$$
\begin{aligned}
&\simeq \Omega(X)\{C_{ijkl}\eta_{kl} - U_{kk}\tau_{ij} + U_{ik}\tau_{kj} + \tau_{ik}U_{jk}\} \\
&\simeq \Omega(X)\{\frac{1}{2}C_{ijkl}(U_{kl} + U_{lk}) - U_{kk}\tau_{ij} + U_{ik}\tau_{kj} + \tau_{ik}U_{jk}\} \\
&= \ \Omega(X)\{C_{ijkl}U_{kl} - U_{kk}\tau_{ij} + U_{ik}\tau_{kj} + \tau_{ik}U_{jk}\} \qquad (2.269)
\end{aligned}
$$

So

$$g^*_{ij} \ = \ \Omega(X)B^*_{ijkl}U_{kl} + ... \qquad (2.270)$$

with

$$B^*_{ijkl} \ = \ C_{ijkl} - \tau_{ij}\delta_{kl} + \tau_{jl}\delta_{ik} + \tau_{il}\delta_{jk} \qquad (2.271)$$

Similar to the arguments in symmetric deformation space, stability behavior in general deformation space is governed by

$$A^* \ = \ (B^* + (B^*)^T)/2 \qquad (2.272)$$

When

$$\det|A^*| \ = \ 0 \qquad (2.273)$$

for the first time, the system become unstable. Note that A^* is a 9×9 matrix.

As an observation, when a configuration is stable in general deformation space, i.e., none of the eigenvalues of A^* are negative, the configuration would be stable in all deformation spaces.

2.4.2 Deformation of a Periodic Simulation Cell

Basic setup

We will simulate particle systems that interact via short-ranged potentials of cutoff radius r_{cut}. Given any radial function $w(r)$, this can be done by redefining a $\tilde{w}(r)$,

$$\tilde{w}(r) \equiv \begin{cases} w(r) - w(r_{\text{cut}}) - w'(r_{\text{cut}})(r - r_{\text{cut}}), & 0 < r < r_{\text{cut}} \\ 0, & r \geq r_{\text{cut}} \end{cases}, \qquad (2.274)$$

which[2] ensures continuity in both energy and forces as a particle crosses $r = r_{\text{cut}}$. In the case of multiple-component systems, r_{cut} is generalized to a matrix $r_{\text{cut}}^{\alpha\beta}$, where $\alpha \equiv c(i), \beta \equiv c(j)$ are the chemical types of atoms i,j. As a convention, i suggests a "host" atom at the "frame origin", whereas j suggests a "client" atom. We then define

$$\mathbf{x}_{ij} \equiv \mathbf{x}_j - \mathbf{x}_i, \quad r_{ij} \equiv |\mathbf{x}_{ij}|, \quad \hat{\mathbf{x}}_{ij} \equiv \frac{\mathbf{x}_{ij}}{r_{ij}}. \tag{2.275}$$

Quantities such as the pair force $\mathbf{f}_{ij} \equiv \left(-\tilde{V}'(r_{ij})\right)\hat{\mathbf{x}}_{ij}$ are understood as the force on j *due to* i. $\mathbf{f}_{ij}$ should be *parallel* to $\mathbf{x}_{ij}$ when the potential is *repulsive*, a mnemonic device.

The supercell is a parallelepiped, which can be tiled in space indefinitely if desired. The three edges are row vectors

$$\mathbf{a}_1 = (H_{11}, H_{12}, H_{13}), \quad \mathbf{a}_2 = (H_{21}, H_{22}, H_{23}), \quad \mathbf{a}_3 = (H_{31}, H_{32}, H_{33}), \tag{2.276}$$

in Cartesian coordinates, with $H_{\mu\nu}$ forming a 3×3 matrix $\mathbf{H}$. [3] The position of particle i is specified by a row vector, $\mathbf{s}_i = (s_{i1}, s_{i2}, s_{i3})$, with $s_{i\mu}$'s usually satisfying

$$0 \leq s_{i\mu} < 1, \quad \mu = 1..3, \tag{2.277}$$

and the Cartesian coordinate of this particle, $\mathbf{x}_i$, also a row vector, is

$$\mathbf{x}_i \;=\; s_{i1}\mathbf{a}_1 + s_{i2}\mathbf{a}_2 + s_{i3}\mathbf{a}_3 \;=\; \mathbf{s}_i\mathbf{H}, \tag{2.278}$$

where $s_{i\mu}$ has the geometrical interpretation of the fraction of the μth edge to build

[2] An alternative is to define $\tilde{w}(r) \equiv w(r)\exp(r_s/(r - r_{\text{cut}}))$ which has all derivatives continuous at $r = r_{\text{cut}}$. Another efficient scheme for the LJ6-12 potential is $w(r) \equiv 4\varepsilon[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} + \left(2\left(\frac{\sigma}{r_{\text{cut}}}\right)^{18} - \left(\frac{\sigma}{r_{\text{cut}}}\right)^{12}\right)\left(\frac{r}{\sigma}\right)^{6} - 3\left(\frac{\sigma}{r_{\text{cut}}}\right)^{12} + 2\left(\frac{\sigma}{r_{\text{cut}}}\right)^{6}]$, which expands in r^6 instead of r, and avoids using `sqrt` or `exp`.

[3] This labelling scheme is literally followed in both my C ($1-3$ becomes $0-2$) and Fortran source codes, irrespective of *internal storage arrangements*. That is, I may sacrifice efficiency for clarity in Fortran (where *columns* are stored contiguously), to achieve easy-to-read correspondence with C source codes, where this arrangement *is* computationally more efficient for edge vector operations.

$\mathbf{x}_i$.

The volume of the supercell is

$$\Omega = |\det H| = |\mathbf{a_1} \cdot (\mathbf{a_2} \times \mathbf{a_3})| > 0. \tag{2.279}$$

The inverse of the $\mathbf{H}$ matrix $\mathbf{B} \equiv \mathbf{H}^{-1}$ satisfies

$$\mathbf{I} = \mathbf{HB} = \mathbf{BH}. \tag{2.280}$$

If we define row vectors

$$\mathbf{b}_1 \equiv (B_{11}, B_{21}, B_{31}), \quad \mathbf{b}_2 \equiv (B_{12}, B_{22}, B_{32}), \quad \mathbf{b}_3 \equiv (B_{13}, B_{23}, B_{33}), \tag{2.281}$$

then (2.280) is equivalent to

$$\mathbf{a}_i \cdot \mathbf{b}_j \equiv \mathbf{a}_i \mathbf{b}_j^T = \delta_{ij}. \tag{2.282}$$

Since $\mathbf{b}_1$ is perpendicular to both $\mathbf{a}_2$ and $\mathbf{a}_3$, it must be collinear with the normal direction $\mathbf{n}$ of the $\mathbf{a}_2/\mathbf{a}_3$ plane: $\mathbf{b}_1 \equiv |\mathbf{b}_1|\mathbf{n}$. And so by (2.282),

$$1 = \mathbf{a}_1 \cdot \mathbf{b}_1 = \mathbf{a}_1 \cdot (|\mathbf{b}_1|\mathbf{n}) = |\mathbf{b}_1|(\mathbf{a}_1 \cdot \mathbf{n}). \tag{2.283}$$

But $|\mathbf{a}_1 \cdot \mathbf{n}|$ is nothing other than the *thickness* of the supercell along the $\mathbf{a}_1$ edge. Therefore, the thicknesses (distances between two parallel surfaces) of the supercell are,

$$d_1 = \frac{1}{|\mathbf{b}_1|}, \quad d_2 = \frac{1}{|\mathbf{b}_2|}, \quad d_3 = \frac{1}{|\mathbf{b}_3|}. \tag{2.284}$$

The general design of the simulation should allow for deformation of $\mathbf{H}$ that includes rotational components, even though one may choose to impose the constraint of symmetric deformation later, whose dynamics is derived in section 2.4.2. In general one should use the Lagrangian strain $\underline{\underline{\eta}}$, a true rank-2 tensor under coordinate transformation, to measure the deformation of a MD supercell as it is unlikely to be infinitesimal. To define $\underline{\underline{\eta}}$, one needs a reference $\mathbf{H}_0$ of a previous time, with $\mathbf{x}_0 = \mathbf{s}\mathbf{H}_0$ and $d\mathbf{x}_0 = (d\mathbf{s})\mathbf{H}_0$, and then imagine that with $\mathbf{s}$ and $d\mathbf{s}$ fixed, $d\mathbf{x}_0$ is transformed to $d\mathbf{x} = (d\mathbf{s})\mathbf{H}$, under $\mathbf{H}_0 \to \mathbf{H} \equiv \mathbf{H}_0\mathbf{K}$.

The Lagrangian strain is defined by the change in the differential line length,

$$dl^2 = d\mathbf{x}d\mathbf{x}^T \equiv d\mathbf{x}_0(\mathbf{I} + 2\underline{\underline{\eta}})d\mathbf{x}_0^T, \tag{2.285}$$

where by plugging in $d\mathbf{x} = (d\mathbf{s})\mathbf{H} = (d\mathbf{x}_0)\mathbf{H}_0^{-1}\mathbf{H} = (d\mathbf{x}_0)\mathbf{K}$, $\underline{\underline{\eta}}$ is seen to be

$$\underline{\underline{\eta}} = \frac{1}{2}\left(\mathbf{H}_0^{-1}\mathbf{H}\mathbf{H}^T\mathbf{H}_0^{-T} - \mathbf{I}\right) = \frac{1}{2}\left(\mathbf{K}\mathbf{K}^T - \mathbf{I}\right). \tag{2.286}$$

Because $\underline{\underline{\eta}}$ is a symmetric matrix, it always has three mutually orthogonal eigendirections $\mathbf{x}_1\underline{\underline{\eta}} = \mathbf{x}_1\eta_1$, $\mathbf{x}_2\underline{\underline{\eta}} = \mathbf{x}_2\eta_2$, $\mathbf{x}_3\underline{\underline{\eta}} = \mathbf{x}_3\eta_3$. Along those directions, the line lengths are changed by factors $\sqrt{1+2\eta_1}$, $\sqrt{1+2\eta_2}$, $\sqrt{1+2\eta_3}$, which achieve extrema among *all* line directions. Thus, as long as η_1, η_2 and η_3 oscillate between $[-\eta_{\text{bound}}, \eta_{\text{bound}}]$ for some chosen η_{bound}, any line segment at $\mathbf{H}_0$ can be lengthened by no more than $\sqrt{1+2\eta_{\text{bound}}}$ and shortened by no less than $\sqrt{1-2\eta_{\text{bound}}}$. That is, if we define length measure

$$L(\Delta\mathbf{s}, \mathbf{H}) \equiv \sqrt{\Delta\mathbf{s}\mathbf{H}\mathbf{H}^T\Delta\mathbf{s}^T}, \tag{2.287}$$

then so long as η_1, η_2, η_3 oscillate between $[\eta_{\min}, \eta_{\max}]$, there is

$$\sqrt{1+2\eta_{\min}}\, L(\Delta\mathbf{s}, \mathbf{H}_0) \quad \le \quad L(\Delta\mathbf{s}, \mathbf{H}) \quad \le \quad \sqrt{1+2\eta_{\max}}\, L(\Delta\mathbf{s}, \mathbf{H}_0). \tag{2.288}$$

One can use the above result to define a *strain session*, which begins with $\mathbf{H}_0 \equiv \mathbf{H}$ and during which no line segment is allowed to *shrink by less than a threshold* $f_c \le 1$*, compared to its length at* $\mathbf{H}_0$. This is equivalent to requiring that,

$$f \equiv \sqrt{1 + 2\left(\min(\eta_1, \eta_2, \eta_3)\right)} \quad \le \quad f_c. \tag{2.289}$$

Whenever this condition is violated, the session terminates and a new session starts with the current $\mathbf{H}$ as the new $\mathbf{H}_0$. In my implementation of the $\mathcal{O}(N)$ molecular dynamics program, this is associated with a repartitioning of the supercell into equal-size bins, and is called a strain-induced bin repartitioning.

The purpose of a bin partition and a strain session is the following: it can be a very demanding task to determine if atoms i,j are within r_{cut} or not, *for all possible ij combinations.* [4] Formally, this requires checking

$$r_{ij} \equiv L(\Delta\mathbf{s}_{ij}, \mathbf{H}) \quad \leq \quad r_{\text{cut}}. \tag{2.290}$$

Because $\mathbf{s}_i$, $\mathbf{s}_j$ and $\mathbf{H}$ are *mobile* – they differ from step to step, it appears that we have to do this at each step. This $\mathcal{O}(N^2)$ complexity would indeed be the case but for the observation that, in most MD, MC and static minimization applications, $\mathbf{s}_i$'s and $\mathbf{H}$ often change *only a little* from the previous step. Therefore, once we ensured that (2.290) hold at some previous step, we can devise a *sufficient condition* to test whether (2.290) still must hold *now* or not, at a much smaller cost. Only when this sufficient condition breaks down, which is taken to be less frequent, do we resort to a more complicated search and check in the fashion of (2.290).

My implementation of the above idea is as follows: I associate each $\mathbf{s}_i$ with a *semi-mobile* reduced coordinate $\mathbf{s}_i^a$ called atom i's *anchor*. At each step, I check if $L(\mathbf{s}_i - \mathbf{s}_i^a, \mathbf{H})$, that is, the current distance between i and its anchor, is greater than a certain $r_{\text{drift}} \geq r_{\text{drift}}^0$ or not. If it is not, then $\mathbf{s}_i^a$ does not change; if it is, then I redefine $\mathbf{s}_i^a \equiv \mathbf{s}_i$ at this step, which is called atom i's *flash* incident. At i's flash, atom

[4] It is often more efficient to count *pairs* if the potential function allows for easy use of such "half-lists", such as pair- or EAM potentials. In these scenarios we pick a unique "caretaker" among i and j to store the information about the ij-pair, that is, a particle's "personal list" only keeps possible pairs that are under its own care. For load-balancing it is best if the responsibilities are distributed evenly among particles. We use a pseudo-random choice of "*if $i+j$ is odd and $i>j$, or if $i+j$ is even and $i<j$, then i is the caretaker; otherwise it is j.*" As $i>j$ is "uncorrelated" with whether $i+j$ is even or odd, significant "load imbalance" is unlikely to occur even if the indices correlate strongly with the atoms' positions.

i is required to update records [5] of all atoms whose anchors satisfy

$$L(\mathbf{s}_j^a - \mathbf{s}_i^a, \mathbf{H}_0) \;\; \leq \;\; r_{\text{list}} \equiv \frac{r_{\text{cut}} + 2r_{\text{drift}}^0}{f_c}. \tag{2.291}$$

Note that the distance is between *anchors* instead of *atoms* [6], and the length is measured by $\mathbf{H}_0$, not the current $\mathbf{H}$. (2.291) nominally takes $\mathcal{O}(N)$ work per flash, but I accelerate it to $\mathcal{O}(1)$ per flash by partitioning the supercell into $m_1 \times m_2 \times m_3$ bins at the start of the session, whose thicknesses by $\mathbf{H}_0$ (see (2.284)) are required to be greater than or equal to r_{list}:

$$\frac{d_1(\mathbf{H}_0)}{m_1}, \frac{d_2(\mathbf{H}_0)}{m_2}, \frac{d_3(\mathbf{H}_0)}{m_3} \;\; \geq \;\; r_{\text{list}}. \tag{2.292}$$

The bins deform with $\mathbf{H}$ and remains commensurate with it, that is, its $\mathbf{s}$-partition $1/m_1$, $1/m_2$, $1/m_3$ remains fixed during a strain session. Each bin keeps an updated list[7] of all *anchors* inside. Then, if at the time of i's flash two anchors are separated by more than one bin, there would be

$$L(\mathbf{s}_j^a - \mathbf{s}_i^a, \mathbf{H}_0) \;\; > \;\; \frac{d_1(\mathbf{H}_0)}{m_1}, \frac{d_2(\mathbf{H}_0)}{m_2}, \frac{d_3(\mathbf{H}_0)}{m_3} \;\; \geq \;\; r_{\text{list}}, \tag{2.293}$$

and they cannot possibly satisfy (2.291). Therefore we only need to test (2.291) for anchors within adjacent 27 bins. To synchronize, all atoms flash at the start of a strain session. From then on, atoms flash individually whenever $L(\mathbf{s}_i - \mathbf{s}_i^a, \mathbf{H}) > r_{\text{drift}}$.[8] We see that to maintain anchor lists that captures all solutions to (2.291) among the latest anchors, it takes only $\mathcal{O}(N)$ work per step, and the pre-factor of which is also small because flash events happen quite infrequently for a tolerably large r_{drift}^0.

The central claim of the scheme is that if j is not in i's anchor records (suppose

[5] Parts of the records may be stored in j's if pairs are counted and j happens to be the caretaker of the ij pair.

[6] $\mathbf{s}_i^a = \mathbf{s}_i$, though.

[7] When atom i flashes, it also updates the bin-anchor list if necessary.

[8] If two anchors flash at the same step in a loop, the first flash may get it wrong – that is, missing the second anchor, but the second flash will correct the mistake. The important thing here is to *not* lose an interaction.

i's last flash is more recent than j's), which was created *some time ago* in the strain session, then $r_{ij} > r_{\rm cut}$. The reason is that the *current* separation between the anchor i and anchor j, $L(\mathbf{s}_j^a - \mathbf{s}_i^a, \mathbf{H})$, is greater than $r_{\rm cut} + 2r_{\rm drift}^0$, since by (2.288), (2.289) and (2.291),

$$L(\mathbf{s}_j^a - \mathbf{s}_i^a, \mathbf{H}) \;\geq\; f \cdot L(\mathbf{s}_j^a - \mathbf{s}_i^a, \mathbf{H}_0) \;>\; f \cdot r_{\rm list} \geq f_c \cdot r_{\rm list} = f_c \cdot \frac{r_{\rm cut} + 2r_{\rm drift}^0}{f_c}. \quad (2.294)$$

In fact, we see that the $r_{ij} > r_{\rm cut}$ conclusion maintains if neither i or j currently drifts more than

$$r_{\rm drift} \;\equiv\; \frac{f \cdot r_{\rm list} - r_{\rm cut}}{2} \;\geq\; r_{\rm drift}^0, \quad (2.295)$$

from respective anchors. Put it another way, when we design $r_{\rm list}$ in (2.291), we take into consideration both atom drifts and $\mathbf{H}$ shrinkage which both may bring ij closer than $r_{\rm cut}$, but since the current $\mathbf{H}$ shrinkage has not yet reached the designed critical value, we can convert it to more leeway for the atom drifts.

For multi-component systems, we define

$$r_{\rm list}^{\alpha\beta} \;\equiv\; \frac{r_{\rm cut}^{\alpha\beta} + 2r_{\rm drift}^0}{f_c}, \quad (2.296)$$

where both f_c and $r_{\rm drift}^0$ are species-independent constants, and $r_{\rm drift}^0$ can be thought of as putting a lower bound on $r_{\rm drift}$, so flash events cannot occur too frequently – a self-protection mechanism. At each bin repartitioning, we would require

$$\frac{d_1(\mathbf{H}_0)}{m_1}, \frac{d_2(\mathbf{H}_0)}{m_2}, \frac{d_3(\mathbf{H}_0)}{m_3} \;\geq\; \max_{\alpha,\beta} r_{\rm list}^{\alpha\beta}. \quad (2.297)$$

And during the strain session, $f \geq f_c$, we have

$$r_{\rm drift}^{\alpha} \;\equiv\; \min\left[\min_{\beta}\left(\frac{f \cdot r_{\rm list}^{\alpha\beta} - r_{\rm cut}^{\alpha\beta}}{2}\right), \min_{\beta}\left(\frac{f \cdot r_{\rm list}^{\beta\alpha} - r_{\rm cut}^{\beta\alpha}}{2}\right)\right], \quad (2.298)$$

a time- and species-dependent atom drift bound that controls whether to flash.