

Datenaufteilung:

Wie in bereits bearbeiteten Aufgaben wird die Matrix Zeilenweise auf die Prozesse verteilt. So erhält jede Matrix M/n viele Zeilen wobei M die Zeilenanzahl der Matrix und n die Anzahl der Threads ist. Wobei das Jakobi-Verfahren natürlich 2 solcher Matrixblöcke benötigt während Gauß-Seidel in-place abläuft.

Parallelisierungsschema für das Jacobi-Verfahren:

Im Jakobi verfahren können alle Threads sofort gleichzeitig anfangen ihren Matrixausschnitt zu berechnen. Nachdem ein Thread eine Iteration fertig hat schickt er dem vorangehenden Thread seine erste Zeile und dem Nachfolgendem seine letzte, da diese benötigt werden damit diese ihre jeweils Letzte und Erste berechnen können. Dabei senden die Threads die Zeilen sobald sie sie berechnet haben, also so früh wie möglich und empfangen sie unmittelbar bevor sie sie benötigen also so spät wie möglich. Beim Abbruch nach Iterationsanzahl bricht jeder Thread ab sobald er die gewünschte Anzahl erreicht hat. Beim Abbruch über das maximale Residuum muss zusätzlich, durch eine Barrier sichergestellt werden, dass keiner der Threads mehr als eine Iteration vor allen Anderen ist, damit die Matrix im korrekten Iterationsschritt ausgegeben werden kann. Da im Jakobi-Verfahren die Matrix-Elemente des Letzten Iterationsschrittes noch vorhanden sind, kann man sogar eine Iteration in die Vergangenheit greifen um sicher zu stellen, dass sich alle Elemente auch beim Abbruch nach maximalem Residuum in der richtigen Iteration befinden auch dann wenn ein Thread maximal 1 Iteration vorraus ist.

Kommunikation: Immer dann wenn man die relevanten Zeilen berechnet hat (die Erste und Letzte des eigenen Matrixausschnittes) sendet man diese an die Nachbarn. Man empfängt immer dann wenn man vor der Berechnung einer relevanten Zeile steht um den anderen Prozessen möglichst viel Zeit zu geben ihre Berechnungen zu machen. Beim Abbruch nach Residuum muss man wenn man mehr als einen Iterationsschritt vor einem beliebigen anderem ist warten, damit die korrekte Matrix zusammengesetzt werden kann.

Parallelisierungsschema für das Gauß-Seidel-Verfahren:

Anders als im Jakobi-Verfahren muss der n -te Thread darauf warten, dass der $n-1$ -te Thread ihm seine Letzte Zeile gibt bevor er beginnen darf, mit ausnahme des Ersten Threads. Wenn der Erste dem Zweiten diese Zeile übergeben hat, darf er wieder von vorne seine Zeilen berechnen. Bevor der Letzte beginnt müssen also alle anderen Threads bereits gerechnet haben. Der eErste Thread ist also n Iterationen vor dem letzten Thread.

Kommunikation:

Wie schon beim Jakobi-Verfahren werden die relevanten Zeilen sofort abgeschickt sobald sie berechnet wurden und so spät wie möglich empfangen.

Abbruch:

Beim Abbruch nach Iteration beendet jeder Thread nachdem er die gewünschte Anzahl erreicht hat seine Berechnung wie gehabt. Da der erste Thread n Iterationen vor dem letzten ist wird die Zeit also um n Iterationen dauer erhöht, davon ausgehend das diese etwa gleich lang brauchen. Bei hoher anzahl an iterationen (wovon wir ausgehen) sollte dies nicht ausschlaggebend sein. Der Abbruch nach Residuum im Gauß-Seidel-Verfahren ist nicht deterministisch. Man bricht dann ab wenn alle Elemente das mindest Residuum erreicht haben dadurch erhält man eine Matrix die der Anforderung der Genauigkeit entspricht allerdings unter Umständen immer anders aussieht, da die Threads in unterschiedlichen Iterationen sein können. Wenn man wie beim Jakobi-Verfahren eine Barrier benutzt um die Iterationen fest zu halten läuft das Programm nur noch sequentiell ab. Man könnte auch jedem Thread n viele Matrixausschnitte halten lassen, um die Richtige Matrix konstruieren zu können, was allerdings ein immenser Speicheraufwand wäre.