Monday 7:30-9:20 in 2076/2082

Handouts: PS#1, Landau/Paez Ch. 4 excerpts, Session #4

Gameplan: Announcements, recaps, and brief overview,
Then finish up the Session 2 activities.

Announcements:
- We have a TA: Dr. Sung Yong Park, who works with David Stroud. Wide experience in computational physics. He'll help out in the 2076/2082 sessions.
- PS#1 is due next Monday (or Tuesday if holiday).
- Mostly a check that you're up to speed on the basic tools. The problem is on page 33 of Landau/Paez
- handouts: compute \( \frac{1}{n} \) and \( \frac{2}{n} \) on a computer.
- There will be hints available from the web page.
- Use the examples given, email for help.
- Thursday night is our session here for questions:
- Lecture notes available online from web page.
- On Wednesday we'll recap then start numerical integration
- as a useful example of error analysis.

Quickies: (questions?)
- Unix directories: 
  
  - \$ ls ...
  - \$ cd session_4 Session_2

- Create area with
  - \$ make-area
  - \$ make

- Given C routines from C++ => see comments in GSL programs
Makefiles - we'll see more examples. Why use them?
  1. changing options
  2. many files together
  
  => look at make-dft-trap-menu

  VARIABLE = stuff -> $(VARIABLE) substitutes

  -bash -lgsl
  
  -lgsl
  
  (gsl library code (like o files)
  
  libgsl.a <= arcade)

  Output with cout => look at quadratic-equation-2.cpp
  
  #include <iostream>
  
  manipulate stream

  cout << rel_error << endl;    // first time
  cout << scientific << rel_error << endl; // now in scientific (default)
  cout << scientific << setprecision(16)
      << rel_error << endl;    // 16 digits
  cout << fixed << setprecision(8) << setw(8) << rel_error << endl;
  
  to get things to line up.

  To output to a file: just like cout but [my-file.out]
  
  #include <fstream>
  
  ofstream my-out ("my-file.out");
  my-out << rel_error << endl;
  
  and so on
Recap of simple calculation: $\epsilon < \epsilon_m < 6 \times 10^{-8}$ in single precision

What's wrong with this argument? (It's easy to go wrong!)

$\epsilon = 5 \times 10^{-8} (1 + \epsilon_m)$

$\Rightarrow 10^7 \times \epsilon = 5 \times 10^1 (1 + \epsilon_m)$

$\Rightarrow 10^3 \times \epsilon + 1 = 1.5 \times 10^6$

but actually it is 1.53. Why? (Discussion not time.)

- Run demo, use make demo, a guess: check everything.
- Indent ad print code, do this first, empirically.
- Guess what $10^8$ will do.

What other ways can errors accumulate?

**Multiplication error:**

$Z_3 = Z_2 \times Z_3 \Rightarrow Z_3 = Z_2 (1 + \epsilon_3) = Z_2 (1 + \epsilon_2) (1 + \epsilon_3)$

$\Rightarrow \epsilon_3 = \epsilon_2 + \epsilon_3 \Rightarrow$ add up errors

But: $|\epsilon| \leq \epsilon_m$ with sign under! $\epsilon_3$ could be greater or less than $\epsilon_2$.

Often: uncorrelated errors that are more-or-less random.

$\Rightarrow \epsilon \equiv \epsilon_m \text{ when } -1 \leq s \leq 1$

with $<s^2> = 0$ and $<s^3> = 1$.

Possible fits:

```
-2 2
```

or

```
-4 4
```

How can you check?

Now $N$ multiplications:

$\epsilon_{\text{total}} = \epsilon_1 + \epsilon_2 + \ldots + \epsilon_N$

$\Rightarrow \epsilon_{\text{total}} \leq \epsilon_m \left( s_1 + s_2 + \ldots + s_N \right)$

but consider:

$\epsilon_{\text{total}} = \epsilon_1 + \epsilon_2 + \ldots$

$\Rightarrow \epsilon_{\text{total}} \leq \epsilon_m \left( s_1 + s_2 + s_3 + \ldots \right)$

$\Rightarrow \epsilon_{\text{total}} \leq \epsilon_m \left( s_1 + s_2 + s_3 + \ldots \right)$

$\Rightarrow \epsilon_{\text{total}} \leq \epsilon_m \left( s_1 + s_2 + s_3 + \ldots \right)$

$\Rightarrow \epsilon_{\text{total}} \leq \epsilon_m \left( s_1 + s_2 + s_3 + \ldots \right)$

The formula works.
Next time: integration approximation error $E_{\text{approx}} = \frac{x}{N^2}$, possibly:

\[ E_{\text{total}} = \frac{x}{N^2} + \sqrt{\frac{N}{N}} \]

\[ \Rightarrow \text{optimum } N \]

Test it! Thread:

Challenge: are errors really random? (random in

Session 2: errors calculating spherical Bessel functions

\[ e^{k \rho^2} = \sum_{l=0}^{\infty} (2l+1) j_l(k \rho) P_l(\cos \phi) \]

\[ J_l(x) \xrightarrow{\text{large } l} x^{l+1} \text{ for } x \ll l \]

\[ J_l(x) \xrightarrow{\text{large } x} \frac{\sin(x-x_0)}{x} \text{ for } x \gg l \]

or:

\[ J_{l+1}(x) = \frac{x^{l+2}}{x} j_l(x) - J_{l-1}(x) \text{ (up)} \]

\[ J_{l-1}(x) = \frac{x^{l-1}}{x} j_l(x) - J_{l+1}(x) \text{ (down)} \]

Up: fix $x$, start with \( J_0(x) = \frac{\sin x}{x} \), \( J_1(x) = \frac{\sin(x-x_0)}{x^3} \) \( \Rightarrow J_{2n+1} \Rightarrow J_{2n+3} \Rightarrow J_0(x) \)

Down: start with only $J_{\text{max}}(x)$, $J_{\text{max}+2}(x)$ make up

\[ J_{\text{max}}(x) \xrightarrow{\text{maximize } x_0} J_{\text{max}+2}(x) \Rightarrow J_0(x) \]

Then rescale to known $J_0(x)$.

Why is one way better or worse?

Hint: how many solutions to diff eq?

Key: mix of good and bad solutions $\Rightarrow$ go in direction that bad piece shrinks