Nano Simbox: An OpenCL-accelerated Framework for Interactive Molecular Dynamics

M. O'Connor*[abc], P. Tew[b], B. Sage[b], S. McIntosh-Smith[a], D. R. Glowacki[abcd]

[a]Department Of Computer Science, University Of Bristol, UK - [b]Pervasive Media Studio, Bristol, UK - [c]School Of Chemistry, University of Bristol, UK - [d]PULSE Institute, Stanford University,USA

Introduction

With increasing computational power and the maturation of new consumer grade human-computer interaction devices, interactive computational experiments are becoming feasible[1].
The Nano Simbox is an OpenCL-accelerated interactive molecular dynamics framework extending the award winning danceroom Spectroscopy [2], currently being used to provide engaging educational content. The framework shows the potential to become a scientific tool for exploring complex systems using human intuition and visualisation.

Theory and Algorithms

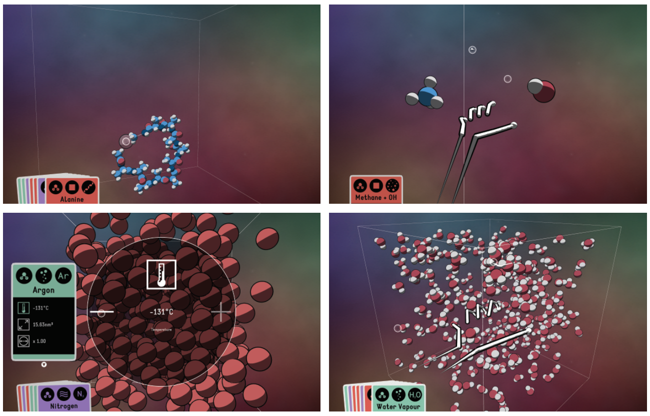
The Nano Simbox (NSB) carries out a molecular dynamics simulation involving N atoms. Every timestep the forces on the atoms are computed and integrated via a Velocity Verlet integration scheme.
In NSB, the forces acting on the particles can be characterised as:

$$F_{\mathrm{total}} = F_{\mathrm{int}} + F_{\mathrm{ext}}.$$

F_{int} is the internal force vector acting on the system, consisting of non-bonded forces between all atoms and bonded forces governing the interaction between atoms in a molecule.
F_{ext} is the external force vector acting on the system - the force that arises as a result of user interaction. Each interaction is projected into the system and represented as a Gaussian field centered at $\boldsymbol{r_j}$ with standard deviation σ in the simulation space. The interaction between N particles with and M user interaction points is then calculated as

$$F_{\mathrm{ext}} = \sum_{i=1}^{N}\sum_{j=1}^{M} c_i c_j \frac{(\boldsymbol{r_i} - \boldsymbol{r_j})}{\sqrt{2\pi}\sigma^3} e^{-\frac{(r_i - r_j)^2}{2\sigma^2}}$$

The term $\boldsymbol{r_i}$ refers to the position of particle i, while c_i and c_j refer to the particle and interaction point gradient scale factors, which can be governed by the user to allow different particles to have different interaction characteristics, and allow interactions to either attract or repel particles.
At each timestep, the above forces are computed and used to update particle velocities and positions, which are then rendered using a custom built game engine. To achieve a target frame rate of 60 FPS, we have investigated a range of multi-core parallelization strategies, including OpenCL on both CPUs and GPUs.



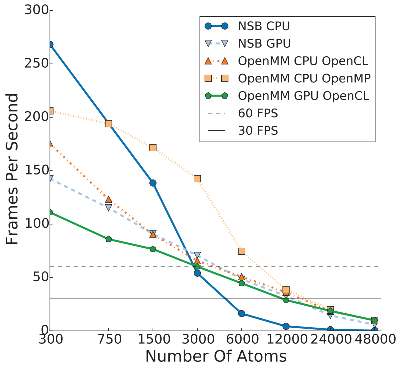
Clockwise from top-left: (1) The user folding a 10-alanine peptide using a touchscreen, (2) the user interactively guiding a CH4 + OH reaction with hand gestures using the LEAP Motion sensor, (3) the user adjusting the temperature on an argon box, (4) a waterbox of 270 water molecules.

The framework has been designed to be flexible so additional methods and features may be added. The current feature set includes:

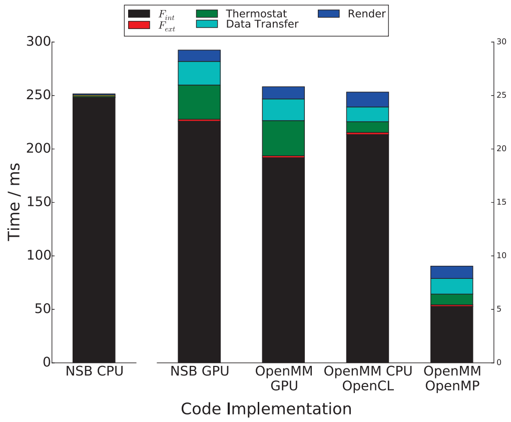
- OpenCL implementation of an MM3 parameterised forcefield with Lennard-Jones non-bonded, bonded and angle terms.
- Integration with the GPU-accelerated MD library OpenMM.
- OpenCL implementation of the Empirical Valence Bond (EVB) method, allowing reactive dynamics.

Performance and Scaling

OpenCL enables the framework to be used across devices from low-end laptops to high-end GPU workstations. The system was benchmarked on a workstation with an Intel i7 2.8 GHz quadcore CPU and an NVIDIA GTX Titan GPU, running an MM3 parameterised waterbox with one continuous user interaction.



Benchmark data showing frames per second obtained with increasing number of particles. CPU and GPU implementations of custom forcefields are compared against integrating with OpenMM on the GPU with OpenCL, CPU with OpenCL, and CPU with OpenMM.



Time per procedure in a single timestep on a waterbox with 6000 atoms.

The vast majority of time is spent in computing F_{int}, which is an $O(N^2)$ operation over all pairs of particles. The OpenCL parallelisation of our custom forcefields show a 10x speedup on the GPU compared to the CPU, but the OpenMM results indicate that there is scope for improvement on all platforms. Of particular note are the results using the OpenMM OpenMP implementation which indicate that a sufficiently optimised implementation will perform as well as a GPU implementation for the scale of simulations that can be run interactively.

Conclusions and Further Work

The Nano Simbox, a high performance framework for interactive molecular dynamics, is an exciting prototype combining high quality graphics, OpenCL accelerated molecular dynamics, and human computer interaction. Further work includes:

- Optimise forcefields for improved performance portability.
- Improve portability of the framework across operating systems.
- Utilise the OpenCL-OpenGL interop to improve performance
- Include additional interaction mechanisms and add stereoscopic rendering capabilities.

References

[1] J. D. Hirst, D. R. Glowacki, and M. Baaden, *Faraday discussions* **169**, 9 (2014).

[2] D. R. Glowacki, M. O'Connor, et al., *Faraday Discussions* **169**, 63 (2014).

nanosimbox.com

m.oconnor@bristol.ac.uk*