$
5	100	[-100:100]	$1.37e-11$	$5.03e-12$
6	100	[-1000:1000]	$2.26e-11$	$2.68e-12$
7	1000	[-10:10]	$4.7e-11$	$5.28e-11$
8	1000	[-100:100]	$6.71e-11$	$5.77e-12$
9	1000	[-1000:1000]	$1.1e-10$	$2.43e-11$

Failure rate 1%