```
clear
syms f(x1,x2) x1 x2 g1(x1,x2) g2(x1,x2) lambda1 lambda2

f(x1,x2) = x1^2 + x2;
dfx1 = diff(f,x1);
dfx2 = diff(f,x2);
fhess = hessian(f)

% g1(x1,x2) = -(x1^2 + x2^2 - 9);
% dg1x1 = diff(g1,x1);
% dg1x2 = diff(g1,x2);

g2(x1,x2) = x1 + x2 - 1;
dg2x1 = diff(g2,x1);
dg2x2 = diff(g2,x2);
g2hess = hessian(g2);

%x1 = 0.5;
%x2 = 0.5;
% dfx1 = double(subs(dfx1));
% dfx2 = double(subs(dfx2));
% dg1x1 = double(subs(dg1x1));
% dg1x2 = double(subs(dg1x2));
% dg2x1 = double(subs(dg2x1));
% dg2x2 = double(subs(dg2x2));

%solution = solve(dfx1,dfx2)
solution = solve(dfx1 - lambda2*dg2x1 == 0,...
    dfx2 - lambda2*dg2x2 == 0, g2 == 0);
lambda2 = double(solution.lambda2)
green_x1 = solution.x1
green_x2 = solution.x2

laplac = fhess - lambda2*g2hess
eigenvalues = eig(laplac)
% Positive semi-definite

jacobian = [double(dg2x1), double(dg2x2)]
y = [-1, 1]
coeff = y*laplac*y'




% The coefficient is always greater than zero, and so we do not have a
% constrained maximum
```

fhess(x1, x2) =

```
[ 2, 0]
[ 0, 0]
```

```
lambda2 =

     1


green_x1 =

1/2


green_x2 =

1/2


laplac(x1, x2) =

[ 2, 0]
[ 0, 0]


eigenvalues(x1, x2) =

 0
 2


jacobian =

     1     1


y =

    -1     1


coeff(x1, x2) =

2
```

Published with MATLAB® R2018b