TaylorAnim3.mws

An Animation of Taylor and Maclaurin Series Converging to Their Generating Functions

Author: Carl Devore <devore@math.udel.edu>

15 April 2001

The author, Carl Devore, can be reached by email at <devore@math.udel.edu>, by paper mail at 655 E3 Lehigh Rd, Newark DE 19711 USA,

or by telephone at (302) 831-3176 or (302) 738-2606.

If you use this program, please send me a brief email, either to discuss it, or just to say "hi". I am frequently willing to write Maple applications for free if the subject interests me. I welcome all suggestions for improvement.

Introduction

The primary purpose of this worksheet is to teach the reader about Taylor series. Maple's superb animation commands are ideal for this. I have seen several other Maple programs that animate Taylor series. I believe that this program is far faster and more versatile than any other program that does something similar. Please let me know if you disagree. Note that this progran can handle functions with discontinuities.

The secondary purpose is to teach the Maple programmer about Maple's not-well-known facilities for fast numerical computation provided by the evalhf command.

To get the best view of the animations, you should maximize both your Maple window and the worksheet window within the Maple window. In addition, you can direct the plot output to a separate window by clicking Optons then Plot Display then Window. My opinion is that the best view for these animation is acheived by setting the playback speed to a very slow value (by using the animation controls on the lower toolbar). (The controls do not appear until you click on an animation.) Since animations take up a lot of memory, it is probably best if you delete some of them after viewing. If you don't, the scrolling speed of the worksheet may become slow. You can delete them by clicking on Edit, them Remove Ouput, then From Worksheet. You can also direct all the plot output to another window and then close each window after it is viewed.

```
> restart;
```

This command sends me an email if this worksheet is run on a Unix computer.

```
> if kernelopts(platform) = "unix" then system(`echo 'TaylorAnim' | mailx -sTaylorAnim devore@math.udel.edu`) fi:
```

```
> TaylorAnim:=
proc(f :: procedure # function to plot
,cntr # expansion point for Taylor series
,xrange :: range # plotting interval
,ord :: nonnegint # highest degree polynomial to plot
)
option `Copyright (c) Carl James DeVore, Jr.`;

local
a, b # left and right members of xrange
,deg # degree of current polynomial
,TaySeries # the Taylor series, stored as an array
,x0 # evalf(cntr)
,npts # number of x-values to evaluate at
,X # array of x-values
,Pts # array of all polynomials evaluated at all x-values
,Plot # basic plot of the function
,Curve # numbers extracted from Plot
,txtx, txty # x,y positions at which to print text
,ylow, yhigh # range from the original plot
,largs # list of optional arguments
,yrange # user-specified y-range
,Discont # value of user-supplied 'discont' option
,Color # user-specified color for the polynomials
,k # generic index
;

Digits:= trunc(evalhf(Digits));
UseHardwareFloats:= true;

x0:= evalf(cntr);

# Deconstruct the x-range.
a:= evalf(op(1, xrange));
b:= evalf(op(2, xrange));
if x0 < a or x0 > b then error cntr, `is not in the range`, xrange fi;

# Parse optional arguments
largs:= [args[5..-1]];
yrange:= NULL;
Discont:= NULL;
Color:= COLOR(RGB, 0., 0., 1.); # Default to blue
for k in largs do
```

```
if k::`..` then yrange:= k
elif k::`=`(identical('discont'), identical(true)) then Discont:= k
elif member(lhs(k), {'color', 'colour'}) then Color:= `plot/color`(rhs(k))
fi
od;
largs:= `plot/options2d`(op(remove(A -> A::`..` or A::`=`(identical('discont'), identical(true)), largs)));

# Use regular, adaptive plot command.
plot(f, xrange, yrange, Discont);
# Put the color spec, if any, inside the CURVES
Plot:= CURVES(op(op(1,%)), op(select(x->op(0,x)=COLOUR or op(0,x)=COLOR, %)));

# Extract all numbers from Plot
Curve:= map(op, [op(select(type, Plot, listlist))]);

# Save adaptively chosen x-values as a hardware float array. This is array type used by the evalhf command.
# Note that the "datatype= float[8]" and "order= C_order" are necessary.
X:= rtable(remove(`=`, map(P->P[1], Curve), FAIL), datatype= float[8], order= C_order);
# op([2,2]) applied to a 1-D rtable gives the upper limit of indices. Since the lower limit is 1 by
# default, this gives the number of elements. Note that once you are inside the evalhf environment, it is
# impossible to determine the size of an array; it must be passed in.
npts:= op([2,2], X);

# Approximate y-range from the plot if the user didn't specify it.
if yrange=NULL then
Curve:= remove(`=`, map(P->P[2], Curve), FAIL);
ylow:= min(op(Curve));
yhigh:= max(op(Curve));

# Add a margin
yrange:= yhigh-ylow;
yhigh:= yhigh + .1*yrange;
ylow:= ylow - .1*yrange
else
ylow:= evalf(op(1, yrange));
yhigh:= evalf(op(2, yrange))
fi;

# Find a good place for putting text on the graph.
txtx:= .9*a+.1*b;
txty:= .9*yhigh+.1*ylow;

# Construct array of Taylor coeffs and degrees.
# op([1..-3], ...) returns the array of coeffiecients and exponents from a taylor series.
# The extra -1's at the end are needed as stop signals for the subsequent computation.
TaySeries:=
rtable(evalf([op(1..-3, taylor(evalf(f(x)), x= x0, ord+1)), -1, -1]), datatype= float[8], order= C_order);

#Evaluate Taylor polys at all x-values and for all degrees and store results in the array Pts
Pts:= rtable(1..npts, 0..ord, datatype= float[8], order= C_order);
evalhf
(proc(X # x-values to evaluate at
,a # center point of series
,TaySeries # the series in sparse polynomial / array form, terminated with two -1's
,ord # maximum degree
,Pts # array of evaluations to be returned
,npts # number of x-values
)
local
d # degree of current term
,`x-a`, `(x-a)^d` # exactly what they look like
,S # sum of series so far
,j, k # indices
;
# Attempting to use Horner form for floating-point polynomial evaluation is worthless.
# The roundoff error is extreme.
for k to npts do
`x-a`:= X[k]-a;
`(x-a)^d`:= 1;
j:= 2; # Position of first exponent
S:= 0;
for d from 0 to ord do
# TaySeries is stored in the form [coeff, expon, coeff, expon, ...] where only nonzero
# coefficients are stored.
if d = TaySeries[j] then # We found the exponent.
S:= S+TaySeries[j-1]*`(x-a)^d`;
j:= j+2
fi;
`(x-a)^d`:= `(x-a)^d`*`x-a`;
Pts[k,d]:= S
od
od
end proc
(X, x0, TaySeries, ord, Pts, npts)
);
```

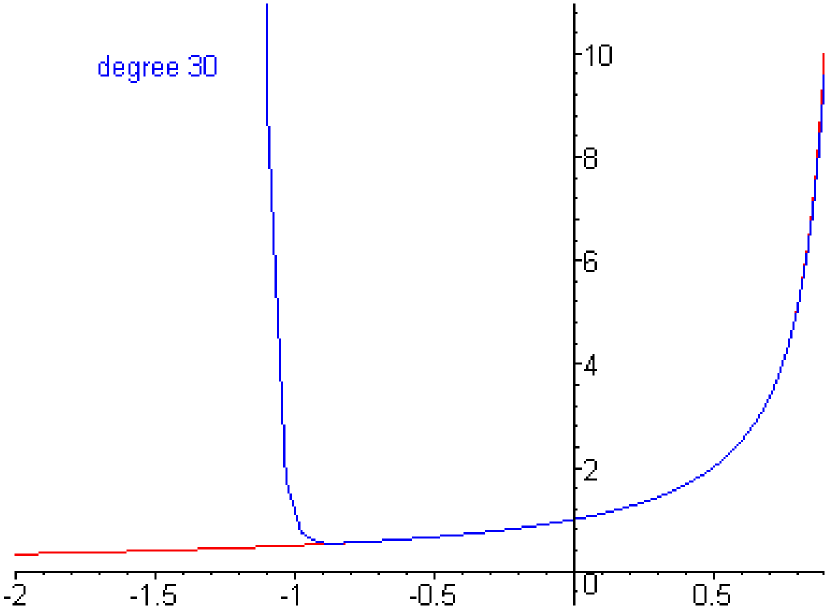
```
#Return the animated plot structure
PLOT
(ANIMATE
(seq([CURVES([seq([X[k], Pts[k,deg]], k= 1..npts)], Color)
,Plot
,TEXT([txtx,txty], cat(`degree `, deg), ALIGNRIGHT)
,Color
,VIEW(a..b, ylow..yhigh)
,largs
]
,deg= 0..ord
)
)
)
end proc:
```

Probably the most basic example is 1/(1-x), the fundamental geometric series. We know that $\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$ for $|x| < 1$. That's the same as saying that the sequence of polynomials $1, 1+x, 1+x+x^2, 1+x+x^2+x^3$, ... converges to 1/(1-x) for $|x| < 1$.

```
> f:= x->1/(1-x):
```

Note that I express f in Maple's operator notation rather than expression notation. For this first plot, I won't include the discontinuity at x = 1 in the plot interval. I'll show how to do that later. The second parameter to TaylorAnim is the point around which the Taylor series is expanded. The third parameter is the plot interval. The fourth parameter is highest degree polynomial to plot. Optional additional parameters can be any of the modifiers that you can put on Maple's display command, like axes= frame, scaling= constrained, thickness= 3, or font= [HELVETICA, 14].

```
> TaylorAnim(f, 0, -2..0.9, 30);
```



Remember to click on the graph and use the animation controls. Click on the << button to slow the playback speed.

What are some of the significant features of the converging sequence of functions that you see in the animation?

Think about that and then expand the subsection for the answer.

The features that I was hoping that you'd notice are

1. The convergence expands outwards from x=0.

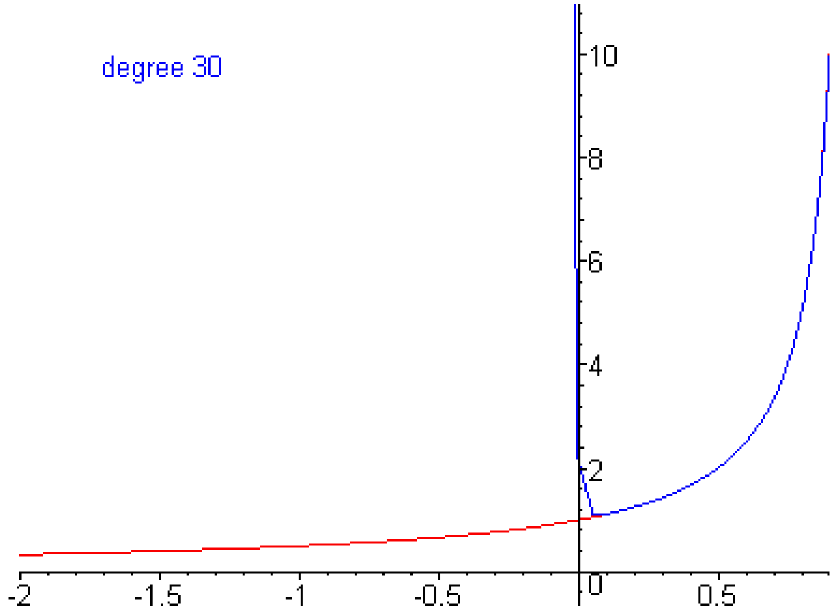
2. There is no convergence for x < -1 (even though the function is perfectly smooth there).

Feature 1 is a key property of Taylor series. There are other sequences of polynomials that converge to a given function (over a specified interval) even faster than the Taylor polynomials. But the Taylor polynomials have the "best" convergence, in a well-defined sense, among those sequences whose convergence expands outwards from a single point. Also, the Taylor polynomials are much easier to compute than those other sequences of polynomials. In all the subsequent animations, notice how the convergence expands outwards from single point, the second parameter of the TaylorAnim command.

Feature 2 has to do with the radius of convergence of the power series. We expanded the Taylor series using a center of x=0. The function is undefined at x=1, a distance of 1 from the central point. Therefore the radius of convergence is at most 1. There can be no convergence for x at a distance more than 1 from 0. Now, you might think that this is trivial because in this case it is obvious that $\sum_{n=0}^{\infty} x^n$ cannot converge for any x < -1 because the terms would be getting larger in absolute value. In the next animation, we do one where it is not so trivial.

Based on the radius of convergence, what do you expect to happen if we expand around x=1/2?

```
> TaylorAnim(f, 1/2, -2..0.9, 30);
```



Try a few more expansion points.

```
> #TaylorAnim(f, -1/2, -3..0.9, 30);
```

```
> #TaylorAnim(f, -2, -6..0.9, 30);
```

The above two animations show that the Taylor series about those central points have wider intervals of convergence than those in the first two animations. Why not use them instead? Well, they are useful for many purposes. But the series for x=0 is easier to work with, especially if you have to work with pencil and paper. We can use Maple's taylor command to see the Taylor polynomials for that last animation. I'll list them up to degree 8.

```
> for n to 9 do taylor(f(x), x= -2, n) end;
```

$$\frac{1}{3}+\mathrm{O}(x+2)$$

$$\frac{1}{3}+\frac{1}{9}(x+2)+\mathrm{O}((x+2)^{2})$$

$$\frac{1}{3}+\frac{1}{9}(x+2)+\frac{1}{27}(x+2)^{2}+\mathrm{O}((x+2)^{3})$$

$$\frac{1}{3}+\frac{1}{9}(x+2)+\frac{1}{27}(x+2)^{2}+\frac{1}{81}(x+2)^{3}+\mathrm{O}((x+2)^{4})$$

$$\frac{1}{3}+\frac{1}{9}(x+2)+\frac{1}{27}(x+2)^{2}+\frac{1}{81}(x+2)^{3}+\frac{1}{243}(x+2)^{4}+\mathrm{O}((x+2)^{5})$$

$$\frac{1}{3}+\frac{1}{9}(x+2)+\frac{1}{27}(x+2)^{2}+\frac{1}{81}(x+2)^{3}+\frac{1}{243}(x+2)^{4}+\frac{1}{729}(x+2)^{5}+\mathrm{O}((x+2)^{6})$$

$$\frac{1}{3}+\frac{1}{9}(x+2)+\frac{1}{27}(x+2)^{2}+\frac{1}{81}(x+2)^{3}+\frac{1}{243}(x+2)^{4}+\frac{1}{729}(x+2)^{5}+\frac{1}{2187}(x+2)^{6}+\mathrm{O}((x+2)^{7})$$

$$\frac{1}{3}+\frac{1}{9}(x+2)+\frac{1}{27}(x+2)^{2}+\frac{1}{81}(x+2)^{3}+\frac{1}{243}(x+2)^{4}+\frac{1}{729}(x+2)^{5}+\frac{1}{2187}(x+2)^{6}+\frac{1}{6561}(x+2)^{7}+\mathrm{O}((x+2)^{8})$$

$$\frac{1}{3}+\frac{1}{9}(x+2)+\frac{1}{27}(x+2)^{2}+\frac{1}{81}(x+2)^{3}+\frac{1}{243}(x+2)^{4}+\frac{1}{729}(x+2)^{5}+\frac{1}{2187}(x+2)^{6}+\frac{1}{6561}(x+2)^{7}+\frac{1}{19683}(x+2)^{8}+\mathrm{O}((x+2)^{9})$$

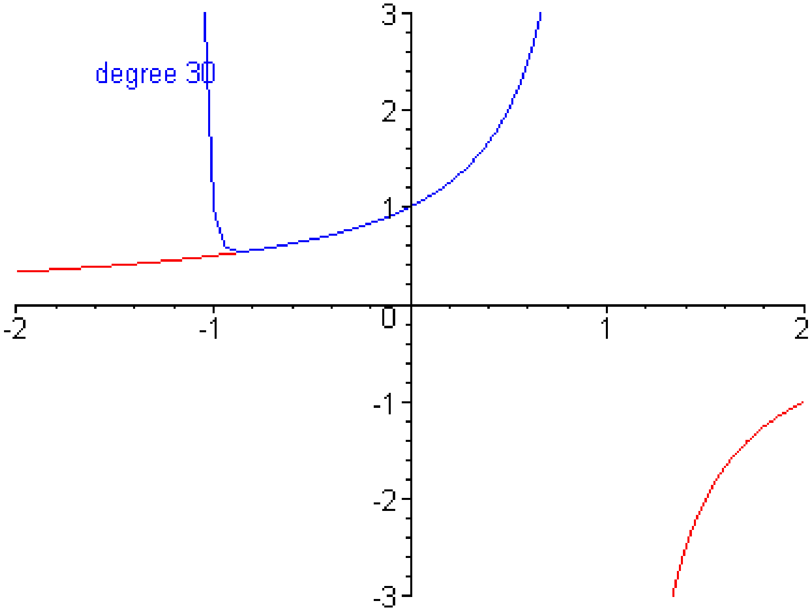
(The O's at the end are Maple's way of indicating that there are other terms in the true, infinite, Taylor series.)

The pattern is relatively simple, but it's not as simple as $\sum_{n=0}^{\infty} x^n$. See if you can discover the pattern. Then use the ratio test to prove that the radius of convergence is 3.

Taylor series expanded about x=0 are often relatively simple. They are distinguished by the name Maclaurin series. So a Maclaurin series is nothing more than a special type of Taylor series, and a Taylor series is a special type of power series.

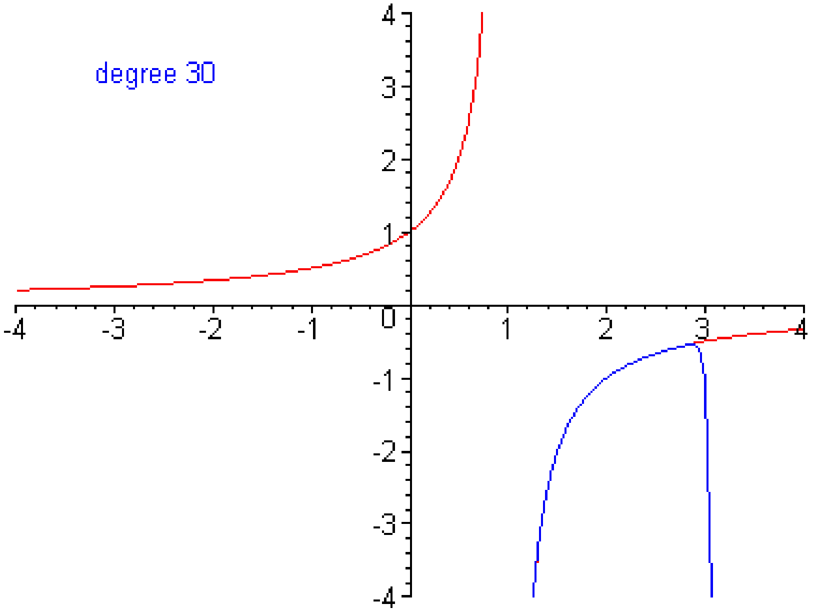
The TaylorAnim command can handle functions that "blow-up" (go to infinity). Just like an ordinary plot command applied to such functions, for this case we need to specify a y-range and the option "discont= true". These options must come after the required first four arguments.

```
> TaylorAnim(f, 0, -2..2, 30, -3..3, discont= true);
```



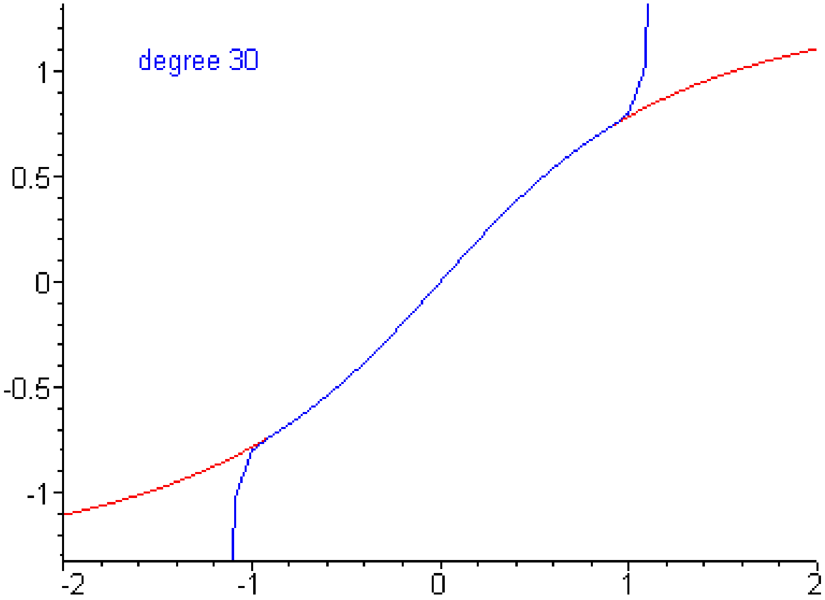
Our function is smooth for x > 1 also, so we should be able to find Taylor series there also. If I expand the Taylor series about x = 2, what do you think the radius of convergence will be?

```
> TaylorAnim(f, 2, -4..4, 30, -4..4, discont= true, scaling= constrained);
```



Even if the function is perfectly smooth everywhere, its Taylor series may still have a finite radius of convergence.

```
> TaylorAnim(arctan, 0, -2..2, 30, axes= frame);
```



The reason that this series has a radius of convergence of only 1 is very easy to figure out using complex numbers, but that is a just little bit beyond the scope of this course.

In the above animation, notice how the polynomials only change on every other iteration. That is because the even-order derivatives evaluate to zero at zero. The Maclaurin series only has odd-degree terms. (That is why functions symmetric with respect to the origin are called odd.)

```
> taylor(arctan(x), x=0, 19);
```

$$x-\frac{1}{3}x^{3}+\frac{1}{5}x^{5}-\frac{1}{7}x^{7}+\frac{1}{9}x^{9}-\frac{1}{11}x^{11}+\frac{1}{13}x^{13}-\frac{1}{15}x^{15}+\frac{1}{17}x^{17}+\mathrm{O}(x^{19})$$

But if I expand around a point other than 0, there is no special even-odd property.

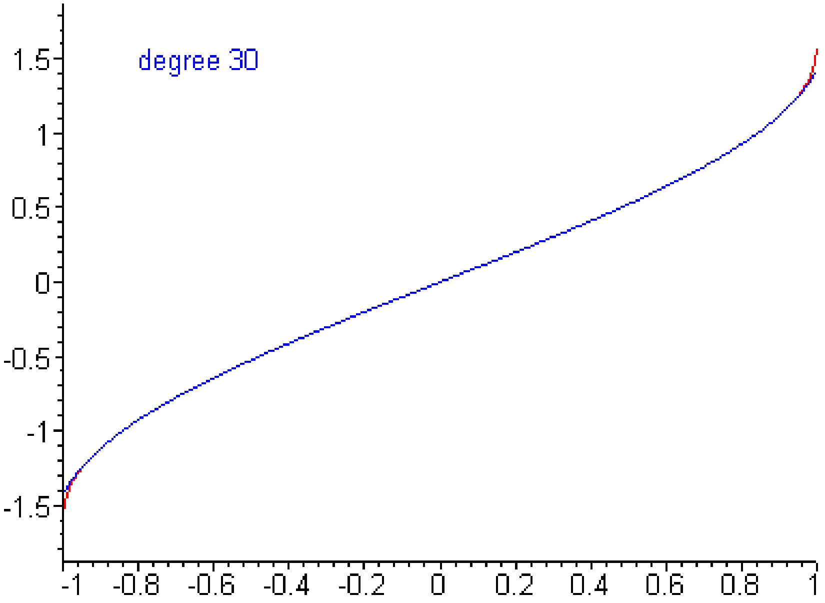
```
> taylor(arctan(x), x=1, 9);
```

$$\frac{1}{4}\pi+\frac{1}{2}(x-1)-\frac{1}{4}(x-1)^{2}+\frac{1}{12}(x-1)^{3}-\frac{1}{40}(x-1)^{5}+\frac{1}{48}(x-1)^{6}-\frac{1}{112}(x-1)^{7}+\mathrm{O}((x-1)^{9})$$

Now we do a function that is only defined on a finite interval. Notice how slow the convergence is at the endpoints.

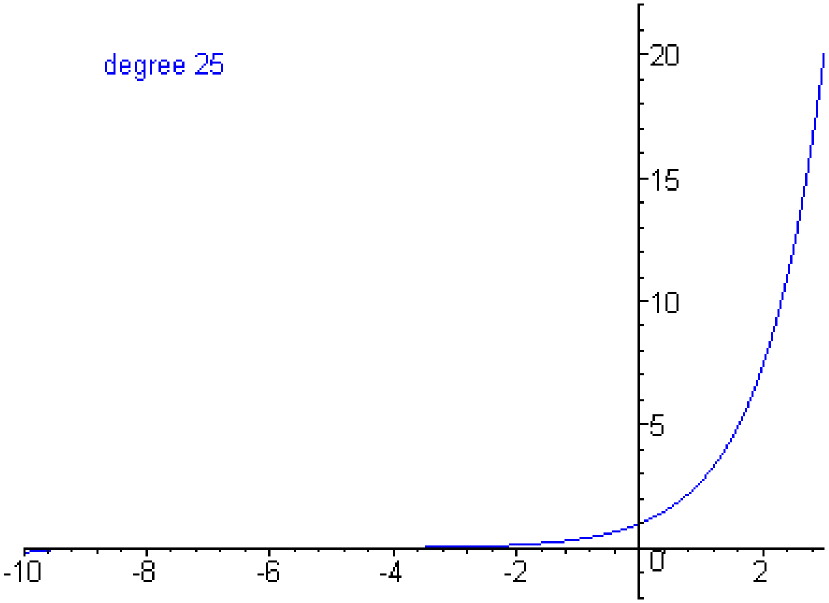
```
> plots[setoptions](numpoints= 200);
```

```
> TaylorAnim(arcsin, 0, -1..1, 30, axes= frame, discont= true);
```



The Maclaurin series for exp(x), sin(x), and cos(x) converge everywhere.

```
> TaylorAnim(exp, 0, -10..3, 25);
```



You can change the color of the polynomials.

```
> #TaylorAnim(sin, 0, -20..20, 30, axes= frame, color= green);
```

The speed of that convergence is amazing considering that the *overall* shape of sine's graph is very different from a polynomial. Sine is wavy forever, but all polynomials have tails that straighten out.

You can also change the color of the primary function.

```
> plots[setoptions](color= green);
```

```
> #TaylorAnim(cos, 0, -20..20, 30, axes= frame);
```

Cosine is an even function, so all terms of its Maclaurin series have even degree.

```
> taylor(cos(x), x=0, 16);
```

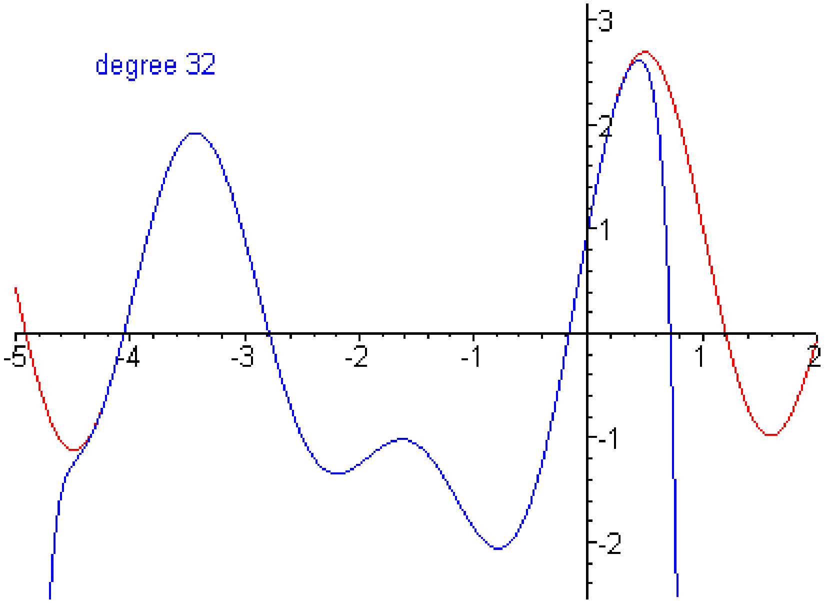
$$1-\frac{1}{2}x^{2}+\frac{1}{24}x^{4}-\frac{1}{720}x^{6}+\frac{1}{40320}x^{8}-\frac{1}{3628800}x^{10}+\frac{1}{479001600}x^{12}-\frac{1}{87178291200}x^{14}+\mathrm{O}(x^{16})$$

Even extremely complicated and wavy functions can have Taylor series that converge everywhere as long the derivatives of all orders exist. But don't make the mistake of assuming that just because the derivatives of all orders exist that the Taylor series necessarily converges everywhere. We saw with the arctangent that that was not true.

```
> f:= x->exp(cos(x))*sin(x)+cos(2*x)+sin(Pi*x):
```

```
> plots[setoptions](color= red);
```

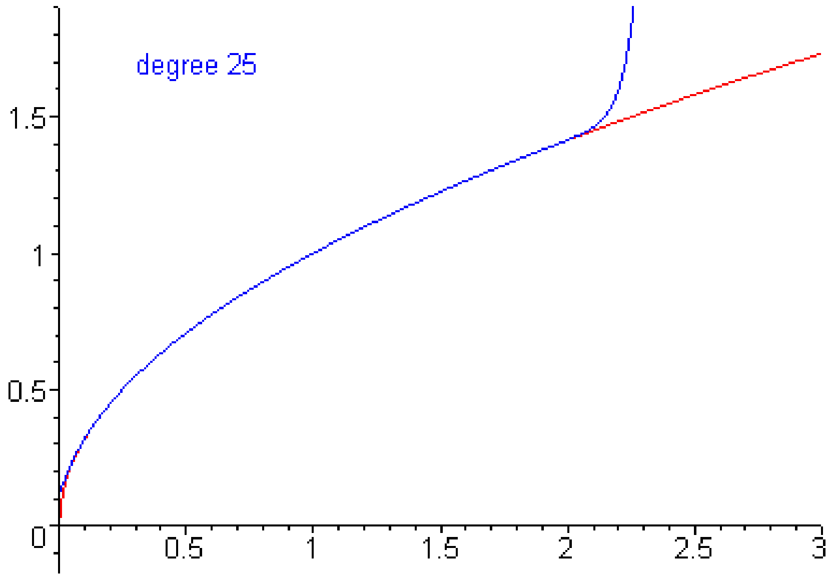
```
> TaylorAnim(f, -2, -5..2, 32);
```



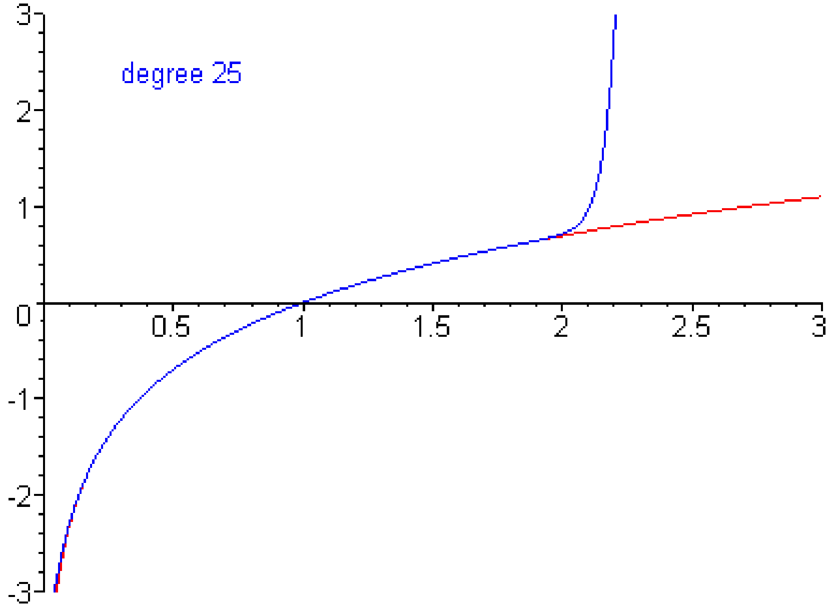
The amazing thing about this is that the value of the function at any point can be determined just by knowing it's value and all it's derivatives values at a single point. Another way of looking at it is this: All of the turns of the function may be determined by knowing, on a minute scale, how it turns at a single point.

Now see what you can discover on your own about Taylor series by using this plotting tool.

```
>  TaylorAnim(x->sqrt(x), 1, 0..3, 25);
```

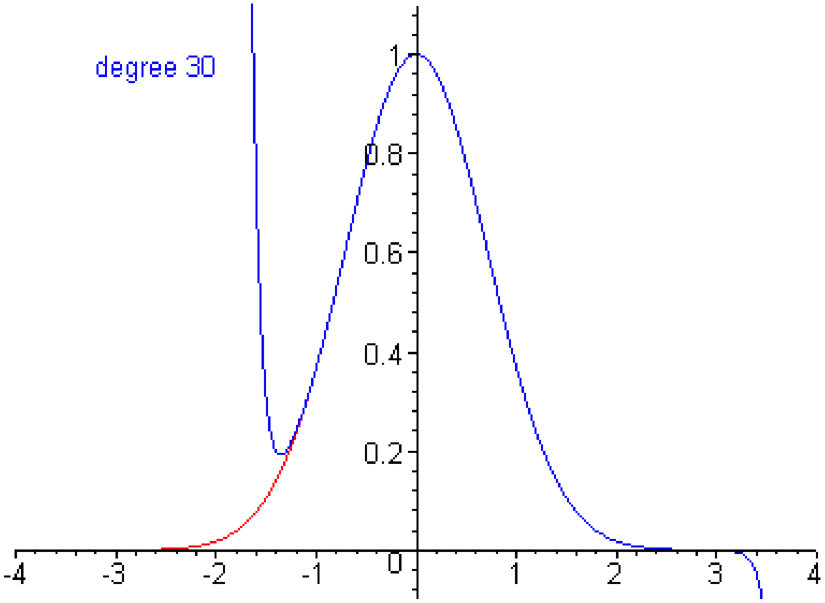


```
> TaylorAnim(x->ln(x), 1, 0..3, 25, discont= true, -3..3);
```



In the following animation, notice how wild some of the polynomials get at x-values sufficiently far from the central point.

```
> TaylorAnim(x->exp(-x^2), 1, -4..4, 30);
```



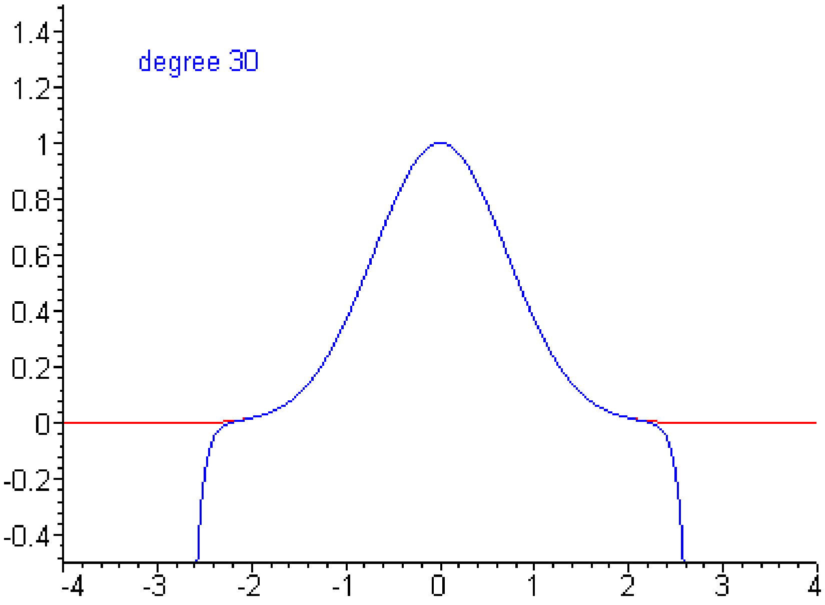
If you don't want to see those false sharp corners on some of the polynomials, you could set 'numpoints' to a higher value. The computation time will only be increased slightly, but the memory usage is increased significantly.

```
> plots[setoptions](numpoints= 300);
```

```
> TaylorAnim(x->exp(-x^2), -1, -4..4, 30):
```

For this function, if we expand about x=0, they won't be so wild.

```
> TaylorAnim(x->exp(-x^2), 0, -4..4, 30, -0.5..1.5, axes= frame);
```



```
> plots[setoptions](numpoints= 50);
```

The following function is extremely important in statistics. It gives the area under the "bell curve".

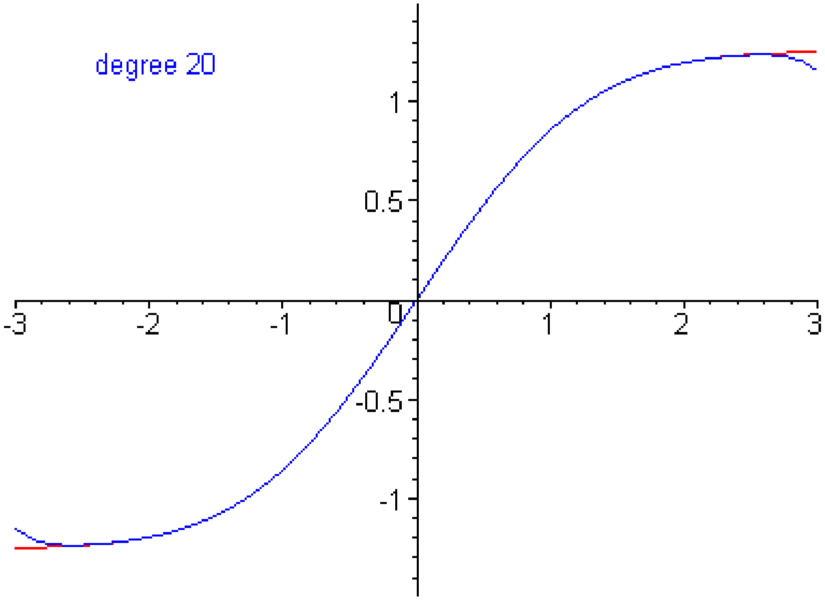
```
> f:= x -> int(exp(-t^2/2), t= 0..x):
```

The antiderivative cannot be expressed in terms of elementary functions. But the Maclaurin series converges very fast.

```
> taylor(f(x), x=0, 16);
```

$$x-\frac{1}{6}x^3+\frac{1}{40}x^5-\frac{1}{336}x^7+\frac{1}{3456}x^9-\frac{1}{42240}x^{11}+\frac{1}{599040}x^{13}-\frac{1}{9676800}x^{15}+O(x^{16})$$

```
> TaylorAnim(f, 0, -3..3, 20);
```



The inverse of the above function is also extremely important in statistics. But it cannot even be expressed as an integral. However, its derivatives can be found with the Inverse Function Theorem. Maple allows us to express the inverse this way:

```
> g:= x -> solve(f(y)=x, y):
```

The diff command will not give a very useful derivative.

```
> diff(g(x), x);
```

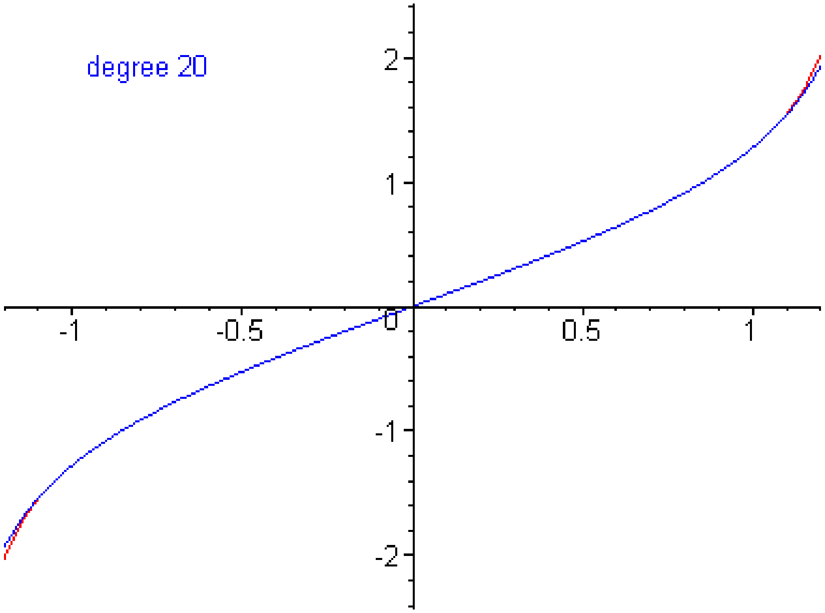
$$\frac{1}{\mathbf{e}^{(-\mathrm{RootOf}(\mathrm{erf}(\_Z)\sqrt{\pi}-\sqrt{2}\,x)^2)}}$$

But, amazingly, the taylor command will give a useful answer.

```
> taylor(g(x), x=0, 10);
```

$$x+\frac{1}{6}x^3+\frac{7}{120}x^5+\frac{127}{5040}x^7+\frac{4369}{362880}x^9+\mathrm{O}(x^{10})$$

```
> TaylorAnim(g, 0, -1.2..1.2, 20);
```


degree 20
2
1
-1
-2
-1
-0.5
0
0.5
1

>