Week 7 Hand-in

NumIntro 2019

Department of Computer Science
University of Copenhagen

Casper Lisager Frandsen <fsn483@alumni.ku.dk>

Version 1
Due: October 30th, 08:00

Contents

1. Problem 1 **3**
(a) 3
(b) 3
(c) 3
(d) 4

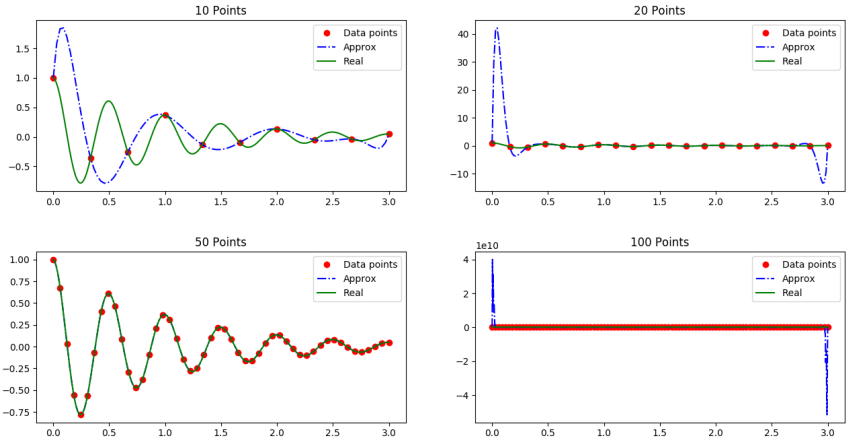
2. Problem 2 **4**
(a) 4
(b) 5

1. Problem 1

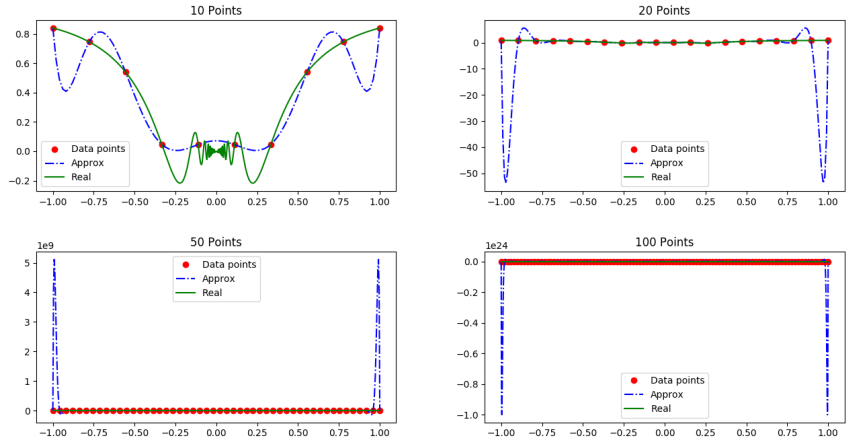
This Exercise has been implemented in "Q0013.py".

(a)

(b)



(c)



The approximate functions have been plotted with 10 times as many points as there are data points in the corresponding plot, and the "real" functions have been plotted with 1000 points. It is interesting to note that as we use more and more data points to approximate the function, it becomes significantly more precise in the middling parts of the interval. However, at the outer edges of the interval, the imprecisions rise extremely quickly, with the approximate functions reaching extremely large values.

(d)

We start by writing the function as per Theorem 6.1.2 in the book:

$$|f(x) - p_n(x)| = \frac{1}{(n+1)!} f^{n+1}(\mathcal{E}x) \prod_{i=0}^{n} (x - x_i)$$

We start by finding an upper bound for $f^{n+1}(\mathcal{E}x)$, with the help of the general Leibniz rule:

$$\begin{aligned} f^{n+1}(\mathcal{E}x) &= \left(\frac{d}{dt}\right)^{n+1} (e^{-t} cos(4\pi t)) \\ &= \sum_{k=0}^{n+1} \binom{n+1}{k} \left(\frac{d}{dt}\right)^{(n+1-k)} (e^{-t}) \cdot \left(\frac{d}{dx}\right)^{k} cos(4\pi t) \end{aligned}$$

We can bound each of the derivative terms here:

$$\left(\frac{d}{dt}\right)^{(n+1-k)} (e^{-t}) \leq e^0 = 1$$

$$\left(\frac{d}{dx}\right)^{k} cos(4\pi t) \leq 1 \cdot (4\pi)^k$$

We insert this into $f^{n+1}(\mathcal{E}x)$:

$$f^{n+1}(\mathcal{E}x) \leq \sum_{k=0}^{n+1} \binom{n+1}{k} (4\pi)^k$$

We can simplify this by using this expansion:

$$\sum_{k=0}^{n} \binom{n}{k} x^k = (1+x)^n$$

$$\sum_{k=0}^{n+1} \binom{n+1}{k} (4\pi)^k = (1+4\pi)^{n+1}$$

$$f^{n+1}(\mathcal{E}x) \leq (1+4\pi)^{n+1}$$

Because $t \in [0, 3]$, we can bound the following:

$$\prod_{i=0}^{n} (x - x_i) \leq 3^n$$

Thus, we have a bound on $|f(x) - p_n(x)|$:

$$|f(x) - p_n(x)| \leq \frac{3^n \cdot (1+4\pi)^{n+1}}{(n+1)!}$$

2. Problem 2

(a)

This Exercise has been implemented in "Q0013.py".

(b)

I have been unable to get the code for the "Divided Differences" and "Newton-Horner" algorithms to work in the Q files, as the book pseudocode is not very clear. The plot for the Newton-Horner algorithm is then for one that does almost the right thing, except all the values are significantly further from zero than they should be. The general form of the function is correct though, and it has a running time extremely close to what the correct one would have.
However, I can still comment on what I think would happen, if they were implemented. Based purely on the number of iterations the algorithms go through, it would seem like the Lagrange polynomial function would be slowest, as it will loop through $(n * (n-1))$ elements in total, whereas the other two reduce the second factor significantly by limiting the second loop to $n - j$ iterations instead of $n - 1$. This means that while all the algorithms have an $O(n^2)$ asymptotic running time, the Lagrangian one has significantly larger constant factors. In the plot below, we can see that the running time is polynomial for both algorithms, as expected, and that the Newton-Horner method is significantly faster, but only with regards to constant factors. Any significant outliers are likely due to cache misses inside the computer.

Figure 1: Lagrangian method

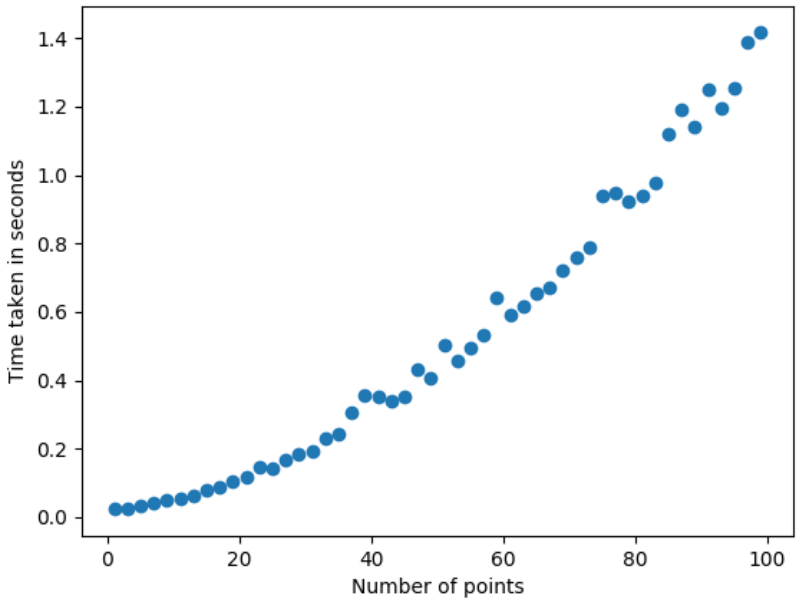


Figure 2: Incorrect Newton-Horner method