Challenge 2: Packing Tetrahedra

March 12, 2015

The task of identifying dense crystal structures can be thought of in (at least) two ways: as a global optimization problem over particle/cell degrees of freedom or the final state of a compression process. It isn't immediately clear at the outset which of these two approaches would most reliably return the densest packing of regular solids. On the one hand, there are excellent global optimization procedures that can digest high dimensional problems efficiently while preserving tolerance to rough fitness landscapes (Particle Swarm Optimization and Genetic Algorithms are the primary players in this space). However, these methods are heuristic and when they fail it's not often clear how to proceed. In addition, each method exhibits common failure mechanisms related to population collapse and premature convergence, which can be difficult to address in a systematic way. Unfortunately, gradient-based methods are straight out since we're lacking anything resembling a smooth potential between particles and the optimization landscape is dense in low density, jammed states. In contrast, the physical interpretation of optimizing through a compression process is straight-forward and robust (assuming infinite computation time). By including some annealing procedure (simulated annealing or parallel tempering, the latter being used here), locally jammed states can be relieved through rare expansion moves as the system attempts to converge to a dense packing in the high pressure limit. While the present solution uses a constant pressure (isobaric-isothermal) Monte Carlo method with parallel tempering to converge dense solid packings, it would be really interesting to throw a PSO solution together that builds off of it to see how the two methods compare!

As explained above, the method used here is a more or less off-the-shelf (although using a custom implementation) Monte Carlo simulation sampling states at constant NPT. The system is defined by a cell tensor $\underline{\underline{h}}$ and set of regular tetrahedra represented by their four vertices $[\underline{r}_0, \underline{r}_1, \underline{r}_2, \underline{r}_3]$. At each step, a cell shape move is performed with probability p_{cell} by generating a symmetric, random strain tensor $\underline{\underline{\epsilon}}$ and updating the cell according to:

$$\underline{\underline{h}}^{t+1} = \left(\underline{\underline{I}} + \underline{\underline{e}}\right) \cdot \underline{\underline{h}}^t$$

During a cell shape update, the particle positions are moved by simply maintaining the fractional coordinates of their center of mass across the transformation i.e.

$$\underline{s} = \left(\underline{\underline{h}}^t\right)^{-1} \cdot \underline{r}_{com}^t$$

$$\underline{r}_{com}^{t+1} = \underline{\underline{h}}^{t+1} \cdot \underline{s}$$

If a collision is detected, this process is simply reversed to undo the move. In order to sample states at equilibrium with an externally imposed pressure, cell shape moves are accepted according to the appropriate Metropolis criterion:

$$p_{acc} = \min\left(1, e^{-\beta P\left(V^{t+1} - V^t\right)}\right)$$

Similarly, particles are perturbed with either an intrinsic rotation or center-of-mass translation. A particle move is rejected if the move results in a collision with another particle and accepted otherwise.

While the above algorithm is sound, quenching a system to high pressure and hoping to converge rapidly onto a valid packing structure while avoiding jammed states is a losing proposition. The strategy used here to anneal out of jammed states at high pressures is to include parallel tempering steps, periodically swapping the pressures of two simulations running in parallel. This trick is used extensively in the particle-based simulation community to help the system sample outside of deep, narrow minima. The analogy with standard parallel tempering to pressure moves is straightforward; we propose swapping the states of two systems at different pressures P_0 and P_1, accepting the move with probability:

$$p_{acc} = \min\left(1, \frac{e^{-\beta(P_0 V_1 + P_1 V_0)}}{e^{-\beta(P_1 V_1 + P_0 V_0)}}\right)$$

Results

Briefly, the method appears to work quite well although the absolute densest known packings were unfortunately not able to be reproduced.

The best solution observed had a packing fraction of 82.5% and is reproduced below.
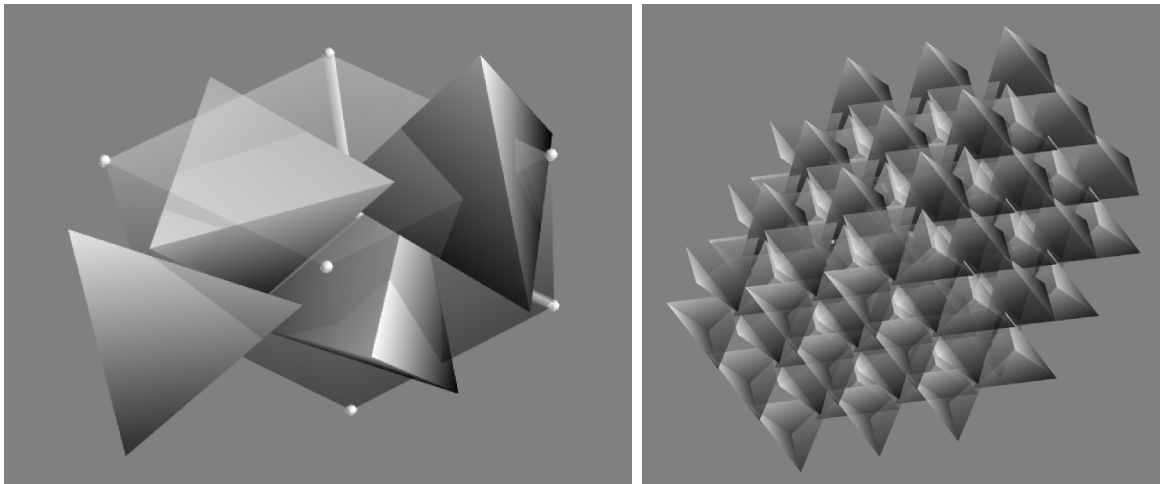

Figure 1: Best observed packed structure run with: n_particles=4, n_drivers=5, p=[25, 100, 250, 500, 1000], n_steps=10^6.