Lecture 3: Numbers for crunching

Sourendu Gupta

TIFR Graduate School

Computational Physics 1
February 8, 2010

Outline

Bits, bytes and words

In a modern computer memory each unit of memory is a switch (bit) which is either off (meaning 0) or on (meaning 1). Switches are organized into groups of 8 called bytes. A single byte can represent the numbers

$$00000000 = 0, 00000001 = 1, \cdots, 11111111 = 255.$$

On most machines, groups of 4 bytes are called a word. Most numbers in a computer are 1 word long. You associate one word in the computer's memory through a declaration such as:

C example	FORTRAN example
`int number;`	`integer number`
`float another;`	`real another`

Numbers of type `complex`, `double` or `long int` are two words long.

Problem 1: Write your age in octal (base 8). Write out the current year in octal. Write a Mathematica code to convert a number from decimal notation to octal and vice versa.

Integers and machine infinity

Subsets of integers are implemented on computers. The size of M depends on the length of a word. Machine integers are the set

$$\mathcal{I} = \{x \mid |x| \leq M\}.$$

Addition of machine integers is clearly not closed: for example, $1 + M$ cannot be accommodated in a machine. Typically, any integer $|x| > M$ is called machine infinity. We will denote this by the symbol ∞.
One could extend the definition of machine integers to

$$\mathcal{I}_+ = \mathcal{I} \cup \{\infty\}.$$

This has other problems.

Problem 2: On a certain machine $M = 2147483647$. How many bytes constitute a word on this machine?

Groups

If a set S is

- **closed** under a binary operation $\otimes$,
- has an **unique identity** $\mathbf{1} \in S$ such that $x \otimes \mathbf{1} = x$,
- each element has an **unique inverse**, $x \otimes x^{-1} = \mathbf{1}$,
- and the operation is **associative**, $x \otimes (y \otimes z) = (x \otimes y) \otimes z$ for all $x, y, z \in S$,

then $\{S, \otimes\}$ is a group.

Integers

The set of integers, Z, is closed under addition; 0 is the identity; for any integer z, $-z$ is the inverse; the order of addition does not matter. Hence $\{Z, +\}$ is a group.

.

Machine integers

The set of machine integers, $\mathcal{I}$, is not closed under addition. $\mathcal{I}_+$ is closed; 0 is the identity; but ∞ has no inverse element. Hence $\{\mathcal{I}, +\}$ and $\{\mathcal{I}_+, +\}$ are not groups.

.

Floating point numbers and machine precision

In decimal notation every real number can be represented in the form

$$\pm 0.d_1 d_2 d_3 d_4 d_5 \cdots \times 10^n,$$

where each digit $d_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ (except d_1 which cannot take the value 0), the number of digits can be anything, and the exponent n is an integer. Only a subset of such numbers can be represented in one word.
Floating point numbers are signed numbers of the form

$$\pm 0.d_1 d_2 \cdots d_p \times 10^n,$$

where p is the maximum number of digits allowed and n is an integer in the range $|n| \leq N$. The numbers p and N can be dependent on the word length and other features of the representation. The machine precision, ϵ_m, is the largest number for which $1 + \epsilon_m = 1$.

Problem 3: Take a make-believe machine which has $p = 2$ and $N = 2$. Which real numbers can be represented in it? What is the machine precision on such a number.

How computer arithmetic is done

A floating point number is stored in normalized form, *i.e.*, the first digit after the decimal point is not zero. When adding two numbers, the smaller is de-normalized by shifting the digits until the exponents match. In this process one can lose some digits altogether. Here is an example on a make-believe machine with $p = 2$,

$$\begin{aligned} 0.48E{+}1 + 0.16E{-}2 &= 0.48E{+}1 + 0.01E{-}1 \\ &= 0.48E{+}1 + 0.00E{+}0 \\ &= 0.48E{+}1 + 0.00E{+}1 \\ &= 0.48E + 1 \end{aligned}$$

This procedure gives rise to the notion of machine precision. On a machine with $p = 2$ what is ϵ_m?

Problem 4: See the C/FORTRAN codes associated with this lecture set. What do they do?

Loss of associativity

Addition of real numbers is associative, *i.e.*, the order of addition does not matter. For example,

$$(0.98 + 0.006) + 0.004 = 0.98 + (0.006 + 0.004) = 0.99.$$

For floating point numbers the order matters. For example, with $p = 2$ and $N = 2$ one has

$$\begin{aligned}(0.98E{+}0 + 0.60E{-}2) + 0.40E{-}2 &= 0.98E{+}0 + 0.40E{-}2 = 0.98E{+}0\\ 0.98E{+}0 + (0.60E{-}2 + 0.40E{-}2) &= 0.98E{+}0 + 0.10E{-}1 = 0.99E{+}0.\end{aligned}$$

Multiplication of real numbers is also associative. However, multiplication of floating point numbers is not associative. Here is an example,

$$\begin{aligned}(0.99E{+}0 \times 0.50E{+}0) \times 0.20E{+}1 &= 0.49E{+}0 \times 0.20E{+}1 = 0.98E{+}0,\\ 0.99E{+}0 \times (0.50E{+}0 \times 0.20E{+}1) &= 0.99E{+}0 \times 0.10E{+}1 = 0.99E{+}0.\end{aligned}$$

Catastrophic loss of precision

Problem 5: We have shown that $0.50E{+}0 \times 0.20E{+}1$ is not uniquely equal to 1. Construct an example showing that $a + b = 0$ does not have an unique solution for b given a.

The lack of an unique inverse operation for addition and multiplication sometimes causes you to lose all precision in the middle of a computation. For example, if you want to do the computation $171 \times (1/17) - 10$ with $p = 2$ you will have

$$\begin{aligned} & 0.17E{+}3 \times (0.10E{+}1/0.17E{+}2) - 0.10E{+}2 \\ & = 0.17E{+}3 \times 0.59E{-}1 - 0.10E{+}2 \\ & = 0.10E{+}2 - 0.10E{+}2 = 0. \end{aligned} \tag{1}$$

If you are aware of the possibility of such loss of precision you may want to examine the expression and find that $171 \times (1/17) - 10 = 1/17$. Then the maximally accurate computation would be $0.10E{+}0/0.17E{+}2 = 0.59E{-}1$.

Intervals

Problem 6: An interval on real numbers is the set

$$(a, b) = \{x | a \leq x < b\}.$$

A floating point number is not a real number; but we can try to associate it with some interval on real numbers.

Arithmetic has a natural definition on intervals;

$$\begin{aligned} (a, b) + (c, d) &= (a + c, b + d) \\ (a, b) \times (c, d) &= (ac, bd). \end{aligned}$$

Is it possible to interpret floating point numbers as intervals? If so, how does one resolve the problems of non-associativity and the non-existence of inverses of addition and multiplication? Is this representation useful in understanding how uncertainties in arithmetic propagate through a computation?

Outline

The nature of physical measurements

Any single instance of a physical measurement usually yields a reading on an instrument. There is a finite resolution for every measurement. Conventionally one does not quote more digits than are measured. For example, Avogadro's number is

$$N_A = 6.022\ 141\ 79\ (30) \times 10^{23}.$$

The number in brackets is an estimate of the error in the two least significant digits. One does not write

$$N_A \stackrel{?}{=} 6.022\ 141\ 793\ 152\ (300\ 000) \times 10^{23},$$

although it lies in the interval specified in the previous instance, because the extra digits lie beyond the resolution of the measurement.
In this sense, all physical measurements are floating point numbers. When you manipulate results of measurements, you should be worried about subtraction and division, the propagation of errors, and catastrophic loss of significance.

The nature of physical theories

Every physical theory is a map from observation to prediction. For example, classical mechanics predicts the future behaviour of a system of particles given observations of its present state. Typically, given measurements of the position, $\mathbf{x}_0$, and momentum, $\mathbf{p}_0$, at a time t_0, one obtains predictions for the position, $\mathbf{x}(t; \mathbf{x}_0, \mathbf{p}_0, t_0)$, and momentum, $\mathbf{p}(t; \mathbf{x}_0, \mathbf{p}_0, t_0)$, at any time t. If there are uncertainties in the measurement of the initial state, there will also be uncertainties in the final state.

> **Problem 7**: Draw phase space trajectories for a one-dimensional harmonic oscillator. If there are errors in the measurements of initial position and momenta, δx_0 and δp_0, show how they affect the uncertainties in predictions. How is this different for the anharmonic oscillator with $V(x) = x^4$?

Using real numbers to model physical theories can give a wrong impression of great precision. The main problem of rigorous computation, that of the propagation of errors, is at the heart of predictability and testability in the sciences. This is not emphasized by standard mathematical analysis.

Outline

Convergent and absolutely convergent

Usually one thinks of a sum of a series as the operations—

$$\sum_{n=1}^{\infty} T_n = \cdots ((((T_1 + T_2) + T_3) + T_4) + T_5) + \cdots$$

When the brackets can be changed around without changing the result, the series is said to be unconditionally convergent. The exceptions are those series which are not absolutely convergent.
The Riemann rearrangement theorem states that by a rearrangement a series which is not absolutely convergent may be made to yield any result at all.

Problem 8: Prove that

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} + \cdots = \ln 2,$$

$$1 - \frac{1}{2} - \frac{1}{4} + \frac{1}{3} - \frac{1}{6} - \frac{1}{8} + \frac{1}{5} \cdots = \frac{1}{2} \ln 2.$$

Summing in floating point

When the series is convergent as well as absolutely convergent, does floating point arithmetic change the results? Take two series—

$$S = \sum_{n=1}^{\infty} T_n \qquad \text{and} \qquad S' = \sum_{n=1}^{\infty} T'_n,$$

where $T'_n = T_n(1+\epsilon_n)$. We choose ϵ_n to be of the order of the machine precision, ϵ. Then

$$|S' - S| = \left| \sum_{n=1}^{\infty} \epsilon_n T_n \right| \leq \epsilon \sum_{n=0}^{\infty} |T_n|.$$

Since the series is absolutely convergent, the error is bounded. Numerical results for series which are absolutely convergent can be bounded in error by this means. However, this result also shows that floating point summation of series which are not absolutely convergent can go terribly wrong when done in floating point.

Bad behaviour of convergent series

The Taylor series expansion of the sine function is

$$\sin x = \sum_{n=1}^{\infty} \frac{(-1)^{i-1} x^{2i-1}}{(2i-1)!}.$$

This series is absolutely convergent for all values of x. When x is small enough then just a few terms of the series give a good approximation to the result. For example, when $x = 0.01$, the error in neglecting all but the first term is less than 1%. When x is large, the terms increase significantly before they begin to decrease. Many terms have to be summed, some very large, before it converges to a value in the range ± 1. In the process there may be catastrophic loss of significance (see the Mathematica program).

Problem 9: Given a (large) value of x, for which value of n does the term reach its maximum? (Hint: If x is large enough, then the n would be large enough to use the Stirling approximation.) How large is the maximum value? Does this give us limits on using the Taylor series expansion for evaluating a sine?

How to sum a power series

Given a power series

$$S(x) = \sum_{i=0}^{N} a_i x^i,$$

what is the most stable method of summing it? The following are equivalent over real numbers—

$$\begin{aligned} S(x) &= (a_0 + ((a_1 x) + ((a_2 (x^2)) + \cdots))) && (2) \\ S(x) &= (a_0 + x(a_1 + x(a_2 + \cdots))) && (3) \\ S(x) &= (((a_N x + a_{N-1})x + a_{N-2}) + \cdots). && (4) \end{aligned}$$

The number of arithmetic operations that you need to perform in evaluating the expression in (1) is $\mathcal{O}(N^3)$, whereas for the other two forms it is $\mathcal{O}(N)$. Is the accuracy of these different forms also different?

Outline

References and further reading

1. Numerical Recipes, by W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Cambridge University Press. Treat this as a well written manual for a ready source of building blocks. Read the introductory chapter for a description of arithmetic on a computer.
2. The IEEE 754-2008 floating point standard. This is the most widely used standard for floating point arithmetic. You should read this article in Wikipedia carefully enough to understand what problems it addresses.
3. Read, compile and run the C or FORTRAN code associated with this lecture. What does it do?
4. Several points in this lecture are illustrated in the associated Mathematica notebooks. These notebooks also have additional examples. They are given in `tar.gz` format. Unpack them using the `tar -xvzf` command.