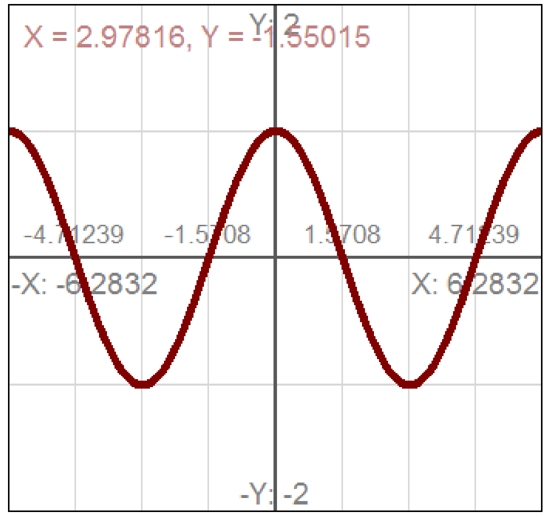
Taylor Polynomials II

Once we understand how to find Taylor polynomials, the next question is: how many terms do we need to find, and how good is the approximation? Let's consider a couple different functions and look at the graphs. We can then visually compare the quality of the approximation. After we do this for several example functions, we will consider error estimation.

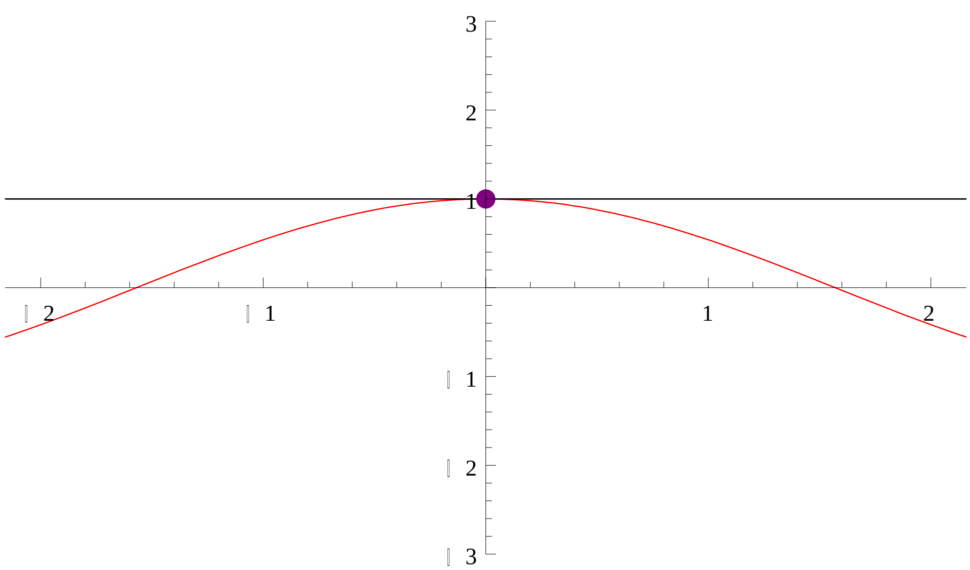
Example 1. Let's start with the example of *cos(x).*



The graph of the regular cosine function is shown for two complete periods. This is a fairly complicated function to model with a polynomial, but let's see how it works.

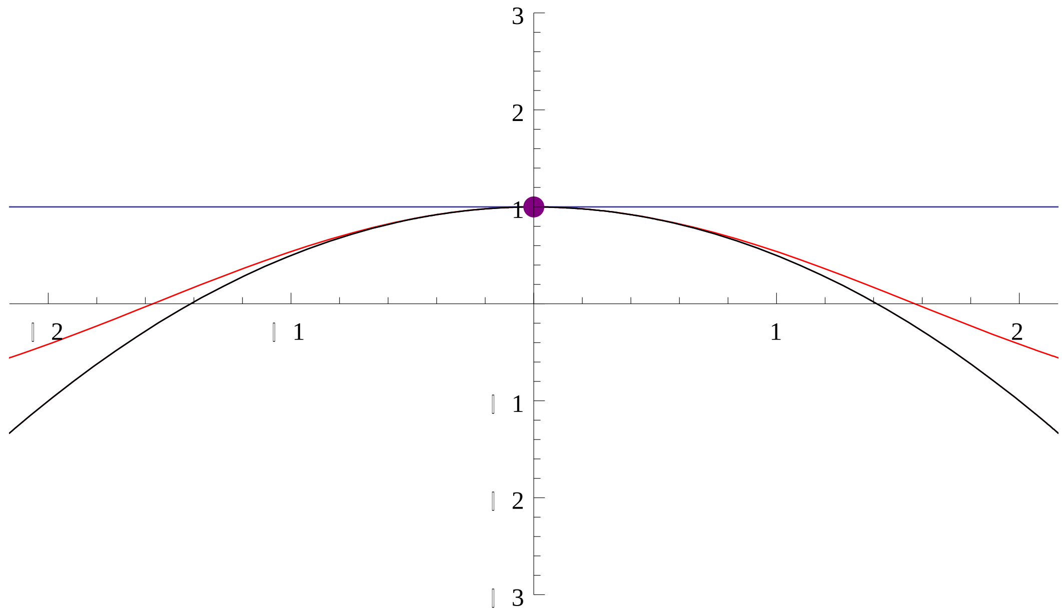
The Taylor polynomial expansion begins with f(c). For the sake of simplicity here, let's let *c*=0. In this case*, f(0)=cos(0)=1*. So our 0-order Taylor polynomial is just *y=1*. Since cosine has no linear term in the expansion (we can see this by taking the derivative and letting *c*=0 again, we get a zero). So *y=1* is also the first order Taylor polynomial as well.

The graph below shows both the original cosine function and the Taylor polynomial centered at zero.



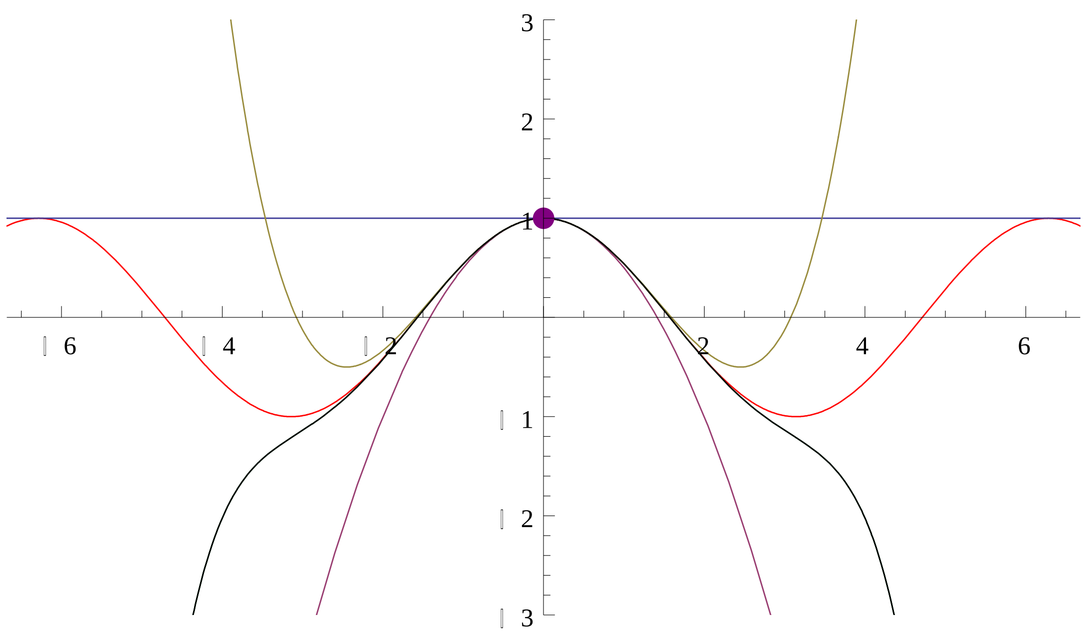
We've zoomed in on *x*=0. For small angles (in radians, of course), *y=1* is a decent approximation to *f(x)=cos(x).* By a small angle here, I'm thinking -0.4<*x*<0.4 based on the graph. That's about 22°. I'd estimate the error is around 0.1. Of course, the tolerable error size depends on your application. Outside these values, the approximation becomes increasingly worse, until it reaches a maximum at multiples of π.

Now let's consider the second (and third) order Taylor polynomial for cosine.



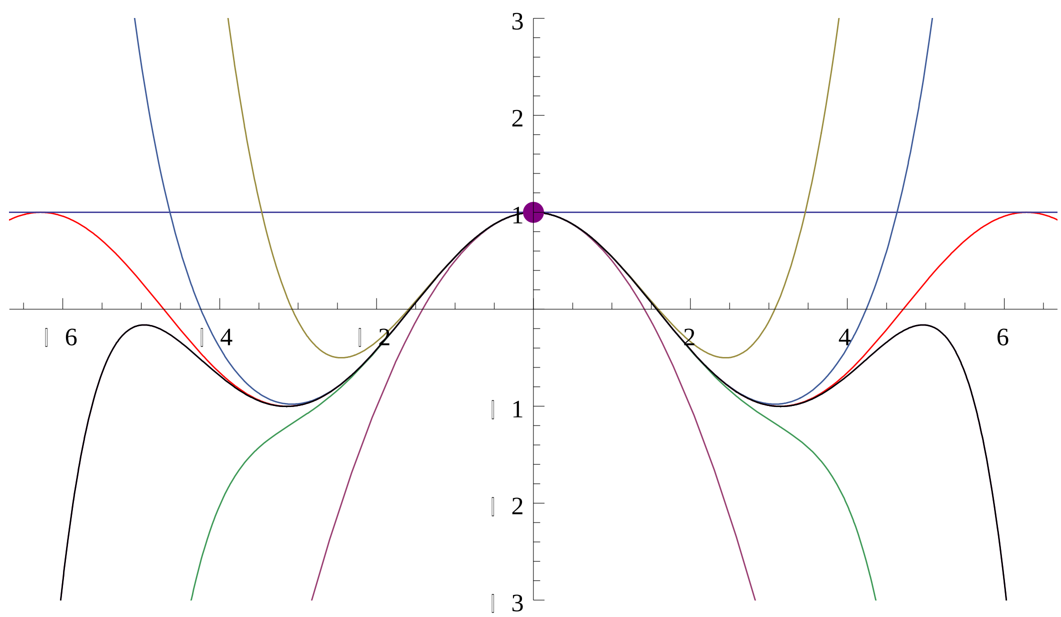
That new bottom curve is the new approximation. You can work out for yourself that the approximation is now $y = 1 - \frac{x^2}{2}$. This approximation shows substantial improvement over the 0-order approximation because the graphs are practically on top of each other well past our ±0.4 points. The separation becomes visible in our graph only around ±1 radian≈57°.

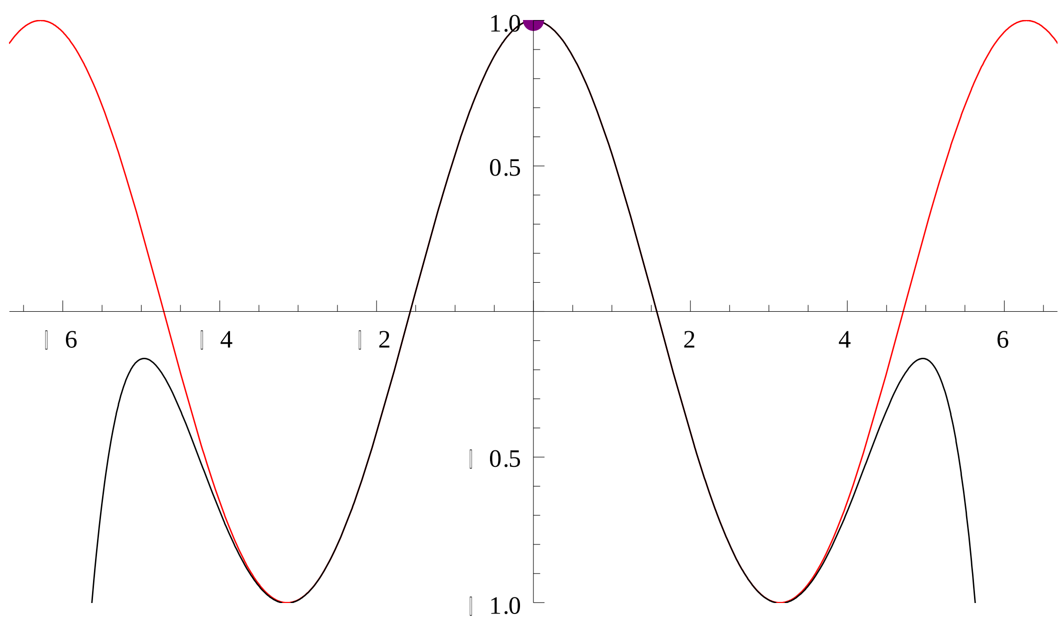
Let's consider the next two Taylor polynomials for *f(x)=cos(x)*. The 6th-degree Taylor polynomial is now



$y = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720}$. With each new additional term, the errors close to zero will get smaller and smaller (much less than 1% for small angles), and decent quality estimates (under the 0.1 we've been using) will extend further and further from zero. If we take still more terms, we can do still better. The graph below shows the 8th and 10th degree polynomials, up to

$$y = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{40320} - \frac{x^{10}}{3628800}.$$

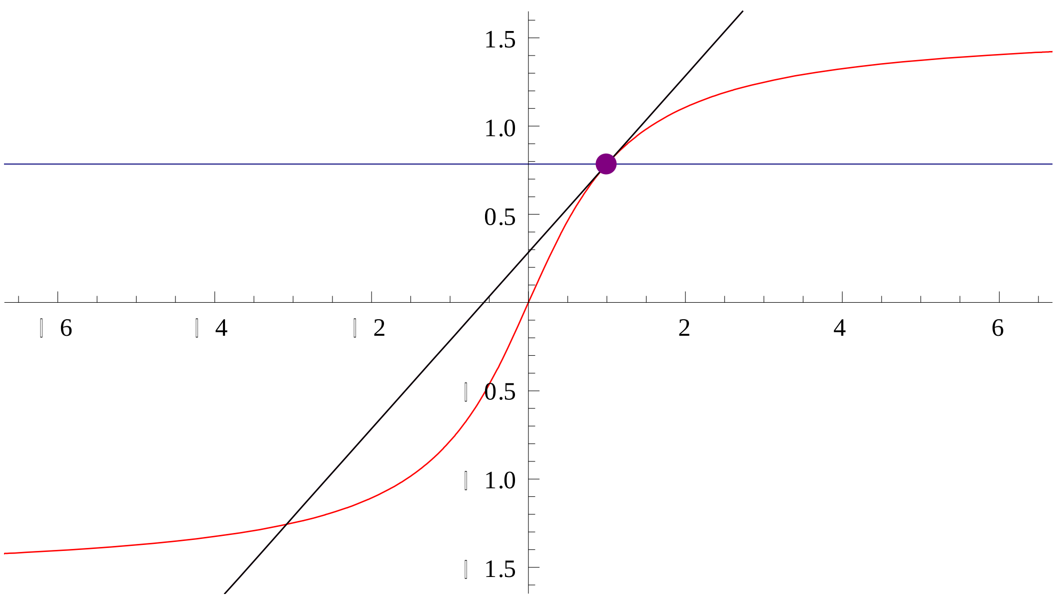




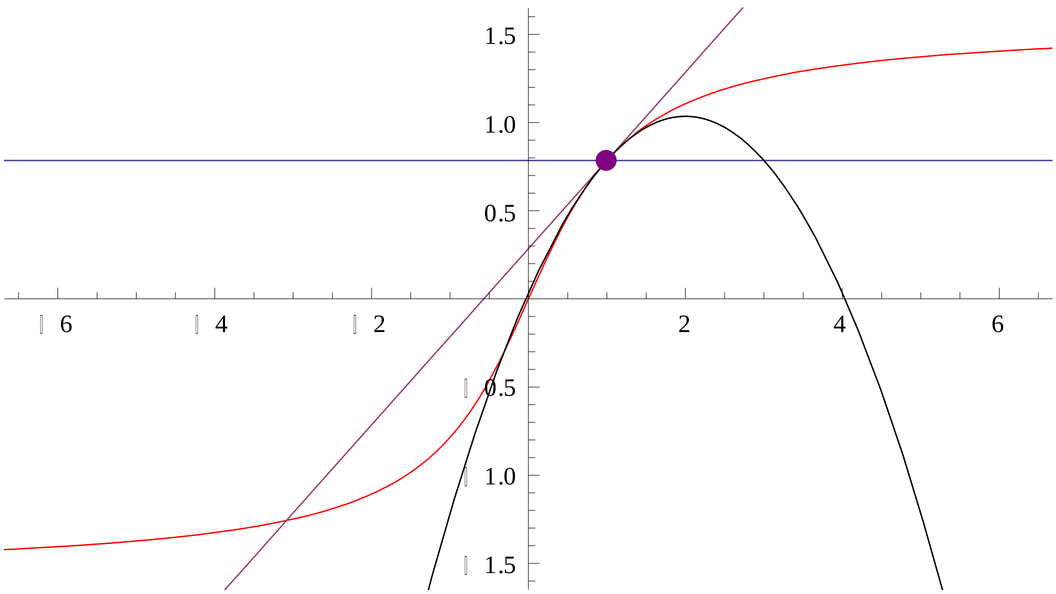
Taylor polynomials are always approximations, but as we increase the number of terms, and our estimates improve we can get arbitrarily good approximations for any value of *x*, even if the approximation is centered at 0. If we take an infinite number of terms, we can reproduce the function exactly. This is a Taylor series.

Practice: Find the next two terms of the Taylor polynomial for cosine (that's up to the 14th-degree term). Obtain the graph of the polynomial. (You can use a program called GraphCalc to print it out.) Estimate from the graph for what angles (in radians) the graph is with 0.1 of the original cosine function. Use that number to calculate the actual error (based on the value of *cos(x)* at that point and the value of the estimate at the same point). Based on your equation, establish a formula for the successive terms of the Taylor series (your formula may be in terms of the series, rather than orders of the Taylor polynomial).

Example 2. For the second example, let's consider $f(x) = \tan^{-1}(x)$, and let's choose *c*=1 this time. The Taylor polynomial for this function has terms of all orders but calculating derivatives is much harder. So, let's consider the zeroth-order polynomial. This is $y = \tan^{-1}(1) = \frac{\pi}{4}$. As you can see from the graph, the constant function is an extremely poor approximation of the polynomial, which is why the linear, or first-order Taylor polynomial is usually the smallest order used on a practical level. That approximation is $y = \frac{\pi}{4} + \frac{1}{2}(x-1)$. The graph of both approximations, together with the original inverse tangent function is shown below.



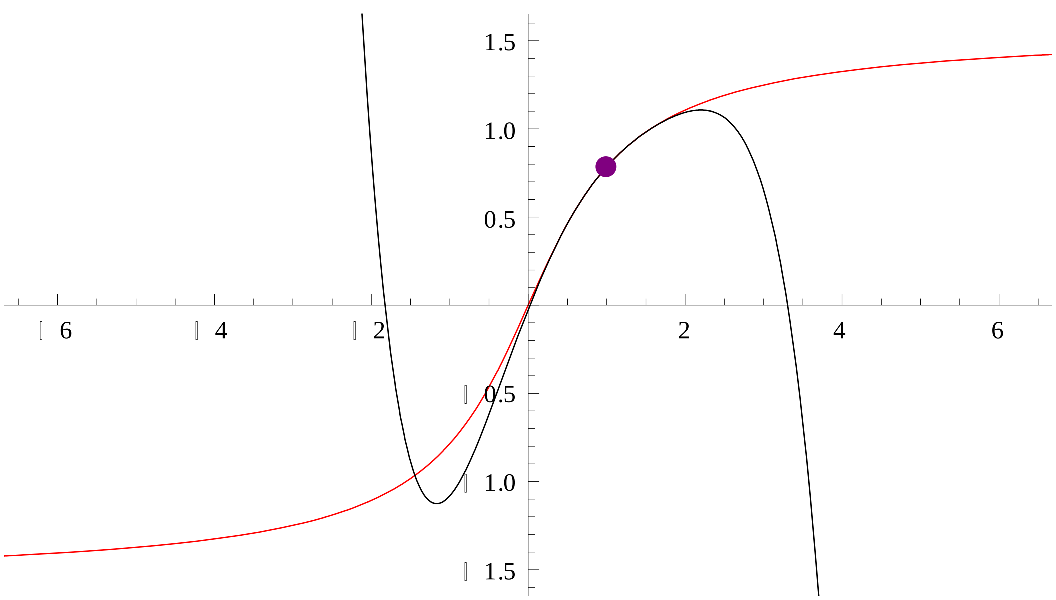
The first-degree approximation is not bad for tangent values close to one radian, but basically just a rough estimate. As with the cosine graph, the second order $y = \frac{\pi}{4} + \frac{1}{2}(x-1) - \frac{1}{4}(x-1)^2$ does a much better job:



Look how close that approximation is now all the way back into negative values! Unlike the cosine function, the inverse tangent graph (in part because we've shifted away from the origin) has significant asymmetry in the range of values for which it is a good approximation.

Let's jump of several steps and see how good we can get from the fifth-degree approximation

$$y = \frac{\pi}{4} + \frac{1}{2}(x-1) - \frac{1}{4}(x-1)^2 + \frac{1}{12}(x-1)^3 - \frac{1}{40}(x-1)^5.$$



We can see a couple interesting features here. The second-degree approximation actually does a better job at *x*= -0.5 than does the 5th degree equation. However, as expected, the match for values close to one radian is phenomenal. One of the curious things about this Taylor expansion is that the fourth-degree term is missing. This is repeated in the 8^{th}-degree term as well, and all other multiples of 4, those powers are missing. This will make the job of coming up with a Taylor series formula much more difficult.

There is always a trade-off between the number of terms you use and the value of the error. This is generally why we don't use Maclaurin polynomials for everything. Rather than using more terms, in terms of calculation effort, it's better to change the center of expansion than to use more and more terms of the series. However, if our intention is to work with the series (in the form of a general expression for the terms), centering the expression at zero makes the series easier to write, and the infinite series is an arbitrarily close match at any point.

Practice: Calculate the true error for $f(x) = \tan^{-1}(x)$ at the point *x*= -0.5 and at *x*= 0.5. Which Taylor polynomial is the best match in each case? Can you explain why the 4^{th}-degree (and other multiples of the 4^{th} degree) vanish?

Error Estimates. So far, we've been calculating true errors: the actual difference between a particular approximation and the true value of the function. While this is always the most accurate, what if we want to determine how many terms we will need to get less than a given error range, without actually working out more terms of the series than we will actually end up needing? This is where the error function comes in.

The formula for Taylor polynomials our textbook gives is:

$$f(x) = f(c) + f'(c)(x-c) + \frac{f''(c)(x-c)^2}{2!} + \frac{f'''(c)(x-c)^3}{3!} + \ldots + \frac{f^{(n)}(c)(x-c)^n}{n!} + R_n$$

where R_n is the remainder. That formula is given separately, and we can think of it as the next non-zero term in the series: $R_n = \dfrac{f^{(n+1)}(z)(x-c)^{n+1}}{(n+1)!}$. All the values are familiar except the *z*, which represents some number (unknown to us) at which we are evaluating the *n*+1^{st} derivative. For some functions we will be able to choose a maximum value for this number that covers the entire function. In other cases, we will restrict the domain over which the error estimate applies, and use the maximum of function in that range. Because there are two different scenarios here, we will do two examples.

Example 3. For the cosine function we can choose a maximum value over all derivatives that can be used for $f^{(n+1)}(z)$ since the derivatives of cosine are always *±sin(x)* or *±cos(x),* we know the maximum value of any derivative is always ±1. Since we don't care about the sign when we are calculating errors (since they are themselves absolute values), we can replace $f^{(n+1)}(z)$ with 1 in any cosine or sine function error estimate.

Let's consider the scenario where we'd like to expand *f(x)=cos(x)* function as a Maclaurin polynomial, as we did in Example 1, and we'd like to determine how many terms we will need to get an error estimate of less than 0.0001 for some given value of x. Let's say, *x*= -1 radians. Because we are using cosine and a Maclaurin series, we get the following: $0.0001 \geq \left| \dfrac{(1)(x)^{n+1}}{(n+1)!} \right|$. If we replace *x* with -1, this reduces further to $0.0001 \geq \left| \frac{1}{(n+1)!} \right|$. Getting *n* on one side we get: $(n+1)! \geq \frac{1}{0.0001} = 10,000$. For what *n* is ($n$+1)!≥10,000? When *n*=7, *n*+1=8 and 8!=40,320. This means we will need 7 terms in the series to get our Taylor polynomial within 0.0001 of the true value of cosine at *x*= -1 radians.

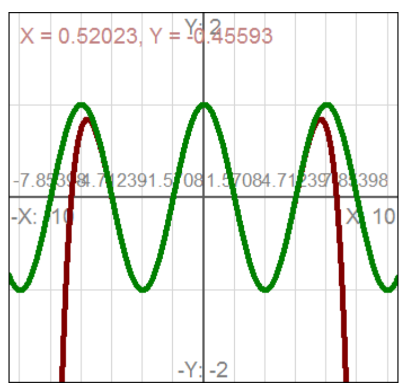
To be sure, this is an estimate. It may very be true with fewer terms, but we know that we won't need **more** terms than this.

Practice: Repeat this process for *f(x)=sin(x)*. Estimate the number of terms you will need to approximate *f(x)* with a Maclaurin polynomial at *x*=π, with an error of 0.001. Because x^{n+1} does not reduce away quite as easily as in our example, you may want to evaluate the expression numerically in your calculator table.

Example 4. Consider the function $f(x)=e^x$. If we expand this function around x=1, we get the following Taylor polynomial (out to 5 terms): $y = e + e(x-1) + \frac{1}{2}e(x-1)^2 + \frac{1}{6}e(x-1)^3 + \frac{1}{24}e(x-1)^4 + \frac{1}{120}e(x-1)^5$.... We'd like to know, how many terms do we need to approximate this function within 0.0001 at *x*=e. This is a relatively straightforward derivative in that every $f^{(n)}(x) = e^x$. But this function has no maximum value. It never gets smaller than zero, but blows up to infinity. In this case, we will choose *z* to be either *c* or *x* (since there are also no critical points). We will test both values, and choose the larger value (the value that makes the function larger, that is) for the $f^{(n+1)}(z)$ term as *z*. Since $e^e > e^1$, we will use $f^{(n+1)}(e) = e^e$ (notice that if *x*<1, we would use e^1). Our error expression then becomes $0.0001 \geq \left| \frac{(e^e)(e-1)^{n+1}}{(n+1)!} \right|$. I will have to evaluate this numerically in my calculator to get an answer. In scientific notation 0.0001=1×10^{-4}, so I need a number smaller than this (i.e. something with a 10^{-5} next to it). Entering this function on the Y= screen and replacing n with x, I find that *n* = 11.

x	y
0	26.0393
1	22.3714
2	12.8135
3	5.50429
4	1.89158
5	0.54171
6	0.13297
7	0.02856
8	0.00545
9	0.00094
10	0.00015
11	2E-005
12	3E-006

Practice: Let's do this another way. Consider the 5th-degree Taylor polynomial for $f(x)=e^x$ given in Example 4. How large (or small) an *x* can I use and still be within an error 0.0001. Here, *n* is give (*n*=5) and I am looking for *x* in the expression. For the values larger than one, $f^{(n+1)}(z) = e^x$, and for values smaller than one, $f^{(n+1)}(z) = e^1$. Estimate your answers to 3 decimal places.



A1: $y = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \frac{x^{12}}{12!} - \frac{x^{14}}{14!}$, 10% error @±4.4 radians≈0.104, series $\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}$

A2: x=0.5, 5th-degree≈0.00038, x= -0.5, 2nd-degree≈0.08; 4th derivative is 0 at x=1

A3: ≈14

A4: 0.455<x<1.501