401 HW 2

James Oswald

September 2020

2.1 Continuity, differentiability review

A) Give an example of a function $f : \mathbb{R} \to \mathbb{R}$ that is continuous everywhere on its domain but has at least one point at which it is not differentiable.

Using the definition of a continuous function on the real numbers and the definition of a function being differentiable at a point:

A function f is continuous on the real numbers iff:

$$\forall x \in \mathbb{R} : f(x) \in \mathbb{R} \wedge \lim_{n \to x} f(n) = f(x)$$

Or put more simply, the function is connected with no gaps in its plot. A function f is differentiable at a point x iff:

$$f'(x) \neq DNE := \lim_{h \to 0} \frac{f(p+h) - f(p)}{h} \neq DNE$$

Which basically just states that the derivative of f exists at x, or more graphically, the plot of f is "Smooth" at x. Using these two definitions, we can easily prove such a function exists just by thinking about the graphs of functions that would satisfy these conditions. We're looking for a connected function with a point where the curve isn't smooth. The first function that comes to mind is $f(x) = |x|$ since thinking about it visually fulfills all the requirements. It is fully connected and therefore continuous, but has no derivative at 0. The proof for continuity is trivial, we know $|x|$ is fully connected since at all points (even 0) the limit of f at n as $n \to x$ is the same as $f(x)$. We can also prove that $|x|$ is not differentiable at x by plugging into the definition.

$$\lim_{h \to 0} \frac{|0+h| - |0|}{h} = \lim_{h \to 0} \frac{|h|}{h}$$

We see that by splitting this up into the left and right limits, we get it approaching -1 from the left limit and 1 from the right limit, therefore the limit does not exist, and the function is not differentiable at 0.

B) Give an example of a function $f : \mathbb{R} \to \mathbb{R}$ that is not continuous at 0.

Since we're allowed to use a piecewise function, we can easily just define a piecewise function that is explicitly split at 0, for this I will use:

$$f(x) = \begin{cases} 1 & x \leq 0 \\ 0 & x > 0 \end{cases}$$

intuitively we know this is discontinuous due to the large gap in its plot, however I will also formalize this by showing it violates the formal definition of continuity of a function at a point provided in part (A). A function f is continuous on the real numbers iff:

$$\forall x \in \mathbb{R} : f(x) \in \mathbb{R} \wedge \lim_{n \to x} f(n) = f(x)$$

We see that at x = 0:

$$\lim_{n \to 0} f(n) = f(0)$$

$f(0) = 1$ but the limit $\lim_{n\to 0} f(n)$ does not exist since approaching from the left yields the limit to be 1 while approaching from the right yields the limit to be 0. Therefore $\lim_{n\to 0} f(n) \neq f(0)$ and the function is not continuous at 0.

2.1.1 Floating point and related topics

A) Write down the binary expansion of 50.5.

I assume this means the IEEE-754 Single Precision floating point expansion. Following the conversion algorithm, I start by converting the whole part, 50 to binary 110010. I then convert the decimal portion .5 to binary 1. I then combine them into 110010.1 which I can then convert to binary scientific notation $1.100101 \cdot 2^5$ The exponent for a single precision float will be the exponent from the sci notation plus 127 so, $5 + 127 = 132$ which in binary will be 10000100. Our mantissa will be the bits after the decimal from our scientific notation form 100101, padded by 0s at the end. Finally we put it all together, 1 sign bit, positive in this case (so 0), followed by 8 bits of exponent and the 23 mantissa bits. Using + here to represent concatenation the I get a final answer of:

0+10000100+10010100000000000000000 = 01000010010010100000000000000000

B) Consider the gaps between representable floating point numbers. Which of the following has the larger gap between it and the next larger representable floating point number? 2 or 201

It should be obvious from the way floating points are stored that 201 has a larger gap than 2 between the next floating point number, since as you gain exponent you loose room for precision in the mantissa which ties in with the fun fact that half of all representable floating point numbers are in the range [-1, 1] and you get less and less precise the further you go out. Just to prove this, I've brute forced the calculations:

Converting 2 to a float, we get 01000000000000000000000000000000, the next largest representable float will be 01000000000000000000000000000001 which is 2.0000002384185791015625 in decimal. Converting 201 to a 32 bit IEEE 754 float we get 01000011010010010000000000000000, the next largest representable float will be 01000011010010010000000000000001 which is 201.0000152587890625 in decimal. Comparing the distances we get .0000002384185791015625 < .0000152587890625 which proves 201 has a larger gap between it and the next representable number.

2.2 Bisection search

I've taken the liberty of not copying questions for coding questions. See attached Matlab file hw223.m for implementation.
Code for all 3 parts:

Editor - C:\Users\James\Desktop\program\School\ICSI-401-Numerical-Methods\Homewo

hw223.m

```
%2.3 Part 2
F1 = @(x) x^5 + x + 1;
%using the formula for digits of precision from 2.2
disp(bisectionSearch(F1, -1, 1, 5*10^(-4-1)));

%2.3 Part 3
F2 = @(x) sin(x);
disp(bisectionSearch(F2, -pi/2, 5*pi/2, 5*10^(-4-1)));

%2.3 Part 1
function z = bisectionSearch(F, a, b, delta)
    %IVT Check, F(a) and F(b) have seperate signs,
    %IE there must be a 0 between a and b
    if F(a) * F(b) > 0
        z = NaN;
        return;
    end
    %precision check to see if we're acceptably close
    %to the root
    while abs(a - b) > delta
        %Finding the midpoint
        z = (a + b) / 2;
        %checking in which dir we overshot
        if F(a) * F(z) < 0
            b = z;
        else
            a = z;
        end
    end
end
```

Output For part 2 and 3:

```
>> diary
>> hw223
   -0.7549


   1.1984e-05


>> diary off
>>
```

For part 2, We graph F1 to see that it indeed find the correct root. For part 3, we see it found the root at 0, but was stopped by my selection of delta before getting too close.

2.3 Newton's method

Part 1 See attached Matlab file hw224.m file for implementation.

```
Editor - C:\Users\James\Desktop\program\School\ICSI-401-Numerical-Methods\Homework2\hw224.m
hw223.m  hw224.m  +

%Testing corectness of part 1 against an online result I found from
%https://www.whitman.edu/mathematics/calculus_online/section06.03.html
F1 = @(x) x^2-3;
F1p = @(x) 2*x;
disp(newtonsMethod(F1, F1p, 2, 2));

%2.4 Part 1
function wk = newtonsMethod(F, Fprime, w0, k)
    iter = 0;
    wk = w0;
    while iter <= k
        wk = wk - F(wk) / Fprime(wk);
        iter = iter + 1;
    end
end
```

Diary file output for part 1.

```
>> diary diary224.txt
>> hw224
    1.7321


>> diary off
```

I was lucky to be able to test my code out using a sample problem I found online, and get what appears to be a correct result with those parameters.

Part 2 Suppose that you want to find the points at which the graphs of two functions $f(x)$ and $g(x)$ intersect. Write down a function $H(x)$ such that $f(x)$ and $g(x)$ intersect at a point z if and only if $H(z) = 0$.

The function $H(x) = f(x) - g(x)$ will equal 0 exclusively where $f(x)$ and $g(x)$ intersect. This is because whenever $f(x)$ and $g(x)$ intersect, $f(x) = g(x)$ which implies $f(x) - g(x) = 0$ which is exactly what we need as our condition for $H(x)$.

Part 3 Find the point of intersection of $f(x) = x$ and $g(x) = x\log(x)$:

- Write down a function $H(x)$ as above, such that the roots of $H(x)$ are exactly the points of intersection of x and $x\log(x)$

  Using the function from above we get: $H(x) = x - x\log(x)$

- Write down the Newton iteration equation (i.e., w_{k+1} in terms of wk, $H(x)$, and $H'(x)$) that you would use to find roots of $H(x)$.

$$w_{k+1} = w_k - \frac{H(w_k)}{H'(w_k)}$$

- Use your Matlab implementation of Newton's method with $k = 50$ and some appropriate starting point w_0 to find an approximation wk to the root x_* of $H(x)$. In order to choose w_0, you can plot $H(x)$ and identify a point visually close to x_*.

  I visually Identified the point 5 as being close to the true root x_* I plugged in with $k = 50$ and ran the experiment.

```
%2.4 Part 3
F1 = @(x) x - x*log(x);
F1p = @(x) -log(x);
disp(newtonsMethod(F1, F1p, 5, 50));

%2.4 Part 1
function wk = newtonsMethod(F, Fprime, w0, k)
    iter = 0;
    wk = w0;
    while iter <= k
        wk = wk - F(wk) / Fprime(wk);
        iter = iter + 1;
    end
end
```

Which yielded what looks like the correct result

```
>> diary diary2243.txt
>> hw224
    2.7183

>> diary off
```