

# Wissenschaftliches Rechnen – Paralleles Höchstleistungsrechnen

Christian Engwer

[http://wwwmath.uni-muenster.de/num/Vorlesungen/WissenschaftlichesRechnen\\_SS11/](http://wwwmath.uni-muenster.de/num/Vorlesungen/WissenschaftlichesRechnen_SS11/)

Abgabe 9. 5. 2011. Abgabe der Programmieraufgaben bis per Email an [christian.engwer@uni-muenster.de](mailto:christian.engwer@uni-muenster.de), schriftliche Abgabe Dienstags in der Vorlesung.

- 
- Alle Programmierübungen müssen per Email und in ausgedruckter Form abgegeben werden.
  - Achten sie darauf, ihr Programm ordentlich zu formatieren und gut zu kommentieren.  
**In Zukunft wird die Form mit in die Bewertung eingehen.**
- 

## ÜBUNG 1 HIGH PERFORMANCE COMPUTING.

In dieser Übung wollen wir für zwei verschiedene numerische Anwendung (sogenannte “kernel”) untersuchen wieviele Rechenoperationen pro Sekunde heute möglich sind. Implementieren Sie folgende mathematische Operationen:

1. *Matrixmultiplikation.* Gegeben zwei Matrizen  $A, B \in \mathbb{R}^{n \times n}$  dann ist das Matrixprodukt  $C = AB$  gegeben durch

$$c_{i,j} = \sum_{k=1}^n a_{i,k} b_{k,j} \quad .$$

2. *Gauß-Seidel 2d.* Es sei  $\Omega_n^d = \{(i_0, \dots, i_{d-1}) \in \mathbb{Z}^d \mid \forall 0 \leq k < d : 0 \leq i_k < n\}$ . Für die Gitterfunktionen  $u^m : \Omega_n^2 \rightarrow \mathbb{R}$  definiere die Iteration

$$u^{m+1}(i, j) = \frac{1}{4} (u^{m+1}(i-1, j) + u^{m+1}(i, j-1) + u^m(i, j+1) + u^m(i+1, j)) \quad (i, j) \in [1, n-1]^2.$$

In beiden Fällen liegen Daten, welche in einem 2D Feld organisiert sind zugrunde. Aus Performancegründen wollen wir die 2D Struktur auf ein 1D Feld abbilden.

Implementieren Sie die Kernel in C/C++ und verwenden sie als Datenstruktur die in der Vorlesung besprochenen Vektorklassen `std::vector`. Bestimmen Sie für jede Operation die Anzahl der Fließkommaoperationen und ermitteln Sie die Geschwindigkeit des Programmes in “Millionen Fließkommaoperationen pro Sekunde”.

Zur Zeitmessung können Sie die Funktionen aus der ersten Übung, welche in `nbstopwatch.hh` zur Verfügung gestellt werden, verwenden.

Vergessen Sie nicht die Felder mit sinnvollen Daten (nicht 0.0) zu initialisieren, z. B.  $u(i, j) = i + j$ , oder wie in der Vorlesung mit Zufallswerten.

Übersetzen Sie das Programm mit maximaler Optimierungsstufe. Für den GNU C/C++ Compiler ist etwa `-O3 -funroll-loops` empfehlenswert.

Verbessern Sie die Cachenutzung durch Kacheln und ermitteln Sie die Beschleunigung.

Stellen Sie die Ergebnisse in graphischer Form, d. h. MFLOPS über Problemgröße dar. Zur Darstellung können Sie z.B. das Programm `gnuplot` (siehe Hinweis) verwenden. Diskutieren Sie die Form der Kurven, insbesondere warum die MFLOP-Rate wann absinkt. Vergleichen Sie die Ergebnisse von Matrixmultiplikation und Gauß-Seidel-Iteration.

Bei der Gauß-Seidel-Iteration können Sie mit anderen Durchlaufreihenfolgen experimentieren. Z. B. im 2d-Fall das Schachbrett muster: Bearbeite zuerst alle Indizes mit  $i+j$  gerade, dann die mit  $i+j$  ungerade. Falls möglich, testen Sie ihre Programme auf verschiedenen Rechnern.

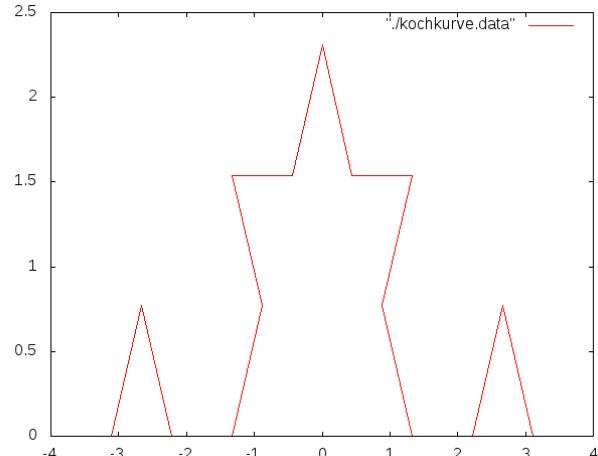
## Hinweise zu gnuplot

gnuplot ist ein Programm zur Datenanalyse und Visualisierung. Sie finden es auf den Pool Rechnern und können es mit dem Befehl gnuplot starten. Anschliessend erhalten sie eine Eingabeaufforderung. Hier können Sie Befehle eingeben um Daten darstellen zu lassen. Gnuplot interpretiert die Datendateien als Tabellen von Daten, wobei die Einträge einer Zeile durch Whitespace getrennt sind, z.B der Dateinhalt `data` und der von

```
gnuplot> plot "data" with lines
```

erzeugter Gnuplot Graph:

```
-4.000000  0.000000
-4.000000  0.000000
-3.111111  0.000000
-2.666667  0.769800
-2.222222  0.000000
-1.333333  0.000000
-0.888889  0.769800
-1.333333  1.539601
-0.444445  1.539601
-0.000000  2.309401
0.444444  1.539601
1.333333  1.539601
0.888889  0.769800
1.333333  0.000000
2.222222  0.000000
2.666667  0.769800
3.111111  0.000000
4.000000  0.000000
```



Den Inhalt verschiedener Dateien können sie darstellen, in dem Sie eine Komma-separierte Liste angeben, z.B:

```
gnuplot> plot "data1" with lines, "data2" with lines
```

Wenn Sie mehrere Daten in eine Datei schreiben wollen, können sie einfach verschiedene Spalten verwenden. Implizit geht gnuplot davon aus, dass Sie die ersten zwei Spalten verwenden können, sie können aber mit `using` explizit angeben, welche darzustellen sind:

```
gnuplot> plot "data1" using 2:3 with lines, "data2" using 1:3 with lines
```

Weitere Information erhalten direkt in Gnuplot durch den Aufruf von `help` oder auf der Homepage <http://www.gnuplot.info/>. Desweiteren hat die FU Berlin ihre Kursunterlagen unter <http://userpage.fu-berlin.de/~voelker/gnuplotkurs/gnuplotkurs.html> online gestellt.

Auf der Vorlesungshomepage finden Sie ausserdem das Programmskelett aus der Übungsstunden, in welchem bereits eine Ausgabe im Gnuplot Format implementiert ist.