Why Crank-Nicolson

#Crank-Nicolson #numerical

We integrate the **TDGL** (or Cahn-Allen) equation::

$$\partial_t u = \partial_{xx} u + C(t)u - \mathcal{F}[u^3] \quad (1\text{D})$$

$$\partial_t u = \Delta u + C(t)u - \mathcal{F}[u^3] \quad (2\text{D})$$

with the Crank-Nicolson scheme (in Fourier space: Crank-Nicolson in Fourier space).
The reason is that Implicit and Explicit Euler algorithm do not integrate correctly the dynamics of an initially flat profile $u(x) = u_0$ (0D case).

Explicit and implicit Euler for an initially flat profile

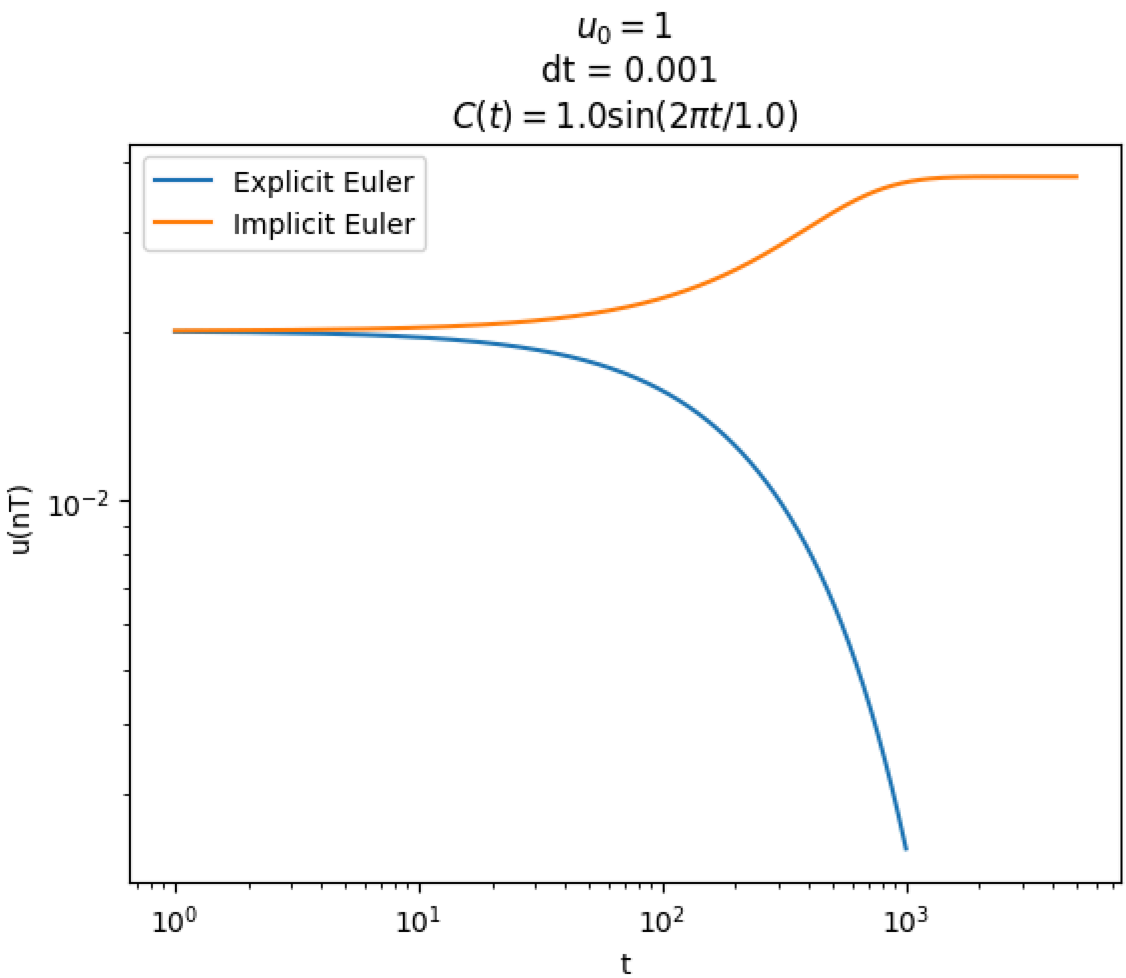
Here we integrate the **1D** TDGL equation with Implicit or Explicit Euler's algorithm (in Fourier space) starting from a state $u(x, t=0) = 1$. During the dynamics $C(t)$ oscillates as

$$C(t) = \sin(2\pi t)$$

So the average value of the oscillation is zero, this means that we should see

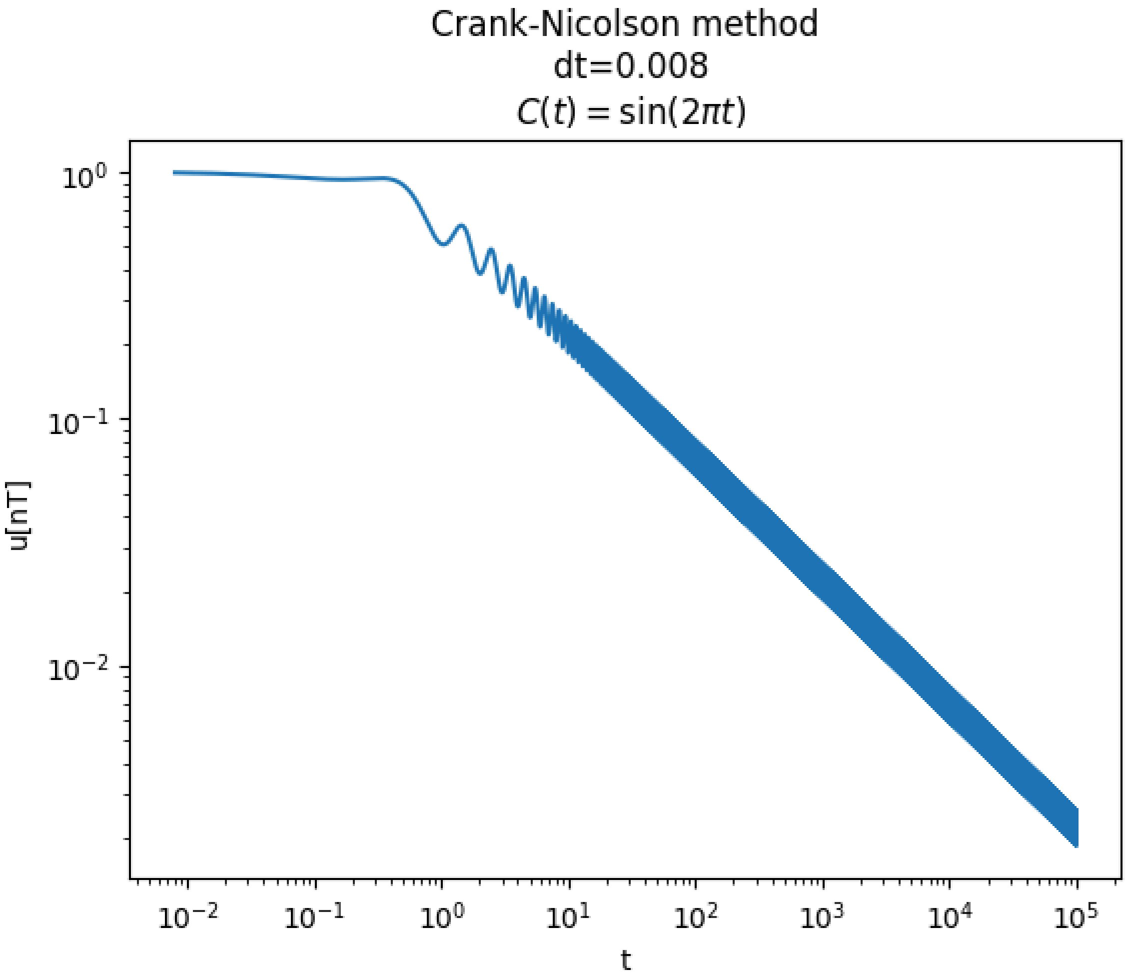
$$u(t) \sim t^{-\frac{1}{2}}$$

Instead we see an exponential time dependance!



Crank-Nicolson for an initially flat profile

Instead, integrating the same system with Crank-Nicolson scheme (in Fourier space), leads to the correct power-law decay!

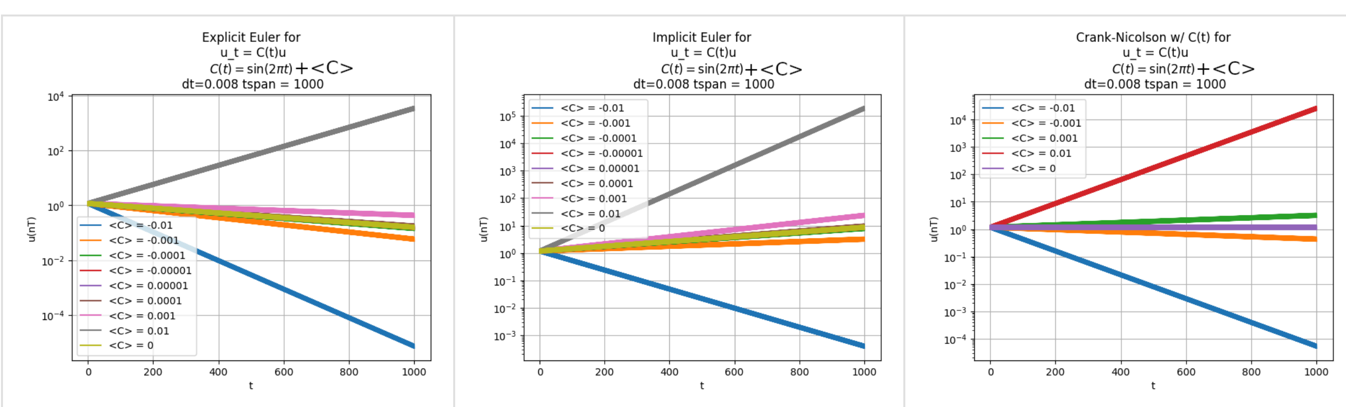


Linear dynamics

Crank-Nicolson in Fourier space

You can see better why the Euler schemes are not good, by integrating **only the linear part** of the equation. This time there is no need of Fourier transform as the equation to integrate is

$$\partial_t u = C(t)u$$



You can see that $u(t)$ grows or decays exponentially fast also when the average value $\langle C \rangle = 0$ if you use Implicit or Explicit Euler schemes. While Cranck Nicholson works fine.

Crank-Nicolson in Fourier space

To integrate the TDGL equation, we apply a Fourier transfrom in x, so

$$\partial_t u = \partial_{xx} u + C(t)u - \mathcal{F}[u^3]$$

becomes ($u(x,t) \to \mathcal{F}[u(x,t)] = U_q(t)$)

$$\partial_t U_q = -q^2 U_q + C(t)U_q - \mathcal{F}[\mathcal{F}[u^3]]_q$$

So you get rid of the space derivatives and you use the Crank-Nicolson scheme to integrate the equation in time for a small timestep dt. Then you do the inverse fourier transform and you retrieve $u(x, t + dt)$. Then you repeat.

Crank-Nicolson scheme

It is formulated by taking an average of the formulas of Implicit and Explicit schemes:

- Explicit Euler:

$$U(t + dt) = U(t) + [C(t)U(t) - \mathcal{F}[u^3](t)]dt$$

- Implicit Euler:

$$U(t + dt) = U(t) + [C(t + dt)U(t + dt) - \mathcal{F}[u^3](t + dt)]dt$$

If we average the two expression (sum them and divide by 2):

$$U(t + dt) = U(t) + \frac{dt}{2}[C(t)U(t) - \mathcal{F}[u^3](t) + C(t + dt)U(t + dt) - \mathcal{F}[u^3](t + dt)]$$

Now we make an **approximation** in order to get an explicit formula for $U(t + dt)$ if you know $U(t)$:

$$\mathcal{F}[u^3](t + dt) \to \mathcal{F}[u^3](t)$$

After this approximation, we isolate $U(t + dt)$ and we find

$$U(t + dt) = U(t)\frac{(1 + \frac{dt}{2}C(t))}{(1 - \frac{dt}{2}C(t + dt))} - \frac{\mathcal{F}[u^3](t)dt}{(1 - \frac{dt}{2}C(t + dt))}$$

As you need to compute the FFT of $u^3(t)$ at each step, you need, after each step dt, compute the IFFT to get $u(x, t)$, then compute $u^3(x, t)$ and then its FFT. Then you can proceed with the next step.

NOTE: The same approximation we make to use the Crank-Nicolson scheme, would be applied within the Implicit Euler scheme, if you want to implement it.