



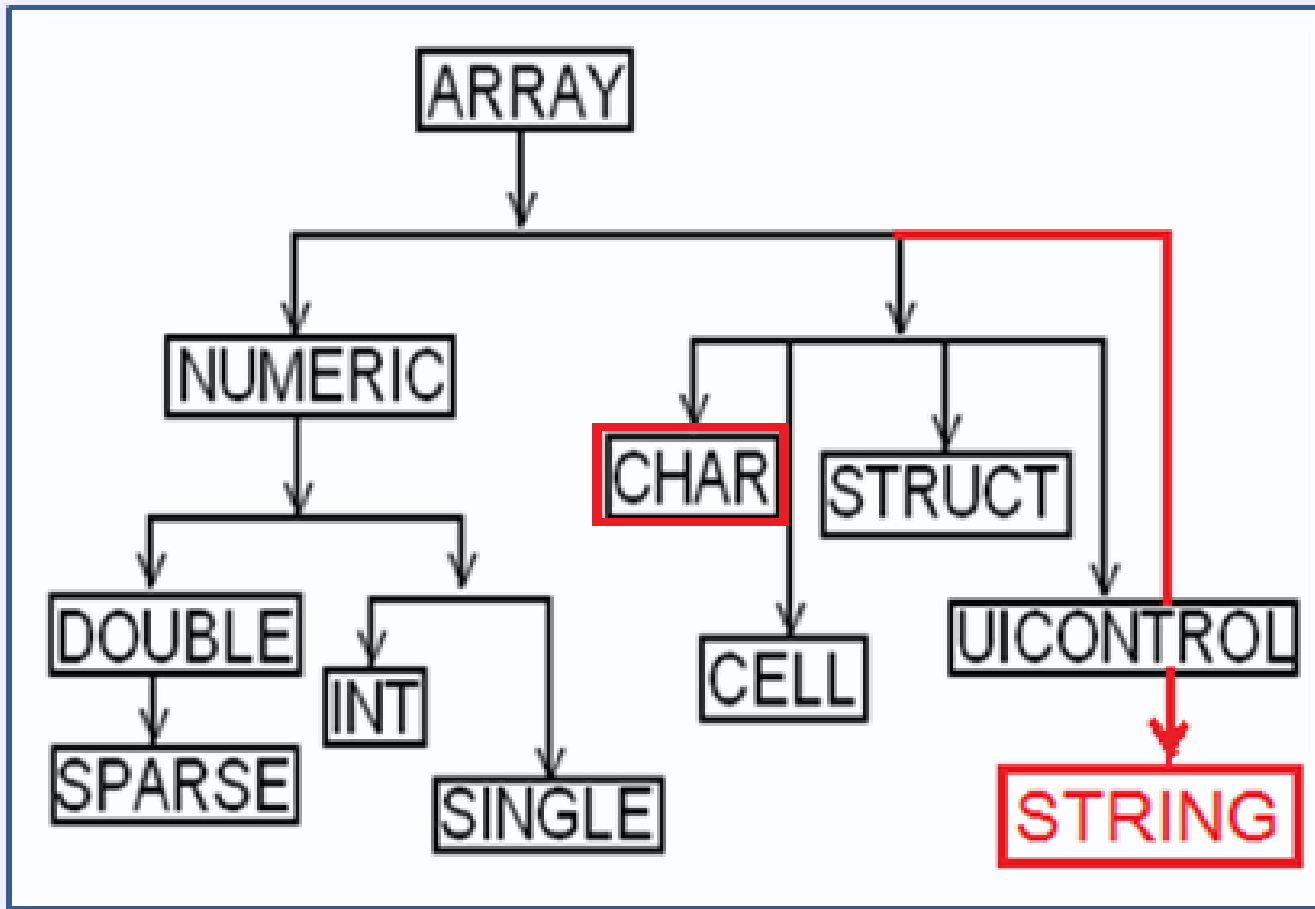
Пакеты научных вычислений

Лекция 2. Иерархия классов MatLab. Классы Double, Char, String, Cell

Курбатова Наталья Викторовна, к.ф.-м.н.,
доцент кафедры математического
моделирования, мехмат, ЮФУ



Схема иерархии классов





Содержание:

Array - всё Array!

- ndims , size, length , numel
- display

Numeric, Double

- Конструкторы матриц
- Функции от элементов матриц и определяющие свойства матриц (eig, det, rank, cond...)
- СЛАУ. Метод Гаусса.
- Построение простейших графиков
- Операции I/O

String, Char, Cell

- Конструкторы
- Основные функции классов



Array

Наследуемые потомками основные функции:

`N=ndims(a)` – количество размерностей

`[m, p] = size(a)` – число строк, столбцов

`numel` – общее количество элементов (равно $m * p$)

`length(a)` – по умолчанию: число строк

`length(a,2)` – по умолчанию: число столбцов (длина вдоль столбцов, т.е. второй размерности)

`display (a);` – точка с запятой не подавляет вывода



Numeric. Double

Конструкторы матриц:

- **zeros** – из нулей
- **eye** – единичной
- **ones** – из единиц
- **rand**, **randi**, **randn** – случайных
- **magic(n)** – матрица со свойствами магического квадрата, суммы любой строки и столбца равны (существует не для любого порядка)
- **blkdiag** – блочно-диагональной:

```
C={ones(3),randi(20,2,4),rand(3), triu(ones(2))}; % C∈Cell  
blkdiag(C{:}) % C∈Cell - массив ячеек
```



Конструкторы матриц:

- **triu** – верхнетреугольной

$\text{triu}(A)$; $\text{triu}(A,-k)$; $\text{triu}(A,k)$;

- **tril** – нижнетреугольной

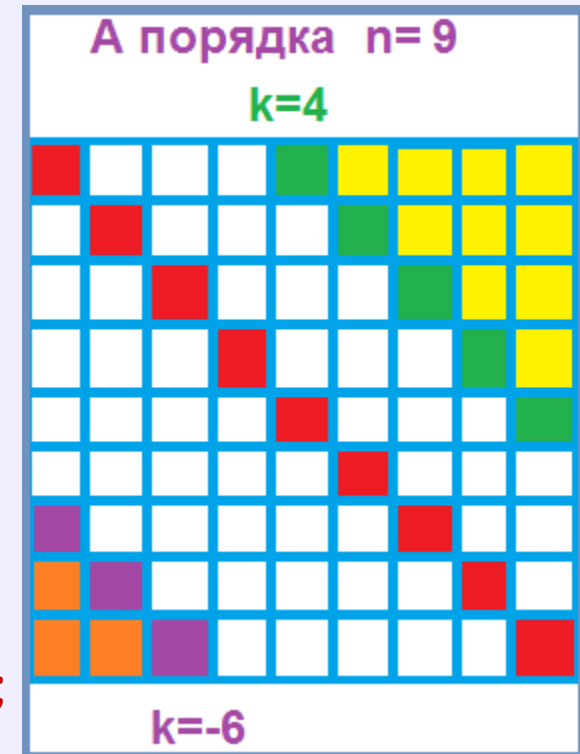
$\text{tril}(A)$; $\text{tril}(A,-k)$; $\text{tril}(A,k)$;

- **diag** – диагональной

$a = \text{diag}(A)$; $a = \text{diag}(A,k)$; $a = \text{diag}(A,-k)$;

$A = \text{diag}(a)$; $A = \text{diag}(a,k)$; $A = \text{diag}(a,-k)$;

$k > 0$, целое, A – матрица, a – вектор





Векторное редактирование матриц по условию

Синтаксические форма поиска элементов по условию:

- $\text{LogicalMatrix} = \mathbf{A} \text{ RelationOperation } \mathbf{B}$
RelationOperation: $>$, $<$, $=$, ... \mathbf{A} - matrix, \mathbf{B} – константа или матрица...

Синтаксические форма редактирования элементов по условию:

- $\mathbf{A}(\mathbf{A} \text{ RelationOperation } \mathbf{B}) = \text{const}$ – присвоение элементам матрицы, отвечающим условию, новых значений

Пример

$$\mathbf{A} = [0, 0, -2; 0, -1, -1; -2, -2, 0]$$

$$\text{LogicalMatrix} = \mathbf{A} < -1$$

$$\mathbf{A}(\mathbf{A} < -1) = 3$$

LogicalMatrix – какой вид?

А матрица А ?

$\mathbf{A} =$			
	0	0	-2
	0	-1	-1
	-2	-2	0



Анализ элементов матриц: функция: find

Синтаксические формы поиска элементов по условию:

- `index=find(A)` % индексы элементов $A \neq 0$
в `A(index)` или в `A(index(:))` находятся ненулевые элементы;
- `[i,j]=find(AnyCondition)` % *i,j* индексы элементов, удовлетворяющие условию, f.e., `AnyCondition={A>value или A<value или A==value,...}`;

Элементы $A(i,j)$ совпадают с A_{cond} !

Brain Storm!

`index=find(A)` % определите содержание `index`!

`[i,j]=find(A==-2)` % определите содержание `i,j`!

A =			
	0	0	-2
	0	-1	-1
	-2	-2	0



Логический анализ элементов матриц: функции: all, any

Анализ операций отношения матриц:

- Анализируем \forall отношение, f.e. $C=A==B$, результат соразмерная логическая матрица C из **нулей** (нет равенства соответствующих элементов) и **единиц** (выполняется равенство).
- $TrueFalse=all(all(C)); TrueFalse=1$, если все элементы в C равны **1** или **true**, здесь A, B – двумерные матрицы и $C=A==B$
- $TrueFalse=any(any(C)); TrueFalse=1$, если хотя бы один элемент A равен соответствующему в B (т.е. матрица C имеет хоть одно истинное значение)

Проверьте для произвольной матрицы a , является ли матрица $a'*a$ – симметричной?



Анализ элементов матриц. Примеры

Поиск элементов с заданным условием:

*% Генерируем целочисленную матрицу
% размера 2x3, величина элементов
% принадлежит отрезку [-9,9]:*

```
a=randi([-9,9],2,3); n=nnz(a)
```

*% Находим anz - ненулевые элементы
% и их индексы i,j, такие, что $a(i(k),j(k)) = anz(k)$, $k=1:n$*

```
[i,j,anz]=find(a);
```

```
[i,j,anz] % упорядочены, см. ans в CommandWindow  
for k=1:n,anz(k)=a(i(k),j(k)),end % testing
```

А что будет в результате? Объясните!

```
[k]=find(a)
```

```
a(k(:)) % см. Рис.
```

```
>> [i, j, anz]
ans =
     1     1    -7
     1     2     9
     2     2    -3
     1     3     2
     2     3    -5

>> a
a =
    -7     9     2
     0    -3    -5
```

Рис.



Функции от матриц

- ✓ $\sin(a)$, $\cos(a)$, $\tan(a)$ – a в радианах
- ✓ $\text{sind}(b)$, $\text{cosd}(b)$, $\text{tand}(b)$ – b в градусах (degrees)
- ✓ atan , asin , acos – (традиционно)
- ✓ $\log(x) \sim \ln(x)$; \log_{10} – десятичный логарифм
- ✓ imag , real – (мнимая часть элементов и вещественная)

```

12 - b=[pi/3 pi/2; 3*pi/2 pi]
13 - a=sin(b)
14
Command Window
a: 2x2 double =
    0.8660    1.0000
   -1.0000    0.0000
1.0472    1.5708
4.7124    3.1416

```

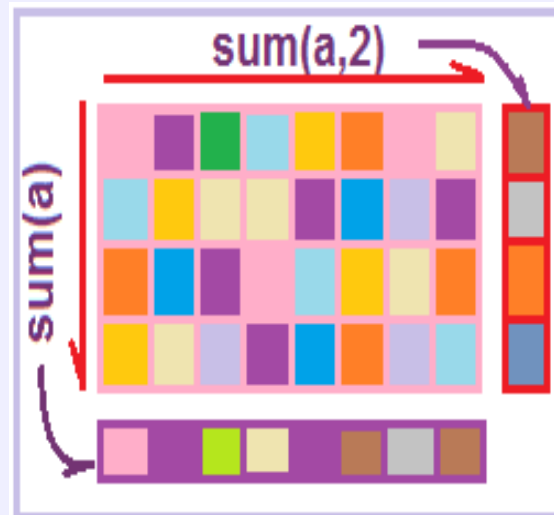
sum, prod, abs, min, max, sort – функции выполняют операцию по колонкам, вдоль строк – default, f.e. **sum(a)~sum(a,1)**;

```

15 - a=ones(3), b= repmat([1,2,3],3,1)
16 - a=a.*b, s=sum(a)
17
Command Window
s: 1x3 double =
    3    6    9

a =
    1    2    3
    1    2    3
    1    2    3

```



sum(a,2); по строкам, вдоль всех колонок

sum(sum(a)) ?



Функции, определяющие природу матриц

- $s=\det (a)$ – вычисляется значение определителя
- $[v,D]=\text{eig}(a)$ – v – матрица из собственных векторов-столбцов, D – диагональная матрица, на диагонали $D(i,i)$ – собственные значения
- $\text{cond}(a)$ – число обусловленности; характеризует устойчивость решения
- issymmetric (ishermitian for complex), isfull , issparse , isequal
- $\text{rank}(a)$ – вычисляется ранг матрицы
- $\text{inv}(a)$ или $a^{(-1)}$ вычисление обратной матрицы ☹
- $\text{norm}(a,p)$ →

Виды норм:

p	Matrix	Vector
1	$\max(\text{sum}(\text{abs}(X)))$	$\text{sum}(\text{abs}(X))$
2	$\max(\text{svd}(X))$	$\text{sum}(\text{abs}(X).^2)^{(1/2)}$
real value p	-	$\text{sum}(\text{abs}(X).^p)^{(1/p)}$
Inf	$\max(\text{sum}(\text{abs}(X')))$	$\max(\text{abs}(X))$
-Inf	-	$\min(\text{abs}(X))$

Разложение svd:

$$[U,S,V] = \text{svd}(X);$$

$$X = U*S*V'$$

U,V - унитарные матрицы,

$$(U*U')=I, (U'*U)=I,$$

здесь ' - транспонирование и комплексное сопряжение

Эвклидова норма где? Ищем!



СЛАУ . Метод Гаусса

$$Ax=b$$

$$\begin{cases} 3x + y + z = 5 \\ -x + y = 1 \\ x + 2y - z = 5 \end{cases} \quad A = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 1 & 0 \\ 1 & 2 & -1 \end{bmatrix} \quad b = \begin{bmatrix} 5 \\ 1 \\ 5 \end{bmatrix}$$

B ML:

```
>>A=[3,1,1; -1 1 0; 1, 2, -1]
```

```
>>b =[5 1 5]'
```

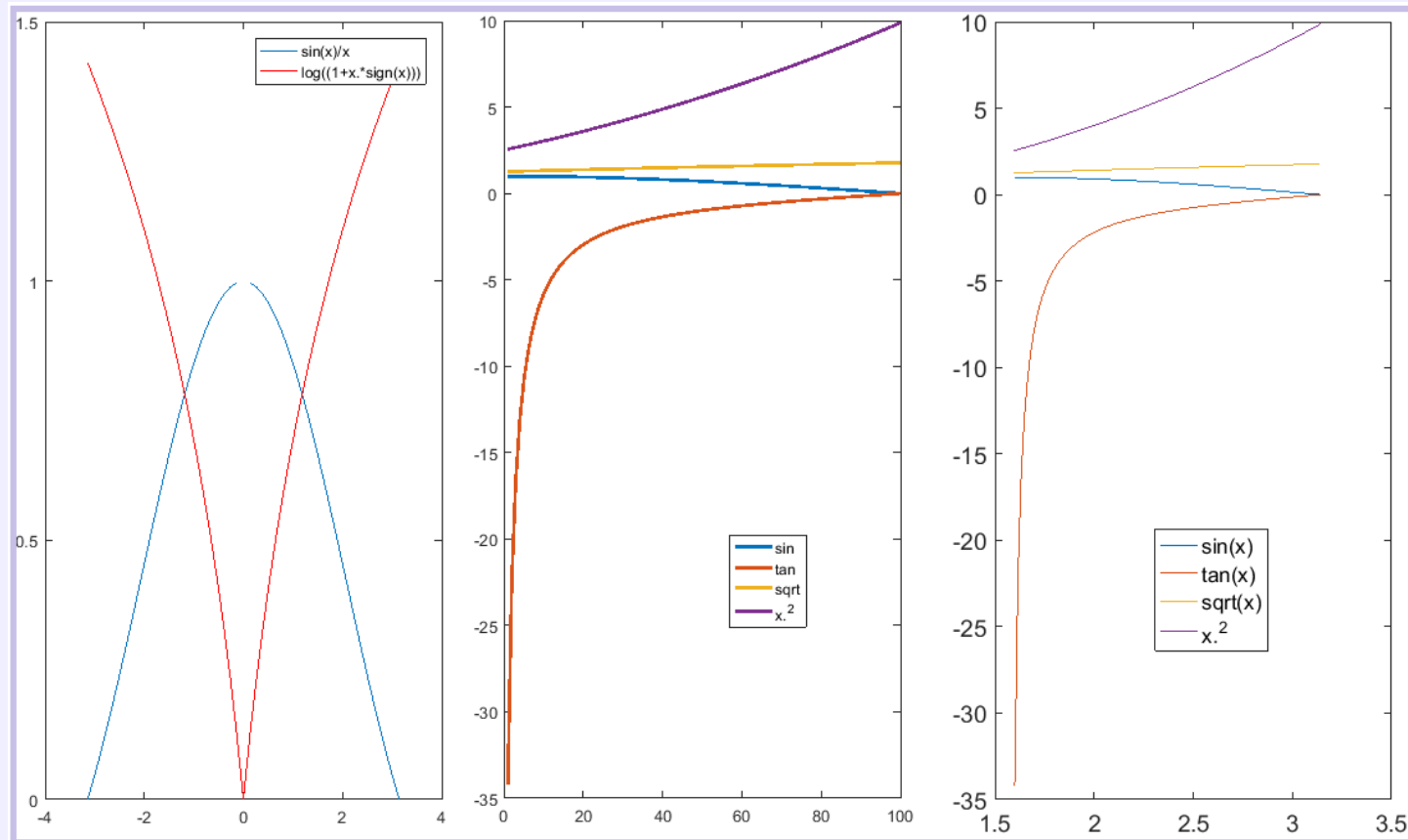
```
>>x=A\b % Метод Гаусса или обратное деление
```



Graph.m

```
clear
x=-pi/24:pi;
y= sin(x)./x; % векторная
операция
z=log((1+x.*sign(x)));
subplot(1,3,1);
plot(x,y); hold on;
plot(x,z,'r-');
legend('sin(x)/x',...
'log((1+x.*sign(x)))');
subplot(1,3,2);
% clear x % (если хотите)
x=linspace(1.6,3.1415,100); %
z=[sin(x)', tan(x)', sqrt(x)' x.^2'];
r=plot(z);
legend({'sin','tan','sqrt','x.^2'});
subplot(1,3,3);
plot(x,z);
legend({'sin(x)','tan(x)',...
'sqrt(x)','x.^2'});
set(r(:),'linewidth',2);
set(gca,'fontsize',14);
```

Простейшие графики объектов Double



subplot(m,n,k) % m×n графических окон, активное k-окно;

hold on меняет свойство поля **NextPlot** объекта **Figure**:

с **replace** (по умолчанию на) **add**;

Чем это чревато?

x=linspace(a,b,n) % вектор из n – точек на отрезке [a,b];

gca (graphic current axes) - активные оси

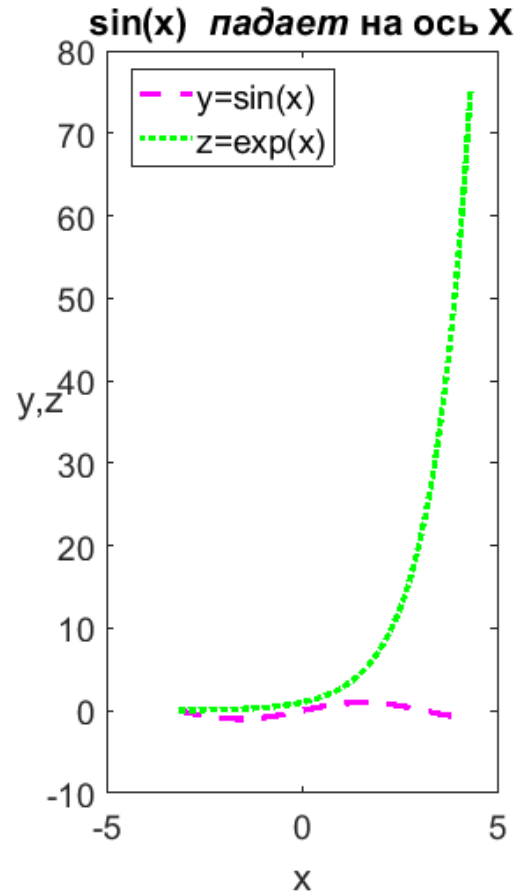


Графики функций разного роста. PLOTYY

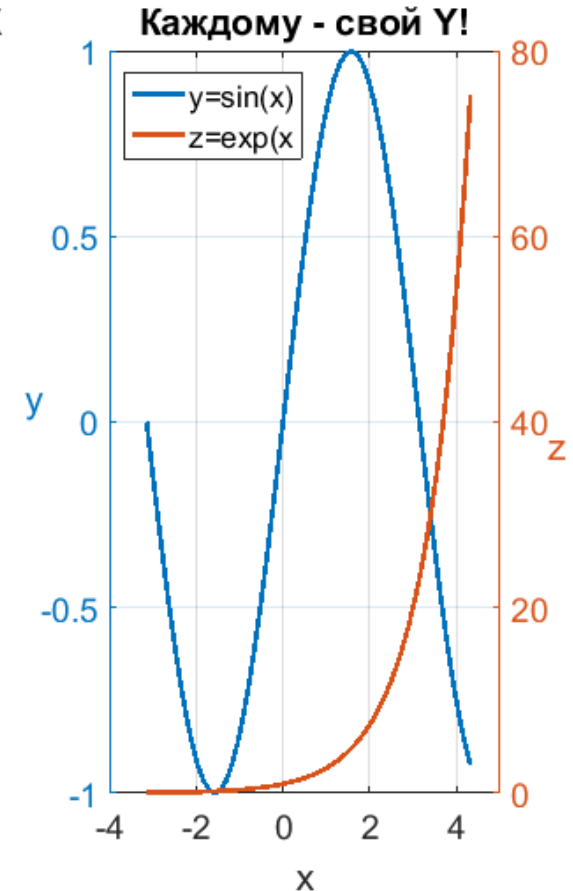
DifferentIncrease.m

```
x=-pi:pi/24:1.4*pi; y= sin(x);  
z=exp(x);  
subplot(1,2,1);  
title('sin(x) {\it падает} на ось X')  
plot(x,y,'m--','linewidth',2); hold  
on  
plot(x,z,'c','linewidth',2);  
lg1=legend('y=sin(x)','z=exp(x)');  
lg1.FontSize=14;  
xlabel('x'),  
yl=ylabel('y,z','rotation',0),  
yl.FontSize=14  
set(gca,'fontsize',14)  
subplot(1,2,2);  
title('Каждому – свой Y!')  
[haxes,line1,line2]=plotyy(x,y,x,z);  
line1.LineWidth=2;  
line2.LineWidth=2;  
xlabel('x'); grid on;  
ylabel(haxes(1),'y','rotation',0);  
ylabel(haxes(2),'z','rotation',0);  
legend('y=sin(x)','z=exp(x)');  
set(haxes(1),'fontsize',14);  
set(haxes(2),'fontsize',14)
```

PLOT



PLOTYY



При использовании команд set и get имена полей структуры свободны от регистра!
В нотации имяОбъекта.имяСвойства=ЗначениеСвойства **важен регистр!**



String array

Конструктор строк – двойные кавычки (" ")

```
1 clear
2 %% STRING Array
3 %% Creation
4 S="MEXMAT FOREVER!"
5 S1=isletter(S);
S: 1x1 string =
"MEXMAT FOREVER!"
```

S - СКАЛЯР!

Контроль букв в строке - **ISLETTER**

```
5 S1=isletter(S);
6
S1: 1x15 logical array =
 1  1  1  1  1  1  0  1  1  1  1  1  1  1  0
```

```
1 clear
2 %% STRING Array
3 %% Creation
4 S="MEXMAT FOREVER!"
5 S1=isletter(S);
6 S2=split(S) %split(S,delimiter,dim)
7 length(S2)
8 %% Edition
9 S3=S2(1)+".SFEDU" % concatenation
10 s3=lower(S3) % S3=upper(S3) %register
11 erase(S3,' ') % S3=erase(S3,' ')
12 S3
```

Command Window

```
New to MATLAB? S
S3 =
"MEXMAT .SFEDU"
s3 =
"mexmat .sfedu"
ans =
"MEXMAT.SFEDU"
```

S2: 2x1 string =
"MEXMAT "
"FOREVER!"

Чему равен результат строки 12?

SPLIT - разделение строки на элементы (слова) столбца; пробел разделитель по умолчанию (см.S2)

UNIQUE(STRING) выбор неповторяющихся *char* в **STRING**

ERASE – удаление заданного элемента в строке (см.S3)

LOWER, UPPER – преобразование к соответствующему регистру (см.S3)



Char - character array

Конструктор строк Char – одинарные кавычки (' ')

```
Command Window
>> whos

Name      Size  Bytes  Class
a         1x6   12     char
b         1x1    8     double

Workspace
Name      Value
a         'letter'
b         1
```

```
2 - a=upper('letter') % as for string
3 - l=length(a);
4 - it=a(4:end) % Total Expense Ratio
5 - res=imag(it)
6 - b='ТОТ ПОТОП'; % almost Palindrom
7 - b1=fliplr(b)
```

```
Command Window
K>> b1
b1 = ПОТОП ТОТ

K>> whos b res
Name      Size  Bytes  Class
b         1x9   18     char
res       1x3   24     double
```

В чем отличия от STRING?

Почему **a занимает 12 байт**

а **res – 24 байта?**

Справедливо для Char и String:

upper, lower – конвертирование строки в заглавные или в строковые (выбор регистра)

fliplr – (сальто) инверсия слева направо (сравни b, b1! *А Уверен и не реву?*)

length(CHAR); strlen(STRING) – количество характеристик;

>>(length(b) % равно 9;

>>strlen('TRUE') % равно 4, хотя 'TRUE' **скаляр!**

Вычисления в классе Double – двойной точности – 8 байт!

А flipud как работает? *Мозговой штурм*) ¹⁷



Char , String. Редактирование

Связь массивов строк, ячеек и характеристик: **StrArr=string(CellChar)** – преобразование в соразмерный с **CellChar** массив **StrArr** из строк (**string** – конвертирование)

```
Help Center | mathworks.com/help/matlab/ref/string.html
Documentation | Examples | Functions | Apps | Videos | Answers

A = {'Mercury','Gemini','Apollo';...
     'Skylab','Skylab B','ISS'}

A = 2x3 cell
     {'Mercury'}  {'Gemini' }  {'Apollo'}
     {'Skylab' }  {'Skylab B'}  {'ISS'   }

str = string(A), str(1,2)

str = 2x3 string
     "Mercury"  "Gemini"   "Apollo"
     "Skylab"   "Skylab B"  "ISS"

ans =
     "Gemini"
```

str2mat – конвертирование характеристик в матрицу (в рамках машинного хранения , как колонку)

```
10 B=str2mat(A);
    str 2x3 string
    B =
    Mercury
    Skylab
    Gemini
    Skylab B
    Apollo
    ISS
```

```
K>> A = ['Mercury','Gemini','Apollo';...
         'Skylab','Skylab B','ISS']
Dimensions of matrices being concatenated are not
consistent.
```

←-----**Что не так?**
Как поправить?
Сформулируем правило!

MercuryGeminiAppolo
SkyLab SkyLab B ISS



Массив ячеек Cell и Char

объединяет разнотипные объекты в *контейнер*

- `C=cell(size)` – резервирует структуру массив ячеек заданием размера
- `C={A,B,D,...}` – `A,B,D,...` - \forall элементы MatLab

```
Command Window
C =
 2x1 cell array

 'my mind is strong'
 'my mind is resting'
TF =
 logical
 0
ans =
 logical
 1

Editor - F:\Nata\2017\Matlab 2017\2023\Лекции Work Folder\MyPresentation\Le
KurbatovaStart.m x ForLecture2.m x StringChar.m x +
1 - clear
2 - %% comparisen
3 - C={'my mind is strong';...
4 - 'my mind is resting'}
5 - TF=strncmpi(C{1},C{2}) % C(1),C(2)
6 - strncmp(C{1},C{2},10)
```

```
Command Window
ind =
     4
S =
my mind is strongmy mind is resting
ans =
 1x12 logical array
 1 1 1 1 1 1 1 1 1 1 1 1
```

```
[ind]=findstr(C{1},'mind')
S=[C{1},C{2}]; CS=string(C) % Размер CS?
[Ced,ind1]=unique(S);
S(ind1)==Ced %
[Ced,ind1]=unique(CS);
```

Сравнение:

- `ScalarTrueOrFalse = strcmp(S1,S2)` – сравнение характеристик в целом
- `[LogicalVector]=S1==S2;` или `eq(S1,S2);`
- `TF=strncmp(S1,S2,n);` % - первых n элементов
- `TF=strcmpi(S1,S2)` % full comparisen

Конвертирование: String ↔ Char ↔ Double ↔ Cell

Charac=char(Str);

Класс String относительно
новый для MatLab!

Синтаксис двойных кавычек –
ошибка для предыдущих версий!

Функционал работы со строками
покрывается в классе CHAR !

В помощь массив ячеек Cell:

CS=strsplit(S,dlm) – конвертирует
каждое слово **S**, заключенное между
разделителями **dlm** в **S ∈ Char** в элемент
CS ∈ Cell; **dlm** пробел – по умолчанию

strsplit('Диван незаразен на вид - ПАЛИНДРОМ!');

→

```
res=double('xyz') % ASCII code
% res=120 121 122
str2double('pi/2'); % вычислит!
Charac=num2str(ClassDouble) % ∈ Char
legend(Charac)
str2double('5+0.17i') % 5.0000 + 0.1700i
str2num(['1 2';'3 4']) % ans =
      1   2
      3   4
```

Str=string(Charact);

Если не хватает функционала в классе
характеристик CHAR, то конвертируем в класс
строк String и наоборот!

```
ans =
      1x6 cell array
      Columns 1 through 6
      'Диван' 'незаразен' 'на' 'вид' '-' 'ПАЛИНДРОМ!'
```



Cell ↔ Double

`cell2mat` с учетом согласования размерностей

Пример 1: Пусть c – массив ячеек из пункта (3)
`r=cell2mat(c), % c = {[1] [2 3 4]; [5; 9] [6 7 8; 10 11 12]}`

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$$

`mat2cell` `r=mat2cell(a,m,n)`, здесь m – вектор количества строк подматриц; n – столбцов подматриц;

`num2cell` `r=num2cell(a,m,n)`, экспортрование данных типа `double` и `char`

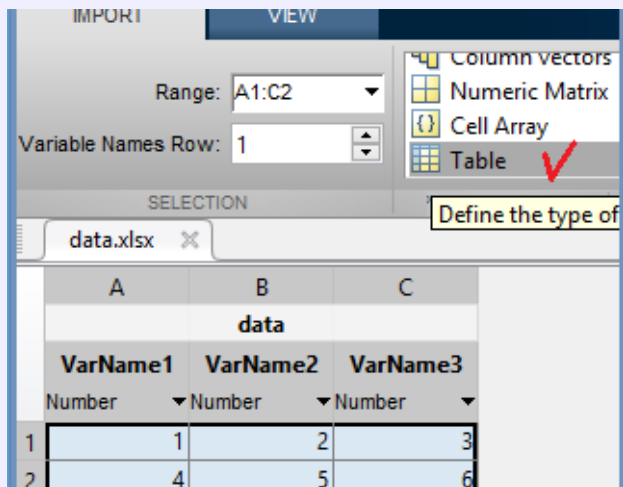
- `[DoubleV]=Cell2mat(C)` – см. Пример1
- `[CellV]=mat2cell(DoubleV)`
- `[CellV]=num2cell(DoubleV)`

```
>> a=randi(4,4,4),  
>> r=mat2cell(a,[1,3],[2,2])  
  
a =  
    1     3     2     1  
    3     0     1     2  
    1     2     0     3  
    0     2     1     1  
  
r =  
    [1x2 double]    [1x2 double]  
    [3x2 double]    [3x2 double]
```



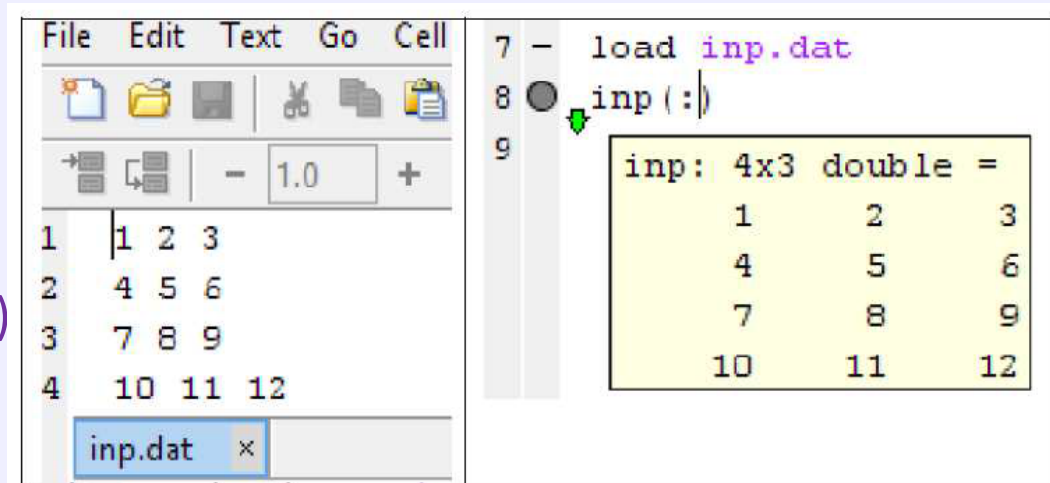
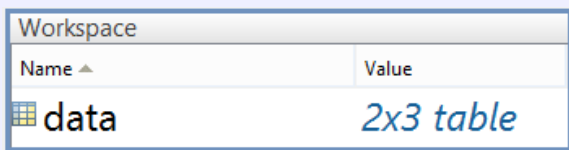
Операции I/O ~ load/save

Import



Home → Import Data
Выбирайте нужный тип файла!
...правильный тип данных в ML!

load, save:



Двоичные файлы:

`load file % file.mat (saved before)`

Текстовые файлы:

`load file.txt % inp.dat`



xlsread , xlsxwrite (Excel)

inex.xls

	A	B
1	0,1869	0,7547
2	0,4898	0,276
3	0,6463	0,7655
4	0,7094	0,7952
5	0,7547	0,1869
6	0,6797	0,4456

- Путь здесь не указан, т.к. файл *.xls (*.xlsx) в рабочей папке
- Имя листа – на английском (лучше! зависит от версии, на русском может быть ошибка)
- [STATUS,MESSAGE] = xlsxwrite(file,array,sheet,range), file, array – обязательные параметры

```
10 - exmatr=xlsread('inex.xls')
11 ● → d
12      |
      | exmatr: 6x2 double =
      |
      | 0.1869    0.7547
      | 0.4898    0.2760
      | 0.6463    0.7655
      | 0.7094    0.7952
      | 0.7547    0.1869
      | 0.6797    0.4456
```




Чтение разнородной таблицы в Excel. Пример

[a,b,c]=xlsread('data1.xlsx')

a - вещественная часть таблицы

b – массив ячеек, содержащий нетипичный формат, например, даты;

c – массив ячеек, объединяющий **a** и **b**.

	A	B	C	D
1	01.01.2023	2	3	1
2	04.01.2023	5	6	4
3	07.01.2023	7	12	0,3
4		1	2	2
5	13.01.2023	2	7	3

a =

```
2.0000    3.0000    1.0000
5.0000    6.0000    4.0000
7.0000   12.0000    0.3000
1.0000    2.0000    2.0000
2.0000    7.0000    3.0000
```

b =

```
5x1 cell array
'01.01.2023'
'04.01.2023'
'07.01.2023'
''
'13.01.2023'
```

c =

5x4 cell array

```
'01.01.2023'    [2]    [ 3]    [ 1]
'04.01.2023'    [5]    [ 6]    [ 4]
'07.01.2023'    [7]   [12]   [0.3000]
[          NaN]    [1]    [ 2]    [ 2]
'13.01.2023'    [2]    [ 7]    [ 3]
```



fscanf, fprintf (форматированный текст)

`Fid=fopen(name, opts)`

`A=fscanf(Fid, format, [m,n])` % [m,n] – размер A

`fclose(fid);`

Если `n=inf`, то чтение в `m` строк до исчерпания столбцов

```
- fid1=fopen('inpnum.txt','r')
- A=fscanf(fid1,'n=%d eps=%g x0=%f\n',[3,4])
```

```
A: 3x4 double =
    1.0000    2.0000    3.0000    4.0000
    0.2000    0.0500    0.0001    0.0000
    3.2564    3.1419    3.1417    3.1415
```

```
clear
- fid=fopen('inpinf.txt','r')
- A=fscanf(fid,'%d %g\n',[2,inf])
```

```
A: 2x4 double =
    1.0000    2.0000    3.0000    4.0000
    0.2000    0.0500    0.0001    0.0000
```

	-	1.0	+	÷	1.1	×
1	n=1	eps=2.0e-01	x0=3.2564			
2	n=2	eps=5.0e-02	x0=3.1419			
3	n=3	eps=1.0e-04	x0=3.1417			
4	n=4	eps=1.0e-05	x0=3.1415			

inpnum.txt ×	
1	1 2.0e-01
2	2 5.0e-02
3	3 1.0e-04
4	4 1.0e-05
inpinf.txt ×	



Опции форматированного I/O

<code>\n</code>	Переход на новую строку
<code>\t</code>	Горизонтальная табуляция
<code>\b</code>	Возврат назад на один символ

пробелы в строке формата записываются как пробелы

<code>%c</code>	Последовательность символов
<code>%d</code>	Десятичные числа
<code>%e, %f, %g</code>	Числа с плавающей запятой
<code>%i</code>	Целое число со знаком
<code>%o</code>	Восьмеричное целое число со знаком
<code>%s</code>	Ряд символов без пробелов
<code>%u</code>	Десятичное целое число со знаком
<code>%x</code>	Шестнадцатеричное целое число со знаком

ТЕМАТИЧЕСКАЯ СПРАВКА:

https://www.mathworks.com/help/matlab/matlab_prog/formatting-strings.html



Спасибо за внимание!

“Как бы медленно ты ни шёл, не останавливайся!”

孔子 (Конфуций)