Array-Oriented Computation of Electron Matrix Elements

David Dams

August 11, 2024

Contents

1 Theory

The non-relativistic electronic Hamiltonian can be expressed as:

$$H = T_{ab}a^\dagger b + V_{ab}a^\dagger b + U_{abcd}a^\dagger b^\dagger cd \tag{1}$$

where T_{ab}, V_{ab}, and U_{abcd} are the kinetic, nuclear, and interaction matrix elements, respectively and the Einstein summation convention is used. The non-orthonormality of the basis can be accounted for via the overlap matrix S, with elements:

$$\{a^\dagger, b\} = S_{ab} \tag{2}$$

Takaeta et al. have derived non-recursive expressions for each of these matrices in a Gaussian basis. These expressions can be recast in terms of SIMD-friendly array-oriented program-

ming to work efficiently with JAX. Loosely speaking, this means turning loops into JAX builtins, such as matrix products and convolutions.

2 Primitive and Contracted Matrix Elements

A primitive gaussian $p_i(\vec{A}, \alpha, l_x, l_y, l_z)$ is defined as

$$p_i(\vec{A}, \alpha, l_x, l_y, l_z) = (x - A_x)^{l_x}(y - A_y)^{l_y}(z - A_z)^{l_z} e^{-\alpha(\vec{A}-\vec{r})^2} \tag{3}$$

A contracted basis vector v_i of a Gaussian basis is represented by a linear combination of primitive Gaussians.

$$v_i = \sum_j c_{ij} p_j \tag{4}$$

Consequently, the primitive and contracted (c) one-electron matrix elements (a matrix with only two indices) are related by

$$M^{(c)}_{ab} = \sum_{ij} c_{ai} c_{bj} M_{ij} \tag{5}$$

And the two-electron matrix elements by

$$U^{(c)}_{abcd} = \sum_{ijkl} c_{ai} c_{bj} c_{ck} c_{dl} U_{ijkl} \tag{6}$$

The JAX-friendly implementation of these equations is quite straightforward. Challenging is computing primitive matrix elements, discussed next.

Since our expressions will involve at most four gaussians, we number them from 1 to 4 and consider the means $(\vec{A}, \vec{B}, \vec{C}, \vec{D})$ and "inverse standard deviations" $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, while we denote the angular momenta of the i-th gaussian as l_i, such that $l_i = (l_{i,x}, l_{i,y}, l_{i,z})$. In this way $M[12], U[1234]$, denotes the primitive gaussian matrix element between gaussians 1,2 and 1,2,3,4, respectively.

To prepare, we first define a few common variables and operators.

$$
\begin{aligned}
\gamma = \gamma_1 &= \alpha_1 + \alpha_2 \\
\gamma_2 &= \alpha_3 + \alpha_4 \\
f_j(l_1, l_2, a, b) &= \partial_x^j (a+x)^{l_1} (a+x)^{l_2}|_{x=0} \\
[x] &= \lfloor x + 1 \rfloor \\
\vec{P} &= \frac{\alpha_1 \vec{A} + \alpha_2 \vec{B}}{\alpha_1 + \alpha_2} \\
\vec{Q} &= \frac{\alpha_3 \vec{A} + \alpha_4 \vec{B}}{\alpha_3 + \alpha_4} \\
d_{xy} &= \vec{y} - \vec{x} \\
\text{Conv}_3 : a, b, c &\to \text{Conv}[\text{Conv}[a, b], c]
\end{aligned}
$$

Additionally, we denote the boys function with $\text{boys}(I, x)$ and an "array inflation function" as

$$\text{inflate}(a)_{2I} = a_I \tag{7}$$
$$\text{inflate}(a)_{2I+1} = 0 \tag{8}$$

3 Overlap

The overlap matrix elements can be expressed as a scalar times a product of inner products.

$$S[12] = a \cdot \prod_{i \leq 3} \sum_{I=0}^{[l_{1,i}+l_{2,i}]} b_{I,i}(l_1, l_2, \vec{d}_{AP}, \vec{d}_{BP}) c_I \tag{9}$$

where

$$
\begin{aligned}
a &= \frac{\pi}{\gamma}^{3/2} e^{-\alpha_1 \alpha_2 \vec{d}_{AB}^2 / \gamma} \\
b_{I,i} &= f_{2I}(l_{1,i}, l_{2,i}, \vec{d}_{AP,i}, \vec{d}_{BP,i}) \\
c_i &= \frac{(2i-1)!!}{(2\gamma)^i}
\end{aligned}
$$

4 Kinetic

The kinetic matrix elements expressed in terms of overlap matrix elements can be implemented directly from the paper, eqn. 2.14.

In principle, they can also be obtained by differentiating the overlap matrices. If we denote by $p_{\vec{\mu_1},\sigma_1}(x)p_{\vec{\mu_2},\sigma_2}(x)$ a product of two gaussians and apply shift vectors $\vec{a},\vec{b}$ to the mean of the second one, we have

$$\mathrm{div}_b\mathrm{grad}_a \int dx p_{\mu_1,\sigma_1}(x)p_{\mu_2+a+b,\sigma_2}(x)|_{a=b=0} = \int dx p_{\mu_1,\sigma_1}(x)\mathrm{div}_b\mathrm{grad}_a p_{\mu_2,\sigma_2}(x+a+b)|_{a=b=0}$$
$$= \int dx p_{\mu_1,\sigma_1}(x)\Delta p_{\mu_2,\sigma_2}(x)$$

Which means we can write a function that accepts two shifts, applies them to the "column gaussians" and returns the overlap matrix $\mathrm{overlap} : a, b \to S$. Then, the kinetic matrix is computed as

$$T = \mathrm{div}_b\mathrm{grad}_a\mathrm{overlap}(a,b)|_{a=b=0}.$$

5 Nuclear

Introducing the nucleus position vector $\vec{N}$ and $\epsilon = \frac{\gamma}{4}$, we write

$$V[12] = o \cdot \sum_{i=0}^{[l_{1,1}+l_{2,1}]} \sum_{j=0}^{[l_{1,2}+l_{2,2}]} \sum_{k=0}^{[l_{1,3}+l_{2,3}]} A_{1,i}A_{2,i}A_{3,i}F_{i+j+k} = o \cdot \sum_I F_I \sum_{i+j+k=I} A_{1,i}A_{2,i}A_{3,i} \tag{10}$$

$$= o \cdot \sum_I F_I \mathrm{Conv}_3[A_1, A_2, A_3]_I \tag{11}$$

where

$$F_I = \mathrm{boys}(I, \vec{d}^2_{PN} \cdot \gamma) \tag{12}$$

$$o = 2\pi/\gamma e^{-\alpha_1\alpha_2\vec{d}^2_{AB}/\gamma} \tag{13}$$

The quantity $A_{c,I}$ is defined as

$$A_{c,I} = \sum_{i-2r-u=I, r\leq[i/2], u\leq[i/2]-r} a_{c,i} b_r c_{c,I,u} \tag{14}$$

where

$$\begin{aligned} a_{c,i} &= i!(-1)^i f_i(l_{1,c}, l_{2,c}, \vec{d}_{AP,c}, \vec{d}_{BP,c}) \\ b_r(\epsilon) &= \frac{\epsilon^r}{r!} \\ c_{c,I,u}(p) &= \frac{(\vec{d}_{NP,c})^{I-u}}{(I-u)!} \cdot \frac{(-1)^u \epsilon^u}{u!} \end{aligned}$$

By defining $v = \text{inflate}(b)$, the sum over $2r$ can be transformed into a sum over a new contiguous index j, such that the ab term becomes

$$e_L = \sum_{i-j=L, j\leq i} a_i v_j = \sum_{i, L\geq 0} a_i v_{i-L} = v_{L,i} a_i \tag{15}$$

Similarly, defining

$$d'_{c,I,L} = \frac{(\vec{d}_{NP,c})^{2I-L} (-1)^{L-I} \epsilon^{L-I}}{(2I-L)!(L-I)!} \tag{16}$$

where we allow non-zero entries only for $[L/2] \geq L - I \geq 0$, we finally obtain

$$A_{c,I} = \sum_{L-u=I, u\leq[L/2]} e_L d_{c,L,u} = \sum_L d'_{c,I,L} e_L \tag{17}$$

In conclusion, a nuclear matrix element decomposes into a triple convolution of matrix-vector-products.

6 Interaction

Similar to the nuclear matrix elements, we write

$$U[1234] = \sum_I F_I \text{Conv}_3[A_1, A_2, A_3]_I \tag{18}$$

where

$$F_I = \text{boys}(I, \vec{d}^2_{PQ}/(\gamma_1+\gamma_2)) \tag{19}$$
$$\tag{20}$$

The quantity $A_{c,I}$ is

$$A_{c,I} = \sum_{r_1\leq[i_1/2],r_2\leq[i_2/2],u\leq[(i_1+i_2)/2]-r_1-r_2} a_{c,i_1,r_1} b_{c,i_2,r_2} d_{c,I+u,u}, \tag{21}$$

where

$$I = i_1 + i_2 - 2(r_1+r_2) - u \tag{22}$$
$$a_{c,i_1,r_1} = \frac{f_{i_1}(l_{1,c}, l_{2,c}, \vec{d}_{AP,c}, \vec{d}_{BP,c}) i_1!}{r_1!(i_1-2r_1)!(4\gamma_1)^{i_1-r_1}} \tag{23}$$
$$b_{c,i_2,r_2} = \frac{(-)^{i_2} f_{i_2}(l_{3,c}, l_{4,c}, \vec{d}_{CQ,c}, \vec{d}_{DQ,c}) i_2!}{r_2!(i_2-2r_2)!(4\gamma_2)^{i_2-r_2}} \tag{24}$$
$$d_{c,I+u,u} = \frac{(I+u)!(-)^u(\vec{d}_{QP})^{I-u}}{u!(I-u)!\delta^I} \tag{25}$$
$$\delta = \frac{1}{4\gamma_1} + \frac{1}{4\gamma_2} \tag{26}$$

We now rewrite

$$a_{c,L} = \frac{1}{L!} \sum_{r_1\leq[i_1/2]}^{i_1-2r_1=L} \frac{f_{i_1}(l_{1,c}, l_{2,c}, \vec{d}_{AP,c}, \vec{d}_{BP,c}) i_1!}{(4\gamma_1)^{i_1}} \frac{(4\gamma_1)^{r_1}}{r_1!} \tag{27}$$
$$b_{c,L} = \frac{1}{M!} \sum_{r_2\leq[i_2/2]}^{i_2-2r_2=M} (-)^{i_2} \frac{f_{i_2}(l_{1,c}, l_{2,c}, \vec{d}_{CQ,c}, \vec{d}_{DQ,c}) i_2!}{(4\gamma_2)^{i_2}} \frac{(4\gamma_2)^{r_2}}{r_2!} \tag{28}$$

Due to the sums in the first and second line being restricted, they can not be directly translated to cross-correlations.

Instead, one can rewrite a by defining

$$v_{2r_1} = \frac{1}{r_1!(4\gamma_1)^{r_1}} \tag{29}$$

$$v_{2r_1+1} = 0 \tag{30}$$

$$w_{c,i} = f_i(l_{1,c}, l_{2,c}, \vec{d}_{AP,c}, \vec{d}_{BP,c}) i!(4\gamma_1)^i \tag{31}$$

This results in

$$a_{c,L} L! = \sum_{j \leq i}^{i-j=L} w_{c,i} v_j = \sum_{i,L \geq 0} w_{c,i} v_{i-L} \equiv \sum_i v_{L,i} w_{c,i} \tag{32}$$

by promoting v to a matrix such that $v_{L,i} = v_{i-L}$. The rewriting for b proceeds analogously. Then, defining

$$c_{c,K} = \sum_{L+M=K} a_{c,L} b_{c,M} = \mathrm{Conv}[a_c, b_c]_K \tag{33}$$

we can write

$$A_I = \sum_{L+M-u=I} a_L b_M d_{I+u,u} = \sum_{u \leq K}^{K-u=I} c_K d_{K,u} = \sum_{I \geq 0}^{K} c_K d_{K,K-I} \equiv \sum_K e_{I,K} c_K$$

(34)

where

$$e_{I,K} = \frac{K!(-)^{K-I} p_x^{2I-K}}{(K-I)!(2I-K)!\delta^I} \tag{35}$$

is allowed non-zero entries only for $[K/2] \geq K - I \geq 0$.

In conclusion, an interaction matrix element decomposes into a convolution of matrix-vector-products, where the vectros themselves are obtained from convolutions.

7 Just-in-time compilation and performance

To be compatible with JAX JIT, the functions discussed here are compiled with intermediate tensors of static shape, set by the largest possible angular momentum in the basis set. The definition of the matrices d, e in the nuclear, interaction matrix element computations are crucial for setting the correct summation limits.

Performance on GPU depends (probably, mostly) on XLA's ability to fuse operators, because GPUs can do a lot of stuff in parallel, but memory access / individual ops aren't that fast.