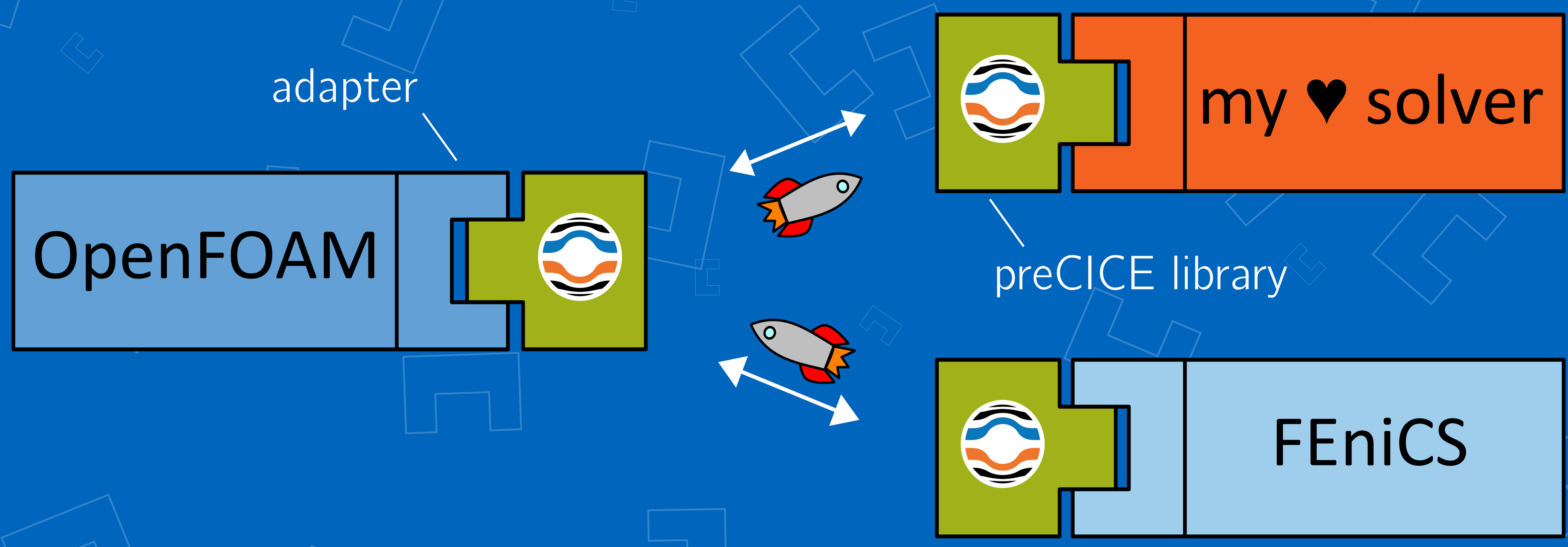
Higher-Order Time-Stepping in Multiphysics using preCICE

Poster Download



***Niklas Vinnitchenko**[*,†], **Benjamin Rodenberg**[*], **Hans-Joachim Bungartz**[*], **Benjamin Uekermann**[‡]*

[*] SCCS, TUM School of Computation, Information and Technology

[†] Alfred-Wegener-Institut, Helmholtz-Zentrum für Polar- und Meeresforschung

[‡] Usability and Sustainability of Simulation Software, Institute for Parallel and Distributed Systems, University of Stuttgart

niklas.vinnitchenko@tum.de

Plain language summary

To reduce the complexity of simulating multiphysics problems, one can follow a divide-and-conquer approach by dividing the problem into several smaller models and solving them instead with existing software. The software library preCICE helps realize this approach. It is a coupling library treating each solver as a black box to ensure that no compatibility dependencies among coupling participants exist. [1] With the help of time interpolation, preCICE can also maintain the convergence order of each solver. The partitioned heat equation is solved using different implicit Runge-Kutta schemes in FEniCS to show this. To this end, an automated approach of defining the weak equation employing higher-order Runge-Kutta schemes from Firedrake, a different FEM library [2] was adapted to FEniCS. As a more intricate example, a fluid-structure scenario is investigated. A solid, flexible, and fixed flap resides on the bottom of a fluid-filled channel.

Automated Higher-Order Implicit Time-Stepping Schemes in FEniCS

Consider as an example the following third-order RadauIIA method:

$$\begin{array}{c|cc} 1/3 & 5/12 & -1/12 \\ 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4 \end{array}$$

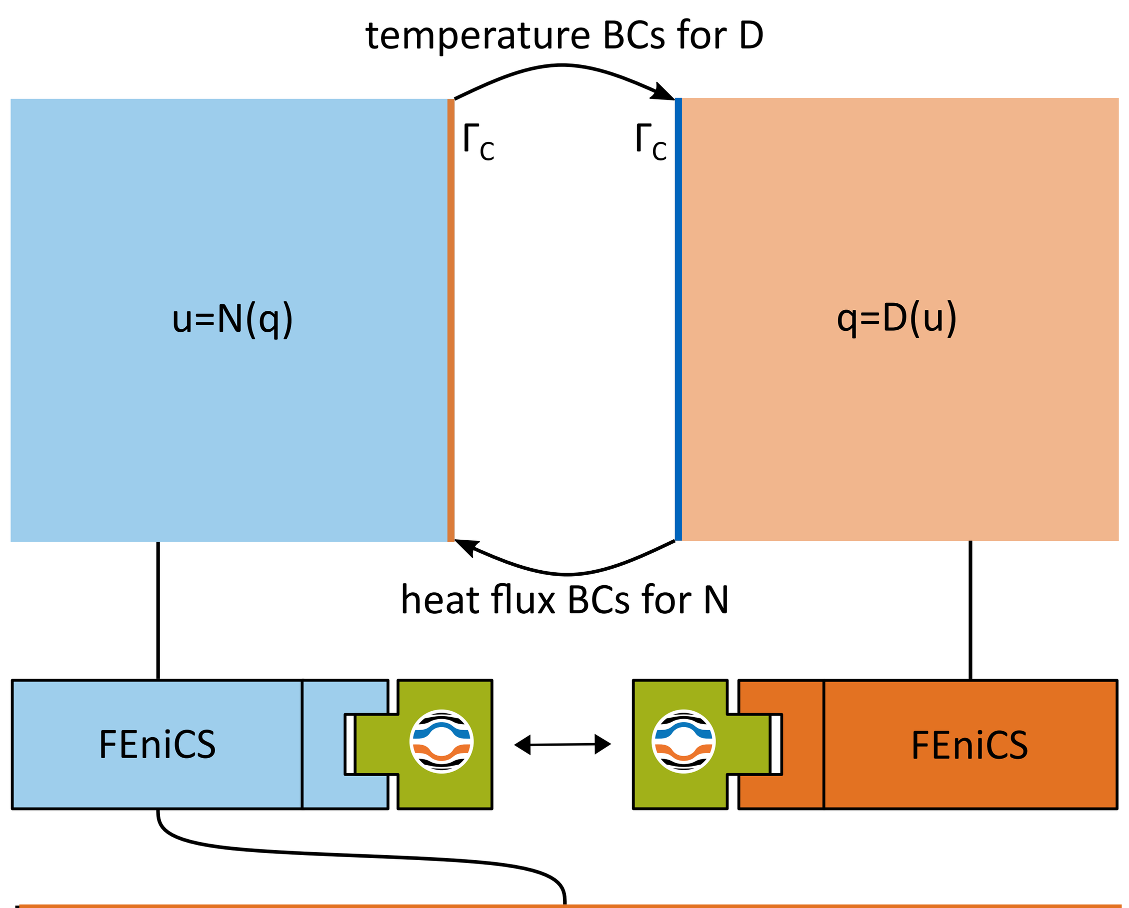
and the heat equation's weak form:

$$\int_\Omega \frac{\partial u}{\partial t} v \,\mathrm{d}x + \int_\Omega \nabla u \nabla v \,\mathrm{d}x - \int_{\partial\Omega} v \frac{\partial u}{\partial n} \,\mathrm{d}s = \int_\Omega f v \,\mathrm{d}x$$

Following [2], the implementation of this approach looks for the example above as follows for Dirichlet boundaries [3]:

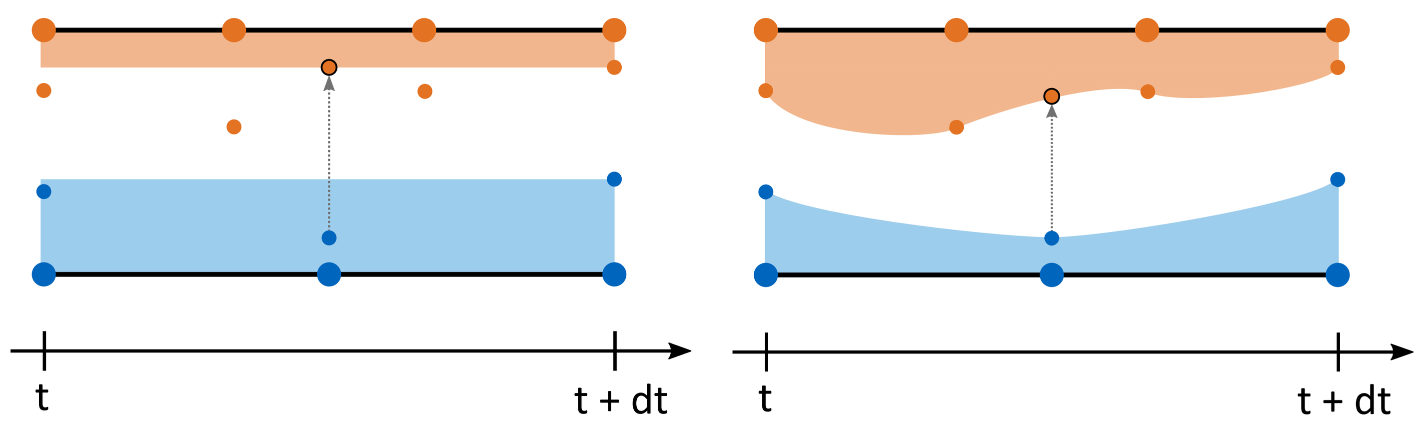
```
# init rhs f1, f2
# init test functions v1, v2
# init trial functions k1, k2
# create array us of length 2
us[0] = u_n + 5/12 * dt * k1 - 1/12 * dt * k2
us[1] = u_n + 3/4 * dt * k1 - 1/4 * dt * k2
F  = inner(k1, v1)*dx
F += inner(grad(us[0]), grad(v1))*dx
F -= f1*v1*dx
F += inner(k2, v2)*dx
F += inner(grad(us[1]), grad(v2))*dx
F -= f2*v2*dx
```

Partitioned heat equation

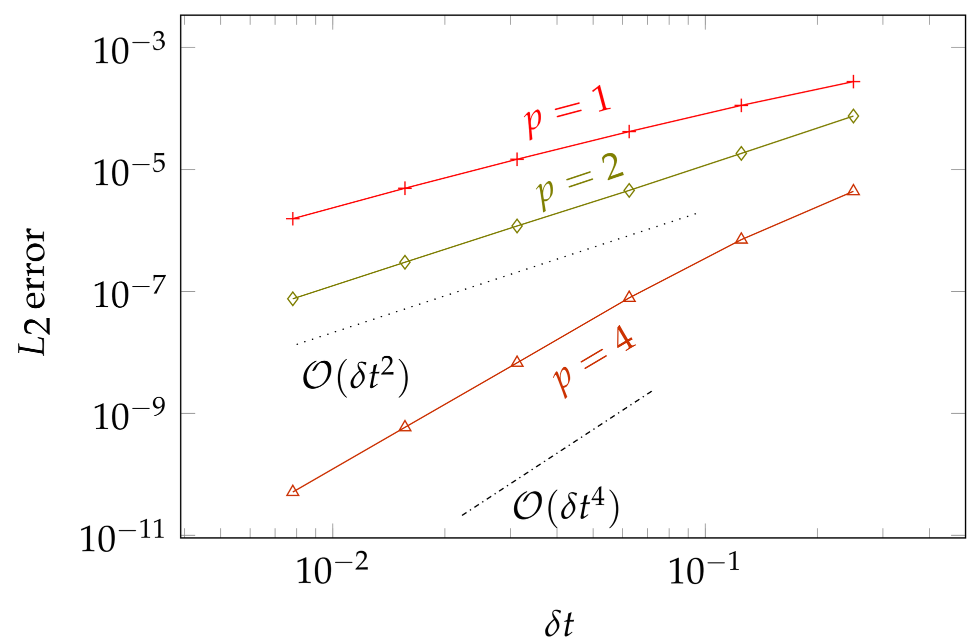



```
import precice
# some initialization steps ...
while participant.is_coupling_ongoing():
  dt = participant.get_max_time_step_size()
  q  = participant.read_data(dt)
  u  = solve_heat_equation(q)
  participant.write_data(u)
  participant.advance(dt)
```

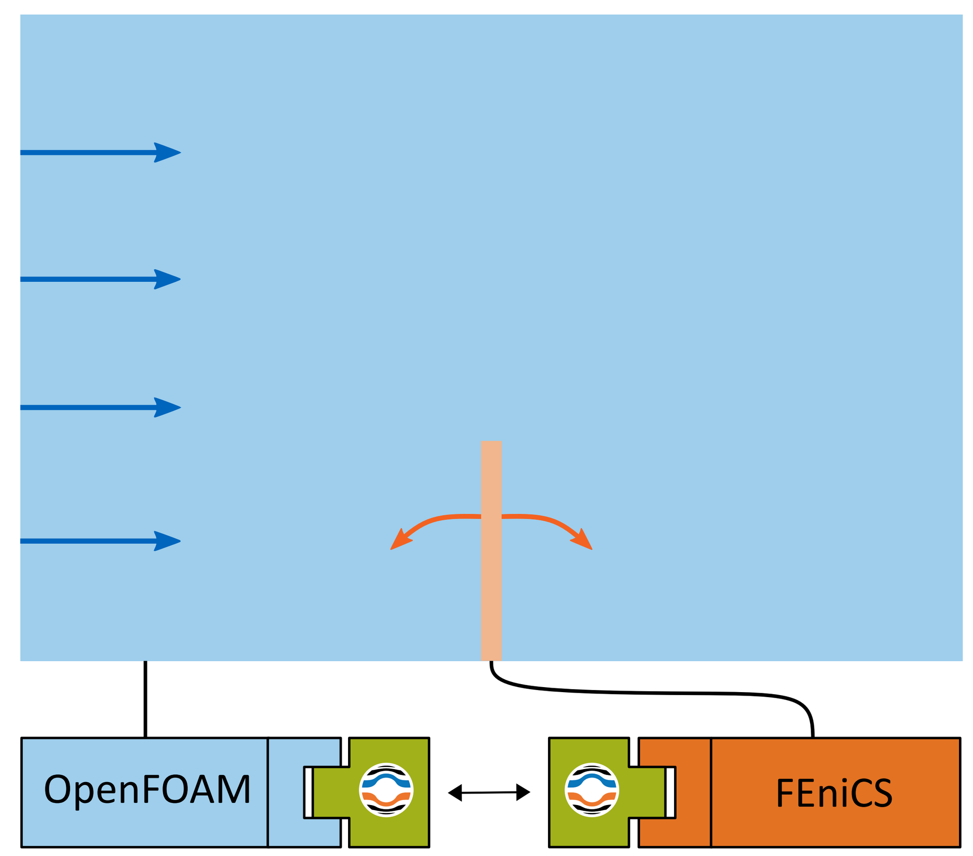
Since preCICE v3, it is possible to sample non-constant boundary conditions from participants.



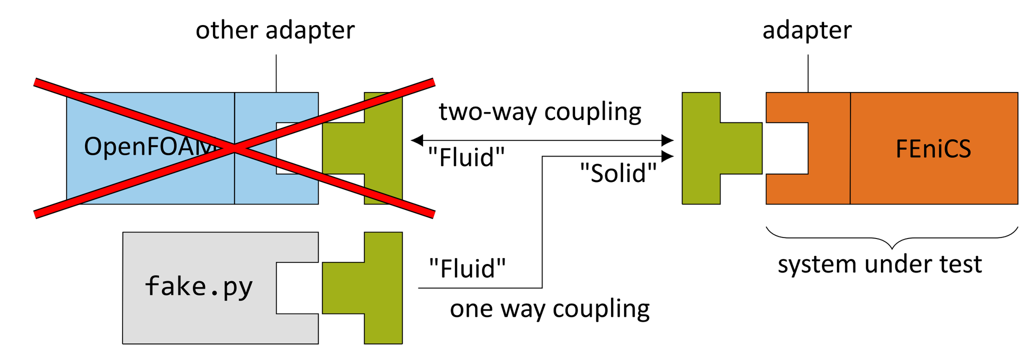
Convergence study for the partitioned heat equation with the 3-stage LobattoIIIC method (for different waveform degrees p)



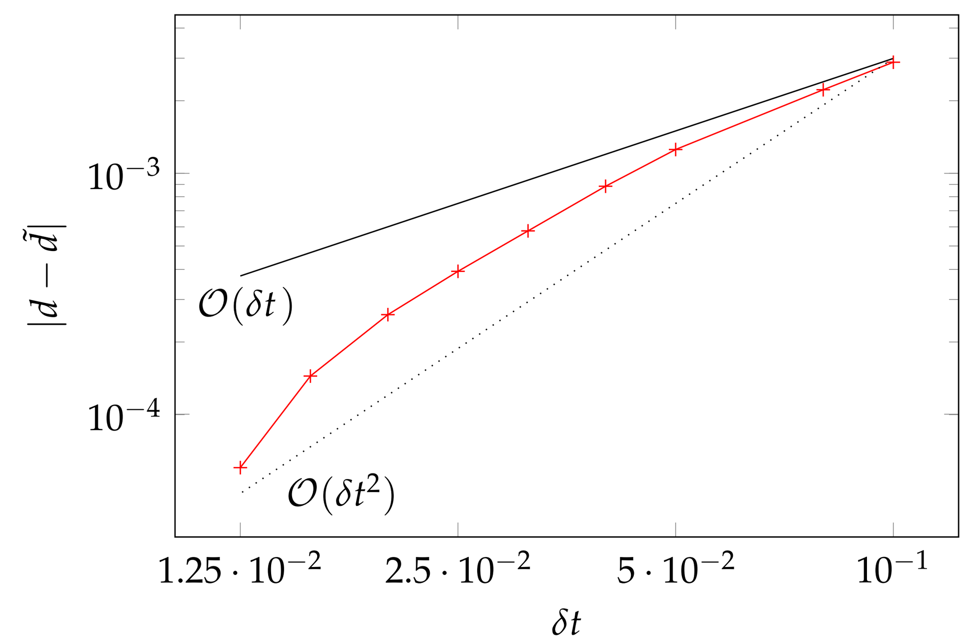
Perpendicular Flap



Testing with mocked participants



Convergence study using FEniCS for the flap and `fake.py` for the fluid



[1] Chourdakis G, Davis K, Rodenberg B et al. preCICE v2: A sustainable and user-friendly coupling library [version 2; peer review: 2 approved]. Open Res Europe 2022, 2:51 doi: 10.12688/openreseurope.14445.2

[2] Patrick E. Farrell, Robert C. Kirby, and Jorge Marchena-Menéndez. "Irksome: Automating Runge–Kutta Time-Stepping for Finite Element Methods." In: ACM Trans. Math. Softw. 47.4 (Sept. 2021). issn: 0098-3500. doi: 10.1145/3466168

[3] Vinnitchenko, N. (2024). Evaluation of Higher-Order Coupling Schemes with FEniCS-preCICE. https://mediatum.ub.tum.de/doc/1732367/1732367.pdf

ALFRED-WEGENER-INSTITUT
HELMHOLTZ-ZENTRUM FÜR POLAR- UND MEERESFORSCHUNG

University of Stuttgart
Germany

Technical University of Munich