

## GROMACS - Bug #150

### Wrong Mega-Flops accounting for Coulomb [W3] in double precision

06/01/2007 01:35 PM - Mikhail empty

<b>Status:</b>	Closed	<b>Difficulty:</b>	uncategorized
<b>Priority:</b>	High		
<b>Assignee:</b>	Erik Lindahl		
<b>Category:</b>	mdrun		
<b>Target version:</b>	3.3.1		
<b>Affected version - extra info:</b>			
<b>Affected version:</b>			

#### Description

When running DPPC test (downloaded from [www.gromacs.org](http://www.gromacs.org)) in double precision GROMACS outputs wrong numbers in Mega-Flops Accounting region for Coulomb [W3].

For example, compiled with GNU compiler (3.4.6 & 4.1.0) outputs the following:

Computing:	M-Number	M-Flops	% of Flops
LJ	13782.627985	454826.723505	-0.0
Coulomb	11511.419253	310808.319831	-0.0
Coulomb [W3]	-36019137.110528	-2881530968.842240	100.1
Coulomb [W3-W3]	2307.047275	539849.062350	-0.0
Coulomb + LJ	6741.516862	256177.640756	-0.0
.....			

#### In single precision:

Computing:	M-Number	M-Flops	% of Flops
LJ	13783.368842	454851.171786	10.8
Coulomb	11508.531708	310730.356116	7.3
Coulomb [W3]	1476.843692	118147.495360	2.8
Coulomb [W3-W3]	2304.909673	539348.863482	12.7
Coulomb + LJ	6736.020809	255968.790742	6.1
.....			

Values in all other other lines are close to each other in single and double precision.

This issue is reproduced also with Intel 9.1 compiler.

icc/double:

Coulomb [W3]	0.097144	7.771520	0.0
Coulomb [W3-W3]	2305.530631	539494.167654	13.1

icc/single:

Coulomb [W3]	1475.951089	118076.087120	2.8
Coulomb [W3-W3]	2304.629050	539283.197700	12.7

It appears also in VILLIN and LYS/Cut tests (and everywhere Coulomb [W3] is used I suppose).

The issue is reproduced on Intel Xeon 5355 and Dual Core AMD Opteron Processor 285.

#### History

##### #1 - 06/07/2007 11:00 PM - David van der Spoel

I have reproduced the problem.

##### #2 - 09/07/2007 05:00 PM - David van der Spoel

Erik, it seems that the variables outeriter and inneriter are not passed correctly to the double precision version of the assemblyloops. That is, the single precision routines give reasonable values, whereas the double precision outeriter is 0 and the inneriter is large. The C-version of the double precision loops behaves as it should.

**#3 - 09/13/2007 07:34 AM - Erik Lindahl**

Fixed in head and release-3-3-patches. This was due to a stack pointer being restored too early in assembly. I've checked all related kernels too, but they seem to be unaffected. No effect on results.