Computational Physics

Round-Off Errors

02/03/2009

Outline

Functions

```
# include <iostream.h>

int Factorial (int aI) {
      int j = 1;

      for (int i=1; i < (aI + 1); i++) {
            ...
      }
      return j;
}

main() {
      ...
      k = Factorial(N);
}
```

1. The input variables are termed *arguments*.
2. A function can be called with either variables or constants as parameters.

Functions

```
# include <iostream.h>
# include <math.h>

int Factorial (int aI) {
      int j = 1;

      for (int i=1; i < (aI + 1); i++) {
            ...
      }
      return j;
}

main() {
      ...
      k = Factorial(N);
      k = factorial(N);
}
```

1. The input variables are termed *arguments*.
2. A function can be called with either variables or constants as parameters.

Outline

Computational Errors

- Human Errors
  - Blunders
- Random Errors
  - Acts of Nature
- Approximation Errors

$$e^x \approx \sum_n^N (-x)^n / n!$$

- Range Errors
- Round-Off Errors

Computational Errors

- Human Errors
  - Blunders
- Random Errors
  - Acts of Nature
- Approximation Errors

$$e^x \approx \sum_n^N (-x)^n / n!$$

- Range Errors
- Round-Off Errors

Round-Off Errors

Machine Accuracy ϵ

The largest number such that

$$1.0 + \epsilon = 1.0$$

Computer Representation x_c:

$$x_c = x(1 + \epsilon_x) \quad |\epsilon_x| \leq \epsilon$$

Round-Off Errors

- Example: A Simple Sum $S = \sum \frac{1}{n}$

$$S_{up} = \sum_{n=1}^{N} \frac{1}{n} = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{N}$$

round-off error

$$S_{down} = \sum_{n=N}^{1} \frac{1}{n} = \frac{1}{N} + \frac{1}{N-1} + \frac{1}{N-2} + \ldots + 1$$

$$S_{up} \neq S_{down}$$

more precise

Subtractive Cancelation Errors

Subtractive Cancelation Errors

$$a = b - c \quad \longrightarrow \quad a_c = b_c - c_c \qquad a_c = a(1+\epsilon_a)$$

$$a(1+\epsilon_a) = b(1+\epsilon_b) - c(1+\epsilon_c)$$

$$1+\epsilon_a = 1+\epsilon_b b/a - \epsilon_c c/a$$

$$\boxed{\epsilon_a = \epsilon_b b/a - \epsilon_c c/a}$$

if a is small then $b \approx c$

$$\epsilon_a \approx b/a(\epsilon_b - \epsilon_c)$$

If you subtract two large numbers and end up with a small one, there will be less significance in the small one.

Subtractive Cancelation

- Example: Subtractive Cancelation

$$a\,x^2 + b\,x + c = 0$$

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad x'_{1,2} = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

if $b^2 >> 4ac$

then for:

$b>0$ x_1 & x'_2 are imprecise → use x_2 & x'_1

$b<0$ x_2 & x'_1 are imprecise → use x_1 & x'_2

Multiplicative Errors

- Multiplicative Errors

$$a = b * c \longrightarrow a_c = b_c * c_c$$

$$a_c = a(1+\epsilon_a)$$

$$a(1+\epsilon_a) = b(1+\epsilon_b) * c(1+\epsilon_c)$$

$$1+\epsilon_a \simeq 1+\epsilon_b+\epsilon_c$$

$$\boxed{\epsilon_a = \epsilon_b + \epsilon_c}$$

Since ϵ_b and ϵ_c can have opposite signs the total error can be larger or smaller than the individual errors