

## GROMACS - Bug #1166

### g\_order is incorrect for unsaturated carbons

02/27/2013 10:35 PM - Chris Neale

<b>Status:</b> New	
<b>Priority:</b> Low	
<b>Assignee:</b> David van der Spoel	
<b>Category:</b> analysis tools	
<b>Target version:</b> future	
<b>Affected version - extra info:</b> 4.5.4	<b>Difficulty:</b> uncategorized
<b>Affected version:</b> 4.5.5	
<b>Description</b>	
The problem (which exists in 4.5.4), and a patch, are located here <a href="http://www.mail-archive.com/gmx-users@gromacs.org/msg41543.html">http://www.mail-archive.com/gmx-users@gromacs.org/msg41543.html</a>	
I haven't checked 4.6, but there is no mention of a fix to g_order in the 4.6 release notes: <a href="http://www.gromacs.org/About_Gromacs/Release_Notes/Versions_4.6.x">http://www.gromacs.org/About_Gromacs/Release_Notes/Versions_4.6.x</a>	
Please note that this patch does not fix g_order for all options, but should be useful for anybody who wants to fully implement the fix.	
I realize that it is most desirable that I upload a fully fixed version of g_order, but I don't have time at the moment. I have therefore opened this issue so that at least it doesn't get forgotten about. A quick fix would be to have the current code throw an error when encountering unsaturated carbons, since these values are wrong (though widely reported in the literature based on g_order).	

### History

#### #1 - 01/16/2014 08:12 PM - Reid Van Lehn

I encountered the bug that Chris refers to here and wanted to chime in on its origin in the existing g\_order code. The source code as stands works by following an approximation for S<sub>cd</sub> nicely derived in this paper: <http://scitation.aip.org/content/aip/journal/jcp/109/6/10.1063/1.476823>. Namely, for saturated carbons, the code defines a new set of molecular axes on the carbon atom C(N) for which the order parameter is calculated, defining a z-axis from C(N-1) to C(N+1), a y-axis that bisects the H-C-H bond angle, and a x-axis orthogonal to the C(N-1)-C(N)-C(N+1) plane. It then makes use of the relation of  $S_{cd} = 2/3 S_{xx} + 1/3 S_{yy}$ , where S<sub>cd</sub> is the desired order parameter that captures the orientation of the C-D vector for this carbon with respect to a reference axis and S<sub>xx</sub> and S<sub>yy</sub> are order parameters reflecting the orientation of the molecular X and Y axes defined above relative to the reference axes. However, the prefactors in this expression (2/3 and 1/3) emerge from the assumed tetrahedral geometry of the central carbon and are related to the approximate cos<sup>2</sup> and sin<sup>2</sup> of 109.5 degrees, the presumed H-C-H bond angle. This expression is thus NOT valid for unsaturated carbons due to a different bond angle geometry. Similarly, the expression  $S_{cd} = 2/3 S_{xx} + 1/3 S_{yy}$  also implicitly cancels out a cross term related to S<sub>xy</sub> due to the assumed equivalence of two hydrogen/deuterium atoms bonded to the central carbon; again this cross term cannot be ignored for an unsaturated bond with only a single bonded hydrogen.

The correct case for handling the an unsaturated bond is derived in the reference above, leading to the relation:  $S_{cd} = 1/4 S_{zz} + 3/4 S_{yy} - \sqrt{3}/2 S_{yz}$ , where the signs in front of the different terms depends on the geometry of the double bond and how the axis S<sub>z</sub> is defined. These prefactors assume an idealized 120 degree bond angle appropriate for the sp<sup>2</sup> carbons included in a double bond. In the version of the g\_order code I looked at (v. 4.6.1), the molecular axis S<sub>z</sub> is correctly redefined for unsaturated bonds for the C9 carbon ONLY, but the incorrect S<sub>cd</sub> expression is then used. The reason C9 results look so wrong compared to C10 (in DOPC/POPC) is likely because the errors of incorrectly defining the S<sub>z</sub> axis for C10 and using the incorrect S<sub>cd</sub> definition lead to a fortuitous cancellation of errors, yielding somewhat correct results for C10 but not C9. I've patched my own local version of g\_order to use the correct expression for S<sub>cd</sub> and am able to reproduce results similar to those of Poger and Mark (<http://pubs.acs.org/doi/abs/10.1021/ct900487a>) and similar to the POPC results reported by Piggot et al (<http://pubs.acs.org/doi/abs/10.1021/ct3003157>). I haven't looked at Chris' patch which I'm sure works fine, but the solution I'm suggesting is in the spirit of the original implementation of g\_order and thus might be easy to implement.

The solution to this fix is thus to fix the definition of S<sub>cd</sub> for unsaturated carbons, and ensure that the molecular axes are always correct - specifically, S<sub>z</sub> always needs to be parallel to the double bond (currently only correct for the C9 carbon as defined in the GROMOS force field, for example). I have a heavily modified version of g\_order that makes these changes amongst others, but is probably too different from the current version to be useful as is. If anyone wants I can clean up the code to remove unnecessary changes and upload it to at least serve as a template for fixing this bug in the future.

#### #2 - 01/16/2014 10:34 PM - David van der Spoel

- Affected version set to 4.5.5

thanks for looking into this issue. your description seems to be rather discouraging as we can not have tools that implicitly assume we are using a certain force field, and it even sounds like the lipid types are hardcoded. we used to have more such tools but are trying to phase them out. does the paper you mention give any formulae that are not dependent on knowing the geometry?

**#3 - 09/02/2014 01:48 PM - Erik Lindahl**

David: g\_order already has an option -unsat that should be used for unsaturated carbons, and the fix here seems to only concern that case. However, the present patch is a bit too dirty to apply right away, so this will have to wait until somebody has time to clean it up (or rewrite g\_order properly).

**#4 - 09/02/2014 04:31 PM - Chris Neale**

Sure, but then g\_order should probably write out a note that it is incorrect for unsaturated carbons.

**#5 - 06/19/2015 12:27 AM - Erik Lindahl**

Reid: If you could attach your modified g\_order source code to this page, it would be great. It might not make it in time for 5.1, but having it there will mean we do add it next time.

**#6 - 06/30/2015 03:46 PM - Chris Neale**

- File gmx\_order\_4.5.4\_modifiedbyChrisNeale.c added

This is the same file for which I originally posted the patch (although a few of the more verbose debugging output statements have been commented out here).

**#7 - 06/18/2017 04:40 PM - Thomas Piggot**

Some of my recent work has been looking at comparing different tools for the calculation of these order parameters. The modified version of g\_order here from Chris still doesn't perform this calculation correctly for the double bond. Reid sent me his code and with some slight modifications I have a version that works correctly for these united-atom unsaturated double bonds. There are also some further modifications that calculate the individual hydrogen atom order parameters in united-atom methylene groups.

As per the comments above by Reid, this version of g\_order: is quite different to the original version (e.g. it takes in a .dat file that defines things for the calculation a la g\_density/gmx density); is just designed to calculate deuterium order parameters (both for a complete membrane and radially outwards from an object); and it is based upon g\_order 4.6.x and so is still written in C (and my coding skills are not good enough to easily make the conversion to C++). However, I can provide this version here, if that will help towards solving this bug?

Cheers

Tom

**#8 - 02/06/2020 02:32 PM - Pacho Ramos**

What is the status of this with 2019.x versions :/

Reading <https://pubs.acs.org/doi/pdf/10.1021/acs.jctc.7b00643> I was also wondering about the results for unsaturated residues are being properly calculated with -unsat option

Thanks

**Files**

---

g_order.patch	7.67 KB	02/27/2013	Chris Neale
gmx_order_4.5.4_modifiedbyChrisNeale.c	33 KB	06/30/2015	Chris Neale