Preparation for Fourier Series, Taylor Series:

a visual approach to approximation

©Mike May, S.J., maymk@slu.edu, Saint Louis University

>

Background and introduction

Why we look to Calculus while studying Linear Algebra
One of the challenging topics of linear algebra is the use of inner product spaces to project onto a subspace. The underlying concept that is often missed is that the process gives the best approximation of a vector with a vector in a given subspace.

This process has a nice visualization with Fourier series, but the computations are challenging enough that students often miss the point of approximation. Since most students in linear algebra have had at least a year of calculus, it is useful to refer back to Taylor series where approximation has a similar visualization.

A fast review of Taylor Polynomials
One of the ongoing themes of calculus is polynomial approximation of functions.

The tangent line can be described as the "best linear approximation" to a curve at a point.

When we first introduced second derivatives, we looked at the "best quadratic approximation" of a curve at a point. This was a quadratic polynomial that p(x) that approximated f(x) at a given point x=c, with

$$p(c) = f(x),\ \ p'(c) = f'(c), \text{ and } p''(c) = f''(c).$$

We have continued this theme by studying Taylor series, which comes at the end of the chapter on sequences and series. In the flurry of computations on sequences and series, it is all to easy to loose sight of the idea that a Taylor polynomial is the "best nth degree polynomial approximation" to a function at a point. A visual approach makes this clear. (If a picture is worth 1000 words, how much is a movie worth?) The visual approach also makes concepts like the interval of convergence easier to understand.

We start with the technical details. Put your cursor in the red section below, and hit the return key. This should load the plots routines.

```
> with(plots):
Warning, the name changecoords has been redefined
```

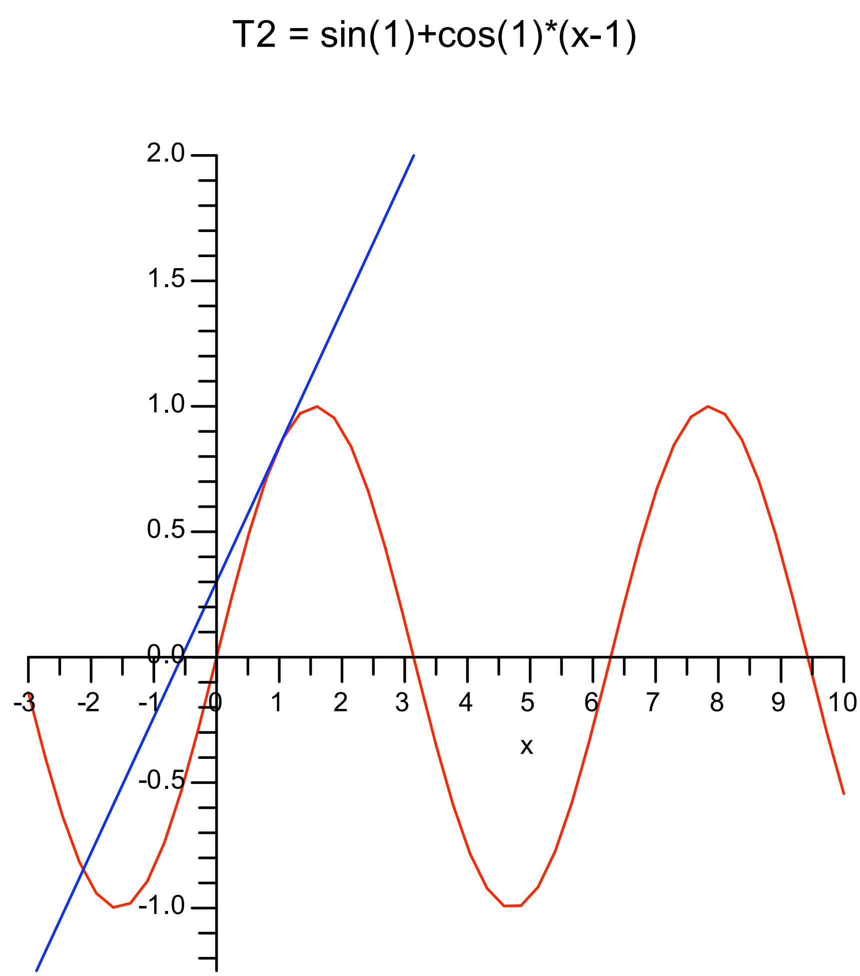
We next view an example. The example we look at is the sin function expanded about the point x=0. We look at the plot of sin(x) plotted on the same axis with the nth degree Taylor polynomial T_n, where n goes from 2 to 20 by 2.

The first example, an animation on a nice sequence of Taylor polynomials

```
> func := sin(x);
minx := -3:  maxx := 10: aboutx := 1:
miny := -1.25: maxy := 2:
mindeg := 2: degsteps := 12: bydeg :=2:
A := display(seq(
plot(convert(taylor(func,x=aboutx, mindeg + bydeg*i), polynom),
x=minx..maxx,y=miny..maxy, color=blue,
title=substring("T"||(mindeg + bydeg*i)||" = "||(convert
(convert(taylor(func,x=aboutx, mindeg + bydeg*i),polynom),
string)),1..60),
titlefont=[HELVETICA, 14]),
i=0..degsteps), insequence=true):
B := animate(func,x=minx..maxx,y=miny..maxy,frames=degsteps+1,
color=red):
print(` `||(convert(func, string))||` vs its Taylor
polynomial`);
display(A,B,view=[minx..maxx,miny..maxy]);
```

$$func := \sin(x)$$

sin(x) vs its Taylor polynomial



To run the animation

To run the animation, click once on the picture above. A box should appear around the picture. If you see the box, you should see a series of buttons, like you might see on a tape recorder. From left to right, the buttons are: back one frame, stop, start, advance one frame; a bow and slider for the current frame; a drop down choice of forward, down and back, or backwards for direction; a drop down choice of stop after one cycleor run in a loop, and a ontrols for the speed. Click the on button and watch the animation.

Taylor polynomials and interval of convergence.

An example with a finite radius of convergence

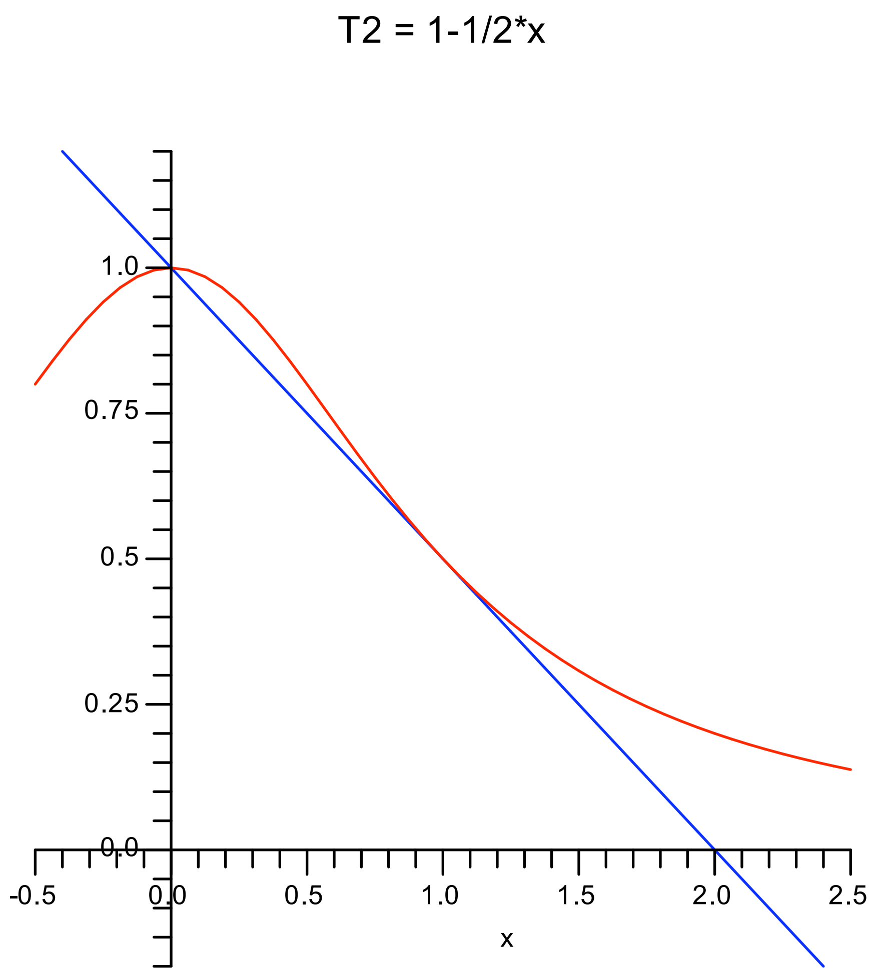
The first animation we looked at has a Taylor series that works everywhere. To make the Taylor polynomial work as an approximation on a bigger interval we simply increased the degree of the polynomial. We now consider a case where the approximation is only good in a small region, no matter how high the degree.

Execute the following animation:

```
> func := 1/(1 + x^2);
minx := -0.5:  maxx := 2.5: aboutx := 1:
miny := -0.2: maxy := 1.2:
mindeg := 2: degsteps := 15: bydeg :=8:
for i from mindeg by bydeg to mindeg + bydeg * degsteps do
T[i] :=convert(taylor(func, x=aboutx, i), polynom): od:
A := display(seq(plot(T[mindeg + bydeg*i],
x=minx..maxx,y=miny..maxy, color=blue,
title=substring("T"||(mindeg + bydeg*i)||" = "||(convert
(convert(taylor(func,x=aboutx, mindeg + bydeg*i),polynom),
string)),1..60),
titlefont=[HELVETICA, 14]),
i=0..degsteps), insequence=true):
B := animate(func,x=minx..maxx,y=miny..maxy,frames=
degsteps+1, color=red):
print(` `||(convert(func, string))||` vs its Taylor
polynomial`);
display(A,B,view=[minx..maxx,miny..maxy]);
```

$$\text{func} := \frac{1}{1+x^2}$$

1/(1+x^2) vs its Taylor polynomial



Notice that the Taylor polynomial approximation of 1/(1 + x^2) centered at x = 1.5 only gives a good approximation for x between 0 and 3, no matter how high the degree.

Final Instructions:

When you complete the worksheet, print it out and hand it in.
(One way to print is to use the "Export as...RTF" option under the file menu. RTF format means it is a Word document.)

>

>