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], Jesper Grooss[2]

[1]Technical University of Denmark (DTU), Kgs. Lyngby, Denmark, [2]DHI Group, Hoersholm, Denmark

Motivation

- In the pursuit of faster modelling tools to simulate water wave dynamics we investigate modern many-core computing architectures in order to accelerate the simulations by computing in parallel using a graphic card (GPU).
- A parallel solution scheme is investigated and developed. Designed to utilize the massively parallel processors with CUDA C.
- The simulation tool MIKE 21 HD is developed by the international company DHI and is one of their most utilized commercial products. Improvements in simulation speed will increase the amount of solvable problems and 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

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

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

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

- Solution scheme: Alternating Direction Implicit (ADI) method.
- Many tri-diagonal matrix systems have to be solved each time step.

Parallelization/Solution Approach

Two different parallel solution schemes are developed.

- $S1$: Single CUDA thread solving one tri-diagonal system.
  - Same solution algorithm as MIKE 21 HD (Thomas algorithm).
- $S2$: Several CUDA threads solving one tri-diagonal system.
  - Adds more parallelism.
  - Parallel solution algorithms investigated: PCR, CR-PCR.

Crucial Optimization

- Reduce global memory access.
- Access memory coalesced to fully utilize the available hardware.

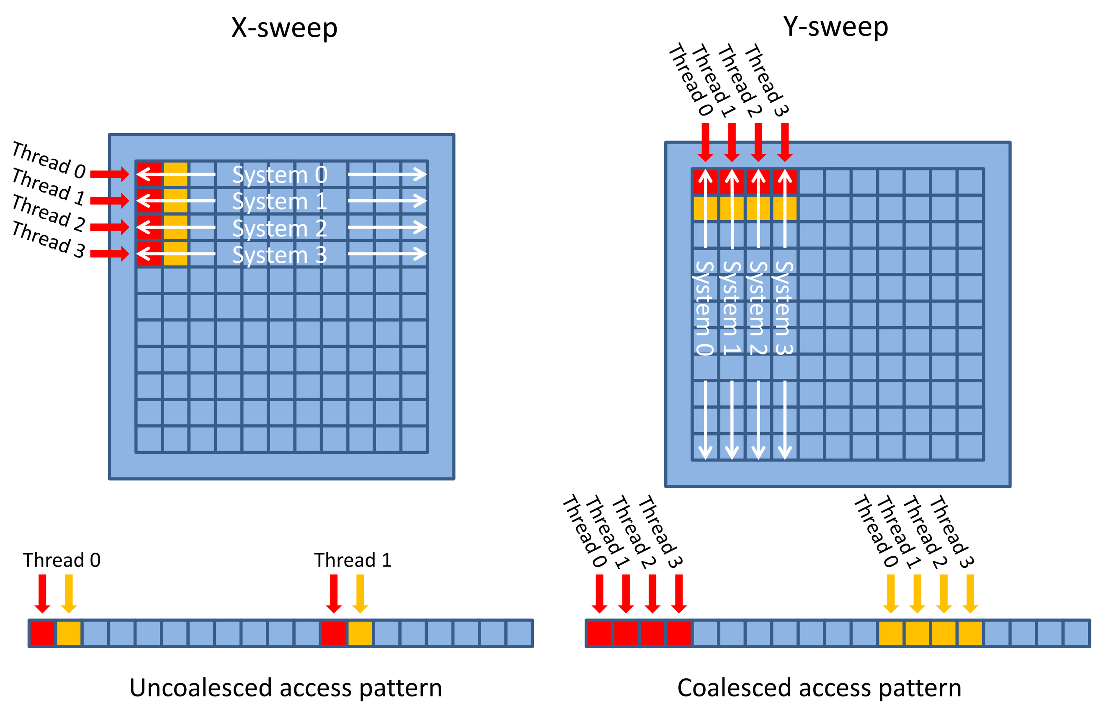
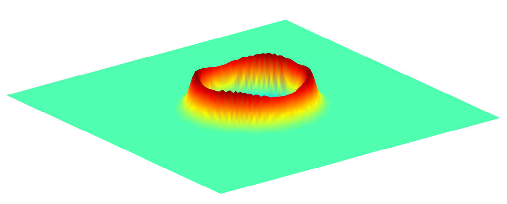
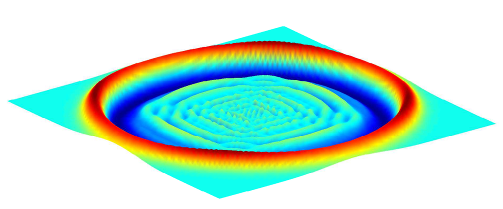


Figure 1: *Access pattern for $S1$ for an x- and y-sweep.*

- Determine core functionality and perform calculations cleverly by transposing arrays so the structure of the program is maintained.





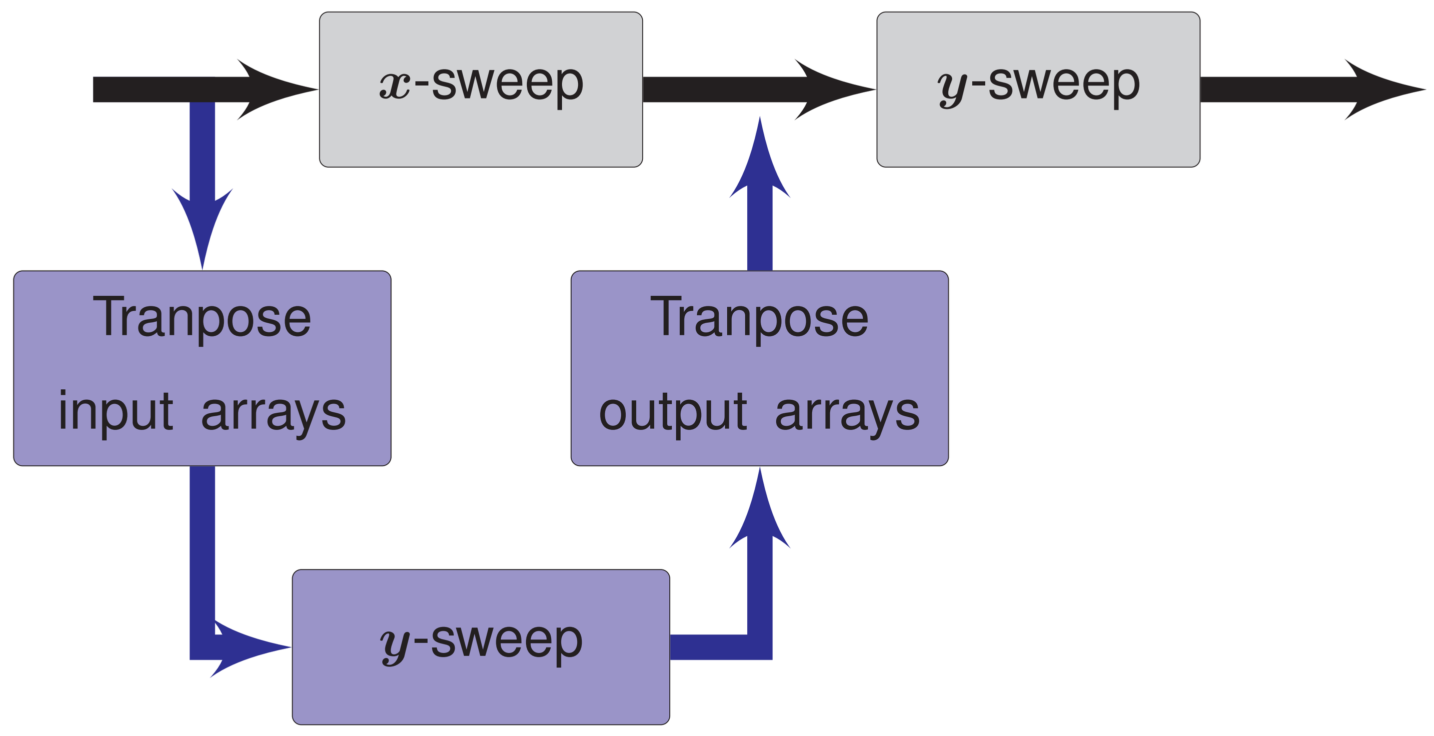


Figure 2: *Flowchart through one simulation time step in the $S1$ approach shown in gray and the modified x-sweep in blue.*

Results

- $S1$ beneficial for large systems.
  - Scaling $\mathcal{O}(n)$ against $\mathcal{O}(n^2)$ for the sequential implementation.
- $S2$ beneficial for small systems.

In double-precision

- Possible to solve systems in between 128×128 to 3584×3584
  - Minimum 35x speedup. *One hour simulation in less than 2 min.*
  - Maximum 82x speedup. *One hour simulation in less than 45 sec.*
- Solving a 3072×3072 system in double-precision on the GPU twice as fast as a 512×512 system on the CPU.
- Achieve exact same solution as MIKE 21 HD.

In single-precision

- Obtain 145x and 203x speedup for $S1$ and $S2$, respectively.
- Reduced precision compared to MIKE 21 HD.

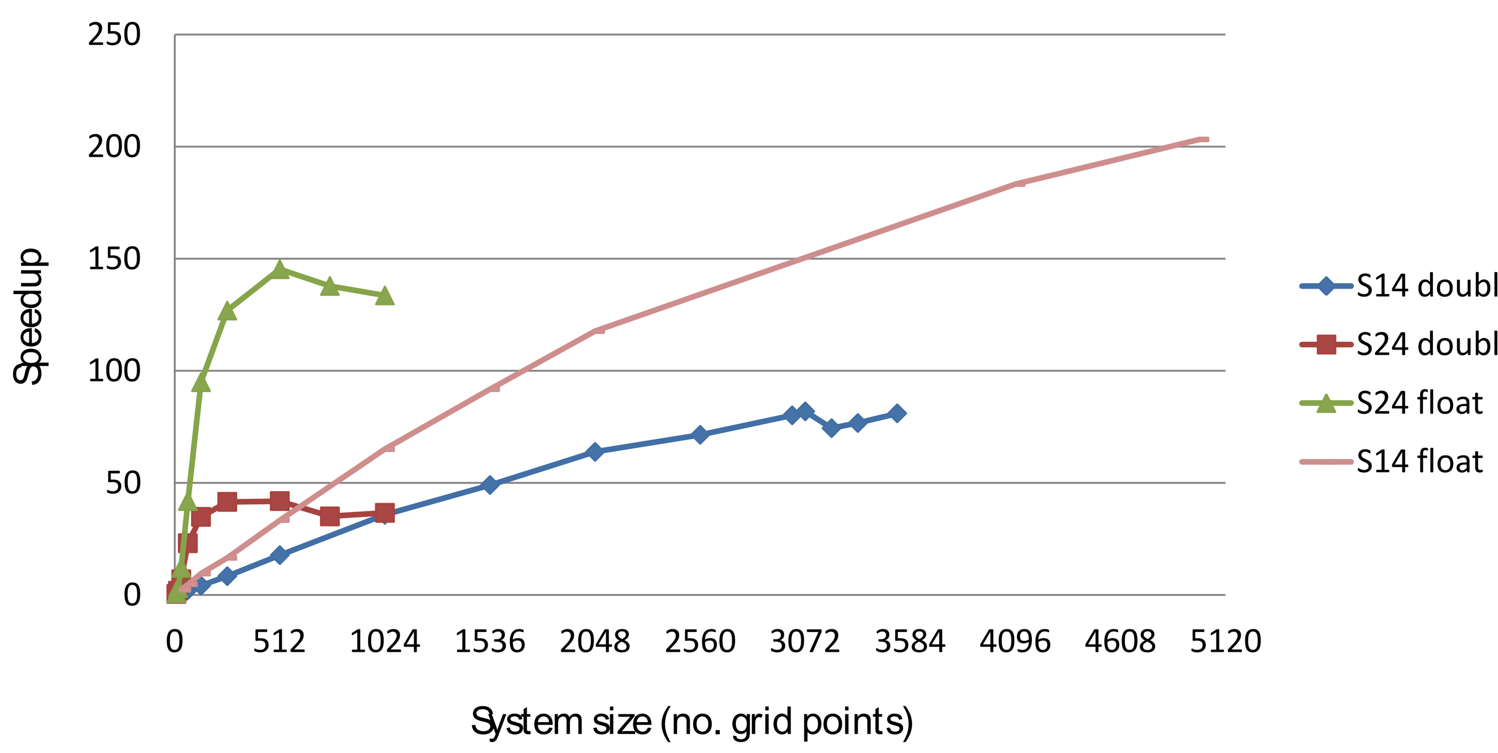


Figure 3: *Speedup of $S1$ and $S2$ in single- and double-precision compared to an corresponding CPU implementation. Executed on a NVIDIA GeForce GTX 590*

Conclusion and further research

- Accelerated MIKE 21 simulation speed dramatically by a formulated parallel solution scheme for execution on many-core architecture.
- Enabled DHI to solve larger or more detailed systems.
- Next step: Investigating the precision impact of double- vs. single-precision; using mixed-precision to core math calculation.