

<p style="text-align: center;"><b>1. Übungssektion: – Computerorientierte Theoretische Physik</b> <b>Schleifen, Parallelisierung, OpenMP</b></p>
--

**Erster Übungstermin dieser Sektion: 25.10.2018, Die Nachbereitungsaufgaben sind in der folgenden Übung nach Abschluss dieser Sektion abzugeben/zu besprechen.**

**Vorbereitungsaufgabe 1 : *Generelle Vorbereitung für den Kurs***

1. In diesem Kurs werden vor allem Bibliotheken in C/C++ unter Linux verwendet (Petsc, Slep3c und ITensor etc.). Sollten Sie die Grundzüge von C/C++ (Schleifen, Funktionen, Variablen, Objekte) nicht beherrschen, so sollten Sie sich diese vor der ersten Übung anschauen und lernen (und mehr wird nicht als Voraussetzung am Anfang erwartet). Grundsätzlich werden zwar auch Lösungen in anderen Programmiersprachen akzeptiert, aber nicht in allen Fällen gibt es die nötigen Schnittstellen zu den Bibliotheken oder vergleichbare Bibliotheken in den entsprechenden Sprachen (ggf. vorher besprechen).
2. Beschaffen Sie sich bitte einen PC-Pool Zugang und probieren Sie einfache C/C++-Programme mit Makefile zu erstellen.
3. Es wird für jede Übung vorausgesetzt, dass Sie sich alle Aufgaben vorher anschauen und erste Lösungsstrategien entwickelt haben. Damit wir uns auf Fragen und Probleme mit den Aufgaben in der Übung konzentrieren können.
4. Es wird erwartet, dass jedes Gruppenmitglied vorbereitet ist und dass die Vorbereitungen gemeinsam erarbeitet und besprochen werden.
5. Die Bearbeitung erfolgt ähnlich wie in einem Projekt orientiert an einer bestimmten Fragestellung, die untersucht werden soll.

**Vorbereitungsaufgabe 2 : *Zeitmessung, OpenMP***

1. Informieren Sie sich (z.B. im Internet), wie Zeiten in einem Computerprogramm gemessen werden können, um die Performance zu analysieren. Insbesondere ist es wichtig für die Performanceanalyse, die Zeitauflösung zu kennen.
2. Informieren Sie sich bitte über OpenMP in C/C++ Programmen! Die folgenden Aspekte wollen wir untersuchen: Parallelisierung von Schleifen, Synchronisation von Threads (z.B. omp critical), Parallelisierung mittels tasks.

**Aufgabe 3 : *Hauptspeicher versus Cache; Ausführungsreihenfolge***

1. Die Art wie die Daten im Speicher abgelegt werden und wie diese abgerufen werden, hat heute einen entscheidenden Einfluss auf die Geschwindigkeit einer Simulation. Suchen Sie sich ein numerisches Problem aus der Physik (z.B. ein diskretisiertes mehrdimensionales Integral mit einer Faltung) an dem Sie diesen Einfluss untersuchen können und führen Sie eine systematische Studie durch.
2. Versuchen Sie auf ähnliche Weise zu untersuchen, inwieweit es bei ineinander verschachtelten Schleifen sich die Ausführungsgeschwindigkeit ändert, wenn Sie Rechenoperationen zwischen den verschachtelten Schleifen verschieben.

#### **Aufgabe 4 :** *Parallelisierung mit OpenMP*

1. Verwenden Sie das Beispiel aus der letzten Aufgabe (oder ein anderes physikalisches) umzu zeigen, wie mittels OpenMP parallelisiert werden, und wie es die Ausführungsgeschwindigkeit verändert. Dabei soll sowohl eine einfache Parallelisierung der Schleife als auch mittels `omp tasks` untersucht werden.
2. Untersuchen Sie auch den Einsatz von Synchronisierungen und bei den `tasks` den Einsatz von Abhängigkeiten.
3. Bei allen Teilaufgaben, achten Sie auch darauf, dass ihr Code richtig rechnet.