yiyang3
1503849

Exercise 1 p.305 6.7

Part b

Script:

```
H1=[-6 6;6 0];
H2=[6 -6;-6 0];
H3=[-6 6;6 -12];
H4=[6 -6;-6 12];
res = [eig(H1) eig(H2) eig(H3) eig(H4)];
```

result:

(0,0) H1	(0,-1) H2	(-1,-1) H3	(1,0) H4
-9.70820393249937	-3.70820393249937	-15.7082039324994	2.29179606750063
3.70820393249937	9.70820393249937	-2.29179606750063	15.7082039324994
Not definite	not definite	negative definite	positive definite
saddle point	saddle point	local maximum	local minimum

Exercise 2

Part a

Pk = rPk-1

Pk = r^2*Pk-2

Pk = r^3*Pk-3

……

So let Pk = Po* r^k

We use Gauss-Newton Method to do non linear least square fit.

The main part of the script is as follows:

```
r=2; p0=0.1;                    % I take initial as 2 and 0.1 based on observation
for i=1:30                      % loop to update p0 and r, until coverge
   d=Jaco(p0,r)\error(p0,r);    % Jaco is a 2d array of  partial derivative of r^k and p0
                                % [ -k*p0*r^k  -r^k]  where k =1,2,3,4..8
                                   % error is p0*r^k-Pk where k=1,2,..8  Pk = 0.19,0.36,0.69,1.3,….14
   r=r+d(1)                     % update
   p0=p0+d(2)
   if norm(error)<0.0001        % determine if converge
        %converge
   end   end
```

Finally, we have p0 = 0.1616, r = 1.7494

yiyang3
1503849

Part b

```
k = [1 1;1 2;1 3 ;1 4; 1 5; 1 6; 1 7 ; 1 8;];

p = log[0.19  0.36  0.69 1.3 2.5 4.7 8.5 14]^T;

res = k/p;

p0=exp(res(1));

r = exp(res(2));
```

We get p0 = 0.1058, r = 1.8642

Compare to the non linear fit, the linear fit has a slightly smaller p0 and a slightly larger r. The difference Is acceptable and both models works.