Python On Resonance (PyOR)

Everybody can simulate NMR

Author: Vineeth Thalakottoor

Email: vineethfrancis.physics@gmail.com

Tutorial 16: Cartesian Operator Basis

In this tutorial we will see how to make cartesian operator basis for any number of spin half particles.

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

In [21]:
```
pathSource = '/media/HD2/Vineeth/PostDoc_Simulations/Github/PyOR_V1/Source'
```

In [22]:
```
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 *
```

In [23]:
```
Slist1 = [1/2,1/2]
hbarEQ1 = True
System = PyOR.Numerical_MR(Slist1,hbarEQ1)
```

In [24]:
```
Index = False
Normal = True
Basis, dic = System.ProductOperators_SpinHalf_Cartesian(Index,Normal)
```

In [25]:
```
dic
```

Out[25]:
```
['Id1 Id2 ',
 'Id1 Ix2 ',
 'Id1 Iy2 ',
 'Id1 Iz2 ',
 'Ix1 Id2 ',
 'Ix1 Ix2 ',
 'Ix1 Iy2 ',
 'Ix1 Iz2 ',
 'Iy1 Id2 ',
```

```
 'Iy1 Ix2 ',
 'Iy1 Iy2 ',
 'Iy1 Iz2 ',
 'Iz1 Id2 ',
 'Iz1 Ix2 ',
 'Iz1 Iy2 ',
 'Iz1 Iz2 ']
```

In [26]:
```
Matrix(Basis[6])
```

Out[26]:
$$\begin{bmatrix} 0 & 0 & 0 & -0.5i \\ 0 & 0 & 0.5i & 0 \\ 0 & -0.5i & 0 & 0 \\ 0.5i & 0 & 0 & 0 \end{bmatrix}$$

In [27]:
```
System.OP_InnerProduct(Basis[0],Basis[0])
```

Out[27]:
```
1.0
```

In [28]:
```
OpB = System.String_to_Matrix(dic, Basis)
```

```
['', 'Ix2', 'Iy2', 'Iz2', 'Ix1', 'Ix1Ix2', 'Ix1Iy2', 'Ix1Iz2', 'Iy1', 'Iy1Ix2', 'Iy1Iy2',
'Iy1Iz2', 'Iz1', 'Iz1Ix2', 'Iz1Iy2', 'Iz1Iz2']
```

In [29]:
```
OpB["Ix1Iy2"]
```

Out[29]:
```
array([[0.+0.j , 0.+0.j , 0.+0.j , 0.-0.5j],
       [0.+0.j , 0.+0.j , 0.+0.5j, 0.+0.j ],
       [0.+0.j , 0.-0.5j, 0.+0.j , 0.+0.j ],
       [0.+0.5j, 0.+0.j , 0.+0.j , 0.+0.j ]], dtype=complex64)
```

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

Python On Resonance (PyOR)

Everybody can simulate NMR

Author: Vineeth Thalakottoor

Email: vineethfrancis.physics@gmail.com

Tutorial 16: Plus Minus Z Operator Basis

In this tutorial we will see how to make plus minus z operator basis for any number of spin half particles.

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

In [1]:
```
pathSource = '/media/HD2/Vineeth/PostDoc_Simulations/Github/PyOR_V1/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 *
```

In [3]:
```
Slist1 = [1/2,1/2]
hbarEQ1 = True
System = PyOR.Numerical_MR(Slist1,hbarEQ1)
```

In [4]:
```
Sx,Sy,Sz = System.SpinOperator()
Sp,Sm = System.PMoperators(Sx,Sy)
```

In [5]:
```
sort = 'negative to positive'
Index = False
Normal = True
Basis, coh, dic = System.ProductOperators_SpinHalf_PMZ(sort,Index,Normal)
```

In [6]:
```
dic
```

Out[6]:
```
['Im1 Im2 ',
 'Id1 Im2 ',
 'Im1 Id2 ',
```

```
'Im1 Iz2 ',
'Iz1 Im2 ',
'Id1 Id2 ',
'Id1 Iz2 ',
'Im1 Ip2 ',
'Iz1 Id2 ',
'Iz1 Iz2 ',
'Ip1 Im2 ',
'Id1 Ip2 ',
'Iz1 Ip2 ',
'Ip1 Id2 ',
'Ip1 Iz2 ',
'Ip1 Ip2 ']
```

In [7]:
```
coh
```

Out[7]:
```
[-2, -1, -1, -1, -1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2]
```

In [12]:
```
Matrix(Basis[0])
```

Out[12]:
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1.0 & 0 & 0 & 0 \end{bmatrix}$$

In [9]:
```
Matrix(System.Adjoint(Basis[0]))
```

Out[9]:
$$\begin{bmatrix} 0 & 0 & 0 & 1.0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

In [10]:
```
OpB = System.String_to_Matrix(dic, Basis)
```

```
['Im1Im2', 'Im2', 'Im1', 'Im1Iz2', 'Iz1Im2', '', 'Iz2', 'Im1Ip2', 'Iz1', 'Iz1Iz2', 'Ip1Im
2', 'Ip2', 'Iz1Ip2', 'Ip1', 'Ip1Iz2', 'Ip1Ip2']
```

In [11]:
```
Matrix(OpB["Im1Im2"])
```

Out[11]:
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1.0 & 0 & 0 & 0 \end{bmatrix}$$

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

Python On Resonance (PyOR)

Everybody can simulate NMR

Author: Vineeth Thalakottoor

Email: vineethfrancis.physics@gmail.com

Tutorial 16: Spherical tensor Operator Basis

In this tutorial we will see how to make spherical tensor operator basis for any number of spin half particles.

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

In [26]:
```
pathSource = '/media/HD2/Vineeth/PostDoc_Simulations/Github/PyOR_V1/Source'
```

In [27]:
```
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 *
```

In [28]:
```
Slist1 = [1/2,1/2]
hbarEQ1 = True
System = PyOR.Numerical_MR(Slist1,hbarEQ1)
```

In [29]:
```
sort = 'negative to positive'
Index = False
Basis_half, coherence_half, dic_half = System.ProductOperators_SpinHalf_SphericalTensor(sc
```

```
Coherence Order:  [0, -1, 0, 1]
LM state:  [(0, 0), (1, -1), (1, 0), (1, 1)]
```

In [30]:
```
Matrix(Basis_half[0])
```

Out[30]:
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1.0 & 0 & 0 & 0 \end{bmatrix}$$

In [31]:
```
print("Coherence order: ", coherence_half)
```

```
Coherence order:  [-2, -1, -1, -1, -1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2]
```

In [32]:
```
print(dic_half)
```

```
['Im1 Im2 ', 'Id1 Im2 ', 'Im1 Id2 ', 'Im1 Iz2 ', 'Iz1 Im2 ', 'Id1 Id2 ', 'Id1 Iz2 ', 'Im1
Ip2 ', 'Iz1 Id2 ', 'Iz1 Iz2 ', 'Ip1 Im2 ', 'Id1 Ip2 ', 'Iz1 Ip2 ', 'Ip1 Id2 ', 'Ip1 Iz2 ',
'Ip1 Ip2 ']
```

In [33]:
```
"""
Orthonormalization check
""";
System.OP_InnerProduct(Basis_half[0],Basis_half[0])
```

Out[33]:
```
0.9999999999999996
```

In [34]:
```
"""
Orthonormalization check
""";
System.OP_InnerProduct(Basis_half[0],Basis_half[1])
```

Out[34]:
```
0.0
```

In [35]:
```
OpB = System.String_to_Matrix(dic_half, Basis_half)
```

```
['Im1Im2', 'Im2', 'Im1', 'Im1Iz2', 'Iz1Im2', '', 'Iz2', 'Im1Ip2', 'Iz1', 'Iz1Iz2', 'Ip1Im
2', 'Ip2', 'Iz1Ip2', 'Ip1', 'Ip1Iz2', 'Ip1Ip2']
```

In [36]:
```
OpB["Im1Im2"]
```

Out[36]:
```
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [1., 0., 0., 0.]])
```

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