rY_operator

December 7, 2023

1 Program to compute $rY^1_\mu(\Omega)$ operator expansion in $SU(3)$ tensors for actinides

The expansion is given by

$$\left[r_\sigma Y^1_\mu(\hat{r}_\sigma)\right]^{L_o=J_o=1}_{M_{J_o}} = \frac{(-1)^\eta}{\sqrt{3}} \sum_{(\lambda_o,\mu_o)k_o} \sum_{l,l'} \left\langle \eta', l', \frac{1}{2} \middle\| r_\sigma Y^1(\hat{r}_\sigma) \middle\| \eta, l, \frac{1}{2} \right\rangle \times \langle (\eta', 0), 0, l'; (0, \eta), 0, l || (\lambda_o, \mu_o), k_o, 1 \rangle_{\rho_o=1} \left\{ a^\dagger_{(\ldots}\right.$$

where $\sigma = \pi, \nu$, the operator acts on two shells η, η' and the Edmunds convention on Wigner-Eckart theorem is used.

[1]:
```
# Libraries
import numpy as np
import pandas as pd
```

[2]:
```
# Coefficients and reduced matrix elements
SU3coeffs = pd.read_csv("./adat_coupling_coeffs_actinides/coeffs.csv")
rYrmes = pd.read_csv("./rY_double_bar_rmes_actinides/rY_double_bar_rmes.csv")
couplings = pd.read_csv("./adat_couplings_actinides/couplings.csv")
```

[3]:
```
def SU3_rY_expansion(eta1, eta2):

    # Orbital angular momentum values in each shell
    L1 = [l for l in range(eta1,-1,-2)]
    L2 = [l for l in range(eta2,-1,-2)]

    # Double bar reduced matrix elements <  1, l1 || rY^1(Ω) ||  2, l2 >
    tensor_rYrmes = rYrmes[(rYrmes["eta1"]==eta1) & (rYrmes["eta2"]==eta2)]

    # Extraction of ( _o,  _o) irreps of the ( 1, 0)x(0,  2) couplings
    tensor_irreps = couplings[(couplings["lam1"]==eta1) &␣
↪(couplings["mu2"]==eta2)][["lam","mu"]].to_numpy()

    # Extraction of < ( 1, 0), 0, l1; (0,  2),0, l2|| ( _o,  _o),K_o, l_o >␣
↪coefficients
    tensor_coeffs = pd.DataFrame()
    for irrep in tensor_irreps:
```

```
        tensor_coeffs = pd.concat([tensor_coeffs,SU3coeffs[(SU3coeffs["lam1"]==eta1) & (SU3coeffs["mu2"]==eta2) &
                                                (SU3coeffs["lam3"]==irrep[0]) & (SU3coeffs["mu3"]==irrep[1])]])

    # Calculation of sum over l1 and l2
    expansion = []

    for irrep in tensor_irreps:

        s = 0

        # Sum over l1 and l2
        for l1 in L1:
            for l2 in L2:

                # Gets rY double bar reduced matrix element
                GETrYrme = tensor_rYrmes[(tensor_rYrmes["l1"]==l1) & (tensor_rYrmes["l2"]==l2)]
                rYrme = 0 if GETrYrme.empty else float(GETrYrme["rme"])

                #if rYrme == 0:
                #    print("< ",eta1,",",l1,"|| rY ||",eta2,",",l2,"> = 0")

                # Gets coefficient
                GETrYcoeff = tensor_coeffs[(tensor_coeffs["l1"]==l1) & (tensor_coeffs["l2"]==l2) &
                                        (tensor_coeffs["lam3"]==irrep[0]) & (tensor_coeffs["mu3"]==irrep[1])]
                rYcoeff = 0 if GETrYcoeff.empty else float(GETrYcoeff["coeff"])

                #if rYcoeff == 0:
                #    print("<(",eta1,",0),",l1,";(0,",eta2,"),  ",l2," ||(",irrep[0],",",irrep[1],"), 1 > = 0")

                # Sumation
                s += rYrme*rYcoeff

        expansion.append([(-1)**eta2*s/np.sqrt(3), irrep])

    return expansion

def print_format(expansion):

    string = ""
```

```
    for term in expansion:
        if term[0] <= 0:
            string += " {0}{{a+a}}({1},{2})".format(round(term[0],4), term[1][0], term[1][1])
        else:
            string += " + {0}{{a+a}}({1},{2})".format(round(term[0],4), term[1][0], term[1][1])

    print(string)
```

2 Proton operator

This operator destroys a proton in $\eta = 6$ and creates one proton in $\eta' = 5$. The expansion is given by

$$\left[r_\pi Y^1_\mu(\hat{r}_\pi)\right]^{L_o=J_o=1}_{M_{J_o}} = \frac{1}{\sqrt{3}} \sum_{(\lambda_o,\mu_o)k_o} \sum_{l,l'} \left\langle \eta' = 5, l', \frac{1}{2} \middle\| r_\pi Y^1(\hat{r}_\pi) \middle\| \eta = 6, l, \frac{1}{2} \right\rangle \times \langle (5,0),0,l';(0,6),0,l||(\lambda_o,\mu_o),k_o,1\rangle_{\rho_o=}$$

```
[4]: protonrY = SU3_rY_expansion(5,6)
     print(protonrY,  "\n")
     print_format(protonrY)
```

```
[[-0.3709121274256688, array([5, 6])], [0.20574642859123457, array([4, 5])],
[0.5315093604157286, array([3, 4])], [-0.4596360788911335, array([2, 3])],
[-1.7435442029972494, array([1, 2])], [1.7236270942936072, array([0, 1])]]

 -0.3709{a+a}(5,6) + 0.2057{a+a}(4,5) + 0.5315{a+a}(3,4) -0.4596{a+a}(2,3)
-1.7435{a+a}(1,2) + 1.7236{a+a}(0,1)
```

3 Neutron operator

This operator destroys a proton in $\eta = 7$ and creates one proton in $\eta' = 6$. The expansion is given by

$$\left[r_\nu Y^1_\mu(\hat{r}_\nu)\right]^{L_o=J_o=1}_{M_{J_o}} = -\frac{1}{\sqrt{3}} \sum_{(\lambda_o,\mu_o)k_o} \sum_{l,l'} \left\langle \eta' = 6, l', \frac{1}{2} \middle\| r_\nu Y^1(\hat{r}_\nu) \middle\| \eta = 7, l, \frac{1}{2} \right\rangle \times \langle (6,0),0,l';(0,7),0,l||(\lambda_o,\mu_o),k_o,1\rangle_{\rho_o}$$

```
[5]: neutronrY = SU3_rY_expansion(6,7)
     print(neutronrY,  "\n")
     print_format(neutronrY)
```

```
[[-0.34101916024129764, array([6, 7])], [-0.24401383796675546, array([5, 6])],
[0.38332444381144287, array([4, 5])], [0.5486539571391357, array([3, 4])],
[-0.6483787072210015, array([2, 3])], [-2.076417292237013, array([1, 2])],
[2.1612625533664307, array([0, 1])]]

 -0.341{a+a}(6,7) -0.244{a+a}(5,6) + 0.3833{a+a}(4,5) + 0.5487{a+a}(3,4)
-0.6484{a+a}(2,3) -2.0764{a+a}(1,2) + 2.1613{a+a}(0,1)
```

[ ]:

```
```

[ ]:

```
```