```
function dGdt =
forest_human(t,G,S,v,rho,phi,alpha,omega,X,tau,s,a_g,a_f,n,m,...
    p_f,q_f,p_g,q_g,p_pl1,p_pl2,p_pl3,p_pl4,p_cr,p_B,d_c,d_e)

% G(1) represents proportion of forest cover,
% G(2) represents proportion of grass cover,
% G(3) represents preference for forests,
% G(4) represents preference for grasses,


%% recruitment function
% alpha is the maximum recruitment rate, phi represents the
% calibrated grass cover threshold constant, S is soil moisture content,
% omega represents the recruitment transition constant
recruitment = alpha/(1 + exp((G(2)/(G(1) + G(2)) - phi/((0.5 - S)))/omega));

%% J(x_F) human influence on forest term in model, conditional on the
preference for forest
% X represents the diffusion of practices threshold
%tau represents the diffusion of practices rate and rho represents the
%maximum potential human influence

human_influenceF = rho/(1 + exp((X - G(3))/tau));

% J(x_G) human influence on grassland term in model, conditional on the
preference for forest
human_influenceG = rho/(1 + exp((X - G(4))/tau));

% J(x_A) human influence on agriculture term in model, conditional on the
preference for forest
human_influenceA = rho/(1 + exp((X - (1 - G(3) - G(4)))/tau));

%% revenue generated from harvesting on a seven year cycle
for int=1:m % initial investment (p_pl1)
if int==1 || rem(int,7)==1
    ini=int;
    p_pl1=p_pl1;
else
    p_pl1=0;
    ini=0;
end

if int==2 || int==4 || int==6 || rem(int,7)==2 || rem(int,7)==4 || rem(int,
7)==6
    mt=int;
    p_pl2=p_pl2; % maintenance costs of plantation (p_pl2)
else
    p_pl2=0;
    mt=0;
end

if int==3 || int==5 || rem(int,7)==3 || rem(int,7)==5
    delta=int;
    p_pl3=p_pl3; % revenue generated from trimming trees every 2 years
(p_pl3)
```

```
else
    p_pl3=0;
    delta=0;
end

if rem(int,7)==0 % revenue from cutting the tree down in entirety
    h=int;
    p_pl4=p_pl4;
else
    p_pl4=0;
    h=0;
end

% discounting each stage of the harvest rotation
% d_e is the economic discount rate, ini, mt, delta and h are the number of
% years discounted
p_pl1 = p_pl1*(((1/(1 + d_e))^ini) - ((1/(1 + d_e))^(ini + 1)))/(1 - (1/(1 +
d_e)));
p_pl2 = p_pl2*(((1/(1 + d_e))^mt) - ((1/(1 + d_e))^(mt + 1)))/(1 - (1/(1 +
d_e)));
p_pl3 = p_pl3*(((1/(1 + d_e))^delta) - ((1/(1 + d_e))^(delta + 1)))/(1 - (1/
(1 + d_e)));
p_pl4= p_pl4*(((1/(1 + d_e))^h) - ((1/(1 + d_e))^(h + 1)))/(1 - (1/(1 +
d_e)));

a(int)=sum(p_pl1);
b(int)=sum(p_pl2);
c(int)=sum(p_pl3);
d(int)=sum(p_pl4);

int=int+1;

end
p_pl=sum(a) + sum(b) + sum(c) + sum(d);

%% penalty for insufficient natural land reserves
if G(1) + G(2)>0.2
    p_B=0;
else
    p_B=p_B;

end

%% discounted utility values for land cover variables

% d_c is the conservation discount rate, d_e is the economic discount rate,
% n is the discount time horizon for conservation projects, m is the
% discount time horizon for economic projects

% conservation/profit term of forest cover, based on scarcity hypothesis
% and financial gains
u_f(F) = q_f*(1/(1 + d_c))*(1 - (1/(1 + d_c))^n)/(1 - (1/(1 + d_c)))*(1 -
G(1)) +...
    p_f*(1/(1 + d_e))*(1 - (1/(1 + d_e))^m)/(1 - (1/(1 + d_e)));
```

```
% p_f is the profit from natural forest cover, q_f represents the
% conservation value of forest

%conservation/profit term of grass cover, based on scarcity hypothesis
% and financial gains
u_g(G) = q_g*(1/(1 + d_c))*(1 - (1/(1 + d_c))^n)/(1 - (1/(1 + d_c)))*(1 -
G(2)) +...
    p_g*(1/(1 + d_e))*(1 - (1/(1 + d_e))^m)/(1 - (1/(1 + d_e)));
% p_g is the profit from natural grass cover, q_g represents the
% conservation value of grassland

% utility of converted land is purely economic
u_a(A) = 0.57*p_cr*(1/(1 + d_e))*(1 - (1/(1 + d_e))^m)/(1 - (1/(1 + d_e)))
+ ...
    0.43*p_pl - p_B*(0.2 - G(1) - G(2));
% p_B is the penalty for insufficient natural land cover, p_pl is the
% profit from plantations, p_cr is the profit from crops
% 0.2 is the maximum natural land requirement, 0.57 is the proportion of
% converted land used for crops, 0.43 is the proportion of converted land
% used for plantations

%% variable equations
dGdt = [
    recruitment*G(1)*G(2) - v*G(1) + human_influenceF*G(1)*(1 - G(1)) - ...
    human_influenceG*G(1)*G(2) - human_influenceA*G(1)*(1 - G(1) - G(2)) +
a_f*(1 - G(1) - G(2));...

    v*G(1) - recruitment*G(1)*G(2) + human_influenceG*G(2)*(1 - G(2)) - ...
    human_influenceF*G(2)*G(1) - human_influenceA*G(2)*(1 - G(1) - G(2)) +
a_g*(1 - G(1) - G(2));...

    s*((u_f(F) - u_a(A))*G(3)*(1 - G(3) - G(4)) + (u_f(F) -
u_g(G))*G(3)*G(4));...

    s*((u_g(G) - u_f(F))*G(4)*G(3) + (u_g(G) - u_a(A))*G(4)*(1 - G(3) -
G(4)))];

% v is the natural disturbance rate, a_f is the abandonment of plantation to
forest,
% a_g is the abandonment of crops to grassland, s represents the social
learning habit


%% to constrain the model output between 0 and 1
if G(3)>=1 && dGdt(3)>0
    dGdt(3)=0;
    dGdt(4)=dGdt(3)+dGdt(4);
end

if G(4)>=1 && dGdt(4)>0
    dGdt(4)=0;
    dGdt(3)=dGdt(3)+dGdt(4);
end

if G(1)>=1 && dGdt(1)>0
```

```
    dGdt(1)=0;
    dGdt(2)=dGdt(2)+dGdt(1);
end

if G(2)>=1 && dGdt(2)>0
    dGdt(2)=0;
    dGdt(1)=dGdt(2)+dGdt(1);
end



% Parameter planes two-variable ode model

tspan=0:5000;
options=odeset('NonNegative',1:4);

epsilon=0.02;

%% parameter values to be input into model
S=0.2;
v=0.02;
rho=0.15; % can be 0.05,0.15,0.25
phi=0.149;
alpha=0.2;
omega=0.0667;
X=0.5;
tau=0.075;
s=0.42;
a_g=0.0057;
a_f=0.0043;
n=28;
m=7;
p_f=0.05;
q_f=0.1;
p_g=0.35;
q_g=0.03;
p_pl1=-0.4;
p_pl=-1.2;
p_pl3=-0.4;
p_pl4=12;
p_cr=1;
p_B=0.01;
d_c=0.001;
d_e=0.1;

p=2601; %number of parameter combinations to run the model simulations more
quickly
  A=ones(p,10); % 10 represents the number of initial conditions
  B=ones(p,10);
  C=ones(p,10);
  D=ones(p,10);
  H=ones(p,10);
  I=ones(p,10);
  J=ones(p,10);
  K=ones(p,10);
```

```
  L=ones(p,10);
  M=ones(p,10);

  A1=zeros(p,1);
  B1=zeros(p,1);
  C1=zeros(p,1);
  D1=zeros(p,1);
  H1=zeros(p,1);
  I1=zeros(p,1);
  J1=zeros(p,1);
  K1=zeros(p,1);
  L1=zeros(p,1);
  M1=zeros(p,1);
  O=zeros(p,1);


for e=0.05:0.1:0.95 % loop through initial forest composition
    for b=0.05:0.1:0.95

G0=[e;(1-e)*b;0.2;0.3];

for i=1:1
    for j=1:1

  E=zeros(i,1);


    for n=1:2:101 %loop through set of parameters
        for m=1:1:51

%simulate ode

[t,G]=ode45(@(t,G)forest_human(t,G,S,v,rho,phi,alpha,omega,X,tau,s,a_g,a_f,n,
m,...

p_f,q_f,p_g,q_g,p_pl1,p_pl2,p_pl3,p_pl4,p_cr,p_B,d_c,d_e),tspan,G0,options);

% initial min and max points to find oscillations
max_pt(j)=0;
min_pt(j)=500;% categorizing output into cater


        for t=4985:5000 % loop through the last 15 years to find oscillations


                    if G(t)>max_pt(j)
                        max_pt(j)=G(t); %find greatest proportion of forest
cover in over the last 15 years

                    end

                    if G(t)<min_pt(j)
                        min_pt(j)=G(t); %find lowest proportion of forest
cover in over the last 15 years
                    end
```

```
            t=t+1;
    end


            if abs(max_pt(j)-min_pt(j))>epsilon %see if oscillations
occur
                E(i)=10; %set point equal to 10 for cycle in
future use of pcolor plot

            else
                E(i)=G(5000);% otherwise equilibrium is the last
result from the ode simulation

            end

        j=j+1;
        i=i+1;

    end

    end


for c=1:10
%% categorizing output into caterogies of full grass cover (0), mostly
% grassland (3), mostly forest cover (5), full forest cover (7) and
oscillations (10)
if 0<=e && e<0.1 && b>=0.1*(c)-0.06 && b<0.1*(c)+0.03
A(E<=0.05,c)=0;
A(E>0.05 & E<0.5,c)=3;
A(E==10,c)=10;
A(E>=0.5 & E<0.9,c)=5;
A(E>=0.9 & E<=1.1,c)=7;

elseif 0.1<=e && e<0.2 && b>=0.1*(c)-0.06 && b<0.1*(c)+0.03
B(E<=0.05,c)=0;
B(E>0.05 & E<0.5,c)=3;
B(E==10,c)=10;
B(E>=0.5 & E<0.9,c)=5;
B(E>=0.9 & E<=1.1,c)=7;

elseif 0.2<=e && e<0.3 && b>=0.1*(c)-0.06 && b<0.1*(c)+0.03
C(E<=0.05,c)=0;
C(E==10,c)=10;
C(E>=0.05 & E<0.5,c)=3;
C(E>=0.5 & E<0.9)=5;
C(E>=0.9 & E<=1.1,c)=7;

elseif 0.3<=e && e<0.4 && b>=0.1*(c)-0.06 && b<0.1*(c)+0.03
D(E<=0.05,c)=0;
D(E==10,c)=10;
D(E>0.05 & E<0.5,c)=3;
D(E>=0.5 & E<0.9,c)=5;
D(E>=0.9 & E<=1.1,c)=7;
```

```
elseif e>=0.4 && e<0.5 && b>=0.1*(c)-0.06 && b<0.1*(c)+0.03
H(E<=0.05,c)=0;
H(E==10,c)=10;
H(E>0.05 & E<0.5,c)=3;
H(E>=0.5 & E<0.9,c)=5;
H(E>=0.9 & E<=1.1,c)=7;

elseif e>=0.5 && e<0.6 && b>=0.1*(c)-0.06 && b<0.1*(c)+0.03
I(E<=0.05,c)=0;
I(E==10,c)=10;
I(E>=0.05 & E<0.5,c)=3;
I(E>=0.5 & E<0.9,c)=5;
I(E>=0.9 & E<=1.1,c)=7;

elseif e>=0.6 && e<0.7 && b>=0.1*(c)-0.06 && b<0.1*(c)+0.03
J(E<=0.05,c)=0;
J(E==10,c)=10;
J(E>0.05 & E<0.5,c)=3;
J(E>=0.5 & E<0.9,c)=5;
J(E>=0.9 & E<=1.1,c)=7;

elseif e>=0.7 && e<0.8 && b>=0.1*(c)-0.06 && b<0.1*(c)+0.03
K(E<=0.05,c)=0;
K(E==10,c)=10;
K(E>0.05 & E<0.5,c)=3;
K(E>=0.5 & E<0.9,c)=5;
K(E>=0.9 & E<=1.1,c)=7;

elseif e>=0.8 && e<0.9 && b>=0.1*(c)-0.06 && b<0.1*(c)+0.03
L(E<0.05,c)=0;
L(E==10,c)=10;
L(E>0.05 & E<0.5,c)=3;
L(E>=0.5 & E<0.9,c)=5;
L(E>=0.9 & E<=1.1,c)=7;

elseif e>=0.9 && e<1 && b>=0.1*(c)-0.05 && b<0.6*(c)+0.03
M(E<=0.05,c)=0;
M(E==10,c)=10;
M(E>0.05 & E<0.5,c)=3;
M(E>=0.5 & E<0.9,c)=5;
M(E>=0.9 & E<=1.1,c)=7;
%
end

end
    end


    end
end

end

%% determine if system is bistable by comparing final output from all intial
```

```
conditions
% all intial conditions lead to limit cycles (10), at least one initial
% condition leads to limit cycles (12), only one equilibrium point for all
% initial conditions (value of equilibrium point), multiple equilibrium
points for different intial
% conditions, multistable (15)
for id=1:length(A)

if A(id,1)==10 && A(id,2)==10 && A(id,3)==10 && A(id,4)==10 && A(id,5)==10 &&
A(id,6)==10 && A(id,7)==10 && A(id,8)==10 && A(id,9)==10 && A(id,10)==10;
    A1(id)=10;

elseif A(id,1)==10 || A(id,2)==10 || A(id,3)== 10 || A(id,4)==10 || A(id,
5)==10 || A(id,6)==10 || A(id,7)==10 || A(id,8)==10 || A(id,9)==10 || A(id,
10)==10 ;
    A1(id)=12;

elseif A(id,1)==A(id,2) && A(id,2)==A(id,3) && A(id,3)==A(id,4) && A(id,
4)==A(id,5) && A(id,5)==A(id,6) && A(id,6)==A(id,7) && A(id,7)==A(id,8) &&
A(id,8)==A(id,9) && A(id,9)==A(id,10);
   A1(id)= A(id,3);

else
A1(id)=15;

end


end
for id=1:length(B)

if B(id,1)==10 && B(id,2)==10 && B(id,3)==10 && B(id,4)==10 && B(id,5)==10 &&
B(id,6)==10 && B(id,7)==10 && B(id,8)==10 && B(id,9)==10 && B(id,10)==10;
    B1(id)=10;

elseif B(id,1)==10 || B(id,2)==10 || B(id,3)== 10 || B(id,4)==10 || B(id,
5)==10 || B(id,6)==10 || B(id,7)==10 || B(id,8)==10 || B(id,9)==10 || B(id,
10)==10;
    B1(id)=12;

elseif B(id,1)==B(id,2) && B(id,2)==B(id,3) && B(id,3)==B(id,4) && B(id,
4)==B(id,5) && B(id,5)==B(id,6) && B(id,6)==B(id,7) && B(id,7)==B(id,8) &&
B(id,8)==B(id,9) && B(id,9)==B(id,10);
   B1(id)= B(id,3);

else
B1(id)=15;

end

end

for id=1:length(C)

if C(id,1)==10 && C(id,2)==10 && C(id,3)==10 && C(id,4)==10 && C(id,5)==10 &&
```

```
C(id,6)==10 && C(id,7)==10 && C(id,8)==10 && C(id,9)==10 && C(id,10)==10;
    C1(id)=10;

elseif C(id,1)==10 || C(id,2)==10 || C(id,3)== 10 || C(id,4)==10 || C(id,
5)==10 || C(id,6)==10 || C(id,7)==10 || C(id,8)==10 || C(id,9)==10 || C(id,
10)==10;
    C1(id)=12;

elseif C(id,1)==C(id,2) && C(id,2)==C(id,3) && C(id,3)==C(id,4) && C(id,
4)==C(id,5) && C(id,5)==C(id,6) && C(id,6)==C(id,7) && C(id,7)==C(id,8) &&
C(id,8)==C(id,9) && C(id,9)==C(id,10);
   C1(id)= C(id,3);

else
C1(id)=15;

end

end

for id=1:length(D)

if D(id,1)==10 && D(id,2)==10 && D(id,3)==10 && D(id,4)==10 && D(id,5)==10 &&
D(id,6)==10 && D(id,7)==10 && D(id,8)==10 && D(id,9)==10 && D(id,10)==10;
    D1(id)=10;

elseif D(id,1)==10 || D(id,2)==10 || D(id,3)== 10 || D(id,4)==10 || D(id,
5)==10 || D(id,6)==10 || D(id,7)==10 || D(id,8)==10 || D(id,9)==10 || D(id,
10)==10;
    D1(id)=12;

elseif D(id,1)==D(id,2) && D(id,2)==D(id,3) && D(id,3)==D(id,4) && D(id,
4)==D(id,5) && D(id,5)==D(id,6) && D(id,6)==D(id,7) && D(id,7)==D(id,8) &&
D(id,8)==D(id,9) && D(id,9)==D(id,10);
   D1(id)= D(id,3);

else
D1(id)=15;

end

end

for id=1:length(H)

if H(id,1)==10 && H(id,2)==10 && H(id,3)==10 && H(id,4)==10 && H(id,5)==10 &&
H(id,6)==10 && H(id,7)==10 && H(id,8)==10 && H(id,9)==10 && H(id,10)==10;
    H1(id)=10;

elseif H(id,1)==10 || H(id,2)==10 || H(id,3)== 10 || H(id,4)==10 || H(id,
5)==10 || H(id,6)==10 || H(id,7)==10 || H(id,8)==10 || H(id,9)==10 || H(id,
10)==10;
    H1(id)=12;

elseif H(id,1)==H(id,2) && H(id,2)==H(id,3) && H(id,3)==H(id,4) && H(id,
```

```
4)==H(id,5) && H(id,5)==H(id,6) && H(id,6)==H(id,7) && H(id,7)==H(id,8) &&
H(id,8)==H(id,9) && H(id,9)==H(id,10);
   H1(id)= H(id,3);

else
   H1(id)=15;

end

end

for id=1:length(I)

if I(id,1)==10 && I(id,2)==10 && I(id,3)==10 && I(id,4)==10 && I(id,5)==10 &&
I(id,6)==10 && I(id,7)==10 && I(id,8)==10 && I(id,9)==10 && I(id,10)==10;
    I1(id)=10;

elseif I(id,1)==10 || I(id,2)==10 || I(id,3)== 10 || I(id,4)==10 || I(id,
5)==10 || I(id,6)==10 || I(id,7)==10 || I(id,8)==10 || I(id,9)==10 || I(id,
10)==10;
    I1(id)=12;

elseif I(id,1)==I(id,2) && I(id,2)==I(id,3) && I(id,3)==I(id,4) && I(id,
4)==I(id,5) && I(id,5)==I(id,6) && I(id,6)==I(id,7) && I(id,7)==I(id,8) &&
I(id,8)==I(id,9) && I(id,9)==I(id,10);
   I1(id)= I(id,3);

else
   I1(id)=15;

end

end

for id=1:length(J)

if J(id,1)==10 && J(id,2)==10 && J(id,3)==10 && J(id,4)==10 && J(id,5)==10 &&
J(id,6)==10 && J(id,7)==10 && J(id,8)==10 && J(id,9)==10 && J(id,10)==10;
    J1(id)=10;

elseif J(id,1)==10 || J(id,2)==10 || J(id,3)== 10 || J(id,4)==10 || J(id,
5)==10 || J(id,6)==10 || J(id,7)==10 || J(id,8)==10 || J(id,9)==10 || J(id,
10)==10;
    J1(id)=12;

elseif J(id,1)==J(id,2) && J(id,2)==J(id,3) && J(id,3)==J(id,4) && J(id,
4)==J(id,5) && J(id,5)==J(id,6) && J(id,6)==J(id,7) && J(id,7)==J(id,8) &&
J(id,8)==J(id,9) && J(id,9)==J(id,10);
   J1(id)= J(id,3);

else
   J1(id)=15;

end
```

```
end

for id=1:length(K)

if K(id,1)==10 && K(id,2)==10 && K(id,3)==10 && K(id,4)==10 && K(id,5)==10 &&
K(id,6)==10 && K(id,7)==10 && K(id,8)==10 && K(id,9)==10 && K(id,10)==10;
    K1(id)=10;

elseif K(id,1)==10 || K(id,2)==10 || K(id,3)== 10 || K(id,4)==10 || K(id,
5)==10 || K(id,6)==10 || K(id,7)==10 || K(id,8)==10 || K(id,9)==10 || K(id,
10)==10;
    K1(id)=12;

elseif K(id,1)==K(id,2) && K(id,2)==K(id,3) && K(id,3)==K(id,4) && K(id,
4)==K(id,5) && K(id,5)==K(id,6) && K(id,6)==K(id,7) && K(id,7)==K(id,8) &&
K(id,8)==K(id,9) && K(id,9)==K(id,10);
   K1(id)= K(id,3);

else
   K1(id)=15;

end

end

for id=1:length(L)

if L(id,1)==10 && L(id,2)==10 && L(id,3)==10 && L(id,4)==10 && L(id,5)==10 &&
L(id,6)==10 && L(id,7)==10 && L(id,8)==10 && L(id,9)==10 && L(id,10)==10;
    L1(id)=10;

elseif L(id,1)==10 || L(id,2)==10 || L(id,3)== 10 || L(id,4)==10 || L(id,
5)==10 || L(id,6)==10 || L(id,7)==10 || L(id,8)==10 || L(id,9)==10 || L(id,
10)==10;
    L1(id)=12;

elseif L(id,1)==L(id,2) && L(id,2)==L(id,3) && L(id,3)==L(id,4) && L(id,
4)==L(id,5) && L(id,5)==L(id,6) && L(id,6)==L(id,7) && L(id,7)==L(id,8) &&
L(id,8)==L(id,9) && L(id,9)==L(id,10);
   L1(id)= L(id,3);

else
   L1(id)=15;

end

end

for id=1:length(M)

if M(id,1)==10 && M(id,2)==10 && M(id,3)==10 && M(id,4)==10 && M(id,5)==10 &&
M(id,6)==10 && M(id,7)==10 && M(id,8)==10 && M(id,9)==10 && M(id,10)==10 ;
    M1(id)=10;

elseif M(id,1)==10 || M(id,2)==10 || M(id,3)== 10 || M(id,4)==10 || M(id,
```

```
5)==10 || M(id,6)==10 || M(id,7)==10 || M(id,8)==10 || M(id,9)==10 || M(id,
10)==10 ;
    M1(id)=12;

elseif M(id,1)==M(id,2) && M(id,2)==M(id,3) && M(id,3)==M(id,4) && M(id,
4)==M(id,5) && M(id,5)==M(id,6) && M(id,6)==M(id,7) && M(id,7)==M(id,8) &&
M(id,8)==M(id,9) && M(id,9)==M(id,10) ;
   M1(id)= M(id,3);

else
M1(id)=15;

end

end

for id=1:length(A1)

if A1(id)==10 && B1(id)==10 && C1(id)==10 && D1(id)==10 && H1(id)==10 &&
I1(id)==10 && J1(id)==10 && K1(id)==10 && L1(id)==10 && M1(id)==10;
    O(id)=10;

elseif A1(id)==10 || B1(id)==10 || C1(id)== 10 || D1(id)==10 || H1(id)==10 ||
I1(id)==10 || J1(id)==10 || K1(id)==10 || L1(id)==10 || M1(id)==10;
    O(id)=12;

elseif A1(id)==12 || B1(id)==12 || C1(id)== 12 || D1(id)==12 || H1(id)==12 ||
I1(id)==12 || J1(id)==12 || K1(id)==12 || L1(id)==12 || M1(id)==12 ;
    O(id)=12;

elseif A1(id)==B1(id) && B1(id)==C1(id) && C1(id)==D1(id) && D1(id)==H1(id)
&& H1(id)==I1(id) && I1(id)==J1(id) && J1(id)==K1(id) && K1(id)==L1(id) &&
L1(id)==M1(id);
   O(id)= C1(id);

else
   O(id)=15;

end

end

%% save O as a mat file to use to plot parameter planes


% plot parameter planes
function mat= vec2mat(vec,x)

load('O.mat')

x=51; % m is the first set of parameters that you loop through
vec=O(1:2601); % 2601 represents the number of parameter combinations
y=(length(vec))/x; % n is the second set of parameters that you loop through
mat=(reshape(vec,x,y));
```

```
n=1:2:101; % parameter 1 to plot
m=1:1:51;  % parameter 2 to plot

pcolor(n,m,mat) % pcolor(x,y,c) x=m, y=n and c=mat
plotbrowser
end
```