ChEE 201
University of Arizona
Computer Reading 5
Taylor Series Approximations

Students learned the Taylor Series Approximation in Reading 4. Here, we'll build upon that knowledge and add a few more details so we can handle functions that have more than one variable. Some more computer programming tools will also be discussed, like how to add comments to a computer program so it is easier to see what the program is doing. This makes sense as the programs keep getting longer and longer with more complexity.

At the end of this section, students will be able to:
1) use the Taylor series expansion to represent a function of two variable as a linear equation
2) use successive approximations with Taylor series' when a single approximation has problems
3) see the utility of adding comments to a computer program

<u>Second Order Taylor Series Approximation:</u>
The Taylor series approximation was given in the past section as:

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)\frac{1}{1!}(x_{i+1}-x_i) + f''(x_i)\frac{1}{2!}(x_{i+1}-x_i)^2 + \cdots + f^{(n)}(x_i)\frac{1}{n!}(x_{i+1}-x_i)^n$$

We looked at several examples like using the Taylor series to find ln(x), e^x, sin(x), etc. But what can we do if we have a function of more than one variable? Let's say that you wanted to do a Taylor series expansion to find $e^2\ln(2)$? Our general function now would be $e^x\ln(y)$. There is a way to break up the Taylor series for two variables which uses partial derivatives. The general form is:

$$f(x_{i+1}, y_{i+1}) \approx f(x_i, y_i) + \left.\frac{\partial f}{\partial x}\right|_{x_i, y_i}\frac{1}{1!}(x_{i+1}-x_i) + \left.\frac{\partial f}{\partial y}\right|_{x_i, y_i}\frac{1}{1!}(y_{i+1}-y_i) + \left.\frac{\partial^2 f}{\partial x^2}\right|_{x_i, y_i}\frac{1}{2!}(x_{i+1}-x_i)^2 +$$

$$\left.\frac{\partial^2 f}{\partial y^2}\right|_{x_i, y_i}\frac{1}{2!}(y_{i+1}-y_i)^2 + \frac{2}{2!}\left(\left.\frac{\partial^2 f}{\partial x \partial y}\right|_{x_i, y_i}\right)\left((x_{i+1}-x_i)(y_{i+1}-y_i)\right)\cdots + \textit{higher order terms}$$

Here, we see that things have gotten more complex because we now have to take the derivative with respect to more than one variable at a time and also evaluate the function more often. Just to complete the first order corrections, we have to add two terms to the approximate answer evaluated at the starting point. Then we add three more terms to get to the second order approximations. Let's work through our example of $e^2\ln(2)$.

We will begin by selecting x_{i+1} as 2 and y_{i+1} as 2. What do you think are good values for x_i and y_i? Write them down and then continue on to the next paragraph to compare.

A good choice for x_i is 0 because we know that $e^0 = 1$. $y_i = 1$ is also good because we know ln(1)=0. Now that we have that, let's find the first derivative of our function with respect to (w.r.t) x:

$$\frac{\partial f}{\partial x} = \frac{\partial}{\partial x}\left(e^x \ln(y)\right) = e^x \ln(y)$$

And the first derivative of our function w.r.t y:

$$\frac{\partial \mathrm{f}}{\partial \mathrm{y}} = \frac{\partial}{\partial \mathrm{y}}\left(\mathrm{e}^{\mathrm{x}} \ln(\mathrm{y})\right) = \mathrm{e}^{\mathrm{x}}\frac{1}{\mathrm{y}}$$

Then we evaluate both functions at x_i,y_i to get:

$$\left.\frac{\partial f}{\partial x}\right|_{x_{i.}, y_i} = e^0 \ln(1) = (1)(0) = 0 \qquad \left.\frac{\partial f}{\partial y}\right|_{x_{i.}, y_i} = e^0\frac{1}{1} = (1)(1) = 1$$

So, let's put together our approximation through the end of the first order terms:

$$f\left(x_{i+1}, y_{i+1}\right) \approx (1)(0) + (0)\left(\frac{2-0}{1}\right) + (1)\left(\frac{2-1}{1}\right) + \cdots = 0 + 0 + 1 = 1$$

Our actual solution (by using a calculator), should be 5.1217. Our true error and true relative error are 4.1217 and 0.80475.

So far, so good. Let's now do the second derivatives and evaluate those at x_i,y_i:

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\right)\left(e^x \ln(y)\right) = e^x \ln(y) \qquad \left.\frac{\partial^2 f}{\partial x^2}\right|_{x_{i.}, y_i} = e^0 \ln(1) = (1)0 = 0$$

$$\frac{\partial^2 f}{\partial y^2} = \frac{\partial}{\partial y}\left(\frac{\partial}{\partial y}\right)\left(e^x \ln(y)\right) = \frac{\partial}{\partial y}\left(\frac{e^x}{y}\right) = -\frac{e^x}{y^2} \qquad \left.\frac{\partial^2 f}{\partial y^2}\right|_{x_{i.}, y_i} = -\frac{e^0}{1^2} = -\frac{(1)}{1} = -1$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x}\left(\frac{\partial}{\partial y}\right)\left(e^x \ln(y)\right) = \frac{\partial}{\partial y} e^x \ln(y) = \frac{e^x}{y} \qquad \left.\frac{\partial^2 f}{\partial x \partial y}\right|_{x_{i.}, y_i} = \frac{e^0}{1} = \frac{1}{1} = 1$$

And now we plug all of this into our equation and we get:

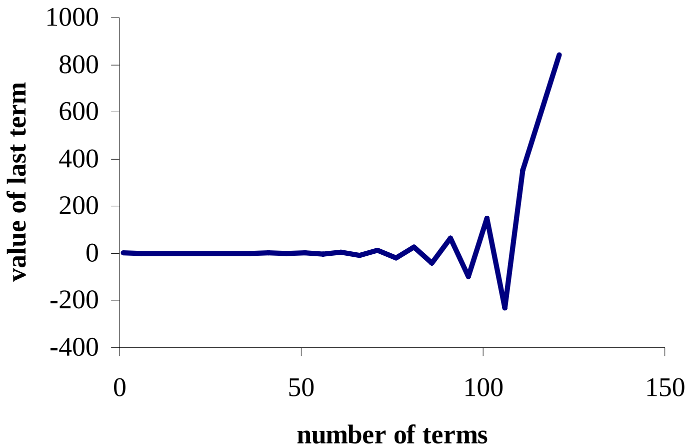
$$f\left(x_{i+1}, y_{i+1}\right) \approx 1 + \frac{0}{2}(2-0)^2 - \frac{1}{2}(2-1)^2 + \frac{2(1)}{2}\left((2-0)(2-1)\right) + \cdots = 1 + 0 - \frac{1}{2} + 2 = \frac{5}{2}$$

At this point, our true error is 2.62 and the true relative error is 0.5115. This is still too high of an error so we could go to the next higher derivative to add third order corrections, which would add another four terms to continue to improve our answer. We'll stop here with our example, though. You should notice that the two variable functions often converge slower than other functions so more terms may be needed. For the chemical engineering students, they will find that multivariable Taylor series approximations will be useful in their Control theory course later on in their academic careers.

There are a few things to be careful of when we are doing the higher order terms, which is to make sure we use the right differences between the i+1 and i values. The next is to make sure we get the prefactor correct (like remembering the 2 in the equation above). Other than that, we just need to be careful and keep our numbers and terms clear and orderly on paper.

<u>Successive Approximations:</u>

There are some times when the functions we are trying to evaluate cause problems for us when we try to apply the Taylor series. One example that will come back in the next homework is when you try to do a Taylor series expansion for ln(x) using 1 as your x_i value and try to use that to find the ln of any number larger than 2.0. You could go back to your fourth homework solution and try running the program for ln(2.1), as an example. You can increase the number of steps by changing the stopping value in your program or in the solution key if your program didn't work. No matter how high you increase the number of steps you'll let the program go, your value will never converge down to a final value. Try implementing a msgbox right before your program attempts to return a value to Excel and have it report the value of the last term that was calculated. If you count up high enough in the steps, the function diverges. This is because the upper part of your infinite sum $(2.1\text{-}1)^n$ starts to outstrip your denominator, n, and the function begins too grow to rapidly. For any value of Δx^n where x is larger than 1, we'll have divergence. Let's look at the specific case here when we are trying to find ln(2.1). We have a plot on the next page of the value of the last term in the series as a function of the number of steps included in the series for ln(2.1). We see that after about 65 terms or so that the function becomes unstable as the terms keep getting larger and larger in magnitude as they oscillate from positive to negative. This shows graphically what is happening to the final term, which would be added to the total sum at that point and lead that function to diverge as well.



There are still ways to implement our Taylor series approximation, but we just need to be a little more clever. A good analogy is to think of mountain climbers. They might not be able to hike all the way to the top of the mountain in just one day. Instead, they hike so far on the first day and then stop. Then they continue on the next day. We can do the same type of "resting" by using a successive approximation. On the first calculations to find the ln of a number between 2 and three, we'll move from our starting point of 1 to an ending point at 2 since we still get convergence here, even though it took 145 steps. Then, we'll use that answer to find the approximate value we are trying to find. Let's examine how this would come about in terms of the equations by starting with the general Taylor series equation:

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)\frac{1}{1!}(x_{i+1} - x_i) + f''(x_i)\frac{1}{2!}(x_{i+1} - x_i)^2 + \cdots$$

If we weren't doing successive approximations, we would have just done:

$$\ln(2.1) \approx \ln(1) + f'(1)\frac{1}{1!}(2.1 - 1) + f''(x_i)\frac{1}{2!}(2.1 - 1)^2 + \cdots$$

where x_i is 1 and x_{i+1} is 2.1.

Let's instead replace this by a shifted function where we first find the Taylor series for ln(2):

$$\ln(2) \approx \ln(1) + f'(1)\frac{1}{1!}(2 - 1) + f''(x_i)\frac{1}{2!}(2 - 1)^2 + \cdots = \ln(2)_{appr}$$

where x_i is still 1, but x_{i+1} is 2.0 since we know this will converge to an answer. We end up with $\ln(2)_{appr}$ as a number we can then use to find the shifted successive approximation in a new Taylor series:

$$f(2.1) \approx \ln(2)_{app1} + f'(2)\frac{1}{1!}(2.1 - 2) + f''(2)\frac{1}{2!}(2.1 - 2)^2 + \cdots$$

where x_i is 2.0 and x_{i+1} is 2.1. Note that you have to be careful in this second approximation that you are evaluating the derivatives now at your new x_i, which is 2. In our original infinite sum for getting $\ln(2)_{appr}$, we would have evaluated them at 1.

This will allow us to find the ln all the way up to 3. If we wanted to go higher, we'd have to go through two approximations, one from 1 to 2 and then one from 2 to 3 before we could do our final sums to get the ln of the number we wanted. You will implement the above equations in a VBA program to approximate the value of the ln of a number between 2 and 3 so ask questions about these parts if you need help.

<u>Adding Comments to Computer Code:</u>

Some of our programs are now getting long enough that it is starting to become attractive to add some extra language around the code that will help us remember what we did and which variables are being used for what.

Let's go back to HW 3 and use the example from there to show what we mean. It's been a while, but here is what the code looked like for problem number one on computer homework 3:

```
Option Explicit
Function approx(x, n)
Dim sum As Single, i As Single

sum = 0

For i = 1 To n
   sum = sum + (-1) ^ (i + 1) * (x ^ i) / i
Next i

approx = sum

End Function
```

Looking at this code, can you even remember what this was supposed to do? Can you look through the code and tell what the answer might be? What is the variable *sum* doing for us? Let's rewrite this program and add some comments to it so we can see how to do this well. To add a comment, simply begin the line with an apostrophe, '.

```
Option Explicit
Function approx(x, n)
'this program is to approximate the value of  ln(1+x) to n terms
'x and n are sent from Excel

'sum is going to be the variable that we manipulate to find ln(1+x)
'i is going to be our counting variable for our loop.  It will go from 1 to n
Dim sum As Single, i As Single

'sum is set equal to zero to remove any previous values that may be in the computer for sum
sum = 0

'this loop is where we do our infinite series we derived in the homework on paper
For i = 1 To n
   sum = sum + (-1) ^ (i + 1) * (x ^ i) / i
Next i

'now our answer is sent back to the Excel file
approx = sum

End Function
```

Was this easier to examine now that we've added the comment lines? It should be. In fact, even if you weren't part of writing a specific program, a well commented program becomes easy for you to break apart and figure out. You should consider commenting your programs as they get longer and more complex so that you can easily open them up and remember what you were doing.