

1 Решение СЛАУ с помощью LU-разложения

1.1 Цель работы

Цель работы – закрепление навыков работы в ППП GNU Octave на примере решения задач линейной алгебры.

Для достижения поставленной цели используется программное обеспечение GNU Octave или Scilab.

1.2 Краткая теоретическая справка

Эффективным методом решения системы линейных алгебраических уравнений является метод разложения на треугольные матрицы, или LU-разложение. Алгоритмы этого метода близки к методу исключения Гаусса, хотя вычисления могут производиться в различной последовательности. LU-разложение матрицы A можно представить в виде $A=LU$, где L – нижнетреугольная матрица с единичной диагональю, а U – верхнетреугольная матрица. При этом решение СЛАУ, при известном LU-разложении, сводится к решению системы:

$$\begin{cases} Ux = y \\ Ly = b \end{cases} \quad (1.1)$$

Вначале находится LU-разложение матрицы, затем решается второе уравнение, а затем первое системы (1.1).

Существует порядка десяти вариантов реализации разложения. Чаще всего используются, так называемых *ijk*-формы со следующим обозначением индексов:

- k – номер исключаемой переменной;
- i – номер строки, т.е. модифицируемого уравнения;
- j – номер столбца, т.е. коэффициента в модифицируемом уравнении.

Тогда общую основу всех алгоритмов удобно определить тройкой вложенных циклов вида.

Для ____

Для ____

Для ____

$$a_{ij} = a_{ij} - l_{ik}a_{kj}$$

Далее приведены используемые в данной работе *ijk*-алгоритмы.

1.3 Алгоритмы LU-разложения

LU-разложение, *ikj*-алгоритм

Для $i = 2, \dots, N$

Для $k = 1, \dots, i - 1$

$$a_{i,k} = a_{i,k} / a_{k,k}$$

Для $j = k + 1, \dots, N$

$$a_{i,j} = a_{i,j} - a_{i,k} \cdot a_{k,j}$$

Увеличить j

Увеличить k

Увеличить i

Данный алгоритм позволяет пересчитать i -ю строку матрицы **A** в i -ю строку матриц **L** и **U**. Каждые $1, 2, \dots, j - 1$ строки участвуют в определении j -й строки матриц **L** и **U**, но сами больше не модернизируются. Таким образом, доступ к элементам матрицы **A** производится по строкам. Исключение по строкам. Модификации отложенные.

LU-разложение, *ijk*-алгоритм

Для $i = 2, \dots, N$

Для $j = 2, \dots, i$

$$a_{i,j-1} = a_{i,j-1} / a_{j-1,j-1}$$

Для $k = 1, \dots, j - 1$

$$a_{i,j} = a_{i,j} - a_{i,k} \cdot a_{k,j}$$

Увеличить k

Увеличить j

Для $j = i + 1, \dots, N$

Для $k = 1, \dots, i - 1$

$$a_{i,j} = a_{i,j} - a_{i,k} \cdot a_{k,j}$$

Увеличить k

Увеличить j

Увеличить i

Доступ к элементам матрицы **A** производится по строкам. Исключение по строкам. Модификации отложенные. Первый цикл по j элементы i -й строки матрицы **L**. Второй цикл по j элементы i -й строки **U**.

LU-разложение, *kij*-алгоритм

Для $k = 1, \dots, N - 1$

Для $i = k + 1, \dots, N$

$$a_{i,k} = a_{i,k} / a_{k,k}$$

Для $j = k + 1, \dots, N$

$$a_{i,j} = a_{i,j} - a_{i,k} \cdot a_{k,j}$$

Увеличить j

Увеличить i

Увеличить k

Доступ к элементам матрицы **A** производится по строкам. Исключение по столбцам. Модификации немедленные.

LU-разложение, *kji*-алгоритм

Для $k = 1, \dots, N - 1$

Для $s = k + 1, \dots, N$

$$a_{s,k} = a_{s,k} / a_{k,k}$$

Доступ к элементам матрицы **A** производится по столбцам. Исключение по столбцам. Модификации немедленные.

```

    Увеличить s
    Для j = k + 1, ..., N
        Для i = k + 1, ..., N
             $a_{i,j} = a_{i,j} - a_{i,k} \cdot a_{k,j}$ 
        Увеличить i
    Увеличить j
Увеличить k

```

В ходе работы требуется реализовать представленные алгоритмы в виде функций с именами вида LUikj().

Далее с помощью реализованных функций получить время, затрачиваемое на LU-разложение, для произвольно заданной матрицы порядка 100, 200, ..., 1500. Построить полученные зависимости. Проанализировать полученные результаты. Полученные результаты и выводы привести в отчёте.

1.4 Решение систем с треугольными матрицами

С помощью одной из функций, реализованных ранее для получения LU-разложения, решить тестовую СЛАУ вида

$$\begin{cases} 1,83x_1 + 4,34x_2 - 7,49x_3 + 11,07x_4 = 1 \\ -2,15x_1 + 4,94x_2 - 3,89x_3 + 6,48x_4 = 2 \\ -0,50x_1 + 1,94x_2 - 3,32x_3 + 4,33x_4 = 3 \\ -4,27x_1 + 8,45x_2 - 7,71x_3 + 12,30x_4 = 4 \end{cases}.$$

Результат тестирования привести в отчёте.

После получения LU-разложения требуется реализовать 2 функции для решения системы вида (1.1). Данный процесс состоит из решения двух уравнений с треугольными матрицами, сначала с **L**, а затем с **U**. Далее приведен пример функции (*backsubL()*) для решения уравнения вида **Ly = b**:

```

function Y = backsubL(A, B)
n = length(B);
Y = zeros(n, 1);
Y(1) = B(1);
for k = 2:n
    Y(k) = (B(k) - A(k,k-1:-1:1)*Y(k-1:-1:1,1));
end

```

Необходимо проработать программную реализацию. После чего реализовать аналогичную функцию для решения уравнения вида $Ux = y$ с именем *backsubU()* и проверить правильность реализации на примере приведенной выше СЛАУ с помощью стандартных средств Octave (Scilab).

Полученные результаты и выводы привести в отчёте.

1.5 Содержание и требования к оформлению отчета

Отчет должен содержать титульный лист, название работы и цель работы, исходные данные, результаты расчетов, таблицы и графики, анализ результатов и выводы по работе.

Оформление должно соответствовать ОС ТУСУР 01-2013 "работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления".