BLG202E: Numerical Methods Homework #1

Uğur Ali Kaplan - 150170042

February 27, 2019

1. For $x \in \mathbb{R}$, $\beta \in \mathbb{N}$, $t \in \mathbb{N}$ and $0 \leq d_i \leq \beta - 1$ it can be written

$$x = \pm(\frac{d_0}{\beta^0} + \frac{d_1}{\beta^1} + \frac{d_2}{\beta^2} + ... + \frac{d_{t-1}}{\beta^{t-1}}) \times \beta^e \tag{1}$$

where β is the base of the number system and t is the precision. Since β and t needs to be natural numbers, given that $\epsilon \in \mathbb{R} \setminus \mathbb{Q}$ and $\pi \in \mathbb{R} \setminus \mathbb{Q}$ it is not possible to represent ϵ and π in a floating point system.

It may be possible to represent them with other systems such as *arbitrary-precision arithmetic* system but this is not relevant to this question.

The answer is **NO**.

2. There exists such a system where $\beta = 7$ and $t = 2$, then we can represent $\frac{8}{7}$ as

$$\frac{8}{7} = (\frac{1}{7^0} + \frac{1}{7^1}) \times 7^0 \tag{2}$$

3. To find the *rounding unit*, we can try to see $num + \epsilon = num$ because any value under ϵ will be regarded as 0 by the machine. Since I don't have a calculator, I have written the following function:

```
function to_be_found = find_eps
  to_be_found = 1.0;
  while (1.0 + to_be_found / 2 != 1.0)
  to_be_found = to_be_found / 2;
  endwhile
  return
endfunction
```

Listing 1: Finding Rounding Unit

When called, this function returns `ans = 2.2204e-16` and typing `eps` into the console prints `ans = 2.2204e-16`. If one does not believe their own eyes, we can further ensure it by returning a boolean value.

```
function is_eps = find_eps
  to_be_found = 1.0;
  while (1.0 + to_be_found / 2 != 1.0)
  to_be_found = to_be_found / 2;
  endwhile
  is_eps = (to_be_found == eps);
  return
endfunction
```

Listing 2: Making Sure We Have Found The Rounding Unit

This returns `ans = 1`.

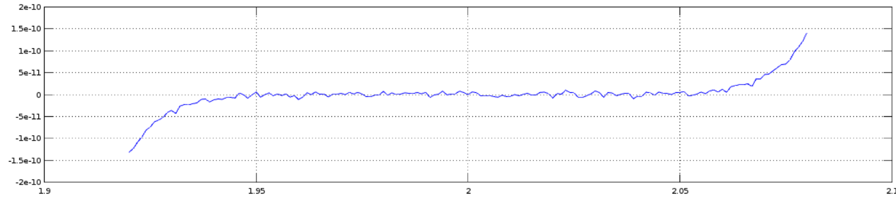

Figure 1: Graph of the nested evaluation

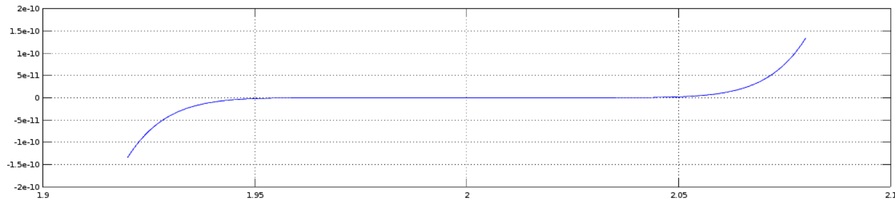

Figure 2: Graph of the function

4. (a)

$$f(x) = (x-2)^9 = x^9 - 18x^8 + 144x^7 - 672x^6 + 2016x^5 - 4032x^4 + 5376x^3 - 4608x^2 + 2304x - 512 \quad (3)$$

To use nested evaluation on (3), we can write it as:

$$f(x) = x(x(x(x(x(x(x(x(x-18)+144)-672)+2016)-4032)+5376)-4608)+2304)-512 \quad (4)$$

After writing the necessary code we can see the graphs of the function(Figure 2) and the nested evaluation(Figure 1).

```
A = 1.92:(2.08-1.92)/160:2.08; % Creates 161 equidistant points
f = @(x) (x - 2) .^ 9; % Function
f2 = @(x) x .* (x .* (x .* (x .* (x .* (x .* (x .* (x .* (x - 18) + 144) - 672) +
    2016) - 4032) + 5376) - 4608) + 2304) - 512; % Nested Evaluation Form

% Plot
subplot(2,1,1);
plot(A, f2(A));
subplot(2,1,2);
plot(A, f(A));
```

Listing 3: Creating the graphs

(b) Let's choose a point where both graphs differ from each other. $f(1.948) = -2.7799 \times 10^{-12}$ and $f_2(1.948) = -8.0149 \times 10^{-12}$. Following some of the steps for f_2 easily shows us the *round-off error*. Error accumulates as the time goes on.

```
>>1.948 - 18
ans = -16.052
>> ans * 1.948
ans = -31.269
>> ans + 144
ans =  112.73
>> ans * 1.948
ans =  219.60
```