Math 230 — Hour Exam 2

November 23, 2015

General Directions. This is an open-book, open-notes, open-computer test. However, you may not communicate with any person, except me, during the test. You have the full class period (50 minutes) in which to do the test. Put your answer to each question in the space provided (use the backs of pages if you need more space). Be sure to **show your work**! I give partial credit for incorrect answers if you show correct steps leading up to them; conversely, I do not give full credit even for correct answers if it is not clear that you understand where those answers come from. Good luck.

This test contains 4 questions on 4 pages.

Question 1 (10 Points). If **V** is a vector containing n numbers, define the i^{th} partial sum of **V** to be the sum of the numbers in positions i through n of **V**. More formally, the i^{th} partial sum of **V** is

$$\sum_{j=i}^{n} \vec{V}_j$$

Assuming that `V`, `n`, and `i` are variables that have already been assigned values, write a Matlab `for` loop, and any necessary accompanying code, to compute the i^{th} partial sum of **V**. Your code should leave the sum in a variable named `partialSum`.

Question 2 (15 Points). Suppose that $f(x)$ is a function, and that x_l and x_u are two numbers such that $x_l < x_u$, $f(x_l) < 0$, $f(x_u) > 0$ and f is continuous between x_l and x_u. Then you can crudely estimate where $f(x) = 0$ by stepping through the interval between x_l and x_u in steps of some fixed size (the size determining how accurately you approximate the root), stopping as soon as you see a value of $f(x) \geq 0$. More specifically, if n is the number of steps you want, you would compute Δx as $(x_u - x_l) / n$, then test whether $f(x_l + \Delta x) \geq 0$. If it is, stop, $x_l + \Delta x$ is your estimate of the root. Otherwise go back and test whether $f(x_l + 2\Delta x) \geq 0$, stop if so, otherwise try $f(x_l + 3\Delta x)$, and so forth until you finally find a value at which $f(x) \geq 0$ (which you know you will, since $f(x_u) > 0$ and $x_l + n\Delta x = x_u$).

(As an example, suppose $f(x) = x - 1$, $x_l = 0$, $x_u = 2$, and $n = 3$. Then Δx would be 2/3, your first test would be $f(2/3)$ which is less than 0, so you would try again at $f(4/3)$, find that it is greater than 0, and so return 4/3 as the estimated root.)

Here is the header for a Matlab function that takes f, x_l, x_u, and n as arguments. Write a body for this function so that it computes and returns an estimate of a root of f, using the algorithm outlined above. You may assume that `f`, `xl`, `xu`, and `n` meet the requirements given for them above; do not include code to check those requirements in your answer.

```
function [root] = stepOff( f, xl, xu, n )
```

Question 3 (15 Points). Recall the function we developed to take another function, *f*, and an *x* value as arguments, and to approximate $f'(x)$ by simulating the limit definition using a small but non-zero *h*. We used a constant *h* in that function, but ideally *h* would depend on *x*, since "small" is a relative notion. Here is a version of the derivative function based on this idea:

```
function [y] = derivative( f, x )
   h = x * 1e-9;
   y = ( f(x+h) – f(x) ) / h;
end
```

The only problem with this definition is that if *x* is 0, *h* will also be 0 and the calculation of *y* will fail, trying to divide 0 by 0. Write a variation on the `derivative` function given above that normally sets `h` to `x` × 10^{-9} (as the above version does), but that sets `h` to the constant 10^{-30} if `x` is between -10^{-20} and $+10^{-20}$.

Question 4 (10 Points). Suppose **V** is a vector of length 3. Write a Matlab logical expression that evaluates to true exactly when each element of **V** is less than or equal to the element after it, and at least 2 of the elements are equal to each other. Your expression should evaluate to false in all other cases.

(For example, your expression would evaluate to true for the vectors [1, 1, 5], [1, 5, 5], and [5, 5, 5] but would be false for [1, 2, 3] and [3, 3, 0].)