//AUTHOR: DANIEL CRISP

//DATE: SATURDAY, NOVEMBER 26th, 2016

//DESC: GRAPHICS

#QUESTION 7: Meshgrid Command

MATLAB CODE:

function [ X , Y ] = meshcmd( LowPoint , HighPoint , Points )

n = Points; %number of points

%F = Function;

low = LowPoint; %[1 10]

high = HighPoint; %[6 15]

length = high - low;

x = linspace(low(1),high(1),n);

y = linspace(low(2),high(2),n);

[X,Y] = meshgrid(x,y);

%F = input('\nEither enter a Z(n,n) matrix for use in X-Y surface plot.\n Or enter "0" to continue without plotting.\n');

%F = X.\*exp(-X.^2-Y.^2); %Example

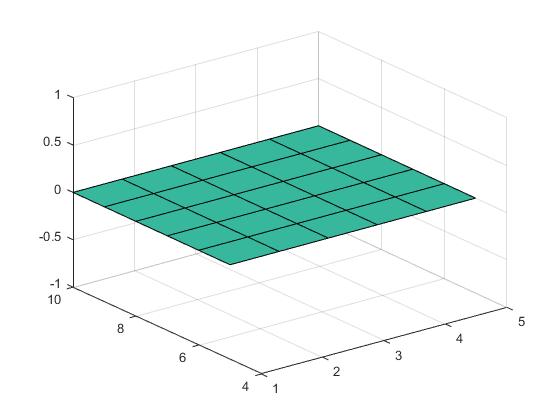
F = zeros(n,n); %Example of (n x n) matrix of zeros

surf(X,Y,F);

end

MATLAB OUTPUT:

>>[S] = meshcmd([1 10],[6 15], 6 );



#QUESTION 8: Polygon Rotation

Rotate a polygon ABCD a quarter, half, full and 5 cycles

around the origin if A( 0,0), B(4, 1), C(2, -4), D(-1, -3).

MATLAB CODE:

function rotate( Origin, Points, Degrees )

%Rotate a given set of points entered as [p1;p2;p3] about the origin

%using a linspace in degrees.

%Convert vargins to more useful names

o = Origin; %[0 0];

pts = Points;

deg = Degrees;

%Set pts about origin, get magnitudes, close line by appending first

%point to end of pts

[lim,pts] = relate\_inputs(o,pts);

%Draw Lines

plotline(pts);

axis([-lim+o(1) lim+o(1) -lim+o(2) lim+o(2)]);

Npts = length(pts(:,1));

for theta = deg

R = [cos(theta) -sin(theta);sin(theta) cos(theta)]; %2D Rotation matrix

pts\_new=R\*pts(1:Npts,:)'; %Rotate points

plotline(pts\_new'); %Send pts\_new to plotline as [pt1;pt2..]

end

end

function plotline(pts)

dim = length(pts(1,:));

x = pts(:,1);

switch dim

case 2

y = pts(:,2);

line('XData',x,'YData',y);

case 3 %IGNORE: havn't yet included 3D rotation matrix

z = pts(:,2);

line('XData',x,'YData',y,'ZData',z);

end

end

function [lim,pts] = relate\_inputs(o, pts)

Npts = length(pts(:,1));

r(1:Npts) = 0;

for i = 1:Npts

pts(i,:) = pts(i,:)-o(1,:); %Shift points for rotation about origin

r(i) = sqrt(sum(pts(i,:).^2));

end

pts(Npts+1,:)=pts(1,:); %Close image by appending pts with duplicate of first.

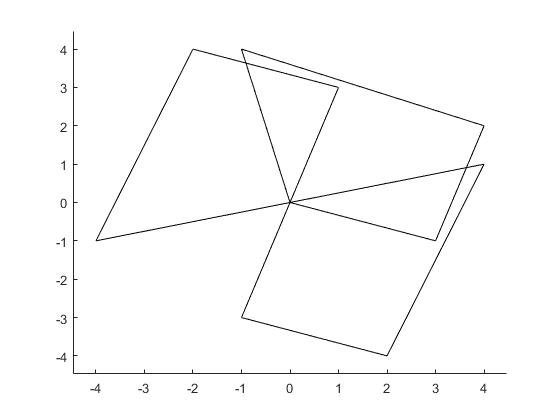
lim = max(r);

end

MATLAB OUTPUT:

>>polyPoints = [0 0; 4 1; 2 -4; -1 -3]; full = 2\*pi;

>>rotate([0 0],polypts,[0 0.25\*full 0.5\*full full 5\*full])



#QUESTION 9: Quiver Function

MATLAB CODE: Unsure of what V=Rmag\*sin(phi) meant, or v=[-3,3], I chose to show Force=V x B.

% vectorField.m

% Author: Daniel Crisp

%Clear, Clean

clear; clc;

script = 'vectorField.m';

%Set Parameters

B = [0, 0, -.3]; %B-Field into the page (-Zhat)

Bmag = sqrt(sum(B.^2));

%Define grid

Nx = 10;

x = 0:1:Nx;

y=x-(0.5\*Nx); %Shift center of range from x to 0.

%[X, Y, Z] = meshgrid(x,y,y);

[X, Y] = meshgrid(x,y);

%Compute Vector Field

Vx = X; Vy = Y;

Vmag=sqrt(Vx.\*Vx+Vy.\*Vy); %distance from origin

Fx = Vy\*Bmag;

Fy = -Vx\*Bmag;

Fmag = Vmag\*Bmag; %Assuming angle = 90;

%Plot Vector Field

quiver(Vx, Vy, Fx, Fy, 'r');

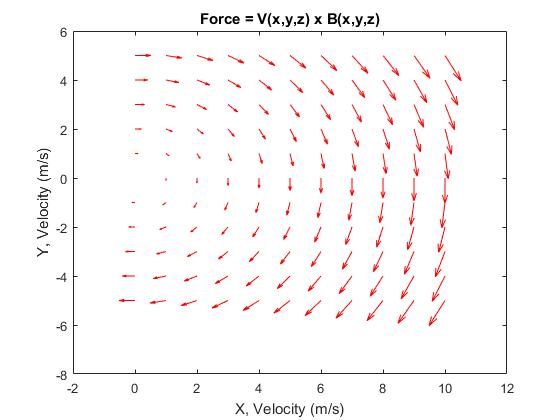
title('Force = V(x,y,z) x B(x,y,z)')

xlabel('X, Velocity (m/s)')

ylabel('Y, Velocity (m/s)')

MATLAB OUTPUT:

run('vectorField.m')



#QUESTION 10: Dumbell Rotation

Using dumbbell program, rotate three spheres connected as

triangle. Make any necessary assumption.

% rotateDumbbell

% Make hgtransform object and rotate it

% around object center which is displaced

% from the origin.

%

%% create two spheres connected by a line

[X,Y,Z]=sphere(12);

hs1=surf(X,Y,Z,'FaceColor',[0.9 0.9 0.9]);

hold on

hs2=surf(X+3,Y,Z,'FaceColor',[0.6 0.6 0.6]);

hs3=surf(X+1.5,Y+2.5981,Z,'FaceColor',[0.3 0.3 0.3]);

hold off

axis equal

hL1=line([1 2],[0 0],[0 0],'LineWidth',7);

A = cos(pi/3); B = sin(pi/3);

hL2=line([1\*A 2\*A],[1\*B 2\*B],[0,0],'LineWidth',7);

hL3=line([1\*A+1.5 2\*A+1.5],[2\*B 1\*B],[0,0],'LineWidth',7);

axis([-2 4 -2 4 -2 3]);

xlabel('x');ylabel('y');zlabel('z')

%% aggregate into Dumbbell object

hDumbbell=hgtransform('Parent',gca);

set(hs1,'Parent',hDumbbell);

set(hs2,'Parent',hDumbbell);

set(hs3,'Parent',hDumbbell);

set(hL1,'Parent',hDumbbell);

set(hL2,'Parent',hDumbbell);

set(hL3,'Parent',hDumbbell);

drawnow

%% rotate around center of dumbbell

% at r=(1.5, 0, 0)

Delta=[1.5, 0, 0];

Nth=60;

theta=linspace(0,0.2\*10\*pi,Nth);

v=1

for ith=1:Nth

Mt1=makehgtform('translate',-Delta);

M=makehgtform('yrotate',theta(ith));

Mt2=makehgtform('scale',[v, v, v]);

set(hDumbbell,'Matrix',Mt2\*M\*Mt1);

drawnow

pause(0.5);

% Mt2=makehgtform('scale',[v+0.01,v+0.01,v+0.01]);

%v=v+.01;

end

MATLAB OUTPUT:

