ECON 715, Optimization

Michael Bar, San Francisco State University

Table of Contents

Unconstrained Optimization

```
clear %Clearing memory
close all
```

Multiproduct competitive firm

Consider a firm sells 2 different products, **taking the market prices as given**. The quantities of these products are Q_1, Q_2 and prices P_1, P_2. The revenue is therefore: $R(Q_1, Q_2) = P_1Q_1 + P_2Q_2$. The cost function is $C(Q_1, Q_2) = 2Q_1^2 + Q_1Q_2 + 2Q_2^2$.

Declaring variables, and defining the revenue, cost and profit functions.

```
syms Q1 Q2 P1 P2 %Declaring symbolic variables
R = P1*Q1 + P2*Q2; %Revenue
C = 2*Q1^2 + Q1*Q2 + 2*Q2^2; %Cost function
Pai = R - C; %Profit
```

Solving for optimal quantities:

```
[Q1_opt, Q2_opt] = solve(gradient(Pai,[Q1,Q2])==0, [Q1,Q2]);
display([Q1_opt;Q2_opt])
```

ans =

$$\begin{pmatrix} \frac{4\,P_1}{15} - \frac{P_2}{15} \\ \frac{4\,P_2}{15} - \frac{P_1}{15} \end{pmatrix}$$

Substituting market prices:

```
Supply = subs([Q1_opt, Q2_opt], [P1,P2], [12,18])
```

Supply = $\begin{pmatrix} 2 & 4 \end{pmatrix}$

You can change the market prices, and see how quantities change:

```
Supply = subs([Q1_opt, Q2_opt], [P1,P2], [15,15])
```

```
Supply = (3   3)
```

Multiproduct Monopoly

Consider a monopoly that sells 2 different products to two markets, with demands:

$$P_1(Q_1, Q_2) = 55 - Q_1 = Q_2$$
$$P_2(Q_1, Q_2) = 70 - Q_1 - 2Q_2$$

Defining the demand, revenue, cost and profit functions:

```
P1(Q1,Q2) = 55 - Q1 - Q2; %Demand in market 1
P2(Q1,Q2) = 70 - Q1 - 2*Q2; %Demand in market 2
R = P1*Q1 + P2*Q2; %Revenue
C = Q1^2 + Q1*Q2 + Q2^2; %Cost function
Pai = simplify(R - C); %Profit
```

Solving for optimal quantiies, equilibrium prices and maximal profit:

```
[Q1_opt, Q2_opt] = solve(gradient(Pai)==0);
P1_opt = double(P1(Q1_opt, Q2_opt));
P2_opt = double(P2(Q1_opt, Q2_opt));
Pai_opt = double(Pai(Q1_opt, Q2_opt));

disp('Quantities')
```

```
Quantities
```

```
double([Q1_opt;Q2_opt])
```

```
ans = 2×1
    8.0000
    7.6667
```

```
disp('Prices')
```

```
Prices
```

```
double([P1_opt;P2_opt])
```

```
ans = 2×1
   39.3333
   46.6667
```

```
disp('Profit')
```

```
Profit
```

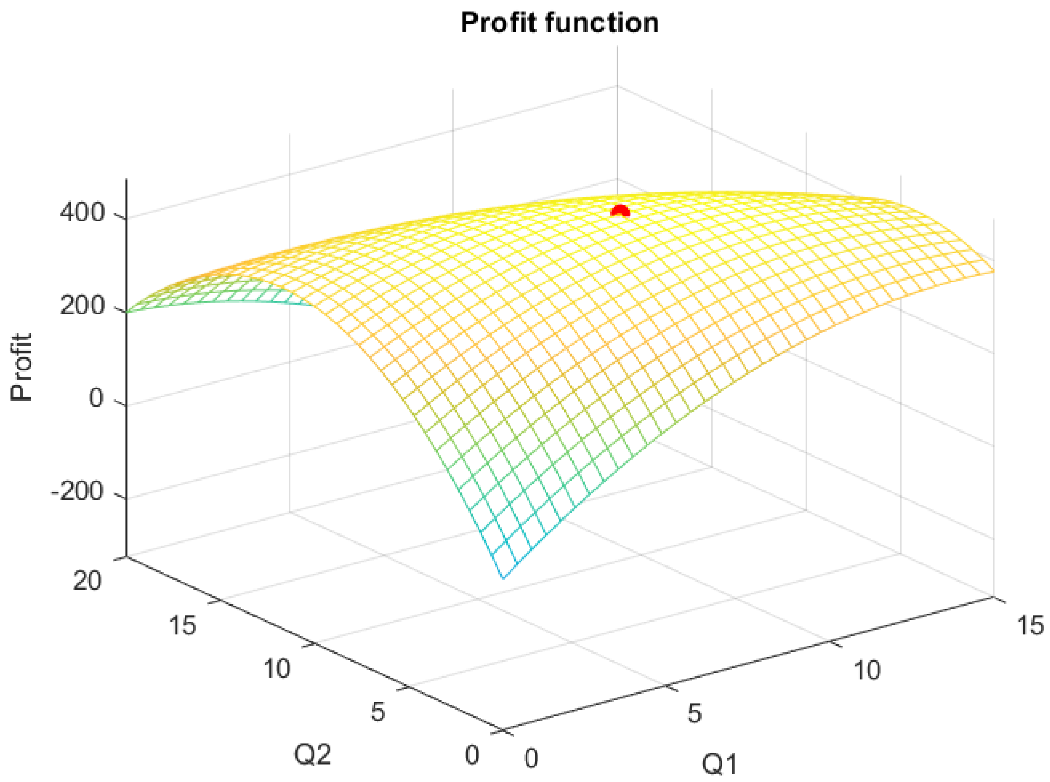
```
disp(Pai_opt)
```

```
  488.3333
```

Plotting the profit function:

```
fmesh(Pai, [0 15 0 20] )
% fsurf(Pai, [0 15 0 20] )
```

```
hold on
plot3(Q1_opt, Q2_opt, Pai_opt,'.r','MarkerSize',25) %Point of tangency - stationary point
hold off
title('Profit function')
xlabel('Q1')
ylabel('Q2')
zlabel('Profit')
```



Constrained Optimization

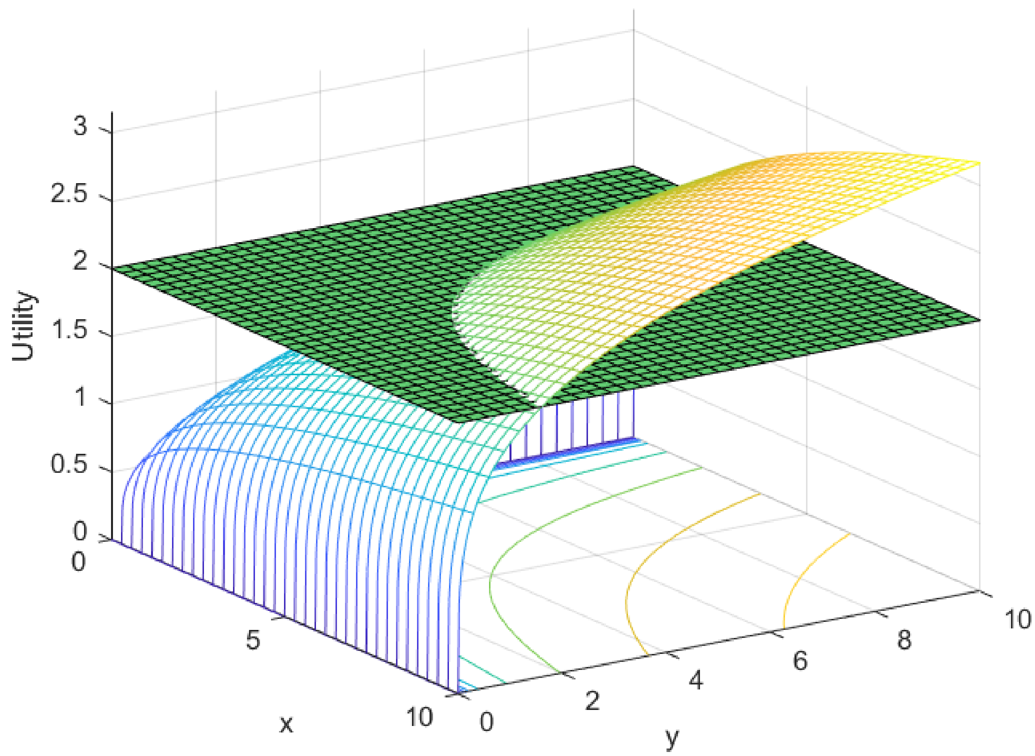
Utility function visualization

Plotting the graph of $u(x, y) = x^{0.25}y^{0.25}$, a **strictly concave** function.

```
syms x y %Declaring symbolic variables
a = 0.25; b = 0.25;
u(x,y) = x^a*y^b; %Cobb-Douglas utility function
fmesh(u, [0 10 0 10], 'ShowContours','on')
hold on
xlabel('x')
ylabel('y')
zlabel('Utility')
```

Slicing the utility function, to illustrate indifference curves.

```
fsurf(2, [0 10 0 10] )
hold off
view([56.40 23.40]) %Rotate angle
```

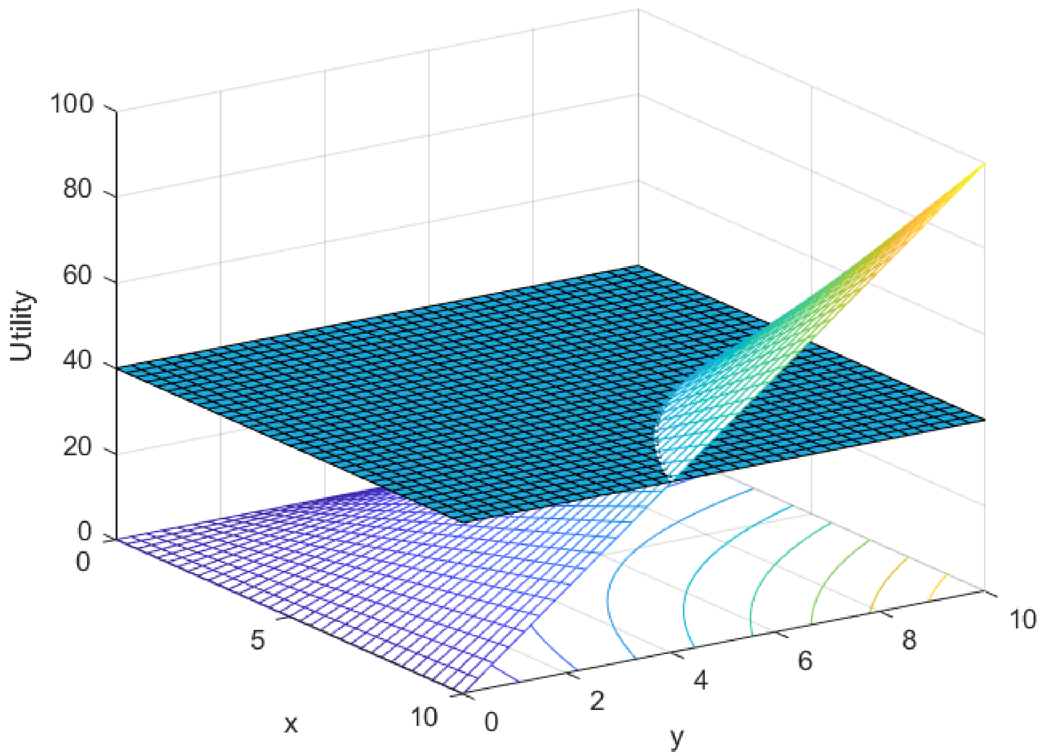


Plotting the graph of $u(x, y) = xy$, not concave, but **strictly quisiconcave** function.

```
syms x y %Declaring symbolic variables
a = 1; b = 1;
u(x,y) = x^a*y^b; %Cobb-Douglas utility function
fmesh(u, [0 10 0 10], 'ShowContours','on')
hold on
xlabel('x')
ylabel('y')
zlabel('Utility')
```

Slicing the utility function, to illustrate indifference curves.

```
fsurf(40, [0 10 0 10] )
hold off
view([56.40 23.40]) %Rotate angle
```



Notice that the level curves (indifference curves) of a strictly quisiconcave utility function have the same shape as the level curves of a strictly concave utility function, and upper contour sets are strictly convex in bot cases. Thus, strict concavity of utility fucntions is not necessary for the "right shape" of indifference curves.

Consumer's demand

Setting up the lagrange:

```
syms x y p_x p_y I k a %Declaring symbolic variables
u(x,y) = a*log(x) + (1-a)*log(y); %Utility function defined
L(x,y,k) = u(x,y) - k*(p_x*x + p_y*y - I); %Lagrange function defined
% k is the Lagrange multiplier
```

Solving for consumer demand:

```
[x,y,k] = solve(gradient(L,[x,y,k])==0, [x,y,k])
```

x =

$$\frac{\mathrm{I}\, a}{p_x}$$

y =

$$-\frac{\mathrm{I}\ (a-1)}{p_y}$$

k =

$$\frac{1}{I}$$

Substituting values of p_x, p_y, I:

```
demand = subs([x,y],[a,p_x,p_y,I],[0.5,3,0.75,100])
```

demand =

$$\left(\frac{50}{3} \quad \frac{200}{3}\right)$$

```
double(demand)
```

```
ans = 1×2
   16.6667   66.6667
```