# Implementation of Comsol in Simulink S-Functions, Revisited

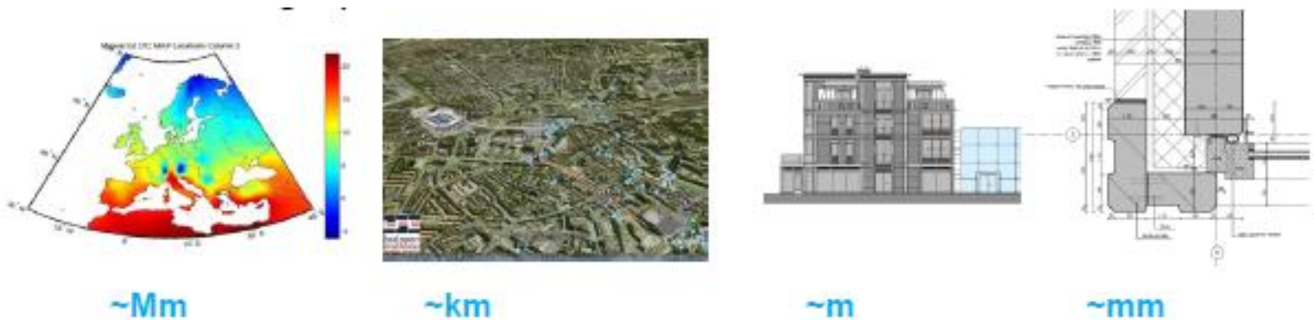
Jos van Schijndel

COMSOL CONFERENCE 2014 CAMBRIDGE

Technische Universiteit Eindhoven University of Technology

Where innovation starts

# Scale levels Building physics

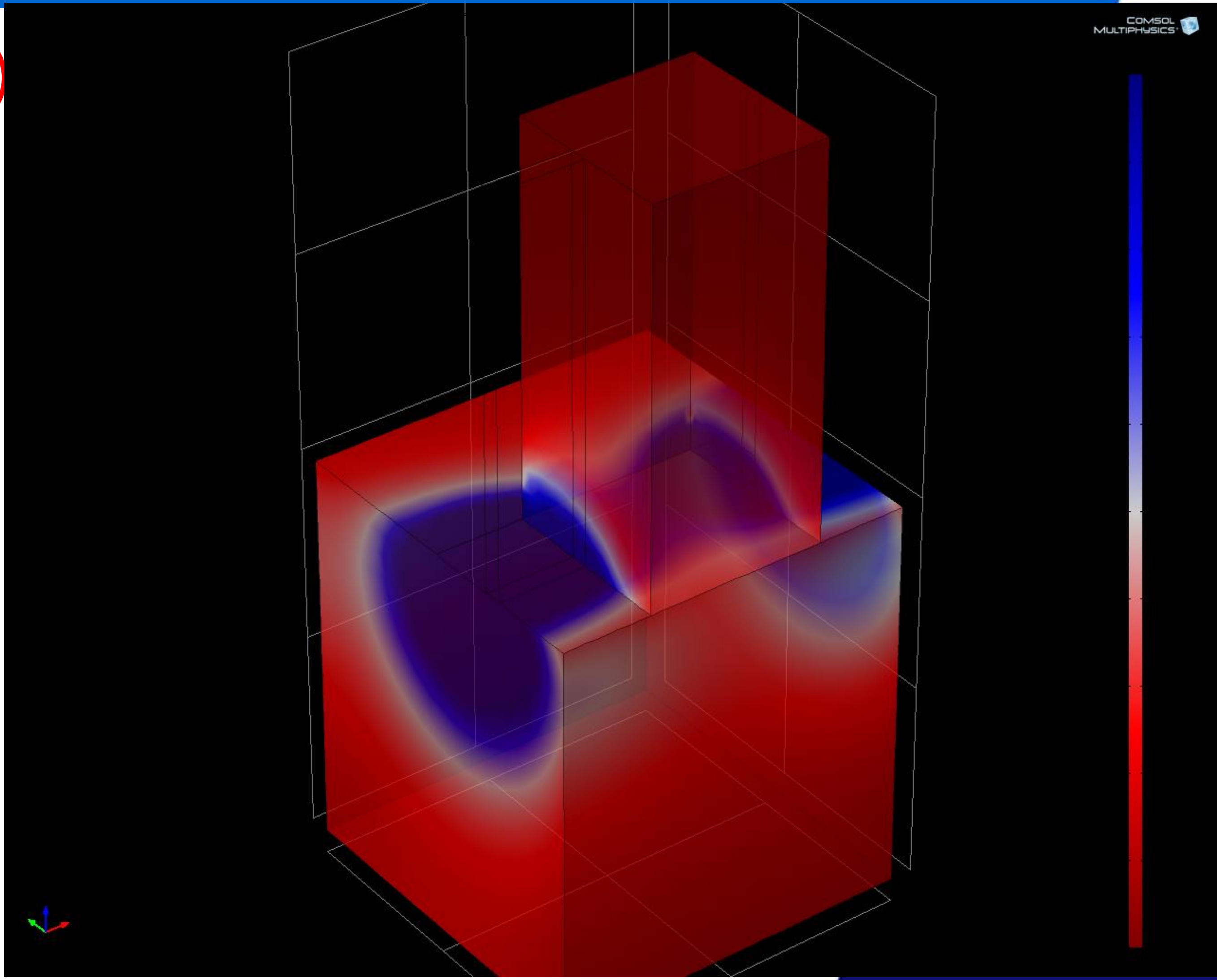



Scale levels, from left to right: EU; Urban area; Building; Material;

- [mm] Material Physics
- [m] Building Physics
- [km] Urban Physics
- [Mm] … Physics

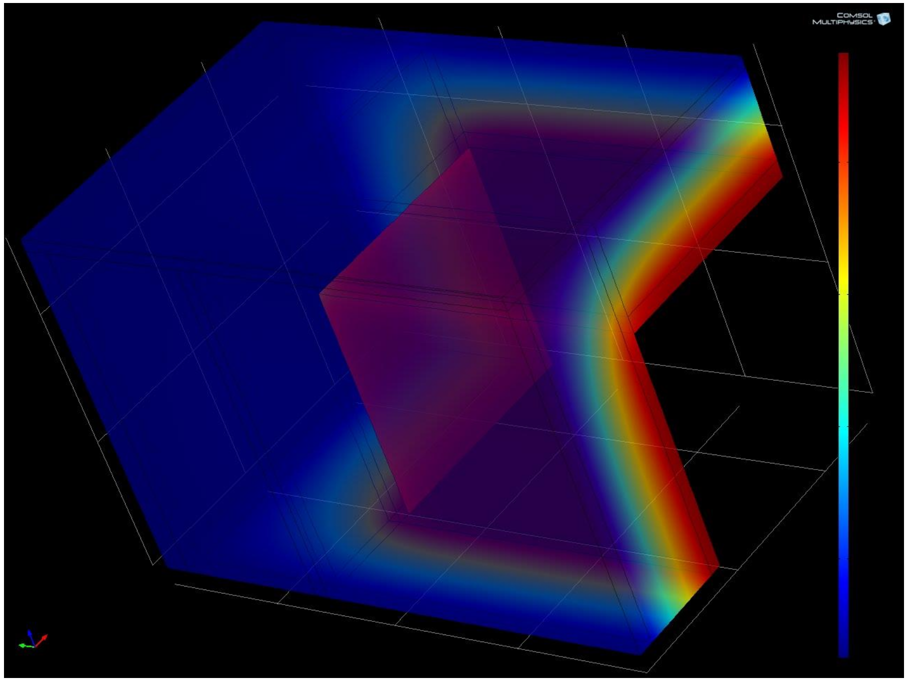
# Scale level [mm]  Material Physics
# Moisture induced damages

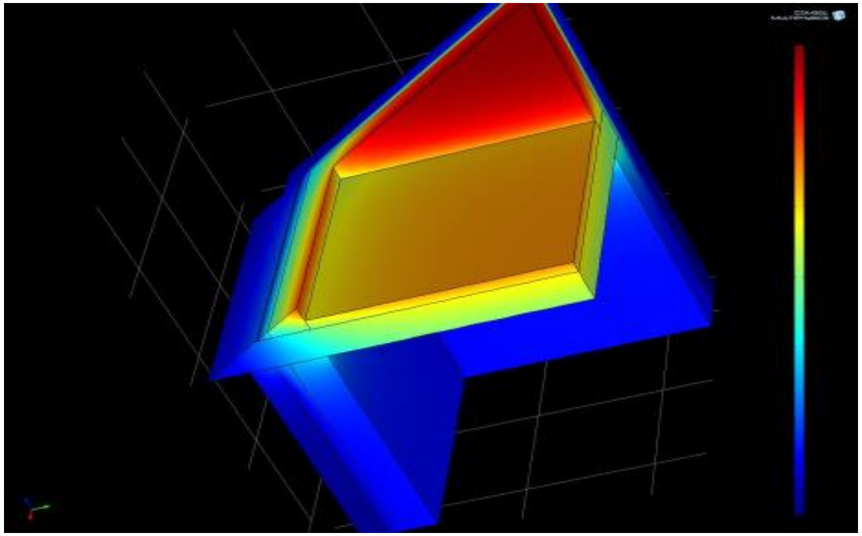
Scale levels, from left to right: EU; Urban area; Building; Material;



# Scale level [cm]  Building systems Physics
# Thermal bridges

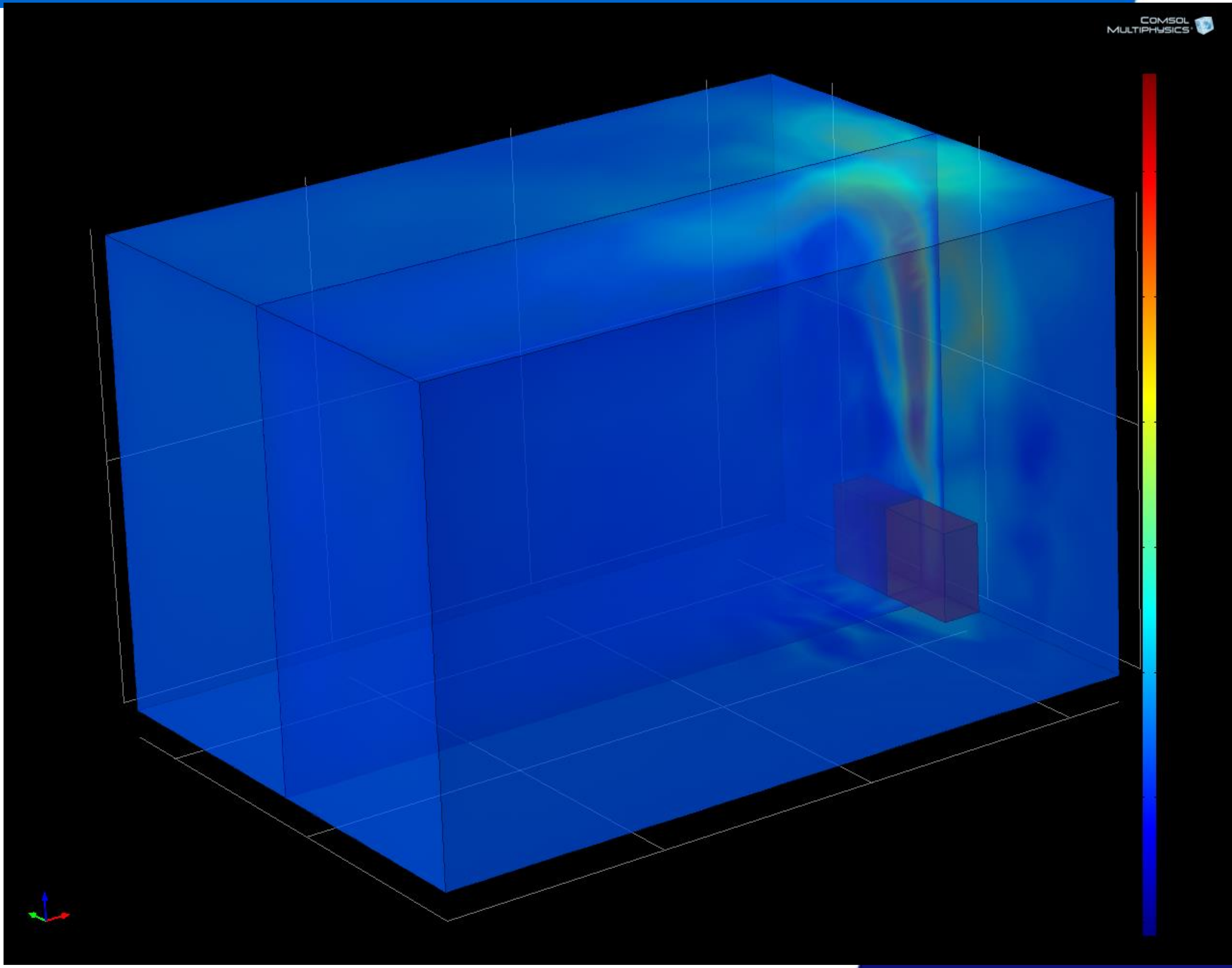
Scale levels, from left to right: EU; Urban area; Building; Material;





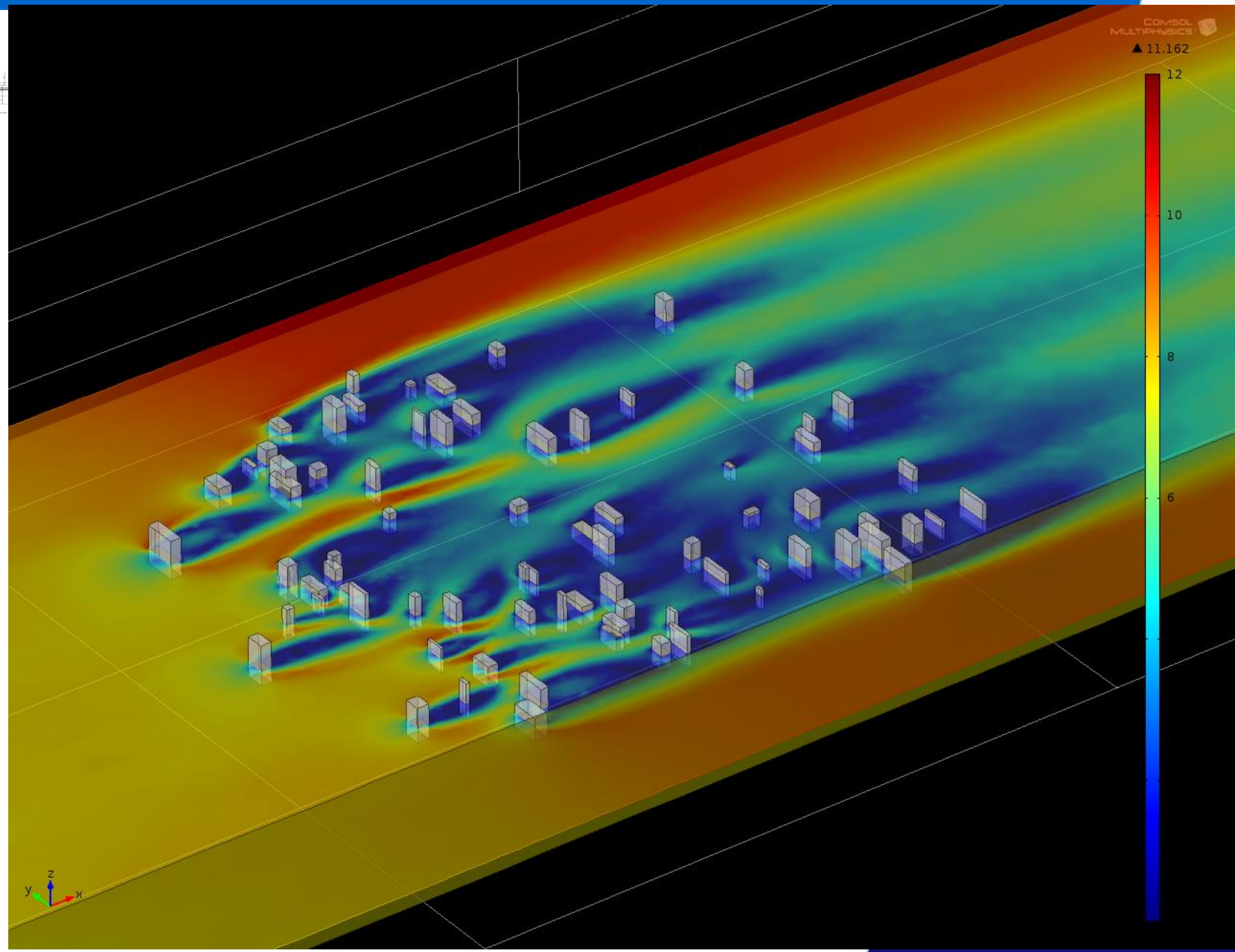
# Scale level [m] Building Physics
# Indoor climate performance & design


Scale levels, from left to right: EU; Urban area; Building; Material;

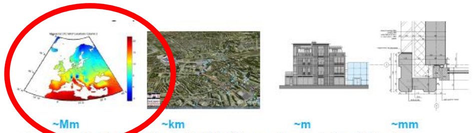


# Scale level [km]  Urban physics
# Urban climate performance

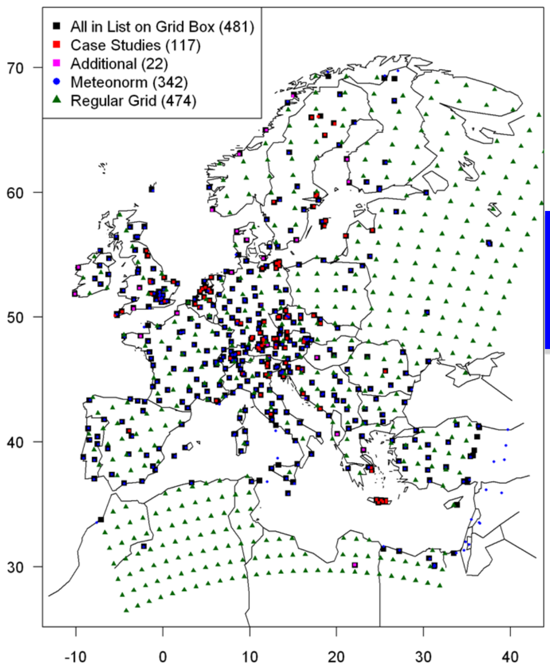

Scale levels, from left to right: EU; Urban area; Building; Material;



# Scale level [Mm] EU physics
# EU climate scale performance & design

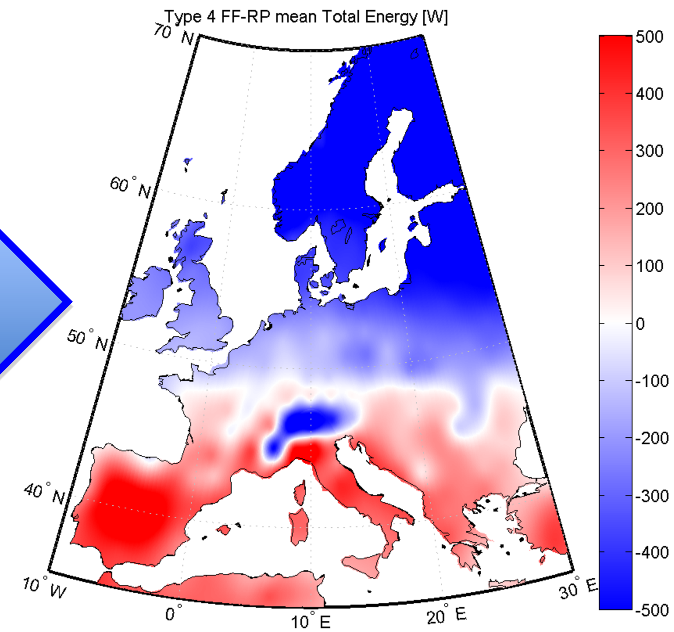



Scale levels, from left to right: EU; Urban area; Building; Material;




**Building simulation (HAMBase)**

**Museum models (Classification)**



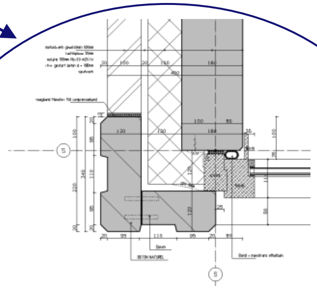

Technische Universiteit **Eindhoven** University of Technology

# Computational Building Physics
# PSE: Problem solving environment

Multi Buildings
‘HAMBase’
MatLab

Multi Details
PDE
Comsol

Multi Systems & Control
ODE
SimuLink



Technische Universiteit **Eindhoven** University of Technology

# Implementation of Comsol in Simulink S-Functions, Revisited

## CONTENTS

- Why?
  - SimuLink has powerful unique capabilities
  - Alternative for Standard export from Comsol to SimuLink
- How?
  - Using S-Functions SimuLink and
  - Comsol-Matlab code
- Examples
- Conclusions

## CONTENTS

- Why?
  - SimuLink has powerful unique capabilities
  - Alternative for Standard export from Comsol to SimuLink
- How?
  - Using S-Functions SimuLink and
  - Comsol-Matlab code
- Examples
- Conclusions

# SimuLink

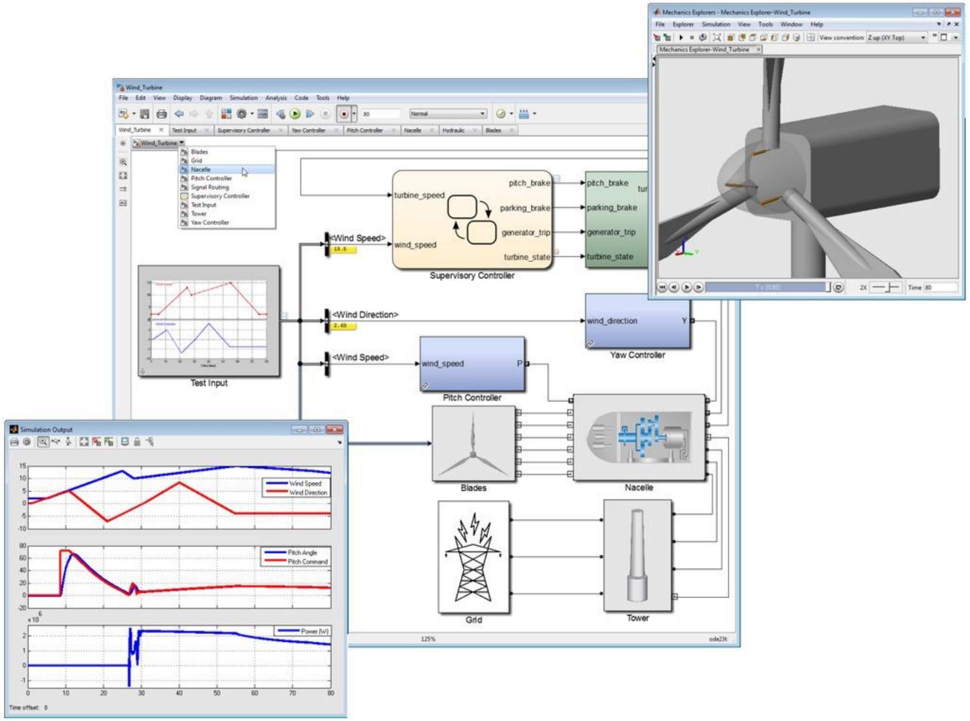
Simulink is widely used in

control theory and

digital signal processing

for multidomain simulation

and Model-Based Design.[2][3]



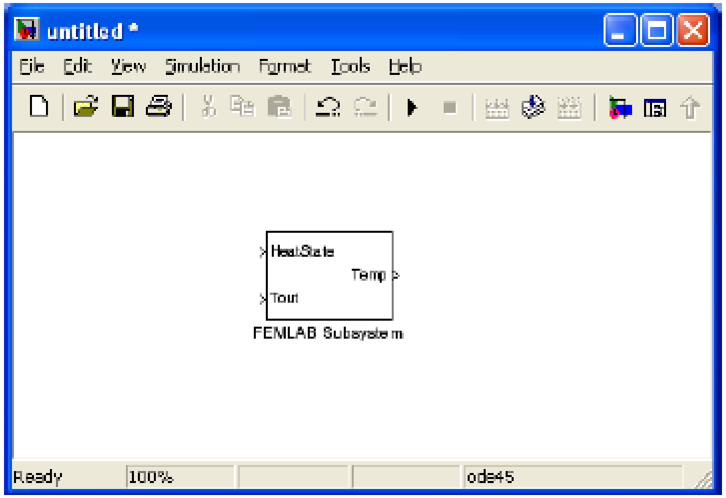


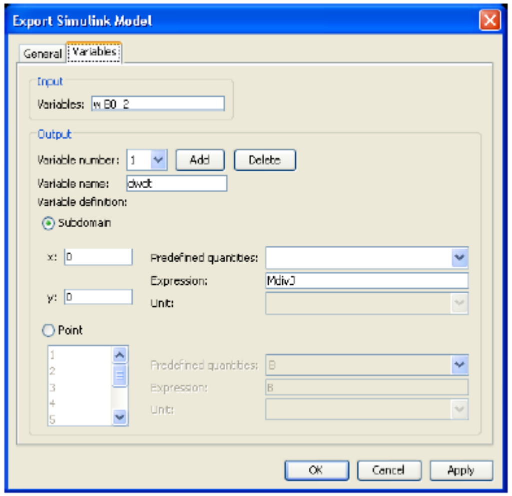
2. ^ "The Successful development process with MATLAB Simulink in the framework of ESA's ATV project" (PDF). Vega Group PLC. Retrieved 2011-11-01.
3. ^ Model Based Design Accelerates the Development of Mechanical Locomotive Controls, SAE 2010 Commercial Vehicle Engineering Congress, October 2010, Chicago, IL, USA, Session: Model Based Design & Embedded Software Development (Part 1 of 2), Paper 2010-01-1999

# CONTENTS

- Why?
  - SimuLink has powerful unique capabilities
  - Alternative for Standard export from Comsol to SimuLink
- How?
  - Using S-Functions SimuLink and
  - Comsol-Matlab code
- Examples
- Conclusions

# Comsol standard export to SimuLink




*A COMSOL Multiphysics Subsystem block in Simulink with two inputs and one output.*

Because COMSOL Multiphysics models usually are stiff, we recommend using an implicit stiff ODE solver like `ode15s` in Simulink's **Simulation Parameters** dialog box.

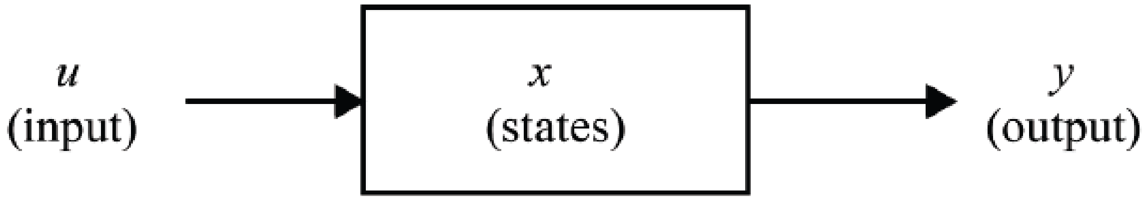
**However, SimuLink solvers cannot handle complex (non linear) Comsol models**

## CONTENTS

- Why?
  - SimuLink has powerful unique capabilities
  - Alternative for Standard export from Comsol to SimuLink
- How?
  - Using S-Functions SimuLink and
  - Comsol-Matlab code
- Examples
- Conclusions

Technische Universiteit Eindhoven University of Technology

# S-Function in SimuLink




$$y = f_0(t, x, u) \qquad \text{(output)}$$

$$\dot{x}_c = f_d(t, x, u) \qquad \text{(derivative)}$$

$$x_{d_{k+1}} = f_u(t, x, u) \qquad \text{(update)}$$

where $x = x_c + x_d$

# Step 1: Dummy S-Function in SimuLink

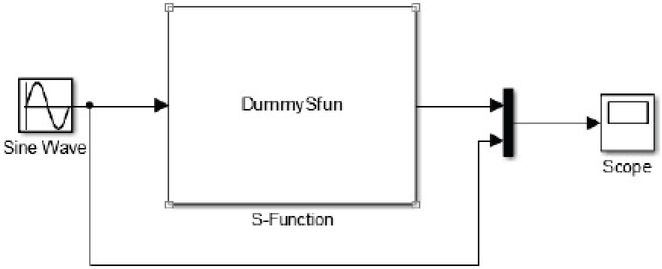
**Appendix 1**

```
function [sys,x0,str,ts] = DummySfun(t,x,u,flag)
% Diskrete S-function Working almost empty for Comsol
comsolmodel=[];
%Init my variables
tstap=eval(get_param(gcs,'fixedstep'));  % time step in [s]
u0=0;      %
switch flag,
  case 0,
    [sys,x0,str,ts] = mdlInitializeSizes(comsolmodel,tstap,u0);
  case 2,
    sys = mdlUpdate(t,x,u,comsolmodel,tstap);
  case 3,
    sys = mdlOutputs(t,x,u,comsolmodel,tstap);
  case 9,
   otherwise
    error(['unhandled flag = ',num2str(flag)]);
end

function [sys,x0,str,ts] = mdlInitializeSizes(comsolmodel,tstap,u0)
nT=1; %Dummy replace with number of nodes if neccessary
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = nT;
sizes.NumOutputs     = 1;
sizes.NumInputs      = 1;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = u0*zeros(nT,1);
str = [];
ts  = [tstap 0];

function sys = mdlUpdate(t,x,u,comsolmodel,tstap)
disp(['u =' num2str(u(1)) ])
disp(['calc ' num2str(t) ' to ' num2str(t+tstap) ])
sys=1; %Dummy replace with nT number of nodes if neccessary

function sys = mdlOutputs(t,x,u,comsolmodel,tstap)
sys=1; %Dummy replace with nT number of nodes if neccessary
```



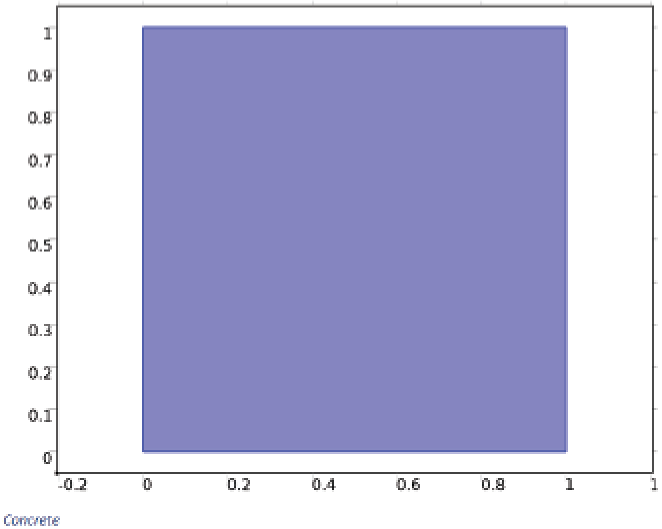
## CONTENTS

- Why?
  - SimuLink has powerful unique capabilities
  - Alternative for Standard export from Comsol to SimuLink
- How?
  - Using S-Functions SimuLink and
  - Comsol-Matlab code
- Examples
- Conclusions

# Step 2: Simple Heating model in Comsol exported to Matlab

**Appendix 3**

```
import com.comsol.model.*
import com.comsol.model.util.*
Cmodel = ModelUtil.create('Model');
Cmodel.modelPath('D:\04_Comsol44\SimuLink');
Cmodel.modelNode.create('comp1');
Cmodel.geom.create('geom1', 2);
Cmodel.mesh.create('mesh1', 'geom1');
Cmodel.physics.create('ht', 'HeatTransfer', 'geom1');
Cmodel.geom('geom1').run;
Cmodel.study.create('std1');
Cmodel.study('std1').feature.create('time', 'Transient');
Cmodel.study('std1').feature('time').activate('ht', true);
Cmodel.geom('geom1').feature.create('sq1', 'Square');
Cmodel.geom('geom1').run;
Cmodel.material.create('mat1');
Cmodel.material('mat1').name('Concrete');
Cmodel.material('mat1').set('family', 'concrete');
Cmodel.material('mat1').propertyGroup('def').set('density', '2300[kg/m^3]');
Cmodel.material('mat1').propertyGroup('def').set('thermalconductivity', '1.8[W/(m*K)]');
Cmodel.material('mat1').propertyGroup('def').set('heatcapacity', '880[J/(kg*K)]');
Cmodel.material('mat1').selection.geom('geom1', 2);
Cmodel.material('mat1').selection.set([1]);
Cmodel.material('mat1').set('family', 'concrete');
Cmodel.name('Simple2D_fase2.mph');
Cmodel.physics('ht').feature.create('hs1', 'HeatSource', 2);
Cmodel.physics('ht').feature('hs1').selection.set([1]);
Cmodel.physics('ht').feature('hs1').set('Q', 1, '1000');
Cmodel.study('std1').feature('time').set('tlist', 'range(0,3600,24*3600)');
Cmodel.sol.create('sol1');
Cmodel.sol('sol1').study('std1');
Cmodel.sol('sol1').feature.create('st1', 'StudyStep');
Cmodel.sol('sol1').feature('st1').set('study', 'std1');
Cmodel.sol('sol1').feature('st1').set('studystep', 'time');
Cmodel.sol('sol1').feature.create('v1', 'Variables');
Cmodel.sol('sol1').feature('v1').set('control', 'time');
Cmodel.shape('shape1').feature('shfun1');
Cmodel.sol('sol1').feature.create('t1', 'Time');
Cmodel.sol('sol1').feature('t1').set('tlist', 'range(0,3600,24*3600)');
Cmodel.sol('sol1').feature('t1').set('plot', 'off');
Cmodel.sol('sol1').feature('t1').set('plotfreq', 'tout');
Cmodel.sol('sol1').feature('t1').set('probesel', 'all');
Cmodel.sol('sol1').feature('t1').set('probes', {});
Cmodel.sol('sol1').feature('t1').set('probefreq', 'tsteps');
Cmodel.sol('sol1').feature('t1').set('atolglobalmethod', 'scaled');
Cmodel.sol('sol1').feature('t1').set('atolglobal', 0.0010);
Cmodel.sol('sol1').feature('t1').set('estrat', 'exclude');
Cmodel.sol('sol1').feature('t1').set('maxorder', 2);
Cmodel.sol('sol1').feature('t1').set('control', 'time');
Cmodel.sol('sol1').feature('t1').feature.create('fc1', 'FullyCoupled');
Cmodel.sol('sol1').feature('t1').feature('fc1').set('jtech', 'once');
Cmodel.sol('sol1').feature('t1').feature('fc1').set('damp', 0.9);
Cmodel.sol('sol1').feature('t1').feature('fc1').set('maxiter', 5);
Cmodel.sol('sol1').feature('t1').feature.create('d1', 'Direct');
Cmodel.sol('sol1').feature('t1').feature('d1').set('linsolver', 'pardiso');
Cmodel.sol('sol1').feature('t1').feature('fc1').set('linsolver', 'd1');
Cmodel.sol('sol1').feature('t1').feature('fc1').set('jtech', 'once');
Cmodel.sol('sol1').feature('t1').feature('fc1').set('damp', 0.9);
Cmodel.sol('sol1').feature('t1').feature('fc1').set('maxiter', 5);
Cmodel.sol('sol1').feature('t1').feature.remove('fcDef');
Cmodel.sol('sol1').attach('std1');
```



# Step 3: Additional manual setting

**Appendix 2**

```
Cmodel.physics('ht').feature('init1').set('T', 1, '293.15[K]');
Tmid_all5(1)=293.15;
for i=1:24
Cmodel.sol('sol1').feature('t1').set('tlist', 'range(0,3600)');
Cmodel.sol('sol1').runAll;
u=mphinterp(Cmodel,'T','coord',[0.5;0.5],'Solnum','end')
Tmid_all5(i+1)= u;
ustr=[num2str(u) '[K]'];
Cmodel.physics('ht').feature('init1').set('T', 1, ustr);
end
```

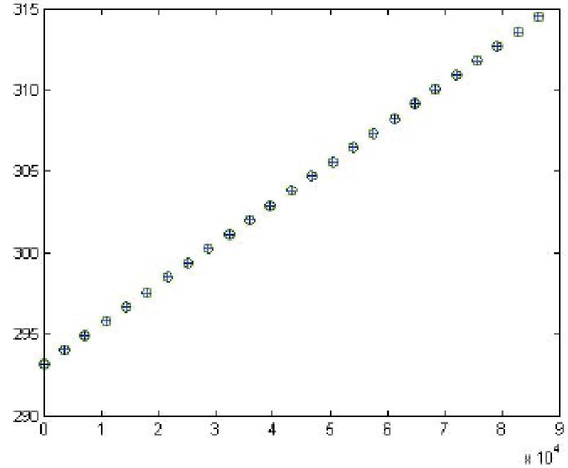
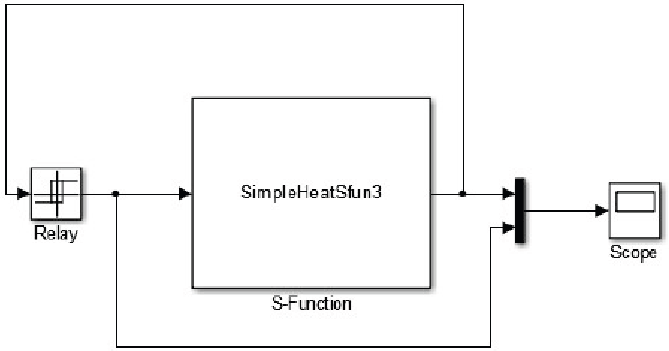



**Figure 5.** Comparison with a simulation of the whole period at once (+) and manually stepping (o)

# Step 4: Final implementation



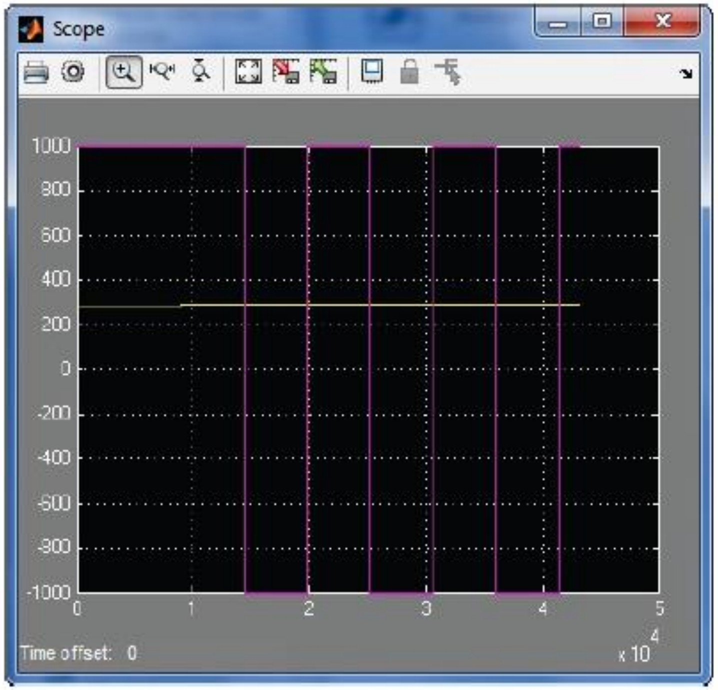



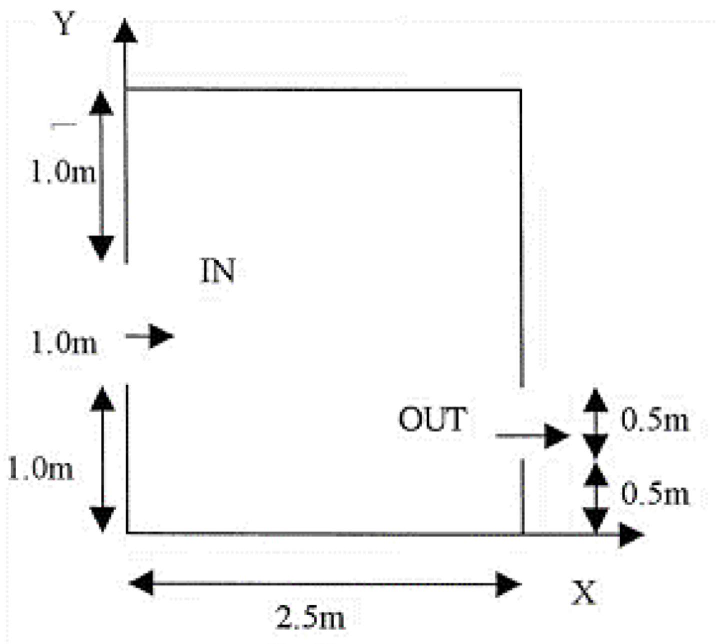

**Figure 7.** The simulated result

Technische Universiteit **Eindhoven** University of Technology

# CONTENTS

- Why?
  - SimuLink has powerful unique capabilities
  - Alternative for Standard export from Comsol to SimuLink
- How?
  - Using S-Functions SimuLink and
  - Comsol-Matlab code
- Examples
- Conclusions

# Numerical case study




$$\frac{\partial u}{\partial t} = -\frac{\partial (uu)}{\partial x} - \frac{\partial (vu)}{\partial y} - \frac{\partial p}{\partial x} + \frac{1}{\mathrm{Re}} \nabla^2 u$$

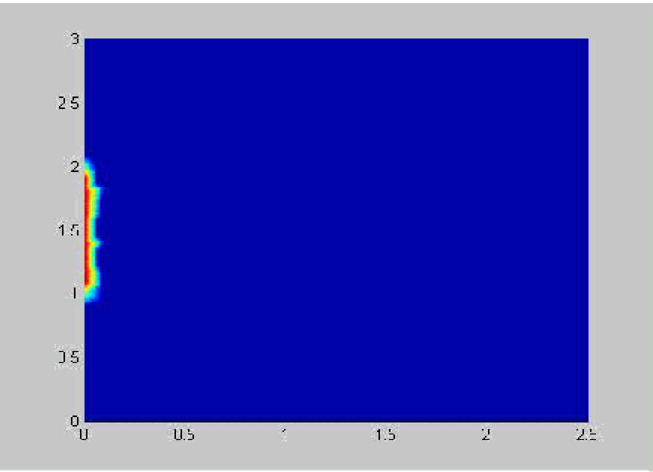
$$\frac{\partial v}{\partial t} = -\frac{\partial (uv)}{\partial x} - \frac{\partial (vv)}{\partial y} - \frac{\partial p}{\partial y} + \frac{1}{\mathrm{Re}} \nabla^2 v + \frac{Gr}{\mathrm{Re}^2} T$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

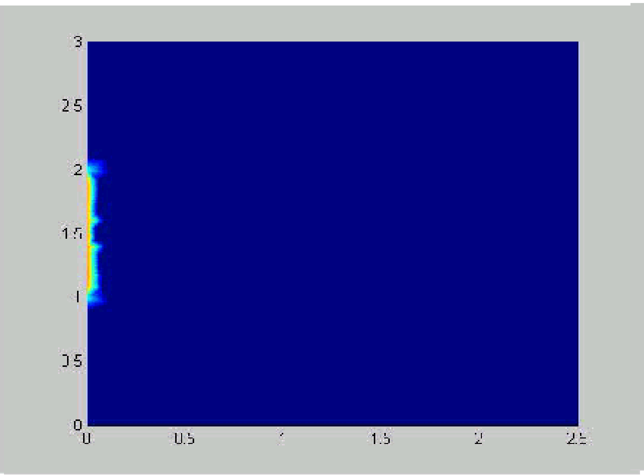
$$\frac{\partial T}{\partial t} = -\frac{\partial (uT)}{\partial x} - \frac{\partial (vT)}{\partial y} + \frac{1}{\mathrm{Re}\,\mathrm{Pr}} \nabla^2 T$$

Sinha et al. 2000 Energy and Buildings 32, pp121-129

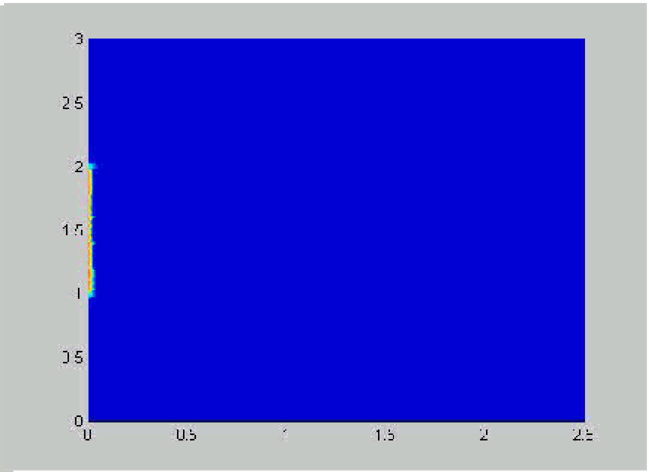
# Simulation using Comsol




Re = 50; Gr =0

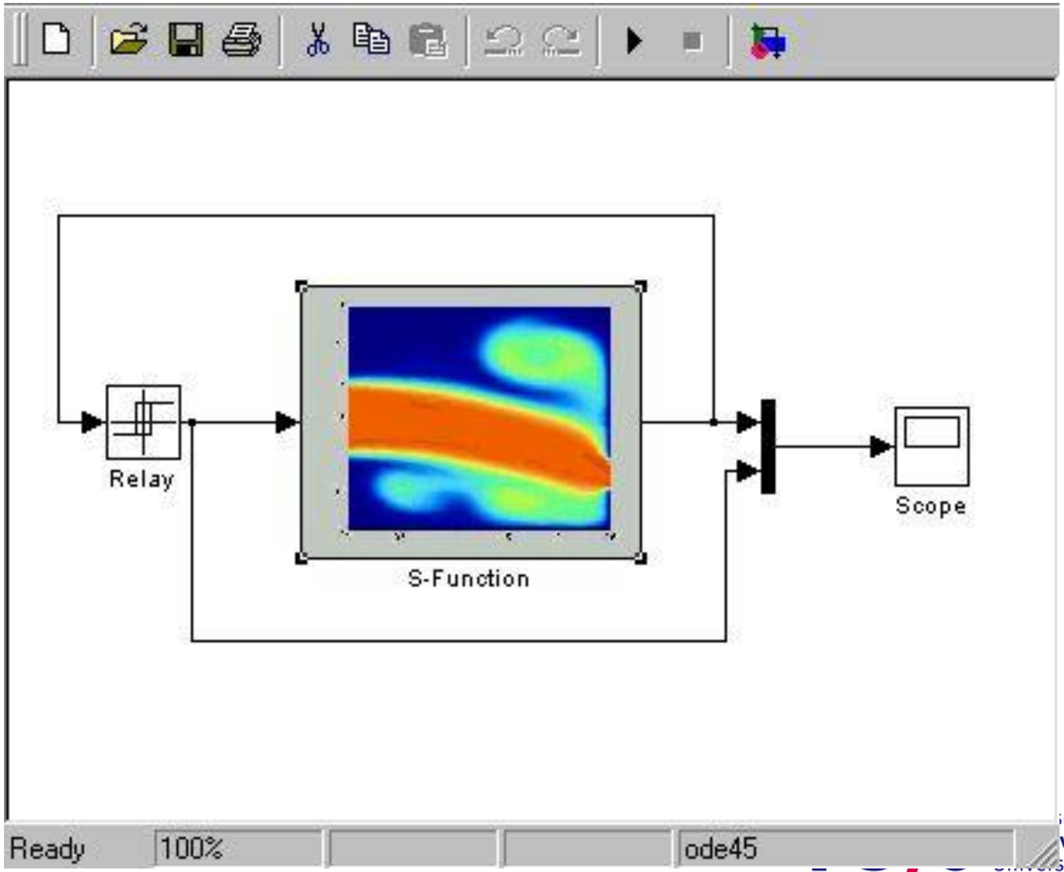



Re = 1000; Gr =0



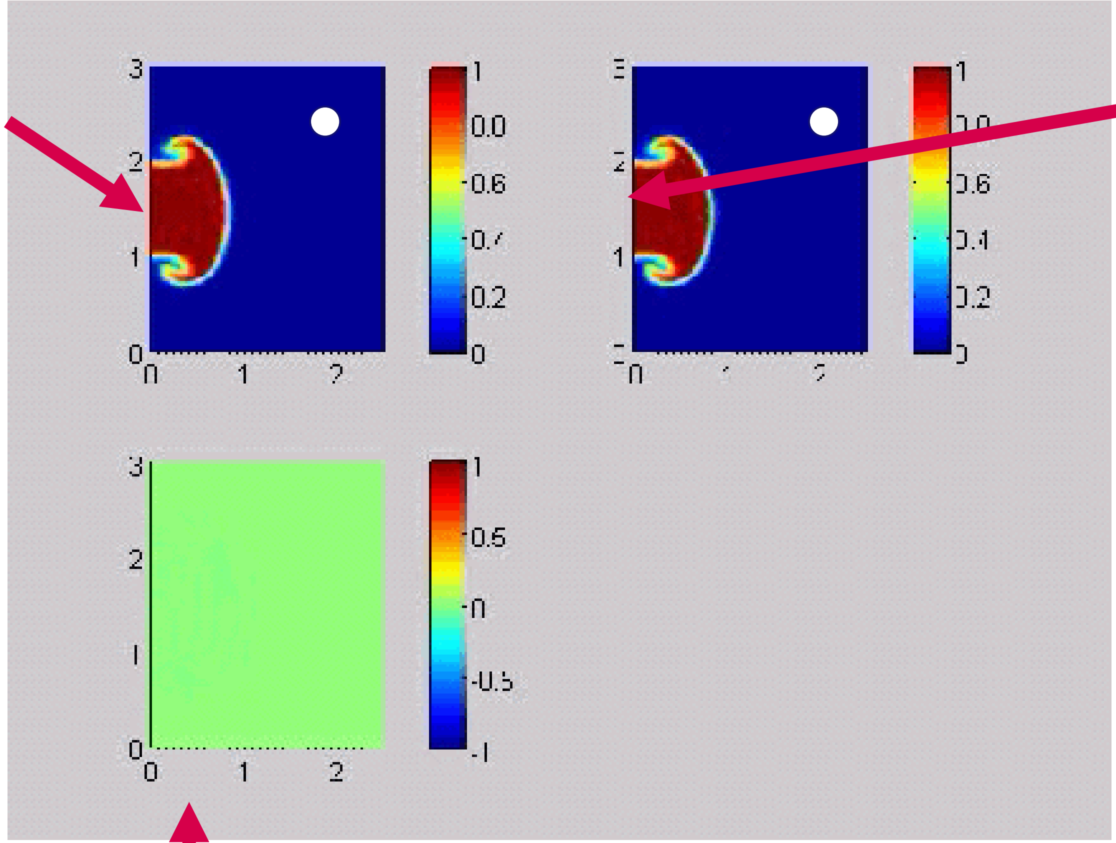

Re = 1000; Gr =~$10^7$

# Implementation in S-Function, target



# Switching sensitivity without buoyancy

Switching:
<0.30 hot air
>0.50 cold air




Switching:
<0.32 hot air
>0.48 cold air

Difference between top figures

## CONTENTS

- Why?
  - SimuLink has powerful unique capabilities
  - Alternative for Standard export from Comsol to SimuLink
- How?
  - Using S-Functions SimuLink and
  - Comsol-Matlab code
- Examples
- Conclusions

# Conclusions

- It is concluded that Comsol models can also be exported to SimuLink by writing an appropriate S-Function.

- The advantage of this approach is that the special solvers of Comsol can be used in the SimuLink environment. This can lead to significant improvement of the simulation duration time.

- **Thank you for your attention**

- **Questions?**