Python On Resonance (PyOR)

Everybody can simulate NMR

Author: Vineeth Thalakottoor

Email: vineethfrancis.physics@gmail.com

Tutorial 3: Initial Density Matrix Part 2

In previous tutorial, we saw how to generate the Equlibrium Initial Density Matrix for single spin half particle (Proton). In this tutorial we will see how to generate the Equlibrium Initial Density Matrix for single electron.

Load Python packages and define path to the source file "PythonOnResonance.py"

In [1]:

```
pathSource = '/media/HD2/Vineeth/PostDoc_Simulations/Github/PyOR_G/Source'
```

In [2]:

```
from IPython.display import display, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))
import sys
sys.path.append(pathSource)

import PythonOnResonance as PyOR

import time
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import rc
%matplotlib notebook
import sympy as sp
from sympy import *
```

Generating Spin System

In [3]:

```
"""
Define Spin quantum numbers of your spins in "Slist1".
Slist1[0] is spin of first particle and Slist1[1] is spin of second particle.
""";

Slist1 = [1/2]
```

In [4]:

```
"""
Define Planck constant equals 1.
Because NMR spectroscopists are more interested to write Energy in frequency units.
if False then hbarEQ1 = hbar
""";

hbarEQ1 = True
```

In [5]:

```
"""
Generate Spin Operators
""";

System = PyOR.Numerical_MR(Slist1,hbarEQ1)

"""
Sx, Sy and Sz Operators
""";
Sx,Sy,Sz = System.SpinOperator()

"""
S+ and S- Operators
""";
Sp,Sm = System.PMoperators(Sx,Sy)
```

Zeeman Hamiltonian in Lab Frame

In [6]:

```
"""
Gyromagnetic Ratio
Gamma = [Gyromagnetic Ratio spin 1, Gyromagnetic Ratio spin 1, ...]
""";
Gamma = [System.gammaE]

"""
Define the field of the spectromter, B0 in Tesla.
"""
B0 = 9.4

"""
Define the chemical Shift of individual spins
Offset = [chemical Shift spin 1, chemical Shift spin 1, ..]
"""
Offset = [20] # Offset frequency in Hz

"""
Function "LarmorF" give the list Larmor frequencies of individual spins in lab frame
"""
LarmorF = System.LarmorFrequency(Gamma,B0,Offset)

Hz = System.Zeeman(LarmorF,Sz)
```

```
Larmor Frequency in MHz:  [263455.54347774]
```

Initialize Density Matrix

In [7]:

```
"""
We will generate Initial Density Matrix in two ways:
First we will generate a density matrix as we prefer say, Sz.
Second we will create density matrix at thermal equlibrium

First Case
""";

Thermal_DensMatrix = False

if Thermal_DensMatrix:
    Hz_EnUnit = System.Convert_FreqUnitsTOEnergy(Hz)
    HT_approx = False # High Temperature Approximation is False
    T = 300 # Temperature in Kelvin
    rho_in = System.EqulibriumDensityMatrix(Hz_EnUnit,T,HT_approx)
    rhoeq = rho_in.copy()
else:
    rho_in = np.sum(Sz,axis=0) # Initial Density Matrix
    rhoeq = np.sum(Sz,axis=0)  # Equlibrium Density Matrix
    print("Trace of density metrix = ", np.trace(rho_in))
```

```
Trace of density metrix =  0j
```

In [8]:

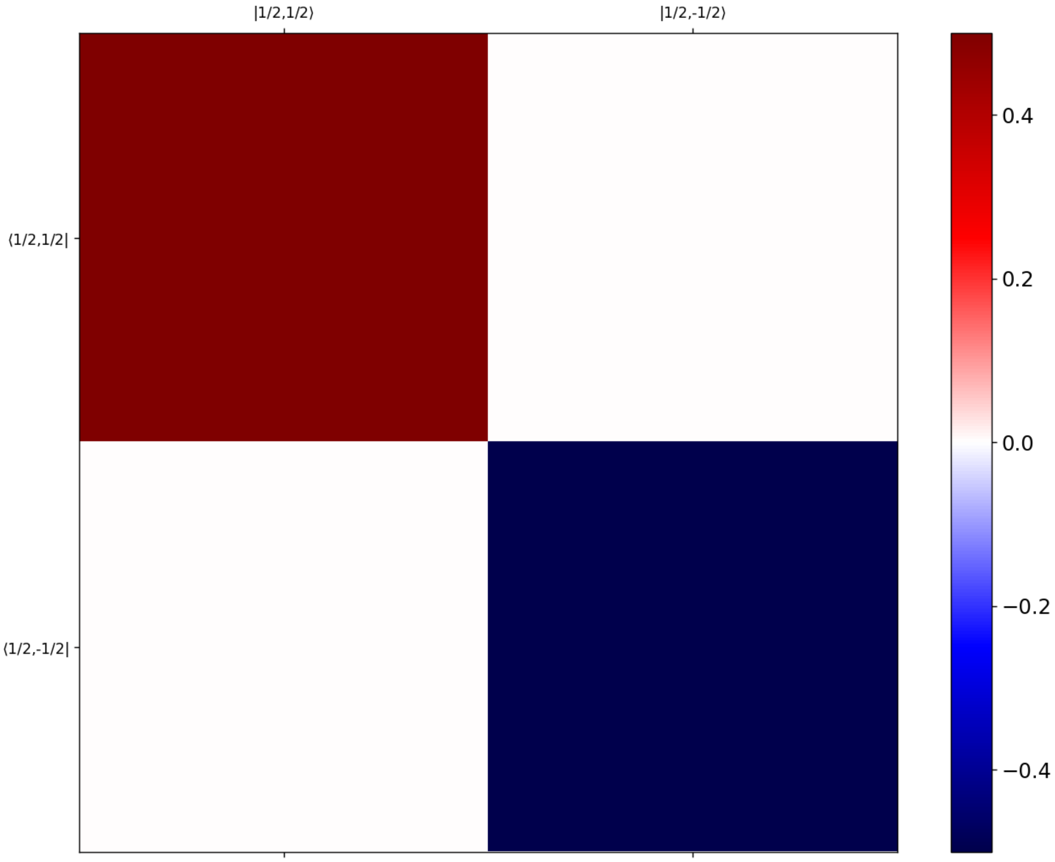
```
Matrix(rho_in)
```

Out[8]:

$$\begin{bmatrix} 0.5 & 0 \\ 0 & -0.5 \end{bmatrix}$$

In [9]:

```
System.MatrixPlot(1,rho_in.real)
```



```
/media/HD2/Vineeth/PostDoc_Simulations/Github/PyOR_G/Source/PythonOnResonance.py:669: User
Warning: FixedFormatter should only be used together with FixedLocator
  ax.set_xticklabels([''] + labelx,fontsize=10)
/media/HD2/Vineeth/PostDoc_Simulations/Github/PyOR_G/Source/PythonOnResonance.py:670: User
Warning: FixedFormatter should only be used together with FixedLocator
  ax.set_yticklabels([''] + labely,fontsize=10)
```

In [10]:

```
"""
Second Case: Initial Desnity Matrix at Thermal Equlibrium
""";

Thermal_DensMatrix = True
```

```
if Thermal_DensMatrix:
    Hz_EnUnit = System.Convert_FreqUnitsTOEnergy(Hz)
    HT_approx = False # High Temperature Approximation is False
    T = 1.2 # Temperature in Kelvin
    rho_in = System.EqulibriumDensityMatrix(Hz_EnUnit,T,HT_approx)
    rhoeq = rho_in.copy()
else:
    rho_in = np.sum(Sz,axis=0) # Initial Density Matrix
    rhoeq = np.sum(Sz,axis=0)  # Equlibrium Density Matrix
    print("Trace of density metrix = ", np.trace(rho_in))
```

```
Trace of density metrix =  0.9999999999999999
```

In [11]:

```
"""
Population Difference between two states
"""
print("Population difference Between the two states = ", rho_in[1][1] - rho_in[0][0])
```

```
Population difference Between the two states =  (0.999946865209809+0j)
```

In [12]:

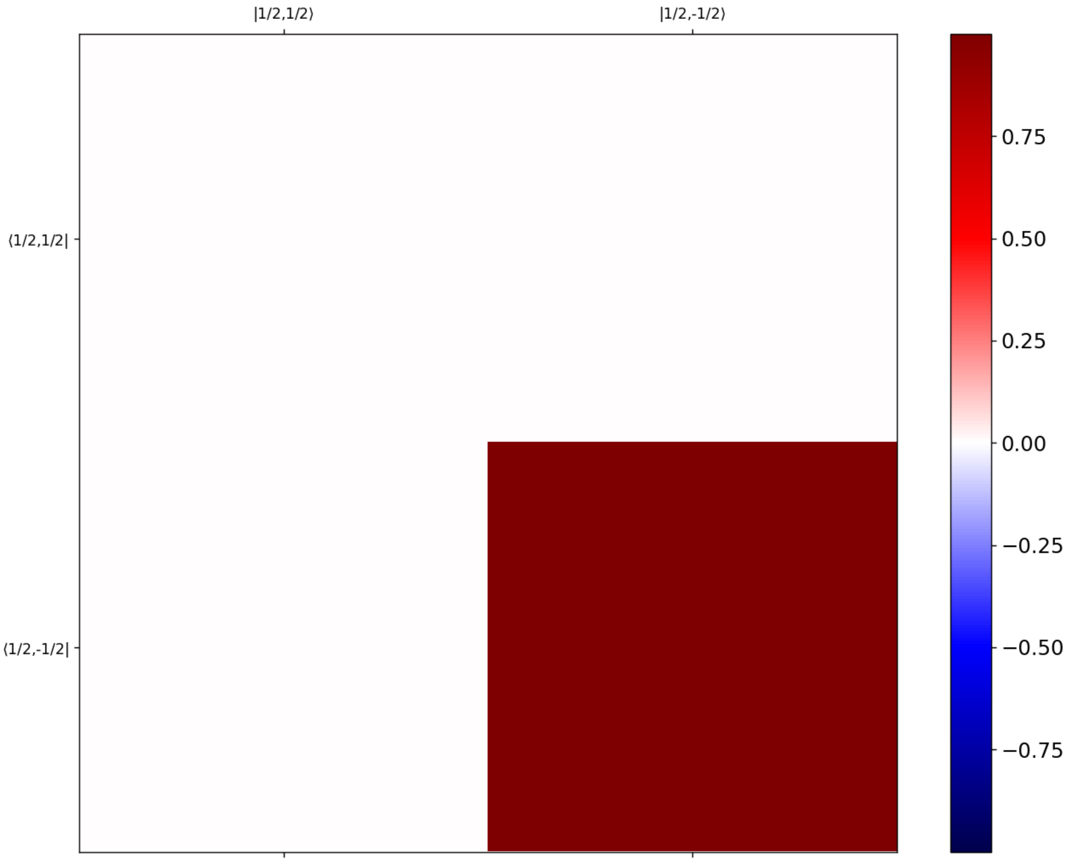
```
Matrix(rho_in)
```

Out[12]:

$$\begin{bmatrix} 2.65673950954855 \cdot 10^{-5} & 0 \\ 0 & 0.999973432604904 \end{bmatrix}$$

In [13]:

```
System.MatrixPlot(2,rho_in.real)
```



Next tutorial: Initial Density Matrix Part 3

In this lecture you will see how to generate initial density matrix for two spin system a electron and a proton at thermal equlibrium

Any suggestion? write to me

If you see something is wrong please write to me, so that the PyOR can be error free.

vineethfrancis.physics@gmail.com

In [ ]:

```
```