A GPU accelerated Lennard-Jones system for immersive molecular dynamics simulations in Virtual Reality

Nitesh Bhatia, Erich A. Müller, and Omar Matar

Imperial College London, South Kensington, London, SW7 2AZ, United Kingdom
{n.bhatia,e.muller,o.matar}@imperial.ac.uk
https://www.imperial.ac.uk/chemical-engineering

Abstract. Interactive tools and immersive technologies make teaching more engaging and complex concepts easier to comprehend. are designed to benefit training and education. Molecular Dynamics (MD) simulations numerically solve Newton's equations of motion for a given set of particles (atoms or molecules). Improvements in computational power and advances in virtual reality (VR) technologies and immersive platforms may in principle allow the visualization of the dynamics of molecular systems allowing the observer to experience first-hand elusive physical concepts such as vapour-liquid transitions, nucleation, solidification, diffusion, etc. Typical MD implementations involve a relatively large number of particles N = O(10^4) and the force models imply a pairwise calculation which scales, in case of a Lennard-Jones system, to the order of O(N^2) leading to a very large number of integration steps. Hence, modelling such a computational system over CPU along with a GPU intensive virtual reality rendering often limits the system size and also leads to a lower graphical refresh rate. In the model presented in this paper, we have leveraged GPU for both data-parallel MD computation and VR rendering thereby building a robust, fast, accurate and immersive simulation medium. We have generated statepoints with respect to the data of real substances such as CO2. In this system the phases of matter viz. solid liquid and gas, and their emergent phase transition can be interactively experienced using an intuitive control panel.

Keywords: virtual reality, gpu, molecular dynamics

1 Introduction

Molecular dynamics (MD) is fundamentally used for studying the physical movements, interactions of molecules and phase transitions between solid, liquid and vapour states (34; 13). These concepts can be reasonably easy to understand. However, the mathematical statements and the consequences of Newtonian principles for a system of interacting molecules, where intermolecular forces relying on their potential energies as well as overall system's temperature, volume and pressure are, hard to imagine and, often daunting to the student. E.g., in (33; 40), a 2D MD simulation system is

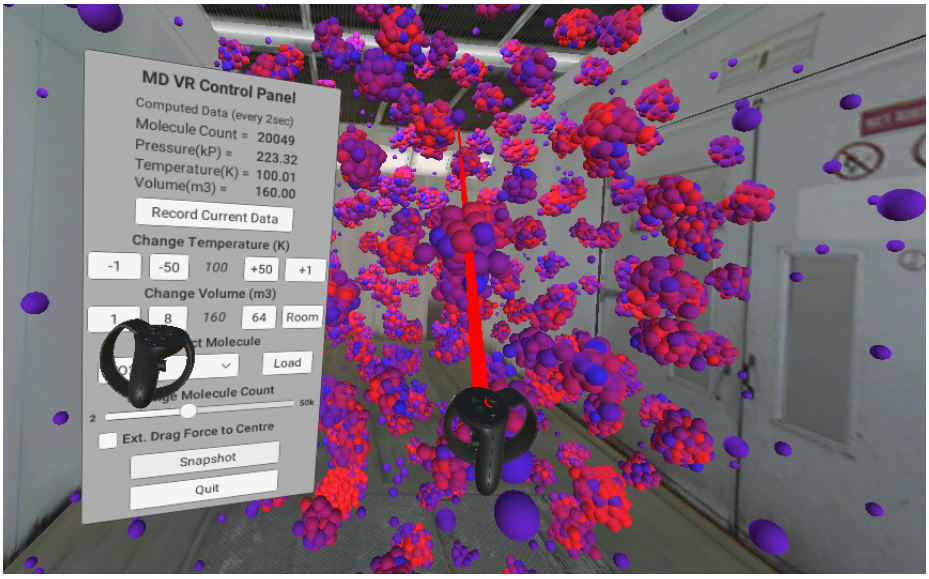


Fig. 1: Molecular Dynamics Virtual Reality (MD VR) Environment with 20,000 Lennard-Jones Molecules showcasing Liquid-Vapour coexistence. The colour of the molecules represents the magnitude of their velocity where blue represents low and red represents the high velocity.

presented, the former relying on Javascript and a browser and latter on Java3D. The students can develop an understanding of various MD phenomena by interactive controls and following a variety of student exercises. Educators have always explored new ways of teaching hard-to-grab MD concepts using simulations. With the advent of modern virtual and mixed reality devices such as Oculus Rift (29), HTC Vive (14) and Microsoft Hololens (26) and the improvements in computational hardware, virtual educational platforms can now offer potential benefits for training and education in a simulated setting (8). The 3D interactive medium provided by a well designed virtual reality (VR) system for pedagogy promotes inquiry-based learning which helps in better understanding of molecular dynamics and interactions compared to a didactic approach such as classroom teaching. This kind of approach has been reported in literature (39; 17; 42; 35). There are two main challenges. First, the research area is inherently cross-disciplinary and therefore requires deeper exchanges between virtual reality and computational molecular engineering. Second, real-time simulations in VR require low latency both in visualisation and computations that can be a bottleneck with any MD system. Typical MD implementations involve a relatively large number of particles (N = 10^4) and the force models imply a pairwise calculation which scales, in case of a Lennard-Jones system, to the order of O(N^2) leading to a very large number of integration steps (7). Hence, modelling such a computational system along with a graphical intensive virtual reality rendering could be limiting in terms of the system size which may lead to a lower graphical refresh rate.

The emergence of programmable general-purpose GPU architectures such as stream processing provides one way to exploit parallelisation and achieve performance boost (22). Hence, a fundamental difference in kind of computations best suitable for a GPU involves structural parallelisation compared to a CPU. By using high-level languages such as C and C++ and shader languages such as DirectX, HLSL, GLSL and Metal one can easily program any data-parallel algorithm over GPU (3). A holistic review of GPU programming has already been reported in literature (30). A general principle while programming on GPU to bake computational variables into graphics specific data structures like 2D and 3D textures (12; 21). Additionally, most modern GPUs are equipped with computational APIs and hardware to process data using shaders and shader cores (23). Using these technologies, several academic and commercial molecular dynamics software such as GROMACS, NAMD and AMBER have already been GPU-accelerated (2; 1; 31). A number of recent works have been reported on the implementation on GPU of particle systems immersed in force fields, interactions with rigid bodies or with other particles (19; 18; 9). Parallel processing, therefore, plays an important role in interactive simulation as it can first, push the performance of a molecular dynamics simulation and second, the shared-memory can be directly accessed in GPU for rendering graphics in VR thereby reducing the considerable bottleneck in data-exchange as seen in CPU centric visualisations. However, the enhancement in speed, scalability and program execution time can vary among GPU hardware, VR rendering requirements and the implemented algorithms. Table 1 shows a comparison of GPU execution cores and computing performance present in recent VR devices (as of 2019). In this paper, we have tried to address the above-mentioned challenges. Our focus is on the use of virtual reality (VR) in the context of simulating immersive molecular dynamics (MD) for the enhancement of education and interactive learning experience of students. The platform is aimed at educating about dynamics of molecular systems allowing the observer to experience first-hand elusive physical concepts such as vapour-liquid transitions, nucleation, solidification, and diffusion while surrounded by jiggling molecules. This MD VR simulation system is specifically developed for PC-VR that consists of a tracked headset, a pair of hand controllers and a PC equipped with a GPU. To derive an efficient mapping of MD with VR, we have leveraged a single GPU for both data-parallel MD computations and VR rendering thereby resulting in a robust, fast, accurate and immersive simulation medium with significant runtime savings. The scope of our work is on considering a generic MD system consisting of dense, uniform, single-component fluids, where the dynamics supported by classical mechanics. Complex systems such as molecular liquids, solvent and fluids, and the interfacial phenomena are not considered. The paper is organised as follows. In the next section (Section 2), we describe the methodology and the implementation details are covered in Sections 3, 4 and 5. Simulation results are discussed in Section 6. In Section 7, we have covered a comparative analysis of performance using two different GPUs. Section 8 concludes the paper with a summary of the results and the future work.

Table 1: Performance Comparison of GPUs present in PC-VR and Mobile-VR platforms in terms of core count and gigaflops (GFLOPS) compute performance.

Device	GPU	Cores	GFLOPS
Oculus Rift or HTC Vive (PC-VR)	Nvidia RTX2080	2944	8920
Oculus Rift or HTC Vive (PC-VR)	Nvidia GTX1080	2560	8228
Oculus Rift or HTC Vive (PC-VR)	Nvidia RTX2070	2304	6497
Oculus Rift or HTC Vive (PC-VR)	Nvidia GTX1070	1920	5783
Sony PlayStation VR	AMD Liverpool	1152	1843
Samsung GearVR Galaxy S10	Adreno 640	384	954
Microsoft Hololens 2	Adreno 630	256	727
Oculus Quest or Vive Focus	Adreno 540	256	567
Oculus Go	Adreno 530	256	407
Microsoft Hololens 1	Intel HD-Graphics	80	184

2 Methodology

Our goal is to simulate the classic N-body MD problem using GPU (32). To simulate such a system in VR perceivable at a human scale, a scaled representation of the molecules and the inter molecular forces is required. In the current system, we have considered the Lennard-Jones potential for representing forces between coarse-grained spherical molecules. In a Lennard-Jones system, the molecules attract each other at long range but strongly repel one another at they come closer. This system and equations are described in section 3. The primary objective for an efficient compute performance is that the MD VR simulation should run entirely on the GPU where the CPU mainly acting as a controller. We need to define positions, velocities and masses of the molecules on the GPU, update the positions according to intermolecular force influence, render the molecules in VR. The impulse on each molecule and the positions are computed on GPU with the help of a compute shader. Then using a GPU again, the molecules are rendered as spheres using a geometry shader. The compute and rendering methodology is further described in section 4. For designing an interactive MD simulation framework in VR for teaching and learning, we have considered several different ingredients such as a 3D VR environment and a control panel based intuitive user interface. The design functionality is further covered in section 5.

3 Scaled Lennard-Jones MD System for VR

3.1 Representation of Lennard-Jones MD System

The model consists of N spherical molecules interacting pair-wise as the system numerically resolves the step-by-step solution of the classical equations of motion. To build such a system following conditions are required to be satisfied:

Initial Condition An initial condition acts as a starting point to begin the force and motion computations for MD simulation in a mixture of N molecules. As a convenient

choice, the system is initialised with 5000 molecules at random positions in a predefined volume. The number of molecules and volume is kept modifiable by the student as described in Section 5.

Interaction Potential The forces acting on molecules derive from potential functions, representing the potential energy of the system for the specific geometric arrangement of the molecules. It is given by equation 1.

$$F(r_1, r_2, \ldots, r_n) = -\nabla U(r_1, r_2, \ldots, r_n) \tag{1}$$

Given any two molecules *i* and *j* separated by distance $r_{ij} = |r_i - r_j|$ the potential energy U is computed pairwise for all the molecules is given by equation 2

$$U(r_1, r_2, \ldots, r_n) = \sum_{i<j} u(r_{ij}) \tag{2}$$

One of the most famous and commonly used pair potentials for MD systems is the Lennard-Jones potential defined in equation 3, which was originally proposed for liquid argon (16; 27; 4). This pair potential is a sum of two inverse-power-law terms, a harshly repulsive term with exponent 12 and a softer, attractive (negative) term with exponent 6.

$$u(r) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] & r \leq r_c \\ 0 & otherwise \end{cases} \tag{3}$$

Based on Equation 1 the resulting inter-molecular forces arising from the Lennard-Jones potential (see, Equation 3) have the form as mentioned in equation 4 and shown in Figure 2 for CO2 molecule.

$$f(r) = \begin{cases} 48\frac{\epsilon}{r} \left[\left(\frac{\sigma}{r}\right)^{12} - 0.5 \left(\frac{\sigma}{r}\right)^{6} \right] & r \leq r_c \\ 0 & otherwise \end{cases} \tag{4}$$

System Evolution and Motion The evolution of the system over time can be followed by solving Newton laws of motion for all molecules in the MD system. The force is computed pairwise using equation 4 which results in molecular movement. The change in velocity and position is computed using verlet algorithm as shown in equation 5 and 6 (11).

$$v(\Delta t) = \frac{f(t)}{m} \Delta t \tag{5}$$

$$r(\Delta t) = v(t)\Delta t + \frac{f(t)}{2m} (\Delta t)^2 \tag{6}$$

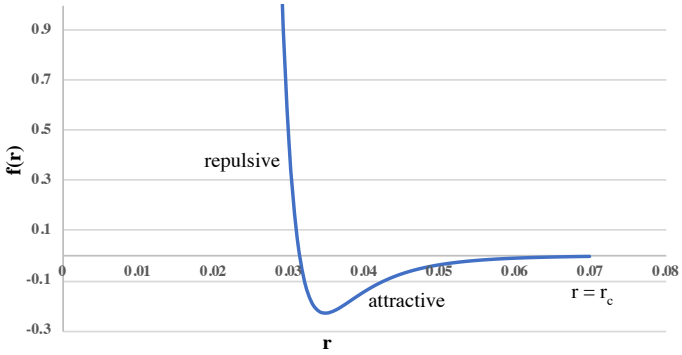


Fig. 2: Curve representing the change in Lennard-Jones force with inter molecular distance. The positive slope describes the attraction and the negative slope represents the repulsion between two molecules.

Energy Conservation and Thermodynamic Equilibrium Any system following the principle of Newtonian dynamics conserves the total energy E by maintaining a predefined temperature T for thermodynamic equilibrium (5). The conservation of the total energy E is given by equation 7 where E_{KE} represents instantaneous kinetic energy given by equation 8 and the relation with temperature T and pressure P are given in 9 and 10.

$$E(r_1, r_2, \ldots, r_n) = E_{KE}(r_1, r_2, \ldots, r_n) + U(r_1, r_2, \ldots, r_n); \tag{7}$$

$$E_{KE}(r_1, r_2, \ldots, r_n) = \frac{1}{2} m \sum_{i=1}^{N} v(r_i)^2 \tag{8}$$

$$T = \frac{2E_{KE}}{3k_B N} \tag{9}$$

$$P = \frac{Nk_B T}{V} + \frac{1}{3V} \sum_{i=1}^{N} r_i . f(r_i) \tag{10}$$

3.2 Scaling MD to Virtual World

In scientifically correct MD simulations the Δt time steps are of the order of 10^{-12} milliseconds (15). Hence, the systems require a good amount of computational power for getting accurate results. For instance, simulating 1000 molecules for the equivalent of one second requires about 10^{15} floating-point operations or 10^6 GFLOPS of computing power. Modelling such system over GPU would still take months of computation time. For rendering any VR content at desired 60 to 90 frames per second, the resulting Δt time steps are of milliseconds order resulting in a reduced requirement

of 10^{-3} GFLOPS (10). This is achieved by scaling standard parameters such as mass, length and energy. While still maintaining scientifically correct results, the scaling can reduce the necessity of a high compute power thereby opening doors for existing VR systems with good GPUs. In VR, making the nanoscale objects visibly large pose another challenge. Hence, visually correct scaling of physical quantities is required for accurate physical simulations. Therefore to represent the MD system in VR so to be manipulated approximately to scale in relationship with each other, we comprised on scaling the units of length, energy, mass and Boltzmann constant. In Table 2 the sample data is shown for carbon dioxide along with the scaling parameters.

Table 2: System of units and scaling factors used in MD VR simulations.

Physical Quantity	Units	CO2 Parameters	Scaling Factor	Scaled CO2 parameters for VR
Length	σ	2.8 x 10^{-10} m	10^{8}	0.028 m
Energy	ϵ	2.69 x 10^{-21} J	10^{18}	0.00269 J
Mass	m	7.30 x 10^{-26} Kg	10^{25}	0.73 Kg
Boltzmann Constant	kB	1.38 x 10^{-23} J/k	10^{18}	0.0000138 J/k

4 Computational and Rendering Methodology

The methodology implemented is based on three modules that include data initialisation using CPU, MD processing over GPU using compute shaders and graphics rendering using surface shaders. Communication and shared data between each stage are shown in Fig. 3. The VR system is programmed in Unity 3D and the corresponding scripts are coded using C# and DirectX11 supported shader languages (HLSL). The functionality of each stage is described in the following sections.

4.1 MD Data Structure and Initialisation

The most common coarse grain model for molecules are represented by spherical balls, with radii equal to their van der Waals radii and mass equal to their molar mass. (6; 25). In our model, the data structure of molecule consists of its position (as 3D Vector) in MD system, velocity (as 3D Vector), diameter (as float) and its molar mass (as float) as shown in Figure 3. The molecules with random positions and velocity are initialised in CPU as an array of Unity's Compute Buffer (named as MoleculeBuffer). That is data type is responsible for transferring data to and from the CPU's memory (RAM) to GPU's memory (VRAM) which can be accessed by Compute Shaders.

4.2 MD Computations using Compute Shaders

A compute shader in Unity is a program that is executed on the GPU regardless of the rendering pipeline (20). For data-parallel algorithms, it can dramatically improve

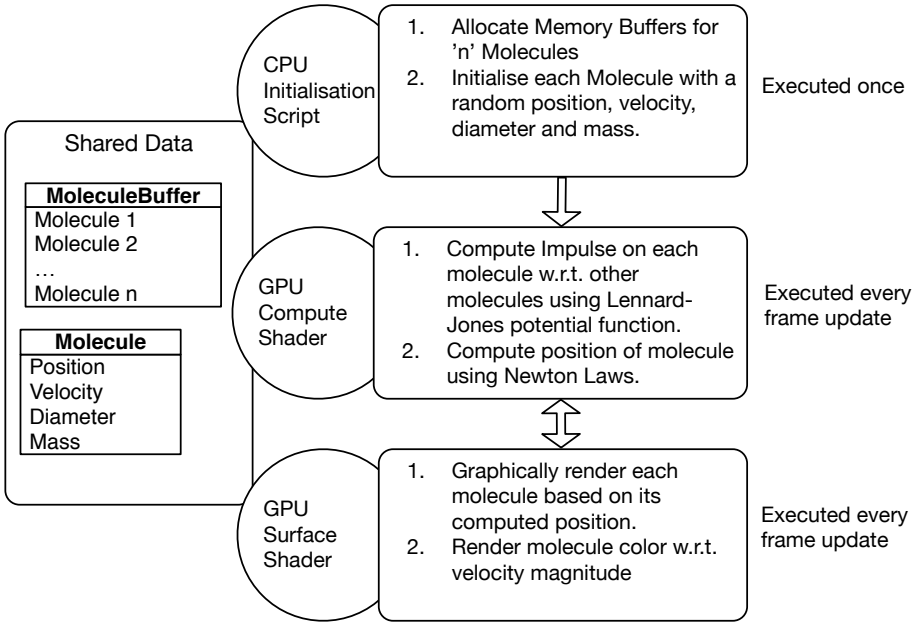


Fig. 3: Graphical representation of Compute and Render Pipeline used in MD VR System.

computational performance. For the MD computations, the compute shader consists of two kernels.

1. The first kernel calculates the total force on each molecule for every other molecule using equation 3. The Lennard-Jones potential exhibit strong directional forces however, at a cutoff distance, the forces become negligible and hence can be ignored. To reduce the number of pair-wise force computations we have considered a cut-off distance of 2.5 times diameter of the molecule (24).
2. The second kernel computes the position and updated velocity of the molecule using 5 and 6. The velocity is further scaled using 9 based on the temperature input selected by the user to maintain thermodynamic equilibrium. To eliminate surface effects periodic boundary conditions are used in MD (41). In our system, moving molecules effectively wrap around the Unity's boundary limits in such a way that a molecule leaving one face enters through the opposite face.

4.3 VR Rendering using Surface Shaders

The molecules are rendered as an instance of a single molecule modelled using spherical mesh object. A surface shader attached to the molecule runs in parallel on GPU. It locally reads the compute shader's data for each molecule and renders the mesh at its associated position. The colour of the molecule is defined over a range of velocity magnitude that varies from blue at low velocity to red at high velocity. We have

utilised Unity's indirect mesh rendering technique to further speed up the rendering (37).

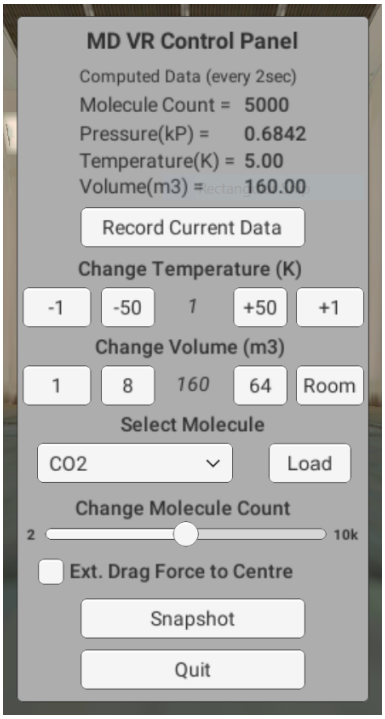
5 VR Environment and Controls

In this MD VR system, the molecules are rendered in a fixed volume. The volume is surrounded by a 3D model of a large room for an additional immersion. The 3D model is downloaded from Unity Asset Store (36). In the resulting simulation, the student standing inside the room can see the molecules floating in the air and interacting with each other based on MD model. A crucial part of any VR system is how the users may interact with the system. Oculus Rift comes with handheld remote controllers. These controllers allow interaction with the VR interface using buttons. The simulation parameters can be manipulated using a control panel as shown in Figure 4a. Using Unity's XR plug-in framework, the primary trigger is programmed to show the control interface attached to the left hand while a pointer to interact with the control panel appears on the right hand (38). The secondary trigger is utilised for user locomotion which is programmed to be restricted inside the room. The control interface consists of following features that can be interactively modified at runtime by the user.

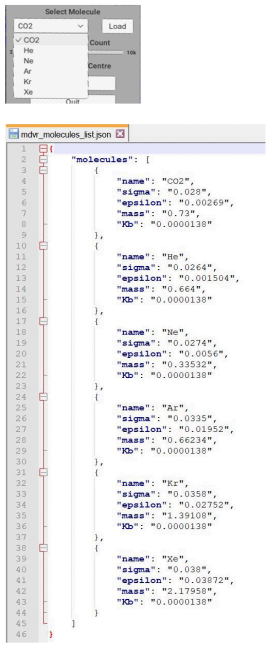
- **Simulation Data:** The control panel also displays the state of MD system updated every 2 seconds, in terms of number of molecules, computed pressure and temperature as well as the volume. A "Record Current Data" button is provided that stores the current simulation data in a text file for offline analysis.
- **Temperature:** For changing the system temperature (energy) four buttons are provided that change the temperature in the intervals of 1K and 50K.
- **Volume:** For changing the volume occupied by the molecules, four buttons are provided that change the volume to a cube of size 1 m^3, 8 m^3, 64 m^3 and the largest volume occupying the entire room of size 160 m^3.
- **Molecule Type:** The basic properties of a molecule such as a name, sigma, epsilon, mass and K_b is stored in as an editable JSON file as shown in Figure 4b. The program reads this JSON file and a populated drop-down list is presented to the student for selecting different molecules during VR simulation.
- **Molecule Count:** This slider interface helps in changing the number of molecules.
- **External Drag:** For collating the molecules toward the centre, a checkbox is provided that applies a uniform radial acceleration to all the molecules towards the centre of the volume.
- **Snapshot:** Using this button, the student can capture the active viewpoint as a PNG image similar to the one shown in Figure 1.
- **Quit:** This button is used for quitting the program.

6 Simulation Results

The primary objective of simulating a Lennard-Jones MD in VR is to have an interactive system to explore the effect of intermolecular forces concerning temperature and



(a) Control Panel User Interface



(b) (top) Molecule list as a drop-down and the molecule properties as described in a user editable JSON file (bottom).

Fig. 4: MD VR Control Panel Design.

density of molecules on the complete phase behaviour of a molecule such as Carbon di oxide (CO_2). Any ordinary matter is found in one of three phases: vapour, liquid or solid based on the temperature. The vapour (gas) phase exists at high temperatures whereas the solid phase is found at low temperatures with liquid phase in between. As shown in Figure 5 an example of gas phase where molecules overcome the intermolecular forces due to high velocity and continue to move with a kinetic energy that is proportional to the temperature.

It is possible to move continuously from the solid to the vapour phase or liquid to the vapour phase and vice versa by changing molecule density and temperature. Figure 6 shows an example of vapour to solid phase transition and the time development of kinetic and potential energies of CO_2 at low density and low temperature. As the temperature is lowered, the heat is taken out for maintaining thermodynamic equilibrium. This causes nucleation of molecules and overall the reduction of mean kinetic energy which results in transition of CO_2 from vapour to solid phase. This effect is also visible in high pressure. At the nucleation stage, the molecules being tightly packed

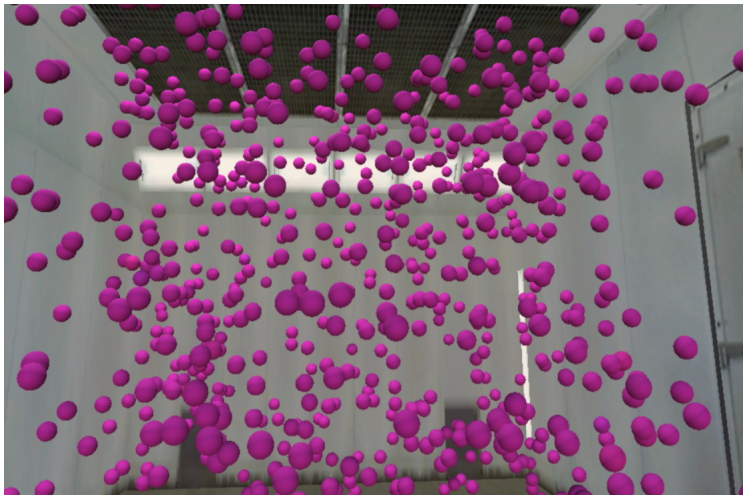
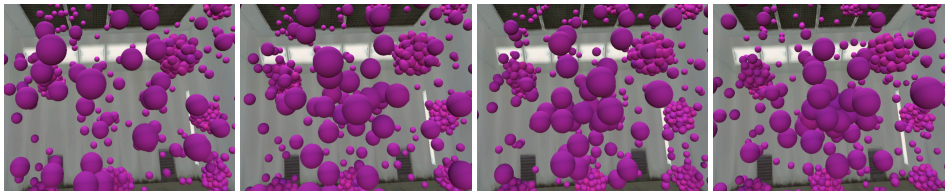


Fig. 5: Molecules in vapour phase at low pressure and high temperature.

possess a higher kinetic energy. This leads to transition of gas to liquid phase where the nucleated molecules form clusters of high potential energy molecules that move over time as shown in Figure 7. Figure 8 an example of solid phase. At low temperature, nucleated molecules form close packed clusters. In some cases, they naturally form a hexagonally close packed structures. The clusters move slowly and vibrate proportional to their kinetic energy.

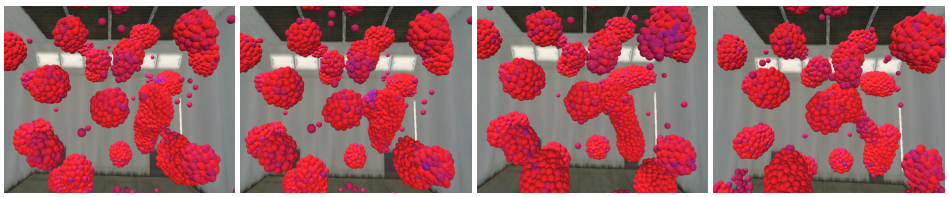


(a) Frame 1 (b) Frame 4 (c) Frame 7 (d) Frame 10

Fig. 6: Phase transition in molecules from vapour phase to the formation of solid clusters.

7 System Performance

We have tested the application on two machines viz. a Laptop and a Desktop both capable of rendering VR. We used HP Omen Laptop having Nvidia GTX1070 Max-Q



(a) Frame 1 (b) Frame 4 (c) Frame 8 (d) Frame 16

Fig. 7: Liquid phase and moving molecule clusters at high pressure and high temperature.

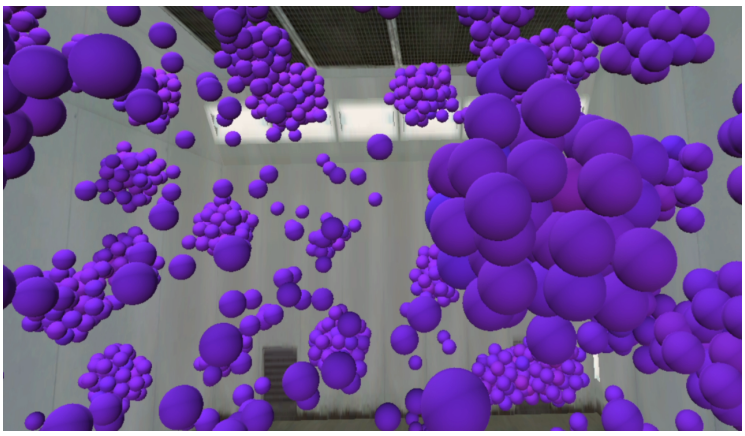


Fig. 8: Solid phase and formation of static clusters at low pressure and low temperature.

GPU and HP EliteDesk 800 G3 Desktop having Nvidia GTX 1080 GPU. The system details are shown in Appendix A. The results show a comparison of frame-rate achieved at steady-state for the number of molecules. When measuring the frame rates the orientation of Oculus HMD was kept facing the visualisation. In all the cases, FPS drops exponentially with increasing the number of molecules. We have also included the performance comparison of running the same application without VR.

On Oculus Rift, a frame rate above 75 frames per second (fps) is desired for best performance. On GTX1080 with VR, MD system with up to 25,000 molecules can be rendered with up to 90 fps while on GTX1070 Max-Q this limit reduces to 15,000 molecules due to lower compute power and thermal throttling. This limit is much higher when running the same simulations without VR, i.e. on-screen. Upon increasing the number of molecules, as the computation load on GPU increases the fps starts to drop. Note that in the case of Oculus Rift, the maximum frame rate VR gets restricted to 90 fps as an upper limit. Additionally, due to asynchronous spacewarp (ASW) feature in Oculus runtime, the software tries to compensate for frame drops by introducing additional frames which are evident with jittery fps in 9 between 45 fps to 90 fps. (28).

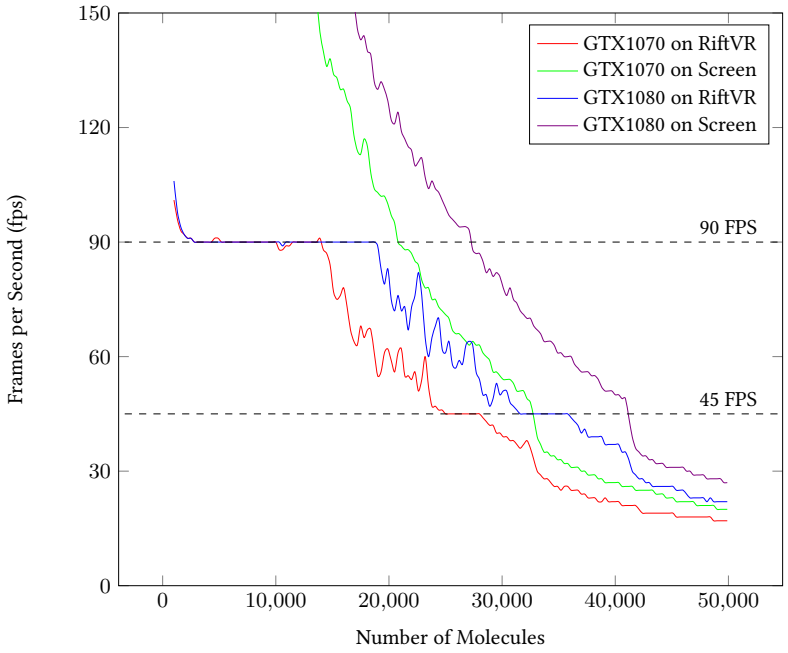


Fig. 9: FPS benchmarks using two GPUs and MD VR application running in VR and screen (non-VR) mode. The plots show a comparison of frame-rate achieved at steady-state for the number of molecules simulated in MD VR.

8 Conclusions

In this paper we described the design and implementation of data-parallel MD simulations using GPU, resulting in a robust, fast, accurate and immersive VR system. This modelling technique can compute pair-wise interactions of the order of O(N^2) and render also a large number of molecules maintaining a respectable display refresh rate. We plan to optimise this system to leverage more advanced shader programs for GPU resource utilisation. With the introduction of newer VR hardware, techniques such as foveated rendering can significantly unlock performance speedups. The current system is scalable from mobile GPU platforms to large scale GPU farms however, we need to test the scalability, performance and speed gains. Additionally, we plan to conduct student user-studies in future to demonstrate the effectiveness of this MD VR system in the context of teaching and learning.

9 Acknowledgements

Funding through Imperial College London Pedagogy Transformation programme is gratefully acknowledged.

Bibliography

[1] Abraham, M.J., Murtola, T., Schulz, R., Páll, S., Smith, J.C., Hess, B., Lindahl, E.: Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX **1**, 19–25 (2015)

[2] Anderson, J.A., Lorenz, C.D., Travesset, A.: General purpose molecular dynamics simulations fully implemented on graphics processing units. Journal of computational physics **227**(10), 5342–5359 (2008)

[3] Buck, I.: High level languages for gpus. In: SIGGRAPH Courses. p. 109 (2005)

[4] Buckingham, R.A.: The classical equation of state of gaseous helium, neon and argon. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences **168**(933), 264–283 (1938)

[5] Callen, H.B.: Thermodynamics and an introduction to thermostatistics (1998)

[6] Duncan, B.S., Olson, A.J.: Approximation and characterization of molecular surfaces. Biopolymers: Original Research on Biomolecules **33**(2), 219–229 (1993)

[7] Elsen, E., Vishal, V., Houston, M., Pande, V., Hanrahan, P., Darve, E.: N-body simulations on gpus. arXiv preprint arXiv:0706.3060 (2007)

[8] Freina, L., Ott, M.: A literature review on immersive virtual reality in education: state of the art and perspectives. The International Scientific Conference eLearning and Software for Education **1**(133), 10–1007 (2015)

[9] Georgii, J., Westermann, R.: Mass-spring systems on the gpu. Simulation modelling practice and theory **13**(8), 693–702 (2005)

[10] Germann, T.C., Kadau, K.: Trillion-atom molecular dynamics becomes a reality. International Journal of Modern Physics C **19**(09), 1315–1319 (2008)

[11] Grubmüller, H., Heller, H., Windemuth, A., Schulten, K.: Generalized verlet algorithm for efficient molecular dynamics simulations with long-range interactions. Molecular Simulation **6**(1-3), 121–142 (1991)

[12] Harris, M.: Mapping computational concepts to gpus. In: ACM SIGGRAPH 2005 Courses. p. 50. ACM (2005)

[13] Hirst, J.D., Glowacki, D.R., Baaden, M.: Molecular simulations and visualization: introduction and overview. Faraday discussions **169**, 9–22 (2014)

[14] HTC: HTC Vive product page. https://www.vive.com, accessed: 2019-09-20

[15] Jo, J.C., Kim, B.C.: Determination of proper time step for molecular dynamics simulation. Bulletin of the Korean Chemical Society **21**(4), 419–424 (2000)

[16] Johnson, J.K., Mueller, E.A., Gubbins, K.E.: Equation of state for lennard-jones chains. The Journal of Physical Chemistry **98**(25), 6413–6419 (1994)

[17] Jones, L.L., Jordan, K.D., Stillings, N.A.: Molecular visualization in chemistry education: the role of multidisciplinary collaboration. Chemistry Education Research and Practice **6**(3), 136–149 (2005)

[18] Kolb, A., Latta, L., Rezk-Salama, C.: Hardware-based simulation and collision detection for large particle systems. In: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware. pp. 123–131. ACM (2004)

[19] Kruger, J., Kipfer, P., Konclratieva, P., Westermann, R.: A particle system for interactive visualization of 3d flows. IEEE Transactions on visualization and computer graphics **11**(6), 744–756 (2005)
[20] Lammers, K.: Unity shaders and effects cookbook. Packt Publishing Ltd (2013)
[21] Luebke, D., Harris, M.: General-purpose computation on graphics hardware. In: Workshop, SIGGRAPH. vol. 33 (2004)
[22] Luebke, D., Harris, M., Govindaraju, N., Lefohn, A., Houston, M., Owens, J., Segal, M., Papakipos, M., Buck, I.: Gpgpu: general-purpose computation on graphics hardware. In: Proceedings of the 2006 ACM/IEEE conference on Supercomputing. p. 208. ACM (2006)
[23] Luebke, D., Humphreys, G.: How gpus work. Computer **40**(2), 96–100 (2007)
[24] Mecke, M., Winkelmann, J., Fischer, J.: Molecular dynamics simulation of the liquid–vapor interface: The lennard-jones fluid. The Journal of chemical physics **107**(21), 9264–9270 (1997)
[25] Mezey, P.G.: Shape in chemistry: an introduction to molecular shape and topology. Wiley-VCH (1993)
[26] Microsoft: Microsoft Hololens product page. https://www.microsoft.com/en-us/hololens, accessed: 2019-09-20
[27] Müller, E.A., Gubbins, K.E.: Molecular-based equations of state for associating fluids: A review of saft and related approaches. Industrial & engineering chemistry research **40**(10), 2193–2211 (2001)
[28] Oculus: Asynchronous Spacewarp blog post. https://developer.oculus.com/blog/asynchronous-spacewarp, accessed: 2019-09-20
[29] Oculus: Oculus Rift product page. https://www.oculus.com/rift, accessed: 2019-09-20
[30] Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A.E., Purcell, T.J.: A survey of general-purpose computation on graphics hardware. Computer graphics forum **26**(1), 80–113 (2007)
[31] Purawat, S., Ieong, P.U., Malmstrom, R.D., Chan, G.J., Yeung, A.K., Walker, R.C., Altintas, I., Amaro, R.E.: A kepler workflow tool for reproducible amber gpu molecular dynamics. Biophysical journal **112**(12), 2469–2474 (2017)
[32] Rapaport, D.: Molecular dynamics simulation. Computing in Science & Engineering **1**(1), 70–71 (1999)
[33] Schroeder, D.V.: Interactive molecular dynamics. American Journal of Physics **83**(3), 210–218 (2015)
[34] Steinfeld, J.I., Francisco, J.S., Hase, W.L.: Chemical kinetics and dynamics, vol. 3. Prentice Hall Englewood Cliffs (New Jersey) (1989)
[35] Stone, J.E., Gullingsrud, J., Schulten, K.: A system for interactive molecular dynamics simulation. In: Proceedings of the 2001 symposium on Interactive 3D graphics. pp. 191–194 (2001)
[36] Store, U.A.: Paint booth uaa business. https://assetstore.unity.com/packages/3d/environments/uaa-business-paint-booth-120410, accessed: 2019-09-20
[37] Technologies, U.: Graphics.DrawMeshInstancedIndirect function description. https://docs.unity3d.com/ScriptReference/Graphics.DrawMeshInstancedIndirect.html, accessed: 2019-09-20

[38] Technologies, U.: XR Plugin Architecture unity. https://docs.unity3d.com/Manual/XRPluginArchitecture.html, accessed: 2019-09-20
[39] Thorsteinsson, G., Shavinina, L.: Developing an understanding of the pedagogy of using a virtual reality learning environment (vrle) to support innovation education. The Routledge International Handbook of Innovation Education. Edited by LV Shavinina. Oxford: Routledge. ISBN-10 **415682215**, 456–470 (2013)
[40] Vormoor, O.: Quick and easy interactive molecular dynamics using java3d. Computing in Science & Engineering **3**(5), 98–104 (2001)
[41] Yeh, I.C., Hummer, G.: System-size dependence of diffusion coefficients and viscosities from molecular dynamics simulations with periodic boundary conditions. The Journal of Physical Chemistry B **108**(40), 15873–15879 (2004)
[42] Zyda, M.: From visual simulation to virtual reality to games. Computer **38**(9), 25–32 (2005)

A Appendix: VR System configurations used for performance computations

For teaching chemical engineering through virtual reality, we acquired several Oculus VR headsets and computers under pedagogy transformation funding. For computing the performance of MD VR system, we have used Oculus Rift VR Headset tethered to two different systems, one laptop class GPU and one desktop-class GPU. The hardware and software configurations are described below.

Hardware Configuration 1

- HP Omen 15t Laptop
- Oculus Rift with Controllers
- Intel Core i7 8750H Processor
- Nvidia Geforce GTX1070 Max-Q GPU
- 16 GB RAM

Hardware Configuration 2

- HP EliteDesk 800 G3 Desktop
- Oculus Rift with Controllers
- Intel Core i7 7700 Processor
- Nvidia Geforce GTX1080 GPU
- 64 GB RAM

Software Configuration

- Microsoft Windows 10 Build 1803
- DirectX 11 supported GPU drivers
- Oculus Runtime and SDK
- Unity 2018 with Education Licence
- Microsoft Visual Studio 2017 Community Edition