Basic Error Principle

Every numeric method should be accompanied by an error estimate. If such a formula is lacking, is extremely complicated, or is impractical because it involves information (for instance, on derivatives) that is not available, the following may help.

Error Estimation by Comparison. *Do a calculation twice with different accuracy. Regard the difference $\tilde{a}_2 - \tilde{a}_1$ of the results $\tilde{a}_1, \tilde{a}_2$ as a (perhaps crude) estimate of the error ϵ_1 of the inferior result $\tilde{a}_1$.* Indeed, $\tilde{a}_1 + \epsilon_1 = \tilde{a}_2 + \epsilon_2$ by formula (4*). This implies $\tilde{a}_2 - \tilde{a}_1 = \epsilon_1 - \epsilon_2 \approx \epsilon_1$ because $\tilde{a}_2$ is generally more accurate than $\tilde{a}_1$, so that $|\epsilon_2|$ is small compared to $|\epsilon_1|$.

Algorithm. Stability

Numeric methods can be formulated as algorithms. An **algorithm** is a step-by-step procedure that states a numeric method in a form (a "**pseudocode**") understandable to humans. (See Table 19.1 to see what an algorithm looks like.) The algorithm is then used to write a program in a programming language that the computer can understand so that it can execute the numeric method. Important algorithms follow in the next sections. For routine tasks your CAS or some other software system may contain programs that you can use or include as parts of larger programs of your own.

Stability. To be useful, an algorithm should be **stable**; that is, small changes in the initial data should cause only small changes in the final results. However, if small changes in the initial data can produce large changes in the final results, we call the algorithm **unstable**.

This "*numeric instability*," which in most cases can be avoided by choosing a better algorithm, must be distinguished from "*mathematical instability*" of a problem, which is called "*ill-conditioning*," a concept we discuss in the next section.

Some algorithms are stable only for certain initial data, so that one must be careful in such a case.

PROBLEM SET 19.1

1. **Floating point.** Write 84.175, −528.685, 0.000924138, and −362005 in floating-point form, rounded to 5S (5 significant digits).
2. **Small differences of large numbers** may be particularly strongly affected by rounding errors. Illustrate this by computing $0.81534/(35 \cdot 724 - 35.596)$ as given with 5S, then rounding stepwise to 4S, 3S, and 2S, where "stepwise" means round the rounded numbers, not the given ones.
3. **Order of terms**, in adding with a fixed number of digits, will generally affect the sum. Give an example. Find empirically a rule for the best order.
4. **Nested form.** Evaluate

$$f(x) = x^3 - 7.5x^2 + 11.2x + 2.8$$
$$= ((x - 7.5)x + 11.2)x + 2.8$$

at $x = 3.94$ using 3S arithmetic and rounding, in both of the given forms. The latter, called the *nested form*, is usually preferable since it minimizes the number of operations and thus the effect of rounding.

5. **Quadratic equation.** Solve $x^2 - 30x + 1 = 0$ by (4) and by (5), using 6S in the computation. Compare and comment.
6. Solve $x^2 - 40x + 2 = 0$, using 4S-computation.
7. **Instability.** For small $|a|$ the equation $(x - k)^2 = a$ has nearly a double root. Why do these roots show instability?
8. **Overflow and underflow** can sometimes be avoided by simple changes in a formula. Explain this in terms of $\sqrt{x^2 + y^2} = x\sqrt{1 + (y/x)^2}$ with $x^2 \geqq y^2$ and x so large that x^2 would cause overflow. Invent examples of your own.