System Simulation Using VHDL-AMS: Modeling Multiple Physical Domains for HEV Applications

Xiao Hu
Scott Stanton
Vincent Delafosse
ANSYS Inc.
Introduction

- Overview of VHDL-AMS
- Applications:
  - Battery: Electrochemistry
  - Heat Exchanger: Fluid, Thermal
What is VHDL-AMS?

**Very High Speed Integrated Circuit Hardware Description Language – Analog and Mixed-Signal**

VHDL-AMS is a strict superset of IEEE Std. 1076
Why Use VHDL-AMS

- Standard Format Allows Model Portability
  - Different engineering groups within same company
  - With Sub-Contractors
  - Between different simulators
- Multi-level Modeling
  - Different levels of abstraction of model behavior
- Multi-domain Modeling
  - Electrical, Thermal, Magnetic, Mechanical, etc
- Mixed-signal Modeling
  - Supports analog and digital modeling
**VHDL-AMS Model Construct**

- **Entity**
  - Interface description of a subsystem or physical device
  - Specifies input and output ports to the model

- **Architecture**
  - Behavior description
  - Can be dataflow, structural, procedural, etc
  - Modeling can deal with both analog (continuous) and digital (discrete) domains

![Diagram of VHDL-AMS model with Entity and Architecture 1, 2, 3 connections to input and output ports]
**VHDL-AMS Code for a Resistor**

- **Entity**
  - Interface description of a subsystem or physical device
  - Specifies input and output ports to the model

```vhdl
LIBRARY Ieee;
use Ieee.electrical_systems.ALL;
ENTITY vhdl_resistor IS
  GENERIC (resistance : resistance := 1.0e3);
  PORT (QUANTITY temp : IN real := 300.0;
        TERMINAL p : electrical;
        TERMINAL n : electrical);
END ENTITY vhdl_resistor;
```

The resistor model has one model constant, one input quantity, and two terminals.
VHDL-AMS Code for a Resistor

- **Architecture**
  - Description of the model and no solving information is required

```vhdl
ARCHITECTURE arch_vhdl_resistor OF vhdl_resistor IS
QUANTITY voltage ACROSS current THROUGH p TO n;
BEGIN

voltage == current * resistance;

END ARCHITECTURE arch_vhdl_resistor;
```

No solving information is needed!!

```vhdl
ARCHITECTURE arch_vhdl_resistor OF vhdl_resistor IS
QUANTITY voltage ACROSS current THROUGH p TO n;
BEGIN

voltage == current * resistance * (1.0+1.0e-2*(temp-300.0)+1.1e-3*(temp-300.0)**2);

END ARCHITECTURE arch_vhdl_resistor;
```

A second architecture is possible!!
The capacitor entity has one model constant, one input, and two terminals

The model description essentially has two lines, one for initial condition and one for the governing equation!!
Open VHDL-AMS Basic Library
PM Motor example
Open VHDL Basic Library

```
ARCHITECTURE behav of Id IS
CONSTANT del : TIME := T_PROP * 1 sec;
BEGIN
  PROCESS (CLK, DIN)
  BEGIN
    IF (CLK='1') THEN
      Q <= DIN AFTER del;
      QB <= NOT DIN AFTER del;
    END IF;
  END PROCESS;
END behav;
```
The modeling of side-force by a tire.

Diagram of side force of Tire

Vehicle speed $V_v$
Rotational Speed of tire $\omega_v$
Slip rate $\beta$
Slip
Load $mg$
Side angle

Perpendicular load 380[kgf]

Traction force, Side force [kgf]

Side angle

Gap of a rack and rotational center : $R$
Sidewall equivalent radius : $R_c$

Force of Rack $F$

Ref. ISBN 4-381-08855-7
VHDL-AMS code

Translation <> Rotation

\[ T = R \times F \]
\[ V = \omega \times R \]

Slip rate

\[ Slip = (V_v - \omega_v) / V_v \]

Friction Coeff.

\[ \beta = \int \omega dt \]
\[ \mu = F_{\text{lookup}}(Slip, \beta) \]

Equations

\[ Fs = \mu \cdot mg \]
\[ J \dot{\omega} = (T - R \times Fs) \]

begin
\[ T == R \times F ; \]
\[ V == \omega \times R ; \]
if(vv'dot < 0.0) use
\[ \text{slip_rate} == (vv-ww)/vv ; \]
else
\[ \text{slip_rate} == (ww-vv)/ww ; \]
end use ;

\[ \text{beta} == \omega \text{integ} ; \]
\[ \text{mur} == \text{lookup}_\text{SideFC}(	ext{slip_rate, abs(beta)}) ; \]
\[ \text{Fs} == \text{mur} \times \text{mass} \times 9.8 \times \text{sign(beta)} ; \]
\[ J \times \omega \text{dot} == (T- Fs*R) ; \]
end architecture beh ;

• VHDL-AMS model expresses definitional equations directly.
Nonlinear side force model

Side force of Tire [N]

- Nonlinear side force mapping.
- Confirm reducing steering force by assist motor.
How to Use a VHDL-AMS Model in Simplorer?

- Drag and drop the VHDL-AMS model as if it is built-in.
- Double click to launch the property window as if it is built-in.
What About PDEs – Boundary Value Problems?

- **Example:**
  \[
  \begin{cases}
  \frac{d\Phi}{dx} = x \\
  \Phi(0) = \phi_{bc}
  \end{cases}
  \]

- **Entity**

```vhdl
ENTITY steady_state_boundary_value IS
  generic (phibc: real := 0.0);
END ENTITY steady_state_boundary_value;
```

- **Architecture**

```vhdl
ARCHITECTURE arch_steady_state_boundary_value OF steady_state_boundary_value IS
  quantity phi1,phi2,phi3,phi4,phi5 : real;
  constant h : real := 0.25;
BEGIN
  phi1 == phibc;
  (phi3-phi1)/(2.0*h) == 1.0*h;
  (phi4-phi2)/(2.0*h) == 2.0*h;
  (phi5-phi3)/(2.0*h) == 3.0*h;
  (1.0*phi3-4.0*phi4+3.0*phi5)/(2.0*h) == 4.0*h;
END ARCHITECTURE arch_steady_state_boundary_value;
```
What About PDEs – Initial Value Problems?

• Example:

\[
\frac{\rho C_p}{k} \frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}
\]

\[
T(x,0) = 20000x \quad x \in [0, 0.005]
\]

\[
T(x,0) = 20000(0.01 - x) \quad x \in [0, 0.01]
\]

\[
T(0.0, t) = 0.0
\]

\[
T(0.01, t) = 0.0
\]

• Entity

ENTITY transient_diffusion IS
generic ( 
  rho: real := 2000.0;
  k:   real := 2.0;
  Cp:  real := 1000.0);
END ENTITY transient_diffusion;

• Architecture (main part)

IF (domain = quiescent_domain) USE

T1 == 20.0;
T2 == 60.0;
T3 == 100.0;
T4 == 60.0;
T5 == 20.0;

ELSE

 rho*Cp*T1'dot == -(N1p - N0p)/h;
 rho*Cp*T2'dot == -(N2p - N1p)/h;
 rho*Cp*T3'dot == -(N3p - N2p)/h;
 rho*Cp*T4'dot == -(N4p - N3p)/h;
 rho*Cp*T5'dot == -(N5p - N4p)/h;

END USE
Newman’s 1d Electrochemistry Model in Simplorer

- Electrochemical Kinetics
- Solid-State Li Transport
- Electrolytic Li Transport

\[ \eta = (\phi_s - \phi_a) - U \]
\[ i_0 = k(c_e)^a_z (c_{z,max} - c_{z,e})^a_z (c_{z,e})^a_z \]
\[ \frac{\partial (\varepsilon_c c_e)}{\partial t} = \nabla \cdot (D_e \nabla c_e) + \frac{1-t^+}{F} j^{Li} \]
\[ j^{Li} = a_z i_o \left\{ \exp \left[ \frac{\alpha_c F}{RT} \eta \right] - \exp \left[ -\frac{\alpha_c F}{RT} \eta \right] \right\} \]

Lithium Ion Batteries

\[ \text{Li}(1-x)\text{CoO}_2 + \text{Li}_x\text{C} \xrightarrow{j} \text{LiCoO}_2 + \text{C} \]

Results from Simplorer

Results from Newman
Governing Equations

The governing equations of porous electrode model of the lithium-ion battery (Electrochemical Systems, 3rd by John Newman)

\[
\varepsilon \frac{\partial c}{\partial t} = \nabla \cdot (\varepsilon D \nabla c) - \frac{i_2 \cdot \nabla t_+^0}{z_+ v_+ F} + \frac{a j_n (1 - t_+^0)}{v_+}
\]

\[
i_2 = -\kappa \nabla \phi_2 + \frac{\kappa RT}{F} (1 - t_+^0) \nabla \ln c
\]

\[
I - i_2 = -\sigma \nabla \phi_1
\]

\[
a F j_n = \nabla \cdot i_2
\]

\[
 j_n = i_0 \left\{ \exp \left( \frac{\alpha_a F}{RT} \eta_s \right) - \exp \left( - \frac{\alpha_c F}{RT} \eta_s \right) \right\}
\]
LIBRARY leee;
use leee.thermal_systems.all;
use leee.fluidic_systems.all;
use leee.math_real.ALL;
use leee.electrical_systems.ALL;
ENTITY battery IS
    generic ( 
        diffD : real := 7.5e-11; 
        epslonn: real := 0.357; 
        epslonp: real := 0.444; 
        epslonfp: real := 0.259; 
        sigman : real := 100.0; 
        sigmap: real := 3.8; 
        diffDsn: real := 3.9e-14; 
        diffDsp: real := 1.0e-13; 
        kn: real := 2.334e-11; 
        kp: real := 2.334e-11; 
        Bruggn: real :=1.5;  
        Bruggp: real :=1.5; 
        zp : real := 1.0;  
        mup: real := 1.0; 
        sp : real :=1.0; 
        n : real := 1.0; 
    )
    PORT ( 
        TERMINAL negative : electrical; 
        TERMINAL positive : electrical 
    );
END ENTITY battery;

Note that the model has two terminals and the rest are simply parameters of the model.
Entities are Passed to Model Properties Window

![Entities are Passed to Model Properties Window](image-url)

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Unit</th>
<th>Evaluated Value</th>
<th>Description</th>
<th>Callback</th>
<th>Overide</th>
<th>Property Type</th>
<th>Show Pin</th>
<th>Si</th>
</tr>
</thead>
<tbody>
<tr>
<td>dffd</td>
<td>7.5e-011</td>
<td></td>
<td>7.5e-011</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>epslon</td>
<td>0.357</td>
<td></td>
<td>0.357</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>epslonc</td>
<td>0.444</td>
<td></td>
<td>0.444</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>epslonf</td>
<td>0.172</td>
<td></td>
<td>0.172</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>epslonfc</td>
<td>0.259</td>
<td></td>
<td>0.259</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>egna</td>
<td>100</td>
<td></td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sigmc</td>
<td>3.8</td>
<td></td>
<td>3.8</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dffdsa</td>
<td>3.3e-014</td>
<td></td>
<td>3.3e-014</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dffdsc</td>
<td>1e-013</td>
<td></td>
<td>1e-013</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ka</td>
<td>2.334e-011</td>
<td></td>
<td>2.334e-011</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kc</td>
<td>2.334e-011</td>
<td></td>
<td>2.334e-011</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>brugga</td>
<td>1.5</td>
<td></td>
<td>1.5</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bruggc</td>
<td>1.5</td>
<td></td>
<td>1.5</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zd</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mup</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sp</td>
<td>-1</td>
<td></td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>i</td>
<td>17.5</td>
<td></td>
<td>17.5</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>csomax</td>
<td>26390</td>
<td></td>
<td>26390</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>csmmax</td>
<td>22860</td>
<td></td>
<td>22860</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>alfa</td>
<td>0.5</td>
<td></td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>delta</td>
<td>5.0e-005</td>
<td></td>
<td>5.0e-005</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>deltax</td>
<td>0.0001</td>
<td></td>
<td>0.0001</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>deltac</td>
<td>0.00183</td>
<td></td>
<td>0.00183</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rea</td>
<td>1.25e-005</td>
<td></td>
<td>1.25e-005</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rnc</td>
<td>8e-006</td>
<td></td>
<td>8e-006</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>96485.3</td>
<td></td>
<td>96485.3</td>
<td></td>
<td></td>
<td></td>
<td>Real</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
VHDL-AMS Implementation:
Architecture

governing equation for concentration in particles in anode

4.0/3.0*(***)(1.0*hpa, 3.0)-"**"(0.0*hpa, 3.0)*cs1_1 dot = -(diffDsa*(cs1_2-cs1_1)/hpa*4.0*(1.0*hpa) )
4.0/3.0*(***)(2.0*hpa, 3.0)-"**"(1.0*hpa, 3.0)*cs1_2 dot = -(diffDsa*(cs1_3-cs1_2)/hpa*4.0*(2.0*hpa) + diffDsa*(cs1_2-cs1_1)/hpa*4.0*(2.0*hpa) )
4.0/3.0*(***)(3.0*hpa, 3.0)-"**"(2.0*hpa, 3.0)*cs1_3 dot = -(diffDsa*(cs1_4-cs1_3)/hpa*4.0*(3.0*hpa) + diffDsa*(cs1_3-cs1_2)/hpa*4.0*(3.0*hpa) )
4.0/3.0*(***)(4.0*hpa, 3.0)-"**"(3.0*hpa, 3.0)*cs1_4 dot = -(diffDsa*(cs1_5-cs1_4)/hpa*4.0*(4.0*hpa) + diffDsa*(cs1_4-cs1_3)/hpa*4.0*(4.0*hpa) )
4.0/3.0*(***)(5.0*hpa, 3.0)-"**"(4.0*hpa, 3.0)*cs1_5 dot = -(diffDsa*(cs1_6-cs1_5)/hpa*4.0*(5.0*hpa) + diffDsa*(cs1_5-cs1_4)/hpa*4.0*(5.0*hpa) )
4.0/3.0*(***)(6.0*hpa, 3.0)-"**"(5.0*hpa, 3.0)*cs1_6 dot = -(diffDsa*(cs1_7-cs1_6)/hpa*4.0*(6.0*hpa) + diffDsa*(cs1_6-cs1_5)/hpa*4.0*(6.0*hpa) )
4.0/3.0*(***)(7.0*hpa, 3.0)-"**"(6.0*hpa, 3.0)*cs1_7 dot = -(diffDsa*(cs1_8-cs1_7)/hpa*4.0*(7.0*hpa) + diffDsa*(cs1_7-cs1_6)/hpa*4.0*(7.0*hpa) )
4.0/3.0*(***)(8.0*hpa, 3.0)-"**"(7.0*hpa, 3.0)*cs1_8 dot = -(diffDsa*(cs1_9-cs1_8)/hpa*4.0*(8.0*hpa) + diffDsa*(cs1_8-cs1_7)/hpa*4.0*(8.0*hpa) )
4.0/3.0*(***)(9.0*hpa, 3.0)-"**"(8.0*hpa, 3.0)*cs1_9 dot = -(diffDsa*(cs1_10-cs1_9)/hpa*4.0*(9.0*hpa) + diffDsa*(cs1_9-cs1_8)/hpa*4.0*(9.0*hpa) )
4.0/3.0*(***)(10.0*hpa, 3.0)-"**"(9.0*hpa, 3.0)*cs1_10 dot = -(diffDsa*(cs1_11-cs1_10)/hpa*4.0*(10.0*hpa) + diffDsa*(cs1_10-cs1_9)/hpa*4.0*(10.0*hpa) )
4.0/3.0*(***)(11.0*hpa, 3.0)-"**"(10.0*hpa, 3.0)*cs1_11 dot = -(diffDsa*(cs1_12-cs1_11)/hpa*4.0*(11.0*hpa) + diffDsa*(cs1_11-cs1_10)/hpa*4.0*(11.0*hpa) )
4.0/3.0*(***)(12.0*hpa, 3.0)-"**"(11.0*hpa, 3.0)*cs1_12 dot = -(diffDsa*(cs1_13-cs1_12)/hpa*4.0*(12.0*hpa) + diffDsa*(cs1_12-cs1_11)/hpa*4.0*(12.0*hpa) )
4.0/3.0*(***)(13.0*hpa, 3.0)-"**"(12.0*hpa, 3.0)*cs1_13 dot = -(diffDsa*(cs1_14-cs1_13)/hpa*4.0*(13.0*hpa) + diffDsa*(cs1_13-cs1_12)/hpa*4.0*(13.0*hpa) )
4.0/3.0*(***)(14.0*hpa, 3.0)-"**"(13.0*hpa, 3.0)*cs1_14 dot = -(diffDsa*(cs1_15-cs1_14)/hpa*4.0*(14.0*hpa) + diffDsa*(cs1_14-cs1_13)/hpa*4.0*(14.0*hpa) )
4.0/3.0*(***)(15.0*hpa, 3.0)-"**"(14.0*hpa, 3.0)*cs1_15 dot = -(diffDsa*(cs1_16-cs1_15)/hpa*4.0*(15.0*hpa) + diffDsa*(cs1_15-cs1_14)/hpa*4.0*(15.0*hpa) )
4.0/3.0*(***)(16.0*hpa, 3.0)-"**"(15.0*hpa, 3.0)*cs1_16 dot = -(diffDsa*(cs1_17-cs1_16)/hpa*4.0*(16.0*hpa) + diffDsa*(cs1_16-cs1_15)/hpa*4.0*(16.0*hpa) )
4.0/3.0*(***)(17.0*hpa, 3.0)-"**"(16.0*hpa, 3.0)*cs1_17 dot = -(diffDsa*(cs1_18-cs1_17)/hpa*4.0*(17.0*hpa) + diffDsa*(cs1_17-cs1_16)/hpa*4.0*(17.0*hpa) )
4.0/3.0*(***)(18.0*hpa, 3.0)-"**"(17.0*hpa, 3.0)*cs1_18 dot = -(diffDsa*(cs1_19-cs1_18)/hpa*4.0*(18.0*hpa) + diffDsa*(cs1_18-cs1_17)/hpa*4.0*(18.0*hpa) )
4.0/3.0*(***)(19.0*hpa, 3.0)-"**"(18.0*hpa, 3.0)*cs1_19 dot = -(diffDsa*(cs1_20-cs1_19)/hpa*4.0*(19.0*hpa) + diffDsa*(cs1_19-cs1_18)/hpa*4.0*(19.0*hpa) )
4.0/3.0*(***)(20.0*hpa, 3.0)-"**"(19.0*hpa, 3.0)*cs1_20 dot = -(sn1*4.0*(20.0*hpa) + diffDsa*(cs1_20-cs1_19)/hpa*4.0*(20.0*hpa) )
Results: Newman 1d Model

Simplorer’s Results

White’s Results

Results: Newman 1d Model

Simplorer’s Results

White’s Results

Results: Newman 1d Model

Concentration profiles due to different temperatures

Discharge curves due to different temperatures

Heat Exchanger for IGBT Package

• IGBTs used in high power devices generate a lot of heat.
  – Temperature goes high
  – To maintain proper temperature is thus important

• A thermal model, suitable for coupling with circuit simulator, can be implemented in Simplorer using VHDL-AMS.
• The real problem is highly three dimensional and complex.
  – 3d flow inside the cooling pipes
  – 3d conduction everywhere else
  – 1d flow assumption is made

• Parallel-flow heat exchanger

\[
\begin{align*}
-C_p \rho \frac{\partial T_h}{\partial t} &= \dot{m}_h C_p \frac{\partial T_h}{\partial x} + h_B T_h - h_B T_c \\
-C_p \rho \frac{\partial T_c}{\partial t} &= \dot{m}_c C_p \frac{\partial T_c}{\partial x} + h_B T_c - h_B T_h \\
T_h (0) &= T_{h,0} \\
T_c (0) &= T_{c,0}
\end{align*}
\]

The above derivation can be found in many heat transfer text books: Incropera : Fundamentals of heat and mass transfer
Steady State Solution for the Heat Exchanger Model

- Exact solution exists for steady state under constant properties
- For $m_{h} = m_{c} = 0.4; C_{p_{h}} = C_{p_{c}} = 1000; h = 1000; B_{h} = B_{c} = 0.15; \rho_{h} = \rho_{c} = 1000$

\[
\begin{align*}
0 &= 400 \frac{dT_{h}}{dx} + 150T_{h} - 150T_{c} \\
0 &= 400 \frac{dT_{c}}{dx} + 150T_{c} - 150T_{h}
\end{align*}
\]

- All SI units above above
- Analytical solution

\[
\begin{align*}
T_{h} &= 400 + 100e^{-0.75x} \\
T_{c} &= 400 - 100e^{-0.75x}
\end{align*}
\]
Validation against Analytical Solution

Th-Tc (K) vs. x (m)

- **Simplorer: VHDL-AMS**
- **Analytical**

© 2010 ANSYS, Inc. All rights reserved.
Test 1: Zero Heat Transfer Coefficient

- With zero heat transfer coefficient, the flows in the two pipes are decoupled. Outlet temperatures should follow inlet temperatures exactly with delay.
- Numerical diffusion caused slight reduction of amplitude

Inlet Temperatures

Outlet Temperatures
Test 2 : Positive Heat Transfer Coefficient

- Mean temperature change is about 50K, which is consistent with the steady state case.
- The shape of the temperature curve is preserved (with shift of mean value), which is the expected behavior for convection equations.
- Notice the delay of about 5 sec at the outlet. This amount of delay is corresponding to the length of 1 meter and convection speed of 0.22 m/s.

Inlet Temperatures

Outlet Temperatures
Test 3: Different Temperature Oscillation Frequencies

- Note that the two inlets have different frequencies. This will distort temperature shape at outlet because the two convection equations are coupled.
- Mean temperature change is still 50 K as expected.
As expected, two discrete frequencies are observed for both outlet temperature signals after a 8-point Discrete Fourier Series.

- DC value not shown due to their rather large values.
What is Not Suitable for VHDL-AMS?

- **PDEs in 3D**
  - Discretization becomes complex without help of a commercial code.
  - It becomes tedious to write discretized equations.
  - Efficiency

- **If a specialized solver is needed to solve the governing equations efficiently, then VHDL-AMS might not be as efficient.**
  - 1D Euler equation in fluid dynamics
    - Flux difference splitting (Roe’s method)
Conclusion

- VHDL-AMS is perfect for users to include a customer model in system level simulations.

- VHDL-AMS programmers only need to write the governing equations after discretization. Solver and post-processing is provided by the simulating environment, for instance, Simplorer.

- VHDL-AMS can easily handle devices governed by algebraic, ODEs, and PDEs in 1D or even 2D.

- VHDL-AMS can NOT replace CFD, FEA, Maxwell, or specialized solvers.
Thank you!!
Workshop : Create a RC Circuit Using VHDL-AMS in Simplotrer
Create a RC Circuit Using VHDL-AMS in Simplorer

- Start Simplorer V9.
- Right Mouse Button (RMB) on Models, and select Add Definition...
- Change the Name to vhdl_resistor.
- Press the OK button.
Create a RC Circuit Using VHDL-AMS in Simplorer

- Select VHDL Model Editor>Edit Entry>Terminal.
- Select Library and Package.
- Select electrical_system.ALL
- Do NOT press the OK button yet.
Create a RC Circuit Using VHDL-AMS in Simplorer

- Select Terminal.
- Add two terminals as shown below by pressing the add button.
- Do NOT press the OK button yet.
Create a RC Circuit Using VHDL-AMS in Simplorer

- Select Quantity.
- Add one quantity as shown below.
- Do NOT press the OK button yet.

![Edit Model dialog box with a selected quantity](image)
Create a RC Circuit Using VHDL-AMS in Simplorer

- Select Generic
- Add one generic as shown below.
- Press the OK button.
Create a RC Circuit Using VHDL-AMS in Simplorer

• Select the vhdl_resistor tab and the entity part of the VHDL-AMS should look as follows.

• Note that you could have edited the entity directly. But as a beginner, you may find using the panel easier.
Create a RC Circuit Using VHDL-AMS in Simplorer

- Select the arch: arch_vhdl_resistor tab.
- For the architecture, we will edit the content directly. The architecture should look as follows.

```vhdl
ARCHITECTURE declaration arch_vhdl_resistor ARCHITECTURE arch_vhdl_resistor OF vhdl_resistor IS

QUANTITY voltage ACROSS current THROUGH p TO n;
BEGIN

    voltage == current * resistance;

END ARCHITECTURE arch_vhdl_resistor;

END ARCHITECTURE vhdl_resistor;
```
Create a RC Circuit Using VHDL-AMS in Simplorer

- Select VHDL Model Editor>Check Syntax. Syntax check should give no error in the Message Manager window.
Create a RC Circuit Using VHDL-AMS in Simplorer

- Select VHDL Model Editor > Update Project. Import Component window shows up.
- This window is to create a component out of your model. At this stage, you just need to click on OK
Create a RC Circuit Using VHDL-AMS in Simplorer

• Your model is now part of the project, and ready to be used.
• Double click on the design name.
• This will make the schematic available again. You can now drag and drop this component on the schematic.
• You can edit the symbol for the component. This part is skipped here since it is not critical.
Create a RC Circuit Using VHDL-AMS in Simplorer

- Follow the same procedure to create the model for the capacitor.
- You are ready to connect them to create a RC circuit.
Appendix B: VHDL-AMS Source Code for the Boundary Value Problem

--------- VHDLAMS MODEL steady_state_boundary_value ---------
--------- ENTITY DECLARATION steady_state_boundary_value ---------
ENTITY steady_state_boundary_value IS
    generic (phibc : real := 0.0);
END ENTITY steady_state_boundary_value;

--------- ARCHITECTURE DECLARATION
arch_steady_state_boundary_value ---------
ARCHITECTURE arch_steady_state_boundary_value OF steady_state_boundary_value IS

quantity phi1, phi2, phi3, phi4, phi5 : real;
constant h : real := 0.25;

BEGIN

phi1 == phibc;
(phi3-phi1)/(2.0*h) == 1.0*h;
(phi4-phi2)/(2.0*h) == 2.0*h;
(phi5-phi3)/(2.0*h) == 3.0*h;
(1.0*phi3-4.0*phi4+3.0*phi5)/(2.0*h) == 4.0*h;

END ARCHITECTURE arch_steady_state_boundary_value;
--------- END VHDLAMS MODEL steady_state_boundary_value ---------

© 2010 ANSYS, Inc. All rights reserved.
--- VHDLAMS MODEL transient_diffusion ---
--- ENTITY DECLARATION transient_diffusion ---
ENTITY transient_diffusion IS
  generic (
    rho: real := 2000.0;
    k:   real := 2.0;
    Cp:  real := 1000.0
  );
END ENTITY transient_diffusion;
ARCHITECTURE DECLARATION arch_transient_diffusion

ARCHITECTURE arch_transient_diffusion OF transient_diffusion IS

quantity T1, T2, T3, T4, T5, N0p, N1p, N2p, N3p, N4p, N5p : real;
constant h : real := 2.0e-3;

BEGIN

N0p == -k*(4.0*T1-0.5*(T1+T2))/h;
N1p == -k*(T2-T1)/h;
N2p == -k*(T3-T2)/h;
N3p == -k*(T4-T3)/h;
N4p == -k*(T5-T4)/h;
N5p == -k*(0.5*(T4+T5)-4.0*T5)/h;

IF (domain = quiescent_domain) USE

T1 == 20.0;
T2 == 60.0;
T3 == 100.0;
T4 == 60.0;
T5 == 20.0;

ELSE

rho*Cp*T1'dot == -(N1p - N0p)/h;
rho*Cp*T2'dot == -(N2p - N1p)/h;
rho*Cp*T3'dot == -(N3p - N2p)/h;
rho*Cp*T4'dot == -(N4p - N3p)/h;
rho*Cp*T5'dot == -(N5p - N4p)/h;

END USE;

END ARCHITECTURE arch_transient_diffusion;

END VHDLAMS MODEL transient_diffusion;