TLK-Thermo GmbH

Simulator Suite

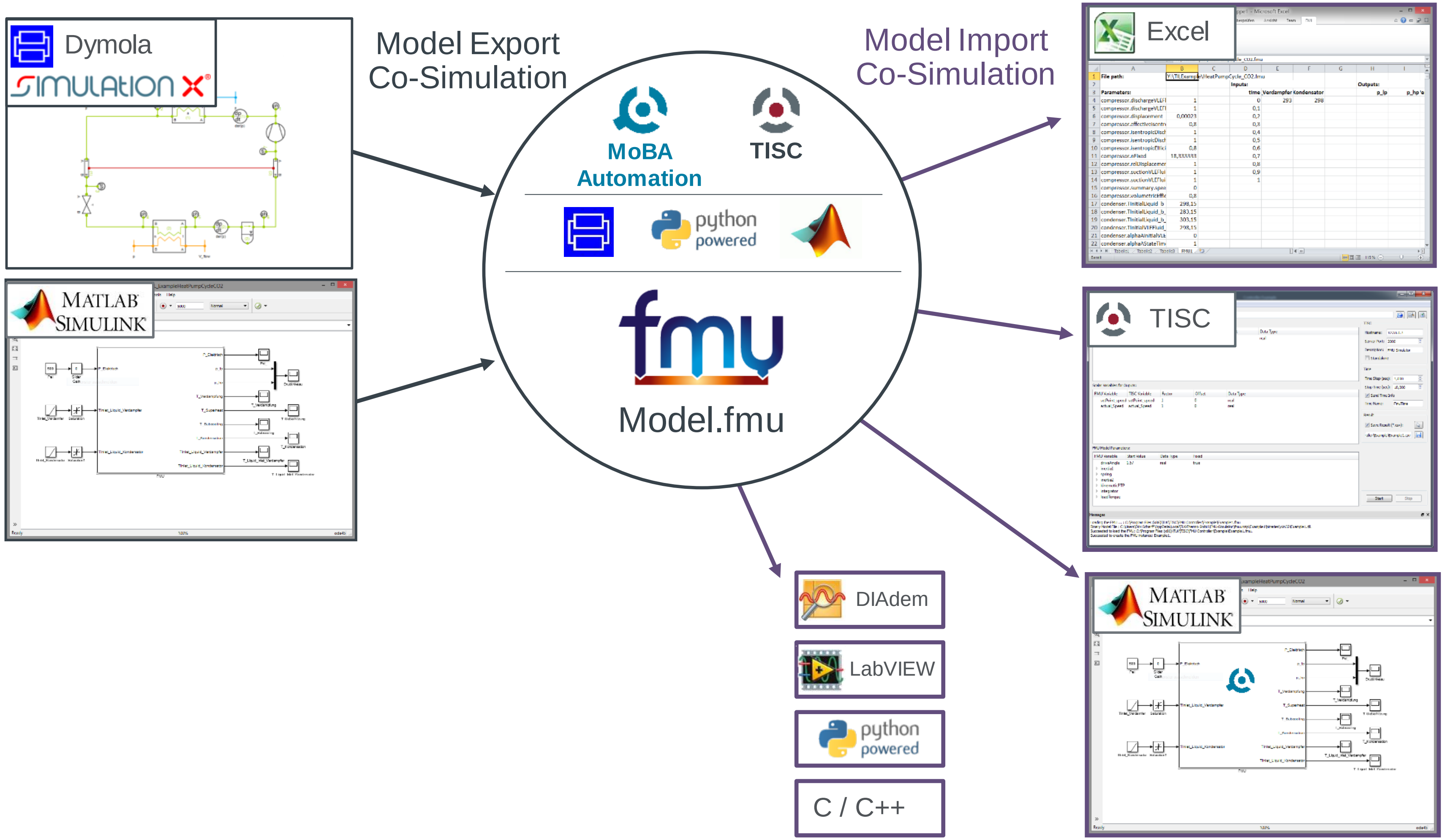
Software package for the simulation and evaluation of models

Introduction

Functional Mock-Up Interface

Products:

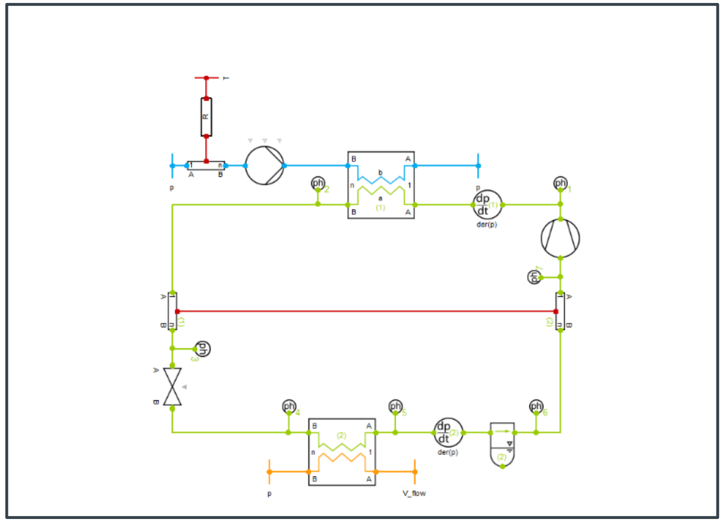
- TLK-Simulator for Excel
- TLK-Simulator for LabVIEW
- TLK-Simulator for TISC
- TLK-Simulator for TRNSYS
- TLK-Simulator for Simulink
- TLK-Simulator for Python
- TLK-Simulator for C/C++

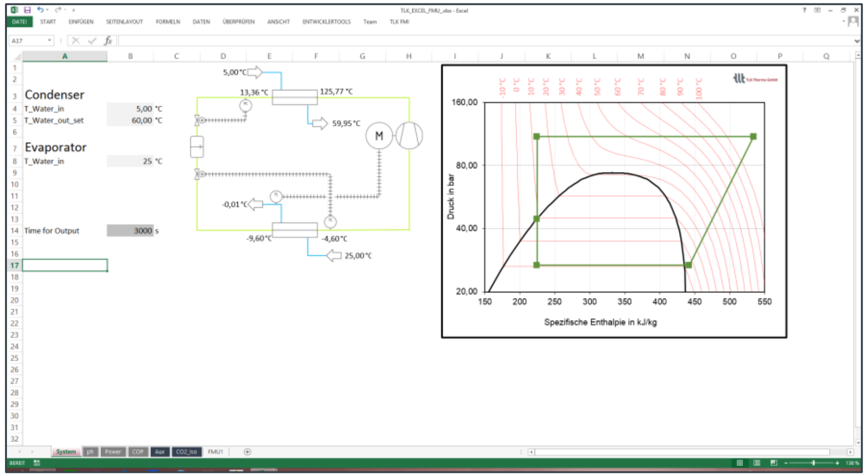


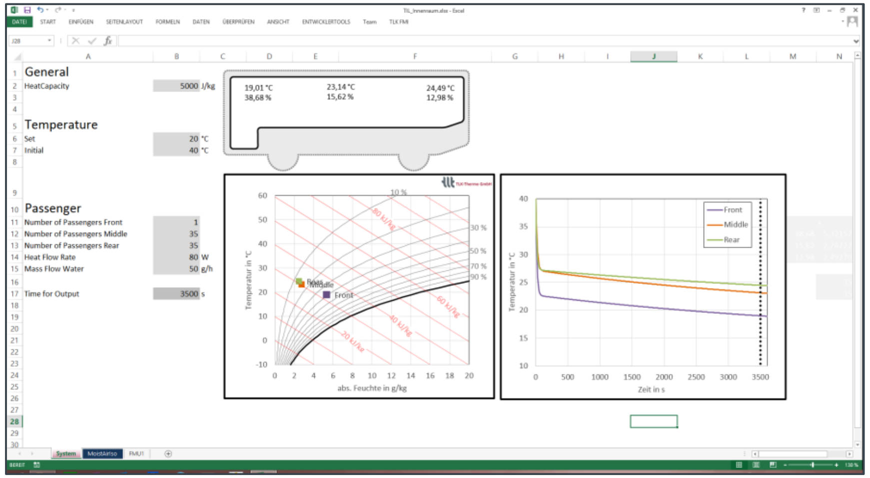
Introduction

Using sophisticated models in Excel

- Connect existing models
- Simulation and Visualization
- Real-Time Simulation and Parameter Variation



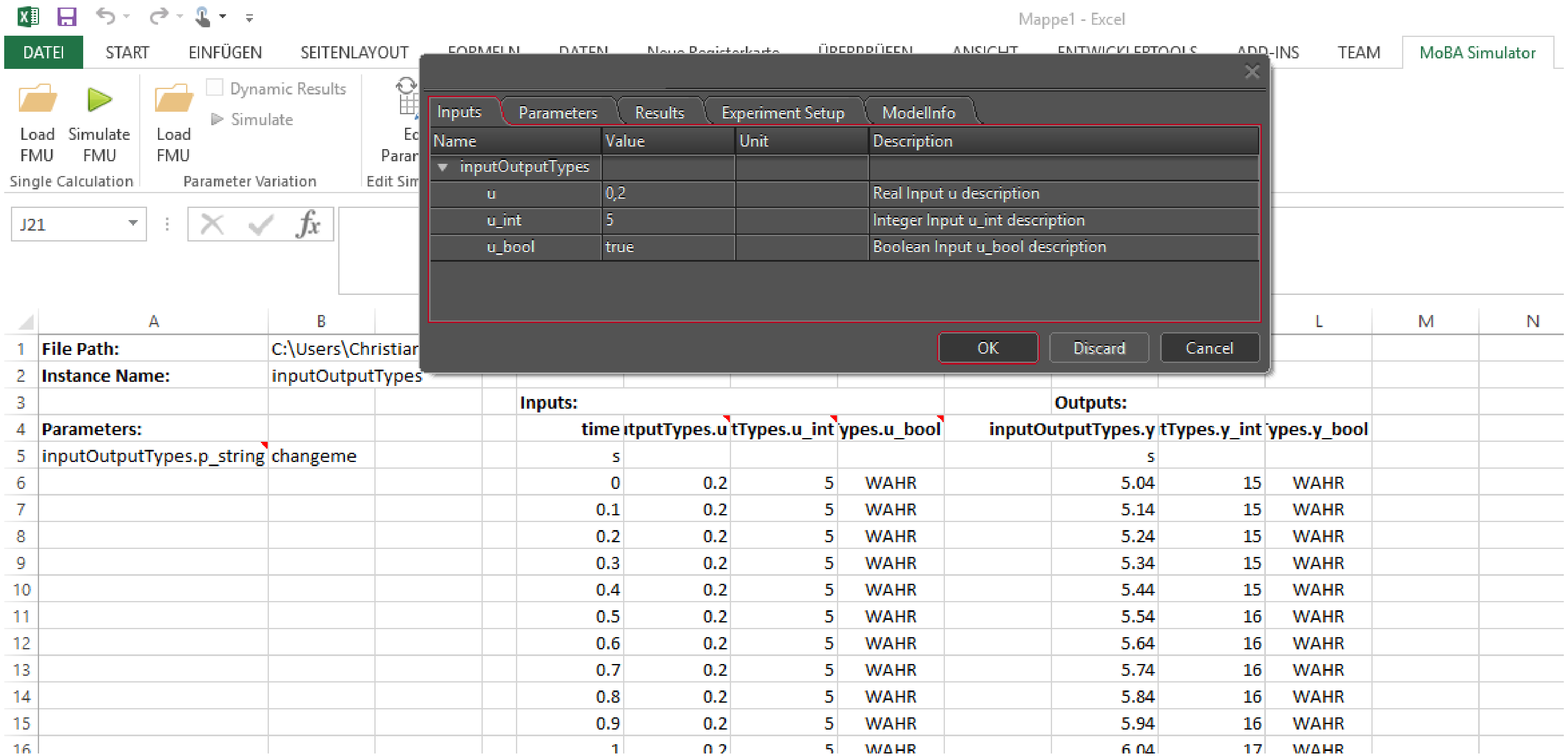




TLK-Thermo GmbH

Interfacing and Simulation

TLK-Simulator for Excel



Interfacing and Simulation

Solving Methods

ODE-Solver:

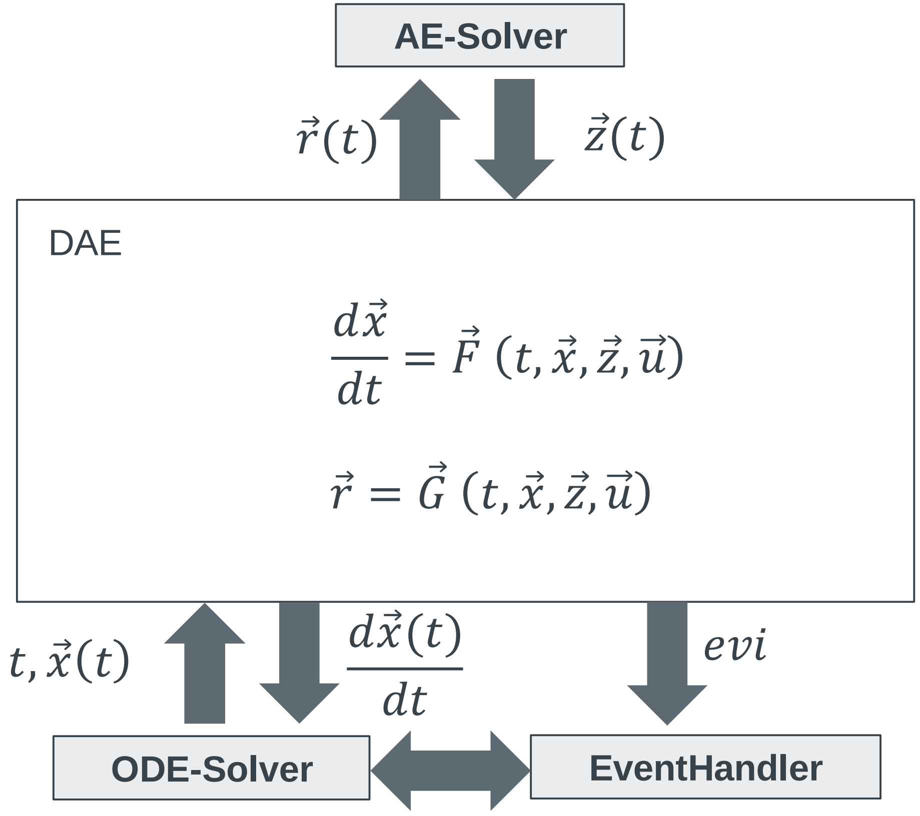
- A Sundials CVode(S) (BDF)
- A Sundials ARKode (Runge-Kutta)
- A Explicit Euler
- D Modified DASSL

Algebraic Solver:

- A Sundials Kinsol
- A Modified Newton Raphson (including Line-Search)

Legend for level of product maturity:

- A Available Product
- D Development Status



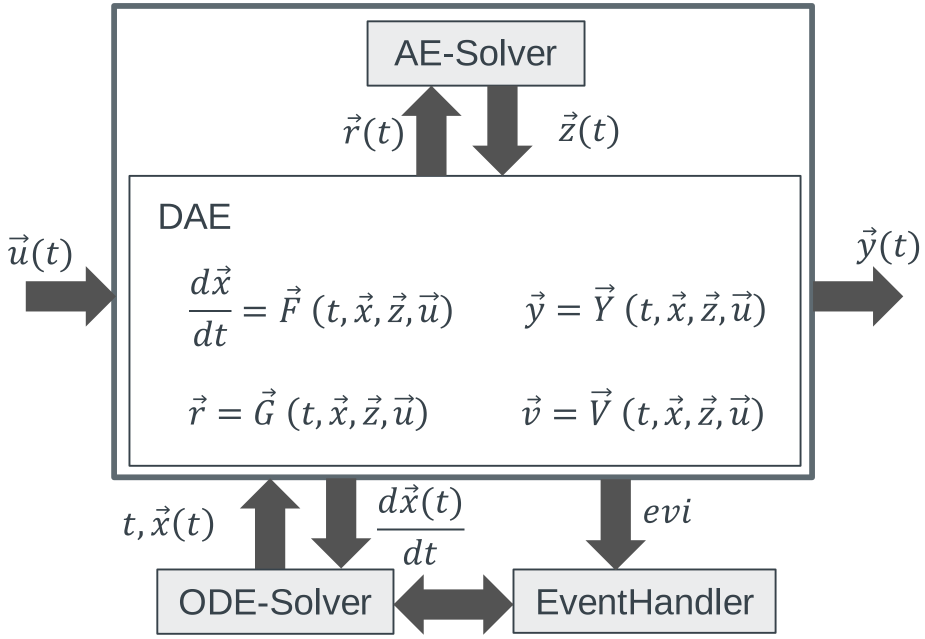
Interfacing and Simulation

FMI Cross-Check of current tool versions

Source of FMU		Degree of fulfilment (Model Exchange / Co-Simulation)	
Software	**Version**	**FMI 1.0**	**FMI 2.0**
AMESim	14, 15	37,5% (6/16)	100% (6/6)
CATIA	R2015x, R2016x	90% (36/40)	100% (36/36)
DS-FMU_Export_from_Simulink	2.1, 2.1.1, 2.2, 2.3	97,3% (74/76)	100% (74/74)
dSPACE_TargetLink	2018-B	-	100% (6/6)
Dymola	2015FD01 - 2019FD01	100% (148/148)	100% (128/128)
FMIToolbox_MATLAB	2.1, 2.3	100% (36/36)	100% (26/26)
JModelica.org	1.15	87,5% (7/8)	100% (8/8)
MapleSim	2016.2, 2018, 2019	100% (30/30)	94,4% (34/36)
MWorks	2016	71,4% (20/28)	100% (4/4)
Silver	3.2, 3.3, 3.5	100% (29/29)	100% (6/6)
SimulationX	3.7.41138, 4.0.4	77,7% (21/27)	88,8% (24/27)

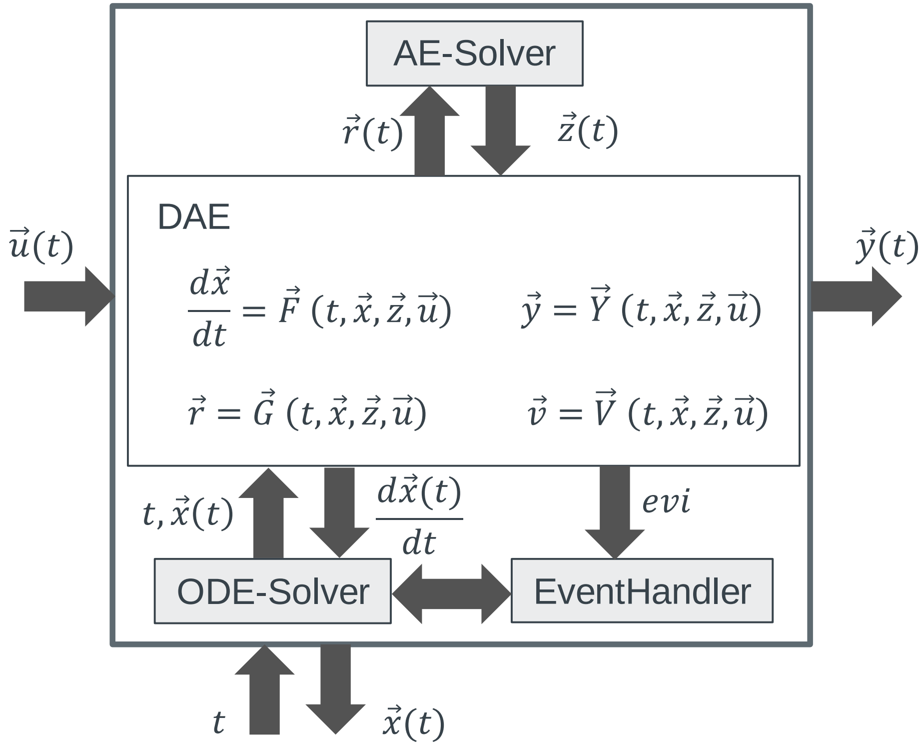
Model Analysis

Mathematics of FMUs and DAEs



Model Exchange FMU

(no ODE-Solver)



Co-Simulation FMU

(contains ODE-Solver)

Model Analysis

Partial Derivatives introduced in FMI 2.0

- Derivatives w.r.t. **differential states** x

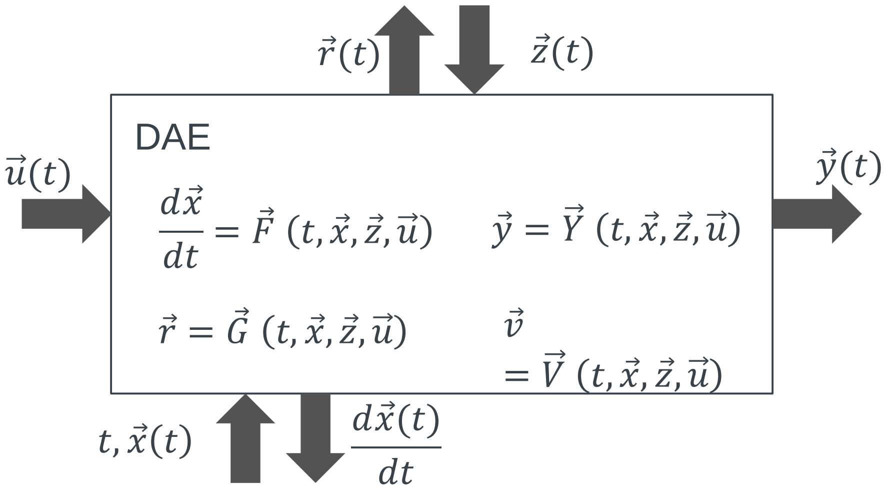
$$J_x = \frac{dF}{dx} = \frac{d\frac{dx}{dt}}{dx}$$

  - Implicit ODE-Solver
  - Finding the steady state of dynamic models

Derivatives w.r.t. **inputs** u:

$$J_u = \frac{dF}{du} = \frac{d\frac{dx}{dt}}{du}$$

  - Sensitivities (CVodeS)
  - Optimal Control



If derivatives are not available, they are calculated numerically

Model Analysis

Detailed Mathematical Analysis

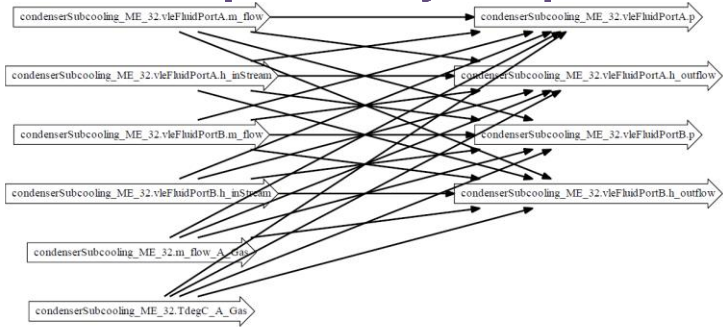
State Space Form

The State Space Form of the system has 0 differential state, 3 inputs and 7 outputs.

A =
B =
C =
D =

	fan.p_h_Pa	fan.V_flow_in_m3ps	fan.n_Hz
fan.P_loss	0.017	0.0134	0.00361
fan.P_loss_blade	0.00606	0.011	-0.00603
fan.P_loss_impact	0	-3.74e+03	-3.1e+03
fan.dp	-645	-489	-2.65e+03
fan.P_shaft	1.09e+03	0.989	204
fan.P_hyd	161	43.4	28.9
fan.V_flow	176	-28.4	0.000579

Direct Dependency Graph



Determination of Time Constants

Contribution to Physical States:

Number	Time Constant	Contribution to States
40	0.0655 s	40.3% to heatPumpCycle_R407C.evaporator.vleFluidCell_a[20].h 25.6% to heatPumpCycle_R407C.condenser.vleFluidCell_a[1].h 8.7% to heatPumpCycle_R407C.evaporator.vleFluidCell_a[19].h 3.3% to heatPumpCycle_R407C.condenser.vleFluidCell_a[2].h 3.0% to heatPumpCycle_R407C.condenser.portB_a.p 2.3% to heatPumpCycle_R407C.evaporator.vleFluidCell_a[18].h 1.9% to heatPumpCycle_R407C.condenser.vleFluidCell_a[3].h 1.1% to heatPumpCycle_R407C.condenser.vleFluidCell_a[5].h 1.0% to heatPumpCycle_R407C.condenser.vleFluidCell_a[6].h 1.0% to heatPumpCycle_R407C.condenser.vleFluidCell_a[4].h
41	0.0655 s	40.3% to heatPumpCycle_R407C.evaporator.vleFluidCell_a[20].h 25.6% to heatPumpCycle_R407C.condenser.vleFluidCell_a[1].h 8.7% to heatPumpCycle_R407C.evaporator.vleFluidCell_a[19].h 3.3% to heatPumpCycle_R407C.condenser.vleFluidCell_a[2].h 3.0% to heatPumpCycle_R407C.condenser.portB_a.p 2.3% to heatPumpCycle_R407C.evaporator.vleFluidCell_a[18].h 1.9% to heatPumpCycle_R407C.condenser.vleFluidCell_a[3].h 1.1% to heatPumpCycle_R407C.condenser.vleFluidCell_a[5].h 1.0% to heatPumpCycle_R407C.condenser.vleFluidCell_a[6].h 1.0% to heatPumpCycle_R407C.condenser.vleFluidCell_a[4].h

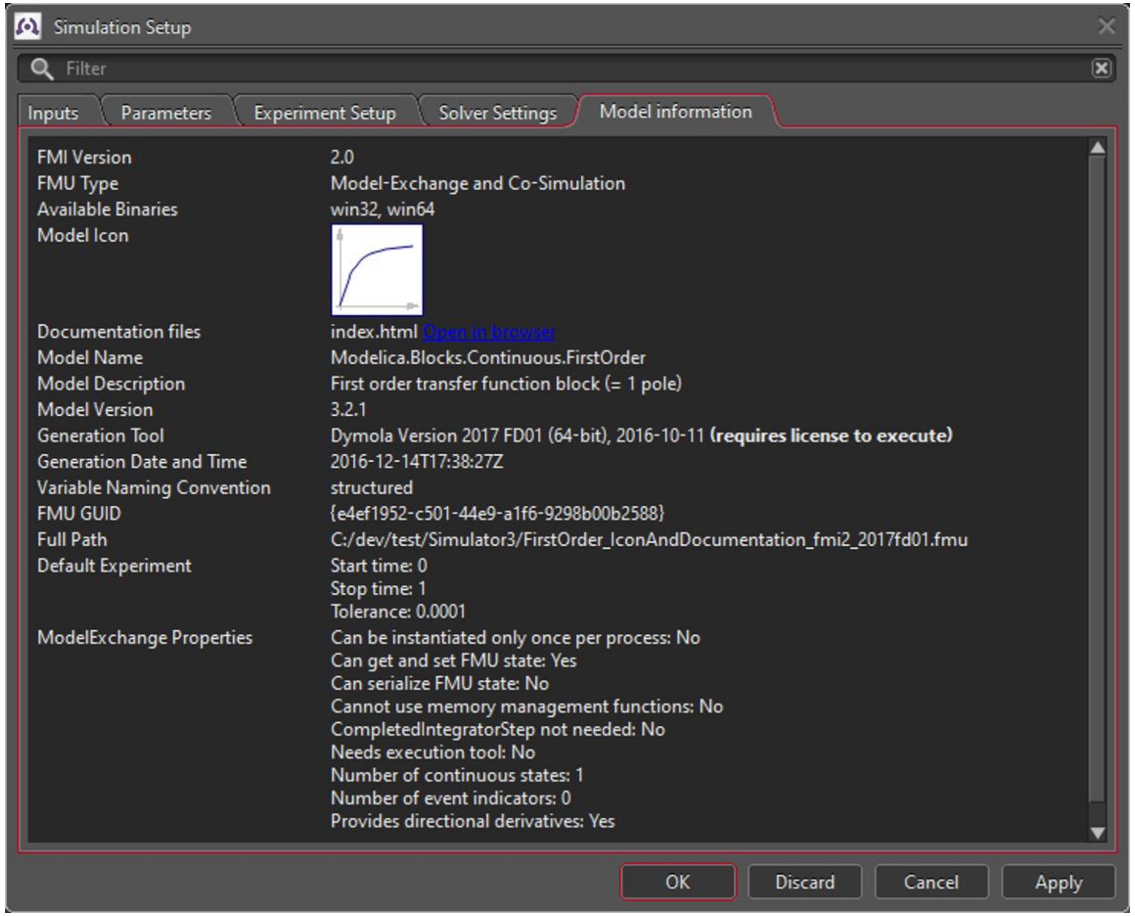
Sensitivities

- _SolverDebug
- automotivePITuning
  - sens[automotivePITuning.compressor.rotatoryFlange.phi,automotivePITuning.u1]
  - sens[automotivePITuning.compressor.rotatoryFlange.phi,automotivePITuning.u]
  - sens[automotivePITuning.condenser.moistAirCell[1, 1].H_WallPlusFilm,automotivePITuning.u1]
  - sens[automotivePITuning.condenser.moistAirCell[1, 1].H_WallPlusFilm,automotivePITuning.u]
  - sens[automotivePITuning.condenser.moistAirCell[1, 1].pressureDropState,automotivePITuning.u1]
  - sens[automotivePITuning.condenser.moistAirCell[1, 1].pressureDropState,automotivePITuning.u]
  - sens[automotivePITuning.condenser.moistAirCell[1, 2].H_WallPlusFilm,automotivePITuning.u1]
  - sens[automotivePITuning.condenser.moistAirCell[1, 2].H_WallPlusFilm,automotivePITuning.u]
  - sens[automotivePITuning.condenser.moistAirCell[1, 2].pressureDropState,automotivePITuning.u1]
  - sens[automotivePITuning.condenser.moistAirCell[1, 2].pressureDropState,automotivePITuning.u]
  - sens[automotivePITuning.condenser.moistAirCell[1, 3].H_WallPlusFilm,automotivePITuning.u1]

Model Analysis

Detailed Model Information

- Helpful model information about the simulated system
- Various options to customize the simulation in detail, for example by selecting and setting the solver (e.g. Sundials CVode, Sundials ARKode, Explicit Euler)
- Existing solvers are explicitly selected to solve complex thermodynamic systems



Thank you

If you have any questions,
please don’t hesitate to contact us at
simulator@tlk-thermo.com

Or your contact person
Johannes Schulz
j.schulz@tlk-thermo.com

TLK-Thermo GmbH
Rebenring 31
38106 Braunschweig
www.tlk-thermo.com

Tel.: +49/531/390 76 - **241**
Fax: +49/531/390 76 - 29