basis_bug

April 24, 2024

[2]:
```
from pyscf import scf,gto,dft,ao2mo
import numpy as np
from gbasis.integrals.electron_repulsion import electron_repulsion_integral
from gbasis.wrappers import from_pyscf
from gbasis.parsers import parse_gbs, make_contractions
```

[3]:
```
#Pyscf Water Molecule
mol = gto.M(
    atom = '''
        O    0.00000000    0.00000000    0.11767600
        H    0.00000000    0.76026300   0.47070300
        H    0.00000000   -0.76026300   -0.47070300
            ''',
)
mol.unit='B' # or 'B'
mol.basis = 'sto-6g'
mol.cart=False
#mol.verbose=5
mol.spin=0
mol.charge=0
mol.symmetry=False
mol.build()

#Basis from Pyscf
pb_basis =  from_pyscf(mol)
```

[4]:
```
#Build Water Via gbasis
basis_file ="../Downloads/sto-6g.1.gbs"
atcoords=np.array([[0.00000000, 0.00000000, 0.11767600],
                   [0.00000000, 0.76026300, 0.47070300],
                   [0.00000000, -0.76026300, -0.47070300],
                   ])
atoms = ['O', 'H', 'H']

basis_dict = parse_gbs(basis_file)

#basis from Pyscf
```

```
gb_basis = make_contractions(basis_dict, atoms, atcoords, "p")
```

[5]:
```
for i, (shell1,shell2) in enumerate(zip(pb_basis,gb_basis)):
    print(f"Contraction shell #{i}")
    print(f"   Center: {shell1.coord==shell2.coord}")
    print(f"   Angular momentum: {shell1.angmom==shell2.angmom}")
    print(f"   Primitive coefficients {np.isclose(shell1.coeffs.T,shell2.coeffs.
 ↪T)}")
    print(f"   Primitive exponents {np.isclose(shell1.exps,shell2.exps)}")
    print(f"   Primitive normalization constant {np.isclose(shell1.
 ↪norm_cont,shell2.norm_cont)}")
    print()
```

```
Contraction shell #0
   Center: [ True  True  True]
   Angular momentum: True
   Primitive coefficients [[ True  True  True  True  True  True]]
   Primitive exponents [ True  True  True  True  True  True]
   Primitive normalization constant [[ True]]

Contraction shell #1
   Center: [ True  True  True]
   Angular momentum: True
   Primitive coefficients [[ True  True  True  True  True  True]]
   Primitive exponents [ True  True  True  True  True  True]
   Primitive normalization constant [[ True]]

Contraction shell #2
   Center: [ True  True  True]
   Angular momentum: True
   Primitive coefficients [[ True  True  True  True  True  True]]
   Primitive exponents [ True  True  True  True  True  True]
   Primitive normalization constant [[ True  True  True]]

Contraction shell #3
   Center: [ True  True  True]
   Angular momentum: True
   Primitive coefficients [[ True  True  True  True  True  True]]
   Primitive exponents [ True  True  True  True  True  True]
   Primitive normalization constant [[ True]]

Contraction shell #4
   Center: [ True  True  True]
   Angular momentum: True
   Primitive coefficients [[ True  True  True  True  True  True]]
   Primitive exponents [ True  True  True  True  True  True]
   Primitive normalization constant [[ True]]
```

[6]:
```
#Evaluate coulomb integrals

coulomb_gb =␣
 ↪electron_repulsion_integral(gb_basis,transform=None,notation="chemist")
coulomb_pb =␣
 ↪electron_repulsion_integral(pb_basis,transform=None,notation="chemist")
```

[7]:
```
np.isclose(coulomb_gb[0,0],coulomb_pb[0,0])
```

[7]:
```
array([[ True,  True,  True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True,  True,  True],
       [ True,  True,  True,  True,  True, False, False],
       [ True,  True,  True,  True,  True, False, False],
       [ True,  True,  True,  True,  True, False, False],
       [ True,  True, False, False, False,  True,  True],
       [ True,  True, False, False, False,  True,  True]])
```

[8]:
```
coulomb_gb[0,0],coulomb_pb[0,0]
```

[8]:
```
(array([[ 4.78749087e+00,  6.75883992e-01,  0.00000000e+00,
         -2.45393544e-18,  0.00000000e+00,  5.57141577e-01,
          4.82241466e-01],
        [ 6.75883992e-01,  1.11737934e+00,  0.00000000e+00,
          1.66346011e-17,  0.00000000e+00,  8.53815151e-01,
          7.90817027e-01],
        [ 0.00000000e+00,  0.00000000e+00,  1.11738128e+00,
          0.00000000e+00,  0.00000000e+00,  3.59142920e-01,
         -3.16887860e-01],
        [-2.45393544e-18,  1.66346011e-17,  0.00000000e+00,
          1.11738128e+00,  0.00000000e+00,  1.66767484e-01,
         -2.45244293e-01],
        [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
          0.00000000e+00,  1.11738128e+00,  0.00000000e+00,
          0.00000000e+00],
        [ 5.57141577e-01,  8.53815151e-01,  3.59142920e-01,
          1.66767484e-01,  0.00000000e+00,  8.80546496e-01,
          4.95511323e-01],
        [ 4.82241466e-01,  7.90817027e-01, -3.16887860e-01,
         -2.45244293e-01,  0.00000000e+00,  4.95511323e-01,
          8.24028784e-01]]),
 array([[ 4.78749087e+00,  6.75883991e-01,  0.00000000e+00,
          0.00000000e+00,  1.72150413e-18,  5.57141577e-01,
          4.82241466e-01],
        [ 6.75883991e-01,  1.11737934e+00,  0.00000000e+00,
          0.00000000e+00,  1.53423846e-17,  8.53815151e-01,
```

```
  7.90817027e-01],
[ 0.00000000e+00,  0.00000000e+00,  1.11738128e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  1.11738128e+00,  0.00000000e+00,  3.59142920e-01,
 -3.16887860e-01],
[ 1.72150413e-18,  1.53423846e-17,  0.00000000e+00,
  0.00000000e+00,  1.11738128e+00,  1.66767484e-01,
 -2.45244293e-01],
[ 5.57141577e-01,  8.53815151e-01,  0.00000000e+00,
  3.59142920e-01,  1.66767484e-01,  8.80546496e-01,
  4.95511323e-01],
[ 4.82241466e-01,  7.90817027e-01,  0.00000000e+00,
 -3.16887860e-01, -2.45244293e-01,  4.95511323e-01,
  8.24028784e-01]]))
```

[9]:
```
print(np.isclose(np.round(coulomb_gb[5][0],8),np.round(coulomb_pb[1][0],8)))
```

```
[[False False False False  True False False]
 [False False False False  True False False]
 [False False False False  True False False]
 [False False False False  True False False]
 [ True  True  True  True False False False]
 [False False False False False False False]
 [False False False False False False False]]
```