

all sections back to back.

Name: _____ ANSWER _____ Section: __03/04__

Practice for Exam1.

Is an A possible?	
yes	no

At the end of the 50minutes, SUBMIT your script under the EXAM link on Canvas. You do not have time to zip so do not zip. Circle whether or not an A is possible so I can grade more efficiently. Your grade will be available tomorrow. If this grade is less than 90, you MUST come on Wednesday to take the real exam.

I do not answer questions during exams. Due to the vocabulary being new to you, you mis-use it and I have to "interpret" your question. I'd rather not. ☹

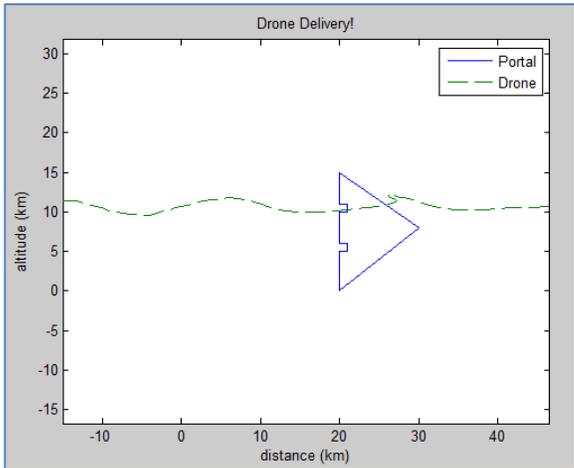
DO AS MUCH AS YOU CAN. DO YOUR BEST. 30PTS IS ALREADY GOOD FOR EXTRA CREDIT ASSIGNMENTS!

Google translate and WolframAlpha is permitted. But do not SEARCH anything related to MATLAB online.

DO NOT EMAIL/SHARE FILE WITH THE NEXT SECTIONS. IT IS CALLED CHEATING.

Problem: It's the year 2070 and drone-delivery is common! The drone should have a pretty straight trajectory, but due to winds, building, airplanes, flying humans, super-heros, etc... the trajectory is "all over the place".

You can only control the starting altitude of the drone. Once it starts, the drone has an automated path. Find a possible starting altitude that would successfully go into a portal ship, as shown on the figure. Once inside, someone will



intercept so do not worry about the rest of the trajectory.

The wind coefficient k_{wind} , the super-hero coefficient $k_{superHeros}$, and the starting altitude y_{start} of the drone are known but subject to change. The trajectory of the drone (y vs. x) is defined using the following equations, where n is a fixed range of values from -12 to 15 radians at all times:

$$x(n) = n \cos(n) + k_{wind} * n$$

$$y(n) = \sin(n) + \frac{n}{k_{superHeros}} + y_{start}$$

May the force be with you! Fill the missing pieces of the table by re-using your code.

k_{wind}	$k_{superHeros}$	y_{start} (km)	Success: no, top, or bottom door?
3.2	20	10.7	top door
3.2	2	10.7	no
-13.2	2	10.7	no
-2	2	10	bottom door
0	-4	4	no

Who knew a math function could be so funky... ☹

This warning message is normal. Let it be!

```
Command Window
Warning: Imaginary parts of complex X and/or Y
arguments ignored
> In DroneDelivery at 23
```

(turn page over)

Complete what is missing of the Engineering process below:

Step1 (5pts):

Givens: kwind, ksuperhero, sometimes y_start

Find: y_start

Step2: see picture page1. The portal's coordinates are below.

Step3: equations given page1

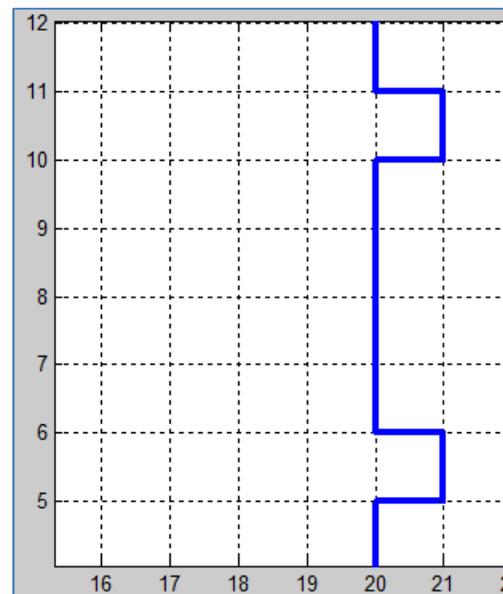
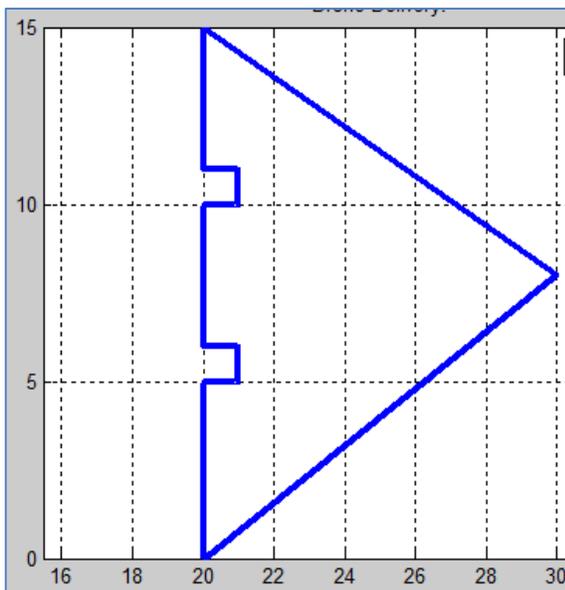
Step4: The delivery is a success if the drone goes through one of the specified door: top or bottom.

Step5: This is only feasible with the help of a computational tool!

Step6: Check with my picture with case 1 from the table.

Step7a/b. in the script file. (As a guide, my code is 29lines top to bottom)

The script is required to have hardcoded the variables that are subject to change. Plot the drone's trajectory and the portal on the same figure. The plot should be properly formatted overall as the labs have done. Use line specifiers to differentiate between portal and drone. No output should show in the command window.



Zoomed in for the doors

Rubric. If you are done, here is a checklist!

Step1	5pts
valid and meaningful file name	5pts
name, section, purpose	3pts
cleanup commands	3pts
algorithms/comments	5pts
spacing of code	5pts
proper variable names	5pts
hardcode variables	5pts

data for trajectory: n, x, y	15pts
data for portal	15pts
correct plot	10pts
plot formatting	5pts
table filled in properly	8pts
simple use of parentheses	3pts
no outputs in cmd wd	2pts
(leeway)	6pts

```
%drone delivery through portal
%by Prof.Liron (sections 3/4)

clc;
clear;
close all;

%define start of travel
kWind = 0;
kSuperHeros=-4;
yStart = 10;

%travel coordinates
myN = linspace(-12,15,50);
xDrone = myN.^cos(myN) + kWind*myN; %ONE DOT
yDrone = sin(myN) + myN/kSuperHeros + yStart; %NO DOTS

%portal x/y coordinates
xPortal = [20 20 21 21 20 20 21 21 20 20 30 20];
yPortal = [0 5 5 6 6 10 10 11 11 15 8 0]; %7.5 accepted too

%plot all and legend and title
plot(xPortal, yPortal, xDrone,yDrone, '--'); %or use method with hold on;
axis equal;
title('Drone Delivery!');
xlabel('distance (km)');
ylabel('altitude (km)');
legend('Portal', 'Drone');

%no output. run to see figure
```

Name: _____ **ANSWER** _____ Section: 08/09

Is an A possible?	yes	no
-------------------	-----	----

Practice for Exam1.

At the end of the 50minutes, SUBMIT your script under the EXAM link on Canvas. You do not have time to zip so do not zip. Circle whether or not an A is possible so I can grade more efficiently. Your grade will be available tomorrow. If this grade is less than 90, you MUST come on Wednesday to take the real exam.

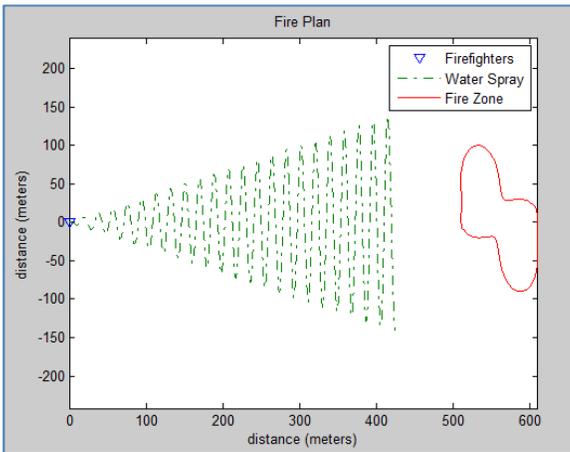
I do not answer questions during exams. Due to the vocabulary being new to you, you mis-use it and I have to “interpret” your question. I’d rather not. ☹

DO AS MUCH AS YOU CAN. DO YOUR BEST. 30PTS IS ALREADY GOOD FOR EXTRA CREDIT ASSIGNMENTS!

Google translate and WolframAlpha is permitted. But do not SEARCH anything related to MATLAB online.

DO NOT EMAIL/SHARE FILE WITH THE NEXT SECTION. IT IS CALLED CHEATING.

Problem: Fires are ravaging the Northwest coast. You manufactured a new water spray system but you need to test some possible settings of the machine. Your goal is to **find the best “Spray setting”** so it reaches a majority of the fire. The fire size and type are also known but subject to change due to winds and weather!



The fire contour (y vs. x) is defined by the following parametric equations:

$$x(r, \phi) = 10 * r * \cos(\phi) * fire_{size} + 560$$

$$y(r, \phi) = 10 * r * \sin(\phi) * fire_{size} + 5$$

These equations have 2 independent variables:

First, ϕ is a range of angles from 0 to 2π .

r is then defined by:

$$r(\phi) = |\cos(fireType * \phi)| \sin(2 * fireType * \phi)$$

Create them in the correct order. Let MATLAB handle it! All will be fine!

The machine’s spray (y vs. x) is much simpler but requires a lot of new angles β this time from 0 to 45π !

$$x(\beta) = \beta * Spray_{setting}$$

$$y(\beta) = \beta * \cos(\beta)$$

Fill the missing pieces of the table by re-using your code. (For simplicity, only test with integers).

Fire Type	Fire Size	Spray Setting	Width of fire contained? (in y direction)	Depth of fire contained? (in x direction)
1	5	3	Not hit!	Not Hit!
1	5	4	Mostly	About 50% only
3	7	5	No	All of it
6	4	6	All of it	All of it

Who knew a math function could be so funky... ☹

(turn page over)

Complete what is missing of the Engineering process below:

Step1 (5pts):

Givens: Fire type, Fire size,

Find: spray setting

Step2: see picture page1.

Step3: equations given page1

Step4: none

Step5: This is only feasible with the help of a computational tool!

Step6: Check with my picture with case 1 from the table.

Step7a/b. in the script file. (As a guide, my code is 32lines top to bottom)

The script is required to have hardcoded the variables that are subject to change. Plot the fire, the firefighter (located at 0,0), and the water spray on the same figure. The plot should be properly formatted overall as the labs have done. Use line specifiers to differentiate between each plot. The firefighter should have a marker. No output should show in the command window.

Rubric. If you are done, here is a checklist!

Step1	5pts
valid and meaningful file name	5pts
name, section, purpose	3pts
cleanup commands	3pts
algorithms/comments	5pts
spacing of code	5pts
proper variable names	5pts
hardcode variables	5pts
data for fire: angles, r , x, y	15pts
data for spray: angles, x, y	15pts
correct plot	10pts
plot formatting	5pts
table filled in properly	8pts
simple use of parentheses	3pts
no outputs in cmd wd	2pts
(leeway)	6pts

```
%fires in California (Sections 8/9)
%by Prof.Liron

clc;
clear;
close all;

%fire/spray characteristics
fireType=6;
fireSize=4;
kSpray = 6;

%data points for fire zone
angFire = linspace(0,2*pi);
rad = abs(cos(fireType*angFire)).^sin(2*fireType*angFire); %ONE DOT
xsFire = 10*rad.*cos(angFire)*fireSize+560;%ONE DOT
ysFire = 10*rad.*sin(angFire)*fireSize+5;%ONE DOT

%water jet
angSpray = linspace(0,45*pi,2000);
xsSpray=angSpray*kSpray; %NO NO DOT
ysSpray=angSpray.*cos(angSpray); %ONE DOT

%plot all on 1 figure
plot(0,0,'v',xsSpray,ysSpray,'-.',xsFire,ysFire); %or use hold on method

%plot formatting
axis equal;
title('Fire Plan');
xlabel('distance (meters)');
ylabel('distance (meters)');
legend('Firefighters','Water Spray','Fire Zone');

%no output. run to see figure
```