

# GPU Accelerated Simulations

Popular Science Summary

Authors: Axel Goteman  
Vilhelm Roxling

Supervisors: Hidling Elmqvist (Dassault Systemes)

Michael Doggett (Dep of CS, LTH)

Examiner: Flavius Gruian

One of the key components in our modern society is the ability to simulate. By simulations, the industry can design new cars, phones, aircrafts etc., without having to go through prototype after prototype, allowing us the cheap and high-tech products most of us rely on in our day-to-day life. However, good as simulations are today there are still severe limitations on what can be simulated. Two of the largest limiting factors are how hard simulations are to design and how long they take to run. The first of these problems are tackled by the Modelica programming language, which is designed for easy set-up of simulations. We have attacked both these problems by using GPUs to speed-up Modelica simulations more than 5 times, and extending the capability of Modelica to handle collisions between objects.

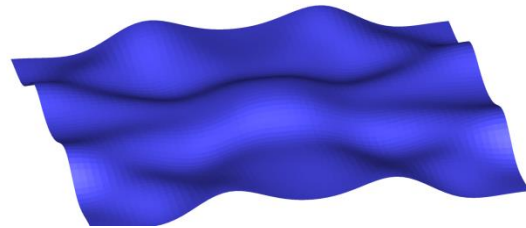
A modern GPU (graphics card) has more processing power, that is, can make more calculations per second, than a modern CPU. They are not used often however, as it is much harder to write programs that utilizes the capabilities of the GPU. This is because they are designed for effective processing of graphics, meaning doing the same operation for the thousands of pixels that makes up the screen. Any program that wishes to utilize the GPU needs to mimic this working method, i.e. doing thousands of identical operations, one for each element in a set.

## GPUs in Modelica

Modelica works by having the user set up a number of equations, giving the rules for what to simulate and how it should behave. Our GPU version extracts parts from these rules that have a structure which fits the GPU needs mentioned above. These parts are then separated from the rest of the simulation by being run on the GPU. The GPU and CPU then communicate necessary information during the simulation.

Typical problems where GPU usage is beneficial would be fluid/flow simulations. Wave simulation is such a problem, which can be useful in creation of for example wave power plants. When testing our prototype on such a problem, the simulation time decreased to less than a fifth

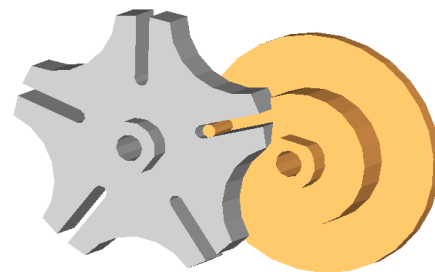
of the original simulation time.



Wave simulations done using our prototype

## Collisions in Modelica

One of the areas where Modelica has previously been lacking was collision simulations. Our prototype handles objects of any shape and size, by using an algorithm for contact reaction that isn't based on the shape of the object. Instead it allows very tiny penetrations between objects and uses this penetration to calculate the correct collision response. The prototype is also capable of handling anything from zero to tens of thousands of objects.



Our prototype simulating a part of a watch