Molecular Integration Simulation Toolkit - interfacing novel integrators with Molecular Dynamics codes

Molecular Integration Simulation Toolkit:
Interfacing novel integrators with Molecular Dynamics codes

Iain Bethune[1]*, Elena Breitmoser[1], Ben Leimkuhler[2]
*** ibethune@epcc.ed.ac.uk**
[1]EPCC, The University of Edinburgh [2]School of Mathematics, The University of Edinburgh

Background

Sampling the conformational space of complex macromolecules with Molecular Dynamics is hard! Despite significant effort in simulation software (e.g. GROMACS, NAMD, AMBER...) able to take advantage of hardware such as GPUs and parallel supercomputers, access to timescales of biological relevance – milliseconds to seconds – is still far from routine.

The ExTASY project proposes a set of **Ex**tensible **T**ools for **A**dvanced **S**ampling and anal**Y**sis which take a three-pronged approach to the sampling problem:

- *More sampling* by using massive 'ensemble' simulations. 100s – 1000s of loosely coupled MD simulations executed automatically on HPC systems.
- *Smarter sampling* by coupling simulation with analysis tools to determine which regions of conformational space are well-sampled and promote exploration of under-sampled regions.
- *Faster sampling* by better integration algorithms, coupling the latest numerical schemes to existing highly-optimised MD codes to increase the base simulation time tep, and incorporate collective variable data on-the-fly.

The Molecular Integration Simulation Toolkit (MIST) library tackles the last of these objectives. The library is designed to meet the need of two user groups - for developers it provides a simple but powerful API for developing and testing new integration schemes; for users a suite of integrators is available which may be coupled to codes including GROMACS and AMBER with negligible loss of performance. Thus the complexity barrier associated with 'production' MD codes is overcome and new algorithms can quickly get into the hands of users.

MIST Library Architecture

MIST is implemented as a C++ library which provides two key abstractions, each implemented as an abstract class:

- The *System* represents the current simulation state, and a simple API provides access to the properties of each particle. In addition, accessors are provided for global quantities such as the potential energy, and a single function call updates the forces on the particles given the current state of the system. Adapters are provided for each supported MD code.
- An *Integrator* has a `single method`, which integrates the system from time `t` to `t+dt` and is implemented using only the System API, so completely independent of any `particular MD` code.

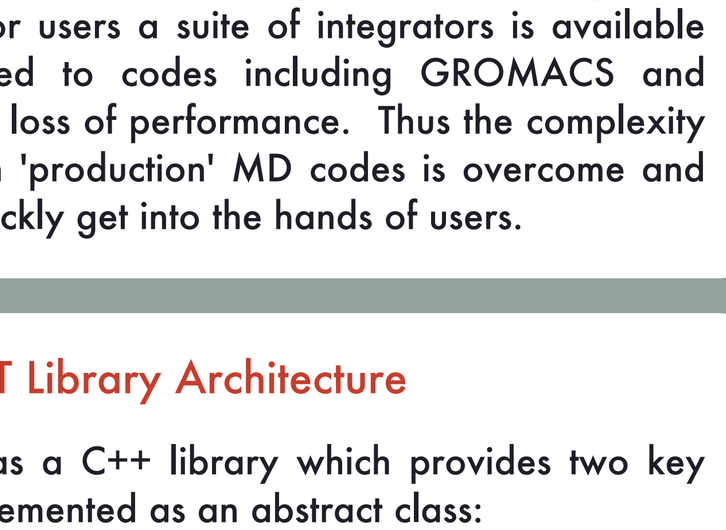

Figure 1: Schematic representation of MIST library structure

Performance Tests

MIST is designed to simplify integrator development without incurring a significant performance overhead. This is achieved by inserting MIST library calls into the 'host' MD code via source-code patches. To measure the impact of using MIST compared to standard GROMACS, we simulated a 50Å^3 box of water molecules using a flexible TIP3P model from the CHARMM forcefield for a total of over 11,000 atoms. We used Verlet integration and an NVE ensemble with a time step of 1 fs. The input files for this system can be found in the examples directory of the MIST distribution.
CPU performance is measured on ARCHER, a Cray XC30 with Intel Ivy Bridge processors, and GPU performance on Hydra, a machine with Intel Sandy Bridge processors and an Nvidia Tesla K20m GPU. When using CPUs only, MIST matches the performance of native Gromacs to within 2%, and when using the GPU, at worst the overhead of using MIST was around 5%.

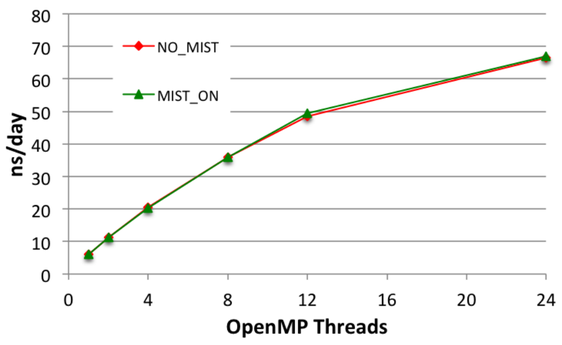

Figure 2: GROMACS performance on ARCHER using OpenMP

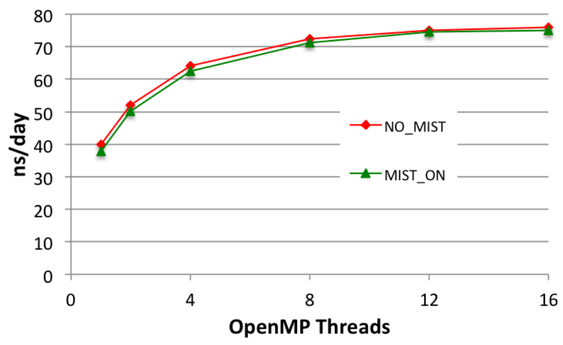

Figure 3: GROMACS performance on Hydra using CUDA + OpenMP

Download MIST

MIST is freely available (BSD Licence) from:

www.extasy-project.org/mist

The following features are currently available in the MIST library:

- AMBER, GROMACS and NAMD-Lite plugins (OpenMP only)
- Velocity Verlet, Verlet Leapfrog, Langevin Dynamics, Yoshida
- Continuous Tempering, Temperature Accelerated MD
- Access to individual force-field components
- Constraints (SHAKE/RATTLE and Symmetric Newton's method)

Please download and experiment with the code - we welcome feedback, contributions and suggestions!

The ExTASY project is funded by NSF and EPSRC [EP/K039490/1].
This work used the ARCHER UK National Supercomputing Service.

ExTASY
Extensible Tools for Advanced Sampling and Analysis

epcc

EPSRC
Engineering and Physical Sciences Research Council

The University of Edinburgh