

---

## Parallelisierung mit MPI (Gauß-Seidel: 500 Punkte)

Parallelisieren Sie das GS-Verfahren in dem `partdiff`-Programm gemäß dem Parallelisierungsschema.

Beachten Sie dabei folgende Anforderungen:

- Abbruch
  - Es gibt hier zwei Fälle, die auf Korrektheit der Parallelisierung zu prüfen sind:
    1. Abbruch nach fester Iterationszahl (beide Störfunktionen)
    2. Abbruch nach Genauigkeit (beide Störfunktionen) - von der Wertigkeit her entspricht dieser Aspekt **mehr als 50%** der Punkte
  - Dabei soll nach gleicher Iterationszahl das Ergebnis (Matrix und Fehlerwert) identisch bleiben.
  - Beim Abbruchkriterium „erreichte Genauigkeit“ muss die parallele Variante nicht unbedingt bei derselben Iteration wie die sequentielle abbrechen. Es kann bis zu  $|\text{nprocs}|$  Iterationen mehr gerechnet werden. Die Matrix und der Fehlerwert für  $i + |\text{nprocs}|$  Iterationen verglichen mit dem sequentiellen Programm, welches nach  $i + |\text{nprocs}|$  Iterationen abbricht, muss aber gleich sein! Dabei gilt weiterhin, dass der Lauf deterministisch sein muss, d.h. ein wiederholter Aufruf endet immer in der gleichen Iteration.
  - Überprüfen Sie, dass die Ergebnisse mit 24 Prozessen auf zwei Knoten richtige Ergebnisse liefern.
  - Sie dürfen keine mathematischen Abkürzungen nutzen, die es erlauben würden, nur auf Grundlage eines prozesslokalen Residuums über den Abbruch nach Genauigkeit zu entscheiden. Der Abbruch darf lokal nur auf Grundlage des globale maximalen Residuums über alle Prozesse aus der gleichen Iteration entschieden werden.
- Code
  - Zu keinem Zeitpunkt darf ein Prozess die gesamte Matrix im Speicher halten. Die Matrix soll auf alle Prozesse gleichmäßig verteilt werden.
  - Das Programm muss mit beliebigen Prozesszahlen funktionieren (auch für  $p == 1$  oder  $p \gg N$ ). Einfach das ganze Programm kontrolliert abzurechnen, reicht nicht.
  - Erstellen Sie eine eigene Funktion für die MPI-Parallelisierung des GS-Verfahrens.
  - Jacobi muss dabei weiterhin **parallel** funktionieren.
  - **Hinweis:** Sie können die in den Materialien bereitgestellte `displayMatrixMPI`-Funktion als Grundlage für die parallele Ausgabe der Matrix benutzen. Beachten Sie, dass diese Funktion Werte überschreibt und daher erst ganz am Ende aufgerufen werden sollte.

- Dokumentieren Sie Ihren Code ausreichend.
- Laufzeit
  - Das Programm darf mit zwei Prozessen nicht langsamer als die sequentielle Variante sein. Es muss ein zufriedenstellender Speedup erreicht werden.
- Kommunikation
  - Nutzen Sie synchronisierende Funktionen (`MPI_Ssend` und `MPI_Issend`) um ihr Programm auf mögliche Verklemmungen zu untersuchen. Für die finale Abgabe können sie diese Funktionen gegen die Standard-Funktionen wieder austauschen.
  - Jeder nichtblockierende Kommunikationsaufruf muss mit einem passenden `MPI_Wait` oder einem erfolgreichen `MPI_Test` abgeschlossen werden. Anderenfalls ist der Aufruf falsch.

## Abgabe

Abzugeben ist ein gemäß den bekannten Richtlinien erstelltes und benanntes Archiv. Das enthaltene und gewohnt benannte Verzeichnis soll folgenden Inhalt haben:

- Alle Quellen, aus denen Ihr Programm besteht; gut dokumentiert (Kommentare bei geänderten Code-Teilen!)
  - Erwartet werden die Dateien `Makefile`, `askparams.c`, `partdiff.c` und `partdiff.h`.
- Ein `Makefile`
- **Keine** Binärdateien!

Senden Sie das Archiv an `hr-abgabe@wr.informatik.uni-hamburg.de`.