

## GROMACS - Task #2336

### improve on relative tolerance tests for tabulated interaction

12/07/2017 03:51 AM - Roland Schulz

<b>Status:</b>	Closed
<b>Priority:</b>	Normal
<b>Assignee:</b>	Erik Lindahl
<b>Category:</b>	core library
<b>Target version:</b>	2019
<b>Difficulty:</b>	hard
<b>Description</b>	
Both ICC 17.4 and 18.1	
<pre>[ RUN      ] SplineTableTest/0.PmeCorrection ../src/gromacs/tables/tests/splinetable.cpp:157: Failure Failed Interpolation inconsistency for table PMECorr 1 function(s) in table, testing index 0 First failure at x = 3.5600000000000023 Function value when evaluating function &amp; derivative: 0.28089874184092334 Function value when evaluating only function:      0.28089874184092339  [ RUN      ] SplineTableTest/1.Sinc ../src/gromacs/tables/tests/splinetable.cpp:220: Failure Failed Failing comparison with function for table Sinc 1 function(s) in table, testing index 0 Test range is ( 0.10000000000000001 , 10 ) Tolerance                = 1e-10 First failure at      x = 7.72300000000000105 Reference function     = 0.1283742279838776 Test table function    = 0.1283742279838776 Reference derivative   = 0.000289162613504236 Test table derivative  = 0.00028916261362489458</pre>	
<b>Related issues:</b>	
Related to GROMACS - Task #2353: improve on relative tolerance for constructi...	<b>New</b>

#### Associated revisions

##### Revision 68113c4e - 12/19/2017 02:36 AM - Mark Abraham

Fix table tests and improve table construction

Since compilers are allowed to use different FMA constructs, we now allow the consistency check to deviate a few ulps.

For sinc and other extreme functions that oscillate, the scan over the definition range to locate the minimum quotient between the 1st and 4th derivative to set the table spacing exposes some delicate errors. Basically, it is not possible to have arbitrarily low relative errors for the derivative for a function that has large magnitude in the same place. For now we reduce the test interval for sinc(); this should anyway not be relevant for normal well-behaved MD functional forms.

Fixes #2336.

Change-Id: I5f999ae871ae21ddc5b59cf78ad8bd27fe2df622

##### Revision 80dd3f5b - 12/21/2017 10:54 AM - Mark Abraham

Update double-precision test configurations

These changes improve coverage of double precision, using more release mode, particularly with latest gcc and icc, and using 128-bit SIMD, which have been cases that were buggy recently. The other aspects of the configurations that have been modified have been non-critical. Where appropriate, brief rationales are recorded. This resolves an old TODO item in the post-submit matrix.

Fixed a sign mismatch in initializing an OpenCL variable that didn't need to be initialized.

Noted relevant new TODOs.

Refs #2300, #2325, #2326, #2334, #2335, #2336, #2337, #2338

Change-Id: I131fa1a6776d1e7809799c3f931a1fc8100fcdc9

## History

---

### #1 - 12/07/2017 03:53 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2336](#).  
Uploader: Roland Schulz ([roland.schulz@intel.com](mailto:roland.schulz@intel.com))  
Change-Id: gromacs~release-2018~I5f999ae871ae21ddc5b59cf78ad8bd27fe2df622  
Gerrit URL: <https://gerrit.gromacs.org/7302>

### #2 - 12/07/2017 06:04 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2336](#).  
Uploader: Mark Abraham ([mark.j.abraham@gmail.com](mailto:mark.j.abraham@gmail.com))  
Change-Id: gromacs~release-2018~I131fa1a6776d1e7809799c3f931a1fc8100fcdc9  
Gerrit URL: <https://gerrit.gromacs.org/7303>

### #3 - 12/11/2017 01:29 AM - Mark Abraham

gcc 7.1 on dev-purley01 for AVX512 in double and release mode fails in the second case the same way (different value for the derivative, though)

```
[ RUN      ] SplineTableTest/1.Sinc
../src/gromacs/tables/tests/splinetable.cpp:220: Failure
Failed
Failing comparison with function for table Sinc
1 function(s) in table, testing index 0
Test range is ( 0.100000000000000001 , 10 )
Tolerance          = 1e-10
First failure at   x = 7.72300000000000105
Reference function = 0.1283742279838776
Test table function = 0.1283742279838776
Reference derivative = 0.000289162613504236
Test table derivative = 0.00028916261360602399

[ FAILED ] SplineTableTest/1.Sinc, where TypeParam = gmx::CubicSplineTable (17 ms)
```

### #4 - 12/11/2017 02:13 AM - Mark Abraham

And same failure with icc 18 and gcc 7.2 in double and release mode with AVX2\_256 on my laptop (but not clang-4.0.1).

### #5 - 12/11/2017 06:42 AM - Mark Abraham

- Target version set to 2018-beta2

gcc 7.1 in double and release mode fails in the second case the same way (different value for the derivative, though)

```
[ RUN      ] SplineTableTest/1.Sinc
../src/gromacs/tables/tests/splinetable.cpp:220: Failure
Failed
Failing comparison with function for table Sinc
1 function(s) in table, testing index 0
Test range is ( 0.100000000000000001 , 10 )
Tolerance          = 1e-10
First failure at   x = 7.72300000000000105
Reference function = 0.1283742279838776
Test table function = 0.1283742279838776
Reference derivative = 0.000289162613504236
Test table derivative = 0.00028916261360602399

[ FAILED ] SplineTableTest/1.Sinc, where TypeParam = gmx::CubicSplineTable (17 ms)
```

**#6 - 12/11/2017 06:43 AM - Mark Abraham**

- Status changed from New to Fix uploaded

**#7 - 12/12/2017 12:41 AM - Erik Lindahl**

Fixed properly now, I think.

There were two issues:

1) As Roland noticed, the compiler might evaluate some FMA operations in different order, which can lead to very small differences depending on whether we evaluate both function & derivative or only one of them. I've updated this to use a test with a tolerance of only a few ulps instead of requesting exact match. However, we shouldn't push the consistency check as far as the normal table tolerance.

2) The sinc() problem was not related to the tests, but the code we use to determine table spacing from the smallest quotient of the first and 3rd/4th derivative. Since this oscillates more for sinc() than any normal MD function, it requires more point to find the true minimum (otherwise we use too few points in the table). Instead of just jacking up the number of points, I modified the screening to occur in two stages. Our typical MD functions are much smoother than sinc() and won't be affected, but it's a good example of how aggressive unit tests help us find latent issues in the code!

**#8 - 12/12/2017 08:50 AM - Mark Abraham**

- Target version changed from 2018-beta2 to 2018-beta3

**#9 - 12/12/2017 12:25 PM - Erik Lindahl**

- Assignee set to Erik Lindahl

- Difficulty hard added

- Difficulty deleted (uncategorized)

Apparently only fixed improperly :-)

The problem is still in the construction of tables, not the unit test. The core issue is that we use the quotient of the first and fourth derivative of a function to determine the minimum spacing so as to fulfill a relative tolerance. This works perfect for normal well-behaved MD functions, but for general functions we bump into

1) 4th derivatives of oscillating functions (sinc) are not very nice

2) All the zeros mean we have both infinite values, and ones very close to zero

3) For our requirement on the relative tolerance, I'm worried that we in some cases (for pathological functions) end up with **insanely** small spacing, and consequently too large tables. Basically, the finer we look, the larger the probability that we hit one such location.

So, I think I have to revisit that. I'll give it a bit of time today, but if it's too complex we'll just remove sinc as a test for now.

**#10 - 12/12/2017 12:47 PM - Mark Abraham**

sinc as a test is ok, but the range [0.1,3] won't hit these issues

**#11 - 12/15/2017 10:25 AM - Erik Lindahl**

The fix in gerrit should now be fine for release-2018.

There's nothing major wrong with the code, but the challenge is rather that it doesn't work to specify a relative accuracy for regions where the function or derivative might be zero. For all the normal functions we use as nonbonded potentials this won't be any issue. For master I'll have to come up with a slightly different way of either specifying or interpreting the requested tolerance when constructing tables.

**#12 - 12/19/2017 02:33 AM - Mark Abraham**

- Tracker changed from Bug to Task

- Subject changed from Intel compiler table test fails in double to improve on relative tolerance tests for tabulated interaction

- Category set to core library

- Status changed from Fix uploaded to Closed

- Target version changed from 2018-beta3 to 2019

- Affected version deleted (2018-beta1)

opened [#2353](#) for the future task

**#13 - 12/19/2017 02:34 AM - Mark Abraham**

- Related to Task #2353: improve on relative tolerance for constructing tables added