We have the differential equation



We simulate it with 64-bits floating point. For a large value of n the calculated solution will be:



My attempt was to first solve it by hand:

The two roots: $r\_{1}=\frac{1}{2} r\_{2}=\frac{1}{3}$

$$y\_{n}=C\left(\frac{1}{2}\right)^{n}+D\left(\frac{1}{3}\right)^{n}$$

$$y\_{0}=C+D=1 y\_{1}=C\frac{1}{2}+D\frac{1}{3}=\frac{1}{2} (1-D)\frac{1}{2}+D\frac{1}{3}=\frac{1}{2} $$

$$\frac{1}{2}-\frac{1}{2}D+D\frac{1}{3}=\frac{1}{2}$$

$$-\frac{3}{6}D+D\frac{2}{6}=0 D=0 C=1$$

$$y\_{n}=\left(\frac{1}{2}\right)^{n}$$

I don’t see where the floating point inequarcy would lead to a wrong anwer for large n. Can you show where it does that?

Question 2:

A second order inhomoegeneous differential equation has the solution



If one are given two start values one would obtain that it is:



If the equation is simulated on a pc with 64-bits floating point one would get for large n that the computed result would be



From the assignment we see that C=0 and D=1 and A as a guessed solution is 1. How can you show that this would create overflow?