

Анимация в Matlab

Пример #01. Анимированный график на плоскости

```
% Рисуем картинку
x = 0:0.01:1;
y = x;
plot(x,y)
% Захватим кадр
f = getframe;

% преобразуем полноцветное изображение в палитровое
% Полноцветное изображение хранится в f.cdata.
% Оно имеет размер NxMx3.
% Вместо него получаем палитровое изображение
% im размера NxM, цвет каждого пиксела которого
% определяется цветовой картой (палитрой) map.
% 256 - количество цветов в палитре
[im,map] = rgb2ind(f.cdata,256);

% Задаем массив, в котором будем хранить кадры анимации
% Результатом работы rgb2ind может быть двумерный или
% четырехмерный массив.
% Так что введение дополнительного (третьего) измерения -
% дело вынужденное.
% Значение индекса четвертого измерения (10) -
% количество кадров будущей анимации.
% Таким образом мы заранее резервируем место под массив im.
im(1,1,1,10) = 0;

% Цикл рисования и запоминания кадров
% Рисуем очередной кадр, захватываем его (getframe)
% и добавляем в массив im.
for k = 1:10
    y = x.^k;
    plot(x,y)
    f = getframe;
    im(:,:,1,k) = rgb2ind(f.cdata,map);
end

% Записываем полученную анимацию в файл
% DelayTime - определяет время задержки между кадрами анимации
% LoopCount - задает число повторений.
% Один раз анимация воспроизводится всегда,
% так что LoopCount=0 означает, что анимация будет
% воспроизведена один раз,
% при LoopCount=1 анимация выполняется дважды и т.д.
% LoopCount=inf закикливает анимацию (повторяет ее бесконечное
% число раз).
imwrite(im,map,'test.gif','DelayTime',0,'LoopCount',inf)
```

Пример #02. Анимированный график в пространстве

```
[x, y] = meshgrid( [0:0.01:1] );
t = linspace( 0, pi, 100);
for k =1:100
z = sin(2*pi*x).*sin(3*pi*y).*sin(2*t(k));
surf(x,y,z);
shading flat;
colormap(copper);
axis( [0 1 0 1 -1 1]);
M(k) = getframe;
end
movie(M,1,30)
```

Задание 2. Считывание координат точек из графического окна

Вызов функции `ginput` приводит к появлению в текущем графическом окне управляемого мышью перекрестия. Щелчок левой клавишей мыши в любой точке графического окна приводит к добавлению ее координат в соответствующие массивы. После того, как указанное в аргументе функции `ginput` число команд будет считано, перекрестие автоматически пропадает.

Если вызвать функцию `ginput` без указания числа точек, то считывание координат будет продолжаться до тех пор, пока вы не нажмете клавишу Enter.

```
clc
clear
ezplot('sin(x)')
hold on
ezplot('cos(x)')
[x ,y] = ginput(3)
```

2

Выполните пример. Сохраните `[x ,y]` в текстовый файл с помощью команды `dlmwrite('data.txt',[x, y])`.

Откройте файл, убедитесь, что данные записаны. Сравните содержимое файла с выводом в Command Window. Данные были записаны по столбцам. Извлеките данные из файла с помощью команд

```
xy=dlmread('data.txt')
xy(1:3, 1) % первый столбец
xy(1:3, 2) % второй столбец
```

Задание 3. Функция area

Пример #01. Сопоставьте данные `Y` и график. Сделайте вывод.

```
Y = [1, 5, 3;
     3, 2, 7;
     1, 5, 3;
     2, 6, 1];
figure
area(Y)
```

Пример #02

```
Y = [1, 5, 3;  
     3, 2, 7;  
     1, 5, 3;  
     2, 6, 1];  
figure  
h = area(Y, 'LineStyle', ':', 'LineWidth', 2);  
set(h(1), 'FaceColor', [0, 0.25, 0.25]);  
set(h(2), 'FaceColor', [0, 0.5, 0.5]);  
set(h(3), 'FaceColor', [0, 0.75, 0.75]);
```

Выполните примеры #01 и #02. На их основе создайте трехцветный флаг из 3 полос. Полосы должны быть одинаковой ширины

Базовые типы данных

Класс Array

Пример #01. Количество размерностей многомерного массива ndims

```
clc  
ndims(ones(1)) % вернет 2  
ndims(ones(2,2)) % вернет 2  
ndims(ones(2,3,4)) % вернет 3  
ndims(ones(2,3,4,5)) % вернет 4
```

3

Пример #02. Вектор размерностей массива size

```
clc  
size(ones(1))  
size(ones(2,2))  
size(ones(2,3,4))  
size(ones(2,3,4,5))
```

Пример #03. Длина большей размерности length

```
clc  
length(ones(7,5,3))  
length(ones(1,3,9))
```

Пример #04. Длина большей размерности length

```
clc  
length(ones(7,5,3))  
length(ones(1,3,9))
```

Пример #05. Количество элементов массива numel

```
clc  
numel(ones(7,5,3)) % вернет 105  
numel(ones(1,3,9)) % вернет 27
```

Пример #06. Команда disp

```
clc  
disp([1 2 3]); % не подавляется ;
```

Класс Numeric

Пример #01. Результат выполнения isnumeric равен 1, если данные это числа

```
clc  
clear  
a=rand(2,3)  
b=isnumeric(a) % вернет 1  
c=islogical(b) % b - это логическая единица  
d=isnumeric(c) % логическая единица это не число
```

Класс Double

- скаляр – тоже массив, минимальный элемент размера (1,1);
- в памяти матрица хранится как вектор по столбцам;
- диапазон вещественных чисел [realmin, realmax].

Специальные переменные

Имя	Результат
ans	Последний результат; если выходная переменная не указана, то MATLAB использует переменную ans.
eps	Точность вычислений с плавающей точкой; определяется длиной мантиссы и для PC eps = 2.220446049250313e-016
realmax	Максимальное число с плавающей точкой, представимое в компьютере; для PC realmax = 1.797693134862316e+308.
realmin	Минимальное число с плавающей точкой, представимое в компьютере; для PC realmin = 2.225073858507202e-308.
pi	Специальная переменная для числа π : pi=3.141592653589793e+000.
i, j	Специальные переменные для обозначения мнимой единицы
inf	Специальная переменная для обозначения символа бесконечности
NaN	Специальная переменная для обозначения неопределенного значения - результата операций типа: 0/0, inf/inf.
computer	Специальная переменная для обозначения типа используемого компьютера; для PC - PCWIN.
version	Специальная переменная для хранения номера используемой версии системы MATLAB.

Способы создания объектов

Пример #01. Задание объектов перечислением

```
clc
clear
a = [1; 2; 3] % вектор-столбец;
a = [1 2 3] % вектор-строка;
a = [1, 2, 3] % вектор-строка
a1=2 ; step_=3 ; end_=12
a=[a1:step_:end_] % вектор с элементами арифметической прогрессии
```

Пример #02. Задание объектов с помощью специальных матриц

```
clc
clear
% матрица 3x4, заполненная случайными числами в диапазоне от 0 до 1.
a=randn(3, 4)
% матрица 5x5, заполненная случайными числами в диапазоне от -50 до 50.
b = randi([-50 50],5)
c=magic(3) % магический квадрат или матрица Дюрера
d=ones(2) % квадратная матрица из единиц
e=eye(4)
nn = [1; 2; 3; 4];
f= diag(nn) % диагональная матрица
```

Пример #03. Задание объектов из имеющихся путем выделения частей

```
clear
clc
A=magic(5) % магический квадрат 5 порядка
B=A(1:2,3) % выделили элементы 1 и 2 из 3-го столбца
C=A(:,3:end) % выделили все элементы в столбцах с 3-го по 5-й
D=A(2:3,3:5) % выделили из 2-й и 3-й строк элементы с 3-го по 5-й
```

5

Пример #04. Выделение треугольных матриц

```
clc
clear
A=magic(5) % магический квадрат 5 порядка
B=triu(A) % верхняя треугольная матрица, полученная из A
C=tril(A) % нижняя треугольная матрица, полученная из A
D=diag(A) % диагональ матрицы A
```

Пример #05. Редактирование блоков матриц

```
clc
clear
A=ones(5) % матрица 5 порядка из единиц
A(3:end, 3:end)=5 % теперь в матрице A девять пятерок
A(1:2, 1:2)=4 % теперь в матрице A четыре четверки
A(1:2, 3:5)=7 % теперь в матрице A шесть семерок
```

Пример #06. Задание объектов с помощью импортирования из файла *.xlsx

```
clc
clear
A=xlsread('D:\mydata.xlsx')
```

Содержимое файла mydata.xlsx на диске D, например, может быть следующее:

	A	B	C
1	1	2	3
2	4	5	6
3	7	8	9

Импортирование можно выполнить с помощью команды меню:
File → ImportData.

Пример #07. Импортирование из файла *.txt

Содержимое файла data.txt

```
SunDay MonDay TuesDay WednesDay ThursDay FriDay SaturDay
95.01 76.21 61.54 40.57 55.79 70.28 81.53
73.11 45.65 79.19 93.55 75.29 69.87 74.68
60.68 41.85 92.18 91.69 81.32 90.38 74.51
48.60 82.14 73.82 41.03 0.99 67.22 93.18
89.13 44.47 57.63 89.36 13.89 19.88 46.60
```

```
clear
clc
% Скрипт импорта из текстового файла
filename = 'D:\data.txt'; % файл должен существовать!
delimiterIn = ' '; % разделитель столбцов
headerlinesIn = 1; % одна строка заголовков
A = importdata(filename,delimiterIn,headerlinesIn);
% disp - вывод текста или массива
for k = [1:7]
    disp(A.colheaders{1, k}) % вывод заголовка
    disp(A.data(:, k)) % вывод данных
    disp(' ') % пропуск строки
end
```

Результат:

```
SunDay
95.0100
73.1100
60.6800
48.6000
89.1300
```

```
Monday
    76.2100
...
Saturday
    81.5300
    74.6800
    74.5100
    93.1800
    46.6000
```

Пример #08. Операции над форматированными файлами

`line = fgetl(fid)` — возвращает строку из файла с идентификатором `fid` с удалением символа конца строки. Если функция `fgetl` обнаруживает конец файла, то она возвращает значение `-1`;

```
% чтение файла и вывод его построчно
clc
clear
fid = fopen('data.txt');

tline = fgetl(fid);
while ischar(tline) % является ли элемент символьным массивом
    disp(tline)
    tline = fgetl(fid);
end

fclose(fid);
```

7

`line = fgets(fid)` — возвращает строку из файла с идентификатором `fid`, не удаляя символ конца строки. Если функция `fgets` обнаруживает конец файла, то она возвращает значение `-1`;

`line = fgets(fid,nchar)` — возвращает не больше чем `nchar` первых символов строки. После признака конца строки или конца файла никакие дополнительные символы не считываются

```
% чтение всего файла и вывод по 3 символа в строке
clc
clear
fid = fopen('data.txt');

tline = fgets(fid,3);
while ischar(tline)
    disp(tline)
    tline = fgets(fid,3);
end

fclose(fid);
```

`count = fprintf(fid,format,A,...)` — форматирует данные, содержащиеся в действительной части матрицы `A`, под контролем строки `format` и записывает их в файл с идентификатором `fid`. Функция `fprintf` возвращает число записанных байтов. Значение идентификатора `fid` — целое число, возвращаемое функцией `fopen`.

```
% форматированная запись в файл
x = 0:.1:1;
A = [x; exp(x)];

fileID = fopen('exp.txt','w');
fprintf(fileID,'%6s %12s\n','x','exp(x)');
fprintf(fileID,'%6.2f %12.8f\n',A);
fclose(fileID);
```

Файл exp.txt

```
      x      exp(x)
0.00    1.00000000
0.10    1.10517092
...
1.00    2.71828183
```

Символы формата	
Символ	Описание
%c	Последовательность символов; параметр ширины поля определяет количество считываемых символов
%d	Десятичное число
%e, %f, %g	Число с плавающей точкой
%i	Целое число со знаком
%o	Восьмеричное число со знаком
%s	Последовательность символов
%u	Десятичное целое число со знаком
%x	Шестнадцатеричное целое число со знаком
[...]	Последовательность символов

Специальные символы в строках формата	
Символ	Описание
\n	Новая строка
\t	Горизонтальная табуляция
\b	Возврат на один символ
\r	Возврат каретки

\f	Новая страница
\\	Обратный слеш
\\" или "	Одиночная кавычка
%%	Процент

Параметры спецификаторов формата		
Символ	Описание	Пример
-	Выравнивание преобразованных аргументов по левому краю	%-5.2d
+	Всегда печатать знак числа (+ или -)	%+5.2d
0	Заполнение нулями вместо пробелов	%05.2d
Цифры	Определяет минимальное число знаков, которые будут напечатаны	%6f
Цифры после точки	Определяет количество символов, печатаемых справа от десятичной точки	%6.2f

$A = \text{fscanf}(\text{fid}, \text{format})$ — читает все данные из файла с идентификатором, fid , преобразует их согласно значению параметра format и возвращает в виде матрицы A . Значение идентификатора fid — целое число, возвращаемое функцией fopen . Параметр format представляет собой строку, определяющую формат данных, которые необходимо прочесть;

```

clc
clear
% создали файл и записали в него 8 строк
x = 100*rand(8,1);
fileID = fopen('nums1.txt', 'w');
fprintf(fileID, '%4.4f\n', x);
fclose(fileID);

% открыли файл
fileID = fopen('nums1.txt', 'r');

% определили формат
formatSpec = '%f';

% считали из файла в A
A = fscanf(fileID, formatSpec)

fclose(fileID);

```

$[A, \text{count}] = \text{fscanf}(\text{fid}, \text{format}, \text{size})$ — считывает количество данных, определенное параметром size , преобразует их в соответствии с параметром format и возвращает вместе с количеством успешно считанных элементов count . Параметр size — это произвольный аргумент, определяющий количество считываемых данных. Допустимы следующие значения:

`n` — чтение `n` элементов в вектор-столбец;

`inf` — чтение элементов до конца файла и помещение их в вектор-столбец, содержащий такое же количество элементов, что и в файле;

`[m,n]` — считывает столько элементов, сколько требуется для заполнения матрицы размера `mхn`. Заполнение происходит по столбцам. Величина `n` (но не `m`!) может принимать значение `Inf`.

```
clc
clear
% создали файл и записали в него 5 строк по 2 столбца
x = 1:1:5;
y = [x;rand(1,5)];
fileID = fopen('nums2.txt','w');
fprintf(fileID,'%d %4.4f\n',y);
fclose(fileID);

% открыли файл на чтение
fileID = fopen('nums2.txt','r');

% определили формат
formatSpec = '%d %f';
% определили размер - 2 столбца и до конца файла
sizeA = [2 Inf];

% считали из файла в A
A = fscanf(fileID,formatSpec,sizeA)

% закрыли файл
fclose(fileID);

% необходимо транспонировать A, так как считывание производится
% по строкам, а данные хранятся по столбцам
A=A'
```

ФУНКЦИЯ	ОПИСАНИЕ
<code>fclose</code>	Закрытие всех или одного файла
<code>feof</code>	Если достигнут конец файла, то истина, иначе - ложь
<code>ferror</code>	Информация об ошибках ввода / вывода файла
<code>fgetl</code>	Читать строку из файла, удаление символов конца строки
<code>fgets</code>	Читать строку из файла, сохраняя символы конца строки
<code>fopen</code>	Открыть файл или получить информацию об открытых файлах
<code>fprintf</code>	Запись данных в текстовый файл
<code>fread</code>	Чтение данных из двоичного файла

frewind	Переместить индикатор позиции файла в начало открытого файла
fscanf	Чтение данных из текстового файла
fseek	Переместить в указанную позицию в файле
ftell	Позиция в открытом файле
fwrite	Запись данных в двоичный файл

Задания

1. Создать магический квадрат 5-го порядка. Сформировать из него матрицы размеров 2x2, 2x3, 3x1, 1x5, выделяя сначала, начиная с элемента (1,1), а потом – начиная с элемента (5,5). Записать полученные матрицы в файл. Считать матрицы из файла.
2. В файл Excel записать квадратную матрицу. Выполнить импортирование из файла в переменную. Вывести на экран матрицу, ее квадрат, и обратную к ней.
3. Записать в файл таблицу, содержащую значения x , $\sin(x)$, $\cos(x)$. Прочитать из файла содержащуюся в нем информацию и вывести на экран. Построить графики по полученным данным.
4. Записать в файл квадратную матрицу, содержащую случайные числа. Прочитать из файла информацию и вывести на экран по одному числу в строке. Найти сумму элементов матрицы.

Некоторые методы линейной алгебры

Пример #01. Функции min, max, sum, prod

```

Clc
clear
A=diag([0 1 2])+ones(3)
prod(prod(A)) % вернет 6
sum(sum(A)) % вернет 12
min(min(A)) % вернет 1
max(max(A)) % вернет 3

B=[1 2; 3 4]
sum(B) % вернет 4 и 6 - сумма по столбцам
prod(B) % вернет 3 и 8 - произведение по столбцам

```

Пример #02. Обратная матрица и определитель матрицы

Если матрица A является квадратной и невырожденной, уравнения $AX = I$ и $XA = I$ имеют одинаковое решение X . Это решение называется матрицей обратной к A , обозначается через A^{-1} и вычисляется при помощи функции `inv`.

```
clc
clear
A= pascal (3)
d = det (A)
X = inv (A)
```

Пример #03. Степень матрицы и поэлементное умножение

```
clc
clear
A= pascal (3)
% степень определена для квадратной матрицы
B = A^2
% поэлементное умножение
C = A.^2
```

Решение систем линейных уравнений

12

Если СЛАУ имеет хотя бы одно решение, ее называют **совместной**, если же решений нет – **несовместной**. Если совместная СЛАУ имеет ровно одно решение, её именуют **определённой**, если бесконечное множество решений – **неопределённой**.

Рассмотрим примеры, когда система определенная

$X = A \setminus B$ - обозначает решение матричного уравнения $AX = B$

$X = B / A$ - обозначает решение матричного уравнения $XA = B$.

Пример #04. Решение СЛАУ при помощи знака обратной косой черты

```
clc
clear
A = [4 1 2;
     3 7 1;
     2 2 8];
b = [7; 11; 12];
x = A \ b

% Вместо знака обратной косой черты можно было вызвать функцию
mldivide
x = mldivide(A, b)

% Проверка или вычисление невязки
b - A*x
```

Пример #05. Решение СЛАУ с помощью `insolve`

```
clc
clear
A = [4 1 2;
     3 7 1;
     2 2 8];
b = [7; 11; 12];
x=insolve(A,b)
```

Пример #06. 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) % по СТРОКАМ по возрастанию
E = sort(A,2, 'descend') % по СТРОКАМ по убыванию
```

Пример #07. Определение индексов ненулевых элементов - `find`

```
clear
clc
A =[3 0 0; 0 7 0; 0 0 256]
[i, j, x] = find(A)
```

Результат

A =			i =		j =		x =
3	0	0	1		1		3
0	7	0	2		2		7
0	0	256	3		3		256

Пример #07. `find` с условием

```
clear
clc
A =[3 0 0; 0 7 0; 0 0 256]
% возвратит номера индексов и логическое значение 1
[i, j, x] = find(A>7); [i,j,x]
```

Результат

```
ans =
     3     3     1
```

Пример #08. 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
clc
A=eye(3)% единичная матрица
lambda = eig(A) % собственные значения
D = eig(A, 'matrix') % собственные значения в виде матрицы
[V,D] = eig(A) % собственные значения и правые собственные
векторы A.
V\A*V % собственные значения различны, а собственные векторы
независимы
```

Задание

1. Найти решение СЛАУ. Выполнить проверку.

№	СЛАУ
1	$\{x_1+2x_2+3x_3=3,$ $3x_1+5x_2+7x_3=0,$ $x_1+3x_2+4x_3=1\}$
2	$\{x_1+x_2+x_3=0,$ $4x_1+2x_2+x_3=1,$ $9x_1+3x_2+4x_3=3\}$
3	$\{x_1+x_2-3x_3=2,$ $3x_1-2x_2+x_3=-1,$ $2x_1+x_2-2x_3=0\}$
4	$\{x_1+2x_2-4x_3=31,$ $5x_1+x_2+2x_3=29,$ $3x_1-x_2+x_3=10\}$

2. Найти собственные векторы и собственные числа. Выполнить проверку.