CSL361 Problem set 1: Numerical computations - pitfalls

January 7, 2014

1 Discretization

Because of the "discrete" nature of the floating point numbers, numerical problem solving requires discretization. Here are a few examples.

1. Derive the trapezoidal rule for numerical integration which goes somewhat like:
$$\int_a^b f(x)dx \simeq \sum_{k=0}^{n-1} \frac{1}{2}h[f(x_k) + f(x_{k+1})]$$
for discrete values of x_k in interval $[a, b]$. Compute $\int_0^\pi sin(x)dx$ using trapezoidal rule (use $h = 0.1$, $h = 0.01$ and $h = 0.001$) and compare with the exact result.

2. Consider the differential equation
$$\begin{aligned} y'(x) &= 2xy(x) - 2x^2 + 1, \quad 0 \leq x \leq 1 \\ y(0) &= 1 \end{aligned}$$
   (a) Show/verify that the exact solution is the function $y(x) = e^{x^2} + x$.

   (b) If we approximate the derivative operation with a divided difference
   $$y'(x_k) = (y_{k+1} - y_k)/(x_{k+1} - x_k)$$
   then show that solution can be approximated by the iteration
   $$\begin{aligned} y_{k+1} &= y_k + h(2x_ky_k - 2x_k^2 + 1), \quad k = 0, 1, \ldots, n \\ y_0 &= 1 \end{aligned}$$
   where $x_k = kh$, $k = 0, 1, \ldots, n$ and $h = 1/n$.

(c) Use $h = 0.1$ to solve the differential equation numerically and compare (plot) your answers with the exact solution.

3. (a) Derive the Newton's iteration for computing $\sqrt{2}$ given by

$$\begin{aligned} x_{k+1} &= \tfrac{1}{2}[x_k + (2/x_k)], \quad k = 0, 1, \ldots \\ x_0 &= 1 \end{aligned}$$

(b) Show the Newton's iteration takes $O(\log n)$ steps to obtain n decimal digits of accuracy.

(c) Numerically compute $\sqrt{2}$ using Newton's iteration and verify the rate of convergence.

4. We have discussed rounding errors in the class. Truncation errors are those that occur because of termination of a computational process after a finite number of steps. Which of the following are rounding errors and which are truncation errors?

(a) Replace $sin(x)$ by $x - (x^3/3!) + (x^5)/5! \ldots$

(b) Use 3.1415926536 for π.

(c) Use the value x_{10} for $\sqrt{2}$, where x_k is given by Newton's iteration above.

(d) Divide 1.0 by 3.0 and call the result 0.3333.

2 Unstable and Ill-conditioned problems

1. Consider the differential equation

$$\begin{aligned} y'(x) &= (2/\pi)xy(y - \pi), \quad 0 \leq x \leq 10 \\ y(0) &= y_0 \end{aligned}$$

(a) Show/verify that the exact solution to this equation is

$$y(x) = \pi y_0 / [y_0 + (\pi - y_0)e^{x^2}]$$

(b) Taking $y_0 = \pi$ compute the solution for

i. an 8 digit rounded approximation for π

ii. a 9 digit rounded approximation for π

What can you say about the results?

2. Solve the system

$$\begin{array}{rcrcl} 2x & - & 4y & = & 1 \\ -2.998x & + & 6.001y & = & 2 \end{array}$$

using any method you know. Compare the solution with the solution to the system obtained by changing the last equation to $-2.998x+6y=2$. Is this problem stable?

3. Examine the stability of the equation

$$x^3 - 102x^2 + 201x - 100 = 0$$

which has a solution $x^* = 1$. Change one of the coefficients (say 201 to 200) and show that $x^* = 1$ is no longer even close to a solution.

3 Unstable methods

1. Consider the quadratic

$$ax^2 + bx + c = 0, \quad a \neq 0$$

Consider $a = 1$, $b = 1000.01$, $c = -2.5245315$. Suppose that $\sqrt{b^2 - 4ac}$ is computed correctly to 8 digits, what is the number of digits of accuracy in x? What is the source of the error?

2. Show that the solution to the quadratic can be re-written as

$$x = -2c/(b^2 + \sqrt{b^2 - 4ac})$$

Write a program to evaluate x for several values of a, b and c (with b large and positive and a, c of moderate size). Compare the results obtained with the usual formula and the formula above.

3. Consider the problem of determining the value of the integral

$$\int_0^1 x^{20} e^{x-1} dx$$

If we let

$$I_k = \int_0^1 x^k e^{x-1} dx$$

Then, integration by parts gives us (please verify)

$$\begin{aligned} I_k &= 1 - kI_{k-1} \\ I_0 &= \int_0^1 e^{x-1}dx = 1 - (1/e) \end{aligned}$$

Thus we can compute I_{20} by successively computing $I_1, I_2, \ldots$. Compute the (k, I_k) table with a program, plot, and see if it makes sense. What are the errors due to?

Compare the results with that obtained using the following recursion (which you can *easily!* derive by integrating by parts twice).

$$I_k = (1/\pi) - [k(k-1)/\pi^2]I_{k-2}, \quad k = 2, 4, 6, \ldots$$

4. The standard deviation of a set of numbers $x_1, x_2, \ldots, x_n$ is defined as

$$s = (1/n)\sum_{i=1}^{n}(x_i - \bar{x})^2$$

where $\bar{x}$ is the average. An alternative formula that is often used is

$$s = (1/n)\sum_{i=1}^{n} x_i^2 - \bar{x}^2$$

   (a) Discuss the instability of the second formula for the case where the x_i are all very close to each other.

   (b) Observe that s should always be positive. Write a small program to evaluate the two formulas and find values of $x_1, \ldots, x_{10}$ for which the second one gives negative results.

5. Under fixed point arithmetic, if we add n numbers, then either we get the exact answer or we get an "overflow" error. This is not true for floating point arithmetic. Consider the `Matlab` program sumtrouble.py. You might want to play with the program, changing n or h or computing the difference between the computed sums and the true ones in order to understand what is happening.

   (a) Is $sum1$ equal to the true value? If not, why not? (Please try be specific - it is not enough to say "round off caused the error" (in typical IITD style).

   (b) Are any of $sum2$ and $sum3$ equal to their true value? If not, why not? Why are they different?

   (c) Ditto for $sum4$ and $sum5$.