DTU Informatics
Technical University of Denmark

Smoke Simulation in Fire Engineering using CUDA

Stefan Lemvig Glimberg, PhD-student, DTU Informatics
slgl@imm.dtu.dk

INTRODUCTION

We present a Computational Fluid Dynamics (CFD) solver, for smoke propagation, using the parallel architecture of graphics hardware. The purpose is to investigate the possibilities of fast approximation techniques in combination with the powers of GPUs, and test usability from an engineering point of view. CFD tools for smoke propagation help engineers analyze security risks at fire scenes. Based on several case studies, our solver has shown to be an efficient tool for interactive smoke simulation. Even at high resolutions the solver performs at interactive rates on a single GPU. The parallel architecture of the GPU makes it an excellent computational unit for compute intensive tasks like CFD.

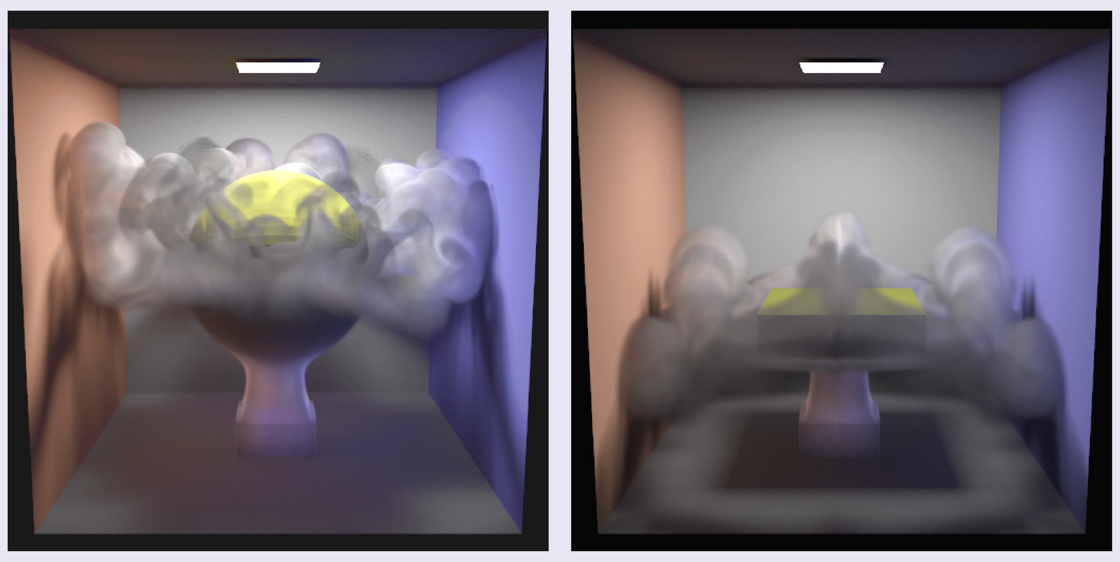


Figure 1. Two test setup examples with a static smoke source at the bottom middle of the room. The smoke propagates in curls around the obstacles placed above the smoke source.

METHOD

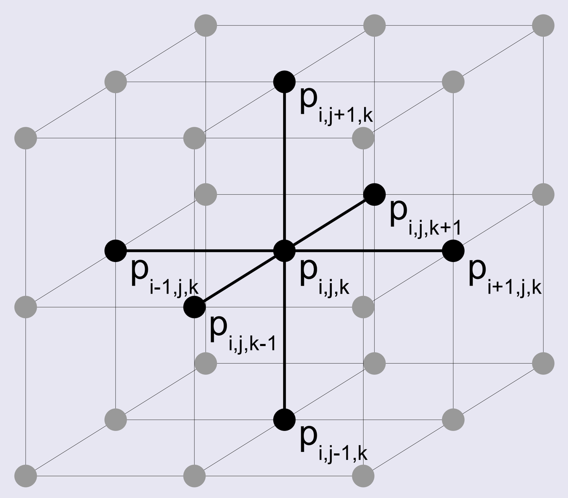
A fractional step method was utilized and implemented to solve the incompressible form of the Navier-Stokes equations

$$\frac{\partial u}{\partial t} = -(u \cdot \nabla)u - \frac{1}{\rho}\nabla p + \nu \nabla^2 u + f$$
$$\nabla \cdot u = 0 .$$

The fractional step method divides the N-S equations into individual parts, yielding four coupled sub-equations to be solved

$$\frac{\partial u_1}{\partial t} = f ,$$
$$\frac{\partial u_2}{\partial t} = -(u_1 \cdot \nabla)u_1 ,$$
$$\frac{\partial u_3}{\partial t} = \nu \nabla^2 u_2 ,$$
$$\frac{\partial u_4}{\partial t} = -\frac{1}{\rho}\nabla p .$$

After the last fractional update, the continuity constraint must be satisfied, so that the velocity field is non-divergent, i.e. $\nabla \cdot u_4 = 0$. A collocated regular grid representation has been chosen, because it fits the GPU programming model well. Spatial derivatives are approximated using second order central differences.



$$\nabla p_{i,j,k} = \begin{bmatrix} \frac{p_{i+1,j,k} - p_{i-1,j,k}}{2\Delta x} \\ \frac{p_{i,j+1,k} - p_{i,j-1,k}}{2\Delta y} \\ \frac{p_{i,j,k+1} - p_{i,j,k-1}}{2\Delta z} \end{bmatrix}$$

Figure 2. An example of a stencil from a collocated grid. The collocated grid fits the programming model well. Spatial derivatives are easily computed by the CUDA kernels using finite difference schemes.

We use a pressure projection technique to solve the fourth equation in order to satisfy the continuity constraint. This projection requires a Poisson equation to be solved

$$\nabla^2 p = \nabla \cdot u_3 ,$$

where the only unknown is the pressure p. Since the differential operators are approximated with finite differences, the equation is simply a system of linear equations. We have implemented an iterative multigrid-based solver for these problems. The solver performs very well compared to a traditional Jacobi solver.

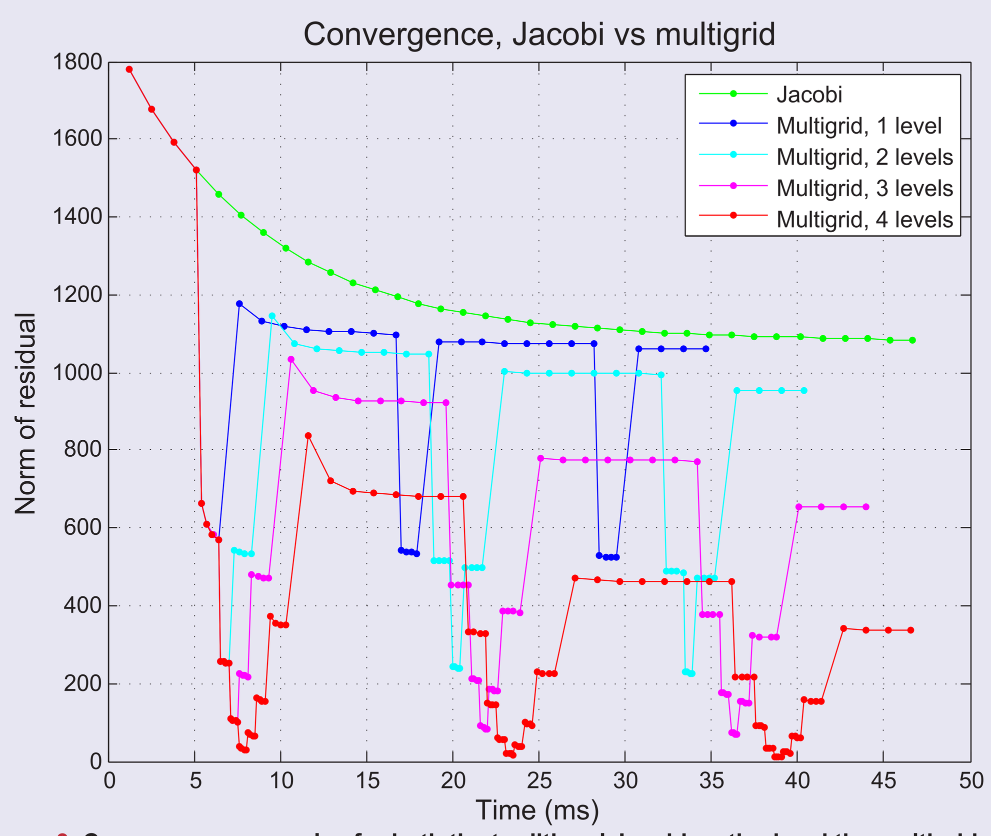


Figure 3. Convergence examples for both the traditional Jacobi method and the multigrid method. The peaks in the multigrid curves are due to the continuous restrictions of the linear systems. Notice how the multigrid method has improved the convergence significantly when it returns to the original system of equations.

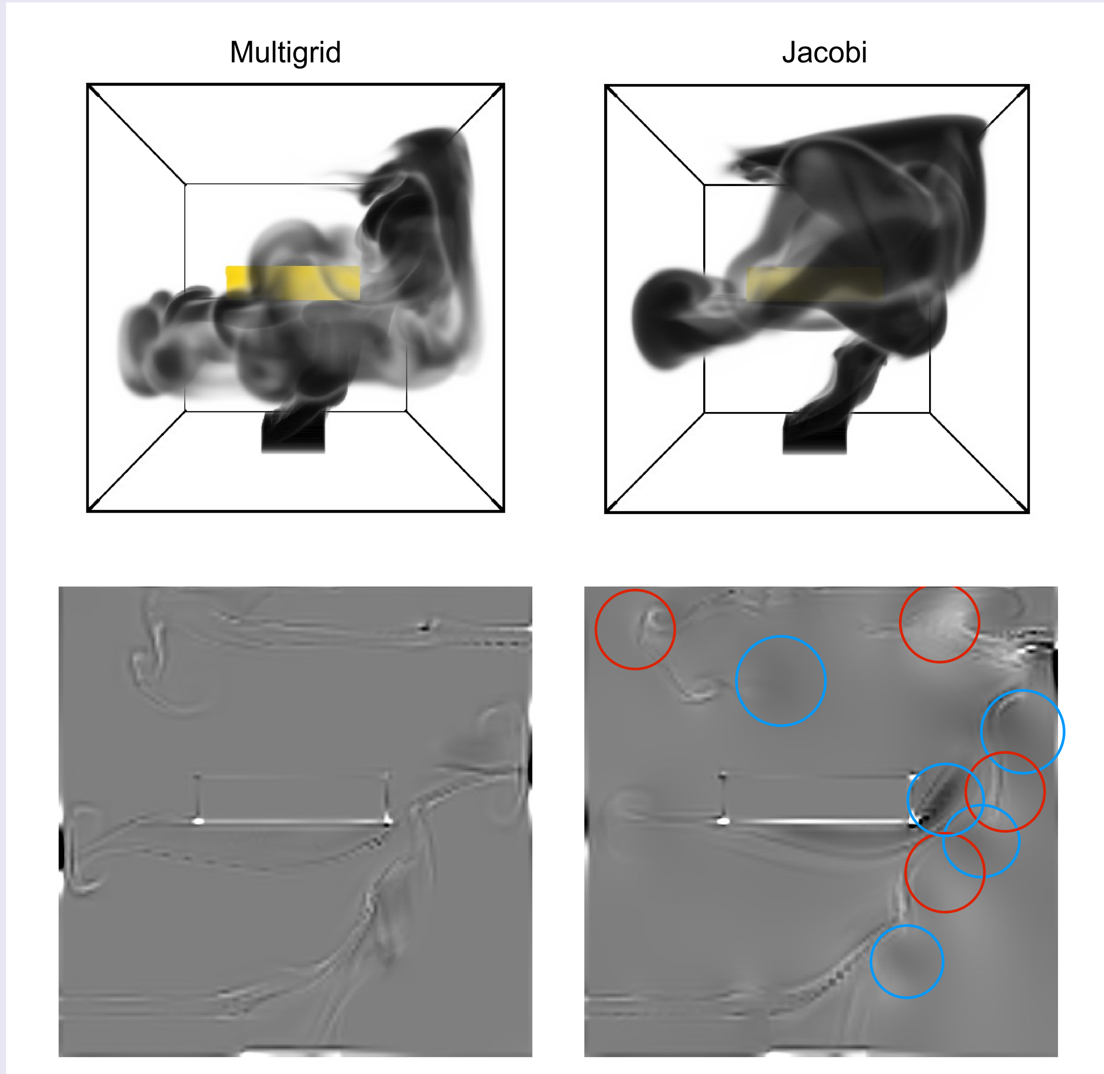
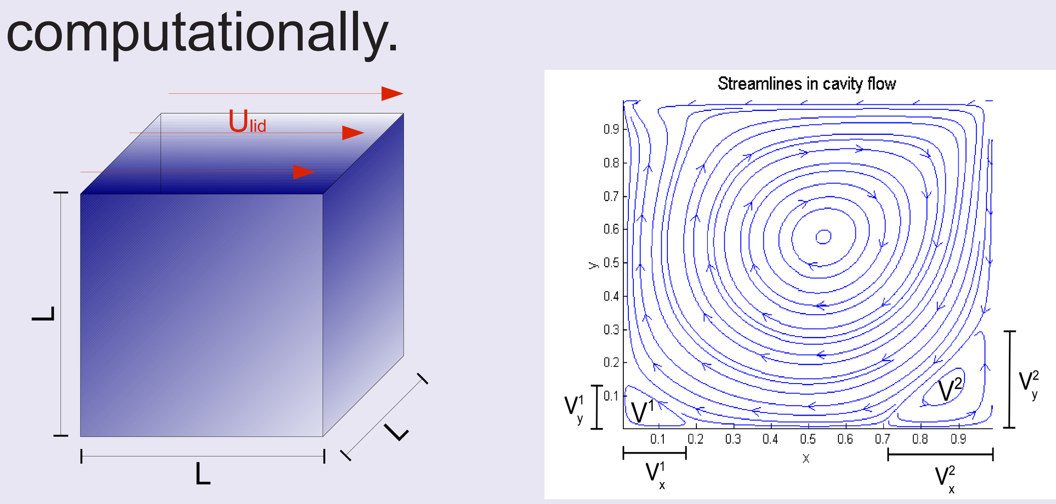


Figure 4. Simulation comparison of the multigrid and the Jacobi methods. The number of iterations was choosen so that the total time consumption are the same for both methods. *Top*: Details along with the rotational motions, are much more apparent with the multigrid method. *Bottom*: The divergence field at the center slice. Medium gray equals zero divergence. Only the Jacobi method suffers from low-frequency errors.

RESULTS

A range of case studies have successfully confirmed the useability of our CFD solver. One example is the *shear-driven cavity flow*. A rectangular cavity is modeled with a lid that is moving at a constant velocity U_{lid}. The streamline profile inside the cavity is measured and compared to a range of reference materials that are produced both experimentally and computationally.



V \ Res.	64^2	128^2	256^2	129^2 (Ghia*)
v^1_x	0.16	0.19	0.22	0.22
v^1_y	0.12	0.14	0.16	0.18
v^2_x	0.29	0.30	0.31	0.30
v^2_y	0.36	0.36	0.35	0.35

Figure 5. The shear-driven cavity setup and a 2D velocity streamline profile. The sizes of the secondary vortices in the streamline profile are close to the references in the last column of the table. **Ghia U., Ghia K. N., Shin C. T.: High-re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. Journal of Computational Physics 48, December 1982.*

FUTURE WORK

As part of my PhD-study I will continue working with GPU-based methods for solving partial differential equations. The purpose is to investigate efficient and robust solvers for large-scale Poisson type problems, and how to fit them to the parallel architecture of GPUs.

Note. This work was done as part of my masters thesis at the Department of Computer Science, University of Copenhagen, in collaboration with Rambøll Denmark. I therefore thank the involved supervisors, Kenny Erleben (KU) and Jens Chr. Bennetsen (Rambøll).