

Physics 115/242; Peter Young

Expressions that are mathematically equivalent may not be computationally equivalent

We have learned that subtracting numbers that are nearly equal in magnitude leads to a loss of precision. Here we consider a simple example, namely roots of a quadratic equation, and show how the expression for the root can be reexpressed in a different way, which is *mathematically equivalent*, but which does not involve subtraction of nearly equal quantities, and so is numerically *not equivalent*. A brief discussion is given in Landau and Páez, Sec. 3.4.

The roots of the quadratic equation

$$ax^2 + bx + c = 0 \tag{1}$$

are, of course,

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{2}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}. \tag{3}$$

However, evaluation of these expressions numerically can lead to difficulty if b is very large since (assuming without loss of generality that $b > 0$) the two terms in the numerator of x_1 ($-b$ and $\sqrt{b^2 - 4ac}$) almost cancel, and so we lose precision. In this limit we can expand the square root in Eq. (2), and neglect $-4ac$ in the numerator of Eq. (3), to get

$$x_1 \simeq -\frac{c}{b}, \quad x_2 \simeq -\frac{b}{a}. \tag{4}$$

Here is an example of output, obtained from a program in single (32 bit) precision, for $a = 1, b = 10,000, c = 1$, for which $x_1 \simeq -10^{-4}, x_2 \simeq -10^4$:

```
x1 = 0.00000000,    x2 = -10000.00000000
```

We see that x_2 is correct, as expected, but x_1 is completely wrong. The value $x_1 = 0$ comes because the representation of b and $\sqrt{b^2 - 4ac}$ in the computer is *identical* in single precision arithmetic for these values of a, b and c .

One way to improve the situation would be to use double precision. This would work in the present situation but would still fail if b were much larger still. Better is to rearrange the formula for x_1 so that *the problem of subtracting two nearly identical numbers does not arise*.

If we multiply the expression for x_1 in the numerator and denominator by $b + \sqrt{b^2 - 4ac}$ we find

$$x_1 = -\frac{2c}{b + \sqrt{b^2 - 4ac}}, \quad (5)$$

which does not involve subtracting nearly equal quantities. Using Eq. (5) rather than Eq. (2) for x_1 the computer output is now

```
x1 = 0.00010000    x2 = -10000.00000000
```

which is correct.

The moral of this handout is that although Eqs. (2) and (5) and mathematically equivalent, they are not computationally equivalent because the roundoff errors are quite different.

Note that if b is large in magnitude but *negative*, the difficulty would arise in calculating x_2 rather than x_1 . Hence a well written code for solving a quadratic equation would test the sign of b and use

$$x_1 = -\frac{2c}{b + \sqrt{b^2 - 4ac}}, \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (6)$$

if $b > 0$, and

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = -\frac{2c}{b - \sqrt{b^2 - 4ac}}, \quad (7)$$

if $b < 0$. In this way, an accurate expression for both roots would be obtained even if b is large in magnitude.

(*Note:* A professional code would also do other tests to make sure that the program didn't "bomb" when executed, such as checking that $a \neq 0$ and that the discriminant ($b^2 - 4ac$) is not negative.)