Higher-order ice-sheet modelling accelerated by multigrid on graphics cards

Christian F. Brædstrup and David L. Egholm
Department of Geoscience, Aarhus University
E-mail: Christian.fredborg@geo.au.dk

Research Goals

- Explore the potential of GPU acceleration
- Build fast and robust solvers for ice dynamics
- Integrate GPU solver into existing landscape creation framework

1. Introduction

The need for high-resolution ice simulations in exploring the dynamics of ice sheets and glaciers places a high demand on computational resources, particularly when full-Stokes or higher-order ice dynamical approximations are involved. To meet this demand, our research focuses on using specialized general purpose graphics processing units (GPGPU) for processing most of the numerical work during simulations.

Here we explore the performance of graphics cards when applied in higher-order ice sheet simulations. We use the depth-integrated iSOSIA model [Egholm, 2011] for solving a higher-order benchmark problem [Pattyn, 2008], and we measure the speedup provided by the GPGPU technology when combined with specialized GPGPU multigrid algorithms.

2. General purpose graphics processing unit (GPGPU)

Recent developments in graphics card hardware have sparked the interest of applying GPGPU-technology to a large set of computational problems. The GPGPU technology combines significant computational resources with a small form factor, bringing a new type of parallel supercomputing to ordinary desktop workstations.

The newest GPGPUs have more then 2500 low-frequency computing cores in one GPGPU-card, and modern laptops are often designed with more then 256 computing cores in their standard graphics cards. The resources are therefore already readily available.

However, the unique memory construction of graphics cards means that algorithms need to be specially designed for the hardware in order to run most efficiently.

3. FAS Multigrid

The Full Approximation Scheme (FAS) multigrid method offers a framework for writing robust and fast solvers. A typical solver contains: A smoother, a set of interpolation routines and a scheme for switching grid level.

The general idea is to smooth the errors of a given non-linear set of equations. The smoother is a Red-Black Gauss-Sidel algorithm in our solver. Each grid update is based on neighbour cell values, and smoothing errors with long wavelengths therefore requires many iteration to converge.

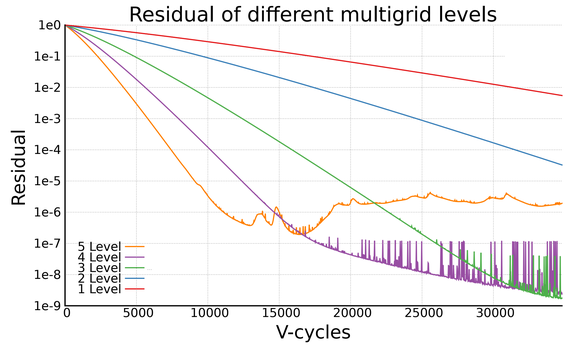


Figure 2: Residual of Multigrid. Test is [Pattyn, 2008] at 40 km.

To accelerate convergence the errors are therefore smoothed on several grids of varying resolution, and corrections are interpolated from coarse to fine grids. Using this scheme, long-wavelength residuals on fine grids become short-wavelength residuals on coarse grids.
The scheme is modular in the sense that different smoother, interpolation and cycle routines can be combined. Figure 5 shows a multigrid V- and W- cycle. Residuals are transferred to coarse grids and corrections are transferred back to the fine grids. The V-cycle is the most common multigrid cycle but many alternatives .

4. Numerical experiments

There are several important questions when testing the GPGPU multigrid solver:

- Is the multigrid (MG) component accelerating convergence?
- Is the solver able to scale to large grid sizes?
- Is the GPGPU faster then the CPU?

To answer these questions Test A from [Pattyn, 2008] has been used as a test experiment (See figure 1). This test generally requires many iteration before the non-linear higher-order stress components converge.

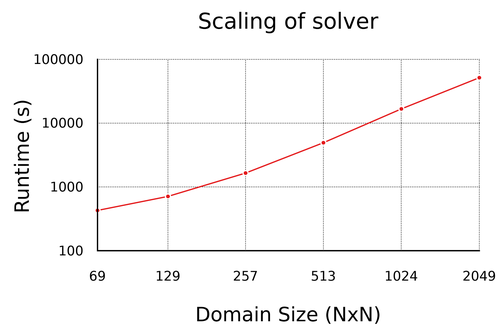


Figure 3: Scaling of the GPU Solver at different grid resolutions. Using a 80 km test.

Figure 2 shows how the multigrid component accelerates the convergence by reducing the iteration count. Without multigrid (1 level) the test converges (defined as a residual below $1e^{-3}$) after ~50000 V-cycles, whereas the multigrid algorithm using 5 grid levels convergences after ~7500 V-cycles.

Figure 3 shows the scaling of the solver to very large grids. The Tesla M2070 series of GPU's hold 6 GB of on-board RAM which currently allows the solver grids upto 2049x2049 and beyond. The current max of the solver is 4097x0497 (not shown). With the rapid development of GPU's and price of memory this limit will properly only increase with time.

Figure 4 shows the overall speedup produced by the GPU-multigrid algorithm. The figure shows the runtime difference between 1 CPU core running without multigrid (1 grid level) and a Tesla M2070 GPU operating with and without multigrid. Without multigrid (red curve), the GPU solver shows nearly a 100 times speedup on large grids, whereas the addition of multigrid increases the speedup factor to almost 250 for 5 grid levels.

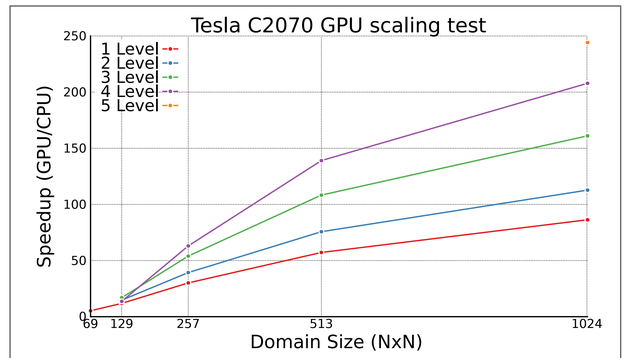
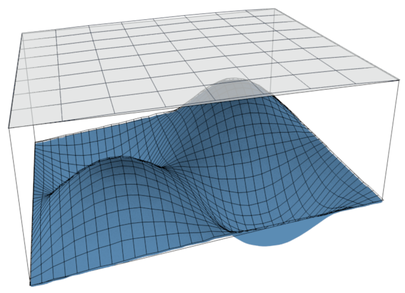
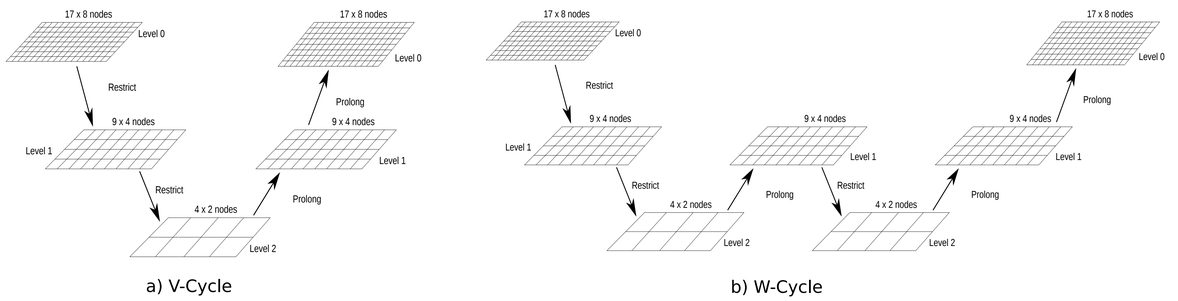


Figure 4: Speedup of GPU vs. CPU version. The speedup is based on Test A [Pattyn, 2008] using the 80 km test.



Figur 1: Overview of the Test setup. Published in [Pattyn, 2008].



Figur 5: Different examples of Multigrid cycles.

5. Conclusion

Many researchers in the glaciological community need fast and robust higher-order ice flow solvers.
Using advances in high-performance computing, our research shows that GPU's may provide a possible route towards reduced compute times and increased access to high-resolution models of glaciers and ice sheets without the need for very expensive supercomputers.

Future work

- Integrate solver with hydrology, erosion and sliding routines.
- Experiment with different multigrid routines.
- Test solver against realistic landscapes.

References:
[Pattyn, 2008] - Pattyn, Frank, et al. "Benchmark experiments for higher-order and full Stokes ice sheet models (ISMIP-HOM)." The Cryosphere Discussions 2.1 (2008): 111-151.
[Egholm, 2011] - Egholm, David L., et al. "Modeling the flow of glaciers in steep terrains: The integrated second-order shallow ice approximation (iSOSIA)." Journal of Geophysical Research: Earth Surface (2003–2012) 116.F2 (2011).