Exam 2 Practice Problems

This is a list of problems that will help you prepare for Exam 2. They are optional and will not be graded.

Root Finding

1. (a) Use the bisection method to approximate where the functions $y = x^3$ and $y = 3x^2 - 1$ intersect. Calculate x_4 of the bisection method with an initial interval of $[0, 2]$. Here, $x_0 = 1$ is the initial midpoint.
   Answer: $x_4 = \frac{11}{16} = 0.6875$

   (b) How many steps of the bisection method are needed to determine the root with an error of at most $(1/2) * 10^{-8}$.
   hint: $\frac{\ln(10^{-8})}{\ln(2)} = -26.575$, *when rounded to three decimal places.*
   Answer: $n = 28$

2. (a) Use the bisection method to approximate the root of the function $f(x) = x^3 - 9x + 2$ that is on the interval $[1, 5]$. Calculate x_4 of the bisection method with an initial interval of $[1, 5]$. Here, $x_0 = 3$ is the initial midpoint.
   Answer: $x_4 = \frac{23}{8} = 2.875$

   (b) How many steps of the bisection method are needed to determine the root with an error of at most $(1/2) * 10^{-10}$.
   hint: 1.) $\ln(4) = \ln(2^2) = 2\ln(2)$
   hint: 2.) $\frac{\ln(10^{-10})}{\ln(2)} = -33.219$ *rounded to three decimal places.*
   Answer: $n = 36$

3. Problem 5 from section 3.2
   Answer: $x_{n+1} = \frac{2Rx_n}{x_n^2 + R}$ **Answer:** $x_5 = 4.999976821853597$ Newtons method problems on the exam will be written so that a calculator is not needed.

4. Problem 15 from section 3.2
   Answer: $x_1 = \frac{1}{2}$

5. Use Newton's method to approximate the root of $f(x) = x^3 - 9x + 2$ with an initial guess $x_0 = 4$. Compute x_4, which is the fourth iteration of Newton's method.
   Answer: $x_4 = 2.8820215337441586$ Newtons method problems on the exam will be written so that a calculator is not needed.

6. **Answer:** The solution for this problem is given in the lecture notes from 3-8

   (a) Verify that when Newton's method is used to compute $\sqrt[3]{R}$ (by solving the equation $x^3 = R$), the sequence of iterates is defined by

$$x_{n+1} = \frac{1}{3}\left(2x_n + \frac{R}{x_n^2}\right)$$

   (b) Fill in lines 3 and lines 7 of the code below so that when the code is run, it will print off the 10th iteration of Newton's method described in part a.

Code 1:

```
1 R = 5
2 f = lambda x: x**3 - R
3 df = lambda x: #fill in the code
4 n = 10
5 x_old = 3
6 for i in range(n):
7     x_new = #fill in code
8     x_old = x_new
9 print(x_new)
```

   (c) In the code from part b, x_new represents the nth iteration of newtons method x_n. Modify the code from part b to print the approximate root from Newton's method x_n, such that $|f(x_n)| < 10^{-8}$. *Hint: One way to do this is with a while loop.*

Polynomial Interpolation

7. Section 4.1 problem 3
   Answer:

$$\ell_0(x) = -\frac{1}{40}(x-1)(x-3)(x-4)$$
$$\ell_1(x) = \frac{1}{12}(x+1)(x-3)(x-4)$$
$$\ell_2(x) = -\frac{1}{8}(x+1)(x-1)(x-4)$$
$$\ell_3(x) = \frac{1}{15}(x+1)(x-1)(x-3)$$

8. Section 4.1 problem
   Answer: Verify that $p(x)$ and $q(x)$ interpolate the data. For example the verification for the first data point with $p(x)$ is: $p(1) = 5(1)^3 - 27(1)^2 + 45(1) - 21 = 5 - 27 + 45 - 21 = 50 - 48 = 2$
   Answer: This does not violate the uniqueness part of the theorem since $q(x)$ is an order 4 polynomial, and the interpolating polynomial of degree 3 or less is unique.

9. Section 4.1 problem 9
   Answer: The answer for this problem is given in the textbook.

10. Section 4.1 problem 10a
    Answer:

$$\begin{array}{cccccc}
0 & 7 & & & \\
 & & 2 & & \\
2 & 11 & & 5 & \\
 & & 17 & & 1 \\
3 & 28 & & 9 & \\
 & & 35 & & \\
4 & 63 & & &
\end{array}$$

$$p_3(x) = 7 + 2x + 5x(x-2) + 1x(x-2)(x-3)$$

11. Section 4.1 problem 16 *start with* $p(x) = a_0 + a_1(x-0) + a_2(x-0)(x-1)$. *Find* a_0 *and* a_1, *and then find* $p'(x)$ *and use* $p'(\alpha) = 2$ *to solve for* a_2 *in terms of* α.
    Answer: The solution for this problem is given in the lecture notes from (3-8).

Numerical Differentiation

12. Section 4.3 problem 2
    Answer: The answer to this problem is given in the textbook

13. Section 4.3 problem 8a
    Answer: The answer to this problem is given in the textbook

14. Consider the table of values:

x	-1	0	1	2	3
$f(x)$	$\frac{1}{3}$	1	3	9	27

The code below uses the forward difference method to approximate $f'(-1)$, $f'(0)$, $f'(1)$, and $f'(2)$.

Code 2:

```
import numpy as np
x = np.array([-1,0,1,2,3.])
y = np.array([1/3,1,3,9,27])
n = x.shape[0]
forward_diff = np.zeros(n)
forward_diff[:n-1] = (y[1:] - y[:n-1])/(x[1:]-x[:n-1])
backward_diff = np.zeros(n)
central_diff = np.zeros(n)
```

(a) Assuming this code is run in python, write the variable forward_diff as python output or as a vector.
Answer: [2/3,2,6,18,0]

(b) Add the line of code that uses array slicing along with backwards difference formula to approximate the values $f'(0)$, $f'(1)$, $f'(2)$, and $f'(3)$ and assign them to the last four entries of the variable backward_diff.
Answer: backward_diff [1:] = (y[1:] − y[:n−1])/(x[1:] − x[:n−1])

(c) Write out code that uses a for loop along with the central difference formula to approximate the values $f'(0)$, $f'(1)$, $f'(2)$ and assigns them to the middle three elements of the variable central_diff . (This can be written in 2 lines of code.)
Answer:

Code 3:

```
for i in range(1,n-1):
    central_diff[i] = (y[i+1]-y[i-1])/(x[i+1]-x[i-1])
```

Numerical Integration

15. Section 5.1 problem 12
Answer: $\frac{1}{4}10^{-4}$

16. Approximate the integral

$$\int_1^2 \frac{dx}{x^2}$$

with the left-hand rule, right-hand rule, midpoint rule, and trapezoid rule. Use the partition

$$P = \left\{1, \frac{4}{3}, \frac{5}{3}, 2\right\}$$

.

Answer: LHR: $\frac{769}{1200}$, RHR: $\frac{469}{1200}$, MPR: $\frac{78796}{160083}$, TR: $\frac{619}{1200}$

17. Section 5.1 problem 19
Answer: 2.5×10^{-5}

18. Determine the smallest number of sub-intervals n that guarantees an error of $\epsilon = 10^{-5}$ when the trapezoid rule is used to approximate the integral

$$\int_1^2 \frac{1}{1+x}\,\mathrm{d}x.$$

hint: $\sqrt{\frac{1}{48\times 10^{-5}}} = 40.825$ *rounded to three decimal places*
Answer: The solution to this is given in the notes from 3-8

19. **Answer:** The solution to this is given in the notes from 3-8
The left-hand rule with even spacing is:

$$\int_a^b f(x)\,\mathrm{d}x \approx h\sum_{i=0}^{n-1} f(x_i)$$

The following python code contains a function that returns the left-hand rule approximation given the arguments f the function $f(x)$, a the lower bound of the integral, b the upper bound of the integral, and n the number of sub-intervals:

Code 4:

```python
import numpy as np
def left_hand_rule(f,a,b,n):
    x = np.linspace(a,b,n+1)
    h = (b-a)/n
    sum = 0
    for i in range(n):
        sum = sum + f(x[i])
    sum = h*sum
    return sum
f = lambda x: (x+1)**2
a = -1
b = 1
n = 4
LHR = left_hand_rule(f,a,b,n)
print(LHR)
```

(a) If this code is run in python, what is printed?

(b) Replace lines 6, 7, and 8 that corresponds to using the right-hand rule to approximate the integral. Recall the right hand rule:

$$\int_a^b f(x)\,\mathrm{d}x \approx h \sum_{i=0}^{n-1} f(x_{i+1}).$$

(c) If the code with the replacement from part b is run in python, what is printed?