LECTURE NOTE #6

Numerical Errors

Numerical Methods should be sufficiently accurate to meet the requirement of a particular engineering problem. They also should be precise enough for adequate engineering design.

Numerical “error” represents both the inaccuracy and the imprecision of our predictions by numerical methods. It can be categorized as:

*Truncation error*
: results when approximations are used to represent exact mathematical procedures

*Round-off error (or Chopping error)*
: results when numbers having limited significant figures are used to represent exact numbers

1. **Truncation Error**

   The term *truncation error* generally refers to the error involved in using a truncated or finite summation to approximate the sum of an infinites series such as The Taylor Series.

2. **Round-off Error**

   Let's say, a computer converts a value of 0.4 to a normalized floating-point value with eight-bit (bit: a 1-binary-digit) mantissas, e.g., converting 0.4 to a binary number (a base-2 system):

   \[ 0.4 \times 10^0 = (0110011001100...)_2 \]

   To eight bits, we get \( 0.4 \rightarrow (01100110)_{10} \). Converting \( (01100110)_{10} \) back to a base-10 system gives:
\[
(01100110)_2 = 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 0 \times 2^{-5} + 1 \times 2^{-6} + 1 \times 2^{-7} + 0 \times 2^{-8}
\]

\[
= \frac{1}{4} + \frac{1}{8} + \frac{1}{64} + \frac{1}{128} = 0.398437500
\]

Therefore, because of using a base-2 representation, it cannot precisely represent certain exact base-10 numbers. This discrepancy introduced by omission of significant figures is called round-off error. So, you can say, “Numerical round-off errors are directly related to the manner in which numbers are stored in a computer.

By the way, you may have a question about what I mean by “significant figures”. That’s a good question. Definition of Significant Figures is “a number are those that can be used with confidence” (by Chapra and Candle, Numerical Methods for Engineers), for example, if we have the number 45300, then we have four groups of 10000, five groups of 1000, and three groups of 100. In a normalized format, the number can be expressed as

\[
\begin{align*}
\underline{4.53} & \times 10^4 \\
underline{4.530} & \times 10^4 \\
\underline{4.5300} & \times 10^4 
\end{align*}
\]

which indicate that each representation of the number 45300 has different level of confidence; \(4.53 \times 10^4\) means that we are sure about this number upto the second decimal point so that the significant number of this representation is three.
3. Computer Arithmetic

A single precision floating-point number used in the IBM 3000 and 4300 series consists of a 1-binary-digit (bit) sign indicator, a 7-bit exponent with a base of 16, and a 24-bit mantissa. 24 binary digits correspond to between 6 and 7 decimal digits so that we can assume that this number has at least 6 decimal digits of precision for the floating-point number system.

For example, consider the machine number

\[
\begin{array}{c|c|c}
\text{sign} & \text{exponent} & \text{mantissa} \\
\hline
0 & 000010 & 1011001100000100000000000 \\
\end{array}
\]

The leftmost bit is a zero, which indicates that the number is positive. The next seven bits, 0000010, are equivalent to the decimal number:

\[+(0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0) = 2\]

Then, this machine number represents the decimal number:

\[+16^2 \times \left[ (1/2)^1 + (1/2)^3 + (1/2)^4 + (1/2)^7 + (1/2)^8 + (1/2)^{14} \right] \]

\[= 179.0156097412109375\]
However, the next largest machine number is

\[
\text{signof} \quad \text{signof} \quad \text{exponent} \quad \text{mantissa}
\begin{array}{c}
0 \quad 0 \quad 000010 \quad 101100110000010000000001
\end{array}
\]

= 179.0156402587890625

and the next smallest machine number is

\[
\text{signof} \quad \text{signof} \quad \text{exponent} \quad \text{mantissa}
\begin{array}{c}
0 \quad 0 \quad 000010 \quad 101100110000001111111111
\end{array}
\]

= 179.0156097412109375

That is, the original machine number must represent not only 179.0156097412109375, but many real numbers that are between this number and its nearest machine number neighbors. To be precise, the original machine number is used to represent any real numbers in the interval: [179.0156097412109375, 179.0156402587890625]
Example: Numerical Errors Led to System Failure at Dhahran, Saudi Arabia?

The Patriot Missile
Air defense mobile missile system
Defend against incoming aircraft and cruise missile
Operate a few hours at a time
“Scudbuster”

Scud Missile (Al Hussein)

It turns out that the cause was an inaccurate calculation of the time since boot due to computer arithmetic errors. Specifically, the time in tenths of second as measured by the system's internal clock was multiplied by 10 to produce the time in seconds. This calculation was performed using a 24 bit fixed point register (radix point is fixed. What's the other way to store a number in computer?!?). In particular, the value 1/10, which has a non-terminating binary expansion, was chopped at 24 bits after the radix point. The small chopping error, when multiplied by the large number giving the time in tenths of a second, lead to a significant error.

Indeed, the Patriot battery had been up around 100 hours, and an easy calculation shows that the resulting time error due to the magnified chopping error was about 0.34 seconds. (The number 1/10 equals $\frac{1}{2^4} + \frac{1}{2^5} + \frac{1}{2^8} + \frac{1}{2^{10}} + \frac{1}{2^{12}} + \frac{1}{2^{13}} + ...$. In other words, the binary expansion of 1/10 is .000 1100 1100 1100 1100 1100 1100 1100... Now the 24-bit register in the Patriot stored instead .000 1100 1100 1100 1100 1100 introducing an error of .000 0000 0000 0000 0000 0000 1100 1100... binary, or about 0.000000095 decimal.

Multiplying by the number of tenths of a second in 100 hours gives $0.000000095 \times 100 \times 60 \times 60 \times 10 = 0.34$.) A Scud travels at about 1,676 meters per second, and so travels more than half a kilometer in this time: $1676 \times 0.34 = 569.84$ meters. This was far enough that the incoming Scud was outside the "range gate" that the Patriot tracked. Ironically, the fact that the bad time calculation had been improved in some parts of the code, but not all, contributed to the problem, since it meant that the inaccuracies did not cancel.
Target acquisition procedure in the Patriot missile system

Patriot missile launching