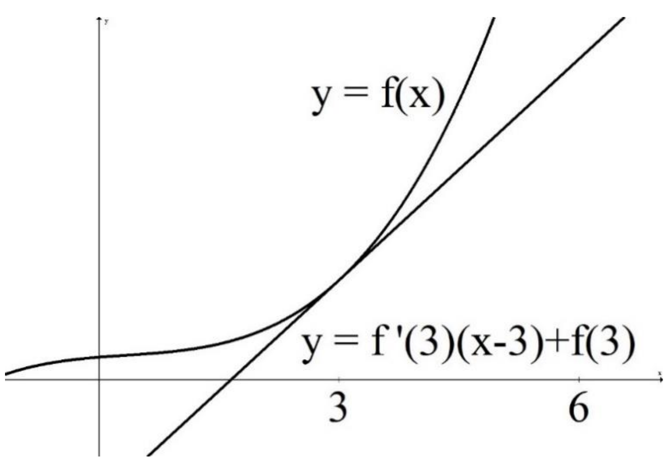
Closing Tue: Taylor Notes 1, 2
Closing Thu: Taylor Notes 3
Final: Sat, June 2, 5:00pm, KANE 130
Eight pages, covers everything.

Taylor Notes 1 (TN 1):
Tangent Line Error Bounds
Goal: Approximate functions with tangent lines and get error bounds.

Def'n: The **first Taylor polynomial for f(x) based at b** is

$$T_1(x) = f(b) + f'(b)(x - b)$$



Entry Task: Find the 1st Taylor polynomial for $f(x) = \sqrt{x}$ at $x = 4$.

- Use it to estimate $\sqrt{4.5}$.
- Use your calculator to find the difference between this estimate and the actual value.

Bounding the Error

Given an interval around $x = b$ (*i.e.* $b - a \leq x \leq b + a$).

Tangent Linear Error Bound Thm

If $|f''(x)| \leq M$ for all *x*, then

$$|f(x) - T_1(x)| \leq \frac{M}{2}|x - b|^2.$$

To use

Step 1: Find $f''(t)$.

Step 2: Find an upper bound (max) for |f''(t)| on the interval. Put this in for M in thm.

Step 3: Plug in x = "an endpoint" to get *worst case* error bound.

Example: Using the theorem, give an error bound for $|\sqrt{x} - T_1(x)|$ based at $x = 4$ on the interval $[3.5, 4.5]$.

Example: $f(x) = \ln(x)$ at $b = 1$.

(a) Find the 1st Taylor polynomial.

(b) Use the error bound formula to find a bound on the error over the interval J = [1/2, 3/2]

(c) Find an interval around b = 1 where the error is less than 0.01.

x	$f(x)$	$T_1(x)$	$\lvert f(x) - T_1(x) \rvert$
1	*0*	*0*	*0*
1.2	*0.1823*	*0.2*	*0.01768*
1.4	*0.3364*	*0.4*	*0.06353*
0.9	*-0.1053*	*-0.1*	*0.00536*

Proof of error bound for x > b:

Start with $f(x) - f(b) = \int_b^x f'(t)dt$.

Do integration by parts,

(with $u = f'(t)$, $dv = dt$,

$du = f''(t)$, $v = t - x$)

to get

$$f(x) - f(b) = f'(b)(x-b) - \int_b^x (t-x)f''(t)dt$$

Rearrange to get

$$f(x) - f(b) - f'(b)(x-b) = \int_b^x (x-t)f''(t)dt$$

Thus,

$$|f(x) - T_1(x)| = \left|\int_b^x (x-t)f''(t)dt\right|$$

Then note

$$\left|\int_b^x (x-t)f''(t)dt\right| \le \int_b^x (x-t)|f''(t)|dt$$

$$\le M \int_b^x (x-t)dt$$

$$\le \frac{M}{2}(x-b)^2.$$

Note about "Bounds":

An upper **bound**, *M*, is a number that is always bigger than the function. The smallest possible upper bound is sometimes called a *tight* bound.

Examples: Find any upper **bound** (if it is easy to do so, find a *tight* upper bound).

1. $|\sin(5x)|$ on [0,2π]

2. $|x-3|$ on [1,5]

3. $\left|\frac{1}{(2-x)^3}\right|$ on [-1,1]

4. $|\sin(x) + \cos(x)|$ on [0,2π]

5. $\left|\cos(2x) + e^{2x} + \frac{6}{x}\right|$ on [1,4]

Example (you do):

Let $f(x) = x^{1/3}$ and $b = 8$.

(a) Find the 1st Taylor Polynomial.

(b) Give a bound on the error over the interval $J = [7,9]$.

(TN 2/ 3): Higher Order Approx.

The **2^{nd} Taylor Polynomial** (or quadratic approximation) is given by

$$T_2(x) = f(b) + f'(b)(x - b) + \frac{1}{2}f''(b)(x - b)^2$$

The **quadratic error bound theorem** (Taylor's inequality) states:
on a given interval [a,b],
if $|f'''(x)| \leq M$, then

$$|f(x) - T_2(x)| \leq \frac{M}{6}|x - b|^3$$

Example:
Find the 2^{nd} Taylor polynomial for $f(x) = x^{1/3}$ based at b = 8 and find the error bound on the interval J = [7,9].

Taylor Approximation Idea:

If two functions have ***all*** the same derivative values, then they are the same function (up to a constant).

To explain, let's compare derivatives of f(x) and T_2(x) at b.

$$T_2(x) = f(b) + f'(b)(x-b) + \tfrac{1}{2} f''(b)(x-b)^2$$

$$T_2'(x) = f'(b) + \frac{1}{2} f''(b) 2(x-b) = f'(b) + f''(b)(x-b)$$

$$T_2''(x) = f''(b)$$

$$T_2'''(x) = 0$$

Now plug in x = b to each of these.

What do you see?

Why did we need a ½ ?

What would T_3(x) look like?

n^{th} Taylor polynomial

$$f(b) + f'(b)(x-b) + \frac{1}{2}f''(b)(x-b)^2 + \frac{1}{3!}f'''(b)(x-b)^3 + \cdots + \frac{1}{n!}f^{(n)}(b)(x-b)^n$$

In sigma notation:

$$T_n(x) = \sum_{k=0}^{n} \frac{1}{k!} f^{(k)}(b)(x-b)^k$$

Taylor's Inequality (error bound):

on a given interval [a,b],

if $|f^{(n+1)}(x)| \leq M$, then

$$|f(x) - T_n(x)| \leq \frac{M}{(n+1)!}|x-b|^{n+1}$$

Side Note:
For a fixed constant, *a*, the expression $\frac{a^k}{k!}$ goes to zero as k goes to infinity.

So the expression $\frac{1}{(n+1)!}|x - b|^{n+1}$, will always go to zero as *n* gets bigger.

Which means that the error goes to zero, unless something unusual is happening with M, which will see in examples later.
