

Messung 1 (Threads → Sekunden)

Bei Messung 1 sieht man nur eine grobe Proportionalität zwischen Threadanzahl und Berechnungszeit. Dass die Zahl der Threads hier nicht sauber skaliert, wird wahrscheinlich am Overhead liegen, der für jeden neuen Thread eingeführt wird: Einerseits müssen die neuen Threads überhaupt erstellt werden und Speicherplatz zugewiesen bekommen; andererseits muss die Matrix auch den Threads zugewiesen werden, was je nach Teilbarkeit von Threads und Matrixgröße unterschiedlich einfach, und damit auch zeiteffizient erfolgt. Ab 9 Threads ist wahrscheinlich ein point of diminishing returns, wo die Aufteilung der Aufgaben auf die Threads so kleinteilig wird, dass der Overhead dadurch größer ist als die Einsparung der Rechenzeit.

Messung 2 (Interlines → Sekunden)

Beim Ändern der Matrixgröße, bzw. Interlines zeigt sich eine eindeutigere Proportionalität zwischen ihr und der Berechnungszeit. Dies ist auch logisch, da die Komplexität der Berechnung hier $O(n)$ ist, also jedes zusätzliche Matricelement erfordert eine weitere Rechnung (bzw. Rechnung pro Iteration).