Lecture 4 Review

- Representation of single & double precision real numbers.
- Either representation has a finite precision.
- Code to determine machine precision for single precision numbers.
- Example of variable declaration (single precision real).
- Example of for loop.
- Simple example of a “here document” in shell scripting.

http://www.cplusplus.com/doc/tutorial/variables.html
http://www.cplusplus.com/doc/tutorial/control.html
NUMERICAL INTEGRATION

NEED TO PERFORM \((\text{NUMERICAL})\) INTEGRATION

From \text{TIME TO TIME}.

\[ \int_{a}^{b} f(x) \, dx = A \]

\text{BASIC TECHNIQUE IS STRAIGHT FORWARD}

\[
\int_{a}^{b} f(x) \, dx = \lim_{h \to 0} \left[ h \sum_{i=1}^{(b-a)/h} f(x_i) \right]
\]
Many different integration techniques. Basic idea is to "sum over boxes" more abstractly.

\[ \int_{a}^{b} f(x) \, dx \approx \sum_{i=1}^{N} f(x_i) \omega_i. \]

"Weights"

Actually, no real need to have boxes all of same width; width can vary.

However, let's start simply.

\[ h = \frac{b-a}{N-1} \]

Box widths all equal.

\[ x_i = a + (i-1)h, \quad i = 1, N \]

We construct trapezoids.
Consider area of single trapezoid

\[ \sum_{i=1}^{N} f(x_i) dx \approx h \left( \frac{f_i + f_{i+1}}{2} \right) = \frac{1}{2} h f_i + \frac{1}{2} h f_{i+1} \]

In terms of our std integration formula

\[ \sum_{i=1}^{N} f(x_i) dx \approx \sum_{i=1}^{N} f(x_i) \omega_i , \quad \omega_i = \frac{1}{2} \]

For the full range \([a, b] \]

Add contributions from each sub-interval

\[ \int_{a}^{b} f(x) dx \approx \frac{1}{2} f_1 + \frac{1}{2} f_2 + \frac{1}{2} f_3 + \cdots + \frac{1}{2} f_N \]

Note: Each interval point counts twice

Hence, \( \omega_i = \{ \frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2} \} \)

Main advantage: Easy to understand

Easy to code

\[ N \text{ must be odd} \]

Don't Forget
Slightly fancier technique: Simpson’s Rule

Tries to account for curvature of f(x)
For small intervals f(x) \approx \Delta x^2 + \beta x + \gamma

Still keep intervals equally spaced.

Will evaluate integral over adjacent intervals (as before).

N must be odd.

Area of each interval is integral of parabola

\[ \int_{x_i}^{x_{i+1}} (ax^2 + \beta x + \gamma) \, dx = \frac{a}{3} x_i^3 + \frac{\beta}{2} x_i^2 + \gamma x_i \]

Now, consider interval [-1, 1]

\[ \int_{-1}^{1} (ax^2 + \beta x + \gamma) \, dx = \frac{2a}{3} + 2\gamma \]
Also notice $f(-1) = 2 - \beta + \gamma$ $f(0) = \gamma$ $f(1) = 2 + \beta + \gamma$

$\Rightarrow x = \frac{f(1) + f(-1) - f(0)}{2}$

$\beta = \frac{f(1) + f(-1)}{2}$

$\gamma = f(0)$

So,

$\int_1^0 (x^2 + \beta x + \gamma)dx = \frac{f(-1)}{3} + \frac{4f(0)}{3} + \frac{f(1)}{3}$

Expand to consider 2 adjacent intervals.

$\int_{x_i}^{x_i+h} f(x)dx = \int_{x_i}^{x_i+h} f(x)dx + \int_{x_i}^{x_i} f(x)dx \approx \frac{h}{3} f_i + \frac{4h}{3} f_i + \frac{h}{3} f_{i+1}$

Don't Forget
So...

\[ \int_a^b f(x) \, dx = \frac{h}{3} f_1 + \frac{4h}{3} f_2 + \frac{2h}{3} f_3 + \frac{4h}{3} f_4 + \frac{2h}{3} f_5 + \ldots + \frac{4h}{3} f_{N-1} + \frac{h}{3} f_N \]

In terms of our weight formula:

\[ \int_a^b f(x) \, dx = \sum_{c=1}^{N} f(x_c) \, w_c \]

\[ w_c = \left\{ \frac{h}{3}, \frac{4h}{3}, \frac{2h}{3}, \frac{4h}{3}, \frac{2h}{3}, \ldots, \frac{4h}{3}, \frac{h}{3} \right\} \]

\( N \) weights, \( N = \text{odd} \)

BTW, easy to show:

\[ \sum_{c=1}^{N} w_c = (N-1) h \]
Simpson’s Rule

Display simpson’s rule source code (too large for 1 slide).

http://www.physics.smu.edu/devel/coan/3340/simpson.cc
Monte Carlo Integration

Throw darts at dart board

Probability of landing in circle proportional to area of circle

But area = integral under curve

Consider upper quadrant $0 \leq x \leq 1, 0 \leq y \leq 1$

Area in quad = $\frac{1}{4} \int_0^1 \int_0^1 dy \, dx$
\[ A = \int_0^1 \sqrt{1-x^2} \, dx \]

"Throw a dart" means pick random number. Random \( \leq \text{can't be predicted.} \)
Practically, done by function call.

Pick random \( x \) \( (0 \leq x \leq 1) \)

Pick random \( y \) \( (0 \leq y \leq 1) \)

If \( y \leq \sqrt{1-x^2} \), then dart in quadrant.

Repeat many times, say, \( N \)
Calculate how many darts \( -\text{circle} \)

\[ A = \left( \frac{N_o}{N_T} \right) \times \text{area of square} \]
// computes some random numbers

#include <iostream>

using namespace std;

int main()
{
    srand((time(0))); // "seed" the random number generator
    int r = random();

    // RAND_MAX is largest random number. it is an integer. built-in.

    cout << " random number: " << (double) random()/RAND_MAX << endl;
    cout << " random number: " << random() << endl;
    cout << " random number: " << random() << endl;

    cout << "The value of RAND_MAX is " << RAND_MAX << endl;
    return 0;
}
My head is exploding.

I need something to read quietly, at my own pace.

Link also available from PHYS 3340 links page
Summary

- Trapezoid and Simpson’s rule for numerical integration.
- Monte Carlo (“dart throwing”) technique for integration.
- Code for random number generation.
- C++ features: if, declaring functions, random, ...
- Just a peek at plotting: gnuplot.

Don’t suffer in silence. Scream for help!!!