Session 1 follow-ups

Types of error

Reminder of power laws
Session 1 follow-ups

- **Session 1 guides returned with “✓” if satisfactory**
  - “−” means something to fix/finish and then turn in for upgrade
  - “+” means “above and beyond” in your answers

- **See Session 2 notes for more itemized Session 1 follow-ups**

- **Highlighted comments**
  - wide range of C++ backgrounds (from none to most)
    - procedural: focus on formula ($area = \pi \times radius^2$)
    - object oriented: focus on circle and things about it
    - here: classes for data hiding, encapsulation, extensions (cf. functions and libraries)
  - learn to use an editor efficiently: e.g., globally change float to double; show line numbers, . . . (ask if you don’t know how)
  - `using namespace std;` instead of `std::cin`
    - like the last name used to distinguish students named Jane
  - variable choice: loop indices `i, j, k` are fine since local; use self-documenting variable names, but not crazy-long
Session 1 follow-ups (cont.)

More highlighted comments

Verification: how do you detect i) \( A = 3r^2 \), ii) \( A = \pi r \) errors?

Python: same pseudocode!

Key consequences of limited total of machine numbers

- there are maximum and minimum numbers (overflow and underflow)
- \( z_c = z(1 + \epsilon) \) with \(-\epsilon_m < \epsilon < \epsilon_m\)
- you only determined it to 1–2 digits; don’t give more!

How do you explain 1.00 \( \cdots \) 011921 when you look for the machine precision? Hint: what is \( 2^{-23} \)?

Pitfalls of makefiles: extra spaces. Why use one?

\[ \Rightarrow \text{recipe for constructing the code (compiler switches, multiple files, machine dependencies, \ldots)} \]

GSL: use libraries when possible. Later: LAPACK

Questions?

Plan: finish marked Session 1 tasks then move to Session 2
Programming notes (that you need to know)

- **Quick check:** `float x = 1/2; // never do this!` cout `<<` x `<<` endl; // what is printed?
  - The expression with `<<`'s is called a “stream”. (More later!)

- Suppose `int eps=1;`
  - `eps++` ➞ add 1 to `eps` ➞ `eps=2` (C++, get it?)
  - `eps+=5` ➞ add 5 to `eps` ➞ `eps=6`
  - `eps*=3` ➞ multiply `eps` by 3 ➞ `eps=3`
  - `eps/=2` ➞ add 1 to `eps` ➞ `eps=0` (0.5 if float)

- `endl, setprecision` are called “manipulators” (more later)
  - control output (new line, how many digits, etc.)
  - in contrast to C-style formatting with `printf`

```cpp
double answer = sqrt(2.);
cout << "The answer is " << fixed << setprecision(4) << answer << " miles." << endl;
printf("The answer is %.4f miles.\n", answer);
```

Both yield: The answer is 1.4142 miles.
Types of error (see Session 2 notes for more)

1. “Blunders” ➔ use compiler to catch typos, omissions, etc.
   - be careful of mixing up variable1 and variable2

2. Round-off errors ➔ because finite (relative) precision
   - $1 + \epsilon_m = 1$ so we lose digits
   - $\epsilon_m$ is the maximum relative error in representing floating-point numbers
   - don’t confuse smallest number with machine precision
   - 7 digits in single precision; 15-16 digits in double precision
   - worst case is “subtractive cancellation” (e.g., in derivative):
     \[
     \begin{array}{c}
     1.2345 \\
     +1.2344 \\
     \hline
     2.4689
     \end{array}
     \Rightarrow 5 \text{ digits}
     \begin{array}{c}
     1.2345 \\
     -1.2344 \\
     \hline
     0.0001
     \end{array}
     \Rightarrow \text{only 1 digit!}
     \]

3. Approximation errors (next time)
   - example: $e^x \approx 1 + x + x^2/2$ for small $x$
   - approximation (or truncation) error $\approx x^3/3!$ (1st omitted term)
Power laws: two ways to plot

If \( y = Cx^\alpha \), where \( \alpha = 1, 2, \ldots \) or \( -1 \) or \( 1/2 \) or \( -1/2 \) or \( \ldots \)

then \( \log_{10} y = \log_{10} Cx^\alpha = \log_{10} C + \alpha \log_{10} x \)

or \( Z = D + \alpha W \)
Error plots in arXiv:1312.6876

“Systematic expansion for infrared oscillator basis extrapolations”

by R.J. Furnstahl, S.N. More, T. Papenbrock

[Sushant took 6810 in 2013]

First check against solvable model problem (square well on left)

Ask: does it work as well as it should? ➔ Are slopes correct?
Follow-ups  Errors  Power law

**Error plots in arXiv:1312.6876**

“Systematic expansion for infrared oscillator basis extrapolations”

by R.J. Furnstahl, S.N. More, T. Papenbrock  [Sushant took 6810 in 2013]

First check against solvable model problem (square well on left)

Ask: does it work as well as it should? \(\implies\) Are slopes correct?

Dick Furnstahl  6810: Session 2