

Econ 353 – Final Project  
Mark Gillis  
University of Victoria

## Mathematica Model Portfolio

## **Introduction**

The following is a portfolio of Mathematica models that I created predominately for the purpose of teaching. The models are all various economics models that I have worked with in the last few years. Because these models were created for the purpose of teaching they don't really have any applicable 'experiments' (as compared to a GAMS model etc). The models are intended to show the results of comparative statics exercises graphically. For example, with a supply and demand diagram I can instantly show, graphically, how a change in demand elasticity will effect the demand curve.

As this was a learning process for me, some of my earlier models are a little bit more basic. Creating several different models though really helped me learn Mathematica and allowed me branch out and program really complex ones (like my "Pièce de résistance"). Unfortunately because all of the models are mostly just one really long line of code, I couldn't really do a step-by-step description of my coding within the notebooks. Instead I opted to write about particular highlights of my models in this document.

As side note, there are a few characters that Mathematica uses that are not recognized by word, so when I copied and pasted my coding into this document it caused some weird things to show up. Using the attached CD is probably the easiest way to look at the coding.

## **CD**

Attached is a CD of all my notebooks so that you can work with the models themselves, as opposed to just looking at the pictures. You are also able to see all the coding that run my models as well.

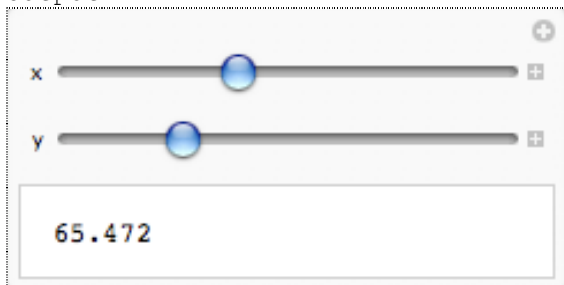
Just open up whichever notebook you want, evaluate the notebook, and that should be it!

## Manipulate

The key portion of all of my models is Mathematica's 'Manipulate' feature. This feature lets the user create an interactive frame that allows them to change parameter values and see the results instantly. In other words, Manipulate allows the user to run instantaneous comparative statics exercises. For example:

```
Input-> Manipulate[x y, {{x, 10}, 5, 15}, {{y, 10}, 5, 15}]
```

Output->



The Manipulate frame above is solving for 'X\*Y'. The sliders allow the user to change the values of X and Y to anywhere between 5 and 15, and then see the results instantly.

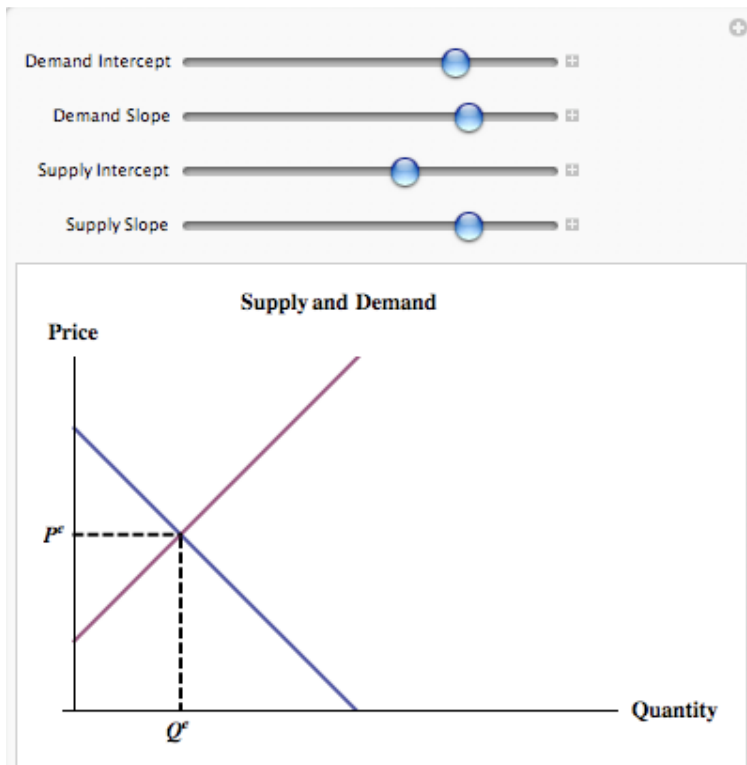
Manipulate becomes particularly interesting when it is used in conjunction with the 'Plot' feature. Doing this allows the user to change any parameters, like the above example, and then see the results graphically. This combination is the cornerstone for all of my models.

# Model 1 – Supply and Demand

## Description

This model is a basic Supply and Demand model. The demand and supply curves are programmed using inverse functions. The model allows the user to change both the slope and the intercepts of the functions. The model is displayed in the general form, so there are no real values assigned to the equilibrium price or quantity. Instead I chose to display these values as  $P^e$  and  $Q^e$ .

## Picture



## Coding

```
Manipulate[
  Plot[{a + b*x, c + d*x}, {x, 0, 15}, AxesOrigin -> {0, 0},
    AxesLabel -> {"Quantity", "Price"}, LabelStyle -> Directive[Bold,
    12], PlotRange -> {0, 10}, Ticks -> {{{(a - c)/(d - b), "
    SuperscriptBox["Q", "e"] "}}, {{
    a + b*((a - c)/(d - b)), "
    SuperscriptBox["P", "e"] "}}}, PlotStyle -> Directive[Thick],
    Ticks -> {Automatic, Automatic}, TicksStyle -> Directive[12],
    Epilog -> {{Thick, Dashed, Line[{{((a - c)/(d - b)), 0}, {(a -
    c)/(d - b), (a + b*((a - c)/(d - b))}}]}, {Thick, Dashed,
    Line[{{0, (a + b*((a - c)/(d - b))}, {(a - c)/(d - b),
    a + b*((a -
```

```
c)/(d - b))}}]]}}, AspectRatio -> Automatic, PlotLabel -> "Supply and Demand", {{a, 8, "Demand Intercept"}, 2, 10}, {{b, -1, "Demand Slope"}, -4, -.2}, {{c, 2, "Supply Intercept"}, 5, 0}, {{d, 1, "Supply Slope"}, 4, .2}}
```

## Programming Highlights

There are two particular portions of this coding that appear in many of my other models also, and I wanted to shed a little light on them.

The first is the feature Epilog. This feature renders a graphic after the rest of the coding has been run/updated. So basically by adding this Epilog within Manipulate allows me to create a graphic that updates instantly every time a parameter value is changed.

I use Epilog in all my models to create a line from the axis to the equilibrium point. The 'Line' feature allows me to create a line between two or more points within the plot. Therefore in this model I use Epilog to create the line graphic from  $\{Q^e, 0\}$  to  $\{Q^e, P^e\}$ .

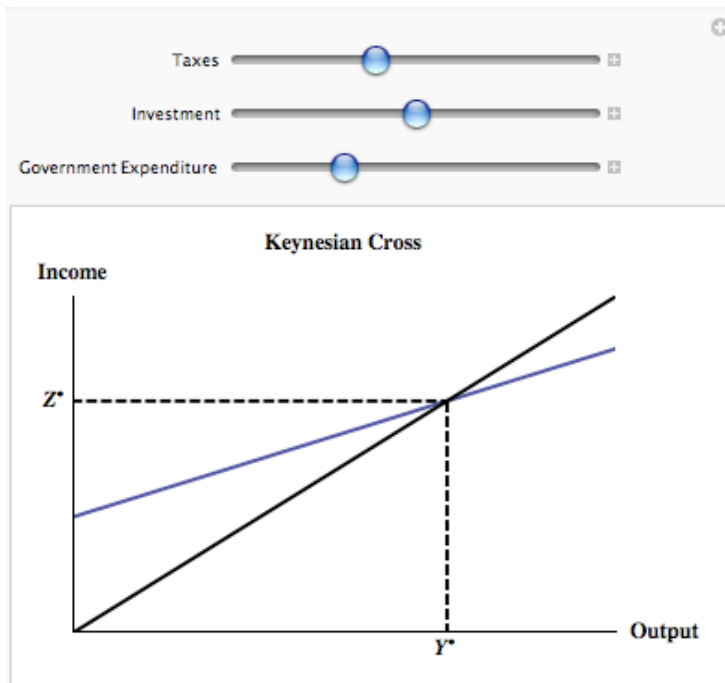
The second feature I wanted to explain was Ticks. Ticks is a feature that allows me to specify what I want as place markers on the axes (words, numbers, etc). In this model I used the Ticks feature to create the points  $\{Q^e, P^e\}$ .

## Model 2 – Keynesian Cross

### Description

The following is a general form Keynesian Cross diagram. The expenditure line is programmed in the form  $Z = C + I + G$ , where  $C = c_0 + c_1(Y-T)$ . The model allows the user to change government expenditure, investment and taxes. The equilibrium output and income is denoted by  $Z^*$  and  $Y^*$ .

### Picture



### Coding

```
Manipulate[ Plot[{c0 + c1 (Y - T) + Iv + G}, {Y, 0, 45},
  Epilog -> {{Thick, Line[{{0, 0}, {45, 45}}]}, {Thick, Dashed,
    Line[{{((c0 + Iv + G - c1 T)/(1 - c1)), 0}, {(c0 + Iv + G - c1
  T)/(1 - c1)}, (c0 + Iv + G - c1 T)/(1 - c1)}]}, {Thick, Dashed,
    Line[{{0, ((c0 + Iv + G - c1 T)/(1 - c1))}, {(c0 + Iv + G - c1
  T)/(1 - c1)}, ((c0 + Iv + G - c1 T)/(1 - c1))}}]}], AxesOrigin ->
{0, 0}, PlotStyle -> Directive[Thick], LabelStyle -> {Bold},
PlotRange -> {{0, 45}, {45, 0}}, Ticks -> {{{((c0 + Iv + G - c1 T)/(1 -
c1)), "SuperscriptBox[\"Y\", \"*\"] "}}, {{{(c0 + Iv + G -
c1 T)/(1 - c1)}, "SuperscriptBox[\"Z\", \"*\"] "}}, PlotLabel
-> "Keynesian Cross", AxesLabel -> {"Output", "Income"}, AxesStyle
-> Directive[Bold, 12]], {{T, 3, "Taxes"}, 0, 8}, {{Iv, 10,
"Investment"}, 5, 15}, {{G, 5, "Government Expenditure"}, 3,
10}]
```

## **Programming Highlights**

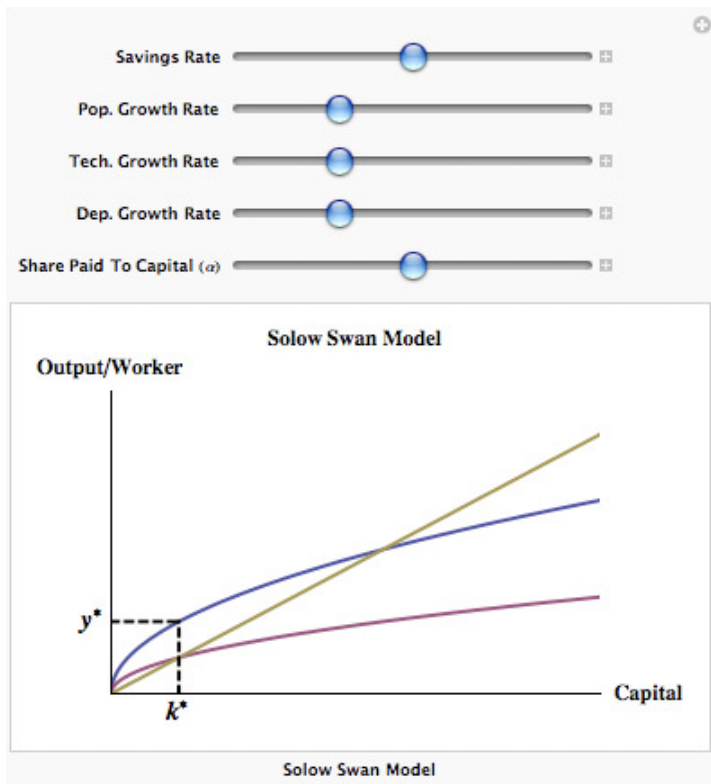
The programming used for this model was quite similar to the supply and demand model. I had originally wanted to program an IS-LM model, but I discovered that someone else had programmed a very good one on Mathematica's demonstration page and so I decided to try something different.

## Model 3 – Solow Swan Model

### Description

The following is a Solow Swan model with technological growth. The blue line represents the per worker output, the yellow line represents the capital widening line, and the purple line is the saving per worker function.

### Picture



### Coding

```
Manipulate[Plot[{k^a, s*k^a, k (n + g + t)}, {k, 0, 20},
  AxesLabel -> {"Capital", "Output/Worker"},
  AxesOrigin -> {0.0, 0.0}, PlotLabel -> "Solow Swan Model", PlotStyle
-> Directive[Thick], PlotRange -> {{0, 20}, {0, 7}}, Ticks ->
{{{(s/(n + g + t))^(1/(1 - a)), "SuperscriptBox["k", "*" ]}
}, {(s/(n + g + t))^(a/(1 - a)), "SuperscriptBox["y", "*" ]}
"}}, TicksStyle -> Directive[Bold, 15], LabelStyle ->
Directive[Bold, 12], Epilog -> {{Thick, Dashed, Line[{{(s/(n + g
+ t))^(1/(1 - a)), 0}, {(s/(n + g + t))^(1/(1 - a)), (s/(n + g
+ t))^(a/(1 - a))}}]}, {Thick, Dashed, Line[{{0, (s/(n + g +
t))^(a/(1 - a))}, {(s/(n + g + t))^(1/(1 - a)), (s/(n + g + t))^(a/(1 -
a))}}]}], {{s, .5, "Savings Rate"}, 0, 1}, {{n, .1, "Pop. Growth
```

```
Rate"}, .025, .3}, {{g, .1, "Tech. Growth Rate"}, .025, .3}, {{t,  
.1, "Dep. Growth Rate"}, .025, .3}, {{a, .5,  
"Share Paid To Capital (a)"}, .4, .6}, LabelStyle -> {Bold},  
FrameLabel -> "Solow Swan Model"]
```

## **Programming Highlights**

This model is pretty similar to my previous ones, although some of the Epilog line functions are slightly more complicated than previous.

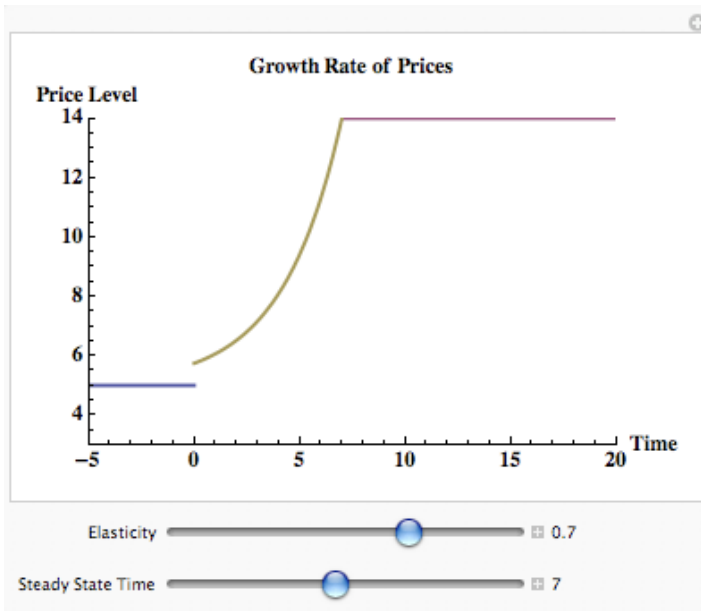
## Model 4 – Price Model

### Description

The models start to get a little bit more complex now. When I started this project I wanted one of my big centerpieces to be an exchange rate model. So I went to Dr. Voss and he supplied me with some of his notes from Econ 401 (Adv. Macroeconomics). He showed me two different models, the first being this Price model.

This price model describes the growth path of prices when there is an unexpected increase in the money supply growth rate. All of the variables in this model are in log form. Originally money supply is growing at a constant rate, and then when money supply is increased there is a jump in prices and then a growth over several periods. At some steady state time,  $T$ , the price growth rate levels out and remains constant.

### Picture



### Coding

```
Manipulate[
  Plot[{If[t < 0, mm], If[t >= T, mmm],
    If[0 <= t <= T, (mm + (a^(T - t) (mmm - mm)))]}, {t, -5, 20},
  PlotRange -> {{-5, 20}, {3, 14}}, PlotStyle -> Directive[Thick],
```

```
LabelStyle -> Directive[Bold, 12], PlotLabel -> "Growth Rate of  
Prices",  
AxesLabel -> {"Time", "Price Level"},  
AxesOrigin -> {-5, 3}], {{a, .7, "Elasticity"}, .01, 1,  
Appearance -> "Labeled"}, {{T, 7, "Steady State Time"}, 0, 15,  
Appearance -> "Labeled"}, ControlPlacement -> {Bottom, Bottom}]
```

## **Programming Highlights**

Figuring out how to program the stepwise function took me quite some time. I eventually figured out that IF statements were the most efficient solution. These statements can be seen quite clearly in the first line of coding. The methodology of the IF statement is as follows: `If[condition,t,f]` gives `t` if the condition is true, and gives `f` if the condition is false. The condition I used was the range of the function (ex.  $t < 0$ ), and then the ‘`t`’ portion would be the actual function.

The IF statement ended up becoming the cornerstone for a lot of my work. I found it to be the most effective way to program stepwise functions and also impose lots of little restrictions on my models.

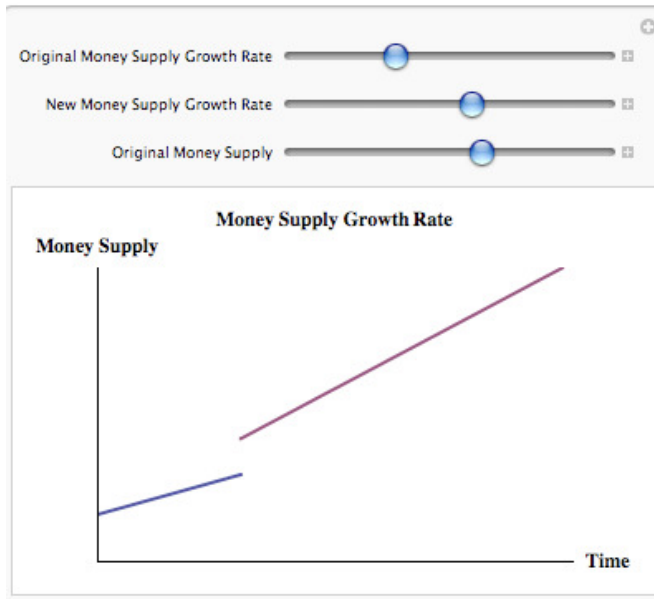
## Model 5 – Exchange Rate Model

### Part 1 – Money Supply Growth

#### Description

This money supply growth model is quite simple, although it is the first step in developing the exchange rate model. The model shows money supply growing at some constant rate until time = 5, when there is an increase in the money supply growth rate.

#### Picture



#### Coding

```
Manipulate[
  Plot[{If[0 <= t <= T, u*t + mbar], If[t >= T, uu*t + mbar]}, {t, 0,
  10}, PlotStyle -> Directive[Thick], PlotLabel -> "Money Supply Growth
  Rate", LabelStyle -> Directive[Bold, 14], AxesLabel -> {"Time",
  "Money Supply"}, Ticks -> None, AxesStyle -> Directive[12],
  PlotRange -> {{0, 10}, {0, 25}}, {{u, 1, "Original Money Supply
  Growth Rate"}, .25, 3}, {{uu, 2, "New Money Supply Growth Rate"}, 1,
  3}, {{mbar, 4, "Original Money Supply"}, 1, 6}]
```

## Part 2 – Exchange Rate Growth Path

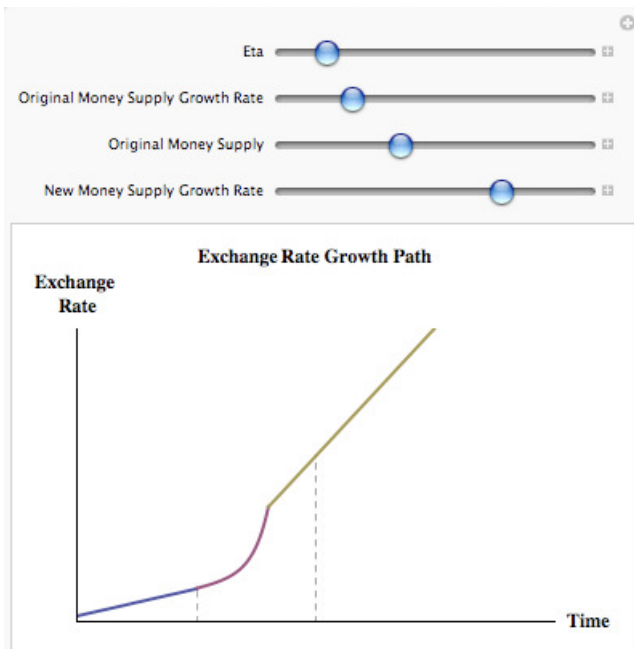
### Description

This is the second model that I developed with the help of Dr. Voss. It describes the growth path of the exchange rate and what happens when there is an unanticipated increase in the money supply. The change in the money supply was shown in the last model.

This model is another, relatively complex, stepwise function. The function solves for the exchange rate as a function of the elasticity parameter  $\eta$ , the original and new money supply growth rates, and of the initial money supply level. All of the variables are not in log form. I also decided to remove all the values on the axes as they weren't really necessarily in showing the general form.

This model is also kind of neat because if you change the parameters appropriately you can also make this model display what happens when there is an unexpected decline in the money supply growth rate.

### Picture



## Coding

```
Manipulate[
  Plot[{If[t < 0, mbar + u*t + (h/(1 + h))/(1 - (h/(1 + h))) u],
    If[0 <= t < T, mbar + u*t + ((h/(1 + h))/(1 - (h/(1 + h))))*u +
    (h/(1 + h))^(T - t)* T*(uu - u) + ((h/(1 + h))/(1 - (h/(1 +
    h))))*(h/(1 + h))^(T - t) (uu - u)], If[t >= T, mbar + uu*t + (h/(1 +
    h))/(1 - (h/(1 + h))) uu]}], {t, -5, 15}, PlotRange -> {{-5, 15}, {0,
    45}}, PlotStyle -> Directive[Thick], AxesOrigin -> {-5, 0}, AxesStyle
-> Directive[12, Bold], AxesLabel -> {"Time", "Exchange Rate"},
PlotLabel -> "Exchange Rate Growth Path", Ticks -> None, LabelStyle ->
Bold, Epilog -> {{Dashed, Line[{{0, 0}, {0, (mbar + u*0 + (h/(1 +
h))/(1 - (h/(1 + h))) u)}}]}, {Dashed, Line[{{5, 0}, {5, mbar + uu*5
+ (h/(1 + h))/(1 - (h/(1 + h))) uu}}]}]}, {{h, .25, "Eta"}, .01, 2},
{{u, 1, "Original Money Supply Growth Rate"}, .25, 3}, {{mbar, 5,
"Original Money Supply"}, 2, 10}, {{uu, 3, "New Money Supply Growth
Rate"}, 1, 5}]
```

## Programming Highlights

There is nothing particularly new that hasn't already been commented on. Once I had figured out how to build a stepwise function in the Price Model this model was pretty straightforward in terms of programming. What made this model so interesting for me was that it was one that I had never worked on before.

## Model 6 - Pièce de résistance – Insurance Demand Model

### Description

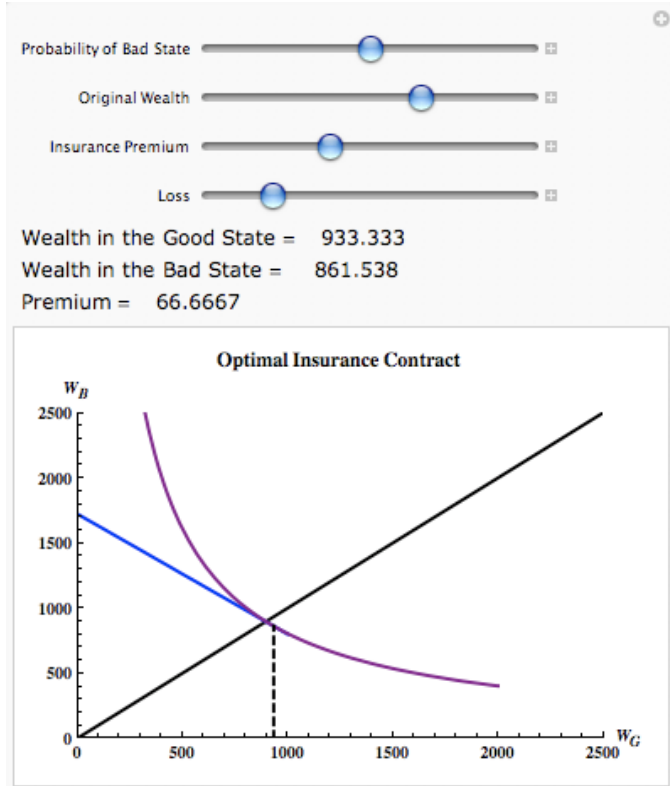
This model solves for the optimal amount of insurance a risk averse individual should purchase.

The first portion of the notebook is the derivation of the optimal wealth values. The risk averse individual's expected utility of wealth function is:  $EU(w) = p \cdot \text{Log}(B) + (1-p) \cdot \text{Log}(G)$ , where  $p$  is the probability of the bad state occurring, and 'B' and 'G' represent the wealth in the bad state and the wealth in the good state respectively. I also set up the budget constraint:  $\text{Budget} = M - G - Ls z - B z + G z$ , where  $M$  is initial wealth,  $L$  is the loss in the bad state, and  $z$  is the cost of insurance per dollar of coverage.

Using the Lagrangian optimization method I solved for the general form of the equilibrium wealth in the good and bad state (being subject to the budget constraint). With these general form solutions I was able to use the plot feature to plot the individual's indifference curve and their budget constraint.

Using the Manipulate feature the user is able to change the probability of loss, the loss amount, the initial wealth, and the insurance premium. If the indifference curve disappears and is replaced by a red dot at the initial bundle point this is because the individual has dropped out of the market for insurance and is 'consuming' at their initial bundle.

## Picture



## Coding

```
Manipulate[ If[(-M + M p + Ls z - Ls p z)/(-1 + z) <= M,
  Plot[{If[G <= M, (-G + M + G z - Ls z)/z], If[(-M + M p + Ls z - Ls p
z)/(-1 + z) <= M, Exp[((1 - p)*Log[G] - p*Log[(M p - Ls p z)/z]) -
(1 - p)* Log[(-M + M p + Ls z - Ls p z)/(-1 + z)]]/-p]]}, {G, 0,
2000}, AxesLabel -> {" SubscriptBox["W", "G"] ", "
SubscriptBox["W", "B"] "}, PlotRange -> {{0, 2500}, {0, 2500}},
  AxesStyle -> Directive[Bold], LabelStyle -> Bold, Epilog -> {Thick,
Dashed, Line[{{(-M + M p + Ls z - Ls p z)/(-1 + z), 0}, {-M + M p +
Ls z - Ls p z)/(-1 + z), (M p - Ls p z)/z}}]},
  Prolog -> {Thick, Black, Line[{{0, 0}, {2500, 2500}}]},
  PlotStyle -> {{Thick, Blue}, {Thick, Purple}},
  PlotLabel -> "Optimal Insurance Contract",
  Plot[{If[G <= M, (-G + M + G z - Ls z)/z], G}, {G, 0, 2500},
  AxesStyle -> Directive[Bold],
  AxesLabel -> {" SubscriptBox["W", "G"] ",
" SubscriptBox["W", "B"] "},
  PlotStyle -> {{Thick, Blue}, {Thick, Black}}, AxesOrigin -> {0, 0},
  PlotRange -> {{0, 2500}, {0, 2500}}, AxesStyle -> Directive[Bold],
  Epilog -> {PointSize[Large], Red, Point[{M, M - Ls}]},
  PlotLabel -> "Optimal Insurance Contract", LabelStyle -> Bold]],
{{p, .5, "Probability of Bad State"}, .1, .9}, {{M, 1000, "Original
Wealth"}, 500, 1250}, {{z, .52, "Insurance Premium"}, .3, .9}, {{Ls,
200, "Loss"}, 50, 900}, Text[ Style["Wealth in the Good State = "
Dynamic[ If[(-M + M p + Ls z - Ls p z)/(-1 + z) <= M,
Dynamic[(-M + M p + Ls z - Ls p z)/(-1 + z)], M]], 12, "Label"]],
```

```

Text[Style[ "Wealth in the Bad State = " Dynamic[
  If[(-M + M p + Ls z - Ls p z)/(-1 + z) <= M, Dynamic[(M p - Ls p
z)/z],
  M - Ls]], 12, "Label"]], Text[Style[ "Premium = "
Dynamic[
  If[(-M + M p + Ls z - Ls p z)/(-1 + z) <= M,
  Dynamic[M - (-M + M p + Ls z - Ls p z)/(-1 + z)], 0]], 12,
"Label"]]]

```

## Programming Highlights

This model took me quite a while to program. Setting up the initial problem and solving for the equilibrium values was pretty straightforward, but trying to plot it was quite a challenge.

As the budget constraint is a function of both the good and bad state wealth values Mathematica would plot it from the Y axis down to the X axis, but this isn't correct. An individual is not able to consume at any point that has a greater wealth in the good that is greater than their initial bundle. Therefore I had to restrict the budget constraint to stop at the initial bundle by using the If statement:  $\text{If}[G \leq M, (-G + M + G z - Ls z)/z]$ . Note that  $G \leq M$  is: Wealth in the Good State less than or equal to initial wealth. Therefore using this kind of restriction in an If statement allows me to make sure that the individual will never consume off their budget curve.

Doing restrictions like the above one was very common in this model. Another similar issue arose with the equilibrium wealth values. If the parameters of the model were changed in a particular way it would cause the equilibrium wealth values to be at some point off the 'restricted' budget curve. The individual would be trying to transfer a portion of the 'loss' to the good state, which is impossible.

To fix this I had to use If statements with the restriction  $(-M + M p + Ls z - Ls p z)/(-1 + z) \leq M$ , to make sure that they couldn't consume off of their budget line. The formula  $(-M + M p + Ls z - Ls p z)/(-1 + z)$  is the general form for the equilibrium wealth in the good state. This restriction basically says that 'if the equilibrium wealth in the good state is less than or equal to the initial endowment then...'. Using this formula also allowed me to create a restriction that was flexible to parameter changes caused by the manipulate feature. As the equilibrium wealth in the good state changes, from perhaps a change in

the initial wealth, so does this restriction. This restriction can also be found in the text portion (right at the end of the coding) of my model.

The other portion I wanted to highlight was the If statement found right after Manipulate. This statement basically allowed me to create two plots (which look exactly the same), and which one shows is dependant on whether or not the restriction holds. This was all due to an issue caused by the Epilog feature.

I wanted to use Epilog to draw a line from the axes to the equilibrium bundle. The issue arose if the 'equilibrium bundle' was off of the budget constraint (the same issue found above), because Epilog would still draw a line between the axis and this point (so there would be a line going from the axis to a blank portion of the plot).

The issue is that you can't use an If statement in conjunction with Epilog. Anytime the If statement didn't hold would cause Mathematica to give me an error because, so far as I understand, Epilog was trying to render the graphic '{Null}'.

Therefore, in short, I had to use this initial If statement to create two separate plots so that Epilog would render this line correctly. The first plot is used anytime that the individual buys insurance, and therefore the Epilog line is applicable. The second plot occurs when the individual doesn't buy insurance, and Epilog only shows a red dot at the initial bundle.