



TECHNISCHE
UNIVERSITÄT
DRESDEN

Entwicklung eines Modelica Compiler BackEnds für große Modelle

Jens Frenkel

Dresden, 19.02.14

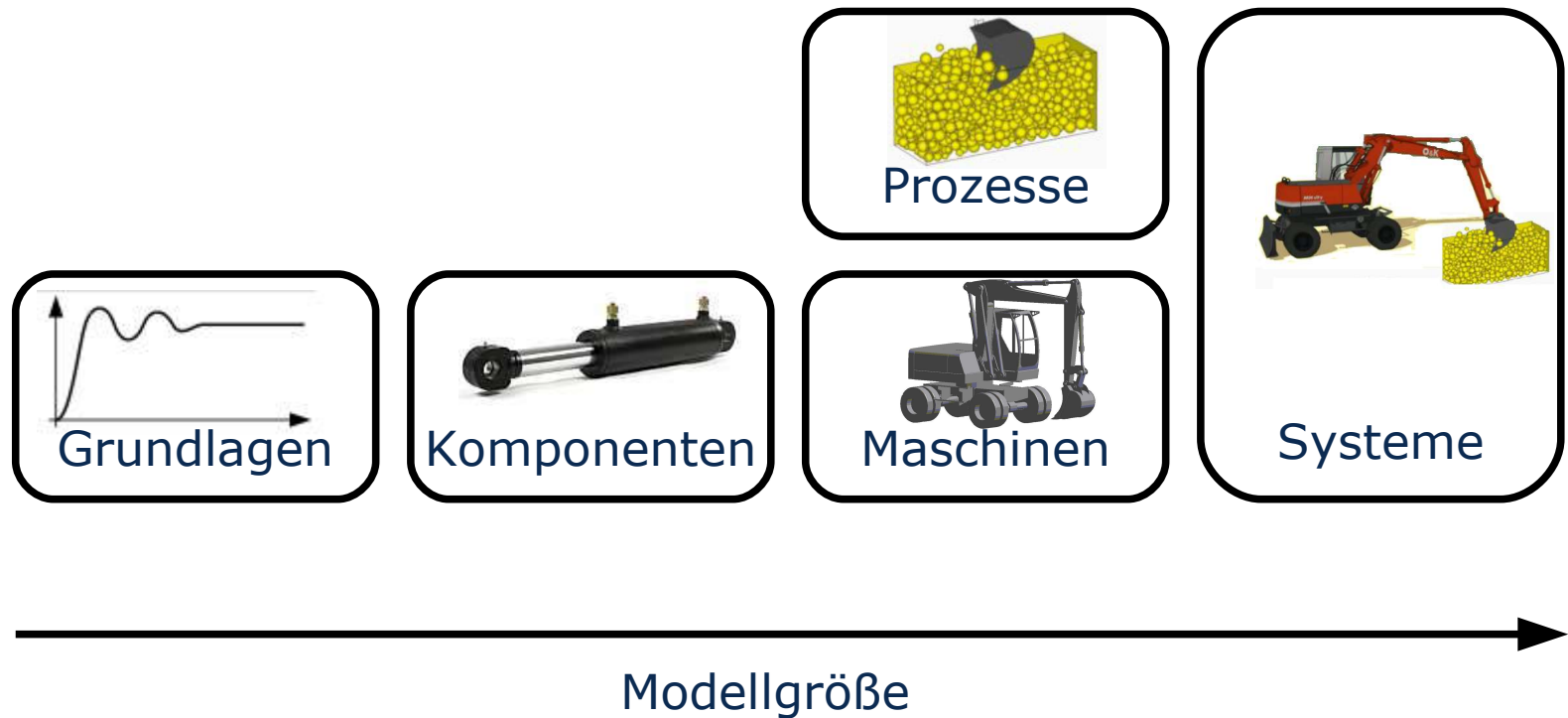


DRESDEN
concept
Exzellenz aus
Wissenschaft
und Kultur

1. Einführung und Motivation
2. Definition und Aufbau Compiler BackEnd
3. Implementation und Entwicklung Compiler BackEnd
4. Messungen an praxisrelevanten Beispielen
5. Zusammenfassung und weiterführende Aufgaben

1 Einführung und Motivation

Modellbildung und Simulation - Trend

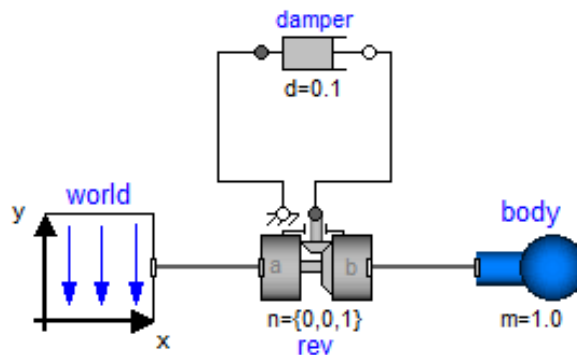


1 Einführung und Motivation

Modelica

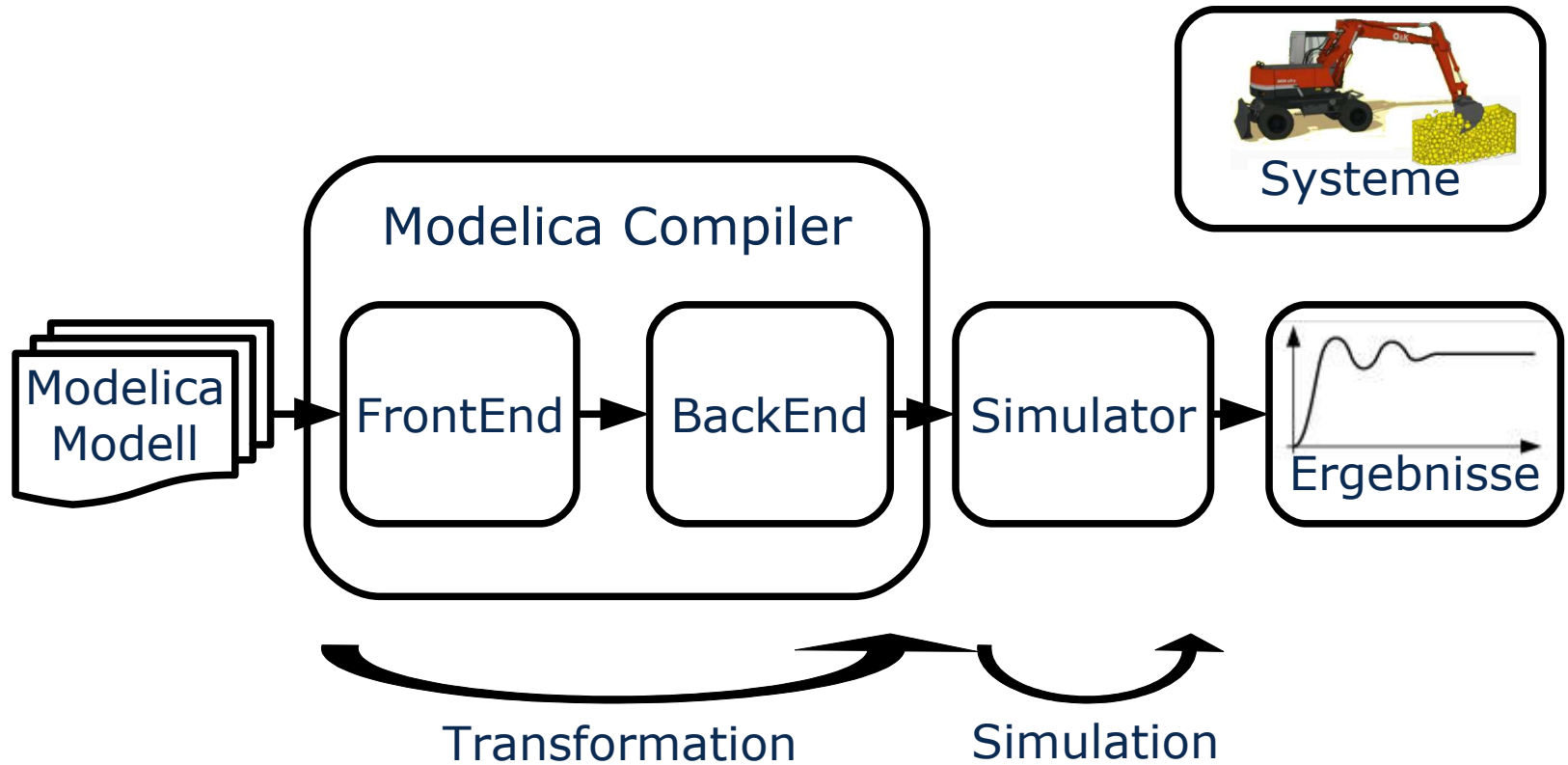


- Modellierungssprache
 - Gleichungsbasiert
 - Objektorientiert
- Wiederverwendbarkeit
- Wartungsfreundlich



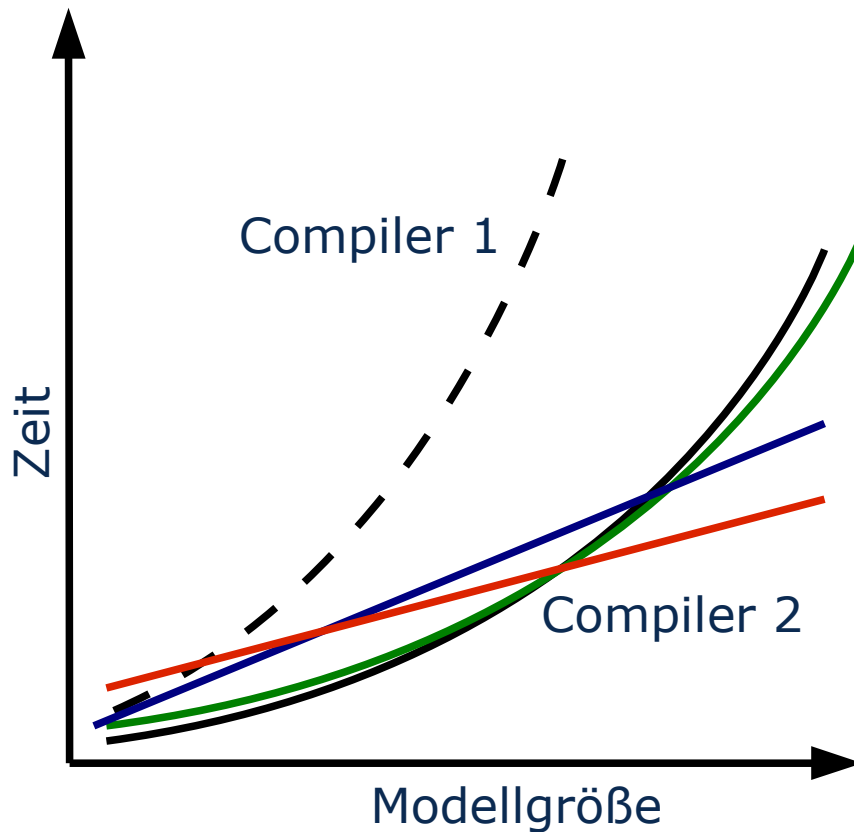
```
model Pendel
parameter Real m=1,L=1;
constant Real g=9.81;
Real x(start=L),y(start=0);
Real dx,dy;
Real Fo;
equation
der(x) = dx;
der(y) = dy;
m*der(dx) = -x*Fo/L;
m*der(dy) = -m*g-Fo*y/L;
x^2+y^2=L^2;
end Pendel;
```

1 Einführung und Motivation



1 Einführung und Motivation

Wie skalieren Modelica Compiler?



- FrontEnd
- BackEnd
- Erstellung Simulator

1 Einführung und Motivation

Modelica Compiler skalieren nichtlinear

- Transformierbare Modellgröße ist praktisch begrenzt
- BackEnd
 - Skaliert nichtlinear
unabhängig von Modellierungsweise

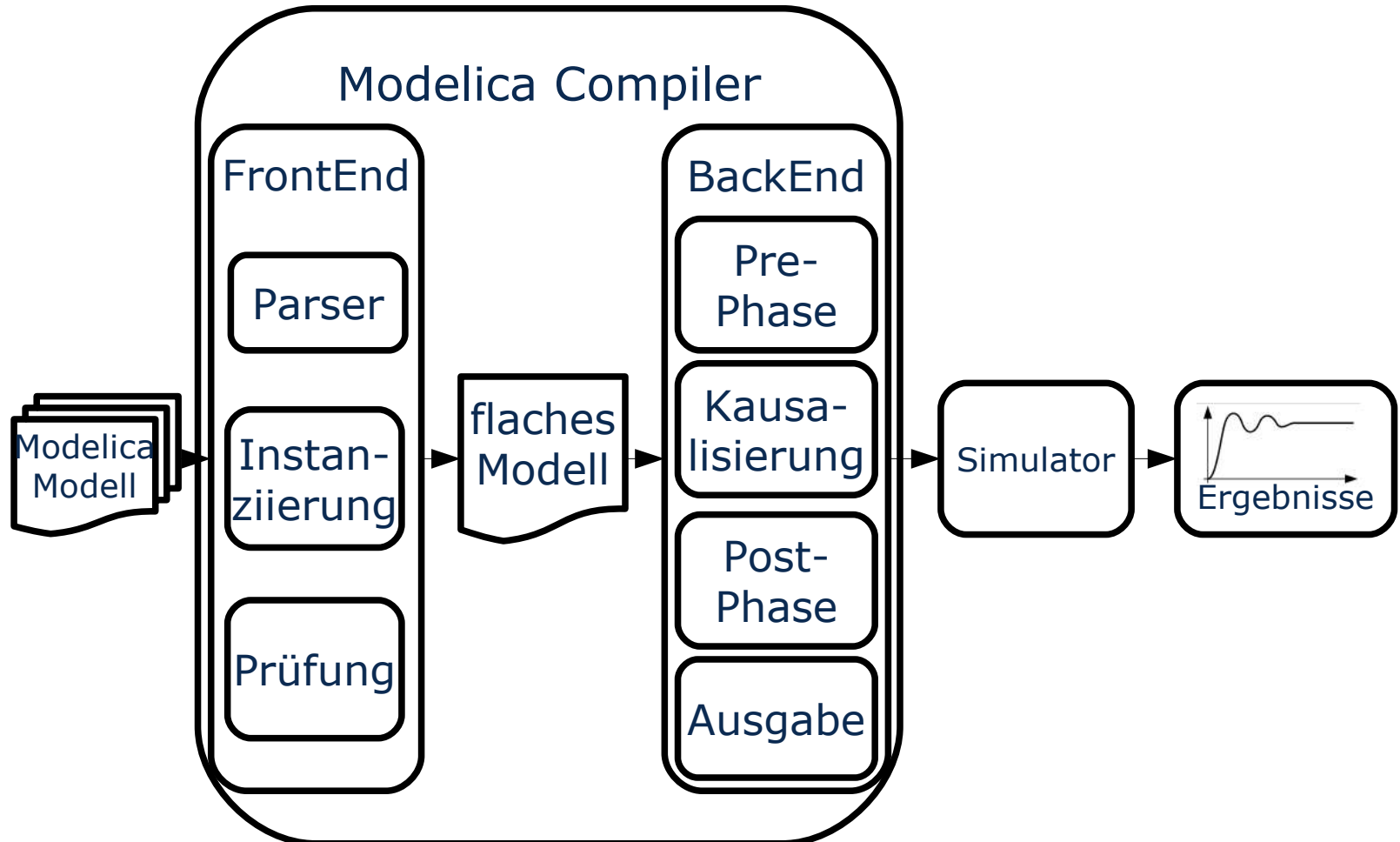
Ziele:

- Ursachen für nichtlinearen Verlauf des BackEnds
- Ist linearer Verlauf für BackEnd möglich?

OpenModelica Compiler



2 Definition und Aufbau Compiler BackEnd



- Akausales System → kausales System
- Symbolische Optimierung

2 Definition und Aufbau Compiler BackEnd

Kausalisieren

Modelica Modell:

- Hybrides differential-algebraisches Gleichungssystem (DAE)
 - Zeitkontinuierlicher Anteil
 - Zeitdiskreter Anteil

$$F(x(t), \dot{x}(t), u(t), y(t), t, q(t_e), q_{pre}(t_e), p, c(t_e)) = 0$$

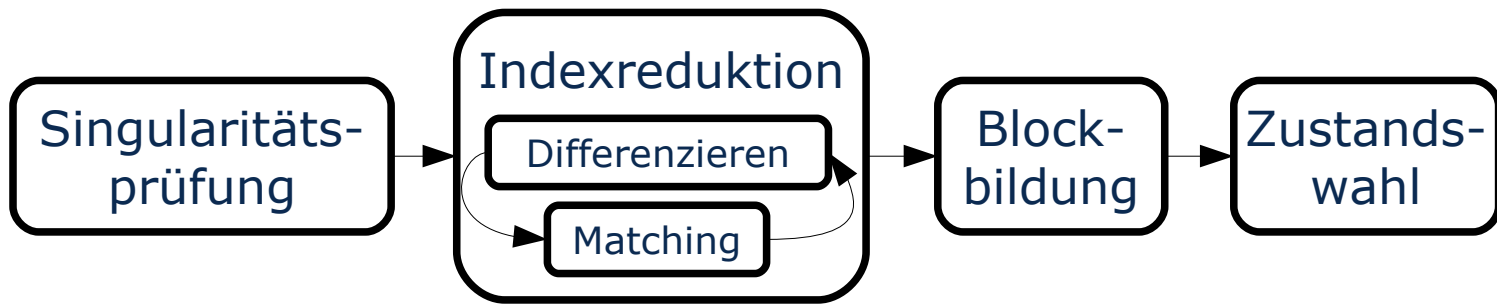
- Integrationsalgorithmus für gewöhnliche Differentialgleichung (ODE)
 - Allgemeines Interface

$$\dot{x} = f(x(t), t)$$



2 Definition und Aufbau Compiler BackEnd

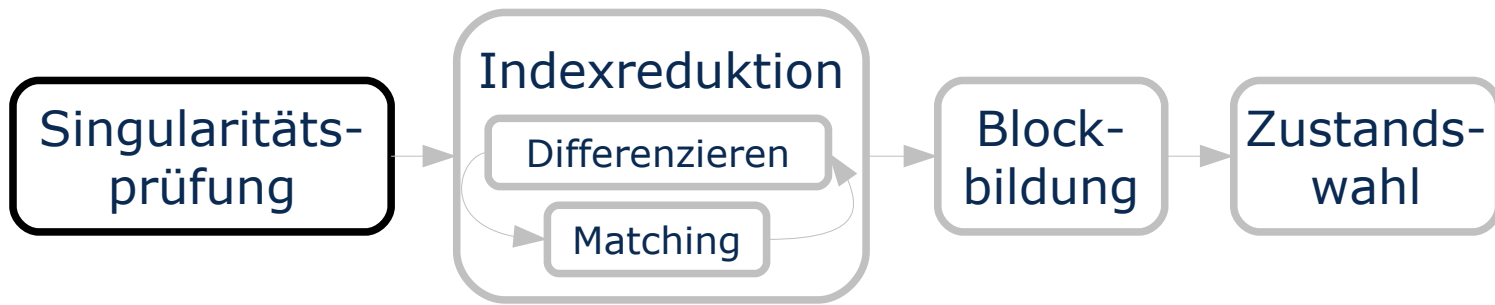
Kausalisieren



2 Definition und Aufbau Compiler BackEnd

Kausalisieren - Singularitätsprüfung

- Frühzeitig symbolisch strukturell singuläre System erkennen
- Terminierung Indexreduktion
- Matching Algorithmus



2 Definition und Aufbau Compiler BackEnd

Kausalisieren - Singularitätsprüfung

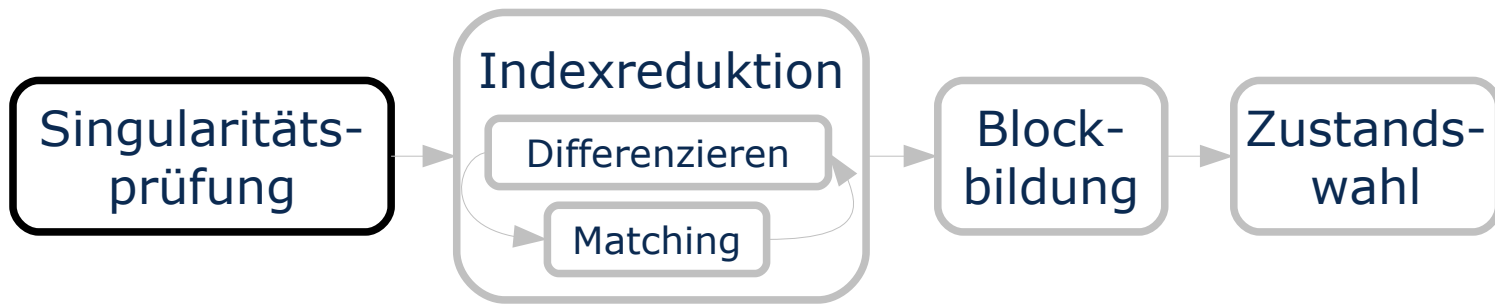
$$f_1(v_1, v_3, v_4) = 0$$

$$f_2(v_2, v_3) = 0$$

$$f_3(v_2, v_3) = 0$$

$$f_4(v_4) = 0$$

	v_1	v_2	v_3	v_4
f_1	1		1	1
f_2		1	1	
f_3		1	1	
f_4				1



2 Definition und Aufbau Compiler BackEnd

Kausalisieren - Singularitätsprüfung

$$f_1(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4) = 0$$

$$f_2(\mathbf{v}_2, \mathbf{v}_3) = 0$$

$$f_3(\mathbf{v}_2, \mathbf{v}_3) = 0$$

$$f_4(\mathbf{v}_4) = 0$$

	\mathbf{v}_1	\mathbf{v}_2	\mathbf{v}_3	\mathbf{v}_4
f_1	1		1	1
f_2		1	1	
f_3		1	1	
f_4				1



2 Definition und Aufbau Compiler BackEnd

Kausalisieren - Singularitätsprüfung

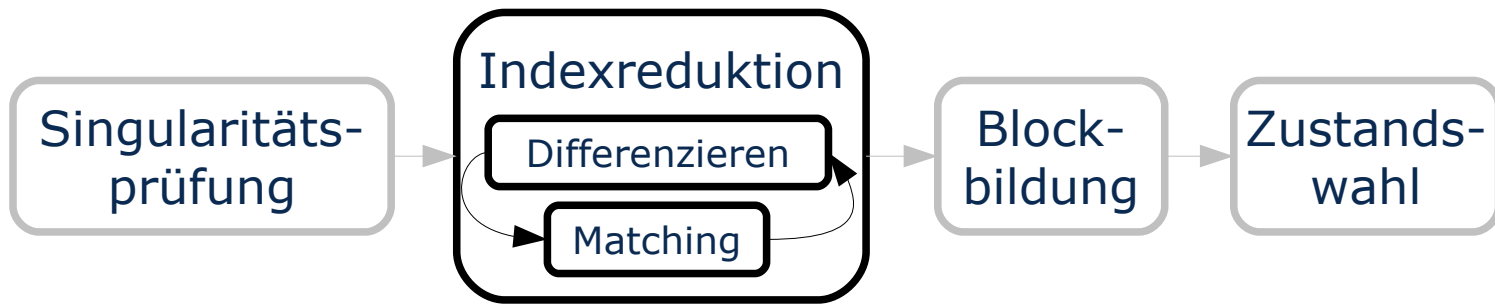
- Frühzeitig symbolisch strukturell singuläre Systeme erkennen
- Terminierung Indexreduktion
- Matching Algorithmus
- Im ungünstigsten Fall nichtlineare Laufzeit
- Matching Algorithmus PF+ erreicht lineare Laufzeit für praxisnahe Systeme



2 Definition und Aufbau Compiler BackEnd

Kausalisieren – Indexreduktion

- Numerische Löser für ODE-Systeme (Integrationsalgorithmus)
- Differentiationsindex $v_d=0$
- Pantelides Algorithmus
 - Matching Algorithmus
 - Minimal singuläres Set (MSS) differenzieren
- Lineare Laufzeit, wenn Differentiationsindex unabhängig von Systemgröße



2 Definition und Aufbau Compiler BackEnd

Kausalisieren – Blockbildung

	v_4	v_2	v_3	v_1
f_4	1			
f_3		1	1	
f_2		1	1	
f_1	1	1		1

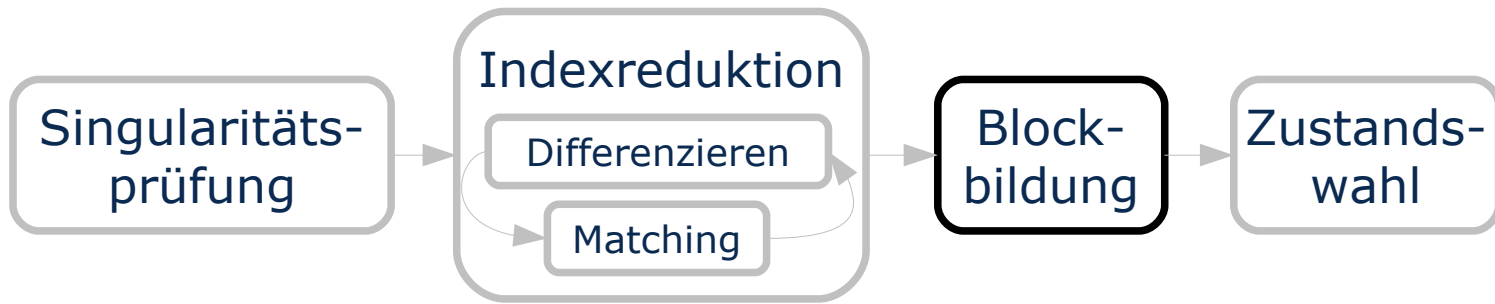
$$f_4(v_4)=0$$

$$f_3(v_2, v_3)=0$$

$$f_2(v_2, v_3)=0$$

$$f_1(v_1, v_3, v_4)=0$$

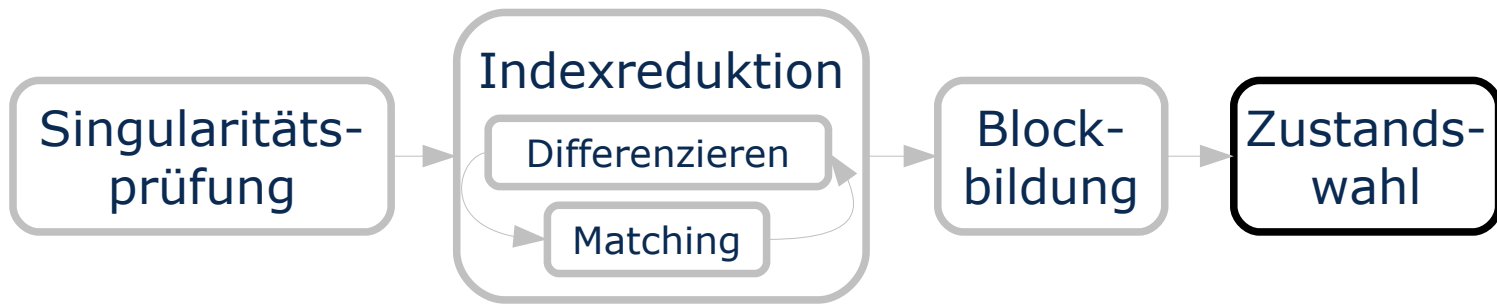
- Tarjans Algorithmus
 - Lineare Laufzeit



2 Definition und Aufbau Compiler BackEnd

Kausalisieren – Zustandswahl

- Indexreduziertes System
 - *Darunterliegende* ODE
 - Differenzierte Gleichungen (Zwangsbedingungen)
- Stabilisierung
 - Dummy Derivative Methode mit dynamischer Zustandswahl
- Lineare Komplexität für praktisch relevante Modelle



2 Definition und Aufbau Compiler BackEnd

Finden und Entfernen trivialer Gleichungen

- Grundlegender Optimierungsalgorithmus
- Triviale Gleichungen
 - Alias-Gleichungen
 - Zeitunabhängige Gleichungen
- Anteil trivialer Gleichungen an Beispielen aus der Modelica Standard Bibliothek 44%-73%
- Entfernen aller trivialen Gleichungen aus
 - Post-Phase: meist lineare Laufzeit
 - Pre-Phase: nichtlineare Laufzeit
- Zeitersparnis für Indexreduktion
- Iteration muss begrenzt werden
 - Nach 2. Durchlauf >97% entfernt

BackEnd

a=b;
a=-b;
Pre-Phase

Kausa-
lisierung

Post-Phase

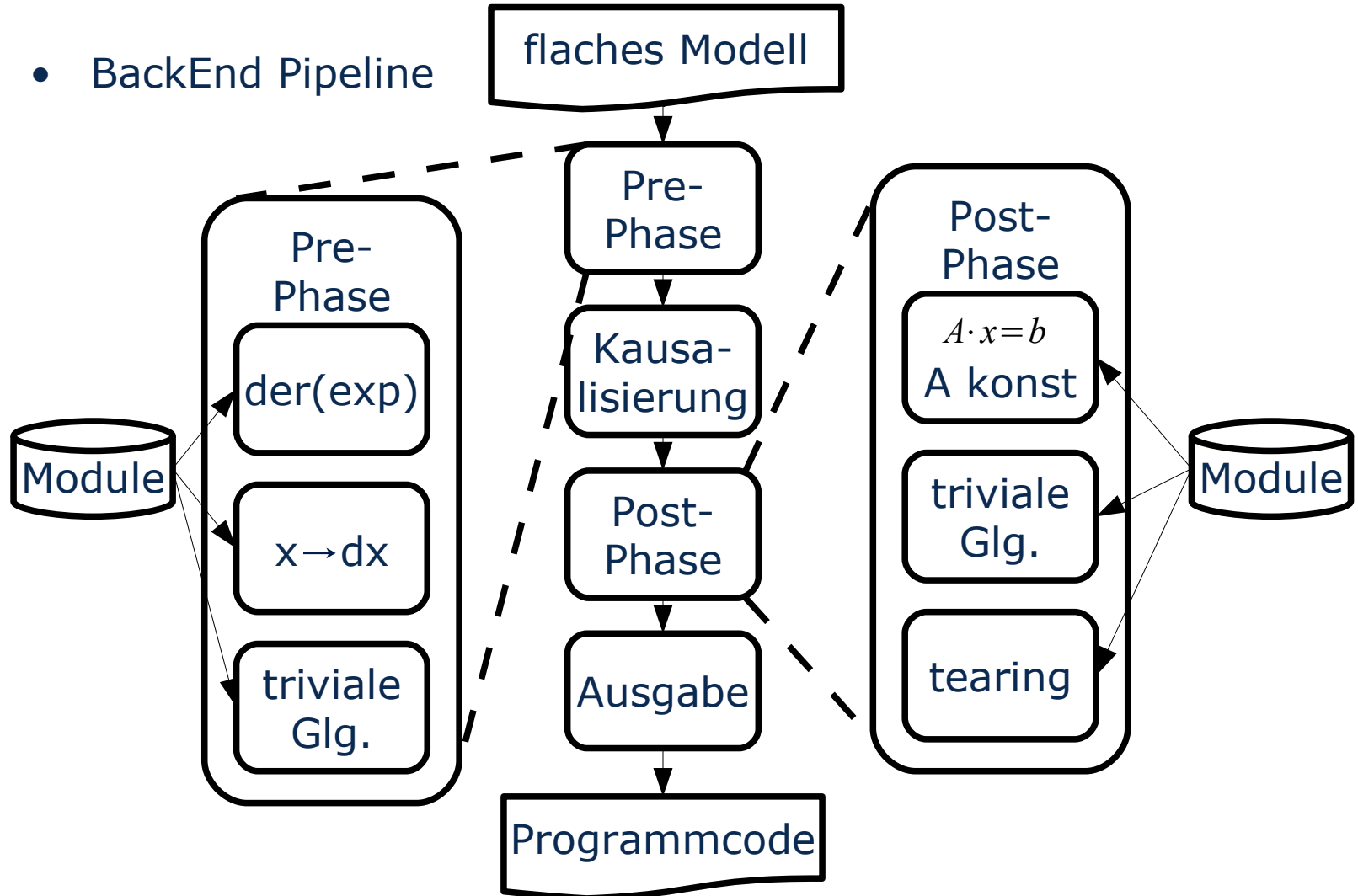
Phase

Ausgabe

u=2*pi*R;
x=sin(u);

3 Implementation und Entwicklung

- BackEnd Pipeline



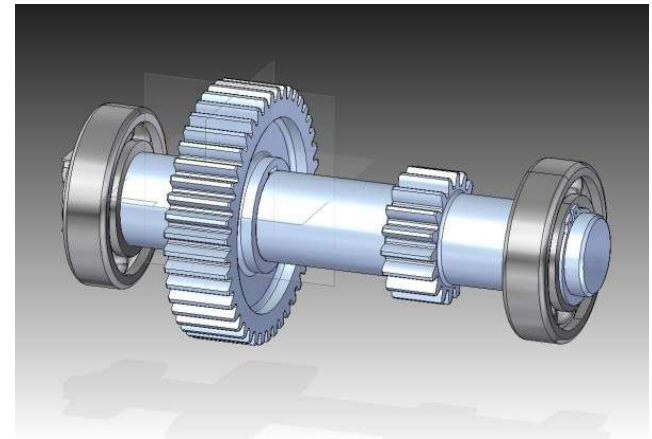
3 Implementation und Entwicklung

- BackEnd Pipeline
 - Flexible Strukturierung von Daten und Prozessen
- Algorithmen für Module
 - Implementierung
 - Vergleich
 - Optimierung
- Testumgebung
 - Konformität
 - Symbolik
 - Numerik
 - Fehlererkennung
 - Performance
 - Modelimark

4 Messungen an praxisrelevanten Beispielen

Modelle

- Skalierbar
- Synthetisch
- Praktisch relevant durch Einsatz als Teil eines Gesamtmodells
- **Eindimensionaler rotatorischer Mehrfachschwinger**
- Dreidimensionales Mehrfachpendel
- Ebene Verkettung von Viergelenken
- Elektrischer Leiter
- Radlader CASE 921E



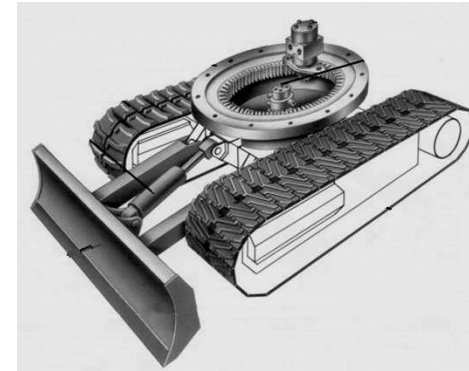
(a) Getriebewelle

Quellen: (a) Florian Ditzer: <http://florianditzer.wordpress.com/getriebewelle/>

4 Messungen an praxisrelevanten Beispielen

Modelle

- Skalierbar
- Synthetisch
- Praktisch relevant durch Einsatz als Teil eines Gesamtmodells
- Eindimensionaler rotatorischer Mehrfachschwinger
- **Dreidimensionales Mehrfachpendel**
- Ebene Verkettung von Viergelenken
- Elektrischer Leiter
- Radlader CASE 921E



(a) Kettenfahrwerk



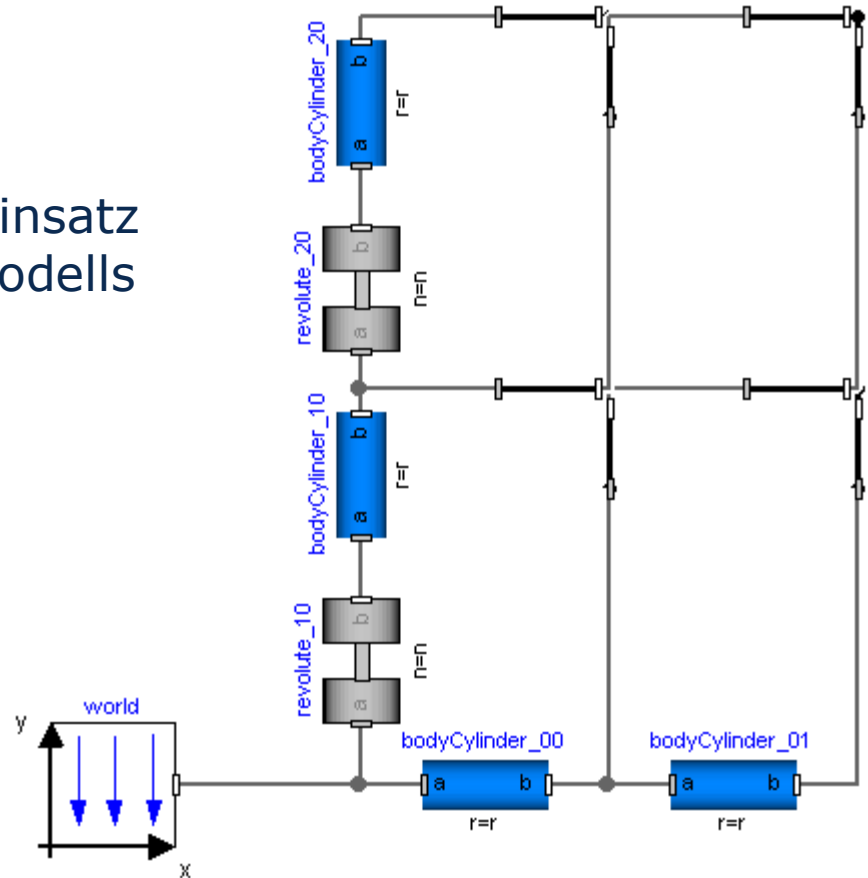
(b) Eimerkettenbagger

Quellen: (a) Dr.-Ing. D. Lemser: Hydraulikbagger auf Rädern oder Ketten Teil1, Taunusstein, Tiefbau 9/2003
www.baumaschine.de, (b) <http://www.die-weserhuette.de/eimerkettenbagger.htm>

4 Messungen an praxisrelevanten Beispielen

Modelle

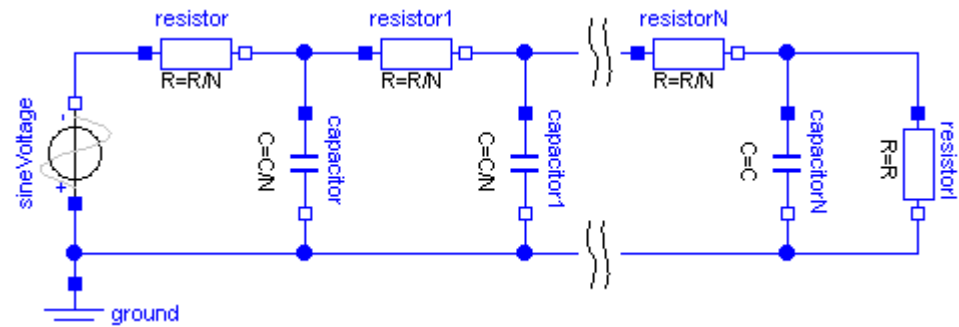
- Skalierbar
- Synthetisch
- Praktisch relevant durch Einsatz als Teil eines Gesamtmodells
- Eindimensionaler rotatorischer Mehrfachschwinger
- Dreidimensionales Mehrfachpendel
- **Ebene Verkettung von Viergelenken**
- Elektrischer Leiter
- Radlader CASE 921E



4 Messungen an praxisrelevanten Beispielen

Modelle

- Skalierbar
- Synthetisch
- Praktisch relevant durch Einsatz als Teil eines Gesamtmodells
- Eindimensionaler rotatorischer Mehrfachschwinger
- Dreidimensionales Mehrfachpendel
- Ebene Verkettung von Vierecken
- **Elektrischer Leiter**
- Radlader CASE 921E



4 Messungen an praxisrelevanten Beispielen

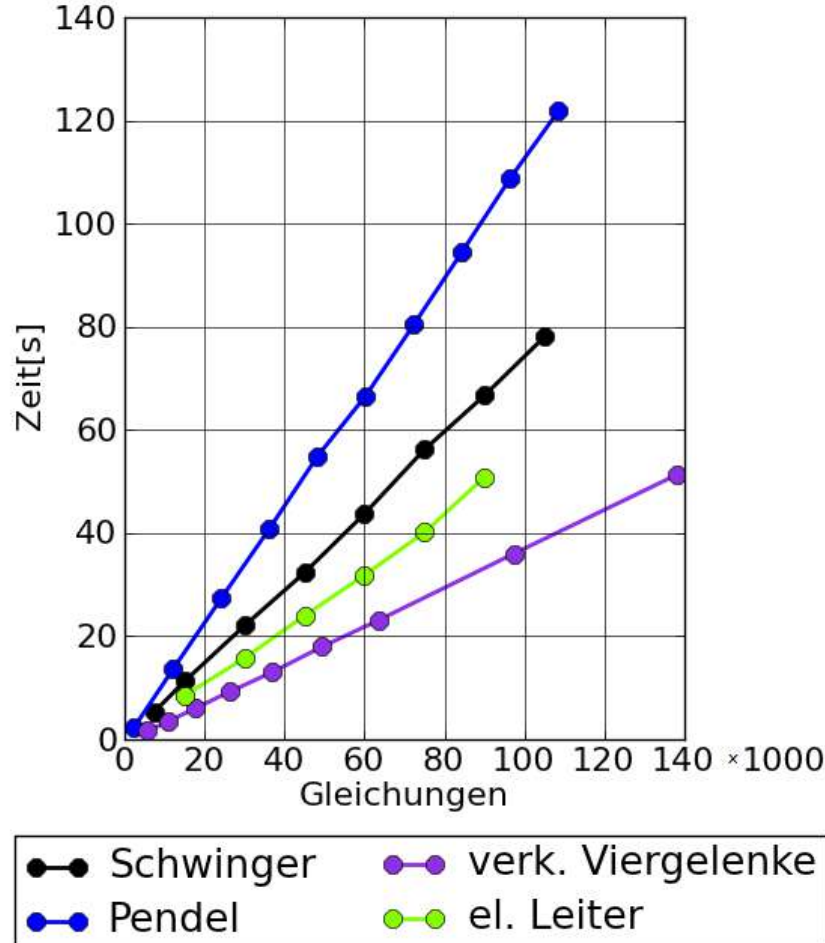
Modelle

- Skalierbar
- Synthetisch
- Praktisch relevant durch Einsatz als Teil eines Gesamtmodells
- Eindimensionaler rotatorischer Mehrfachschwinger
- Dreidimensionales Mehrfachpendel
- Ebene Verkettung von Viergelenken
- Elektrischer Leiter
- **Radlader CASE 921E**

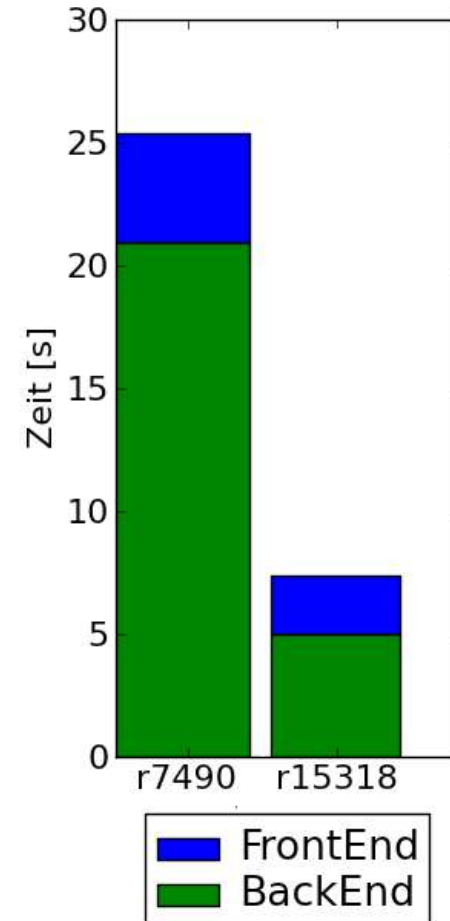


4 Messungen an praxisrelevanten Beispielen

Ergebnisse für Benchmark mit skalierbaren Beispielen



Vergleich OMC Versionen Radlader CASE 921E



5 Zusammenfassung

- Schritte der symbolischen Analyse detailliert analysiert und dokumentiert
- Linear skalierendes Modelica Compiler BackEnd
 - Mechanik und Elektrik
- Wesentlich beschleunigter Transformationsprozess
- Ursachen für nichtlinearen Verlauf des BackEnd
 - Ineffektiv oder ineffizient umgesetzte Algorithmen
- Algorithmen
 - Implementiert
 - Untersucht auf Effizienz
- Konzepte für Modelica Compiler Entwicklung ausgearbeitet, umgesetzt, getestet
 - BackEnd-Pipeline
 - Modelimark

5 weiterführende Aufgaben

- Vollständig linear skalierender OpenModelica Compiler
 - FrontEnd
 - Kompilierung generierten Programmcodes für Simulator
- Weitere Ansätze für effiziente Transformation
 - Parallelisierung
 - Modellstruktur ausnutzen
(weniger Quellcode für Simulator)
- Erweiterung Modelica Sprache
 - Strukturvariable Systeme
 - Partielle Differentialgleichungen

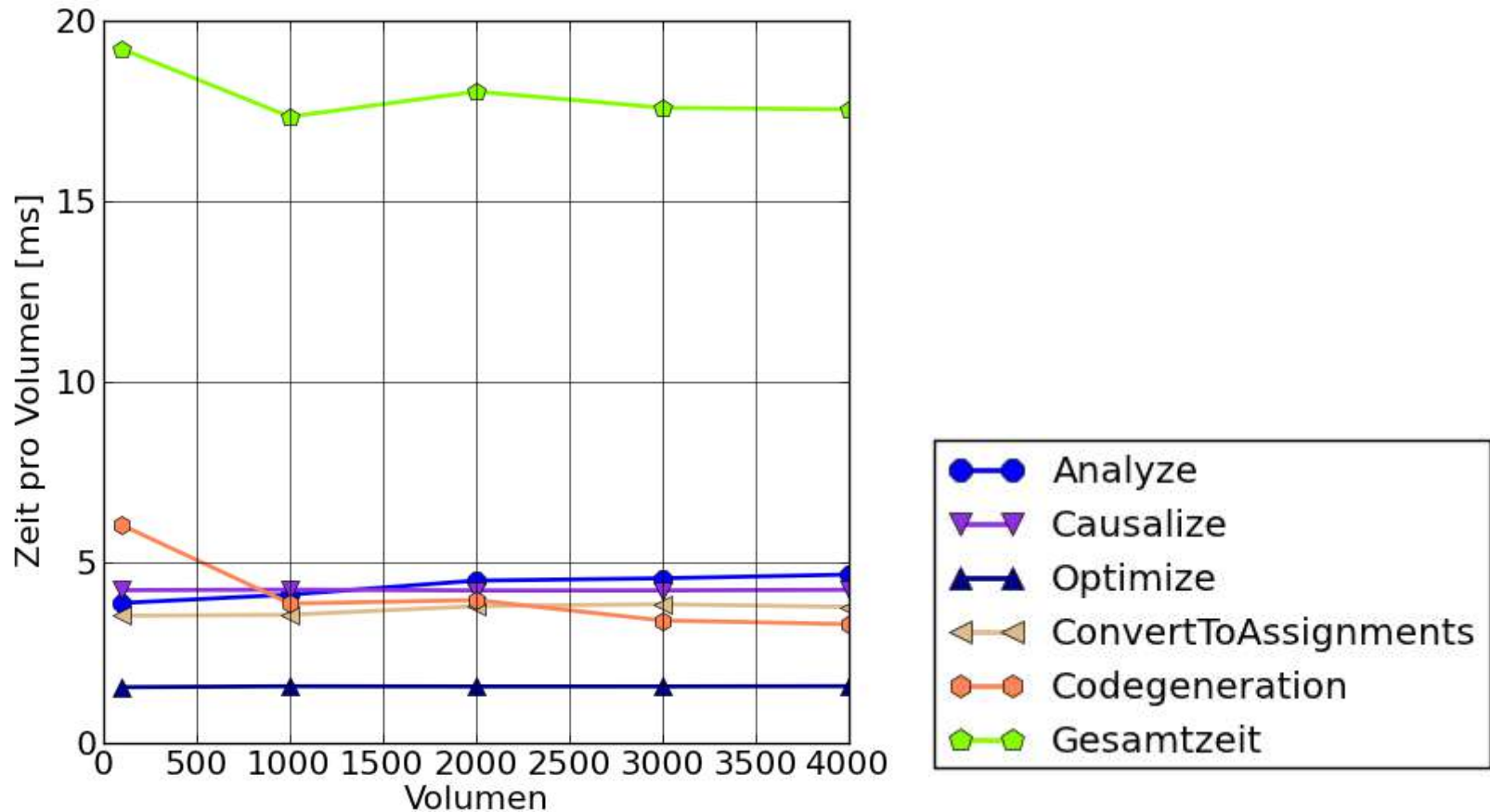
Vielen Dank für Ihre Aufmerksamkeit!

www.qucosa.de:

<http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-132032>

Diskussion

Benchmark des Hydraulik-Modells – lange Leitung





»Wissen schafft Brücken.«