Лабораторная работа 3

«Решение СЛАУ методом прогонки»

Выполнил

Студент 1 группы, 2 курса

Быченок Егор

1. **Постановка задачи**

Разработать программу численного решения СЛАУ методом прогонки. Найти вектор приближённого решения y\*, а также рассчитать относительную погрешность вида для следующей матрицы:

n = 11, m = 4, k = 1

Недиагональные элементы выбираются из чисел 0, -1, -2, -3, -4 произвольным образом,

Диагональные элементы:

, 2in

= +

Задать вектор решений:

y = (1, 2, …, N + 1),

а вектор свободных членов задать умножением:

d = Ay

1. **Входные данные**

n = 11, m = 4, k = 0

A =

[

4 3 0 0 0 0 0 0 0 0 0 0

-1 5 4 0 0 0 0 0 0 0 0 0

0 -1 6 5 0 0 0 0 0 0 0 0

0 0 -1 7 6 0 0 0 0 0 0 0

0 0 0 -1 8 7 0 0 0 0 0 0

0 0 0 0 -1 9 8 0 0 0 0 0

0 0 0 0 0 -1 10 9 0 0 0 0

0 0 0 0 0 0 -1 11 10 0 0 0

0 0 0 0 0 0 0 -1 12 11 0 0

0 0 0 0 0 0 0 0 -1 13 12 0

0 0 0 0 0 0 0 0 0 -1 14 13

0 0 0 0 0 0 0 0 0 0 -1 15

]

y = ( 1 2 3 4 5 6 7 8 9 10 11 12 )

1. **Листинг программы**

**void** solveWithTridiagonalMatrixAlgorithm() {  
 *//прямая прогонка* **float**\* alpha = **new float**[n-1];  
 **float**\* beta = **new float**[n];  
 **float** denominatorTemp;  
  
 alpha[0] = A[0][1] / A[0][0];  
 beta[0] = d[0] / A[0][0];  
 **for** (**int** i = 1; i < n - 1; i++) {  
 denominatorTemp = A[i][i] - A[i][i-1] \* alpha[i-1];  
 alpha[i] = A[i][i+1] / denominatorTemp;  
 beta[i] = (d[i] - A[i][i-1] \* beta[i-1]) / denominatorTemp;  
 }  
  
 beta[n - 1] = (d[n-1] - A[n-1][n-2] \* beta[n-2]) / (A[n-1][n-1] - A[n-1][n-2] \* alpha[n-2]);  
  
 *//выводим коэффициенты после прямой прогонки* cout << **"alpha: ( "**;  
 **for** (**int** i = 0; i < n-1; i++) {  
 cout << alpha[i] << **" "**;  
 }  
 cout << **")"** << endl;  
  
 cout << **"beta: ( "**;  
 **for** (**int** i = 0; i < n-1; i++) {  
 cout << beta[i] << **" "**;  
 }  
 cout << **")"** << endl;  
  
 *//обратная прогонка* yApprox[n - 1] = beta[n - 1];  
 **for** (**int** i = n - 2; i >= 0; i--) {  
 yApprox[i] = beta[i] - alpha[i] \* yApprox[i+1];  
 }  
}

*//вычисляем относительную погрешность вида*

**float** calculateError() {

**float** error;  
 **float** normError = 0;  
 **float** yRealNorm = 0;

*//вычисляем*   
 **for** (**int** i = 0; i < n; i++) {  
 yRealNorm += abs(yReal[i]);  
 }

*//вычисляем*   
 **for** (**int** i = 0; i < n; i++) {  
 normError += abs(yReal[i] - yApprox[i]);  
 }  
 *//вычисляем относительную погрешность*

error = normError / yRealNorm;  
 **return** error;

1. **Выходные данные**

alpha = ( 0.75 0.695652 0.746753 0.774518 0.797765 0.816513 0.832061 0.845161 0.856354 0.866029 0.874477 )

beta = ( 2.5 4.08696 5.98701 7.87259 9.78659 11.7156 13.6565 15.6065 17.5635 19.5263 21.4937 )

= ( 0.9999995232 2.000000715 2.999999046 4.000000954 4.999999523 6.000000477 6.999999523 8.000000954 8.999999046 10.00000095 10.99999905 12 )

Error = 1.069826965e-07

1. **Выводы**

Для трехдиагональной матрицы метод прогонки, имеет линейную сложность, а значит требует меньше времени и вычислительной мощности для работы. Относительная погрешность маленькая, что совпадает с теорией, т.к. матрица диагонально преобладающая, а метод прогонки устойчив для таких матриц.