



**3D**EXPERIENCE

# MODELICA / DYMOLA and SIMULIA Abaqus working together

Tomáš Ivančík  
SIMULIA Sales

[tomas.ivancik@3ds.com](mailto:tomas.ivancik@3ds.com)

*MODELICA seminář 2015*  
26.6.2015

# 3DS Technologies | Broadening the Portfolio

Simulation  
For Product, Nature and Life

SolidWorks  
Simulation



CATIA  
Analysis



CAD Design  
Simulation

Abaqus



FEA Multiphysics  
Simulation

Isight



Process  
Integration &  
Design  
Optimization

Tosca



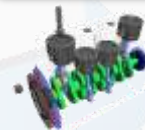
Non-parametric  
Optimization

fe-safe



Fatigue

Simpack



Multi-Body  
Dynamics



3DEXPERIENCE®



Uniting the  
Virtual & Real  
Worlds for All  
Industries

SolidWorks

CATIA

SIMULIA

Geensoft

Dymola

System  
Simulation



SFE



Conceptual  
Engineering

Simpoe



Plastic  
Molding



# What is Realistic Simulation?

“Realistic Simulation” is a simulation that is physically realistic and “life like” in every way



Courtesy Mechanical Design and Analysis Corporation, 2010 SCC

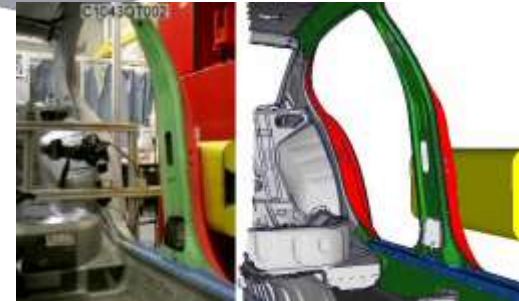


“Predictive Crashworthiness Simulation in a Virtual Design Process without Hardware Testing”, Jürgen Lescheticky, Hariakto Hooputra and Doris Ruckdeschel, BMW Group, SIMULIA Summer Conference, May 2010

Statistical distribution of impact damage



Courtesy of University of Zagreb, 2010 SCC



Courtesy of BMW Group, 2010 SCC

# Why simulate instead of test?

## *Physical Testing*

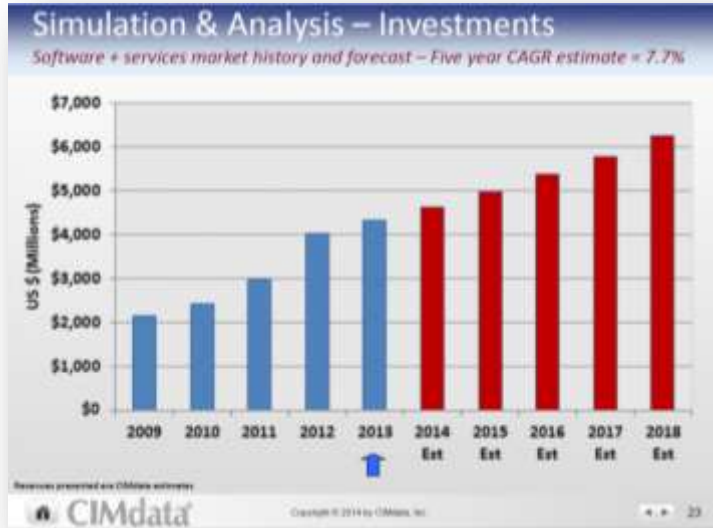
- ▶ Very expensive (often \$100,000/test)
- ▶ Time-consuming (weeks or months)
- ▶ Can only be done in an expensive lab
- ▶ Requires exotic equipment
- ▶ Can only be done at certain times
- ▶ Says *what* happened (“pass/fail”), but not *why*

## *Virtual Testing (Realistic Simulation)*

- ▶ Cheap (\$500/simulation-result)
- ▶ Quick (hours)
- ▶ Can be done anywhere
- ▶ Uses only a computer and software
- ▶ Can be done 24/7/365
- ▶ Says *what* happened AND *why*

# Simulation & Analysis is a Central, Up-front Role

*It must be... recognized and managed as a strategic capability...*



Courtesy CIMdata

## Simulation & Analysis Governance A Strategy to Advance the Value of S&A

April 2014

*“simulation must be a strategic capability”*

*“simulation is the fastest growing segment of PLM”*

*“simulation has a central, up-front role”*

**-CIMdata**

CIMdata, Inc.  
3909 Research Park Drive  
Ann Arbor, MI 48108  
+1 734 668 9922  
www.CIMdata.com

**CIMdata**

Questions  
world.

# Living Heart Project

- ▶ Simulation is at the forefront of a revolution in cardiac care



- ▶ Visit the Living Heart Project in 3D

**The Living Heart**

**SIMULIA Spearheads The Living Heart Project**  
Bringing the medical community together for improved patient care

**3DIXcite**  
Realtime Deformable Visualization

**SIMULIA**  
COMMUNITY NEWS  
May 2014

**IN HARMONY**  
SIMULATING PRODUCT  
NATURE, AND LIFE

**SIMULIA**  
Community News  
May 2014



SIMULIA  
Community News  
May 2014

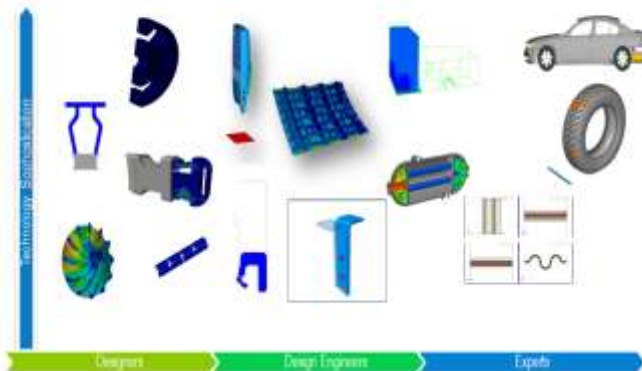
# Metóda konečných prvků – MKP (FEM)

...je numerická metoda sloužící k simulaci průběhů napětí, deformací, vlastních frekvencí, proudění tepla, jevů elektromagnetismu, proudění tekutin atd. na vytvořeném fyzikálním modelu. Její princip spočívá v diskretizaci spojitého kontinua do určitého (konečného) počtu prvků, přičemž zjišťované parametry jsou určovány v jednotlivých uzlových bodech.



One tool for all

Simulation solution covering Designers to Simulation Experts



# Solutions for the entire range of industries

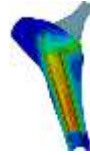
## Consumer Packaged Goods

Plastic and Glass Forming  
Conveyor Systems  
Container Drop  
Pressure Analysis  
Thermal Analysis  
Bottle Sealing  
Adhesives



## Life Sciences

Tissue Modeling  
Surgical Equipment  
Stents  
Drug Delivery  
Orthopedics  
Medical Packaging



## Aerospace & Defense

Avionics  
Landing Gear  
Aerostructures  
Aeroengines  
Composites  
Defense Systems  
Space Systems



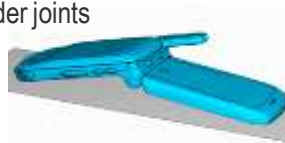
## Automotive & Transportation

Chassis  
Body  
Tires  
Interiors  
Crashworthiness  
Brake Systems  
Powertrain  
Electronics



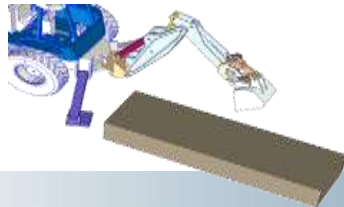
## High Tech

Thermal cycling of solder joints  
Drop Testing  
Vibration Analysis  
Semiconductors  
Circuit Boards  
Hand-held Devices  
Computers & Peripherals



## Industrial Equipment

Nonlinear Stress Analysis  
Thermal Analysis  
Cyclic Loading  
Flexible Multibody Dynamics  
Soil-Structure Interaction



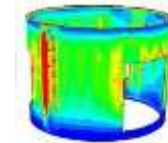
## Architecture & Construction

Earthquake loading  
Structural integrity due to fire  
Concrete analysis  
Soil-pore interaction  
Failure limits



## Energy

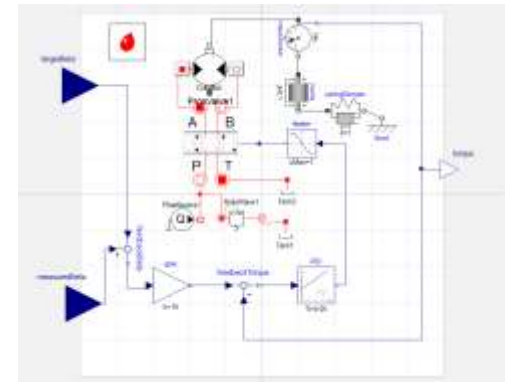
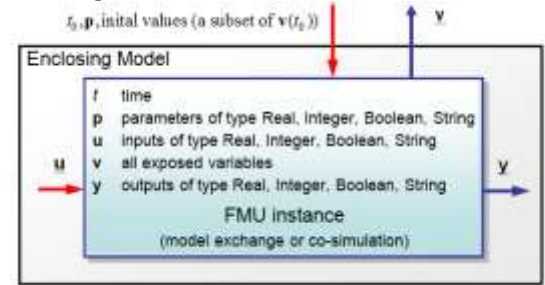
Wave loading on rigs and pipelines  
Piping and pressure vessels  
Thermal analysis  
Blast loading  
Drop or Impact





# What is a Functional Mockup Unit (FMU)?

- ▶ A self-describing simulation model
  - ▷ Adhering to an open standard titled Functional Mockup Interface (FMI)
- ▶ With scalar input and output variables
- ▶ Commonly provided in file format (e.g. mymodel.fmu)
- ▶ Conceptually similar to
  - ▷ User subroutines
  - ▷ Dymola DLL files used in the Abaqus / Dymola interface
  - ▷ Substructure / superelement files



Dymola component

# What is a Functional Mockup Unit (FMU)?

- ▶ Two general categories of FMUs are available
  - ▷ Model Exchange format – the FMU provides equation information
    - ▶ More difficult to handle
    - ▶ Easier to generate
  - ▷ Co-Simulation format – the FMU is capable of performing its own time integration
    - ▶ Easier to handle
    - ▶ More difficult to generate
- ▶ We will discuss only the Co-Simulation format

# What is a Functional Mockup Unit (FMU)?

File format (e.g. mymodel.fmu)

- ▶ Important components
  - ▷ modelDescription.xml
    - ▶ Describes the model, including
      - ▷ Input/output variables
      - ▷ Parameters available
      - ▷ Unit definitions
  - ▷ Shared libraries
  - ▷ Source files

Every FMU is distributed with its own zip file. This zip file has the following structure:

```
// Structure of zip file of an FMU
modelDescription.XML // Description of FMU (required file)
model.png // Optional image file of FMU icon
documentation // Optional directory containing the FMU documentation
  _main.html // Entry point of the documentation
  <other documentation files>
sources // Optional directory containing all C sources
  // all needed C sources and C header files to compile and link the FMU
  // with exception of: fmiTypesPlatform.h , fmiFunctionTypes.h and fmiFunctions.h
binaries // Optional directory containing the binaries
  win32 // Optional binaries for 32-bit Windows
    <modelIdentifier>.dll // DLL of the FMI implementation
    // (build with option "MT" to include run-time environment)
    <other DLLs> // The DLL can include other DLLs
    // Optional object Libraries for a particular compiler
  VisualStudio8 // Binaries for 32-bit Windows generated with
    // Microsoft Visual Studio 8 (2005)
    <modelIdentifier>.lib // Binary libraries
  gcc3.1 // Binaries for gcc 3.1.
  ...
win64 // Optional binaries for 64-bit Windows
  ...
linux32 // Optional binaries for 32-bit Linux
  <modelIdentifier>.so // Shared library of the FMI implementation
  ...
linux64 // Optional binaries for 64-bit Linux
  ...
resources // Optional resources needed by the FMU
  < data in FMU specific files which will be read during initialization;
  also more folders can be added under resources (tool/model specific).
  In order for the FMU to access these resource files, the resource directory
```

# FMU Attractions

- ▶ Open standard
- ▶ Increasing vendor adoption
- ▶ Intellectual property hiding

Tools supporting FMI	Model-Exchange		Co-Simulation	
	Export	Import	Slave	Master
Adams		Planned	Available	Available
AMESim	Available	Available	Available	Planned
ANSYS SImplorer		Planned	Planned	
ASim - AUTOSAR Simulation	Available		Available	
Atego Ace		Available		Available
@Source	Available			
Building Controls Virtual Test Bed				Available
CATIA	Available	Available	Available	Available
ControlBuild	Available	Available	Available	Available
Cosimato		Available		Available
Cybernetica CENIT		Available		Planned
Cybernetica ModelFit		Available		Available
DSHplus	Planned		Planned	
Dynola	Available +	Available	Available +	Available
EnergyPlus			Planned	Available
FMI Add-in for Excel				Available +

# History of the standard

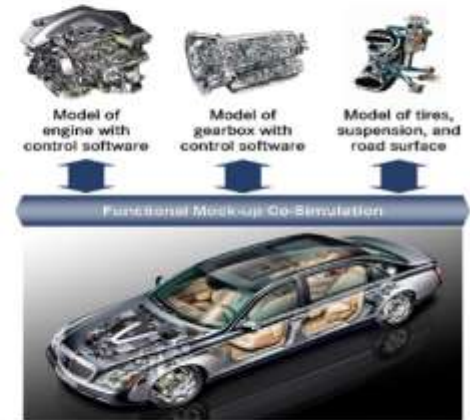
Began as “Modelisar Project”

## Project MODELISAR

Standardizing simulation of multi-domain physical models with control system models

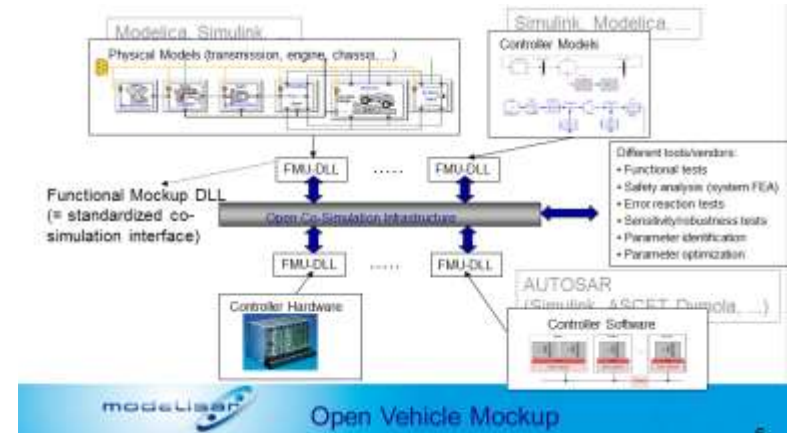
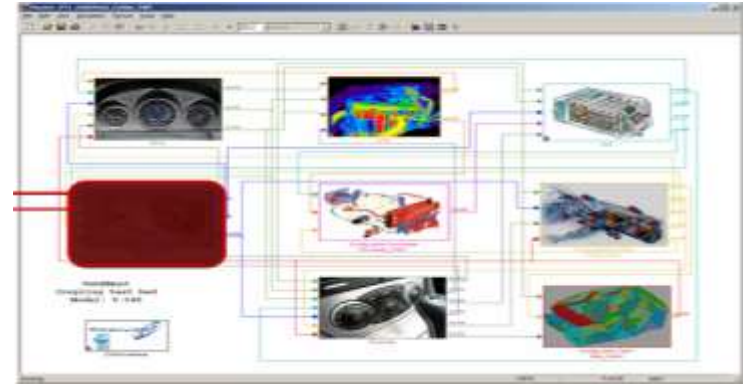
Major project goals

- Easy use of simulation technology for a virtual integration of multi-domain systems
  - Joint simulation of models being developed with different tools
  - Interface for simulation with AUTOSAR control units in the loop
- Develop a proposal for an open standard, the FMI-standard
- Prove the proposed standard by means of several real world use-case scenarios



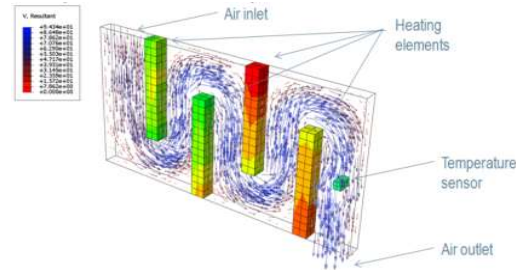
# Use of FMUs

- ▶ FMUs are designed for simulations where they are integrated into a system of components, comprised of either
  - ▷ Other FMUs or
  - ▷ Other simulation codes (e.g. Abaqus)
- ▶ FMUs are designed to be “slaves” in a simulation, with control over only their internal state.
  - ▷ They must be directed by a co-simulation “master”

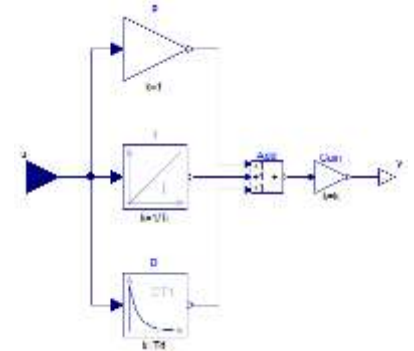


# Example

- ▶ This example simulation considers a simple electrically controlled heat exchanger illustrated below

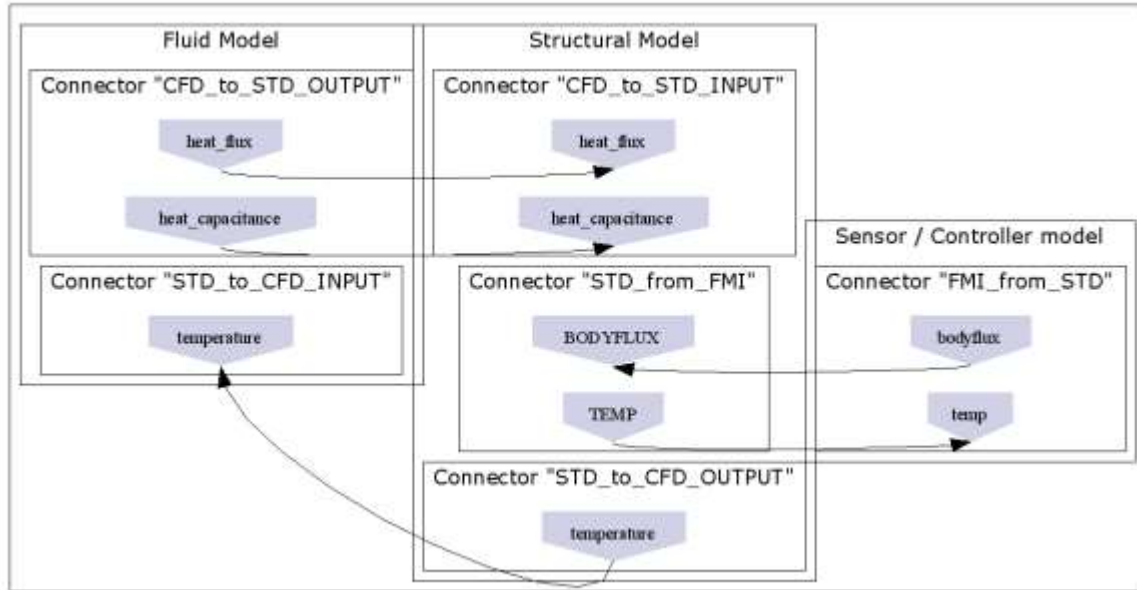


- ▶ The simulation comprises an Abaqus/Standard model of the solid heating element, an Abaqus/CFD model of the airflow through the passages, and an FMU representation of a controller. The FMU implementation models a traditional Proportional Integral Derivative (PID) controller that seeks to obtain an air outlet temperature of 50degC quickly without overshoot.



# Example

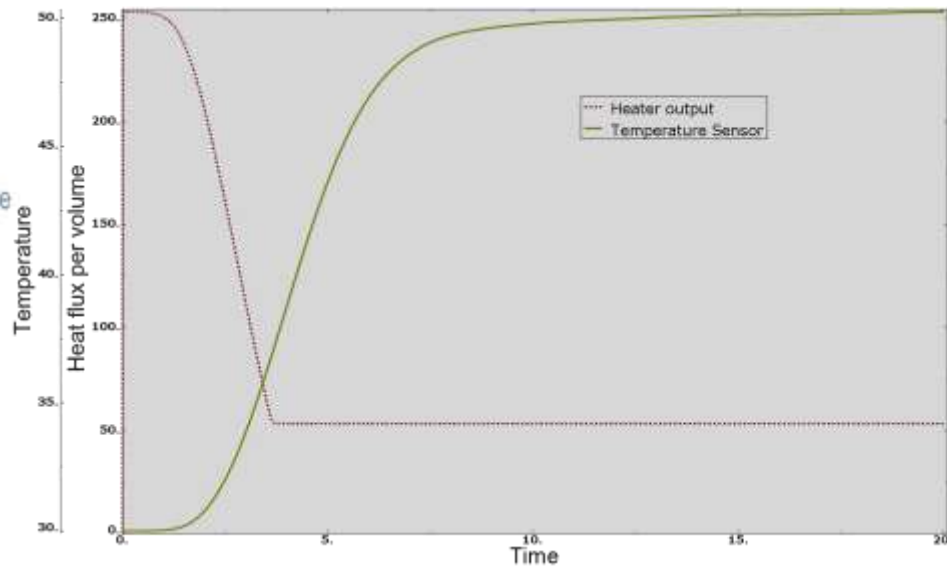
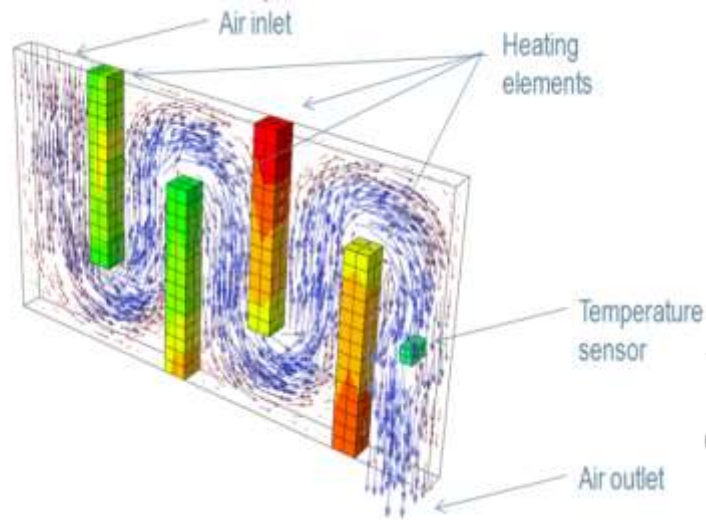
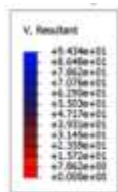
- ▶ The configuration file defines the following relationships





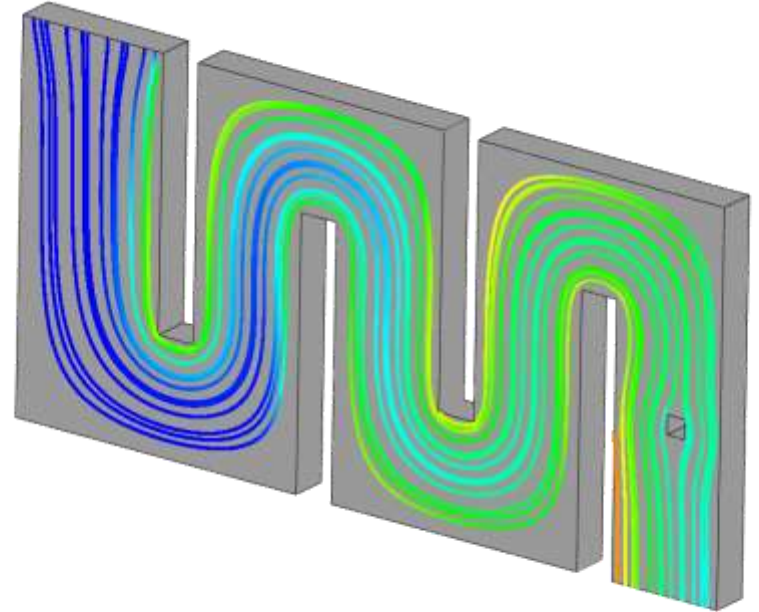
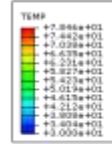
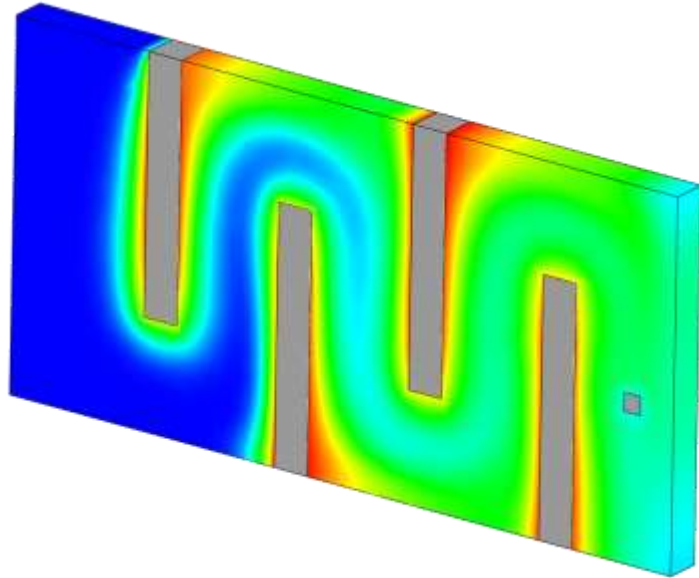
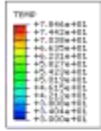
# Example

## Results



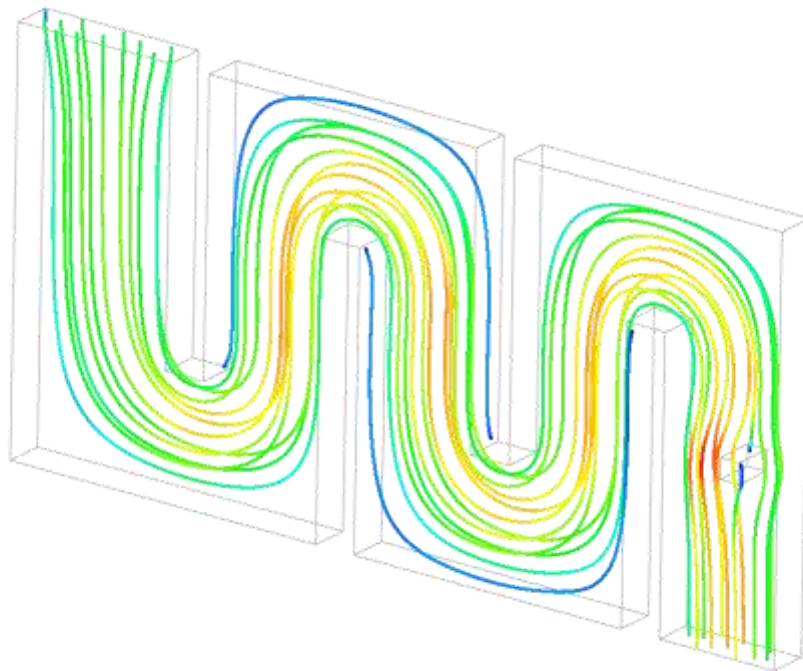
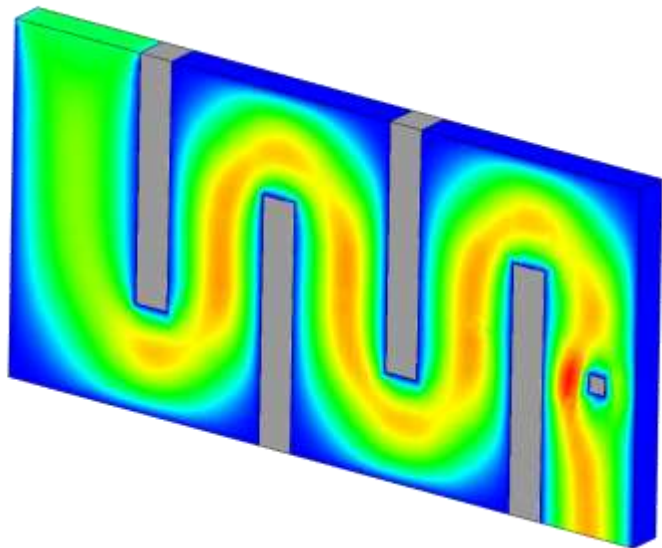
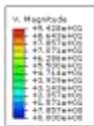
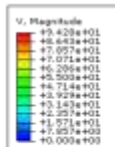
# Example

## Results: Temperature



# Example

Results: Velocity



# Example

<http://www.3ds.com/fileadmin/PRODUCTS/SIMULIA/PDF/tech-briefs/Auto-High-Fidelity-Anti-Lock-Brake-System-Simulation-10.pdf>



## Abaqus Technology Brief

TB-10-COS-1  
Revised: April 2010

### High Fidelity Anti-Lock Brake System Simulation Using Abaqus and Dymola

#### Summary

Accurate simulation of an anti-lock brake system (ABS) requires detailed modeling of separate subsystems in different physical domains. Creating refined models of the brake, wheel, and control components with a single analysis tool is difficult, if not impossible. The strategy of co-simulation can be adopted to meet this challenge; different simulation tools can be used simultaneously to create multi-disciplinary and multi-domain coupling.

In this Technology Brief, a co-simulation approach using Abaqus and Dymola is used to achieve a realistic system-level simulation of an ABS. The tire, wheel, brake caliper mechanism, and road are simulated with a detailed Abaqus finite element model while the brake system control algorithm and hydraulics are simulated with Dymola.



#### Key Features and Benefits

- Abaqus and Dymola co-simulation allows for the coupled, time domain simulation of compliant structures embedded in complex, logically controlled systems
- Accurate, detailed static and dynamic non-linear tire modeling in Abaqus

#### Background

An anti-lock brake system can be viewed as an assembly of mechanical and logical subsystems. The mechanical system consists of the tire, wheel, disc, and brake caliper hardware, while the logical system consists of the hydraulics and control electronics.

Abaqus, with its strong nonlinear continuum capabilities and versatile modeling features, has proven to be a valuable tool for tire simulations. Dymola, with its ability to efficiently model logical abstractions, is an ideal candidate for the simulation of the hydraulic and control systems.

Neither tool in isolation is an ideal choice for conducting a high-fidelity system-level simulation of the entire ABS. To this end, a co-simulation approach is demonstrated in which Abaqus and Dymola are coupled at run-time to simulate the dynamics of the system in a way that cannot be achieved with either software acting alone. The structural response of the wheel system and the logical response of the control system are exchanged between Abaqus and Dymola in a synchronized manner. The braking loads thus applied to the wheel are controlled by the ABS electronic logic based on input from the mechanical system.

consists of slowing the wheel assembly from an initial velocity of 10 m/s with a ramped brake pedal force.

The mechanical state of the brake system is used as the controlling signal; it is used by Dymola to actuate the caliper in the Abaqus model such that braking forces are applied and modulated so that the wheel will not lock, or fully slip on the road. This workflow is discussed in detail in the following sections.

#### Brake control system model in Dymola

The block diagram of the Dymola logical model is shown in Figure 1. The braking system consists of a single brake caliper cylinder connected to a master cylinder via a three

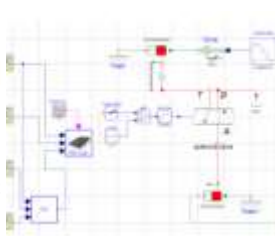


Figure 1: Block diagram of the brake system logic and hydraulics

The control mechanism is simulated by making the opening state of the three port valve depend on wheel acceleration and slip. In the control algorithm [1], the required slip is an wheel angular velocity  $\omega$ , angular acceleration rolling radius  $r$  and hub longitudinal velocity  $v_h$ . The slip signals to the controller are sampled with a period of  $\Delta t$  ms. The longitudinal slip is calculated as

$$\lambda = \frac{v_h - r\omega}{v_h}$$

ABS is triggered when the wheel deceleration falls below the prescribed threshold  $-a$ . At that moment, pressure is held constant until the slip exceeds a threshold  $\lambda_s$ ; when point pressure is dropped to a certain value.

Pressure is then held again until a positive acceleration  $A$  is reached, then the pressure is increased until the acceleration drops to  $a$ . At this stage pressure is increased only via alternative rolls and increase commands. This case allows the peak portion of the friction characteristic curve to be traversed slowly before the unstable side of the curve is reached. The cycle begins again once the acceleration threshold is crossed.

From the regular control cycle, the controller is deactivated when the longitudinal velocity is under a certain  $v$ . In addition, a limit parameter is used to reset the control algorithm if it remains in one state for an excessive

#### Tire, road and brake model in Abaqus

The Abaqus model, shown in Figure 2, includes the tire, wheel, brake caliper and rotor subassembly, and the road. The tire is first pressurized and placed in contact with the road under the vehicle weight acting on the wheel. A steady state transport analysis is then performed in Abaqus/Standard to compute the state of stress and deformation in the tire corresponding to a given forward velocity with no braking. Optionally, a cornering radius can be specified as well.

The tire in its free-rolling condition is then imported into Abaqus/Explicit for the braking co-simulation with Dymola. As Abaqus/Explicit computes the state of stress and deformation in the rolling tire, the wheel angular velocity and acceleration are communicated at frequent intervals to Dymola via the Abaqus sensors. The required braking pressure is computed by Dymola and communicated back to Abaqus/Explicit to be applied to the brake caliper cylinder. The brake pads are pressed against the brake rotor to produce a braking torque that decelerates the wheel assembly.

#### Co-simulation scheme

The following non-iterative co-simulation scheme is used:

- At each time increment of the Abaqus simulation, sensor information is computed and communicated to Dymola via a socket-based interface.
- Dymola reads the Abaqus sensor information as inputs to the control logic and integrates in time with an increment size equal to that used by Abaqus/Explicit. The actuating signal computed in Dymola is communicated back to Abaqus, which applies this newly computed load in the next increment.
- The process is repeated until the simulation time is completed.

#### Co-simulation results

Figure 3 shows the hub longitudinal and circumferential velocities during the co-simulation. A difference between the two quantities represents a slipping condition. The slip



threshold is crossed.

In this regular control cycle, the controller is designed so the longitudinal velocity is under a certain limit, a limit parameter is used to reset the algorithm if it remains in one state for an excessive

time. An alternative is to use a model-based algorithm, which is popular in production. An alternative is to use a model-based algorithm of which can be found in [2].



Figure 2: Abaqus tire, road and brake model

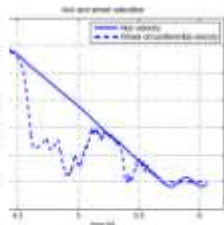


Figure 3: Hub longitudinal and circumferential velocities during co-simulation

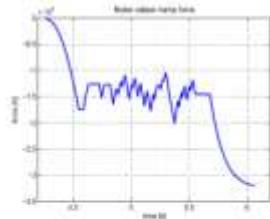


Figure 4: Brake caliper clamping force during co-simulation

threshold is crossed. In this regular control cycle, the controller is able to prevent wheel lock-up during the co-simulation. The rapid force build-up and release are clearly visible.

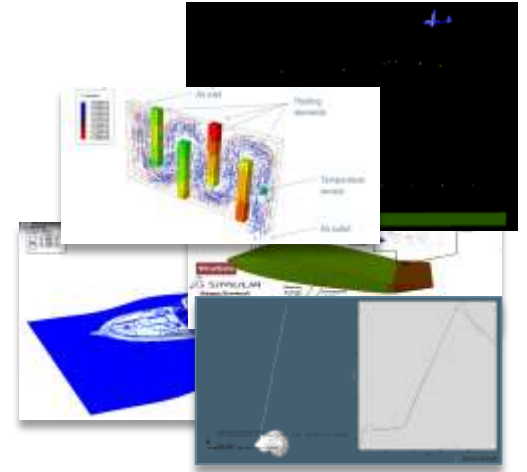
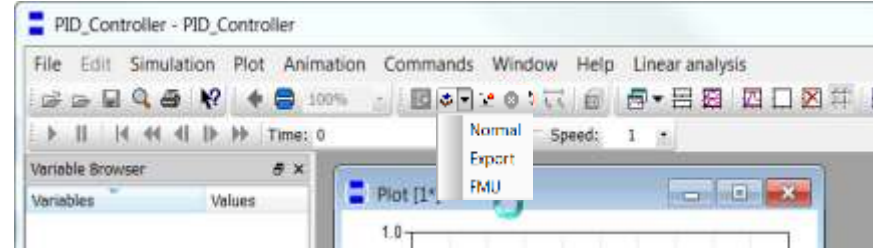
High-fidelity tire model in Abaqus is combined with a hydraulic braking control system model in Dymola, highlighting how different simulation packages can be integrated to perform realistic system-level simulations. The existing advanced modeling capabilities in Abaqus can thus be applied to a larger class of real-world operating conditions. The co-simulation approach can be extended to other



IF WE ask the right questions we can change the world.

# Using FMUs in your work

- ▶ Create them with Dymola
  - ▷ Note that the FMU interface will replace the current Dymola DLL interface
  - ▷ All Dymola DLL QC tests have equivalent FMU based tests
- ▶ Write your own
  - ▷ An SDK is provided by <http://www.qtronic.de/en/fmusdk.html>
    - ▶ This kit is used by Jeff, Jaesu, David Fox for creating FMUs from c-code



# Our Values

IF WE  SHOW  
THE **DREAM**  
IS POSSIBLE

> WE CAN **INSPIRE** PEOPLE <  
TO CREATE IT

IF WE  HAVE  
THE **PASSION**  
TO LEARN

> WE CAN **EXPAND** CREATIVITY <  
TO NAVIGATE THE FUTURE

IF WE  **CHALLENGE**  
THE STATUS QUO

> WE CAN **IMAGINE** NEW HORIZONS <  
TO IMPROVE THE WORLD

IF WE  **BRING OUR**  
**COMMUNITY**  
TOGETHER

> WE CAN **BUILD** HARMONY <  
TO ACHIEVE OUR GOALS

