

5.4 Доступ к элементам матрицы

Доступ к элементам матриц осуществляется при помощи двух индексов - номеров строки и столбца, заключенных в круглые скобки, например команда $B(2,3)$ выдаст элемент второй строки и третьего столбца матрицы B [2].

Для выделения из матрицы столбца или строки следует в качестве одного из индексов использовать номер столбца или строки матрицы, а другой индекс заменить двоеточием.

Например, на рисунке 5.6 представлена запись строки матрицы A в вектор z . Также можно осуществлять выделение блоков матриц при помощи двоеточия. Например, на рисунке 5.7 из матрицы P выделен блок отмеченный желтым цветом. Если необходимо посмотреть переменные рабочей среды MATLAB, в командной строке необходимо набрать команду «whos» (рисунок 5.8).

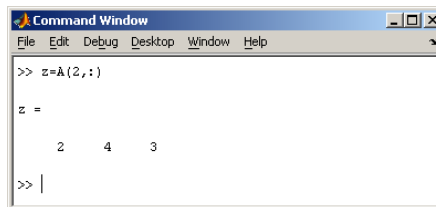


Рисунок 5.6. Запись строки матрицы A в вектор z

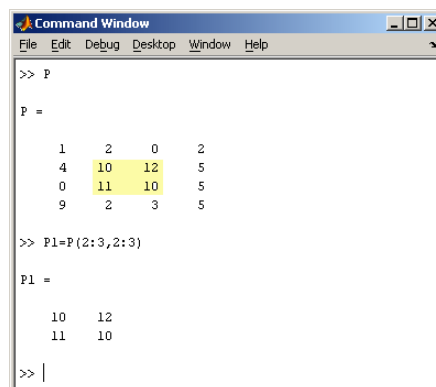


Рисунок 5.7. Выделение блоков матриц через двоеточие

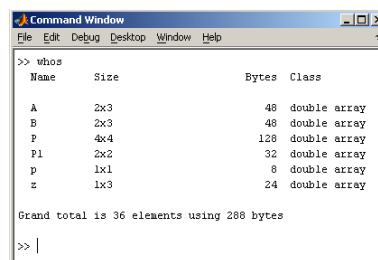


Рисунок 5.8. Применение командной строки «whos» для матричного исчисления

Видно, что в рабочей среде содержатся один скаляр (p), четыре матрицы (A , B , P , $P1$) и вектор-строка (z).

5.5 Основные матричные операции

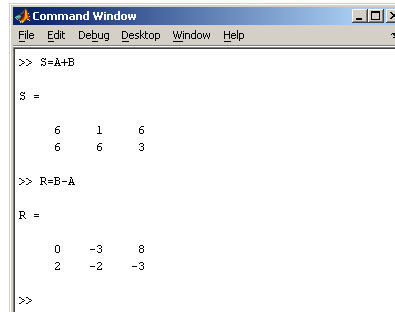
При использовании матричных операций следует помнить, что для сложения или вычитания матрицы должны быть одного размера, а при перемножении число столбцов первой матрицы обязано равняться числу строк второй матрицы [8]. Сложение и вычитание матриц (рисунок 5.9), так же как чисел и векторов, осуществляется при помощи знаков плюс и минус. Умножение элементов матрицы в исходном тексте программы обозначается знаком звездочка «*». На рисунке 5.10 введена матрица размером 3×2 .

Умножение матрицы на число тоже осуществляется при помощи звездочки, причем умножать на число можно как справа, так и слева. Возведение квадратной матрицы в целую степень,

производится с использованием оператора «^» (рисунок 5.11). Проверьте полученный результат, умножив матрицу P саму на себя [15].

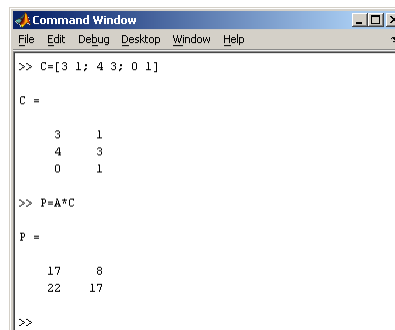
Заполнение прямоугольной матрицы нулями производится встроенной функцией «zeros» (рисунок 5.12). Единичная матрица создается при помощи функции «eye» (рисунок 5.13). Матрица, состоящая из единиц, образуется в результате вызова функции «ones» (рисунок 5.14).

MATLAB предоставляет возможность заполнения матриц случайными числами.



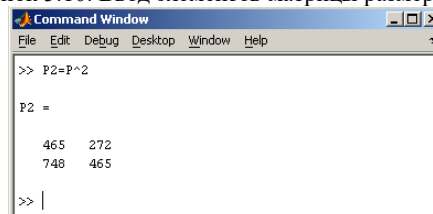
```
Command Window
File Edit Debug Desktop Window Help
>> S=A+B
S =
     6     1     6
     6     6     3
>> R=B-A
R =
     0    -3     8
     2    -2    -3
>>
```

Рисунок 5.9. Процедура сложения и вычитания матриц в MATLAB



```
Command Window
File Edit Debug Desktop Window Help
>> C=[3 1; 4 3; 0 1]
C =
     3     1
     4     3
     0     1
>> P=A*C
P =
    17     8
    22    17
>>
```

Рисунок 5.10. Ввод элементов матрицы размером 3x2

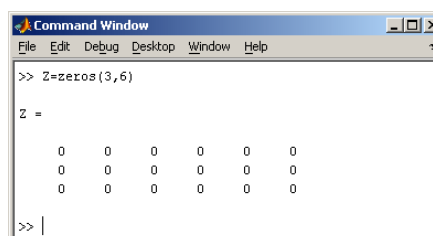


```
Command Window
File Edit Debug Desktop Window Help
>> P2=P^2
P2 =
    465    272
    748    465
>> |
```

Рисунок 5.11. Процедура возведения квадратичной матрицы в целую степень

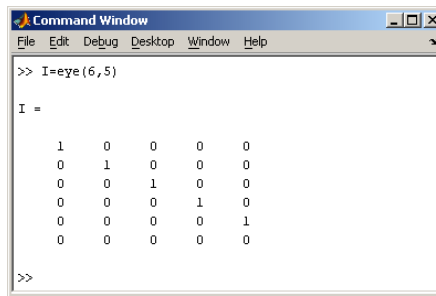
5.6 Создание матриц специального вида

Результатом функции «rand» является матрица чисел, равномерно распределенных между нулем и единицей, а функции «randn» - матрица чисел, распределенных по нормальному закону с нулевым средним и единичной дисперсией. Функция «diag» формирует диагональную матрицу из вектора, располагая элементы по диагонали [15].



```
Command Window
File Edit Debug Desktop Window Help
>> Z=zeros(3,6)
Z =
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
>> |
```

Рисунок 5.12. Заполнение прямоугольной матрицы нулями через встроенную функцию «zeros»



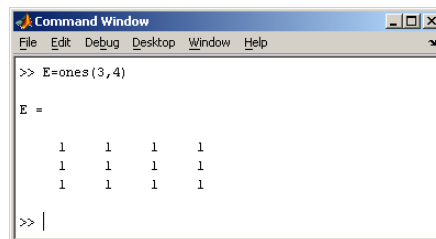
```
>> I=eye(6,5)

I =

     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
     0     0     0     0     0

>>
```

Рисунок 5.13. Создание единичной матрицы функцией «eye»



```
>> E=ones(3,4)

E =

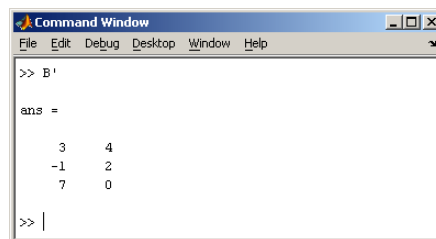
     1     1     1     1
     1     1     1     1
     1     1     1     1

>> |
```

Рисунок 5.14. Создание единичной матрицы функцией «ones»

5.7 Матричные вычисления

MATLAB содержит множество различных функций для работы с матрицами [8]. Так, например, транспонирование матрицы производится при помощи знака апострофа «'».



```
>> B'

ans =

     3     4
    -1     2
     7     0

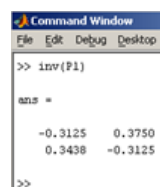
>> |
```

Рисунок 5.15. Программа транспонирования матрицы знаком апострофа «'»

Также следует отметить, что при транспонировании массива (операция «.') строки просто заменяются столбцами:

```
A.'
ans =
     1     4     1
     2     5     4
     3     6     8.
```

При транспонировании матрицы (операция «'» без точки) результатом является транспонированная матрица, для комплексных чисел выполняется операция комплексного сопряжения. Нахождение обратной матрицы проводится с помощью функции «inv» для квадратных матриц (рисунок 5.16). Псевдообратную матрицу можно найти с помощью функции «pinv». Более подробно про обработку матричных данных можно узнать, если вывести список всех встроенных функций обработки данных командой «help datafun», а затем посмотреть информацию о нужной функции, например «help max».



```
>> inv(P1)

ans =

    -0.3125    0.3750
     0.3438   -0.3125

>>
```

Рисунок 5.16. Вычисление обратной квадратной матрицы функцией «inv»

5.8 Разные операции над матрицами

Над матрицами можно производить разнообразные операции [2]. Например, с помощью команды `diag` возможно формировать или извлекать диагонали матрицы. Функция `X=diag(v)` формирует квадратную матрицу `X` с вектором `v` на главной диагонали.

Функция `X=diag(v,k)` формирует квадратную матрицу `X` порядка `length(v)+abs(k)` с вектором `v` на `ke`-той диагонали. Функция `v=diag(X,k)` извлекает из матрицы `X` диагональ с номером `k`; при `k>0` - это номер `k`-й верхней диагонали, при `k<0` - это номер `k`-й нижней диагонали. Функция `B=reshape(A,m,n)` возвращает матрицу размером `[m x n]`, сформированную из элементов матрицы `A` путем их последовательной выборки по столбцам.

5.9 Функции для создания стандартных матриц

Для создания матриц используется специальные символы [12]. Функции для создания стандартных матриц приведены в таблице 5.3.

Таблица 5.3. Функция для создания стандартных матриц

Функция	Результат и примеры вызовов
<code>zeros</code>	Нулевая матрица <code>F=zeros(4,5)</code> <code>F=zeros(3)</code> <code>F=zeros([3 4])</code>
<code>eye</code>	Единичная прямоугольная матрица (единицы расположены на главной диагонали) <code>I=eye(5,8)</code> <code>I=eye(5)</code> <code>I=eye([5 8])</code>
<code>ones</code>	Матрица, целиком состоящая из единиц <code>E=ones(3,5)</code> <code>E=ones(6)</code> <code>E=ones([2 5])</code>
<code>rand</code>	Матрица, элементы которой – случайные числа, равномерно распределенные на интервале <code>(0,1)</code> <code>R=rand(5,7)</code> <code>R=rand(6)</code> <code>R=rand([3 5])</code>
<code>randn</code>	Матрица, элементы которой – случайные числа, распределенные по нормальному закону с нулевым средним и дисперсией, равной единице <code>N=randn(5,3)</code> <code>N=randn(9)</code> <code>N=randn([2 4])</code>
<code>diag</code>	1) диагональная матрица, элементы которой задаются во входном аргументе – векторе <code>D=diag(v)</code> 2) диагональная матрица со смещенной на <code>k</code> позиций диагональю (положительные <code>k</code> – смещение вверх, отрицательные – вниз), результатом является квадратная матрица размера <code>length(v)+abs(k)</code> <code>D=diag(v,k)</code> 3) выделение главной диагонали из матрицы в вектор <code>d=diag(A)</code> 4) выделение <code>k</code> -й диагонали из матрицы в вектор <code>d=diag(A,k)</code>

Например, для ввода матрицы `A(3x3)` можно задать следующую команду:

```
>> A=input('Введите матрицу 3*3').
```

Также приведем пример ввода матрицы `3*3`:

```
[1 2 3;4 5 6;7 8 9] <Enter>
```

```
A =
```

```
1 2 3
4 5 6
7 8 9.
```

Удобной возможностью системы программирования MATLAB является конструирование матрицы из матриц меньших размеров.

Рассмотрим пример конструирования большой матрицы из нескольких матриц меньших размеров. Пусть заданы 4 матрицы разных размерностей:

$$M1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad M2 = \begin{bmatrix} -2 & -3 & -5 \\ -1 & -5 & -6 \end{bmatrix},$$

$$M3 = \begin{bmatrix} 9 & 8 \\ -7 & -5 \\ 1 & 2 \end{bmatrix}, \quad M4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Требуется составить из этих 4 матриц 1 одну блочную матрицу `M` вида:

$$M = \begin{bmatrix} M1 & M2 \\ M3 & M4 \end{bmatrix}.$$

Можно считать, что соответственно, матрицей получения в рабочей среде использовать оператор: `>> M=[M1 M2; M3 M4]`.

Для создания векторов матриц также можно использовать специальные функции.

5.10 Сервисные функции

Перечислим сервисные функции MATLAB для операции над матрицами [15]:

- `who` – выводит имена всех определенных в рабочей среде переменных:

```
>> a=rand(3);
```

```
>> who
```

Your variables are:

a;

- whos – выводит информацию о переменных рабочей среды:

```
>> whos
```

```
Name Size Bytes Class
```

```
a 3x3 72 double array
```

```
Grand total is 9 elements using 72 bytes;
```

- size – возвращает размер массива:

```
>> v=[1 2 3];
```

```
>> s=size(v)
```

```
s =
```

```
3
```

ИЛИ

[M,N] = size(s) M – число

```
>> s=ones(2,3)
```

```
s =
```

```
1 1 1
```

```
1 1 1
```

```
>> [M,N]= size(s)
```

```
M =
```

```
2
```

```
N =
```

```
3.
```

- max и min – вычисляет вектор-строку, содержащую максимальные и минимальные элементы

в каждом столбце матрицы:

```
>> v=rand(3)
```

```
v =
```

```
0.1934 0.5417 0.3784
```

```
0.6822 0.1509 0.8600
```

```
0.3028 0.6979 0.8537
```

```
ma =
```

```
0.6822 0.6979 0.8600.
```

Для того чтобы узнать не только значения максимальных или минимальных элементов, но и их номера в столбцах или строках, используют следующий вызов:

```
[Y,I] = max(X,[],DIM)
```

```
[Y,I] = min(X,[],DIM).
```

Здесь Y – значения максимальных элементов; I – номера элементов; DIM – номера размерности: 1 – по столбцам; 2 – по строкам.

Обычно (по умолчанию) вычисляется выражение вида: ([Y,I] = min(X,[],DIM).