
Table of Contents

.....	1
Problem 1:	1
Problem 2:	3
Problem 3	4
Problem 4	6

% Academic License

% Engineering 006 Homework

%

% These programs, solves the 4 problems in homework 2 prompt.

%

% Eng 6, Spring 2016

% Name: <Chengeng Xiao>

% Student ID: <913186040>

% Lab Section: <A07>

Problem 1:

Problem 1.1

```
a1 = 1:1:50;
lambda1 = 1550 / 1000; % ? in ?m
nCore1 = 1.5; % record nCore1
nCladding1 = 1.45; % record nCladding1
V1 = 2 * pi * a1 / lambda1 * sqrt(nCore1 ^ 2 - nCladding1 ^ 2); %
    Calculate V
figure(1); % Initialize figure 1
plot(a1, V1); % Draw plot
hold on; % hold the figure
title('Problem 1.1): Plot of core radius a to frequency V'); % set
    title
legend('V = 2 * pi * a / lambda * sqrt(nCore ^ 2 - nCladding ^ 2)'); %
    set legend
xlabel('a (core radius)'); % set x-axis
ylabel('V'); % set y-axis
```

% Problem 1.2

```
a2 = 2.405 / 2 / pi * lambda1 / sqrt(nCore1 ^ 2 - nCladding1 ^ 2); %
    Calculate alpha2
fprintf('Problem 1.2):\n'); % Print with format
fprintf('The permissible radius design parameter for a single mode to
    propagate through such a fiber is %f.\n', a2); % Print with format
```

% Problem 1.3

```
beta2 = - 21.7 / 10 ^ 12; % Change ?2 into ps^2 / nm
c1 = 3 * 10 ^ 5; % Speed of wave
```

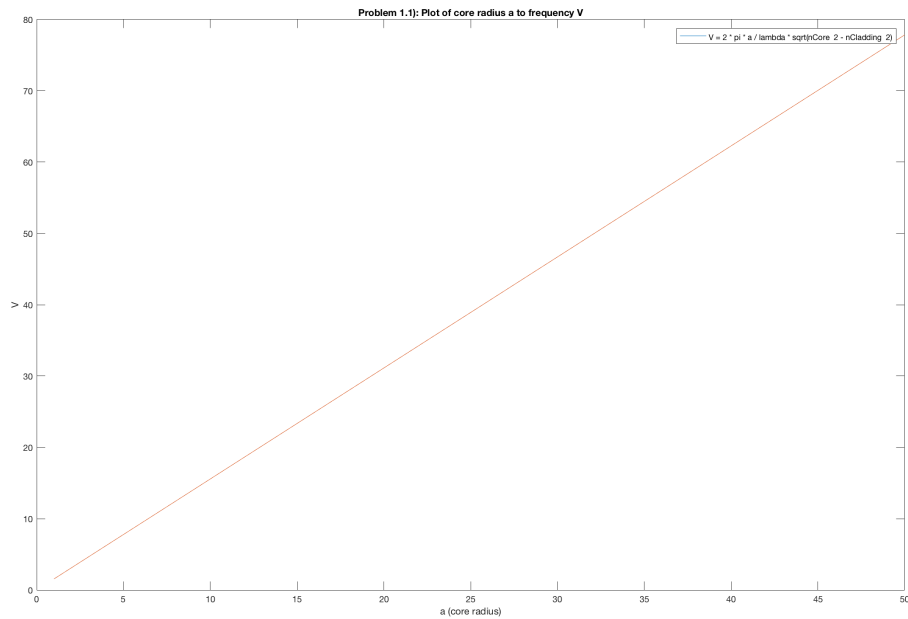
```

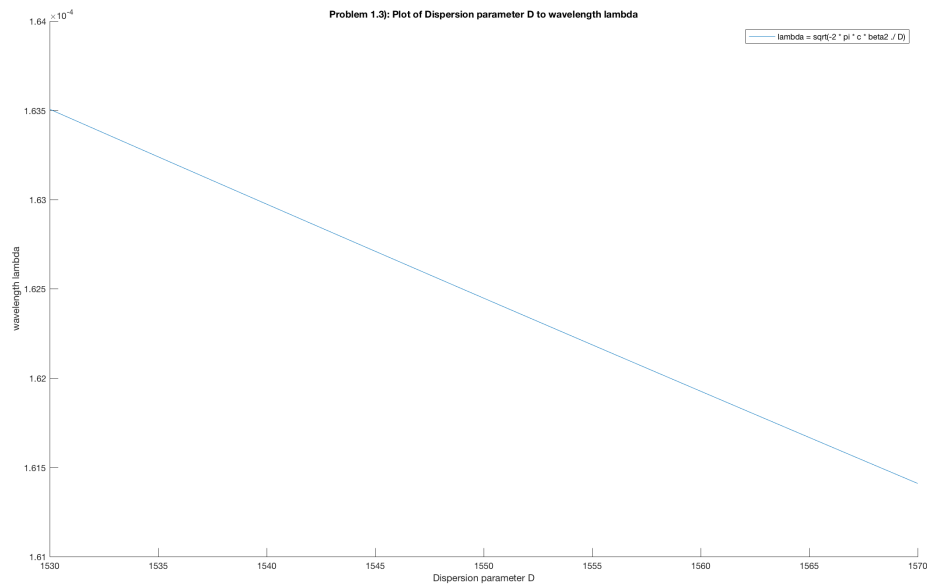
D1 = 1530:1:1570; % Dispersion parameter
lambda2 = sqrt(-2 * pi * c1 * beta2 ./ D1); % Calculate ?
figure(2) % Initialize figure 2
hold on; % hold the figure
plot(D1, lambda2); % Draw plot
title('Problem 1.3): Plot of Dispersion parameter D to wavelength
      lambda'); % set title
legend('lambda = sqrt(-2 * pi * c * beta2 ./ D)'); % set legend
xlabel('Dispersion parameter D'); % set x-axis
ylabel('wavelength lambda'); % set y-axis

```

Problem 1.2):

The permissible radius design parameter for a single mode to propagate through such a fiber is 1.544795.





Problem 2:

Problem 2.1

```
x1 = 0:0.1:2 * pi; % set list x1
y1 = 4 / pi * sin(pi * x1); % Calculate y1
sum = y1; % add y1 to sum
figure(3) % Initialize figure 3
hold on; % hold the figure
plot(x1, sum); % Draw plot
title ('Problem 2: sine x to y plots'); % set title
legend ('y1 = 4 / pi * sin(pi * x)'); % set legend
xlabel ('x'); % set x-axis
ylabel ('y'); % set x-axis
ax1 = subplot(1,1,1); % set subplot
hold(ax1,'on'); % hold the subplot
```

% Problem 2.2

```
x2 = 0:0.1:2 * pi; % set list x2
y2 = 4 / 3 / pi * sin(3 * pi * x2); % Calculate y2
sum = sum + y2; % add y2 to sum
plot(ax1, x2, sum); % Draw subplot
legend ({'y1 = 4 / pi * sin(pi * x)'; 'n=3'}); % set legend
hold(ax1,'on'); % hold subplot
```

% Problem 2.3

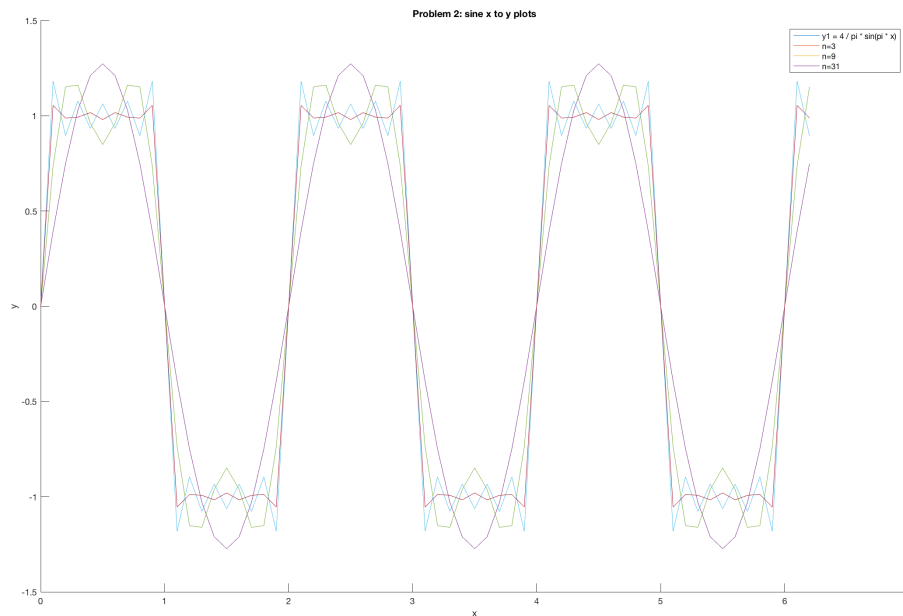
```
x3 = 0:0.1:2 * pi; % set list x3
y3 = 4 / 5 / pi * sin(5 * pi * x3) + 4 / 7 / pi * sin(7 * pi * x3) +
    4 / 9 / pi * sin(9 * pi * x3); % Calculate y3
sum = sum + y3; % add y3 to sum
plot(ax1, x3, sum); % Draw subplot
legend ({'y1 = 4 / pi * sin(pi * x)'; 'n=3'; 'n=9'}); % set legend
```

```

hold(ax1,'on'); % hold subplot

% Problem 2.4
x4 = 0:0.1:2 * pi; % set list x4
y4 = 4 / 11 / pi * sin(11 * pi * x4) + 4 / 13 / pi * sin(13 * pi * x4)
+ 4 / 15 / pi * sin(15 * pi * x4) + 4 / 17 / pi * sin(17 * pi * x4)
+ 4 / 19 / pi * sin(19 * pi * x4) + 4 / 21 / pi * sin(21 * pi * x4)
+ 4 / 23 / pi * sin(23 * pi * x4) + 4 / 25 / pi * sin(25 * pi * x4)
+ 4 / 27 / pi * sin(27 * pi * x4) + 4 / 29 / pi * sin(29 * pi * x4) +
4 / 31 / pi * sin(31 * pi * x4); % Calculate y4
sum = sum + y4; % add y4 to sum
plot(ax1, x4, sum); % Draw subplot
legend({'y1 = 4 / pi * sin(pi * x)'; 'n=3'; 'n=9'; 'n=31'}); % set
legend
hold on; % hold the figure

```



Problem 3

Problem 3.1

```

temperatureCityA = [56.0, 60.4, 65.8, 72.4, 70.5, 76.4, 80.6, 75.8,
70.3, 68.9, 61.5, 54.8]; % Save temperature of city A in an array
temperatureCityB = [44.3, 49.8, 56.8, 65.5, 73.1, 78.6, 79.4, 76.2,
70.5, 61.3, 54.7, 46.7]; % Save temperature of city B in an array
month = 1:1:12; % set month
figure(4); % Initialize figure 4
plot(month, temperatureCityA, 'r', month, temperatureCityB, 'b'); %
Draw plot
title('Problem 3.1): Plot of monthly temperature of city A and B'); %
set title

```

```

legend ('CityA', 'CityB'); % set legend
xlabel ('Month'); % set x-axis
ylabel ('Temperature'); % set y-axis

% Problem 3.2
fprintf('Problem 3.2:\nThe hottest month in CityA is July, the
    temperature is 80.6.\n');

logicHotterB = temperatureCityA < temperatureCityB; % Find the months
    have higher temperature in CityB in comparision to CityA
hotterMonthsB = find(logicHotterB == 1); % Show the months with higher
    temperature in CityB
fprintf('logicHotterB:\n'); % Print with format
disp = logicHotterB; % Display logicHotterB
fprintf('The months have higher temperature in CityB in comparision to
    CityA.\n'); % Print with format
disp(hotterMonthsB) % Display hotterMonthsB

% Problem 3.3
tempDiff = CityBTemp - CityATemp; % Calculate the monthly temperature
    difference
leastDiff = min(abs(tempDiff)); % Find the minimun absolute difference
leastMonth = find(leastDiff == tempDiff); % Find the month of least
    difference
fprintf('Problem 3.3:\n'); % Print with format
fprintf('The month shows the least temperature difference is %d and
    the temperature difference is %f\n', leastMonth, leastDiff); % Print
    with format

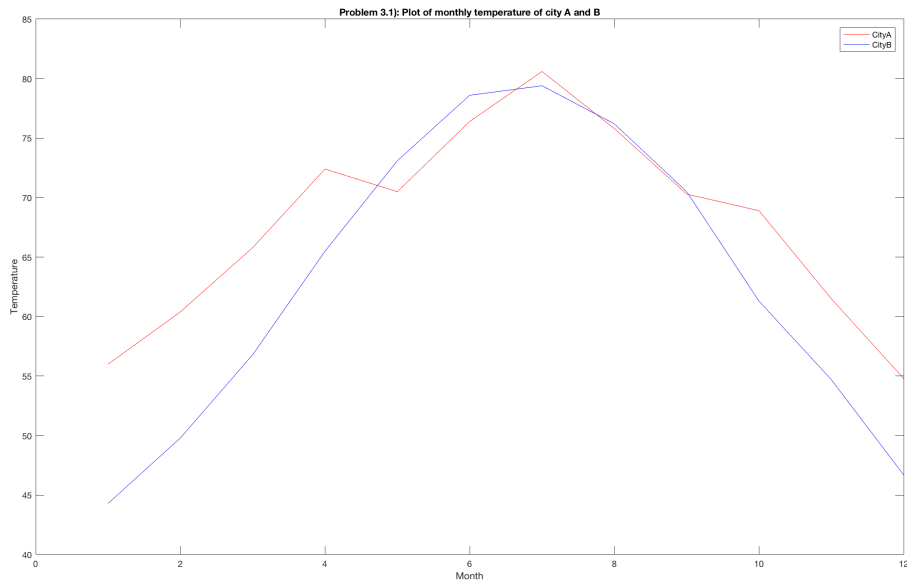
Problem 3.2:
The hottest month in CityA is July, the temperature is 80.6.
logicHotterB:
The months have higher temperature in CityB in comparision to CityA.

ans =

    1    1    1    1

Problem 3.3:
The month shows the least temperature difference is 9 and the
    temperature difference is 0.200000

```



Problem 4

Problem 4.a

```
is = inputdlg('Please enter the scores'); % Get the input string
rawScores = str2float(is{:}); % Convert input string numbers in to a
    numeric array
highestScore = max(rawScores); % Find the highest score
normScores = rawScores / highestScore * 100; % Normalize these scores
    and put it in a new row array
os = sprintf('Problem 4.a:\nThe raw scores are:\n'); % Overload output
    string into os
disp(os); % Display the output
disp(rawScores); % Display the rawScores
os = sprintf('The normalized scores are:\n'); % Overload output string
    into os
disp(os); % Display the output
disp(normScores); % Display the normScores

% Problem 4.b.1
sumScore = sum(normScores); % Calculate the sum of scores
lengthScores = length(normScores); % Get the length of normScores
    array
avg1 = sumScore / lengthScores; % Calculate average in first way
fprintf('Problem 4.b.1:\nThe average value calculated by the first
    method is %f.\n', avg1); % Print the results with format

% Problem 4.b.2
avg2 = mean(normScores); % Calculate average of normScores array with
    the built-in average function
fprintf('Problem 4.b.2:\nThe average value calculated by the second
    method is %f.\n', avg2); % Print the results with format
```

Undefined function or variable 'str2float'.

Error in CGXHW2 (line 112)

*rawScores = str2float(is{:}); % Convert input string numbers in to a
numeric array*

Published with MATLAB® R2016a