```
> #Title: IVT
  #Author: Gord Clement, June 2011
  #Description: For a given function, interval and intermidiate value u, this procedure determines
      if the conditions

      #                  of the intermediate value theorem hold. If so, all possible values c on the
      interval such that f(c) = u

      #                  are calculated. An animation shows the function being plotted across the
      interval and the c values are
    #                 labelled when reached
  #Usage:
  #Call : IVT(function, interval, value)
  #function : function to be used for the theorem, note : must be continuous on given interval

      #interval: interval to be used in the theorem, entered in standard Maple notation, ie [a,b]
      would be entered x=a..b

      #value: intermediate value to be used in theorem, note: must be between the function
      evaluated at the end points of the interval

IVT :=proc(expr, range, value)
   #local variable declarations
   local A, B, fAta, fAtb, fullList, cList, var, maxy, miny, output, tempc, mainplot, atext, btext, ctext,
      cplot, i, clist, step, fullplot, fplot, tempx, j;
    #extract variable and endpoints from range
    var := op( 1, range);
   A := evalf(op(1, op(2, range ) ));
   B := evalf(op(2, op(2, range) ));

   #find min and max y values for plotting purposes
   miny := minimize(expr, var=A ..B);
   maxy := maximize(expr, var=A ..B);

  #calculate animation step size
  step := evalf( (B − A) / 40 );
   #calculate f(a) and f(b)
  fAta := subs(var=A, expr);
 fAtb := subs(var=B, expr);

  #check if function is continuous
  if (not(iscont(expr, var=A ..B,'closed'))) then
      output
       := ("Error: The Intermediate Value Theorem does not apply since the function is not
      continuos on the given range");
  #make sure interval given is correct
  elif (A ≥ B) then
     output := ("Error:  a must be less than b");

 #check if value is between f(a) and f(b)
  elif ((fAta < value < fAtb) or (fAtb < value < fAta)) then
 #calculate all potential c values
 fullList := evalf(solve(expr = value, var, dropmultiplicity = true ));
if (nops([fullList]) = 1) then
     #if just one value

     tempc := fullList;

    #plot a, b, and intermediate value labels

     mainplot := plot([[[A, t, t = miny - (maxy - miny)/10 ..maxy], [B, t, t = miny
 - (maxy - miny)/10 ..maxy], [t, value, t = A - (B - A)/10 ..B + (B - A)/10 ], [t, fAta, t = A
 ..B], [t, fAtb, t = A ..B]], thickness = [1, 1, 2, 1, 1], linestyle = [dash, dash, solid, dash, dash],
color = [black, black, red, black, black]) :
   atext := plots[textplot]([A, maxy + (maxy - miny)/20, typeset("a = ", evalf(A, 2))], font
 = [TIMES, ROMAN, 14]) :
   btext := plots[textplot]([B, maxy + (maxy - miny)/20, typeset("b = ", evalf(B, 2))], font
 = [TIMES, ROMAN, 14]) :

   #plot c labels
  ctext := plots[textplot]([tempc, value - (maxy - miny)/4.5, typeset("c1 ≈ ", evalf(tempc,
3))], font = [TIMES, ROMAN, 14]) :
  cplot :=  plot([tempc, t, t = value- (maxy - miny)/5 ..value], thickness = 1, linestyle = dash,
color = black) :
  #initialize x value for animation
   tempx := A :
  fullplot := plots[display]([mainplot, atext, btext]) :
  output := [fullplot];
  for i from 1 to 40 do
       tempx := tempx + step :
       #plot function up to animation value
       fplot := plot(expr, var = A .. tempx, thickness = 2, color = black) :

      #if x has crossed c value, also plot the c labels
      if (is(tempx < tempc) )then
          fullplot := plots[display]([mainplot, atext, btext, fplot]) :
      else
          fullplot := plots[display]([mainplot, atext, btext, fplot, ctext, cplot]) :
      end if;
```

```
       # add plot to list
       output := [op(output), fullplot];
   end do;
     #animate list
    plots[display]( [op(output) ], view = [A - (B - A)/10 ..B + (B - A)/10, default], insequence
  = true, labels = ["", ""] );
else
      #multiple potential c values
     clist := [ ];
     #find all values between a and b
     for i from 1 to nops( [fullList]) do
           tempc := fullList[i];
            if (is(tempc < B) and is(tempc > A)) then
                  clist := [op(clist), tempc];
           end if;
    end do;

    #plot a, b and value labels
    mainplot := plot( [ [A, t, t = miny - (maxy - miny)/10 ..maxy], [B, t, t = miny
  - (maxy - miny)/10 ..maxy], [t, value, t = A - (B - A)/10 ..B + (B - A)/10 ], [t, fAta, t = A
  ..B], [t, fAtb, t = A ..B] ], thickness = [1, 1, 2, 1, 1], linestyle = [dash, dash, solid, dash, dash],
  color = [black, black, red, black, black] ) :
    atext := plots[textplot]( [A, maxy + (maxy - miny)/20, typeset("a = ", evalf(A, 2)) ], font
  = [TIMES, ROMAN, 14] ) :
    btext := plots[textplot]( [B, maxy + (maxy - miny)/20, typeset("b = ", evalf(B, 2)) ], font
  = [TIMES, ROMAN, 14] ) :
    fullplot := plots[display]( [mainplot, atext, btext]) :
    output := [fullplot];
   clist := sort(clist) : #sort c values from low to high
   cplot := [ ] :
    ctext := [ ] :
    tempx := A;
    for i from 1 to nops(clist) do
          #create c label plots
          tempc := clist[i];
          cplot := [op(cplot), plot( [tempc, t, t = value- (maxy - miny)/5 ..value], thickness = 1,
  linestyle = dash, color = black ) ] :
```

```
            ctext := [op(ctext), plots[textplot]([tempc, value - (maxy - miny)/4.5, typeset("c", i,
        " ≈ ", evalf(tempc, 3))], font = [TIMES, ROMAN, 14])] :
         end do;
         for i from 1 to 40 do
           #create animation
           tempx := tempx + step :
           fplot := plot(expr, var = A .. tempx, thickness = 2, color = black) :
          fullplot := [mainplot, atext, btext, fplot];
          for j from 1 to nops(clist) do
              #plot c values as the x value crosses them
              if (is(tempx ≥ clist[j]) ) then
                  fullplot := [op(fullplot), ctext[j], cplot[j]] :
             end if;

           end do;
         fullplot := plots[display]([op(fullplot)]) :
         output := [op(output), fullplot];
        end do;
        #animate
        plots[display]([op(output)], view = [A - (B - A)/10 ..B + (B - A)/10, default], insequence
         = true, labels = ["", ""]);
      end if;
    else
         #error message if intermediate value is not between f(a) and f(b)
        output := cat("Error: The Intermediate Value Theorem does not apply since ",
         convert(value, string), " is not between f(a) = ", convert( fAta, string), "and f(b) = ",
         convert( fAtb, string));

    end if;

    end proc:
> IVT(x^2, x = 0 ..2, 2)
> IVT(x, x =-3 ..2, 1)


> IVT(x^3 - 3/2 x^2 - 3/2 x + 15, x =-2 .. 5/2, 14)

>
```