

Операции с матрицами и свойства матриц

Пример 1. Определение размера матрицы

```
clear
ndims(eye(2,3))
ndims(rand(2,3,4))
[m,n]=size(ones(3,4))
r=size(ones(3,4,5))
maxdim=length(zeros(7,5,3)) % определяется длина большей
размерности
numel(ones(7,5,3)) % общее количество элементов
Объясните назначение конструкторов: eye, ones, zeros, rand
```

Пример 2. Контроль типов

```
clear
a=rand(2,3)
b=isnumeric(a) % b=1, но
c=islogical(b) % b -логическая единица
d=isreal(c) % логическая единица не имеет мнимой части
isinf(1/0), isnan(0/0) % объясните нюансы (/ - slash)
```

Пример 3. Редактирование строк и блоков матриц

```
clc
clear
A=ones(5) % матрица 5 порядка из единиц
A(3:end, 3:end)=5 % элементы блока третьего порядка равны пяти
A(1:2, 1:2)=[1,2;3,4] % блок второго порядка в матрице A заменен на
матрицу правой части
C=blkdiag(A, eye(3), rand(7)) % построение диагональной матрицы
spy(C) % графическая визуализация ненулевых элементов
C(1:2, :)=[], spy(C) % редуцированы первые две строки
C(:, 2:2:end)=0, spy(C) % обнулены четные столбцы
```

Пример 4. Функции min, max, sum, prod

```
clear
A=diag([0 1 2])+ones(3)
B=prod(A) % определите тип результата
prod(prod(A)) % определите тип результата
sum(sum(A))
min(min(A))
max(max(A))
B=[1 -2; 3+i 4]
abs(B), abs(B(2,1))
sum(B) % по результатам команд
prod(B) % объясните,
prod(prod(B)), %как работают функции min, max, sum, prod, abs,
сравните случаи, когда аргумент является вектором и матрицей
```

Пример 5. Сортировка - sort

```
clear
clc
A =[3 2 1; 7 8 9; 5 4 6]
B = sort(A) % сортирует по столбцам по возрастанию
```

```

C = sort(A, 'descend') % по убыванию
D = sort(A,2) % вдоль строк (по столбцам) по возрастанию
E = sort(A,2, 'descend') % вдоль строк (по столбцам) по убыванию

```

Пример 6. find поиск элементов ненулевых или соответствующих заданному условию

```

clear
clc
A=[3 0 0; 0 7 0; 0 0 256]
[i, j, x] = find(A) % определяются индексы [i,j] и ненулевые
%значения A(i,j), записанные в вектор x
[i, j, x] = find(A>7); % определяются векторы индексов i и j для
значений A(i,j) >7, вектор x состоит из логических единиц

```

Элементы линейной алгебры

Пример 7. Обращение матриц

Если матрица A является квадратной и невырожденной, уравнения $AX = E$ и $XA = E$ (E – единичная матрица) имеют одинаковое решение X . Это решение называется матрицей обратной к A , обозначается через A^{-1} и вычисляется с помощью функции `inv(A)` или `A^(-1)`.

```

clear
A= pascal (3)
d = det (A) % вычисляется определитель
X = inv (A) % вычисляется обратная матрица
A*X == eye(size(A)) % объясните результат (eye – конструктор
единичной матрицы)
B=A*X - eye(size(A)) % постройте, используя подходящий конструктор,
матрицу B

```

2

Пример 8. Собственные значения и собственные векторы матрицы – eig

Собственным значением и собственным вектором квадратной матрицы A называются скаляр λ и вектор v , удовлетворяющие условию $Av = \lambda v$

```

clear
A = [4 1 2;
3 7 1;
2 2 8];
lambda = eig(A) % здесь собственные значения различны
% и выходной параметр – вектор
[V,D] = eig(A) % собственные значения на главной диагонали
диагональной матрицы D, и все собственные векторы (с.в.) в
матрице записаны по столбцам, с. в. независимы
V\A*V % равносильно решению D=A*V(inv(V)) матричного уравнения
AV=DV

```

Пример 9. Норма и число обусловленности матрицы

```

clear
A= rand(1,4)
norm(A) % норма вектора, без второго уточняющего (тип нормы)
аргумента совпадает с Евклидовой нормой
B=A+A'
cond(B) % norm(B)*norm(inv(B)) число обусловленности – характеризует
устойчивость решения системы

```

rank(B) % ранг матрицы, важно при оценке вырожденности системы

Факторизация матриц и решение СЛАУ

Систему называют определенной, если она имеет единственное решение.

$X = A \setminus B$ (\ - backslash) - является решением матричного уравнения $AX = B$ (\ - backslash)

$X = B/A$ - является решением матричного уравнения $XA = B$.

Пример 10. Решение СЛАУ методом Гаусса

```
clear
A = [4 1 2;
3 7 1;
2 2 8];
b = [7; 11; 12];
x = A\b % в ML для вычисления x реализован алгоритм Гаусса
% функция mldivide равносильна операции деления (\)
x = mldivide(A, b)
% Проверка: вычисление невязки или нормы вектора невязки
b - A*x, norm(b - A*x)
```

Пример 11. Факторизация матриц

```
clear
A = [4 1 2;
3 7 1;
2 2 8];
b = [7; 11; 12];
[l,u]=lu(A) % l - нижняя треугольная, u - верхняя треугольная
матрицы такие, что l*u=A
[Q,R]=qr(A) % R - верхняя треугольная, Q - унитарная (Q*Q'=E)
матрицы такие, что Q*R=A
V=A*A' % V - симметричная матрица с положительными элементами
all(all(V==V')) % объясните, что же здесь проверили (может быть, для
понимания поможет реструктурирование команды)
R=chol(B) % если B - положительно определенная матрица, то R'*R = B,
где R - верхняя треугольная матрица с положительными элементами на
диагонали
```

3

Пример 12. Решение СЛАУ с помощью linsolve

```
clc
clear
eig([4 1 2;
3 7 1;
2 2 8])
b = [7; 11; 12];
x=linsolve(A,b) % изучите справку linsolve - как управлять выбором
решателей в зависимости от свойств матрицы системы
```

Задания для самостоятельного (творческого) решения.

- 1) Придумайте пример системы линейных алгебраических уравнений в матричной форме $Ax=b$
 - a) выберите высокий порядок ($n > 1000$) матрицы A и используйте для её построения - функцию rand

- b) факторизуйте матрицу A с помощью процедуры $[L, U] = \text{lu}(A)$ и на основе факторизованной системы $L^*U^*x=b$ получите решение
 - c) факторизуйте матрицу A с помощью процедуры $[Q, R] = \text{qr}(A)$ и на основе факторизованной системы $Q^*R^*x=b$ получите решение
 - d) сравните время поиска решений, а также их точность (по норме невязки); используйте функции `clock` и `etime`, см. `help`).
- 2) Придумайте пример системы $Ax=b$, имеющей положительно определенную матрицу порядка, большего тысячи. Решите систему с помощью
- a) `linsolve(A, b)`
(идеально использовать `opts.POSDEF=true`, `linsolve(A, b, opts)` с опцией положительной определённости, но в некоторых версиях эта опция уже не поддерживается)
 - b) метода Холецкого (используйте факторизацию `R=chol(A)`)
 - c) метода Гаусса

сравните точность и время поиска решений .

Замечание. На всех этапах построения решений убедитесь, что все решения получены верно.