

# Laufzeitoptimierung in LARSIM

Oliver Gronz

Fachbereich Informatik  
Fachhochschule Trier

-

Arbeitsgruppe Modellbildung und Simulation  
Fachbereich VI - Physische Geographie  
Universität Trier

17. Februar 2009



**FACHHOCHSCHULE TRIER**

Hochschule für Technik, Wirtschaft und Gestaltung  
University of Applied Sciences

**Informatik**

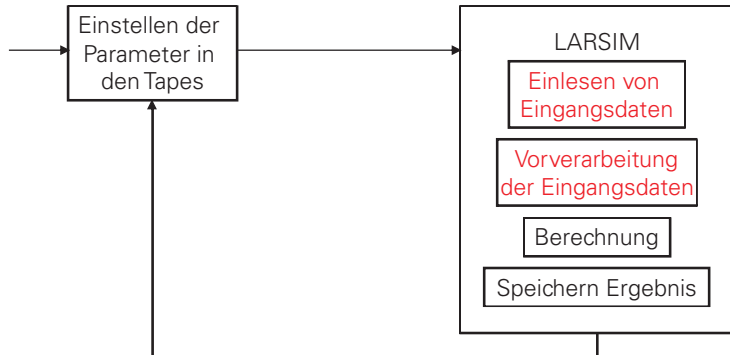


**Universität Trier**

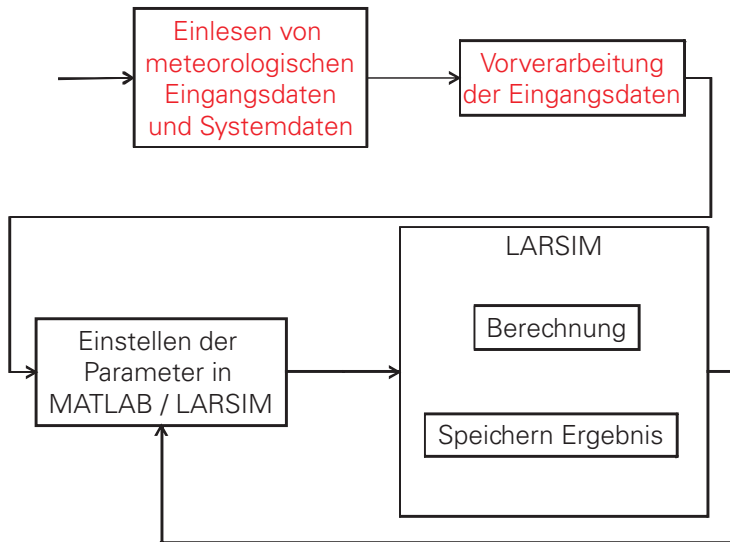
- 1 Einleitung
- 2 Optimierung des Programmablaufs
- 3 Effiziente Implementierung von Modellkonzepten
- 4 Parallelisierung
- 5 Zusammenfassung und Diskussion

- Im Rahmen des Projekts "Räumlich verteilte Parametrisierung des Bodenmoduls in LARSIM" große Anzahl von Modellläufen notwendig:
  - z. B. bei Sensitivitätsanalysen / Monte-Carlo-Simulationen
  - z. B. bei Optimierung unter Verwendung genetischer Algorithmen
- ⇒ Notwendigkeit einer effizienten Simulation

# Momentaner Ablauf



# Ablauf mit MATLAB-LARSIM



- Speicherung von Niederschlag auf den Blattoberflächen
- Modellannahme:

$$K_{Inz} = 0,2mm \cdot LAI$$

# Blattflächenindizes

Landnutzung	Blattflächenindex LAI											
	Jan.	Feb.	März	Apr.	Mai	Juni	Juli	Aug.	Sep.	Okt.	Nov.	Dez.
versiegelt*	10	10	10	10	10	10	10	10	10	10	10	10
Acker**	0,3	0,3	0,3	1,0	2,3	3,7	3,8	3,5	2,4	1,2	0,3	0,3
Weinbau	1,0	1,0	1,0	1,5	2,0	3,5	4,0	4,0	4,0	1,5	1,0	1,0
Intensivobstbau	2,0	2,0	2,0	2,0	3,0	3,5	4,0	4,0	4,0	2,5	2,0	2,0
Brache (bewachsen)	2,0	2,0	3,0	4,0	5,0	5,0	5,0	5,0	5,0	3,0	2,5	2,0
unversiegelt, unbewachsen	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
intensives Grünland	2,0	2,0	2,0	3,0	3,5	4,0	4,0	4,0	3,5	3,0	2,5	2,0
Feuchtflächen	2,0	2,0	3,0	4,0	5,0	5,0	5,0	5,0	5,0	3,0	2,5	2,0
extensives Grünland	2,0	2,0	2,0	3,0	3,5	4,0	4,0	4,0	3,5	3,0	2,5	2,0
locker baumbestanden	2,0	2,0	3,0	4,5	5,5	5,5	5,5	5,5	5,5	4,0	2,5	2,0
Nadelwald	11	11	11	11	11	11	11	11	11	11	11	11
Laubwald	0,5	0,5	1,5	4,0	7,0	11	12	12	11	8,0	1,5	0,5
Mischwald	3,0	3,0	4,0	6,0	8,0	11	11,5	11,5	11	9,0	4,0	3,0
Wasser	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

\* Fiktiver Wert, um Benetzungs- und Muldenverluste auf versiegelten Flächen zu berücksichtigen

\*\* Mittelwert für verschiedene Anbausorten

## Variante 1

```
1 for i = 1:numberOfElements
2   for j = 1:elements.numberOfSubElements(i)
3     elements.subElement(i).capacityOfInterceptionStorage(j) = ...
4       0.2 * leafAreaIndices(elements.subElement(i).landUse(j), ...
5         currentMonth);
6   end
7 end
```



- Dauer der Berechnung aller Kapazitäten aller Teileinzugsgebiete aller Elemente des gesamten Nahe-Gebietes für 1 Jahr:

27 Minuten

## Variante 2

```
1 for i = 1:numberOfElements
2   elements.subElement(i).capacityOfInterceptionStorage = ...
3     0.2 * leafAreaIndices(elements.subElement(i).landUse,...
4     currentMonth);
5 end
```

- Dauer der Berechnung aller Kapazitäten aller Teileinzugsgebiete aller Elemente des gesamten Nahe-Gebietes für 1 Jahr:

**3 Minuten**

## Variante 3

```
1 leafAreaIndices = 0.2 * leafAreaIndices;
2
3 for i = 1:numberOfElements
4     elements.subElement(i).capacityOfInterceptionStorage = ...
5         leafAreaIndices(elements.subElement(i).landUse, currentMonth);
6 end
```

- Dauer der Berechnung aller Kapazitäten aller Teileinzugsgebiete aller Elemente des gesamten Nahe-Gebietes für 1 Jahr:

2 Minuten

- *(Lediglich monatliche Berechnung)*
- *(Berechnung aller Teileinzugsgebiete aller Elemente in einer Multiplikation)*
- *(Berechnung aller Zeitschritte aller Teileinzugsgebiete aller Elemente in einer Multiplikation)*
- Verwendung von `parfor`

# Was wird z. B. bei einer Monte-Carlo-Simulation berechnet?

- Für alle Parameterkonfigurationen werden alle Zeitschritte berechnet
- Für alle Zeitschritte werden alle Elemente berechnet
- Für alle Elemente werden alle Teileinzugsgebiete berechnet
- Für alle Teileinzugsgebiete werden alle Komponenten des Wasserhaushalts simuliert
- Für alle Elemente wird die Translation und Retention im Gerinne simuliert

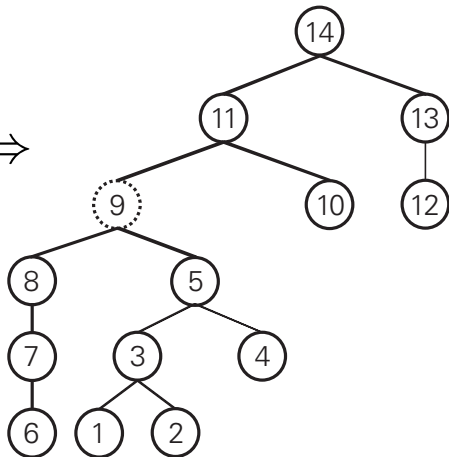
# Was kann parallel berechnet werden?

## Pseudocode

```
1 parfor alle Parameterkonfigurationen (z. B. 10.000)
2   for alle Zeitschritte
3     for alle Elemente
4       parfor alle Subelemente dieses Elements
5         Simuliere Wasserhaushalt...
6       Simuliere Translation und Retention
```



# Parallelisierung von unabhängigen Elementen



6. Iterationsschritt

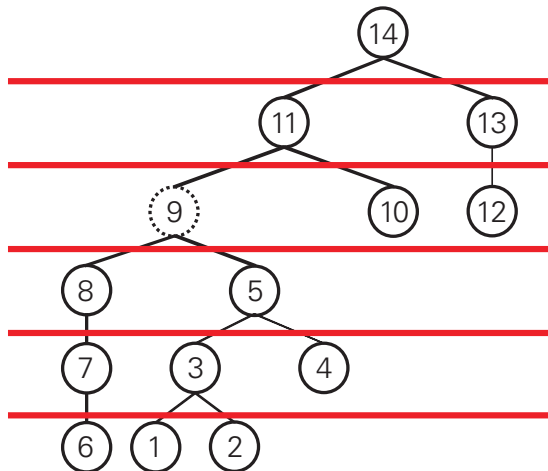
5. Iterationsschritt

4. Iterationsschritt

3. Iterationsschritt

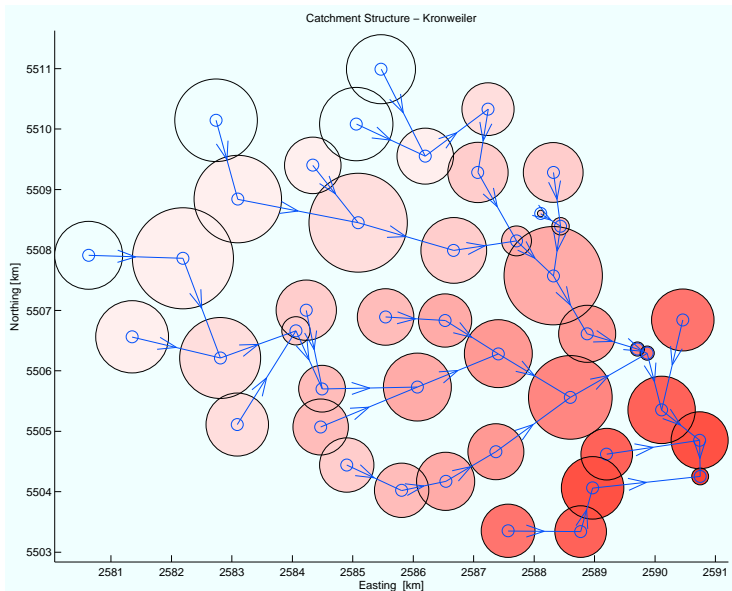
2. Iterationsschritt

1. Iterationsschritt



⇒ **6 Schritte anstatt 14**

# Beispielvisualisierung der Level



- Hängt von "Form" des Einzugsgebiets ab:
- Im schlechtesten Fall: keiner (eigentlich sogar ein Nachteil)
- Im besten Fall beträgt der maximale Level für  $n$  Elemente  $\lceil \log_2(n+1) \rceil - 1$
- Im Nahe-Gebiet: im Mittel Faktor 19

## Also: Was kann parallel berechnet werden? (Version 2)

### Pseudocode

```
1 parfor alle Parameterkonfigurationen (z. B. 10.000)
2   for alle Zeitschritte
3     parfor alle Elemente
4       Simuliere Wasserhaushalt...
5   for alle Level
6     Simuliere Translation und Retention für Elemente dieses Levels
```

- Zielkonflikte bei Implementierung:
  - Laufzeit
  - Lesbarkeit
  - Speicherbedarf
  - Entwicklungsaufwand
- Konkreter Vorteil: Abhängig von Umgebung:
  - "Alte Eingangsdaten" bzw. einmaliger Lauf: kein Vorteil
  - Schneller Datenträger, Single Core: ca. Faktor 2-5
  - Cluster, mehrere Kerne: ca. Faktor 20-50