CSC338. Homework 7

Due Date: Wednesday March 11, 9pm

Please see the guidelines at https://www.cs.toronto.edu/~lczhang/338/homework.html

What to Hand In

Please hand in 2 files:

- Python File containing all your code, named `hw7.py`.
- PDF file named `hw7_written.pdf` containing your solutions to the written parts of the assignment. Your solution can be hand-written, but must be legible. Graders may deduct marks for illegible or poorly presented solutions.

If you are using Jupyter Notebook to complete the work, your notebook can be exported as a .py file (File -> Download As -> Python). Your code will be auto-graded using Python 3.6, so please make sure that your code runs. There will be a 20% penalty if you need a remark due to small issues that renders your code untestable.

Make sure to remove or comment out all matplotlib or other expensive code before submitting your homework!

Submit the assignment on **MarkUs** by 9pm on the due date. See the syllabus for the course policy regarding late assignments. All assignments must be done individually.

```
import math
import numpy as np
```

Question 1.

Part (a) – 4 pt

Consider the problem of finding the roots of the functions f_1, f_2, f_3 and f_4. What is the (absolute) condition number of each problem?

1. $f_1(x) = sin(\frac{x}{100})$, at $x = 0$
2. $f_2(x) = x^3 - 5x^2 + 8x - 4$, at $x = 2$
3. $f_3(x) = xcos(20x) - x$, at $x = 0$
4. $f_4(x) = x^3$, at $x = 0$

Include your solution in your PDF writeup. Show your work.

Part (b) – 6 pt

What is the convergence rate of each of the following sequences? Your answer should be either "linear", "superlinear but not quadratic", "quadratic", "cubic", or "something else". Include your solution in your PDF writeup. Show your work.

1. $x_n = 10^{-2n}$
2. $x_n = 2^{-n^2}$
3. $x_n = 2^{-n \log n}$

Question 2. Interval Bisection

Part (a) – 4 pt

Write a function `bisect` that returns a list of intervals where the root of the function `f(x)` lies. Each interval should be half the size of the previous, and should be obtained using the interval bisection method.

```
def bisect(f, a, b, n):
    """Returns a list of length n+1 of intervals
    where f(x) = 0 lies, where each interval is half
    the size of the previous, and is obtained using
    the interval bisection method.

    Precondition: f continuous,
                  a < b
                  f(a) and f(b) have opposite signs

    Example:
    >>> bisect(lambda x: x - 1, -0.5, 2, n=5)
    [(-0.5, 2),
     (0.75, 2),
     (0.75, 1.375),
     (0.75, 1.0625),
     (0.90625, 1.0625),
     (0.984375, 1.0625)]
    """
```

Part (b) – 4 pt

During lecture, we compared the following two computations of the midpoint between floating-point numbers a and b:

```
a = 5
b = 5.1

mid1 = a + (b - a) / 2
mid2 = (a + b) / 2
```

Using a toy floating-point system, we saw in class that `mid2` can be outside of the range of [a, b].

Use the same idea to construct two floating-point values `float_a` and `float_b` where `(float_a + float_b) / 2` is not bewteen `float_a` and `float_b`. Recall that Python uses IEEE Double Precision for floating-point arithmetic (i.e., round to even in base 2 with a mantissa of 53 bits).

Discuss how you arrived at the answer in your PDF writeup.

```
float_a = 0
float_b = 0
```

Part (c) – 2pt

Suppose you would like to use the interval bisection method to find the root of a function $f(x)$, starting with an interval (a, b).

What is the minimum number of interval bisection iterations necessary to guarantee that your estimate of a root is accurate to 10 decimal places?

Include your solution in your PDF writeup.

Question 4. Fixed-Point Iteration

Part (a) – 4 pt

Write a function `fixed_point` to find the fixed-point of a function `f` by repeated application of `f`. The function should return a list of values `[x, f(x), f(f(x)), ...]`.

```
def fixed_point(f, x, n=20):
    """ Return a list lst = [x, f(x), f(f(x)), ...] with
    `lst[i+1] = f(lst[i])` and `len(lst) == n + 1`

    >>> fixed_point(lambda x: math.sqrt(x + 1), 3, n=5)
    [3,
     2.0,
     1.7320508075688772,
     1.6528916502810695,
     1.6287699807772333,
     1.621348198499395]
    """
```

Part (b) – 6 pt

To find a root of the equation

$$f(x) = x^2 - 3x + 2 = 0$$

we can consider fixed-point problems involving the following different functions:

1. $g_1(x) = \frac{x^2+2}{3}$
2. $g_2(x) = \sqrt{3x - 2}$
3. $g_3(x) = 3 - \frac{2}{x}$

Use the `fixed_point` function to generate the fixed-point iterations for each of these functions. Choose the starting value of x to be $x_0 = 3$.

Does fixed-point iteration converge for each of the functions? Include the output of your call to `fixed_point` in your PDF writeup. If the iteration converges, what is the approximate the convergence rate of convergence? (linear, superliner, quadratic, etc).

Include your solution in your PDF writeup.