Numerical Analysis HW #1

Ozaner Hansha

February 12, 2019

Problem 1

Consider the following function:

$$f(x) = \frac{1 - \cos x}{x^2}$$

Part a

Problem: Prove the following:

$$\lim_{x \to 0} f(x) = \frac{1}{2}$$

Solution: First let us plug the Taylor expansion of $\cos x$ into $f(x)$:

$$\begin{aligned}
f(x) &= \frac{1 - \cos x}{x^2} \\
&= \frac{1 - (1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots)}{x^2} \\
&= \frac{\frac{x^2}{2!} - \frac{x^4}{4!} + \frac{x^6}{6!} - \cdots}{x^2} \\
&= \frac{1}{2!} - \frac{x^2}{4!} + \frac{x^4}{6!} - \cdots
\end{aligned}$$

The limit as x approaches 0 of this series becomes clear:

$$\begin{aligned}
\lim_{x \to 0} f(x) &= \lim_{x \to 0} \frac{1 - \cos x}{x^2} \\
&= \lim_{x \to 0} \left(\sum_{n=1}^{\infty} \frac{x^{2n-2}}{(2n)!} (-1)^{n+1} \right) \\
&= \lim_{x \to 0} \left(\frac{1}{2!} - \frac{x^2}{4!} + \frac{x^4}{6!} - \cdots \right) \\
&= \frac{1}{2!} - 0 + 0 - \cdots = \frac{1}{2}
\end{aligned}$$

Part b

Problem: Find $\alpha \in \mathbb{R}$ such that:

$$(\exists c \in \mathbb{R})\ \lim_{x\to 0} \frac{|f(x) - 1/2|}{x^\alpha} = c \neq 0$$

Solution: Recall that in part a, $f(x)$ was shown to be equal to:

$$\begin{aligned} f(x) &= \frac{1-\cos x}{x^2} = \sum_{n=1}^{\infty} \frac{x^{2n-2}}{(2n)!}(-1)^{n+1} \\ &= \frac{1}{2!} - \frac{x^2}{4!} + \frac{x^4}{6!} - \cdots \end{aligned}$$

And so:

$$\begin{aligned} \frac{|f(x) - 1/2|}{x^\alpha} &= \frac{\left|\left(\frac{1}{2!} - \frac{x^2}{4!} + \frac{x^4}{6!} - \cdots\right) - 1/2\right|}{x^\alpha} \\ &= \frac{\left|-\frac{x^2}{4!} + \frac{x^4}{6!} - \frac{x^6}{8!}\cdots\right|}{x^\alpha} \end{aligned}$$

At this point it should be obvious the lowest degree term (which has the largest value as $x \to 0$) is 2. This leaves us with $\alpha = 2$:

$$\lim_{x\to 0} \frac{|f(x) - 1/2|}{x^2} = \lim_{x\to 0} \frac{\left|-\frac{x^2}{4!} + \frac{x^4}{6!} - \frac{x^6}{8!}\cdots\right|}{x^2} = \lim_{x\to 0} \left|-\frac{1}{4!} + \frac{x^2}{6!} - \frac{x^4}{8!}\cdots\right| = \frac{1}{4!}$$

Part c

Problem: Write a Matlab program to calculate $f(x)$ and its error $|f(x) - 1/2|$ for the values $10^{-1}, 10^{-2}, \cdots, 10^{-8}$.

Solution:

x	$f(x)$	$\|f(x) - 1/2\|$
10^{-1}	4.9958e-01	4.1653e-04
10^{-2}	5.0000e-01	4.1667e-06
10^{-3}	5.0000e-01	4.1674e-08
10^{-4}	5.0000e-01	3.0387e-09
10^{-5}	5.0000e-01	4.1370e-08
10^{-6}	5.0004e-01	4.4450e-05
10^{-7}	4.9960e-01	3.9964e-04
10^{-8}	0	5.0000e-01

Part d

Problem: Why doesn't the error, and thus the result, converge in part c as it was proven to in part b?

Solution: As the table shows, the result starts to converge from iteration 1 to 4, but at the 5th iteration the error actually increases and it generally gets worse from there. This happens because the numbers involved in computing $f(x)$ become to small to represent in a double precision float. Consider $x = 10^{-8}$:

$$\cos(10^{-8}) = 0.99999999999999995 0\ldots$$

As expected, cos's value approaches 1 as its input approaches 0. Note, however, that the number of digits represented by a double precision float is about 16 ($2^{-53} \approx 10^{-16}$). Using truncation, this number is at the limit of not being rounded to 1:

$$fl(\cos(10^{-8})) = 0.9999999999999999$$

Assuming the above hasn't already been rounded to 1, when the next operation of the function takes place $(1 - \cos x)$ we have the following:

$$1 - 0.9999999999999999 = 1 \times 10^{-16}$$

Note that this value definitely runs into the machine epsilon, that is to say it is too small in magnitude to be represented by a double and so is instead represented by the nearest value 0. Now that the value of the numerator is 0, the true function and the calculation now differ completely:

$$\underbrace{\frac{1-\cos x}{x^2} = \frac{1 \times 10^{-16}}{(10^{-8})^2} \approx 0.5}_{\text{true}} \qquad \underbrace{\frac{fl(1-\cos x)}{x^2} = \frac{0}{(10^{-8})^2} = 0}_{\text{calculation}}$$

The problem is the same for $x = 10^{-5}, 10^{-6}$ and 10^{-7} just less pronounced because the round off error does not totally zero out the output like it does in the 10^{-8} case. Ultimately, if any of the intermediate calculations performed result in a value that is too small to precisely represented (i.e. has a large relative error) then that error will propagate to further steps, possibly rendering the entire calculation incorrect.

Problem 2

Consider the following function:

$$f(x) = x^3 + x - 4$$

Part a

Problem: Show that $f(x)$ has at least one root in the interval $[1, 4]$.

Solution: Recall the intermediate value theorem. That is, for any continuous function $f : [a, b] \to \mathbb{R}$, the following must be true:

$$(\forall y \in [\min(f(a), f(b)), \max(f(a), f(b))]) \ (\exists c \in (a, b)) \ y = f(c)$$

Since a root of f is simply a value c such that $f(c) = 0$, we can simply evaluate the f at the bounds of the interval I to find:

$$f(1) = -2 \qquad f(4) = 64$$

And because:

$$-2 < 0 < 64$$

the intermediate value theorem holds and thus this function must have at least 1 root.

Part b

Problem: Show that $(\forall x \in \mathbb{R}) \ f'(x) > 0$. Use this to show that $f(x)$ has only 1 real root.

Solution: First note that the derivative of f is:

$$f'(x) = 3x^2 + 1$$

It is should be clear that this is strictly positive, but to spell it out:

$$\begin{aligned} x^2 &\geq 0 && \text{(even-valued exponentiation)} \\ 3x^2 &\geq 0 && \text{(ordered field multiplication)} \\ 3x^2 + 1 &\geq 1 && \text{(ordered field addition)} \end{aligned}$$

Now for the second part of the question, we have shown that f has at least 1 root in part a. Now let us assume that it has at least 2 distinct roots r_1 and r_2. If this was the case then the mean value theorem tells us:

$$(\exists c \in \mathbb{R}) \ f'(c) = \frac{f(r_1) - f(r_2)}{r_1 - r_2} = 0$$

But this contradicts our previously established result of $(\forall x \in \mathbb{R}) \ f'(x) > 0$. Thus f has only 1 real root.

Part c

Problem: Write Matlab programs to approximate the root of f using both Newton and the bisection method to within a tolerance of 10^{-3} (for the input in the bisection method and for the iterations in Newton's method). For Newton's method use an initial guess of $x_0 = 1$ and for the bisection method use the same interval $[1, 4]$. Give each iteration and the total number of iterations for both programs.

Solution:

Iteration	Bisection	Newton
1	2.500000000000000	1.500000000000000
2	1.750000000000000	1.387096774193548
3	1.375000000000000	1.378838947597994
4	1.562500000000000	1.378796701230898
5	1.468750000000000	
6	1.421875000000000	
7	1.398437500000000	
8	1.386718750000000	
9	1.380859375000000	
10	1.377929687500000	
11	1.379394531250000	
12	1.378662109375000	

Problem 3

Problem: Note the following recurrence relation:

$$x_{n+1} = 2 - (1+c)x_n + cx_n^3$$

This sequence will converge to the value $s = 1$ for certain c, assuming x_0 is sufficiently close to s. Find those c values. Also find which of them lead the sequence to converge quadratically.

Solution: Recall that for all recurrence relations of the form $x_{n+1} = f(x_n)$ that converge to a value s, the following holds true:

$$s = \lim_{n\to\infty} x_n = \lim_{n\to\infty} f(x_n) = f(s)$$

The fixed point theorem tells us that this recurrence relation will converge to s when $|f'(s)| < 1$. Using this, we can solve for which values of c this condition holds true when $s = 1$:

$$\begin{aligned} f(s) &= 2 - (1+c)s + cs^3 \\ f'(s) &= -(1+c) + 3cs^2 \\ f'(1) &= -(1+c) + 3c = 2c - 1 \end{aligned}$$

Plugging this into the convergence condition we have:

$$\begin{aligned}|f'(1)| = |2c-1| &< 1\\ -1 < 2c - 1 &< 1\\ 0 < 2c &< 2\\ 0 < c &< 1\end{aligned}$$

And so x_n will only converge to 1 for $c \in (0,1)$. To find where this convergence is quadratic, we have to remember that pth order convergence means that $f^{(i)}(s) = 0$ and $f^{(p+1)}(s) \neq 0$ for positive integers $i < p$. In our case this means:

$$f'(1) = 2c - 1 = 0 \implies c = \frac{1}{2}$$

And to verify that the second derivative is non-zero at this value of $c = \frac{1}{2}$:

$$\begin{aligned}f''(s) &= 6cs\\ f''(1) &= 6c\\ &= 6\left(\frac{1}{2}\right) = 3 \neq 0\end{aligned}$$

And so x_n will only converge quadratically to 1 for $c = \frac{1}{2}$.

Problem 4

Problem: Use Matlab to implement Newton's method and solve the following system of equations:

$$\begin{cases} x - x^2 - y^2 = 0\\ y - x^2 + y^2 = 0\end{cases}$$

Use the initial guess $(0.5, 0.5)$, give all iterations, and stop iterating until two successive iterates agree to 14 decimal places.

Solution: The more general fixed point iteration for a nonlinear system of equations using Newton's method is given below:

$$\begin{bmatrix} f_x(x_0,y_0) & f_y(x_0,y_0)\\ g_x(x_0,y_0) & g_y(x_0,y_0)\end{bmatrix}\begin{bmatrix} x_{n+1} - x_n\\ y_{n+1} - y_n\end{bmatrix} = \begin{bmatrix} -f(x_0,y_0)\\ -g(x_0,y_0)\end{bmatrix}$$

Using Matlab to implement this iteration we arrive at the following approximations:

Iteration	x_n	y_n
0	0.500000000000000	0.500000000000000
1	1.000000000000000	0.500000000000000
2	0.812500000000000	0.437500000000000
3	0.773719879518072	0.420557228915663
4	0.771848952636680	0.419645658001209
5	0.771844506371371	0.419643377620421
6	0.771844506346038	0.419643377607081
7	0.771844506346038	0.419643377607081

Problem 5

Problem: Solve the same system as in problem 4 but use Matlab's `fsolve` routine to do it. In particular, use the following code:

```
options = optimset('Display', 'iter');
x0 = [0.5,0.5]
[x,fval] = fsolve(@fcnns,x0,options)|
```

Solution: Running this code, with the system defined in `fccns.m`, returns the following approximation and error:

```
x = 0.771844506371479    0.419643377620486

fval = 1.0e-10 *

   -0.250830467507512
   -0.146170298087611
```

Note that `fsolve` defaults to 10 decimal places of accuracy (tol = 10^{-10}) and so it only agrees with problem 4's answer to the 10th decimal place.