Math 164: Week 2A

Fall 2024

Course announcements

- Lecture notes and Python code will be shared at https://github.com/chrismkkim/Howard-Math164
- Install Python on your personal computer. Homework and exam problems will involve coding in Python
- Install Jupyter notebook (interactive coding environment) to run the Python code covered in class

2.1: Evaluating the value of a function

How do we numerically evaluate functions that are not polynomials. In other words, a function is NOT in the form

$$f(x) = a_0 + a_1 x + ... a_n x^n \tag{1}$$

Examples of non-polynomial functions are: $sin(x), cos(x), e^x$

The exact values of these functions are known at some x's
$sin(0) = 0, cos(0) = 1, sin(\pi/2) = 1, cos(\pi/2) = 0$

How can we evaluate them at arbitrary x's?
$sin(0.1), cos(0.1)$?

Taylor's Theorem

A smooth function $f(x)$ can be approximated around x_0 by

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + ... + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \tag{2}$$

$$+ \frac{f^{(n+1)}(s)}{(n+1)!}(x - x_0)^{n+1} \tag{3}$$

Left hand side: Non-polynomial
Right hand side: Polynomial in x if we can evaluate $f'(x_0), f''(x_0)...$

Taylor's theorem approximates a non-polynomial $f(x)$ with a polynomial.

- $h = x - x_0$ is a small number.
- s is a number between x_0 and $x = x_0 + h$.

- n-th order Taylor polynomial.

$$T_n(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + ... + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \tag{4}$$

x_0 is selected so that the derivatives $f'(x_0), f''(x_0)$... are known. Then, we don't have to evaluate the non-polynomial functions.

- Remainder

$$R_n(x) = \frac{f^{(n+1)}(s)}{(n+1)!}(x - x_0)^{n+1} \tag{5}$$

This is not a polynomial in x because s depends on x. The non-polynomial function $f^{(n+1)}(s)$ has to be evaluated at arbitrary s.

Algorithm for evaluating non-polynomial functions

1. Use the n-th order Taylor polynomial to evaluate the function (it's an approximation)

$$T_n(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + ... + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \tag{6}$$

2. Use the remainder term to evaluate the error due to the approximation.

$$R_n(x) = \frac{f^{(n+1)}(s)}{(n+1)!}(x - x_0)^{n+1} \tag{7}$$

An important issue to consider

- How do we determine n, the number of terms to include in $T_n(x)$?
- Choose n so that $T_n(x)$ is a good approximation of $f(x)$.
- The remainder should be smaller than a small number ϵ.

$$|R_n(x)| = |f(x) - T_n(x)| < \epsilon$$

Typically, $R_n(x)$ becomes smaller as n gets larger. So, find a large enough n so that $R_n(x)$ is smaller than ϵ.

A worked-out example

Evaluate $\cos(0.15)$ with error smaller than $\epsilon = 10^{-5}$.

Find a Taylor approximation of $\cos(x), x = 0.15$ around $x_0 = 0$.

Evaluate the derivatives of $\cos(x_0)$ and use them in the Taylor approximation.

$\cos' x = -\sin x,\ \cos'' x = -\cos x,\ \cos''' x = \sin x,\ \cos'''' x = \cos x$

$\cos' x_0 = 0,\ \cos'' x_0 = -1,\ \cos''' x_0 = 0,\ \cos'''' x_0 = 1$

Taylor approximation of $\cos(x)$ at $x_0 = 0$ is:

$T_n(x) = 1 - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 - \frac{1}{6!}x^6 + ... + (-1)^n \frac{1}{(2n)!}x^{2n}$

Remainder term is

$R_n(x) = \frac{\cos^{(n+1)}(s)}{(n+1)!}x^{(n+1)}$

How large is the error of Taylor approximation? Estimate how large the remainder term is.

See if we can control the absolute value of $R_n(x)$

$$|R_n(x)| = \frac{|\cos^{(n+1)}(s)|}{(n+1)!}|x|^{(n+1)}$$

Notice that the derivatives of $\cos(x)$ are in the form of $\pm\sin(x), \pm\cos(x)$. This means that

$$|\cos^{(n+1)}(s)| \leq 1$$

Hence,

$$|R_n(x)| = \frac{1}{(n+1)!}|x|^{(n+1)}$$

Now, evaluate $R_n(x)$ for different values of n in Python .