Assignment 9 ENGR 220

Table of Contents

Ethan Vosburg

Problem 1

Write a MATLAB program to help a user decide how to handle a medical issue. The user should enter his/her temperature. If the temperature is between 96°F and 99°F, the output should tell the user to stay home and rest. If the temperature is over 99°F and less than 103°F, the output should tell the user to take aspirin and rest; if the temperature is greater than or equal to 103°F, or less than 96°F the user should get to the hospital immediately.

```
if ~exist('usrInput', 'var')
    usrInput = input('Enter your temperature in degrees Fahrenheit: ');
end

usrInput

if (usrInput >= 96 && usrInput <= 99)
    disp('Stay home and rest.')
elseif (usrInput > 99 && usrInput < 103)
    disp('Take aspirin and rest.')
elseif (usrInput >= 103 || usrInput < 96)
    disp('Get to the hospital immediately.')
else
    disp('Invalid input.')
end
```

```
usrInput =

    34

Get to the hospital immediately.
```

Problem 2

First National Bank pays 5.5 percent annual interest (5.5% APR) on savings and Second National Bank pays 4.5 percent annual interest (4.5% APR); interest is compounded monthly. Your plan is to start with $30 and to deposit $30 at the end of each month. Use a while loop to determine the time it will take (in years) to accumulate $1,000,000 in an account in Second National Bank compared to First National Bank. Report how long it takes at each bank in years.

```
while (true)
    firstBank = 30;
    secondBank = 30;
    firstBankTime = 0;
    secondBankTime = 0;
    while (firstBank < 1000000)
        firstBank = firstBank + 30;
        firstBank = firstBank * 1.0055;
        firstBankTime = firstBankTime + 1;
    end
    while (secondBank < 1000000)
        secondBank = secondBank + 30;
        secondBank = secondBank * 1.0045;
        secondBankTime = secondBankTime + 1;
    end
    if (firstBankTime < secondBankTime)
        fprintf('It will take %d years to reach $1,000,000 in First National
 Bank.\n', firstBankTime);
        fprintf('It will take %d years to reach $1,000,000 in Second National
 Bank.\n', secondBankTime);
        break;
    end
end
```

```
It will take 950 years to reach $1,000,000 in First National Bank.
It will take 1116 years to reach $1,000,000 in Second National Bank.
```

Problem 3

The electricity accounts of residents in a very small town are calculated as follows:

If 500 units or fewer are used, the cost is 2 cents per unit. If more than 500 but not more than 1000 units are used, the cost is $10 for the first 500 units and 5 cents for every unit in excess of 500. If more than 1000 units are used, the cost is $35 for the first 1000 units plus 10 cents for every unit in excess of 1000. A basic service fee of $5 is charged no matter how much electricity is used. Write a program that enters the following five consumptions into a vector and uses a “for loop” to calculate and display the total charge for each one:

Consumption levels: 200, 500, 700, 1000, 1500 units.

```
consumption = [200, 500, 700, 1000, 1500];
for i = 1:length(consumption)
    if (consumption(i) <= 500)
        charge = consumption(i) * 0.02;
    elseif (consumption(i) > 500 && consumption(i) <= 1000)
        charge = 10 + ((consumption(i) - 500) * 0.05);
```

```
    elseif (consumption(i) > 1000)
        charge = 35 + ((consumption(i) - 1000) * 0.1);
    end
    charge = charge + 5;
    fprintf('The charge for %d units is $%.2f\n', consumption(i), charge);
end
```

```
The charge for 200 units is $9.00
The charge for 500 units is $15.00
The charge for 700 units is $25.00
The charge for 1000 units is $40.00
The charge for 1500 units is $90.00
```

Probelm 4

Polycrystalline metals are generally composed of relatively homogeneous grains. The ASTM grain size measurement is determined by:

$$N = 2^{(n-1)}$$

where the ASTM grain size (n) is determined by counting the number of grains (N) in a one-square-inch sample observed at 100X magnification.

Probelm 4a

Write a MATLAB function called NumGrains to find the number of grains in a one-square-inch sample at 100X magnification when the ASTM grain size is known. (So the function should find N when n is known). Use the function to find the number of grains for n = 1 to n = 20. Make the function a local function in your script file so your instructor can run it after you upload it.

```
problem4aData = NumGrains(1:20);
```

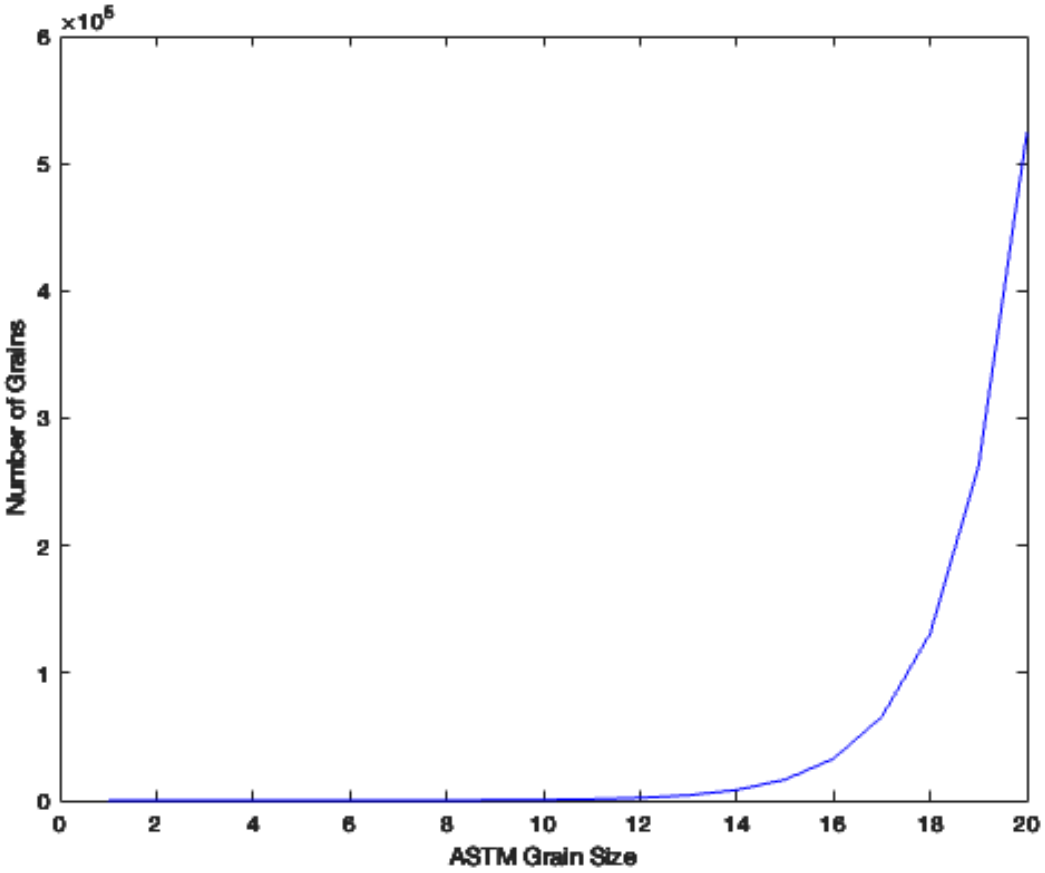
Probelm 4b

Create a fully labeled plot of your results.

```
figure(40);
problem4bPlot = axes;

plot(1:20, problem4aData, 'b-', 'Parent', problem4bPlot);

ylabel(problem4bPlot, 'Number of Grains');
xlabel(problem4bPlot, 'ASTM Grain Size');
```



Problem 5

A rocket is launched vertically. At time t = 0, the rocket's engine shuts down. At that time, the rocket has reached an altitude of 500 meters and is rising at a velocity of 125 meters/second. Gravity then takes over. The height of the rocket as a function of time is given by the kinematic equation:

$$h(t) = -\frac{9.8}{2}t^2 + 125t + 50$$

Problem 5a

Create a local function called Height that accepts time as an input and returns height of the rocket. Use your function in your solutions to parts b and c of this problem.

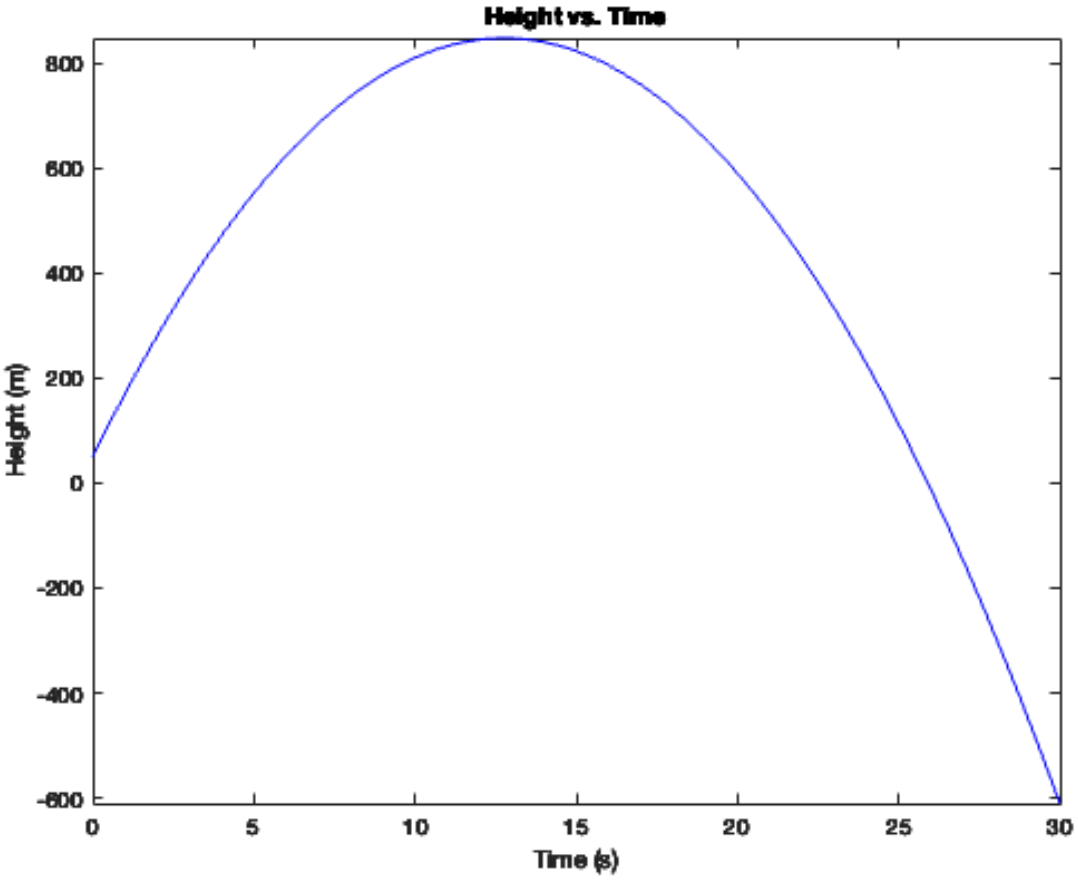
```
% Function at bottom
```

Problem 5b

Plot height vs. time for $0 \leq t \leq 30$ seconds. Use an increment of 0.5 seconds when creating your time vector.

```
figure(50);
problem5bPlot = axes;
```

```
fplot(@(t) Height(t), [0, 30], 'b-', 'Parent', problem5bPlot);
ylabel(problem5bPlot, 'Height (m)');
xlabel(problem5bPlot, 'Time (s)');
title(problem5bPlot, 'Height vs. Time');
```



Problem 5c

Find the time when the rocket starts to fall back to the ground. Use the max() function to find the exact value. Check to make sure your answer is consistent with your graph.

```
heightArray = [[0:0.5:30]' Height([0:0.5:30])'];
[value index] = max(heightArray);
maxHeight = heightArray(index(2), 1);

fprintf('\nThe rocket starts to fall back to the ground at %.2f seconds.\n',
 maxHeight);
```

The rocket starts to fall back to the ground at 13.00 seconds.

Functions

```
function N = NumGrains(n)
```

```
    N = 2.^(n-1);
end

function height = Height(t)
    height = -((9.8 / 2) * t.^2) + (125 * t) + 50;
end
```

Published with MATLAB® R2022b