GROMACS - Bug #2192

grompp should read floats (e.g charge) from data files to double, to avoid accumulating round-off error

05/29/2017 01:12 AM - Mark Abraham

Status:	Accepted		
Priority:	Low		
Assignee:	Berk Hess		
Category:	preprocessing (pdb2gmx,grompp)		
Target version:	2021		
Affected version - extra info:		**Difficulty:**	uncategorized
Affected version:	git master		

Description

On the 3.3M lignocellulose benchmark, gmx grompp warns that total charge is -0.000267. In double precision, it does not warn. Each moleculetype in the topology claims a qtot of zero, so probably this is an accumulation issue.

More generally, we should use double anywhere that we haven't measured that performance is worth improving via running in single, just like every other HPC application.

Associated revisions

Revision 635eb138 - 06/07/2017 06:44 PM - Berk Hess

Avoid grompp charge warning with rounding

Even though the grompp total charge check uses double for summation,
there are rounding errors for each charge when charges are stored
in single precision. Now the charge check rounds the net charge of
molecules to integer when the difference is less than the maximum
possible sum of charge rounding errors.

Fixes #2192.

Change-Id: I4e24620ed4ff0901b297db4689e75f0befd23944

History

#1 - 05/29/2017 01:25 AM - Mark Abraham

- Description updated

#2 - 06/02/2017 09:41 AM - Berk Hess

- Status changed from New to Accepted

- Assignee set to Berk Hess

All charge summation in grompp (and even in mdrun for PME corrections) is done in double precision. So I don't see directly why things differ here between a single and double build. Maybe reading and storing each charge in single precision leads to a difference in the last significant bit of some charges which is visible in the double.
If that is the case, the check in sum_q in topio.cpp should not check against 1e-6, but against something like 0.5*GMX_REAL_MIN*atoms->nr.

#3 - 06/02/2017 10:42 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #2192.
Uploader: Berk Hess (hess@kth.se)
Change-Id: gromacs~master~I4e24620ed4ff0901b297db4689e75f0befd23944
Gerrit URL: https://gerrit.gromacs.org/6687

#4 - 06/02/2017 10:45 AM - Berk Hess

- Tracker changed from Feature to Bug

- Status changed from Accepted to Fix uploaded

- Target version set to 2018

- Affected version set to git master

I would say this is a bug, since with PME grompp will generate a warning (not in v2016) and -maxwarn 1 is needed.
I pushed up a change that increases the tolerance for rounding for large molecules. Now we tolerate larger errors for large molecules. The only way to avoid that is reading and storing the charges in double precision in grompp.

#5 - 06/02/2017 11:01 AM - Berk Hess

- Subject changed from grompp should accumulate total charge in double precision to grompp charge precision leads to incorrect total charge warning

#6 - 06/05/2017 04:00 PM - Erik Lindahl

This is a really old issue. I remember looking into it, and the problem is likely that the individual charges themselves with values like 0.3 cannot be represented exactly in floating-point data (and obviously the error is much larger in single), and then it doesn't help that we perform the summation in double.

#7 - 06/08/2017 08:07 AM - Berk Hess

- Status changed from Fix uploaded to Resolved

Applied in changeset 635eb138152e351d7bdaeb2575c8cddae8ec5f78.

#8 - 06/27/2017 11:10 PM - Mark Abraham

- Subject changed from grompp charge precision leads to incorrect total charge warning to grompp should read floats (e.g charge) from data files to double, to avoid accumulating round-off error

- Status changed from Resolved to Accepted

- Target version changed from 2018 to 2019

Retargeted the remaining part of the task to 2018

#9 - 10/22/2018 01:20 PM - Paul Bauer

The reading is already done in double (see toppush.cpp line 241 and 324). Precision is lost when assigned to the underlying data structure in line 484. So fixing the rest would mean changing this in the t_atoms.

#10 - 12/03/2018 01:25 PM - Paul Bauer

- Target version changed from 2019 to 2020

Changing the representation in the TPR would lead to a number of different issues, so this likely won't happen for 2019.

#11 - 12/20/2019 12:11 PM - Paul Bauer

- Target version changed from 2020 to 2021