

GROMACS - Feature #1837

Design of new table classes

10/03/2015 02:38 PM - David van der Spoel

Status:	Closed	
Priority:	High	
Assignee:	David van der Spoel	
Category:		
Target version:		
Difficulty:	uncategorized	
Description		
<p>In order to support potentials in (user) tables one or more C++ classes are needed that are</p> <ul style="list-style-type: none">a) self-containedb) testablec) efficient to evaluated) use not too much memory such as to flood the cache <p>a) The table modules must be simple with a minimum of dependencies and preferentially no knowledge of the content. Table classes must provide energy and force evaluation for reference purposes</p> <p>b) Both generation of tables and reading of tables must be tested</p> <p>c) The non-bonded kernel has high demands on efficiency. For this reason tables need to use aligned memory, and multiple tables may need to be combined into one (think Coulomb, Dispersion, Repulsion), again without the table class needing to know what it contains.</p> <p>d) Seeing that it is possible to have different tables for all atomtypes in a biomolecule, e.g. ~40 atomtypes, there may be 500-1000 tables, each containing 12 reals per point and around 1000 data points. In single precision that amounts to 12 Mb. More than level 1 cache but less than level 3 cache for most modern CPUs if I'm not mistaken.</p>		
Related issues:		
Related to GROMACS - Task #1852: Remove group scheme		New

History

#1 - 10/03/2015 02:39 PM - David van der Spoel

- Status changed from New to Closed

#2 - 10/03/2015 02:39 PM - David van der Spoel

- Status changed from Closed to Accepted

#3 - 10/03/2015 02:42 PM - David van der Spoel

And the tables have to be

e) accurate to a user specified tolerance when generated

e) We need at least QuadraticSpline tables (implemented in <https://gerrit.gromacs.org/#/c/4734/>) and CubicSpline tables (to be done).

#4 - 10/03/2015 04:29 PM - David van der Spoel

Make that 50 Mb for the amount of memory.

In addition we need to update the manual since there is no description of the quadratic splines.

#5 - 10/03/2015 06:10 PM - Gerrit Code Review Bot

Gerrit received a related patchset '105' for Issue [#1837](#).

Uploader: David van der Spoel (davidvanderspoel@gmail.com)

Change-Id: I41aaf1f7c7994888967be2f40f51d548b4e69169

Gerrit URL: <https://gerrit.gromacs.org/4734>

#6 - 10/03/2015 09:22 PM - David van der Spoel

Maybe it would be good anyway to split the math and the physics.

E.g. by having the quadratic spline and the cubic spline table classes as part of the math subdirectory (as Teemu suggested in the patch 4734) and another class combining them into a form that the kernels would like to have. In this manner we can separately check the spline interpolations and the physics.

#7 - 10/07/2015 06:48 PM - Mark Abraham

I suggest the client module at setup time

- specifies the functional form (e.g. inspects forcerec for what the user wants, and prepares callback functions that will do the right thing. David maybe has that organized already for Ewald? I have WIP code that does this for 1-4 interactions. I can probably do the same for e.g. walls (that requires user tables) with <https://gerrit.gromacs.org/5183> as a base.)
- calls the spline code with the above callbacks, and it handles the interpolations, filling new vectors
- passes those vectors to table-layout code that knows what various hardware kernels would like (simplest if we just do that for all possible hardware)
- coordinates matching tables to actual kernels (e.g. we construct an instance of TabulatedXXXXKernelRunner, whose run() method knows to pass the right flat C-style array in the right function argument to the actual kernel)

The combinatorial problem of a user who wants different tabulated functional forms for non-trivial numbers of distinct atomtype pairs is not really one we can solve well (or need to). But OpenMP scales increasingly better as the problem becomes memory bound :-).

#8 - 02/26/2016 11:55 AM - Mark Abraham

- *Related to Task #1852: Remove group scheme added*

#9 - 10/20/2016 10:13 AM - Erik Lindahl

- *Status changed from Accepted to Resolved*

New table class merged in <https://gerrit.gromacs.org/#/c/5661/>

This solves most points raised in the original description, including transparent SIMD support in the nonbonded kernels. Tables can also be initialized from a vector of data - how that is read from a file is a job for a separate module, though.

However, the memory usage depends on the accuracy requested by the user. It is difficult to combine automation of the table density calculation with guarantees about not using too much memory, not to mention that some force fields (Hello, OPLS) can use ~ 100 atom types in pathological cases. Using group-based tables for each individual atom type will always be a disaster for performance on modern CPUs.

#10 - 03/17/2018 12:31 AM - Mark Abraham

- *Status changed from Resolved to Closed*