РАБОТА С МАТРИЦАМИ
И
РЕШЕНИЕ СЛАУ
В ПАКЕТЕ
MATLAB

Вектор-строка и вектор-столбец

Все в Matlab – матрицы! Индексация начинается с 1 !!!

Matlab	Математика
a=[3; -1; 7] – разделение элементов вектора-столбца идет через «;»	$a = \begin{pmatrix} 3 \\ -1 \\ 7 \end{pmatrix}$
b=[3 -1 7] – разделение элементов вектора-столбца идет через «пробел»	$b = \begin{pmatrix} 3 & -1 & 7 \end{pmatrix}$

Действия с векторами

Оператор/ функция	Действие
dot(a,b)	Скалярное произведение
cross(a,b)	Векторное произведение
length(a)	Длина вектора *a* (для столбца с предыдущего слайда =3)

Матрица A

$$\begin{pmatrix} 1 & 0 & \boxed{0} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Элемент,
A (1,3)

Строка,
A(3,:)

Столбец, A(:,1)

Задание матрицы

Ф	A = [12 62 18; 93 -8 15; 12 -15 81]
КС	A = 12 62 18 93 -8 15 12 -15 81

Задание по строкам, строки разделяются символом «;». Символ «;» отделяет два соседних столбца

Задание матрицы специальными функциями

Функция	Результат
zeros(3,2)	Нулевая матрица размерности 3x2 (3 строки, 2 столбца)
ones(2,4)	Матрица из «1» размерности 2x4
A = [12 62 93]; B = diag(A) C=diag(A,k)	B- квадратная матрицы у которой на диагонали стоят элементы вектора A, вне диагонали – нули; у C диагональ смещена на k (вверх, если k>0, вниз, если k<0)
eye(m,n), eye(n)	Единичная матрица $m \times n$. Если матрица квадратная, то допустимо указывать только одно число, n.

Задание специальных матриц

Функция	Результат
pascal(n)	Симметричная версия матрицы Паскаля размерностью nxn. Для n=3: $\begin{matrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{matrix}$
randi(v,m,n)	Матрица из случайных чисел из диапазона v, размерности mxn.
magic(n)	«Магическая» матрица nxn (суммы строк и столбцов одинаковы). Для n=3: $\begin{matrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{matrix}$

Задание матрицы через цикл

Сформировать матрицу *A* размерности 3x3 по правилу:

$$A_{i,j} = \begin{cases} i + j, & i \neq j \\ 1, & i = j \end{cases}$$

Задание матрицы через цикл

Ф	N=3 % matrix size
	A=ones(N,N) % all elem.=1
	for i=1:N %rows
	  for j=1:N %columns
	     A(i,j)=i+j ????? i=j
		N=3; % matrix size
		for i=1:N %rows
		  for j=1:N %columns
		     A(i,j)=i+j
		for i=1:N % diag
		       A(i,i)=1

Условный оператор

if (условие истинно)
  делать
else
  делать (выполняется в случае, если условие не истинно)
end

Возможно «краткая (укороченная)» версия условного оператора:
if (условие истинно)
  делать
end

Составные условия

<table>
<tr><th>Символ</th><th>Операция</th></tr>
<tr><td>==</td><td>равно</td></tr>
<tr><td>~=</td><td>Не равно</td></tr>
<tr><td>>, <</td><td>Больше, меньше</td></tr>
<tr><td>>=,<=</td><td>Больше или равно, меньше или равно</td></tr>
<tr><td>~</td><td>не</td></tr>
<tr><td>||</td><td>или</td></tr>
<tr><td>&&</td><td>и</td></tr>
</table>

Условный оператор -2

if (условие1 истинно)
   делать (выполняется в случае, если условие1 истинно)
elseif (условие2 истинно)
   делать (выполняется в случае, если условие2 истинно)
else
 делать (выполняется в случае, если ни одно из условий не истинно)
end

Задание матрицы через цикл (универсальная функция)

```
function A=creat_matrix(N)
for i=1:N %rows
  for j=1:N %columns
    if (i==j)
        A(i,i)=1;
    else
        A(i,j)=i+j;
  end%if
end%for j
end%for i
 A;
end
```

Трассировка функции, или как это работает (N=3)

i	j	if (i==j)	A(i,j)
1	1	true	1
1	2	false	3
1	3	false	4
2	1	false	3
2	2	true	1
2	3	false	5
3	1	false	4
3	2	false	5
3	3	true	1

Вызов функции задания матрицы и результат работы

```
>> clear all
>> A=creat_matrix(5)

A =

     1     3     4     5     6
     3     1     5     6     7
     4     5     1     7     8
     5     6     7     1     9
     6     7     8     9     1

>>
```

Очищает все имеющиеся переменные

Действия с матрицами

Оператор/ функция	Действие
+, -	Сложение, вычитание. Количество элементов должно быть одинаково
*	Умножение по правилам матричной алгебры.
.* , ./	Поэлементное умножение, поэлементное деление
‘ (апостроф)	Транспонирование
\	Матричное деление (алгоритм выбирает Matlab)
inv(A)	Вычисление обратной матрицы

Функции для работы с матрицами

Функция	Действие
[d, ind] = sort(b)	Сортировка вектора b. Результат записывается в вектор d, в векторе ind сохраняются индексы исходных позиций элементов. Может работать с матрицами (см. Help)
[d, ind] = max(b)	возвращает максимум вектора b; если b - матрица, то max (b) - строка, содержащая максимальное значение каждого столбца. ind – индекс максимального элемента
[d, ind] = min (b)	возвращает минимум вектора b

Функции для работы с матрицами

Функция	Действие
[S, L] = bound(A)	Возвращает мин. (S) и макс. (L) элементы вектора A; если A –матрица, то возвращает строку из мин. и макс. элементов
[S, L] = bound(A, ‘all’)	Возвращает мин. (S) и макс. (L) элементы матрицы A
d = mean(b)	возвращает среднее значение вектора b; если b - матрица, то возвращается строка, содержащая среднее значение каждого столбца.

Слияние матриц

Функция	Результат
A = ones(1,4); B = zeros(1,4); C = [A B]	Слияние по строкам >>C = *1×8* B A 1 1 1 1 0 0 0 0
D = [A;B]	Слияние по столбцам >>D = *2×4* A 1 1 1 1 0 0 0 0 B
A = ones(2,3) B = zeros(2,2) D = horzcat(A,B)	Слияние двух матриц в одну (число строк должно быть одинаково) >>D = *2×5* 1 1 1 0 0 B A 1 1 1 0 0

Расширение матриц (поэлементное)

Функция	Результат
Ф	A = [10 20 30; 60 70 80] A(3,4) = 1
КС	A = 2×3 10 20 30 60 70 80 A = 3×4 10 20 30 0 60 70 80 0 0 0 0 1

Расширение матриц (матрицами)

Функция	Результат
Ф	A(4:5,5:6) = [2 3; 4 5]
КС	A = *5×6* 10 20 30 0 0 0 60 70 80 0 0 0 0 0 0 1 0 0 0 0 0 0 2 3 0 0 0 0 4 5

A(с какой строки : по какую, с какого столбца : по какой)

Выделение подматрицы

```
A =

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

Функция	Результат

B=A(i,:)	Выделить из матрицы A i-тую строку
	`>> A(1,:)`
	`ans =`
	`1     3     4     5     6`

B=A(:,j)	Выделить из матрицы A j-тый столбец
	`>> A(:,3)`
	`ans =`
	`4`
	`5`
	`1`
	`7`
	`8`

Основные характеристики матрицы

Функция	Результат
det(A)	Определитель
rank(A)	Ранг
trace(A)	След

Нормы матрицы

Для вычисления различных вариантов нормы используется функция вида:

n=norm(M,k)

M – матрица/вектор, для которого необходимо вычислить норму

k – параметр, определяющий какую норму надо вычислить. Наиболее распространенные значения параметра k:

1 – L1 норма (максимальная столбцовая норма)

2 – евклидова норма (по умолч.)

inf – неопределенная норма (максимальная строчная норма)

Нормы вектора

Для вычисления различных вариантов нормы используется функция вида:

n=vecnorm(M, k)

M – матрица/вектор, для которого необходимо вычислить норму. Результат зависит от вида M:
M – вектор, n – число, евклидова норма
M – матрица, n – строка, содержащая нормы для каждого столбца

Значения параметра k такие же, как в функции norm

Числа обусловленности матрицы

Для вычисления различных вариантов числа обусловленности используется функция вида:

cond(M,k)

M – матрица/вектор, для которого необходимо вычислить число обусловленности

k – параметр, определяющий какое число обусловленности надо вычислить. Значения параметра k аналогичны параметрам в функции norm.

Пример вычисления характеристик матрицы

Ф	КС
D=det(A) R=rank(A) T=trace(A) N1=norm(A,1) N2=norm(A,2) NInf=norm(A,Inf) C1=cond(A,1) C2=cond(A,2) CInf=cond(A,Inf)	D = 1.0119e+04 R = 5 T = 5 N1 = 31 N2 = 25.5919 NInf = 31 C1 = 24.3644 C2 = 13.8764 CInf = 24.3644

Собственные числа и собственные вектора

Функция	Результат
lambda = eig(A)	Возвращает собственные числа матрицы A
[V, D] = eig(A)	В матрице V расположены собственные вектора матрицы A, в матрице D на диагонали расположены собственные числа матрицы A, соответствующие собственным векторам, приведенным в V.

Собственные числа и собственные вектора (пример)

<table>
<tr><td>Ф</td><td>lambda = eig(A)
[V, D] = eig(A)</td></tr>
<tr><td>КС</td><td><pre>
V =

   -0.1465   0.1977   -0.2954   0.8544   -0.3495
   -0.1725   0.2696   -0.6982   -0.4980   -0.4025
   -0.2286   0.5719   0.6331   -0.1362   -0.4486
   -0.4421   -0.7355   0.1464   -0.0551   -0.4890
    0.8373   -0.1420   0.0547   -0.0193   -0.5248

D =

   -8.4280   0   0   0   0
         0   -6.2488   0   0   0
         0   0   -4.0708   0   0
         0   0   0   -1.8443   0
         0   0   0   0   25.5919
</pre></td></tr>
</table>

Разложения матриц

Функция	Результат
[L,U] = lu(A)	LU-разложение матрицы, L – нижняя треугольная матрицы с 1 на диагонали; U — верхняя треугольная матрица
[Q,R] = qr(A)	QR-разложение матрицы: QQ^T=E(единичная матрица) R – верхняя треугольная матрица

LU разложение матриц (пример)

Ф	`[L,U] = lu(A)`
КС	L = $\begin{matrix} 0.1667 & -0.7333 & -0.8095 & -0.8864 & 1.0000 \\ 0.5000 & 1.0000 & 0 & 0 & 0 \\ 0.6667 & -0.1333 & 1.0000 & 0 & 0 \\ 0.8333 & -0.0667 & -0.0952 & 1.0000 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 \end{matrix}$ U = $\begin{matrix} 6.0000 & 7.0000 & 8.0000 & 9.0000 & 1.0000 \\ 0 & -2.5000 & 1.0000 & 1.5000 & 6.5000 \\ 0 & 0 & -4.2000 & 1.2000 & 8.2000 \\ 0 & 0 & 0 & -6.2857 & 9.3810 \\ 0 & 0 & 0 & 0 & 25.5530 \end{matrix}$

QR разложение матриц (пример)

Ф	`[Q,R] = qr(A)`
КС	see output below

```
Q =

   -0.1072    0.6044    0.5245    0.4583   -0.3715
   -0.3216   -0.7676    0.2938    0.3262   -0.3387
   -0.4288    0.1594   -0.7685    0.2998   -0.3321
   -0.5361    0.1187    0.1742   -0.7482   -0.3293
   -0.6433    0.0779    0.1328    0.1842    0.7270


R =

   -9.3274  -10.5067  -11.3644  -11.7932  -11.7932
         0    3.0999    0.1928    0.3523    0.6749
         0         0    5.0807    0.3758    0.7572
         0         0         0    7.2563    0.8816
         0         0         0         0   -9.4928
```

Решение СЛАУ вида Ax=b

Варианты	Действия
x = inv(A)*b	Вычисление обратной матрицы
x = A\b	Левое матричное деление
R = rref(A1)	Метод Гаусса, A1 – расширенная матрица системы: A1 =[A b], решение – последний столбец матрицы ответа (R=[I x] или x = R(:,end))
x=linsolve(A,b,opt)	Методы линейной алгебры, выбор алгоритма задается параметром <u>opt</u> (<u>не обязателен</u>)

Вычисление времени выполнения операций

Время выполнения операция вычисляют с помощью пары операторов:

tic
исследуемые действия
t=toc

В переменой t будет находится время выполнения операций, размещенных между tic-toc

Функция для решения СЛАУ

```
function res=my_lsolve(A)
 for i=1:length(A(:,1)) %rows
   b(i)=1;
 end% for
 b=b';%transp.
tic
 x1 = inv(A)*b;
t1=toc;
tic
 x2 = A\b;
t2=toc;
tic
R = rref([A b]);
x3 = R(:,end);
t3=toc;
tic
 x4=linsolve(A,b);
t4=toc;
T=[t1 t2 t3 t4];
X=[x1 x2 x3 x4];
res=[T;X];
end
```

Функция для решения СЛАУ (результат работы)

```
>> my_lsolve(A)

ans =

    0.0001    0.0001    0.0038    0.0001
   -0.0566   -0.0566   -0.0566   -0.0566
    0.0368    0.0368    0.0368    0.0368
    0.0554    0.0554    0.0554    0.0554
    0.0634    0.0634    0.0634    0.0634
    0.0679    0.0679    0.0679    0.0679
```

Вызов функции

Время работы каждого способа

Столбцы ответов для каждого способа