```
#Title:` FTC Squeeze
 #Author: Gord Clement, January 2011
 #Description : Animation to illustrate a key step in the proof of the fundamental theorem of calculus.
 #                    Shows h →0 +  and the absolute maximum and minimum on [x, x + h] converging to x
 #Usage: No user input required

  output ≔ [ ] :
 #define function with lots of local maxima and minima
 g ≔ (x − 1) · (x − 2) ·x· (x + 1) · (x + 2) + 10 :
 #plot of function
 origplot ≔ plot(g, x =−2.5 ..2.5 , y = 5 ..20, tickmarks = [0, 0], thickness = 2, labels = ["", ""]) :
 #plot of vertical line for x
 xplot ≔ plot( [ -2.005, y, y = 5 ..15 ], x =−2.5 ..2.5 , y = 5 ..20, linestyle = dash, color = black, labels
       = ["", ""]) :
 #label for x
 xlabel ≔ plots[textplot]( [ -2.005, 15.5, typeset(x) ]) :
 #initalize H
 H ≔ 4 :
 for i from 0 to 48 do
       #step through H values
      H ≔ H − 0.08;
      #find absolute max and min values
      my ≔ minimize(g, x =−2.005 .. -2.005 + H);
      My ≔ maximize(g, x =−2.005 .. -2.005 + H);
       #plot for max and min values
       minplot ≔ plot(my, x =−2.005 .. -2.005 + H, color = blue, thickness = 2, labels = ["", ""]) :
       maxplot ≔ plot(My, x =−2.005 .. -2.005 + H, color = green, thickness = 2, labels = ["", ""]) :

      #find x value corresponding to min

       mx ≔ -0.565; #default to the one x value that numerically slips through the cracks
       mxList ≔ evalf (solve(g = my, x) ) assuming real; #find all possible x values for the min

        if (nops( [mxList]) = 1 ) then
             mx ≔ mxList : #if just one, use that value
       else
             #find a value in the interval [x,x+h]
             for j from 1 to nops( [mxList]) do
                     if ( -2.006 ≤ mxList[ j] ≤ -2.004 + H) then
                             mx ≔ mxList[ j];
                     end if;
             end do;
      end if;

      #find x value corresponding to the max
      MxList ≔ evalf (solve(g = My, x) );
      Mx ≔ -1.66; #default to the one value the numerically slips through the cracks
      if (nops( [MxList]) = 1 ) then
           Mx ≔ MxList :
      else
        for j from 1 to nops( [MxList]) do
           if ( -2.006 ≤ MxList[ j] ≤ -2.004 + H) then
```

```
              Mx := MxList[j];
          end if;
       end do;
     end if;

     #plots to label m and M
     mtext := plots[textplot]([mx, 16, typeset(m)]) :
     Mtext := plots[textplot]([Mx, 16.5, typeset(M) ]) :
     mplot := plot([mx, y, y = my..16], x =-2.5 ..2.5, y = 5 ..20, linestyle = dash, color = black, labels
      = ["", ""]);
     Mplot := plot([Mx, y, y = My..16.5], x =-2.5 ..2.5, y = 5 ..20, linestyle = dash, color = black, labels
      = ["", ""]);

      #plots to label x+h
      xhplot := plot([-2.005 + H, y, y = 5 ..17], x =-2.5 ..2.5, y = 5 ..20, linestyle = dash, color = black,
     labels = ["", ""]);
     xhlabel := plots[textplot]([-2.005 + H, 17.5, typeset(x + h)]) :

       #put all the pieces together
      fullplot := plots[display]( [origplot, xplot, minplot, maxplot, xhplot, xlabel, xhlabel, mtext, Mtext,
     mplot, Mplot], title = typeset( "h=", evalf( H/10, 3 ) ), caption = typeset("The Maximum value ",
     f(M), " is plotted in green. The minimum value ", f(m), " is plotted in blue."), labels = ["", ""] ) :
 #add full plot to the list
 output := [op(output), fullplot] :

 end do:
  #animate list
 plots[display]([op(output)], insequence = true, view = [-2.5 ..2.5, 5 ..20])
```