GM04: MATLAB Exercise 1 Solution

```
%%%%% By N. Ovenden, T. L. Ashbee and A. Whitfield

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q1: Matrix operations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A = [5 3 1 -1; 2 -3 6 1; 6 4 1 3]
B = [1 3; 3 -4; 2 0]
C = [6 0 6 -1; 9 2 -1 5]

[A B]
[A; C]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q2: Matrix operations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%(a)
D = [2:7; 29:-3:14]

%(b)
D(3,:)  =  D(1,:) + D(2,:)

%(c)
D(4,:)  =  (D(1,:) + D(2,:))/2

%(d)
D = D + 0.1*rand(4,6).*D

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q3: Matrix operations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
A = [5 2 4; 1 7 -3; 6 -10 0]
B = [11 5 -3; 0 -12 4; 2 6 1]
C = [7 14 1; 10 3 -2; 8 -5 9]

%(a)
A*inv(A)
B*inv(B)
C*inv(C)

%(b)
A*(B+C)
A*B+A*C

%(c)
[V,D] = eig(A)
V*D*inv(V)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q4: Solving linear equations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%(a)
A = [4 -2 6; 2 8 2; 6 10 3]
b = [8; 4; 0]

x = inv(A)*b
% or
x = A\b

% test error
abs(A*x-b)

%(b)
A = [2 3 -1 4; 1 1 -3 5; 7 1 3 4; 5 4 3 -11]
b = [23; 11; 12; 14]

x = inv(A)*b
```

```
% or
x = A\b

% test error
abs(A*x-b)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q5: Simple plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x_vec = -3: 6/20: 3
y_vec = (x_vec.^2 + 1).^3.*x_vec.^3
figure(1)
plot(x_vec, y_vec) % see end of document for plot

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q6: Features of plot command
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x_vec = -2: 4/20: 2;

figure(2)
Q6_fig=plot(x_vec, cosh(x_vec))
hold on
plot(x_vec, sinh(x_vec),'--')
hold on
plot(x_vec, tanh(x_vec),':')

xlabel('x')
ylabel('y')
legend('cosh(x)', 'sinh(x)','tanh(x)')
xlim([-1 1])

saveas(Q6_fig, 'Ex1_Q6_fig', 'jpg')

% see end of document for plot

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

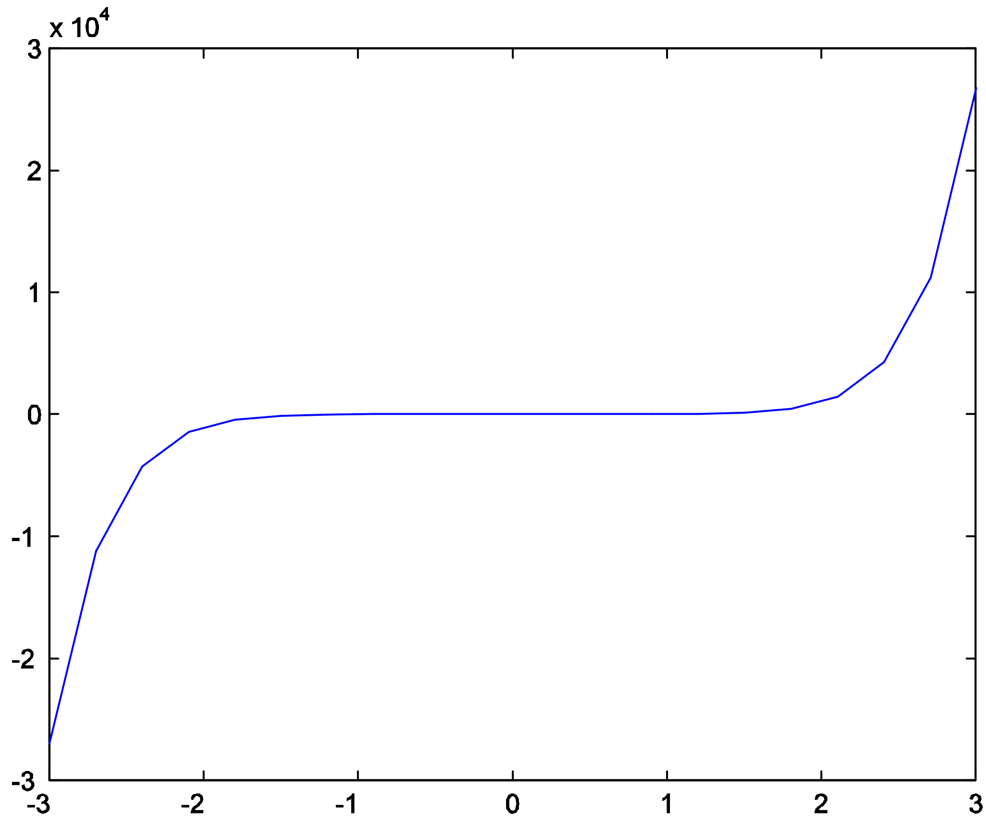
```
%%%%% Q7: Numerical approximation of infinite series
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n_vec100 = 1:1:100;
sum(1./n_vec100.^2) - pi^2/6

n_vec1000 = 1:1:1000;
sum(1./n_vec1000.^2) - pi^2/6

n_vec10000 = 1:1:10000;
sum(1./n_vec10000.^2) - pi^2/6

sum((-1).^(n_vec10000+1)./n_vec10000) - log(2)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q8: Functions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

-------------------------------------------------------
-------------- m-file spherecircle.m ------------------

function [area_circle, perimeter_circle, volume_sphere, ...
surface_area_sphere] = spherecircle(radius)

area_circle = pi*radius^2;

perimeter_circle = pi*2*radius;

volume_sphere = 4*pi*radius^3/3;

surface_area_sphere = 4*pi*radius^2;

--------------------------------------------------------
--------------------------------------------------------

% example call
[A, P, V, SA] = spherecircle(1)
```

```
--------------------------------------------------------
----------------- m-file cylinder.m --------------------

function [volume, surface_area] = cylinder(radius, length)

volume = pi*radius^2*length;
surface_area = 2*pi*radius*length + 2*pi*radius^2;

--------------------------------------------------------
--------------------------------------------------------

% example call
[V, SA] = cylinder(1,1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q9: Functions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

--------------------------------------------------------
---------------- m-file cross_product.m ----------------

function [c] = cross_product(a, b)

c(1) = a(2)*b(3) - a(3)*b(2);
c(2) = a(3)*b(1) - a(1)*b(3);
c(3) = a(1)*b(2) - a(2)*b(1);

--------------------------------------------------------
--------------------------------------------------------

% example call
a = [1,2,3];
b = [4,5,6];
cross_product(a, b)

% check with MATLAB's built-in function
cross(a, b)
```

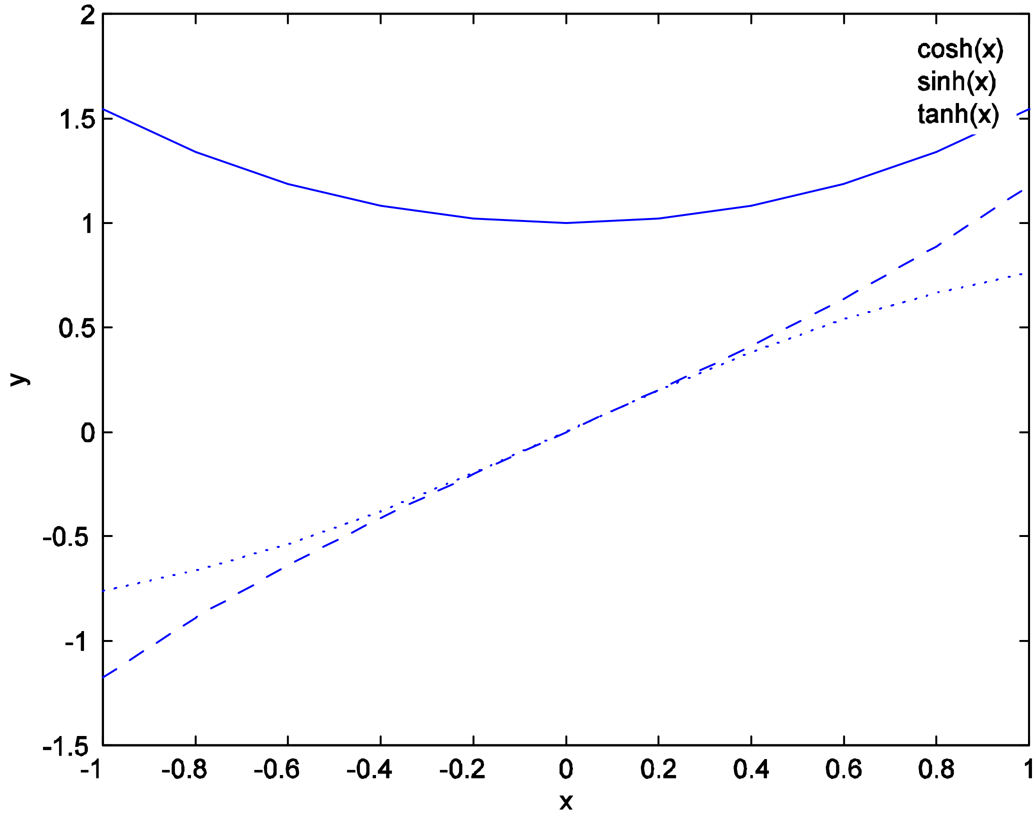
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q10: Numerical integration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

--------------------------------------------------------
----------- m-file simpson_integration.m ---------------

function [ans] = simpson_integration(f, a, b)

ans = (b-a)*(f(a) + 4*f((a+b)/2) + f(b))/6;

--------------------------------------------------------

% example call
f = @(x)(x.^2);
simpson_integration(f, 0, 1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q11: Mesh plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x = -4:8/40:4;
y = -4:8/40:4;

[xx,yy] = meshgrid(x,y);

zz = 1.8^(-sqrt(xx.^2+yy.^2)).*sin(xx).*cos(yy/2);

figure(3)
mesh(xx,yy,zz)
xlabel('x')
ylabel('y')
zlabel('z')
title('My mesh plot');

figure(4)
surf(xx,yy,zz)
xlabel('x')
```

```
ylabel('y')
zlabel('z')

z_max = max(max(zz));
z_min = min(min(zz));

cont_vec = z_min: (z_max-z_min)/19: z_max;
figure(5)
contour(xx,yy,zz,cont_vec)
```

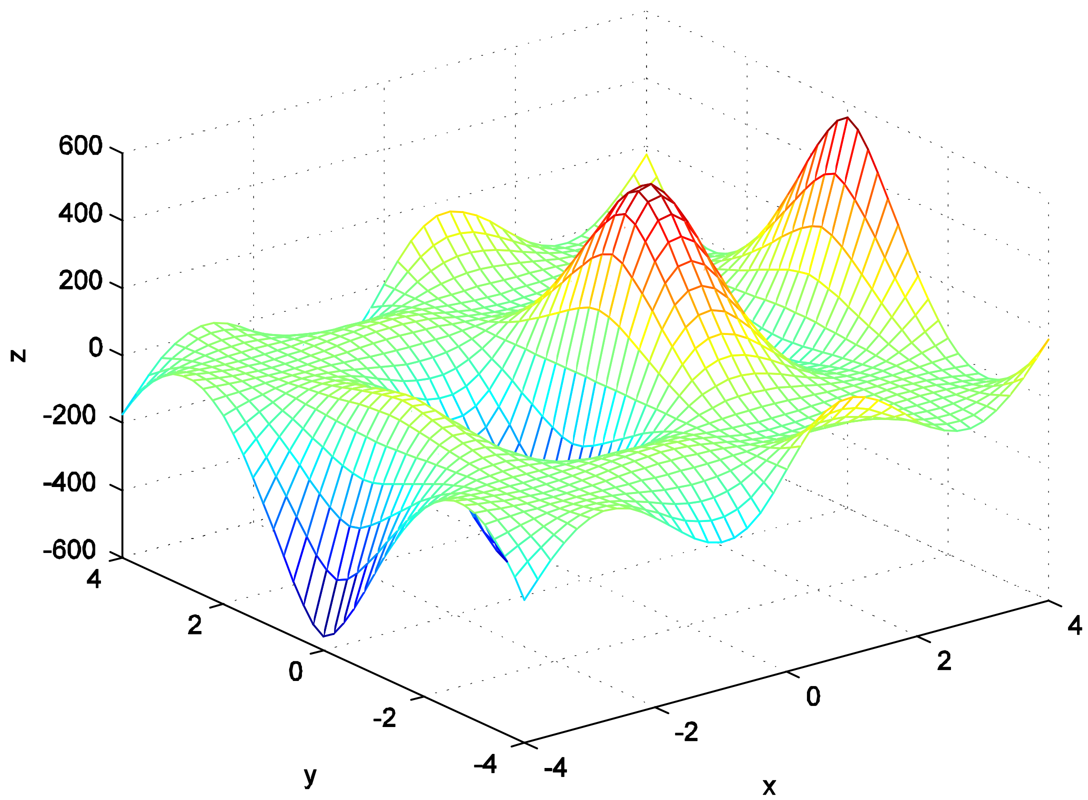
Q5:



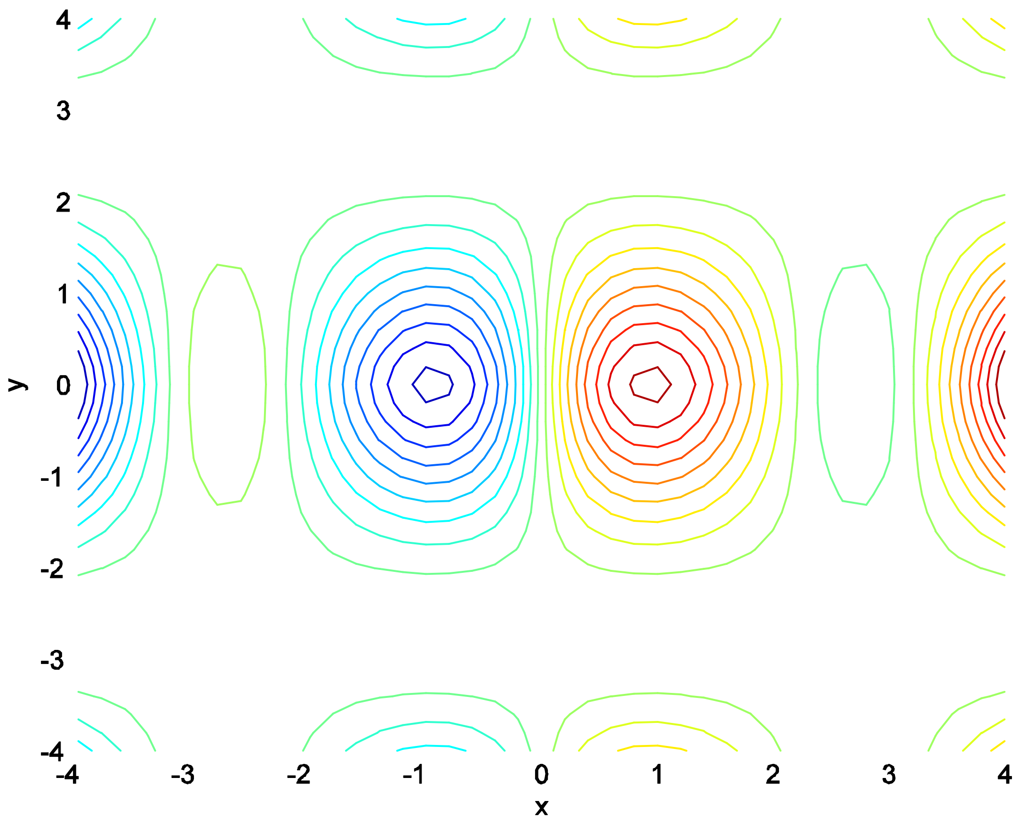
Q6:



Q11 (mesh plot)



Q11 (contour plot)



Q11 (surface plot)