Q1

It's hard to figure it out exactly from just the plot of error. We can tell that it is probably along the lines of $error = O(dx^p)$ for some $1 < p < 2$. We can infer this by eyeballing the slope of the graph. The slope looks like it is roughly between 1 and 2 (paying attention to the quirk that the axis aren't drawn to the same physical scale).

To get a more exact solution, I fit a straight line through the 4 $(\log(dx), \log(error))$ points. As it turns out, np.polyfit tells me that the slope is 1.424. Since we have only 4 data points, I take only 2 significant digits. I have included a plot of $dx^{1.4}$ too.

As we can see, they have similar slopes.

Q2

First, we notice that with CFL=0.7, the solution at $t = 1$ is pretty good. However, with CFL=0.75, the solution is terrible. As can be (kind of) seen, CFL=0.75 produces a solution that oscillates wildly at low x.

The instability is from the AB3 iteration. Hence, we can conclude that the method is stable for a maximum CFL which is somewhere in between 0.7 and 0.75. Equivalents, since $dx = 0.002$ and $c = 1$, we know that the maximum allowable dt is in the range $[0.7 * 0.002, 0.75 * 0.002] = [0.0014, 0.0015]$

What we can conclude is that $\lambda \Delta t$ must fall outside of the stability range. A quick look at the eigen values of C tells us that the largest eigenvalue is 2i.