## SOME DEFINITIONS

Let $x_T$ denote the true value of some number, usually unknown in practice; and let $x_A$ denote an approximation of $x_T$.

The <u>error</u> in $x_A$ is

$$error(x_A) = x_T - x_A$$

The <u>relative error</u> in $x_A$ is

$$rel(x_A) = \frac{error(x_A)}{x_T} = \frac{x_T - x_A}{x_T}$$

<u>Example</u>: $x_T = e$, $x_A = \frac{19}{7}$. Then

$$error(x_A) = e - \frac{19}{7} \doteq .003996$$

$$rel(x_A) = .00147$$

We also speak of <u>significant digits</u>. We say $x_A$ has $m$ significant digits with respect to $x_T$ if the magnitude of $error(x_A)$ is $\leq 5$ units in the $(m+1)^{st}$ digit, beginning with the first nonzero digit in $x_T$. Above, $x_A$ has 3 significant digits with respect to $x_T$.

## SOURCES OF ERROR

This is a very rough categorization of the sources of error in the calculation of the solution of a mathematical model for some physical situation.

(E1) Modelling Error: As an example, if a projectile of mass $m$ is travelling thru the earth's atmosphere, then a popular description of its motion is given by

$$m\frac{d^2\mathbf{r}(t)}{dt^2} = -mg\mathbf{k} - b\frac{d\mathbf{r}}{dt}$$

with $b \geq 0$. In this, $\mathbf{r}(t)$ is the vector position of the projectile; and the final term in the equation represents friction. If there is an error in this a model of a physical situation, then the numerical solution of this equation is not going to improve the results.

(E2) Blunders: In the pre-computer era, these were likely to be arithmetic errors. In the earlier years of the computer era, the typical blunder was a programming error. These were usually easy to find as they generally resulted in absurd calculated answers.

Present day "blunders" are still often programming errors. But now they are often much more difficult to find, as they are often embedded in very large codes which may mask their effect. Some simple rules:
(i) Break programs into small testable subprograms.
(ii) Run test cases for which you know the outcome.
(iii) When running the full code, maintain a skeptical eye on the output, checking whether the output is reasonable or not.

(E3) Observational Error: The radius of an electron is given by

$$(2.81777 + \varepsilon) \times 10^{-13} \text{ cm}, \qquad |\varepsilon| \leq .00011$$

This error cannot be removed, and it must affect the accuracy of any computation in which it is used. We need to be aware of these effects and to so arrange the computation as to minimize the effects.

(E4) Rounding/chopping Error: This is the main source of many problems, especially problems in solving systems of linear equations. We later look at the effects of such errors.

(E5) Approximation Error: This is also called "discretization error" and "truncation error"; and it is the main source of error with which we deal in this course. Such errors generally occur when we replace a computationally unsolvable problem with a nearby problem that is more tractable computationally.

For example, the Taylor polynomial approximation

$$e^x \approx 1 + x + \frac{1}{2}x^2$$

contains an "approximation error".
The numerical integration

$$\int_0^1 f(x)\,dx \approx \frac{1}{n}\sum_{j=1}^{n} f\left(\frac{j}{n}\right)$$

contains an approximation error.
Finally, the numerical differentiation formula

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

contains an approximation error.

## LOSS OF SIGNIFICANCE ERRORS

This can be considered a source of error or a consequence of the finiteness of calculator and computer arithmetic. We begin with some illustrations.

Example. Define

$$f(x) = x\,[\text{sqrt}(x+1) - \text{sqrt}(x)]$$

and consider evaluating it on a 6-digit decimal calculator which uses rounded arithmetic. The values of $f(x)$, taken from the text in Section 2.2:

| $x$ | Computed $f(x)$ | True $f(x)$ |
|---:|---:|---:|
| 1 | .414210 | .414214 |
| 10 | 1.54340 | 1.54347 |
| 100 | 4.99000 | 4.98756 |
| 1000 | 15.8000 | 15.8074 |
| 10000 | 50.0000 | 49.9988 |
| 100000 | 100.000 | 158.113 |

What happened?

Example. Define

$$f(x) = \frac{1 - \cos x}{x^2}, \qquad x \neq 0$$

Values for a sequence of decreasing positive values of $x$ is given in Section 2.2 of the text, using a past model of a popular calculator. The calculator carried 10 decimal digits, and it used rounded arithmetic.

| $x$ | Computed $f(x)$ | True $f(x)$ |
|---|---|---|
| 0.1 | 0.4995834700 | 0.4995834722 |
| 0.01 | 0.4999960000 | 0.4999958333 |
| 0.001 | 0.5000000000 | 0.4999999583 |
| 0.0001 | 0.5000000000 | 0.4999999996 |
| 0.00001 | 0.0 | 0.5000000000 |

Consider one case, that of $x = .001$. Then on the calculator:

$$\begin{aligned} \cos(.001) &= .9999994999 \\ 1 - \cos(.001) &= 5.001 \times 10^{-7} \\ \frac{1 - \cos(.001)}{(.001)^2} &= .5001000000 \end{aligned}$$

The true answer is $f(.001) = .4999999583$. The relative error in our answer is

$$\frac{.4999999583 - .5001}{.4999999583} = \frac{-.0001000417}{.4999999583} \doteq -.0002$$

There 3 significant digits in the answer. How can such a straightforward and short calculation lead to such a large error (relative to the accuracy of the calculator)?

When two numbers are nearly equal and we subtract them, then we suffer a “loss of significance error” in the calculation. In some cases, these can be quite subtle and difficult to detect. And even after they are detected, they may be difficult to fix.

The last example, fortunately, can be fixed in a number of ways. Easiest is to use a trigonometric identity:

$$\cos(2\theta) = 2\cos^2(\theta) - 1 = 1 - 2\sin^2(\theta)$$

Let $x = 2\theta$. Then

$$\begin{aligned} f(x) &= \frac{1-\cos x}{x^2} = \frac{2\sin^2(x/2)}{x^2} \\ &= \frac{1}{2}\left[\frac{\sin(x/2)}{x/2}\right]^2 \end{aligned}$$

This latter formula, with $x = .001$, yields a computed value of .4999999584, nearly the true answer. We could also have used a Taylor polynomial for $\cos(x)$ around $x = 0$ to obtain a better approximation to $f(x)$ for small values of $x$.

## A MORE SUBTLE EXAMPLE

Evaluate $e^{-5}$ using a Taylor polynomial approximation:

$$e^{-5} \approx 1 + \frac{(-5)}{1!} + \frac{(-5)^2}{2!} + \frac{(-5)^3}{3!} + \cdots + \frac{(-5)^n}{n!}$$

With $n = 25$, the error is

$$\left| \frac{(-5)^{26}}{26!} e^c \right| \le 10^{-8}$$

Imagine calculating this polynomial using a computer with 4 digit decimal arithmetic and rounding. To make the point about cancellation more strongly, imagine that each of the terms in the above polynomial is calculated exactly and then rounded to the arithmetic of the computer. We add the terms exactly and then we round to four digits.

See the table of results in Section 2.2 of the text. It gives a result of 0.009989 whereas the correct answer is 0.006738 to four significant digits.

To understand more fully the source of the error, look at the numbers being added and their accuracy. For example,

$$\frac{(-5)^3}{3!} = -\frac{125}{6} \rightarrow -20.83$$

in the 4 digit decimal calculation, with an error of magnitude $0.00333\ldots$ Note that this error in an intermediate step is of same magnitude as the true answer 0.006738 being sought. Other similar errors are present in calculating other coefficients, and thus they cause a major error in the final answer being calculated.

**Whenever a sum is being formed in which the final answer is much smaller than some of the terms being combined, then a loss of significance error is occurring.**

## NOISE IN FUNCTION EVALUATION

Consider using a 4-digit decimal calculator (with rounding) to evaluate the two functions

$$\begin{aligned} f_1(x) &= x^2 - 3 \\ f_2(x) &= 9 + x^2\left(x^2 - 6\right) = [f_1(x)]^2 \end{aligned}$$

Note that $f_2(x) = [f_1(x)]^2$. On our calculator,

$$f_1(x) \text{ is } \begin{cases} < 0, & 0 \le x \le 1.731 \\ = 0, & x = 1.732 \\ > 0, & x \ge 1.733 \end{cases}$$

However,

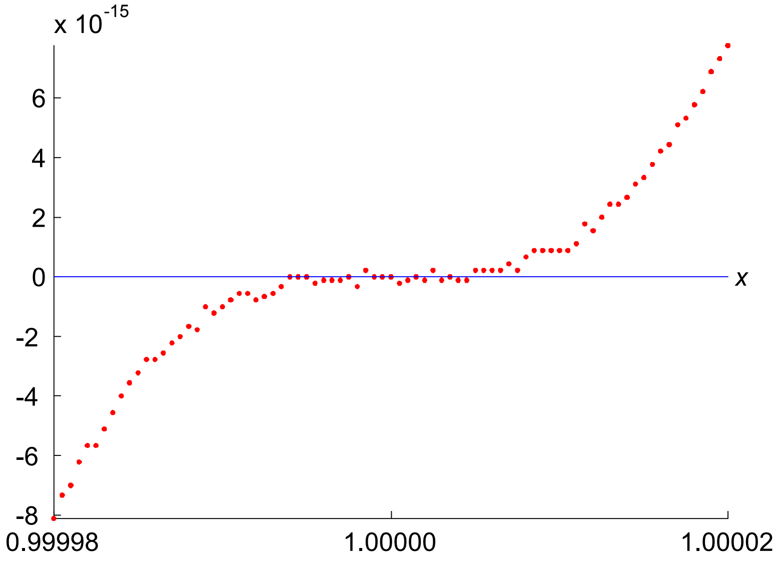
$$f_2(x) \text{ is } \begin{cases} > 0, & 0 \le x \le 1.725 \\ = 0, & 1.726 \le x \le 1.738 \\ > 0, & x \ge 1.739 \end{cases}$$

Thus $f_2(x)$ has 13 distinct zeros on this calculator; whereas we know that $f_2(x)$ has only the zeros $\pm$sqrt(3) $\doteq$ 1.732. What happened in our calculations?

Whenever a function $f(x)$ is evaluated, there are arithmetic operations carried out which involve rounding or chopping errors. This means that what the computer eventually returns as an answer contains noise. This noise is generally "random" and small. But it can affect the accuracy of other calculations which depend on $f(x)$. For example, we illustrate the evaluation of

$$\begin{aligned} f(x) &= -1 + 3x - 3x^2 + x^3 \\ &= -1 + x\,(3 + x\,(-3 + x)) \end{aligned}$$

which is simply $(x-1)^3$ and has only a single root at $x = 1$. We use *MATLAB* with its IEEE double precision arithmetic.



## UNDERFLOW ERRORS

Consider evaluating

$$f(x) = x^{10}$$

for $x$ near 0. When using IEEE single precision arithmetic, the smallest nonzero positive number expressible in *normalized* floating-point format is

$$m = 2^{-126} \doteq 1.18 \times 10^{-38};$$

see the table on IEEE single precision arithmetic with $E = 1$ and $(a_1 a_2 \dots a_{23})_2 = (00\dots0)_2$.Thus $f(x)$ will be set to zero if

$$\begin{gathered} x^{10} < m \\ |x| < m^{\frac{1}{10}} \doteq 1.61 \times 10^{-4} \\ -0.000161 < x < 0.000161 \end{gathered}$$

## OVERFLOW ERRORS

Attempts to use numbers that are too large for the floating-point format will lead to *overflow* errors. These are generally fatal errors on most computers. With the IEEE floating-point format, overflow errors can be carried along as having a value of $\pm\infty$ or *NaN*, depending on the context. Usually an overflow error is an indication of a more significant problem or error in the program and the user needs to be aware of such errors.

When using IEEE single precision arithmetic, the largest nonzero positive number expressible in *normalized* floating-point format is

$$m = 2^{128}\left(1 - 2^{-24}\right) \doteq 3.40 \times 10^{38}$$

see the table on IEEE single precision arithmetic with $E = (254)_{10}$ and $(a_1 a_2 \ldots a_{23})_2 = (11\ldots1)_2$.Thus $f(x)$ will overflow if

$$x^{10} > m$$

$$|x| > m^{\frac{1}{10}} \doteq 7131.6$$