

Projekt 1 – Jacobi- und Gauß-Seidel-Verfahren

Sarah Lutteropp und Johannes Sailer

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung



Aufgabenstellung

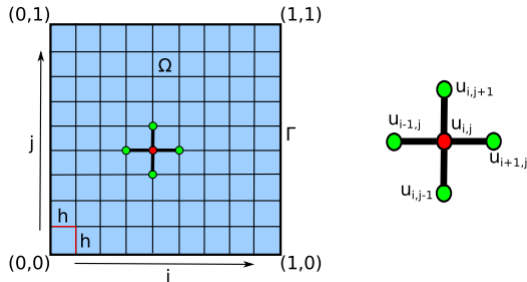
Mathematischer Hintergrund

Parallelisierung

Experimentelle Auswertung

Fazit

Approximation von Stoffkonzentrationen



Löse $Au = b$

TODO

$$\frac{\sum_{i,j} |u_{i,j}^{(k)} - u_{i,j}^{(k-1)}|}{size * size} \leq TOL$$

Vorteile

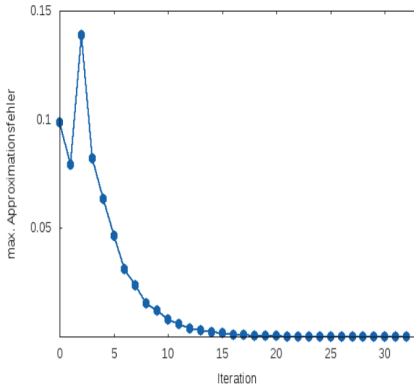
- Sprunglos
- Implementierung mit
`#pragma omp reduce`

Nachteile

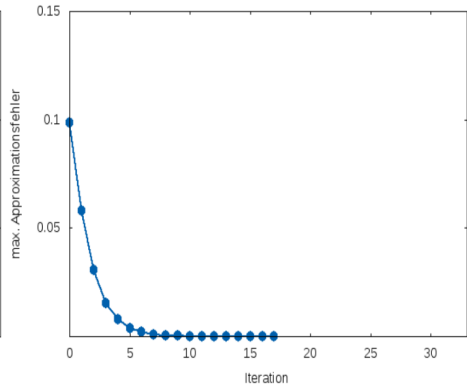
- Maximum der Differenzen
wäre exakter

Beide Verfahren konvergieren.

Jacobi-Verfahren für $h=1/4$



Gauß-Seidel-Verfahren für $h=1/4$



Keine Abhängigkeiten innerhalb einer Iteration

```
#pragma omp parallel for private(j, i) reduction(+:diff) collapse(2)
for (j = 1; j < size - 1; j++)
{
    for (i = 1; i < size - 1; i++)
    {
        a1[CO(i,j)] = a0[CO(i, j - 1)]
                      + a0[CO(i - 1, j)]
                      + a0[CO(i, j + 1)]
                      + a0[CO(i + 1, j)]
                      + functionTable[CO(i, j)];
        a1[CO(i,j)] *= 0.25;

        diff += fabsf(a1[CO(i,j)] - a0[CO(i,j)]);
    }
}
```

Zusätzliche Optimierung: SSE-Vektorinstruktionen

TODO: SSE-Bild

- Abhängigkeiten innerhalb einer Iteration:

$$u_{i,j}^{k+1} = \frac{1}{4} u_{i,j-1}^{k+1} + u_{i-1,j}^{k+1} + u_{i,j+1}^k + u_{i+1,j}^k + h^2 f(x_i, y_j)$$

- 1. Möglichkeit: Wavefront

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 2 | 3 | 4 | 5 |
| 3 | 4 | 5 | 6 |
| 4 | 5 | 6 | 7 |

Nachteile Wavefront

- Schlecht für Cache

| | | | | | | | |
|----|----|----|----|----|----|---|---|
| 0 | 1 | 2 | 3 | 0 | | | |
| 4 | 5 | 6 | 7 | 4 | 1 | | |
| 8 | 9 | 10 | 11 | 8 | 5 | 2 | |
| 12 | 13 | 14 | 15 | 12 | 9 | 6 | 3 |
| | | | | 13 | 10 | 7 | |
| | | | | 14 | 11 | | |
| | | | | 15 | | | |

- Aufwändige Berechnung der Indizes
- Geringe Parallelität bei kleinen Diagonalen
- Allgemein großer Overhead

TODO

Auswertung ohne Abbruchkriterium

TODO

TODO

TODO