Modelica-based simulation of building and district energy systems

SESSION 3: Modelica overview

- Text editor
- Modelica code, variables and parameters
- Exercise 2: “HelloWorld” in Modelica
- Exercise 3: Getting physical: Cooling an object with air

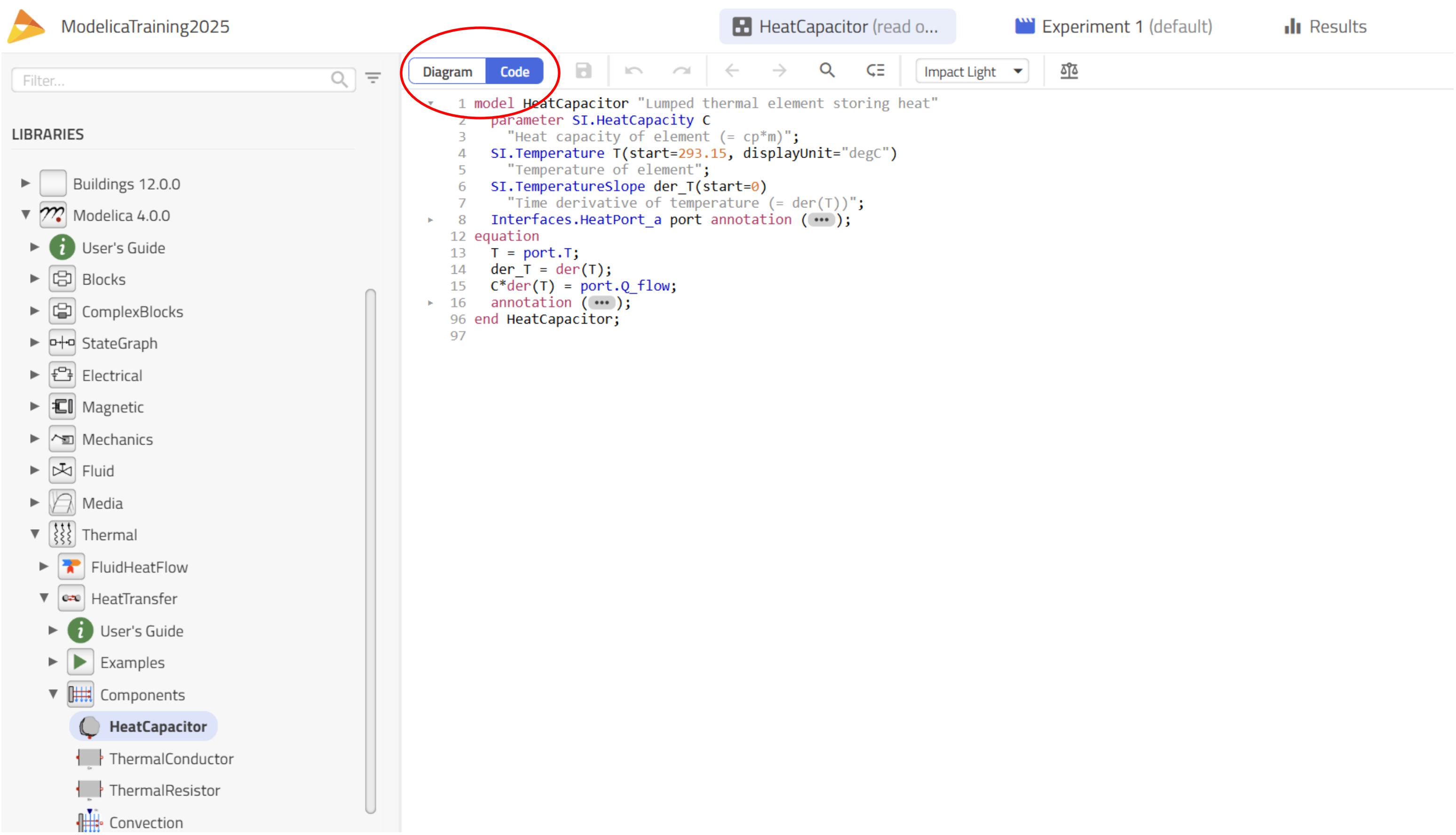
Alessandro Maccarini

SESSION 3: Modelica overview

- **Text editor**
- Modelica code, variables and parameters
- Exercise 2: “HelloWorld” in Modelica
- Exercise 3: Getting physical: Cooling an object with air

Modelon Impact Code Editor

- The *text view* of a class can be access by clicking the Code button
- Make sure the class you want to view is selected in the package browser:
  - **Double-click** on it



Dymola text editor

- In the default view annotations are not displayed.
- Show them by either:
  - Clicking on the small arrow
  - Or right click the three dots

```
model HeatCapacitor "Lumped thermal element storing heat"
  parameter SI.HeatCapacity C
    "Heat capacity of element (= cp*m)";
  SI.Temperature T(start=293.15, displayUnit="degC")
    "Temperature of element";
  SI.TemperatureSlope der_T(start=0)
    "Time derivative of temperature (= der(T))";
  Interfaces.HeatPort_a port annotation (•••);
equation
  T = port.T;
  der_T = der(T);
  C*der(T) = port.Q_flow;
  annotation (•••);
end HeatCapacitor;
```

SESSION 3: Modelica overview

- Text editor
- **Modelica code, variables and parameters**
- Exercise 2: “HelloWorld” in Modelica
- Exercise 3: Getting physical: Cooling an object with air

Modelica code

A Modelica "Hello World" model

$$\dot{x} = 1 - x$$

```
model HelloWorld
  Real x;
equation
  der(x) = 1-x;
end HelloWorld;
```

The code starts with the keyword ***model***, which is used to indicate the start of the model definition. The ***model*** keyword is followed by the model name (in this example we chose *HelloWorld*)

Declaration of variables. The variable x is declared as ***Real*** (there are other types such as parameters, constant etc..)

Equations. We can use the operator ***der().*** We don't need to write a "program" to solve the differential equation!

The code ends with the keyword ***end*** and the name of the model

Modelica code

We can make our model a bit more readable by adding description

```
model HelloWorld "A simple first differential equation"
  Real x "State variable";
equation
  der(x) = 1-x "My equation"
end HelloWorld;
```

We can use quoted text

The **state** of a dynamical system is a collection of variables (state variables) that permits prediction of the future development of the system

Modelica code

Variable declaration

```
prefix type name “description”;
```

- prefix
    - No prefix — variable — can change with time
    - `parameter` — parameter — constant with time, may be modified
    - `constant` — parameter — constant with time, may not be modified

- type
    - `Real` — floating point variable e.g. 1.6, -2.3e-5
    - `Integer` — integer variable e.g. 1, 4, 133
    - `Boolean` — boolean variable e.g. false, true
    - `String` — string e.g. “from file”

```
model SimpleEquation "A simple first differential equation"
  Real x "State variable";
  parameter Real r=5 "radius";
  constant Real pi=3.14;
  parameter Real A=pi*r^2;
equation
  der(x) = A-x "My equation";
end SimpleEquation;
```

Deafult equations (involving parameters) may be written in the parameter declaration

Modelica code

- As we have seen already, Modelica allows us to describe model behavior in terms of differential equations
- But the initial conditions we choose are just as important as the equations
- How to define the initial conditions?
  - Use ***initial equation***

```
model SimpleEquation "A simple
  Real x "State variable";
  parameter Real r=5 "radius";
  constant Real pi=3.14;
  parameter Real A=pi*r^2;
initial equation
  x=6;
equation
  der(x) = A-x "My equation";
end SimpleEquation;
```

If you don't define the initial conditions,
The tool gives you a warning and it uses 0

SESSION 3: Modelica overview

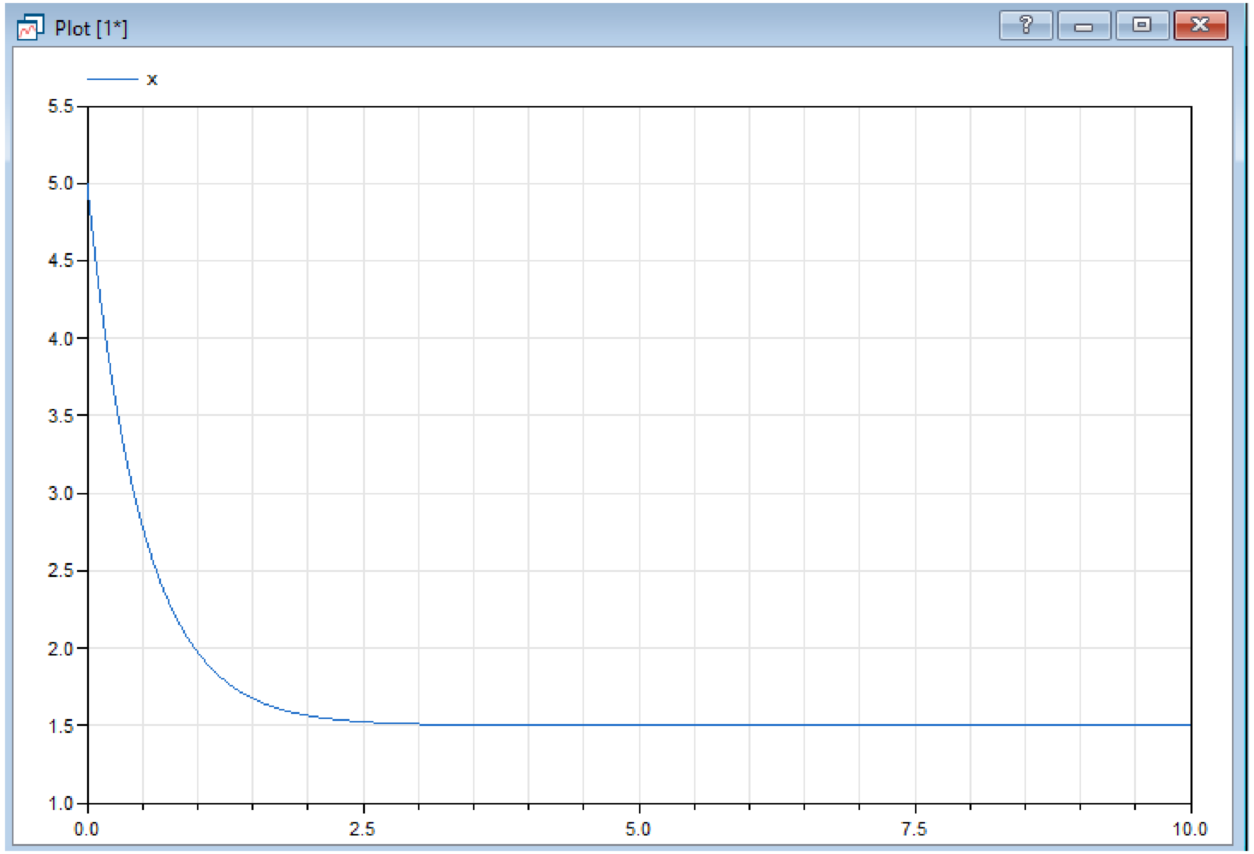
- Text editor
- Modelica code, variables and parameters
- **Exercise 2: “HelloWorld” in Modelica**
- Exercise 3: Getting physical: Cooling an object with air

Exercise2: "HelloWorld" in Modelica

- Write a Modelica model (named "HelloWorld") to solve the following differential equation:

$$\dot{x} = 3 - 2x$$
$$x(0) = 5$$

- Simulate the model for 10s
- Plot *x*



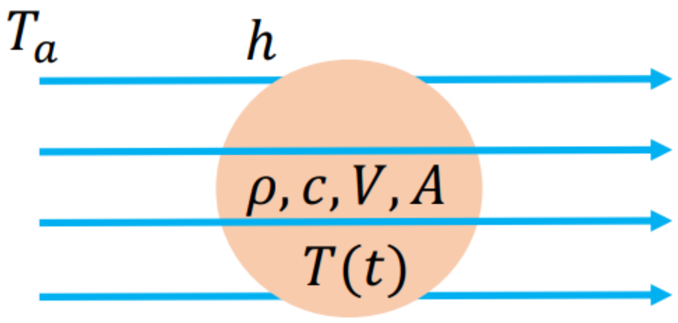
SESSION 3: Modelica overview

- Text editor
- Modelica code, variables and parameters
- Exercise 2: “HelloWorld” in Modelica
- **Exercise 3: Getting physical: Cooling an object with air**

Exercise3: Getting physical

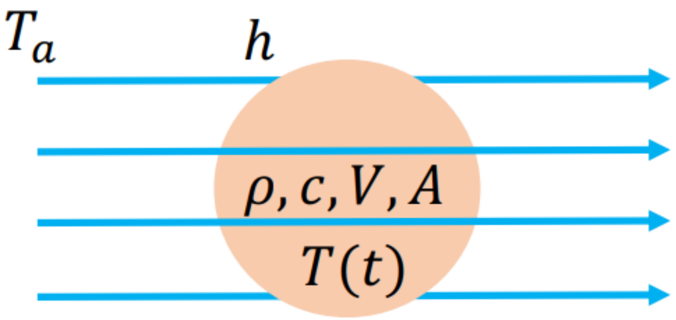
Physical phenomenon:

A small rock is being cooled by an air stream



Exercise3: Getting physical

Physical phenomenon:
A small rock is being cooled by an air stream



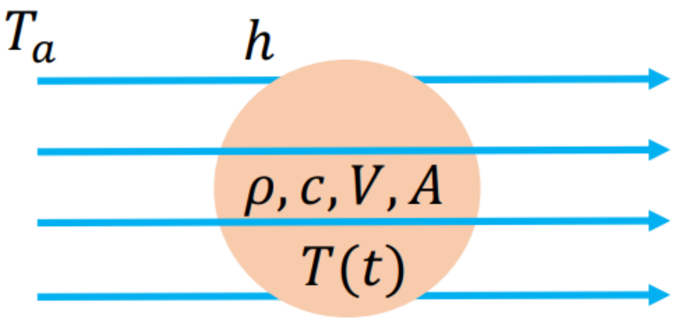
Physical equation:
We are interested in simulating the temperature of the rock over time. This can be described by the differential equation and initial condition below:

$$\rho V c_v \frac{dT}{dt} = hA(T_a - T)$$

$$T(t = 0) = T_0$$

Exercise3: Getting physical

Physical phenomenon:
A small rock is being cooled by an air stream



r=0.1 m
rho=2230 kg/m^3
c=880 J/kg K
h=1000 W/m^{2}K
T_a=273.15+10 K
T_0=273.15+20 K

Physical equation:
We are interested in simulating the temperature of the rock over time. This can be described by the differential equation and initial condition below:

$$\rho V c_v \frac{dT}{dt} = hA(T_a - T)$$

$$T(t=0) = T_0$$

Variables
(that change over time)

T is the rock temperature [K]

Initial condition

T_0 is the initial rock temperature [K]

Parameters

T_a is the air temperature [K]

ρ is the rock density [kg/m^3]

V is the rock volume [m^3]

c_v is the rock specific heat capacity [J/kg-K]

h is the rock-air heat transfer coefficient [W/m^2-K]

A is the rock surface area [m^2]

Try to write the Modelica code in the text editor, and simulate for 500 s

Exercise3: Getting physical

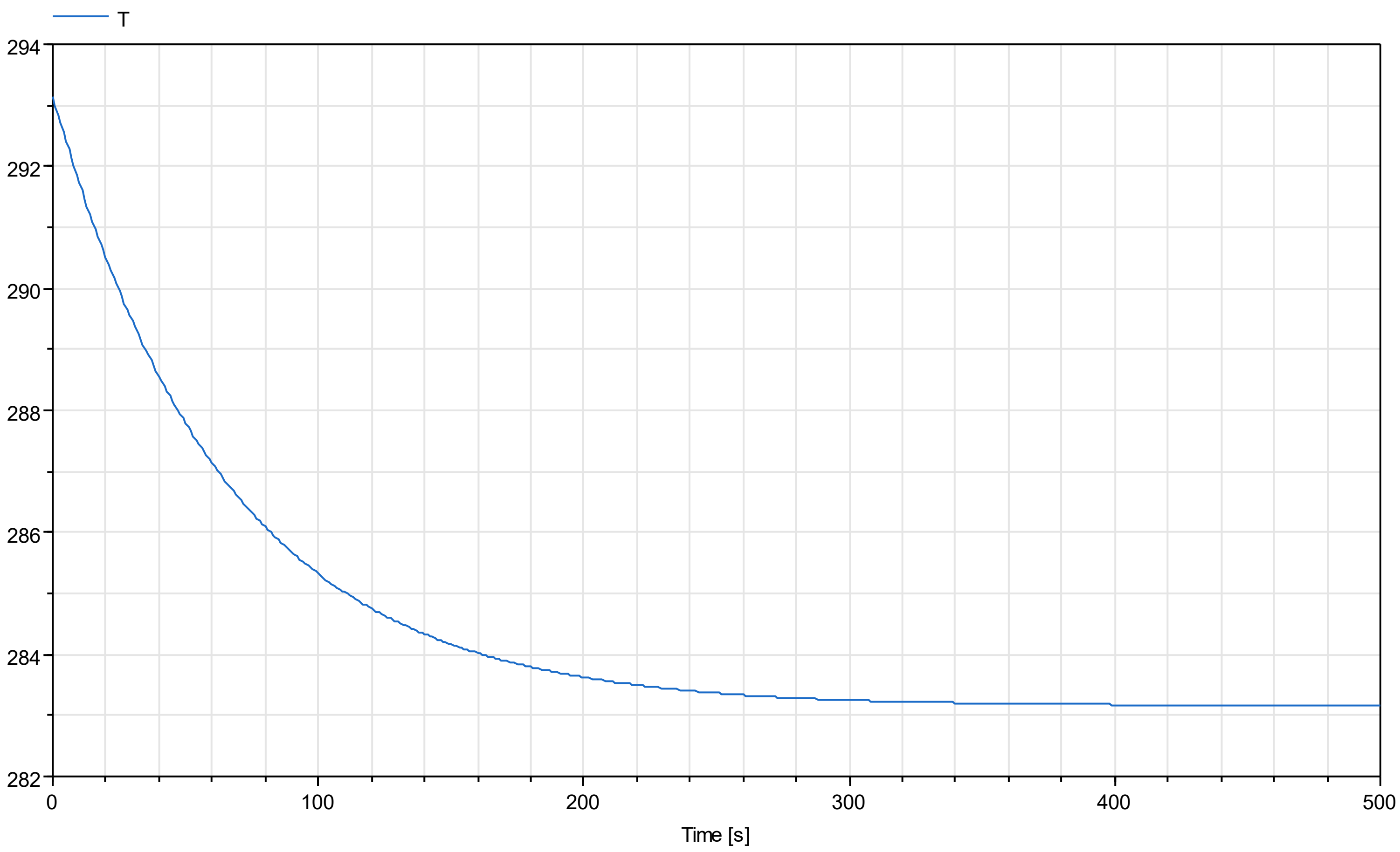
Step 1:
Declare all the parameters and variables

```
model Exercise3_noUnits "This is a model of a rock in air stream"
  parameter Real r = 0.1 "Density of rock";
  parameter Real rho = 2230 "Density of rock";
  parameter Real c = 880 "Specific heat capacity of rock";
  parameter Real h = 1000 "Heat tranfer coefficient";
  parameter Real T_a = 273.15+10 "Temperature of air stream";
  parameter Real T_0 = 273.15+20 "Initial temperature of rock";
  parameter Real V = (4/3)*3.14*r^3 "Volume of rock";
  parameter Real A = 4*3.14*r^2 "Surface area of rock";
  Real T "Temperature of rock";
initial equation
  T = T_0 "Used before simulation to compute initial values";
equation


end Exercise3_noUnits;
```

Step 2:
Add the equation for the heat transfer

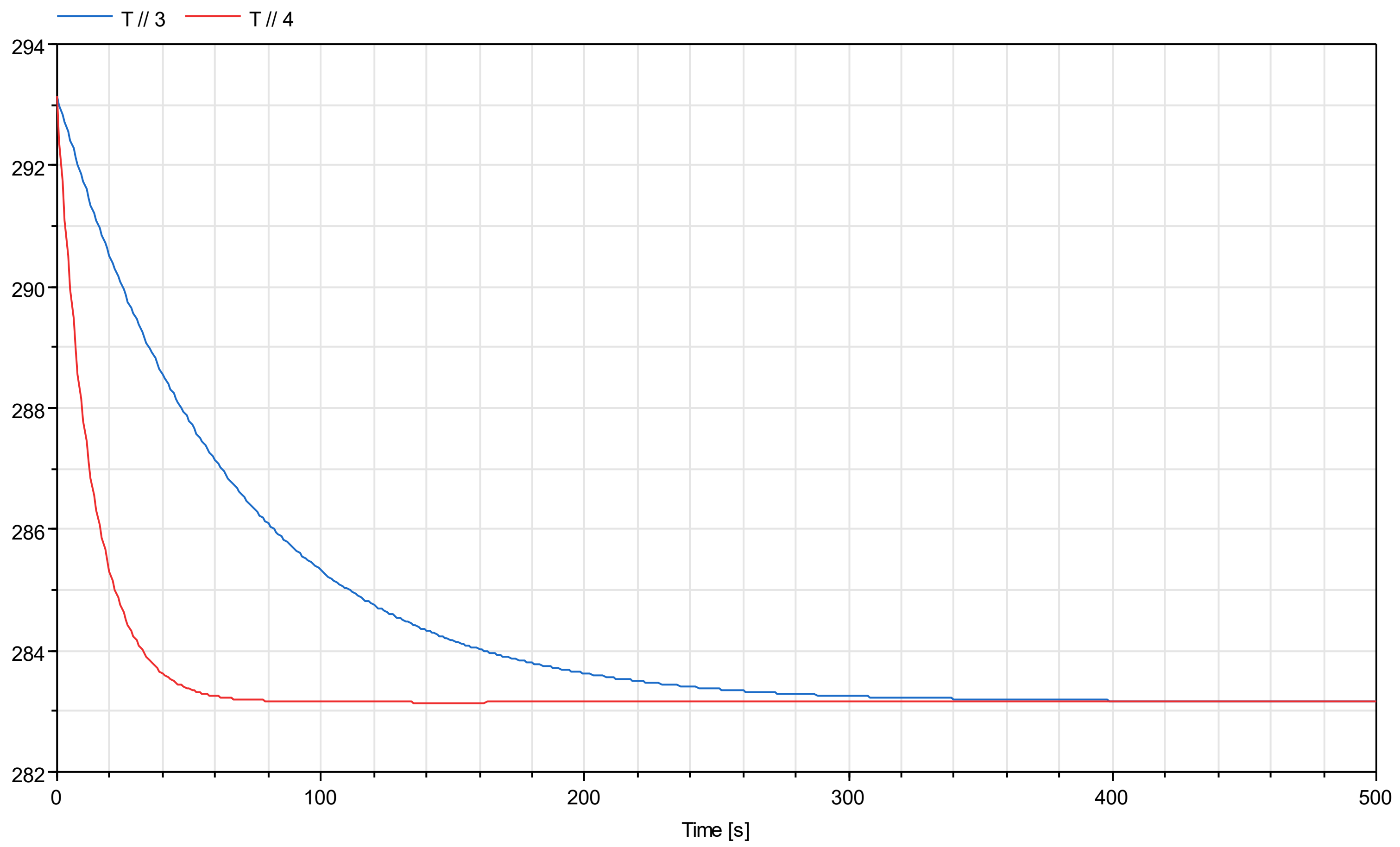
```
model Exercise3_noUnits "This is a model of a rock in air stream"
  parameter Real r = 0.1 "Density of rock";
  parameter Real rho = 2230 "Density of rock";
  parameter Real c = 880 "Specific heat capacity of rock";
  parameter Real h = 1000 "Heat tranfer coefficient";
  parameter Real T_a = 273.15+10 "Temperature of air stream";
  parameter Real T_0 = 273.15+20 "Initial temperature of rock";
  parameter Real V = (4/3)*3.14*r^3 "Volume of rock";
  parameter Real A = 4*3.14*r^2 "Surface area of rock";
  Real T "Temperature of rock";
initial equation
  T = T_0 "Used before simulation to compute initial values";
equation
  rho*V*c*der(T) = h*A*(T_a-T);

end Exercise3_noUnits;
```

Exercise3: Getting physical



Now try to change the value of the paramter h (heat transfer coefficient) from 1000 to 5000. What happens? Does it make sense physically?

Exercise3: Getting physical



Exercise3: Getting physical

How can we add units?

```
model Exercise3_noUnits "This is a model of a rock in air stream"
  parameter Real r = 0.1 "Density of rock";
  parameter Real rho = 2230 "Density of rock";
  parameter Real c = 880 "Specific heat capacity of rock";
  parameter Real h = 1000 "Heat tranfer coefficient";
  parameter Real T_a = 273.15+10 "Temperature of air stream";
  parameter Real T_0 = 273.15+20 "Initial temperature of rock";
  parameter Real V = (4/3)*3.14*r^3 "Volume of rock";
  parameter Real A = 4*3.14*r^2 "Surface area of rock";
  Real T "Temperature of rock";
initial equation
  T = T_0 "Used before simulation to compute initial values";
equation
  rho*V*c*der(T) = h*A*(T_a-T);

end Exercise3_noUnits;
```

```
model Exercise3 "This is a model of a rock in air stream"
  parameter Modelica.Units.SI.Length r = 0.1 "Density of rock";
  parameter Modelica.Units.SI.Density rho = 2230 "Density of rock";
  parameter Modelica.Units.SI.SpecificHeatCapacity c = 880 "Specific heat capacity of rock";
  parameter Modelica.Units.SI.CoefficientOfHeatTransfer h = 1000 "Heat tranfer coefficient";
  parameter Modelica.Units.SI.Temperature T_a = 273.15+10 "Temperature of air stream";
  parameter Modelica.Units.SI.Temperature T_0 = 273.15+20 "Initial temperature of rock";
  parameter Modelica.Units.SI.Volume V = (4/3)*3.14*r^3 "Volume of rock";
  parameter Modelica.Units.SI.Area A = 4*3.14*r^2 "Surface area of rock";
  Modelica.Units.SI.Temperature T "Temperature of rock";
initial equation
  T = T_0 "Used before simulation to compute initial values";
equation
  rho*V*c*der(T) = h*A*(T_a-T);

end Exercise3;
```