Guide to running GROMACS

What is GROMACS?

GROMACS (**GRO**ningen **MA**chine for **C**hemical **S**imulations) is a molecular dynamics package primarily designed for simulations of proteins, lipids and nucleic acids. It was originally developed in the Biophysical Chemistry department of the University of Groningen, and is now maintained by contributors in universities and research centers across the world. GROMACS is one of the fastest and most popular molecular dynamics software packages available, and can run on GPUs as well as CPUs. It is free, open source released under the GNU General Public License. Starting from version 4.6, GROMACS is released under the GNU Lesser General Public License.

GROMACS Functionality

GROMACS supports all the usual algorithms one might expect from a modern molecular dynamics implementation, but there are also quite a few features that make it stand out:

- GROMACS provides extremely high performance compared to all other programs. A lot of algorithmic optimizations have been introduced in the code. The innermost loops are written in assembly with optimized kernels for most common CPU extensions such as SSE/SSE2/SSE4 and AVX.
- Hybrid-CPU/GPGPU support for efficient usage of modern GPU hardware.
- GROMACS is user-friendly, with topologies and parameter files written in clear text format.
- There is no scripting language – all programs use a simple interface with command line options for input and output files.
- GROMACS is written in C and can be run in parallel, using standard MPI communication. Hybrid-MPI/OpenMP implementation is able to push the scaling limits for small to medium systems.
- GROMACS contains several state-of-the-art algorithms that make it possible to extend the time steps in simulations significantly, and thereby further enhance performance without sacrificing accuracy or detail.
- GROMACS is Free Software, available under the GNU General Public License.
- GROMACS contains several state-of-the-art algorithms that make it possible to extend the time steps in simulations significantly, and thereby further enhance performance without sacrificing accuracy or detail.
- The package includes a fully automated topology builder for proteins, even multimeric structures. Building blocks are available for the 20 standard amino acid

Ewrop & Chymru: Buddsoddi yn eich dyfodol
Cronfa Datblygu Rhanbarthol Ewrop
Europe & Wales: Investing in your future
European Regional Development Fund
ERDF
Llywodraeth Cymru
Welsh Government

residues as well as some modified ones, the four nucleotide and four deoxinucleotide residues, several sugars and lipids, and some special groups like hemes and several small molecules.

- There is ongoing development to extend GROMACS with interfaces both to quantum chemistry and bioinformatics/databases.

Floating point operation precision

GROMACS can be installed using the single and double precisions. We would suggest the use of the double precision calculation if you are using a fine grained force field, but coarse grained representation can be used using single precision. Basically, in the single precision the calculation can be done faster, but the numerical precision is lower than the slower but higher precision double precision. This guide is confined to the double precision version of the code.

Step 1 - Log in

The example used in this guide is configured to run on the Swansea Sandy Bridge cluster. Connect to *login.hpcwales.co.uk* with your HPC Wales user credentials using your preferred method (e.g. PuTTY from a Windows machine or ssh from any Linux terminal), then **`ssh sw-sb-log-001`** to connect to the Swansea system.

The steps below involve typing commands (**`in bold font`**) in the terminal window.

Step 2 - Load a GROMACS module

A number of GROMACS binary packages are available. Note that in common with most other software packages on the system, these are built with the Intel compiler.

- List pre-installed GROMACS versions:

  ```
  module avail gromacs
  ```

  Load the MPI double precision version (4.6.1-mpi_d):

  ```
  module load gromacs/4.6.1-mpi_d
  ```

- Confirm the loaded modules. All dependencies are handled automatically via the module file:

  ```
  module list
  ```

Step 3 - Create a directory

From your home directory, create a directory to hold the GROMACS data:

```
cd ~
mkdir gromacs
```

Step 4 - Obtain a test case

A number of benchmark test cases are provided with the installation. Choose the **`d.dppc`** example at:

```
/app/chemistry/gromacs/4.6.1/sb/intel-13.0/intel-4.1/examples/d.dppc
```

Copy the two benchmark job scripts, *Gromacs_4.6.1_dppc-pme.SLURM.q* and *Gromacs_4.6.1_dppc-cutoff.SLURM.q*, to your user space:

```
cd ~/gromacs
cp -rip /app/chemistry/gromacs/4.6.1/sb/intel-13.0/intel-4.1/examples/d.dppc/*.q .
```

The benchmark runs using Version 4.6.1 of GROMACS are for a phospholipid membrane, consisting of 1024 dipalmitoylphosphatidylcholine (DPPC) lipids in a bilayer configuration with 23 water molecules per lipid, for a total of 121,856 atoms.

Two data sets – Cutoff and PME – are used. Cutoff uses a pair of cutoffs for the neighbour list and for the Coulomb forces, while PME uses the Particle-Mesh Ewald method for long-range electrostatics. Both test cases run 5,000 steps of 2 femtoseconds for a total of 10 picoseconds simulation time.

Step 5 - Submit a job

You are now ready to run these test cases with the supplied job scripts.

- From your working directory, submit the job using:

  ```
  sbatch Gromacs_4.6.1_dppc-pme.SLURM.q
  ```

- Check the job queue by: `squeue`

  The run comprises two steps, initially executing grompp, the GROMACS pre-processor, followed by the MPI run on 64 cores. When the latter commences, a file called *GROMACS.d.dppc-pme.output.n64.<Job_ID>.log* is created in the *gromacs* directory that holds the job output - many temporary files are generated and routed to the user's scratch directory, /scratch/$USER/*gromacs.d.dppc-pme.*<*Job_ID*>, created by the job (where <*Job_ID*> is the ID assigned by the queuing system).

- The job should take around one minute using 64 cores if the case runs successfully. The file *GROMACS.d.dppc-pme.output.n64.<Job_ID>.log* should contain all the output and point to successful completion of the job.

- Compare your job output with the output file,

  /app/chemistry/gromacs/4.6.1/sb/intel-13.0/intel-4.1/examples/d.dppc/GROMACS.d.dppc-pme.output.n64.log

Step 6 - More Test Cases

Two additional cases are provided at

```
/app/chemistry/gromacs/4.6.1/sb/intel-13.0/intel-4.1/examples/ion_channel
```

and

```
/app/chemistry/gromacs/4.6.1/sb/intel-13.0/intel-
```

```
4.1/examples/lignocellulose
```

Try running these using Step 4 and Step 5 described above for `d.dppc`. More test cases are distributed with GROMACS. They can be found at:

```
/app/chemistry/gromacs/4.6.1/sb/intel-13.0/intel-4.1/regressiontests-4.6
```

A PDF version of the latest (and earlier versions of the manual) are available at:

ftp://ftp.gromacs.org/pub/manual/manual-4.6.7.pdf

A variety of useful tutorials are also available at:

http://www.gromacs.org/Documentation/Tutorials

References

- Official GROMACS website: http://www.gromacs.org/
- A list of papers referencing the development of GROMACS include the following:
  - Berendsen, et al. (1995) Comp. Phys. Comm. **91**: 43-56. (DOI, Citations of this paper)
  - Lindahl, et al. (2001) J. Mol. Model. 7: 306-317. (DOI, Citations of this paper)
  - van der Spoel, et al. (2005) J. Comput. Chem. **26**: 1701-1718. (DOI, Citations of this paper)
  - Hess, et al. (2008) J. Chem. Theory Comput. **4**: 435-447. (DOI, Citations of this paper)
  - Pronk, et al. (2013) *Bioinformatics* **29** 845-854. (DOI, Citations of this paper)
- GROMACS User Documentation, http://www.gromacs.org/Documentation