The problem is that \texttt{g\_density} builds up a histogram starting from the bottom of the box. If the bilayer is centered in this box, then the center of the bilayer actually varies (because the box height varies) and so there is error in the form of shaking noise added to the resulting profiles.

You can find more information about the problem and ways to circumvent it in a mailing list post. In particular, it is possible to use a fairly convoluted \texttt{trjconv} scheme to get the correct results with the standard \texttt{g\_density} (see the post below). However, nobody is doing this and quite a few published density profiles along the bilayer normal are wrong because of this:

http://gromacs.5086.n6.nabble.com/problem-with-g-density-center-td4454827.html;cid=1362008104264-151

This is separate from the normalization error that was fixed in version 4.5.2:


I have also attached my own modified version of \texttt{g\_density} from gromacs 4.0.5, which might be useful as a template for somebody else to make the proper changes in a neater way. What I did was to construct the histogram outward from the center of the box. I realize that this is not going to be what people want in all cases, but would be a good option at least. Note that I have only fixed the \texttt{calc\_density()} function, and not the \texttt{calc\_electron\_density()} function so analogous changes would need to be made in all such functions in order to fix all functionality of \texttt{g\_density}.

It's not really a bug because users can get what they want with the linked \texttt{trjconv} routines. Still, this plus the unexpected behaviour of \texttt{trjconv -center} can lead to a fair amount of spread/flattening of density profiles that the regular user doesn't realize.

A simple solution for now would be to add comments to \texttt{g\_density -h} about the way to use \texttt{trjconv} preprocessing to get the right output when using the current version of \texttt{g\_density}, although this will require the modification to \texttt{trjconv} so that it centers the COM and not the location of \texttt{(max-min)/2}.

### Associated revisions

**Revision ee58a7ba - 06/27/2014 08:01 PM - Erik Lindahl**

Update \texttt{g\_density} to handle bilayers better

As suggested by Chris Neale, we now properly perform the binning relative to the center rather than merely shifting the center of the output. While at it, I have also added a selection for the group to center on (useful when the membrane is not in the center of the system, say with a membrane protein), and an option that allows us to perform the binning in relative coordinates and output the average dimensions. The last is useful when there are large box fluctuations. The tool now also has an expanded help section with a few recommendations for bilayers.

Fixes #1168.

Change-Id: id1323111f1d6cd4858f99e8f44a0266667100a406
I'd like to add two more small issues in `g_density` that I think might want to be clarified or changed and thought this might be an appropriate place. Notably, the bCenter flag does not center based on mass if the -dens flag is set to either number or charge, and actually the centering itself is added incorrectly. These bugs are for the GMX 4.6 version of `g_density`.

If bCenter is turned on, the help documentation reads: "Shift the center of mass along the axis to zero. This means if you axis is Z and your box is bX, bY, bZ, the center of mass will be at bX/2, bY/2, 0". This shift is performed in the function center_coords, where the COM of the entire system is determined by iterating over all atoms in the topology, weighting the current position of the atom by its mass determined from the atoms->atom[i].m value in line 141.

The problem is that in the function gmx_density, the masses of all atoms are overwritten by either 1 (for number density) or the charge (for charge density) in lines 520-533. This implies that bCenter is actually centering the system based on one of these two quantities, which may not be desirable and is not clear from documentation. The COM should be correctly calculated for mass density or electron density calculations so this may not affect results a great deal, but I think it is worth noting. It may be safer to just allocate a new array that holds the value of the appropriate mass/charge/electron density/number for the weighting rather than overwriting a value in the topology.

The other problem lies in the calculation and application of the shift vector for centering - on line 153 the rvec shift is calculated using rvec_sub(box_center, com, shift); shift is thus defined as box_center - com = shift. However, this vector is then SUBTRACTED from all coordinates on line 162 with rvec_dec(x0[i], shift). This is incorrect - instead, the shift vector needs to be added to all coordinates. This can be fixed by changing rvec_dec(...) to rvec_inc(...). I verified that this correctly moves the COM to box_center for all axes except "axis", which is moved to 0.

Gerrit received a related patchset '1' for Issue #1168.
Uploader: Erik Lindahl (erik@kth.se)
Change-Id: Id132311f1d6cd485899e8f44a0266667100a406
Gerrit URL: https://gerrit.gromacs.org/3690

- Status changed from New to Fix uploaded
- Status changed from Fix uploaded to Resolved
- Target version changed from future to 5.0