# Department of Computer Science
# CPSC 302 Midterm Examination (U. Ascher)

Oct. 17, 2001. NAME:____________________________

Number of pages: 5 Signature:____________________________

Time: 50 minutes STD. NUM: ____________________________

You are permitted a calculator and one $8\frac{1}{2} \times 11$ sheet of handwritten notes to assist you in answering the questions.

For the short answer questions, be as concise as possible. The weight of each question is given in parentheses. The total number of marks is 100 (approximately 2 marks/min). If you run out of space for a question, use the reverse side. Show all your work.

**Good Luck!**

| Q1 | Q2 | Q3 | Q4 | TOTAL |
|---|---|---|---|---|
| | | | | |
| 10 | 30 | 30 | 30 | 100 |

---

1. **[10 marks]** How many distinct positive numbers can be represented in a floating point system using base $\beta = 10$, precision $t = 2$ and exponent range $L = -9$, $U = 10$?

   (Assume normalized fractions and don't worry about underflow.)

2. Consider the polynomial function

$$\begin{aligned} f(x) &= (x-2)^9 \\ &= x^9 - 18x^8 + 144x^7 - 672x^6 + 2016x^5 - 4032x^4 + 5376x^3 - 4608x^2 + 2304x - 512. \end{aligned}$$

I wrote a MATLAB script which evaluated this function at 161 equidistant points in the interval $[1.92, 2.08]$ using two methods:

(a) Apply Horner's method for evaluating the polynomial at the right hand side; recall that for a polynomial $p(x) = \sum_{i=0}^{n} a_i x^i$, Horner's method for evaluating $p$ at a given value of $x$ reads

$$\begin{aligned} &p = a_n; \\ &for\ i = 1:n \\ &\quad p = p * x + a_{n-i} \end{aligned}$$

(b) Calculate $(x-2)^9$ directly.

The results are plotted in Figure 1.

(a) **[15 marks]** Explain the difference between the two graphs in *one short paragraph.*

(b) **[15 marks]** Suppose you were to apply the bisection routine from the course notes (or from Assignment 2) to find a root of this function, starting from the interval $[1.92, 2.08]$ and using the Horner evaluation method, to an absolute tolerance $10^{-6}$. Circle the correct outcome:

i. The routine will terminate with a root $p$ satisfying $|p-2| \leq 10^{-6}$.
ii. The routine will terminate with a root $p$ *not* satisfying $|p-2| \leq 10^{-6}$.
iii. The routine will not find a root.

Justify your choice in one short sentence. *Note: No justification, or two justifications, produce no marks.*

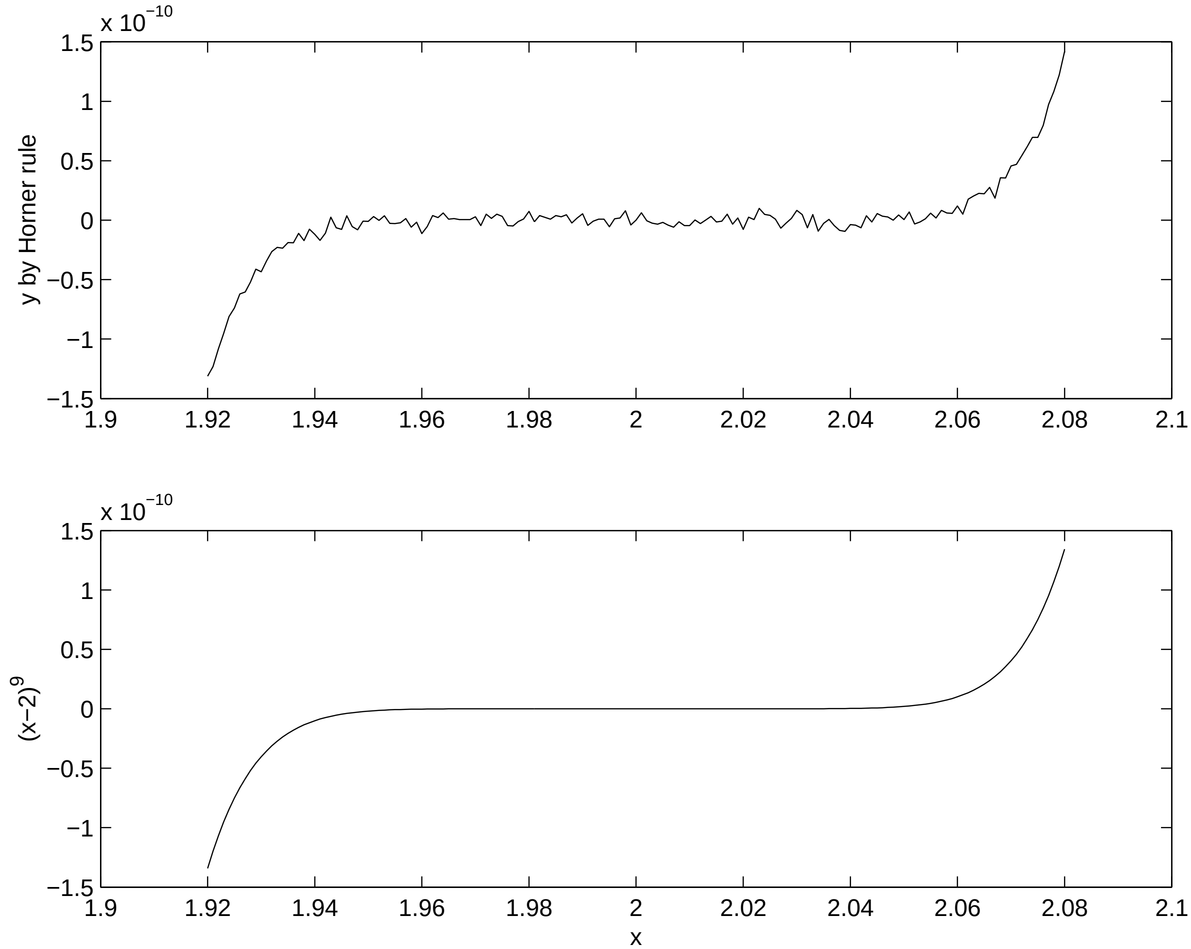


Figure 1: Graphs of the function of Question (1b).

3. **[30 marks]** Consider finding the root of a given nonlinear function $f(x)$, known to exist uniquely in a given interval $[a, b]$, using one of the following three methods: *Bisection, Newton* and *secant*. For each of the following three instances of $f(x)$, one of these methods has a distinct advantage over the other two. Match problems and methods and justify briefly.

   (a) $f(x) = x - 1$ on the interval $[0, 1.8]$.

   (b) $f(x)$ is given in Figure 2 on $[0, 4]$.

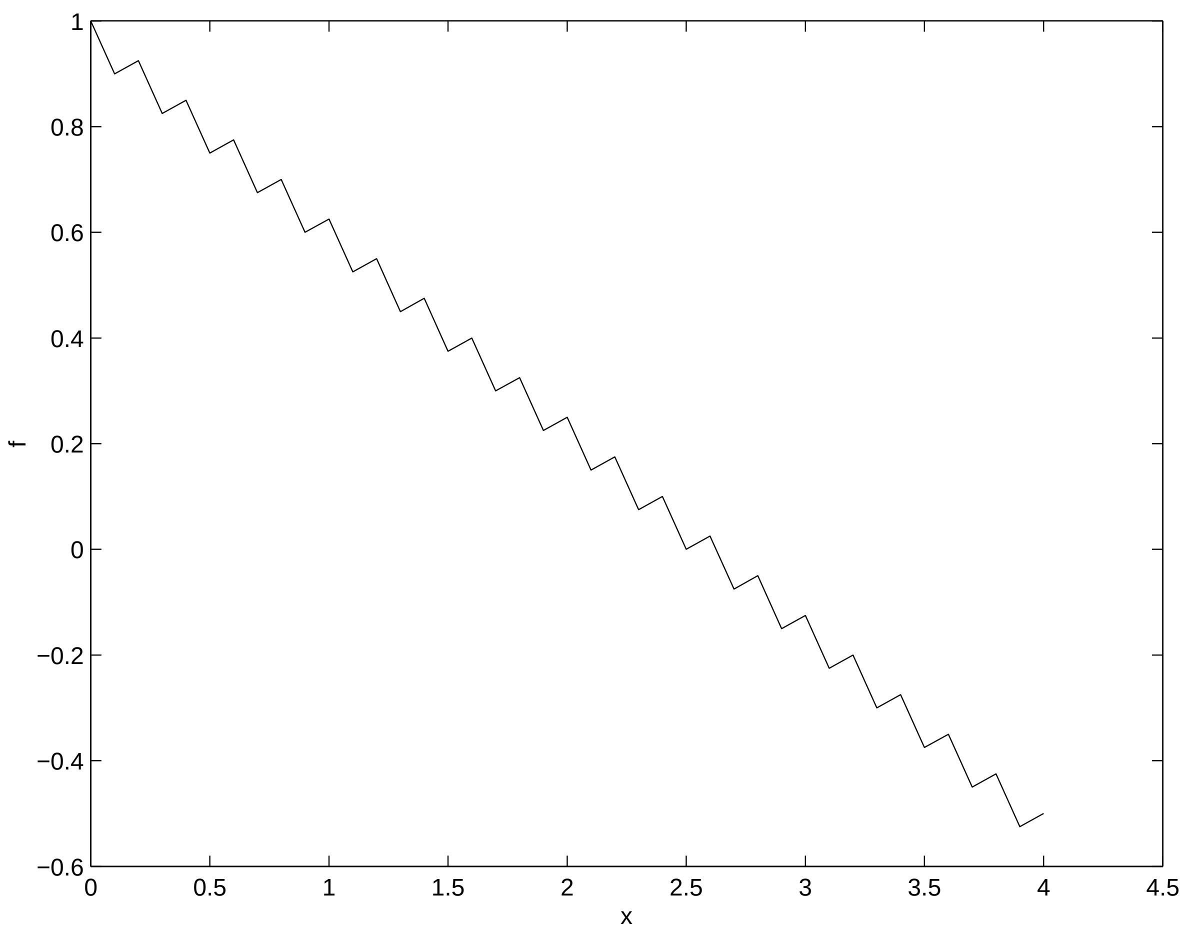


Figure 2: Graph of the function of Question (3b).

   (c) $f \in C^5[0.1, 0.2]$, the derivatives of $f$ are all bounded in magnitue by 1, and $f'(x)$ is hard to specify explicitly or to evaluate.

4. **[30 marks]**

Let $A$ and $T$ be two nonsingular, $n \times n$ real matrices. Furthermore, suppose we are given two matrices $L$ and $U$ such that $L$ is unit lower triangular, $U$ is upper triangular, and

$$TA = LU.$$

Devise an algorithm that will solve the problem

$$A\mathbf{x} = \mathbf{b}$$

for any given vector $\mathbf{b}$ in $O(n^2)$ floating point operations. First explain briefly yet clearly why your algorithm requires only $O(n^2)$ flops (you may assume without proof that solving an upper triangular or a lower triangular system requires only $O(n^2)$ flops). Then specify your algorithm in detail (including the details for lower and upper triangular systems) using pseudocode or a MATLAB script.