

Report for frontCollision

Simulated with: lib.managers.crankNicolson.default

Simulation constants:

baseDensity: 1	chemicalPotential: 1	dt: 0.005
dx: 0.200	g: -1	hbar: 1
healingLength: 0.707	mass: 1	plotFPS: 1000.000
plotPause: 0.001	plotStep: 10	plotYMax: 2
plotYMin: -2	r: 0.125	tCount: 1000
tMax: 5	tMin: 0	velocity: 0
x0: 0	xCount: 100	xMax: 10
xMin: -10		

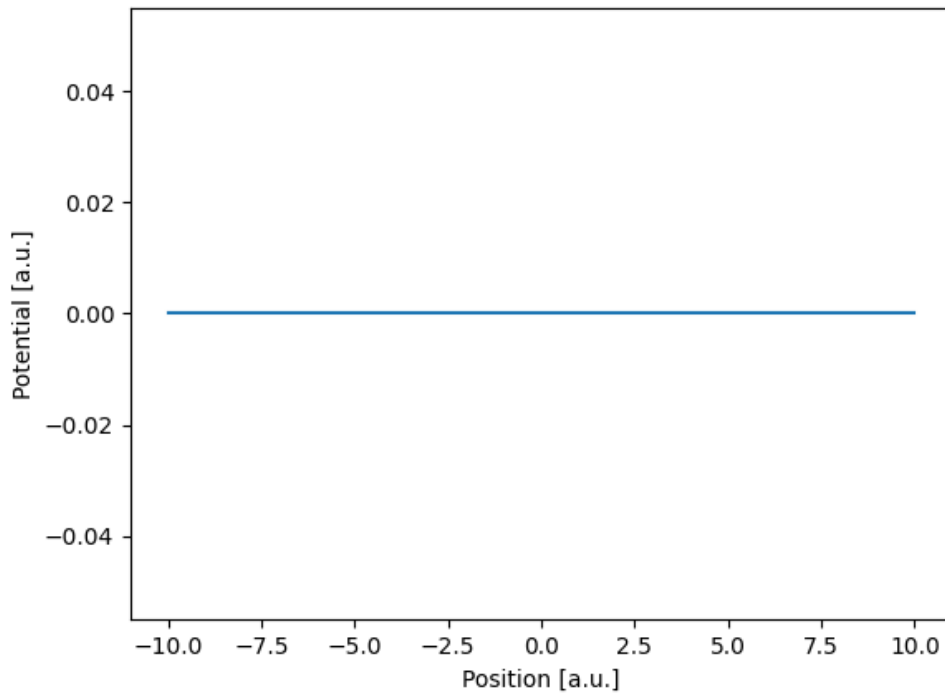
Wave function:

```
def waveFunction(x, t):    eta = 1        kappal = 1        omegal = (kappal**2 +
eta**2) / 2        v1 = kappal        kappa2 = -1        omega2 = (kappa2**2 + eta**2) /
2        v2 = kappa2        x1 = -3        x2 = 3        wf1 = eta * jnp.exp(1j * kappal *
(x - x1) - 1j * omegal * t) / jnp.cosh(eta * ((x - x1) - v1 * t))        wf2 = eta
* jnp.exp(1j * kappa2 * (x - x2) - 1j * omega2 * t) / jnp.cosh(eta * ((x - x2) -
v2 * t))        return wf1 + wf2
```

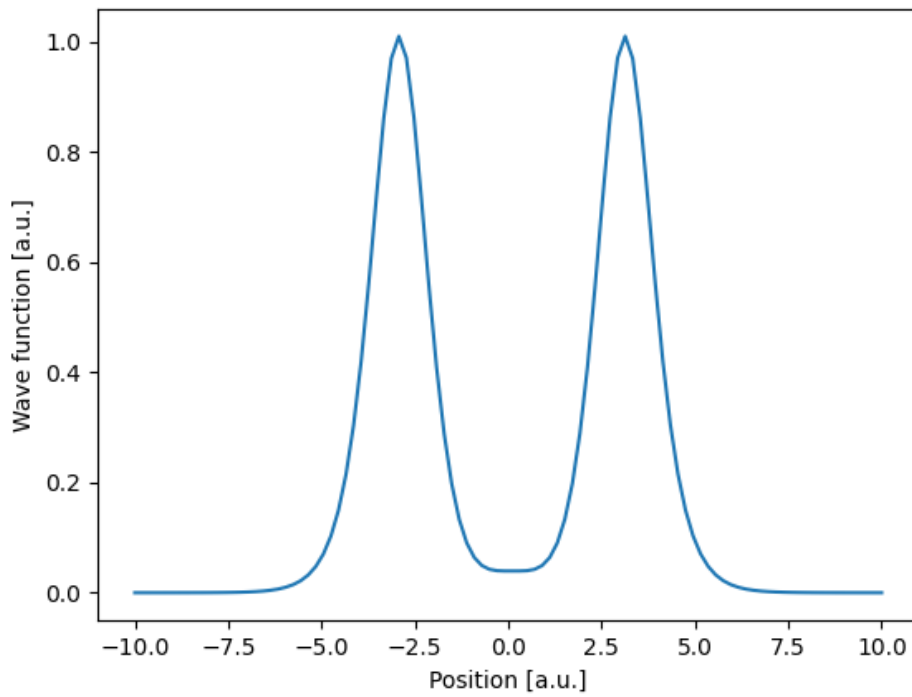
Potential function:

```
def V(x, t):    return jnp.zeros_like(x)
```

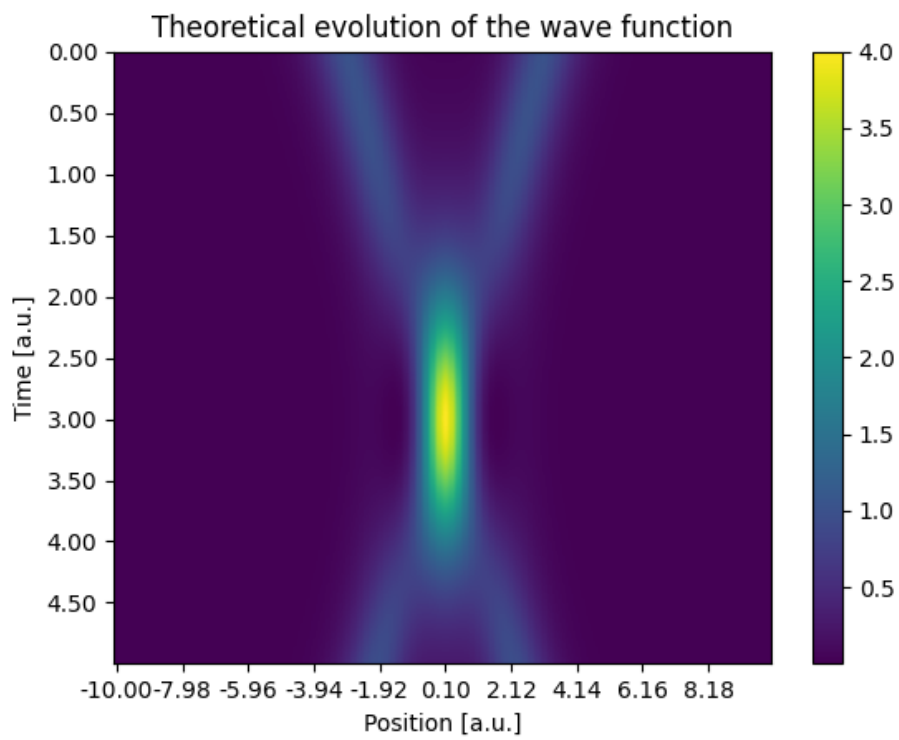
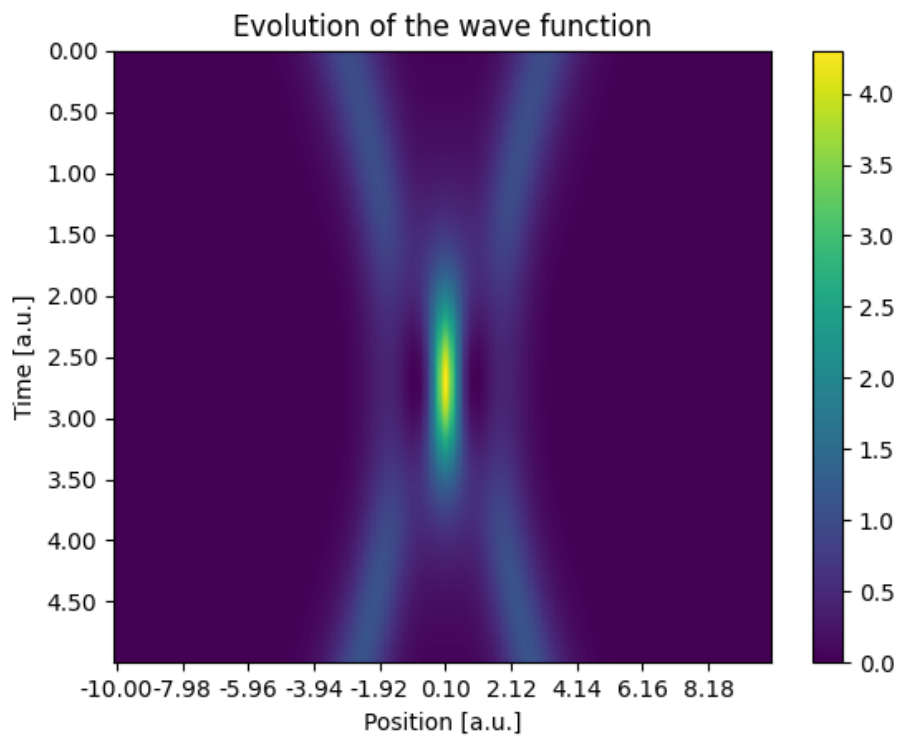
Potential at time $t=0$



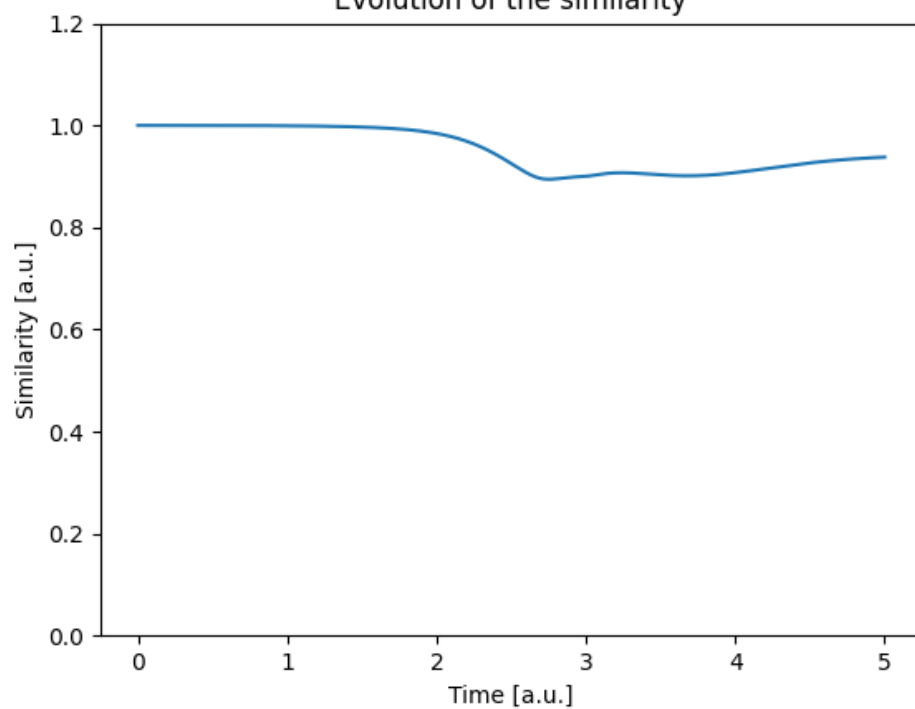
Wave function at time $t=0$



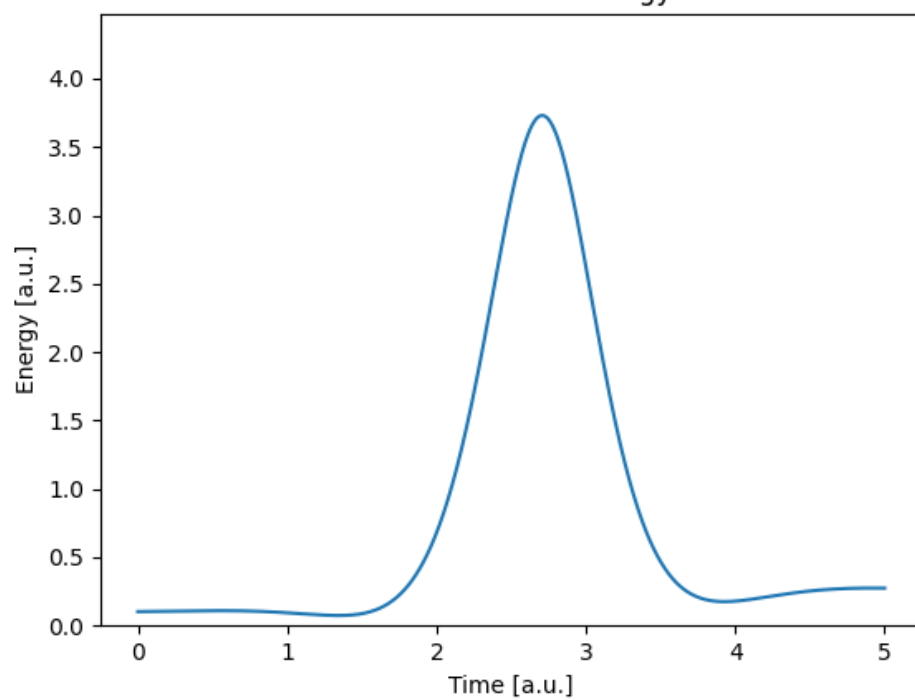
Results



Evolution of the similarity



Evolution of the energy



Evolution of the norm

