ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПММ

Кафедра вычислительной математики

МЕТОДЫ РЕШЕНИЯ СИСТЕМ С РАЗРЕЖЕННЫМИ МАТРИЦАМИ

СПОСОБЫ ХРАНЕНИЯ И ПРЕДСТАВЛЕНИЯ РАЗРЕЖЕННЫХ МАТРИЦ, ОПЕРАЦИИ НАД НИМИ

Методические указания
к спецкурсу

для студентов 3 курса
дневного и вечернего отделений
факультета ПММ

Составители:
И.А.Блатов
Т.Н.Глушакова
М.Е.Эксаревская

Воронеж – 2002

СОДЕРЖАНИЕ

§1. Введение

Строгого определения разреженной матрицы нет, но есть “нестрогие” определения, некоторые из которых мы здесь приведем.

О п р е д е л е н и е 1.1. ***Разреженная матрица*** (РМ) – это матрица, у которой “много” элементов равно нулю.

О п р е д е л е н и е 1.2. ***Разреженная матрица*** – это матрица, для которой использование алгоритмов , учитывающих наличие нулей, позволяет добиться экономии машинного времени и памяти по сравнению с традиционными методами.

РМ возникают при решении многих прикладных задач. Назовем некоторые из них:

1) дискретизация уравнений математической физики – разностные схемы и метод конечных элементов;
2) задачи линейного программирования (теория оптимизации);
3) задачи теории электрических цепей.

Основная задача курса *– научиться строить эффективные алгоритмы решения систем линейных алгебраических уравнений (СЛАУ)* $AU = f$ *(*A *– РМ), т.е. пытаться оптимизировать процесс решения с точки зрения затрат машинной памяти и времени.*

Возможности решения этой задачи связаны с игнорированием нулей матрицы A за счет того, что:

1) арифметитические операции с нулями не производятся;
2) нули не обязательно хранить в машинной памяти.

§2. Способы хранения и представления разреженных матриц

Все способы хранения РМ заключаются в том, чтобы хранить только ненулевые элементы матрицы или, может быть, небольшое количество нулей вместе с ними.

2.1. Разреженный строчный формат (РСФ)

Это наиболее широко используемая форма хранения РМ. Пусть есть прямоугольная $n \times m$ матрица $A = \left\{ a_{ij} \right\}$. Для ее представления в РСФ нужно три одномерных массива:

1) AN – массив ненулевых элементов матрицы A;
2) JA – массив соответствующих столбцовых индексов ненулевых элемен- тов матрицы A;
3) IA – так называемый “массив указателей ” – целочисленный массив, i-я компонента которого указывает, с какой позиции массивов AN и JA начинается описание i-й строки матрицы A. Здесь предусмотрена дополнительная компонента , которая является последней и указывает номер первой свободной позиции в массивах AN и JA.

Таким образом, описание i-й строки матрицы A хранится в позициях с $IA(i)$ до $[IA(i+1)-1]$ массивов AN и JA за исключением равенства I $IA(i+1)=IA(i)$, означающего, что i-я строка пуста. Следовательно, элементы записываются в массив по порядку следования строк. Если A имеет m строк , то массив IA содержит $(m+1)$ позицию.

Данный способ представления называют полным, т.к. представлена вся матрица A.

В зависимости от того, как записываются в каждой строке столбцовые индексы в массиве JA (по порядку возрастания или нет), различают упорядоченное или неупорядоченное представление соответственно.

Неупорядоченные представления нужны для алгоритмических удобств: результаты большинства матричных операций получаются неупорядоченными, и упорядочение их требует дополнительных затрат машинного времени, в то время как большинство алгоритмов для РМ не требует, чтобы представления были упорядоченными.

З а м е ч а н и е. Всюду в дальнейшем мы будем иметь дело с вещественными матрицами.

<u>Задача 1.</u> Написать для матрицы А упорядоченное представление в РСФ.

$$
\begin{array}{rc}
N \text{ столбцов :} & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
A = & \begin{pmatrix} 1 & 1 & 0 & 2 & 0 & 0 & 0 \\ 4 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.
\end{array}
$$

N позиций:	1	2	3	4	5	6	7	8
AN:	1	1	2	4	5	3	6	
JA:	1	2	4	1	3	5	6	
IA:	1	4	6	8	8			

З а м е ч а н и е. В первой позиции массива IA всегда стоит 1 .

<u>Задача 2.</u> По массивам AN, JA, IA восстановить матрицу A (с точностью до нулевых столбцов справа).

AN:	1	2	3	4	5	8	9
JA:	6	7	1	2	3	4	5
IA:	1	3	5	6	8		

Разобьем массивы $AN,\ JA$ по строкам:

$$
\begin{array}{rccccccc}
N \text{ позиции:} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
AN: & 1 & 2 & 3 & 4 & 5 & 8 & 9 \\
JA: & \underline{6\quad 7} & & \underline{1\quad 2} & & \underline{3} & \underline{4\quad 5} &
\end{array}
$$

Таким образом, в матрице A 4 строки и 7 столбцов, причем в 1-ой строке в 6 столбце стоит 1, в 7-м столбце – 2 и т.д.

$$
\begin{array}{cc}
 & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
A= & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 2 \\ 3 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 9 & 0 & 0 \end{pmatrix}
\end{array}
$$

(над столбцами: N столбцов : 1 2 3 4 5 6 7)

<u>Задача 3.</u> Написать для матрицы из задачи 1 полное, но неупорядоченное представление.

2.2. Разреженный столбцовый формат (РСтФ)

Здесь элементы хранятся не по строчкам, как в РСФ, а по столбцам. Столбцовые представления могут также рассматриваться и как строчные представления транспонированных матриц. Таким образом, в массиве JAT указывается строчный индекс соответствующего элемента, а элементы IAT указывают, с какой позиции начинается описание очередного столбца матрицы A.

<u>Задача 4.</u> Написать для матрицы A из задачи 1 упорядоченное столбцовое представление.

а)

$$
\begin{array}{rcccccccc}
N \text{ позиций:} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
ANT: & 1 & 4 & 1 & 5 & 2 & 3 & 6 & \\
JAT: & 1 & 2 & 1 & 2 & 1 & 3 & 3 & \\
IAT: & 1 & 3 & 4 & 5 & 6 & 7 & 8 &
\end{array}
$$

<u>Задача 5.</u> Транспонировать матрицу A из задачи 1 и написать для нее упорядоченный РСФ, сравнить результат с результатом задачи 5.

<u>Задача 6.</u> Записать матрицу A в неупорядоченном РСтФ.

$$
A=\begin{pmatrix} 0 & 0 & 1 & 3 & 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & 0 & 1 & 0 & 0 \end{pmatrix}.
$$

2.3. Строчный разреженный формат хранения симметричных матриц

Для симметричной матрицы $A=\left\{a_{ij}\right\}_{i,\,j=1}^{n}$ $(a_{ij}=a_{ji})$ достаточно хранить лишь ее диагональ и верхний (нижний) треугольник. При этом можно указать два способа хранения:

1) строчное представление диагонали и верхнего (нижнего) треугольника (РСФБД);
2) выделение диагональных элементов матрицы A в отдельный массив AD, а разреженным форматом представляется только верхний (нижний) треугольник матрицы A (причем в этом представлении диагональ считается нулевой) (РСФД).

<u>Задача 7.</u> Записать симметричную матрицу

$$A=\begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 3 & 0 \\ 1 & 1 & 0 & 3 \end{pmatrix}$$

а) в РСФБД;
б) в РСФД.

а) $AN:$ 2 1 1 1 3 3
$JA:$ 1 4 2 4 3 4
$IA:$ 1 3 5 6 7

б) $AN:$ 1 1
$JA:$ 4 4
$IA:$ 1 2 3 3 3
$AD:$ 2 1 3 3

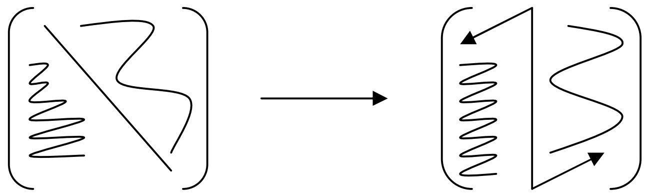
2.4. Диагональная схема хранения (ДСХ) ленточных матриц

О п р е д е л е н и е 2.1. Квадратная матрица $A=\left\{a_{ij}\right\}_{i,\,j=1}^{n}$ называется $(2m+1)$–диагональной или ***ленточной***, если $a_{ij}=0$ для всех $i,\,j$ таких, что $|\,i-j\,|>m$. Число $(2m+1)$ – это ***ширина ленты***, m – ***полуширина***.

Если $m<<n$, то такую матрицу следует рассматривать как разреженную.

Для хранения $(2m+1)$–диагональной несимметричной матрицы выделяется двумерный массив AD $(n\times(m+1))$, а для симметричной – $(n\times(m+1))$. Столбцами этого массива являются ненулевые диагонали матрицы A, которые хранятся, начиная с самой левой диагонали (самый левый столбец) и кончая самой правой (самый правый столбец) следующим образом:

сначала в столбец записывается главная диагональ, нижние кодиагонали – в остальных концах слева со сдвигом на одну позицию вниз при каждом смещение влево, а верхние кодиагонали – в правой части массива со сдвигом на одну позицию вверх при каждом смещении вправо. Схематично это можно изобразить так:



У симметричных матриц, как правило, записывают нижний треугольник.

Задача 8. Для квадратной матрицы A

1) найти полуширину ленты m;
2) указать размеры матрицы AD;
3) построить ДСХ.

а) $A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 8 & 6 & 0 & 0 & 0 \\ 0 & 8 & 3 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 4 & 10 & 0 & 0 \\ 0 & 0 & 0 & 10 & 5 & 11 & 12 \\ 0 & 0 & 0 & 0 & 9 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 \end{pmatrix}$;   б) $A = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 2 & 0 & 5 \end{pmatrix}$.

а) 1) $m = 2$;
2) $AD(7 \times 5)$;
3) $AD = \begin{pmatrix} & & 1 & 0 & 0 \\ & 0 & 2 & 8 & 6 \\ 0 & 8 & 3 & 0 & 0 \\ 9 & 0 & 4 & 10 & 0 \\ 0 & 10 & 5 & 11 & 12 \\ 0 & 9 & 6 & 0 & \\ 0 & 0 & 7 & & \end{pmatrix}$;

б) 1) $m = 2$;
2) $AD(5 \times 3)$;
3) $AD = \begin{pmatrix} & & 1 \\ & 0 & 2 \\ 1 & 0 & 0 \\ 0 & 0 & 3 \\ 2 & 0 & 5 \end{pmatrix}$.

Задача 9. Указать взаимно - однозначное соответствие между положением элемента в матрице A и его положением в массиве AD.

2.5. Профильная схема хранения (ПСХ) ленточных матриц

Эта схема предназначена для хранения симметричных ленточных матриц, когда внутри ленты находится слишком много нулей, и предыдущая схема становится нецелесообразной. Здесь хранится только нижняя полулента.

Введем числа $b_i = i - j_{\min}(i)$, где i – номер строки, $j_{\min}(i)$ – минимальный столбцовый индекс первого ненулевого элемента в i-й строке матрицы A, т.е. первый ненулевой элемент i-й строки находится на b_i позиций левее правой диагонали.

О п р е д е л е н и е 2.2. ***Профилем матрицы*** $A = \{a_{ij}\}_{i,j=1}^{n}$ называется сумма b_i: $pr(A) = profile(A) = \sum_{i=1}^{n} b_i$.

О п р е д е л е н и е 2.3. ***Оболочкой матрицы A*** называется совокупность элементов $\{a_{ij} : 0 < i - j \le b_i\}$.

Все элементы оболочки и диагональные элементы, упорядоченные по строкам, хранятся в одномерном массиве AN, причем диагональный элемент данной строки помещается в ее конец.

$$\dim AN = pr(A) + n.$$

В целочисленном массиве DA указываются номера диагональных элементов в массиве AN.

Таким образом, при $i > 1$ элементы i-й строки находятся в позициях от $[DA(i-1)+1]$ до $DA(i)$. Единственный элемент a_{11} первой строки хранится в $AN(1)$.

Задача 10. Для матрицы $A = \begin{pmatrix} 10 & & & & \\ 0 & 13 & & & \\ 0 & 0 & 17 & & \\ 0 & 1 & 0 & 18 & \\ 0 & 0 & 0 & 2 & 20 \end{pmatrix}$

1) подсчитать профиль, найти размерность массива AN;
2) построить ПСХ.

N позиции:	1	2	3	4	5	6	7	8
AN:	10	13	17	1	0	18	2	20
DA:	1	2	3	6	8			

$1)b_1 = 0,\ b_2 = 0,\ b_3 = 0,\ b_4 = 2,\ b_5 = 1,$

$pr(A) = 3,\ \dim AN = 3 + 5 = 8.$

Задача 11. По массивам AN и DA восстановить матрицу A:

AN: 9 8 7 6 5 4 3 2 1

DA: 1 3 5 6 7 9

§ 3. Операции над РМ. Алгебра РМ

Оперировать с РМ труднее, т.к. они заданы в упакованной форме. Мы будем работать с матрицами, которые заданы в РСФ (РСтФ). Поэтому любой алгоритм разбивается на **два этапа**.

1. Символический – определяется структура разреженности результата, который хотим получить, а также позиции элементов исходных РМ, с которыми нужно проводить арифметические действия, т.е. идет работа с векторами IA, JA; здесь же определяется объем оперативной памяти, необходимый для хранения промежуточных результатов и выходной информации.

2. Численный – непосредственно выполняются численные операции. В результате получаем число, матрицу или вектор.

3.1. Списки

3.1.1. Хранение списков, целых списков, кольцевых целых списков

Определение 3.1. ***Списком*** называется совокупность ячеек (позиций), связанных в том или ином порядке. Каждая ячейка содержит элемент списка и номер ячейки, в которой хранится следующий элемент списка.

Внутри каждого списка, по предположению, повторений нет.

В общем случае схема хранения списка состоит из трех массивов:

1) массив позиций N;
2) массив элементов A;
3) массив $NEXT$ – указатель позиций следующих элементов, и добавляется указатель начала списка IP.

Задача 12. Написать схему хранения чисел a, b, c, d в указанном порядке, если они хранятся в массиве A следующим образом:

N:	1	2	3	4	5	6	7
A:		b		d	a		c
Ответ: $NEXT$:		7		2	4;		$IP = 5$.

Задача 13. В каком порядке должны храниться числа $a,\ b,\ c,\ d,\ e,\ f$, если схема хранения выглядит следующим образом:

N: 1 2 3 4 5 6

A:	a	b	c	d	e	f
$NEXT$:		6	5	3	2	1

$IP = 4$

Ответ: $d,\ c,\ e,\ b,\ f,\ a$.

О п р е д е л е н и е 3.2. Если элементы списка являются целыми числами, то такой ***список*** называется ***целым***.

Пусть есть некоторый целый список A, элементы которого могут принимать значения $\{1,\ 2, \dots,\ n\}$.

О п р е д е л е н и е 3.3. Число n – максимальное значение элементов в списке – называется ***размахом списка***.

Пусть m – число элементов списка. Если $m \ll n$, то список называется разреженным целым списком (РЦС). Примерами таких списков могут служить массивы JA, IA. Целый список можно хранить с помощью одномерного массива длины n (скажем, JA) и указателя начала входа IP.

Задача 14. Записать целый список $3,\ 8,\ 4,\ 2$ в компактной форме.

N позиций:	1	2	3	4	5	6	7	8
JA:		3	8	2				4

$IP = 3$.

Число, хранимое в каждой позиции JA, дает одновременно значение элемента списка и адрес следующего элемента; таким образом, массив $NEXT$ уже не нужен.

Список называется кольцевым, если в его последнюю позицию поместить указатель на начальную позицию. У кольцевого списка нет ни начала, ни конца, но он требует хранимого отдельно указателя входа, который указывает на любую занятую позицию. Благодаря этому можно хранить несколько непересекающихся списков (кольцевых или нет) в одномерном массиве длины n, равной максимальному размаху списков, с использованием указателя входа IP.

Задача 15. Для непересекающихся списков $A_1,\ A_2,\ A_3$ написать схему хранения в одном массиве.

$A_1:\ 2,\ 8,\ 6,\ 3;\quad A_2:\ 5,\ 4,\ 0;\quad A_3:\ 7,\ 1,\ 10$

N позиций:	1	2	3	4	5	6	7	8	9	10
JA:	10	8	2	9	4	3	1	6	5	7

IP : 2 5 7

3.1.2. Слияние РЦС

Пусть заданы РЦС $A_1,\ A_2, \dots,\ A_n$. Результатом их слияния является список B, который содержит все элементы списков A_i ($i = 1,\dots,n$) без повторений (внутри каждого списка повторений нет).

Задача 16. Слить три списка A, B, C в один список M, где A: 2, 7, 3, 5; B: 3, 11, 5; C: 7, 2.

Список A записываем в M полностью, затем добавляем те элементы B, которых в A нет, потом элементы C, которых нет в A и B. Таким образом, получим следующий список M: 2, 7, 3, 5, 11.

Просматривать каждый раз список M, чтобы внести новый элемент, нерационально. Поэтому вводится массив переключателей Y и переключатель p. Обычно $p = 1$.

Число позиций массива переключателей должно быть равно максимальному размаху сливаемых списков (в данной задаче 11). В начальный момент во все позиции Y засылаем нули. Далее просматриваем первый из сливаемых списков, и если очередной просматриваемый элемент равен i, то в i-ю позицию массива переключателей засылается значение переключателя (то есть 1). При просмотре каждого текущего элемента следующего списка (пусть он равен j) мы обращаемся к массиву переключателей и добавляем элемент j в список M только в том случае, когда соответствующая позиция ”не включена”, т.е. в позиции переключателя стоит ноль. Таким образом мы просматриваем все списки.

Для данной задачи массив переключателей будет следующим:

N позиции:	1	2	3	4	5	6	7	8	9	10	11	
Y:	0	0	0	0	0	0	0	0	0	0	0	в начальный момент
	0	1	1	0	1	0	1	0	0	0	0	после просмотра A
	0	1	1	0	1	0	1	0	0	0	1	после просмотра B
	0	1	1	0	1	0	1	0	0	0	1	после просмотра C

3.1.3. Метод переменного переключателя (ПП)

При обработке РМ приходится сливать несколько групп списков. В целях экономии памяти используется только один массив переключателей (ЦМП), число позиций которого равно порядку матрицы (или числу строк, или числу столбцов).

В соответствие с алгоритмом предыдущего пункта перед каждым новым слиянием нужно было бы засылать во все позиции ЦМП нули, на что уходит значительное число машинных операций. Для экономии вводят переменный переключатель (ПП) p.

Сначала $p = 1$. Затем после каждого слияния группы списков p увеличивается на единицу (т.е. $p = p + 1$), что избавляет от необходимости засылки нулей во все позиции ПП.

Задача 17. Слить три списка A, B, C в M_1 и два списка D, E в M_2, используя ПП p.

$$\begin{cases} A: & 11, & 12, & 13 \\ B: & 4, & 5, & 11 \\ C: & 7, & 8, & 4 \end{cases}; \quad \begin{cases} D: 6,\ 7,\ 8,\ 9 \\ E: 7,\ 8,\ 4 \end{cases} \quad \begin{matrix} M_1: 11, 12, 13, 4, 5, 7, 8 \ (p=1); \\ M_2: \ 6,\ 7,\ 8,\ 9, 1, 2 \quad (p=2). \end{matrix}$$

N позиции:	1	2	3	4	5	6	7	8	9	10	11	12	13
Y:	0_2	0_2	0	0_1	0_1	0_2	0_{1_2}	0_{1_2}	0_2	0	0_1	0_1	0_1

3.2. Сложение разреженных векторов (РВ)

3.2.1. Метод расширенного вещественного накопителя (РВН)

На входе даны векторы A, B в РСФ, на выходе должны получить вектор $C = A + B$ в том же формате.

Алгоритм

1. **Символический этап**: определяются позиции ненулевых элементов C путем слияния списков JA, JB в JC.
2. **Численный этап**

   1^0. Вводится РВН X, число позиций которого равно размаху списка JC. Во все позиции вектора X с номерами из JC засылаем нули.

   2^0. Просматриваем JA и AN и прибавляем к содержимому позиций вектора X содержимое соответствующих позиций AN. Затем прибавляем BN к X.

   3^0. Выбираем из X нужные числа, чтобы сформировать CN, с помощью JC: $CN(i) = X(JC(i))$ и записываем их в соответствующие столбцы.

Задача 18. Сложить векторы A, B методом РВН.

$$\begin{cases} AN: & 0.2 & 0.3 & 0.4 & -0.7 \\ JA: & 10 & 3 & 7 & 4 \end{cases}; \qquad \begin{cases} BN: & 0.6 & 0.7 & 0.5 \\ JB: & 5 & 4 & 10 \end{cases}.$$

1. Сливаем списки JA, JB в JC: 10 3 7 4 5.
2. Вводим РВН X:

N позиции:	1	2	3	4	5	6	7	8	9	10	
X:			0	0	0		0			0	в начальный момент
			0.3	−0.7			0.4			0.2	прибавили AN
				0.7	0.6					0.5	прибавили BN
			0.3	0	0.6		0.4			0.7	

CN: 0.7 0.3 0.4 0 0.6

З а м е ч а н и е. Если мы не хотим, чтобы в CN были записаны нули, нужно “выбросить” нулевой элемент и соответствующий ему столбец. Тогда для данной задачи вектор C будет следующим:

$$\begin{array}{lllll} CN: & 0.7 & 0.3 & 0.4 & 0.6 \\ JC: & 10 & 3 & 7 & 5 \end{array}.$$

3.2.2. Метод расширенного целого указателя (РЦУ)

На входе даны векторы A, B в РСФ, на выходе должны получить вектор $C = A + B$ в том же формате. Здесь в качестве накопителя используется вектор CN.

Алгоритм

1. **Символический этап**: слияние списков JA, JB в JC.
2. **Численный этап**

1^0. Вводится расширенный целый массив указателей (РЦМУ) IX, размер которого равен размаху списка JC. Он заполняется с помощью просмотра массива JC, и $IX(j)$ указывает, в какой позиции массива CN будет храниться j-я компонента массива C, т.е. $C(j)$. Начальное состояние IX - нулевое.

2^0. В позиции CN засылаем нули. Затем сканируем JA, AN в CN, определяя с помощью IX позиции CN, где должны накапливаться значения ненулевых элементов. Затем сканируем JB, BN в CN:

$$CN(IX(JA(i))) = CN(IX(JA(i))) + AN(i);$$
$$CN(IX(JB(i))) = CN(IX(JB(i))) + BN(i).$$

<u>Задача 19.</u> Сложить вектора A, B методом РЦУ.

$$\begin{cases} AN: & 0.4 \quad 0.6 \quad 0.8 \quad -0.14 \\ JA: & 10 \quad\ 7 \quad\ 3 \quad\quad 4 \end{cases}; \quad \begin{cases} BN: & 0.1 \quad 0.2 \quad 0.3 \quad 0.4 \\ JB: & 9 \quad\ 5 \quad\ 3 \quad\ 1 \end{cases}.$$

1. N позиции: 1 2 3 4 5 6 7 — N позиции соответствующего столбца в JC

JC: 10 7 3 4 9 5 1 — N столбцов в CN

2. 1) N позиции: 1 2 3 4 5 6 7 8 9 10 — N элемента в JC (CN)

IX: 0 0 0 0 0 0 0 0 0 0

7   3 4 6   2   5 1

2) N позиции: 1 2 3 4 5 6 7

CN: 0.4 0.6 0.8 −0.14 — сканируем AN

0.3   0.1 0.2 0.4 — сканируем BN

$\overline{0.4}$ $\overline{0.6}$ $\overline{1.1}$ $-\overline{0.14}$ $\overline{0.1}$ $\overline{0.2}$ $\overline{0.4}$ — элементы CN

Таким образом, CN: 0.4 0.6 1.1 −0.14 0.1 0.2 0.4

JC: 10 7 3 4 9 5 1

3.3. Сложение РМ

Пусть на входе заданы две РМ A, $B\,(n\times m)$ в неупорядоченном РСФ. На выходе требуется получить $C = A + B$ в том же формате.

Для решения этой задачи нам потребуется вспомогательный ЦМП Y и РВН X. Размерность этих массивов равна числу столбцов A и B: $\dim Y = \dim X = m$. В основе алгоритма лежит идея поочередного сложения строк матриц A и B как РВ с помощью РВН.

Алгоритм

1. **Символический этап** (формирование JC, IC)

   1^0. $i = 1$ (номер строки).

   2^0. С помощью IA, JA и IB, JB определяем списки, соответствующие i-м строчкам матриц A и B и сливаем эти списки методом ПП с помощью ЦМП Y. Список, полученный в результате слияния, помещаем в 1-ю свободную позицию массива JC.

   3^0. Полагаем $i = i + 1$ и идем на 2^0. Повторяем действия до тех пор, пока $i \le n$. В результате получаем JC.

   4^0. Для получения IC достаточно на каждом этапе запоминать первую свободную позицию JC.

2. **Численный этап** (последовательное сложение строк A, B с помощью РВН)

   1^0. $i = 1$.

   2^0. С помощью JC, IC определяем список, полученный в результате слияния списков столбцовых индексов i-й строки массивов JA и JB.

   3^0. С помощью этого списка и РВН X складываем i-е строки матриц A и B, как РВ. Результат сложения помещаем в текущие свободные позиции вектора CN в соответствии с порядком столбцов из JC.

   4^0. $i = i + 1$ и идем на 2^0. Повторяем действия до тех пор, пока $i \le n$. В результате получим CN.

Замечание. После каждого сложения строк РВН X должен быть приведен в нулевое состояние.

Задача 20. Даны две матрицы A, B. Найти $C = A + B$.

$$\left\{\begin{array}{l} AN:\ 2\quad -1\quad 4\quad 3\quad 3\quad 7\quad -2\quad -1\quad 1\quad 1 \\ JA:\ \underline{3\quad 5}\quad \underline{1\quad 3\quad 4\quad 5}\quad \underline{1\quad 6}\quad \underline{2\quad 4}\ ; \\ IA:\ 1\quad 3\quad 7\quad 9 \end{array}\right.$$

$$\left\{\begin{array}{rl} BN: & 1 \quad -1 \quad 5 \quad -2 \quad 4 \quad 6 \quad -1 \quad 1 \quad 2 \\ JB: & \underline{1 \quad 3 \quad 6} \quad \underline{5} \quad \underline{1 \quad 2} \quad \underline{2 \quad 3 \quad 4} \\ IA: & 1 \quad 5 \quad 9 \quad 12 \quad 15 \end{array}\right. .$$

1. JC: $\underline{3 \quad 5 \quad 1 \quad 6} \quad \underline{1 \quad 3 \quad 4 \quad 5} \quad \underline{1 \quad 6 \quad 2} \quad \underline{2 \quad 4 \quad 3}$

   IC: $1 \quad 5 \quad 9 \quad 12 \quad 15$

   N позиции: $1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6$

   Y: $0_{1_{2_3}} \quad 0_{3_4} \quad 0_{12_4} \quad 0_{2_4} \quad 0_{1_2} \quad 0_{1_3}$

2. N позиции: $1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6$

$$\begin{array}{rcccccc l} X: & 0 & & 0 & & 0 & 0 & \text{заполнение 1 строки} \\ & & & 2 & & -1 & & \text{прибавили } AN \\ & \underline{1} & & \underline{-1} & & \underline{\quad} & \underline{5} & \text{прибавили } BN \\ & 1 & & 1 & & -1 & 5 & \text{результат} \end{array}$$

$$\begin{array}{cccccc l} 0 & 0 & 0 & 0 & 0 & 0 & \text{заполнение 2 строки} \\ 4 & & 3 & 3 & 7 & & \text{прибавили } AN \\ \underline{\quad} & & \underline{\quad} & \underline{\quad} & \underline{-2} & & \text{прибавили } BN \\ 4 & & 3 & 3 & 5 & & \text{результат} \end{array}$$

$$\begin{array}{cccccc l} 0 & 0 & 0 & 0 & 0 & 0 & \text{заполнение 3 строки} \\ -2 & & & & & -1 & \text{прибавили } AN \\ \underline{4} & \underline{6} & & & & \underline{\quad} & \text{прибавили } BN \\ 2 & 6 & & & & -1 & \text{результат} \end{array}$$

$$\begin{array}{cccccc l} 0 & 0 & 0 & 0 & 0 & 0 & \text{заполнение 4 строки} \\ & 1 & & 1 & & & \text{прибавили } AN \\ & \underline{-1} & \underline{1} & \underline{2} & & & \text{прибавили } BN \\ & 0 & 1 & 3 & & & \text{результат} \end{array}$$

AN: $1 \quad -1 \quad 1 \quad 5 \quad 4 \quad 3 \quad 3 \quad 5 \quad 2 \quad -1 \quad 6 \quad 0 \quad 3 \quad 1$

3.4. Скалярное умножение двух РВ

На входе даны два вектора A, B размерности n в РСФ. На выходе нужно получить число P – скалярное произведение A и B.

Для этого вводится РЦМУ IX, заполнение которого производится точно так же, как в п.3.2.2, но с помощью одного из сомножителей, например, с помощью JA. Символический этап здесь отсутствует, сразу начинается численный этап.

Сначала заполняем IX по JA. Если k – число элементов в JB, то для $i=1,\dots,k$ проверяем $IX(JB(i))$. Если это ноль, то переходим к следующей позиции массива JB; если нет, то $P = P + BN(I)\cdot AN(IX(JB(i)))$ и переходим к следующей позиции JB.

Другими словами, делаем следующее:

1) $P = 0$;
2) для $i = 1,\dots,k$

$$IX_{JA}(JB(I)) = 0$$

да $\swarrow$ $\searrow$ нет

$$i = i+1 \qquad P = P + BN(I)\cdot AN(IX_{JA}(JB(i)))$$

$$i = i+1$$

<u>Задача 21.</u> Найти скалярное произведение векторов

$$\begin{cases} AN: 1 \quad 2 \quad 3 \quad 4 \quad 5 \\ JA: 3 \quad 7 \quad 8 \quad 1 \quad 2 \end{cases}; \qquad \begin{cases} BN: 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11 \\ JB: 4 \quad 3 \quad 2 \quad 1 \quad 6 \quad 8 \end{cases}.$$

N позиции: 1 2 3 4 5 6 7 8

IX_{JA}: 4 5 1 0 0 0 2 3

$P = 0$, $i = 1,\dots,8$

$i=1$: $IX_{JA}(JB(1)) = 0$

$i=2$: $IX_{JA}(JB(2)) = 1 \neq 0$

$P = 0 + BN(2)\cdot AN(1) = 7\cdot 1 = 7$

$i=3$: $IX_{JA}(JB(3)) = 5 \neq 0$

$P = 7 + BN(3)\cdot AN(5) = 7 + 8\cdot 5 = 47$

$i=4$: $IX_{JA}(JB(4)) = 4$

$P = 47 + 9\cdot 4 = 83$

. . . .

3.5. Умножение РМ на РВ (результат – заполненный вектор)

На входе – матрица A и вектор B в РСФ, на выходе – вектор $C = AB$ как обычный одномерный массив. Здесь строки матрицы A рассматриваются, как РВ, и скалярно умножаются на РВ B с помощью рассмотренного в п.3.4 алгоритма.

Алгоритм

1. РЦМУ IX (начальное состояние – нулевое) заполняется с помощью вектора B.
2. $i = 1$.
3. С помощью IA определяем, в каких позициях JA содержится описание i-й строки.
4. Просматривается участок JA, соответствующий описанию i-й строки. Если компонента РМЦУ с номером $JA(k)$ (где k- текущая просматриваемая позиция) равна нулю, то переходим к просмотру следующей позиции; если не равна нулю, то производим умножение соответствующих компонент массивов AN и B и результат умножения прибавляем к содержимому ячейки $C(i)$.
5. Если просмотр строки окончен, то полагаем $i = i + 1$ и возвращаемся к 3. Повторяем до тех пор, пока не пройдем все строки.

Схематично этот **алгоритм** можно изобразить следующим образом.

$k = 1$

$C(k) = 0$

$IA(k+1) = IA(k)$

нет — да

$i = \overline{IA(k), IA(k+1) - 1}$ $\qquad$ $k = k + 1$

$IX(JA(I)) = 0$

да — нет

$i = i + 1$ $\qquad$ $C(k) = C(k) + AN(I) \cdot BN(IX_{JA}(JA(i)))$

$i = i + 1$

$k = k + 1$

<u>Задача 22.</u> Умножить РМ на РВ (результат – заполненный вектор).

N позиции:	1	2	3	4	5	6	7	8	9	10	11	12
AN:	1	3	5	7	9	2	4	6	8	10	2	6
JA:	1	4	6	3	2	5	7	8	9	10	1	3
IA:	1	4	9	11	12	13						

BN:	1	2	4	5	7	8	10
JB:	1	2	4	5	7	8	10

N позиции:	1	2	3	4	5	6	7	8	9	10
IX_{JB}:	1	2	0	3	4	0	5	6	0	7

$k=1;\quad C(1)=0;\quad IA(2)\ =IA(1)$

$i=1,\dots,3:\ i=1:\quad IX(JA(1))=IX(1)=1\neq 0$

$\qquad C(1)=0+1\cdot BN(1)=1$

$\qquad i=2:\quad IX(JA(2))=IX(4)=3\neq 0$

$\qquad C(1)=1+3\cdot BN(3)=1+3\cdot 4=13$

$\qquad i=3:\quad IX(JA(3))=IX(6)=0$

$k=2;\ C(2)=0;\ IA(3)\ \neq IA(2)$

$i=4,\dots,8:$

$\qquad i=4:\quad IX(JA(4))=IX(3)=0$

$\qquad i=5:\quad IX(JA(5))=IX(2)=2$

$\qquad C(2)=0+9\cdot BN(2)=9\cdot 2=18$

. . . .

3.6. Умножение РМ на заполненный вектор (результат – заполненный вектор)

Этот алгоритм незначительно отличается от предыдущего. Разница в том, что РМЦУ заполняется по k-й строке матрицы A, а не по вектору B. Здесь, по-видимому, можно обойтись без РМЦУ следующим образом.

Алгоритм

1. $k=1$.
2. Просматриваем содержимое k–й строки в JA и умножаем соответствующие компоненты AN на компоненты, накапливая результат в соответствующей ячейке $C(k)$ вектора C.

Схема решения

$k=1$

$C(k)=0$

$IA(k+1)\ =IA(k)$

нет — да

$i=IA(k),IA(k+1)-1$ $\qquad k=k+1$

$C(k)=C(k)+AN(i)\cdot B(JA(i))$

$k=k+1$

<u>Задача 23.</u> Умножить РМ A из задачи 22 на заполненный вектор

$B=(2\quad 4\quad 6\quad 8\quad 10\quad 9\quad)^T$.

$k=1;\ C(1)=0;\ IA(2)\ \neq IA(1)$

$i=1,\dots,3:$

$\qquad i=1:\quad C(1)=0+1\times 2=2$

$$i = 2: \quad C(1) = 2 + 3 \cdot 7 = 23$$
$$i = 3: \quad C(1) = 23 + 5 \cdot 3 = 38$$
$k = 2$; $C(2) = 0$; $IA(3) \neq IA(2)$

$i = 4,\ldots,8$:
$$i = 4: \quad C(2) = 0 + 7 \cdot 6 = 42$$

. . . .

3.7. Умножение РМ

На входе заданы РМ $A(n \times m)$, $B(m \times l)$ в неупорядоченном РСФ. На выходе должны получить матрицу $C = AB$ в РСФ.

Частным случаем является умножение РМ на РВ, когда результат –- РВ.

Алгоритм

1. Символический этап – определение векторов JC, IC. Для него потребуется ЦМП Y, размерность которого равна l. Начальное состояние Y – нулевое.

1^0. $i = 1$. $IC(1) = 1$.

2^0. Просматриваем по порядку столбцовые индексы элементов i-й строки в массиве JA. Для каждого просматриваемого индекса j в массиве столбцовых индексов JB просматриваем содержимое j-й строки. При этом для каждого ненулевого элемент j-й строки выполним следующие операции. Пусть просматриваемый индекс равен k. Если k-я позиция массива переключателей не “включена”, то “включаем” её и помещаем индекс k в первую свободную позицию массива JC. Если позиция уже “включена”, то переходим к просмотру следующего элемента вектора JB, ничего не добавляя к JC.

3^0. После окончания просмотра i-й строки матрицы A компоненте вектора $IC(i+1)$ присваиваем порядковый номер первой свободной позиции JC.

4^0. $i = i + 1$ и идем на 2^0. Значение ПП p увеличиваем на единицу.

Алгоритм

$Y(1) = Y(2) = \mathbf{K} = Y(l) = 0$

$i = 1$: $IC(1) = 1$

$n_j = 1$ (N элемента в JC)

$IA(i) = IA(i+1)$

нет да

$i = i + 1$

$n = \overline{IA(i), IA(i+1) - 1}$:

$$j = JA(n)$$

$$IB(i) = IB(i+1)$$

нет да

для $m = \overline{IB(j), IB(j+1)-1}$

$$k = JB(m)$$

$$Y(k) = i$$

нет да

$Y(k) = i$; $JC(n_j) = k$; $n_j = n_j + 1$

$IC(i+1) = n_j$; $i = i + 1$

2. Численный этап – определяется вектор CN. <u>Основная идея</u> – умножение с помощью РВН, размерность которого равна числу столбцов матрицы C (т.е. l), поэтому матрица C будет записываться по строкам.

<u>Пример</u>. Найти произведение двух матриц $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$.

Р е ш е н и е

Накопитель $\begin{array}{cc} x_1 & x_2 \\ (0, & 0) \end{array}$ – начальный момент. Берем первый элемент текущей строки матрицы A и умножаем его на все элементы соответствующей строки матрицы B и т.д.:

$$x_1 = 0,\quad x_2 = 0;\quad x_1 = x_1 + a_{11}b_{11},\quad x_2 = x_2 + a_{11}b_{12};$$

$$x_1 = x_1 + a_{12}b_{21},\quad x_2 = x_2 + a_{12}b_{22};\quad c_{11} = x_1,\quad c_{12} = x_2.$$

Рассмотрим **численный этап** в общем случае.

1^0. $i = 1$.

2^0. С помощью IC определяем, в каких позициях JC находится описание i-й строки матрицы C.

3^0. Просматриваем участок JC, соответствующий описанию i-й строки, и в позиции РВН X с номерами столбцовых индексов i-й строки этого участка засылаем нули.

4^0. С помощью IA определяем, в каком участке JA содержится описание i-й строки матрицы A.

5^0. Просматриваем выделенный участок (i-ю строку) массива JA. Для каждого просматриваемого элемента k, стоящего в m-й позиции массива JA, выполняем следующие операции:

a) определяем с помощью IB участок массива JB, описывающий k-ю строку массива B;

b) просматриваем выделенный участок JB и для каждого просматриваемого элемента j, стоящего в n-й позиции массива JB, полагаем $X(j) = X(j) + AN(m) \cdot BN(n)$.

6^0. Выбираем из РВН значения i-й строки матрицы C и присваиваем их соответствующим компонентам CN.

7^0. Если просмотр i-й строки завершен, то полагаем $i = i+1$ и возвращаемся на 2^0.

Алгоритм

$i = 1$

$IC(i) = IC(i+1)$

нет да

$i = i+1$

для $j = \overline{IC(i), IC(i+1)-1}$: $X(JC(j)) = 0$

$IA(i) = IA(i+1)$

нет да

$i = i+1$

для $m = \overline{IA(i), IA(i+1)-1}$: $k = JA(m)$

$A = AN(m)$

для $n = \overline{IB(k), IB(k+1)-1}$: $j = JB(n)$

$X(j) = X(j) + A \cdot BN(n)$

для $j = \overline{IC(i), IC(i+1)-1}$: $CN(j) = X(JC(j))$

$i = i+1$

Задача 24. Найти произведение двух РМ A и B.

$$\begin{cases} AN: 1 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \\ JA: 3 \quad 5 \quad 2 \quad 4 \quad 1 \quad 5 \\ IA: 1 \quad 3 \quad 5 \quad 7 \end{cases},$$

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 2 & 0 & 3 & 0 \\ 4 & 0 & 0 & 0 & 5 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 6 \\ 0 & 4 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 5 \end{pmatrix}.$$

1) $Y(1) = Y(2) = Y(3) = 0$;

$i = 1$: $IC(1) = 1$,

$n_j = 1$; $n = 1, 2$:

$n = 1$: $j = 3$, $m = 4$:

$k = 2$, $Y(2) = 1$,

$JC(2) = 3$, $n_j = 3$,

$IC(2) = 3$;

$i = 2$: $n = 3, 4$:

$n = 3$: $j = 2$, $m = 2, 3$:

$m = 2$: $k = 2$, $Y(2) = 2$,

$JC(3) = 2$, $n_j = 4$;

$m = 3$: $k = 3$, $Y(3) = 2$,

$JC(4) = 3$, $n_j = 5$,

$IC(3) = 5$;

$n = 4$: $j = 4$,

……………………………

$m = 2$: $k = 2$, $Y(2) = 2$,

$JC(3) = 2$, $n_j = 4$;

$m = 3$: $k = 3$, $Y(3) = 2$,

$JC(4) = 3$, $n_j = 5$,

$IC(3) = 5$;

$n = 4$: $j = 4$, $m = 5$,

$k = 1$, $Y(1) = 2$

…………………………………

$$\begin{cases} JC: 2 \quad 3 \quad 2 \quad 3 \quad 1 \quad 1 \quad 3 \\ IC: 1 \quad 3 \quad 6 \quad 8 \end{cases}$$

2) $i = 1$: $j = 1, 2$:

$j = 1$: $X(2) = 0$;

$j = 2$: $X(3) = 0$.

$m = 1, 2$;

$m = 1$: $k = 3$, $A = 1$,

$n = 4$: $j = 2$, $X(2) = 4$;

$m = 2$: $k = 5$, $A = 1$;

$n = 6$: $j = 3$, $X(3) = 5$;

……………………………

CN : 4 5 6 12 6 4 25

$n = 3$: $j = 2$, $m = 2, 3$:

CN : 4 5 6 12 6 4 25

3.8. Транспонирование РМ

На входе дана матрица $A(n \times m)$ в неупорядоченном РСФ. На выходе должны получить A^T - транспонированную матрицу, заданную в РСФ.

З а м е ч а н и е. Если задано строчное представление матрицы A, то его можно рассматривать как столбцовое представление A^T и наоборот, поэтому задача транспонирования A сводится к нахождению РСТФ матрицы A.

Если просматривать массивы «в лоб», то это очень неэффективно. Поэтому вводятся m целых и m вещественных списков длины n и m – указателей первой свободной позиции каждого списка. В начальный момент времени все списки пустые, т.е. первая свободная позиция каждого списка – первая.

На практике списки организуются непосредственно в массивах JAT и ANT, а указатели первой свободной позиции каждого списка – в IAT. Таким образом, дополнительной памяти не требуется, и перемещение данных в памяти машины сведено к минимуму.

Алгоритм

С помощью массивов JA и IA просматриваем по порядку строки матрицы A.

1. $i = 1$.
2. С помощью IA определяем, в каких позициях JA содержится описание i-й строки матрицы A.
3. Просматриваем содержимое i-й строки в JA и для каждого j (где j - столбцовый индекс просматриваемого элемента) добавляем число i в первую свободную позицию j-го целого списка, увеличивая при этом указатель первой свободной позиции списка на единицу. В соответствующую позицию j-го вещественного списка добавляем соответствующий элемент списка AN.
4. $i = i + 1$.
5. Объединяем по порядку их нумерации вещественные и целые списки, получаем списки JAT и ANT. Список IAT получается с помощью подсчета числа элементов заполняемых списков.

З а м е ч а н и я

1. С помощью этого алгоритма получается упорядоченное представление транспонированной матрицы даже в том случае, если представление исходной матрицы было неупорядоченным.
2. Применение данного алгоритма к матрице A дважды позволяет получить из неупорядоченного РСФ матрицы A её упорядоченное представление.

Задача 25. Транспонировать матрицу A из задачи 22 (номер списка – это номер столбца).

вещественные списки целые списки

1) 1 2	6) 5	1) 1 4	6) 1
2) 9	7) 4	2) 2	7) 2
3) 7 6	8) 6	3) 2 5	8) 2
4) 3	9) 8	4) 1	9) 3
5) 2	10) 10	5) 2	10) 3

$$\begin{cases} ANT: & 1 \quad 2 \quad 9 \quad 7 \quad 6 \quad 3 \quad 2 \quad 5 \quad 4 \quad 6 \quad 8 \quad 10 \\ JAT: & \underline{1 \quad 4} \quad \underline{2} \quad \underline{2 \quad 5} \quad \underline{1} \quad \underline{2} \quad \underline{1} \quad \underline{2} \quad \underline{2} \quad \underline{3} \quad \underline{3} \\ IAT: & 1 \quad 3 \quad 4 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11 \quad 12 \quad 13 \end{cases}$$

Задача 26. Получить из неупорядоченного представления в РСФ матрицы A из задачи 22 её упорядоченное представление (применяя алгоритм $A \to A^T$ дважды).

3.9. Перестановка строк и столбцов разреженной матрицы

3.9.1. Перестановка столбцов

Дана матрица $A(n \times m)$. Пусть J – целочисленный вектор размерности m, который задает перестановку столбцов $J = \{J(1), J(2), \ldots J(m)\}$. Здесь $J(i)$ указывает номер, который после перестановки должен иметь i-ый столбец.

Перестановка сводится к преобразованию JA по следующей схеме: $JAP(i) = J(JA(i))$, где $JAP(i)$ – номер столбца после перестановки.

Таким образом, на входе дана матрица A в РСФ и вектор перестановок J. На выходе требуется получить РСФ матрицы с переставленными столбцами.

Алгоритм сводится только к преобразованию массива JA: по порядку просматриваются все его компоненты, и каждая компонента $JA(i)$ заменяется на $J(JA(i))$. Численные значения ненулевых компонентов, хранимые в AN, остаются прежними, IA также не меняется.

Задача 27. В матрице A из задачи 22 переставить 2 и 5, 4 и 7 столбцы.

N позиции: 1 2 3 4 5 6 7 8 9 10

$$J = \{1,\ 5,\ 3,\ 7,\ 2,\ 6,\ 4,\ 8,\ 9,\ 10\}$$

$JAP(1) = J(JA(1)) = J(1) = 1$

$JAP(2) = J(JA(2)) = J(4) = 7$

$JAP(3) = J(6) = 6$

$JAP(4) = J(3) = 3$

$JAP(5) = J(2) = 5$

$JAP(6) = J(5) = 2$

$JAP(7) = J(7) = 4$

$JAP(8) = J(8) = 8$

$JAP(9) = J(9) = 9$

$JAP(10) = J(10) = 10$

$JAP(11) = J(1) = 1$

$JAP(12) = J(3) = 3$

3.9.2. Перестановка строк

Перестановка строк матрицы соответствует перестановке столбцов транспонированной матрицы, поэтому алгоритм перестановки строк получается комбинированием двух предыдущих алгоритмов.

Задача 28. Переставить 1 и 5 сроки матрицы A из задачи 22.

Алгоритм решения

1. Транспонировать матрицу A (получить матрицу A^T) .
2. Переставить 1-ый и 5-ый столбцы матрицы A^T (получить ATP).
3. Транспонировать матрицу ATP .

§4. Метод Гаусса для РМ

Рассмотрим СЛАУ вида $Ax = f$, где $x \in R^n$, $f \in R^n$, A – симметричная матрица $n \times n$. Решение этой системы будем искать с помощью метода Гаусса. Будем считать, что ведущий элемент гауссова исключения всегда находится на главной диагонали матрицы A, то есть перестановка строк и столбцов не нужна.

Особенности Гауссова исключения

1. Существует две модификации – по строкам и по столбцам, которые отличаются порядком выполнения операций исключения. Обычно используют гауссово исключение по столбцам, но для РМ используют гауссово исключение по строкам.

2. Вначале строим LU – разложение матрицы A, а затем решаем две треугольные системы $LU = f$, $Ux = y$, где U – верхняя треугольная матрица с единичными диагональными элементами, L – нижняя треугольная.

Если главные миноры A отличны от нуля, то матрицу A можно представить в виде $A = U^T \tilde{D} U$, где $\tilde{D}$ – диагональная матрица. Если матрица A еще и положительно определена, то разложение можно представить в виде $A = U^T U$ – ***разложение Холецкого.***

4.1. Построение разложения $U^T \tilde{D} U$

На входе задана матрица A в верхнетреугольном неупорядоченном РСФ: массивы AN, JA, IA, AD. На выходе должны получить матрицу в РСФД и диагональную матрицу $\tilde{D}$.
Особенности символического этапа –- мы должны исключить элементы ниже главной диагонали, а у нас только верхний треугольник, поэтому сразу мы не можем определить позицию тех элементов, которые нужно исключать при обработки текущей i-й строки. Оказывается, что столбцовые индексы i-й строки, которые нужно обнулить, совпадают со строчными индексами элементов i-ого столбца выше главной диагонали той матрицы, которая получилась к данному моменту при реализации алгоритма. Таким образом, на i-м шаге мы должны пройти по i-ому столбцу, определить все строчные индексы ненулевых элементов i-ого столбца –- это и будут те столбцовые индексы элементов i-й строки, которые нужно обработать.

4.1.1. Символический этап

4.1.1.1. Составление ассоциированного списка i-ого столбца

Ассоциированный список (АС) i-ого столбца – это список номеров тех строк, которые участвуют при обработке i-й строки в методе Гаусса. Фактически каждому столбцу ставится в соответствии совокупность строк, причем каждая строка относится только к одному столбцу (то есть ни к какому другому она уже не приписывается), поэтому к i-ому столбцу приписываются только те строки матрицы A, у которых ненулевой элемент в i-м столбце является первым ненулевым элементом данной строки справа от диагонали.

Алгоритм

1. $i = 2$ (так как первую строку обрабатывать не надо).
2. Проходим по $(i-1)$-й строке справа от диагонали. Предположим, что первый ненулевой столбцовый индекс равен j, тогда $(i-1)$-ую строку приписываем j-му столбцу и помещаем номер этой строки в первую позицию j-го АС. Если действовать таким образом, то к моменту обработки строки АС будет уже сформирован.
3. $i = i+1$ и повторяем все действия до тех пор, пока не пройдем все строки.

АС каждого столбца удобно хранить как кольцевой РЦС, и все эти списки можно хранить в целочисленном массиве JP размерности n, а массив указателей будет показывать вход в каждый список.

4.1.1.2. Алгоритм символического этапа

1. Для $i = 1$ переносим портрет первой строки матрицы A в портрет пер-
вой строки матрицы U без изменений.
2. $i = 2$.
3. Заканчиваем формирование i-ого АС.

4. Просматриваем i-й АС и для каждого элемента j этого списка определяем список только тех столбцовых индексов ненулевых элементов j-й строки, номер которых больше i.
5. Сливаем все полученные списки, определенные на 4, и список i-й строки матрицы с помощью ПП. Результатом слияния будет портрет i-й строки матрицы U.
6. Приписываем список, полученный на 5, к первой свободной позиции массива JU.
7. Определяя число элементов списка, полученной на 5, вычисляем очередную компоненту массива IU, которая будет указывать начало $(i+1)$-й строки (значением этой компоненты будет номер первой свободной позиции массива JU после присоединения к нему списка, полученного на 5).
8. $i = i+1$ и обрабатываем следующую строку, перейдя на 3, и так до тех пор, пока $i \le n$.

Результатом будет портрет матрицы U.

4.1.2. Численный этап

4.1.2.1. Составление АС i-го столбца. Отличия от символического этапа

1. На символическом этапе каждая строка приписывалась только к одному столбцу. Здесь же каждая строка должна быть приписана ко всем тем столбцам, с которыми она имеет ненулевое пересечение, то есть если в j-ом столбце i-й строки $(i < j)$ содержится ненулевой элемент, то она должна быть приписана к j-ому столбцу, так как важны численные значения элементов.

2. Нам потребуется упорядоченное представление матриц A и U, поэтому перед реализацией численного этапа нужно перейти от неупорядоченного представления к упорядоченному двукратным транспонированием.

4.1.2.2. Алгоритм численного этапа

1. $i = 1$. Определяем в AN и JA содержимое первой строки матрицы A, делим эту строку на диагональный элемент a_{11} (который хранится в первой позиции массива AD) и помещаем первую строку в соответствующие позиции массива UN. $\tilde{D}(1) = AD(1)$.
2. $i = i+1$.
3. Просматриваем $(i-1)$-ю строку матрицы U (массив JU). Если $(i-1)$-я строка пуста, то переходим к 5. Если нет, то определяем столбцовый индекс j первого ненулевого элемента этой строки правее главной диагонали. Приписываем $(i-1)$-ю строку к АС j-го столбца.

4. Полагаем $IUP(i-1) =<$ номер позиции в JU, содержащей столбцовый индекс первого ненулевого элемента $(i-1)$-й строки правее главной диагонали $>$.
5. 1^0. С помощью массива IU определяем в JU описание i-й строки матрицы U. В позиции РВН X с номерами столбцовых индексов выделенного участка (если он непустой) и номером строки i засылаем нули.

   2^0. В i-ю позицию РВН X помещаем элемент $AD(i)$. Если i-я строка не пуста, то в позиции РВН, соответствующие портрету i-й строки матрицы A из массива AN.
6. Просматриваем АС i-ого столбца. Если он пустой, то переходим к 7. Если непустой, то для каждого элемента j этого списка делаем следующие операции:

   а) с помощью элементов $IUP(j)$ определяем участки массивов JU, UN, в которых содержится описание элементов j-й строки матрицы U, имеющих столбцовые индексы $\geq i$;

   б) умножаем элементы выделенного участка (массива UN) на число $(-U_{ij} \cdot \tilde{D}(j))$ (элемент U_{ij} находим в UN);

   в) прибавляем новые значения элементов выделенного участка к содержимому соответствующих ячеек РВН;

   г) полагаем $IUP(j) = IUP(j)+1$;

   д) приписываем j-ю строку к АС k-го столбца, где k – столбцовый индекс ненулевого элемента с номером $IUP(j)$, следующий за i-м в j-й строке матрицы U.

7. Если просмотр АС закончен, то выбираем элемент РВН $X(i)$, соответствующий диагональному элементу этой i-й строки, и помещаем его в i-ю позицию массива $\tilde{D}$, а содержимое РВН делим на элемент, содержащийся в i-й позиции РВН (то есть на $\tilde{D}(j)$), после чего в РВН содержится i-я строка матрицы U.
8. Выбираем из РВН ненулевые элементы i-й строки матрицы U (за исключением i-го столбца, в котором находится 1) и помещаем их в UN таким образом, чтобы j-ому столбцу соответствовал элемент $X(j)$.
9. $i = i+1$ и переходим к 3.

4.1.3. Примеры

Задача 29. Найти треугольное разложение для матрицы A.

$$A=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 4 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 5 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 6 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 7 \end{pmatrix}; \quad \begin{array}{lcccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ AN: & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ JA: & \underline{6} & \underline{4} & \underline{5} & \underline{5} & \underline{7} & \underline{6} & \underline{7} & \underline{6} \\ IA: & 1 & 2 & 4 & 6 & 8 & 9 & & \\ AD: & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \end{array}.$$

1. Символический этап

N строки	N столбца	N АС (N столбца)
1	– 6	1) –
2	– 4, 5	2) –
3	– 5, 7	3) –
4	– 6, 7	4) 2
5	– 6	5) 2, 3
6	–	6) 1, 4, 5
7	–	7) 3, 4

$$\begin{array}{lccccccccccccc} JU: & \underline{6} & \underline{4} & \underline{5} & \underline{5} & \underline{7} & \underline{5} & \underline{6} & \underline{7} & \underline{5} & \underline{7} & \underline{7} \\ IU: & 1 & 2 & 4 & 6 & 9 & 11 & 12 & & & & \end{array}$$

2. Численный этап

N АС (N столбца)	
1) –	$\tilde{D}(1)=1$
2) –	$i=2: \quad IUP(1)=1$
3) –	$X(4)=0+1/2=1/2$
4) 2	$X(5)=0+1/2=1/2$
5) 2, 3	$X(2)=0+2/2=1$
6) 1, 4, 5	$\tilde{D}(2)=2$
7) 3, 4	$i=3: \quad IUP(2)=2$
	$X(5)=1/3$
	$X(7)=1/3$
	$X(3)=1$

$$\tilde{D}(3) = 3$$

$$i = 4: \quad IUP(3) = 4$$

$$X(5) = 0 - \frac{1}{4} / \frac{15}{4} = -1/15$$

$$X(6) = 0 + 1/\frac{15}{4} = 4/15$$

$$X(7) = 0 + 1/\frac{15}{4} = 4/15$$

$$X(4) = 4 - 1/4 = 15/4$$

$$\tilde{D}(4) = 15/4$$

$j = 2:$ столбцовые индексы 4, 5.

$$4: \quad UN(2)\cdot(-a_{24}/\tilde{D}(2)) = UN(2)\cdot(-AN(2)/\tilde{D}(2)) = -\frac{1}{2}\cdot\frac{1}{2} = -\frac{1}{4}$$

$$5: \quad UN(3)\cdot(-a_{24}/\tilde{D}(2)) = -\frac{1}{2}\cdot\frac{1}{2} = -\frac{1}{4}$$

$$IUP(2) = 3, \quad IUP(2) = 4$$

$$j = 3: \quad IUP(3) = 4, 5$$

$$i = 5: \quad IUP(4) = 6, 7$$

$$X(6) = 0 + 1 - 16/225 = 209/225$$

$$X(7) = 0 - 1/9 - 16/225 = -41/225$$

$$X(5) = 0 + 5 - 1/4 - 1/9 + 4/225 = 4\frac{197}{300}$$

$$\tilde{D}(5) = 19/4 - 1/9 - 4/225 = \mathbf{K}$$

$$UN: \quad 1 \quad \frac{1}{2} \quad \frac{1}{2} \quad \frac{1}{3} \quad \frac{1}{3} \quad -\frac{1}{15} \quad \frac{4}{15} \quad \frac{4}{15}\mathbf{K}$$

<u>Задача 30.</u> По портрету (структуре) несимметричной РМ A определить максимальное число ненулевых элементов после приведения матрицы к треугольному виду.

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} \times & \times & & \times \\ \times & & \times & \\ & \times & & \times \\ \times & & \times & \end{pmatrix} \end{array} \rightarrow \begin{pmatrix} \times & \times & & \times \\ \otimes & \oplus & \times & \oplus \\ & \otimes & \oplus & \times \\ \otimes & + & \otimes & \oplus \end{pmatrix} \rightarrow \begin{pmatrix} \times & \times & & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{pmatrix}$$

Здесь $\oplus$ – ненулевой элемент из других строк, $\otimes$– обнуление элемента исходной матрицы, $+$ – обнуление элемента из других строк.

4.2. Обратный ход метода Гаусса

Он состоит в решении системы

$$U^T \tilde{D} U x = f \, , \tag{1}$$

где $\tilde{D}$ –- диагональная матрица, U – верхнетреугольная с единицами на главной диагонали.

Обратный ход метода Гаусса для системы (1) заключается в решении трех систем.

$$\begin{cases} U^T z = f & (2) \\ \tilde{D} w = z \quad . & (3) \\ U x = w & (4) \end{cases}$$

Таким образом, нужно решить две системы с треугольными матрицами и одну с диагональной.

Решаем систему (2) при помощи прямой подстановки: по порядку, начиная с первой, просматриваем строки матрицы U^T и вычисляем компоненты по формулам

$$z_1 = f_1 / U_{11}^T = f_1 \, , \qquad z_i = f_i - \sum_{j=1}^{i-1} U_{ij} z_j \, .$$

Для системы (3) имеем $w_i = z_j \tilde{D}_{ii}$. Система (4) решается обратной подстановкой: $x_n = w_n \, , \quad x_i = w_i - \sum\limits_{j=i+1}^{n} U_{ij} x_j \, .$

Можно также для решения системы (4) перейти от РСФ к столбцовому, после чего система (4) решается аналогично (2).

4.3. Вывод РМ на печать или экран

Для представления матрицы можно выбрать одну из следующих форм.

1. Представление в виде полной матрицы

Для каждой строки значения ненулевых элементов загружаются в полный вещественный массив, которому предварительно придано начальное нулевое состояние. Строка массива выводится на печать или дисплей, и алгоритм переходит к обработки следующей строки матрицы. Очень удобно было бы различать визуально внутрипортретные нули (т.е. нули, перемещенные в AN или AD вследствие взаимного сокращения при вычислении) и внепортретные нули (т.е. нули, о которых заранее известно, что они будут точными нулями, и которые поэтому не включаются в JA). В позициях, соответствующих внепортретным нулям, можно печатать нечисловой символ, например $*$.

Разумеется, практически этот метод приложим к тем ситуациям, когда достаточно исследовать малую часть матрицы или сама матрица достаточна мала.

2. **Для каждой строки печатается её номер**, а затем ненулевые элементы этой строки и за каждым из них – в скобках – соответствующий столбцовый индекс. Еще лучше было бы упорядочить ненулевые элементы перед печатанием. Достоинство этого метода в том, что он сокращает пространство, занимаемое выводимой строкой; однако он не дает такого ясного представления о взаимном расположении соседних строк, как первый метод.

3. **Портрет матрицы можно вывести на устройство с высокой разрешающей способностью,** например, дисплей или матричное печатающее устройство. Нужно только, чтобы матрица была не слишком велика или чтобы было достаточно её рассматривать по частям.

4.4. Выбор порядка исключения в методе Гаусса (упорядочение строк и столбцов)

В процессе гауссова исключения происходит заполнение матрицы, причем объем и структура этого заполнения весьма существенно зависят от выбора порядка исключения. Поэтому следует выбирать такую строку, в которой больше нулей, – чтобы минимизировать число ненулевых элементов, когда строка «обрушивается» на все другие строки, а также столбец с максимальным числом нулей, чтобы использовать поменьше строк.

О п р е д е л е н и е. Для элемента a_{ij} произведение числа ненулевых элементов в i-ой строке и j-ом столбце называется ***ценой Марковица*** элемента a_{ij}.

Ненулевой элемент a_{ij} следует выбирать так, чтобы цена Марковица этого элемента была минимальной или не слишком большой. Эта идея называется ***стратегией Марковица***. Она позволяет оптимизировать выбор ведущего элемента.

Но такой выбор влияет на устойчивость процесса гауссова исключения, так как если ведущий элемент мал, то возможна потеря устойчивости вычислений.

4.5. Вычислительные ошибки в гауссовом исключении

При реализации метода Гаусса приходится выполнять арифметические действия типа $b = a - lU$. Для операций с плавающей точкой границы ошибки обычно устанавливаются следующим образом: $fl(x \mathbf{o}\, y) = (x \mathbf{o}\, y)(1+e)$, где символ $\mathbf{o}$ обозначает одну из элементарных операций $+, -, \times, /$; $(x \mathbf{o}\, y)$ – точный результат операции; $fl(x \mathbf{o}\, y)$ – округленный результат; $|\, e \,| \le e_M$, где e_M – машинная точность.

Пусть $|\, a \,|, |\, b \,| < a$, тогда для оценки погрешности имеем

$$|\, e \,| \le a \cdot e_M \cdot \frac{1}{1-e_M} \cdot \left(\frac{1}{1-e_M} + 2 \right).$$

Таким образом, для того чтобы вычислительная погрешность была не слишком велика, не следует допускать чрезмерного роста чисел $a,\ l,\ U$. А в методе Гаусса $a,\ l,\ U$ – элементы k-х промежуточных матриц. Поэтому вводится показатель промежуточного роста в методе Гаусса $a_k = \max \left| a_{ij}^k \right| \quad \left(A_k = \left\{ a_{ij}^k \right\} \right)$, который должен быть не слишком велик.

4.6. Стратегия, осуществляющая компромисс между оптимизацией устойчивости и оптимизацией алгоритма

Пусть сделаны первые k шагов метода Гаусса с выбором главного элемента по столбцу. Возьмем число $U:\ 0 < U << 1$. Определим в k-ом столбце множества $R_{st} = \left\{ a_{ik} : \left| a_{ik} \right| \ge U \cdot \max \left| a_{ij}^k \right| \right\}$, R_{sp} –- множество ненулевых элементов k-го столбца, предпочтительных с точки зрения разреженности, например таких, для которых цена Марковица не превосходит некоторого фиксированного числа.

Выбор ведущего элемента осуществляется в множестве $R_{piv} = R_{sp} \cap R_{st}$, ориентируясь на R_{sp} или R_{sp}.

Литература

1. Писсанецки С. Технология разреженных матриц / С. Писсанецки. – М .: Мир, 1988.
2. Джордж А., Лю Д. Численные методы решения больших разреженных систем уравнений / А. Джордж, Д. Лю. – М.: Мир, 1984.

3. Эстербю О., Златев З. Прямые методы для разреженных матриц / О. Эстербю, З. Златев. – М.: Мир, 1987.
4. Тьюарсон Р. Разреженные матрицы / Р. Тьюарсон. – М.: Мир, 1977.

Составители: Блатов Игорь Анатольевич
Глушакова Татьяна Николаевна
Эксаревская М.Е.

Рецензент Покорная И.Ю.

Редактор Тихомирова О.А.

Заказ № от 2002 г. Тир. 50 экз. Лаборатория оперативной полиграфии ВГУ