GROMACS - Feature #2309

Feature # 2307 (Closed): Implement MiMiC-based QM/MM functionality in GROMACS

Implement MiMiC QM/MM workflow in the GROMACS core

11/28/2017 01:08 PM - Viacheslav Bolnykh

<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>Viacheslav Bolnykh</td>
</tr>
<tr>
<td>Category:</td>
<td>core library</td>
</tr>
<tr>
<td>Target version:</td>
<td>2019</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>hard</td>
</tr>
</tbody>
</table>

Description

In order to support the new QM/MM workflow inside of the GROMACS core we need to add linking against MiMiC communication library and implement the protocol (as written in attached pdf file). Moreover, we need to modify the local topology building to extend the local exclusion lists (both for serial and parallel configurations) with the intermolecular exclusion of interactions between QM atoms (since they are going to accounted in the QM code).

Associated revisions

Revision 1113f056 - 10/12/2018 10:34 AM - Viacheslav Bolnykh

Implemented QM/MM workflow inside core GROMACS
Implemented MiMiC integrator along with the generation of inter-molecular exclusion lists for QM atoms both in serial and parallel parts of the code

Fixes #2309

Change-Id: I35073ad69752e22968c0e2821e2fd9f0c465b2a

History

#1 - 11/28/2017 01:32 PM - Viacheslav Bolnykh
Presumably, this has to be done in a separate function called do_mimic() - similar to do_md but without things that we won't need (e.g. integrator or thermostat)

#2 - 11/28/2017 02:16 PM - Aleksei Iupinov
- Parent task set to #2307

#3 - 11/28/2017 05:58 PM - Mark Abraham
- Target version changed from 2018 to 2019

#4 - 11/30/2017 11:41 PM - Erik Lindahl
I'm fairly certain we want a specific do_mimic() function in Gromacs (at least not unless there is a very active core developer taking care of it), but a better solution seems to be if you work on implementing the exchange APIs you need so these can be called either from C++ or (better) Python, and make them general enough to work with any code.

Have you discussed QM/MM interfaces with the people from MolISSI? Since they are working along similar lines, it seems like a good idea to standardize on an interface.

#5 - 12/01/2017 09:39 AM - Viacheslav Bolnykh
Erik Lindahl wrote:

I'm fairly certain we want a specific do_mimic() function in Gromacs (at least not unless there is a very active core developer taking care of it), but a better solution seems to be if you work on implementing the exchange APIs you need so these can be called either from C++ or (better) Python, and make them general enough to work with any code.

We are trying to avoid tight coupling between QM and MM codes (we do not build a single binary for the QM/MM). Instead we have two programs running in parallel interacting through the communication layer. Currently we are using the client-server mechanism implemented in the MPI which allows to attach processes pool of one communicator to another one as a remote group. After that the interaction is done using standard MPI calls on the intercommunicator (through the communication library API). Currently, in our fork of GROMACS we directly implement the protocol, described in...
the PDF document (from communication point of view - we just send particular data in a particular order, which, I agree, is not very robust). Within this strategy the only way I think it may be possible to create a callable API is to run a separate thread in GROMACS, which will be only handling communication with a partner code (CPMD in our case but in this approach it will be rather general). Messages received by this thread will be decoded into particular commands that GROMACS will perform. Should I proceed with this approach or is there another possible solution?

Have you discussed QM/MM interfaces with the people from MolSSI? Since they are working along similar lines, it seems like a good idea to standardize on an interface.

Thank you very much for this suggestion - we will do this in the nearest future.

#6 - 12/30/2017 11:29 PM - Erik Lindahl
Sorry, I just saw a typo above: I'd like to avoid a specific communications layer for each other program, and rather write general ones that are used by more codes.

The only reason for using threads is really if something must be handled asynchronously, and I don't see any obvious need for that - we can't start with the next timestep until the data is available, and there's no point in getting the data earlier than it's needed, so why does the communication have to be handled in its own thread?

#7 - 01/03/2018 11:20 AM - Viacheslav Bolnykh

Erik Lindahl wrote:

Sorry, I just saw a typo above: I'd like to avoid a specific communications layer for each other program, and rather write general ones that are used by more codes.

The only reason for using threads is really if something must be handled asynchronously, and I don't see any obvious need for that - we can't start with the next timestep until the data is available, and there's no point in getting the data earlier than it's needed, so why does the communication have to be handled in its own thread?

Yes, this is correct - there is no need for a separate thread. For now I have implemented the naive communication protocol by just sending data in a certain order but I will later re-work it to turn it into API calls.

#8 - 01/05/2018 07:09 PM - Gerrit Code Review Bot
Gerrit received a related DRAFT patchset '1' for Issue #2309.
Uploader: Viacheslav Bolnykh (bolnykh@gmail.com)
Change-Id: gromacs~master~I35073ad69752e22968c1e2821e29f0c465b2a
Gerrit URL: https://gerrit.gromacs.org/7445

#9 - 10/08/2018 06:53 PM - Mark Abraham
- Status changed from New to Fix uploaded

#10 - 10/12/2018 10:45 AM - Viacheslav Bolnykh
- Status changed from Fix uploaded to Resolved

Applied in changeset 1113f0566319750c05503c4b965e49b5eb0bd85c1.

#11 - 10/12/2018 12:22 PM - Mark Abraham
- Status changed from Resolved to Accepted

Berk made some final feedback minutes after I submitted the code, please consider addressing them

#12 - 10/15/2018 05:15 PM - Mark Abraham
- Status changed from Accepted to Closed

That got fixed

Files

| protocol_doc.pdf | 227 KB | 11/28/2017 | Viacheslav Bolnykh |