Ice Simulation Using GPGPU

Shadi Alawneh, Dennis Peters and Roelof C. Dragt‡
Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, NL, Canada
‡Faculty of Mechanical, Maritime and Material Engineering, Delft University of Technology, Delft, The Netherlands
{shadi.alawneh, dpeters}@mun.ca, sanderdragt@gmail.com

Abstract

Simulation of the behaviour of a ship operating in pack ice is a computationally intensive process to which General Purpose Computing on Graphical Processing Units (GPGPU) can be applied. In this work we present an efficient parallel implementation of such a simulator developed using the NVIDIA Compute Unified Device Architecture (CUDA). We have conducted an experiment to measure the relative performance of the parallel and serial versions of the simulator by running both versions on several different ice fields for several iterations to compare the performance. Our results show speed up of up to 55 times, reducing simulation time for a small ice field from over 40 minutes to about 45 seconds. Also, we have conducted another experiment to validate our numerical model of ship operations in 2D pack ice. Using a polypropylene vessel and floes, ship-floe and floe-floe interactions are modelled in a model basin and recorded on camera. The video is processed using Image Processing Techniques to track individual floes (and the vessel) to calculate their position and velocity over time. These results are compared with those of a numerical simulation using identical initial conditions.

1. Methodology

The particular problem that we are investigating is to simulate the behaviour of floating ice floes (pack ice, see Fig. 1) as they move under the influence of currents and wind and interact with land, ships and other structures, possibly breaking up in the process. In a two-dimensional model, we model the floes as convex polygons and perform a discrete time simulation of the behaviour of these objects. The goal of this work is to be able to simulate behaviour of ice fields sufficiently quickly to allow the results to be used for planning ice management activities, and so it is necessary to achieve many times faster than real-time simulation.

Figure 1: *Ice Floe[1]*

This project is structured in two components, the *Ice Simulation Engine*, which is the focus of this paper, and the *Ice Simulation Viewer*, which is being developed to display the data produced by the simulation engine. The simulation viewer displays frames of ice field data sequentially to provide its user with a video of a simulation of the field. It is currently used by the STePS^2 software team to help determine the validity of the data calculated by the simulation and will eventually be used to present results to project partners. The Ice Simulation Viewer is being developed in C++ using the Qt [2] user interface framework. Fig. 2 shows a screenshot of the main interface of the Ice Simulation Viewer with ice field loaded.

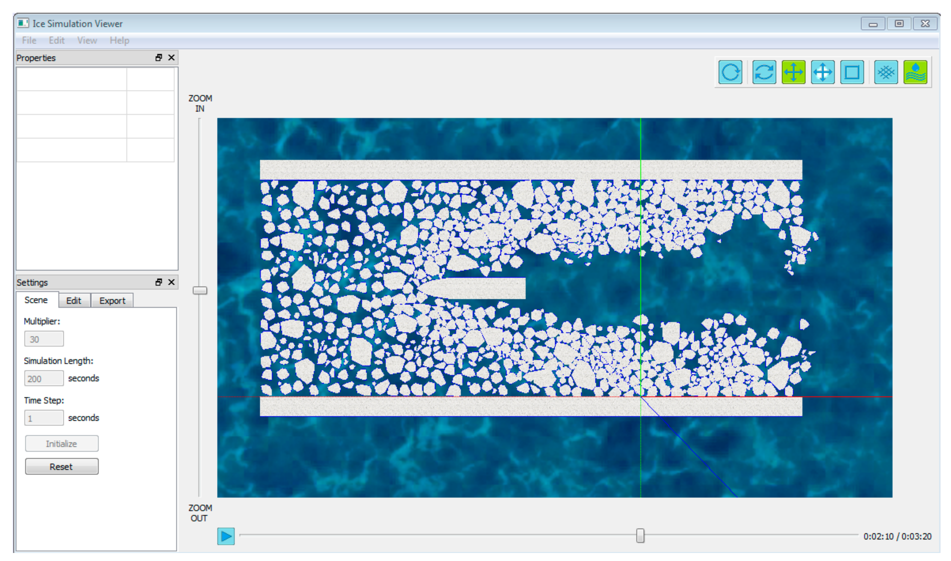

Figure 2: *Ice Simulation Viewer*

The method that we have used in the simulation is called Ice Event Mechanics Modeling (IEMM) which is a concept for rapid simulation of sea ice behavior and interaction mechanics. This method designed to take advantage of massively parallel computations that are possible using GPU hardware. The main idea of the method is to treat ice as a set of discrete objects with very simple properties, and to model the system mechanics mainly as a set of discrete contact and failure events. This method builds a system solution from a large set of discrete events occurring between a large set of discrete objects. The discrete events among the discrete objects are described with simple event equations (event solutions).

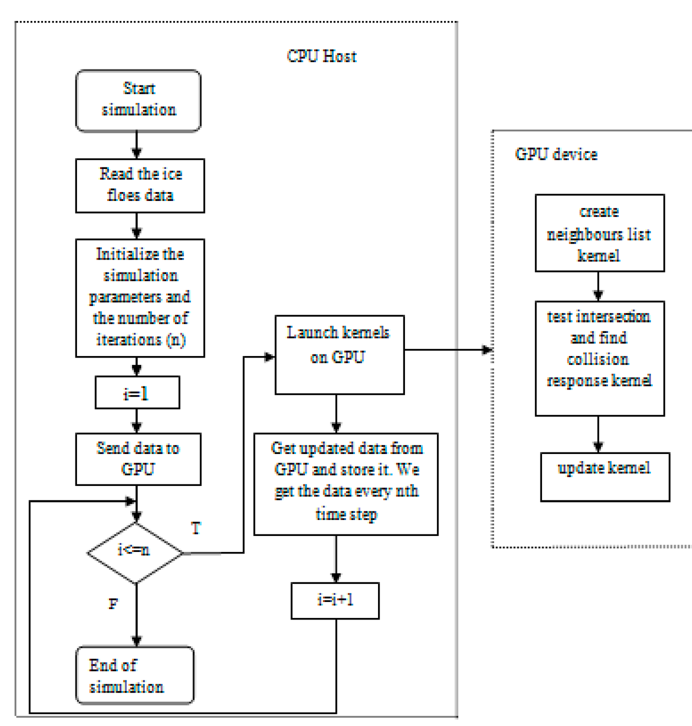

Figure 3: *Ice Simulator Framework*

2. Validation

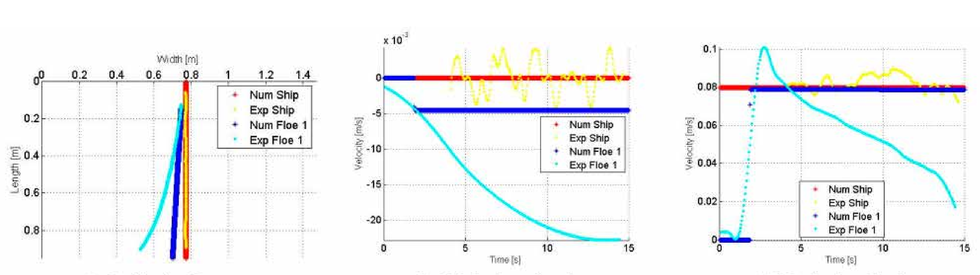

Figure 4: *Comparison between the numerical model (Num) and experimental data (Exp) of a one ship and one floe situation.*

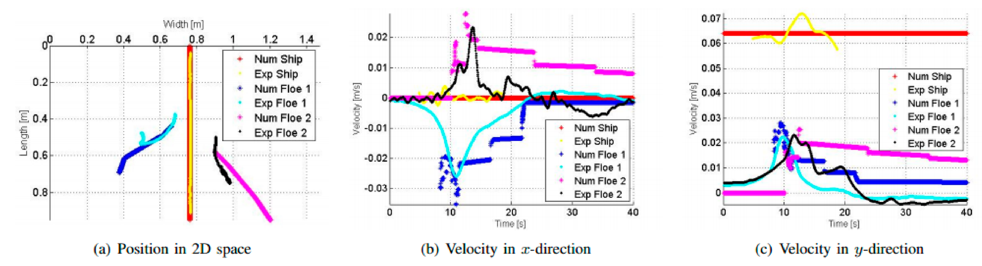

Figure 5: *Pack ice comparison, numerical model (Num) and experimental data (Exp).*

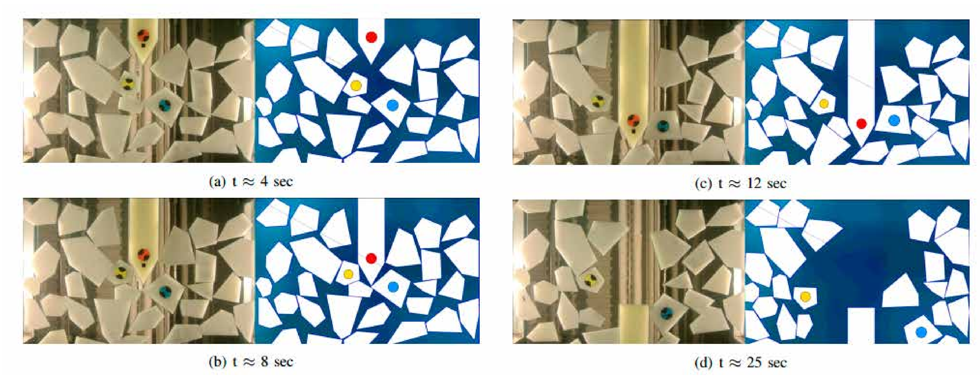

Figure 6: *Comparison between the numerical simulation and the experiments of a single case. The bodies in the numerical model are manually given coloured dots for convenience.*

3. Results

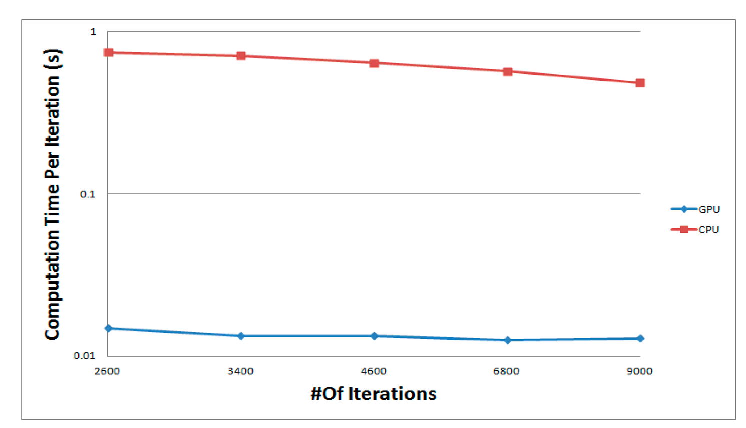

Figure 7: *Computation Time Per Iteration For The 456 Ice field.*

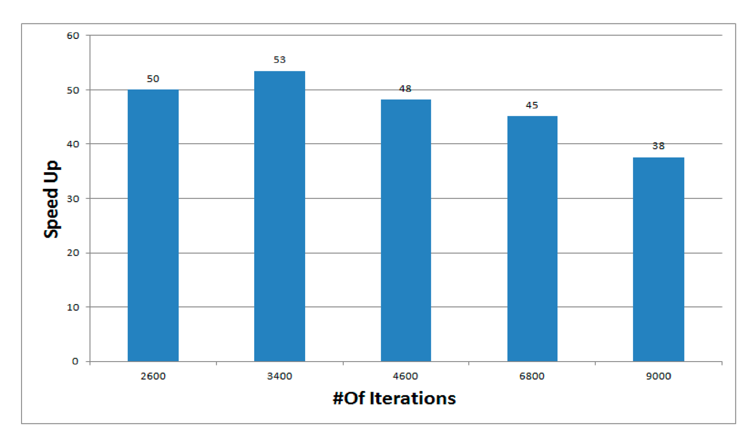

Figure 8: *Speed Up For The 456 Ice Field.*

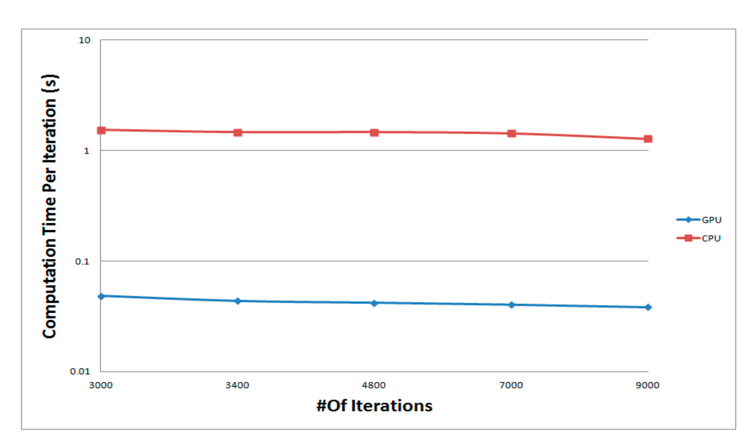

Figure 9: *Computation Time Per Iteration For The 824 Ice Field.*

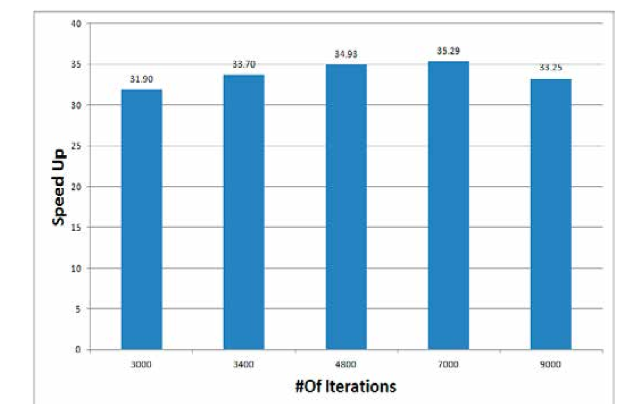

Figure 10: *Speed Up For The 824 Ice Field.*

We have used Intel(R) Xeon(R) CPU @2.27GHz (2 processors) and a GPU Tesla C2050. This card has 448 processor cores, 1.15 GHz processor core clock and 144 GB/sec memory bandwidth

4. Conclusion

The experiments proved performance benefits for simulating the complex mechanics of a ship operating in pack ice. It is clear that GPGPU has the potential of significantly improving the processing time of highly data parallel algorithms. The numerical model shows the general trends which are also visible in the experimental data. Especially in the pack ice scenario, it shows realistic behaviour. However, there are some points where the model needs improvement and where the data collected in this research can prove useful in improving the model.

5. Future Work

The physical models do not as yet model floe splitting, which are necessary for fully functional models and so adding this will be a next step. Also, while the results so far are promising, we have yet to reach the point where the simulation is fast enough to be practically used for planning ice management activities in realistic size ice fields. Further development and optimization are necessary to achieve this.

6. ACKNOWLEDGMENTS

This research has been done under STePS^2 project, under the leadership of Drs. Claude Daley and Bruce Colbourne, and was supported by: ABS, Atlantic Canada Opportunities Agency, BMT Fleet Technology, Husky Oil Operations Ltd, Research and Development Council, Newfoundland and Labrador and Samsung Heavy Industries.

References

[1] Haxon, "Ice floe at oslofjord," mar 2009, http://www.panoramio.com/photo/19618780.

[2] Jasmin Blanchette and Mark Summerfield, *C++ GUI Programming with Qt 4 (2nd Edition) (Prentice Hall Open Source Software Development Series)*, Prentice Hall, 2 edition, Feb. 2008.