GPU-accelerated Computational Fluid Dynamics using Python and CUDA: Introduction

Lars Davidson
Division of Fluid Dynamics
Department of Mechanics and Maritime Sciences
Chalmers University of Technology, Gothenburg, Sweden

August 25, 2022

0.1 Background

This is a project course in the MSc programme Applied Mechanics. It is a mandatory course in late Spring (eight weeks). I had the kick-off meeting with four MSc students on March 24.

0.2 Lap-top

Do you have your own lap-top with Linux and a NVidia compatible graphics card? Install CUDA, AMGX and pyamgx, see Appendix C in **pyCALC-LES** .

1 Testing

- Test if CUDA works.
  - Download and run this code.
- Test if CUDA and pyamgx work.
  - Download and run this code.

2 Methodology

I recommend that the work is split into two parts.

- Re-write modules of **pyCALC-RANS** using the Python/CUDA interface. You find some info here. I have re-written one module in **pyCALC-RANS** and I got a speed-up of more than a factor ten on a Alienware x17 R1 laptop (here's my code). One of the key issues is probably that the amount of data transferred between the CPU to the GPU should be minimized.
- Use the **pyCALC-RANS** code. For running the code, look at the `readme` file. Replace the sparse-matrix Python solvers in **pyCALC-RANS** . Currently I'm using pyamg for the pressure correction equation (a Poisson equation) which

should be replaced with pyamgx which uses AMGX. Here's an example.
For the v_1 and v_2 equations I'm using a GMRES solver. You should find a corresponding GMRES solver for GPUs.

- Instead of starting from **pyCALC-RANS** you may write a simple CFD code from scratch using one of the codes given here.

The first test case could be the lid-driven cavity.
I have recently published my first scientific paper using **pyCALC-LES** .