Advanced Topics in High Performance Scientific Computing [MA5327]
Lecture 1

Manfred Liebmann
Technische Universität München
Chair of Optimal Control
Center for Mathematical Sciences, M17
manfred.liebmann@tum.de

October 12, 2015

Module Description

- **Recommended Prerequisites**
  - Numerical analysis
  - C/C++ programming languages
- **Intended Learning Outcomes**
  - *Upon the completion of the module, students are able to design and implement parallel algorithms for problems in scientific computing for different modern hardware architectures and parallel programming models.*

Lecture and Exercise Content

- **Lecture: First Part**
  - Basic principles of the development of efficient parallel algorithms on modern hardware architectures
- **Lecture: Second Part**
  - *Discussion of case studies for a variety of problems in numerical analysis*
- **Exercise**
  - Investigate practical and theoretical aspects of the implementation of parallel algorithms and programs in C/C++ for different hardware architectures.

Topics: First Part

- **Modern hardware architectures and networks**
  - CPUs, GPUs, accelerators, and interconnects
- **Standard libraries and extensions for parallel programs**
  - MPI, OpenMP, OpenACC, CUDA, intrinsics
- **Design patterns for parallel programs**
  - memory access, vectorization, threads, templates, and automatic code generation.
- **High performance computers**
  - scripting languages, batch systems

Topics: Second Part

- **Case studies**
  - Eigenvalue problems: *Vectorized QR algorithm for small matrices*
  - Iterative solvers: *Algebraic multigrid methods*
  - Fluid dynamics: *Riemann solver for Euler equations*
  - Partial differential equations: *Finite difference methods*
  - Quantum mechanics: *Time dependent Schrödinger equation*
  - Optimal control: *Optimal control of coupled Schrödinger equations*
  - Determinants: *Continuous roots of a complex determinant*
  - Semiclassical approximations: *Semiclassical Quantum Dynamics*

Exercises

- **Goals**
  - Gain experience with parallel programming models
  - Support the development of the software projects
  - Discussion forum
- **Exercise sheets**
  - Programming and theoretical exercises
  - Presentation of programs and solutions

Examination

Students must design and implement a software project related to high performance scientific computing.

A written project report must be submitted and defended in an oral examination at the end of the semester (45 min).

The students must demonstrate with the report that they have gained a deeper understanding of parallel programming models and are able to solve scientific computing problems with efficient parallel algorithms on modern hardware architectures. In the oral examination the students must demonstrate that they are familiar with the program code of the software project and must defend their software design decisions.

Software Projects

- **Project topics**
  - Software project topics can be freely chosen but must be approved
  - Projects must have an affinity to high performance scientific computing
  - Project assignments should be finished by end of November
  - Programming language: C/C++
  - Send project proposals via email: `manfred.liebmann@tum.de`

- **Project reports**
  - 5-10 pages
  - Problem statement
  - Solution methods and algorithms
  - Parallel implementation
  - Performance studies
  - Discussion

Reading List

Introduction to C++ programming for the lecture with a scientific computing angle:

Pitt-Francis, *Guide to Scientific Computing in C++*, Springer 2012

Discussion