

First Order Difference Equations

Dowling17.wmx

TABLE OF CONTENTS

PREFACE	1
DEFINITIONS AND CONCEPTS	2
GENERAL SOLUTION	7
STABILITY CONDITIONS	10
LAGGED INCOME DETERMINATION MODEL	15
THE COBWEB MODEL	16
THE HARROD MODEL	24
OTHER ECONOMIC APPLICATIONS	26
PHASE DIAGRAM ANALYSIS OF STABILITY	30

```
(%i5) load(draw)$ set_draw_defaults(line_width=2, grid = [2,2], point_type = filled_circle,
      background_color = light_gray, draw_realpart=false)$
      fpprintprec:5$ ratprint:false$ kill(all)$
```

```
(%i1) load ("Econ2.mac");
```

```
(%o1) c:/work5/Econ2.mac
```

1 **Preface**

This worksheet is one of a number of wxMaxima files available in the section Economic Analysis with Maxima on my CSULB webpage.

This update corrects some syntax errors, misspelled words, and also improves the clarity of the discussion. We have also used the draw2d function points(xL,yL) for simplicity.

We use Maxima to solve a few of the problems (and draw some of the plots) in Ch. 17 of the supplemental text: Introduction to Mathematical Economics, 3rd ed, (Schaum's Outline Series), by Edward T. Dowling (1992, 2001), McGraw-Hill. This modestly priced text is a bargain with many worked out examples. You should compare the examples worked out "by hand" in this text with what we do using Maxima.

We have slightly changed some of the symbols used in particular problems.

Ted Woollett
<https://home.csulb.edu/~woollett/>
 woollett@charter.net
 Dec. 13, 2021

2 Definitions and Concepts

A difference equation expresses a relationship between a dependent variable and a lagged independent variable (or variables) which changes at discrete intervals of time, for example,

$$I[t] = F(Y[t - 1])$$

with $F(x)$ being some arbitrary function, where I and Y are each measured at the end of each year. Investment during year 4 is some function of national income during year 3, for example: $I[4] = F(Y[3])$, etc.

The order of a difference equation is determined by the greatest number of periods lagged.

A first-order difference equation contains only a time lag of one period, as above.

A second-order difference equation involves a time lag of two periods, in addition to (perhaps) also a time lag of one period, etc.

The change in y as t changes from t to $t + 1$ is called the "first difference of y ".

The FIRST DIFFERENCE of y is written :

$$\Delta y / \Delta t = \Delta y[t] = y[t + 1] - y[t] = \text{change in } y \text{ per period over the time } (t, t+1).$$

In Maxima, $y[t]$ is a (hash) array and `wxMaxima` will display the symbol using subscript notation (if `display2d` is set to `true`, which is the default).

```
(%i2) y[t];
```

```
(%o2) yt
```

```
(%i3) display2d;
```

```
(%o3) true
```

2.1 Example 1

Each of the following is a difference equation of the order indicated:

Order 2:

```
(%i4) I[t] = a* ( Y[t-1] - Y[t-2]);
```

```
(%o4) It = a ( Yt-1 - Yt-2)
```

Order 1:

$$(\%i5) \quad Q[t] = a + b \cdot P[t-1];$$

$$(\%o5) \quad Q_t = b P_{t-1} + a$$

Order 3:

$$(\%i6) \quad y[t+3] - 9 \cdot y[t+2] + 2 \cdot y[t+1] + 6 \cdot y[t] = 8;$$

$$(\%o6) \quad y_{t+3} - 9 y_{t+2} + 2 y_{t+1} + 6 y_t = 8$$

Order 1:

$$(\%i7) \quad \Delta y[t] = 5 \cdot y[t];$$

$$(\%o7) \quad \Delta y_t = 5 y_t$$

The last equation is the same as:

$$(\%i8) \quad y[t+1] - y[t] = 5 \cdot y[t];$$

$$(\%o8) \quad y_{t+1} - y_t = 5 y_t$$

which is the same as:

$$(\%i9) \quad y[t+1] = 6 \cdot y[t];$$

$$(\%o9) \quad y_{t+1} = 6 y_t$$

2.2 First Few Values, history1 (y0, fnext), find_root(expr,x,a,b)

The first few values of a first order difference equation can be found starting from an initial value, $y_0 = y[t=0]$.

For example, suppose $y[t] = 0.5 \cdot y[t-1]^{-1/3}$, a nonlinear difference equation. We take $y_0 = 0.75$ as the starting value.

$$(\%i12) \quad \text{kill}(y_0, y_1, y_2, y_3, y_4) \$$$

$$f(x) := 0.5 \cdot x^{-1/3};$$

$$[y_0 : 0.75, y_1 : f(y_0), y_2 : f(y_1), y_3 : f(y_2), y_4 : f(y_3)];$$

$$(\%o11) \quad f(x) := 0.5 x^{-1/3}$$

$$(\%o12) \quad [0.75, 0.55032, 0.61014, 0.58951, 0.59631]$$

$$(\%i13) \quad \text{history1}(_y_0, \text{fnext}) := ([y_0 : _y_0, y_1 : \text{fnext}(y_0), y_2 : \text{fnext}(y_1), y_3 : \text{fnext}(y_2), y_4 : \text{fnext}(y_3)]) \$$$

For `fnext`, we use our definition of the Maxima function $f(x)$ just above, and use the same starting value y_0 .

```
(%i14) history1 (0.75, f);
```

```
(%o14) [0.75, 0.55032, 0.61014, 0.58951, 0.59631]
```

Decreasing y_0 , but using the same `fnext`,

```
(%i15) history1 (0.55, f);
```

```
(%o15) [0.55, 0.61026, 0.58947, 0.59632, 0.59403]
```

We anticipate the discussion of phase diagram analysis for a first order difference equation (in the last section), and proceed to make a plot of y_t vs y_{t-1} , in which y_t stands for $y[t]$, and y_{t-1} stands for $y[t-1]$. If there exists a stable value of $y[t]$, it should have the property $y[t] = y[t-1]$ and hence lie on the 45 degree line in the first quadrant, which is drawn in the color red below.

For our example, which uses the definition of the Maxima function $f(x)$ above,

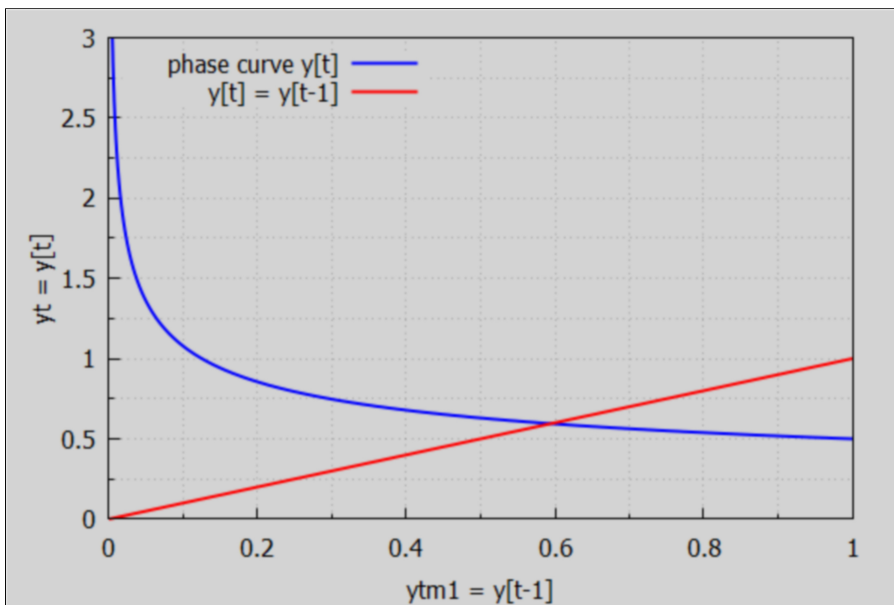
```
(%i16) yt : f(ytm1);
```

```
(yt) 
$$\frac{0.5}{ytm1^{1/3}}$$

```

```
(%i17) wxdraw2d (xlabel = "ytm1 = y[t-1]", ylabel = "yt = y[t]", key_pos = top_left,
  xrange = [0, 3],
  key = "phase curve y[t]", explicit (yt, ytm1, 0, 1), color = red, key = "y[t] = y[t-1]",
  explicit (ytm1, ytm1, 0, 1))$
```

(%t17)



The Maxima function `find_root (expr, x, a, b)` looks for a numerical root of $\text{expr} = 0$ as x varies in the closed interval $[a,b]$.

`find_root (A(x) = B(x), x, a, b)` looks for the root of $A(x) - B(x) = 0$ for x in $[a,b]$.

Let y_{star} be the root of the equation $y = f(y)$, in which the variable is y , and we are again using our definition of the Maxima function $f(x)$ above.

```
(%i18) ystar : find_root (y = f(y), y, 0.4, 0.8);
(ystar) 0.5946
```

The first derivative of the slope of the curve $f(y)$, ie., the curve of y_t vs y_{t-1} , evaluated at a point where $y = f(y)$, tells us whether the point y_{star} is a stable equilibrium for which the given difference equation tends to drive the next value of y (for the next value of t). The "time" t is in the background of this analysis, and in the last section we illustrate the approach to a stable point like y_{star} , using a (cobweb) plot.

We have defined y_t above as a specific Maxima expression depending on y_{t-1} :

```
(%i19) yt;
(%o19)  $\frac{0.5}{y_{t-1}^{1/3}}$ 
```

The local slope of the curve $y_t(y_{t-1})$ is the first derivative of y_t wrt y_{t-1} .

```
(%i20) slope : diff (yt, ytm1);
(slope)  $-\frac{0.16667}{y_{t-1}^{4/3}}$ 
```

For $y_{t-1} > 0$, the slope is thus always a negative number.

What is the local slope value for the "equilibrium point" $y_{t-1} = y_{\text{star}}$?

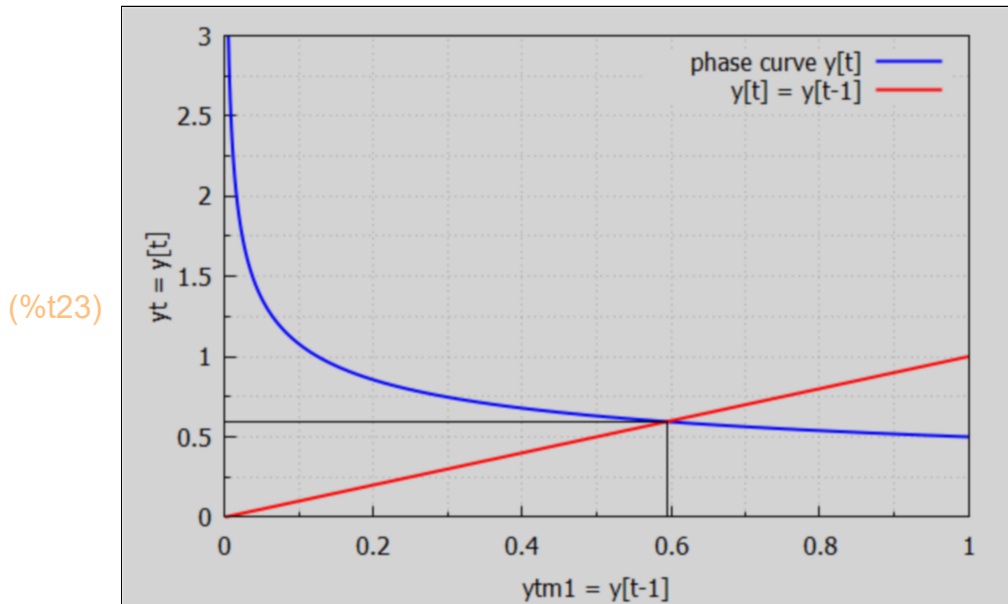
If the MAGNITUDE of the slope at y_{star} is less than 1, then the given difference equation will drive y_t to the equilibrium point.

```
(%i21) ystar;
(%o21) 0.5946
```

```
(%i22) at (slope, ytm1 = ystar);
(%o22) -0.33333
```

The magnitude of the slope of the y_t vs y_{t-1} curve at the value $y = y_{\text{star}}$ is less than 1, so a solution beginning with some y_0 value will converge to $y = y_{\text{star}}$. The slope at y_{star} is a negative number, so successive values of y will oscillate around $y = y_{\text{star}}$ in the process of convergence.

```
(%i23) wxdraw2d (xlabel = "ytm1 = y[t-1]", ylabel = "yt = y[t]", xrange = [0, 3],
key = "phase curve y[t]", explicit (yt, ytm1, 0, 1), color = red, key = "y[t] = y[t-1]",
explicit (ytm1, ytm1, 0, 1), color = black, key = "", line_width = 1,
explicit (ystar, ytm1, 0, ystar), parametric (ystar, yy, yy, 0, ystar))$
```



2.3 Solution of a Difference Equation, Ex. 2

The "solution" of a difference equation defines $y[t]$ for every discrete value of $t = 0, 1, 2, \dots$ and does not contain a difference expression.

Given that the initial value of y is $y_0 = y[0]$, in the example difference equation

$$y[t+1] = b \cdot y[t],$$

a solution is found as follows. By successive substitutions of $t = 0, 1, 2, 3$, etc, we get

$$y[1] = b \cdot y[0],$$

$$y[2] = b \cdot y[1] = b \cdot (b \cdot y[0]) = b^2 \cdot y[0],$$

$$y[3] = b \cdot y[2] = b^3 \cdot y[0],$$

.....

$$y[t] = b^t \cdot y[0], \text{ for } t = 0, 1, 2, 3, \dots$$

This method is called the "iterative method." Since $y[0]$ is constant, note the crucial role b plays in determining values for $y[t]$ as t increases.

3 General Solution for $y[t] = b*y[t-1] + a$

Given a first order difference equation which is LINEAR, (i.e., all the variables are raised to the first power only and there are no cross products)

$$y[t] = b*y[t-1] + a,$$

the general formula for a "definite solution", with y_0 equal to $y[0]$, is

$$\text{if } b \neq 1 \text{ then } y[t] = (y_0 - a/(1-b))*b^t + a/(1-b),$$

$$\text{otherwise, for } b = 1, y[t] = y_0 + a*t.$$

If no initial condition is given, the formula for a general solution can be given in terms of some (as yet undefined) constant A as:

$$\text{if } b \neq 1 \text{ then } y[t] = A*b^t + a/(1-b),$$

$$\text{otherwise, for } b = 1, y[t] = A + a*t.$$

Note that $b^0 = 1$ for any b .

3.1 Maxima Derivation of the General Solution

The Maxima function `solve_rec` (eqn, var, optional-initial-assignments) can be used to solve for hypergeometrical solutions of a linear recurrence equation (eqn) with polynomial coefficients in the variable var.

Optional are one or more "initial conditions" of the form $y[0] = 1, y[3] = 2$, etc, if var = $y[t]$.

To use this function first use `load ("solve_rec");` which is a "contributed package" designed to solve recurrence relations.

```
(%i24) load ("solve_rec");
```

```
(%o24) C:/maxima-5.43.2/share/maxima/5.43.2/share/solve_rec/solve_rec.mac
```

The syntax is

```
solve_rec ( eqn, var, [init] ),
```

in which the brackets around `init` mean that one or more initial condition statements of the form $y[0] = y_0, y[2] = -5$, etc are optional; only `eqn` and `var` are required arguments.

```
(%i26) kill (y0)$
```

```
soln : solve_rec (y[t] = b*y[t-1] + a, y[t], y[0] = y0);
```

```
(soln)  $y_t = b^t y_0 + \frac{a b^t}{b-1} - \frac{a}{b-1}$ 
```

The Maxima function grind prints out the expression in display2d = false style:

```
(%i27) grind(soln)$
      y[t] = b^t*y0+(a*b^t)/(b-1)-a/(b-1)$
```

```
(%i28) rhs(soln);
```

```
(%o28) b^t y0 +  $\frac{a b^t}{b-1}$  -  $\frac{a}{b-1}$ 
```

The coefficient of b^t can be picked out as follows

```
(%i29) coeff(%, b^t);
```

```
(%o29) y0 +  $\frac{a}{b-1}$ 
```

Replacing $a/(b-1)$ with $-a/(1-b)$ throughout, we get the solution given above, for the case that $b \neq 1$.

If we don't supply the optional initial condition, solve_rec produces

```
(%i30) soln : solve_rec (y[t] = b*y[t-1] + a, y[t]);
```

```
(soln) y_t =  $\frac{a b^t}{b-1}$  + %k_1 b^t -  $\frac{a}{b-1}$ 
```

The solve_rec algorithm returns a solution involving an arbitrary constant %k. If we define another arbitrary constant A by:

$$A = a/(b-1) + \%k,$$

which is the coefficient of b^t , we get $y[t] = A*b^t + a/(1-b)$, as given above for $b \neq 1$.

Note that solve_rec did not consider separately the special case $b = 1$.

By setting $b = 1$, we get that case as a solution from solve_rec.

```
(%i31) solve_rec (y[t] = y[t-1] + a, y[t], y[0] = y0);
```

```
(%o31) y_t = y0 + a t
```

which agrees with our $b = 1$ solution above.

3.2 Function DE1(a,b,t,y0) or DE1(a,b,t)

The software file Econ2.mac contains the definition of the Maxima function DE1.

DE1 (a, b, t, y0) returns the value of $y[t]$ governed by the first order difference equation (DE) $y[t] = b*y[t-1] + a$, with $y[0] = y0$.

DE1 (a, b, t) returns the general solution for $y[t]$ governed by the first order difference equation (DE): $y[t] = b*y[t-1] + a$, in terms of an arbitrary constant %A.

3.3 Example 3

Consider the difference equation

$$y[t] = -7*y[t-1] + 16, \text{ with } y[0] = 5$$

in which $a = 16$, the base $b = -7$, and $y0 = 5$.

Let de be a Maxima expression returned by DE1.

```
(%i32) de : DE1 (16, -7, t, 5);
```

```
(de) 3 (-7)^t + 2
```

To define an ordinary Maxima function $y(t)$ at this point, we need to precede the expression de with TWO single quotes (') to force evaluation.

```
(%i33) y(t) := "de;
```

```
(%o33) y(t) := 3 (-7)^t + 2
```

```
(%i34) grind(y(t))$
```

```
3*(-7)^t+2$
```

Check this definition at $t = 0$ and $t = 1$.

```
(%i35) map ('y, [0,1]);
```

```
(%o35) [5, -19]
```

which is equivalent to

```
(%i36) [y(0), y(1)];
```

```
(%o36) [5, -19]
```

$y[1]$'s value should be given in terms of this ordinary Maxima function $y(t)$ using $y(1) = -7*y(0) + 16$, using the given difference equation which must be satisfied for discrete values of $t = 0, 1, 2, \dots$

```
(%i37) -7*5 + 16;
(%o37) -19
```

which is the correct value of $y[1]$.

Without supplying the fourth arg y_0 when calling DE1, we get:

```
(%i38) de : DE1 (16, -7, t);
(de) %A (-7)^t + 2
```

which involves the arbitrary constant $\%A$.

4 Stability Conditions: General Characterization

The exponential expression b^t will generate seven qualitatively different time paths depending on the nature of the base b . The general solution

$$y[t] = A*b^t + c$$

will be dynamically stable only if the "complementary function" $A*b^t$ approaches 0 as t approaches infinity. This then all depends on the general nature of b , as illustrated in the following plots.

If the magnitude of b is greater than 1 ($|b| > 1$), the time path will explode and move farther and farther away from equilibrium. If the magnitude of b is less than 1 ($|b| < 1$), the time path will be damped and will move toward equilibrium.

If b is negative ($b < 0$), the time path will oscillate between positive and negative values. If b is positive ($b > 0$), the time path will be non-oscillating.

Here we can make some plots for different values of b , in the simple case $y[t] = b^t$, implied by the difference equation $y[t] = b*y[t-1]$, with $y[0] = 1$.

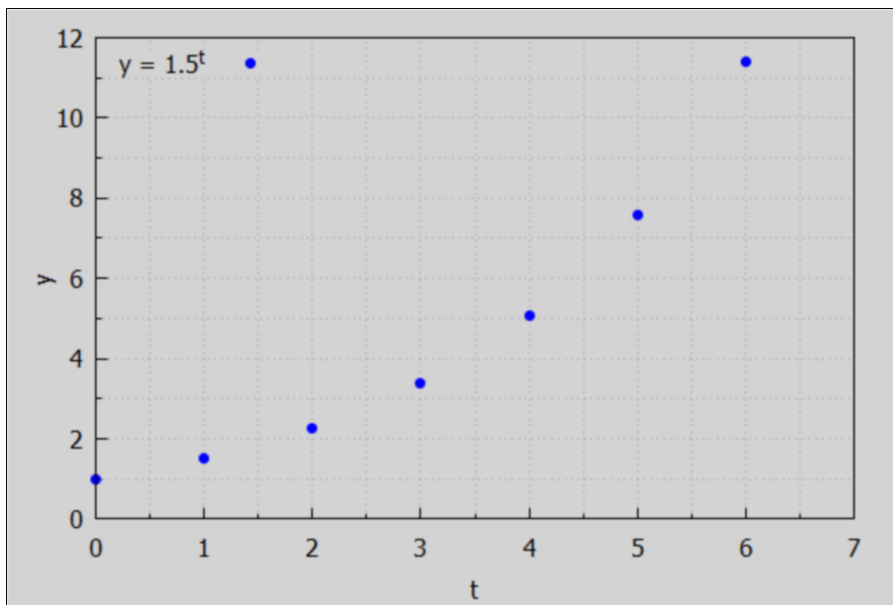
We use the draw2d function points (x_L , y_L) to plot the evolution in time.

```
(%i40) tL : makelist (t, t, 0, 6);
        yL : makelist (1.5^t, t, 0, 6);
(tL) [0, 1, 2, 3, 4, 5, 6]
(yL) [1.0, 1.5, 2.25, 3.375, 5.0625, 7.5938, 11.391]
```

Note that at the top of this file we defined a default point type as a filled circle for draw2d.

```
(%i41) wxdraw2d(yrange = [0,12], xrange = [0, 7], xlabel = "t",
  ylabel = "y", key_pos = top_left, key = "y = 1.5^t",
  points (tL, yL))$
```

```
(%t41)
```



The jump in the value of b^t gets larger each successive discrete value of t we use. This is an example of an "explosive instability."

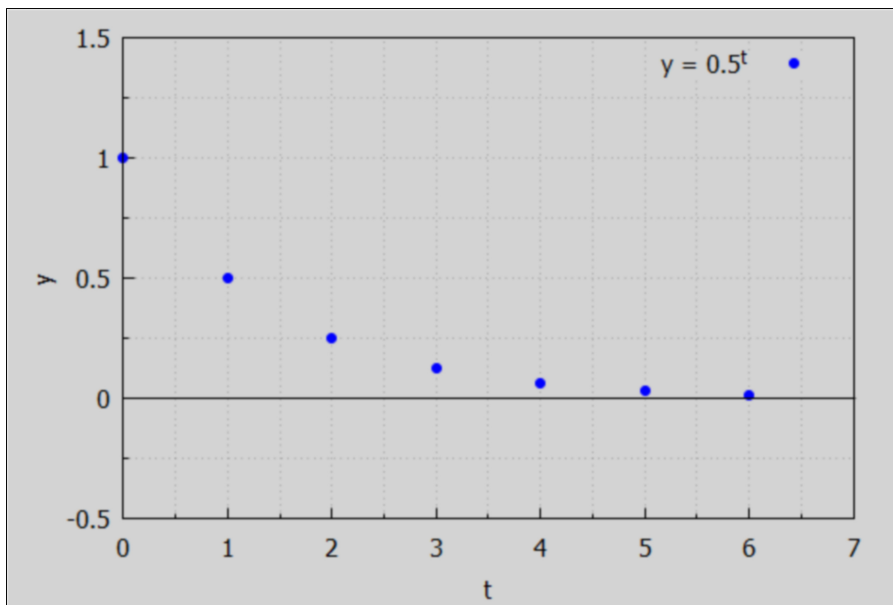
As a contrast, we show convergence to zero with a value of b in the range $0 < b < 1$, say $b = 0.5$, as an example. For $t = 0$, $0.5^0 = 1$.

```
(%i42) yL : makelist (0.5^t, t, 0, 6);
```

```
(yL) [1.0,0.5,0.25,0.125,0.0625,0.03125,0.015625]
```

```
(%i43) wxdraw2d(yrange = [-0.5, 1.5], xrange = [0, 7], xlabel = "t",
  ylabel = "y", key = "y = 0.5^t", points (tL, yL), key = "", color = black,
  line_width = 1, explicit (0,t,0,7))$
```

(%t43)



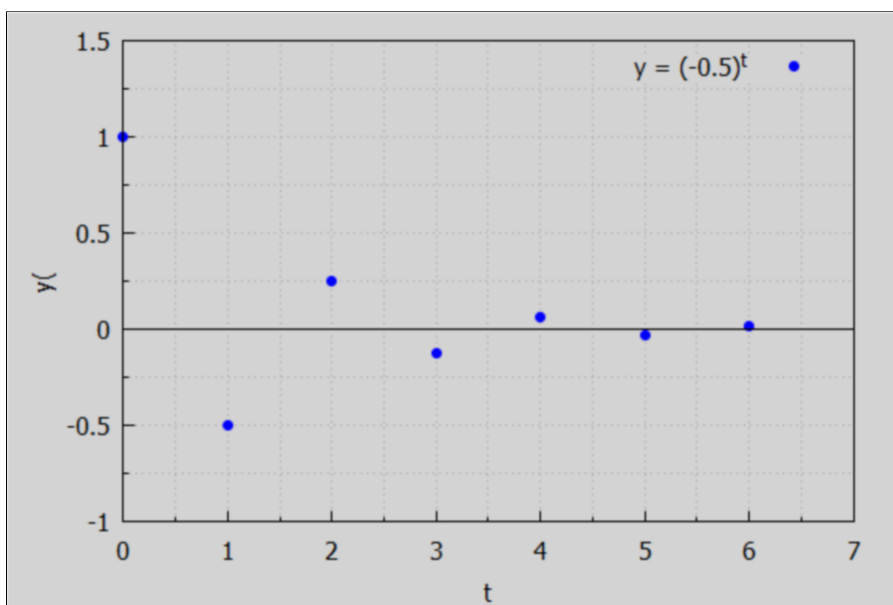
For $-1 < b < 0$ we get oscillating convergence of b^t .
Here is an example with $b = -0.5$.

```
(%i44) yL : makelist ((-0.5)^t, t, 0, 6);
```

```
(yL) [1.0, -0.5, 0.25, -0.125, 0.0625, -0.03125, 0.015625]
```

```
(%i45) wxdraw2d(yrange = [-1, 1.5], xrange = [0, 7], xlabel = "t",
  ylabel = "y", key = "y = (-0.5)^t", points (tL, yL), key = "",
  line_width = 1, color = black, explicit (0, t, 0, 7))$
```

(%t45)



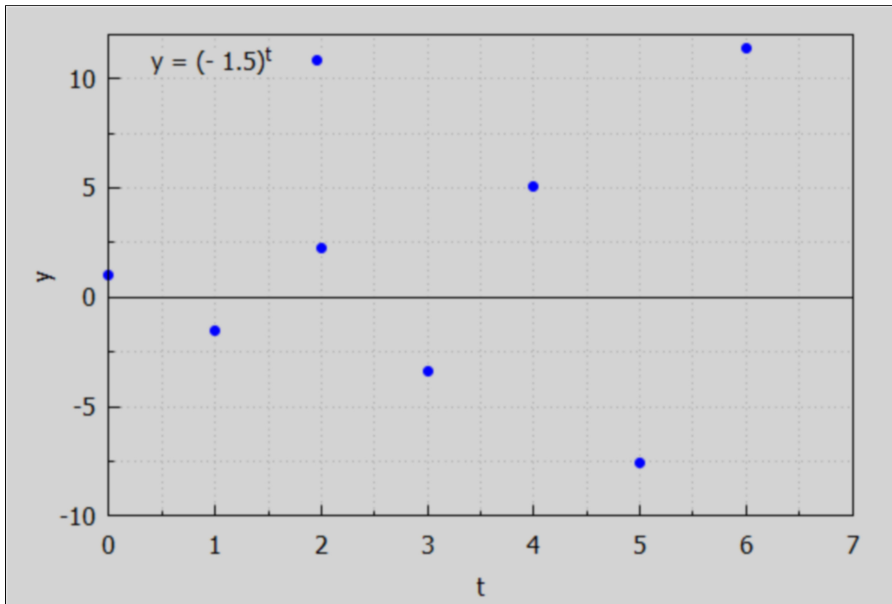
With $b < -1$ we get oscillating explosive instability.
Here is an example with $b = -1.5$.

```
(%i46) yL : makelist ((-1.5)^t, t, 0, 6);
```

```
(yL) [1.0, -1.5, 2.25, -3.375, 5.0625, -7.5938, 11.391]
```

```
(%i47) wxdraw2d(yrange = [-10, 12], xrange = [0, 7], xlabel = "t",  
ylabel = "y", key_pos = top_left, key = "y = (- 1.5)^t",  
points (tL, yL), key = "", line_width = 1, color = black, explicit (0, t, 0, 7))$
```

```
(%t47)
```



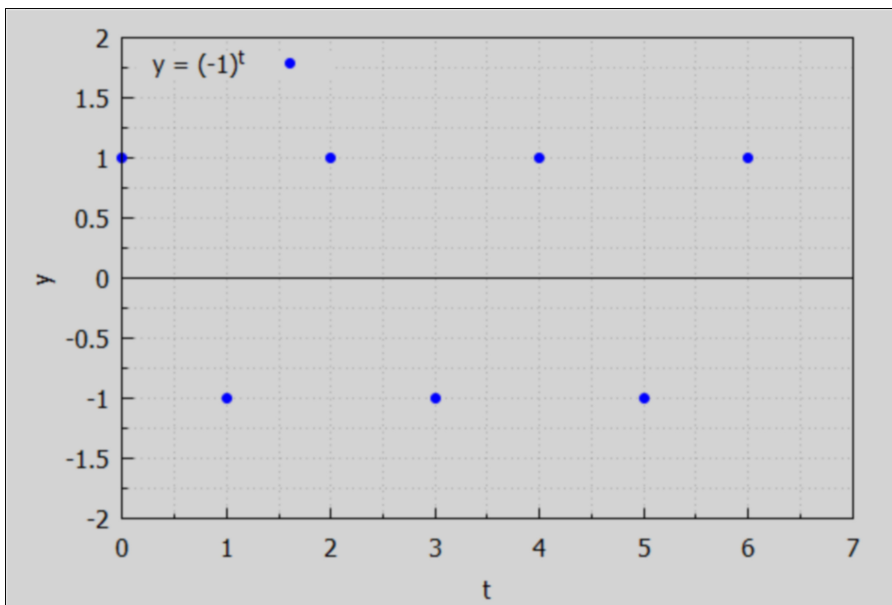
The borderline case $b = -1$ shows oscillating behavior with no convergence and no growth trend.

```
(%i48) yL : makelist ((-1)^t, t, 0, 6);
```

```
(yL) [1, -1, 1, -1, 1, -1, 1]
```

```
(%i49) wxdraw2d(yrange = [-2, 2], xrange = [0, 7], xlabel = "t",
  ylabel = "y", key_pos = top_left, key = "y = (-1)^t",
  points (tL, yL), key = "", line_width = 1, color = black, explicit (0, t, 0, 7))$
```

(%t49)

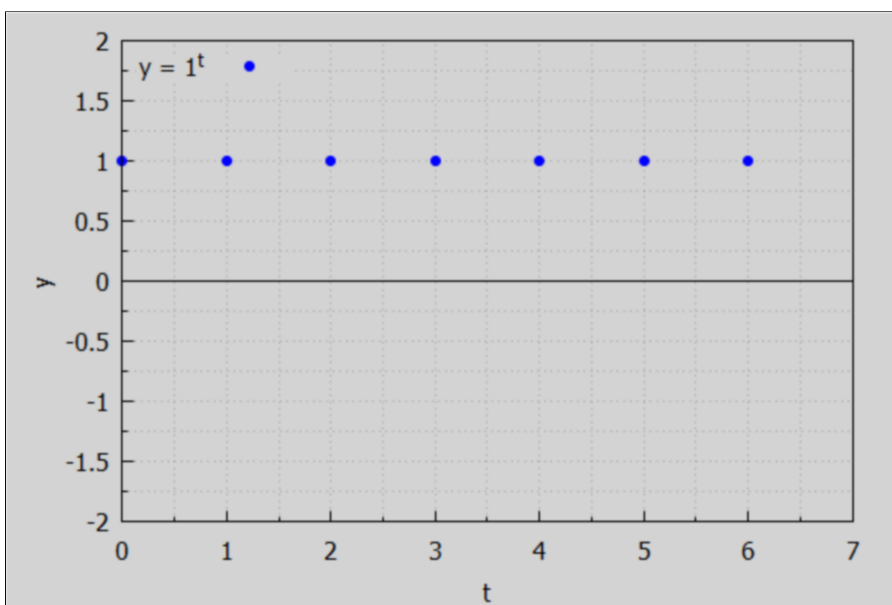


The other borderline case is $b = 1$, for which b^t is a flat line at $y = 1$.

```
(%i50) yL : makelist (1^t, t, 0, 6);
(yL)    [1,1,1,1,1,1,1]
```

```
(%i51) wxdraw2d(yrange = [-2, 2], xrange = [0, 7], xlabel = "t",
  ylabel = "y", key_pos = top_left, key = "y = 1^t",
  points (tL, yL), key = "",
  line_width = 1, color = black, explicit (0, t, 0, 7))$
```

(%t51)



5 Lagged Income Determination Model

In the simple income determination model of Dowling Ch. 2, Sec. 2.3, there were no lags. Now assume that consumption is a function of the previous period's income, and national income Y is the sum of consumption plus investment, and that investment per year is constant.

```
(%i54) /* these are not assignment statements! */
```

$$C[t] = C[0] + c*Y[t-1];$$

$$Y[t] = C[t] + I[t];$$

$$I[t] = I[0];$$

```
(%o52)  $C_t = c Y_{t-1} + C_0$ 
```

```
(%o53)  $Y_t = I_t + C_t$ 
```

```
(%o54)  $I_t = I_0$ 
```

Writing $Y[t]$ in standard form, $Y[t] = c*Y[t-1] + C[0] + I[0]$, and letting C_0 stand for $C[0]$, I_0 stand for $I[0]$, by comparison with our standard form: $y[t] = b*y[t-1] + a$, we identify $b = c$, $a = C_0 + I_0$. Let Y_0 stand for $Y[0]$, the optional initial value arg in our Maxima function DE1 defined in Econ2.mac.

```
(%i55) DE1 (C0 + I0, c, t, Y0);
```

```
(%o55)  $\left( Y_0 - \frac{I_0 + C_0}{1 - c} \right) c^t + \frac{I_0 + C_0}{1 - c}$ 
```

With the "propensity to consume" $c > 0$ and $c < 1$, this solution converges to $(I_0 + C_0)/(1-c)$ with no oscillation, based on our previous examples.

As a numerical example, assume $C_0 = 200$, $I_0 = 100$, $c = 0.9$, $Y_0 = 4500$.

```
(%i56) de : DE1 (300, 0.9, t, 4500);
```

```
(de)  $1500.0 0.9^t + 3000.0$ 
```

Use two single quotes (') in front of the expression de to allow us to define a Maxima function $Y(t)$ which can be mapped onto a list of times.

```
(%i57) Y(t) := "de;
```

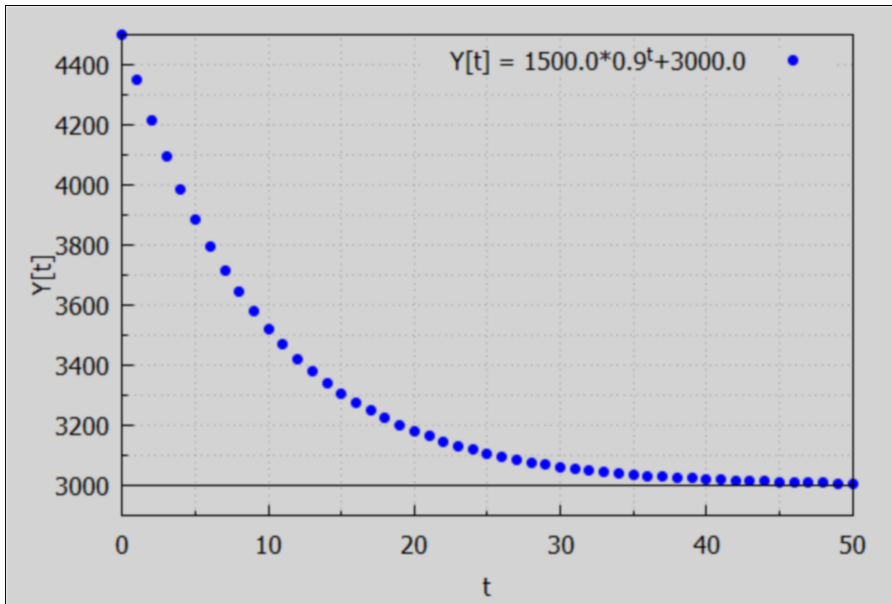
```
(%o57)  $Y(t) := 1500.0 0.9^t + 3000.0$ 
```

```
(%i59) tL : makelist (t, t, 0, 50)$
```

```
YL : map ('Y, tL)$
```

```
(%i60) wxdraw2d(yrange = [2900, 4500], xrange = [0, 50], xlabel = "t",
  ylabel = "Y[t]", key = sconcat ("Y[t] = ", Y(t), points (tL, YL),
  color = black, line_width = 1, key = "", explicit (3000, t, 0, 50))$
```

```
(%t60)
```



6 The Cobweb Model

For many products, such as agricultural commodities, which are planted a year before marketing, current supply depends on last year's prices. This poses interesting stability questions. Let $Q_d[t]$ be the quantity demanded function in terms of prices and $Q_s[t]$ be the quantity supplied function depending on prices the previous year. The equilibrium price during year t ($t = 0, 1, 2, \dots$) is then found by equating $Q_d[t]$ and $Q_s[t]$.

```
(%i63) Qd[t] : c + b*P[t];
  Qs[t] : g + h*P[t-1];
  solns : solve (Qd[t] = Qs[t], P[t]);
```

```
(Qd[t]) b P_t + c
```

```
(Qs[t]) h P_{t-1} + g
```

```
(solns) [P_t = \frac{h P_{t-1} + g - c}{b}]
```

```
(%i64) solns : expand (solns);
```

```
(solns) [P_t = \frac{h P_{t-1}}{b} + \frac{g}{b} - \frac{c}{b}]
```

Use $DE1(a,b,r,y_0)$ to find the solution for $P[t]$, assuming P_0 is the initial value $P[0]$. Here "a" = $(g-c)/b$, "b" = h/b .

(%i65) DE1 ((g-c)/b, h/b, t, P0);

$$(%o65) \left(\frac{h}{b}\right)^t \left(P0 - \frac{g-c}{b\left(1-\frac{h}{b}\right)}\right) + \frac{g-c}{b\left(1-\frac{h}{b}\right)}$$

Evidently the equilibrium price is $P_e = (g-c)/(b-h)$, which appears inside the first term as well as the last term. Let's show an equation with $P[t]$ expressed in terms of P_e .

(%i66) P[t] = (P0 - Pe)*(h/b)^t + Pe;

$$(%o66) P_t = (P0 - Pe) \left(\frac{h}{b}\right)^t + Pe$$

(%i67) de : rhs(%);

$$(de) (P0 - Pe) \left(\frac{h}{b}\right)^t + Pe$$

Now bind the symbol P_e to its symbolic value:

(%i68) Pe : (g - c)/(b - h);

$$(Pe) \frac{g-c}{b-h}$$

To get de fully evaluated, incorporating the above assignment of the symbol P_e , we use two single quotes (') in front of de .

(%i69) de : "de";

$$(de) \left(P0 - \frac{g-c}{b-h}\right) \left(\frac{h}{b}\right)^t + \frac{g-c}{b-h}$$

If $|h| > |b|$, then $|h/b| > 1$, and the time path $P[t]$ explodes.

If $|h| = |b|$, then $|h/b| = 1$ and the time path $P[t]$ oscillates without convergence.

With an ordinary "negative demand function" ($b < 0$) and "positive supply function" ($h > 0$), $h/b < 0$, and the time path will oscillate.

If $|h| < |b|$, then we get convergence to P_e in this ordinary case since $|h/b| < 1$.

Quoting Dowling here:

"In short, when $Q = f(P)$ in supply-and-demand analysis, as is common in mathematics, the supply curve must be flatter than the demand curve for stability. See Example 7 and Problems 17.21 to 17.25. But if $P = f(Q)$, as is typical in economics, the reverse is true; The demand curve must be flatter (or more elastic), than the supply curve if the model is to be stable."

As a numerical example, (we declare t an integer here after finding this version of Maxima has some trouble with combining factors each raised to the power t .

```
(%i71) declare (t, integer)$
      deEx : subst ([c = 86, b = -0.8, g = -10, h = 0.2], de);
(deEx) (P0-96.0) (-1.25)^t 0.2^t+96.0

(%i72) 1.25*0.2;
(%o72) 0.25
```

Maxima is having trouble combining factors with the same exponent (t) here.

Since $-1.25*0.2 = -0.25$, our result is
 $P[t] = (P_0 - 96)*(-0.25)^t + 96$

With the base "b" equal to -0.25, we have oscillatory convergence to the equilibrium price $P_e = 96$ in this numerical example.

6.1 Visualizing the Cobweb

We first ignore time lags and draw $Q(P)$ curves (lines) for both demand (D) and supply (S), in which the supply curve is "flatter" than the demand curve.

```
(%i74) Qd : 86 - 0.8*P;
      Qs : -10 + 0.2*P;
(Qd) 86 - 0.8 P
(Qs) 0.2 P - 10
```

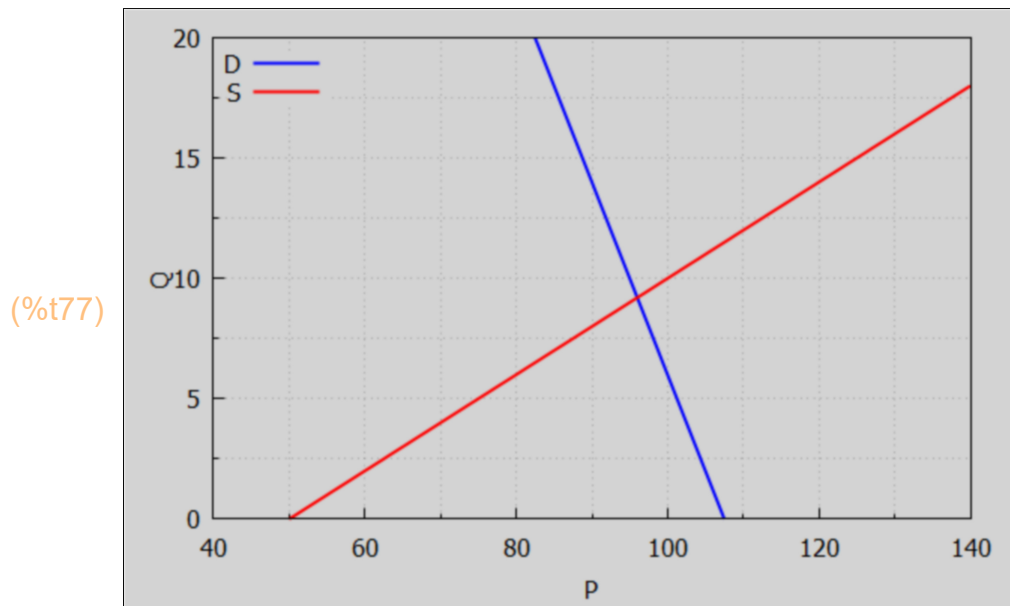
The coordinates of the supply-demand crossing are (P_e, Q_e) in this $Q(P)$ plot.

```
(%i75) Pe : at (P, solve(Qd = Qs, P));
(Pe) 96

(%i76) Qe : at (Qs, P = Pe);
(Qe) 9.2
```

Here we are following the mathematics convention with Q (the dependent variable) on the vertical axis and P (the independent variable) on the horizontal axis. Note that our model parameters imply that $Q_s = 0$ when $P = 50$, meaning that suppliers need a price of at least 50 to supply any of the good being modeled.

```
(%i77) wxdraw2d(xlabel = "P", ylabel = "Q", yrange = [0, 20], xrange = [40, 140],
key_pos = top_left, key = "D", explicit (Qd, P, 40, 140), color = red,
key = "S", explicit (Qs, P, 40, 140))$
```



Suppose the initial ($t = 0$) market price $P_0 = 120$ and let the supply curve predict what the quantity supplied will be in year $t = 1$. Let Q_1 stand for $Q_s[1]$.

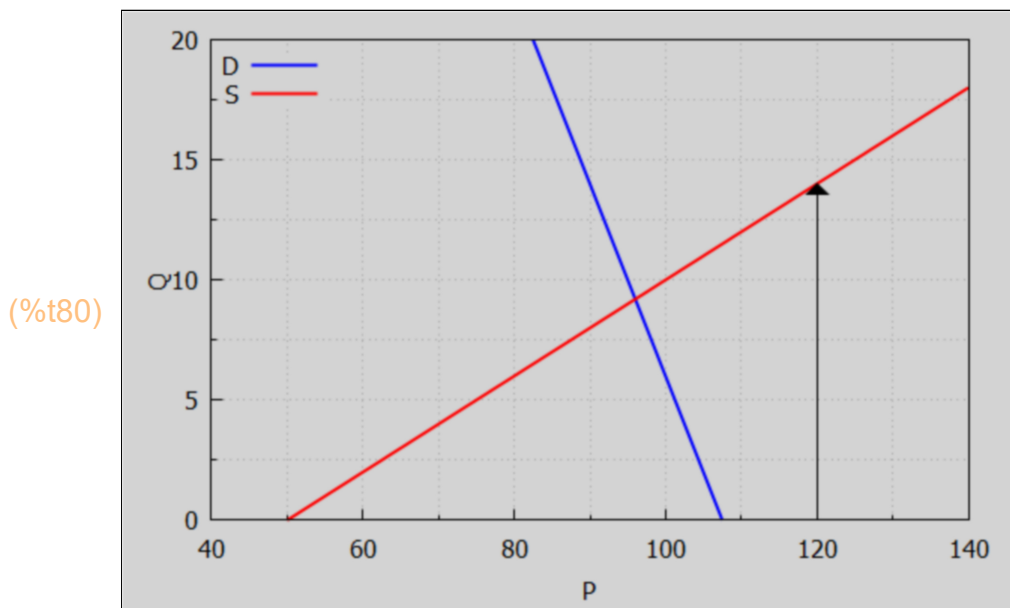
```
(%i78) P0 : 120$
```

```
(%i79) Q1 : at (Qs, P = P0);
```

```
(Q1) 14.0
```

Add a vector to the plot, from point $(120, 0)$ to point $(120, 14)$, recalling the vector syntax used by `draw2d`: `vector ([x,y], [dx, dy])`. For this task, $x = 120$, $y = 0$, (the tail of the vector), and $dx = 0$, $dy = 14$. (dx, dy) describe the orientation and length of the vector, whereas (x, y) describe the tail of the vector. Thus we use `vector ([P0, 0], [0, Q1])`.

```
(%i80) wxdraw2d(xlabel = "P", ylabel = "Q", yrange = [0, 20], xrange = [40, 140],
  key_pos = top_left, key = "D", explicit (Qd, P, 40, 140), color = red,
  key = "S", explicit (Qs, P, 40, 140), key = "", color = black, line_width = 1,
  vector ([P0, 0], [0, Q1]))$
```



Let the demand curve determine the price P_1 offered for quantity $Q_1 = 14$

```
(%i81) soln : solve (Qd = 14, P);
```

```
(soln) [P=90]
```

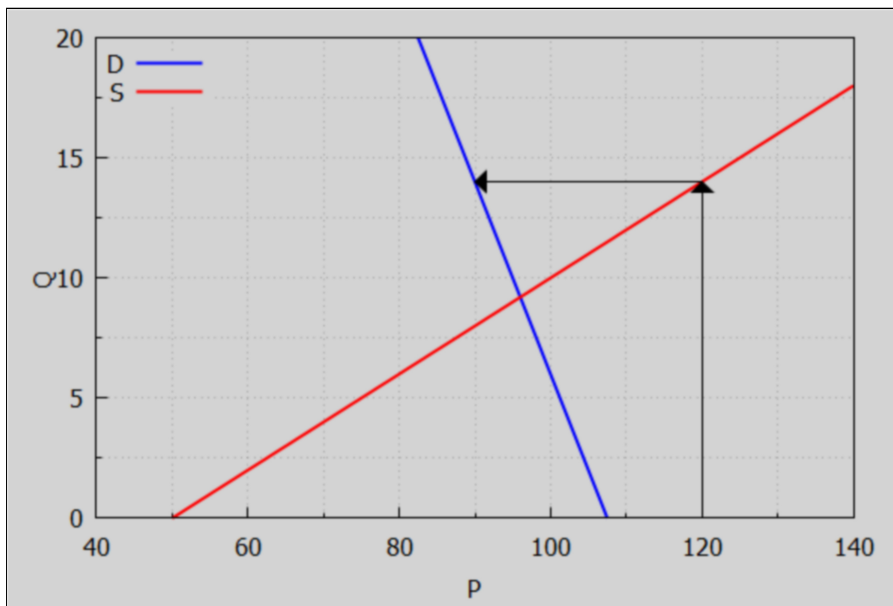
```
(%i82) P1 : at (P, soln);
```

```
(P1) 90
```

Add a vector from (P_0, Q_1) to (P_1, Q_1) , ie., from $(120, 14)$ to $(90, 14)$.

```
(%i83) wxdraw2d(xlabel = "P", ylabel = "Q", yrange = [0, 20], xrange = [40, 140],
  key_pos = top_left, key = "D", explicit (Qd, P, 40, 140), color = red,
  key = "S", explicit (Qs, P, 40, 140), key = "", color = black, line_width = 1,
  vector ([P0, 0], [0, Q1]), vector ([P0, Q1], [P1 - P0, 0]))$
```

(%t83)



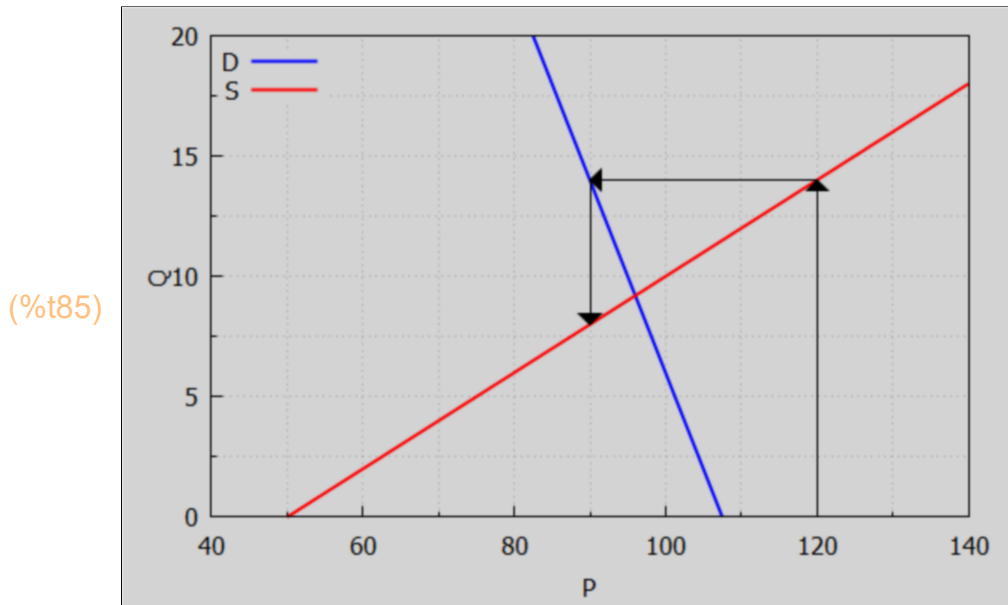
The quantity supplied for year 2, Q_2 , is then determined by the supply curve and the price $P_1 = 90$ offered in year 1.

```
(%i84) Q2 : at (Qs, P = P1);
```

```
(Q2) 8.0
```

Add a vector from $(P_1, Q_1) = (90, 14)$ to $(P_1, Q_2) = (90, 8)$.

```
(%i85) wxdraw2d(xlabel = "P", ylabel = "Q", yrange = [0, 20], xrange = [40, 140],
  key_pos = top_left, key = "D", explicit (Qd, P, 40, 140), color = red,
  key = "S", explicit (Qs, P, 40, 140), key = "", color = black, line_width = 1,
  vector ([P0, 0], [0, Q1]), vector ([P0, Q1], [P1 - P0, 0]),
  vector ([P1, Q1], [0, Q2 - Q1]))$
```



Let the demand curve determine the price P_2 offered for quantity $Q_2 = 8$.

```
(%i86) soln : solve (Qd = Q2, P);
```

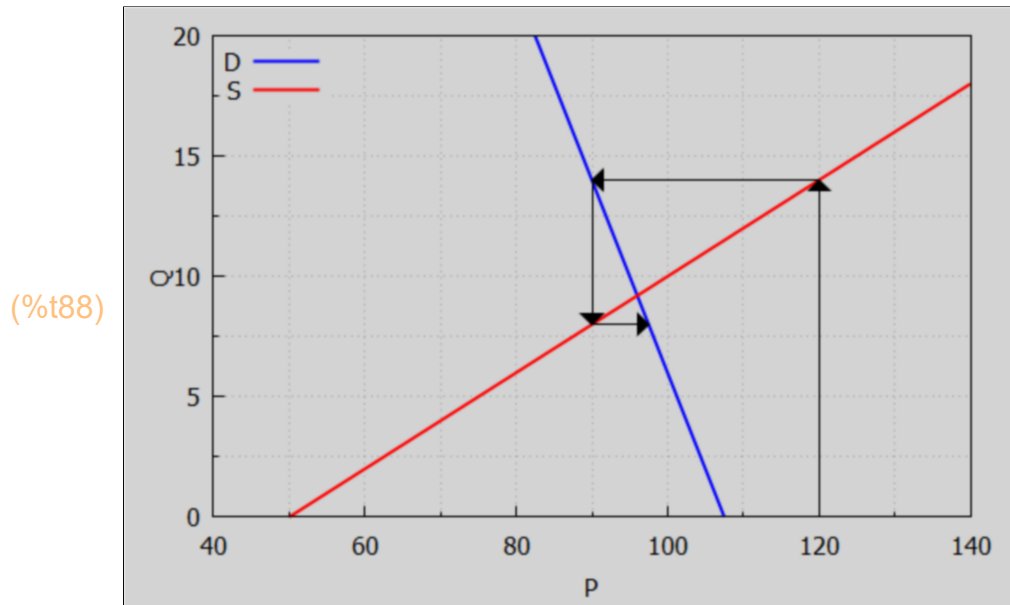
```
(soln) [P = 97.5]
```

```
(%i87) P2 : at (P, soln), numer;
```

```
(P2) 97.5
```

Add a vector from $(P_1, Q_2) = (90, 8)$ to $(P_2, Q_2) = (97.5, 8)$.

```
(%i88) wxdraw2d(xlabel = "P", ylabel = "Q", yrange = [0, 20], xrange = [40, 140],
key_pos = top_left, key = "D", explicit (Qd, P, 40, 140), color = red,
key = "S", explicit (Qs, P, 40, 140), key = "", color = black, line_width = 1,
vector ([P0, 0], [0, Q1]), vector ([P0, Q1], [P1 - P0, 0]),
vector ([P1, Q1], [0, Q2 - Q1]), vector ([P1, Q2], [P2 - P1, 0]))$
```



The quantity supplied for year 3, Q_3 , is then determined by the supply curve and the price $P_2 = 97.5$ offered in year 2.

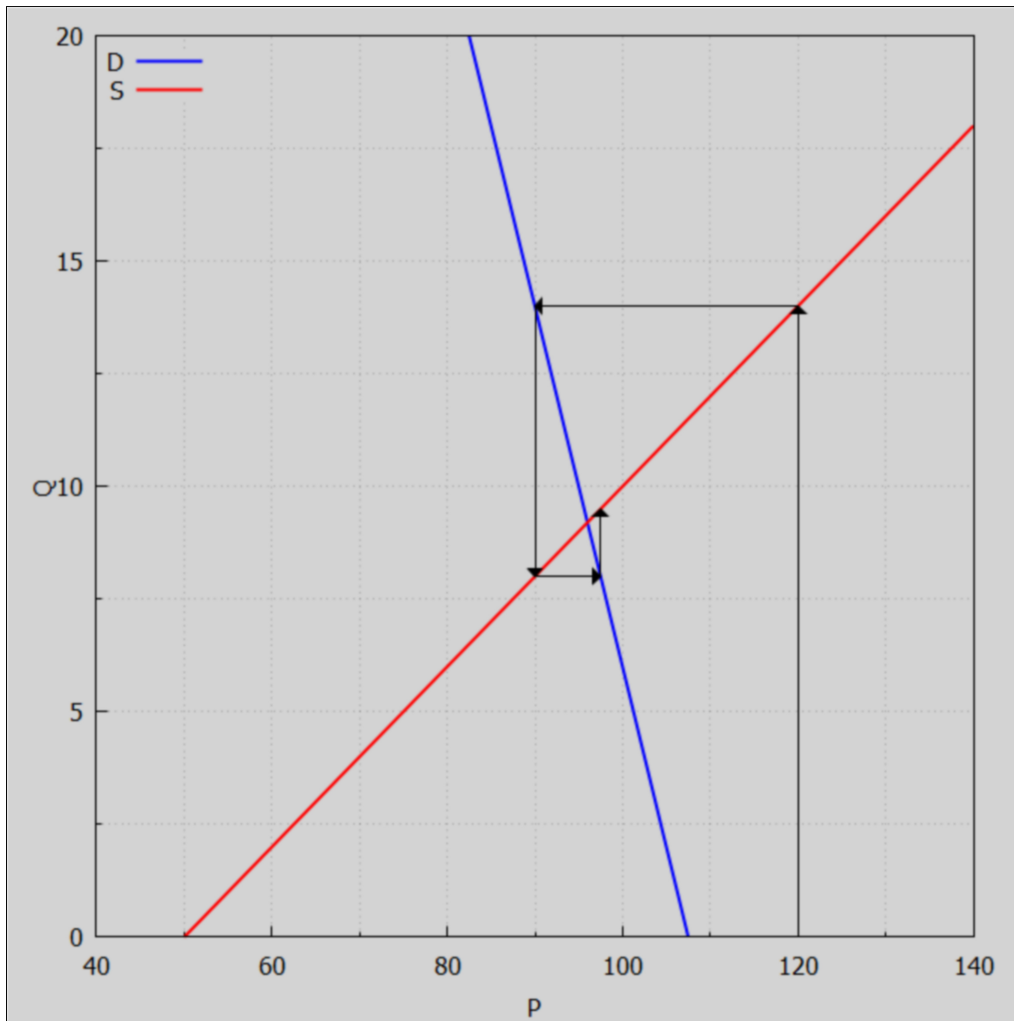
```
(%i89) Q3 : at (Qs, P = P2);
```

```
(Q3) 9.5
```

Add a vector from $(P_2, Q_2) = (97.5, 8)$ to $(P_2, Q_3) = (97.5, 9.5)$.

```
(%i90) wxdraw2d(xlabel = "P", ylabel = "Q", yrange = [0, 20], xrange = [40, 140],
  key_pos = top_left, key = "D", explicit (Qd, P, 40, 140), color = red,
  key = "S", explicit (Qs, P, 40, 140), key = "", color = black, line_width = 1,
  head_length = 1.2, vector ([P0, 0], [0, Q1]), vector ([P0, Q1], [P1 - P0, 0]),
  vector ([P1, Q1], [0, Q2 - Q1]), vector ([P1, Q2], [P2 - P1, 0]),
  vector ([P2, Q2], [0, Q3 - Q2])), wxplot_size = [680, 680];
```

(%t90)



(%o90)

A cobweb pattern is gradually drawn as the vectors converge on the equilibrium value of market price (96).

7 The Harrod Model

The Harrod model attempts to explain the dynamics of growth in the economy. It assumes national saving are some fraction of the national income:

$$S[t] = s Y[t],$$

where s is a constant equal to both the marginal propensity to save (MPS) and the average propensity to save (APS). With both S and Y expressed in dollars, s is dimensionless.

The Harrod model also assumes the "acceleration principle", i.e., investment is proportional to the rate of change of national income over time, in the sense:

$$I[t] = a (Y[t] - Y[t-1]),$$

where a is a constant equal to both the marginal and average capital-output ratios. With both I and Y expressed in dollars, a is dimensionless.

In equilibrium, national investment equals national savings:

$$I[t] = S[t],$$

and hence

$$a (Y[t] - Y[t-1]) = s Y[t], \text{ or}$$

$$(a - s) Y[t] = a Y[t-1].$$

a is normally larger than 1 and s is positive but less than 1, so $(a-s) \neq 0$, and we can normally divide by $(a - s)$ to get

$$Y[t] = (a/(a-s)) Y[t-1].$$

Comparing with our standard form $y[t] = b*y[t-1] + a$, the base $b = a/(a-s)$ and there is no additional constant term. Find $Y[t]$ using DE1, using Y_0 to stand for $Y[0]$.

(%i91) DE1(0, a/(a - s), t, Y0);

(%o91)
$$\frac{Y_0 a^t}{(a-s)^t}$$

(%i92) grind(%)\$

$$(Y_0 * a^t) / (a-s)^t$$

So our solution is $Y[t] = (a/(a-s))^t Y_0$.

The base b of the exponent t is $b = a/(a-s)$, in which $0 < s < 1$ and $a > 1$ normally, so $a/(a-s)$ is positive but greater than 1, so we have non-oscillatory explosive behavior of national income. Income will expand indefinitely with no bounds.

7.1 Example 8: Warranted Rate of Growth

Continuing with the Harrod model, the "rate of growth between successive periods" is a pure number defined as

$$G = (Y[1] - Y[0]) / Y[0]$$

```
(%i94) Y[t] := (a/(a - s))^t*Y0;
      G : (Y[1] - Y0)/Y0, ratsimp;
```

$$(Y[t]) \quad Y_t := \left(\frac{a}{a-s} \right)^t Y_0$$

$$(G) \quad - \frac{s}{s-a}$$

We can absorb the minus sign out front of G with the following trick:

```
(%i95) G : (-num(G))/(-denom(G));
```

$$(G) \quad \frac{s}{a-s}$$

The "warranted rate of growth" G_w is the rate of growth which is needed to have an equilibrium between savings and investment each year. Since we found $Y[t]$ and then the rate of growth G assuming that equilibrium,

$$G_w = s/(a-s).$$

If $s = 0.12$, for example, and $a = 2.12$, the "warranted rate of growth is"

```
(%i96) at (G, [s = 0.12, a = 2.12]);
```

```
(%o96) 0.06
```

ie., 6 per cent.

8 Other Economic Applications

8.1 17.28: Compound Interest Formula from a Difference Equation

Derive the formula for the value $P[t]$ of an initial amount of money P_0 deposited at i interest (decimal interest rate) for t years when compounded annually.

When interest is compounded annually, the principal P increases according to

$$P[t+1] = P[t] + i P[t] = (1 + i) P[t],$$

which is mathematically equivalent to (let $t \rightarrow t-1$ everywhere)

$$P[t] = (1 + i) P[t-1],$$

a form which encourages us to use our difference equation function $DE1(a,b,t,y_0)$, in a case $b = (1+i) \neq 1$ since $i > 0$, and $a = 0$, and we let P_0 stand for $P[0]$.

```
(%i98) kill(P0)$
      de : DE1(0, 1+i, t, P0);
```

$$(de) \quad P_0 (i+1)^t$$

Value of principal after t years is then

$$P[t] = P_0 (1 + i)^t.$$

We used this in Dowling Ch. 8, Sec. 6.

8.2 Prob 17.29: Market Price vs Prev. Yr. Level of Inventory

Assume

$$Q_d[t] = c + z P[t],$$

$$Q_s[t] = g + h P[t].$$

Also assume that the market price is determined not only by the usual market clearing mechanisms (the pursuit of $Q_s = Q_d$) but also by the level of inventory ($Q_s - Q_d$) available the previous year:

$$P[t+1] = P[t] - a (Q_s[t] - Q_d[t]), \text{ for } t = 0, 1, 2, \dots$$

We also make the reasonable assumption that an increase in inventory will tend to reduce the price, and likewise, a decrease in inventory will tend to increase the market price.

So we assume $a > 0$.

Let $I[t] = Q_s[t] - Q_d[t]$, then $P[t+1] = P[t] - a I[t]$. A larger $I[t]$ value produces a smaller $P[t+1]$ provided $a > 0$.

Substituting for Q_s and Q_d , solving for $P[t+1]$ in terms of $P[t]$, and then replacing $t+1 \rightarrow t$ and $t \rightarrow (t-1)$, we get

$$P[t] = B P[t-1] + A,$$

with $B = 1 + a(z - h)$, $A = a(c - g)$.

Let P_0 stand for $P[0]$, and find $P[t]$ using DE1.

```
(%i99) de : DE1 (A, B, t, P0);
```

```
(de) B^t (P0 - A/(1-B)) + A/(1-B)
```

Evidently, $P_e = A/(1 - B)$.

```
(%i101) kill(Pe)$
```

```
de1 : ratsubst (Pe, A/(1-B), de);
```

```
(de1) -B^t Pe + Pe + B^t P0
```

```
(%i102) coeff(de1, B^t);
```

```
(%o102) P0 - Pe
```

```
(%i103) de2 : %*B^t + Pe;
```

```
(de2) Pe + B^t (P0 - Pe)
```

We give this equation a name for later convenience (Prob. 17.30).

(%i104) soln : P[t] = de2;

(soln) $P_t = P_e + B^t (P_0 - P_e)$

This is our expression for P[t]. Stability depends on the value of $B = 1 + a(z - h)$, and hence on the values of a, z, and h.

We assume $a > 0$ (as above), and under normal conditions, $z < 0$, $h > 0$, so $a(z-h) < 0$, which in turn implies $B < 1$.

With $B < 1$, we can distinguish five cases:

- 1.) $0 < B < 1$, P[t] converges to $P_e = A/(1-B)$ and is nonoscillatory; this case requires $|a^*(z - h)| < 1$.
- 2.) $B = 0$, $0^t = 0$, $P[t] = P_e = \text{constant}$, and evidently P_e must equal P_0 ; this case requires $a^*(z - h) = -1$.
- 3.) $-1 < B < 0$, P[t] converges to $P_e = A/(1-B)$ and is oscillatory; this case requires $-2 < a^*(z-h) < -1$.
- 4.) $B = -1$, uniform oscillation between P_0 and $(2*P_e - P_0)$; this case requires $a^*(z-h) = -2$.
- 5.) $B < -1$, oscillatory explosion in value; this case has $a^*(z-h) < -2$.

8.3 Prob. 17.30: Numerical Example of 17.29

As a numerical example, take (in 17.29 above) $c = 120$, $z = -0.5$, $g = -30$, $h = 0.3$, $a = 0.2$, $P_0 = 200$.

Let soln1 be soln (above) in a form in which the explicit dependence on all the model parameters is visible. We first replace P_e in terms of A and B. Next we replace A in terms of a,c,g. Finally, we replace B in terms of a,z,h.

(%i105) soln1 : subst([Pe = A/(1 - B), A = a*(c - g), B = 1 + a*(z - h)], soln);

(soln1) $P_t = (a(z-h) + 1)^t \left(\frac{c-g}{z-h} + P_0 \right) - \frac{c-g}{z-h}$

case1730 is a list of replacement values for all the model constants in our example (Prob. 17.30).

(%i106) case1730 : [c = 120, z = -0.5, g = -30, h = 0.3, a = 0.2, P0 = 200];

(case1730) [c=120,z=-0.5,g=-30,h=0.3,a=0.2,P0=200]

We then find P[t] given these model parameter assumptions:

```
(%i107) soln1730 : at (soln1, case1730);
```

```
(soln1730)  $P_t = 12.5 \cdot 0.84^t + 187.5$ 
```

```
(%i108) grind(%)$
```

```
 $P[t] = 12.5 \cdot 0.84^t + 187.5$ 
```

```
(%i109) at (soln1730, t = 0);
```

```
(%o109)  $P_0 = 200.0$ 
```

With the base $B = 0.84$, $0 < B < 1$, $P[t]$ converges to $P_e = 187.5$ without oscillation.

```
(%i110) rhs(soln1730);
```

```
(%o110)  $12.5 \cdot 0.84^t + 187.5$ 
```

We need to precede % with two single quotes (') to force evaluation to define a Maxima function $f(t)$ with the right hand side of soln1730.

```
(%i111) f(t) := "%;
```

```
(%o111)  $f(t) := 12.5 \cdot 0.84^t + 187.5$ 
```

We now have a defined Maxima function we can use like any other function in Maxima. We create a list of times t (years) using makelist and a list of prices $P[t]$, calling the first list tL and the second list PL .

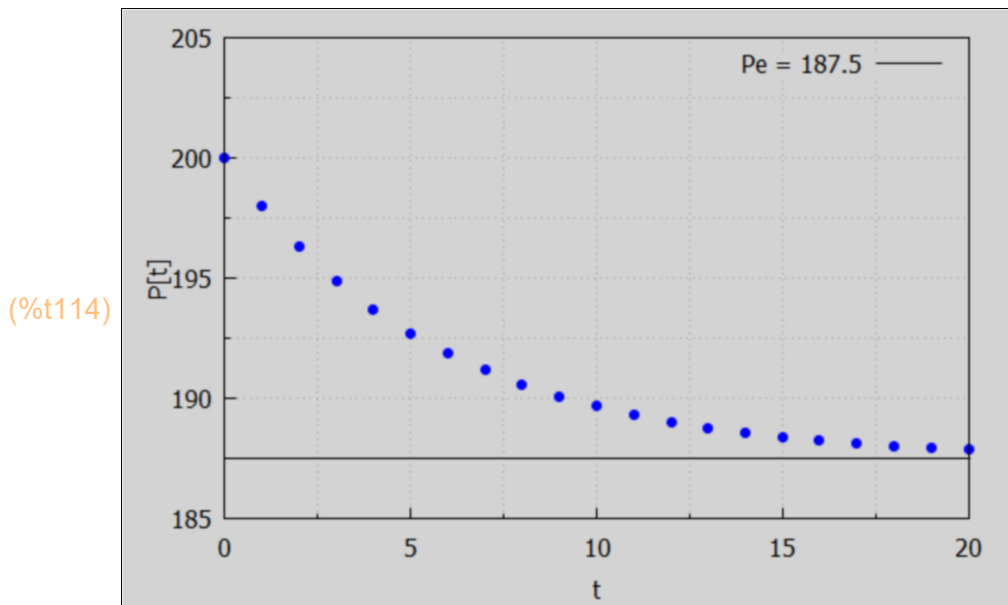
```
(%i113) tL : makelist (t, t, 0, 20);
```

```
PL : map ('f, tL);
```

```
(tL) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
```

```
(PL) [200.0, 198.0, 196.32, 194.91, 193.72, 192.73, 191.89, 191.19, 190.6,
190.1, 189.69, 189.34, 189.04, 188.8, 188.59, 188.41, 188.27, 188.15, 188.04,
187.96, 187.88]
```

```
(%i114) wxdraw2d (xlabel = "t", ylabel = "P[t]", yrange = [185, 205], xrange = [0,20],
  points ( tL, PL ), color = black, line_width = 1, key = "Pe = 187.5",
  explicit (187.5, t, 0, 20))$
```



9 Phase Diagram Analysis for 1st Order Difference Eqns.

Potentially stable equilibrium points in a plot of $y[t]$ vs. $y[t-1]$ are points on the 45 degree line $y[t] = y[t-1]$. If the local "slope" of the curve of $y[t]$ has a magnitude less than 1 (at a crossing point), that point is a point of stable equilibrium.

9.1 Example 10

Make a phase plot of the nonlinear difference equation:

$$y[t] = (y[t-1])^{(1/2)}, \text{ which is equivalent to the difference equation } y[t] = y[t-1]^{(0.5)}.$$

Potential equilibrium points must satisfy $y[t] = y[t-1]$.

If, in the given de, we set both $y[t]$ and $y[t-1]$ equal to y ,

$$y = y^{(1/2)}, \text{ or } y - y^{(1/2)} = 0.$$

This is satisfied for $y = 0$ and $y = 1$.

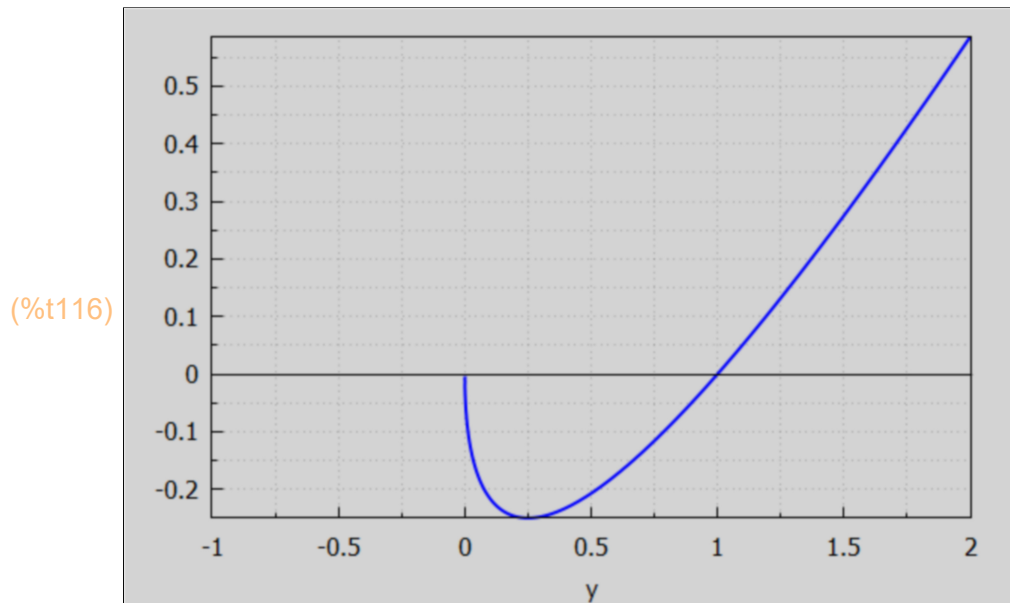
solve cannot deal with nonlinear equations.

```
(%i115) solve (y = y^(1/2), y);
```

```
(%o115) [y=√y]
```

A plot of $f(y)$ vs. y shows where $f(y) = 0$.

```
(%i116) wxdraw2d( xlabel = "y", explicit (y - y^(1/2), y, -1,2),
  color = black, line_width = 1, explicit (0,y,-1,2))$
```



`find_root` can deal numerically with nonlinear equations.

```
(%i117) find_root (y = sqrt(y), y, 0.1, 2);
(%o117) 1.0
```

However, if we try using `findroot (y = sqrt(y), y, -0.1,0.2)` we get an error since if $y < 0$, \sqrt{y} is a complex number.

We can find the limit of the expression $y - \sqrt{y}$ as we approach $y = 0$ from values of $y > 0$.

```
(%i118) limit (y - sqrt(y), y, 0, plus);
(%o118) 0
```

To draw a phase diagram for a first order difference equation, it is convenient to let the symbol `yt` stand for $y[t]$ and the symbol `ytm1` stand for $y[t-1]$.

In terms of these symbols, our difference equation is $yt = ytm1^{0.5} = \sqrt{ytm1}$.
And slope = $\partial(yt)/\partial(ytm1) = \text{diff}(yt, ytm1)$.

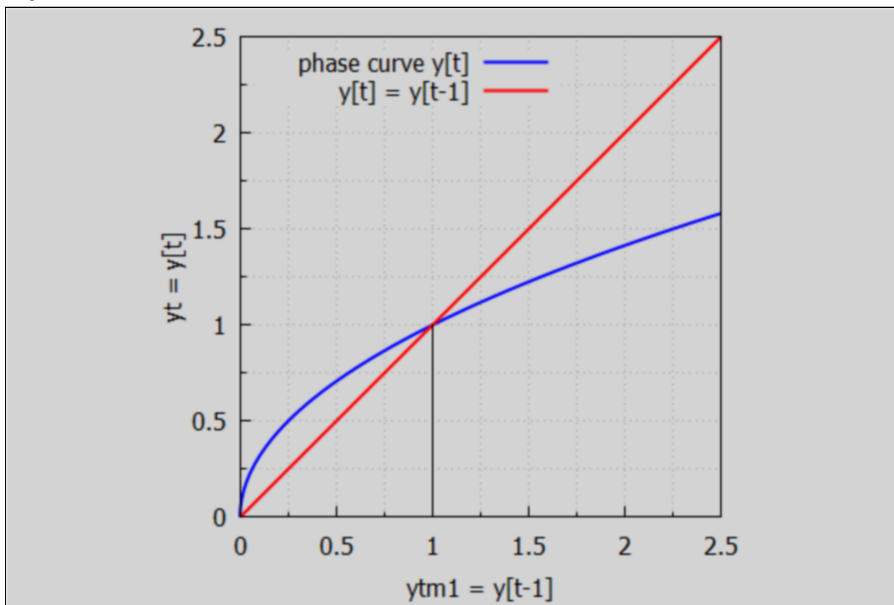
We first draw the y_t vs. y_{tm1} curve, and also the 45 degree line along which $y_t = y_{tm1}$ (the locus of possible equilibrium points).

The first line in the following cell assigns the symbol y_t to the expression $\text{sqrt}(y_{tm1})$. If we use the optional `draw2d` setting: `proportional_axes = 'xy'`, a 45 deg line will look like a 45 deg line.

```
(%i120) yt : sqrt(ytm1);
wxdraw2d (xlabel = "ytm1 = y[t-1]", ylabel = "yt = y[t]", key_pos = top_left,
         proportional_axes = 'xy,
         key = "phase curve y[t]", explicit (yt, ytm1, 0, 2.5), color = red, key = "y[t] = y[t-1]",
         explicit (ytm1, ytm1, 0, 2.5), color = black, key = "", line_width = 1,
         parametric (1,yy,yy,0,1))$
```

(yt) $\sqrt{y_{tm1}}$

(%t120)



Let y_{tp} (y_t -prime) stand for the local slope, as a function of y_{tm1} :

```
(%i121) ytp : diff (yt, ytm1);
```

(ytp) $\frac{1}{2\sqrt{y_{tm1}}}$

The numerical value of the slope of the phase curve at the point $y_{tm1} = 1$ is:

```
(%i122) slope : at (ytp, ytm1 = 1);
```

(slope) $\frac{1}{2}$

The slope at $y_{tm1} = 1$ is $1/2 > 0$, so no oscillation; the slope magnitude < 1 , so $y = 1$ is a point of stable equilibrium.

For a graphical check of stability, let $y_{tm1} = 0.25$ be $y_t[0]$. We can then find succeeding values of y_t from the governing model difference equation: $y_t = \sqrt{y_{tm1}}$. We can show the evolution of succeeding values of $y[t]$, for $t = 1, 2, 3, \dots$, on our phase diagram.

Let $y_j = y[j]$ in this sidebar.

(1): starting at the point ($y_{tm1} = y_0 = y[0]$, $y_t = 0$), draw a vertical line to the intersection with the y_t vs y_{tm1} curve (which determines y_1): $(y_0, 0) \rightarrow (y_0, y_1)$,

(2): draw the horizontal line $(y_0, y_1) \rightarrow (y_1, y_1)$ (ends on the 45 deg line),

(3): draw a vertical line to intersect the y_t vs. y_{tm1} curve (which determines y_2):
 $(y_1, y_1) \rightarrow (y_1, y_2)$,

(4): draw the horizontal line $(y_1, y_2) \rightarrow (y_2, y_2)$ (ends on the 45 deg line),

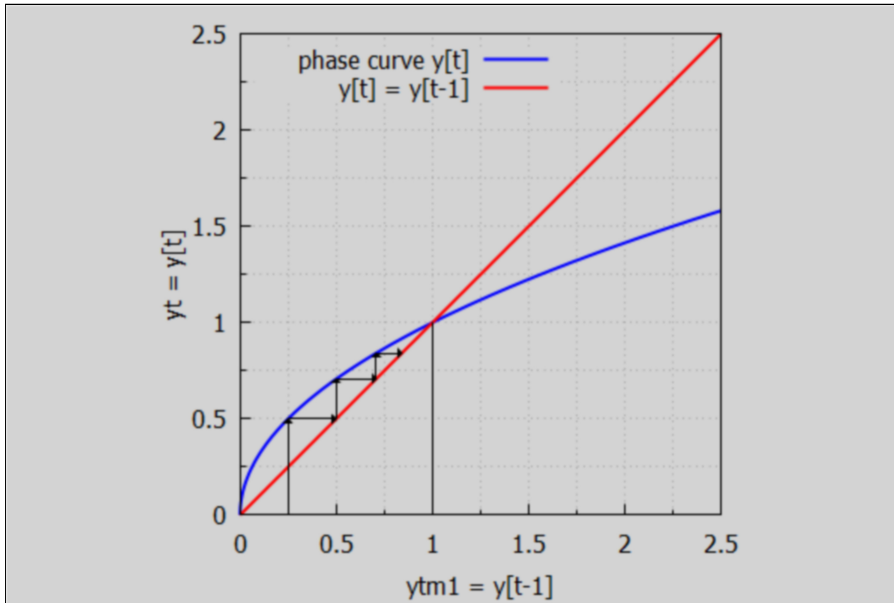
(5): draw the vertical line to intersect the y_t vs. y_{tm1} curve (which determines y_3):
 $(y_2, y_2) \rightarrow (y_2, y_3)$,

and so on.

In the following cell, we use the given model difference equation to define the history of $y[t]$; the values of y_0, y_1, \dots, y_4 , assuming $y_0 = 0.25$.

```
(%i124) kill (y0,y1,y2,y3,y4)$
[y0 : 0.25, y1 : sqrt (y0), y2 : sqrt(y1), y3 : sqrt(y2), y4 : sqrt(y3)];
(%o124) [0.25,0.5,0.70711,0.8409,0.917 ]
```

```
(%i125) wxdraw2d (xlabel = "ytm1 = y[t-1]", ylabel = "yt = y[t]", key_pos = top_left,
    proportional_axes = 'xy,
    key = "phase curve y[t]", explicit (yt, ytm1, 0, 2.5), color = red, key = "y[t] = y[t-1]",
    explicit (ytm1, ytm1, 0, 2.5), color = black, key = "", line_width = 1,
    parametric (1,yy,yy,0,1), head_length = 0.03, vector ([y0, 0], [0, y1]),
    vector ([y0,y1], [y1-y0, 0]), vector ([y1,y1], [0, y2-y1]), vector ([y1,y2],[y2-y1,0]),
    vector ([y2,y2], [0, y3-y2]), vector ([y2,y3], [y3-y2,0]))$
```



An initial value $y_0 = 0.25$ heads closer and closer to $y = 1$ as the value of t increases.

If the initial value starts out at, say $y_0 = 2$, we can again follow the progression as $t = 1, 2, 3, \dots$

Here we redefine the successive $y[t]$ values: y_0, y_1, y_2, \dots

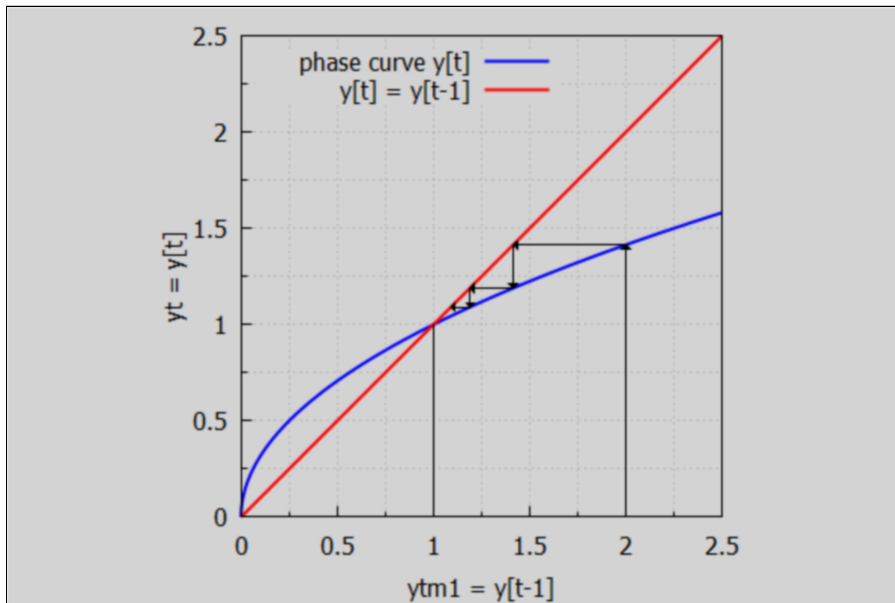
We can then reuse the above graphics instructions without change to show stability from the high side.

```
(%i127) kill (y0,y1,y2,y3,y4)$
[y0 : 2, y1 : sqrt (y0), y2 : sqrt(y1), y3 : sqrt(y2), y4 : sqrt(y3)], numer;
```

```
(%o127) [2, 1.4142, 1.1892, 1.0905, 1.0443]
```

```
(%i128) wxdraw2d (xlabel = "ytm1 = y[t-1]", ylabel = "yt = y[t]", key_pos = top_left,
    proportional_axes = 'xy,
    key = "phase curve y[t]", explicit (yt, ytm1, 0, 2.5), color = red, key = "y[t] = y[t-1]",
    explicit (ytm1, ytm1, 0, 2.5), color = black, key = "", line_width = 1,
    parametric (1,yy,yy,0,1), head_length = 0.03, vector ([y0, 0], [0, y1]),
    vector ([y0,y1], [y1-y0, 0]), vector ([y1,y1], [0, y2-y1]), vector ([y1,y2],[y2-y1,0]),
    vector ([y2,y2], [0, y3-y2]), vector ([y2,y3], [y3-y2,0]))$
```

(%t128)



We again see convergence to the equilibrium value $y = 1$, this time from the high side.

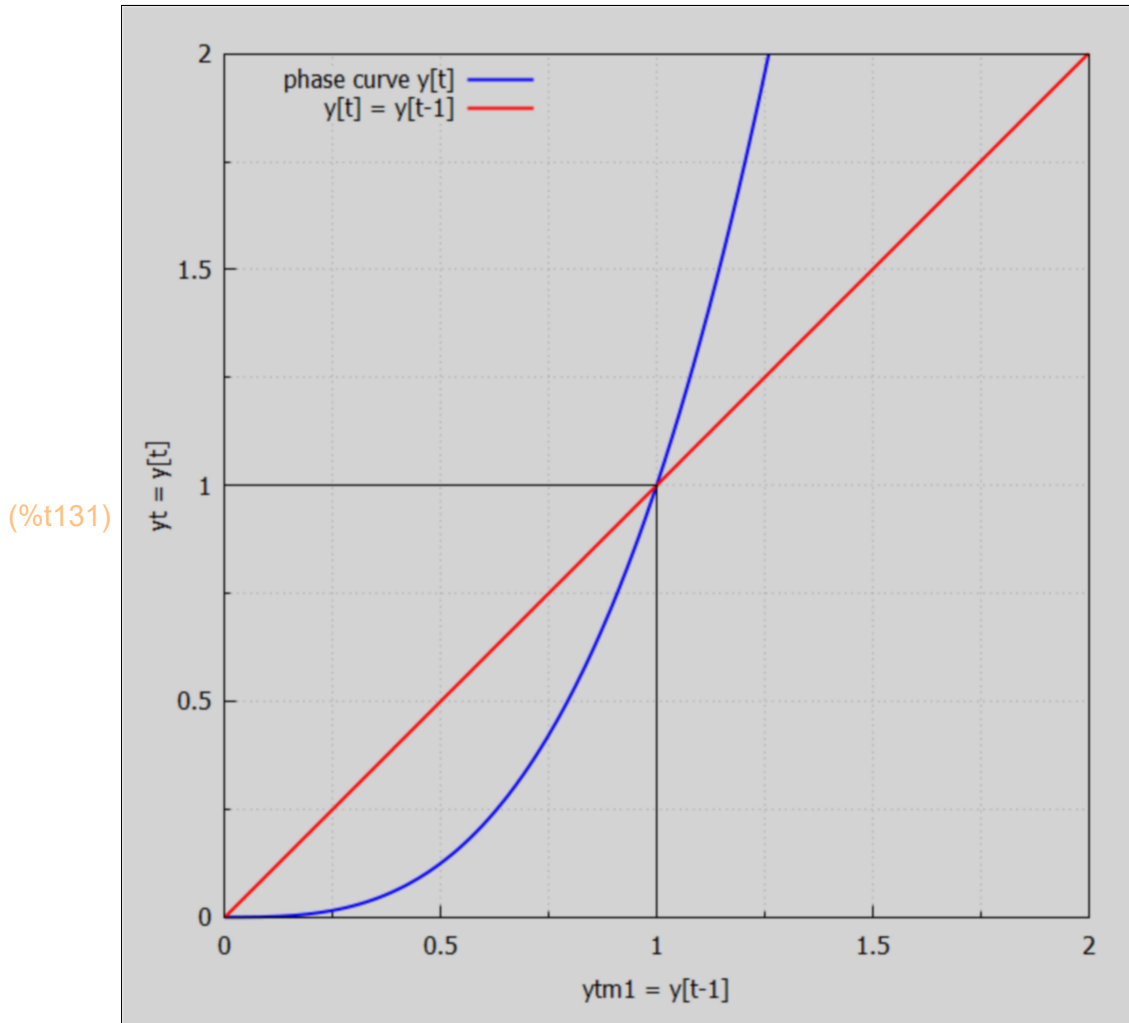
9.2 Prob. 17.31

Construct a phase diagram for the nonlinear difference equation
 $y[t] = y[t-1]^3$
 and check for stability.

Let's start with the phase diagram plot.

```
(%i131) kill(yt, ytm1)$
yt : ytm1^3;
wxdraw2d (xlabel = "ytm1 = y[t-1]", ylabel = "yt = y[t]", key_pos = top_left,
proportional_axes = 'xy, yrange = [0, 2],
key = "phase curve y[t]", explicit (yt, ytm1, 0, 2), color = red, key = "y[t] = y[t-1]",
explicit (ytm1, ytm1, 0, 2), color = black, key = "", line_width = 1,
explicit (1,ytm1,0,1), parametric (1,yy,yy,0,1)), wxplot_size = [680, 680];
```

(yt) $ytm1^3$



We see points of possible stable equilibrium: $y = 0$ and $y = 1$.

The slope at $(0,0)$ is 0 and the slope at $(1,1)$ is 3 (right?). So we infer that $(1,1)$ is not a stable equilibrium point, and $(0,0)$ is a stable equilibrium point.

Visual confirmation with the plot: start with $y_0 = 0.75$, and watch what happens.

```
(%i134) kill(y0,y1,y2,y3,y4,f)$
```

```
    f(x) := x^3;
```

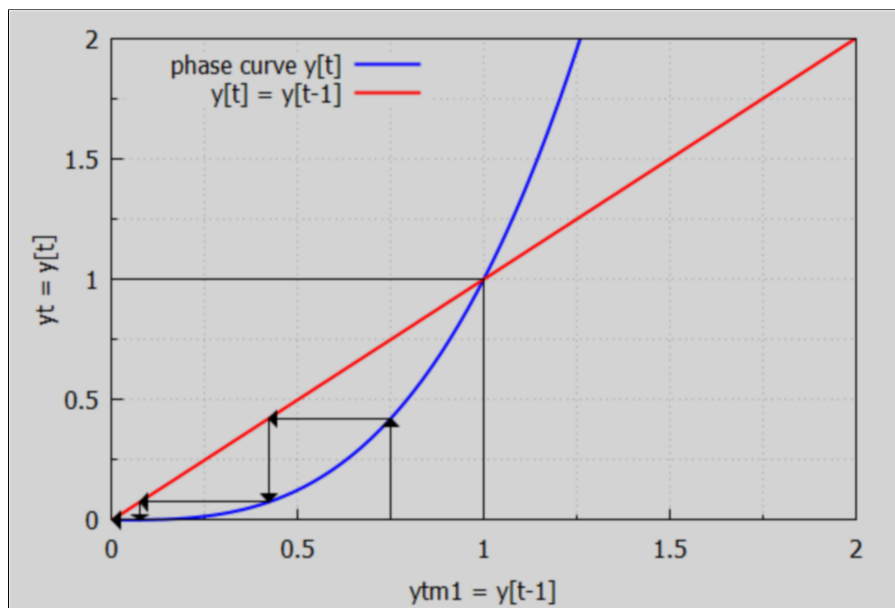
```
    [y0 : 0.75, y1 : f(y0), y2 : f(y1), y3 : f(y2), y4 : f(y3)];
```

```
(%o133) f(x) := x^3
```

```
(%o134) [0.75, 0.42188, 0.075085, 4.2331 10^-4, 7.5851 10^-11]
```

```
(%i135) wxdraw2d (xlabel = "ytm1 = y[t-1]", ylabel = "yt = y[t]",
    yrange = [0,2], key_pos = top_left,
    key = "phase curve y[t]", explicit (yt, ytm1, 0, 2), color = red, key = "y[t] = y[t-1]",
    explicit (ytm1, ytm1, 0, 2), color = black, key = "", line_width = 1,
    explicit (1, ytm1, 0, 1), parametric (1, yy, yy, 0, 1), head_length = 0.03,
    vector ([y0, 0], [0, y1]), vector ([y0, y1], [y1 - y0, 0]), vector ([y1, y1], [0, y2 - y1]),
    vector ([y1, y2], [y2 - y1, 0]), vector ([y2, y2], [0, y3 - y2]), vector ([y2, y3], [y3 - y2, 0]))$
```

```
(%t135)
```



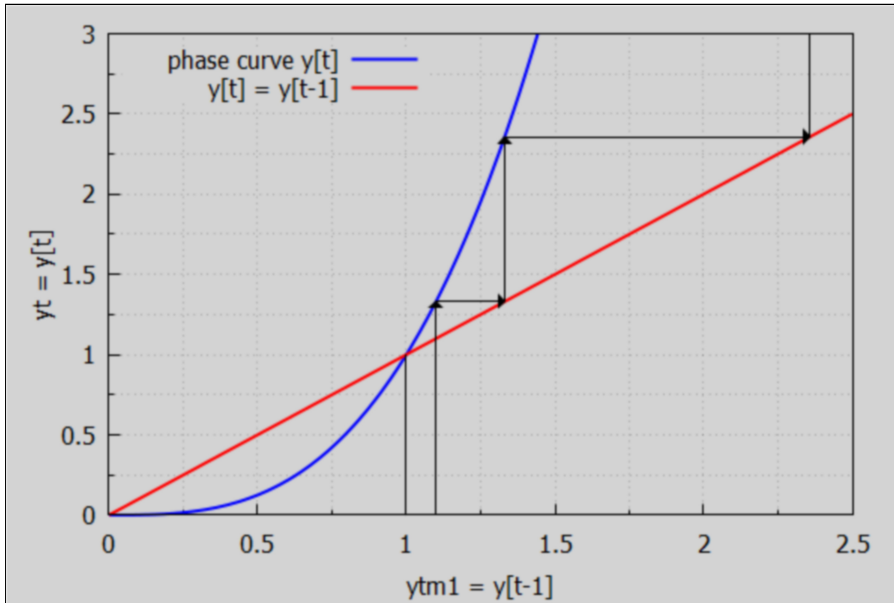
The phase curve takes $y[t]$ away from $y[0] = 0.75$ in decreasing y direction towards $y = 0$. Next we start at $y_0 = 1.1$, on the high side of $y = 1$.

```
(%i136) [y0 : 1.1, y1 : f(y0), y2 : f(y1), y3 : f(y2), y4 : f(y3)];
```

```
(%o136) [1.1, 1.331, 2.3579, 13.11, 2253.2]
```

```
(%i137) wxdraw2d (xlabel = "ytm1 = y[t-1]", ylabel = "yt = y[t]",
    xrange = [0,3], xrange = [0, 2.5], key_pos = top_left,
    key = "phase curve y[t]", explicit (yt, ytm1, 0, 2.5),color = red, key = "y[t] = y[t-1]",
    explicit (ytm1, ytm1, 0, 2.5), color = black, key = "", line_width = 1,
    parametric (1,yy,yy,0,1), head_length = 0.03, vector ([y0, 0], [0, y1]),
    vector ([y0,y1], [y1-y0, 0]), vector ([y1,y1], [0, y2-y1]), vector ([y1,y2],[y2-y1,0]),
    vector ([y2,y2], [0, y3-y2]), vector ([y2,y3], [y3-y2,0]))$
```

(%t137)



Again the phase diagram sends y away from $y[0] = 1$, so the only stable equilibrium point is $y = 0$.

9.3 Prob. 17.32

Construct a phase diagram for the nonlinear difference equation $y[t] = y[t-1]^{-1/4}$ and check for stability.

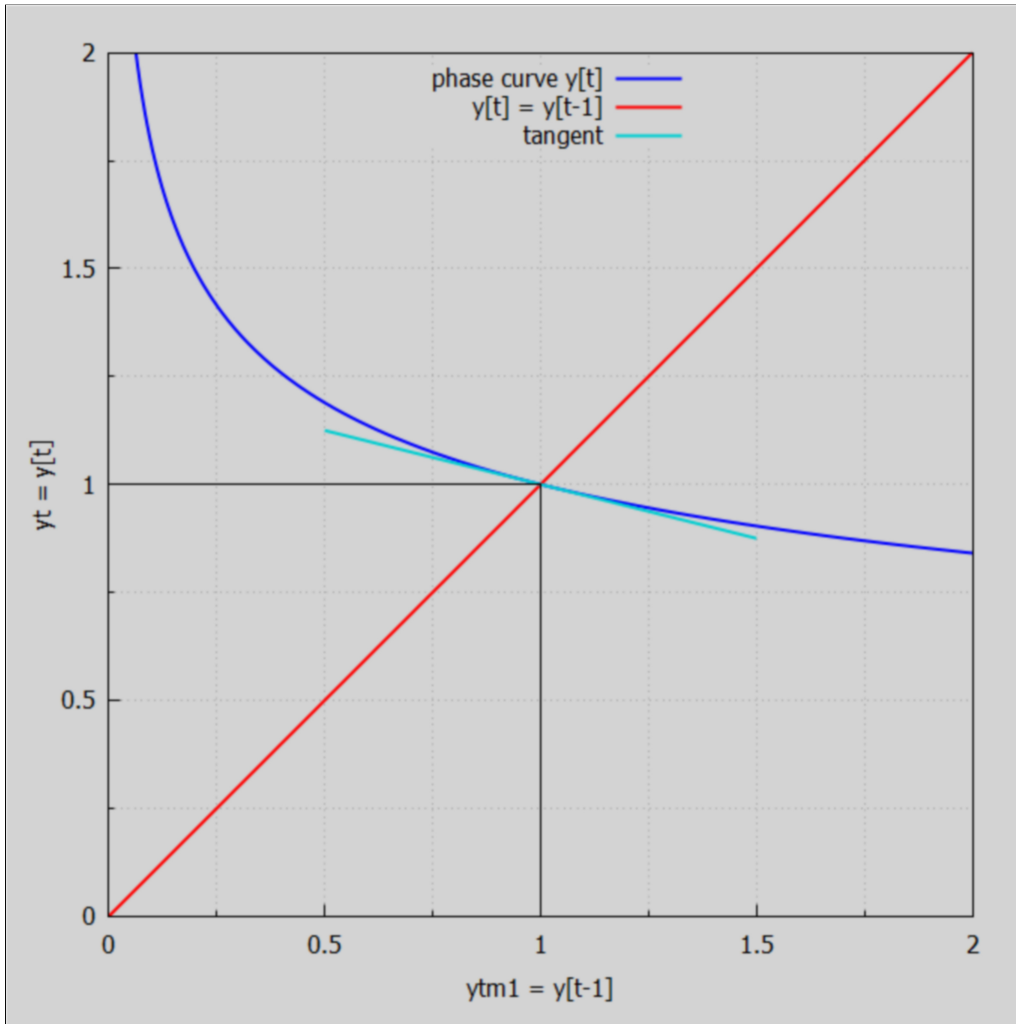
The only possible equilibrium point is $(1,1)$, and the yt vs. $ytm1$ curve has a local slope of $-1/4 = -0.25$ at that point, whose magnitude is less than 1, indicating a true equilibrium point. The negative sign of the local slope at point $(1,1)$ means that the history of $y[t]$ will oscillate around $y = 1$ as convergence proceeds.

We draw a phase diagram plot for this case, and add a line tangent to the curve at point $(1,1)$ whose slope is $-1/4$.

```
(%i140) kill(yt, ytm1)$
yt : ytm1^(-1/4);
wxdraw2d (xlabel = "ytm1 = y[t-1]", ylabel = "yt = y[t]", yrange = [0, 2],
proportional_axes = 'xy, key = "phase curve y[t]", key_pos = top_center,
explicit (yt, ytm1, 0, 2),color = red, key = "y[t] = y[t-1]", explicit (ytm1, ytm1, 0, 2),
color = dark_turquoise, key = "tangent", explicit (-x/4 + 5/4, x, 0.5, 1.5),
color = black, key = "", line_width = 1, explicit (1, ytm1, 0,1),
parametric (1,yy,yy,0,1)), wxplot_size = [680, 680];
```

(yt) $\frac{1}{ytm1^{1/4}}$

(%t140)



(%o140)

We can visualize the convergence in the case $y[0] = y_0 = 0.75$.

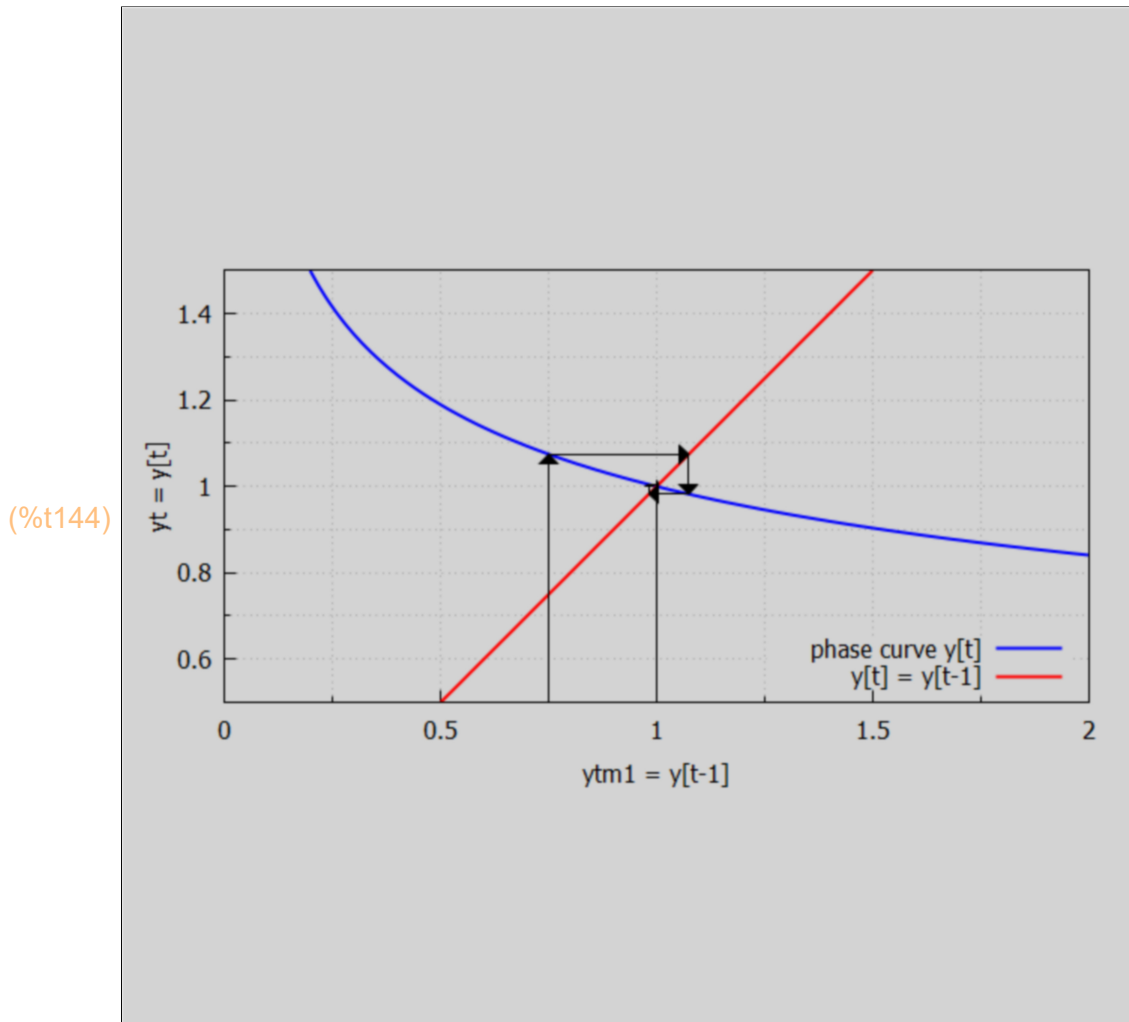
```
(%i143) kill(y0,y1,y2,y3,y4,f)$
f(x) := x^(-0.25);
[y0 : 0.75, y1 : f(y0), y2 : f(y1), y3 : f(y2), y4 : f(y3)];
```

```
(%o142) f(x) := x^-0.25
```

```
(%o143) [0.75, 1.0746, 0.98218, 1.0045, 0.99888]
```

We see that successive values of y (y_1, y_2, y_3, \dots) oscillate between values larger than 1 and values smaller than 1, but converging on $y = 1$.

```
(%i144) wxdraw2d (xlabel = "ytm1 = y[t-1]", ylabel = "yt = y[t]",
  xrange = [0.5, 1.5], key_pos = bottom_right, proportional_axes = 'xy,
  key = "phase curve y[t]", explicit (yt, ytm1, 0, 2), color = red, key = "y[t] = y[t-1]",
  explicit (ytm1, ytm1, 0, 2), color = black, key = "", line_width = 1,
  parametric (1, yy, yy, 0.5, 1), head_length = 0.03, vector ([y0, 0], [0, y1]),
  vector ([y0, y1], [y1 - y0, 0]), vector ([y1, y1], [0, y2 - y1]), vector ([y1, y2], [y2 - y1, 0]),
  vector ([y2, y2], [0, y3 - y2]), vector ([y2, y3], [y3 - y2, 0]), wxplot_size = [680, 680];
```



If we start with a y_0 larger than 1, say 1.25,

```
(%i147) kill(y0,y1,y2,y3,y4,f)$
  f(x) := x^(-0.25);
  [y0 : 1.25, y1 : f(y0), y2 : f(y1), y3 : f(y2), y4 : f(y3)];
```

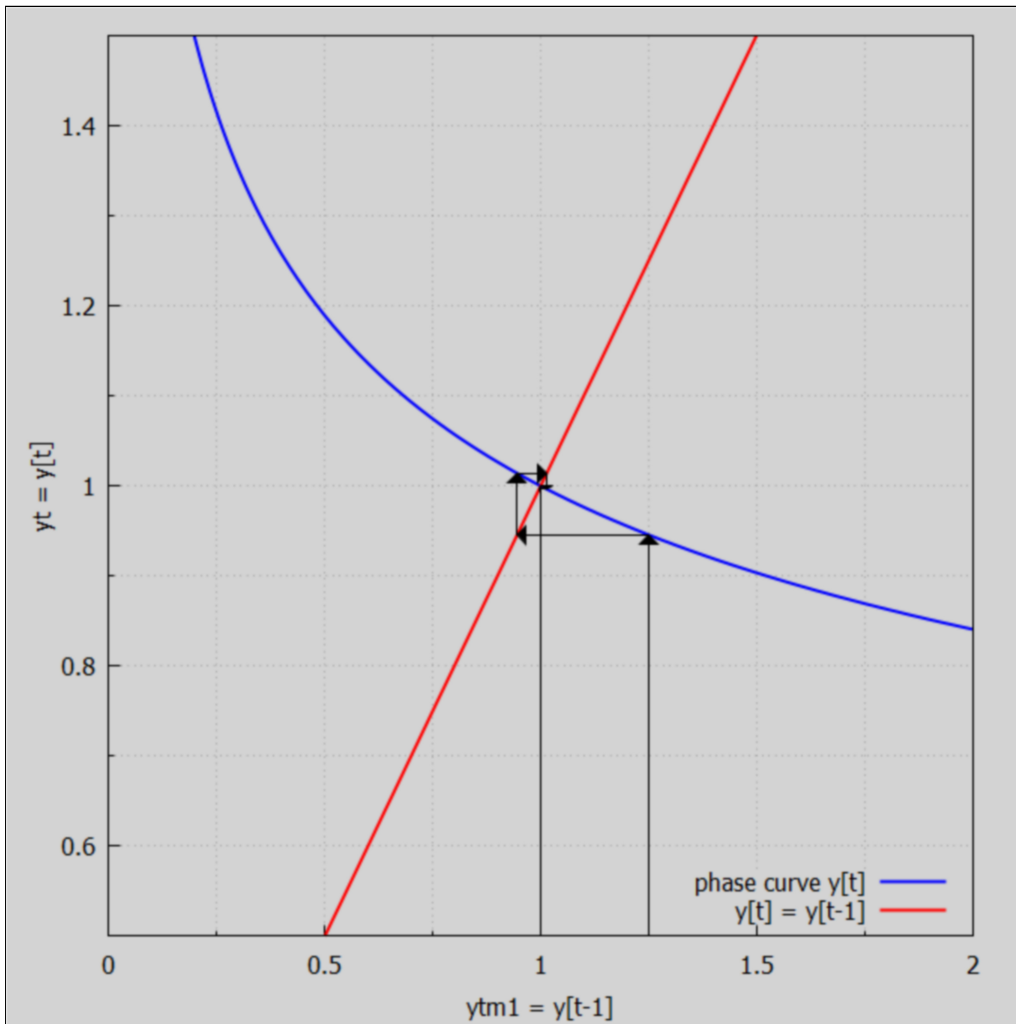
```
(%o146) f(x) := x^-0.25
```

```
(%o147) [1.25, 0.94574, 1.014, 0.99652, 1.0009]
```


we again get oscillating convergence to $y = 1$.

```
(%i148) wxdraw2d (xlabel = "ytm1 = y[t-1]", ylabel = "yt = y[t]",
    xrange = [0.5,1.5], key_pos = bottom_right,
    key = "phase curve y[t]", explicit (yt, ytm1, 0, 2),color = red, key = "y[t] = y[t-1]",
    explicit (ytm1, ytm1, 0, 2), color = black, key = "", line_width = 1,
    parametric (1,yy,yy,0.5,1), head_length = 0.03, vector ([y0, 0], [0, y1]),
    vector ([y0,y1], [y1-y0, 0]), vector ([y1,y1], [0, y2-y1]), vector ([y1,y2],[y2-y1,0]),
    vector ([y2,y2], [0, y3-y2]), vector ([y2,y3], [y3-y2,0])), wxplot_size = [680,680];
```

(%t148)



(%o148)

9.4 Prob. 17.33

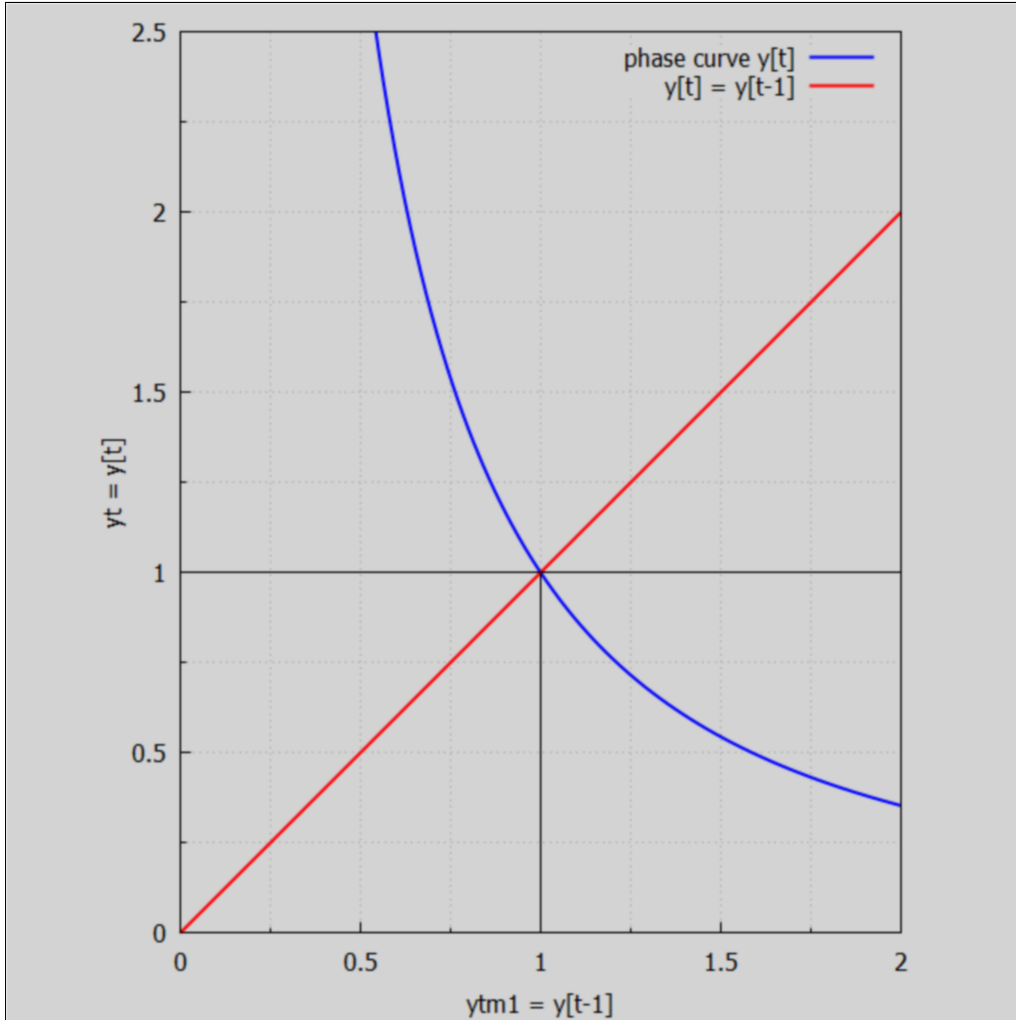
Construct a phase diagram for the nonlinear difference equation $y[t] = y[t-1]^{-3/2}$ and check for stability.

The only possible equilibrium point is $(1,1)$ and the local slope at that point is -1.5 , with a magnitude greater than 1. Hence we have explosive (oscillating) instability.

```
(%i151) kill(yt, ytm1)$
yt : ytm1^(-3/2);
wxdraw2d (xlabel = "ytm1 = y[t-1]", ylabel = "yt = y[t]", key_pos = top_right,
  yrange = [0, 2.5], proportional_axes = 'xy,
  key = "phase curve y[t]", explicit (yt, ytm1, 0, 2), color = red, key = "y[t] = y[t-1]",
  explicit (ytm1, ytm1, 0, 2), color = black, key = "", line_width = 1,
  explicit (1, ytm1,0,2), parametric (1,yy,yy,0,1)), wxplot_size = [680, 680];
```

(yt) $\frac{1}{ytm1^{3/2}}$

(%t151)



(%o151)

For a visual look at the explosive instability, we assume $y[0] = y_0 = 0.75$.

```
(%i154) kill(y0,y1,y2,y3,y4,f)$
f(x) := x^(-1.5);
[y0 : 0.75, y1 : f(y0), y2 : f(y1), y3 : f(y2), y4 : f(y3)];
```

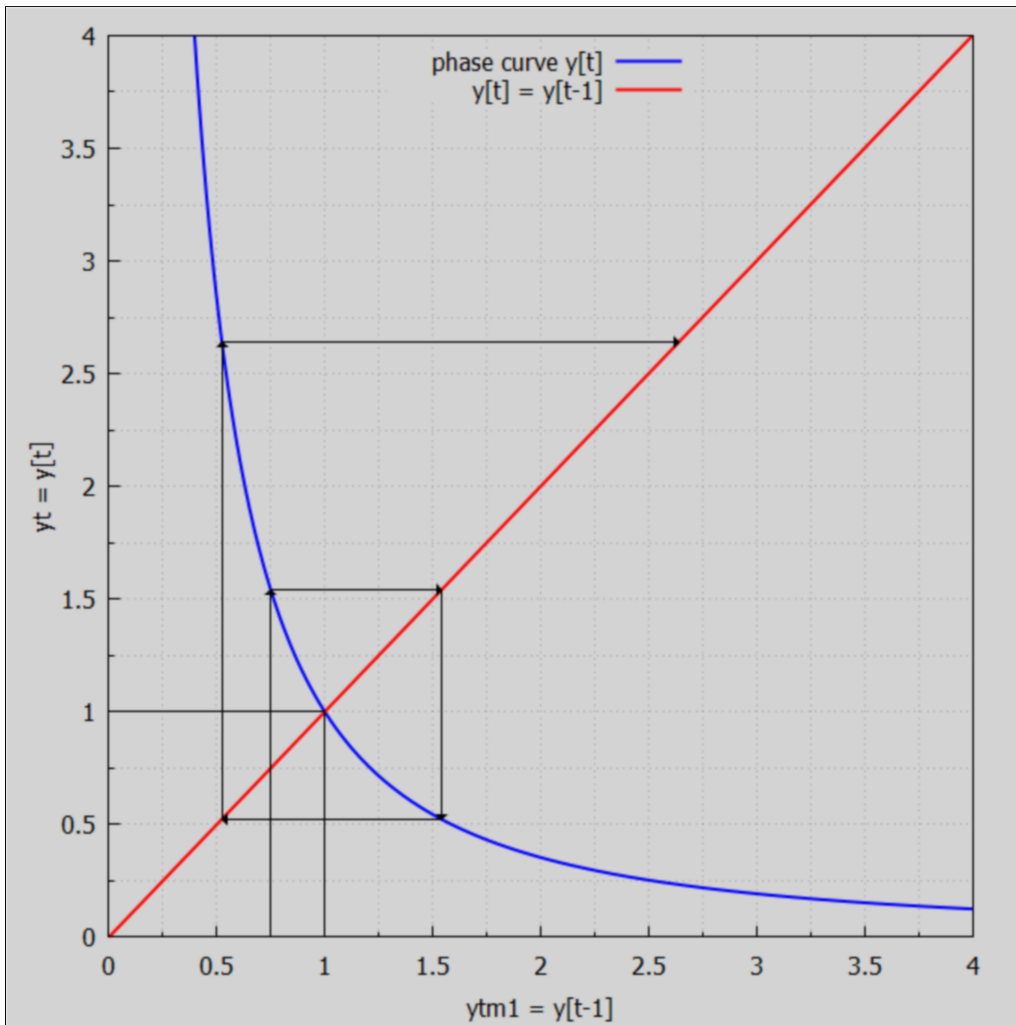
```
(%o153) f(x) := x^-1.5
```

```
(%o154) [0.75, 1.5396, 0.52347, 2.6404, 0.23308]
```

We see y_1, y_2, y_3, \dots values getting further and further away from $y = 1$, with oscillation between values larger than and smaller than $y = 1$.

```
(%i155) wxdraw2d (xlabel = "ytm1 = y[t-1]", ylabel = "yt = y[t]",
    xrange = [0,4], key_pos = top_center,
    key = "phase curve y[t]", explicit (yt, ytm1, 0, 4), color = red, key = "y[t] = y[t-1]",
    explicit (ytm1, ytm1, 0, 4), color = black, key = "", line_width = 1,
    explicit (1, ytm1, 0, 1), parametric (1, yy, yy, 0, 1), head_length = 0.03,
    vector ([y0, 0], [0, y1]), vector ([y0, y1], [y1 - y0, 0]), vector ([y1, y1], [0, y2 - y1]),
    vector ([y1, y2], [y2 - y1, 0]), vector ([y2, y2], [0, y3 - y2]),
    vector ([y2, y3], [y3 - y2, 0])), wxplot_size = [680, 680];
```

(%t155)



(%o155)