



Seminar zu High Performance Computing und Research Software Engineering

In diesem Seminar wollen wir uns mit zwei für unsere und andere Arbeitsgruppen sehr wichtigen und teilweise zusammenhängenden Themenblöcken befassen.

Im **high performance computing (HPC)** geht es insbesondere darum Techniken zu lernen um auf verschiedener Hardware effiziente Software zu schreiben. Insbesondere geht es darum die hohen Rechenleistungen im TeraFLOPS Bereich und höher, also über 10^{12} Gleitkommarechenoperationen in der Sekunde auch tatsächlich zu nutzen und damit auch z.B. komplizierte physikalische Gleichungen in hinnehmbarer Zeit zu Lösen.

Bei **research software engineering (RSE)** geht es darum Methoden der Softwaretechnik auch in wissenschaftlichen Anwendungen einzusetzen und insbesondere auch darum durch gut dokumentierte Open Source Software die Zusammenarbeit in der wiss. Community zu stärken und das „von Grund auf neu schreiben“ von funktional nahezu identischer Software zu vermeiden.

Entsprechend gibt es zu beiden Blöcken jeweils mögliche Themen die ausgearbeitet und präsentiert werden können. Wir sind aber auch offen für eigene Ideen wenn diese in den Rahmen des Seminars passen.

Mögliche Themen für den Bereich HPC sind:

1. Das message-passing (MPI) Konzept für verteiltes Rechnen
2. Thread-parallelisierung z.B. mit OpenMP
3. Rechnungen auf Grafikkarten (GPUs)
4. Die Softwarebibliothek Kokkos als Tool für Thread parallelisierung und GPU-computing
5. Matrixfreie Methoden im Bereich der Finiten Elemente
6. Mehrgitterverfahren
7. Grundlagen paralleler Linearer Algebra (Lösung von Gleichungssystemen etc.)
8. die Bibliotheken BLAS (Basic Linear Algebra Subpackage), LAPack (Linear Algebra Package) ScaLAPack und die Intel (one)MKL

Mögliche Themen für den Bereich RSE sind:

1. Die FAIR Prinzipien für offene Software
2. Versionskontrolle (mit git)
3. integrated development environments (IDEs)
4. Buildsysteme/Buildsystemgeneratoren (z.B. CMake)
5. Erzeugen von (Klassen-)Dokumentation mit Doxygen
6. Test driven development
7. Continuous Integration
8. Was bedeutet Reproduzierbarkeit für Software?
9. Code reviews
10. Typische Arbeitsabläufe auf GitHub und Co. (Forks, Pull request, issues etc.)
11. Modularität von Software
12. Defensive Programmierung mit Asserts etc.

Eine erste Vorbesprechung findet am 30.03.23 um 10:00 Uhr im Raum C311 statt.

Fakultät für Mathematik
und Physik

Jan Philipp Thiele, M.Sc., M.Sc.
Wissenschaftlicher Mitarbeiter
AG Wiss. Rechnen
E-Mail: thiele
@ifam.uni-hannover.de

Leon Kolditz, M.Sc.
Wissenschaftlicher Mitarbeiter
AG Wiss. Rechnen
E-Mail: kolditz
@ifam.uni-hannover.de

Prof. Dr. Thomas Wick
AG Wiss. Rechnen

10. Februar 2023