Gromacs - Feature #879
Normal modes with vsites and/or shells does not work
02/03/2012 07:20 PM - David van der Spoel

Status: Accepted
Priority: Low
Assignee: David van der Spoel
Category: 
Target version: 5.x

Description
Tested normal mode analysis with different small molecules. TIP5P gives NaN everywhere in the mtx file. TIP4P gives strange numbers as well, while models with shells yield weird numbers. I guess that in all cases mdrun should only work on the real atoms and not on the vsites/shells.

Associated revisions
Revision c289d8f7 - 05/22/2014 09:21 AM - Rossen Apostolov
Normal modes don't work currently with virtual sites or shells.
Refs #879.
Change-Id: I1c45b5a4b4c97eff222dccbbbb884e0153ad0c5

History
#1 - 03/05/2012 10:53 AM - Berk Hess
- Status changed from New to Feedback wanted

This is not trivial to fix. You can not have mdrun only work on the normal atoms. You need to put in the dependence of the coordinates which are a function of the mass into the matrix calculation. Herman Berendsen derived a formula for shells. We need to add a formula for virtual sites, which shouldn't be hard. Who has time to look into this?

#2 - 03/05/2012 11:45 AM - David van der Spoel
I will try. Do you have any equations written down? I would say that in both cases the normal mode calculation should be performed on the particles with particle type atom, not vsite or shell. Then, if you displace an atom you have to recompute the virtual sites, and in the case of shells, minimize them. Is it more complex than that? Maybe the mtx file should only store the atomic hessian, otherwise g_nmeig should be changed too.

#3 - 12/29/2012 12:10 AM - Erik Lindahl
- Target version changed from 4.6 to 5.0

No update in 10 months, so changing target to 5.0.

#4 - 01/09/2014 02:01 PM - Rossen Apostolov
- Priority changed from Normal to Low
- Target version changed from 5.0 to 5.x
- Affected version set to 4.5.1
moving to a future version

#5 - 04/22/2014 02:15 PM - Gerrit Code Review Bot
Gerrit received a related patchset ‘1’ for Issue #879.
Uploader: Rossen Apostolov (rossen@kth.se)
Change-Id: i534566ac6eefe1c6124b45f2e24ced7a128477b3
Gerrit URL: https://gerrit.gromacs.org/3390

#6 - 04/22/2014 02:18 PM - Rossen Apostolov
I pushed a patch to disable the combination for now.

#7 - 05/06/2014 04:25 PM - Gerrit Code Review Bot
Gerrit received a related patchset ‘1’ for Issue #879.
Uploader: Rossen Apostolov (rossen@kth.se)
Change-Id: l1c45b5a4b4c97eff222dcbbbb884e0153ad0c5
Gerrit URL: https://gerrit.gromacs.org/3412

#8 - 05/12/2014 02:59 PM - Rossen Apostolov
- Status changed from Feedback wanted to In Progress

#9 - 05/22/2014 04:46 PM - Erik Lindahl
- Status changed from In Progress to Closed

#10 - 05/22/2014 04:48 PM - Erik Lindahl
- Tracker changed from Bug to Feature
- Status changed from Closed to Accepted

"re-opening" this as an accepted feature. For now Rossen's patch will disallow this, but at some point in the future it would be good to have native support for it.

#11 - 09/30/2015 08:08 PM - Gerrit Code Review Bot
Gerrit received a related patchset ‘1’ for Issue #879.
Uploader: David van der Spoel (davidvanderspoel@gmail.com)
Change-Id: lac127f51ef4ebeb246ce9d24176e039778b5952122f
Gerrit URL: https://gerrit.gromacs.org/5146

#12 - 09/30/2015 10:00 PM - Berk Hess
I think that for shells the trick is as follows:
You can write the Hessian for the full system of masses+shells as:
AC^T
CB
where A is the Hessian for the masses and B for the shells. Then the Hessian for the masses coupled by the shells is:
A - C^T B^-1 C
I don't know if this is better than the solution of minimizing the shells for each atom displacement.

11/25/2015
2/3
And how about vsites? That is also a matrix transformation of course. Question is rather, are we missing something in the present implementation?

For vsites you would need to figure out matrix C for every construction type, which is tedious. Since the vsite construction is exact (and I assume handled correctly) in your current change, there is no reason to use the matrix trick. For vsites there is the iterative minimization, which requires setting and extremely low tolerance, as low as the tolerance used to minimize the structure, and it requires iterations that could increase the cost of the matrix calculation by an order of magnitude. But I guess the matrix calculation is still cheap. The matrix trick would avoid those two issues, but both are not really problematic.

OK, now that this passed the build tests in Jenkins we should consider whether we want to put much more effort in this. I can build in a warning about the tolerance for shells. I have some test cases but do not know how easy that is to add to the regression tests seeing that you can only run this really in double precision.