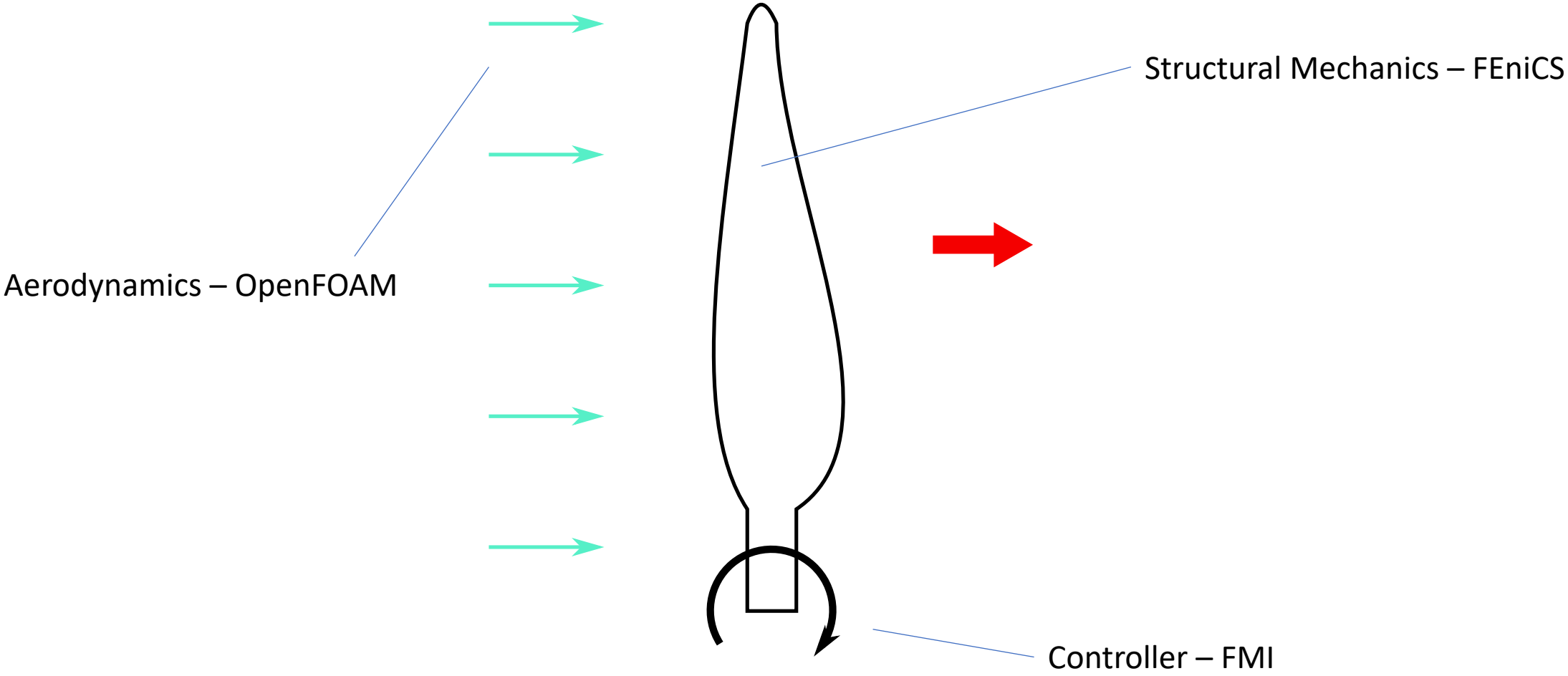
University of Stuttgart
Germany

A preCICE-FMI Runner to Couple Controller Models to PDEs

Question: Can FMI models be coupled to PDE-based solvers via preCICE?



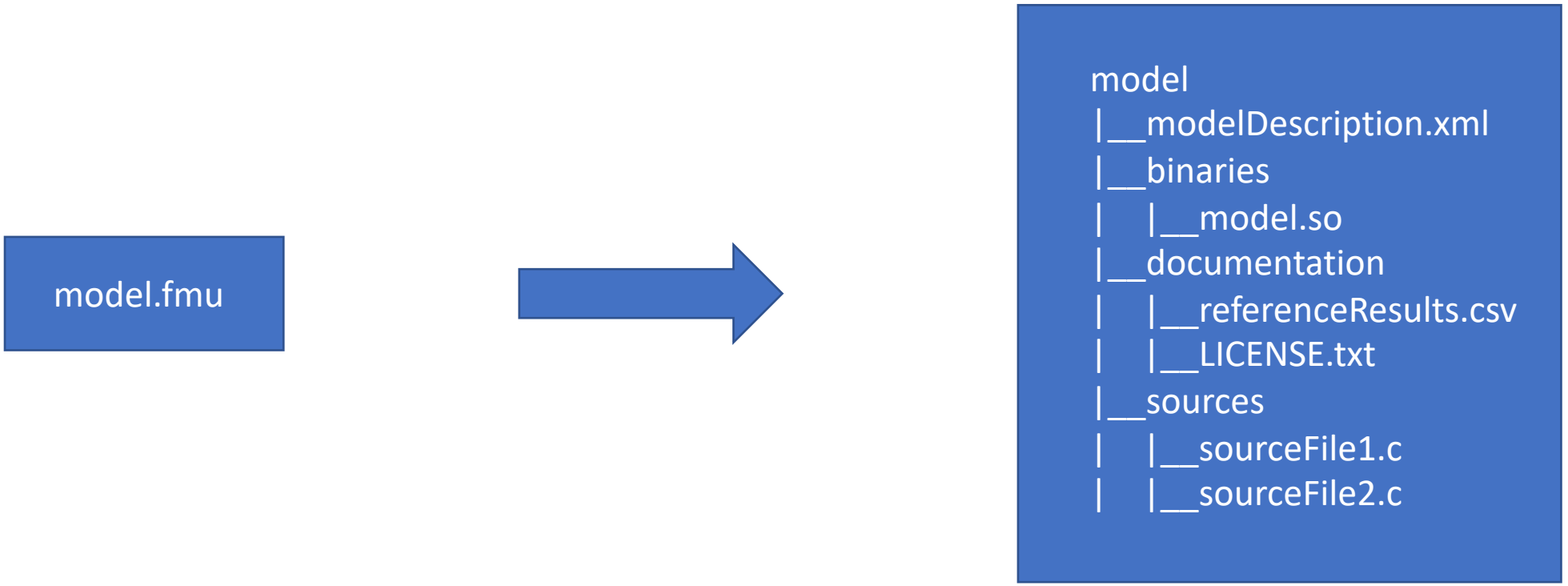
Result: A first preCICE-FMI coupling with limited abilities

Functional Mock-up Interface

Functional Mock-up Interface FMI

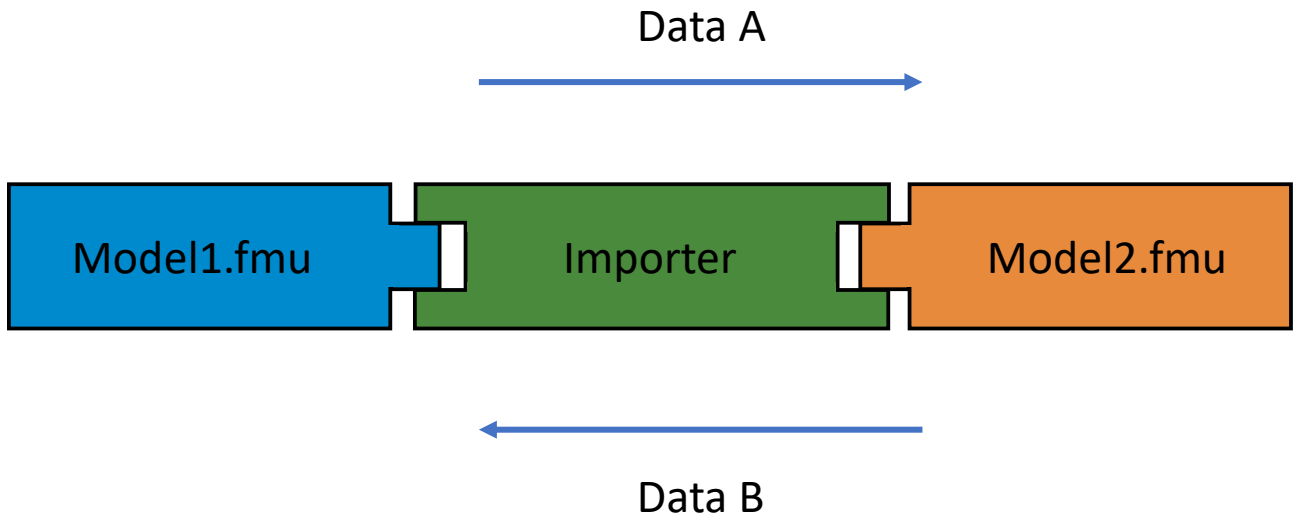
- Standard for the exchange of dynamic simulation models
- Developed by Modelica Association
- Defines a container and an interface

Functional Mock-up Unit FMU



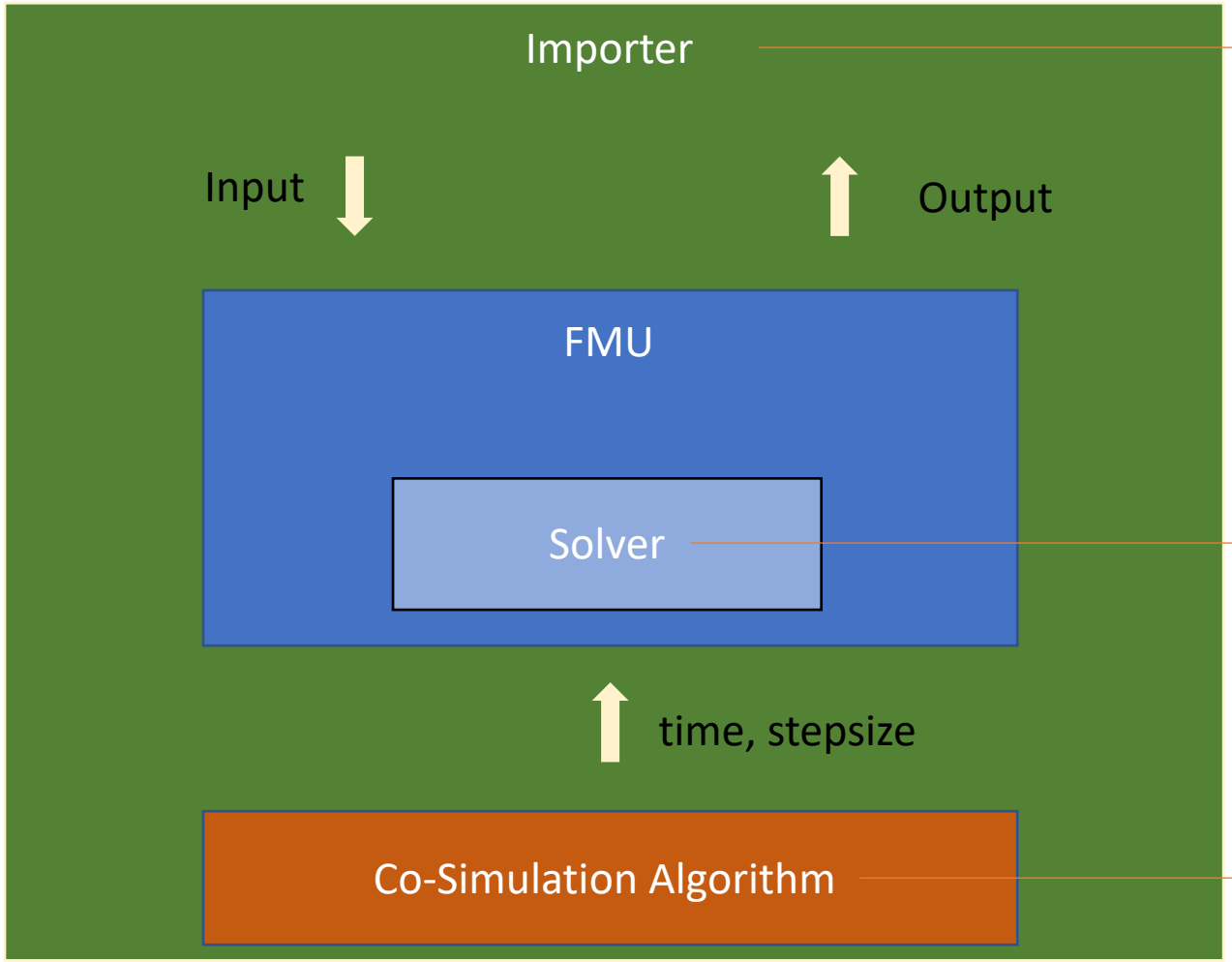
A Functional Mock-up Unit is a ZIP-Archive containing specific files and binaries

FMU Simulation



FMUs are loaded and executed by an importer, which also steers co-simulation

FMU for Co-Simulation



Importer extracts and calls the FMU

FMU includes a solver algorithm to advance in time

Importer includes a co-simulation algorithm to couple the simulation with other models

Writing an Importer in Python

```
import fmpy

fmu = FMUSlave('model.fmu')
fmu.setVariable('a',1)

t=t_0
while t < t_end:
        fmu.doStep(t, stepSize)
        t = t + stepSize
result = fmu.getVariable('b')

fmu.terminate()
```

Python library FMPy allows to load and run FMU models in Python

Advantages

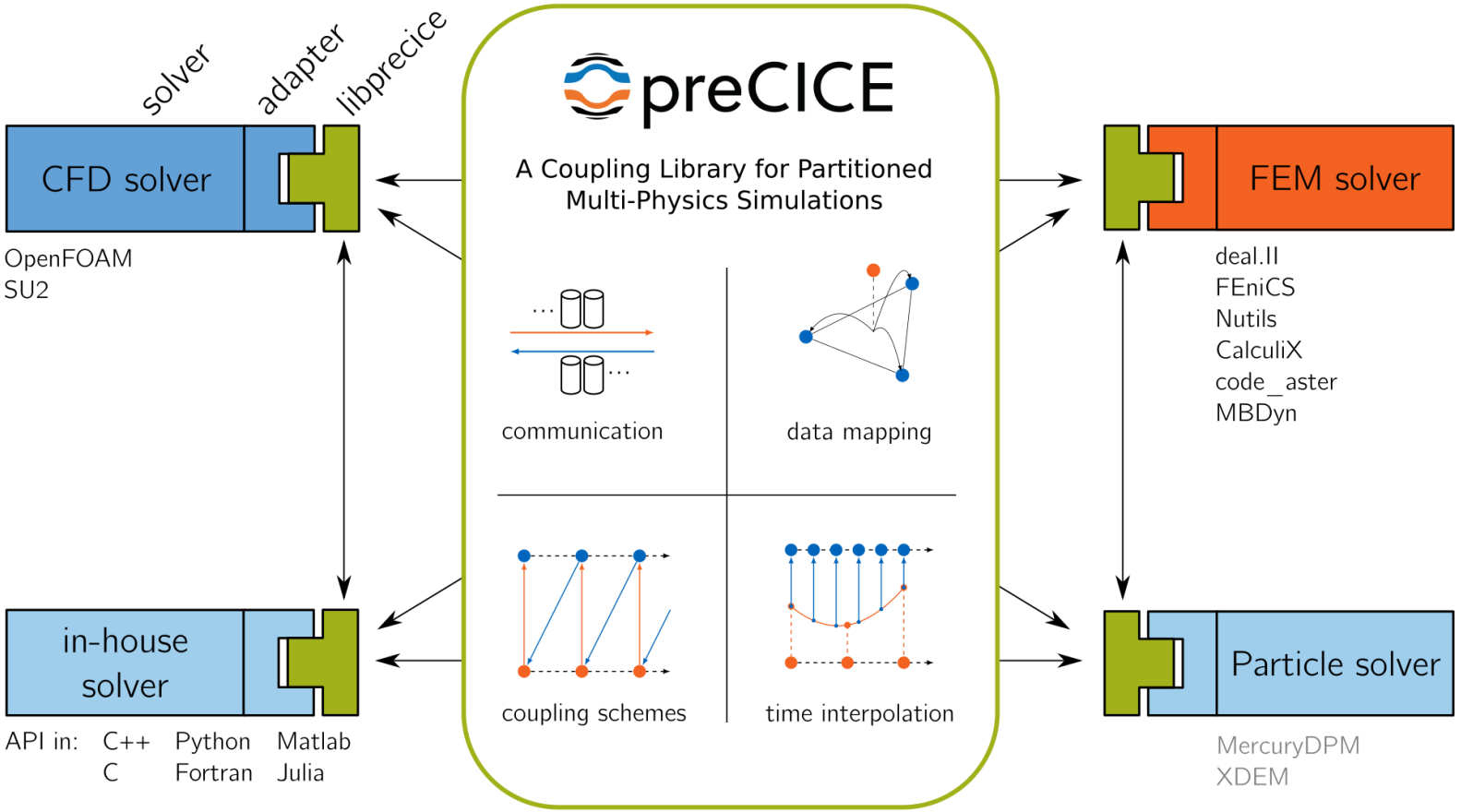
- **Independent standard**: Developed by the Modelica Association, not one tool vendor
- **Widely adopted**: More than 170 tools and many industrial users
- **Protection of Intellectual Property**: Compiled code can be distributed and used without source code

Limitations

- **Restricted:** The standard might forbid or complicate a functionality
- **Built for ODEs:** PDEs can be included in principal but
  - Data mapping is not supported
  - Distributed Memory is not supported

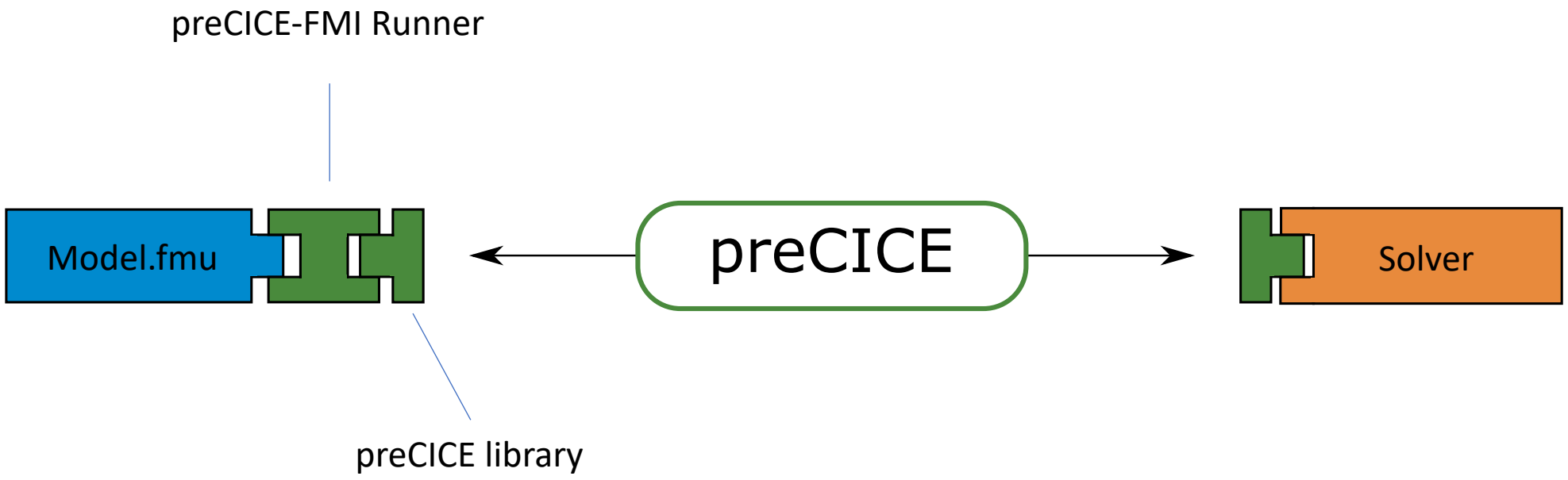
Concept

Overview preCICE



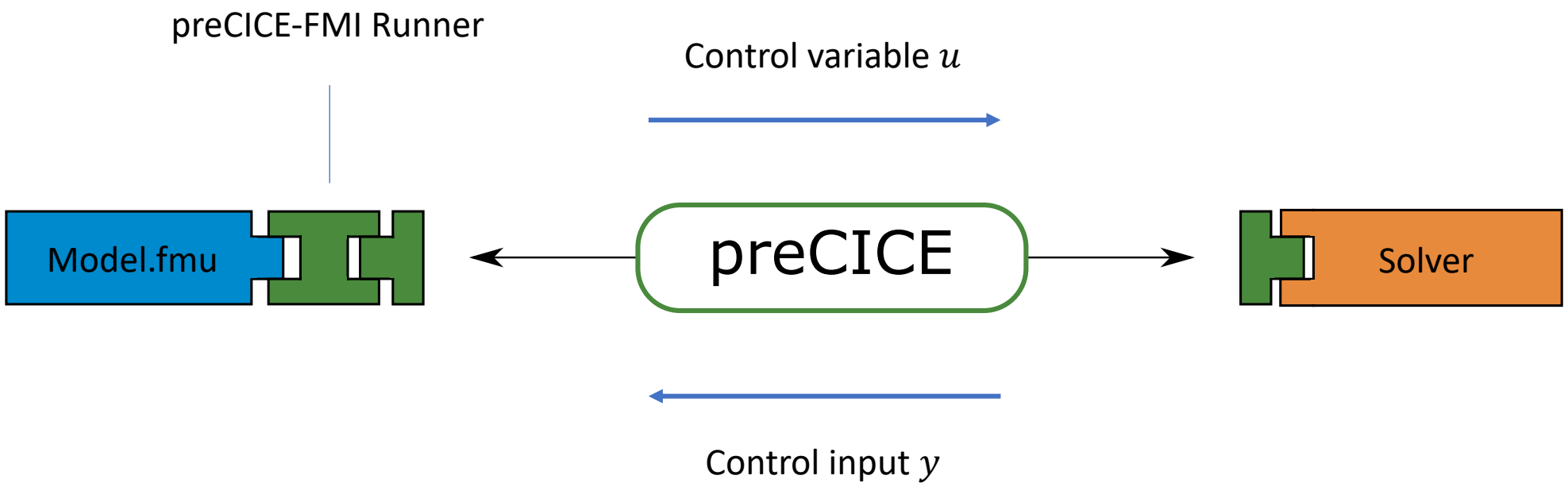
Source: "preCICE v2: A sustainable and user-friendly coupling library"
Gerasimos Chourdakis, Kyle Davis, Benjamin Rodenberg, Miriam Schulte, Frédéric Simonis, Benjamin Uekermann et al., Open Research Europe, 2022, 2:51.

preCICE-FMI Runner



Integrate FMU models in the preCICE coupling

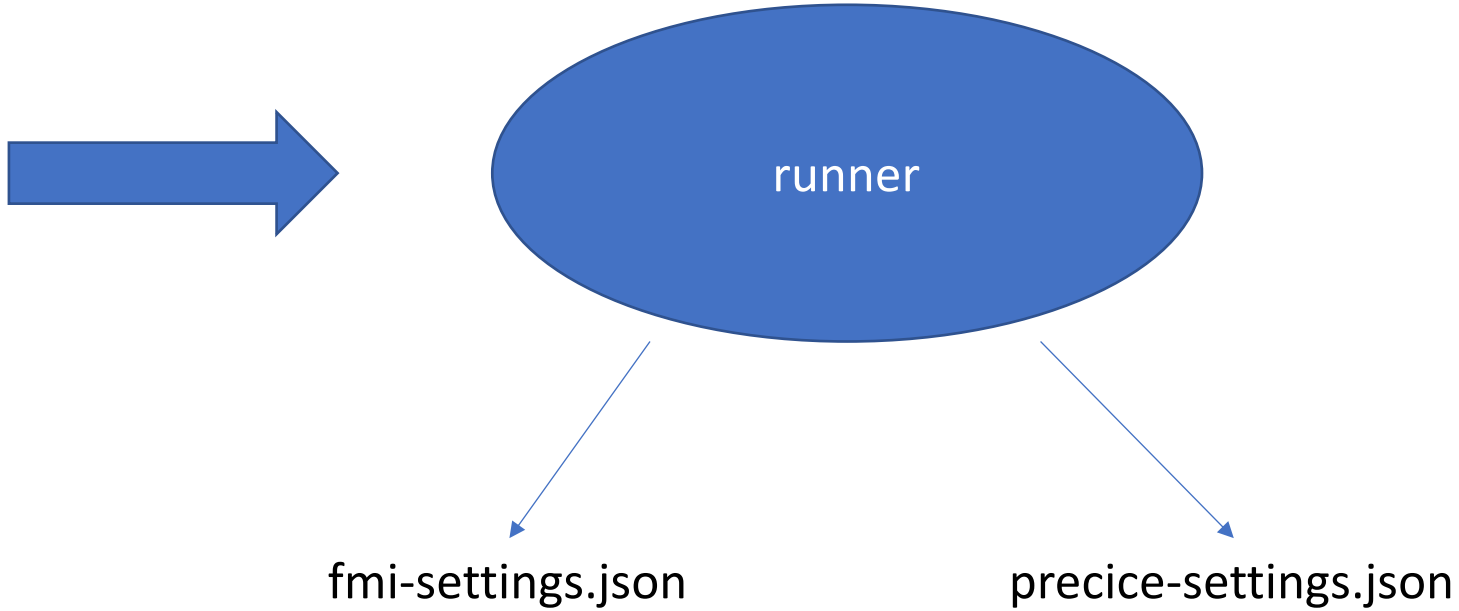
preCICE-FMI Runner for controller models



Use controller models as first step towards a general Runner

Implementation

```
pip install fmiprecice
```



Installation with PyPi

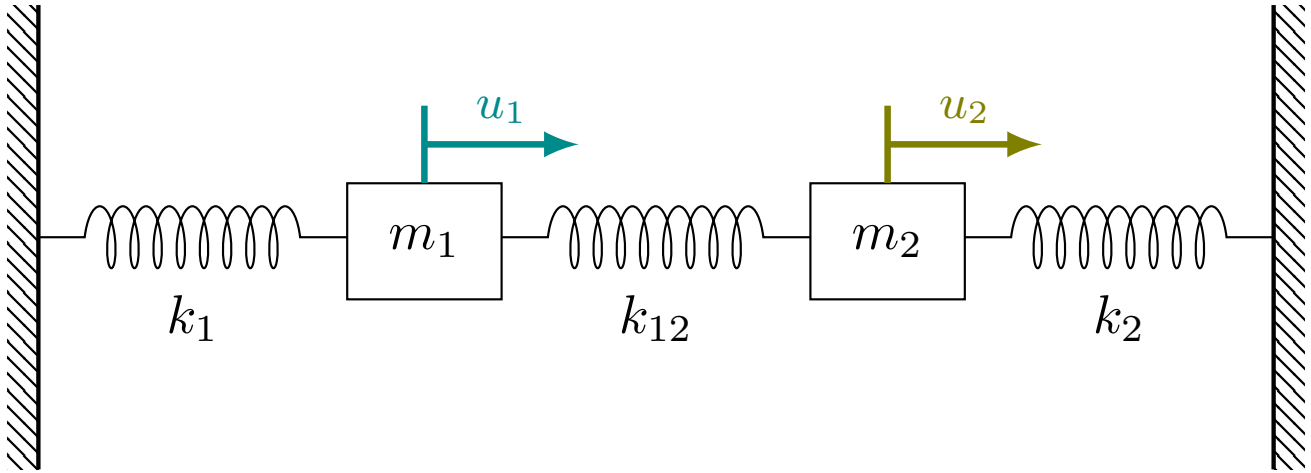
Example: fmi-settings.json

```
"simulation_params": {
        "fmu_file_name": "../model.fmu",
        "output_file_name": "./output.csv",
        "output": ["output_1", "output_2"],
        "fmu_read_data_names": ["read_data_1", "read_data_2"],
        "fmu_write_data_names":["write_data_1", "write_data_2"],
        "fmu_instance_name": "model_1"},
"initial_conditions": {
        "variable_1": 1.0
        "variable_2": 0.0}
```

Define the interaction with the FMU model

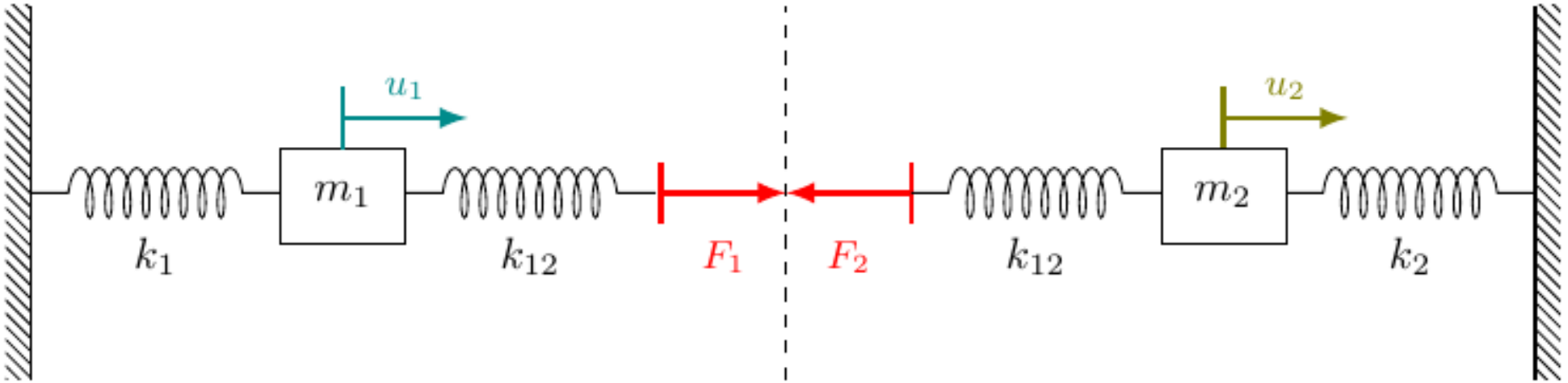
Test Cases

Test Case 1: Mass-spring Oscillator



Goal: Test Runner software for ODE coupling

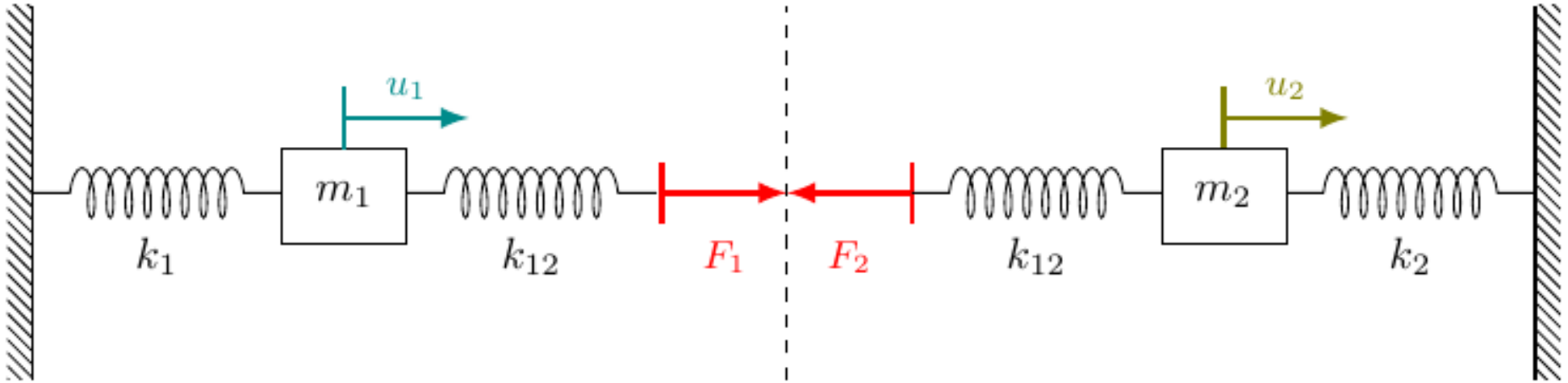
Partitioned setup



$$m_1\ddot{u}_1 = -(k_1 + k_{12})u_1 + F_2(t)$$
$$m_2\ddot{u}_2 = -(k_2 + k_{12})u_2 + F_1(t)$$

Cut spring k_{12} to get two decoupled IVPs with interface forces

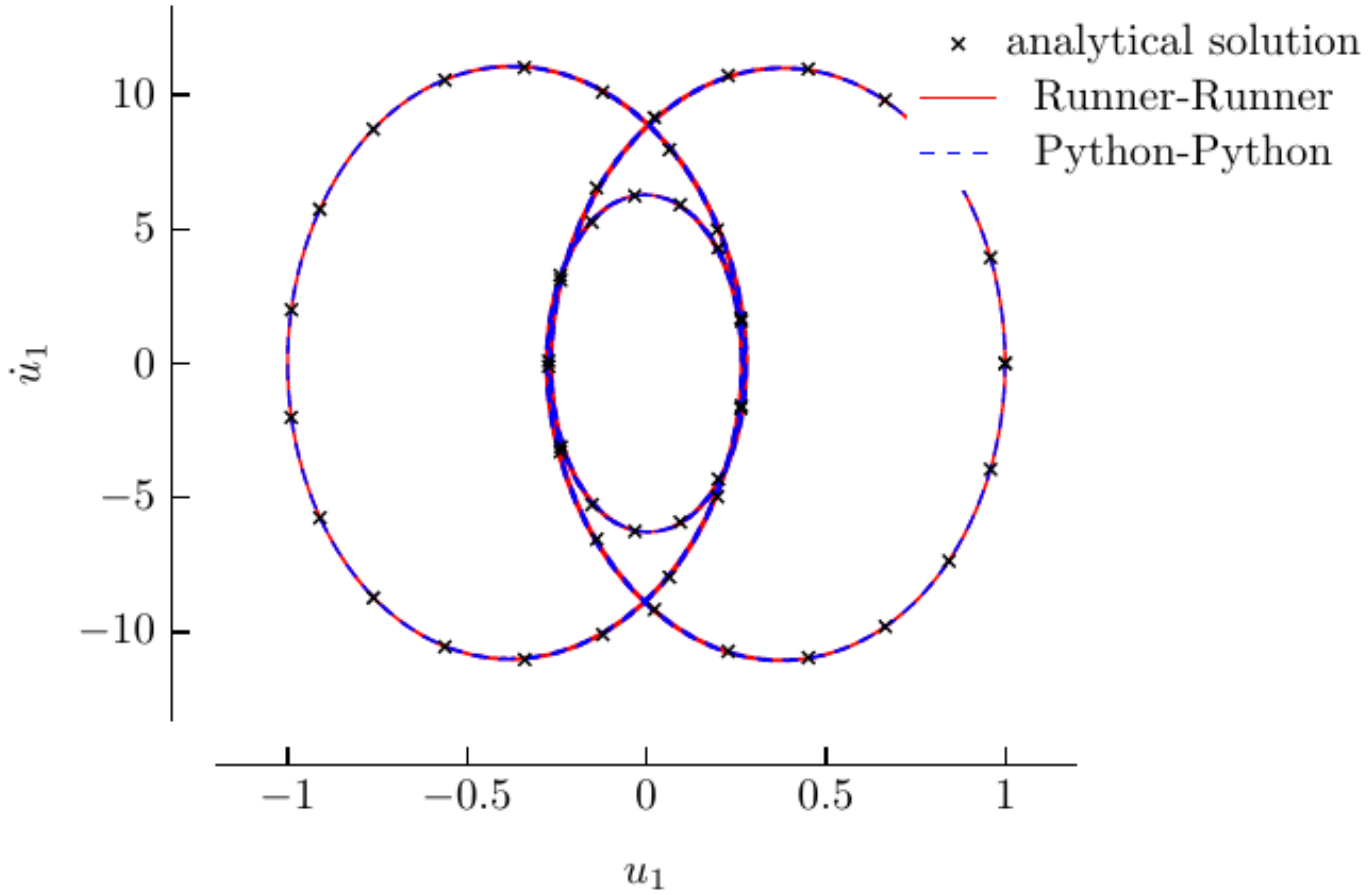
Partitioned setup



Compare the results of

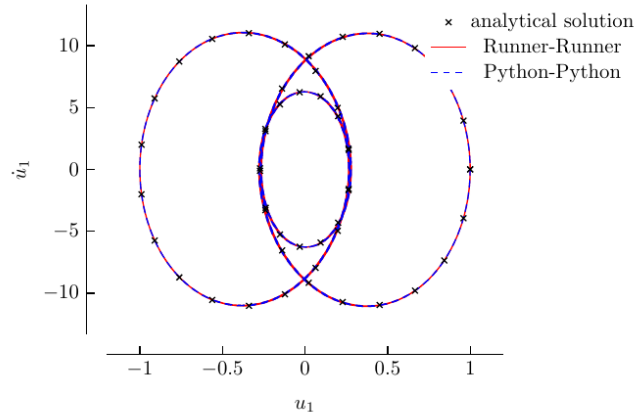
- Analytical calculation
- Numerical computation with Python
- Numerical computation with FMUs and Runner

Results



Trajectory of mass m_1

Results

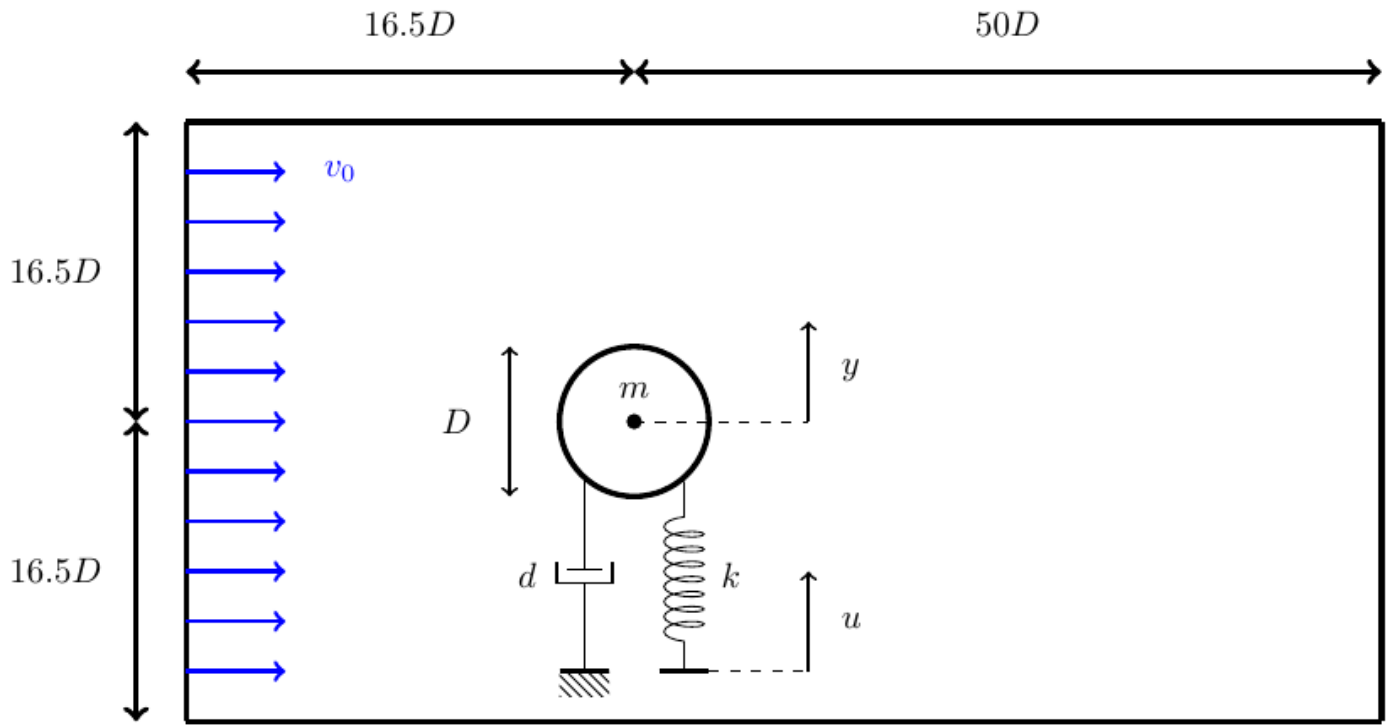


$$\| e \|_\infty (Runner - Analytical) \approx 3.48 \times 10^{-2}$$

$$\Delta \| e \|_\infty (Runner - Python) \approx 1 \times 10^{-4}$$

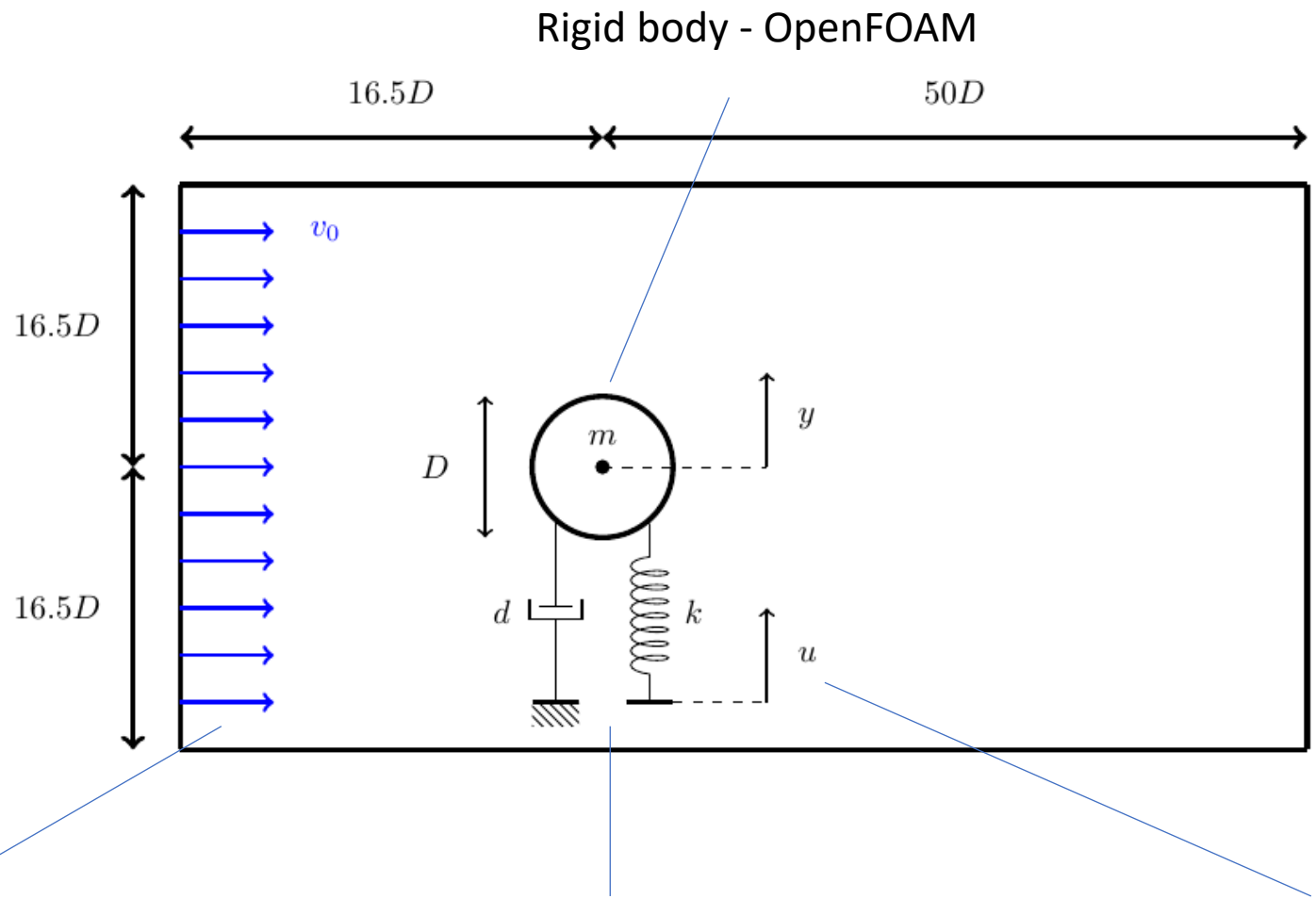
→ Runner works for ODE coupling

Test Case 2: Flow around cylinder

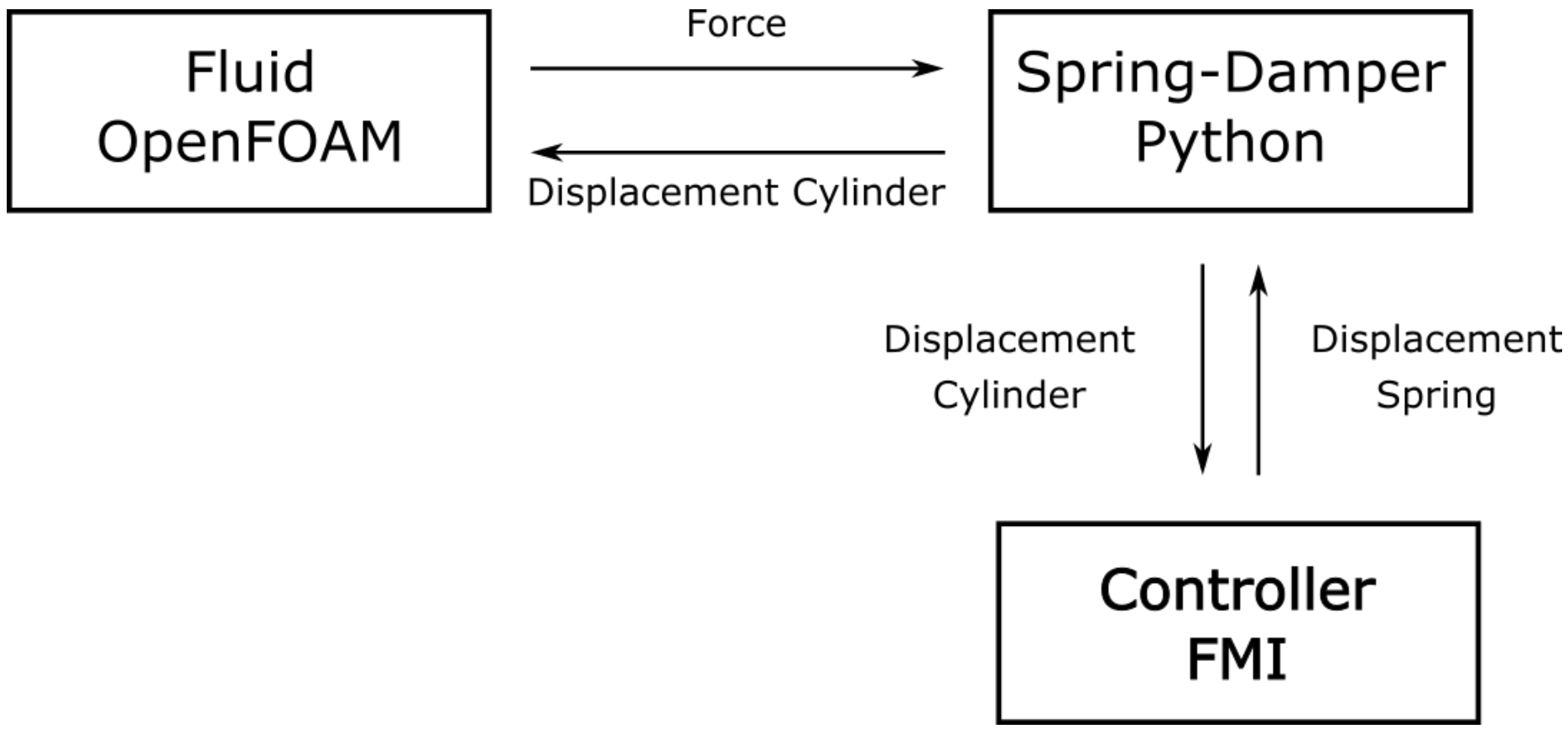


Goal: Test Runner software for PDE coupling

Partitioned setup

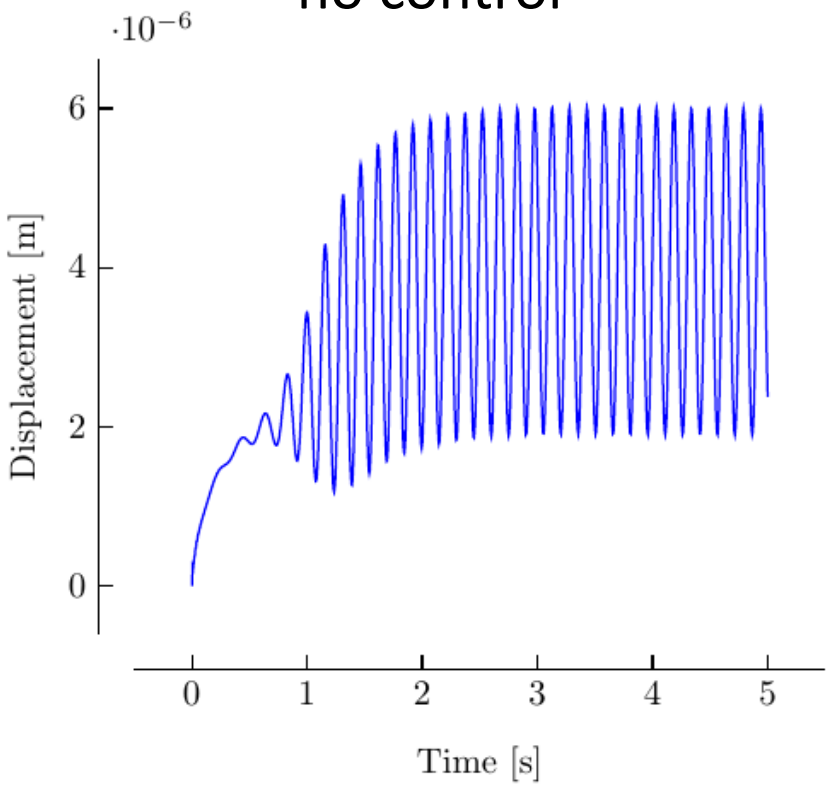


Coupling setup

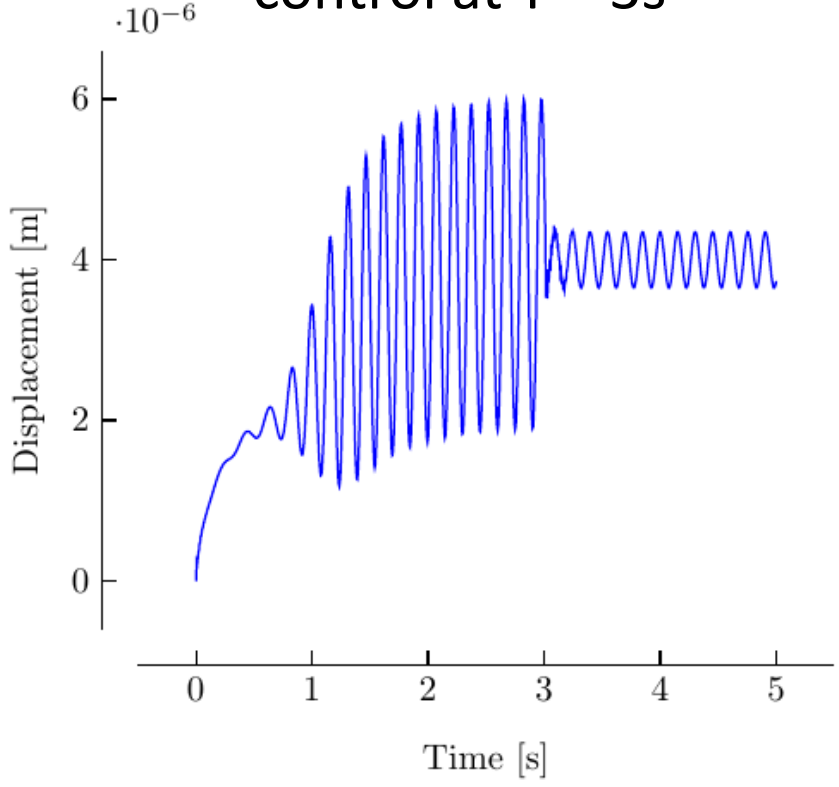


Results

no control



control at T = 3s



FMU controller is coupled, but the results can not be verified

Capabilities of the Runner software

+

- Compatible with FMI version 1, 2 and 3
- Explicit and implicit coupling schemes
- Can be configured for any FMU model

-

- Data exchange only at one vertex
- No handling of FMU vectors
- Compatible with preCICE version 2, not version 3

Conclusion and Outlook

The Runner is limited in its abilities, yet versatile enough for different use cases

→ Overcome minor limitations

→ Explore different coupling scenarios

→ Get user feedback

Summary

- First coupling of FMU models with preCICE
- FMUs can be coupled with PDE-based solvers
- Some limitations remain in the functionality

github.com/precice/fmi-runner

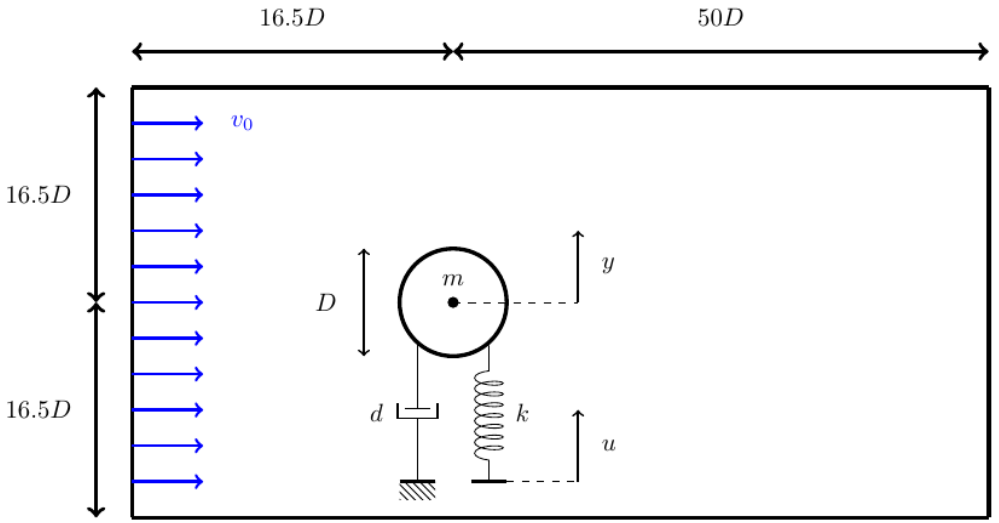
https://doi.org/10.18419/darus-3408

st150067@stud.uni-stuttgart.de

Appendix

Flow around cylinder: Setup

Variable	Value	Unit
D	0.0016	$[m]$
d	0.0043	$[N/s]$
k	69.48	$[N/m]$
m	0.03575	$[kg]$
v_0	0.068	$[m/s]$
Re	108.83	$[-]$

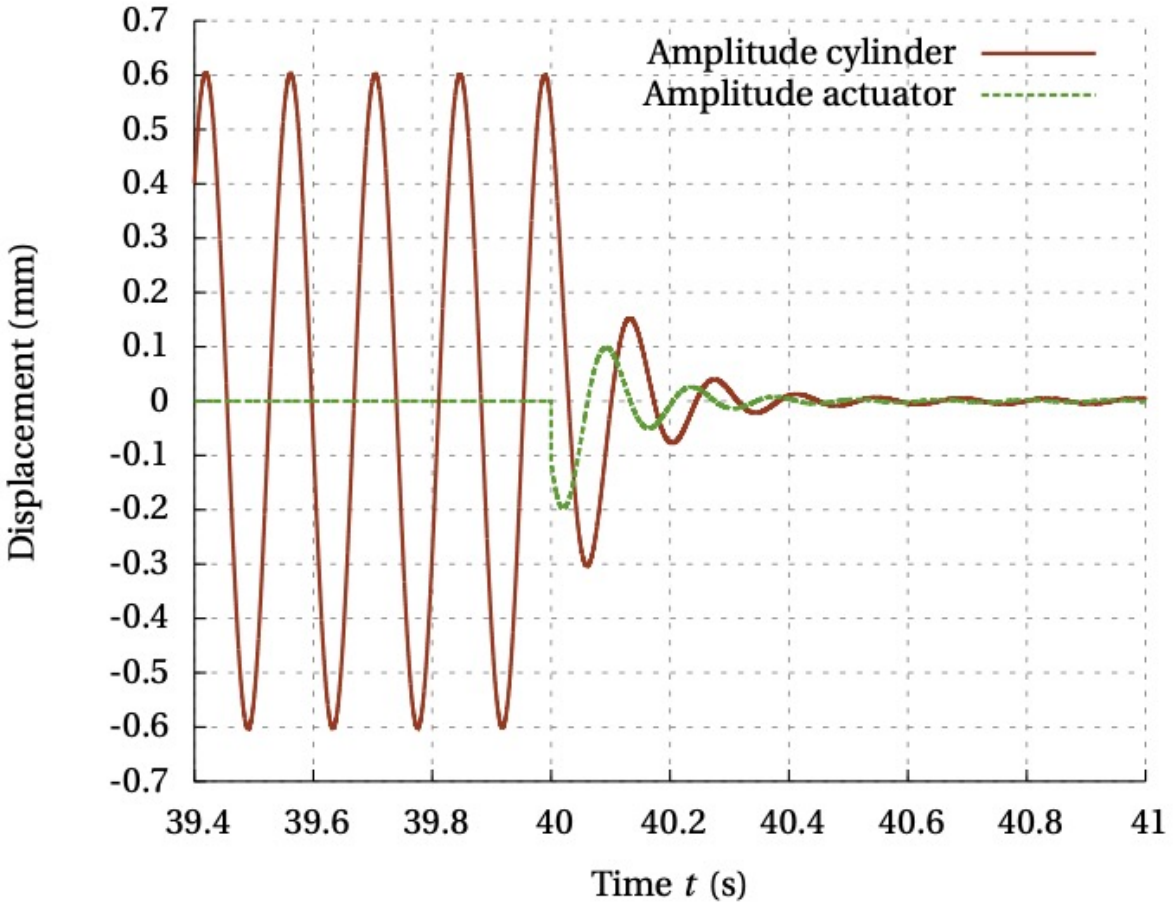


- Laminar flow
- [1] and [2] report a stable oscillation and a lock-in effect for this setup in experiments and numerical simulations

[1]: P. Anagnostopoulus and P.W. Bearman. Response characteristics of a vortex-excited cylinder at low reynolds numbers. *Journal of Fluids and Structures*, 6(1):39–50, 1992.

[2]: Sicklinger, "Stabilized Co-Simulation of Coupled Problems Including Fields and Signals", Technical University of Munich, Dissertation, pp. 126 - 135, https://www.researchgate.net/publication/269705153_Stabilized_Co-Simulation_of_Coupled_Problems_Including_Fields_and_Signals (last visited: 20.04.2023)

Flow around cylinder: Reference results



Frequency of oscillation

- $f_{ref} = 6.5\ Hz$
- $f_{thesis} = 6.5\ Hz$

Amplitude of oscillation

- $\hat{y}_{ref} = 6{\times}10^{-4}\ m$
- $\hat{y}_{thesis} = 2{\times}10^{-6}\ m$

The error was located in the force calculation of OpenFOAM, but could not be corrected.

Source: Sicklinger, “Stabilized Co-Simulation of Coupled Problems Including Fields and Signals”, Technical University of Munich, Dissertation, pp. 134, https://www.researchgate.net/publication/269705153_Stabilized_Co-Simulation_of_Coupled_Problems_Including_Fields_and_Signals (last visited: 20.04.2023)