GROMACS - Bug #1429
Feature # 1292 (Closed): mdrun features to deprecate for 5.0

shell code issues (broken with DD+grid; broken with Verlet scheme)

01/31/2014 07:58 PM - Mark Abraham

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Berk Hess</td>
</tr>
<tr>
<td>Category:</td>
<td>mdrun</td>
</tr>
<tr>
<td>Target version:</td>
<td>4.6.6</td>
</tr>
<tr>
<td>Affected version - extra info:</td>
<td></td>
</tr>
<tr>
<td>Affected version:</td>
<td>4.6.5</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>uncategorized</td>
</tr>
</tbody>
</table>

Description
Removing particle decomposition requires that the regression test perl script stops always triggering -pd when simple grid searching is being used. Currently, the complex/sw test uses the shell-model polarization code, with simple, thus -pd (and vsites and "flexible") constrains). This would have to change to grid+DD when we remove PD.

Using installs of either master or release-4-6 HEAD with

gmxtest.pl complex -only sw\b

by changing the grompp.mdp and/or gmxtest.pl files, I observed:
-pd and simple passes the test with 1-8 tMPI ranks
-pd and grid passes the test with 1-8 tMPI ranks
-dd and simple passes the test with 1 rank, and gives the expected fatal error with DD+simple unsupported (perl script hack required to remove the hard-coded -pd flag)
-dd and grid passes the test with 1 rank, and fails with all higher numbers

The fails are pretty subtle:

comparing energy file reference_s.edr and ener.edr

There are 31 terms in the energy files

There are 5 terms to compare in the energy files

<table>
<thead>
<tr>
<th>Angle</th>
<th>step 19:</th>
<th>71.2642, step 19: 71.1485</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle</td>
<td>step 20:</td>
<td>70.2909, step 20: 70.1638</td>
</tr>
</tbody>
</table>

Files read successfully

So, if there's a problem, then I think it is already present in the code and release-4-6 should get a fatal error for shells + DD and ranks > 1. It could have the same cause as the problem that was swept under the carpet in #993 (the forces may not be very reproducible in single precision), but when I tested release-4-6 in double precision I got the same error with DD, grid, and -ntmpi > 1.

To see if OpenMP was useful as parallelism, I tried to test the shell code with the Verlet scheme (using verlet-buffer-drift=-1 because the test case has a box that's only just large enough for the unmodified rlist), and it segfaulted with both grid and simple, even with just 1 rank and 1 thread. So I believe that needs a fix or a fatal error, also.

The above poses a problem for the proposed removal of PD - the existence of what might be an historical bug reduces the support in 5.0 for shells from "MPI with group scheme and PD" to "serial with group scheme." We have a few open bugs for / suspicions of the shell code (#879, #993, #713). I suggest we proceed with removing PD, and require the complex/sw test to run with 1 rank, unless/until someone wants to find out what the real problem is. Hopefully, with the interest in polarizable models expressed in #1292 that will be sooner rather than later!

Associated revisions
Revision a4c5fb33 - 02/08/2014 09:45 PM - Mark Abraham
Issue fatal errors rather than use broken shell code

05/05/2020
Revision e9e67f53 - 02/19/2014 08:27 PM - Mark Abraham

Fix sw test case

The former version had trouble reproducing the serial results in some cases in parallel. This change replaces the input conf.gro file with the confout.gro file from the former version of the test run in Reference mode, because that starting configuration seems more stable. The former input .gro file had velocities for particles named DW, but it is not known whether this is related to the problem.

Also switched it to use ns-type=grid in line with future changes in this area.

Refs #1429

Revision c949a5ac - 02/21/2014 01:44 AM - Mark Abraham

Reinstate shell code with DD

Further work on the complex/sw test case in the 5.0 regressiontests branch reveals that the initial conditions may have been the reason for the problems observed with DD and more than one node, rather than the implementation.

Refs #1429

Revision debd482d - 06/13/2014 08:32 AM - Berk Hess

Made shells work with the Verlet scheme

The issue was that atoms iso charge group should be put in the box. Additionally water_pol only worked within a charge group, now fixed. Note that with shells working across charge groups, which is always the case with the Verlet scheme, no shell prediction is done. We should implement prediction, since it improves performance.

Fixes #1429.

Change-Id: l2ebfc2d91fc161167f82573b61e11f519c11fd8

Revision 0b396d5f - 06/20/2014 09:17 AM - Berk Hess

Fixed shells with particle decomposition

In the spirit of version 5.0, the recent Verlet scheme fix for shells, commit debd482d, removed the support for PD. Now it's back.

Refs #1429.

Change-Id: l686d4287ec8946e418aa98e739a1a81a0b7f7055

History

#1 - 02/04/2014 08:33 PM - Justin Lemkul

Thanks for pointing this out, Mark. I'm currently working through the code to see which parts I can reuse and which functions I will have to write on my own for our implementation of the Drude model. I remain hopeful that Drude will be fairly straightforward and should play nicely with parallelism, DD, etc because the Drudes are not really special - they're just low-mass particles in our force field. There are some bells and whistles with bonded interactions, temperature, and Thole screening, but those should be fairly simple to add in. Nothing I'm planning to do should be inherently dependent upon PD, unless I am reusing some aspect of the shell code that needs it. If that's the case, I'm sure I'll come across it and will probably just rewrite whatever that function happens to be.

#2 - 02/04/2014 10:16 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #1429.

Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: l18a17f1e232a86a13f4e3b591bd992702a93017b
Of course DD isn't supposed to give the exact same results for different number of domains. But it is odd that for just complex/sw the difference between 1 domain and 2 is larger than our margin. Also the results for 2,3,4,5 domains are all very similar. On the other hand if one takes the results after 20 steps (confout.gro) generated with either 1 or 2 domains as starting configuration (conf.gro), and regenerates the reference values, then the tests pass. So it might just be that the starting configuration is particularly sensitive to the rounding error of DD.

I have implemented Roland's suggestion as a replacement for the sw test case at https://gerrit.gromacs.org/3156

But without charge groups there is not shell prediction. This means that for the test we need reasonable initial shell coordinates, which can done by e.g. running one step of md and using confout.gro. Also not using prediction deteriorates performance, I filed a feature issue #1522 for this.

The uploaded a fix, which was relatively simple. But without charge groups there is not shell prediction. This means that for the test we need reasonable initial shell coordinates, which can done by e.g. running one step of md and using confout.gro. Also not using prediction deteriorates performance, I filed a feature issue #1522 for this.

The uploaded a fix, which was relatively simple. But without charge groups there is not shell prediction. This means that for the test we need reasonable initial shell coordinates, which can done by e.g. running one step of md and using confout.gro. Also not using prediction deteriorates performance, I filed a feature issue #1522 for this.

The uploaded a fix, which was relatively simple. But without charge groups there is not shell prediction. This means that for the test we need reasonable initial shell coordinates, which can done by e.g. running one step of md and using confout.gro. Also not using prediction deteriorates performance, I filed a feature issue #1522 for this.

The uploaded a fix, which was relatively simple. But without charge groups there is not shell prediction. This means that for the test we need reasonable initial shell coordinates, which can done by e.g. running one step of md and using confout.gro. Also not using prediction deteriorates performance, I filed a feature issue #1522 for this.