Long-Run Firm Equilibrium Test Chunk

true

24 Mar 2023

Review: The Firm's Three Problems:

1. ~~Input cost minimization~~
2. ~~Output profit maximization~~
3. Long-Run Readjustment

Perfectly Competitive Input Markets

1. Many, many buyers
2. Homogeneous inputs
3. Perfect information
4. Free entry/exit
5. Firms are price takers (1-3)
6. Profits/lossesare temporary

The Producer's Input-Hiring Problem

1. Objective: Maximize Profit
2. Endogenous (choice) variable(s): Capital, Quantity, Labor
3. Exogenous variable(s): Technology, Market Price, Wage, Return to Capital

Initial Equilibrium (Short Run)

Let's set up the initial equilibrium with the initial values of the market price (P) and quantity (Q__m); planned levels of output (Q__0), labor (L__0), and capital (K__0); actual short-run labor (L), capital (K), and output (Q), and profit (Pi)

```
library(Rsolnp); library(nleqslv)
A = 100; a = 1/3; y1 = 3227.486; w = 10; r = 20
control = list(trace = 0)
P = NULL; Qm = NULL; Q0 = NULL; L0 = NULL; K = NULL; Q = NULL; L = NULL; Pi = NULL
```

When we last left our representative competitive firm, they operated in a market where $P = 0.5$ and maximized profit at $Q_i^* = \sqrt{\frac{31{,}250{,}000}{3}} \approx 3,227.5$ earning a total profit of \$575.8 (whew, whoever wrote this example could have come up with friendlier numbers!). Let's remind ourselves of how to solve the market equilibrium and set the first element of each market variable (`X[1]`) to the initial equilibrium.

```
supply <-function(x) 10000000*x[1] - x[2]
demand <- function(x) 6000000 - 2000000*x[1] - x[2]
marketEq <- nleqslv(c(2,2), function(x) c(supply(x), demand(x)))
P[1] = marketEq$x[1]; Qm[1] = marketEq$x[2]; Q0[1] = 2500
cbind(P, Qm, Q0)
```

```
##        P    Qm   Q0
## [1,] 0.5 5e+06 2500
```

The firm expected to produce 2,500 units and planned its capacity likewise. This meant building 25 units of capital and planning to hire 25 units of labor.

```
cost <- function(x) w*x[1] + r*x[2]
output <- function(x) A*x[1]^a * x[2]^(1 - a)
costMin <- solnp(pars = c(1, 1), fun = cost, ineqfun = output,
  ineqLB = Q0[1], ineqUB = Inf, LB = c(0, 0), control = control)
L0[1] = costMin$pars[1]; K[1] = costMin$pars[2]
cbind(L0, K)
```

```
##      L0  K
## [1,] 25 25
```

Based on this short run guess and the market price of $0.50, the firm chose its output (and, hence its short run variable input) to maximize profits.

```
labor <- function(x) x^3 / (K[1] ^ 2 * A^3)
loss <- function(x) -(P[1]*x - (w*labor(x) + r*K[1]))
piMax <- solnp(pars = 1, fun = loss, LB = 0)
```

```
##
## Iter: 1 fn: -575.8287     Pars:  3227.49095
## Iter: 2 fn: -575.8287     Pars:  3227.49095
## solnp--> Completed in 2 iterations
```

```
Q[1] = piMax$pars; L[1] = labor(piMax$pars); Pi[1] = -tail(piMax$values, 1)
cbind(Q, L, Pi)
```

```
##             Q        L       Pi
## [1,] 3227.491 53.79168 575.8287
```

From this outcome, the firm could lower its costs (and slightly increase profits) in the long run by increasing its capacity (choosing a level of capital that optimizes for 3,227.5 instead of 2,500 as initially expected).

Maybe this happens because independent startups enter a profit-making industry (exit a loss-making industry) or maybe this happens as capital pours into existing firms from investors looking for above-market returns - who knows!

Either way, let's solve it!


WOOHOO!