

Имена переменных, функций и т. п. чувствительны к регистру. Разделителями команд являются: ENTER, «,» или «;». Результат команды, после которой идет ENTER или «,», печатается. Для продолжение команды на следующей строке используется «...».

format short переход в режим с «коротким» представлением чисел с плавающей запятой
format long «длинный» формат при выводе чисел с плавающей запятой
format long «рациональный» формат при выводе чисел с плавающей запятой
help имя_функции справка в командном окне
doc имя_функции справка в справочном навигаторе

1. Числа

$1 + 2*(3 - 4)/5$ вычисление выражения
 $a = 1 + 2*(3 - 4)/5$ вычисление выражения и присваивание результата переменной a
ans результат последнего не присвоенного выражения
 x^y x^y
pi число π
Inf бесконечность ∞
NaN «не-число»
eps машинная точность $2^{-52} = 2.2204 \times 10^{-16}$
realmax максимальное число $(2 - 2^{-52}) \times 2^{1023} = 1.7977 \times 10^{308}$
realmin минимальное положительное нормализованное число $2^{-1022} = 2.2251 \times 10^{-308}$
abs(x) модуль $|x|$
sign(x) знак $\text{sign } x$
sqrt(x) корень квадратный \sqrt{x}
exp(x) e^x
log(x) $\ln x$
log2(x) $\log_2 x$
log10(x) $\log_{10} x$
sin(x) $\sin x$
cos(x) $\cos x$
tan(x) $\text{tg } x$
asin(x) $\arcsin x$
acos(x) $\arccos x$
atan(x) $\text{arctg } x$
floor(x) «пол» $\lfloor x \rfloor$
ceil(x) «потолок» $\lceil x \rceil$
round(x) ближайшее целое $\lfloor x \rfloor$
fix(x) число с отброшенной дробной частью
gcd(m, n) НОД(m, n)
lcm(m, n) НОК(m, n)
rem(m, n) $m - \text{fix}(m/n)n$
mod(m, n) $m - \lfloor m/n \rfloor n$
primes(n) список простых чисел $\leq n$
isprime(n) проверка числа на простоту
factor(n) разложение на простые множители числа n
factorial(n) $n!$
i, j, 1i, 1j мнимая единица
1+1i, 1-2i, 3i комплексные числа
complex(a, b) комплексное число $a+bi$
real(z) действительная часть комплексного числа z
imag(z) мнимая часть комплексного числа z
abs(z) модуль комплексного числа z
angle(z) угол (аргумент) комплексного числа z

z', conj(z) сопряженное число

2. Векторы и матрицы

[1, 2, 3] или [1 2 3] вектор-строка [1, 2, 3]

[1; 2; 3] вектор-столбец $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

[1, 2; 3, 4] матрица $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

(Вместо ; можно нажимать ENTER)

a(1) 1-й элемент вектора (строки или столбца) a

a(end) последний элемент вектора a

A(2, 4) элемент 2-й строки 4-го столбца матрицы A

A(2, end) элемент 2-й строки последнего столбца

A(2, :) 2-я строка матрицы A

A(:, 2) 2-й столбец матрицы A

A(2, :) = [] удаление 2-й строки матрицы A

A(:, 2) = [] удаление 2-го столбца матрицы A

A(2, end) элемент 2-й строки последнего столбца

A([1, 2], [2, 5]) матрица из элементов, стоящих на пересечении 1-й и 2-й строк и 2-го и 5 столбцов

a:b строка, составленная из чисел от a до b с шагом 1

a:h:b строка из чисел от a до b с шагом h

1:10 строка [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

1:2:10 строка [1, 3, 5, 7, 9]

size(A) размеры матрицы A (вектор из двух компонент)

size(A, 1) число строк матрицы A

size(A, 2) число столбцов матрицы A

length(a) длина вектора a

[a, b] конкатенация векторов-строк

[a; b] конкатенация векторов-столбцов

[A, B; C, D] матрица, составленная из блоков

zeros(n) нулевая квадратная матрица порядка n

zeros(m, n) нулевая прямоугольная матрица размера $m \times n$

ones(n), ones(m, n) матрица, заполненная единицами

eye(n), eye(m, n) единичная матрица

rand случайное число на отрезке [0, 1]

randn случайное число, распределенное по

нормальному закону с мат. ожиданием 0 и

среднеквадратическим отклонением 1

rand(n), randn(n) случайные квадратные матрицы порядка n

rand(m, n), randn(m, n) случайные прямоугольные матрицы размера $m \times n$

A + B сумма матриц $A + B$

A - B разность матриц $A - B$

a*A произведение aA числа a на матрицу A

A*B матричное произведение AB

A^n матричная степень A^n

A.*B покомпонентное произведение

A./B покомпонентное деление

A.^B покомпонентная степень

A' сопряженная матрица A^T

A.' транспонированная матрица A^T

A\b решение с.л.у. $Ax = b$

A\B решение матричного уравнения $AX = B$

A/B решение матричного уравнения $YA = B$

inv(A), A^(-1) обратная матрица A^{-1}

det(A) определитель матрицы
rref(A) приведенный ступенчатый вид матрицы
linspace(a, b) вектор из 100 равномерно отстоящих узлов от a до b
linspace(a, b, n) вектор из n равномерно отстоящих узлов от a до b
tril(A) нижнетреугольная часть матрицы
tril(A, k) элементы ниже k -й диагонали ($k = 0$ — главная диагональ, $k > 0$ — выше главной диагонали, $k < 0$ — ниже главной диагонали)
triu(A) верхнетреугольная часть матрицы
triu(A, k) элементы выше k -й диагонали
diag(A) главная диагональ матрицы
diag(A, k) k -я диагональ матрицы
diag(d) формирует диагональную матрицу с элементами из вектора d на диагонали
diag(d, k) формирует матрицу с элементами из d на k -й диагонали
blkdiag(D1, D2, D3, ...) блочно-диагональная матрица
[X, Y] = meshgrid(x, y) генерирование решеток:
[X, Y] = meshgrid([1,2], [11,22,33]) возвращает

$$X = \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 2 \end{bmatrix}, Y = \begin{bmatrix} 11 & 11 \\ 22 & 22 \\ 33 & 33 \end{bmatrix}$$
 Математические функции (**abs**, **sqrt**, **exp**, **log**, **sin** и т. д.), примененные к матрицам, действуют покомпонентно.
 См. также раздел «линейная алгебра»

3. Сохранение и загрузка данных

save имя_файла сохранение значений всех переменных рабочего пространства в **mat**-файле
save имя_файла a1 a2 ... an сохранение значений указанных переменных
load имя_файла загрузка данных из файла
load имя_файла a1 a2 ... an загрузка указанных переменных
save a -ascii сохранение значения переменной в одноименном текстовом файле
load a -ascii загрузка переменной из текстового файла
load a -ascii загрузка переменной из текстового файла
[a1, a2, a3, ...] = textread(имя_файла, формат)
 чтение из текстового файла в переменные $a1, a2, a3, \dots$

4. Графические функции

plot(x, y) график функции
plot(x, y, стиль) график функции с указанием стиля линии:
 c m y r g b w k цвет линии и маркера
 $-$ $--$ $:$ $-\cdot$ стиль линии
 $+$, o , $*$, x , s , d , \wedge , v , $>$, $<$, p , h тип маркера
xlabel('Text') подпись к оси Ox
ylabel('Text') подпись к оси Oy
title('Text') заголовок сверху графика
clf очищает текущее графическое окно
shg выдвигает текущее графическое окно вперед
figure создает новое графическое окно и делает его активным
figure(n) делает активным окно с номером n
hold on переходит в режим сохранения результатов графического вывода
hold off выходит из режима
hold меняет режим

plot(x1, y1, x2, y2, ..., xn, yn) несколько кривых
plot(x1, y1, стиль1, ..., xn, yn, стильn) несколько кривых с указанием их свойств
plot(x, Y) если Y — матрица, то эквивалентно **plot(x, Y(:, 1), ..., x, Y(:, 2))**
legend('текст1', 'текст2', ..., 'текстn') легенда
xlim([xmin, xmax]) диапазон изменения координаты x
ylim([ymin, ymax]) диапазон изменения координаты y
zlim([zmin, zmax]) диапазон изменения координаты z
axis([xmin, xmax, ymin, ymax, zmin, zmax]) диапазоны
axis equal одинаковый масштаб по всем осям
axis square оси координат квадратные
axis on включает отображение осей
axis off выключает отображение осей
grid on включает отображение сетки
grid off выключает отображение сетки
logx(...) аналогично **plot**, но используется логарифмическая шкала по оси Ox
logy(...) аналогично **plot**, но используется логарифмическая шкала по оси Oy
semilog(...) аналогично **plot**, но используется двойная логарифмическая шкала
polar(phi, r) график в полярных координатах
polar(phi, r, стиль) график в полярных координатах с указанием стиля
plot3(x, y, z) график кривой в пространстве
plot3(x, y, z, стиль) график кривой в пространстве с указанием стиля
plot3(x1, y1, z1, x2, y2, z2, ..., xn, yn, zn) несколько графиков в пространстве
plot3(x1, y1, z1, стиль1, ..., xn, yn, zn, стильn) несколько графиков в пространстве с указанием стиля
mesh(X, Y, Z) «проволочная» поверхность
surf(X, Y, Z) закрашенная поверхность
surfl(X, Y, Z) моделирование освещения
colormap палитра задание палитры:
 winter spring summer autumn bone
 copper hot cool gray pink и др.
colorbar отображение цветовой шкалы
hidden on включение режима скрытия невидимых линий
hidden off включение режима отображения невидимых линий
shading faceted режим отрисовки граней
shading interp
shading flat
alpha a задание коэффициента прозрачности
view(az, el) угол обзора (долгота и широта в градусах)
view(x, y, z) точка обзора
contour(X, Y, Z) линии уровня
contour(X, Y, Z, n) n линий уровня
contour(X, Y, Z, [c1, c2, ..., cn]) линии уровня для заданных значений функции
contourf(X, Y, Z) contourf(X, Y, Z, n)
contourf(X, Y, Z, [c1, c2, ..., cn]) то же с закрашиванием промежутков между линиями уровня
camlight headlight размещение источника света в точке наблюдения
camlight right размещение источника света справа сверху от точки наблюдения
camlight left размещение источника света слева сверху от точки наблюдения
camlight то же, что и **camlight right**
camlight(az, el) задание долготы и широты источника света
axis vis3d

```

ezplot('f(x)', a, b) график функции  $f(x)$  на отрезке
[a, b]
ezplot('f(x,y)') график кривой  $f(x, y) = 0$ 
ezplot('x(t)', 'y(t)', a, b) график линии  $x = x(t)$ ,
 $y = y(t)$ ,  $a \leq t \leq b$ 
ezplot3('x(t)', 'y(t)', 'z(t)', a, b) график линии
 $x = x(t)$ ,  $y = y(t)$ ,  $z = z(t)$ ,  $a \leq t \leq b$ 
ezmesh('f(x, y)', [a, b, c, d]) «проволочная»
поверхность  $z = f(x, y)$ ,  $a \leq x \leq b$ ,  $c \leq y \leq d$ 
ezsurf('f(x, y)', [a, b, c, d]) закрашенная
поверхность
ezmesh('x(u, v)', 'y(u, v)', 'z(u, v)', [a, b, c, d])
ezsurf('x(u, v)', 'y(u, v)', 'z(u, v)', [a, b, c, d])
поверхность  $x = x(u, v)$ ,  $y = y(u, v)$ ,  $z = z(u, v)$ ,
 $a \leq u \leq b$ ,  $c \leq v \leq d$ 
ezsurf('f(x, y)', [a, b, c, d]) закрашенная
поверхность
ezcontour('f(x, y)', [a, b, c, d]) линии уровня
ezpolar('r(phi)', [a, b]) график
 $r = r(\phi)$ ,  $a \leq \phi \leq b$  (в полярных координатах)

```

5. Конструкции языка

```

if условие
    команды
end

```

```

if условие
    команды
else
    команды
end

```

```

if условие
    команды
elseif условие
    команды
...
elseif условие
    команды
end

```

```

if условие
    команды
elseif условие
    команды
...

```

```

elseif условие
    команды
else условие
    команды
end

```

```

for переменная = вектор
    команды
end

```

```

for переменная = матрица
    команды
end

```

```

for переменная = матрица
    команды
end

```

```

while условие
    команды
end

```

```

switch выражение
case значение
    команды
case {значение1, значение2, значение3, ...}
    команды
otherwise
    команды
end

```

```

break    немедленный выход из цикла for или while
continue немедленный возврат к проверке условия в
цикле for или while

```

```

all(a)    истина, если и только если все элементы
вектора a ненулевые
all(A)    если A — матрица, то применяет функцию all
к каждому ее столбцу; на выходе — вектор-строка
длины, равной количеству столбцов в A
any(a)    истина, если и только если хотя бы один из
элементов вектора a ненулевой
any(A)    если A — матрица, то применяет функцию any
к каждому ее столбцу; на выходе — вектор-строка
длины, равной количеству столбцов в A
<, >, <=, >=, ==, ~= (поэлементное) сравнение
&, |, ~ поэлементные «и», «или», «не»
&&, || логические «и», «или»

```

6. Массивы структур, массивы ячеек

```

varname.field1, varname.field1 поля структуры
{31, [1, 2], 'Hello'} массив ячеек
a{i} i-й элемент массива ячеек
a(indices) подмассив (срез) массива ячеек

```

7. Программы-сценарии и функции пользователя

Программа-сценарий («скрипт») — это набор команд, записанных в текстовом файле (m-файле). Такие программы работают с переменными рабочего пространства (теми же, с которыми пользователь работает из командной строки). В отличие от них, программы-функции работают со своими локальными переменными.

% начинает строку комментариев

7.1. Функция с подфункциями

```
function [y1, y2, ..., ym] = funcname(x1, x2, ..., xn)
```

команды

```
function [...] = funcname(...)
```

команды

```
function [...] = funcname(...)
```

команды

Область видимости локальных переменных, появляющихся в основной функции не распространяется на подфункции. Область видимости локальных переменных, появляющихся в подфункции, ограничивается этой подфункцией.

7.2. Вложенные (nested) функции

```
function [y1, y2, ..., ym] = func(x1, x2, ..., xn)
```

команды

```

function [...] = func1(...)

    команды

function [...] = func2(...)

    команды

end
end

function [...] = func3(...)

    команды

end

end

function [...] = func4(...)

    команды

end

```

Функции `func1` и `func3` вложены в `func`. Функция `func2` вложена в `func1`. Область видимости локальных переменных, появляющихся в функции (или вложенной подфункции), распространяется на все вложенные подфункции.

8. Суммы, произведения и т.п.

`sum(a)` сумма элементов вектора `a`
`sum(A)` сумма элементов каждого столбца матрицы `A`. Возвращается вектор-строка длины, равной количеству столбцов матрицы `A`
`cumsum(a)` «кумулятивная» сумма
`cumsum(A)` «кумулятивная» сумма для каждого столбца матрицы `A`. Возвращается матрица того же размера, что и `A`
`prod(a)` произведение элементов вектора `a`
`prod(A)` произведение элементов каждого столбца матрицы `A`
`cumprod(a)` «кумулятивное» произведение
`cumprod(A)` «кумулятивное» произведение для каждого столбца матрицы `A`
`diff(a)` вектор разностей
`min(a)` минимальное значение
`max(a)` максимальное значение

9. Линейная алгебра

См. также раздел «матрицы»

`norm(a, p)` p -норма вектора `a`: $\sqrt[p]{\sum_{j=1}^n |a_j|^p}$, где $p \geq 1$.

Возможно значение `p = Inf`
`norm(a)` евклидова норма `norm(a, 2)`
`norm(A, p)` p -норма матрицы `A`. Возможные значения `p = 1, 2, Inf, 'fro'`. Последнее соответствует фробениусовой норме.
`norm(A)` спектральная норма `norm(A, 2)` матрицы `A`
`cond(A, p)` число обусловленности матрицы `A` для p -нормы
`cond(A)` спектральное число обусловленности `cond(A, 2)`

`cond(A)` верхняя оценка для `cond(A, 1)` (оценщик Хэйджера)
`A\b` решение с.л.у. $Ax = b$; псевдорешение для несовместной с.л.у.
`A\B` решение матричного уравнения $AX = B$
`A/B` решение матричного уравнения $YA = B$
`inv(A)` обратная матрица A^{-1}
`pinv(A)` псевдообратная матрица
`[L, U, P] = lu(A)` LU -разложение матрицы `A`: $PA = LU$, где P — перестановочная матрица, L — нижнетреугольная с единичной диагональю, U — верхнетреугольная
`[Q, R] = qr(A)` QR -разложение матрицы `A`: $A = QR$, где Q — унитарная (ортогональная) матрица, R — верхнетреугольная
`B = null(A)` ортонормированный базис пространства решений с.л.у. $Ax = 0$
`B = null(A, 'r')` «рациональный» базис пространства решений с.л.у. $Ax = 0$
`d = eig(A)` собственные числа матрицы `A`
`[Q, D] = eig(A)` возвращается диагональная матрица `D` и матрица `Q`, такие, что $D = Q^{-1}AQ$

10. Интерполяция и аппроксимация данных

`f = polyfit(x, y, n)` возвращает коэффициенты многочлена степени `n`, аппроксимирующего данные
`y = polyval(f, x)` вычисление значения `y` многочлена `f` в точке `x`
`interp1(x, y, xx, 'nearest')` ступенчатая интерполяция
`interp1(x, y, xx)` или `interp1(x, y, xx, 'linear')` кусочно-линейная интерполяция
`interp1(x, y, xx, 'spline')` или `spline(x, y, xx)` кубический сплайн
`interp1(x, y, xx, 'pchip')` или `interp1(x, y, xx, 'cubic')` или `pchip(x, y, xx)` кубический эрмитов интерполянт
`fft(x)` дискретное преобразование Фурье
`ifft(x)` обратное дискретное преобразование Фурье

11. Численное интегрирование

`trapz(x, y)` формула трапеций
`quad(func, a, b)` метод Симпсона
`quad(func, a, b, tol)` метод Симпсона с заданием абсолютной погрешности `tol`
`quadl(func, a, b)` метод Лобатто
`quadl(func, a, b, tol)` метод Лобатто с заданием абсолютной погрешности `tol`
`dblquad(func, a, b, c, d, tol)` двойной интеграл
`triplequad(func, a, b, c, d, e, f, tol)` тройной интеграл

12. Оптимизация и решение систем уравнений

`[x, fval] = fminbnd(func, a, b)` минимизация функции одной переменной на отрезке `[a, b]`; возвращается найденная точка минимума `x` и значение функции в этой точке
`[x, fval] = fminsearch(f, x0)` минимизация функции многих переменных; `x0` — начальное приближение
`[x, fval] = fzero(func, x0)` нуль функции; `x0` — начальное приближение
`[x, fval] = fsolve(func, x0)` решение системы уравнений

13. Обыкновенные дифференциальные уравнения

Задача Коши:

ode45, ode23, ode113 — для нежестких задач;
ode15s, ode23s, ode23t, ode23tb — для жестких задач
[t, y] = ode**(func, [t0, T], y0) решение задачи
Коши для системы диф. уравнений на отрезке
[t0, T]; y0 — начальное значение

14. Symbolic Math Toolbox

s = sym('выражение') создание символьного выражения
syms a b c real создание вещественных символьных переменных
syms a b c unreal создание комплексных символьных переменных
syms a b c то же, что и syms a b c unreal
digits(d) установить количество значащих цифр результата
vpa(s) вычислить символьное выражение
vpa(s, d) вычислить символьное выражение с d цифрами
simplify(s) упрощение символьного выражения
simple(s) перебор разных способов упрощения
expand(s) раскрытие выражения
factor(n) факторизация целого числа
factor(f) факторизация многочлена
subs(s, x, a) подстановка значения
limit(f) $\lim_{x \rightarrow 0} f(x)$
limit(f, a) $\lim_{x \rightarrow a} f(x)$
limit(s, y, a) $\lim_{y \rightarrow a} f(y)$
diff(f) $f'(x)$
diff(f, y) $f'(y)$
diff(f, y, n) $f^{(n)}(y)$
int(f) $\int f(x)dx$
int(f, a, b) $\int_a^b f(x)dx$
int(f, y) $\int f(y)dy$
int(f, y, a, b) $\int_a^b f(y)dy$
solve(s) решение уравнения
solve(s, x) решение уравнения относительно указанной неизвестной
solve(s1, ..., sn) решение системы уравнений
solve(s1, ..., sn, x1, ..., xn) решение системы уравнений относительно указанных неизвестных
syms a b c d x y
solve(a*x^2+b*x+c)
solve(a*x^3+b*x^2+c*x+d)
solve('x^2+x*y+y=3', 'x^2-4*x+3=0')
ans.x, ans.y
solve('x^2+x*y+y=3', 'x^2-4*x+3=0')
dsolve(eq1, ..., eqn, cond1, ..., condn, t) решение системы дифференциальных уравнений
dsolve('Dy+4*y = exp(-t)')
dsolve('Dy+4*y = exp(-t)', 'y(0) = 1')
dsolve('(Dy)^2 + y^2 = 1')
dsolve('(Dy)^2 + y^2 = 1')
dsolve('Dx = y', 'Dy = -x')
dsolve('D2y=-y')

Для матриц с символьными элементами переопределены многие функции линейной алгебры и др.