

# Краткая сводка по языку MATLAB

Н.Ю. Золотых

05.05.08

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 « ... ». % начинает строку комментариев

## 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  
pi число  $\pi$   
Inf бесконечность  $\infty$   
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  $\text{sign } x$   
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  $\lfloor x \rceil$   
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  
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([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 разность матриц

a\*A произведение числа a на матрицу A

A\*B матричное произведение

A^n матричная степень

A.\*B покомпонентное произведение

A./B покомпонентное деление

A.^B покомпонентная степень

A' комплексно сопряженная матрица

A.' транспонированная матрица

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) упрощенный 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

`[X, Y] = meshgrid(x, y)` генерирование решеток  
См. также раздел «линейная алгебра»  
Математические функции, примененные к матрицам  
действуют покомпонентно.

### 3. Сохранение и загрузка данных

`save filename` сохранение значений всех переменных  
рабочего пространства в `mat`-файле  
`save filename a1 a2 ... an` сохранение значений  
указанных переменных  
`load filename` загрузка данных из файла  
`load filename a1 a2 ... an` загрузка указанных  
переменных  
`save a -ascii` сохранение значения переменной в  
одноименном текстовом файле  
`load a -ascii` загрузка переменной из текстового файла

### 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  
  `-- : -. .` стиль линии  
  `+, o, *, x, s, d, ^, 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  
`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, стильn)`  
несколько графиков в пространстве с указанием стиля  
`mesh(X, Y, Z)` «проволочная» поверхность  
`surf(X, Y, Z)` закрашенная поверхность  
`surf1(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])` «проволоч-  
ная» 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. Функции пользователя

### 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.

<code>sum(a)</code>	сумма элементов вектора <code>a</code>
<code>sum(A)</code>	сумма элементов каждого столбца матрицы <code>A</code> . Возвращается вектор-строка длины, равной количеству столбцов матрицы <code>A</code>
<code>cumsum(a)</code>	«кумулятивная» сумма
<code>cumsum(A)</code>	«кумулятивная» сумма для каждого столбца матрицы <code>A</code> . Возвращается матрица того же размера, что и <code>A</code>
<code>prod(a)</code>	произведение элементов вектора <code>a</code>
<code>prod(A)</code>	произведение элементов каждого столбца матрицы <code>A</code>
<code>cumprod(a)</code>	«кумулятивное» произведение
<code>cumprod(A)</code>	«кумулятивное» произведение для каждого столбца матрицы <code>A</code>
<code>diff(a)</code>	вектор разностей
<code>min(a)</code>	минимальное значение
<code>max(a)</code>	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  $p = \text{Inf}$

<code>norm(a)</code>	евклидова норма <code>norm(a, 2)</code>
<code>norm(A, p)</code>	$p$ -норма матрицы <code>A</code> . Возможные значения $p = 1, 2, \text{Inf}, 'Fro'$ . Последнее соответствует фробениусовой норме.
<code>norm(A)</code>	спектральная норма <code>norm(A, 2)</code> матрицы <code>A</code>
<code>cond(A, p)</code>	число обусловленности матрицы <code>A</code> для $p$ -нормы
<code>cond(A)</code>	спектральное число обусловленности <code>cond(A, 2)</code>
<code>condest(A)</code>	верхняя оценка для <code>cond(A, 1)</code> (оценщик Хэйджера)
<code>A\b</code>	решение с.л.у. $Ax = b$ ; псевдорешение для несовместной с.л.у.
<code>A\B</code>	решение матричного уравнения $AX = B$
<code>A/B</code>	решение матричного уравнения $YA = B$
<code>inv(A)</code>	обратная матрица $A^{-1}$
<code>pinv(A)</code>	псевдообратная матрица
<code>d = eig(A)</code>	собственные числа матрицы <code>A</code>
<code>[Q, D] = eig(A)</code>	возвращается диагональная матрица <code>D</code> и матрица <code>Q</code> , такие, что $D = Q^{-1}AQ$

## 10. Интерполяция и аппроксимация данных

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

## 11. Численное интегрирование

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

## 12. Оптимизация и решение систем уравнений

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

## 13. Обыкновенные дифференциальные уравнения

Задача Коши:

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

## 14. Symbolic Math Toolbox

<code>s = sym('выражение')</code>	создание символьного выражения
<code>syms a b c real</code>	создание вещественных символьных переменных
<code>syms a b c unreal</code>	создание комплексных символьных переменных
<code>syms a b c</code>	то же, что и <code>syms a b c unreal</code>
<code>digits(d)</code>	установить количество значащих цифр результата
<code>vpa(s)</code>	вычислить символьное выражение
<code>vpa(s, d)</code>	вычислить символьное выражение с $d$ цифрами
<code>simplify(s)</code>	упрощение символьного выражения
<code>simple(s)</code>	перебор разных способов упрощения
<code>expand(s)</code>	раскрытие выражения
<code>factor(n)</code>	факторизация целого числа
<code>factor(f)</code>	факторизация многочлена
<code>subs(s, x, a)</code>	подстановка значения
<code>limit(s)</code>	$\lim_{x \rightarrow 0} f(x)$
<code>limit(s, a)</code>	$\lim_{x \rightarrow a} f(x)$
<code>limit(s, y, a)</code>	$\lim_{y \rightarrow a} f(y)$
<code>diff(f)</code>	$f'(x)$
<code>diff(f, y)</code>	$f'(y)$
<code>diff(f, y, n)</code>	$f^{(n)}(y)$
<code>int(f)</code>	$\int f(x)dx$

`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.