1. Exercise 1 Gram-Schmidt Procedure

clear;clc;  
v1 = [-2, 2]';  
v2 = [2, 1]';  
[e,a] = orthonormal(v1,v2)

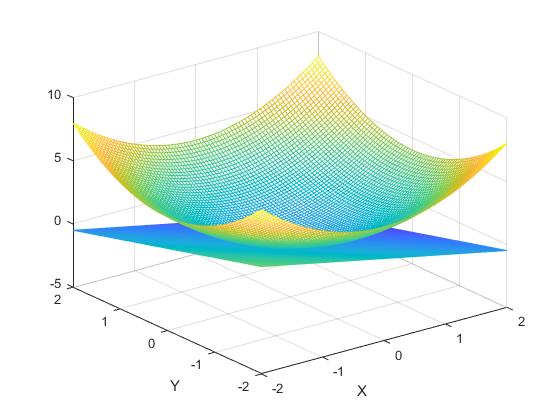
e =  
 -0.7071  
 0.7071  
a =  
 0.7071  
 0.7071

function varargout = orthonormal(varargin)  
% 需要判断一下输出与输入个数，按照公式输入方程即可  
varargout{1} = varargin{1}/norm(varargin{1});  
for j = 2:nargout  
 temp = 0;  
 for i = 1:j - 1  
 temp = temp + varargin{j}'\*varargout{1}\*varargout{1};  
 end  
 t = varargin{j} - temp;  
 varargout{j} = t/norm(t);  
end  
end

[*Published with MATLAB® R2018a*](https://www.mathworks.com/products/matlab)

1. Exercise 2 Tangent Plane

clear;clc;  
paraboloid([-0.5, -0.5]')



创建原函数并求梯度 1

找到所求点的切面 1

画出原函数和切面 并在切点处标记 1

function paraboloid(x)

x0 = x(1);y0 = x(2);

## 创建原函数并求梯度

f =@(x,y) x.^2 + y.^2;  
[X, Y] = meshgrid(-2:0.05:2);  
[fx,fy] = gradient(f(X,Y),0.05); % 以0.05在X Y方向求梯度

## 找到所求点的切面

t = (X == x0) & (Y == y0); % 找到所求点的index  
index = find(t);  
fx0 = fx(index);  
fy0 = fy(index); % 得到切面法线x y  
z = @(x,y) f(x0,y0) + fx0\*(x-x0) + fy0\*(y-y0);

## 画出原函数和切面 并在切点处标记

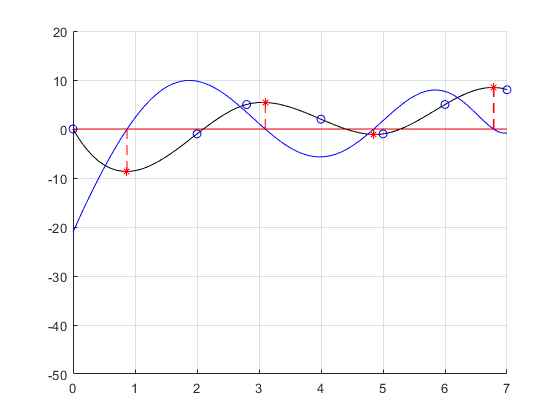
mesh(X,Y,f(X,Y))  
hold on  
mesh(X,Y,z(X,Y))  
plot3(x0,y0,f(x0,y0),'r\*') % 标记切点  
xlabel('X')  
ylabel('Y')

end

[*Published with MATLAB® R2018a*](https://www.mathworks.com/products/matlab)

1. Exercise 3 Polynomial Interpolation

clear;clc  
data = {[0,0],[2,-1],[2.8,5],[4,2],[5,-1],[6,5],[7,8]};  
Polynomial\_Interpolation(data)



画出水平红线y=0 并画出原数据 1

对原数据插值 得到对应Y坐标 1

拟合插值后的函数 并求得一阶导 2

最大最小值 2

function Polynomial\_Interpolation(data)

## 画出水平红线y=0 并画出原数据

x = [];y = [];  
figure  
hold on  
grid on  
line([0 7],[0 0],'color','red')  
axis([0 7 -50 20])  
for i = 1:length(data)  
 x = [x data{i}(1)];  
 y = [y data{i}(2)];  
end  
plot(x,y,'bo')

## 对原数据插值 得到对应Y坐标

X = 0:0.01:7;  
Y = interp1(x,y,X,'spline');  
plot(X,Y,'k');

## 拟合插值后的函数 并求得一阶导

p = polyfit(X,Y,8);  
fp = p(1:8).\*[8:-1:1];  
plot(X,polyval(fp, X),'b')

## 最大最小值

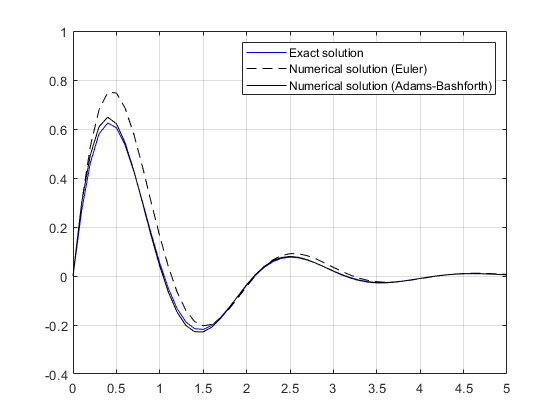
index = find(abs(polyval(fp, X) - 0) < 7e-2); % 这里由于插值得到的Y坐标并不会正好等于0，只能判断趋近于0  
xp = X(index);yp = Y(index);  
plot(xp,yp,'r\*');  
for i=1:length(index)  
 line([xp(i) xp(i)],[0 yp(i)],'color','red','linestyle','--')  
end

end

[*Published with MATLAB® R2018a*](https://www.mathworks.com/products/matlab)

1. Exercise 4 Numerical Methods for ODEs

clear;clc;  
f = @(t,y) -y + 3\*cos(3\*t)\*exp(-t);  
y = @(t) sin(3\*t).\*exp(-t);  
y0 = 0;  
h = 0.1;  
Numerical\_Methods(f,y,y0,h)



初始值 1

以h为间距开始迭代 1

画出三条曲线 2

function Numerical\_Methods(f,y,y0,h)

## 初始值

yeuler = [y0]; % y0  
yadams = [y0 y0 + h\*f(0,y0)]; % y0 y1

## 以h为间距开始迭代

按公式输入

h = 0.1;  
t = 0:h:5;  
i = 1;  
for k = 1:50  
 tempeuler = yeuler(i);  
 tempadams1 = yadams(i);  
 tempadams2 = yadams(i+1);  
 i = i + 1; % 增大yeuler和yadams  
 yeuler = [yeuler tempeuler + h\*f(t(k),tempeuler)];  
 yadams = [yadams tempadams2 + 1.5\*h\*f(t(k+1),tempadams2) - 0.5\*h\*f(t(k),tempadams1)];  
end

## 画出三条曲线

plot(t,y(t),'b') % 这里原函数积分得到  
hold on  
plot(t,yeuler,'k--')  
plot(t,yadams(1:51),'k')  
legend('Exact solution','Numerical solution (Euler)','Numerical solution (Adams-Bashforth)')  
axis([0 5 -0.4 1])  
grid on

end

[*Published with MATLAB® R2018a*](https://www.mathworks.com/products/matlab)