

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МЕТОДА ПРАВОЙ ПРОГОНКИ ПРИ РЕШЕНИИ ЛИНЕЙНЫХ СИСТЕМ УРАВНЕНИЙ

Химич Анна Васильевна

студент, Брянский государственный университет имени академика И. Г. Петровского, РФ, г. Брянск

К решению систем алгебраических и трансцендентных уравнений так или иначе сводится три четверти задач прикладной математики, при этом большую часть из них составляют линейные алгебраические системы, имеющие единственное решение.

Решение задач, связанных с экономикой, физикой, химией и другими науками, в большинстве случаев связано с решением систем линейных уравнений (СЛУ). Их решение является одной из ключевых задач вычислительной линейной алгебры, а также элементарным шагом соответствующего алгоритма при решении различных задач численных методов. Так как задача решения СЛУ является очень важной и практически значимой, многие математики до сих пор обращают на это особое внимание и пытаются разработать более удобные методы решения.

Для решения линейных систем достаточно хорошо известны методы Крамера и Гаусса, а также разработана общая теория. Но по сей день на практике возникает множество различных проблем при решении задач, сводящихся к решению систем линейных уравнений.

Данные методы имеют свои недостатки. Например, недостатком метода Крамера является то, что он слишком трудоёмкий и пользоваться им практически невозможно. С объёмом таких громоздких вычислений не может справиться даже компьютер.

Метод Гаусса менее трудоёмкий. Тот факт, что количество арифметических операций, необходимых для решения линейной системы n -го порядка методом Гаусса, значительно меньше, позволяет использовать данный метод достаточно широко. Но при решении систем высокого порядка применение метода Гаусса становится проблематичным.

На практике нередко возникают плохо обусловленные системы, для которых характерно быстрое нарастание вычислительной погрешности результата, связанное с округлениями, при увеличении порядка системы. Решение таких систем, даже при небольших порядках, вызывает трудности. Существует достаточное количество методов, позволяющих решать такие задачи. Выделяют следующие группы методов: прямые (методы Гаусса, Жордана, квадратного корня и т.п.) и итерационные методы (методы простой итерации, Зейделя, релаксации и т.п.).

Кроме перечисленных методов используются специфические методы прогонки – прямые методы, которые по существу представляют собой модификации метода Гаусса и предназначены для решения систем высокого порядка с сильно разреженными матрицами. Эти матрицы состоят в основном из нулей, а ненулевые элементы в них образуют определенные структуры. Подобные системы часто порождаются разностными и проекционными методами. Применительно к решению краевых задач для дифференциальных уравнений методы прогонки очень эффективны.

Метод правой прогонки используется для решения СЛУ с трёхдиагональными матрицами. Он представляет собой модификацию метода Гаусса. Трёхдиагональная матрица – это матрица, на главной диагонали которой находятся лишь элементы отличные от нуля, а остальные

должны быть равны нулю.

$$\begin{bmatrix} -C_1 & B_1 & 0 & 0 & 0 \\ A_2 & -C_2 & B_2 & 0 & 0 \\ 0 & A_3 - C_3 & B_3 & 0 & 0 \\ 0 & 0 & A_4 - C_4 & B_4 & 0 \\ 0 & 0 & 0 & A_5 - C_5 & 0 \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} -F_1 \\ -F_2 \\ -F_3 \\ -F_4 \\ -F_5 \end{bmatrix}$$

(1)

Рассмотрим СЛУ 5 порядка (рис.1). Отличные от нуля коэффициенты, стоящие на главной диагонали, обозначены буквой С со знаком минус, а коэффициенты, стоящие на двух соседних диагоналях, обозначены буквами А и В соответственно [3]. Достоинством данного способа является значительная экономия памяти ЭВМ, так как она не тратится на бесполезные нули. Для того, чтобы при решении различных других задач матрицы системы имели одинаковый вид, необходимо перед коэффициентами С и правыми частями F поставить знак минус.

Используя описанную систему обозначений, запишем произвольную систему с трёхдиагональной матрицей (М-1)-го порядка в общем виде:

$$\begin{cases} -C_1 y_1 + B_1 y_2 = -F_1, \\ A_m y_{m-1} - C_m y_m + B_m y_{m+1} = -F_m, & m = 2, 3, \dots, M-2, \\ A_{M-1} y_{M-2} - C_{M-1} y_{M-1} = -F_{M-1}. \end{cases}$$

(2)

При М=6 системы (1) и (2) эквивалентны. После использования процедуры прямого хода метода Гаусса к решению системы (2), получится система, матрица которой будет иметь две диагонали: главную и правую. Для того чтобы получить систему с двухдиагональной матрицей, на главной диагонали которой стоят единицы, необходимо каждое уравнение преобразованной системы разделить на диагональный элемент, стоящий в этой строке. Система будет выглядеть следующим образом (при М=6):

$$\begin{pmatrix} 1 & -\alpha_2 & 0 & 0 & 0 \\ 0 & 1 & -\alpha_3 & 0 & 0 \\ 0 & 0 & 1 & -\alpha_4 & 0 \\ 0 & 0 & 0 & 1 & -\alpha_5 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} = \begin{pmatrix} \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{pmatrix}$$

(3)

А рекуррентные формулы обратного хода метода Гаусса примут вид

$$y_m = a_{m+1} \cdot y_{m+1} + \beta_{m+1}, m = M-1, M-2, \dots, 1,$$

(4)

причём $\alpha_M = 0$. (5)

Для определения коэффициентов α_{m+1} и β_{m+1} необходимо заменить в равенстве (4) величину m на $m-1$:

$$y_{m-1} = \alpha_m \cdot y_m + \beta_m, m = M, M-1, \dots, 1$$

(6)

и подставить в (2):

$$\begin{cases} -C_1 y_1 + B_1 y_2 = -F_1, \\ (\alpha_m A_m - C_m) y_m + B_m y_{m+1} = -F_m - A_m \beta_m, m = 2, 3, \dots, M-2, \\ (\alpha_{M-1} A_{M-1} - C_{M-1}) y_{M-1} = (-F_{M-1} - A_{M-1} \beta_{M-1}). \end{cases}$$

В результате получим:

$$\begin{cases} y_1 = \frac{B_1}{C_1} y_2 + \frac{F_1}{C_1}, \\ y_m = \frac{B_m}{(C_m - \alpha_m A_m)} y_{m+1} + \frac{F_m + A_m \beta_m}{(C_m - \alpha_m A_m)}, m = 2, 3, \dots, M-2, \\ y_{M-1} = \frac{(F_{M-1} + A_{M-1} \beta_{M-1})}{(C_{M-1} - \alpha_{M-1} A_{M-1})}. \end{cases}$$

$$\alpha_2 = \frac{B_1}{C_1}, \beta_2 = \frac{F_1}{C_1},$$

После сравнения формул с (4), получим:

$$\begin{cases} \alpha_{m+1} = \frac{B_m}{(C_m - \alpha_m A_m)}, \\ \beta_{m+1} = \frac{F_m + A_m \beta_m}{(C_m - \alpha_m A_m)}, m = 2, 3, \dots, M-2, \end{cases}$$

(7)

$$\alpha_M = 0, \beta_M = \frac{(F_{M-1} + A_{M-1} \beta_{M-1})}{(C_{M-1} - \alpha_{M-1} A_{M-1})}.$$

Вводятся фиктивные величины $\alpha_1, \beta_1, A_1, B_{M-1}$ и y_M .

Пусть $A_1 = 0, B_{M-1} = 0, \alpha_1 = \beta_1 = 0$,

(8)

тогда формулы неизвестных коэффициентов α_m и β_m при всех допустимых значениях m можно будет записать одинаково:

$$\begin{cases} \alpha_{m+1} = \frac{B_m}{(C_m - \alpha_m A_m)}, \\ \beta_{m+1} = \frac{F_m + A_m \beta_m}{(C_m - \alpha_m A_m)}, \end{cases} m = 1, 2, 3, \dots, M-1.$$

(9)

Формулы (8) - (9) дают возможность вычислить все неизвестные коэффициенты α_{m+1} и β_{m+1} . Чтобы получить решения системы (2) используем рекуррентную формулу (4), но сначала необходимо знать значение y_M . Так как $\alpha_M = 0$ (так как $B_{M-1} = 0$), значение y_M можно выбрать любое, например,

$$y_M = 0. \quad (10)$$

После проделанных действий получим алгоритм решения системы (2).

Сначала по формулам (9) с начальными условиями (8) получаются значения

$\alpha_2, \beta_2, \alpha_3, \beta_3, \dots, \alpha_M, \beta_M$. Затем по формуле (4) с начальным условием (10)

получается решение системы: $y_{M-1}, y_{M-2}, \dots, y_1$.

В процессе вычисления величин α_{m+1} и β_{m+1} по формулам (9) происходит деление на величины $(C_m - \alpha_m A_m)$, которые могут обращаться в ноль [3]. В данном случае этот метод не применим. Следовательно, важно знать и заранее проверять условия, при которых можно использовать метод прогонки. Достаточные условия применимости метода прогонки представлены в виде следующей теоремы.

Теорема 1. Если

$$A_m \neq 0, \quad m = 2, 3, \dots, M-1, \quad B_m \neq 0, \quad m = 1, 2, \dots, M-2,$$

$$|C_1| > |B_1|, \quad |C_{M-1}| > |A_{M-1}|, \quad |C_m| \geq |A_m| + |B_m|, \quad m = 2, 3, \dots, M-2,$$

то $|\alpha_m| \leq 1$ для $m = 1, 2, \dots, M$ следовательно данный метод можно использовать [3].

Описанный выше метод прогонки реализован в виде программы, часть кода которой представлена ниже:

```
eps[0]=-A[0][1]/A[0][0];
```

```
et[0]=B[0]/A[0][0];
```

```
for(i=1;i<n;i++){
```

```

z=A[i][i]+A[i][i-1]*eps[i-1];

eps[i]=-A[i][i+1]/z;

et[i]=(B[i]-A[i][i-1]*et[i-1])/z;}

X[n]=(B[n]-A[n][n-1]*et[n-1])/(A[n][n]+A[n][n-1]*eps[n-1]);

for(i=n-1;i>=0;i--)

    X[i]=eps[i]*X[i+1]+et[i];

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

dataGridView3->Rows[j]->Cells[0]->Value=X[j]; } }

```

Программа предназначена для решения систем линейных уравнений методом прогонки и составлена для матриц различного порядка путем добавления и удаления столбцов на форме (рис.1.). При нажатии на кнопку “Добавить столбец”, программа автоматически добавляет новый столбец в таблицу A, а при нажатии кнопки “Удалить столбец” программа, соответственно, удаляет его.

На форме (рис. 1.) помимо вышеназванных текстовых полей имеется также поле “Найти X”. Данное текстовое поле предназначено для нахождения значений столбца X. Процедура ввода осуществляется следующим образом: наводится курсор на поля и вводятся соответствующие значения. Затем нажимается кнопка “Найти X”, и в полях столбца X появляются значения

вектора **X**.

Для того чтобы начать работу с программой, представленной на рисунке (рис. 1.), следует запустить программу, далее в поля таблицы A ввести элементы матрицы. Благодаря встроенным кнопкам мы можем изменять количество столбцов таблицы. В поля, предназначенные для B, вводятся значения элементов вектора b. Далее нажимается кнопка “Найти X”.



Рисунок. 1. Интерфейс программы «Метод прогонки»

В результате в столбце X появятся значения решения системы (рис.2.).

Матрица A

	Column	Column	Column
	10	3	0
	1	9	2
▶	0	4	12
*			

B

	Column
	16
	25
▶	44
*	

X

	Column
▶	1
	2
	3
*	

Удалить столбец Добавить столбец Найти X

Рисунок. 2. Выполнение программы «Метод прогонки»

Следует обратить особое внимание на случай, когда происходит деление на 0. Тогда программа выдаст следующее сообщение (рис.3.).

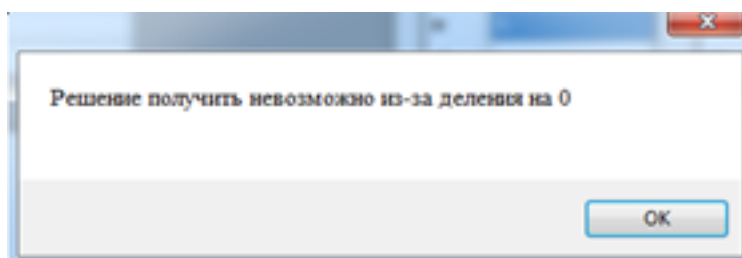


Рисунок. 3. Сообщение об ошибке программы «Метод прогонки»

Программа производит вычисления по методу прогонки и выводит значения в поля, предназначенные для вывода найденных неизвестных ранее значений вектора **X**.

Для тестирования программы была подобрана система с известным решением. Система с

матрицей $A = \begin{bmatrix} 10 & 3 & 0 \\ 1 & 9 & 2 \\ 0 & 4 & 12 \end{bmatrix}$ и вектором $B = \begin{bmatrix} 16 \\ 25 \\ 44 \end{bmatrix}$ имеет точное решение - вектор $x = \{1, 2, 3\}$. В программу задавались соответствующие исходные данные и были получены результаты. Результаты работы программы приведены на рисунке (рис. 2.).

Чтобы полностью убедиться в том, что программа работает исправно, протестируем ее ещё на одном примере, когда происходит деление на 0. В программу задаются значения матрицы. После ввода матрицы A в поля, предназначенные для B, вводятся произвольные значения элементов вектора b. Далее нажимается кнопка "Найти X".

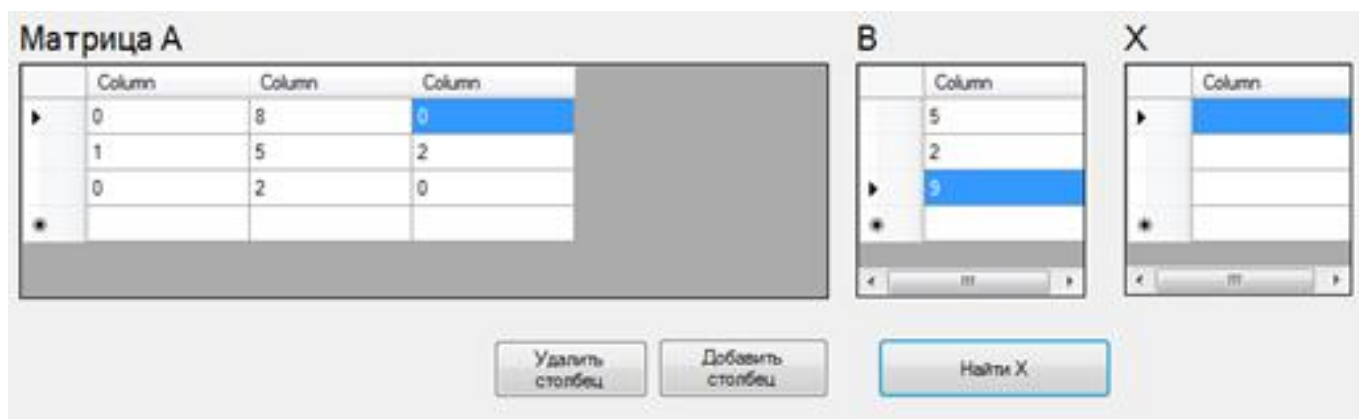


Рис. 4. Выполнение программы «Метод прогонки» при делении на 0

В результате в столбце X не появятся значения решения системы, а на экран выведется сообщение об ошибке (рис.3.).

Список литературы:

1. Стенли Б. Липпман, Жози Лажой. Язык программирования C++. Базовый курс. – Вильямс, 2017 – 1108с.
2. Страуструп Б. Язык программирования C++. Специальное издание – Москва Издательство Бином, 2015 – 1035с.
3. Трубников С.В. Численные методы. Часть 1. – Издательство БГУ, 2005 – 145с.