

Лабораторная работа №6.

Параллельное и распределенное программирование. Технология MPI.

Название лабораторной работы: LU-разложение

Содержание

1. Описание LU-разложения
2. Построение LU-разложения методом исключения Гаусса
3. Цель работы
4. Описание реализации
5. Описание лабораторной работы

Приложение

1. Описание LU-разложения

Под LU -разложением подразумевается представление квадратной матрицы в виде произведения нижнетреугольной матрицы на верхнетреугольную матрицу с единицами на главной диагонали. Такое представление удобно для решения системы линейных алгебраических уравнений, поскольку оно позволяет перейти от решения исходной системы к последовательному решению систем с упомянутыми треугольными матрицами. Кроме того, в памяти компьютера это представление занимает ровно столько места, сколько занимает исходная матрица; исходную матрицу не требуется хранить, поскольку вся информация о ней содержится в получаемых треугольных матрицах (в случае необходимости восстановить исходную матрицу достаточно найти произведение упомянутых треугольных матриц). Следует отметить также, что алгоритм разложения позволяет последовательно сохранять получаемые элементы треугольных матриц на местах, освобождаемых последовательным затиранем определенных элементов исходной матрицы (см. ниже); следовательно, для реализации процесса LU-разложения дополнительного места в памяти вычислительной системы не требуется. Особенно полезно LU -разложение матрицы в том случае, когда решается множество систем линейных алгебраических уравнений с одной и той же матрицей, но с различными правыми частями.

В Приложении приведены точные формулировки используемых утверждений с доказательствами: последние, как известно, необходимы для правильного понимания сформулированных утверждений.

В данном пункте дадим лишь необходимый фрагмент теории (доказательства см. в Приложении).

Рассмотрим квадратную матрицу $A = (a_{ij})_{i,j=1,\dots,n}$. Обозначим A_k квадратную матрицу, образованную пересечением первых k строк и первых k столбцов матрицы A , $A_k = (a_{ij})_{i,j=1,\dots,k}$, где $1 \leq k \leq n$; в частности, матрица A_1 состоит из одного, элемента a_{11} , $A_1 = (a_{11})$, а $A_n = A$. Определители её $\det A_k$, $k = 1, \dots, n$, называются главными минорами матрицы A .

Теорема 1 (об LU-разложении) Квадратная матрица A с ненулевыми главными минорами однозначно представляется в виде произведения LU нижнетреугольной матрицы L , главная диагональ которой состоит из ненулевых элементов, и верхнетреугольной матрицы U с единицами на главной диагонали.

Замечание. С помощью метода индукции аналогичным образом нетрудно установить, что если матрица A — ленточная матрица с шириной ленты $2p + 1$, удовлетворяющая условиям теоремы, то матрицы L и U также можно рассматривать как ленточные, ширина ленты которых не больше, чем ширина ленты у матрицы A . Если A диагональная, то L и U — тоже диагональные.

2. Построение LU-разложения методом исключения Гаусса

Метод исключения Гаусса широко применяется при решении систем линейных алгебраических уравнений. Счастливым обстоятельством является тот факт, что в прямом ходе этого метода получается интересующее нас LU-разложение матрицы системы уравнений.

Метод Гаусса относится к так называемым "точным" методам, т. е. к методам, позволяющим найти точное решение системы линейных алгебраических уравнений за конечное число арифметических действий в предположении, что упомянутые действия выполняются точно.

Однако, не всякая квадратная матрица, даже невырожденная, имеет LU-разложение. В качестве контрпримера вполне подходит матрица перестановки второго порядка, т. е.

матрица $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Это происходит потому, что первый главный минор такой матрицы равен 0. На практике матрицы из вещественных чисел представляются в компьютерах лишь приближенно, поэтому всегда можно считать, что все главные миноры отличные от нуля; однако, они могут быть очень близки к 0, что приводит к резкому ухудшению точности вычислений.

Чтобы избежать этого явления, иногда (в частности, для решения систем линейных уравнений) допускается строить LU-разложение не самой матрицы коэффициентов, а результата некоторой перестановки ее строк и столбцов. При решении системы линейных алгебраических уравнений перестановка строк матрицы коэффициентов соответствует перестановке уравнений исходной системы, а перестановка столбцов, перестановке переменных.

Итак, рассмотрим систему линейных алгебраических уравнений

$$Ax = b, \quad (2.1)$$

где

$$A \stackrel{\text{def}}{=} \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & a_{n,3} & \dots & a_{n,n} \end{pmatrix}, \quad x \stackrel{\text{def}}{=} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix}, \quad b \stackrel{\text{def}}{=} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix}.$$

Поскольку нас интересует только LU-разложение матрицы A, то ограничимся лишь прямым ходом метода Гаусса. Для простоты рассмотрим его в случае $n = 4$: общий случай легко получается из этих рассмотрений. Итак, рассмотрим систему

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}. \quad (2.2)$$

Если $a_{11} \neq 0$, на первом шаге умножаем первое уравнение на $m_{i1} \stackrel{\text{def}}{=} a_{i1}/a_{11}$ и отнимаем его от i -го; здесь $i = 2, 3, 4$. В результате получаем эквивалентную систему

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & a_{34}^{(2)} \\ 0 & a_{42}^{(2)} & a_{43}^{(2)} & a_{44}^{(2)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(2)} \\ b_3^{(2)} \\ b_4^{(2)} \end{pmatrix}, \quad (2.3)$$

где $a_{ij}^{(2)} = a_{ij} - a_{1j}a_{i1}/a_{11}$, $i, j = 2, 3, 4$, $b_i^{(2)} = b_i - b_1a_{i1}/a_{11}$, $i = 2, 3, 4$.

Если же $a_{11} = 0$ (или близко к 0), мы можем поступить одним из следующих способов:

- а) Выбор главного элемента по столбцу: находим строку с максимальным по модулю первым элементом и меняем ее местами с первой строкой.
- б) Выбор главного элемента по строке: находим столбец с максимальным по модулю первым элементом и меняем его местами с первым столбцом.
- в) Выбор главного элемента по всей матрице: находим максимальный по модулю элемент во всей матрице, и при помощи перестановки пары строк и пары столбцов ставим его в верхний левый угол.

После этих преобразований первый шаг производится так же, как и в случае $a_{11} \neq 0$. Вообще говоря, вариантов (а) или (б) (на каждом шаге) достаточно, чтобы можно было построить LU-разложение для любой невырожденной матрицы, однако вариант (в) дает лучшую точность.

Систему (2.3) запишем в виде

$$A^{(2)}x = b^{(2)}, \quad (2.4)$$

где

$$A^{(2)} = M_1A, \quad b^{(2)} = M_1b, \quad (2.5)$$

причем M_1 — нижнетреугольная матрица вида

$$M_1 \stackrel{\text{def}}{=} \begin{pmatrix} 1 & 0 & 0 & 0 \\ -m_{21} & 1 & 0 & 0 \\ -m_{31} & 0 & 1 & 0 \\ -m_{41} & 0 & 0 & 1 \end{pmatrix}. \quad (2.6)$$

Если $a_{22}^{(2)} \neq 0$, то сделаем второй шаг: умножим второе уравнение в соотношении

(2.3) на $m_{i2} \stackrel{\text{def}}{=} a_{i2}^{(2)} / a_{22}^{(2)}$ и вычтем его от i -го; здесь $i = 3, 4$. В результате получаем эквивалентную систему

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & a_{34}^{(3)} \\ 0 & 0 & a_{43}^{(3)} & a_{44}^{(3)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(2)} \\ b_3^{(3)} \\ b_4^{(3)} \end{pmatrix}, \quad (2.7)$$

где $a_{ij}^{(3)} = a_{ij}^{(2)} - a_{2j}^{(2)} a_{i2}^{(2)} / a_{22}^{(2)}$, $i, j = 3, 4$, $b_i^{(3)} = b_i^{(2)} - b_2^{(2)} a_{i2}^{(2)} / a_{22}^{(2)}$, $i = 3, 4$.

Систему (2.7) запишем в виде

$$A^{(3)} x = b^{(3)}, \quad (2.8)$$

где

$$A^{(3)} = M_2 A^{(2)}, \quad b^{(3)} = M_2 b^{(2)}, \quad (2.9)$$

причем M_2 — нижетреугольная матрица вида

$$M_2 \stackrel{\text{def}}{=} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -m_{32} & 1 & 0 \\ 0 & -m_{42} & 0 & 1 \end{pmatrix}. \quad (2.10)$$

Если же $a_{22}^{(2)} = 0$, можно воспользоваться одним из приемов (а) — (в) для выбора главного элемента, однако на сей раз главный элемент надо искать во втором столбце (кроме первой строки), во второй строке (кроме первого столбца) или всей подматрице между элементами с индексами 2,2 и 4,4.

Наконец, предполагая, что $a_{33}^{(3)} \neq 0$, сделаем третий шаг. Умножим третье уравнение в соотношении (2.7) на $m_{i3} \stackrel{\text{def}}{=} a_{i3}^{(3)} / a_{33}^{(3)}$ и вычтем его от i -го; здесь $i = 4$. В результате получаем эквивалентную систему

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & a_{34}^{(3)} \\ 0 & 0 & 0 & a_{44}^{(4)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(2)} \\ b_3^{(3)} \\ b_4^{(4)} \end{pmatrix}, \quad (2.11)$$

где $a_{ij}^{(4)} = a_{ij}^{(3)} - a_{3j}^{(3)}a_{i3}^{(3)}/a_{33}^{(3)}$, $i, j = 4$, $b_i^{(4)} = b_i^{(3)} - b_3^{(3)}a_{i3}^{(3)}/a_{33}^{(3)}$, $i = 4$.

Систему (2.11) запишем в виде

$$A^{(4)}x = b^{(4)}, \quad (2.12)$$

где

$$A^{(4)} = M_3 A^{(3)}, \quad b^{(4)} = M_3 b^{(3)}, \quad (2.13)$$

причем M_3 — нижнетреугольная матрица вида

$$M_3 \stackrel{\text{def}}{=} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -m_{43} & 1 \end{pmatrix}. \quad (2.14)$$

Если использовать формулы (2.4) - (2.6), (2.8) - (2.10), (2.12) - (2.14), то получим

$$A = M_1^{-1} M_2^{-1} M_3^{-1} A^{(4)} \quad (2.15)$$

$$b = M_1^{-1} M_2^{-1} M_3^{-1} b^{(4)} \quad (2.16)$$

Вводя обозначение

$$L \stackrel{\text{def}}{=} M_1^{-1} M_2^{-1} M_3^{-1}, \quad (2.17)$$

найдем

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & m_{32} & 1 & 0 \\ m_{41} & m_{42} & m_{43} & 1 \end{pmatrix}. \quad (2.18)$$

Соотношение (2.15) показывает представление матрицы A в виде произведения нижнетреугольной матрицы L с единицами на главной диагонали на верхнетреугольную матрицу $A^{(4)}$; ввиду единственности такого представления (см. теорему 1) искомое разложение получено. Полагая $U = A^{(4)}$, находим

$$A = LU, \quad (2.19)$$

где L определяется формулой (2.18), а для U справедливо представление

$$U = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & a_{34}^{(3)} \\ 0 & 0 & 0 & a_{44}^{(4)} \end{pmatrix}. \quad (2.20)$$

Рассмотренные при $n = 4$ алгоритмы легко обобщаются на случай произвольного $n = 2, 3, 4, \dots$. Нетрудно подсчитать число операций, необходимых для реализации прямого и обратного ходов метода Гаусса.

Например, для реализации прямого хода число требующихся умножений и делений имеет вид многочлена от n , старший член которого равен $n^3/3$, а число умножений и делений при обратном ходе равно $n(n+1)/2$.

Если число n велико, то реализация метода Гаусса (получение LU-разложения) требует значительной памяти ЭВМ, ибо требуется хранить n^3 элементов матрицы A . Однако алгоритм позволяет размещать в прямом ходе элементы m_{ij} на тех местах в матрице A , которые заполняются нулями в процессе исключения.

Также следует отметить, что в рассмотренном примере мы ищем такое LU-разложение, в котором единицы стоят на диагонали матрицы L , а в приведенной ниже программе ищется разложение, в котором единицы стоят на диагонали матрицы U .

Разница в алгоритмах состоит лишь в том, что в приведенной ниже программе перед вычитанием из последующих строк текущая строка делится на элемент в текущем столбце, и соответствующим образом подправляются коэффициенты, с которыми надо вычитать ее из последующих строк

3. Цель работы

Цель данной работы состоит в следующем

- 1) дать представление о средствах распараллеливания MPI,

- 2) научить элементарным способам использования средства MPI при программировании,
- 3) привить навыки запуска параллельной программы.
- 4) оценить ускорение работы параллельной версии алгоритма в сравнении с последовательной при использовании многоядерных архитектур,
- 5) на основе лабораторной работы сделать выводы о зависимости ускорения от начальных параметров.

Постановка задачи: Требуется вычислить LU-разложение, используя различные варианты метода Гаусса (без выбора ведущего элемента, с выбором ведущего элемента по строкам, с выбором ведущего элемента по столбцам, с выбором ведущего элемента по всей матрице при различных значениях параметра n), дать соответствующие варианты программы, численно исследовать ее устойчивость в зависимости от упомянутого параметра n . искусственно вводя погрешность ϵ в промежуточные вычисления, теоретически обосновать полученные результаты,

4. Описание реализации

На параллельных системах стандарт MPI обычно реализован для алгоритмических языков C, C++ и Fortran. Следующая программа на языке C дает иллюстрацию такого использования для вычисления LU-разложения.

Распараллеливание вышеописанного алгоритма с помощью MPI выглядит следующим образом

```
1 #include <mpi.h>
2 #include <stdio.h>
3
4 #define n 4
5
6 int main(int argc, char *argv[])
7
8 int myrank, nprocs, i, j, k, map[n];
9 double a[n][n];
10 FILE *f;
11
```



```

12 MPI_Init(&argc, &argv);
13 MPI_Comm_rank(MPI_COMM_WORLD, &myrank) ;
14 MPI_Comm_size(MPI_COMM_WORLD, &nprocs) ;
15
16 if (myrank == 0)
17 {
18     f = fopen("lu_4", "r");
19     for(i = 0; i < n; i++)
20         for(j = 0; j < n; j++)
21             fscanf(f, "%lg", wali[j]);
22     fclose(f);
23 }
24
25 MPI_Bcast(a, n*n, MPI_DOUBLE, 0, MPI_COMM_WORLD) ;
26
27 for(i = 0; i < n; i++)
28     map[i] = i % nprocs;
29
30 for(k = 0; k < n-1; k++)
31 {
32     if(map[k] == myrank)
33
34         for(i = k+1; i < n; i++)
35             a[k][i] /= a[k][k];
36
37     MPI_Bcast (&a[k][k+1], n-k-1, MPI_DOUBLE,
38               map[k], MPI_COMM_WORLD) ;
39
40     for(i = k+1; i < n; i++)
41         if(map[i] == myrank)
42             for(j = k+1; j < n; j++)
43                 a[i][j] -= a[i][k]*a[k][j];
44 }

```

```

45 // Printing the entries of the matrix
46
47 for(i = 0; i<n; i++)
48     if(map[i] == myrank)
49     {
50         for(j = 0; j<n; j++)
51             printf ("a[%d] [%d] = %lg, ", i, j, a[i][j]);
52         printf ("\n");
53     }
54
55
56 printf ("\n");
57 MPI_Finalize() ;
58 return 0;
59}

```

Заметим прежде всего, что согласно концепции SIMD (Single Instruction Multiple Data) при использовании стандарта MPI программа копируется во все рассматриваемые параллельные процессы; при этом, конечно, каждый процесс — в зависимости: от своего номера — выполняет специфическую для него работу.

В первой строке программы подключается библиотека `mpi.h`, реализующая стандарт MPI для C, во второй строке подключается библиотека `stdio.h`, содержащая базовые функции ввода-вывода. Строка 4 определяет `n` как целую константу со значением 4. Строка 6 содержит стандартный заголовок функции `main`, в запуске которой и состоит выполнение программы. Строки 8 — 10 содержат описания необходимых переменных. Далее, в строках 12-14 вызываются процедуры MPI. Рассмотрим их подробнее,

Процедура `MPI_Init(&argc, &argv);` инициализирует библиотеку MPI (ее применение обязательно перед тем, как начать использование упомянутой библиотеки); передаваемые ей параметры служат для выборки из аргументов командной строки

тех, которые относятся к библиотеке MPI.

Процедура `MPI_Comm_rank(MPI_COMM_WORLD, &myrank);` определяет номер процесса (он присваивается переменной `myrank`) в группе с коммуникатором

MPI_COMM_WORLD. Этот коммуникатор определяет группу всех процессов, запущенных для решения данной задачи. Процессы в любой группе нумеруются целыми числами от 0 до (число процессов) — 1.

Процедура `MPI_Comm_size(MPI_COMM_WORLD, &nprocs);` позволяет определить число процессов в группе с коммуникатором `MPI_COMM_WORLD` ЕД (это число присваивается выходному параметру `nprocs`). Оно определяется при запуске программы на выполнение при помощи опции `-np` и не должно превышать имеющегося в системе числа процессоров.

Строки 16 — 23 посвящены чтению элементов матрицы из файла «lu_4». Условный оператор нужен для того, чтобы это чтение выполнял только процесс с номером 0. Затем, в строке 25 считанная в 0-м процессе матрица рассылается по всем процессам при помощи вызова процедуры:

```
MPI_Bcast(a, n*n, MPI_DOUBLE, 0, MPI_COMM_WORLD);
```

Параметры этой процедуры имеют следующий смысл:

- первый параметр — указатель на первый элемент массива рассылаемых значений (в данном случае указателем служит `a` — указатель на первый элемент матрицы);
- второй параметр — количество рассылаемых значений (в данном случае их количество равно n^2 , где константа n определяет порядок матрицы),
- третий параметр означает тип рассылаемого значения (здесь должен использоваться тип, определенный в стандарте MPI: в нашем случае это `MPI_DOUBLE`),
- четвертый параметр — номер рассылającego процесса (в нашем случае его номер равен 0),
- пятый параметр — имя коммуникатора группы; поскольку в рассматриваемом случае используется лишь исходная группа параллельных процессов, то здесь должно стоять имя исходного коммуникатора, а именно `MPI_COMM_WORLD`.

В строках 27, 28 вычисляется массив `map`. Его элемент с индексом i равен номеру процесса, «владеющего» строкой с номером i .

В строках 30-43 вычисляется собственно LU-разложение матрицы. С процедурой `MPI_Bcast`, вызываемой здесь, мы уже встречались раньше. Здесь она нужна для рассылки конца текущей строки всем процессам, чтобы каждый процесс мог вычесть этот остаток строки с подходящим коэффициентом из всех имеющихся в его распоряжении строк, находящихся ниже текущей.

В строках 47-53 печатаются элементы полученной матрицы. Элементы на главной диагонали и ниже принадлежат матрице L , остальные — матрице U . Поскольку разные процессы выполняются на разных компьютерах кластера, порядок элементов при печати может быть разным. Как правило, сначала печатаются строки, которыми владеет 0-й процесс, затем 1-й, и т. д. Именно поэтому для каждого элемента печатается не только значение, но и индексы.

5. Описание лабораторной работы

В лабораторной работе требуется выполнить следующие разделы:

- 1) запуск программы;
- 2) распараллеливание LU-разложения без выбора ведущего элемента;
- 3) распараллеливание LU-разложения с выбором ведущего элемента по строкам;
- 4) распараллеливание LU-разложения с выбором ведущего элемента по столбцам;
- 5) распараллеливание LU-разложения с глобальным выбором ведущего элемента (т.е. по строкам и по столбцам);
- 6) изменение конфигурации (масштабирование) параллельной системы;
- 7) построение графиков полученных зависимостей.

1. Запуск программы

- В любом редакторе набрать программу, представленную в пункте 4 и сохранить ее в файле с именем `comp_lu.c` (см. введение),
- Оттранслировать набранную программу (см. введение),
- В любом редакторе набрать матрицу (как последовательность элементов, разделенных пробелами) и сохранить с именем `lu_4` (см. введение),
- запустить результат трансляции на счет (см. введение),
- прочесть результат.

2. Распараллеливание LU-разложения без выбора ведущего элемента

Здесь предлагается исследовать точность вычислений и время счета в зависимости от параметра n . Для измерения времени используется способ, предложенный во введении.

Предупреждение! При исследовании могут получиться весьма неожиданные результаты. После выполнения задания попытайтесь их объяснить.

Порядок действий

- Измените программу так, чтобы вводилась матрица Гильберта, вычисляемая по формуле $a_{ij} = \frac{1}{i+j-1}, i, j = 1, 2, \dots, n$.
- Для значений $n = 2^w, w=9,10,\dots$ запустите программу и определите время и точность вычислений; заполните таблицу.

Таблица - LU -разложения без выбора ведущего элемента

Значение n	Время вычислений, сек.	Точность вычислений
2 ⁹		
....		

3. Распараллеливание LU-разложения с выбором ведущего элемента по строкам

Порядок действий

- Измените программу так, чтобы использовался алгоритм выбора ведущего элемента по строкам (для этого может потребоваться менять местами столбцы матрицы - эти действия должны выполняться в каждом процессе над теми строками, которыми этот процесс владеет: также может потребоваться барьерная синхронизация при помощи процедуры MPI_Barrier).
- Для значений $n = 2^w, w=9,10,\dots$ запустите программу и определите время и точность вычислений; заполните таблицу.

Таблица - LU -разложения с выбором ведущего элемента по строкам

Значение n	Время вычислений, сек.	Точность вычислений
2 ⁹		
....		

4. Распараллеливание LU-разложения с выбором ведущего элемента по столбцам

Порядок действий

- Аналогично предыдущему заданию измените программу так, чтобы использовался алгоритм выбора ведущего элемента по столбцам (это сложнее, чем предыдущий вариант, поскольку здесь может потребоваться менять местами строки, принадлежащие разным процессам — см. процедуры MPI_Send, MPI_Recv).
- Для значений $n = 2^w$, $w=9,10,\dots$ запустите программу и определите время и точность вычислений; заполните таблицу.

Таблица - LU -разложения с выбором ведущего элемента по столбцам

Значение n	Время вычислений, сек.	Точность вычислений
2^9		
....		

5. Распараллеливание LU-разложения с глобальным выбором ведущего элемента

Порядок действий

- Аналогично предыдущему заданию измените программу так, чтобы использовался алгоритм выбора ведущего элемента по строкам и столбцам одновременно.
- Для значений $n = 2^w$, $w=9,10,\dots$ запустите программу и определите время и точность вычислений; заполните таблицу.

Таблица - LU -разложения с глобальным выбором ведущего элемента

Значение n	Время вычислений, сек.	Точность вычислений
2^9		
....		

6. Изменение конфигурации (масштабирование) параллельной системы

Изменить конфигурацию параллельной системы можно изменением количества вычислительных модулей.

Порядок действий

- Провести следующие варианты экспериментального исследования:
 - 1) сохранить 2 параллельных вычислительных модуля

2) сохранить 4 параллельных вычислительных модуля.

3) сохранить 8 параллельных вычислительных модулей

– В каждом из рассмотренных вариантов конфигурации параллельной системы провести вычисления, указанные в пунктах 1-4.

– По полученным результатам для каждой задачи найти относительное ускорение вычислений как отношение времени T_1 , истраченного при запуске на конфигурации 1, ко времени T_i , истраченного при запуске на i -й конфигурации:

$$U_i = \frac{T_1}{T_i} \quad i = 2, 3.$$

– Результаты оформить в виде таблиц вида (приводится лишь один пример таблицы):

Таблица - Распараллеливание LU-разложения

Значение n	Время вычислений, сек.	Точность вычислений	Относительное ускорение, U_i
2^9			
....			

7. Построение графиков полученных зависимостей

Для исследованных случаев построить полигональные графики следующих зависимостей

- 1) зависимости ускорения от параметра LU-разложения,
- 2) зависимости ускорения от числа процессов,
- 3) зависимости точности вычислений и времени счета от параметра.

ЛИТЕРАТУРА

1. Форсайт Дж. Молер К. Численное решение систем линейных алгебраических уравнений. М., 1969. 167 с.
2. Керниган Б., Ригчи Д. Язык программирования C, М.1992. 272 с.
3. Немнюгин С.А. Стесик О.Л. Параллельное программирование для многопроцессорных вычислительных систем. СПб. 2002. 400 с.
4. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб, 2002. 608 с.

П Р И Л О Ж Е Н И Е

Рассмотрим квадратную матрицу $A = (a_{ij})_{i,j=1,\dots,n}$. Обозначим A_k квадратную матрицу, образованную пересечением первых k строк и первых k столбцов матрицы A , $A_k = (a_{ij})_{i,j=1,\dots,k}$, где $1 \leq k \leq n$; в частности, матрица A_1 состоит из одного элемента a_{11} , $A_1 = (a_{11})$, а $A_n = A$. Определители $\det A_k$, $k = 1, \dots, n$, называются главными минорами матрицы A .

Теорема 1 (Об LU -разложении) *Квадратная матрица A с ненулевыми главными минорами однозначно представляется в виде произведения LU нижнетреугольной матрицы L с единицами на главной диагонали и верхнетреугольной матрицы U , главная диагональ которой состоит из ненулевых элементов.*

Доказательство. По условию теоремы

$$\det A_k \neq 0, \quad k = 1, 2, \dots, n. \quad (1)$$

Доказательство теоремы проведём индукцией по k . При $k = 1$ очевидно

$$(a_{11}) = (1) \cdot (a_{11}), \quad (2)$$

так что рассматривая одноэлементные матрицы $L_1 = (1)$, $U_1 = (a_{11})$, из (2) получим $A_1 = L_1 U_1$. Итак, база индукции установлена.

Пусть теперь при некотором k , $k \in \{1, 2, \dots, n-1\}$, известно, что

$$A_k = L_k U_k, \quad (3)$$

где $L_k = (l_{ij}^{(k)})$ — нижнетреугольная матрица порядка k с единицами на главной диагонали, а U_k — верхнетреугольная матрица того же порядка, $U_k = (u_{ij}^{(k)})$. Ввиду определения матрицы L_k имеем $\det L_k = 1$, а из неравенства (1) и свойства $\det A_k = \det L_k \cdot \det U_k$ получаем $\det U_k \neq 0$. Итак,

$$\det L_k = 1, \quad \det U_k \neq 0. \quad (4)$$

Представим матрицу A_{k+1} в клеточной форме

$$A_{k+1} = \begin{pmatrix} A_k & b \\ a & a_{k+1\ k+1} \end{pmatrix},$$

где $a_{k+1\ k+1}$ — элемент исходной матрицы A , a — k -мерная вектор-строка, b — k -мерный вектор-столбец.

Будем находить k -мерную вектор-строку l и k -мерный вектор-столбец u , а также элемент $u_{k+1\ k+1}$ из уравнения

$$\begin{pmatrix} L_k & 0 \\ l & 1 \end{pmatrix} \begin{pmatrix} U_k & u \\ 0 & u_{k+1\ k+1} \end{pmatrix} = \begin{pmatrix} A_k & b \\ a & a_{k+1\ k+1} \end{pmatrix}. \quad (5)$$

Соотношения (5) эквивалентны равенствам

$$L_k U_k = A_k, \quad L_k u = b, \quad l U_k = a, \quad (6)$$

$$l \cdot u + u_{k+1 k+1} = a_{k+1 k+1}, \quad (7)$$

где $l \cdot u$ — скалярное произведение векторов l и u . Первое из равенств (6) заведомо верно, ибо совпадает с (3), второе представляет собой систему линейных уравнений с неособенной матрицей L_k (см. (4)); отсюда однозначно определяется вектор u . Третье из соотношений (6) является системой линейных алгебраических уравнений относительно компонент вектора l ; благодаря второй формуле в (4) вектор l определяется однозначно. Таким образом, в (7) l и u уже определены; поэтому имеем

$$u_{k+1, k+1} = a_{k+1 k+1} - l \cdot u.$$

Если положить

$$L_{k+1} = \begin{pmatrix} L_k & 0 \\ l & 1 \end{pmatrix}, \quad U_{k+1} = \begin{pmatrix} U_k & u \\ 0 & u_{k+1 k+1} \end{pmatrix},$$

то (5) принимает вид

$$A_{k+1} = L_{k+1} U_{k+1}.$$

Шаг индукции завершен; вместе с базой индукции это завершает доказательство. Теорема доказана. ■

Замечание. С помощью метода индукции аналогичным образом нетрудно установить, что если матрица A — ленточная матрица с шириной ленты $2p + 1$, удовлетворяющая условиям теоремы, то матрицы L и U также можно рассматривать как ленточные, ширина ленты которых не больше, чем ширина ленты у матрицы A . Если A — диагональная, то L и U — тоже диагональные.

Следствие 1. В условиях предыдущей теоремы

$$\det A = u_{11} u_{22} \dots u_{nn}. \quad (8)$$

Д о к а з а т е л ь с т в о. Определитель произведения квадратных матриц равен произведению определителей перемножаемых матриц, а определитель треугольной матрицы равен произведению ее диагональных элементов. Теперь для доказательства следствия достаточно воспользоваться формулой (1). ■

Следствие 2. Пусть выполнены условия предыдущей теоремы и $\det A \neq 0$. Тогда существует и единственно разложение

$$A = LDU^{(1)}, \quad (9)$$

где L и U соответственно нижне- и верхне-треугольные матрицы с единицами на главной диагонали, а D — диагональная матрица.

Д о к а з а т е л ь с т в о. Для доказательства этого утверждения достаточно найти диагональную матрицу

$$D = \begin{pmatrix} d_1 & 0 & 0 & \dots & 0 \\ 0 & d_2 & 0 & \dots & 0 \\ 0 & 0 & d_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & d_n \end{pmatrix}$$

и верхнетреугольную матрицу

$$U = \begin{pmatrix} 1 & u_{1,2}^{(1)} & u_{1,3}^{(1)} & \dots & u_{1,n}^{(1)} \\ 0 & 1 & u_{2,3}^{(1)} & \dots & u_{2,n}^{(1)} \\ 0 & 0 & 1 & \dots & u_{3,n}^{(1)} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}$$

так что

$$U = DU^{(1)}. \quad (10)$$

Поскольку

$$\begin{pmatrix} d_1 & 0 & 0 & \dots & 0 \\ 0 & d_2 & 0 & \dots & 0 \\ 0 & 0 & d_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & d_n \end{pmatrix} \begin{pmatrix} 1 & u_{1,2}^{(1)} & u_{1,3}^{(1)} & \dots & u_{1,n}^{(1)} \\ 0 & 1 & u_{2,3}^{(1)} & \dots & u_{2,n}^{(1)} \\ 0 & 0 & 1 & \dots & u_{3,n}^{(1)} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} d_1 & d_1 u_{1,2}^{(1)} & d_1 u_{1,3}^{(1)} & \dots & d_1 u_{1,n}^{(1)} \\ 0 & d_2 & d_2 u_{2,3}^{(1)} & \dots & d_2 u_{2,n}^{(1)} \\ 0 & 0 & d_3 & \dots & d_3 u_{3,n}^{(1)} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & d_n \end{pmatrix},$$

то для выполнения равенства (10) достаточно положить $d_i = u_{i,i}$ и $u_{i,j}^{(1)} = u_{i,j}/d_i$. Единственность такого представления очевидна. ■

Следствие 3. Пусть выполнены условия предыдущего следствия и, кроме того, матрица A симметрична. Тогда существует и единственно представление

$$A = LDL^T, \quad (11)$$

где L и D соответственно нижне-треугольная матрица с единицами на главной диагонали и диагональная матрица, а T означает транспонирование.

Доказательство. Используя предыдущее следствие (см. формулу (9)), с помощью условия $A = A^T$ получаем $LDU^{(1)} = (LDU^{(1)})^T$; учитывая правило транспонирования произведения, отсюда находим $LDU^{(1)} = (U^{(1)})^T DL^T$, так что $A = (U^{(1)})^T DL^T$. Правая часть последнего равенства представляет собой произведение нижне-треугольной матрицы на диагональную и на верхне-треугольную. Ввиду единственности такого представления (см. формулу (9) в следствии 2) имеем $L^T = U^{(1)}$, откуда ввиду формулы (9) получаем соотношение (11). ■

Обозначим круглыми скобками (x, y) скалярное произведение в евклидовом n -мерном пространстве \mathbb{R}^n ,

$$(x, y) = \sum_{i=1}^n x_i y_i.$$

Напомним, что матрица A называется положительно определенной, если выполнено неравенство

$$(Ax, x) > 0 \quad \forall x \in \mathbb{R}^n, \quad x \neq 0. \quad (12)$$

Следствие 4. Пусть выполнены условия теоремы 1, а кроме того, матрица A симметричная и положительно определенная. Тогда существует и единственная нижнетреугольная матрица G такая, что справедливо представление

$$A = GG^T. \quad (13)$$

Доказательство. В рассматриваемых условиях выполнены предположения леммы 3, так что справедливо соотношение (11); воспользуемся им в неравенстве (12):

$$(LDL^T x, x) > 0 \quad \forall x \in \mathbb{R}^n, \quad x \neq 0. \quad (14)$$

Как известно, для любой квадратной матрицы B справедливо тождество

$$(Bx, y) \equiv (x, B^T y) \quad \forall x, y \in \mathbb{R}^n;$$

поэтому соотношение (14) можно переписать в виде

$$(DL^T x, L^T x) > 0 \quad \forall x \in \mathbb{R}^n, \quad x \neq 0. \quad (15)$$

Поскольку L^T — неособенная матрица, то для любого $y \in \mathbb{R}^n$ найдется $x \in \mathbb{R}^n$, так что $y = L^T x$; это означает, что полагая $y = L^T x$ в (15) можно считать вектор y любым вектором из \mathbb{R}^n . Таким образом справедливо неравенство

$$(Dy, y) > 0 \quad \forall y \in \mathbb{R}^n, \quad y \neq 0. \quad (16)$$

Поскольку D — диагональная матрица с элементами d_i на главной диагонали, то $(Dy, y) = \sum_{i=1}^n d_i y_i^2$; следовательно неравенство (16) можно переписать в виде

$$\sum_{i=1}^n d_i y_i^2 > 0 \quad \forall y \in \mathbb{R}^n, \quad y \neq 0. \quad (17)$$

Если положить $y = e^{(i)}$, где $e^{(i)}$ — i -й базисный орт, то из неравенства (17) получим $d_i > 0$; ввиду произвольности i , $i = 1, 2, \dots, n$, приходим к выводу, что диагональ матрицы D состоит из положительных чисел. Введем в рассмотрение диагональную матрицу $D^{1/2}$ с положительными элементами на диагонали по формуле

$$D^{1/2} \stackrel{\text{def}}{=} \begin{pmatrix} \sqrt{d_1} & 0 & 0 & \dots & 0 \\ 0 & \sqrt{d_2} & 0 & \dots & 0 \\ 0 & 0 & \sqrt{d_3} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \sqrt{d_n} \end{pmatrix}.$$

Подставляя очевидное соотношение $D = D^{1/2}(D^{1/2})^T$ в (11), получаем $A = LD^{1/2}(D^{1/2})^T L^T$. Напомним, что для произведения матриц B и C справедливо соотношение $B^T C^T = (BC)^T$; пользуясь этим соотношением предыдущее равенство можно переписать в виде $A = LD^{1/2}(LD^{1/2})^T$. Очевидно, что матрица $G \stackrel{\text{def}}{=} LD^{1/2}$ нижнетреугольная, и, таким образом, из последнего представления матрицы A найдем $A = GG^T$, что и требовалось. ■