list = Y;

for i=Y+1:N+Y-1

list = [num2str(list), ', ', num2str(i)];

end

for i=1:5

if (class(i).seats(1, 1) - class(i).seats(1, 2)) > 4

disp(class(i).title)

end

end

case 4 % east

x(istep+1)=x(istep)+1;

y(istep+1)=y(istep);

end

end

%% animate plot of walk

for istep=2:N

plot(x(1:istep), y(1:istep), ...

x(istep), y(istep), 'ro')

axis equal

drawnow

end

ispin=1;

while stash(ispin) > bet

r=rand;

if r < Pwin

% win

stash(ispin+1)=stash(ispin)+bet;

else

% lose

stash(ispin+1)=stash(ispin)-bet;

end

ispin=ispin+1;

% display the results so far

plot([1:ispin], stash,...

ispin, stash(ispin), 'ro')

axis([0, 1.05\*ispin, 0, inf])

xlabel('spin')

ylabel('stash($) ')

line([0, ispin], [stash(1), stash(1)], 'Color', 'r');

drawnow

end

C = length(colors);

M = length(messages);

BeMine = 0;

SpecialHearts = 0;

for i=1:Candies

icolor = randi(C);

imessage = randi(M);

hearts(i).message = messages(imessage);

hearts(i).colors = colors(icolor);

if strcmp(hearts(i).message, 'Be Mine')

BeMine = BeMine + 1;

elseif strcmp(hearts(i).colors, 'red') && strcmp(hearts(i).message, 'You Rock')

SpecialHearts = SpecialHearts + 1;

end

end

%% animate plot of the motion

for it=1:Nstride:Nt

plot3(x(1:it), y(1:it), z(1:it), 'b',... %draws line as it progresses

x(it), y(it), z(it), 'ro'); %updates marker

axis([-r0, r0, -r0, r0, min(z), max(z)]);

axis square

grid on

xlabel('x (cm)')

ylabel('y (cm)')

title(['Exponential decay with \tau = ', num2str(tau)']);

drawnow

end

----------------------------------------------------

t = linspace(0, Tf, Nt);

for it = 1:Nt

x(it) = r\*cos(a\*(2\*pi)/T\*t(it));

y(it) = r\*sin(b\*(2\*pi)/T\*t(it) + phi);

end

for istep=2:N

idirection=2\*pi\*rand(1, 1);

x(istep) = x(istep-1)+step\*cos(idirection); % x position

y(istep) = y(istep-1)+step\*sin(idirection); % y position

end

if doAnimate

for istep=2:N

plot( x(1:istep), y(1:istep), ...

x(istep), y(istep), 'ro')

axisMax=max(abs([x(1:istep), y(1:istep)]))

axis([-axisMax, axisMax, -axisMax, axisMax]);

axis square

grid on

drawnow

end

else

plot( x, y, ...

x(end), y(end), 'ro')

axisMax=max(abs([x(1:istep), y(1:istep)]));

axis([-axisMax, axisMax, -axisMax, axisMax]);

axis square

grid on

end

Linspace – creates equally spaced points

*var* = linspace(initial value, final value, number of points)

--

length(s) – returns the number of characters in the string sName

--

num2str(x) – returns a string corresponding to the number stored in x

--

lower(s) returns the string s in all lowercase

upper(s) returns the string s in all uppercase

str2num(s) returns a number corresponding to the string s

str2double(s) returns a number corresponding to the string s

(also works with cell arrays of strings, defined later)

sName(4) returns the 4th character in the string sName

sName(4:6) returns the 4th through the 6th characters in the string sName

--

Rand/randi – generates prseudo-random number {rng(‘shuffle’) for real random}

Rand > rando number between [0,1] ; randi > random integer

*var* = (interval min, interval max, # of numbers) à 100 random numbers the default

--

max(v) value of the smallest element of array v

min(v) value of the largest element of array v

--

sum(v) sum of the elements of array v

mean(v) average value of the elements of array v

median(v) median of the elements of array v

mode(v) mode of the elements of array v

std(v) standard deviation the elements of array v

sort(v) returns vector sorted in ascending order

--

Disp – display in command window

disp([‘string in quotes’, num2str(var), ‘don’’t forget comma’s after each different element’])

--

Zeros – set all numbers in an array to 0 ; One – set all number in array to 1

ones(n,m), zeroes(n,m) à n x m matrix (rows then columns)

--

Strcmp – string compare: compares two strings. Returns value “true” if two strings are identical.

This is important because strings cannot in general be tested for

equality with the double equals sign.

--

strcmpi([string](http://www.mathworks.com/help/matlab/ref/strcmpi.html?refresh=true" \l "inputarg_string),[string](http://www.mathworks.com/help/matlab/ref/strcmpi.html?refresh=true#inputarg_string)) compares two strings for equality, ignoring any differences in letter case.

--

Logical Expressions

&& and

|| or

~ not

== is equal to

~= is not equal to

> is greater than

< is less than

>= is greater than or equal to

<= is less than or equal to

Plot – plot points

Plot(v1, v2, ‘-o’

--

Drawnow – animate a plot by sequentially plotting new points ; *must be in a for loop*

*--*

y = tan(sin(x)) - sin(tan(x));

plot(x,y,'--rs','LineWidth',2,...

'MarkerEdgeColor','k',...

'MarkerFaceColor','g',...

'MarkerSize',10)

b blue . point - solid

g green o circle : dotted

r red x x-mark -. dashdot

c cyan + plus -- dashed

m magenta \* star (none) no line

y yellow s square

k black d diamond

w white v triangle (down)

^ triangle (up)

< triangle (left)

> triangle (right)

p pentagram

h hexagram

If

if expression

statements

ELSEIF expression

statements

ELSE

statements

END

==, <, >, <=, >=, or ~=.

Example

if I == J

A(I,J) = 2;

elseif abs(I-J) == 1

A(I,J) = -1;

else

A(I,J) = 0;

End

Switch Case

switch lower(method)

case {'linear','bilinear'}

disp('Method is linear')

case 'cubic'

disp('Method is cubic')

case 'nearest'

disp('Method is nearest')

otherwise

disp('Unknown method.')

end

For Loop

for followed by counter: i, j, k, k, n, m

for R = 1:N

for C = 1:N

A(R,C) = 1/(R+C-1);

end

end

While

==, <, >, <=, >=, or ~=

while expression

statements

END

So long as true, run loop

Cell array

{}

Vector

[ ]

matrix

x(Ns,Nw)

firstName=input('Please enter your first name: ','s');

function [x,y] = funcName(x1, y1, x2,)

y = [u,v] concatenates vectors from u to v