Current Status of the DAE Mode for Solving Large-Scale Simulation Models

Recent Developments

Willi Braun Bernhard Bachmann

University of Applied Sciences Bielefeld
Bielefeld, Germany

February 5, 2018

Motivation

DAE mode is quite useful for large models.

Network	Gen's	Lines	Trafo's	Equations
RETE_C	74	369	583	56386
RETE_E	267	1458	1202	157022
RETE_G	407	6833	2824	593886

Network	Rel. tol.	No. of steps	Sim. time [s]
RETE_C	10^{-6}	146	3.18
RETE_E	10^{-6}	364	15.22
RETE_G	10^{-6}	615	123.19

Motivation

DAE mode is quite useful for large models.

Benchmark	*NE*	*NS*	OD	OS	DA	DD
SimpleAdvection_N_3200	6402	3199	20.81	4.851	3.087	2.561
SimpleAdvection_N_6400	12802	6399	104.9	13.27	6.107	6.781
SimpleAdvection_N_12800	25602	12799	642.2	41.15	19.17	18.38
SteamPipe_N_640	8966	1280	169.2	148.4	158.7	139.3
SteamPipe_N_1280	17926	2560	395.8	316.8	357.8	302.9
SteamPipe_N_2560	35846	5120	1165	651.0	801.9	679.9

Motivation

DAE mode is quite useful for large models.

Benchmark	*NE*	*NS*	OD	OS	DA	DD
SimpleAdvection_N_3200	6402	3199	20.81	4.851	3.087	2.561
SimpleAdvection_N_6400	12802	6399	104.9	13.27	6.107	6.781
SimpleAdvection_N_12800	25602	12799	642.2	41.15	19.17	18.38
SteamPipe_N_640	8966	1280	169.2	148.4	158.7	139.3
SteamPipe_N_1280	17926	2560	395.8	316.8	357.8	302.9
SteamPipe_N_2560	35846	5120	1165	651.0	801.9	679.9

These results have been presented in Prag, where the focus has been on simulation time.
Next step is to enhance the compiling process.

Outline

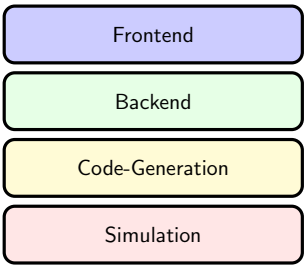
- Outline
  - Implementation Overview
  - Deal with Events
  - Preliminary Results

Outline

- Outline
  - Implementation Overview
  - Deal with Events
  - Preliminary Results

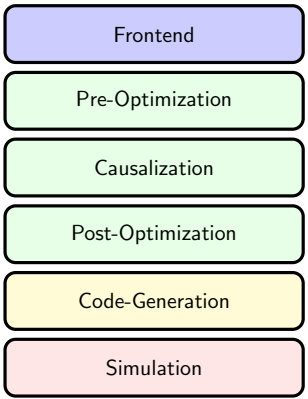
Implementation Overview

First Approach



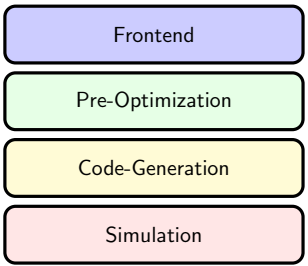
Implementation Overview

First Approach



Implementation Overview

First Approach

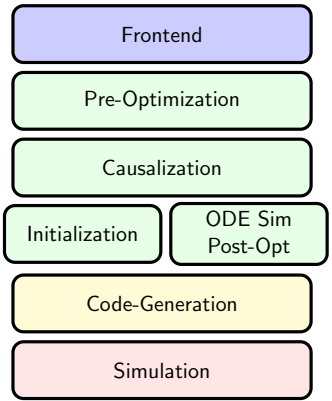


Ideal DAE mode

- Use the system generated by the Frontend and push it to the code generation.
- Skip the causelization process and other time consuming tasks in the Backend.
- This should reduce the compilation time.

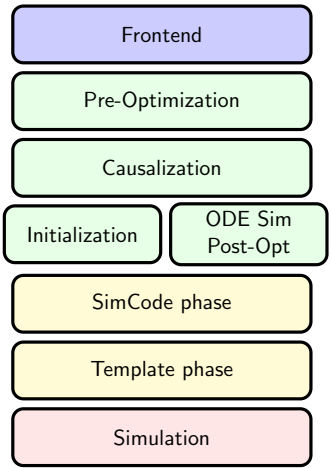
Implementation Overview

First Approach



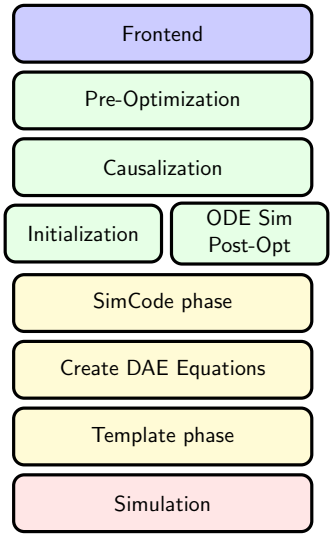
Implementation Overview

First Approach



Implementation Overview

First Approach



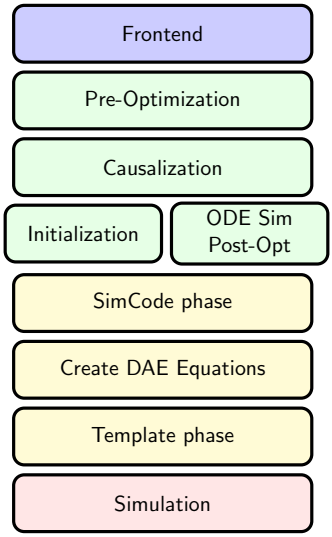
Generate residual equations

- Generation has made based on sorted equations
- The sorting allows to select only dynamic part
- Causelized system is used for the event handling

Implementation Overview

First Approach

FH Bielefeld
University of
Applied Sciences



Generate residual equations

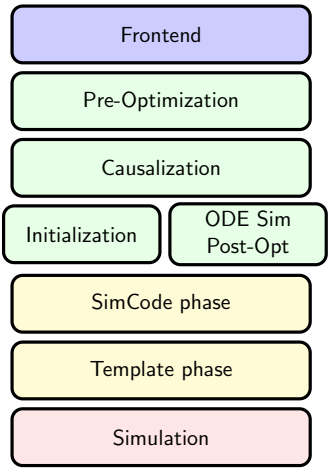
- Generation has made based on sorted equations
- The sorting allows to select only dynamic part
- Causelized system is used for the event handling

Issues

- Additional compilation work
- Needs still Backend modules (e.g. sparsity detection)

Implementation Overview

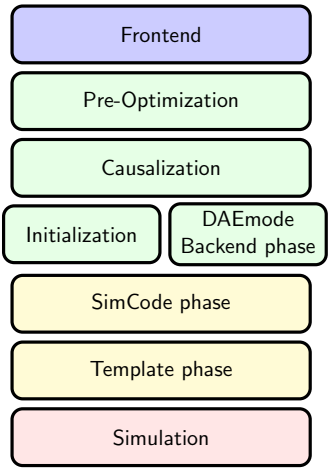
New Approach



Implementation Overview

New Approach

FH Bielefeld
University of
Applied Sciences

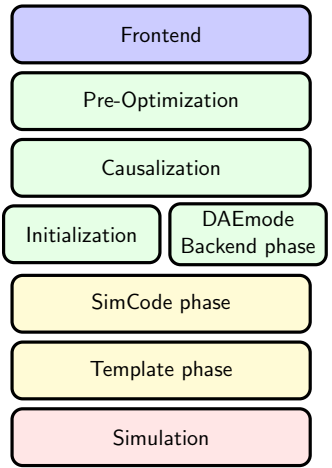


New DAE mode

- Added appropriate Backend modules
- Resolve all equation by adding residual variables
- Introducing auxiliary variables for e.g. CSE variables

Implementation Overview

New Approach



New DAE mode

- Added appropriate Backend modules
- Resolve all equation by adding residual variables
- Introducing auxiliary variables for e.g. CSE variables

How to proceed with events and the discrete variables?

Deal with Events

Event handing in a nutshell

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

$\downarrow$ matching and sorting algorithms transform to

$$\underline{z} = \begin{pmatrix} \dot{\underline{x}}(t) \\ \underline{y}(t) \\ \underline{q}(t_e) \end{pmatrix} = \begin{pmatrix} \underline{f}(\underline{x}(t), \underline{u}(t), \underline{q_{pre}}(t_e), \underline{c}(t_e), \underline{p}, t) \\ \underline{g}(\underline{x}(t), \underline{u}(t), \underline{q_{pre}}(t_e), \underline{c}(t_e), \underline{p}, t) \\ \underline{h}(\underline{x}(t), \underline{u}(t), \underline{q_{pre}}(t_e), \underline{c}(t_e), \underline{p}, t) \end{pmatrix}$$

Deal with Events

Event handing in a nutshell

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

$$\downarrow \text{ matching and sorting algorithms transform to}$$

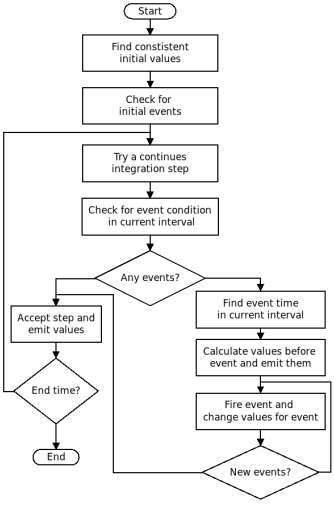
$$\underline{z} = \begin{pmatrix} \dot{\underline{x}}(t) \\ \underline{y}(t) \\ \underline{q}(t_e) \end{pmatrix} = \begin{pmatrix} \underline{f}(\underline{x}(t), \underline{u}(t), \underline{q_{pre}}(t_e), \underline{c}(t_e), \underline{p}, t) \\ \underline{g}(\underline{x}(t), \underline{u}(t), \underline{q_{pre}}(t_e), \underline{c}(t_e), \underline{p}, t) \\ \underline{h}(\underline{x}(t), \underline{u}(t), \underline{q_{pre}}(t_e), \underline{c}(t_e), \underline{p}, t) \end{pmatrix}$$

We get four blocks:

- continuous $\underline{f} \rightarrow$ (functionODE) to evaluate derivative states
  $\underline{g} \rightarrow$ (functionAlgebraics) to evaluate algebraic variables
- discrete $\underline{h} \rightarrow$ discrete algebraic variables included in z
- all $\underline{z} \rightarrow$ (functionDAE) all three blocks together

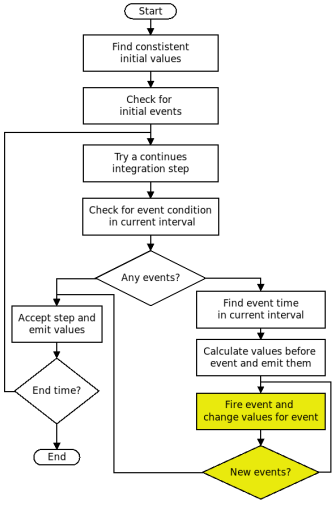
Deal with Events

Event handing in a nutshell



Deal with Events

Event handing in a nutshell



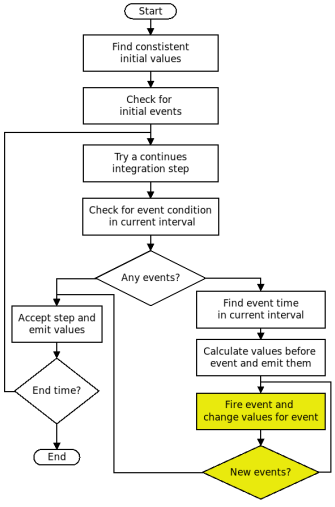
ODE-Mode:

- evaluate $\underline{z} \rightarrow$ with (functionDAE).

$$\underline{z} = \begin{pmatrix} \dot{\underline{x}}(t_e) \\ \underline{y}(t_e) \\ \underline{q}(t_e) \end{pmatrix} = \begin{pmatrix} \underline{f}(\underline{x}(t_e), \underline{q_{pre}}(t_e), \underline{c}(t_e), t) \\ \underline{g}(\underline{x}(t_e), \underline{q_{pre}}(t_e), \underline{c}(t_e), t) \\ \underline{h}(\underline{x}(t_e), \underline{q_{pre}}(t_e), \underline{c}(t_e), t) \end{pmatrix}$$

Deal with Events

Event handing in a nutshell



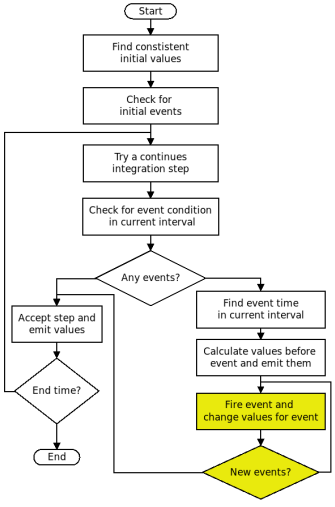
DAE-Mode: CASE h is empty

- Use $\underline{z_{res}} \rightarrow$ (evaluateResiduals) to solve for $\underline{x}(t_e), \underline{\dot{x}}(t_e), \underline{y}(t_e)$
- e.g. with IDACalcIC.

$$\underline{z_{res}} = \underline{k}(\underline{x}(t_e), \underline{\dot{x}}(t_e), \underline{y}(t_e), \underline{c}(t_e), t)$$

Deal with Events

Event handing in a nutshell



DAE-Mode: CASE h non empty, but without algebraic loops

- Causelize only h
- Evaluate h together with $z_{res} \rightarrow$ (evaluateResiduals).
- Solve for $\underline{x}(t_e), \underline{\dot{x}}(t_e), \underline{y}(t_e)$ as in the case above.

$$\begin{pmatrix} \underline{q}(t_e) \\ \underline{z_{res}} \end{pmatrix} = \begin{pmatrix} \underline{h}(\underline{x}(t_e), \underline{\dot{x}}(t_e), \underline{y}(t_e), \underline{q_{pre}}(t_e), \underline{c}(t_e), t) \\ \underline{k}(\underline{x}(t_e), \underline{\dot{x}}(t_e), \underline{y}(t_e), \underline{q_{pre}}(t_e), \underline{c}(t_e), t) \end{pmatrix}$$

Preliminary Results

FH Bielefeld
University of
Applied Sciences

CascadedFirstOrder

Stage:	new DAE mode:			old DAE mode:			ODE mode:		
	time in s			time in s			time in s		
N eqns	6400	12800	25600	6400	12800	25600	6400	12800	25600
Backend	3.316	7.359	15.31	3.142	6.773	14.78	3.117	7.078	13.97
SimCode	0.882	1.728	3.65	2.15	3.99	7.972	0.963	1.516	3.394
Template	1.807	2.709	5.391	1.843	3.685	7.361	1.748	2.627	5.194
Compile	9.427	15.88	28.57	12.5	21.54	39.92	10.18	17.23	31.04
Simulate	2.815	10.69	43.64	2.961	10.89	43.67	4.893	17.31	80.17

Preliminary Results

FH Bielefeld
University of
Applied Sciences

DistributionSystemAC.ScaledExperiments.DistributionSystemLinear

Stage:	new DAE mode:			old DAE mode:			ODE mode:		
	time in s			time in s			time in s		
N eqns	49016	99776	195232	49016	99776	195232	49016	99776	195232
Backend	14.17	33.18	71	19.95	59.49	225.7	18.02	56.6	217.5
SimCode	3.022	6.47	20.6	5.827	14.76	52.01	3.524	9.697	42.22
Template	5.22	9.216	17	7.366	13.24	34.92	5.552	9.427	26.84
Compile	14.88	30.48	55.74	20.33	39.81	72.3	14.84	28.43	55.26
Simulate	x	x	x	1.287	2.939	5.602	39.04	155.6	588.7

Preliminary Results

FH Bielefeld
University of
Applied Sciences

OneDHeatTransferTT_FD

Stage:	new DAE mode:			old DAE mode:			ODE mode:		
	time in s			time in s			time in s		
N eqns	320	640	1280	320	640	1280	320	640	1280
Backend	0.391	0.984	2.5	0.356	0.836	2.049	0.362	0.847	1.635
SimCode	0.064	0.147	0.320	0.253	0.416	1.292	0.067	0.155	0.390
Template	0.117	0.268	0.488	0.154	0.486	0.740	0.153	0.224	0.565
Compile	2.755	4.012	4.693	3.593	5.277	7.277	3.02	3.81	5.372
Simulate	0.056	0.068	0.137	0.089	0.355	0.449	0.211	0.497	0.817

Plans and Outlook

- Add symbolic Jacobian support
- Tweak the compile performance
- Further development of event handling

Plans and Outlook

Questions

- Add symbolic Jacobian support
- Tweak the compile performance
- Further development of event handling