Moving Economic Models from the Chalk Board to the Computer: A Computer Based Assignment Based on a Dynamic Cournot Model

Tim Kochanski
Portland State University, USA

Abstract

This paper presents an iterative model, programmed in Mathematica, which solves time paths for repeated Cournot games allowing us to see how output, price, profits, and market share in the two-firm case change over time when one firm experiences per turn marginal cost reductions. By adjusting the marginal cost reduction rate for one firm and iterating, students can explore the various solutions and gain a better understanding of how the variables in the model diverge over time and the properties of that divergence. More generally, students gain experience designing models and programming in Mathematica and furthermore develop a deeper understanding of iteration in simulations.

Introduction

The use of Mathematica in economics has been widely discussed in the literature, both in the context of model development (Varian, 1993; Varian, 1996; Kendrick *et al.*, 2005), and in terms of learning 'the programming skills to develop one's own models and statistical tools [in graduate school and beyond]' (Belsley, 1997). Although Mathematica has become a useful and established program for economic modelling and simulation, little has been written on introducing Mathematica in the economics classroom.

This paper introduces an extension to the typical Cournot equilibrium found in undergraduate and graduate microeconomic and industrial organisation textbooks. The one-round simultaneous solution Cournot game programmed in Mathematica has been introduced in various sources (Kendrick *et al.*, 2005; Uzawa, 2000). I present a Mathematica-based, iterative model that can be used to help students develop an understanding of the Mathematica programming language and the effects that changing costs have on the Cournot model in a repeated game setting. This reapeated game setting introduces students to the concept of iteration, which is at the core of much dynamic modelling and agent-based computational economics and which is growing more and more prevalent in the economics literature but is essentially non-existant in economics textbooks.[1]

Mathematica is a well-known symbolic computational environment that provides 'powerful modeling, graphical, statistical, and programming facilities' (Belsley, 1997). The notebook and assignment used in this project are designed to allow students to develop a better understanding of how, in a Cournot setting, an innovating firm's marginal cost reductions lead to changes in output, price, market share and profit for each firm in the industry.

With Mathematica's graphing capabilities students gain a better understanding of how iteration turns static models into dynamic ones. With graphical output students are able to examine time paths for the affected variables in the model. Students also learn to become better programmers as they learn by doing the programming assignments. Finally, students develop valuable model building and testing skills so that they can develop notebooks, routines or processes of their own to help explore the constraints, parameters and rates of change within such models. These skills are beneficial in graduate school and beyond and ultimately expand the student's understanding of the economic modelling process.

In the following sections, I will develop the theoretical foundation for the Cournot model and implementation within Mathematica. I will then suggest a Cournot modelling assignment with learning objectives and discuss student responses to the assignment. Finally, I will provide a conclusion.

Background: computational modelling of the Cournot game

The Cournot model and duopoly are usually introduced in introductory and intermediate texts as a more extreme case of oligopoly. The key point being that the actions of one firm affect the performance of another firm in an industry with so few producers. Typical classroom examples include oil cartels, the kinked demand curve, and all the problems associated with trying to organise and maintain a system of collusion. Other examples may include the airline industry regulation and deregulation and how the interactions of a changing business environment with differing cost structures can drastically shift each firm's respective market share, as was the case with Pan Am in the 1980s after deregulation in the airline industry.

Within the computational environment, students can easily adjust the costs of one firm vs. the costs of another firm and solve the model repeatedly over a few time periods arriving at the resulting levels of output, price, profit and market share. Students can also easily adjust the slope and intercept of the demand curve or the initial levels of marginal cost for each firm if they wish to explore behavior at both the elastic and inelastic ends of the demand curve. This particular assignment gives students a better understanding of how one firm's actions in a duopoly can affect the other firm. This creates a deeper understanding of how Pan Am could be stripped of its market share and ultimately its existence during the 1980s. Additionally, students see the powerful price cutting incentive that is always at work in a cartel such as OPEC.

Building the Cournot model in Mathematica: the one round game

The model of Cournot quantity competition was introduced by Augustin Cournot in his 1838 book, *Researches into the Mathematical Principles of the Theory of Wealth*. For simplicity, in this project I impose several common assumptions when constructing the model. I assume a duopoly industry structure where both firms produce homogenous goods. Additionally, I assume both firms start with identical marginal costs but firm 2 experiences gradual decreasing marginal costs in repeated games. Finally, the Cournot solution assumes each firm chooses its optimal level of output as a best response to the hypothesised output of its competitor, and each firm's belief about its competitors output is correct.

An assignment such as this might appear in the middle of an intermediate micro course so that students will have had a sufficient introduction to the Mathematica programming language and some practice at writing programs. Students should be familiar with the symbolic construction of the Cournot solution from lecture so that implementing the model in Mathematica is simply a matter of converting textbook notation into the standard symbolic notation of Mathematica. With even minimal practice this process becomes quite easy.2

Since the market consists of only two firms, the total market quantity supplied is the sum of firm 1 and firm 2's individual outputs.

```
Q = q1 + q2;
```

Next, for my purposes, which are numerical, I choose intercept and slope values then specify a functional form for the inverse market demand. It is linear and downward sloping in this case.

```
a = 10.;
b = 1.;
Price = a - b*Q;
```

Now I define our cost structure by choosing the marginal costs for each firm. In this case I choose initial costs per unit as equal and pick a relatively high initial level ($6 per unit) so that I am sure to operate on the elastic end of the demand function. This typifies high-cost industries such as airlines and ensures a more dramatic effect on the variables I wish to observe when I begin to lower costs for one firm. The students will discover this through the homework assignment. The other commands simply create lists to store our variables in.

```
MC1 = 6.;
MC2 = 6.;
list1 = {x1};
list2 = {x2};
Null
```

Within the game theory context we say that a firm's payoff is the profit it receives from operation. Profit for firm 2 is computed here as the quantity firm 2 produces multiplied by the price minus marginal cost per unit produced. This is the profit function.

```
eqPr2 = Profit2 == q2*(P[q1, q2] - MC2);
```

This line simply substitutes the demand function from above for the more general p[q1,q2].

```
eqPr2 = Expand[% /. P[q1, q2] -> Price]
```

Now, I maximise the profit function. This segment of code first solves the first order condition of the profit function by taking the derivative and solving for zero. The second line solves algebraically in terms of firm 2's quantity since the Cournot game is a quantity choice game. Finally I define this function as firm 2's reaction function.

```
focPr2 = D[eqPr2, q2]
temp2 = Solve[focPr2, q2]
React2 = R2[q1] == q2 /. temp2[[1]]
```

The following output is generated from the previous line of input.

```
R2[q1] == -0.5 (-8. + 1. q1)
```

This is firm 2's reaction function. Notice that firm 2's reaction function depends on the quantity chosen by firm 1. In the Cournot setting, this is firm 2's best response to firm 1's output decision.

Given firm 2's reaction function I can now derive firm 1's reaction function. Everything is defined exactly as above.

```
eqPr1 = Profit1 == q1*(P[q1, q2] - MC1);
eqPr1 = Expand[% /. P[q1, q2] -> Price]
```

```
focPr1 = D[eqPr1, q1]
temp1 = Solve[focPr1, q1];
React1 = R1[q2] == Simplify[q1 /. temp1[[1]]]
```

The following output is generated from the previous input.

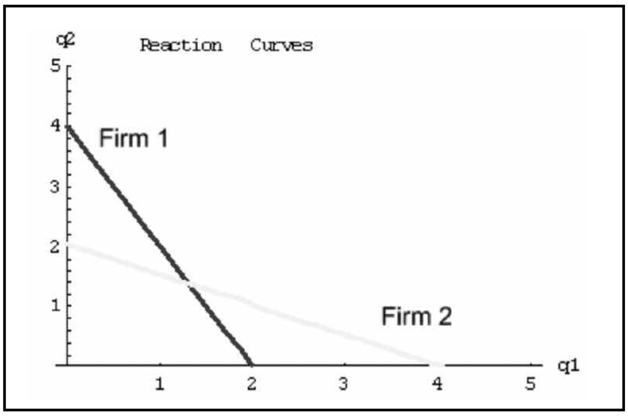
```
R1[q2] == 4 - 0.5 q2
```

This is firm 1's reaction function. If I distribute and simplify firm 2's reaction function above I see they are exactly the same.

Finally, I plot out the reaction functions of both firms. The Cournot equilibrium is designated by the simultaneous solution to the two reaction functions.

```
reactPlot=Plot[{q2/.Solve[focPr1/.{a->10,b-
>1,MC1->6}, q2][[1]],q2/.Solve[focPr2/.{a-
>1,b->1,MC2->6},q2][[1]]},
{q1,0,10}, PlotRange->{0,10},PlotStyle-
>{{RGBColor[0, 0, 1],
Thickness[0.01]},{RGBColor[1,0,0],Thickness
[0.01]}},AxesLabel ->
{”q1”,”q2”},PlotLabel->”Reaction Curves
blue=firm1”]
```

The following output is generated from the previous input:



This is the classic mapping of the reaction functions for firm 1 and firm 2. On the axes are the best response quantities to be produced by each firm. The firms decide what amount to produce by selecting the quantity that corresponds to the intersection of the reaction functions.

To find the exact point where the reaction functions intersect I solve the simultaneous move Cournot game algebraically. I simply set the reaction functions equal to each other and find the point at which they intersect to get the optimal outputs for firm 1 and firm 2. Once I have these values we can solve for the market quantity (firm 1 + firm 2), the equilibrium price (P = 10 – Q), and the profit each firm can expect (e.g. Profit1 = P*q1 – MC1*q1). I also compute the market share of each firm by dividing the firm's output at any time by the market output for that time period (e.g. market share 1 = q1/Q). This is the code that solves for the system of two equations.

```
cournotQ = Simplify[Solve[{React1/.R1[q2]-
>q1,React2/.     R2[q1] -> q2}, {q1,q2}]];
q1cournot = q1 /. %[[1]]
q2cournot = q2 /. %%[[1]]
```

As mentioned above I can now solve for market quantity (firm 1 + firm 2).

```
Qcournot = Q /. {q1 -> q1cournot, q2 ->
q2cournot}
```

I can also solve for the equilibrium price (P = 10 – Q).

```
Pcournot = Simplify[Price/.{q1-
>q1cournot,q2->q2cournot}]
```

I can solve for the profit each firm can expect (e.g. Profit1 = P*q1 – MC1*q1).

```
CournotProfit1 =
Simplify[q1cournot*(Pcournot - MC1)]

CournotProfit2 =
Simplify[q2cournot*(Pcournot - MC2)]
```

And finally, I can compute the market share of each firm by dividing the firm's output at any time by the market output for that time period (e.g. market share 1 = q1/Q).

```
share1 = q1cournot/ Qcournot
share2 = q2cournot/ Qcournot
```

The following output is generated from code similar to all previous input. As you might expect, in a duopoly setting where each firm has identical costs, each firm will produce the same level of output, generate the same amount of profit, and share the same percentage of the market. Given these preliminary results we may now adjust several aspects of the model and observe the resulting outcome.

```
”Cournot quantity for firm 1 then firm 2”
1.33333
1.33333
”Total Cournot market quantity”
2.66667
”Market Share for firm 1 then firm 2”
0.5
0.5
”Cournot market price”
7.33333
”Cournot profit for firm 1 then firm 2”
1.77778
1.77778
```

The Cournot simulation: repeating the game with iteration

The above scenario is well established in the previous literature (Kendrick *et al.* (2005); Uzawa (2000)). Now, that I have established this nice static solution for our two-firm game let's introduce an iterative process that repeats the game. Within this iterative setting we are able to change key variables (marginal cost reduction rate, slope, intercept and initial marginal costs in our case) and chart out the time paths for the solutions we found above in the static example.

The assignment begins with a gradually changing marginal cost for firm 2. For theoretical purposes, assume that firm 2 decides to establish an R&D or systems engineering department that is able to develop technology that will lower production costs in each round. The managers have crunched the numbers and they think they can lower costs in each time period by 5% from the period before.[3] This is a significant cost reduction over time to make the results more dramatic. We project some time paths for how price, quantity, profits and market share will change for the two firms and the industry.

Recall, I am simply repeating the above process which gave us a Cournot equilibrium price and quantity and allowed us to determine industry price and quantity as well as each firm's profit and market share. I am simply repeating the game a number of times, each time lowering firm 2's costs by 5% from the previous period and charting a time path of the results. In essence I am imbedding the previous algorithm for the solution within a 'For' loop to iterate the results for 12 periods, each time lowering the marginal cost of firm 2. (See appendix A for the link to the Mathematica code contained in cournot.nb).

Initiating the program results in a long series of output the begininning and end of which are the following:

```
Round1
  “Cournot quantity for firm 1 then firm 2”
  1.33333
  1.33333
  “Total Cournot market quantity”
  2.66667
  “Market Share for firm 1 then firm 2”
  0.5
0.5
  “Cournot market price”
  7.33333
  “Cournot profit for firm 1 then firm 2”
  1.77778
  1.77778
  “Next period cost for firm 2”
  5.7
  //////////////////////
      /////////////////////
         ////////////////////

Round 12
  ‘Cournot quantity for firm 1 then firm 2”
  0.470934
  3.05813
  “Total Cournot market quantity”
  3.52907
  “Market Share for firm 1 then firm 2”
  0.133444
  0.866556
  “Cournot market price”
  6.47093
  “Cournot profit for firm 1 then firm 2”
  0.221778}
  9.35218}
  “Next period cost for firm 2”
  3.24216
```

Theory tells us that if there are decreasing costs in an industry then supply shifts out leading to lower market price levels and greater levels of output. With the above analysis we can examine how the cost reduction affected market price and quantity. We can see that price did in fact fall and that quantity increased. Graphically we get Figure 1. You might imagine a supply curve shifting out and the points along the demand curve as the equilibrium path that is charted as a result.

As the market price falls and market quantity increases, recall that firm 2 is experiencing lower costs while firm 1's costs are fixed. From our oligopoly theory, this results in firm 2 gaining more and more market share with each round. The data for periods one and 12 above demonstrate this. Figure 2 shows a time series of the market shares.

Given these changes in market share, we see how profits for the two firms diverge under the given changing cost structure in Figure 3.

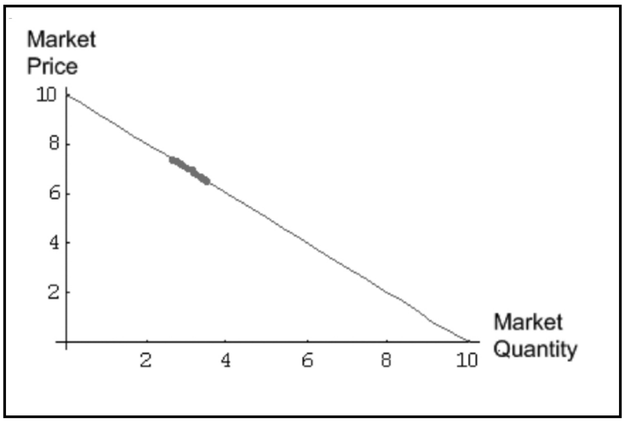


Figure 1

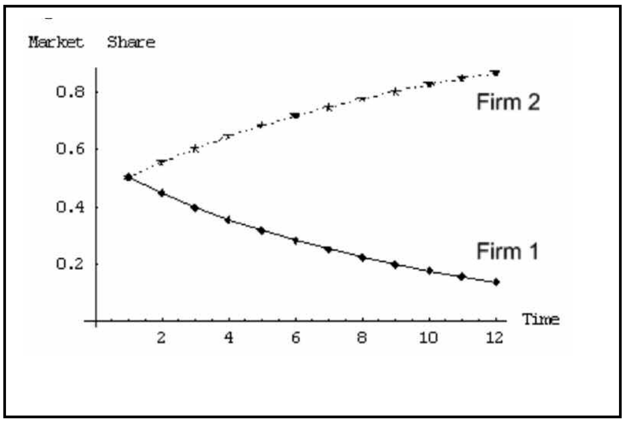


Figure 2

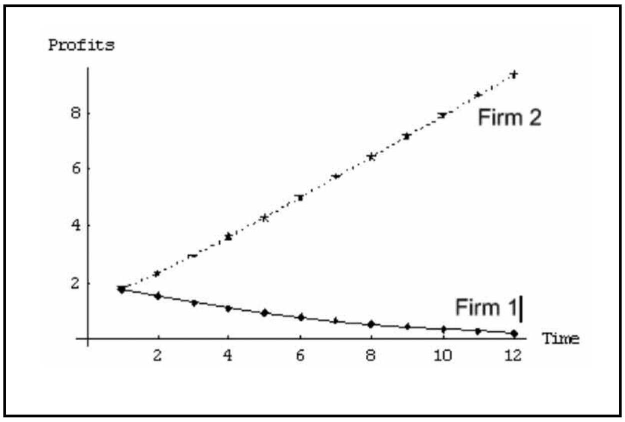


Figure 3

At this point the student is asked to make various alterations to the program and to observe the results. The first alteration involves running the simulation with cost decreasing at both 0.1% and at 10%. At 0.1% the student observes that the change in market share is minimal as is the variation in the other variables. At 10% the student observes that firm 1 experiences negative market share and quantity. The interpretation is that marginal cost is above the market price so the firm must exit the market.

Next, students are asked to comment on the realtionship between marginal cost reductions and market concentration. It becomes clear that firms that are able to lower costs improve their market share, which in turn leads to a more concentrated industry and increased profits. The students are also asked to adjust the slope of the demand curve. Initially, the demand curve has a slope of –1. After adjusting the slope to –3 and –0.25 students are able to determine that the slope of the demand curve does not appear to affect the time path for market share although, as the demand curve becomes more elastic, the cost-reducing firm's profits increase more dramatically. This is because with high initial marginal costs, supply intersects the elastic end of the demand function and by making it flatter it becomes even more elastic. Finally, students are asked to decrease the marginal cost from 6 to 0.5 so that supply intersects the inelastic end of the demand curve. From here they observe a smaller affect on both profit and market share divergence although both firms achieve higher levels of profit.

This model could be used in an upper level class in industrial organisation or microeconomics. Given Mathematica's popularity students can be provided with a Mathematica notebook via e-mail or a class website (blackboard, etc.) and complete the project in the department lab or in another university lab if the universitity has a licence for Mathematica. I learned and use Mathematica because my graduate institution had it installed on all campus computers. It is likely that students would need to be provided with a significant portion if not all of the code.

I should also mention that there are several alternatives to Mathematica and it is likely that any university would have a licence for at least one of them. These alternatives include commercial packages such as Maple, Matlab and Derive as well as freeware packages such as Octave and Mupad. There is also a growing number of papers and web-based content that relies on simulations coded in Microsoft Excel, which is likely to be installed on nearly every PC.

One advantage of using Mathematica is that its notebook layout is very user friendly and functions much like a word processor. Questions and directions can easily be placed within the code itself for any project. A typical set of questions for such a project can be downloaded from Appendix B at the end of this paper.

Ultimately, the students will choose a range of cost structures for firm 2 then run the simulation (iterate the solution process) to obtain a time path for the variables under consideration. They might also explore cases where both firms face decreasing costs but these are decreasing at different rates. Or, a case where costs are initially different but the higher-cost firm reduces cost to that of the other firm over time. Another possibility might involve shifting the demand curve out, as is typified by positive advertising, and observing the effects. This project should give the students a better understanding of how innovation, R&D and cost reductions in general can affect market structure. In an oligopoly setting students should gain some insight about the realtionship between lower costs and market share. Students might also be asked to write a brief essay that generalises this model to an industry like commercial airlines and to comment on how competition and costs can affect the structure of such an industry.

Student response to the assignment

This exercise was conducted as an in-class assignment in an intermediate microeconomic theory course at the University of Alaska Souteast where I held a visiting position in the social science department. The class size was small and only consisted of four students. A laptop cart was delivered to the classroom once a week throughout the semester for in-class Maple programming exercises. This provided an opportunity for students to learn a symbolic programming language gradually over a 15-week period. Any part of the text of *Varian's Intermediate Microeconomics: A Modern Approach* (2002), that dealt with algebraic manipulations or graphical analysis was incorporated into the programming excercises.

We began by simply plotting out supply and demand functions. This led to calculating consumer and producer surplus as well as the welfare effects of taxation. The students were also able to program and solve constrained optimisation problems such as the utility maximisation associated with the construction of the demand curve. Through these excercises and lecture the students gained an understanding of the range of possible effects that could be calculated by changing parameters such as those that typify demand and supply shifts. It was fairly intuitive for them to define a functional form for a parameter such as the vertical intercept term of a supply or demand function and to then plot out a time path of several periods for equilibrium price. This helped them to extend their classic static equilibrium analysis to include cases where a supply curve for example might shift rapidly at first and then decay and slowly converge to its end point as is the case with the linear first order difference equation that defines the marginal cost curve above.

Due to time constraints I was unable to code this particular exercise in Maple. As a result, for this particular exercise the students shared a single computer whose output was projected onto a large screen in the conference room. Each student was given a copy of the homework in advance and one of the more programming-savvy students was voted to do the actual code editing required to complete the exercise. Students gained an appreciation of the changes in market structure and profits that result as firms face increasingly different marginal costs, initial marginal costs and demand slopes. By the end of the course students had an increased confidence in computational modelling and a keen grasp of the iterative procedures.

Although programming in a sybolic language like Mathematica or Maple was intimidating at first, by the end of the course the students had gained an appreciation of the versatility provided by such a program and the ease with which numerically intensive operations could be carried out. Two of the students were junior level business students, one was a senior level psychology major interested in pursuing graduate studies in Psychology and the other an economics major interested in pursuing graduate studies in economics.

Conclusion

Mathematica is a widely used program for theorectical economic modelling. Unfortunately, many students learn late in their careers the benefits of becoming fluent in a mathematical programming language. Given the popularity of Mathematica and the fact that programming in one language tends to facilitate programming in another language there is no reason for instructors not to include a programming element to their undergraudate and first-year graduate courses. Mathematica provides a platform for tweaking and testing models to gain a better understanding of their properties. Furthermore, it provides a powerful creative outlet for the curious student of economics. Programming projects provide a hands-on task that often must be performed in a community lab where students can discuss the pros and cons of various programming styles and refine their coding skills. An extension to this project might generalise the model to include the n-firm case, various cost structures across firms, nonlinear demand curves, or explore slopes of demand curves and the resulting elasticities of demand at the solution point.

Appendix A:
Mathematica Notebook Cournot.nb

Cournot Duopoly Model

Key Assumptions

1) Market consists of only two firms.

2) Each firm provides an identical good.

3) Unit costs are constant but can differ across firms.

4) Each firm's profits depend on their output choice as well as the output of their competitor.

5) A Nash equilibrium for this game is a set of outputs in which each firm chooses its optimal level of output as a best response to the hypothesised output of its competitor, and each firm's belief about its competitors output is correct.

6) This is a simultaneous move game.

Part One

This is a simultaneous move game that results in an equilibrium determined by each firm taking actions to maximise its own profits based on accurate conjectures about the actions of the rival firm. First, we set up our notebook and clear out any variables that may be in memory.

```
Off[General::spell1]
Off[General::spell]
SetAttributes[x1, Constant]
SetAttributes[x2, Constant]
SetAttributes[a, Constant]
SetAttributes[b, Constant]
SetAttributes[MC1, Constant]
SetAttributes[MC2, Constant]
Clear[x1, x2, a, b, MC1, MC2, list1,
list2, temp1, temp2, Profit1,
    Profit2];
Clear[q1, q2, eqPr1, eqPr1, focPr1,
focPr2, R2, CournotProfit1,
    CournotProfit2];
```

Since the market consists of only two firms, The total market quantity supplied is the sum of firm 1 and firm 2's individual outputs. Let's also define the marginal cost of each firm to be 6. Note that given our demand function defined in the next section this puts supply somewhere on the high end or elastic end of the demand curve. We also define some lists to store data that we will generate later.

```
Q = q1 + q2;
MC1 = 6.;
MC2 = 6.;
list1 = {x1};
list2 = {x2};
Null
```

Next, we specify a functional form for the inverse market demand. It is linear and downward sloping in this case, with an intercept of 10 and a slope of negative 1.

```
a = 10.;
b = 1.;
Price = a - b*Q;
```

Within the game theory context we say that a firm's payoff is the profit it receives from the operation. Profit for firm 2 is computed here as the quantity firm 2 produces multiplied by the price minus cost per unit produced. This is the profit function. The second line simply substitutes our demand function from above for the more general p[q1,q2].

```
eqPr2 = Profit2 == q2*(P[q1, q2] - MC2);
eqPr2 = Expand[% /. P[q1, q2] -> Price]
```

Now, we optimise our profit by maximising the profit function. This segment of code first solves the first order condition of our profit function by taking the derivative and solving for zero. The second line solves algebraically in terms of firm two\[CloseCurlyQuote]s quantity since the Cournot game is a quantity choice game. Finally we define this function as firm two\[CloseCurlyQuote]s reaction function.

```
focPr2 = D[eqPr2, q2]
temp2 = Solve[focPr2, q2]
React2 = R2[q1] == q2 /. temp2[[1]]
```

Now that we have firm two\[CloseCurlyQuote]s reaction function we move onto firm 1. Everything is defined exactly as above.

```
eqPr1 = Profit1 == q1*(P[q1, q2] - MC1);
```

```
    eqPr1 = Expand[% /. P[q1, q2] -> Price]
    focPr1 = D[eqPr1, q1]
    temp1 = Solve[focPr1, q1];
    React1 = R1[q2] == Simplify[q1 /.
    temp1[[1]]]
```

Finally, we plot out the reaction functions of both firms. The Cournot equilibrium is designated by the simultaneous solution to these two equations. This is our classic mapping of the reaction curves for firm 1 and firm 2. On the axes are the best response quantities to be produced by each firm. The firms decide what amount to produce by selecting the quantity that corresponds to the intersection of the reaction curves.

```
reactPlot =
    Plot[{q2 /. Solve[focPr1 /. {a -> 10, b ->
    1, MC1 -> 6}, q2][[1]],
    q2 /. Solve[focPr2 /. {a -> 1, b -> 1, MC2
    -> 6}, q2][[1]]}, {q1, 0,
    10}, PlotRange -> {0, 10},
    PlotStyle -> {{RGBColor[0, 0, 1],
    Thickness[0.01]}, {RGBColor[1, 0, 0],
    Thickness[0.01]}}, AxesLabel -> {“q1”,
    “q2”},
    PlotLabel -> “Reaction Curves blue=firm1”]
```

To find the exact point where the reaction functions intersect we solve the simultaneous move Cournot game algebraically. Here we simply set the reaction functions equal to each other and find the point at which they intersect to get our optimal outputs for firm 1 and firm 2. Once we have these values we can solve for the market quantity (firm 1 + firm 2), the equilibrium price (P = 10 – Q), and the profit each firm can expect (e.g. Profit1 = P*q1 – MC1*q1). We also compute the market share of each firm by dividing the firm’s output at any time by the market output for that time period (e.g. market share 1 = q1/Q). This is the code that solves for the system of two equations.

```
    Print[“Cournot quantity for firm 1 then firm
    2”];
    cournotQ =
    Simplify[Solve[{React1 /. R1[q2] -> q1,
    React2 /. R2[q1] -> q2}, {q1,
    q2}]];
    q1cournot = q1 /. %[[1]]
    q2cournot = q2 /. %%[[1]]
    Print[“Total Cournot market quantity”];
    Qcournot = Q /. {q1 -> q1cournot, q2 ->
    q2cournot}
    Print[“Market Share for firm 1 then firm
    2”];
    share1 = q1cournot/ Qcournot;
    share2 = q2cournot/ Qcournot;
    Print[share1];
    Print[share2];
    Print[“Cournot market price”];
    Pcournot = Simplify[Price /. {q1 ->
    q1cournot, q2 -> q2cournot}]
    Print[“Cournot profit for firm 1 then firm
    2”];
    CournotProfit1 =
    Simplify[q1cournot*(Pcournot - MC1)]
    CournotProfit2 =
    Simplify[q2cournot*(Pcournot - MC2)]
```

Part Two

Now that we have a nice static solution for our two-firm game let’s assume that firm 2 decides to establish an R&D and systems engineering department whose mission is to develop technology that will lower production costs. The managers have crunched the numbers and they think they can lower costs in each time period. Let’s project a time path for how profits will change for the two firms. (Recall, we are simply repeating the above process which gave us a Cournot equilibrium price and quantity and allowed us to determine each firm’s profits. We are simply repeating the game a number of times, each time lowering firm 2’s costs by X% from the previous period and charting a time path of the results.) First, we need to reset our variables. Each time you make a change start the simulation by running this section of code, then the next section, then the graphing section.

```
    Clear[x1, x2, a, b, MC1, MC2, list1,
    list2, temp1, temp2, Profit1,
    Profit2];
    Clear[q1, q2, eqPr1, eqPr1, focPr1,
    focPr2, R2, CournotProfit1,
    CournotProfit2];
    Clear[qlist, plist, q1list, q2list];
    a = 10.;
    b = 1.;
    MC1 = 6.;
    MC2 = 6.;
    list1 = {};
    list2 = {};
    qlist = {};
    plist = {};
    q1list = {};
    q2list = {};
    s1list = {};
    s2list = {};
```

Now we can compute the static solution obtained above over a time path. Just click the code below for as many iterations (time periods) as you want, then graph the results.

```
    << Graphics`MultipleListPlot`
    For[i = 0, i < 12, i++,
    Q = q1 + q2;
    Price = a - b*Q;
    eqPr2 = Profit2 == q2*(P[q1, q2] - MC2);
    eqPr2 = Expand[eqPr2 /. P[q1, q2] -> Price];
    focPr2 = D[eqPr2, q2];
    temp2 = Solve[focPr2, q2];
    React2 = R2[q1] == q2 /. temp2[[1]];
    eqPr1 = Profit1 == q1*(P[q1, q2] - MC1);
    eqPr1 = Expand[eqPr1 /. P[q1, q2] -> Price];
    focPr1 = D[eqPr1, q1];
    temp1 = Solve[focPr1, q1];
    React1 = R1[q2] == Simplify[q1 /. temp1[[1]]];
    cournotQ =
    Simplify[Solve[{React1 /. R1[q2] -> q1, React2 /. R2[q1]
    -> q2}, {q1,
q2}]];
Print[“Cournot quantity for firm 1 then firm 2”];
q1cournot = q1 /. cournotQ[[1]];
q2cournot = q2 /. cournotQ[[1]];
AppendTo[q1list, q1cournot];
AppendTo[q2list, q2cournot];
Print[q1cournot];
Print[q2cournot];
Print[“Total Cournot market quantity”];
Qcournot = Q /. {q1 -> q1cournot, q2 -> q2cournot};
AppendTo[qlist, Qcournot];
Print[Qcournot];
Print[“Market Share for firm 1 then firm 2”];
share1 = q1cournot/ Qcournot;
share2 = q2cournot/ Qcournot;
AppendTo[s1list, share1];
AppendTo[s2list, share2];
Print[share1];
Print[share2];
Print[“Cournot market price”];
Pcournot = Simplify[Price /. {q1 -> q1cournot, q2 ->
q2cournot}];
AppendTo[plist, Pcournot];
Print[Pcournot];
Print[“Time Series of the Cournot profit for firm 1 then
firm 2”];
CournotProfit1 = Simplify[q1cournot*(Pcournot -
MC1)];
CournotProfit2 = Simplify[q2cournot*(Pcournot -
MC2)];
x1 = CournotProfit1;
x2 = CournotProfit2;
AppendTo[list1, x1];
AppendTo[list2, x2];
Print[list1];
Print[list2];
Print[“Next period cost for firm 2”];
MC2 = MC2 - .001*MC2;
Print[MC2];
Print[];
Print[]]
```

After you’ve run the simulation, run the code below to plot out a time path of profits for each firm over 12 periods. Use the output generated above and the graphs provided below to answer the questions.

```
pqlist = {{qlist[[1]], plist[[1]]},
{qlist[[2]], plist[[2]]}, {qlist[[3]],
plist[[3]]}, {qlist[[4]], plist[[4]]},
{qlist[[5]],
plist[[5]]}, {qlist[[6]], plist[[6]]},
{qlist[[7]],
plist[[7]]}, {qlist[[8]], plist[[8]]},
{qlist[[9]],
plist[[9]]}, {qlist[[10]], plist[[10]]},
{qlist[[11]],
plist[[11]]}, {qlist[[12]], plist[[12]]}};
plot1 = Plot[p = 10 - q, {q, 0, 10},
PlotStyle -> Hue[.6],
AxesLabel -> {“Quantity”, “Market Price”},
DisplayFunction -> Identity]
plot2 = ListPlot[pqlist, PlotStyle ->
{Hue[0], PointSize[.03]},
AxesLabel -> {“Quantity”, “Market Price”},
DisplayFunction -> Identity]
Show[plot1, plot2, DisplayFunction ->
$DisplayFunction]
ListPlot[s1list, PlotJoined -> True,
DisplayFunction -> Identity]
ListPlot[s2list, PlotJoined -> True,
DisplayFunction -> Identity]
MultipleListPlot[s1list, s2list,
PlotJoined -> True,
DisplayFunction -> $DisplayFunction,
AxesLabel -> {“Time”, “Market Share”}]
MultipleListPlot[{list1, list2},
PlotJoined -> True,
AxesLabel -> {“Time”, “Profits”}]
```

Appendix B: Assignment instructions

Cournot Game: changing cost experiment

Download the Mathematica notebook entitled ‘cournot_homework.nb’ from the class website. Simply read part one and follow the directions provided. In part two, you will complete your assignment by choosing a cost reduction rate for the innovating firm and running the simulation. To do this simply change the following line of code: MC2 = MC2 - .001*MC2; in the cournot_homework.nb file. Given the following questions discuss the time paths for market price, market quantity, firm1 quantity, firm2 quantity, firm1 profit, firm2 profit, firm1 market share and firm2 market share. (Hint: compare the data generated at each period. Also, view the graphs at the end of the output after each simulation.) Our task is to simulate and describe a duopoly where one firm is able to reduce costs over time and the other is faced with constant costs.

Conduct the following experiments:

1. Leave MC2 = MC2 - .001*MC2 as it is and run the simulation. Discuss the effects of this cost structure, over time, on quantities, prices, profits and market shares. What is happening in terms of market supply and demand? How much difference was there between the values at the beginning of the simulation compared to the end?

2. Set MC2 = MC2 - .05*MC2 and run the simulation. Discuss the effects of this cost structure, over time, on quantities, prices, profits and market shares. What is happening in terms of market supply and demand? Do you think firm 2 will ever obtain monopoly power? Are the effects more or less drastic than in part 1?

3. Set MC2 = MC2 - .1*MC2 and run the simulation. Discuss the effects of this cost structure, over time, on quantities, prices, profits and market shares. What is happening in terms of market supply and demand? What do negative quantity and negative market share imply about our two firm model? (Hint: Think in terms of market entry and market exit.)

4. We have already talked about industry concentration ratios and the Herfindahl-Hirschman Index. What affect do lower costs for one firm in an industry have on

industry concentration? Do you think these results would generalise to an industry with say ten firms?

5. Discuss the incentives that firms operating in this type of industry might have to innovate or invest in R&D. Can you think of an industry that might be characterised by this type of model?

6. If we were to change the slope of the demand curve by changing b=1.; how might this affect our results? Try b=3.; and b=.25;

7. Do firms benefit more or less by reducing costs if they are operating at the elastic or inelastic end of the demand function? Marginal costs are currently set as MC1 = 6.; and MC2 = 6.;. This puts supply on the elastic end of the demand curve. Try MC1 = .5; and MC2 = .5;. How does this affect the results?

Notes

[1] The Solow growth model is one exception. Intermediate texts such as Mankiw's *Macroeconomics* (2002) does introduce the concept of iteration when discussing the Solow growth model though computational methods are not discussed. Another exception can be found in introductory texts such as Schiller's *The Economy Today* (2005). Examples here involve the multiplier effects of both fiscal policy and money creation by financial institutions.

[2] Two excellent web-based sources for learning how to program the Cournot model are provided by David A. Kendrick *et al.* (2005) and Masaru Uzawa (2000).

[3] This is a first order linear difference of the form $u_{n+1} = u_n(1\text{-}\rho)$ *where* $\rho=.05$. This function determines the rate of the marginal cost parameter's change within the system. This model could be extended to higher orders and non-linear cases although solutions become increasingly complicated.

Notes

Notebook is available for download at http://geocities.com/monster_economist/me/downloads/cournot_hw.nb

Homework is available for download at http://geocities.com/monster_economist/me/downloads/cournot_hw.doc

References

Belsley, David A. (1999). Mathematica as an Environment for Doing Economics and Econometrics, *Computational Economics*. 14 (1-2), pp. 69-87

Cournot, Augustin (1838). *Researches into the Mathematical Principles of the Theory of Wealth*.

Kendrick, D.A., Mercade, P.R. and Amman, H.M. (2005). *Computational Economics*. Retrieved 9 October 2005, from the University of Texas, Computational Economics Website: http://www.eco.utexas.edu/compeco/

Mankiw, N. Gregory (2002). *Macroeconomics*, 5th ed., Worth Publishers: New York, NY.

Schiller, Bradley R. (2005). *The Economy Today*, 10th ed., McGraw-Hill/Irwin: Columbus, OH.

Uzawa, Masaru. (2000). *Looking at the Cournot-Nash Equilibrium by Using Mathematica Graphics*. Retrieved 9 October 2005, from the Otaru University of Commerce Website: http://www.res.otaru-uc.ac.jp/~uzawa/cal-economics/cournot-e3.pdf

Varian, Hal R. (ed.) (1993), *Economics and Financial Modeling with Mathematica*, Springer-Verlag (Telos): Santa Clara, CA.

Varian, Hal R. (ed.) (1996), *Computational Economics and Finance*, Springer-Verlag (Telos): Santa Clara, CA.

Varian, Hal R. (2002), I*ntermediate Microeconomics: A Modern Approach*, 6th ed. W.W.Norton & Co.: New York, NY.

Contact details

Tim Kochanski
Systems Science and Economics Departments
Portland State University
Portland, OR 99824
USA

Cell: 503-928-0187
Email: tsk@pdx.edu