Madelung_Representation

July 7, 2021

[1]:
```
__author__ = "@Tssp"
__date__   = "17/03/2021"
import sympy as sp
import numpy as np
```

This notebook is aimed to obtain the superfluid hydrodynamic equations of motions through the Madelung representation of the Gross Pitaevskii equation.

[2]:
```
x, t, m, hbar = sp.symbols('x t m \\hbar', real=True)
psi = sp.Function('\\psi', real=False)(x, t)  # wavefunction
rho = sp.Function('\\rho', real=True)(x, t)  # probability density
V   = sp.Function('V', real=True)(x, t)      # Trap Potential
S   = sp.Function('S', real=True)(x, t)      # Madelung Ansatz
v   = sp.Function('v', real=True)(x, t)      # fluid velocity
Pq   = sp.Function('P_Q', real=True)(x, t)   # Quantum Preassure
g   = sp.symbols('g', real=True)             # g constant
```

[3]:
```
Schr_eq = sp.I * hbar * sp.diff(psi, t) + (hbar**2/(2*m) * sp.diff(psi, x, 2) -␣
↪V * psi - g*abs(psi)**2 * psi)
Schr_eq
```

[3]:
$$\frac{\hbar^2 \frac{\partial^2}{\partial x^2}\psi(x,t)}{2m} + i\hbar\frac{\partial}{\partial t}\psi(x,t) - g\psi(x,t)\left|\psi(x,t)\right|^2 - V(x,t)\psi(x,t)$$

Madelung transformation:
$$\psi(x,t) = \sqrt{\rho(x,t)}e^{imS(x,t)/\hbar}$$

[4]:
```
Schr_madelung = sp.simplify(Schr_eq.subs({psi:sp.sqrt(rho)*sp.exp(sp.I*m*S/
↪hbar)})).\
collect(sp.exp(sp.I*m*S/hbar))/sp.exp(sp.I*m*S/hbar)
Schr_madelung
```

[4]:
$$\frac{\hbar^2\left(\frac{2\frac{\partial^2}{\partial x^2}\rho(x,t) - \frac{\left(\frac{\partial}{\partial x}\rho(x,t)\right)^2}{\rho(x,t)}}{4\sqrt{\rho(x,t)}} + \frac{m\left(i\frac{\partial^2}{\partial x^2}S(x,t) - \frac{m\left(\frac{\partial}{\partial x}S(x,t)\right)^2}{\hbar}\right)\sqrt{\rho(x,t)}}{\hbar} + \frac{im\frac{\partial}{\partial x}S(x,t)\frac{\partial}{\partial x}\rho(x,t)}{\hbar\sqrt{\rho(x,t)}}\right)}{2m} + i\hbar\left(\frac{\frac{\partial}{\partial t}\rho(x,t)}{2\sqrt{\rho(x,t)}} + \frac{im\sqrt{\rho(x,t)}\frac{\partial}{\partial t}S(x,t)}{\hbar}\right) - g\sqrt{\rho(x,t)}\left|\sqrt{\rho(x,t)}\right|^2 - V(x,t)\sqrt{\rho(x,t)}$$

[5]:
```python
Schr_madelung = Schr_madelung.expand()
Schr_madelung
```

[5]:

$$\frac{\hbar^2 \frac{\partial^2}{\partial x^2}\rho(x,t)}{4m\sqrt{\rho(x,t)}} - \frac{\hbar^2\left(\frac{\partial}{\partial x}\rho(x,t)\right)^2}{8m\rho^{\frac{3}{2}}(x,t)} + \frac{i\hbar\sqrt{\rho(x,t)}\frac{\partial^2}{\partial x^2}S(x,t)}{2} + \frac{i\hbar\frac{\partial}{\partial x}S(x,t)\frac{\partial}{\partial x}\rho(x,t)}{2\sqrt{\rho(x,t)}} + \frac{i\hbar\frac{\partial}{\partial t}\rho(x,t)}{2\sqrt{\rho(x,t)}} - g\sqrt{\rho(x,t)}\left|\sqrt{\rho(x,t)}\right|^2 - m\sqrt{\rho(x,t)}\frac{\partial}{\partial t}S(x,t) - \frac{m\sqrt{\rho(x,t)}\left(\frac{\partial}{\partial x}S(x,t)\right)^2}{2} - V(x,t)\sqrt{\rho(x,t)}$$

[6]:
```python
Im_eq = Schr_madelung.coeff(sp.I, n=1)
Im_eq
```

[6]:

$$\frac{\hbar\sqrt{\rho(x,t)}\frac{\partial^2}{\partial x^2}S(x,t)}{2} + \frac{\hbar\frac{\partial}{\partial x}S(x,t)\frac{\partial}{\partial x}\rho(x,t)}{2\sqrt{\rho(x,t)}} + \frac{\hbar\frac{\partial}{\partial t}\rho(x,t)}{2\sqrt{\rho(x,t)}}$$

0.1 Imaginary part

Defining:

$$\vec{v} = \frac{\partial S(x,t)}{\partial x}$$

[7]:
```python
Im_eq = (Im_eq * 2 * m/hbar* sp.sqrt(rho)).expand()
Im_eq
```

[7]:

$$m\rho(x,t)\frac{\partial^2}{\partial x^2}S(x,t) + m\frac{\partial}{\partial x}S(x,t)\frac{\partial}{\partial x}\rho(x,t) + m\frac{\partial}{\partial t}\rho(x,t)$$

[8]:
```python
sp.diff(rho*sp.diff(S, x), x)
```

[8]:

$$\rho(x,t)\frac{\partial^2}{\partial x^2}S(x,t) + \frac{\partial}{\partial x}S(x,t)\frac{\partial}{\partial x}\rho(x,t)$$

[9]:
```python
Im_eq.subs({rho*sp.diff(S, x, 2): sp.diff(rho*sp.diff(S, x), x) - sp.diff(S, x)␣
↪* sp.diff(rho, x)})
```

[9]:

$$m\rho(x,t)\frac{\partial^2}{\partial x^2}S(x,t) + m\frac{\partial}{\partial x}S(x,t)\frac{\partial}{\partial x}\rho(x,t) + m\frac{\partial}{\partial t}\rho(x,t)$$

$$m\frac{\partial\rho}{\partial t} - \frac{\partial}{\partial x}\left((\rho\frac{\partial}{\partial x}S) - S - \rho + S + \rho\right) = 0$$

Or:

$$m\frac{\partial\rho}{\partial t} - \frac{\partial}{\partial x}\left((\rho\frac{\partial}{\partial x}S)\right) = 0$$

$$\frac{\partial\rho}{\partial t} - \frac{\partial}{\partial x}\left(\rho v\right) = 0$$

A continuity equation, in 3D

$$\frac{\partial\rho}{\partial t} - \nabla\left(\rho v\right) = 0$$

0.2 Real part

In this part we are going to treat $\sqrt{\rho}$ as an unique variable in order to forbid Python to literally express the derivative and therefore simplify the equations

[10]:
```
sqrt_rho = sp.Function('\sqrt{\\rho}', real=True)(x, t)
```

[11]:
```
Schr_madelung = sp.simplify(Schr_eq.subs({psi:sqrt_rho*sp.exp(sp.I*m*S/hbar)})).
  ↪\
collect(sp.exp(sp.I*m*S/hbar))/sp.exp(sp.I*m*S/hbar)
Schr_madelung
```

[11]:

$$\frac{\hbar^2\left(\frac{\partial^2}{\partial x^2}\sqrt{\rho}(x,t)+\frac{m\left(i\frac{\partial^2}{\partial x^2}S(x,t)-\frac{m\left(\frac{\partial}{\partial x}S(x,t)\right)^2}{\hbar}\right)\sqrt{\rho}(x,t)}{\hbar}+\frac{2im\frac{\partial}{\partial x}S(x,t)\frac{\partial}{\partial x}\sqrt{\rho}(x,t)}{\hbar}\right)}{2m}+ i\hbar\left(\frac{\partial}{\partial t}\sqrt{\rho}(x,t)+\frac{im\sqrt{\rho}(x,t)\frac{\partial}{\partial t}S(x,t)}{\hbar}\right)-g\sqrt{\rho}^3(x,t)-V(x,t)\sqrt{\rho}(x,t)$$

[12]:
```
Schr_madelung = Schr_madelung.expand()
Schr_madelung
```

[12]:

$$\frac{\hbar^2\frac{\partial^2}{\partial x^2}\sqrt{\rho}(x,t)}{2m}+\frac{i\hbar\sqrt{\rho}(x,t)\frac{\partial^2}{\partial x^2}S(x,t)}{2}+i\hbar\frac{\partial}{\partial x}S(x,t)\frac{\partial}{\partial x}\sqrt{\rho}(x,t)+i\hbar\frac{\partial}{\partial t}\sqrt{\rho}(x,t)-g\sqrt{\rho}^3(x,t)- m\sqrt{\rho}(x,t)\frac{\partial}{\partial t}S(x,t)-\frac{m\sqrt{\rho}(x,t)\left(\frac{\partial}{\partial x}S(x,t)\right)^2}{2}-V(x,t)\sqrt{\rho}(x,t)$$

[13]:
```
Re_eq = Schr_madelung.coeff(sp.I, n=0)
Re_eq = sp.simplify(-Re_eq/(sqrt_rho*m))
Re_eq
```

[13]:

$$-\frac{\hbar^2\frac{\partial^2}{\partial x^2}\sqrt{\rho}(x,t)}{2m^2\sqrt{\rho}(x,t)}+\frac{g\sqrt{\rho}^2(x,t)}{m}+\frac{\partial}{\partial t}S(x,t)+\frac{\left(\frac{\partial}{\partial x}S(x,t)\right)^2}{2}+\frac{V(x,t)}{m}$$

The negative term corresponds with the quantum preassure contribution to the fluid. With some manipulation we get the superfluid hydrodynamic equations:

$$P_Q=\frac{\hbar^2}{2m^2}\frac{\nabla^2\sqrt{\rho}}{\sqrt{\rho}}=\frac{\hbar^2}{4m^2\rho}\left(\nabla^2\rho-\frac{(\nabla\rho)^2}{2\rho}\right)$$

[19]:
```
sp.expand(sp.diff(sp.sqrt(rho), x, 2)/sp.sqrt(rho))
```

[19]:

$$\frac{\frac{\partial^2}{\partial x^2}\rho(x,t)}{2\rho(x,t)}-\frac{\left(\frac{\partial}{\partial x}\rho(x,t)\right)^2}{4\rho^2(x,t)}$$

[14]:
```
Re_eq = Re_eq.subs({-hbar**2*sp.diff(sqrt_rho, x, 2)/(2*m**2*sqrt_rho): Pq})
Re_eq
```

[14]:

$$\frac{g\sqrt{\rho}^{2}(x,t)}{m} + \mathrm{P_Q}(x,t) + \frac{\partial}{\partial t}S(x,t) + \frac{\left(\frac{\partial}{\partial x}S(x,t)\right)^{2}}{2} + \frac{V(x,t)}{m}$$

[15]:
```
m*rho*sp.diff(Re_eq, x).subs({sp.diff(S, x): v})
```

[15]:

$$m\left(\frac{2g\sqrt{\rho}(x,t)\frac{\partial}{\partial x}\sqrt{\rho}(x,t)}{m} + v(x,t)\frac{\partial}{\partial x}v(x,t) + \frac{\partial}{\partial x}\mathrm{P_Q}(x,t) + \frac{\partial}{\partial t}v(x,t) + \frac{\frac{\partial}{\partial x}V(x,t)}{m}\right)\rho(x,t)$$

or:

$$m\rho\left(\frac{\partial v}{\partial t} + (v\cdot\nabla)v\right) = -m\rho\nabla P_Q - \rho\nabla V - g\nabla\rho$$

Notice that the pressure depends only on the density. This property makes the condensate a barotropic fluid; as a consequence, surfaces of constant pres- sure are also surfaces of constant density. The quantum pressure is a pure quantum effect, and vanishes if we set Planck's constant equal to zero. It has the same origin as the quantum kinetic energy, i.e. zero point motion, which creates a pressure that opposes any `squashing' or `bending' of the condensate. In a uniform condensate the quantum pressure is zero because n is constant.

This equation is very similar to the classical Euler equation for an inviscid fluid. If in addition, the trapping potential is absent ($V = 0$) then the equation reduces to the classical Euler equation, which describes the motion of a classical fluid without viscosity.