Immobile atoms: Fixing the Virial in Amber11

Andrew Jewett (jewett.ai gmail), Shea Lab
University of California Santa Barbara
Santa Barbara, CA 93106

March 7, 2011

Draft version

1 Introduction:

Pressure regulation and atom position restraints

Members of our group have observed that AMBER has difficulty running MD simulations at constant pressure whenever a large number of atoms are restrained. This is one of the obstacles preventing us from using AMBER in simulations involving large (immobile) assemblies like infinite surfaces or polymers.

The purpose of this document is to 1) call attention to problems with pressure regulation in AMBER 2) point out a simple solution to fix the problem 3) propose code that implements this solution 4) ask for feedback. (If you don't think this approach will work, or if I screwed up, I am curious to hear about it and can be reached at jewett.ai on gmail. Note: I am not using AMBER much these days, but I am curious to read your comments.)

Currently, in AMBER, all pairwise forces between atoms contribute to the virial, regardless of whether they act on mobile or immobile atoms. Furthermore, all atomic positions are rescaled during pressure equilibration, including mobile atoms, immobile atoms (and restraint anchors). Although this is consistent with the way the virial is calculated, this behavior can be undesirable (see figure 1). We hope to convince the development team to make this behavior optional.

To compensate for this proposed change, we suggest that the forces acting on immobilized atoms should be omitted from the virial. . . . *unless* the forces that are used immobilize the atoms (and negate these pairwise forces) *are* included in the virial. (Although I could be wrong, my impression is that presently they are not.) Ideally, for efficiency, these pairwise forces acting on immobilized atoms should not even be calculated. (Although, to implement that modest performance enhancement at this late stage in the AMBER code might require rewriting a substantial amount of code. As an alternative, in appendix A, I included a small modification to the sander source code that we have been using to fix the virial in AMBER.)

This document ended up being a bit long. However the conclusion is simple: *Please include the restraint forces in the virial (and do not automat-*

ically rescale immobile atoms). This solution applies to systems with either harmonic or rigid (belly) restraints.

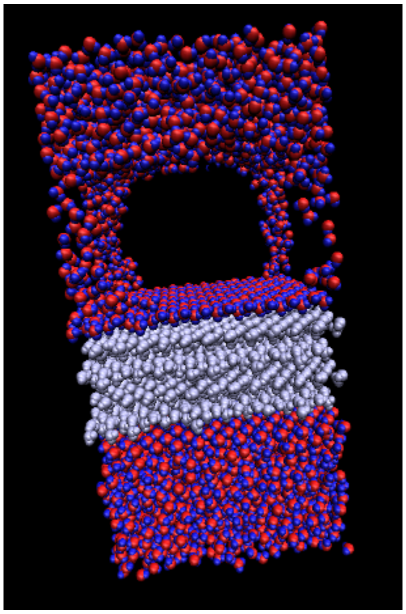

Figure 1: *Snapshot of a simulation containing restrained highly repulsive atoms.* Immobilizing restraints are particularly useful when applied to systems that would otherwise be unstable. As an extreme example, here we show a frame of animation from a simulation of (default TIP3P) water interacting with a slab of immobilized ice (middle) which has been held in place using harmonic restraints under NPT conditions ($T = 300°K$, $P = 1$ bar). The innermost layers of ice (white) have had their charges removed, allowing the remaining Lennard-Jones forces to dominate, causing strong intermolecular repulsion. Under NPT conditions, AMBER continues to expand the simulation box in a vain effort to relax the stress between repulsive (white) molecules, which are held in place by restraints. This cavitation problem goes away if either: 1) the charges in the middle layer are restored, or 2) harmonic restraints are removed allowing the neutral (white) molecules to expand to their equilibrium size.

2 The Virial in the presence of immobile atoms

The virial, W, is used to compute the instantaneous pressure of the system, and it is an ingredient in both of the barostats implemented in AMBER (either Berendsen, or Nose-Hoover, when PIMD is used). It is defined as: $W = -3V \frac{\mathrm{d}U}{\mathrm{d}V}$, where U is the potential energy *per* unit cell, and V is the volume of the unit cell. (There's a nice review of the virial that I found useful in chapter 5.3 of [1], and also [2].) To simplify this discussion, lets assume that the system has a cubic unit cell, and that scaling is isotropic ($V = L^3$, where L is the box size). I will assume that the forces contributing to the virial are dominated by pairwise interactions between atoms (denoted

$u_{\mu\nu}(|\vec{r}_\mu - \vec{r}_\nu|)$, where $\vec{r}_\mu$ and $\vec{r}_\nu$ are the positions of atoms μ and ν). The potential energy (per unit cell) is the sum of all pairwise energies between atoms, and their images in other cells, and can be written as:

$$\begin{aligned} U &= \frac{1}{2}\sum_{\vec{m}\in\mathbb{Z}^3}\sum_{\mu\neq\nu} u_{\mu\nu}\left(|\vec{r}_\mu - (\vec{r}_\nu + L\vec{m})|\right) && (1) \\ & \text{useful notation: } \vec{r}_{\nu,\vec{m}} \;\equiv\; \vec{r}_\nu + L\vec{m} && (2) \\ &= \frac{1}{2}\sum_{\vec{m}\in\mathbb{Z}^3}\sum_{\mu\neq\nu} u_{\mu\nu}\left(\left|\vec{r}_\mu - \vec{r}_{\nu,\vec{m}}\right|\right) && (3) \end{aligned}$$

where $\sum_{\mu\neq\nu}$ is a sum over all pairs of atoms in the system (mobile or immobile), and $\sum_{\vec{m}\in\mathbb{Z}^3}$ is a sum over an atom's images in all surrounding unit cells (located at position $\vec{r}_\nu + L\vec{m}$, where $\vec{m}$, is a vector with integer x,y,z components). The factor of $\frac{1}{2}$ compensates for redundancy of equivalent pairs ($\mu \leftrightarrow \nu$). This yields:

$$W \;=\; -3V\,\frac{\mathrm{d}U}{\mathrm{d}V} \;=\; -L\,\frac{\mathrm{d}U}{\mathrm{d}L} \qquad (\text{where } V = L^3) \qquad (4)$$

$$= -\frac{L}{2}\sum_{\vec{m}\in\mathbb{Z}^3}\left(\sum_{\mu\neq\nu}\frac{\mathrm{d}\vec{r}_\mu}{\mathrm{d}L}\cdot\underbrace{\frac{\partial u_{\mu\nu}\left(\left|\vec{r}_\mu - \vec{r}_{\nu,\vec{m}}\right|\right)}{\partial\vec{r}_\mu}}_{=\,-\vec{f}_{\nu,\vec{m}\to\mu}} + \frac{\mathrm{d}\vec{r}_{\nu,\vec{m}}}{\mathrm{d}L}\cdot\underbrace{\frac{\partial u_{\mu\nu}\left(\left|\vec{r}_\mu - \vec{r}_{\nu,\vec{m}}\right|\right)}{\partial\vec{r}_{\nu,\vec{m}}}}_{=\,-\vec{f}_{\mu\to\nu,\vec{m}}}\right) \quad (5)$$

One can recognize $-\frac{\partial}{\partial\vec{r}_\mu}u_{\mu\nu}\left(\left|\vec{r}_\mu - \vec{r}_{\nu,\vec{m}}\right|\right)$ as the force acting on atom μ exerted by atom ν's image in unit cell indicated by $\vec{m}$ (denoted $\vec{f}_{\nu,\vec{m}\to\mu}$). We now split this sum into terms involving mobile and immobile atoms. We use lower-case indices (i,j) to refer to mobile atoms, and upper-case indices (I,J) to refer to immobile atoms, respectively. Expressed this way, $W =$

$$\begin{aligned} &\frac{L}{2}\sum_{\vec{m}\in\mathbb{Z}^3}\Bigg(\sum_{i\neq j}\left(\frac{\mathrm{d}\vec{r}_i}{\mathrm{d}L}\vec{f}_{j,\vec{m}\to i} + \frac{\mathrm{d}\vec{r}_{j,\vec{m}}}{\mathrm{d}L}\;\vec{f}_{i\to j,\vec{m}}\right) + \sum_{I,j}\left(\frac{\mathrm{d}\vec{r}_I}{\mathrm{d}L}\vec{f}_{j,\vec{m}\to I} + \frac{\mathrm{d}\vec{r}_{j,\vec{m}}}{\mathrm{d}L}\;\vec{f}_{I\to j,\vec{m}}\right) \\ &\quad + \sum_{i,J}\left(\frac{\mathrm{d}\vec{r}_i}{\mathrm{d}L}\vec{f}_{J,\vec{m}\to i} + \frac{\mathrm{d}\vec{r}_{J,\vec{m}}}{\mathrm{d}L}\;\vec{f}_{i\to J,\vec{m}}\right) + \sum_{I\neq J}\left(\frac{\mathrm{d}\vec{r}_I}{\mathrm{d}L}\vec{f}_{J,\vec{m}\to I} + \frac{\mathrm{d}\vec{r}_{J,\vec{m}}}{\mathrm{d}L}\;\vec{f}_{I\to J,\vec{m}}\right)\Bigg) \quad (6) \end{aligned}$$

We use the notation: $\vec{s}_i \equiv \vec{r}_i/L$ to denote the normalized coordinates of atom i. The x,y,z components of $\vec{s}_i$ lie in the range from 0 and 1. Because mobile (unconstrained, lower-case) atom positions are rescaled during pressure equilibration, this means that $\vec{r}_i$ and $\vec{r}_j$ are proportional to L ($\vec{r}_j = L\vec{s}_i$), and consequently, $\vec{s}_i$ and $\vec{s}_j$ are independent of L. Consequently:

$$L\,\frac{\mathrm{d}\vec{r}_i}{\mathrm{d}L} \;=\; \vec{r}_i \quad \text{and} \quad L\,\frac{\mathrm{d}\vec{r}_{j.\vec{m}}}{\mathrm{d}L} \;=\; \vec{r}_{j,\vec{m}} \;=\; \vec{r}_j + L\vec{m} \qquad (7)$$

(See equation 2.) Let us assume, for the moment that immobilized atoms are *not* rescaled during pressure equilibration (see section 2.2). This means that:

$$\frac{\mathrm{d}\vec{r}_I}{\mathrm{d}L} \;=\; \frac{\mathrm{d}\vec{r}_J}{\mathrm{d}L} = 0 \quad \text{and thus} \quad \frac{\mathrm{d}\vec{r}_{I,\vec{m}}}{\mathrm{d}L} \;=\; \frac{\mathrm{d}\vec{r}_{J,\vec{m}}}{\mathrm{d}L} \;=\; \vec{m} \qquad (8)$$

After substituting equations 7 and 8 into 6, and some additional simplification (appendix B) we find that the following terms (denoted ΔW_{im}) are *left out* of the virial:

$$\Delta W_{\mathrm{im}} = \sum_{I} \vec{r}_I \cdot \sum_{\vec{m}\in\mathbb{Z}^3} \left(\sum_{j} \vec{f}_{j,\vec{m}\to I} + \sum_{J>I} \vec{f}_{J,\vec{m}\to I} \right) \tag{9}$$

These terms would otherwise be present if the immobilized atoms (with indices I, and J) *were* scaled with the box-size during pressure equilibration.

These terms (ΔW_{im}) should be subtracted from the virial as calculated by AMBER, which (I strongly suspect) includes all of these pairwise force terms. This correction is necessary since, for the virial, AMBER presently does not include contributions from the restraints used to hold the atoms in place. *Alternately*, one can apply the restraint forces which negate these *before* calculating the virial. (This is really the same thing, because these constraint forces would have contributed $-\Delta W_{\mathrm{im}}$ to the virial.) For completeness, an expression for the full virial is given in equation 10 in appendix B.

We can recognize the sum over $\sum_{\vec{m}\in\mathbb{Z}^3}$ as the *net pairwise force* acting on atom I. Hence we can calculate this correction to the virial knowing only the net force acting on each atom. (This assumes that this net force is dominated by pairwise forces, and does not include harmonic position restraints.) The code for calculating this correction (ΔW_{im}) is included in appendix A.

2.1 Harmonic restraints

The fixed proposed above should be adequate for harmonic restraints, even though, strictly speaking the restrained atoms are not truly immobilized. However in such cases, one can consider the pairwise interactions between *mobile atoms*, and invisible *immobile anchor-points*. When viewed this way, it's really the anchor points which play the role of the immobilized atoms in this case. So long as the force acting on these anchor points is not included in the virial (and the force of the anchor points on other atoms *is* included in the virial), then no correction to the virial is necessary. Hence, the solution for both harmonic restraints and rigid belly constraints is the same: *include restraint forces in the virial.*

2.2 Should immobilized atoms be rescaled?

Perhaps the choice whether to rescale immobilized atoms should be left up to the user. The current behavior of AMBER is not a bug. (The virial appears to be calculated in a way that is consistent with the way the atoms are rescaled during pressure equilibration.) However it can lead to counterintuitive results. Being able to fix the positions of atoms during NPT simulations is a feature that I'd like to see added to AMBER.

Appendices:

A Amending the AMBER virial code

The following function calculates ΔW_{im} and subtracts it from the virial (actually it calculates the x,y,z components of ΔW_{im}, and subtracts them from the x,y,z components of the virial). I have added the following function to the sander source code:

```
!subroutine correct_vir_immobile() modifies the virial
!due to immobilized atoms.  6 arguments are required:
!
!nat   is the number of atoms in the system
!vir() stores the x,y,z components of the virial
!f()   is the net force on each atom (excluding immobilizing restraints)
!x()   is the position of each atom
!igrpfixed() is an array of integers. igrpfixed(i)==0 if atom i is immobilized
!coeff is a coefficient used to correct for discrepancies
!      between the "vir()" array and the traditional virial.

subroutine correct_vir_immobile(nat,igrpfixed,f,x,coeff,vir)
   implicit none
   integer:: i, i3, nat
   _REAL_:: coeff
   integer,          intent(in) :: igrpfixed(*)
   _REAL_,           intent(in) :: f(*)
   _REAL_,           intent(in) :: x(*)
   _REAL_,           intent(inout) :: vir(4)

   do i = 1,nat
      if(igrpfixed(i) == 0) then
         i3 = 3*i-3
         vir(1) = vir(1) - coeff*x(i3+1)*f(i3+1)
         vir(2) = vir(2) - coeff*x(i3+2)*f(i3+2)
         vir(3) = vir(3) - coeff*x(i3+3)*f(i3+3)
      end if
   end do
   return
end subroutine correct_vir_immobile
```

When I invoke this function I have been substituting the “ener%vir” array for the “vir” argument (which differs from the traditional virial by a factor of two). To compensate, I have been setting “coeff” to -0.5. (I tried a few different values for “coeff”, and -0.5 works.)

In principle, you can use this function to correct the virial regardless of whether you use harmonic restraints or rigid “belly” constraints to immobilize your atoms. Because I prefer rigid “belly” constraints, I have been invoking “correct_vir_immobile()” in the following way inside “force.f”

```
if (ibelly == 2) then
   call correct_vir_immobile(natom,ix(ibellygp),f,x,-0.5d0,vir)
```

In this code, rescaling of immobilized atom positions is optional. (If the user wants this feature, then they can set ibelly=2, which is currently unused.) Location: I place this line code directly preceding the line where "bellyf()". (This line was modified slightly as well.)

```
if (ibelly > 0) then
   call bellyf(natom,ix(ibellygp),f) !zero forces on belly atoms
```

Note that additional modifications to the code must be made to prevent the rescaling of immobilized atom coordinates. (In sander at least, the relevant subroutine is "ew_pscale()" located in the file "ew_box.f")

A caveat for simulating infinite polymers and surfaces:

Care should probably be taken whenever the constrained object crosses periodic boundaries of simulation. In that case, *I suspect* we should prevent the box from scaling in that direction. (As an example, we tested AMBER by constraining infinite slab that lies along the XY plane. Consequently, we only allow the periodic box to be rescaled in the Z direction at constant pressure conditions. The slab is quite thick, and we also had to make sure that the slab does not accidentally get divided by the Z-boundary when loaded into AMBER/leap.)

A.1 Testing the amended virial code

If the new virial code is working, one would hope that the density of a liquid should not change when roughly half of the water molecules are immobilized.

The new virial code was tested using a box of unconstrained TIP3P water at constant pressure ($P = 1$ bar, and $T = 300°K$) After equilibration, all of the water molecules lying within a $60\AA$-thick slab were immobilized using rigid (belly) constraints, and the simulation was allowed to continue at the same temperature and pressure, while using the fix to the virial code.

Nevertheless, the tests show a 1% increase in liquid density when the constraints are applied, (See figure 2. More precisely, the corresponding volume is 1% of the *total* volume of the water. However the volume decrease amounts to almost 3% of the remaining volume of the free water.)

I don't know how serious this is, whether it means my virial code does not work, or if the compression is due to some interesting physics. (This is part of the reason I am posting these results publicly. Feel free to contact me (jewett.ai on gmail) if you have suggestions, or want to suggest a better way to test the new virial code.) At any rate, this behavior seems to be an improvement from the original behavior (figure 1).

B Simplifying the virial

I skipped a few steps when deriving equation 9. The are included here. Substituting equations 7 and 8 into equation 6 completely eliminates terms

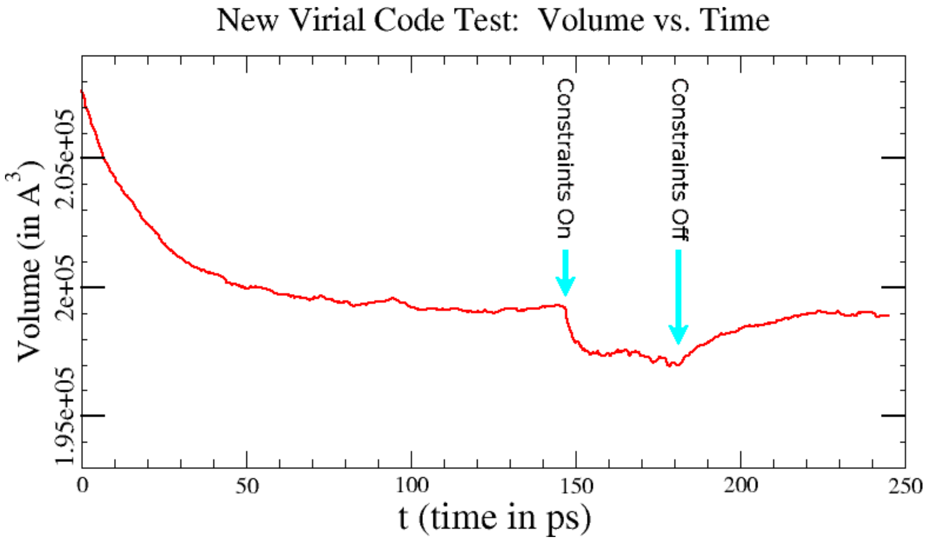

Figure 2: *Volume vs. time for a simulated box of 6550 water molecules at constant pressure, with and without position constraints.* The size of the box was approximately $47 \times 47 \times 90 \mathring{A}^3$. Initially, at $t = 0$, the box had been artificially filled with solvent. During the first $145ps$ of the simulation, the system relaxed to its equilibrium density, without any constraints applied. At $t \approx 145ps$, the conditions of the simulation changed in two ways: 1) Rigid (belly) constraints were applied to the water lying in the central $60\mathring{A}$-thick slab whose center-of-mass z coordinates lie in the range from $5.0\mathring{A} < z < 65.0\mathring{A}$. 2) The molecules in the center of this immobilized slab, $25.0\mathring{A} < z < 45.0\mathring{A}$, were subjected to strong repulsive forces by turning off their charge, as was done in figure 1. The simulations were then continued at constant pressure (holding the box-size fixed in the X and Y directions) until $t = 170ps$, at which point the charges were switched back on. At $t = 180ps$, the constraints were released. This shows that, even after our correction to the virial, constraints do seem to have a modest effect on the density of the liquid.

containing $\vec{f}_{j,\vec{m}\to I}$, $\vec{f}_{J,\vec{m}\to I}$, and reduces the number of terms containing $\vec{f}_{i\to J,\vec{m}}$, and $\vec{f}_{I\to J,\vec{m}}$. We are left with:

$$W = \frac{1}{2}\sum_{\vec{m}\in\mathbb{Z}^3}\left(\sum_{i\neq j}\Big(\vec{r}_i\cdot\vec{f}_{j,\vec{m}\to i} + (\vec{r}_j + L\vec{m})\cdot\vec{f}_{i\to j,\vec{m}}\Big) + \sum_{I,j}\Big((\vec{r}_j + L\vec{m})\cdot\vec{f}_{I\to j,\vec{m}}\Big)\right.$$
$$\left. + \sum_{i,J}\Big(\vec{r}_i\cdot\vec{f}_{J,\vec{m}\to i} + L\vec{m}\cdot\vec{f}_{i\to J,\vec{m}}\Big) + \sum_{I\neq J}\Big(L\vec{m}\cdot\vec{f}_{I\to J,\vec{m}}\Big)\right) \tag{10}$$

This expression for the virial can be written in multiple equivalent ways. (Although it is cleaner to simply calculate the virial in the ordinary way and excluding all the pairwise forces exerted on immobilized atoms.) Note that: $\vec{f}_{i\ \to\ j,\vec{m}}$ (the force from atom i acting on the image of atom j in unit cell $\vec{m}$) is identical to $\vec{f}_{i,-\vec{m}\ \to\ j}$ (the force acting on immobilized atom j, coming from the image of atom i in unit cell $-\vec{m}$).

The terms which (as a result of immobilizing of some of the atoms), are *absent* from this summation (denoted ΔW_{im}), are:

$$\Delta W_{\mathrm{im}} = \frac{1}{2}\sum_{\vec{m}\in\mathbb{Z}^3}\left(\sum_{I,j}\vec{r}_I\cdot\vec{f}_{j,\vec{m}\to I} + \sum_{i,J}\vec{r}_J\cdot\vec{f}_{i\to J,\vec{m}}\right.$$
$$\left. + \sum_{I\neq J}\Big(\vec{r}_I\cdot\vec{f}_{J,\vec{m}\to I} + \vec{r}_J\cdot\vec{f}_{I\to J,\vec{m}}\Big)\right) \tag{11}$$

By using the identities: $\vec{f}_{i\ \to\ J,\vec{m}} = \vec{f}_{i,-\vec{m}\ \to\ J}$ and $\vec{f}_{I\ \to\ J,\vec{m}} = \vec{f}_{I,-\vec{m}\ \to\ J}$, and replacing $\vec{m}$ by $-\vec{m}$, in the sum $\sum_{\vec{m}\in\mathbb{Z}^3}$ (since we are summing over both positive and negative x,y,z components of $\vec{m}$), and combining and eliminating redundant terms (and eliminating the factor of $\frac{1}{2}$), we can justify equation 9.

References

[1] Louis, A., "Computer Simulation Methods in Chemistry and Physics, Part III", Section 5.3, 94-95, (2005) `http://www-thphys.physics.ox.ac.uk/people/ArdLouis/teaching.shtml`

[2] Thompson, A.P., Plimpton, S.J., Mattson, W., "General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions" *J. Chem. Phys.*, **131**, 154107, (2009)