

The following cell sets some defaults for graphs. You may delete it if you prefer. The `fill_color` option is for graphing shaded areas. The color `gray10` appears a black; we use it because in a few applications involving points, the use of black makes the graph look like a blob rather than a collection of points. The `draw_realpart=false` keeps draw from graphing the real part of a complex value (it fails to do so in graphs of implicit functions).

```
(%i62) load(draw)$  
      set_draw_defaults( fill_color=gray40,color=gray10,  
      draw_realpart=false,font_size=8)$  
      kill(all)$
```

A Dynamic Model of Aggregate Demand and Aggregate Supply

This material follows Chapter 13 of Cowen and Tabarrok, Economics. It develops a dynamic version of the model of Aggregate Demand and Aggregate Supply. That is, the annual rates of change of the pertinent values are considered, rather than their levels. For the sake of compactness, AD, LRAS, SRAS are used to name Aggregate Demand, long-run Aggregate Supply, and short-run Aggregate Supply. These are the same as in StaticADAS workbook. Keep in mind that in this workbook, these names apply to rates of change and not to levels.

The first section develops the Dynamic Aggregate Demand Curve and provides some evidence regarding the history of the variables that are involved. The second section outlines a long-run Dynamic Aggregate Supply curve, based on the Solow Growth Curve. The third section considers real shocks to the system. The fourth section develops the short-run Aggregate Supply curve and combines it with the Aggregate Demand curve to show how the model can be applied. The final section considers shocks to the components of the Aggregate Demand curve. [Footnotes appear in brackets.]

1 Dynamic Aggregate Demand

The amount that people are willing and able to spend per year equals the product of the amount of money they hold and the "velocity" of that money. In the cell below, the money supply is M , and the number of times each dollar is spent to purchase a unit of final goods and services during a year, velocity, is v . Thus, $M \cdot v$ measures aggregate demand (AD). [The `kill(all)`, `ratprint:false`, and `fpprintprec:5` commands clear Maxima's memory, suppress printing of some intermediate output, and set the floating-point printing precision. These "housekeeping" commands can be omitted.]

```
(%i1) kill(all)$ ratprint:false$ fpprintprec:5$
```

```
depends([M,v,P,Y],t);
AD : M*v;
```

```
(%o3) [M(t),v(t),P(t),Y(t)]
```

```
(%o4) v M
```

1.1 Conversion of Annual Percentage Terms

We can convert the static expression for the quantity theory to its dynamic counterpart. The percentage change in AD equals the time derivative of AD divided by AD.

```
(%i5) diff(AD,t)/AD; expand(%);
```

```
(%o5) 
$$\frac{v\left(\frac{d}{dt}M\right) + \left(\frac{d}{dt}v\right)M}{vM}$$

```

```
(%o6) 
$$\frac{\frac{d}{dt}M}{M} + \frac{\frac{d}{dt}v}{v}$$

```

The result shows that the annual percentage change in AD (=M*v) equals the sum of the annual percentage changes in M and v.

The quantity theory of money points out that $Mv = P*Y$, where P is the general price level and Y is real output per year. Thus, for any year $M*v = P*Y$. (If v can vary to accommodate changes in M, P, and Y, then the quantity theory is actually an identity.) We can convert $P*Y$ to a dynamic expression, as we did for $M*v$.

```
(%i7) expand(diff(P*Y,t)/(P*Y));
```

```
(%o7) 
$$\frac{\frac{d}{dt}Y}{Y} + \frac{\frac{d}{dt}P}{P}$$

```

1.2 Inflation and Real Output Change

From output %o7, we determine that the annual growth rate of the money supply plus the annual growth rate of velocity must equal the annual inflation rate plus the annual growth in real output. Another way to state this is the the annual growth rate in spending must equal the annual growth rate in nominal output, the latter consisting of the sum of the inflation rate and the real growth rate. We restate this relationship as follows, to derive an expression that relates the inflation rate to the real output rate: $gM + gv = gP + gY$.

We define the function $AD = gM + gv + gY$. This is the inverse aggregate demand curve. That is, it shows the inflation rate as a function of the output growth rate, given the growth in the money supply and velocity. Strictly speaking, the relationship is implicit one: the inflation rate and the real growth rate are determined as a pair. The

expression that we use is to facilitate subsequent analysis, which is easier if one variable is treated as an explicit function of the other one.

```
(%i8) AD(gM, gv, gY) := gM + gv - gY;
```

```
(%o8) AD(gM, gv, gY) := gM + gv - gY
```

The cell below shows the height of the AD curve (the value of gP) at each of two rates of gY. In both cases $gP + gY = 0.05$, as is required by the values of gM and gv.

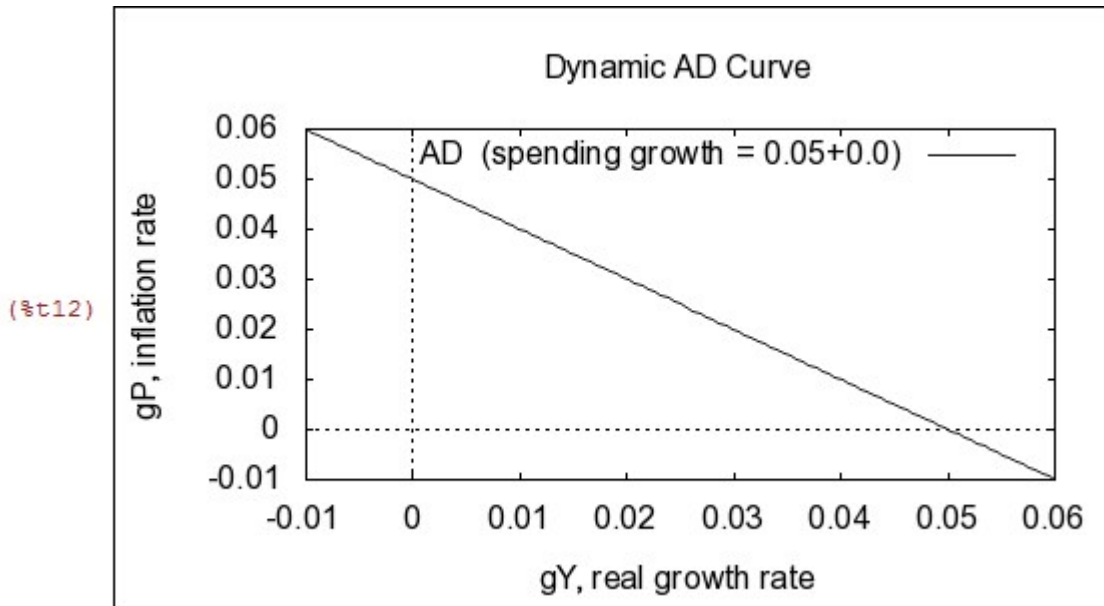
```
(%i9) AD(0.05, 0, 0.03);
      AD(0.05, 0, 0.02);
```

```
(%o9) 0.02
```

```
(%o10) 0.03
```

The graph below shows the AD curve over a range of values. Both $gP < 0$ (deflation) and $gY < 0$ (recession) are allowed.

```
(%i11) [gM0, gv0] : [0.05, 0.0]$
wxdraw2d(title="Dynamic AD Curve", xaxis=true,
yaxis=true, xlabel="gY, real growth rate",
ylabel = "gP, inflation rate", key = concat("AD
(spending growth = ", string(gM0), "+", string(gv0), ")"),
explicit(AD(gM0, gv0, gY), gY, -.01, .06 ) )$
```



1.3 Recent Behavior of the Variables

Before proceeding with the model, we look at the behavior of the four variables for a few years. The file "data.csv" was created from the Economic Report of the President, 2012 and placed in the "myfiles" directory on the C: drive. The first command below reads this file into the wxMaxima worksheet. The second command transposes the data

matrix (printing suppressed), to facilitate list manipulation below. The selection of M2 as the measure of money is arbitrary; the point here is to show something of how these variables have behaved.

```
(%i13) M1 : read_matrix("c:/myfiles/data.csv");
      M2: transpose(M1)$
```

```
(%o13)
[Year  M2   P   Yr ]
[2000  4912  89  11216]
[2001  5425  91  11338]
[2002  5769  92  11543]
[2003  6059  94  11836]
[2004  6404  97  12247]
[2005  6668  100 12623]
[2006  7060  103 12959]
[2007  7485  106 13206]
[2008  8232  109 13162]
[2009  8514  110 12703]
[2010  8796  111 13088]
```

The cells below create lists of values and of rates of change. The first command sets floating-point precision to 3 decimal places, rather than the default 16 places, to make the lists easier to read. The list of gM (annual fractional change of money supply) values is created in three steps: the Mlist is the second row of M2 (because the first row contains headers), the makelist command generates fractional changes for ten years, and the float command shows these values as floating-point values rather than Maxima's default (to show them as exact fractional values).

The creation of gPlist and gYlist are the same as for gMlist, except that the makelist command is embedded inside the float command, saving a step. The gvlist is created from the identity that we developed above: $gM + gv = gP + gY$. The Tlist is created for graphing below. [To produce percentage changes, one could multiply gMlist, gYlist, and gPlist by 100. The cons command constructs a new list, with the label added.]

```
(%
i15) Mlist : M2[2]$
      makelist( (Mlist[i]- Mlist[i-1])/Mlist[i],i, 3, 12 )$
      gMlist : float(%)$ Plist : M2[3]$ gPlist:
        float( makelist( (Plist[i]- Plist[i-1])/Plist[i],i, 3, 12 ) )
      $
      Ylist : M2[4]$ gYlist:
        float( makelist( (Ylist[i]- Ylist[i-1])/Ylist[i],i, 3, 12 ) )
      $
      gvlist : gPlist + gYlist - gMlist$
      Tlist : makelist(2000+i,i,1,10)$
      matrix(cons("Year",Tlist), cons("gM", gMlist), cons
        ("gP",gPlist),
```

```
cons("gY",gYlist),cons("gv",gvlist) ) ;
```

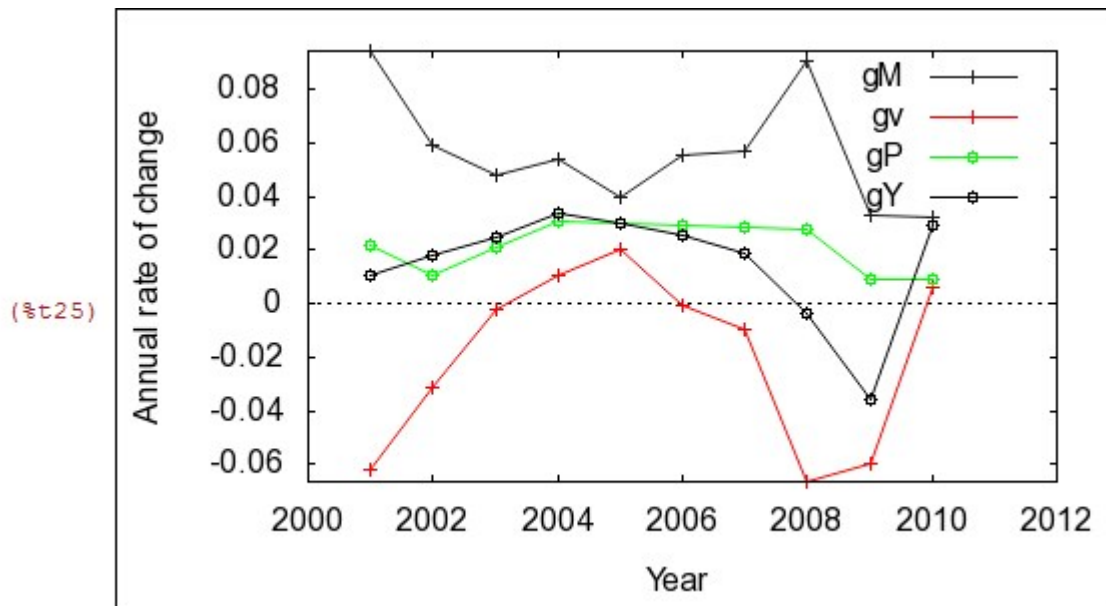
(%o24)

Year	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
gM	0.0946	0.0596	0.0479	0.0539	0.0396	0.0555	0.0568	0.0907	0.0331	0.0321
gP	0.022	0.0109	0.0213	0.0309	0.03	0.0291	0.0283	0.0275	0.00909	0.00901
gY	0.0108	0.0178	0.0248	0.0336	0.0298	0.0259	0.0187	-0.00334	-0.0361	0.0294
gv	-0.0618	-0.031	-0.00183	0.0106	0.0202	-4.69939 10 ⁻⁴	-0.00977	-0.0666	-0.0602	0.00637

The money supply, as measured by M2, has grown throughout this period ($gM > 0$), but at widely varying rates. The velocity has trended downward ($gv < 0$ for most years) and seems to move counter to changes in M. Money changes are partially offset by velocity changes, but the relationship is not very close.

The inflation rate, gP , has been below 4% per year for the entire period and has not exhibited large fluctuations. Real output grew for most of the decade ($gY > 0$), but fell in 2008 and 2009.

(%i25) wxdraw2d(xrange = [2000, 2012], key = "gM", xaxis=true,
 xlabel = "Year", ylabel="Annual rate of change",
 points_joined=true, points(Tlist,gMlist), color = red,
 key = "gv", points(Tlist,gvlist), color = green,point_type=6,
 key = "gP", points(Tlist, gPlist), color = black,
 key = "gY", points(Tlist,gYlist))\$

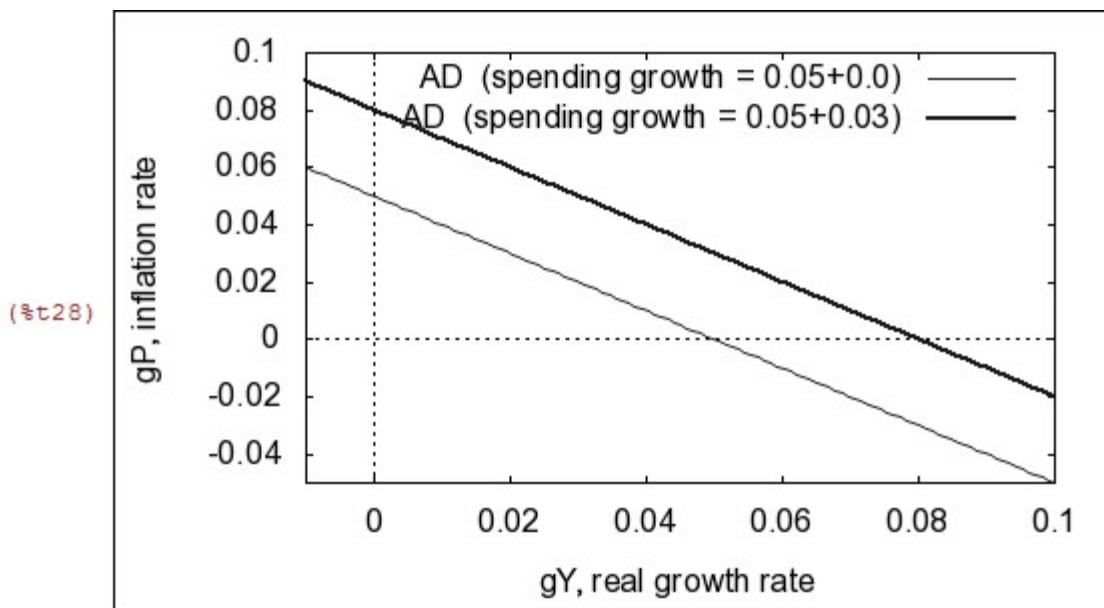


1.4 Shifting the AD Curve

The AD curve for an annual spending growth rate of 5%, shown previously, consists of all combinations of real growth and inflation that sum to 5%. A change in either M or v shifts this curve. Suppose that the money supply continues to grow at 5% per year, but that buyers use each dollar more times per year, so that $gv = gv_0 = 0.0$ is displaced by $gv = gv_1 = 0.03$. The graph below shows that the new AD curve moves rightward (or upward) by 0.03. Now inflation and real output growth must sum to 8% per year.

The monetary authorities might reduce the money supply growth from $g = 0.05$ to $g = 0.04$. If velocity does not change, then the AD curve must shift to the left. Adapt the commands below to confirm this. Likewise, determine what happens if $gM = gM1$ and $gv = gv1$.

```
(%i26) [gM0, gv0] : [0.05, 0.0]$
        [gM1, gv1] : [0.04, 0.03]$
        wxdraw2d( xaxis=true, yrange=[-0.05,0.1],
        yaxis=true, xlabel="gY, real growth rate",
        ylabel = "gP, inflation rate", key = concat("AD
        (spending growth = ",string(gM0),"+", string(gv0),")"),
        explicit(AD(gM0, gv0, gY), gY, -.01, .10 ),
        line_width = 2,key = concat("AD
        (spending growth = ",string(gM0),"+", string(gv1),")"),
        explicit(AD(gM0, gv1, gY), gY, -.01, .10 ) )$
```



2 Long-run Aggregate Supply: The Solow Growth Model

The long-run supply model is based on the Solow growth model. Our variant of this model treats aggregate real output, Y , as a function of four factors: the capital stock, the labor supply, a productivity trend, and a shock. The cell below defines the function and tells Maxima that K and L change with the passage of time. The derivative with respect to time shows that the trend growth of Y equals $b \cdot gK + (1-b) \cdot gL$, where gK and gL are the annual percentage growth rates of capital and labor. [The expand and collect terms commands are used to convert the expression to one that is easily interpreted. Such a conversion is not always possible; Maxima does not know how the output is to be used, so the user might have to manipulate output by hand.]

The derivative with respect to the shock w shows that a one-unit change in w causes a one percent-per-year change in Y .

```
(%i29) depends([K,L],t)$
```

```
Y : exp(a*t + w)*K^b*L^(1-b);
diff(Y,t)/Y$ expand(%)$ trend: collectterms(%, L);
diff(Y,w)/Y;
```

```
(%o30) %ew+a t Kb L1-b
```

```
(%o33) 
$$\frac{(1-b)\left(\frac{d}{dt}L\right)}{L} + \frac{b\left(\frac{d}{dt}K\right)}{K} + a$$

```

```
(%o34) 1
```

The cell below expresses the LRAS as a function of the variables and parameters that are analyzed above. All of these terms are real. Thus, the general price level change rate, gP, does not enter into the equation.

```
(%i35) LRAS(gK,gL,a,b,w):= a + b*gK + (1-b)*gL + w;
```

```
(%o35) LRAS(gK,gL,a,b,w):=a+b gK+(1-b) gL+w
```

The next cell enters a set of values for the system's variables and parameters. Capital grows at 3% per year, and labor grows at 2% per year. The annual trend rate of productivity change is a = 1.5%. The value of b, which reflects relative input productivity equals 0.3. Finally, we set w = 0 to define our base case.

Given these values, the economy's trend growth rate is 3.8% per year. For use below, we bind that value to the name gY0.

```
(%i36) [gK0, gL0, a0, b0, w0] : [0.03, 0.02, 0.015, 0.30, 0.0];
gY0: LRAS(gK0, gL0, a0, b0,w0);
```

```
(%o36) [0.03, 0.02, 0.015, 0.3, 0.0]
```

```
(%o37) 0.038
```

To determine the trend inflation, place gY0 into the AD function. The result is gP0 = 0.012, so the trend inflation rate is 1.2% per year.

```
(%i38) gP0: AD(gM0,gv0, gY0);
```

```
(%o38) 0.012
```

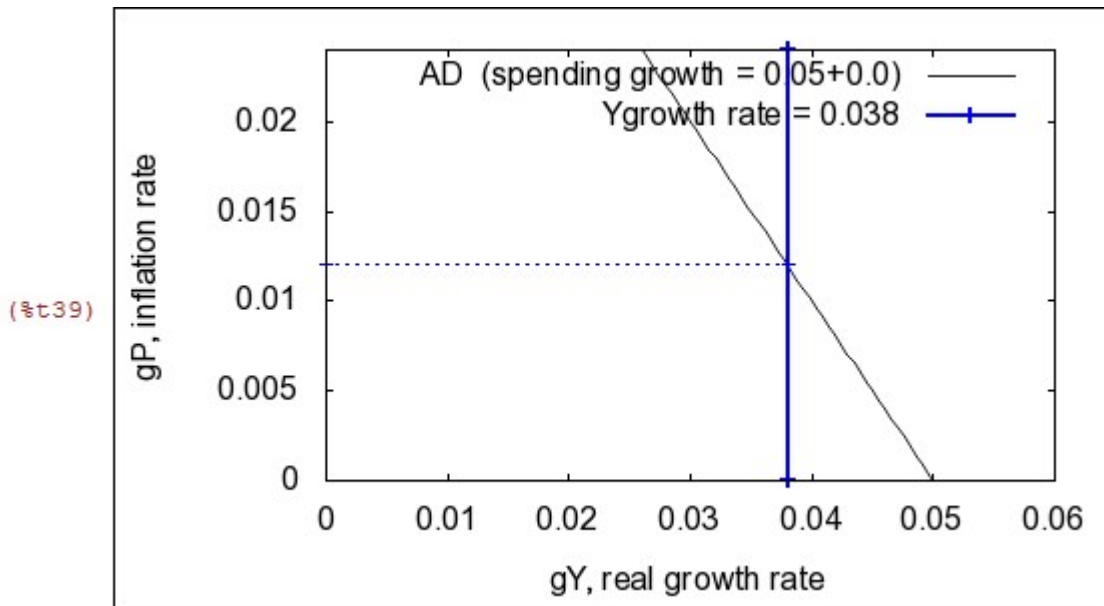
The graph below illustrates the preceding discussion. The long-run aggregate supply curve is a vertical line. It intersects the AD curve at gP = 2.7%.

```
(%i39) wxdraw2d(xlabel="gY, real growth rate", yrange=[0,2*gP0],
ylabel = "gP, inflation rate", key = concat("AD
(spending growth = ",string(gM0),"+", string(gv0),")"),
explicit(AD(gM0, gv0, gY), gY, 0, .06 ), color=blue,
points_joined=true, key = concat("Ygrowth rate =
```

```

",gY0),line_width=2,
points([[gY0,0],[gY0, 2*gP0] ]),
line_type=dots,key="",line_width=1,
points([[0, gP0],[gY0,gP0] ]))
)$

```



The table below shows three values of w and the resulting values of gY and gP . The `cons` command and the `matrix` command are just used to build the table and can be ignored for purposes of following the development of the model.

```

( %
i40) [w1, w2]:[-0.01, 0.01]$
      [gY1,gY2] : [LRAS(gK0, gL0, a0, b0,w1), LRAS(gK0, gL0, a0,
      b0,w2)]$
      [gP1,gP2] : [AD(gM0,gv0, gY1),AD(gM0,gv0, gY2) ]$
      wValues: cons("w", [w1,w0,w2])$
      gYValues: cons("gY",[gY1,gY0, gY2])$
      gPvalues : cons("gP",[gP1, gP0, gP2])$
      matrix(wValues,gYValues,gPvalues);

```

```

( %o46)
      w  -0.01  0.0  0.01
      gY  0.028  0.038  0.048
      gP  0.022  0.012  0.002

```

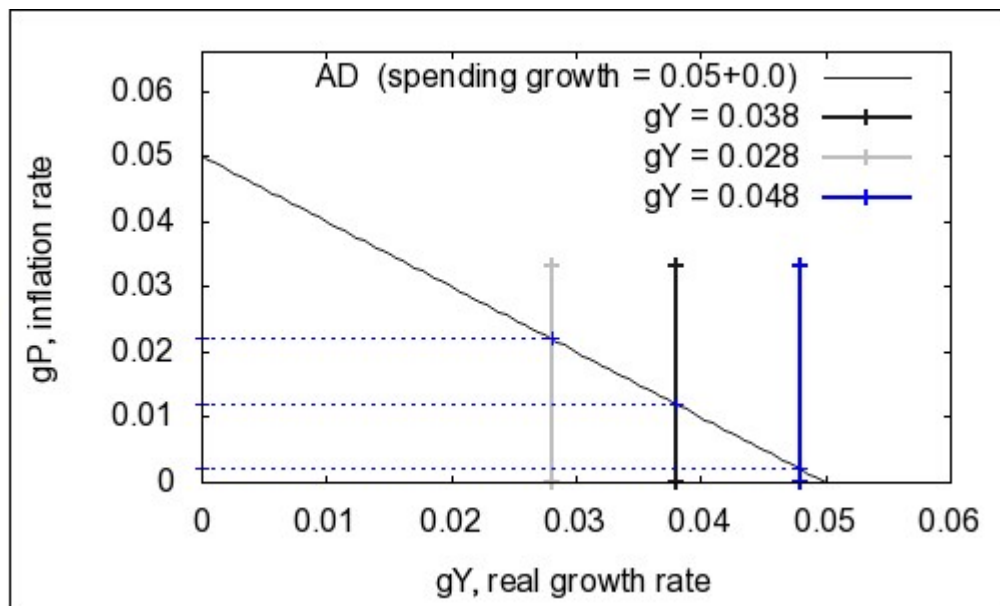
```

( %i47) wxdraw2d(xlabel="gY, real growth rate", yrange=[0,3*gP1],
      ylabel = "gP, inflation rate", key = concat("AD
      (spending growth = ",string(gM0),"+", string(gv0),")"),
      explicit(AD(gM0, gv0, gY), gY, 0, .06 ),
      points_joined=true, key = concat("gY = ",gY0),line_width=2,
      points([[gY0,0],[gY0, 1.5*gP1] ]),
      key = concat("gY = ",gY1),line_width=2,
      color = gray, points([[gY1,0],[gY1, 1.5*gP1] ]),
      key = concat("gY = ",gY2),
      color = blue, points([[gY2,0],[gY2, 1.5*gP1] ]),
      line_type=dots,line_width=1,key="",
      points([[0, gP0],[gY0,gP0] ]), points([[0, gP1],[gY1,gP1] ]),

```

```
points([[0, gP2],[gY2,gP2] ]))$
```

(%t47)



The graph shows that shifts in the LRAS affect both the inflation rate and the rate of real output growth. Exercise: Copy the cell above and delete the commands that relate to the LRAS shift. Then add a second AD curve. Confirm that shifting the AD curve affects the inflationrate only; the growth rate of Y remains constant.

3 Short-run Aggregate Supply

Suppose that a long-run equilibrium has been established and then the system experiences a change in AD. The system will eventually return to the initial gY (nothing has happened that affects the vertical LRAS curve) and the rate of inflation will change by an amount sufficient to ensure this result. The short-run is not nearly so neat, however. In the short run, both gY and gP change as a result of an AD shift.

To illustrate this process, we define a general SRAS curve, the level of which depends on the existing long-run equilibrium. The cell below provides general expressions for gYe and gPe , long-run equilibrium values of gY and gP respectively.

```
(%i48) gYe: LRAS(gK,gL, a, b, w)$
gPe : AD(gM,gv, gYe)$
matrix(["gYe = ",gYe],["gPe = ", gPe]);
```

(%o50)

$$\begin{bmatrix} gYe = & (1-b)gL + b gK + w + a \\ gPe = & gM - (1-b)gL - b gK - w + gv - a \end{bmatrix}$$

As with AD, we state the SRAS by defining its height. We determine the inflation rate, gP , for each value of real output change, gY . We treat the SRAS function as being the initial equilibrium value of gPe , multiplied by a term that values directly with the ratio of gY to the equilibrium value gYe . The exponent c is an elasticity: it is the percentage by

which the inflation rate changes per one-percent change in gY .

The expression above implies that the elasticity of gY with respect to the inflation rate is $1/c$. Thus, our illustrative example assumes that a one percent change in the inflation rate implies a 0.4 percent change in real output.

```
(%i51) gPe*(gY/gYe)^c; radcan(%);
SRAS(gY,a,b,c,gM,gv,gK, gL,w):= ''%;
```

```
(%o51) (gM-(1-b)gL-b gK-w+gv-a)⎛⎜⎝ $\frac{gY}{(1-b)gL+b gK+w+a}$ ⎟⎠^c
```

```
(%o52) 
$$\frac{(gM+(b-1)gL-b gK-w+gv-a) gY^c}{((1-b)gL+b gK+w+a)^c}$$

```

```
(%o53) SRAS(gY,a,b,c,gM,gv,gK,gL,w):=
$$\frac{(gM+(b-1)gL-b gK-w+gv-a) gY^c}{((1-b)gL+b gK+w+a)^c}$$

```

```
(%i54) SRAS(gY,a0,b0,c0,gM0,gv0,gK0, gL0,w0);
```

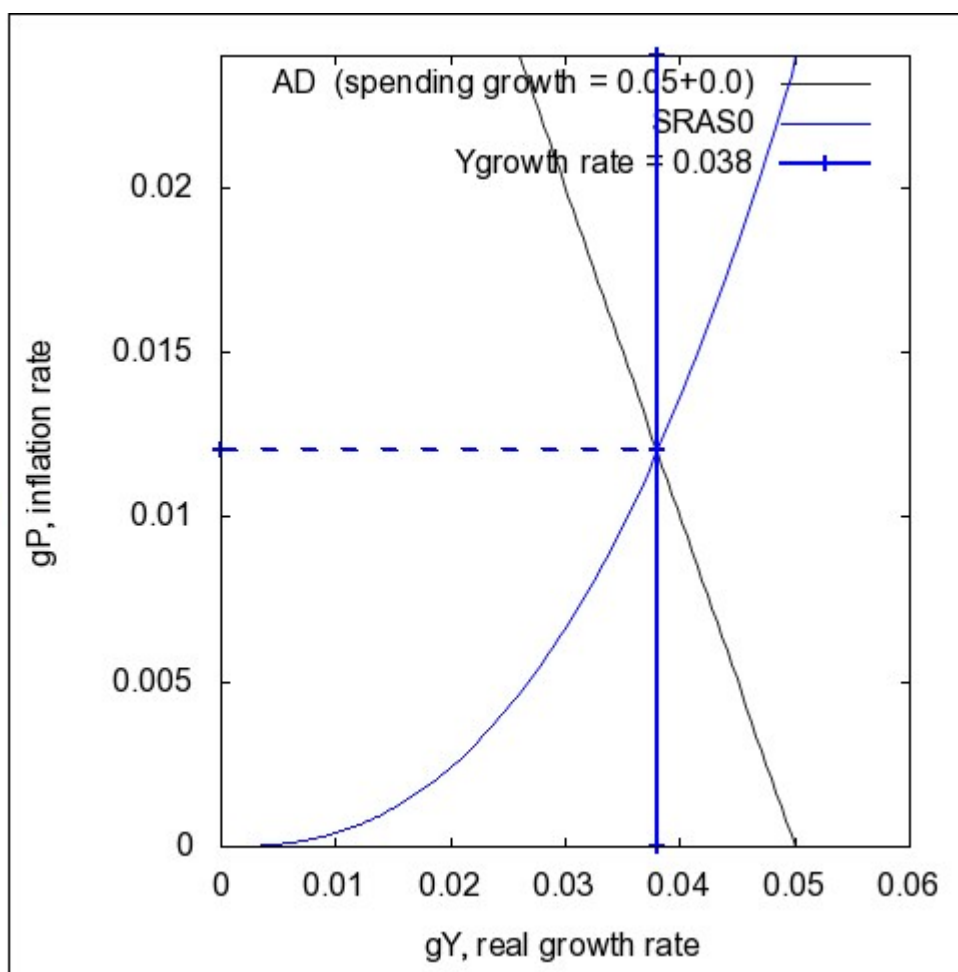
```
(%o54) 
$$\frac{0.012 gY^{c0}}{0.038^{c0}}$$

```

The graph below represents a situation of full equilibrium. The AD curve intersects both the LRAS and the SRAS, so no reason exists for this system to move from the current situation: $gY = 0.038$ and $gP = 0.012$.

```
(%i55) c0:2.5$
wxdraw2d(dimensions=[480,480], xaxis=true, xtics=.01,
yaxis=true, xlabel="gY, real growth rate", yrange=[0,2*gP0],
ylabel = "gP, inflation rate", key = concat("AD
(spending growth = ",string(gM0),"+", string(gv0),")"),
explicit(AD(gM0, gv0, gY), gY, .01, .06 ),
color = blue, key="SRAS0",
explicit(SRAS(gY,a0,b0,c0,gM0,gv0,gK0, gL0, w0),gY,0,.06),
points_joined=true, key = concat("Ygrowth rate =
",gY0),line_width=2,
points([[gY0,0],[gY0, 2*gP0] ]),
line_type=dots,key="",
points([[0, gP0],[gY0,gP0] ])
)$
```

(%t56)



If we stimulate AD with an increase in the rate of expenditure growth we should end up with higher growth and inflation as the AD curve moves up the ASSR curve. We can't use solve, however, to get the short-run equilibrium value for gY . The attempt below shows that gY is expressed as a function of itself, for which no general solution exists. [The solve command seeks an analytical solution. It can fail to find such a solution for either of two reasons. First, as in this example, such a solution might not exist. The second reason is that, though Maxima's catalog of functional types is large, it is not exhaustive. Therefore, Maxima might not have enough information to solve an equation.

```
(%i57) find_root(SRAS(gY,a0,b0,c0,gM0,gv0,gK0, gL0,w0)-  
AD(gM0, gv1, gY), gY, 0, .1);
```

(%o57) 0.0528

When an analytical solution is unattainable, Maxima offers a set of numerical methods, one of which is `find_root`. These approaches require that all parameter values be specified, so that a single number can be found. In the case of `find_root`, Maxima examines the expression at both ends of a range. If the values have the same sign at both ends, Maxima stops and returns an error message. If the signs differ, then Maxima looks for a value such that the function equals zero. We apply `find_root` to the difference between the height of AD and that of SRAS, given the value of $gv1$. The value of gY at which the two curves cross is $gY = 0.052$, approximately. The AD implies that the resulting gP is approximately 0.027.

```
(%i58) gYSR: find_root(SRAS(gY,a0,b0,c0,gM0,gv0,gK0, gL0,w0)=
AD(gM0, gv1, gY), gY, 0.01, 0.09);
gPSR: AD(gM0,gv1,gYSR);
```

```
(%o58) 0.0528
```

```
(%o59) 0.0272
```

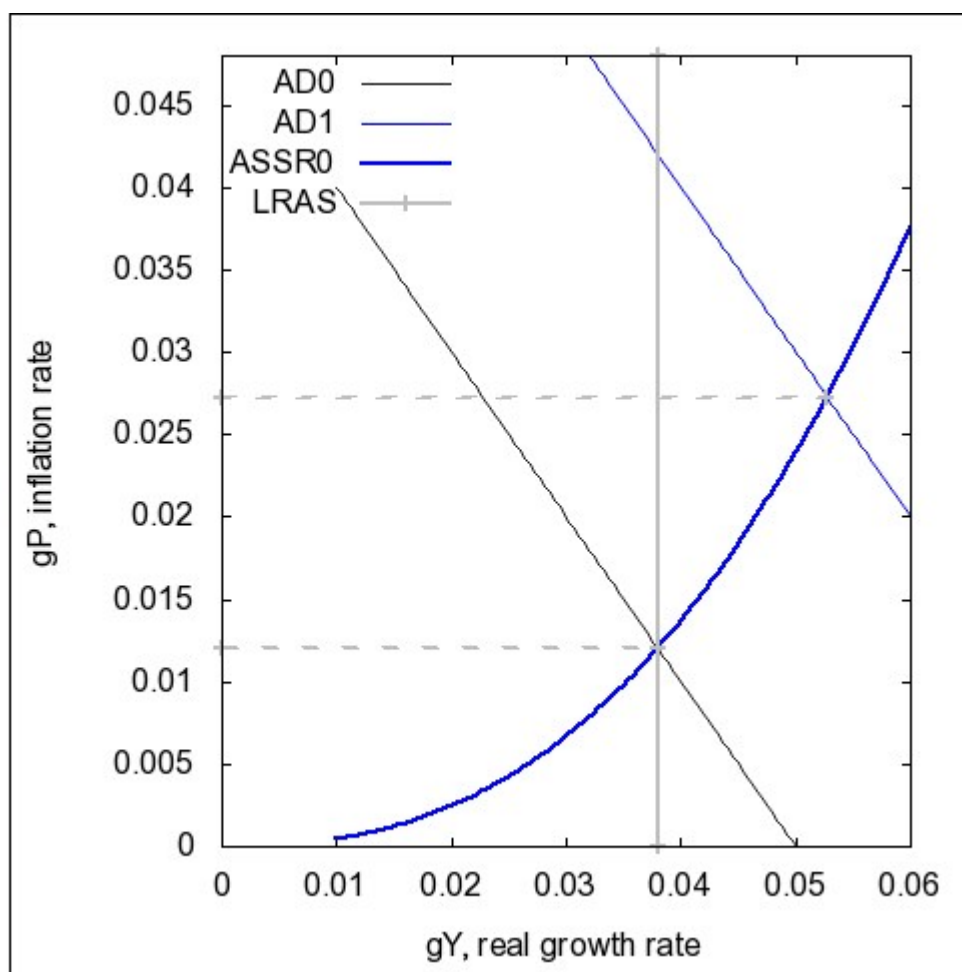
Graphing the new equilibrium allows us to visually verify the solution. The graph below shows the initial AD

```
(%i60) c0:2.5$
wxdraw2d(user_preamble="set key left top",
dimensions=[480,480], xaxis=true, xtics=.01,
yaxis=true, xlabel="gY, real growth rate", yrange=[0,4*gP0],
ylabel = "gP, inflation rate", key = "AD0",

explicit(AD(gM0, gv0, gY), gY, .01, .06 ) ,
color = blue, key = "AD1",

explicit(AD(gM0, gv1, gY), gY, .01, .06 ) ,
line_width = 2, key="ASSR0",
explicit(SRAS(gY,a0,b0,c0,gM0,gv0,gK0, gL0, w0),gY,0.01,.06),
points_joined=true, key = "LRAS", color=gray,
points([[gY0,0],[gY0, 4*gP0] ]),
line_type=dots,key="",
points([[0, gP0],[gY0,gP0] ]),
points([[0, gPSR],[gYSR,gPSR] ]) )$
```

(§t61)



Exercise: Determine the new long-run gP value if the new AD curve persists.

Created with [wxMaxima](#).

Edited with [KompoZer](#).

PDF file produced by [wvhtml2pdf](#).