

# Parallele Numerik - Aufgabe 1

Rebecca Seelos, Alexander Längen, Joshua

25. Juni 2019

# Aufgabe 1 - Teil a

Die Reihenfolge in welcher die IDs der Threads ausgegeben werden ist jedes Mal unterschiedlich und ist auch nicht vorherzusagen.

# Aufgabe 1 - Teil b

-	Number of Threads	N	Time (s)	Speedup
sequential	1	1000	0.002	-
	1	$10^7$	0.239	-
	1	$10^8$	2.388	-
	1	$10^9$	23.642	-
	1	$10^{10}$	239.508	-
manual	2	1000	0.002	1
	2	$10^7$	0.634	-
	2	$10^8$	6.461	-
	2	$10^9$	52.989	-
	4	$10^9$	$\sim 120$	-

# Aufgabe 1 - Teil b

-	Number of Threads	N	Time (s)	Speedup
sequential	1	1000	0.002	-
	1	$10^7$	0.239	-
	1	$10^8$	2.388	-
	1	$10^9$	23.642	-
	1	$10^{10}$	239.508	-
reduction	2	$10^7$	0.121	1.9
	2	$10^{10}$	118.204	2
	4	$10^7$	0.062	3.9
	4	$10^{10}$	59.183	4
	8	$10^{10}$	29.689	8.0
	16	$10^{10}$	29.674	8.1
	32	$10^{10}$	29.551	8.1

# Aufgabe 1 - Teil c

Auflösung N	No. Threads	Time (s)	Speedup
1000	1	4.411	-
1000	2	2.228	1.980
1000	4	1.122	3.931
1000	8	0.563	7.835
1000	16	0.567	7.780
2000	1	17.625	-
2000	2	8.896	1.981
2000	4	4.493	3.923
2000	8	2.254	7.819
2000	16	2.248	7.840

# Aufgabe 1 - Teil c

Auflösung N	No. Threads	Time (s)	Speedup
4000	1	71.761	-
4000	2	35.559	2.010
4000	4	17.952	3.997
4000	8	8.978	7.991
4000	16	8.980	7.991
5000	1	110.143	-
5000	2	55.564	1.982
5000	4	27.951	3.94
5000	8	13.995	7.870
5000	16	14.020	7.856

## Aufgabe 2 - Teil a

1. Race condition auf dem Array a über Index i
2. Threads existieren in gesamter paralleler Region
3. Hier ist die Variable x zunächst global definiert und wird somit implizit zwischen den Threads geshared.
4. f ist global definiert und wird durch jeden Thread private gesetzt.
5. ?

## Aufgabe 2 - Teil b

Werden Matrizen in C zeilenweise im Speicher hinterlegt, ist es zur optimalen Ausnutzung von Caching-Effekten ideal, wenn auch zeilenweise über Einträge der Matrix iteriert wird.



## Aufgabe 2 - Teil c

Die Arbeitspakete pro Thread werden statisch vergeben.

## Aufgabe 3 - Teil a

- ▶ **Speedup:**  $S(n) = T(1)/T(n)$ .  
Der Zusammenhang zwischen serieller und paralleler Ausführungszeit eines Programmes.
- ▶ **Effizienz:** Die Effizienz  $E(n) = S(n)/n$  gibt die relative Verbesserung der Verarbeitungsgeschwindigkeit an.
- ▶ **Auslastung:**  $R(n)/(n * T(n))$ . Gibt an, wie viele Operationen (Tasks) jeder Prozessor im Durchschnitt pro Zeiteinheit ausgeführt hat.
- ▶ **Mehraufwand:**  $R(n) = P(n)/P(1)$ . Beschreibt den bei einem Multiprozessorsystem erforderlichen Mehraufwand für die Organisation, Synchronisation und Kommunikation der Prozessoren.

Aufgabe 1  
○○○○○

Aufgabe 2  
○○○

Aufgabe 3  
○●○○

Aufgabe 4  
○○○

Aufgabe 5  
○○

Aufgabe 6  
○○

Aufgabe 7  
○○

## Aufgabe 3 - Teil b

Aufgabe 1  
○○○○○

Aufgabe 2  
○○○

Aufgabe 3  
○○●○

Aufgabe 4  
○○○

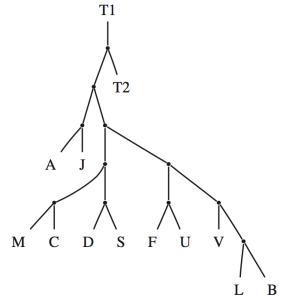
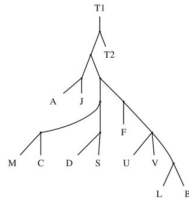
Aufgabe 5  
○○

Aufgabe 6  
○○

Aufgabe 7  
○○

## Aufgabe 3 - Teil c

# Bilder Beispiel



**Abbildung:** copied from  
*Evaluating methods for  
 computer-assisted stematology  
 using artificial benchmark data sets*

# Aufgabe 4 - Teil a

Alle Programmdurchläufe für die Messungen hatten eine Fehlerschranke von 0.000001.

- ▶ **h**: Verfeinerung  $h = \frac{1}{2^l}$
- ▶ **l**:  $l \in \mathbb{N}$

	l=4; h=1/16	l=5; h=1/32	l=6; h=1/64
	0.050s	1.429s	37.296s
	0.041s	1.426s	37.014s
	0.040s	1.392s	37.061s
	0.037s	1.428s	37.416s
	0.051s	1.424s	37.295s
Summe	= 0.044s	= 1.420s	= 37.216s

## Aufgabe 4 - Teil b

- ▶ Innere Schleifendurchgänge direkt voneinander Datenabhängig
- ▶ Schleifen müssen in korrekter Reihenfolge abhängig von äußerer Schleife ablaufen
- ▶ Naive Parallelisierung der äußersten Schleife liefert falsches Ergebnis

## Aufgabe 4 - Teil c

Vorgehensweise zur optimalen Parallelisierung:

- ▶ Innere Schleifen zur Summenberechnung parallel berechnen
- ▶ Matrix a und Vector u explizit shared
- ▶ “+“-Reduction auf Summenvariable firstsum bzw. secondsum

	Laufzeit (seriell)	Laufzeit (parallel)	SpeedUp	Efficiency
$l=4; h=1/16$	0.044s	0.242s	0.182	0.004
$l=5; h=1/32$	1.420s	1.154s	1.231	0.026
$l=6; h=1/64$	37.216s	26.884s	1.384	0.029



## Aufgabe 5 - Teil a + b

- ▶  $u$ : Eine Lösung  $u(x, y)$  muss mind. zweifach differenzierbar sein
- ▶  $f$ :  $f(x, y)$  muss definiert sein auf  $\Omega$
- ▶  $\Gamma$ : ist der Rand von  $\Omega$  und es gilt:  $\Gamma \subset \Omega$

Function:

$$f(x, y) = (N^2 + M^2) * 4 * \pi^2 * \sin(2 * M * \pi * x) * \sin(2 * N * \pi * y)$$

## Aufgabe 5 - c - Visualisiert

- Bei der Lösungsmethodik handelt es sich um eine  $h - FEM$

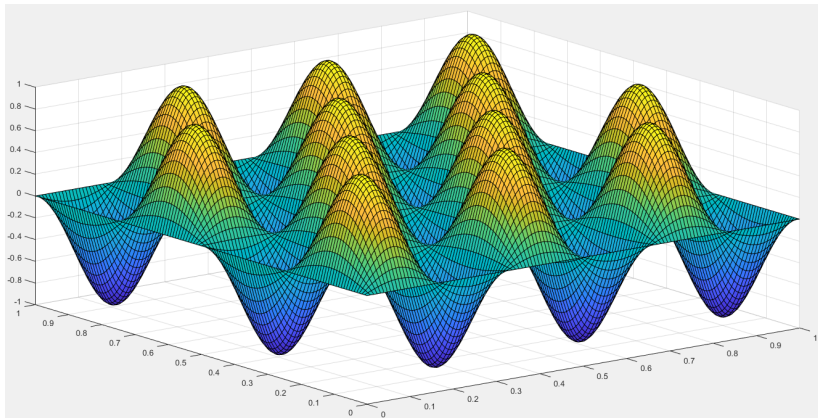


Abbildung:  $l=7$ ;  $h=1/128$ ;  $M=3$ ;  $N=2$

# Aufgabe 6 - Speedup

-	Aufgabe 5(s)	Aufgabe 6(s)	Speedup
l=5; h=1/32	1.450	0.028	56.96
	1.420	0.021	
	1.371	0.026	
	1.475	0.026	
	1.408	0.026	
	=1.424	=0.025	
l=6; h=1/64	15.427	0.140	98.488
	12.546	0.148	
	12.561	0.089	
	12.766	0.142	
	13.081	0.155	
	=13.276	=0.135	
l=7; h=1/128	107.183	1.289	81.113
	105.581	1.188	
	104.562	1.175	

# Aufgabe 6 - Visualisiert

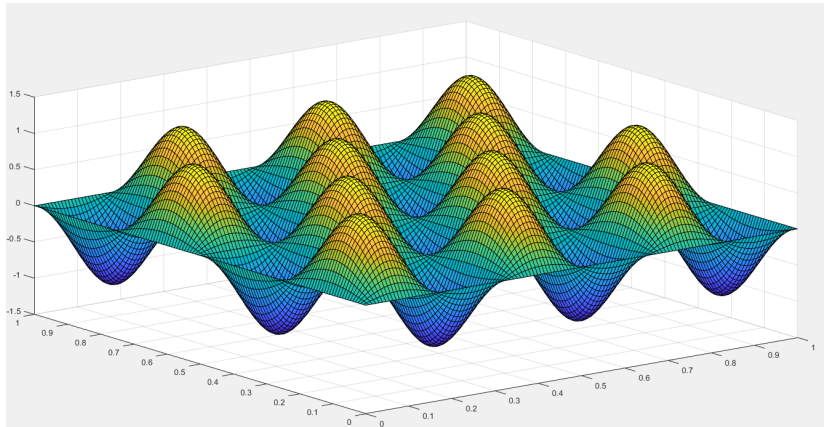


Abbildung: GMRES;  $l=7$ ;  $h=1/128$ ;  $M=3$ ;  $N=2$

# Aufgabe 7

Die Reihenfolge in welcher die IDs der Threads ausgegeben werden ist jedes Mal unterschiedlich und ist auch nicht vorherzusagen.

# References

- ▶ Bein, Thomas. Textkritik: eine Einführung in Grundlagen germanistisch-mediävistischer Editionswissenschaft: Lehrbuch mit Übungsteil. Peter Lang, 2011.
- ▶ Roos, Teemu, Tuomas Heikkilä, and Petri Myllymäki. 'A compression-based method for stemmatic analysis.' *Frontiers in Artificial Intelligence and Applications* 141 (2006): 805.
- ▶ Roos, Teemu, and Yuan Zou. 'Analysis of textual variation by latent tree structures.' *Data Mining (ICDM)*, 2011 IEEE 11th International Conference on. IEEE, 2011.
- ▶ Roos, Teemu, and Tuomas Heikkilä. 'Evaluating methods for computer-assisted stemmatology using artificial benchmark data sets.' *Literary and Linguistic Computing* 24.4 (2009): 417-433.