High-order and multi-rate time stepping with preCICE

Benjamin Rüth

Technical University of Munich, Department of Informatics

preCICE Workshop 2020
Garching, Germany
February 5, 2020
The most important changes:

- You can now (finally) use the same preCICE configuration file for serial and parallel runs.
- The configuration reference can now also be generated in markdown. A recent version is always in the wiki.
- **We cleaned up the duplicated meaning of “timestep”**: preCICE now uses “time windows” and the participants do their “timesteps”.
- Mesh handling is much faster 🚀.
- We moved the Python and (new) Matlab bindings as well as the Fortran Module (formerly known as f2003 bindings) to separate repositories. Btw, the Python bindings are now really pythonic and you can get them through PyPI.
- For the first time, two-level initialization is available, allowing for fast initialization of very large cases 🚀🚀. The feature is, however, still in beta testing and switched off by default. We will have a presentation at the workshop (and afterwards online) about the new initialization concept.
- We restructured the repository a bit: developer tools are now in tools, user tools in extras, native bindings in extras/bindings, and solver dummies in examples. These examples are now also shipped with our binary packages, and you can use them to test your installation.
Dear Benjamin,

I do not really know what a time window is supposed to be and why it is better than a timestep.

Is it really necessary to torture the whole community???

Best regards,

a preCICE user

On 14.02.20 09:32, Benjamin Uekermann wrote:

Dear preCICE Community,

You might have seen it on GitHub already: we have a fresh new release, preCICE v2.0 and since yesterday, all adapters, bindings, and tutorials are compatible.

Breaking news: we have breaking changes 😊. We decided to move to v2.0 to clean up some
How time stepping usually works

\[ u_{\mathcal{D}_0} \to t_{\text{ini}} + \Delta t \]

\[ u_{\mathcal{D}_1} \to \Delta t \]

\[ u_{\mathcal{N}_0} \to t_{\text{ini}} \]

\[ u_{\mathcal{N}_1} \to \Delta t \]
ExaFSA setup

Fluid-acoustics simulation and partitioned setup\(^1\).

<table>
<thead>
<tr>
<th>physics</th>
<th>timescale</th>
<th>solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>small</td>
<td>Ateles</td>
</tr>
<tr>
<td>(A)</td>
<td>small</td>
<td>FASTEST</td>
</tr>
<tr>
<td>(F)</td>
<td>medium</td>
<td>FASTEST</td>
</tr>
<tr>
<td>(S)</td>
<td>large</td>
<td>FEAP</td>
</tr>
</tbody>
</table>

- Quasi-Newton
- Black-box
- High order time-stepping


Benjamin Rüth (TUM) | High-order and multi-rate time stepping with preCICE
Let's use multirate/subcycling!

\[ u_{\varphi_0} \rightarrow u_{\varphi_1} \rightarrow u_{\varphi_2} \]

\[ t_{\text{ini}} \rightarrow t_{\text{ini}} + \Delta t \]

\[ \delta t_{\varphi} \]

\[ u_{N_0} \rightarrow u_{N_1} \rightarrow u_{N_2} \rightarrow u_{N_3} \rightarrow u_{N_4} \rightarrow u_{N_5} \]

\[ \delta t_{N} \]
Let's use multirate/subcycling!

double solver_dt = 0.1; // solver timestep size
double precice_dt; // maximum precice timestep size
double t = 0; // time

precice_dt = precice.initialize(); // e.g. 0.5

while (precice.isCouplingOngoing()){

  ... // reading

  dt = min(precice_dt, solver_dt); // always 0.1
  solver.doTimestep(dt);
  t += dt; // 0.1; 0.2; 0.3; 0.4; 0.5; 0.6, 0.7 ...
  precice_dt = precice.advance(dt); // 0.4; 0.3; 0.2; 0.1; 0.5; 0.4, 0.3 ...

  ... // writing
}

Benjamin Rüth (TUM) | High-order and multi-rate time stepping with preCICE
Let’s use multirate/subcycling!

double solver_dt = 0.1; // solver timestep size
double precice_dt; // maximum precice timestep size
double t = 0; // time

precice_dt = precice.initialize(); // e.g. 0.5

while (precice.isCouplingOngoing()){
    ... // reading + save checkpoint
    
    dt = min(precice_dt, solver_dt); // always 0.1
    solver.doTimestep(dt);
    t += dt; // 0.1; 0.2; 0.3; 0.4; 0.5; 0.1, 0.2 ...
    precice_dt = precice.advance(dt); // 0.4; 0.3; 0.2; 0.1; 0.5; 0.4, 0.3 ...

    ... // writing + restore checkpoint
}

Benjamin Rüth (TUM) | High-order and multi-rate time stepping with preCICE
Let’s use multirate/subcycling!

BCs are constant over window $\Delta t$

\[ u_{\mathcal{D}_0} \rightarrow u_{\mathcal{D}_1} \rightarrow u_{\mathcal{D}_2} \]

\[ u_{\mathcal{N}_5} \rightarrow u_{\mathcal{N}_4} \rightarrow u_{\mathcal{N}_3} \rightarrow u_{\mathcal{N}_2} \rightarrow u_{\mathcal{N}_1} \rightarrow u_{\mathcal{N}_0} \]
Let’s use multirate/subcycling!

BCs are constant over timestep \( \delta t \)

\[
\begin{align*}
\delta t &= u_{\phi_0} \\
\delta t &= u_{\phi_1} \\
\delta t &= u_{\phi_2} \\
\end{align*}
\]

\[
\begin{align*}
u &= \text{readData("u_N")} \\
t_{\text{ini}} + \Delta t &= u_{\phi_1} \\
t_{\text{ini}} &= u_{\phi_2} \\
\end{align*}
\]
Let's use multirate/subcycling!

BCs are interpolated over time $t$

Additional API call

$u = \text{readData}("u_N", t)$
Prototype implementation

libprecice
pyprecice
fenics-adapter
FEniCS
Prototype implementation
Partitioned Heat Equation

\[ D(T) \] \[ \Gamma_D \] \[ \text{preCICE IQN-ILS} \] \[ \Gamma_N \] \[ N(q) \]
Partitioned Heat Equation

<participant name="D">
  <write data name="q1"/>
  <write data name="q2"/>
  <read data name="T1"/>
  <read data name="T2"/>
  <read data name="T3"/>
  <read data name="T4"/>
  <read data name="T5"/>
</participant>

<participant name="N">
  <write data name="T1"/>
  <write data name="T2"/>
  <write data name="T3"/>
  <write data name="T4"/>
  <write data name="T5"/>
  <read data name="q1"/>
  <read data name="q2"/>
</participant>

preCICE
IQN-ILS
Partitioned Heat Equation

Check QN Performance

- different multirate setups $WI(n_D, n_N)$
- QN-WI feeds all samples into Quasi Newton
- interpolate BCs over time
- only linear interpolation & implicit Euler
- compute for $T = 10$

<table>
<thead>
<tr>
<th>QN-WI</th>
<th>$\Delta t$</th>
<th>5.0</th>
<th>0.5</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>WI(1,1)</td>
<td>10.50</td>
<td>7.85</td>
<td>5.45</td>
<td></td>
</tr>
<tr>
<td>WI(1,3)</td>
<td>11.50</td>
<td>8.85</td>
<td>6.60</td>
<td></td>
</tr>
<tr>
<td>WI(1,5)</td>
<td>11.50</td>
<td>8.75</td>
<td>6.77</td>
<td></td>
</tr>
<tr>
<td>WI(3,1)</td>
<td>10.50</td>
<td>8.10</td>
<td>5.43</td>
<td></td>
</tr>
<tr>
<td>WI(3,3)</td>
<td>12.00</td>
<td>9.30</td>
<td>6.36</td>
<td></td>
</tr>
<tr>
<td>WI(3,5)</td>
<td>12.00</td>
<td>9.85</td>
<td>6.89</td>
<td></td>
</tr>
<tr>
<td>WI(5,1)</td>
<td>10.50</td>
<td>8.15</td>
<td>5.43</td>
<td></td>
</tr>
<tr>
<td>WI(5,3)</td>
<td>11.50</td>
<td>9.85</td>
<td>6.82</td>
<td></td>
</tr>
<tr>
<td>WI(5,5)</td>
<td>12.00</td>
<td>9.45</td>
<td>6.41</td>
<td></td>
</tr>
</tbody>
</table>
Partitioned Heat Equation

Check QN Performance

- different multirate setups $\text{WI}(n_D, n_N)$
- QN-WI feeds all samples into Quasi Newton
- interpolate BCs over time
- only linear interpolation & implicit Euler
- compute for $T = 10$

<table>
<thead>
<tr>
<th>QN-WI</th>
<th>$\Delta t$</th>
<th>5.0</th>
<th>0.5</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>WI(1,1)</td>
<td>10.50</td>
<td>7.85</td>
<td>5.45</td>
<td></td>
</tr>
<tr>
<td>WI(1,3)</td>
<td>11.50</td>
<td>8.85</td>
<td>6.60</td>
<td></td>
</tr>
<tr>
<td>WI(1,5)</td>
<td>11.50</td>
<td>8.75</td>
<td>6.77</td>
<td></td>
</tr>
<tr>
<td>WI(3,1)</td>
<td>10.50</td>
<td>8.10</td>
<td>5.43</td>
<td></td>
</tr>
<tr>
<td>WI(3,3)</td>
<td>12.00</td>
<td>9.30</td>
<td>6.36</td>
<td></td>
</tr>
<tr>
<td>WI(3,5)</td>
<td>12.00</td>
<td>9.85</td>
<td>6.89</td>
<td></td>
</tr>
<tr>
<td>WI(5,1)</td>
<td>10.50</td>
<td>8.15</td>
<td>5.43</td>
<td></td>
</tr>
<tr>
<td>WI(5,3)</td>
<td>11.50</td>
<td>9.85</td>
<td>6.82</td>
<td></td>
</tr>
<tr>
<td>WI(5,5)</td>
<td>12.00</td>
<td>9.45</td>
<td>6.41</td>
<td></td>
</tr>
</tbody>
</table>
Partitioned Heat Equation

\[ \phi(\Delta t^2) \]

\[ \Delta t \]

\[ \text{L}^2(\Omega) \text{ error} \]

\[ \text{trapezoidal rule} \rightarrow \text{ideal slope} \]

\[ \text{WI}(5, 2; 1) \]
Partitioned Heat Equation

\[ \Delta t \]

\[ \text{trapezoidal rule} \rightarrow \text{ideal slope} \]

\[ L^2(\Omega) \text{ error} \]

- WI(5,2;1)
- WI(2,2;1)

\[ \Theta(\Delta t^2) \]
Partitioned Heat Equation

\[ L^2(\Omega) \text{ error} \]

\[ \begin{align*}
\Delta t & \quad \text{error} \\
10^{-1} & \quad 10^{-2} \\
10^{-2} & \quad 10^{-3} \\
10^{-3} & \quad 10^{-4} \\
10^{-4} & \quad 10^{-5} \\
10^{-5} & \quad 10^{-6} \\
10^{-6} & \quad 10^{-7} \\
10^{-7} & \quad 10^{-8}
\end{align*} \]

\[ \Theta(\Delta t^2) \]

trapezoidal rule → ideal slope

- WI(5, 2; 1)
- WI(2, 2; 1)
- WI(5, 5; 1)
- WI(5, 5; 2)

Benjamin Rüth (TUM) | High-order and multi-rate time stepping with preCICE
Partitioned Heat Equation

The graph shows the $L^2(\Omega)$ error as a function of the time step size $\Delta t$. The error decreases as $\Delta t$ decreases, following a slope of $O(\Delta t^2)$. Different lines represent different methods:

- $\text{WI}(5, 2; 1)$
- $\text{WI}(2, 2; 1)$
- $\text{WI}(5, 5; 1)$
- $\text{WI}(5, 5; 2)$

The ideal slope is indicated on the graph.
Partitioned Heat Equation

\[ O(\Delta t^2) \]

\[ \text{trapezoidal rule } \rightarrow \text{ideal slope} \]

\[ L^2(\Omega) \text{ error} \]

\[ \tilde{W}(5,2;1) \]
\[ \tilde{W}(2,2;1) \]
\[ \tilde{W}(5,5;1) \]
and \[ \tilde{W}(5,5;2) \]
\[ \tilde{W}(5,2;2) \]
\[ \tilde{W}(2,5;2) \]
FSI Flap
FSI Flap

Convergence study for trapezoidal rule (Fluid) and Newmark $\beta$ method (Solid)

$\|d - d_{\text{ref}}\| [\text{m}]$

$\Delta t [\text{s}]$

$\theta(\Delta t^2)$

- $\text{WI}(1, 1; 1)$
- $\text{WI}(2, 3; 2)$
- $\text{WI}(3, 2; 2)$
FSI Flap

Convergence study for trapezoidal rule (Fluid) and Newmark $\beta$ method (Solid)

![Graph showing convergence study](image)

QN-Iterations

<table>
<thead>
<tr>
<th>$\Delta t$[s]</th>
<th>$0.0025 \cdot 2^0$</th>
<th>$0.0025 \cdot 2^{-1}$</th>
<th>$0.0025 \cdot 2^{-2}$</th>
<th>$0.0025 \cdot 2^{-3}$</th>
<th>$0.0025 \cdot 2^{-4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>WI(1, 1; 1)</td>
<td>4.00</td>
<td>4.50</td>
<td>4.81</td>
<td>5.50</td>
<td>5.64</td>
</tr>
<tr>
<td>WI(2, 3; 2)</td>
<td>5.25</td>
<td>5.63</td>
<td>6.57</td>
<td>7.31</td>
<td>7.42</td>
</tr>
<tr>
<td>WI(3, 2; 2)</td>
<td>4.50</td>
<td>4.75</td>
<td>5.31</td>
<td>5.63</td>
<td>6.33</td>
</tr>
</tbody>
</table>
Conclusion and future work

Conclusion

- $\delta t \neq \Delta t$
- partitioned black-box solvers can efficiently use multirate + QN
- higher order can be reached
- functionality can be hidden inside preCICE

```xml
<participant name="D">
  <write data name="q1"/>
  <write data name="q2"/>
  <read data name="T1"/>
  <read data name="T2"/>
  <read data name="T3"/>
  <read data name="T4"/>
  <read data name="T5"/>
</participant>
```
Conclusion and future work

Conclusion

• $\delta t \neq \Delta t$
• partitioned black-box solvers can efficiently use multirate + QN
• higher order can be reached
• functionality can be hidden inside preCICE

Future work

• real preCICE implementation
• explicit coupling + extrapolation
Conclusion and future work

Conclusion

• $\delta t \neq \Delta t$
• partitioned black-box solvers can efficiently use multirate + QN
• higher order can be reached
• functionality can be hidden inside preCICE

Future work

• real preCICE implementation
• explicit coupling + extrapolation

Interested in details?