CONSERVATION LAWS ON GPUS

A 2.5 DAY SHORT COURSE

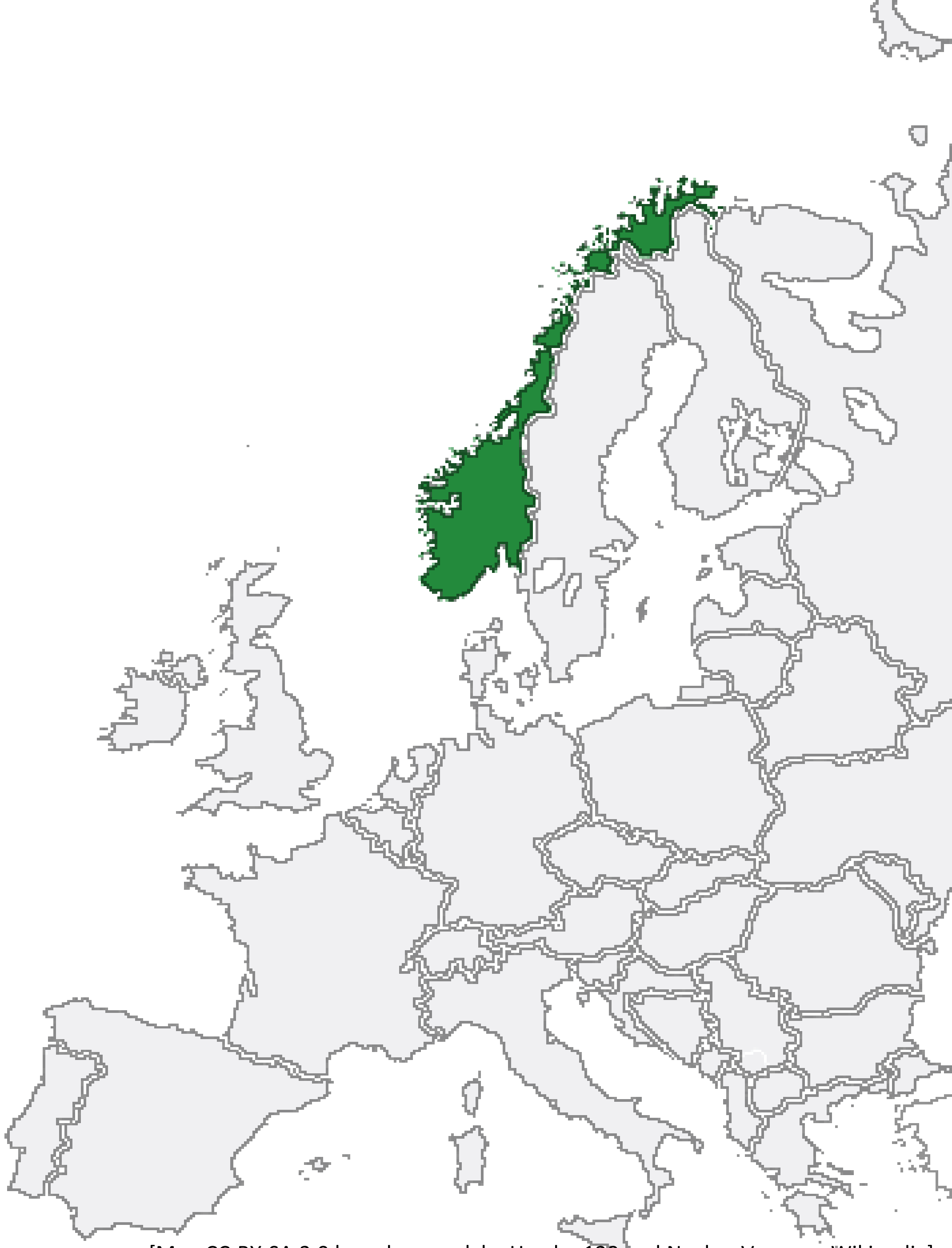
André R. Brodtkorb
Researcher, Department of Mathematics and Cybernetics, SINTEF Digital
Associate Professor, OsloMet – Oslo Metropolitan University

About me

- André R. Brodtkorb
- Ph.D. in scientific computing from the University of Oslo (2010)
- Researcher at SINTEF Digital – Department of Mathematics and Cybernetics
- Associate professor at Oslo Metropolitan University
- Main research interests:
  Numerical simulation, applied mathematics, GPU computing, high performance computing, reproducible research.

SINTEF

- Established 1950 by the Norwegian Institute of Technology.
- The largest independent research organisation in Scandinavia.
- A non-profit organisation.
- Motto: “Technology for a better society”.
- Key Figures*
  - 2100 Employees from 70 different countries.
  - 73% of employees are researchers.
  - 3 billion NOK in turnover
    (about 360 million EUR / 490 million USD).
  - 9000 projects for 3000 customers.
  - Offices in Norway, USA, Brazil,
    Chile, and Denmark.



[Map CC-BY-SA 3.0 based on work by Hayden120 and NuclearVacuum, Wikipedia]

Aim of course

- This is a practically oriented course where the aim is that you work through a lot of tutorials and exercises

- By the end of this course, you should have a Git repository with source code that you have written yourself.

- 2.5 days is not nearly enough to digest all of the material that will be presented.

- Keywords:
  CUDA, Python, OpenCL, C++, Git, logging, measuring performance, parallel reductions, gpu architecture, …

Outline of day 1

- (2 hours lectures, 2 hours tutorial, 3 hour exercise):
- Lecture (90 min): Introduction to GPU computing
  - Motivation for parallel programming, GPU computing, GPU basics
- Tutorial (45 min): Hello world in OpenCL and CUDA
  - Context creation, memory transfers to and from GPU, data types (float vs double), blocks and grids
- Exercise (45 min): GPU computing basics & developing GPU code
  - Matrix addition using PyCuda
  - (if time: Matrix-vector product using PyCuda)

- Tutorial (45 min): Computing Pi with CUDA
  - Serial, OpenMP parallel, shared memory intro, memory to compute ratio
- Exercise (45 min): Computing Pi with CUDA - optimization strategies
  - Compute Pi in PyCuda based on skeleton code

Outline of day 2

- (2 hours lectures, 2 hours tutorial, 3 hour exercise)
- Lecture (45 min): Floating point and best practices for software development
- Tutorial (45 min): Reproducible research
  - Git and version control, unit testing, logging, measuring performance
- Exercise (45 min): Benchmarking and performance
  - Single versus double precision, measuring performance, unit testing

- Lecture (45 min): Advanced GPU architecture (reductions and shared memory)
- Tutorial (45 min): Efficient GPU programming
  - Block size, shared memory reduction, texture memory, prepared call in pycuda, compilation flags, asynchronous memory transfers, asynchronous kernel launches
- Exercise (45+ min): Efficient kernel launches
  - Optimizing addMatrices and computing Pi

Outline of day 3

- (1 hour of lecture, 1 hour tutorial, 2 hour exercise)
- Lecture (45 min): Scalar conservation laws on GPUs (heat eqn, linear wave)
- Tutorial (45 min): Heat equation on the GPU
- Exercise (90 min): Linear wave equation on the GPU