qcrab documentation

Brent R. Westbrook*

E-mail: brent@bwestbro.com

Overlap Integrals

As shown in Ref. 1, the expression for an overlap integral of two primitive Gaussian functions, S, is given by:

$$S = E_{\mathbf{AB}} S_x S_y S_z, \tag{1}$$

where $E_{\mathbf{AB}} = \exp\left[-\frac{\alpha\beta}{\alpha+\beta}|\mathbf{A}-\mathbf{B}|^2\right]$ and S_x, for example, is given by:

$$S_x = \sqrt{\frac{\pi}{\alpha+\beta}} \sum_{i_x=0}^{a_x} \sum_{j_x=0}^{b_x} \binom{a_x}{i_x} \binom{b_x}{j_x} \frac{(i_x+j_x+1)!!}{[2(\alpha+\beta)]^{(i_x+j_x)/2}} (P_x - A_x)^{a_x-i_x} (P_x - B_x)^{b_x-j_x} \tag{2}$$

As a note, $|\mathbf{A}-\mathbf{B}|^2$ is the magnitude of $\mathbf{A}-\mathbf{B}$ ($\sqrt{\sum_i (a_i-b_i)^2}$) squared ($\sum_i (a_i-b_i)^2$), or $(\mathbf{A}-\mathbf{B})\cdot(\mathbf{A}-\mathbf{B})$.

In all of these expressions, α is the exponent for the first primitive Gaussian function, β is the exponent for the second primitive Gaussian, $\mathbf{P} = \frac{\alpha\mathbf{A}+\beta\mathbf{B}}{\alpha+\beta}$, $\mathbf{A}$ and $\mathbf{B}$ are the origins of the basis functions (the Cartesian coordinates of the nuclei they are centered on), and a_x and b_x are the x components of the angular momentum for the Gaussian functions. For example, in an s orbital, $l = 0$, and l_x, l_y, and l_z are given by the tuple (0, 0, 0), while for a p orbital, there are three combinations for which $l = 1 = l_x + l_y + l_z$: (1, 0, 0), (0, 1, 0),

and (0, 0, 1). Consequently, for shells composed of multiple primitive Gaussian functions, we must loop over these combinations, as well as the primitives.

In the code for `overlap_ints`, I first loop over the shells and collect these possibilities, as shown in Listing 1.

Listing 1: Collecting l values for shells

```
let mut ls = Vec::new();
let mut ss = Vec::new();
for (i, shell) in self.0.iter().enumerate() {
    let l = match shell.contr[0].l {
        0 => vec![(0, 0, 0)],
        1 => vec![(1, 0, 0), (0, 1, 0), (0, 0, 1)],
        2 => vec![
            (2, 0, 0),
            (0, 2, 0),
            (0, 0, 2),
            (0, 1, 1),
            (1, 1, 0),
            (1, 0, 1),
        ],
        _ => panic!("unmatched l value {}", shell.contr[0].l),
    };
    let s = vec![i; l.len()];
    ls.extend(l);
    ss.extend(s);
}
```

With the combinations of angular momenta stored in `ls` and the shells associated with them in `ss`, we can iterate over them, effectively iterating over the orbitals themselves. For each combination of orbitals, we compute the overlap integral for the orbitals, S_{ij}, as the sum over primitives, where N is the number of primitives in shell i, and M is the number of primitives in shell j:

$$S_{ji} = S_{ij} = \sum_{n}^{N} \sum_{m}^{M} c_n c_m E_{\mathbf{AB}} S_x S_y S_z \tag{3}$$

Recall that $E_{\mathbf{AB}}$ includes α and β, which are subscripted by n and m in this sum, so $E_{\mathbf{AB}}$ does depend on n and m. Similarly, α and β in Eqn. 2 are α_n and β_m, and $\mathbf{P}$ must

also be computed on each iteration, again because of its dependence on α and β.

References

(1) Hô, M.; Hernández-Pérez, J. M. Evaluation of Gaussian Molecular Integrals I. Overlap Integrals. *The Mathematica Journal* **2012**, *14*.