Lecture 1

Note Title 2018-05-06

Computer Arithmetic

We often want to work with the real number system which consists of all integers, rational and irrational numbers

In a computer we have finite storage for numbers

$\Rightarrow$ not all real numbers can be represented exactly

This can cause problems with arithmetic.

We typically use base 10 decimal system

Computers often use the binary (base 2) system

$(1001.11101)_2 =$

Verify (not to be handed in)

$(1001.11101)_2 = (9.90625)_{10}$

Note that the

$(\quad)_{10} \Leftrightarrow (\quad)_2$

process can lead to errors!

<u>Ex</u> What is $\left(\frac{1}{10}\right)_{10}$ in $(\quad)_2$?

We have

$$\left(\frac{1}{10}\right)_{10} = \left(0.0001\ 1001\ 1001 \ldots\right)_2$$

Since computers have a finite storage the number on the right cannot be stored exactly.

The decimal has to be truncated somehow ...

Hypothetical storage scheme (32 bit)

Normalization

Can write all real numbers in normalized scientific notation

eg $732.5051 =$

$-0.005612 =$

if $x \in \mathbb{R}$ then

How do we truncate a real number to fit our storage mechanism?

<u>Rounding</u> or <u>Chopping</u>

Suppose $x = .a_1 a_2 \cdots a_n a_{n+1} \cdots a_m$ using m digits.

We <u>round</u> to n decimal places by looking at a_{n+1}.

If $a_{n+1} = 0, 1, 2, 3$ or 4

then $x = .a_1 a_2 \ldots a_n$

(after rounding)

If $a_{n+1} = 5, 6, 7, 8, 9$

then $x = .a_1 \; a_2 \cdots (a_n + 1)$

(after rounding)

last digit increased by 1

OR we could chop and simply discard

$$a_{n+1} \quad a_{n+2} \quad \cdots \quad a_m$$

so $x = .a_1 a_2 \cdots a_n$

To quantify error we have

Absolute error

Relative error

to measure the error in an approximation p^* to p.

Significant Digits

p^* approximates p to t significant digits if the relative error is less than

$$5 \times 10^{-t}$$

ie t is the largest integer so that $\frac{|p-p^*|}{|p|} < 5 \times 10^{-t}$.

Floating point arithmetic

Let $fl(x)$ denote the machine representation of x.

If we want to compute $x+y$ on a computer, then the computer returns

Cancellation error

(subtracting nearly equal numbers)

Consider

$fl(x) = 0.d_1 d_1 \cdots d_p \alpha_{p+1} \alpha_{p+2} \cdots \alpha_K \times 10^n$

$fl(y) = 0.d_1 d_2 \cdots d_p \beta_{p+1} \beta_{p+2} \cdots \beta_K \times 10^n$

and $x > y$

We have $fl(fl(x) - fl(y))$

$$= 0.\sigma_{p+1} \sigma_{p+2} \cdots \sigma_K \times 10^{n-p}$$

where

$$0.\sigma_{p+1} \sigma_{p+2} \cdots \sigma_K$$
$$= 0.\alpha_{p+1} \alpha_{p+2} \cdots \alpha_K$$
$$- 0.\beta_{p+1} \beta_{p+2} \cdots \beta_K$$

Example

$p = 0.54617$

$q = 0.54601$

Exact value $r = p - q = 0.00016$

But now with 4 digit rounding.

Example: Consider $f(x) = \dfrac{1-\cos x}{x^2}$

Let $\bar{x} = 1.2 \times 10^{-5}$. Then

$c = fl(\cos(\bar{x})) =$

(rounded to 10 digits)

and $\dfrac{1-c}{\bar{x}^2} =$

Another way to reduce the round off error is to reduce the number of floating point operations

<u>EX</u>. Polynomial evaluation using nested multiplication

$$f(z) = 1.01z^4 - 4.62z^3 - 3.11z^2 + 12.2z - 1.99$$

Example. Solve for x: $ax^2+bx+c=0$

$$x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}, \quad x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$$

Say $b=600$, $a=c=1$.

- What could go wrong?
- How could we reformulate the problem?
- What should we do if b was -600?

Taylor Series (Review)

Taylor's theorem is one of the most important tools for this course.

Assuming a function is sufficiently smooth on an interval $[a, b]$, then we can construct a polynomial approximation $P_n(x)$ to $f(x)$ as follows:

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

where $x, x_0 \in [a, b]$.

How smooth does $f(x)$ have to be?

- $f \in C^n[a, b] \Rightarrow f, f', f'', \ldots, f^{(n)}$ all continuous
- $f^{(n+1)}$ must exist on $[a, b]$

Q How good is this approximation? What error is made?

We have $f(x) = P_n(x) + \text{Error}$
$= P_n(x) + R_n(x)$

Taylor's thm says

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

where ξ is between x_0 & x.

Note: Approximation will be best where x is close to x_0

x_0 is the known expansion point.

Example Find the third Taylor polynomial $P_3(x)$ for $f(x) = \sin(x)$ with $x_0 = 0$.

Soln. $P_3(x) = f(0) + f'(0)(x-0) + \frac{f''(0)}{2}(x-0)^2 + \frac{f'''(0)}{3!}(x-0)^3$

We need f', f'', f'''

$f'(x) = \cos x \Rightarrow f'(0) = \cos(0) = 1$
$f''(x) = -\sin x \Rightarrow f''(0) = -\sin(0) = 0$
$f'''(x) = -\cos x \Rightarrow f'''(0) = -\cos(0) = -1.$

Also $f(0) = 0$.

$\Longrightarrow P_3(x) = x - x^3/6.$

What about the error?

$$R_3(x) = \frac{f''''(\xi)}{4!} x^4$$

$f''''(x) = \sin x$

$$\Longrightarrow R_3(x) = \frac{\sin \xi}{24} x^4$$

Another useful form of Taylor's THM

We have

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2}(x-x_0)^2 + \cdots$$

Let $x = x_0 + h$. Then

$$f(x_0+h) = f(x_0) + f'(x_0)(x_0+h-x_0) + \frac{f''(x_0)}{2}(x_0+h-x_0)^2 + \cdots$$

$$\Rightarrow f(x_0+h) = f(x_0) + f'(x_0)h + f''(x_0)\frac{h^2}{2} + \cdots$$

Ex Use P_1 (linear approximation) to find an approximation to $\sqrt{16.1}$.

Soln.

Throughout this course, we will study numerical methods that solve a problem by constructing a sequence of (hopefully) better ~~&~~ better approximations which converge to the required soln.

A technique is needed to compare the convergence rates of different methods.

Assume the sequence $\{\alpha_n\}$ converges to α:

$$\lim_{n\to\infty} \alpha_n = \alpha$$

We would like to quantify how quickly α_n tends to α.

Consider an example

$$\alpha_n = \sin\left(\frac{1}{n}\right)$$

We have $\alpha_n \to 0$ as $n \to \infty$
so $\alpha = 0$.

Note that $\lim_{n\to\infty} \sin\left(\frac{1}{n}\right)$

is equivalent to

$$\lim_{h\to 0} \sin(h)$$

We will work with the latter
(avoids working with "∞").

Expand $\sin h$ in a Taylor series in powers of h:

We now generalize the underlying concepts.

Consider the following definition:

Suppose $\{\alpha_n\}$ is a sequence that converges to α as $n \to \infty$:

$$\lim_{n \to \infty} \alpha_n = \alpha$$

And assume $\{\beta_n\}$ is a sequence that converges to zero as $n \to \infty$

$$\lim_{n \to \infty} \beta_n = 0$$

If $$|\alpha_n - \alpha| \leq K\,|\beta_n|$$

for n large, where K is a positive constant then we say

$\{\alpha_n\}$ converges to α

with RATE OF CONVERGENCE

$$O(\beta_n) \qquad \text{(big - Oh of } \beta_n\text{)}$$

If $\alpha_n \to \alpha$ with r.o.c. $O(\beta_n)$ then we sometimes write

$$\alpha_n = \alpha + O(\beta_n)$$

In our previous example, we had the sequence

$$\alpha_n = \sin\left(\frac{1}{n}\right)$$

$$\alpha =$$

Note: Usually we compare how fast $\alpha_n \to \alpha$ with how fast

$$\beta_n = \frac{1}{n^p} \longrightarrow 0$$

We are most interested in finding the largest value of p for which

$$\{\alpha_n\} \to \alpha$$

with r.o.c. $O\left(\frac{1}{n^p}\right)$

Another example: $\lim_{n\to\infty} n \sin\left(\frac{1}{n}\right) = 1$

I expanded around $h_0 = 0$. Why?
We want to know what happens as $h \to 0$ so it makes sense to choose $h_0 = 0$.

What about the higher order term?

$$\frac{1}{h} \sin h - 1 = -\frac{h^2}{6} + ch^4 + \ldots$$

Why do we say $O(h^2)$ convergence?

As $h \to 0$, the h^2 terms dominate the convergence since the h^4 term "will be gone to zero" long before the h^2 term.

So the rate of convergence can only be as fast as the h^2 term.