A different calculation scheme for probability-dependent interaction calculation

James Dama

January 5, 2016

Assume that one has a potential that can be written in the following form:

$$U = U(\{\mathbf{r}_i\}) = U(\{p_{i,\alpha}(\{\mathbf{r}_j\})\}, \{\mathbf{r}_i\}) \tag{1}$$

where $\{p_i\}$ is a set of per-particle probabilities of occupying a given state. In our case, the probabilities are a function of the local neighborhood and can be calculated using a normal pair neighbor list.

The set of forces on all particles due to this potential is found from calculations of the derivatives

$$\nabla U = \nabla U(\{\mathbf{r}_i\}) = \nabla U(\{p_{i,\alpha}(\{\mathbf{r}_j\})\}, \{\mathbf{r}_i\}) \tag{2}$$
$$\tag{3}$$

which can be rewritten using the chain rule as

$$\nabla U(\{p_{i,\alpha}(\{\mathbf{r}_j\})\}, \{\mathbf{r}_i\}) = \nabla U(\{p_{i,\alpha}(\{\mathbf{r}_j\})\}, \{\mathbf{r}_i\})|_{\{p_{i,\alpha}(\{\mathbf{r}_j\})\}} + \sum_{i,\alpha} \frac{dU(\{p_{i,\alpha}(\{\mathbf{r}_j\})\}, \{\mathbf{r}_i\})}{dp_{i,\alpha}}|_{\{\mathbf{r}_j\}} \nabla p_{i,\alpha}(\{\mathbf{r}_j\}) \tag{4}$$

There are two ways to consider evaluating this. One is to carry through the derivatives, group terms, and then look for algebraic structure in the result that can simplify the calculation. For instance, when $p_i(\{\mathbf{r}_j\})$ is a function of a local density measured as a sum of pairwise proximity functions with finite range,

$$p_i(\{\mathbf{r}_j\}) = p_i(\rho_i(\{\mathbf{r}_j\})) \tag{5}$$
$$\rho_i(\{\mathbf{r}_j\}) = \sum_{j \neq i} w(r_{ij}) \tag{6}$$
$$\tag{7}$$

then the sum is the same whether it's a sum over all particles or just any set of neighbors that contains all particles within the cutoff of $w(r)$

$$\sum_{j\neq i} w(r_{ij}) = \sum_{j \text{ neigh } i} w(r_{ij}) \tag{8}$$

and the derivative of p_i with respect to the position of any particle k is therefore zero unless k is a neighbor of i. Therefore, with a one-body potential plus finite range pair potential in each state, Eq. 4 above can be rewritten

$$\nabla U(\{p_{i,\alpha}(\{\mathbf{r}_j\})\},\{\mathbf{r}_i\}) = \nabla U(\{p_{i,\alpha}(\{\mathbf{r}_j\})\},\{\mathbf{r}_i\})|_{\{p_{i,\alpha}(\{\mathbf{r}_j\})\}} + \sum_{i,\alpha} \frac{dU(\{p_{i,\alpha}(\{\mathbf{r}_j\})\},\{\mathbf{r}_i\})}{dp_{i,\alpha}}|_{\{\mathbf{r}_j\}} \nabla p_{i,\alpha}(\{\mathbf{r}_j\}) \tag{9}$$

$$= \frac{1}{2} \sum_{i,j \text{ neigh } i,\alpha,\beta} p_{i,\alpha}p_{j,\beta}\nabla u_{\alpha\beta}(r_{ij}) + \sum_{i,\alpha} \frac{dU(\{p_{i,\alpha}(\{\mathbf{r}_j\})\},\{\mathbf{r}_i\})}{dp_{i,\alpha}}|_{\{\mathbf{r}_j\}} \nabla p_{i,\alpha}(\{\mathbf{r}_j\}) \tag{10}$$

$$= \frac{1}{2} \sum_{i,j \text{ neigh } i,\alpha,\beta} p_{i,\alpha}p_{j,\beta}\nabla u_{\alpha\beta}(r_{ij}) + \sum_{i,\alpha} \frac{du_\alpha(p_{i,\alpha})}{dp_{i,\alpha}} \nabla p_{i,\alpha}(\{\mathbf{r}_j\}) + \dots$$

$$\dots + \sum_{i,\alpha} \frac{dU_{\text{pair}}(\{p_{i,\alpha}(\{\mathbf{r}_j\})\},\{\mathbf{r}_i\})}{dp_{i,\alpha}}|_{\{\mathbf{r}_j\}} \nabla p_{i,\alpha}(\{\mathbf{r}_j\}) \tag{11}$$

$$= \frac{1}{2} \sum_{i,j \text{ neigh } i,\alpha,\beta} p_{i,\alpha}p_{j,\beta}\nabla u_{\alpha\beta}(r_{ij}) + \sum_{i,\alpha} \frac{du_\alpha(p_{i,\alpha})}{dp_{i,\alpha}} \nabla p_{i,\alpha}(\{\mathbf{r}_j\}) + \dots$$

$$\dots + \sum_{i,\alpha} \frac{d\frac{1}{2}\sum_{j,k \text{ neigh } j,\beta,\gamma} p_{j,\beta}p_{k,\gamma}u_{\beta\gamma}(r_{jk})}{dp_{i,\alpha}}|_{\{\mathbf{r}_l\}} \nabla p_{i,\alpha}(\{\mathbf{r}_l\}) \tag{12}$$

$$= \frac{1}{2} \sum_{i,j \text{ neigh } i,\alpha,\beta} p_{i,\alpha}p_{j,\beta}\nabla u_{\alpha\beta}(r_{ij}) + \sum_{i,\alpha} \frac{du_\alpha(p_{i,\alpha})}{dp_{i,\alpha}} \nabla p_{i,\alpha}(\{\mathbf{r}_j\}) + \dots$$

$$\dots + \sum_{i,j \text{ neigh } i,\alpha,\beta} p_{j,\beta}u_{\alpha\beta}(r_{ij}) \nabla p_{i,\alpha}(\{\mathbf{r}_l\}) \tag{13}$$

and, recognizing that $\nabla p_{i,\alpha}(\{\mathbf{r}_l\})$ is another pair list sum, we see that the first two terms are simple pair computations but the last requires a twice nested pair list sum. In this equation, the first term is Jaehyeok's 4-2, the second is his 4-1, and the third includes both 4-3 and 4-4. Furthermore, since computing the probabilities is itself a neighbor calculation, either there must be an initial pair list traversal to calculate each of these, or to calculate all of the terms above one must invoke a local neighborhood computation for every term in every one of the pair lists, effectively nesting the pair list traversal one level deeper.

However, there is a simpler, more elegant way to do this based on first computing the $p_{i,\alpha}(\{\mathbf{r}_j\})$, then the $\nabla U(\{p_{i,\alpha}(\{\mathbf{r}_j\})\}, \{\mathbf{r}_i\})|_{\{p_{i,\alpha}(\{\mathbf{r}_j\})\}}$ and $\frac{dU(\{p_{i,\alpha}(\{\mathbf{r}_j\})\}, \{\mathbf{r}_i\})}{dp_{i,\alpha}}|_{\{\mathbf{r}_j\}}$, and then the $\nabla p_{i,\alpha}(\{\mathbf{r}_l\})$. In this case, that amounts to three pair list traversals in sequence. There is no need for nested loops or for modifying the definition of the neighborlist to include neighbors once-removed, as in Jaehyeok's current code.

This is more elegant to me in even just this one simple case. However, the power of this approach becomes much more apparent when considering more general probabilities and changes in the probabilities. This scheme separates the state-specific potential computations from the state probability calculations entirely, so that both may be programmed independently. This enables much more flexible work, and much greater code re-use. One could concevably use identical state probability implementations across pair and pair & few-body-topology force fields, and each state-dependent potential class like pairs or pairs & angles would only need to be programmed once in order to use any state probability definition or implementation.