Optimization Hw 6

Ryan Day

March 27, 2019

```
clear
```

Part A

```
syms f(x1,x2) x1 x2 g(x1,x2) lambda
f(x1,x2) = 4*x1 - 3*x2 + 2*x1^2 - 3*x1*x2 + 4*x2^2;
df1 = diff(f,x1);
df2 = diff(f,x2);

g(x1,x2) = 2*x1 - 1.5*x2 - 5;
dg1 = diff(g,x1);
dg2 = diff(g,x2);

solution = solve(df1 - lambda*dg1 == 0, df2 - lambda*dg2 == 0, g ==
 0);
lambda1 = double(solution.lambda)
x1_a = double(solution.x1)
x2_a = double(solution.x2)
minimum_value_a = double(f(x1_a,x2_a))

% The optimum agrees with the graphical optimum.
```

```
lambda1 =

     7


x1_a =

    2.5000


x2_a =

     0


minimum_value_a =

   22.5000
```

Part B

```
syms f(x1,x2) x1 x2 g(x1,x2) lambda
f(x1,x2) = 4*x1 - 3*x2 + 2*x1^2 - 3*x1*x2 + 4*x2^2;
df1 = diff(f,x1);
df2 = diff(f,x2);

g(x1,x2) = 2*x1 - 1.5*x2 - 5.1;
```

```
dg1 = diff(g,x1);
dg2 = diff(g,x2);

solution = solve(df1 - lambda*dg1 == 0, df2 - lambda*dg2 == 0, g ==
 0);
lambda = double(solution.lambda);
x1_b = double(solution.x1)
x2_b = double(solution.x2)
minimum_value_b = double(f(x1_b,x2_b))

real_change_f = minimum_value_b-minimum_value_a
approximated_change_f = lambda1*(5.1-5)
difference = real_change_f-approximated_change_f

% The lagrange multiplier predicts the change in objective pretty
% accurately, there is only a difference of 0.005 in the approximation
 to
% the real.
```

```
x1_b =

    2.5500


x2_b =

     0


minimum_value_b =

   23.2050


real_change_f =

    0.7050


approximated_change_f =

    0.7000


difference =

    0.0050
```

Published with MATLAB® R2018b

```
syms f(x1,x2,x3) x1 x2 x3 g1(x1,x2,x3) g2(x1,x2,x3) lambda1 lambda2
f(x1,x2,x3) = x1^2 + 2*x2^2 + 3*x3^2;
dfx1 = diff(f,x1);
dfx2 = diff(f,x2);
dfx3 = diff(f,x3);

g1(x1,x2,x3) = x1 + 5*x2 - 12;
dg1x1 = diff(g1,x1);
dg1x2 = diff(g1,x2);
dg1x3 = diff(g1,x3);

g2(x1,x2,x3) = -(-2*x1 + x2 - 4*x3 + 18);
dg2x1 = diff(g2,x1);
dg2x2 = diff(g2,x2);
dg2x3 = diff(g2,x3);

solution = solve(dfx1 - lambda1*dg1x1 - lambda2*dg2x1 == 0,...
    dfx2 - lambda1*dg1x2 - lambda2*dg2x2 == 0,...
dfx3 - lambda1*dg1x3 - lambda2*dg2x3 == 0, g1 == 0, g2 == 0);

lambda1 = double(solution.lambda1)
lambda2 = double(solution.lambda2)
x1 = double(solution.x1)
x2 = double(solution.x2)
x3 = double(solution.x3)
function_value = double(f(x1,x2,x3))
```

```
lambda1 =

    1.9170


lambda2 =

    3.7585


x1 =

    4.7170


x2 =

    1.4566


x3 =

    2.5057
```

```
function_value =

   45.3283
```

Published with MATLAB® R2018b

```
clear
```

Part A

```
syms f(x1,x2) x1 x2 g1(x1,x2) g2(x1,x2) g3(x1,x2) lambda1 lambda2 lambda3

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

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

g2(x1,x2) = -(x1 + x2^2 - 1);
dg2x1 = diff(g2,x1);
dg2x2 = diff(g2,x2);

% G3 is not binding so we leave it out
% g3(x1,x2) = x1 + x2 - 1
% dg3x1 = diff(g,x1);
% dg3x2 = diff(g,x2);

x1 = -2.3723;
x2 = -1.8364;
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 - lambda1*dg1x1 - lambda2*dg2x1 == 0,...
    dfx2 - lambda1*dg1x2 - lambda2*dg2x2 == 0, g1 == 0, g2 == 0);
lambda1 = double(solution.lambda1)
lambda2 = double(solution.lambda2)

% The point is a local optimum since both of the lambdas are binding
```

```
lambda1 =

    0.7785
    0.7785
    0.7785
    0.7785


lambda2 =

    1.0508
    1.0508
```

```
    1.0508
    1.0508
```

Part B

```
clear
syms f(x1,x2) x1 x2 g1(x1,x2) g2(x1,x2) g3(x1,x2) lambda1

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

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

% G2 and G3 are not binding so we leave them out

x1 = -2.5;
x2 = -1.6583;
dfx1 = double(subs(dfx1));
dfx2 = double(subs(dfx2));
dg1x1 = double(subs(dg1x1));
dg1x2 = double(subs(dg1x2));

solution = solve(dfx1 - lambda1*dg1x1 == 0,...
    dfx2 - lambda1*dg1x2 == 0, g1 == 0);

lambda1 = double(solution.lambda1)

% There isn't a solution for lambda that satisfies the equations,
% therefore the point is not a local optimum
```

```
lambda1 =

  0×1 empty double column vector
```

Published with MATLAB® R2018b

```
% Part C
clear
syms f(x1,x2) x1 x2 g2(x1,x2) g3(x1,x2) lambda2

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

g2(x1,x2) = -(x1 + x2^2 - 1);
dg2x1 = diff(g2,x1);
dg2x2 = diff(g2,x2);

% G3 is not binding so we leave it out

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

lambda2_final = lambda2(3)
x1_final = x1(3)
x2_final = x2(3)
optimum_value = double(f(x1_final,x2_final))
```

```
lambda2_final =

    0.4516


x1_final =

   -0.2258


x2_final =

   -1.1072


optimum_value =

   -1.0562
```

Published with MATLAB® R2018b

```
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

```
clear
syms f(h,w) h w g1(h,w) g2(h,w) g3(h,w) lambda1 lambda2 lambda3 lambda4 lambda5

f(h,w) = -(h*w);
dfh = diff(f,h);
dfw = diff(f,w);

g1(h,w) = -(h - 0.6);
dg1h = diff(g1, h);
dg1w = diff(g1, w);

g2(h,w) = -(w - 0.8);
dg2h = diff(g2, h);
dg2w = diff(g2, w);

hyp = sqrt(0.4^2+0.6^2);
side1 = 0.4;
side2 = 0.6;
a = hyp/sqrt((w/2)^2+h^2);
b = side1/(w/2);
c = side2/(h);

g3(h,w) = a-b;
dg3h = diff(g3, h);
dg3w = diff(g3, w);

g4(h,w) = (b-c);
dg4h = diff(g4, h);
dg4w = diff(g4, w);

g5(h,w) = (c-a);
dg5h = diff(g5, h);
dg5w = diff(g5, w);

solution = solve(dfh - lambda1*dg1h - lambda2*dg2h == 0 - lambda3*dg3h
 - lambda4*dg4h - lambda5*dg5h,...
    dfw - lambda1*dg1w - lambda2*dg2w - lambda3*dg3w - lambda4*dg4w -
 lambda5*dg5w == 0, g1 == 0, g2 == 0, g3 == 0, g4 ==0, g5 == 0)
lambda1 = solution.lambda1
lambda2 = solution.lambda2
lambda3 = solution.lambda3
lambda4 = solution.lambda4
lambda5 = solution.lambda5

h = solution.h/2
w = solution.w/2


solution =

  struct with fields:
```

```
          h: [1×1 sym]
    lambda1: [1×1 sym]
    lambda2: [1×1 sym]
    lambda3: [1×1 sym]
    lambda4: [1×1 sym]
    lambda5: [1×1 sym]
          w: [1×1 sym]
```

```
lambda1 =
```

$$\frac{4}{5}$$

```
lambda2 =
```

$$\frac{3}{5}$$

```
lambda3 =
```

$$0$$

```
lambda4 =
```

$$0$$

```
lambda5 =
```

$$0$$

```
h =
```

$$\frac{3}{10}$$

```
w =
```

$$\frac{2}{5}$$

Published with MATLAB® R2018b