УДК 624.073.02

Тажмамат уулу К.- преп. ОшТУ
Tazhmamat uulu K. – teacher OshTU

ЧИСЛЕННАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА БЫСТРОГО ОБРАЩЕНИЯ МАТРИЦ

В статье приводится описание алгоритма быстрого обращения матриц, удобного для одновременного выполнения на нескольких компьютерах.

Ключевые слова: алгоритм, матрицы, компьютеры, параллельное программирование.

NUMERICAL IMPLEMENTATION OF THE ALGORITHM OF FAST MATRIX INVERSION

The paper describes the algorithm of fast matrix inversion, convenient for simultaneous execution on multiple computers.

Keywords: algorithm, matrix, computers, parallel programming.

Реализация алгоритма [1] получилась настолько простой, что ее существенную часть можно привести в объеме настоящей статьи. Реализация выполнена на языке Компонентный Паскаль.

```
PROCEDURE Req* (S:LibMatrices.Matrix;step:INTEGER):LibMatrices.Matrix;
VAR  A1,A,B,C,D,K,L,M,N,Sobr:LibMatrices.Matrix;
size,i,j:INTEGER;
BEGIN
IF  S#NIL  THEN
size:=LEN(S);
A1:= LibMatrices.New(size DIV 2);
(*Инициализация матриц A,B,C,D,K,L,M,N аналогично A1*)
Sobr:=LibMatrices.New(size);

IF  size>1  THEN
IF  ~LibMatrices.IsOne(S)  THEN
i:=0;
WHILE  i<(size DIV 2)  DO
j:=0;
WHILE  j<(size DIV 2)  DO
A[i][j]:=S[i][j];
B[i][j]:=S[i][j+size DIV 2];
C[i][j]:=S[i+size DIV 2][j];
D[i][j]:=S[i+size DIV 2][j+size DIV 2];
INC(j)
END;
INC(i)
END;

A1:=Req(A,step+1);

N:=Req(LibMatrices.Add(D,LibMatrices.Neg(
                                LibMatrices.Mult(LibMatrices.Mult(C,A1),B))),step+1);
```

```
                                M:=LibMatrices.Neg(LibMatrices.Mult(LibMatrices.Mult(N,C),A1));
L:=LibMatrices.Neg(LibMatrices.Mult(LibMatrices.Mult(A1,B),N));
K:=LibMatrices.Add(A1,LibMatrices.Neg(
                                LibMatrices.Mult(LibMatrices.Mult(A1,B),M)));

i:=0;
WHILE  i<(size DIV 2)  DO
j:=0;
WHILE  j<(size DIV 2)  DO
Sobr[i][j]:=K[i][j];
Sobr[i][j+size DIV 2]:=L[i][j];
Sobr[i+size DIV 2][j]:=M[i][j];
Sobr[i+size DIV 2][j+size DIV 2]:=N[i][j];
INC(j)
END;
INC(i)
END
ELSE
Sobr:=LibMatrices.One(size)
END
ELSE
Sobr[0][0]:=1/S[0][0]
END;
END;
RETURN Sobr
END Req;
```

Эта процедура вызывает саму себя два раза рекурсивно. Проверка случая единичной матрицы осуществляется при помощи функции IsOne. Add, Neg, Mult – соответственно, сложение, умножение на -1 и умножение матриц. Рассмотрим подробней ход работы программы. Нижние индексы при переменных обозначают этап рекурсии (см рис. 1). Серой рамкой обозначены те итерации, которые в этом примере присутствуют.

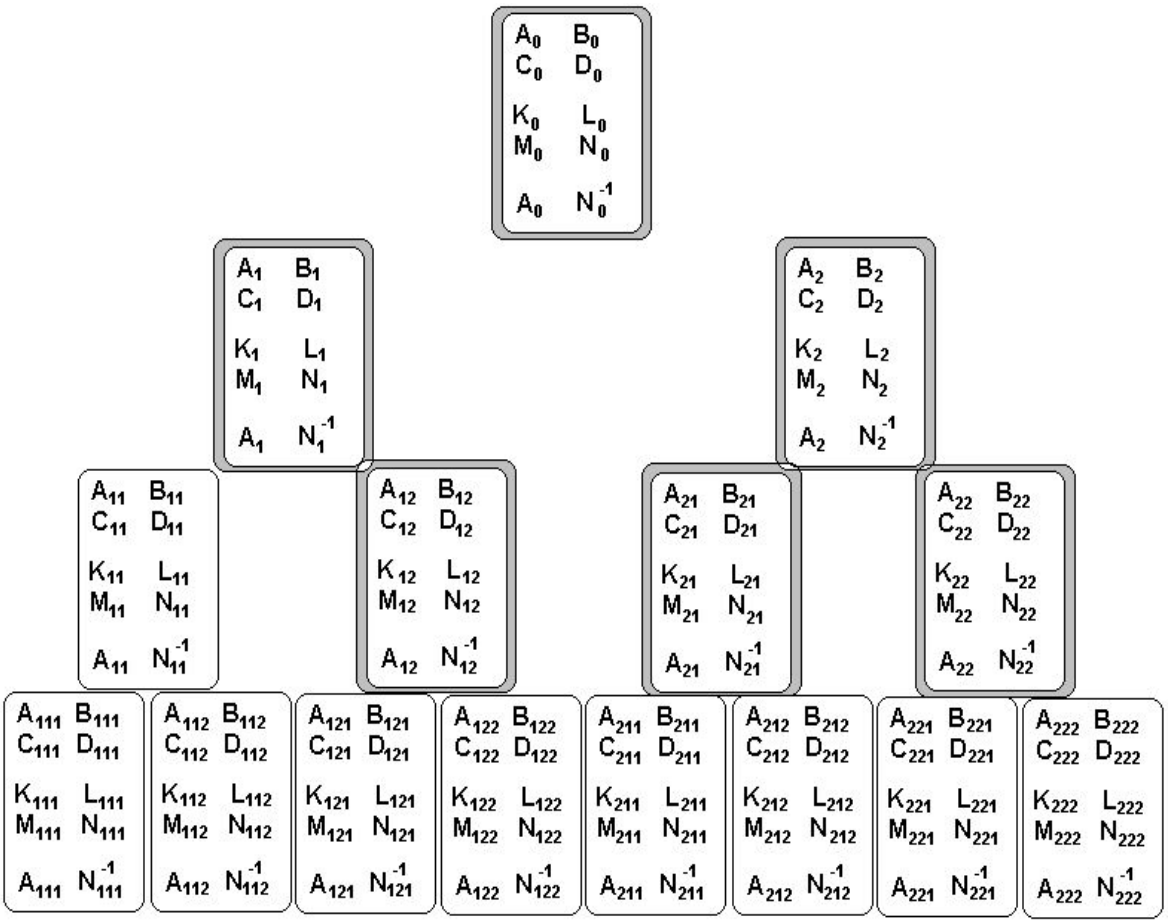

Рис.1. Схема обозначения индексов для разных итераций.

Исходная матрица, приведенная в [1].

$$\begin{vmatrix} 34 & 45 & 43 & 22 & 11 & 67 \\ 43 & 54 & 56 & 33 & 34 & 43 \\ 86 & 76 & 65 & 45 & 23 & 43 \\ 78 & 65 & 32 & 47 & 87 & 77 \\ 99 & 88 & 77 & 66 & 55 & 44 \\ 66 & 22 & 44 & 33 & 22 & 11 \end{vmatrix}$$

Вот результат ее дополнения и разбивки на клетки.

Матрица A_0.

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 34 & 45 \\ 0 & 0 & 43 & 54 \end{vmatrix}$$

Матрица B_0.

$$\begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 43 & 22 & 11 & 67 \\ 56 & 33 & 34 & 43 \end{vmatrix}$$

Матрица C_0.

$$\begin{vmatrix} 0 & 0 & 86 & 76 \\ 0 & 0 & 78 & 65 \\ 0 & 0 & 99 & 88 \\ 0 & 0 & 66 & 22 \end{vmatrix}$$

Матрица D_0.

$$\begin{vmatrix} 65 & 45 & 23 & 43 \\ 32 & 47 & 87 & 77 \\ 77 & 66 & 55 & 44 \\ 44 & 33 & 22 & 11 \end{vmatrix}$$

В свою очередь матрицу A_0 делим на клетки.

Матрица A_1.

$$\begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$$

Матрица B_1.

$$\begin{vmatrix} 0 & 0 \\ 0 & 0 \end{vmatrix}$$

Матрица C_1.

$$\begin{vmatrix} 0 & 0 \\ 0 & 0 \end{vmatrix}$$

Матрица D_1.

$$\begin{vmatrix} 34 & 45 \\ 43 & 54 \end{vmatrix}$$

Матрица A_1 единичная и не требует обращения. В данном случае вторая матрица N_1^{-1} ранга 2, получаемая по формулам (1) [1], равна матрице D_{11}, это надо учесть, чтобы не было путаницы. Разобьем на клетки матрицу N_1^{-1}. При этом получающиеся матрицы с единичным размером вырождаются в скаляры.

Скаляр A_{12}=34

Скаляр B_{12}=45

Скаляр C_{12}=43

Скаляр A_{12}=54

Формулы (1) [1], соответственно, вырождаются в формулы работы со скалярами. Вот результат вычислений по ним, образующий матрицу N_1.

Скаляр K_{12}=-0,54545

Скаляр L_{12}=0,454545

Скаляр M_{12}=0,434343

Скаляр N_{12}=-0,34343

Применив формулы (1) [1], получаем обращенную матрицу K_0

Матрица K_1.

$$\begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$$

Матрица L_1.

$$\begin{vmatrix} 0 & 0 \\ 0 & 0 \end{vmatrix}$$

Матрица M_1.

$$\begin{vmatrix} 0 & 0 \\ 0 & 0 \end{vmatrix}$$

Матрица N_1.

$$\begin{vmatrix} -0{,}54545 & 0{,}454545 \\ 0{,}43434\,3 & -0{,}34343 \end{vmatrix}$$

Вторая матрица N_0^{-1} ранга 4, которую следует обратить, получается по формулам (1) [1].

$$\begin{vmatrix} -64{,}7778 & -77{,}8889 & -265{,}768 & 415{,}6667 \\ -87{,}8889 & -71{,}4444 & -202{,}02 & 471{,}3333 \\ -72{,}1111 & -74{,}5556 & -273{,}889 & 465{,}6667 \\ -75{,}7778 & -125{,}889 & -450{,}222 & 817{,}6667 \end{vmatrix}$$

Делим ее на клетки.

Матрица A_2.

$$\begin{vmatrix} -64{,}7778 & -77{,}8889 \\ -87{,}8889 & -71{,}4444 \end{vmatrix}$$

Матрица B_2.

$$\begin{vmatrix} -265{,}768 & 415{,}6667 \\ -202{,}02 & 471{,}3333 \end{vmatrix}$$

Матрица C_2.

$$\begin{vmatrix} -72{,}1111 & -74{,}5556 \\ -75{,}7778 & -125{,}889 \end{vmatrix}$$

Матрица D_2.

$$\begin{vmatrix} -273{,}889 & 465{,}6667 \\ -450{,}222 & 817{,}6667 \end{vmatrix}$$

В свою очередь матрицу A_2 делим на клетки.

Скаляр A_{21}=-64,7778 Скаляр B_{21}=-77,8889

Скаляр C_{21}=-87,8889 Скаляр A_{21}=-71,4444

Формулы (1) [1], соответственно, вырождаются в формулы работы со скалярами. Вот результат вычислений по ним, образующий матрицу K_2.

Скаляр K_{21}=0,032218 Скаляр L_{21}=-0,03512

Скаляр M_{21}=-0,03963 Скаляр N_{21}=0,029211

Вторая матрица N_2^{-1} ранга 2, которую следует обратить, получается по формулам (1) [1].

$$\begin{vmatrix} -34{,}3169 & 35{,}82303 \\ 21{,}74532 & 237{,}3201 \end{vmatrix}$$

В свою очередь матрицу N_2^{-1} делим на клетки.

Скаляр A_{22}=-34,3169 Скаляр B_{22}=35,82303

Скаляр C_{22}=21,74532 Скаляр A_{22}=237,3201

Формулы (1) [1], соответственно, вырождаются в формулы работы со скалярами. Вот результат вычислений по ним, образующий матрицу N_2.

Скаляр K_{22}=-0,0266 Скаляр L_{22}=0,004015

Скаляр M_{22}=0,002437 Скаляр N_{22}=0,003846

Таким образом, по формулам (1) [1] получаем матрицы N_2, M_2, L_2, K_2.

Матрица K_2.

$$\begin{vmatrix} 0{,}005987 & -0{,}00568 \\ -0{,}10074 & -0{,}02517 \end{vmatrix}$$

Матрица L_2.

$$\begin{vmatrix} -0{,}0313 & 0{,}018054 \\ 0{,}129787 & -0{,}00819 \end{vmatrix}$$

Матрица M_2.

$$\begin{vmatrix} 0{,}00657 & 0{,}013518 \\ -0{,}01134 & 0{,}003042 \end{vmatrix}$$

Матрица N_2.

$$\begin{vmatrix} -0{,}0266 & 0{,}004015 \\ 0{,}002437 & 0{,}003846 \end{vmatrix}$$

Поднимаясь на верхний уровень рекурсии, получаем по формулам (1) [1] получаем матрицы N_0, M_0, L_0, K_0.

Матрца K_0.

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -0{,}01385 & -0{,}0025 \\ 0 & 0 & -0{,}01988 & 0{,}009122 \end{vmatrix}$$

Матрца L_0.

$$\begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0{,}035391 & 0{,}010771 & -0{,}03388 & 0{,}015883 \\ 0{,}032067 & 0{,}001759 & -0{,}00507 & -0{,}03196 \end{vmatrix}$$

Матрца M_0.

$$\begin{vmatrix} 0 & 0 & -0{,}01445 & 0{,}054098 \\ 0 & 0 & 0{,}070627 & -0{,}09493 \\ 0 & 0 & -0{,}02867 & 0{,}040077 \\ 0 & 0 & 0{,}026092 & -0{,}01498 \end{vmatrix}$$

Матрца N_0.

$$\begin{vmatrix} 0{,}005987 & -0{,}00568 & -0{,}0313 & 0{,}018054 \\ -0{,}10074 & -0{,}02517 & 0{,}129787 & -0{,}00819 \\ 0{,}00657 & 0{,}013518 & -0{,}0266 & 0{,}004015 \\ -0{,}01134 & 0{,}003042 & 0{,}002437 & 0{,}003846 \end{vmatrix}$$

Таким образом, применяя один и тот же набор формул, мы получаем обращенную матрицу.

$$\begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0{,}01385 & -0{,}0025 & 0{,}035391 & 0{,}010771302 & -0{,}033875978 & 0{,}015883 \\ 0 & 0 & -0{,}01988 & 0{,}009122 & 0{,}032067 & 0{,}001759268 & -0{,}005074179 & -0{,}03196 \\ 0 & 0 & -0{,}01445 & 0{,}054098 & 0{,}005987 & -0{,}005675465 & -0{,}031300169 & 0{,}018054 \\ 0 & 0 & 0{,}070627 & -0{,}09493 & -0{,}10074 & -0{,}025174176 & 0{,}129786812 & -0{,}00819 \\ 0 & 0 & -0{,}02867 & 0{,}040077 & 0{,}00657 & 0{,}013518236 & -0{,}026596254 & 0{,}004015 \\ 0 & 0 & 0{,}026092 & -0{,}01498 & -0{,}01134 & 0{,}003041566 & 0{,}002436978 & 0{,}003846 \end{vmatrix}$$

Сравним результаты обращения новым способом с обращением по методу «LU (lower/upper) decomposition with pivoting» (см табл. 1)

.

Индекс	33/11/33	88/66/33	77/55/77

Исходная матрица, L/U	-0.0138488170802077**3**	0.00384586042788431**4**	0.02659625393599357
Расширенная матрица, L/U	-0.0138488170802077**3**	0.00384586042788431**4**	0.02659625393599357
Новый метод	-0.0138488170802081**3**	0.00384586042788432	0.02659625393599377

Табл 1. Сравнений результатов расчета по разным методам.

Разница начинается на 15-16 знаках. Результаты по методу L/U для исходной матрицы и расширенной матрицы равны в связи с тем, что последовательность действий по умножению и сложению вещественных чисел в обоих случаях одинакова, поэтому погрешности накапливаются тоже одинаково.

Для избегания громоздкости мы использовали только маленькую исходную матрицу, однако указанный метод при добавлении некоторых прикладных процедур позволяет обращать очень большие матрицы, рангом порядка миллиардов и триллионов.

Литература:

1. Алгоритм быстрого обращения матриц. Тажмамат уулу К.