2D Constant (Numerical)

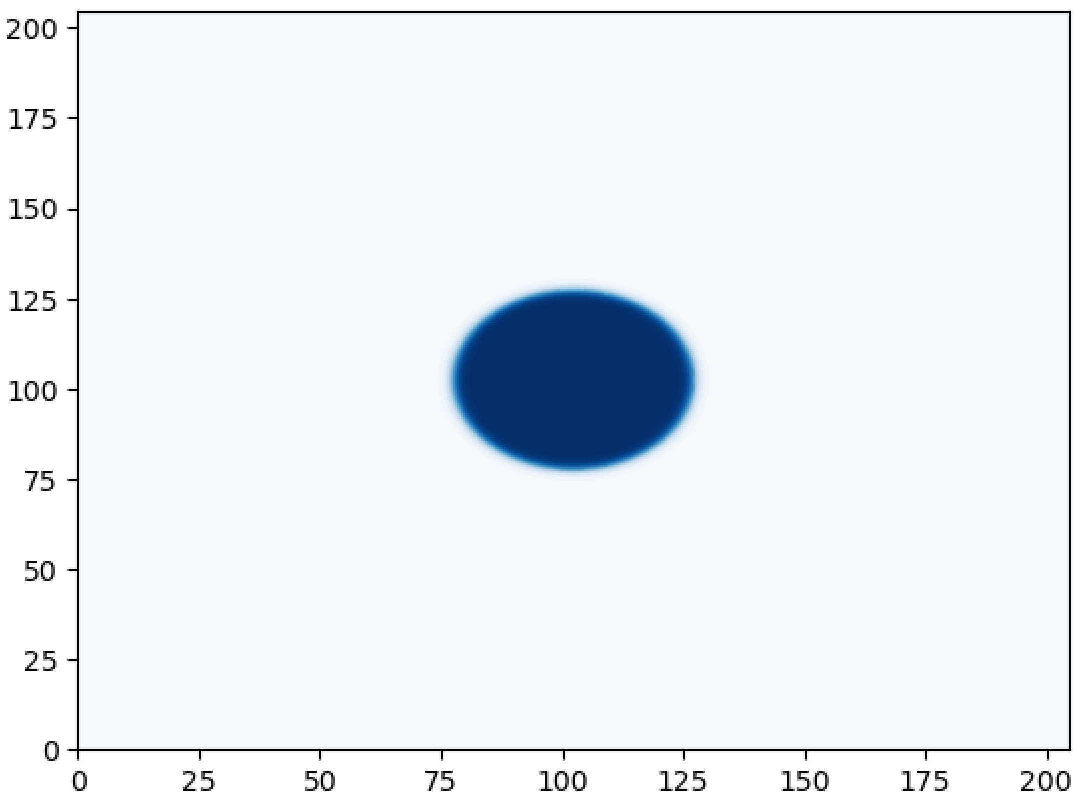
#2D #circular_island #mbc

Here we ran simulations of the 2D TDGL equation considering the initial state to be a circular island of radius R_0

$$u(x, y, t = 0) = \tanh\left((R - R_0)\sqrt{\frac{1}{2}}\right)$$

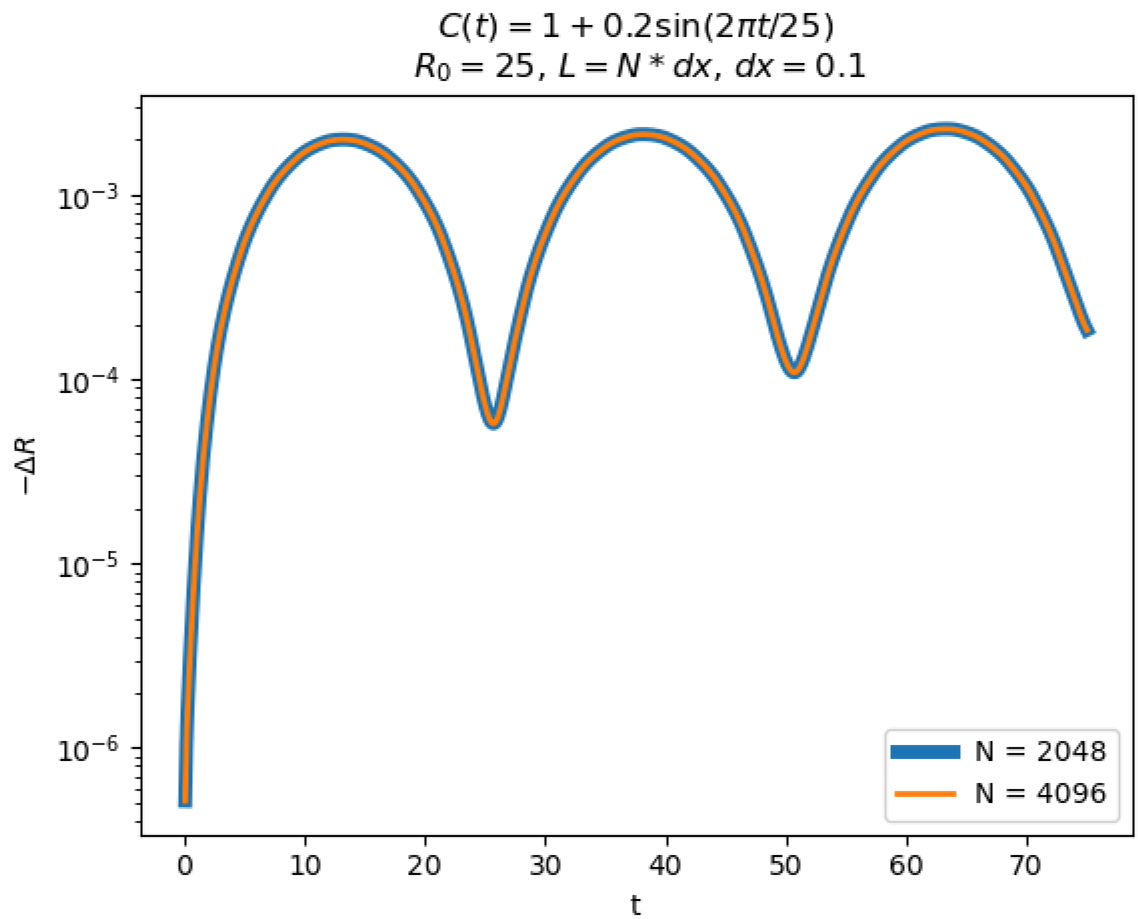
we considered the simulation box lenght

$$L = 204.8 \quad R_0 = 25$$



such that there is no interaction between opposite kinks due to the PBC conditions.
The size of the **simulation box is big enough** for this radius as you can see from here:

$$\Delta R = R_{C(t)} - R_{C=1}$$



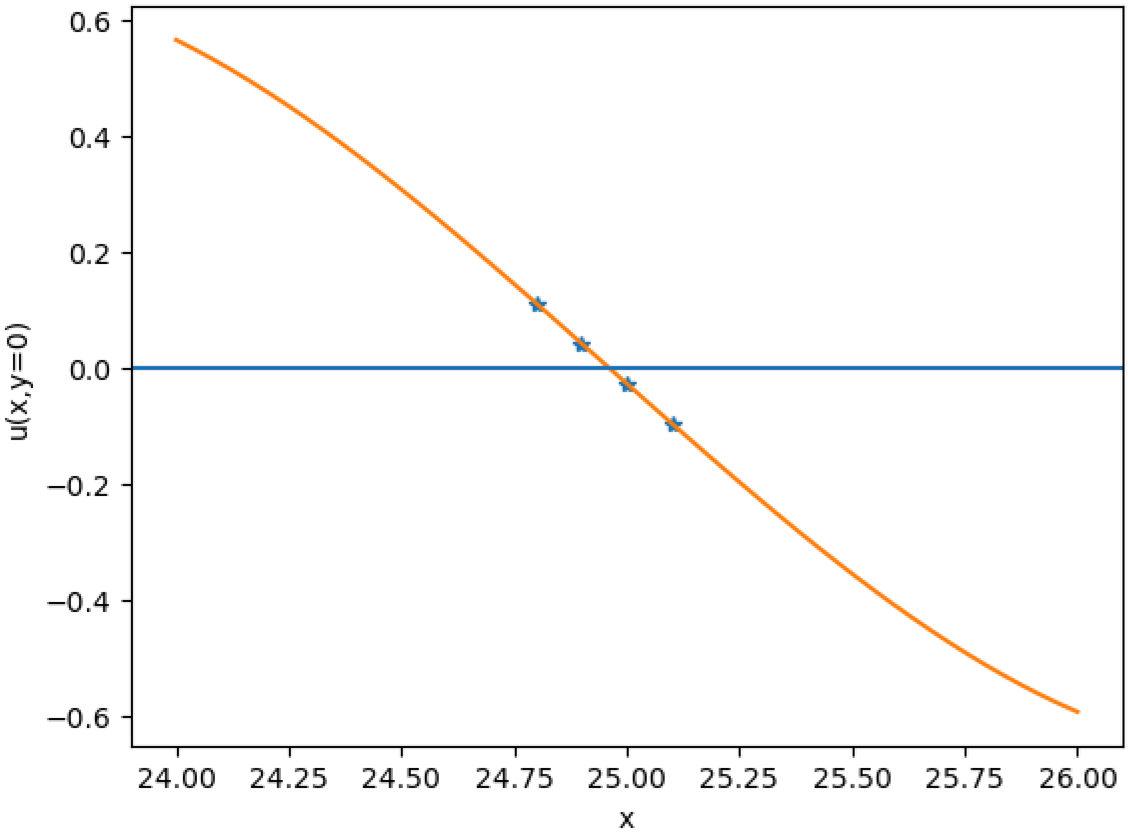
- [How we measure things]
  - [How Radius is measured]
  - [How constant C and C(t) data is compared]
    - [Solution]
- [Checking MBC]
  - [Numerical error]
- [Conclusion]

How we measure things

How Radius is measured

The radius, as a function of time $R(t)$, has been measured by looking at the sections of $u(x, y)$ along the line $y = 0$. Then we focus on the $x > 0$ semi-axis, where there is a kink.
The position of the kink is **estimated** by finding the discrete x_n where u changes sign. Then **4 points** around x_n are considered and a **spline cubic interpolation** is made.
CubicSpline: "piecewise cubic polynomial which is twice continuously differentiable"
The position of the kink is represented by the central root of the polynomial and this is our

estimate for the radius R.



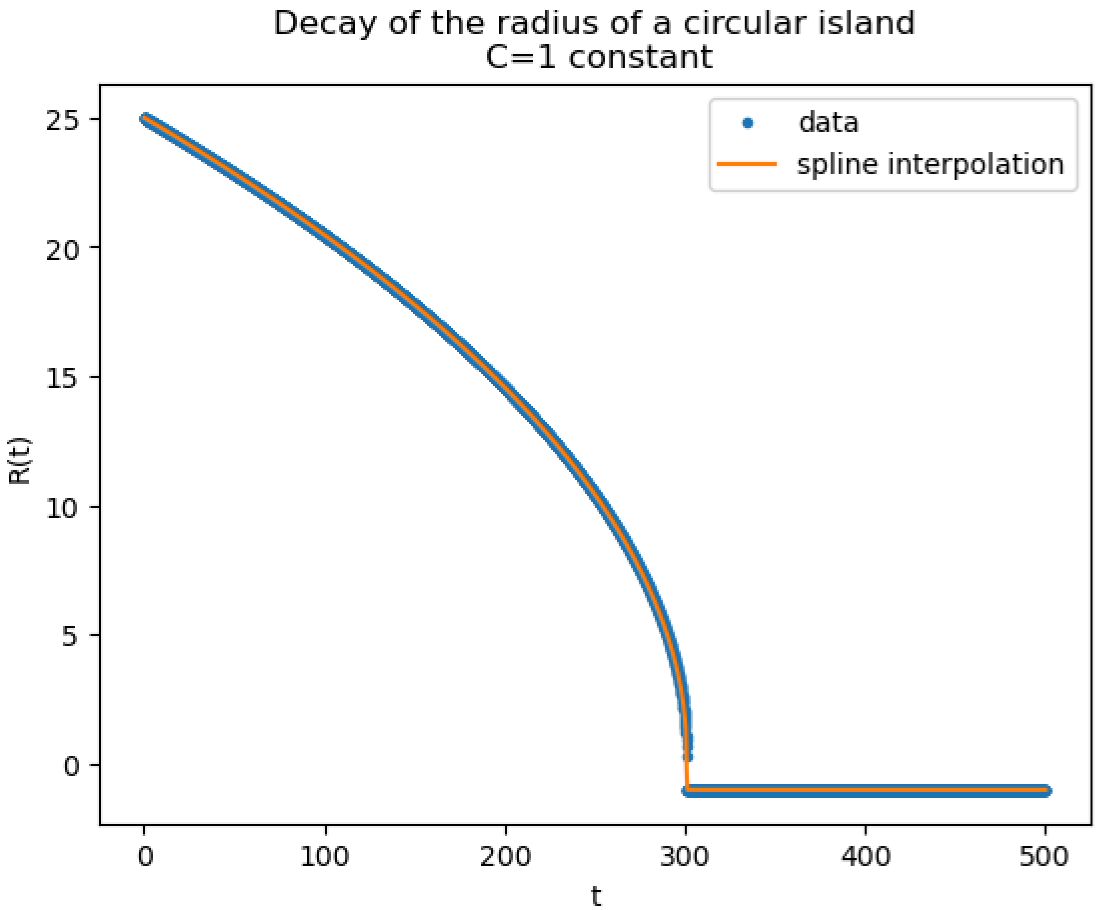
How constant C and C(t) data is compared

As time passes, the curve $R(t)$ deviates more and more from the curve $R_{C=1}(t)$ curve.
If we want to compare $\partial_t R_{C=1}$ and $\partial_t R$ at the **same value of R**, this is a problem, because this **does NOT mean** to compare the two derivatives at the **same time t**.

Solution

We interpolate with a **spline** of degree $k = 4$ (**UnivariateSpline**) the data $R_{C=1}(t)$. Obtaining a **continuos** function

$$f : t \to R_{C=1}(t)$$



Then we compute the **analytical derivative**

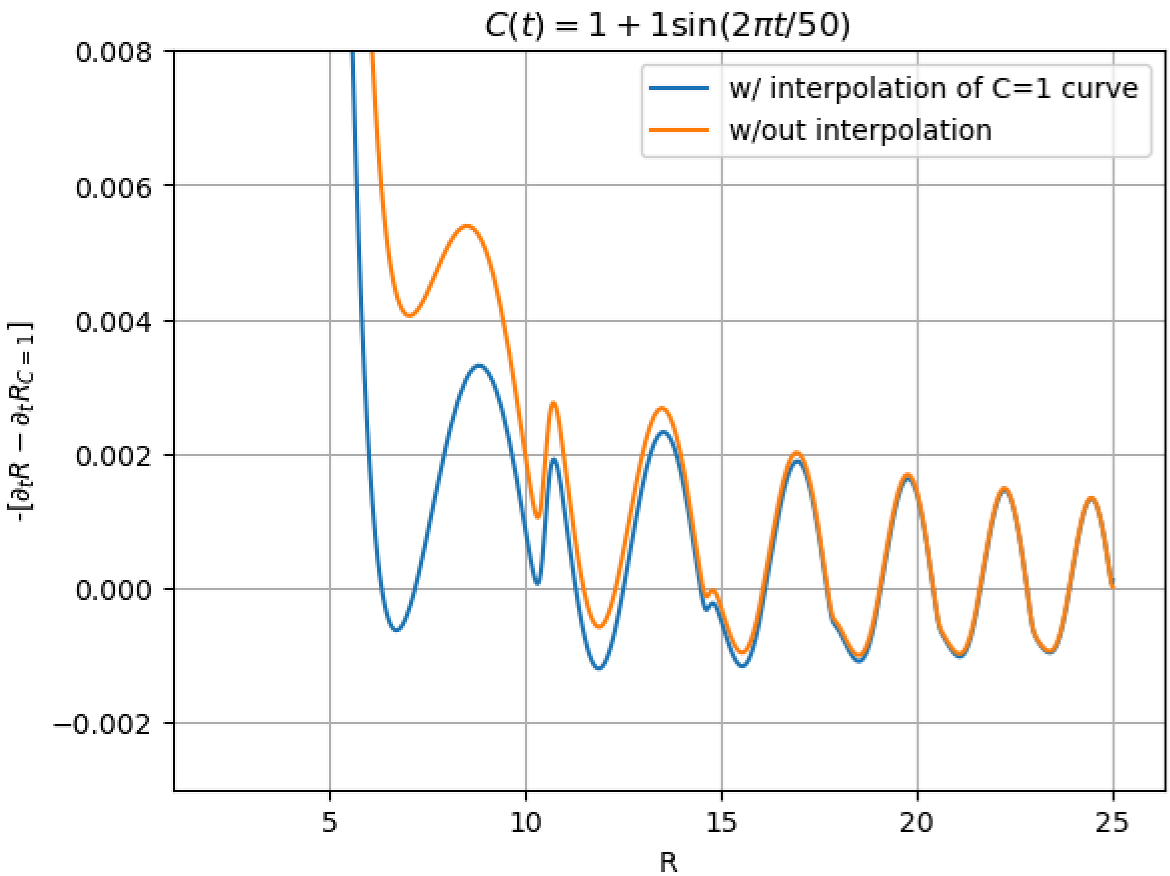
$$g : t \to \partial_t R_{C=1}(t)$$

Then we can find the inverse respect to f of an image R by looking at the root of $f(t) - R$.
And we can compose those functions to obtain

$$\gamma = g(f^{-1}(R)) : R \to \partial_t R_{C=1}$$

Now this function is defined **for any value of** R, so we can compare it with the **numerical derivative** of $R(t)$ such that R is the same for each datapoint compared.
In the plot below, the blue line represent the difference obtained using the interpolation, while the orange curve represents the difference obtained subtracting data points with the same time

t (and on x-axis is plotted the radius of the simulation with $C(t)$).



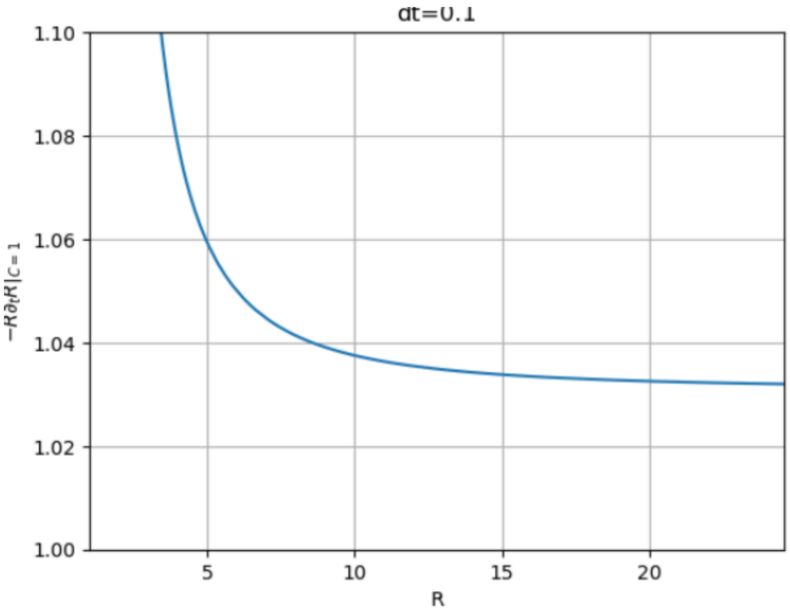
Checking MBC

We expect that, in the large R limit, if C is constant

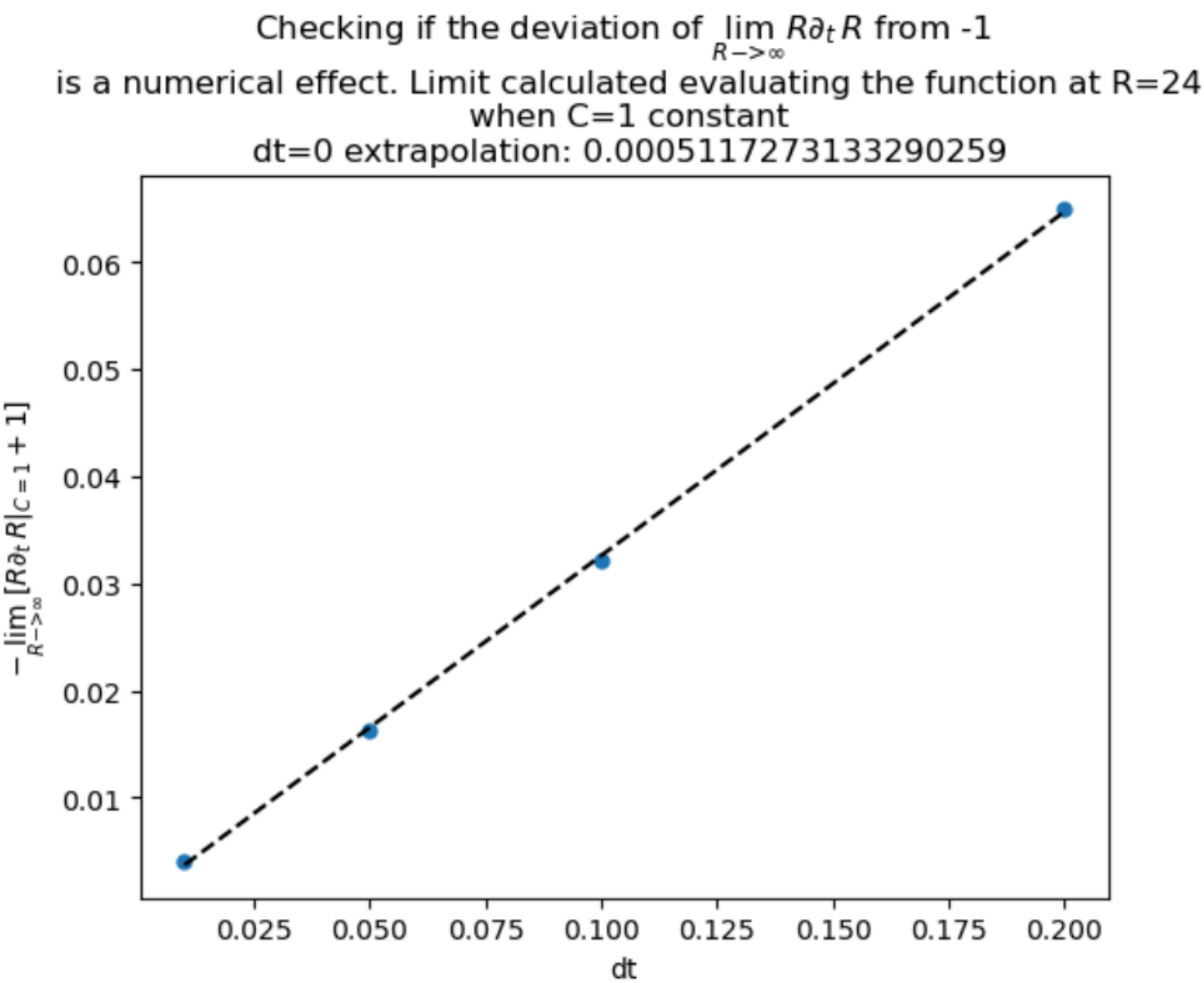
$$\partial_t R = -\frac{1}{R}$$

but we see that, at large R, $R\partial_t R$ saturates to a value different from -1.
Is it due to the time discretization?

Numerical error



We estimate $\lim_{R\to\infty} R\partial R$ by evaluating the function (plotted above) at a large R. We chose $R = 24$ ($R_0 = 25$ is the initial radius). Then we repeat many simulation with $C = 1$ and different values of dt.



The value of the estrapolation to $dt = 0$ represent the order of magnitude of a real (nonnumerical) effect. If we compare it to 1, that is the expected value of the limit.

$$5 * 10^{-4} \ll 1$$

so we conclude **it is just a numerical effect.**

Conclusion

We confirm that the deviation from the MBC law is just a numerical effect. So we state that MBC is correct for large values of R.