

Parallelisierungsschema

Datenaufteilung

Jeder Prozess erstellt nur seine Teilmatrix, dazu muss die Initialisierung der Matrizen umgeschrieben werden. Wieviele Zeilen die Matrix des Prozesses hat kann jeder Prozess selber berechnen.

Am Ende muss die Gesamtmatrix ausgegeben werden. Dazu senden die Prozesse 1-n dem Prozess 0 ihre Teilmatrizen. Dieser kann dann die Gesamtmatrix zusammenbauen indem er die Nachrichten in der Reihenfolge der Ränge der Prozesse empfängt.

Bei der Initialisierung dürfen die Anfangswerte nur an den Rändern der Matrix auftauchen, daran muss die Initialisierung der Matrix je nach Rang angepasst werden. Dazu muss jeder Prozess wissen wie groß die Gesamtmatrix ist.

Die Parameter der Matrix müssen an die Größen der Teilmatrizen angepasst werden. Die Prozesse 1-(n-1) bekommen noch jeweils zwei Zeilen dazu, eine für die Zeile über der ersten Zeile und eine für die Zeile hinter der letzten. Welche Werte diese Zeilen haben empfängt der Prozess i von den Prozessen mit Rang $(i-1)$ und $(i+1)$. Am Anfang werden sie mit den Startwerten initialisiert.

Jacobi

In der 0-ten Iteration hat jeder Prozess die Daten die er zur Berechnung braucht. Also berechnet er zuerst. Am Ende der 0-ten Iteration wartet der Prozess bis alle anderen Prozesse mit der 0-ten Iteration fertig sind. Danach schickt Prozess 0 seine letzte, die Prozesse 1-(n-1) jeweils ihre erste und ihre letzte und Prozess n die erste Zeile an die jeweils davor und dahinter liegenden Prozesse. Danach empfängt der Prozess je nach Rang Zeilen die von den anderen Prozessen berechnet wurden. Dabei muss darauf geachtet werden dass kein Deadlock entsteht. Nach dem Empfang der benötigten Zeilen kann die nächste Iteration beginnen.

Gauß-Seidel

Da bei Gauß-Seidel in jeder Iteration die vorher in der selben Iteration berechneten Werte schon verwendet werden, muss der Prozess warten bis der Prozess vor ihm mit der Iteration fertig ist und ihm die benötigte Zeile geschickt hat.

Der Prozess empfängt am Anfang jeder Iteration die benötigten Zeilen, außer er ist Prozess 0, dann kann er sofort anfangen zu rechnen. Wenn er diese empfangen hat, fängt er an zu rechnen und danach sendet er die letzte Zeile an den nachfolgenden Prozess. Am Ende der Iteration müssen noch die Zeilen für die nächste Iteration empfangen werden. Danach kann sofort die nächste Iteration durchlaufen werden ohne das auf Beenden der anderen Prozesse gewartet werden muss.

Abbruch

Wenn bei einem Prozess eine bestimmte Genauigkeit erreicht wurde, wird dieser beendet, nachdem er seine zuletzt berechneten Werte an Prozess 0 gesendet hat. An die anderen Prozesse sendet er nicht mehr, denn wenn sie weiterrechnen würden, würden nicht alle Prozesse in der selben Iteration stoppen.

Um die anderen Prozesse anzuhalten, wird eine Variable eingeführt, die bei jeder Iteration abgefragt wird, sie ist am Anfang 0. Der fertige Prozess sendet einen Wert $\neq 0$ an alle anderen Prozesse. Diese empfangen den Wert und speichern ihn in ihrer Abbruchvariablen. Da sich die Prozesse bei Jacobi alle in der selben Iteration befinden, bekommt man dabei das selbe Ergebnis wie bei der sequentiellen Variante des Programms.

Bei Gauß-Seidel müssen alle Prozesse ihre Teilmatrizen auf den Zustand bringen, der in der Iteration bestand, in der der erste Prozess aufgehört hat zu rechnen.