Gromacs - Task #1415
Get rid of legacyheaders/
01/02/2014 05:08 PM - Teemu Murtola

Status: New
Priority: Normal
Assignee: Mark Abraham
Category: core library
Target version: 5.x

Description
The legacyheaders/ directory was meant as a temporary workaround to be able to move files to the new directory layout gradually, while keeping the build and the installed headers working all the time. The original idea (when the discussion started back in 2010 or so) was that this would be more or less gone by 5.0, but it has been quite persistently alive. So the goal would be to have all the headers from src/gromacs/legacyheaders/ moved into subdirectories like src/gromacs/domdec/ or src/gromacs/nbnxn/ or something, together with the source files that implement those functions and anything that belongs naturally together with them. Large parts of this should not be rocket science, and if people would contribute on the areas that they know, we could then see what are the problematic areas. The initial division can also be adjusted later as we grow wiser.

There are currently about 150 headers remaining, plus 38 from thread_mpi (the latter should be easy to move, once there is some clarify for #1411). Large parts of this will be easy, but the types/ subdirectory can be challenging.

This would allow us to clearly see the dependencies between the modules and draw reasonable division lines for the different libraries required for, e.g., #1165. Visualizing the dependencies between the parts with something like https://gerrit.gromacs.org/#/c/2908/ would provide a much clearer picture of where we are and what could be done to clarify the structure of the code.

Associated revisions
Revision fdbd0916 - 01/11/2014 02:41 PM - Carsten Kutzner
Move essential dynamics / flooding code to separate directory
Part of #1415 Get rid of legacyheaders.
Additional changes:
- provided doxygen-style documentation for functions in edsam.h
  (partly copied over from edsam.c and removed there to avoid duplication)
- uncrustified edsam.c and edsam.h
Change-Id: If4af498470c5200d50ac1d1b5537917c873380b5

Revision 904e082b - 01/13/2014 08:06 PM - Carsten Kutzner
Move pull code to separate subdirectory
Another part of #1415 Get rid of legacy headers.
Other changes:
- updated pulling-related include guards to new format
- updated documentation for doxygen
- updated .h includes
- uncrustified pull.h and pull_rotation.h
- updated copyright year for all files touched by this commit
Change-Id: lef1ed3543b84f651d1bf01ff1952b342850f5c22
Revision 2b85adec - 01/21/2014 07:37 PM - Mark Abraham

Move more preprocessing files into the module

This removes the need for the gmx_objlib CMake object library, because there is no longer random functionality orphaned in src/programs/gmx (which is there because it was orphaned in src/kernel in release-4-6).

Moved legacyheaders/grompp.h to gmxpreprocess/grompp-impl.h since it only declared structures that are (now) only used in that module (and it no longer #includes stdio.h). gmxpreprocess/grompp.h now only declares gmx_grompp().

Likewise, pdb2gmx, protonate and x2top now have their own single-function headers.

Removed useless declaration of ncontrol from h_db.h

Rewritten grompp.h-related useless dependencies, declarations and definitions from Generalized Born files.

Updated include and added C++ guards for files already within the module (and new ones). Uncrustified, bumped copyrights

Refs #1415

Change-Id: le7967d7d7f61d84d530e3c2d982486a3cd128f

---

Revision d3ac3c5b - 04/01/2014 02:00 AM - Teemu Murtola

Move smalloc.h to utility/

One common header away from legacyheaders/.

Mostly mechanical replacement for the include paths, but removed a few instances where the header was included more than once. Also, reordered the headers a bit, mainly in C++ files.

Part of #1415.

Change-Id: lc438d501ebf5ecc285655a9e82f625eb212f1d1
Revision 2ca0f555 - 04/02/2014 03:11 PM - Teemu Murtola
Move string2.* to utility/cstringutil.*

Another common header away from legacyheaders/.
Some associated #include cleanup.
This reintroduces some unwanted dependencies (like utility/ -> fileio),
but those need to be fixed separately.

Part of #1415.
Change-Id: lffb2820019e2b41ef74bcabcd202364917e43d58

Revision 1fe1fb54 - 04/08/2014 05:38 PM - Teemu Murtola
Move gmx_fatal.* to utility/fatalerror.*

One more included-by-all header to utility/.
Mainly mechanical replacement of include paths, some reordering in C++
files. Minor clean-up in the moved files.
gmx_fatal_collective.h not moved at this point, as it is unclear whether
it should go here, or somewhere else (utility/ should not depend on
t_commrec...).

Now, it's mainly types/simple.h that prevents removing some module-level
dependencies to legacyheaders/.
Finding a good location for that (or its contents, if it needs to be split), and reorganizing the code a bit
more should now allow really putting utility/ at the bottom of the
dependency stack where it belongs.

Part of #1415.
Change-Id: l6e487566d5bd7c9ec7a1455a2d31bd6d45bada3

Revision 83847a3c - 04/16/2014 03:45 AM - Teemu Murtola
Remove sysstuff.h

Replace this with those standard headers that are actually necessary for
the source files. There are likely several transitively covered header
dependencies now, but those can be resolved when they give problems.
One more header off from legacyheaders/.

Part of #1415.
Change-Id: l4cc3f0eec0f546dad929512e532d8efb11da3694
Revision 0d0e128c - 04/16/2014 03:46 AM - Teemu Murtola
Move xvgr.* to fileio/
Part of #1415
Change-Id: lb3d78bce62a4c03c4d72001164921edd55657a2

Revision 826d049d - 04/17/2014 11:12 AM - Teemu Murtola
Move vec.h to math/
Most of the changes are scripted replacement of include paths.
Exceptions:
- #includes in the moved file were cleaned up slightly, and include guards adjusted.
- Places where it was included as "gromacs/legacyheaders/vec.h" were handled manually and the include order adjusted.
- Removal of unnecessary #include "vec.h" from update.h revealed some transitive dependencies that were fixed. Some associated include order cleanup in these files.
- Removed some duplicate includes.
- Remove dependency from linearalgebra/matrix.c to vec.h (only needed for a single sqr()) to avoid a cyclic math <-> linearalgebra dependency.
Part of #1415.
Change-Id: l7ece6eafa22139215510148807c5979f8950a885

Revision 7b9f9426 - 04/20/2014 07:58 PM - Roland Schulz
Move part of macros.h to grompp-impl.h
Part of #1415.
Change-Id: l360ae1f220b6c51e90edbbe136053fd2e1c0ad5b

Revision 3126f116 - 04/27/2014 01:19 PM - Teemu Murtola
Move 'real' definition to utility/real.h
- types/simple.h still includes the new header to avoid touching all code depending on this.
- Go through most files outside gmxlib and mdlib that included types/simple.h, and checked whether headers from utility/ could fulfill the needs. Replaced at least in places where this helped fully remove legacyheaders/ dependencies.
- Some minor changes elsewhere to remove legacyheaders/ dependencies from modules. Now linearalgebra/ and statistics/ don't depend on legacyheaders, improving the layout of the dependency graph significantly.

In the long run, this may be better outside utility/ to better distinguish code that depends on 'real' from parts that do not. Currently, it would create a lot of cyclic dependencies everywhere else.

Part of #1415.

Change-Id: l2aa1890357377d0d14c05b3968e1b36e1023458f

Revision 83fd3f60 - 04/27/2014 09:12 PM - Teemu Murtola

Move rvec and friends to math/vectypes.h

Adjust a few more types/simple.h includes to use only its component parts. Now, math/ has only very few legacyheaders/ dependencies left.

Part of #1415

Change-Id: lffea4fac5c5e0c744cf6deecac4eedae2caf5

Revision 2cc66cb6 - 04/28/2014 04:24 AM - Roland Schulz

Reduced usage of typdefs.h

Preliminary step to move types out of legacyheader.

Renamed parameter called "matrix" because cppcheck gets confused if a parameter is called the same as a type.

Part of #1415

Change-Id: l9207a7b8de7f092bab890da128abb48a8d97b79

Revision 604cd696 - 05/14/2014 09:33 PM - Teemu Murtola

Move physics.* to math/units.*

- Remove some unnecessary #includes.
- Remove the dummy tests; easy to rewrite as usable tests using GTest if someone needs those. They were anyways testing code that no one is still calling, even though it was introduced several years ago.

Now gmxlib is free of *_.test.c files.
Now, only one file using atom_id is keeping math/ dependent on legacyheaders/.

Part of #1415.

Change-Id: i31312b92cf9f5df5129fbd90f31fc27be2f7aaf

---

Revision 2b5c9ac3 - 09/22/2015 01:34 PM - David van der Spoel
Removed legacyheaders/types/commrec_fwd.h from typedef.h

Related to #1415
Change-Id: f9a9dd7f1e62cf040d183370d187f6a23ede78fa

---

Revision 6cc44d3e - 11/03/2015 09:26 PM - David van der Spoel
Moved df_history utility routines from typedefs to new file.

Part of #1415.
Change-Id: la2b9e6e5342cbb5820894fb82ef2908d969bc6

---

Revision 2a9f0384 - 11/11/2015 12:53 AM - David van der Spoel
Removed legacyheaders/typedefs.h

More cleaning up in legacyheaders. Mostly this patch involves shuffling around include files, and adding a couple of forward declarations. Renames gmxlib/typedefs.* to mdtypes/state.* and introduces new utility/int64_to_int.cpp with header.

Introduced new directory mdtypes to resolve circular dependencies.

Right now state.* and history files are the only files living there.

Part of #1415
Change-Id: i5dc59a2ca3bd1f9c5391f4b9deb9f116b30106fc

---

Revision 890ce4e8 - 11/11/2015 10:10 PM - David van der Spoel

Moved inputrec.* from gmxlib and legacyheaders to mdtypes.

Part of #1415
Change-Id: ldb00712d1b6ea9648a36d349f670790aaaf2c6d4
Revision e4034c03 - 11/12/2015 08:44 AM - David van der Spoel
Moved copyright.* to fileio from gmxlib and legacyheaders.
Part of #1415
Change-Id: ld6cfb84cc683f88731fac7d80053aad69828806

Revision 56a3f36 - 11/12/2015 04:37 PM - David van der Spoel
Moved txtdump.* to fileio.
Some cleanups in edsam.cpp too.
Part of #1415
Change-Id: l6cd58951eff1bac0dad75dbd448a958e57c767e7

Revision 99e5a7cc - 11/13/2015 08:05 AM - David van der Spoel
Moved genborn.h to mdlib
Part of #1415
Change-Id: la2d734be40df97426348a153040cf311a3abf7a3

Revision e8e13a0e - 11/15/2015 10:21 AM - David van der Spoel
Merged legacyheaders/types/state.h into mdtypes/state.h
Also merged swap/enum.h into legacyheaders/types/enums.h
to be moved later out of legacyheaders.
Part of #1415, #1854.
Change-Id: le53f190d6798b81c531896de80ef250c60836479

Revision a3be7872 - 11/15/2015 01:46 PM - David van der Spoel
Merged legacyheaders/types/inputrec.h into mdtypes/inputrec.h
Part of #1415.
Change-Id: l39885a01474173170bc6a7a7c2e54e721d64370

History
#1 - 01/07/2014 06:52 PM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #1415.
Gerrit received a related patchset ‘1’ for Issue #1415.
Uploader: Carsten Kutzner (ckutzne@gwdg.de)
Change-Id: If4af498470c5200d50ac1d1b5537917c873380b5
Gerrit URL: https://gerrit.gromacs.org/2961

Gerrit received a related patchset ‘1’ for Issue #1415.
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: le7967d7d7616d684d530e3c2d982486a3c3d12f
Gerrit URL: https://gerrit.gromacs.org/2993

Gerrit received a related patchset ‘1’ for Issue #1415.
Uploader: Teemu Murtola (teemu.murtola@gmail.com)
Change-Id: lc438d501eb5ecc2e5655a9e82f25eb12f1d1
Gerrit URL: https://gerrit.gromacs.org/3311

Gerrit received a related patchset ‘1’ for Issue #1415.
Uploader: Teemu Murtola (teemu.murtola@gmail.com)
Change-Id: llfb2820019e2b41ef74cbaccd202364917e43d58
Gerrit URL: https://gerrit.gromacs.org/3318

Gerrit received a related patchset ‘1’ for Issue #1415.
Uploader: Teemu Murtola (teemu.murtola@gmail.com)
Change-Id: l6e4875b65b5d7c93e7a1455a2d31bd6d45bada3
Gerrit URL: https://gerrit.gromacs.org/3388

Gerrit received a related patchset ‘1’ for Issue #1415.
Uploader: Teemu Murtola (teemu.murtola@gmail.com)
Change-Id: l4cc3f00ec05f46dadb92512e532d8efb11da3694
Gerrit URL: https://gerrit.gromacs.org/3355

Gerrit received a related patchset ‘1’ for Issue #1415.
Uploader: Teemu Murtola (teemu.murtola@gmail.com)
Change-Id: lb3d78bce62a4c03c4d72001164921ed55657a2
Gerrit URL: https://gerrit.gromacs.org/3362
Gerrit received a related patchset '1' for Issue #1415.
Uploader: Teemu Murtola (teemu.murtola@gmail.com)
Change-Id: I7ece6eafa22139215510148807c5979f8950a885
Gerrit URL: https://gerrit.gromacs.org/3363

#10 - 04/16/2014 08:00 AM - Roland Schulz
I think we shouldn't have a new top level folder for each simulation method. This is getting quickly out of hand. We already have: imd, pulling, swap, essentialdynamics. Not as problematic (because the list probably wont grow quickly) but also quite long of a list is math related top-level folders (math, fft, linearalgebra, statistics). Would it make sense to make the last 3 subfolders?

I'm still a bit confused about the dependency checking scripts. What is the easiest approach to use them to know whether moving files into different (sub)folders will create cyclic dependencies?

#11 - 04/16/2014 09:12 AM - Teemu Murtola
In general, I agree that the current subdirectories are possibly too fine-grained. But it is easier to first create smaller units and then combine them, than to try to figure out in advance what kind of bigger units make most sense. So I think that the current approach is reasonable, but we should aim for a smaller amount of top-level folders once it is clear what can be combined and makes sense to combine. Folders with just one or a few files don't really make much sense.

Roland Schulz wrote:

> I think we shouldn't have a new top level folder for each simulation method. This is getting quickly out of hand. We already have: imd, pulling, swap, essentialdynamics.

I agree. My initial thought was to put all these special methods under a specialmd/ folder or similar.

> Not as problematic (because the list probably wont grow quickly) but also quite long of a list is math related top-level folders (math, fft, linearalgebra, statistics). Would it make sense to make the last 3 subfolders?

Possibly. fft/ and linearalgebra/ have a special purpose of wrapping external libraries, and as such, have some relatively complex internal logic. But it would be simpler if there could be dependencies in both directions. This would, e.g., allow putting the definition of real and rvec into math/.

I'm still a bit confused about the dependency checking scripts. What is the easiest approach to use them to know whether moving files into different (sub)folders will create cyclic dependencies?

Currently, you need to run make dep-graphs and view doxygen/depgrahs/module-deps.dot.png. It currently only shows folders one level below src/gromacs/, but that is possible to change (but requires some thinking to keep things clear, both in the graph and elsewhere). I've been thinking of adding a proper cycle detection into the script, but that is complicated by the currently existing cycles. I'll see whether that would still be possible.
#12 - 04/18/2014 04:21 PM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #1415.
Uploader: Teemu Murtola (teemu.murtola@gmail.com)
Change-Id: I2aa1890357377d0d14c05b3968e1b36e1023458f
Gerrit URL: https://gerrit.gromacs.org/3378

#13 - 04/20/2014 08:09 PM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #1415.
Uploader: Roland Schulz (roland@rschulz.eu)
Change-Id: I360ae1f220b6c51e90eddbbe136053fg2e1c0ad5b
Gerrit URL: https://gerrit.gromacs.org/3382

#14 - 04/20/2014 08:40 PM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #1415.
Uploader: Teemu Murtola (teemu.murtola@gmail.com)
Change-Id: Iffeafa4fac5c5e0c744cf6deecac4eedae2caf5
Gerrit URL: https://gerrit.gromacs.org/3383

#15 - 04/21/2014 02:59 AM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #1415.
Uploader: Roland Schulz (roland@rschulz.eu)
Change-Id: I9207a7b8de7f092bab890da128abb48a8d97b79
Gerrit URL: https://gerrit.gromacs.org/3384

#16 - 04/22/2014 06:17 AM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #1415.
Uploader: Teemu Murtola (teemu.murtola@gmail.com)
Change-Id: I31312b92cf9f5d5129f3d27be27afaf
Gerrit URL: https://gerrit.gromacs.org/3388

#17 - 05/20/2014 09:03 PM - Teemu Murtola
Roland Schulz wrote:

I'm still a bit confused about the dependency checking scripts. What is the easiest approach to use them to know whether moving files into different (sub)folders will create cyclic dependencies?

As a reference, https://gerrit.gromacs.org/#/c/3380/ now allows you to run make doc-check, and it will report any new cyclic dependencies that the changes create. Hopefully, existing cycles will not lead to too many confusing results with the algorithm that is currently used. Until it gets merged, you can just cherry-pick it on top of any change (it shouldn't conflict with much of anything). Just try not to push the cherry-picked version. ;)