

# 1 Schema fuer Jacobi

1) Datenaufteilung der Matrix auf die einzelnen Prozesse:

Die N Prozesse bekommen gleich viele Zeilen der Gesamtmatrix als Teilmatrix zugewiesen. Die jeweilige Anzahl an Zeilen  $N_z$  erhält man, indem man die Anzahl der Zeilen der Gesamtmatrix ganzzahlig durch numproc teilt. Geht ein Teilen durch Numproc nicht auf wird durch (numProc-1) geteilt. Bleibt kein Rest, werden die  $N_z$  wieder durch numProc geteilt. Alle Prozesse bekommen dann  $N_z/\text{numProc}$  Zeilen, bis auf den letzten  $(N_z/\text{numProc}) + \text{modulo}(N_z/\text{numProc})$ . Bleibt bei  $N_z/\text{numProc}$  ein Rest bekommen alle Prozesse  $N_z/\text{numProc}$  Zeilen und der letzte den Rest.

Bei der Verteilung muss außerdem Beachtet werden, dass die einzelnen Prozesse zur Berechnung des Abtaststerns jeweils die Randwerte von anderen Prozessen benötigen. Die Teilmatrizen 2 bis N-1 benötigen deshalb zusätzlich jeweils eine Zeile unten und oben. Die Teilmatrix 1 benötigt nur eine Zeile unterhalb, die Teilmatrix N nur eine Zeile oberhalb.

Bei der Aufteilung mit *MPI\_SCATTERV* werden die Teilmatrizen deshalb so aufgeteilt, dass sich jeweils in einer Zeile überlappen. So liegen für den ersten Iterationsschritt auch bereits alle benötigten Randwerte vor.

## 2) Parallelisierungsschema

Nach der Aufteilung mit *MPI\_SCATTERV* haben alle Teilmatrizen alle benötigten Randwerte (siehe 1)). Der erste Schritt in jedem Teilprozess und jedem Iterationsschritt ist deshalb die Berechnung des Abtaststerns für jedes Matrixelement (außer für die Randwerte). Danach müssen die Ergebnisse, die von anderen Prozessen wiederum als Randwerte benötigt werden kommuniziert werden:

Die Prozesse  $i=1$  bis  $i=N-1$  senden ihre zweitunterste Zeile (nicht die Zeile mit den Randwerten, sondern die darüber!) an den darauffolgenden Prozess  $(i+1)$ . Der Prozess  $i+1$  ersetzt seine oberste Zeile mit der empfangenen.

Die Prozesse  $i=2$  bis  $i=N$  senden ihre zweitoberste Zeile an den vorherigen Prozess  $(i-1)$ . Der Prozess  $i-1$  ersetzt seine unterste Zeile mit der empfangenen.

Für einen einzelnen Prozess  $i$  bedeutet das: er führt zuerst die Berechnung durch. Danach sendet er seine zweitunterste Zeile an den nächsten Prozess und seine zweitoberste an den vorherigen Prozess. Anschließend empfängt er jeweils eine Zeile vom vorherigen und nachfolgenden Prozess und ersetzt seine erste und letzte Zeile durch die empfangenen Zeilen. Bei der ersten (letzten) Teilmatrix ist zu beachten, dass diese keine neuen Randwerte für ihre erste (letzte) Zeile empfängt. In diesen Zeilen gelten die konstanten Randbedingungen, die zu Beginn festgesetzt werden.

## 3) Abbruchkriterium

bei fester Iterationszahl:

Dieses Abbruchkriterium ist einfach zu implementieren. Die Schleife zur Berechnung und Kommunikation wird genauso oft durchgeführt, wie die Anzahl an Iterationen ist.

bei Iteration nach Genauigkeit:

jeder Prozess setzt nach jedem Iterationsschritt eine logical Variable auf 0 oder 1. 1 bedeutet, dass das Abbruchkriterium in diesem Prozess erfüllt ist, 0 bedeutet dass es nicht erfüllt ist. Diese Variable wird an den Masterprozess versendet, der mit *MPI\_REDUCE* alle logicals der Teilprozesse empfängt und das Minimum berechnet. Ist das Minimum 0, ist das Abbruchkriterium noch nicht in jedem Teilprozess erfüllt und es muss weiter iteriert werden. Ist das

Minimum 1, kann die Iteration gestoppt werden. Falls dies der Fall ist, sendet der Master mit MPI\_BROADCAST eine logical Variable an alle Teilprozesse, sodass diese 'wissen', dass die Iteration gestoppt werden muss.

## 2 Gauss-Seidel

Die Aufteilung der Matrix geschieht genauso wie bei Jacobi. Verfahren aus Sicht eines einzelnen Prozesses:

Halolinien sind Randlinien(Randbedingungen - oberste und unterste Linie einer Teilmatrix)

1. Empfangen der oberen Haloline vom vorherigen Prozess
2. Berechnung der ersten Zeile der Teilmatrix
3. Versenden dieser Zeile an den vorherigen Prozess
4. Berechnung der restlichen Zeilen der Teilmatrix
5. Versenden der letzten berechneten Zeile an den nächsten Prozess
6. Empfangen der unteren Haloline vom nächsten Prozess

Besonderheiten:

Beim nullten Prozesses entfallen die Schritte 1-3, beim letzten Prozesses entfallen die Schritte 5-6.

Abbruchkriterium nach Iteration:

Hier gibt es keine besonderen Schwierigkeiten. Die Schleife wird von jedem Prozess bis zur gewünschten Iterationszahl durchlaufen. Wie in Aufgabe 7A führt der Masterprozess anschließend mit MPI\_GATHERV die Teilmatrizen wieder zusammen.

Abbruchkriterium nach Genauigkeit:

Zunächst wird wie in Aufgabe 7b geprüft, wann alle Prozesse die gewünschte Genauigkeit erreicht haben. Beim Gauss-Seidel Verfahren besteht nun die Schwierigkeit darin, dass sich die einzelnen Prozesse in verschiedenen Iterationen befinden. Der nullte Prozess hat dabei immer die größte Anzahl an Iterationen durchlaufen, da der nachfolgende Prozess immer auf die letzte berechnete Zeile des vorherigen warten muss. Mit MPI\_(I)REDUCE überprüft der nullte Prozess, die Abbruchkriterien aller Prozesse. Sobald alle die gewünschte Genauigkeit erreicht haben, hört der nullte Prozess auf zu rechnen und alle anderen Prozesse rechnen um Rank Iterationen weiter. So wird sichergestellt, dass alle Prozesse die gleiche Anzahl an Iterationen durchlaufen haben. Mit einem MPI\_REDUCE einer weiteren logical Variable, welches vom nullten Prozess ausgeführt wird, stellt dieser fest ob alle Prozesse fertig sind. Dann werden mit MPI\_GATHERV die Teilmatrizen durch den nullten Prozess zusammengeführt.