Lecture 12

Approximations and Round-Off Errors

- **Review**
  - We know how to define variables (scalars, vectors, and matrices).
  - We know how to write functions, how to write scripts
  - To write a program, we have if statements, for loops and while loops

- **We are ready to move on to the numerical part**
- **From this lecture, we will learn some numerical techniques, and learn how to write matlab programs to implement the numerical techniques**

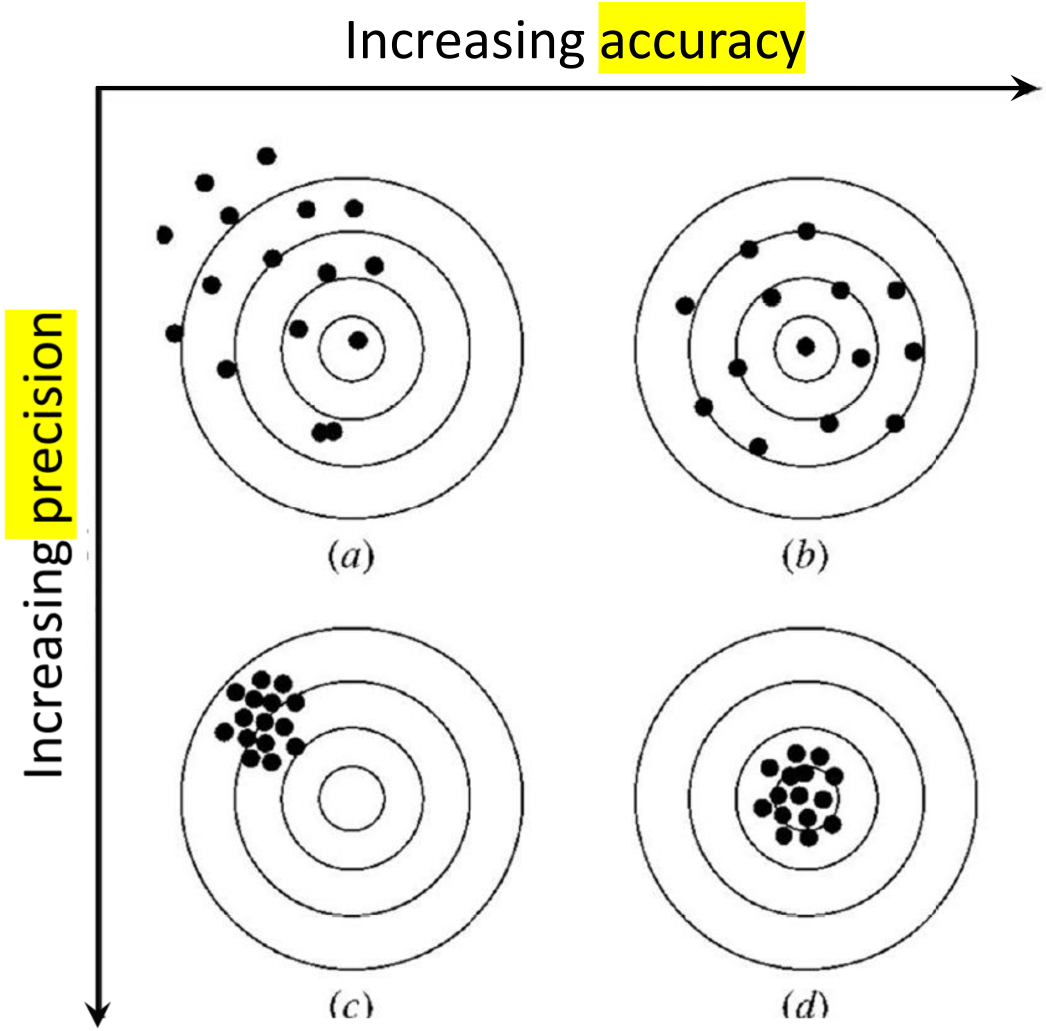
Approximations and Round-Off Errors

- For many engineering problems, we cannot obtain analytical solutions.
- Numerical methods yield approximate results, results that are close to the exact analytical solution. Because we have no idea about the exact solution, we cannot exactly compute the errors associated with numerical methods.

Approximations and Round-Off Errors

- How confident we are in our approximate result?
- How much error is present in our calculation and is it tolerable?
- **Accuracy**. How close is a computed or measured value to the true value
- **Precision** (or *reproducibility*). How close is a computed or measured value to previously computed or measured values.
- **Inaccuracy** (or *bias*). A systematic deviation from the actual value.
- **Imprecision** (or *uncertainty*). Magnitude of scatter.

Approximations and Round-Off Errors



- The center of the bull's eye is the true value;
- The black dots represent the data from a numerical calculation or an experimental measurement

Approximations and Round-Off Errors

➢ **Significant Figures**

- Number of significant figures indicates precision. Significant figures (significant digits) of a number are those that can be *used* with *confidence,* e.g., the certain digits plus one estimated digit.
- Examples

12.3450	6
12.345	5
12.35	4

- Zeros are sometimes used to locate the decimal point, not significant figures.

0.00001753	4
0.0001753	4
0.001753	4

Approximations and Round-Off Errors

➢ Significant Figures

1. All non-zero numbers are significant figure (e.g. 33.2, three)
2. Zeros between two non-zero digits are significant figures (e.g. 2051, four)
3. Leading zeros are not significant (e.g. 0.0032, two)
4. Trailing zeros to the right of the decimal are significant (e.g. 92.00, four)
5. Trailing zeros in a whole number with the decimal shown are significant (for example, 540. , three)
6. Trailing zeros in a whole number with no decimal shown are not significant (for example, 540, two)
7. For a number in scientific notation: $a \times 10^n$, all digits comprising 'a' are significant by the first 6 rules, 10 and n are not significant (e.g. 5.02×10^4)

Approximations and Round-Off Errors

➢ Error Definitions

- True Value = Approximation + Error
- True Error: E_t = True Value – Approximation (+/-)
- True fractional relative error = $\dfrac{\text{True error}}{\text{True value}}$
- True percent relative error, $\varepsilon_t = \dfrac{\text{True error}}{\text{True value}} \times 100\ \%$

In most cases, we have no idea about the true value.

Approximations and Round-Off Errors

- For numerical methods, the true value will be known only when we deal with functions that can be solved analytically (simple systems). In real world applications, we usually don't know the answer. Then we define the approximate percent relative error

$$\varepsilon_a = \frac{\text{Approximation error}}{\text{approximation}} \times 100\ \%$$

How to estimate the approximation error?

Approximations and Round-Off Errors

- Iterative approach – Many times, we solve the problem iteratively. For example, we have equation to solve. We have no idea what the answer should be. Typically we should guess a value, and then we will perform the calculation over and over until the solution stops changing. The approximate percent relative error is defined as

$$\varepsilon_a = \frac{\text{Current approximation} - \text{Previous approximation}}{\text{Current approximation}} \times 100\ \%$$

- Use absolute value
- Computations are repeated until stopping criterion is satisfied.

$$|\varepsilon_a| < \varepsilon_S$$

ε_S is the pre-specified % tolerance based on the knowledge of your solution

- Using the following definition of ε_S and the above criterion, the result is correct to at least n significant figures (**IMPORTANT!**)

$$\varepsilon_S = (0.5 \times 10^{(2-n)})\ \% \qquad (n = 3,\ \varepsilon_S = 0.05\% = 0.0005)$$

Approximations and Round-Off Errors

- Example: The square root of a number ***N*** can be approximated by repeated calculation using the formula

$$\boldsymbol{NG = 0.5\left(LG + \frac{N}{LG}\right)}$$

where ***NG*** stands for next guess and ***LG*** stands for last guess. Write a program that calculates the square root of a number using this method.

It is not known how many times the calculation will be repeated, so we have to use a while loop.

Approximations and Round-Off Errors

- Example:

```
clc
clear
n = 3;     % define number of significant figures
es = 0.5*10^(2-n);  % define % tolerance
N = input('Enter a number to calculate the sqrt:  ')
ov = 1.1;          % define the initial guess of old value
ea = 1.;           % initialize the relative error
while ea > es
    nv = 0.5*(ov + N/ov);     % calculate new value
    ea = 100*abs((nv-ov)/nv);  % calculate relative error
    ov = nv;          % assign new value to old value for
                      % next iteration
end
fprintf('\nThe sqrt of %.2f is: %.2f\n\n', N, nv)
```

$\varepsilon_s = (0.5 \times 10^{(2-n)})\ \%$

$|\varepsilon_a| < \varepsilon_s$

$$\boldsymbol{NG = 0.5\left(LG + \frac{N}{LG}\right)}$$

$$\varepsilon_a = \frac{a_c - a_p}{a_c} \times 100\%$$

Approximations and Round-Off Errors

- **Homework on Canvas**