DEVELOPMENT of a GPU-ACCELERATED MIKE 21 SOLVER for WATER WAVE DYNAMICS

Peter Edward Aackermann[1], Peter Juhler Dinesen Pedersen[1] ⋆,
Allan Peter Engsig-Karup[1], Thomas Clausen[2], and Jesper Grooss[2] ⋆⋆

[1] Technical University of Denmark (DTU), Department of Informatics and Mathematical Modelling, 2800 Kgs. Lyngby, Denmark
[2] DHI Group, 2970 Hoersholm, Denmark

Motivation

With encouragement by the company DHI are the aim of this B.Sc. thesis[3] to investigate, whether if it is possible to accelerate the simulation speed of DHIs commercial product MIKE 21 HD, by formulating a parallel solution scheme and implementing it to be executed on a CUDA-enabled GPU (massive parallel hardware).

MIKE 21 HD is a simulation tool, which simulates water wave dynamics in lakes, bays, coastal areas and seas and it is one of DHIs most applied commercial products. For this reason a drastic improvement in simulation speed has the potential to change the type of optimization problems where MIKE 21 HD is applicable and thereby open new market segments for DHI.

Model Equations and Discretization

MIKE 21 HD simulates water wave dynamics by solving a set of hyperbolic partial differential equations called shallow water equations which are given as

$$\frac{\partial \zeta}{dt} + \frac{\partial p}{\partial x} + \frac{\partial q}{\partial y} = \frac{\partial d}{\partial t} \tag{1}$$

$$\frac{\partial p}{\partial t} + \frac{\partial}{\partial x}\left(\frac{pp}{h}\right) + \frac{\partial}{\partial y}\left(\frac{pq}{h}\right) + gh\frac{\partial \zeta}{\partial x} + \frac{gp\sqrt{p^2+q^2}}{C^2h^2} = 0 \tag{2}$$

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial y}\left(\frac{q^2}{h}\right) + \frac{\partial}{\partial x}\left(\frac{pq}{h}\right) + gh\frac{\partial \zeta}{\partial y} + \frac{gq\sqrt{p^2+q^2}}{C^2h^2} = 0 \tag{3}$$

The solution scheme used is the Alternating Direction Implicit (ADI) method, which results in many tri-diagonal matrix systems, which have to be solved efficiently for each time step.

⋆ For contact the authors: s093066@student.dtu.dk and s093053@student.dtu.dk

⋆⋆ We want to thank Allan Peter Engsig-Karup, Thomas Clausen and Jesper Grooss for supervision and support throughout the project.

[3] B.Sc. thesis: http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6367

Solution Approach

Two different parallel solution schemes are implemented. The first ($S1$) solves each tri-diagonal system in parallel using a single CUDA thread for each system. This approach use the same tri-diagonal solution algorithm as MIKE 21 HD, the Thomas algorithm. The other solution schemes ($S2$) adds more parallelism into the system by using several threads to solve each system in parallel. In order to do this efficient are several parallel solution algorithms investigated. The focus have been on the Parallel Cyclic Reduction (PCR) algorithm and a hybrid algorithm of Cyclic Reduction (CR) and PCR.

Results

We discover that $S2$ are beneficial for small problems, while $S1$ yields better results for larger systems. We have obtained 42x and 80x speedup in double-precision for $S1$ and $S2$ respectively, compared to a representative sequential C implementation of MIKE 21 HD. For comparison can a 3072×3072 system be solved in double-precision on the GPU twice as fast as a 512×512 system on the CPU. Furthermore, the impact of switching to perform calculation in single-precision have been investigated. This resulted in 145x and 203x speedup for $S1$ and $S2$, respectively. We furthermore achieve near linear scaling when using method *S1* compared to a quadratic scaling on the CPU.

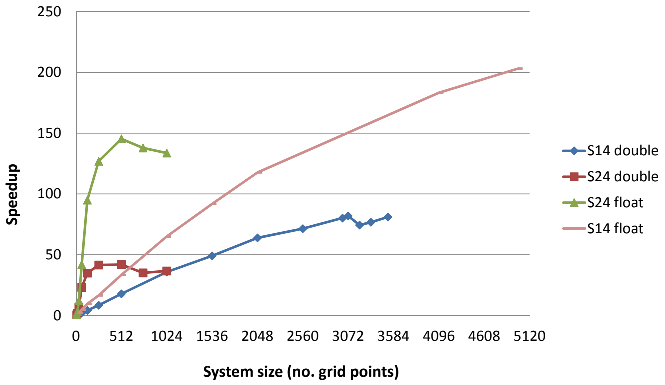

Fig. 1. Speedup of optimized S1 and S2 in single- and double-precision against an implemented CPU version. Result achieved on NVIDIA GeForce GTX 590.

Further research

An investigating of the precision impact of switching from double- to single-precision. Especially, using mixed-precision in the core math calculation is anticipated to be beneficial without losing to much precision.