Computational Physics (6810): Session 2

Dick Furnstahl
Nuclear Theory Group
OSU Physics Department

January 13, 2017

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 $1.00\cdots011921$ when you look for the machine precision? Hint: what is 2^{-23}?
  - Pitfalls of makefiles: extra spaces. Why use one? $\Longrightarrow$ recipe for constructing the code (compiler switches, multiple files, machine dependencies, . . .)
  - GSL: use *libraries* when possible. Later: LAPACK, Armadillo
- *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++` $\Longrightarrow$ add 1 to `eps` $\Longrightarrow$ `eps=2` (C++, get it?)
  - `eps+=5` $\Longrightarrow$ add 5 to `eps` $\Longrightarrow$ `eps=6`
  - `eps*=3` $\Longrightarrow$ multiply `eps` by 3 $\Longrightarrow$ `eps=3`
  - `eps/=2` $\Longrightarrow$ divide `eps` by 2 $\Longrightarrow$ `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`

```
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" $\Longrightarrow$ use *compiler* to catch typos, omissions, etc.
   - be careful of mixing up `variable1` and `variable2`
2. Round-off errors $\Longrightarrow$ because finite (relative) precision
   - $1 + \epsilon_m = 1$ so we lose digits
   - ϵ_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):

$$\begin{array}{r} 1.2345 \\ +1.2344 \\ \hline 2.4689 \end{array} \Longrightarrow \text{5 digits} \qquad \begin{array}{r} 1.2345 \\ -1.2344 \\ \hline 0.0001 \end{array} \Longrightarrow \text{only 1 digit!}$$

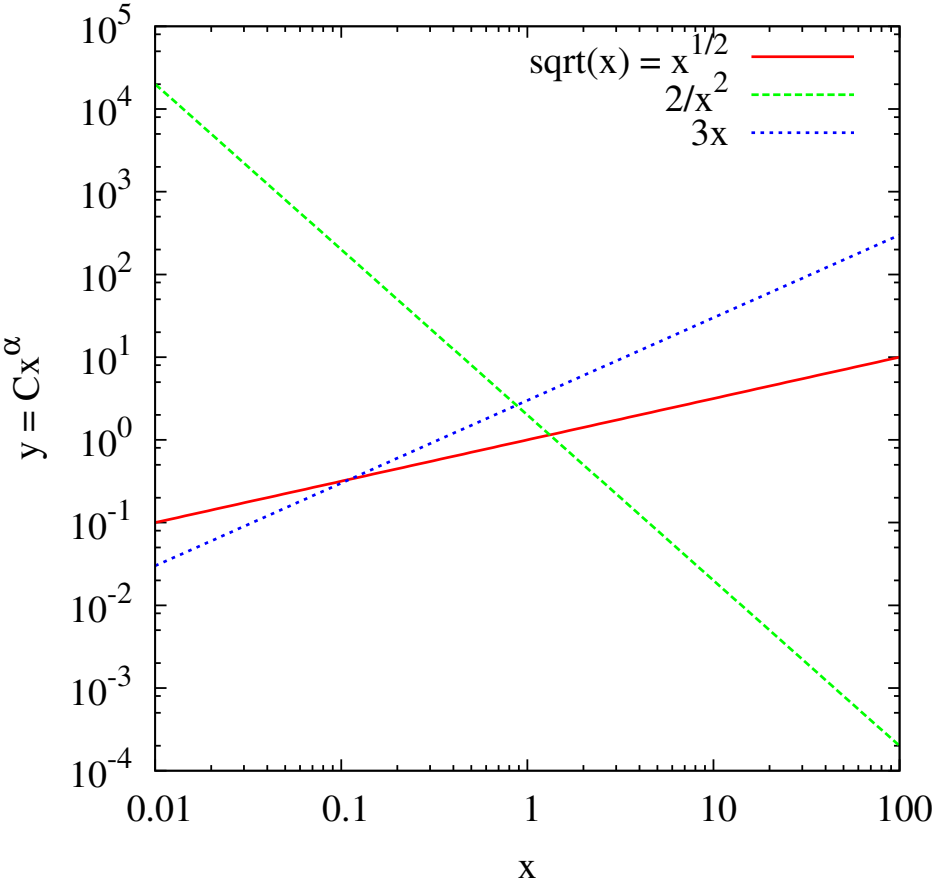
3. Approximation errors (next time)
   - example: $e^x \approx 1 + x + x^2/2$ for small x
   - approximation (or truncation) error $\approx x^3/3!$ (1st omitted term)

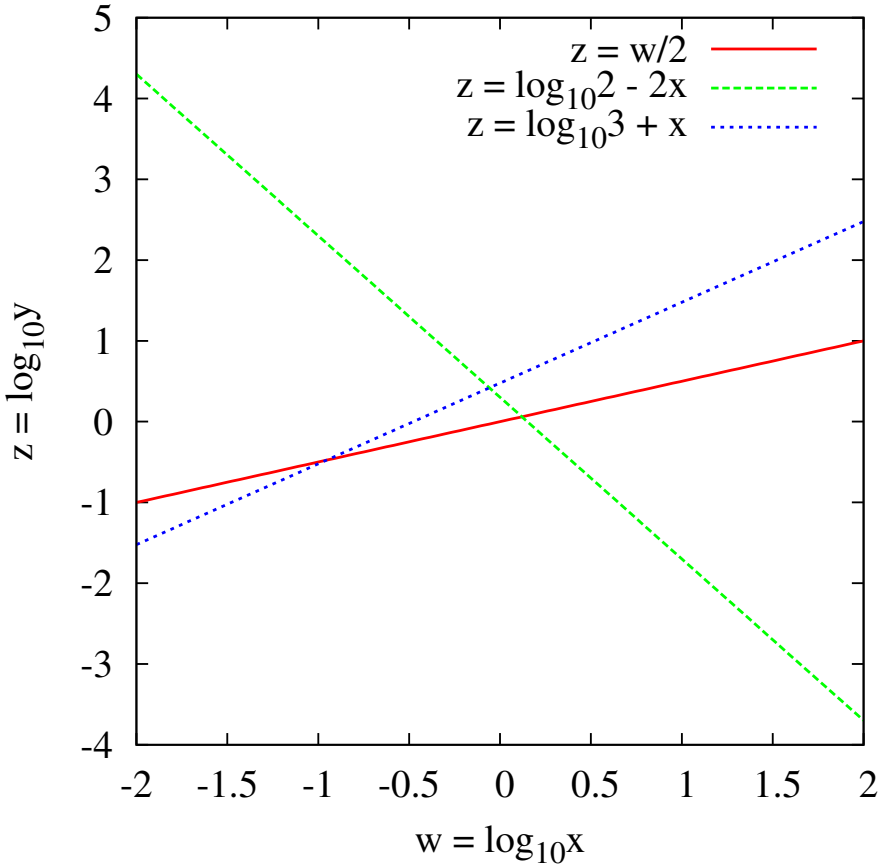
Power laws: two ways to plot

If $y = Cx^{\alpha}$, where $\alpha = 1, 2, \ldots$ or -1 or $1/2$ or $-1/2$ or $\ldots$

then $\underbrace{\log_{10} y}_{z} = \log_{10} Cx^{\alpha} = \underbrace{\log_{10} C}_{D} + \alpha \underbrace{\log_{10} x}_{w}$

or $Z = D + \alpha w$

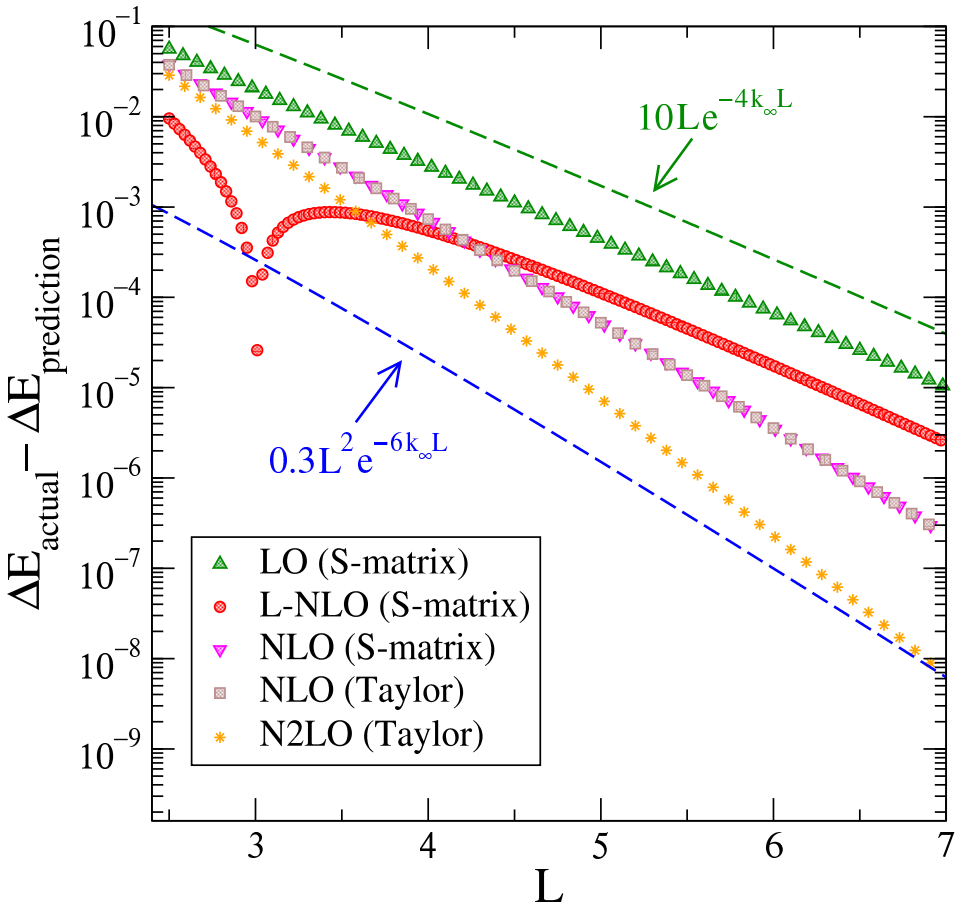


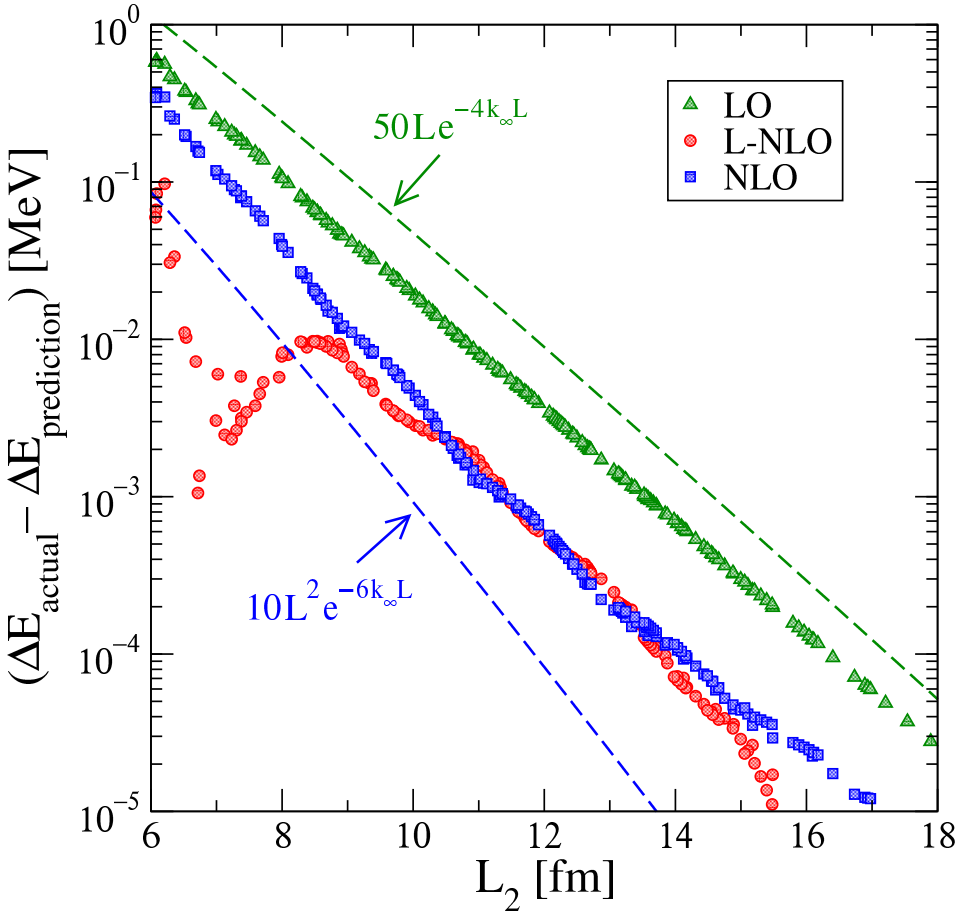


Error plots in arXiv:1312.6876

"Systematic expansion for infrared oscillator basis extrapolations"

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



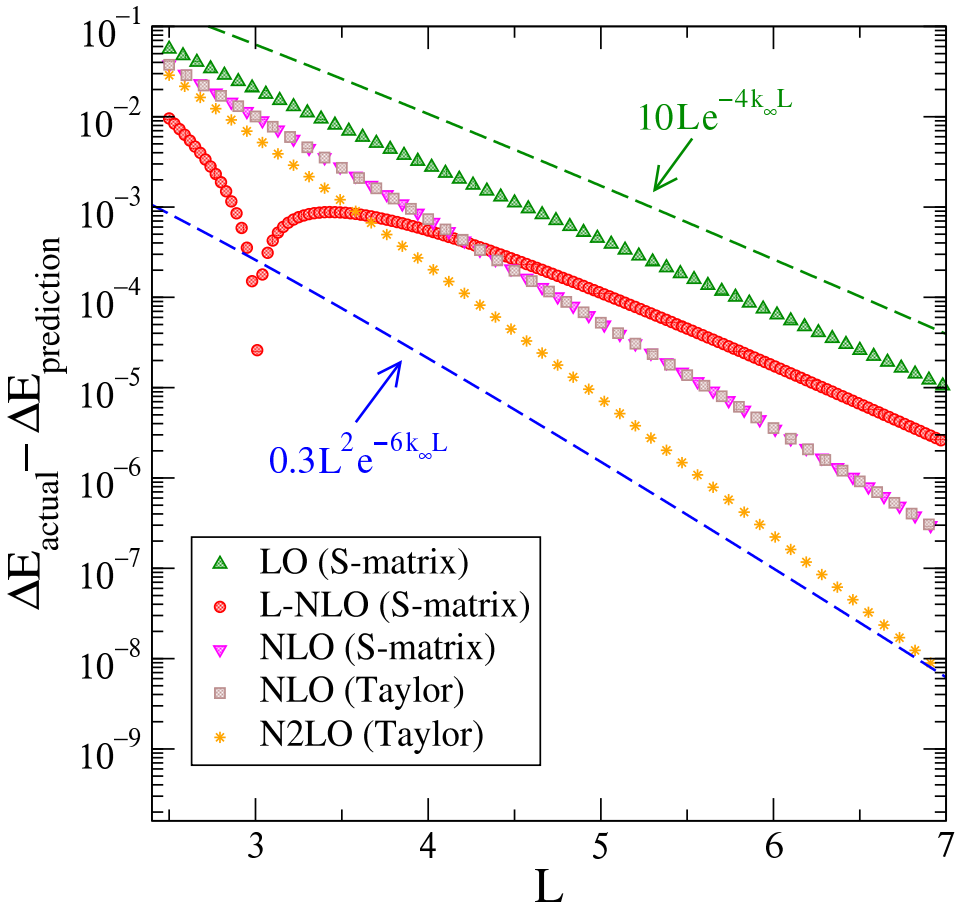


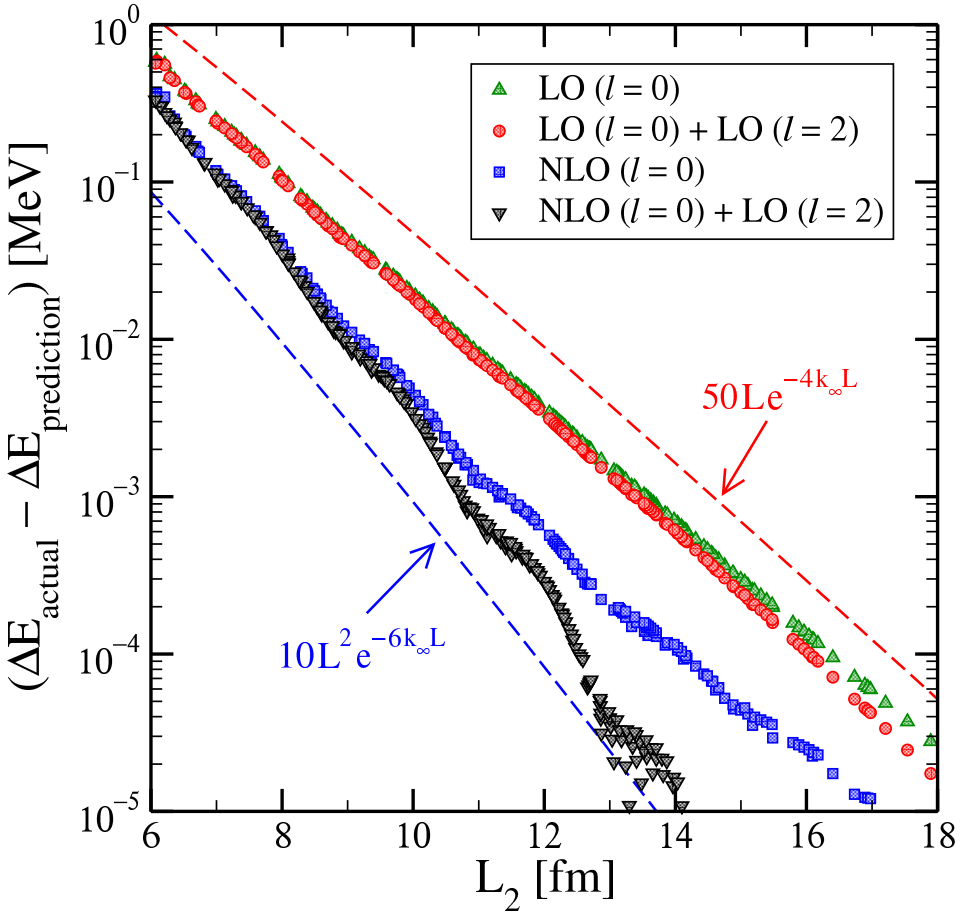
- First check against solvable model problem (square well on left)
- Ask: does it work as well as it should? $\Longrightarrow$ Are slopes correct?

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? $\Longrightarrow$ Are slopes correct?