Solutions to Midterm Exam

Name: ______________________ SS Number: ______________

Instructions: Answer all questions. Partial credit is considered if you state what you would do with an intermediate result if you were able to derive it.

1. (32)
   (a) Write a Matlab function that maps a year to a number expressing the world’s population if population grows at the rate of 1.3% per year and the population is known to be 6,500,000,000 in 2006.

   \[
   \text{function } \text{pop=prob1a(year);} \text{;}
   \]
   \[
   \text{pop} = 6500000000*(1.013)^{\text{(year-2006)}};
   \]
   \[
   \text{end}
   \]

   (b) Write a Matlab function to display a table showing when the world’s population was one eighth of, one quarter of, one half of, and will be twice, four times, eight times what it is in 2006. The left column is the year and the right column is the multiplier (1/8, 1/4, 1/2, etc.).

   \[
   \% \text{Let } B=6500000000 \text{ and let } P \text{ be population.}
   \% \text{Then } P = B*(1.013)^{\text{(year-2006)}};
   \% \text{We want to know at what years } P=B*2^r \text{ for } r=-3,-2,-1,0,1,2,3.
   \% \text{That is, when } B*2^r = B*(1.013)^{\text{(year-2006)}}
   \% \text{Dividing both sides by } B \text{ gives } 2^r = (1.013)^{\text{(year-2006)}}
   \% \text{Taking logs, any base, gives } r*\log(2) = (\text{year-2006})*\log(1.013)
   \% \text{Rearranging gives } \text{year-2006} = r*\log(2)/\log(1.013)
   \% \text{After adding 2006 to both sides, } \text{year}=r*\log(2)/\log(1.013)+2006
   \]
   \[
   \text{function prob1b();}
   \]
   \[
   r = [3 2 1 0 -1 -2 -3];
   \]
   \[
   \text{format rat; } \% \text{rat for pretty print}
   \]
   \[
   \text{year} = \text{floor}(r*\log(2)/\log(1.013)+2006); \% \text{floor for pretty print}
   \]
   \[
   \text{disp(’ year \ multiplier’);}
   \]
   \[
   \text{disp(’ ---- \ ---------’);}
   \]
   \[
   \text{disp([year’ 2.^r’]);} \% \text{transpose for columns}
   \]
   \[
   \text{end}
   \]
2. (33) Recall, in Lab 2, we wrote code to input radii \( R_1, R_2, R_3, R_4 \) of four circles, find centers of those circles so that the \( R_1 \) and \( R_3 \) circles have centers on the \( x \) axis and are tangent at the origin, the \( R_2 \) circle is tangent to the \( R_1 \) and \( R_3 \) circles and above them and the \( R_4 \) circle is tangent to the \( R_1 \) and \( R_3 \) circles and below them. The following code snippet shows how to compute the center of the \( R_3 \) circle.

\[
\begin{align*}
R_1 &= \text{input('Enter radius of circle 1: ')}; \\
R_2 &= \text{input('Enter radius of circle 2: ')}; \\
R_3 &= \text{input('Enter radius of circle 3: ')}; \\
R_4 &= \text{input('Enter radius of circle 4: ')}; \\
\end{align*}
\]

% Find center of circle 2 (tangent to and above circles 1 and 3)
\[
\begin{align*}
g_2 &= ((R2+R3)^2+(R1+R3)^2-(R1+R2)^2)/(2*(R1+R3));  &\quad \text{temp variable} \\
cx2 &= R3-g2;  &\quad \text{x coord, circle 2 center} \\
cy2 &= (R2+R3)*\sin(\arccos(g2/(R2+R3)));  &\quad \text{y coord, circle 2 center}
\end{align*}
\]

Show what additional code is needed to find the centers of the \( R_1, R_3 \), and \( R_4 \) circles and to plot all the circles.

This is one possible answer:

\[
\begin{align*}
\ldots & \quad \text{Input lines as above} \\
\text{hold on;} & \quad \text{do not erase circles} \\
n &= 50; & \quad \text{50 point approximation} \\
\text{angle} &= 0:2*\pi/n:2*\pi; & \quad \text{vector of angles} \\
\ldots & \quad \text{Circle 2 center as above} \\
\end{align*}
\]

% Find center of circle 4 (tangent to and below circles 1 and 3)
\[
\begin{align*}
g_4 &= ((R4+R3)^2+(R1+R3)^2-(R1+R4)^2)/(2*(R1+R3));  &\quad \text{temp variable} \\
cx4 &= R3-g4;  &\quad \text{x coord, circle 4 center} \\
cy4 &= -(R4+R3)*\sin(\arccos(g4/(R4+R3)));  &\quad \text{y coord, circle 4 center}
\end{align*}
\]

% Find circle points and draw the circles
\[
\begin{align*}
x_1 &= -R1+R1*\cos(\text{angle});  &\quad \text{Circle 1 points} \\
x_2 &= cx2+R2*\cos(\text{angle});  &\quad \text{Circle 2 points} \\
x_3 &= R3+R3*\cos(\text{angle});  &\quad \text{Circle 3 points} \\
x_4 &= cx4+R4*\cos(\text{angle});  &\quad \text{Circle 4 points} \\
\text{plot}(x1,y1,x2,y2,x3,y3,x4,y4); &\quad \text{Draw all circles}
\end{align*}
\]

axis equal;
(a) The EPA frequently needs to determine the movement of clouds of pollution in aquifers. Unfortunately, since aquifers are under the ground they cannot take sufficiently many measurements to get the complete story. So, they have to stick probes into the ground. But the cost of sticking a probe is extremely high so they can only stick a few. How can the EPA estimate the shape of the cloud from just a few measurements?

The EPA needs to interpolate the probe data over a volume. One example of a reasonable interpolation method is inverse-distance squared.

(b) What tools does Matlab have for helping with this?

3D matrices, visualization tools such as isosurface. Functions meshgrid and zeros are handy for initializing the matrices. Control constructs such as for and if are needed to compute values at matrix points. Anonymous functions are very handy for passing user-defined interpolators into a visualization function. The functions fopen and fscanf are handy for reading data from a file.

(c) What do those tools require?

A gridded volume and an interpolation algorithm. That is a piece of code that finds a value for each 3D matrix (grid) point from all the given data points.

(d) How can a user satisfy this requirement? (ok, but not necessary, to write some code)

```matlab
function v = inv_dist_squ_interp(x,y,z,dpts);
    [npts ncols] = size(dpts);
    v = zeros(length(x), length(y), length(z));
    % Compute v(i,j,k) for all i,j,k
    for i=1:length(x) % x coordinate
        for j=1:length(y) % y coordinate
            for k=1:length(z) % z coordinate
                % Compute weights of all data points
                w = zeros(1,npts);
                denom = 0.0;
                for l=1:npts
                    ds = (x(i)-dpts(l,1))^2+(y(j)-dpts(l,2))^2+(z(k)-dpts(l,3))^2;
                    w(l) = 1.0/(eps+ds);
                end
                % find the interpolated value at the i,j,k the gridded point
                denom = sum(w);
                v(i,j,k) = sum(dpts(:,4)'.*w/denom);
                end
            end
        end
    end
end
```
(e) Outline the design of a Matlab program to solve (or come close to solving) the EPA’s problem. Each line of your outline should express an important, needed step toward getting a solution.

noindent

i. Settle on a reasonable format for a data file.

ii. Determine variables - name the 2D matrix containing the data, determine its size (which depends on the data), and initialize it. Initialize \(x\), \(y\), and \(z\) vectors corresponding to the chosen grid dimensions and coordinates. Name and initialize the 3D matrix holding grid values.

iii. Read data from file into the 2D matrix.

iv. Interpolate from all 2D points to each 3D point.

v. Visualize the 3D data.