GROMACS - Bug #2024

gmx check does not notice difference in .tpr files that differ only in LJ parameters

08/04/2016 09:21 PM - Chris Neale

Status:	Closed		
Priority:	Low		
Assignee:			
Category:			
Target version:			
Affected version - extra info:		**Difficulty:**	simple
Affected version:	5.1.2		

Description

I created a system with two magnesium ions. Used the -DMG_ALLNER option in the .mdp file (or not) to get different MG parameters with the charmm36 force field from MacKerell's site. The two .tpr files do indeed differ, but gmx check -s1 -s2 does not notice this.

All tested in gmx 5.1.2.

1. There is a difference in these two .tpr files:

```
$ gmx dump -s regmg.tpr >z.reg
$ gmx dump -s allnermg.tpr >z.allner
$ diff z.reg z.allner
1c1
< regmg.tpr:
---
> allnermg.tpr:
12c12
<    ld-seed                         = 3435682522
---
>    ld-seed                         = 2505241084
190c190
<          functype[0]=LJ_SR, c6= 2.22434992e-05, c12= 1.97089389e-09
---
>          functype[0]=LJ_SR, c6= 2.23024417e-05, c12= 1.00746762e-08
3822c3822
<             type[0]={name="MG",nameB="MG"}
---
>             type[0]={name="MGA",nameB="MGA"}
```

But gmx check -s -2 doesn't pick it up:

```
$ gmx check -s1 regmg.tpr -s2 allnermg.tpr
Reading file regmg.tpr, VERSION 5.1.2 (single precision)
Reading file allnermg.tpr, VERSION 5.1.2 (single precision)
comparing inputrec
inputrec->ld_seed (3435682522 - 2505241084)
comparing top
comparing idef
comparing ilist BONDS
comparing ilist G96BONDS
comparing ilist MORSE
comparing ilist CUBICBONDS
comparing ilist CONNBONDS
comparing ilist HARMONIC
comparing ilist FENEBONDS
comparing ilist TABBONDS
comparing ilist TABBONDSNC
comparing ilist RESTRAINTPOT
comparing ilist ANGLES
comparing ilist G96ANGLES
```

```
comparing ilist RESTRANGLES
comparing ilist LINEAR_ANGLES
comparing ilist CROSS_BOND_BOND
comparing ilist CROSS_BOND_ANGLE
comparing ilist UREY_BRADLEY
comparing ilist QANGLES
comparing ilist TABANGLES
comparing ilist PDIHS
comparing ilist RBDIHS
comparing ilist RESTRDIHS
comparing ilist CBTDIHS
comparing ilist FOURDIHS
comparing ilist IDIHS
comparing ilist PIDIHS
comparing ilist TABDIHS
comparing ilist CMAP
comparing ilist GB12
comparing ilist GB13
comparing ilist GB14
comparing ilist GBPOL
comparing ilist NPSOLVATION
comparing ilist LJ14
comparing ilist COUL14
comparing ilist LJC14_Q
comparing ilist LJC_NB
comparing ilist LJ_SR
comparing ilist BHAM
comparing ilist LJ_LR
comparing ilist BHAM_LR
comparing ilist DISPCORR
comparing ilist COUL_SR
comparing ilist COUL_LR
comparing ilist RF_EXCL
comparing ilist COUL_RECIP
comparing ilist LJ_RECIP
comparing ilist DPD
comparing ilist POLARIZATION
comparing ilist WATERPOL
comparing ilist THOLE
comparing ilist ANHARM_POL
comparing ilist POSRES
comparing ilist FBPOSRES
comparing ilist DISRES
comparing ilist DISRESVIOL
comparing ilist ORIRES
comparing ilist ORDEV
comparing ilist ANGRES
comparing ilist ANGRESZ
comparing ilist DIHRES
comparing ilist DIHRESVIOL
comparing ilist CONSTR
comparing ilist CONSTRNC
comparing ilist SETTLE
comparing ilist VSITE2
comparing ilist VSITE3
comparing ilist VSITE3FD
comparing ilist VSITE3FAD
comparing ilist VSITE3OUT
comparing ilist VSITE4FD
comparing ilist VSITE4FDN
comparing ilist VSITEN
comparing ilist COM_PULL
comparing ilist EQM
comparing ilist EPOT
comparing ilist EKIN
comparing ilist ETOT
comparing ilist ECONS
```

```
comparing ilist TEMP
comparing ilist VTEMP
comparing ilist PDISPCORR
comparing ilist PRES
comparing ilist DH/DL_CON
comparing ilist DV/DL
comparing ilist DK/DL
comparing ilist DVC/DL
comparing ilist DVV/DL
comparing ilist DVB/DL
comparing ilist DVR/DL
comparing ilist DVT/DL
comparing atoms
comparing block cgs
comparing block mols
comparing blocka excls
comparing groups
comparing flags
comparing box
comparing box_rel
comparing boxv
comparing x
comparing v

gcq#401: "It's easy to remember: a half a kT is equal to five fourths of a kJ/mol." (Anders Gabrie
lsson)
```

Associated revisions

Revision fc0b6a72 - 12/19/2017 01:57 AM - Erik Lindahl

Use reduced default tolerances for tpx comparison

The tolerances for gmx check are mainly intended
for handling slight statistical deviations, but they
can hide differences between tpr files, when the
user likely wants exact checks. This changes
changes the default relative tolerance to 0.000001
and the absolute tolerance to zero, so that we only
allow for minor differences due to compiler optimization.

Fixes #2024.

Change-Id: I55b882a194d931bf5c36541e25339b6e1eb0a1e4

History

#1 - 08/05/2016 12:37 AM - Mark Abraham

- Description updated

#2 - 08/08/2016 02:27 PM - Berk Hess

- Status changed from New to Accepted

I assume your issue arises because your difference is smaller than the default absolute tolerance of 0.001. Try check -abstol 0.
We might want to make 0 the default value for -abstol.

#3 - 08/09/2016 06:44 AM - Chris Neale

Good catch. If you want to avoid setting a default of zero to a real number comparison, perhaps set the tolerance based on the number of significant digits, not on an absolute real number?

#4 - 08/10/2016 10:19 AM - Berk Hess

But that's effectively what the relative tolerance already does.

#5 - 08/10/2016 03:36 PM - Chris Neale

I see. Thanks for pointing that out.

I don't understand then why the differences of 2.22434992e-05 vs. 2.23024417e-05 and 1.97089389e-09 vs. 1.00746762e-08 were not picked up

(unless gmx check is applying both absolute and relative tolerances? -- if this is the case, is there a good reason to keep the absolute tolerance rather than using a relative tolerance also with zero?)

#6 - 12/17/2017 12:35 PM - Erik Lindahl

Sorry for not addressing a (very) old comment eariler.

The problem with relative tolerances is that if the reference value is zero, the tolerance is also zero. For **tpr** files that might actually be a good idea, but the way the options work it's difficult to have different defaults for different types of files. For a trajectory or energy file, some terms (e.g. forces) are sums of many components but the sum might still be very close to zero. Already for a medium-size trajectory, this is pretty much guaranteed to be the case already for step 1 which gives the user a ton of differences unless we allow for an absolute tolerance by default.

I will upload a fix that uses different defaults just for TPR files.

#7 - 12/17/2017 12:35 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #2024.
Uploader: Erik Lindahl (erik.lindahl@gmail.com)
Change-Id: gromacs~release-2018~I55b882a194d931bf5c36541e25339b6e1eb0a1e4
Gerrit URL: https://gerrit.gromacs.org/7358

#8 - 12/18/2017 05:33 AM - Mark Abraham

Moral: even gmx check is too complex

#9 - 12/19/2017 02:02 AM - Erik Lindahl

- Status changed from Accepted to Resolved

Applied in changeset fc0b6a720137828e1071799526d918947494e9c2.

#10 - 12/20/2017 03:01 PM - Erik Lindahl

- Status changed from Resolved to Closed

Files

allnermg.tpr	61.2 KB	08/04/2016	Chris Neale
regmg.tpr	61.2 KB	08/04/2016	Chris Neale