The University of Western Ontario
Computer Science 2035b
Final Examination - Friday, April 24th, 2015

Professor: John Barron

Last Name	
Given Names	
Student Number	

PLease give first and given names the university has for you (as on your student card). This exam consists of 10 questions (25 pages including this page) worth a total of 205 marks (which will be scaled to 100%). It is an open book exam, course notes and any MatLab book(s) are allowed. No calculators, laptops or cell phones are allowed. All answers are to be written in this booklet. Scrap work may be done on the back of each page; this will not be marked. The exam is 180 minutes long (3 hours) and comprises 35% of your final mark. Should your final exam grade be higher than your midterm exam grade (worth 20% of your final grade), your final exam grade in this course will count for the full 55% of your exam grade.

Please print you full name and student number in the space provided below before you start this exam.

(1)	20 marks		(6)	25 marks	
(2)	20 marks		(7)	15%	
(3)	10 marks		(8)	30%	
(4)	10 marks		(9)	30%	
(5)	25 marks		(10)	20%	
Total out of 205					

Question 1 true/false (20 marks) Choose **one** answer for each question.

(1) MatLab stands for **Mat**hmetics **Lab**atory.   true   <u>false</u>

(2) The inverse of Matrices `A*B` is `inv(A)*inv(B)`, where `inv` is the MatLab inverse function.   true   <u>false</u>

(3) For the systems of equations `A*x=B`, `x` cannot be computed if `A` is non-singular.   true   <u>false</u>

(4) For the 3×3 matrix $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$, `mean(A)` yields 45.   true   <u>false</u>

(5) For $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $A^{-1} = \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix}$ true   <u>false</u>

(6) Least Squares always computes the solution with zero residual.   true   <u>false</u>

(7) Handle graphics allows us to change the appearance of graphical entities that were plotted earlier if the handles for those plots are known.   <u>true</u>   false

(8) A GUI built using GUIDE in MatLab allows the user to design a layout for a GUI without writing a MatLab program.   <u>true</u>   false

(9) Multi-core solutions always run faster than single core solutions if overhead (data transfer and time spent for core allocation) is not considered.   <u>true</u>   false

(10) A GPU (Graphics Processing Unit) solution always run faster than vectorized solutions on a single core if overhead (data transfer) is not considered.   true   <u>false</u>

(11) Numerical integration means MatLab can compute the indefinite integral of any function numerically.   true   <u>false</u>

(12) Numerical integration means MatLab can compute the definite integral of a function numerically.   <u>true</u>   false

(13) `5*randn(1000,1)+10` yields a 1000 component column vector of normal random numbers having a standard deviation of approximately $\sigma = 5.0$ and a mean of approximately $\mu = 10.0$.   <u>true</u>   false

(14) The MatLab function `skewness` measures the symmetry in a distribution of numbers and is available via the Image Processing Processing toolbox.   true   <u>false</u>

(15) MatLab function `polyfit` does a least squares fit of the polynomial coefficients of order n to data of size $\geq n + 1$.   <u>true</u>   false

(16) Pixels with a gradient direction, computed as $\tan^{-1}\left(\frac{I_y}{I_x}\right) \geq \tau$, where I_x and I_y are the intensity derivatives in the x and y dimensions and τ is a threshold, are edgels.   true   <u>false</u>

(17) MatLab 2015a can run the edge detector `'sobel'` on the GPU.   <u>true</u>   false

(18) Histogram equalization either brightens dark parts of an image or darkens bright parts of an image but not both at the same time.   true   <u>false</u>

(19) Histogram equalization of a colour image involves histogram equalization of its 3 colour planes.   true   <u>false</u>

(20) Any calculation on a variable with a `nan` value results in an execution error.   true   <u>false</u>

(2) (20 marks) Consider the evaluation of the division of the two polynomials:

$$f(x) = \frac{1 - \frac{3}{5}x + \frac{3}{20}x^2 - \frac{1}{60}x^3}{1 + \frac{2}{5}x - \frac{1}{20}x^2}.$$

The $*$ indicates where code should be added below. There are 1000 values of x from 0 to 1.

(2a) (10 marks) Give the most efficient vectorized solution for this polynomial evaluation. Efficiency is judged by the number of multiplications required. Write MatLab code to print the first and last values of f.

Answer:

```
x=linspace(0,1,1000);
const1=-3/5;
const2=3/20;
const3=-1/60;
const4=2/5;
const5=-1/20;
% 7 multiplications
x2=x.*x;
x3=x2.*x;
f=(1+const1*x+const2*x2+const3*x3)./(1+const4*x+const5*x2);
or
% 5 multiplications required
f=(1+x*(const1+x*(const2+x*const3)))./(1+x*(const4+const5*x))
% answers of 1.0000000000 and 0.3950617284
fprintf('f(1) [for x being 0] =%14.10f\n',f(1));
fprintf('f(1000) [for x being 1] =%14.10f\n',f(1000));
```

(2b) (10 marks) Give the most efficient GPU solution for evaluating this polynomial.
Answer:

```
x=linspace(0,1,1000);
const1=-3/5;
const2=3/20;
const3=-1/60;
const4=2/5;
const5=-1/20;
x=gpuArray(x);
x2=x.*x;
x3=x2.*x;
f=(1+const1*x+const2*x2+const3*x3)./(1+const4*x+const5*x2);
f=gather(f);
```

(3) (10 marks) Consider the following system of equations:

$$
\begin{aligned}
5x + 6y + 7z &= 8 \\
7x + 6y + 5z &= 3 \\
10x + 12y + 14z &= 16 \\
14x + 15y + 16z &= 30
\end{aligned}
$$

(3a) (5 marks) Set up the matrices for this system of equation as `A*s=B`.

Answer:

$$
A = \begin{bmatrix} 5 & 6 & 7 \\ 7 & 6 & 5 \\ 10 & 12 & 14 \\ 14 & 15 & 16 \end{bmatrix},
$$

$$
B = \begin{bmatrix} 8 \\ 3 \\ 16 \\ 30 \end{bmatrix} \quad \text{and}
$$

$$
s = \begin{bmatrix} x \\ y \\ z \end{bmatrix}
$$

(3b) (5 marks) What solution does MatLab give for s computed using the least squares calculation `s=A\B` for the above system of equations?

Answer:

```
s=A\B does not produce a solution because
A is a singular matrix
```

```
MatLab gives:
Warning: Rank deficient, rank = 2, tol =  2.037010e-14
```

(4) (10 marks) Consider an initial 2D plot of the curve generated using the vectors `x` and `y` with linewidth 1.0 and colour red. Later, we want to change the 2D plot so that the curve is drawn with linewidth 3.0 and colour blue. Do not simply redraw the line. Your answers below should use handle graphics to do this and not re-plot the line. Do **not** use **gca** in your answer. The figure below shows the initial red curve and the final blue curve. The ∗ indicates where code should be added.
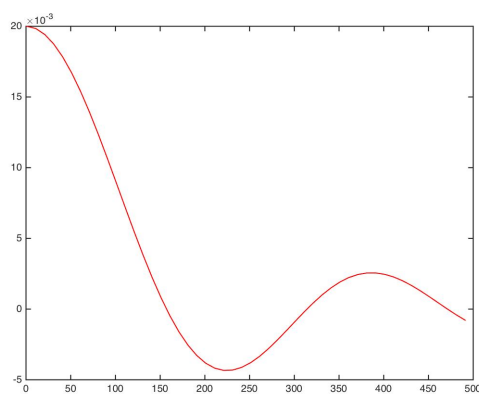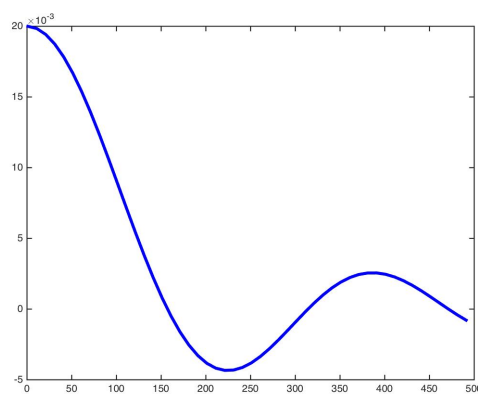
(a)   (b)

Figure 1: (a) The red curve as initially plotted (with colour red and linewidth 1.0) and (b) as it finally appears (with colour blue and linewidth 3.0).

(4a) (5 marks) Show the MatLab code for the initial curve plot:
Answer:

```
x=[1:10:500]';
y=sin(x/50)./x;
figure
```

```
h=plot(x,y,'linewidth',1.0,'color','red');
```

(4b) (5 marks) Show the MatLab code for the final curve plot:

Answer:

```
set(h,'color','blue','linewidth',3.0);
```

(5) (25 marks) Write the missing MatLab code in the appropriate places in the MatLab code below. The * indicates where code should be added.

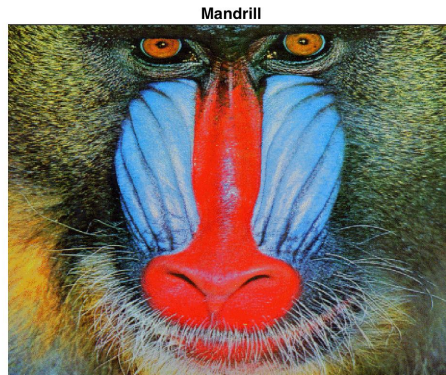(5a) (5 marks) Consider displaying the 2D 8-bit colour Mandrill image.

Figure 2: The 8 bit colour Mandrill image.

Fill in the missing MatLab code below.

Answer:

```
% Load the data for the Mandrill image
% Mandrill is 8 bit colour image, X, with
% a colour map, map

load mandrill X map figure

% imshow(X,map) will also work
image(X)
% Use the colormap specified in the image data file
colormap(map)


% Turn the axes off
axis off
% Add title
title('\fontsize{16} Mandrill')
```

(5b) (5 marks) Consider plotting three sets of y values against a set of x values as shown in the figure below. Add the needed MatLab code below.
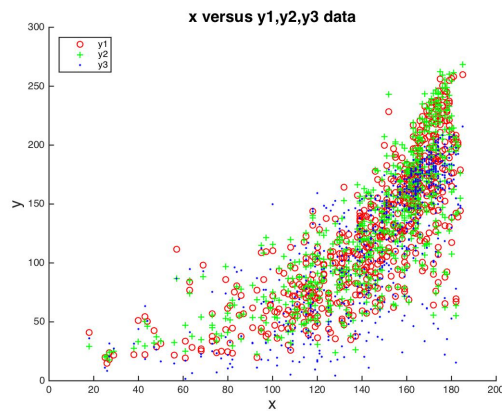

Figure 3: Plot of x versus 3 sets of y values.

Answer:

```
load q5b_data.mat x y1 y2 y3
% Could use scatter in place of plot below
% Plot the first set of data as red o's


figure
% scatter(x,y1,'ro') also works
% plots and scatters can be intermixed
plot(x,y1,'ro')
hold on


% Plot the second set of data as green +'s


% scatter(x,y1,'g+') also works
plot(x,y2,'g+')
hold on


% Plot the third set of data as blue .'s
```

```matlab
% scatter(x,y1,'b.') also works
plot(x,y3,'b.')
hold off % could be hold on or nothing here as well


% Add title and axis labels
title('\fontsize{16} x versus y1,y2,y3 data')
xlabel('\fontsize{16} x')
ylabel('\fontsize{16} y')
% Add a legend
legend('y1','y2','y3','Location','NorthWest');
```

(5c) (5 marks) Consider making a stem plot of some `sample` versus `amplitude` as shown below. Complete the MatLab code below to make this plot.
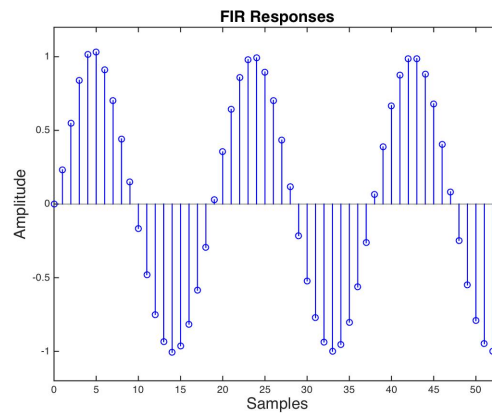

Figure 4: Stem plot of FIR data.

Answer:

```
% Load amplitude data
load amplitudeData sample amplitude
% Create a stem plot using the stem function
figure


stem(sample,amplitude)


% Adjust the axis limits
axis([0 53 -1.2 1.2])

% Add title and axis labels
title('\fontsize{16}FIR Responses')
xlabel('\fontsize{16}Samples')
ylabel('\fontsize{16}Amplitude')
```

(5d) (5 marks) Consider making an area plot of US population data using colormap `'winter'` as shown in the plot below. Fill in the missing statements in the MatLab code to make this plot:
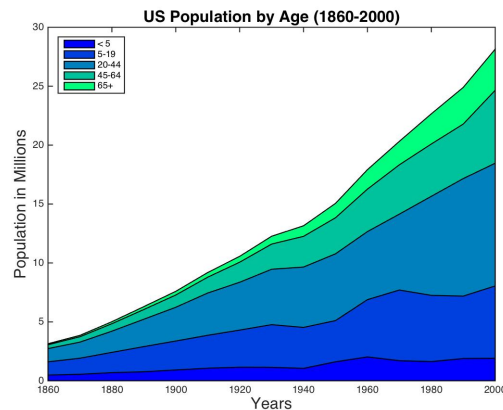

Figure 5: Area plot of US population data with population values divided by 10,000,000.

Answer:

```
% Load US population data by age groups
load PopulationAge years population groups
figure
% Create the area plot using the area function

area(years,population/10000000)
colormap winter

% Add a legend
legend(groups, 'Location', 'NorthWest')
% Add title and axis labels

title('\fontsize{16}US Population by Age (1860-2000)')
xlabel('\fontsize{16}Years')
ylabel('\fontsize{16}Population in Millions')
```

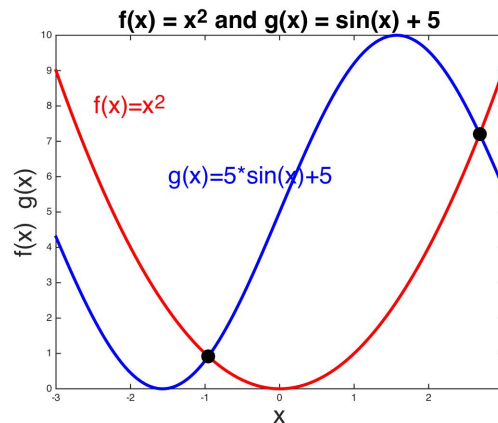(5e) (5 marks) Consider adding text to a 2D plot as shown in the figure below. Add the appro-

Figure 6: Text plotted on a 2D graph.

priate MatLab code to the program below to print the two function formulae and large black dots at the intersection points of the 2 plots to make this figure.

Answer:

```
% Define functions f=x^2 and g=5*sin(x)+5
x=-3.0:0.01:3.0;
f=x.^2;
g=5*sin(x)+5;

% Plot function f
figure
axis([-3,3,-5,15])
plot(x,f,'r-','LineWidth',3)
hold on

% Plot function g
plot(x,g,'b-','LineWidth',3)
hold on

% Add title and axis labels
title('\fontsize{24}f(x) = x^2 and g(x) = sin(x) + 5')
```

```
xlabel('\fontsize{24}x')
ylabel('\fontsize{24}f(x)  g(x)')

% Plot large black dots (markersize 48) at intersection points
% (-0.956,0.916) and (2.686,7.207);
% Setup arrays xeq and yeq

xeq(1)=-0.956;
yeq(1)=0.916;
xeq(2)=2.685;
yeq(2)=7.207;
plot(xeq,yeq,'.k','markersize',48);

% Label the curve for function f in red (fontsize 24)

text(-2.5,8,'\fontsize{24}f(x)=x^2','Color','r');

% Label the curve for function g in blue (fontsize 24)

text(-1.5,6.0,'\fontsize{24}g(x)=5*sin(x)+5','Color','b')
```

(6) (25 marks) Write the missing MatLab code in the MatLab code below. The ∗ indicates where code should be added.

(6a) (5 marks) Consider the following 3D scatter plot generated by plot3 and colorbar in the following MatLab code:
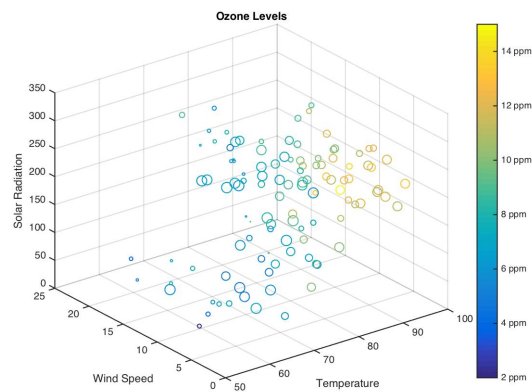
Answer:

Figure 7: 3D plot of Ozone data: temperature versus wind speed versus solar radiation.

```
% Load data on ozone levels
load ozoneData Ozone Temperature WindSpeed SolarRadiation

% Create a 3D plot of data using the plot3 and colorbar functions
% and not scatter3
figure

plot3(Temperature,WindSpeed,SolarRadiation,'o');
colorbar

% Add title and axis labels

title('\fontsize{16}Ozone Levels')
xlabel('\fontsize{16}Temperature')
ylabel('\fontsize{16}Wind Speed')
zlabel('\fontsize{16}Solar Radiation')
```

```
% Add a colorbar with tick labels
colorbar('Location','EastOutside','YTickLabel',...
    {'2 ppm','4 ppm','6 ppm','8 ppm', ...
     '10 ppm','12 ppm','14 ppm'})
```

(6b) (5 marks) Consider the following 3D plot generated by the following MatLab code:
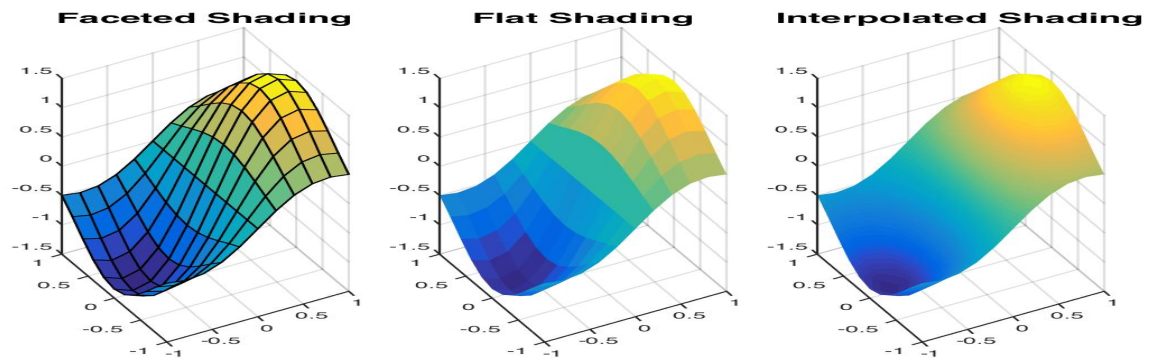
Answer:

Figure 8: 3D shading types.

```
% Create a grid of x and y points
points=linspace(-1,1,11);
[X,Y]=meshgrid(points,points);


% Define the function Z=f(X,Y)
Z=2./exp((X-.5).^2+Y.^2)-2./exp((X+.5).^2+Y.^2);


% Faceted Shading using surf


subplot(1,3,1)
surf(X,Y,Z)
% faceted shading is the default
shading faceted


view(-30,30)
title('\fontsize{16}Faceted Shading')


% Flat Shading using surf


subplot(1,3,2)
```

```
surf(X,Y,Z)
% flat shading
shading flat
```

view(-30,30)

title('\fontsize{16}Flat Shading')

```
% Interpolated Shading using surf
```

```
subplot(1,3,3)
surf(X,Y,Z)
% interpolated shading
shading interp
```

view(-30,30)
title('\fontsize{16}Interpolated Shading')

(6c) (5 marks) Consider the following 3D quiver plot generated by the following MatLab code:
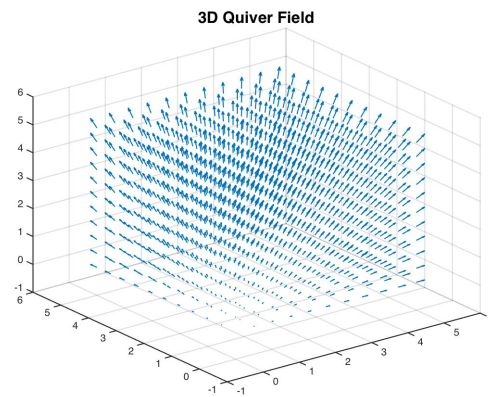
Answer:

Figure 9: 3D quiver field.

```
x=[0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0];
y=[0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0];
z=[0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0];
u=[0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0];
v=[0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0];
w=[0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0];


[X,Y,Z]=meshgrid(x,y,z);
[U,V,W]=meshgrid(u,v,w);


quiver3(X,Y,Z,U,V,W);
axis([-1 6 -1 6 -1 6]);
title('\fontsize{16}3D Quiver Field');
```

(6d) (5 marks) Consider the following 3D plots, generated by the MatLab code, below. The
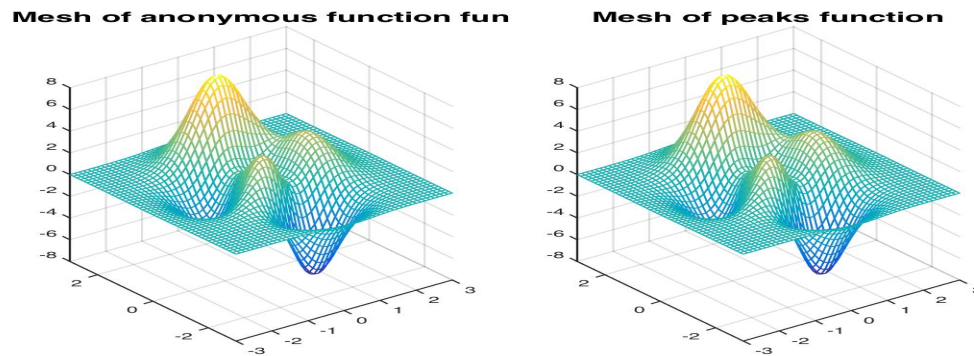

Figure 10: The peaks functions.

MatLab peaks demo function has a default of 49×49 sets of x and y values ranging from -3 to +3.

Answer:

```
x=linspace(-3,3,49);
y=linspace(-3,3,49);
[X,Y]=meshgrid(x,y);
% anonymous peaks function
fun = @(X,Y)(3*(1-X).^2.*exp(-(X.^2)-(Y+1).^2)-...
            10*(X/5-X.^3-Y.^5).*exp(-X.^2-Y.^2)-...
            1/3*exp(-(X+1).^2-Y.^2));
subplot(1,2,1)

% plot using anonymous function
mesh(X,Y,fun(X,Y));

title('\fontsize{16}Mesh of anonymous function fun');
axis([-3 3 -3 3 -8 8]);

subplot(1,2,2)

% plot using peaks function
[X,Y,Z]=peaks(49);
```
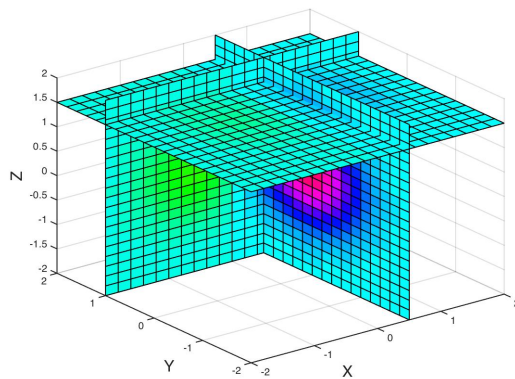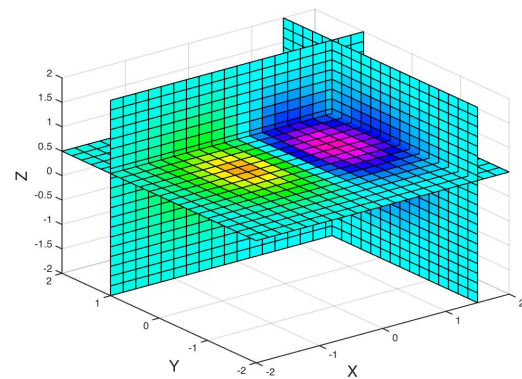
```
mesh(X,Y,Z)
title(\fontsize{16}Mesh of peaks function);
axis([-3 3 -3 3 -8 8]);
```

(6e) (5 marks) Consider the following two 3D slice plots generated by the following MatLab code:

(a)   (b)

Figure 11: Two 3D slice images.

Answer:

```matlab
[X,Y,Z] = meshgrid(-2:0.2:2,-2:0.2:2,-2:0.2:2);
% Volume
V = X.*exp(-X.^2-Y.^2-Z.^2);


Xslice = 0.5;
Yslice = 1.0;
Zslice = 1.5;


figure
slice(X,Y,Z,V,Xslice,Yslice,Zslice)
xlabel('\fontsize{16}X');
ylabel('\fontsize{16}Y');
zlabel('\fontsize{16}Z');
colormap hsv


Xslice = 1.5;
Yslice = 1.0;
Zslice = 0.5;
```

```
figure
slice(X,Y,Z,V,Xslice,Yslice,Zslice)
xlabel('\fontsize{16}X');
ylabel('\fontsize{16}Y');
zlabel('\fontsize{16}Z');
colormap hsv
```

(7) (15 marks) Consider the left grayvalue image shown below. The image has been darkened and has 25 marks salt and pepper noise added to it (the dirty image). The right image shows the image after denoising and brightening (the clean image).

(a) dirty image

(b) clean image

Figure 12: (a) The dirty image is a darkened image with 25 marks salt and pepper noise and (b) the clean image results after denoising and brightening.

Write a short MatLab code segment that takes the initial image (the image on the left) and cleans it up (produces and image like the image on the right).

Answer:

```
close all
I=imread('dirty_image.jpg');
I=squeeze(I(:,:,2));
figure
imshow(I,[0,255]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% We show the code to make the dirty image
% You could not do this...
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% darken the image by dividing by 2
I=I/2;
% add 25% salt and pepper noise
```

```
I=imnoise(I,'salt & pepper',0.25);
figure
imshow(I,[0,255]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

I=medfilt2(I);
I=medfilt2(I);
I=histeq(I);


figure
imshow(I,[0,255]);
print q7_fig2_2015.jpg -djpeg
```

(8) (30 marks) The question concerns the Matlab symbolic arithmetic toolbox. Assume $f(x, y, z)$ is both integrable and differentiable. Assume `f` is a valid MatLab symbolic expression for this function and use it in your answer.

. (8a) (10 marks) Write MatLab code to numerically evaluate this integral:

$$\int_{-1}^{1} \int_{-3}^{3} \int_{-5}^{5} f(x, y, z) dx dy dz.$$

Answer:

```
syms f x y z
f=exp(x)*sin(y)*cos(z)
eval(int(int(int(f,x,-5,5),y,-3,3),z,-1,1))
or another possibility is:
% order of integration is unimportant but
% correct limits must be specified
eval(int(int(int(f,z,-1,-1),y,-3,3),x,-5,5))
% The value is 0
```

(8b) (10 marks) Write MatLab code to numerically evaluate this derivative at $x = 1$, $y = 2$ and $z = 3$.

$$\frac{\partial^3}{\partial x \partial y \partial z} f(x, y, z)$$

Answer:

```
syms f x y z
f=exp(x)*sin(y)*cos(z)
x=1;
y=2;
z=3;
eval(diff(diff(diff(f,z),y),x))
% The value is -2.4470
```

(8c) (10 marks) Consider the following MatLab code:

```
syms A a b c d e f g h i j k l m n o
A = [a b c d;
     d e f g;
     h i j k;
     l m n o];
disp('A(1:2,1:2)')
A(1:2,1:2)
disp('A(3:4,3:4)')
A(3:4,3:4)
A(1:2,1:2)=A(1:2,1:2)*A(3:4,3:4);
disp('A(1:2,1:2)')
A(1:2,1:2)
disp('A')
A
A(3:4,3:4)=A(1:2,1:2)+A(3:4,3:4);
disp('A(3:4,3:4)')
A(3:4,3:4)
disp('A')
A
```

What is printed?

Answer to (8c) here:

```
A(1:2,1:2)
[ a, b]
[ d, e]
A(3:4,3:4)
[ j, k]
[ n, o]
A(1:2,1:2)
[ a*j + b*n, a*k + b*o]
[ d*j + e*n, d*k + e*o]
A
[ a*j + b*n, a*k + b*o, c, d]
[ d*j + e*n, d*k + e*o, f, g]
[         h,         i, j, k]
[         l,         m, n, o]
A(3:4,3:4)
[ j + a*j + b*n, k + a*k + b*o]
[ n + d*j + e*n, o + d*k + e*o]
A
[ a*j + b*n, a*k + b*o,             c,             d]
[ d*j + e*n, d*k + e*o,             f,             g]
[         h,         i, j + a*j + b*n, k + a*k + b*o]
[         l,         m, n + d*j + e*n, o + d*k + e*o]
```

(9) (30 marks) Consider the following 4×4 image (a 2D signal), `f(x,y)`:

```
f = [2 3 4 5;
     3 4 5 6;
     4 5 6 7;
     5 6 7 8];
```

(9a) (5 marks) Show the MatLab code to compute the centered Fourier transform, `g`, of this image:

Answer:

```
g=fftshift(fft2(f))
```

```
Can't show this without a computer but the fft2 yields:
   80.0000 + 0.0000i  -8.0000 + 8.0000i  -8.0000 + 0.0000i  -8.0000 - 8.0000i
   -8.0000 + 8.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   -8.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   -8.0000 - 8.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
and fftshift(fft2(f)) yields:
    0.0000 + 0.0000i   0.0000 + 0.0000i  -8.0000 + 0.0000i   0.0000 + 0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i  -8.0000 - 8.0000i   0.0000 + 0.0000i
   -8.0000 + 0.0000i  -8.0000 - 8.0000i  80.0000 + 0.0000i  -8.0000 + 8.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i  -8.0000 + 8.0000i   0.0000 + 0.0000i
```

(9b) (5 marks) What is frequency, (u, v), of the upper left hand corner of centered **g**:

Answer:

(-1/2,-1/2)

(9c) (5 marks) What is frequency, (u, v), of the lower right hand corner of centered **g**:

Answer:

```
(1/2-1/4,1/2-1/4) = (1/4,1/4)
or
(-1/2+3/4,-1/2+3/4) = (1/4,1/4)
```

(9d) (5 marks) Show the MatLab code for computing the amplitude of centered **g** at frequency $(u, v) = (0, 0)$:

Answer:

```
amplidude=abs(g(3,3))
```

(9e) (5 marks) Show the MatLab code for computing the phase of centered **g** at frequency $(u, v) = (0, 0)$:

Answer:

```
phase=atan(imag(g(3,3))/real(g(3,3)))
```

(9f) (5 marks) Give the MatLab code that only prints out all Fourier transform complex numbers with non-zero imaginary parts.
Answer:

```
fprintf('Complex responses:\n');


g(abs(imag(g))~=0)


Complex responses:
  -8.0000 - 8.0000i
  -8.0000 - 8.0000i
  -8.0000 + 8.0000i
  -8.0000 + 8.0000i
```

(10) (20 marks) This question refers to assignment 4 of this year. In that assignment we computed `average_data(num_of_years,num_of_months,num_of_elements)` where `num_of_years=73`, `num_of_months=12` and `num_of_elements=5`. So there aee $73 \times 12 \times 5$ data items in this array. Recall that `num_of_elements=1` was for average monthly temperature, `num_of_elements=2` was for average monthly rainfall, `num_of_elements=3` was for average monthly snowfall, `num_of_elements=4` was for average monthly precipitation and `num_of_elements=5` was for average monthly snowfall on the ground.

(10a) (10 marks) Give MatLab code to compute the average temperature data and its standard deviation for all 12 months and all 73 years (the average and standard deviation of 10*73 data items).

Answer:

```
% 10a - average temperature and standard deviation of all months and all years
data=squeeze(average_data(:,:,1));
fprintf('Average temperature and standard deviation ');
fprintf('for all months and years:');
fprintf('%8.2f pm %8.2f\n',mean(data(:)),std(data(:)));
```

which produces the output:

```
Average temperature and standard deviation for all months and years:7.64 pm 9.68
```

(10b) (10 marks) Give MatLab code to compute the median temperature data for all 12 months and all 73 years (10*73 data items). Do **not** use the MatLab `median` function in your answer.

Answer:

```
% 10b - median temperature of all months and all years
data=sort(data(:));
median_index=cast((numel(data)/2+0.5),'int32');
fprintf('Median temperature for all months and years:');
fprintf('%8.2f  Number of elements:%d Median index:%d\n',...
        data(median_index),numel(data),median_index);
```

which produces the output:

Median temperature for all months and years:8.08 Number of elements:876 Median index:439