The libint-eigen interface

Laurent Lemmens

Thursday 17th August, 2017 17:34

Contents

1 Terminology

In the LibInt2 basis set context, there is some terminology that should be cleared up. Let's start from the beginning. A *primitive Gaussian* is a function of the following mathematical form:

$$\varphi_{\zeta,l_x,l_y,l_z,\mathbf{R}}(x,y,z) = N(x-X)^{l_z}(y-Y)^{l_y}(z-Z)^{l_z}\exp\left(-\zeta r^2\right), \tag{1}$$

in which

$$r^2 = x^2 + y^2 + z^2\,. \tag{2}$$

l_x, l_y and l_z are called the angular momenta of the primitive. The sum

$$l = l_x + l_y + l_z \tag{3}$$

of the angular momenta determines the type of primitive: $l = 0$ refers to an s-type, $l = 1$ to a p-type, etc. (analogously to naming of the eigenfunctions of the hydrogen atom). The position vector $\mathbf{R}$ specifies the center of the primitive.

Often, a linear combination (also known as a *contraction*) of primitives is taken, leading to a *contracted GTO* (cGTO). These are the functions that are used as basis functions. (Note that a single primitive can also be used as a basis function, in which case the linear combination is just one times that primitive.) *Molecular orbitals* (MOs) are written as a linear combination of basis functions (cGTOs), which in turn serve as a one-electron basis to antisymmetrize into the many-electron basis of the *Slater determinants* (SDs). [1]

2 Are the contracted basis functions normalized?

When constructing a `libint2::BasisSet` object as in say

```
libint2::BasisSet obs ("STO-3G", atoms);
```

the corresponding file `sto-3g.g94` is read (which is located at your `LIBINT_DATA_PATH` environment variable), in which LibInt2 finds the exponents and contraction coefficients for the given basis for a given element.

In `libint2/basis.h`, we can see the following code (edited for brevity):

```
static ... read_g94_basis_library(...){
    ...
    ref_shells[Z].push_back(
        libint2::Shell{...}
```

```
        )
    ...
}
```

which calls a specific constructor of `libint2::Shell`:

```
Shell(...) {
    // embed normalization factors into contraction
   coefficients
    renorm();
}
```

that makes sure that the CGTO is normalized by including the normalization factor in the contraction coefficients. So, **yes**, LibInt2 internally works with normalized basis functions.

References

[1] F. Jensen. *Introduction to Computational Chemistry*. John Wiley & Sons, LTD, second edition, 2007.