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

Recent Developments

Willi Braun  Bernhard Bachmann

FH Bielefeld
University of Applied Sciences

University of Applied Sciences Bielefeld
Bielefeld, Germany

February 5, 2018
Motivation

DAE mode is quite useful for large models.

<table>
<thead>
<tr>
<th>Network</th>
<th>Gen’s</th>
<th>Lines</th>
<th>Trafo’s</th>
<th>Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETE_C</td>
<td>74</td>
<td>369</td>
<td>583</td>
<td>56386</td>
</tr>
<tr>
<td>RETE_E</td>
<td>267</td>
<td>1458</td>
<td>1202</td>
<td>157022</td>
</tr>
<tr>
<td>RETE_G</td>
<td>407</td>
<td>6833</td>
<td>2824</td>
<td>593886</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network</th>
<th>Rel. tol.</th>
<th>No. of steps</th>
<th>Sim. time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETE_C</td>
<td>$10^{-6}$</td>
<td>146</td>
<td>3.18</td>
</tr>
<tr>
<td>RETE_E</td>
<td>$10^{-6}$</td>
<td>364</td>
<td>15.22</td>
</tr>
<tr>
<td>RETE_G</td>
<td>$10^{-6}$</td>
<td>615</td>
<td>123.19</td>
</tr>
</tbody>
</table>
## Motivation

DAE mode is quite useful for large models.

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

DAE mode is quite useful for large models.

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

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

- Implementation Overview
- Deal with Events
- Preliminary Results
Outline

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

First Approach

Frontend
Backend
Code-Generation
Simulation
Implementation Overview
First Approach

- Frontend
- Pre-Optimization
- Causalization
- Post-Optimization
- Code-Generation
- Simulation
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

- Frontend
- Pre-Optimization
- Causalization
- Initialization
- ODE Sim Post-Opt
- Code-Generation
- Simulation
Implementation Overview

First Approach

- Frontend
- Pre-Optimization
- Causalization
- Initialization
- ODE Sim Post-Opt
- SimCode phase
- Template phase
- Simulation
Implementation Overview
First Approach

- Frontend
- Pre-Optimization
- Causalization
- ODE Sim Post-Opt
- Initialization
- SimCode phase
- Create DAE Equations
- Template phase
- Simulation

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
Generate residual equations

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

Issues

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

New Approach

- Frontend
- Pre-Optimization
- Causalization
- Initialization
- ODE Sim
- Post-Opt
- SimCode phase
- Template phase
- Simulation

How to proceed with events and the discrete variables?
Implementation Overview
New Approach

- Frontend
- Pre-Optimization
- Causalization
- Initialization
- DAEmode (Backend phase)
- SimCode phase
- Template phase
- Simulation

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

Frontend
Pre-Optimization
Causalization
Initialization

DAEmode
Backend phase

SimCode phase
Template phase
Simulation

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 handling in a nutshell

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

\[ \downarrow \text{matching and sorting algorithms transform to} \]

\[
\dot{z} = \begin{pmatrix}
\dot{x}(t) \\
y(t) \\
q(t_e)
\end{pmatrix}
= \begin{pmatrix}
f(x(t), u(t), q_{pre}(t_e), c(t_e), p, t) \\
g(x(t), u(t), q_{pre}(t_e), c(t_e), p, t) \\
h(x(t), u(t), q_{pre}(t_e), c(t_e), p, t)
\end{pmatrix}
\]
Deal with Events
Event handling in a nutshell

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

\[ \downarrow \text{matching and sorting algorithms transform to} \]

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

We get four blocks:

**continuous** \( f \rightarrow \text{functionODE} \) to evaluate derivative states
\( g \rightarrow \text{functionAlgebraics} \) to evaluate algebraic variables

**discrete** \( h \rightarrow \) discrete algebraic variables included in \( z \)

**all** \( z \rightarrow \text{functionDAE} \) all three blocks together
Deal with Events
Event handing in a nutshell

Start

Find consistent initial values

Check for initial events

Try a continues integration step

Check for event condition in current interval

Any events?

Accept step and emit values

End time?

End

Find event time in current interval

Calculate values before event and emit them

Fire event and change values for event

New events?
Deal with Events

Event handling in a nutshell

ODE-Mode:

- evaluate $z \rightarrow$ with (functionDAE).

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

Event handling in a nutshell

DAE-Mode: CASE $h$ is empty

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

\[ z_{res} = k(x(t_e), \dot{x}(t_e), y(t_e), c(t_e), t) \]
Deal with Events

Event handling in a nutshell

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

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

\[
\begin{pmatrix}
q(t_e) \\
z_{res}
\end{pmatrix} = \begin{pmatrix}
h(x(t_e), \dot{x}(t_e), y(t_e), q_{pre}(t_e), c(t_e), t) \\
h(x(t_e), \dot{x}(t_e), y(t_e), q_{pre}(t_e), c(t_e), t)
\end{pmatrix}
\]
## Preliminary Results

### CascadedFirstOrder

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

**DistributionSystemAC.ScaledExperiments.DistributionSystemLinear**

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

### OneDHeatTransferTT_FD

<table>
<thead>
<tr>
<th>Stage:</th>
<th>new DAE mode:</th>
<th>old DAE mode:</th>
<th>ODE mode:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time in s</td>
<td>time in s</td>
<td>time in s</td>
</tr>
<tr>
<td></td>
<td>320</td>
<td>640</td>
<td>1280</td>
</tr>
<tr>
<td>N eqns</td>
<td>0.391</td>
<td>0.984</td>
<td>2.5</td>
</tr>
<tr>
<td>Backend</td>
<td>0.064</td>
<td>0.147</td>
<td>0.320</td>
</tr>
<tr>
<td>SimCode</td>
<td>0.117</td>
<td>0.268</td>
<td>0.488</td>
</tr>
<tr>
<td>Template</td>
<td>2.755</td>
<td>4.012</td>
<td>4.693</td>
</tr>
<tr>
<td>Compile</td>
<td>0.056</td>
<td>0.068</td>
<td>0.137</td>
</tr>
</tbody>
</table>
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