## GROMACS - Feature #1168

g_density does not do what users think it does, given PBC over z and constant pressure simulation

02/28/2013 01:35 AM - Chris Neale

<table>
<thead>
<tr>
<th>Status:</th>
<th>In Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Low</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Erik Lindahl</td>
</tr>
<tr>
<td>Category:</td>
<td>analysis tools</td>
</tr>
<tr>
<td>Target version:</td>
<td>future</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>uncategorized</td>
</tr>
</tbody>
</table>

**Description**

The problem is that 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 trjconv scheme to get the correct results with the standard 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:


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


I have also attached my own modified version of 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 calc_density() function, and not the calc_electron_density() function so analogous changes would need to be made in all such functions in order to fix all functionality of g_density.

It's not really a bug because users can get what they want with the linked trjconv routines. Still, this plus the unexpected behaviour of 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 g_density -h about the way to use trjconv preprocessing to get the right output when using the current version of g_density, although this will require the modification to trjconv so that it centers the COM and not the location of (max-min)/2

**Associated revisions**

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

Update 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: id132311f11d6cd4858f99e8f444a026667100a406

**History**

#1 - 06/09/2013 04:14 PM - Reid Van Lehn

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.

#2 - 06/25/2014 02:51 PM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #1168.
Uploader: Erik Lindahl (erik@kth.se)
Change-id: id1323111d6cd485899e8f44a0266667100a406
Gerrit URL: https://gerrit.gromacs.org/3690

#3 - 06/25/2014 02:51 PM - Erik Lindahl
- Status changed from New to Fix uploaded

#4 - 06/28/2014 08:30 AM - Erik Lindahl
- Status changed from Fix uploaded to Resolved

#5 - 06/28/2014 08:30 AM - Erik Lindahl
- Status changed from Resolved to Closed

#6 - 07/15/2014 08:55 AM - Teemu Murtola
- Assignee changed from David van der Spoel to Erik Lindahl
- Target version changed from future to 5.0

#7 - 10/19/2018 11:32 PM - Chris Neale
- File evaluate_gmx_density.tgz added

This bug still exists up (tested in version 2018.3). The effect will be subtle for realistic bilayer fluctuations. I have attached a test set showing that it is still incorrect using magnified changes. Basically, take any single frame and get the density distribution. Then take that frame and double the box z and add it to the original frame to create a 2-frame trajectory. This should have exactly the same density profile from gmx density, but it does not.

The -relative flag does not fix this.

The code that I uploaded in 2013 appears to have been mostly used, but a crucial part was left out, which was to make the slice width static over the entire trajectory. I built in some protection from attempting to write to unallocated memory by building a slice array that was larger than needed based on the first frame, but that was a x2 array-size hack and could fail with sufficiently large fluctuations.

Here's an example of changes I just made to version 5.1.2 to get the behavior I desire for density. I am not completely sure that I didn't introduce any other problems, and I only allowed modifications to one use case, but the resulting code passes the test that I uploaded here.

```c
$ diff gromacs-5.1.2/src/gromacs/gmxana/gmx_density.c gromacs-5.1.2_fixGmxDensity/src/gromacs/gmxana/gmx_density.c
349a350 > real origbox;
372c373,374 < snew((*slDensity)[i], *nslices);
--- > //snew((*slDensity)[i], *nslices);
> snew((*slDensity)[i], *nslices * 10); // allow overflow since we will not be updating this
374a377,378 > invvol = *nslices / (box[XX][XX] * box[YY][YY] * box[ZZ][ZZ]);
> invvol = *nslices / (box[XX][XX] * box[YY][YY] * box[ZZ][ZZ]);
```

03/20/2020 2/3
// not accounting for this, so disallow it
printf("Error, not coded\n");
exit(1);

---

// this only gets used with bRelative, which is currently disallowed

#8 - 10/20/2018 12:49 AM - Justin Lemkul
- Status changed from Closed to In Progress
- Target version changed from 5.0 to future

Updating status to reflect Chris' latest comment about the issue still persisting.

Files

gmx_density.c 15.6 KB 02/28/2013 Chris Neale
evaluate_gmx_density.tgz 3.22 MB 10/19/2018 Chris Neale