Study Questions Homework 1 Introduction to Computational Finance Spring 2023

These are study questions. You are not required to submit solutions (even though the problems are worded like graded assignment problems).

Written Study Exercises

Problem 1.1

Suppose the n-bit 2's complement representation is used to encode a range of integers, $-2^{n-1} \leq x \leq 2^{n-1} - 1$.

1.1.a. If $x \geq 0$ then $-x$ is represented by bit pattern obtained by complementing all of the bits in the binary encoding of x, adding 1 and ignoring all bits in the result beyond the n-th place, i.e., the bit with weight 2^{n-1}. This procedure is also used when $x < 0$ to recover the encoding of $-x \geq 0$. What is the relationship between the binary encoding of $-2^{n-1} \leq x \leq 2^{n-1} - 1$ and the binary encoding of $-x$ in terms of the number of bits n?

1.1.b. Show that simple addition modulo 2^n on the encoded patterns is identical to integer addition (subtraction) for $-2^{n-1} \leq x,\ y \leq 2^{n-1} - 1$. You may ignore results that are out of range, i.e., overflow.

1.1.c. Show how overflow in addition (subtraction) can be detected efficiently.

1.1.d. Multiplying an unsigned binary number by 2 or 1/2 corresponds to shifting the binary representation left and right respectively (a so-called logical shift). Show how multiplying signed integers encoded via 2's complement representation by 2 or 1/2 can be done via a shifting operation (an arithmetic shift).

Problem 1.2

This problem considers the roots of the quadratic equation with a single parameter $\beta > 1$

$$x^2 + 2\beta x + 1.$$

Define the vector-valued function that maps β to the two roots $x_+(\beta)$ and $x_-(\beta)$

$$f : \mathbb{R} \to \mathbb{R}^2, \quad \beta \mapsto \begin{pmatrix} x_+(\beta) \\ x_-(\beta) \end{pmatrix} = \begin{pmatrix} -\beta + \sqrt{\beta^2 - 1} \\ -\beta - \sqrt{\beta^2 - 1} \end{pmatrix}$$

1.2.a What happens to the roots as $\beta \to \infty$?

1.2.b For $\beta > 1$, consider the derivatives with respect to β of each of the roots and derive an approximation of the relative condition number for the vector of roots $f(\beta)$ when β is perturbed slightly. Note that since f is a vector-valued function a vector norm must be used. For your analysis use the standard Euclidean 2-norm, i.e.,

$$v = \begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix} \to \|v\|_2 = \sqrt{\nu_1^2 + \nu_2^2}.$$

to measure the size of the solution and error vectors in $\mathbb{R}^2$. The absolute value be used as the norm of the scalars β and $\Delta\beta$ in $\mathbb{R}$. That is we have

$$\frac{\|f(\beta + \Delta\beta) - f(\beta)\|_2}{\|f(\beta)\|_2} \leq \kappa_{rel} \frac{|\Delta\beta|}{|\beta|}.$$

1.2.c Use the condition number to explain the conditioning of the vector of roots for $\beta > 1$, i.e., is it well-conditioned anywhere on the interval, is it ill-conditioned anywhere on the interval?

Problem 1.3

The evaluation of

$$f(x) = x\left(\sqrt{x+1} - \sqrt{x}\right)$$

encounters cancellation for $x \gg 0$.

Rewrite the formula for $f(x)$ to give an algorithm for its evaluation that avoids cancellation.

Problem 1.4

Consider the summation $\sigma = \sum_{i=1}^{n} \xi_i$ using the following "binary fan-in tree" algorithm described below for $n = 8$ but which clearly generalizes easily to $n = 2^k$:

$$\sigma = \{[(\xi_0 + \xi_1) + (\xi_2 + \xi_3)] + [(\xi_4 + \xi_5) + (\xi_6 + \xi_7)]\}$$

or equivalently

$$\sigma_j^{(0)} = \xi_j, \; 0 \leq j \leq 3$$
$$\sigma_0^{(1)} = \xi_0 + \xi_1, \quad \sigma_1^{(1)} = \xi_2 + \xi_3, \quad \sigma_2^{(1)} = \xi_4 + \xi_5, \quad \sigma_3^{(1)} = \xi_6 + \xi_7$$

$$\sigma_0^{(2)} = \sigma_0^{(1)} + \sigma_1^{(1)}, \qquad \sigma_1^{(2)} = \sigma_2^{(1)} + \sigma_3^{(1)}$$
$$\sigma = \sigma_0^{(3)} = \sigma_0^{(2)} + \sigma_1^{(2)}$$

In general, there will be $k = \log n$ levels and $\sigma = \sigma_0^{(k)}$. Level i has 2^{k-i} values of $\sigma_j^{(i)}$, each of which corresponds to a sum

$$\sigma_j^{(i)} = \xi_{2^i j} + \ldots + \xi_{2^i (j+1)-1}.$$

The algorithm is easily adaptable to n that are not powers of 2.

1.4.a. Derive an expression for the absolute forward error of the method for a fixed $n = 8$ or $n = 16$ and then generalize to $n = 2^k$.

1.4.b. Derive an expression for the absolute backward error of the method for $n = 8$ or $n = 16$ then generalize to $n = 2^k$.

1.4.c. Bound the errors and discuss stability relative to the simple sequential summation algorithm given by, for $n = 8$ but easily generalizable to any n,

$$\sigma = (((((((\xi_1 + \xi_2) + \xi_3) + \xi_4) + \xi_5) + \xi_6) + \xi_7) + \xi_8)$$

or equivalently

$$\sigma = \xi_1$$

$$\sigma \leftarrow \sigma + \xi_i, \quad i = 2, \ldots, 8$$