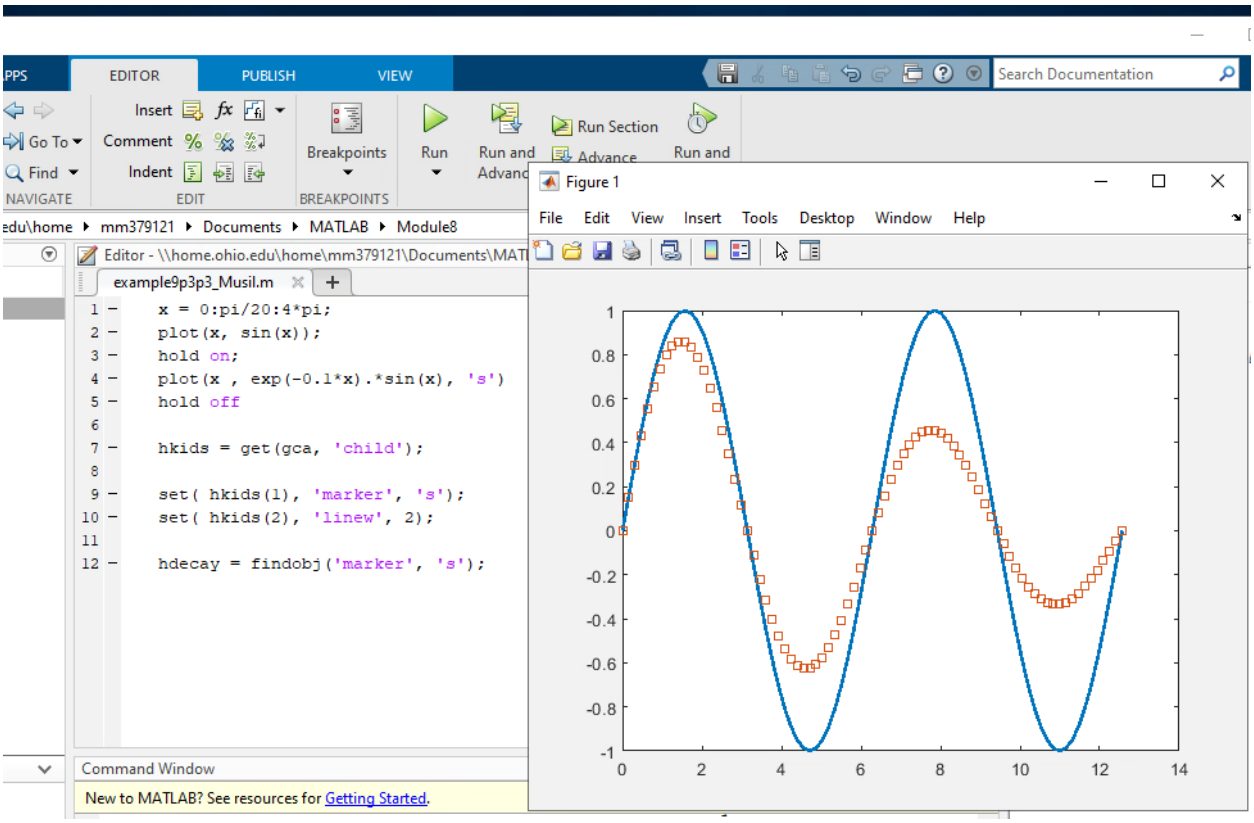
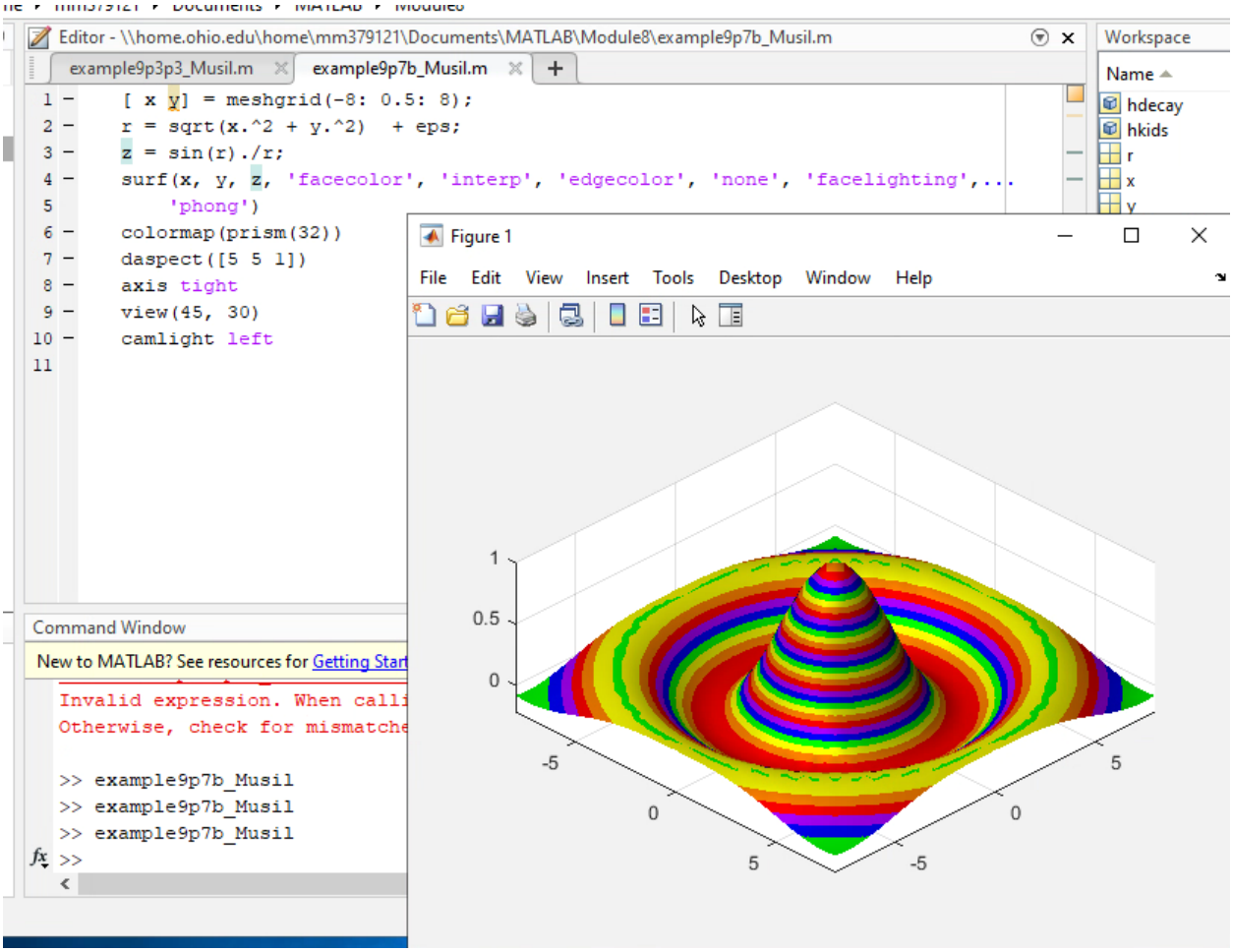
1. Create an m-file, named “example9p3p3_LastName.m”, that has all the MATLAB® statements listed in section 9.3.3 of the textbook (page 218). However, modify it so that the markers of the decaying function are squares instead of circles and set the line width of the other sinusoid to 2 instead of 4.



2. Create an m-file, named “example9p7b_LastName.m”, that has all the MATLAB® statements listed in at the bottom of page 229 and the top of page 230 in section 9.7 of the textbook. However, modify it by implementing the “prism” colormap (which, incidentally, should achieve a Dr. Suess look!) and set the position of the light to azimuth = 45 deg and elevation = 30 deg.



3. Start with the template file langtonsant_template.m and create the basic Langton's Ant animation.

langtons_ant.m

```
% start
clear all
close all

% initialize grid to all zeros (i.e., white)
gridcolor = zeros(100,100);

% specify the bounds of the grid
x_min = 1;  x_max = 100;
y_min = 1;  y_max = 100;

% initialize ant position to the middle of the grid
ant_x = 50;
ant_y = 50;

% initialize the ant direction to east
antdirection = 0;

% create initial plot with handle 'p', specifying the x and y values as
% ant_x and ant_y, set the marker type as a square and set erasemode to
% none so that all points are shown
p = plot(ant_x, ant_y, 'marker', 's');
hold on
p.EraseMode = 'xor';

% Set plot axis to span the grid
axis([0 100 0 100])

% specify the loop variable to indicate if the ant is still inside the grid
inside = 1;

% count the number of steps needed to reach the edge of the grid
total_steps = 0;

while inside == 1
    total_steps = total_steps + 1;

    if gridcolor(ant_x,ant_y) == 0  % if the square is white, then
        antdirection = antdirection + 90;  % turn 90 deg to the right
    else   % otherwise the square is black, so
        antdirection = antdirection - 90;  % turn 90 deg to the left
    end

    % check antdirection and modify if necessary to keep it in the range of
    % 0 to 360 degrees
    if antdirection >= 360
        antdirection = antdirection - 360;
    elseif antdirection < 0
        antdirection = antdirection + 360;
    end

    % the ant always flips the color of the square that it is on
    gridcolor(ant_x,ant_y) = ~gridcolor(ant_x,ant_y);

    if gridcolor(ant_x,ant_y) == 0   % if the grid square is white, then
        % set the Marker edge and face color to white for the x,y point
        set(p,'MarkerEdgeColor', 'w', 'MarkerFaceColor', 'w');
    else  % the grid square is black, so
        % set the Marker edge and face color to black for the x,y point
        set(p,'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'k');
    end
    p = plot(ant_x, ant_y, 'marker', 's');
    p.EraseMode = 'xor';    % execute the graphics set immediately above

    % determine the x and y direction of the ant motion (i.e., the position
    % change for each coordinate)
    ant_motion_x = cosd(antdirection);
    ant_motion_y = sind(antdirection);

    % move the ant to the next square
    ant_x = ant_x + ant_motion_x;
    ant_y = ant_y + ant_motion_y;

    % determine if the ant has move outside the grid; if so, set inside to
    % 0 so that the loop terminates
    if (ant_x<x_min) || (ant_x>x_max)
        inside = 0;
    elseif (ant_y<y_min) || (ant_y>y_max)
        inside = 0;
    end

end
hold off
```

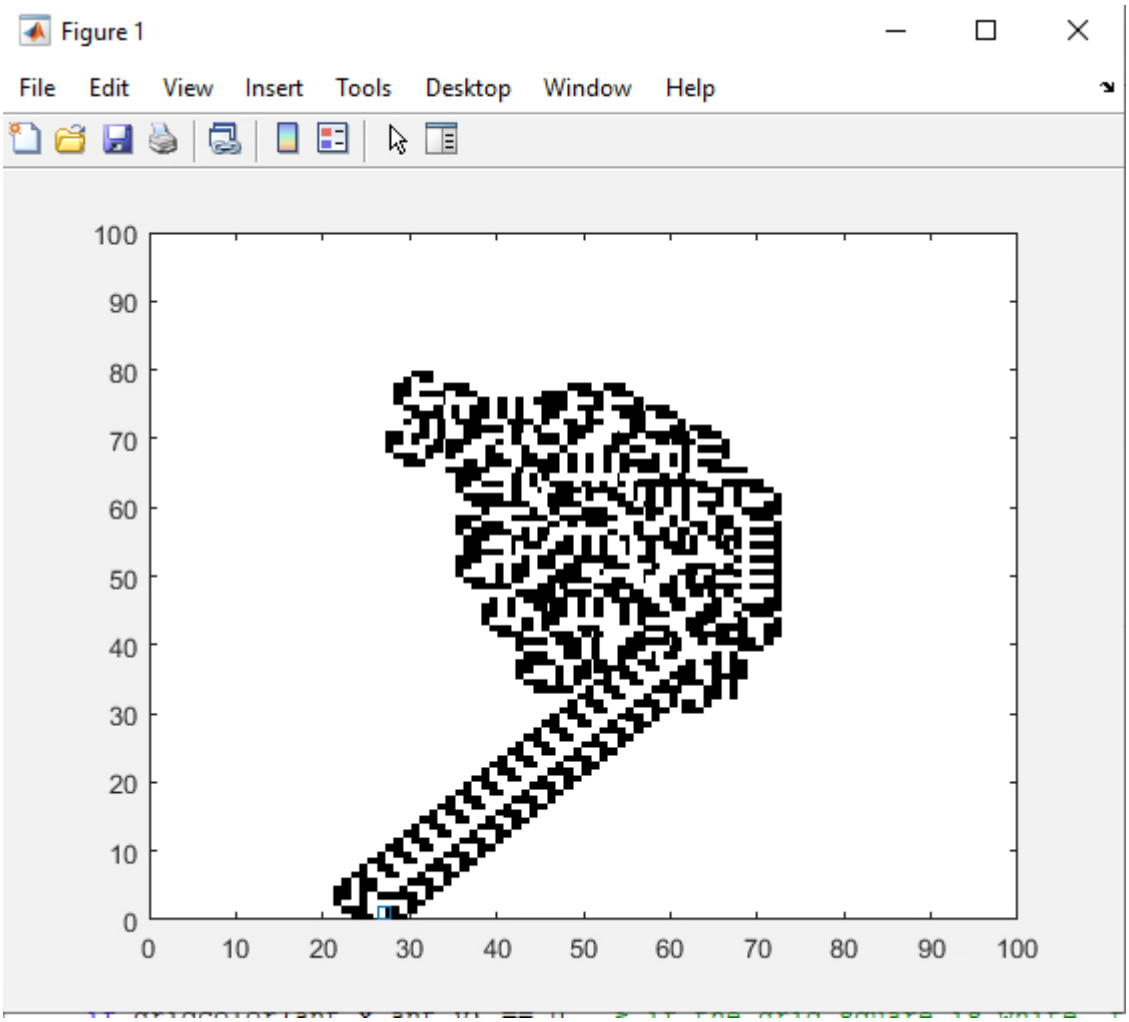


Figure 1
File Edit View Insert Tools Desktop Window Help
100
90
80
70
60
50
40
30
20
10
0
0 10 20 30 40 50 60 70 80 90 100