ПРИКЛАДНАЯ ИНФОРМАТИКА

Лектор Иванов Сергей Евгеньевич

Лабораторная работа 3. Алгоритмы решения систем линейных алгебраических уравнений

Цель работы

Реализовать алгоритмы решения систем линейных алгебраических уравнений СЛАУ.

Теоретические основы

СЛАУ n-го порядка

$$\sum_{j=1}^{n} a_{ij}x_j = b_i; \quad i=\overline{1,n} \qquad \begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \cdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}$$

В векторной форме: $A\bar{x} = \bar{b}$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & \cdots & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \qquad \bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \bar{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Определитель (детерминант) матрицы A n-го порядка:

$$|A| = D = \det A = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & \cdots & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix} = \sum (-1)^k a_{1\alpha}\, a_{2\beta} \cdots\, a_{n\omega}$$

Здесь индексы α, β, ..., ω пробегают все возможные $n!$ перестановок номеров 1, 2, ..., n; k – число инверсий в данной перестановке.

<u>Метод Гаусса</u>

Последовательное исключение неизвестных, приводящее исходную систему к треугольному виду, в котором все коэффициенты ниже главной диагонали равны нулю. Процесс приведения матрицы коэффициентов к треугольному виду называется прямым ходом метода Гаусса.

Возьмем первое уравнение системы и вычтем его из второго, предварительно умножив на такое число, чтобы уничтожился коэффициент при x_1. Затем таким же образом вычтем первое уравнение из третьего, четвертого и т.д.

Получим новую систему, в которой первое уравнение осталось неизменным, а остальные больше не содержат член с x_1, т.е. исключили все коэффициенты первого столбца матрицы, лежащие ниже главной диагонали:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ 0x_1 + a_{22}^{(2)}x_2 + \ldots + a_{2n}^{(2)}x_n = b_2^{(2)} \\ \cdots \\ 0x_1 + a_{n2}^{(2)}x_2 + \ldots + a_{nn}^{(2)}x_n = b_n^{(2)} \end{cases} \qquad A^{(1)} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \cdots & \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}$$

При помощи второго уравнения исключаются коэффициенты при x_2 из третьего, четвертого и последующих уравнений. Продолжая этот процесс, можно исключить из матрицы все коэффициенты, лежащие ниже главной диагонали. Исключение неизвестных повторяется до тех пор, пока в левой части последнего n-го уравнения не останется одно неизвестное x_n

$$a_{nn}^{(n)}x_n = b_n^{(n)}$$

На каждом шаге новые значения коэффициентов определяются через значения на предыдущем шаге согласно

$$a_{ml}^{(k+1)} = a_{ml}^{(k)} - a_{kl}^{(k)} \frac{a_{mk}^{(k)}}{a_{kk}^{(k)}} \qquad b_m^{(k+1)} = b_m^{(k)} - b_k^{(k)} \frac{a_{mk}^{(k)}}{a_{kk}^{(k)}} \qquad k = \overline{1, n-1}; \quad l = \overline{k, n}$$

где m – номер уравнения, из которого исключается x_k; k – номер неизвестного, которое исключается из оставшихся (n–k) уравнений (номер столбца, из которого исключаются элементы); l – номер столбца исходной матрицы.

Обратный ход метода Гаусса состоит в последовательном вычислении x_n, x_{n-1}, ..., x_1 по алгоритму

$$x_k = \frac{1}{a_{kk}^{(k)}} \left[b_k^{(k)} - \sum_{i=k+1}^{n} a_{ki}^{(k)} x_i \right], \quad k = \overline{n, 1}$$

Во время счета необходимо следить, чтобы диагональный элемент $a_{kk}^{(k)} \neq 0$

В противном случае прибегают к перестановке строк матрицы и продолжают расчет.

Если элемент на главной диагонали мал, то эта строка умножается на большие числа, что приводит к значительным ошибкам округления при вычитаниях. Чтобы избежать этого, каждый цикл всегда начинают с перестановки строк. Среди элементов столбца находят главный, т. е. наибольший по модулю в k-м столбце, и перестановкой строк переводят его на главную диагональ, после чего делают исключения.

<u>Метод прогонки</u>

Модификация метода Гаусса, применяемая к системам с *матрицей трехдиагонального типа* (часто встречаются при решении краевых задач для дифференциальных уравнений второго порядка). Каноническая форма:

$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i$;

$i = 1\ldots n$; $a_1 = c_n = 0$

В развернутом виде:

$$
\begin{aligned}
&b_1x_1 + c_1x_2 &&= d_1;\\
&a_2x_1 + b_2x_2 + c_2x_3 &&= d_2;\\
&\quad a_3x_2 + b_3x_3 + c_3x_4 &&= d_3;\\
&\qquad \cdots\\
&\qquad a_{n-1}x_{n-2} + b_{n-1}x_{n-1} + c_{n-1}x_n &&= d_{n-1};\\
&\qquad\qquad a_nx_{n-1} + b_nx_n &&= d_n\,.
\end{aligned}
$$

При этом, как правило, все коэффициенты $b_i \neq 0$.

На этапе **прямого хода** каждое неизвестное x_i выражается через x_{i+1}

$x_i = A_i \cdot x_{i+1} + B_i$ для $i = 1,2, ..., n-1$,

посредством прогоночных коэффициентов A_i и B_i, которые имеют следующий вид

где $e_i = a_i \cdot A_{i-1} + b_i$ (i=2,3, ..., n–1).

Расчет начинается с вычисления коэффициентов

$$A_1 = -\frac{c_1}{b_1}; \qquad B_1 = \frac{d_1}{b_1}$$

На этапе **обратного хода** из последнего уравнения системы при $i = n-1$

$$x_n = \frac{d_n - a_n B_{n-1}}{b_n + a_n A_{n-1}}$$

Далее посредством прогоночных коэффициентов последовательно вычисляем x_{n-1}, x_{n-2}, ..., x_1.

При условии $|\, b_i \,| \geq |\, a_i \,| + |\, c_i \,|$, деление на «0» исключается, и система имеет единственное решение.

Содержание работы

Реализовать средствами ООП (C#, C++) программу решения задачи для систем алгебраических уравнений.

1. Решить систему линейных алгебраических уравнений СЛАУ методом Гаусса. Привести результат решения, а также вид матрицы системы и вектора правой части после завершения прямого хода метода. Примеры СЛАУ для решения.

1. $\begin{cases} 4X_1 + X_2 + X_3 + 2X_4 = 2 \\ X_1 + 3X_2 + 2X_3 - X_4 = 2 \\ 2X_1 - X_2 + 5X_3 + 3X_4 = -1 \\ 4X_1 + 5X_2 + 4X_3 - 4X_4 = 8 \end{cases}$

2. $\begin{cases} X_1 - X_2 + 2X_3 - X_4 = 1 \\ 2X_1 + 3X_3 + X_4 = 4 \\ X_1 + X_2 + 3X_3 - X_4 = 2 \\ 2X_1 + X_2 + 5X_3 - 2X_4 = 3 \end{cases}$

3. $\begin{cases} X_1 + 2X_2 - X_3 - X_4 = 0 \\ 2X_1 + 3X_2 - X_3 + X_4 = 3 \\ 2X_1 + 5X_2 + 2X_3 + X_4 = 3 \\ 3X_1 + 5X_2 + X_3 + 2X_4 = 5 \end{cases}$

7. $\begin{cases} 2X_1 + X_2 + X_3 + X_4 = 1 \\ 2X_1 + 2X_2 + 2X_3 + 3X_4 = -1 \\ 4X_1 + 3X_2 + 3X_3 + 3X_4 = 1 \\ 6X_1 + 4X_2 + 5X_3 + 2X_4 = 4 \end{cases}$

8. $\begin{cases} X_1 + 2X_2 + 2X_3 + X_4 = 0 \\ 2X_1 + 3X_2 + 3X_3 + 2X_4 = 0 \\ 3X_1 + 5X_2 + 4X_3 + 2X_4 = 1 \\ 4X_1 + 7X_2 + 6X_3 + 2X_4 = 2 \end{cases}$

9. $\begin{cases} 2X_1 + X_2 + X_3 + X_4 = 2 \\ 2X_1 + 2X_2 + 2X_3 + 3X_4 = 1 \\ 2X_1 + 2X_2 + 3X_3 + 4X_4 = 0 \\ 2X_1 + 2X_2 + 3X_3 + 5X_4 = -1 \end{cases}$

2. Решить СЛАУ методом прогонки и привести результат решения. Примеры СЛАУ для решения.

1. $\begin{cases} 2X_1 + 2X_2 = 1 \\ -X_1 + 2X_2 - 0{,}5X_3 = 0 \\ X_2 - 3X_3 - X_4 = 2 \\ X_3 + 2X_4 = 2 \end{cases}$

2. $\begin{cases} 3X_1 + X_2 = 5 \\ X_1 + 2X_2 + X_3 = 6 \\ 3X_2 + 9X_3 + 6X_4 = 25 \\ 2X_3 + 4X_4 = 5 \end{cases}$

3. $\begin{cases} 4X_1 + X_2 = 5 \\ X_1 + 3X_2 - 2{,}5X_3 = 2 \\ 1{,}5X_2 - 5X_3 + X_4 = 1 \\ 2X_3 + 4X_4 = 7 \end{cases}$

4. $\begin{cases} 7X_1 - 2X_2 = 5 \\ -2X_1 + 12X_2 + 4X_3 = 8 \\ X_2 - 6X_3 + X_4 = 2 \\ 3X_3 + 5X_4 = 4 \end{cases}$

5. $\begin{cases} 1{,}5X_1 + 0{,}5X_2 = 3{,}2 \\ -X_1 + 2X_2 - 0{,}4X_3 = -1 \\ 2{,}5X_2 + 5X_3 - 2X_4 = 4 \\ X_3 + 3X_4 = 3 \end{cases}$

6. $\begin{cases} 3X_1 + X_2 = 5 \\ X_1 + 4X_2 - X_3 = 3 \\ -X_2 + 5X_3 + X_4 = 12 \\ X_3 + 2X_4 = 6 \end{cases}$

Отчётность по работе

После выполнения лабораторной работы обучаемый представляет отчет. Отчет записывается в личную папку студента на сервере и также отправляется на почту преподавателя.

Отчёт должен содержать:

1. Титульный лист.
2. Название и цель работы.
3. Результаты. Исходный код (C#, C++), блок-схема алгоритма и тестовый расчет примера.
4. Выводы, что выполнено в работе.