FAST CALCULATION OF REFRIGERANT PROPERTIES IN VAPOR COMPRESSION CYCLES USING SPLINE-BASED TABLE LOOK-UP METHOD (SBTL)

Lixiang Li Jesse Gohl John Batteh
Christopher Greiner Kai Wang

For The American Modelica Conference 2018

OUTLINE

- Introduction
- Methodology
- Implementation
- Verification & Validation

INTRODUCTION -- MOTIVATIONS

- Dynamic simulation of two-phase systems requires significant numbers of property calculations
- Typical multi-parameter equation of state (EOS) involves costly function evaluation
  - Short formulation not always available (e.g. R1234yf)
- EOS and the system model use different set of state variables → iteration needed
  - e.g. EOS based on Helmholtz energy use (ρ, T) while the system model uses (p, h)

INTRODUCTION -- MOTIVATIONS

- Use interpolation method instead
- The Spline-Based Table Look-Up (SBTL) Method (Kunick et al, 2015) was chosen
  - Equidistant grid *Searching*
  - Continuous first derivatives
  - Analytic inverse *Iterations*
  - Consistent phase boundary definition *Chattering*
- Speed and robustness proven by CFD and thermal cycle simulations

INTRODUCTION – WHAT'S NEW

- One overall fit over the whole (p, h) domain
- Minimum use of grid transformation
- Implemented in Modelica with external C functions for spline evaluation, inversion and derivatives

METHODOLOGY

- A specific type of quadratic/biquadratic spline on a equidistant grid (Späth, 1995)
  - Node -- where the spline intersects with the raw data points
  - Knot -- where two adjacent pieces of splines meet
  - A node is the mid-point of two neighboring knots
  - A global fit

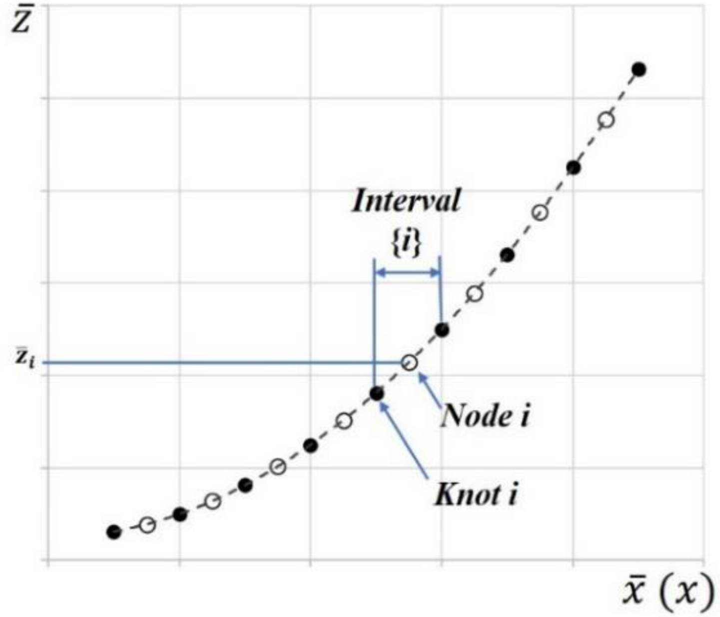


Illustration of the spline interpolation. Knots are represented by solid dots and node by hollow ones.

METHODOLOGY – 1D EXAMPLE

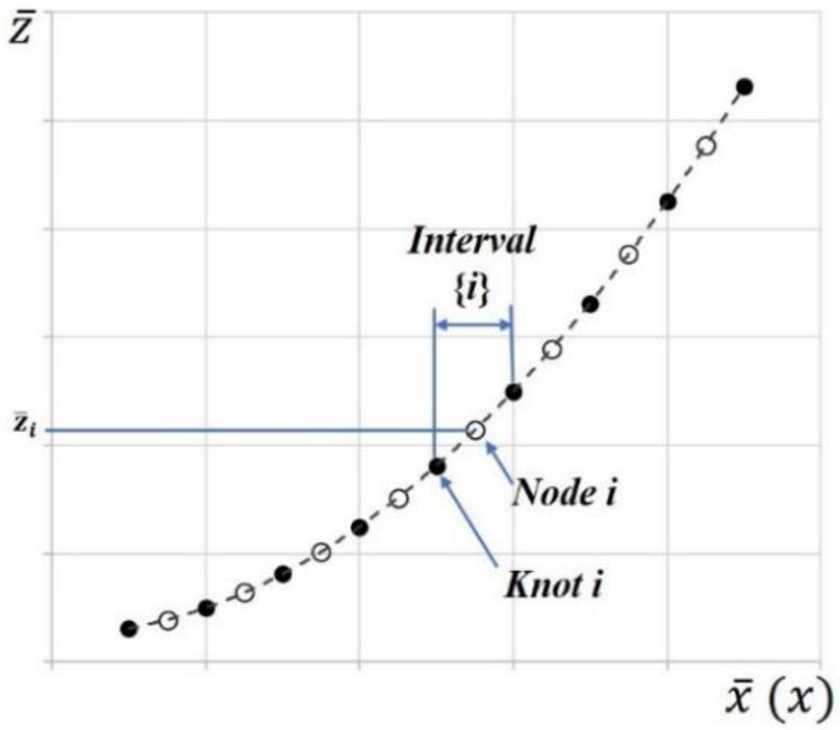


Illustration of the spline interpolation. Knots are represented by solid dots and node by hollow ones.

$$i = \text{floor}\left(\frac{\bar{x} - \bar{x}_1{}^K}{\Delta\bar{x}}\right)$$

$$\bar{z}_{\{i\}}(\bar{x}) = \sum_{k=1}^{3} a_{ik}\,(\bar{x} - \bar{x}_i)^{k-1}$$

$\bar{x}_1{}^K$: first knot of the whole spline
$\Delta\bar{x}$: distant between neighboring nodes (or knots)

Inverse: $$\bar{x}_{\{i\}}{}^{INV}(\bar{z}) = \bar{x}_i + \frac{-a_{i2} \pm \sqrt{a_{i2}{}^2 - 4a_{i3}(a_{i1} - \bar{z})}}{2a_{i3}}$$

Where the sign ($\pm$) equals to sign(a_{i2}).

METHODOLOGY – COORDINATE TRANSFORMATIONS

- Highly non-linear functions → Substantial increase in number of nodes
- End up in larger data files, especially for 2D splines, data file size ~$O({n_{node}}^2)$
- Use proper coordinate transformation, e.g. $\bar{p} = \log(p)$

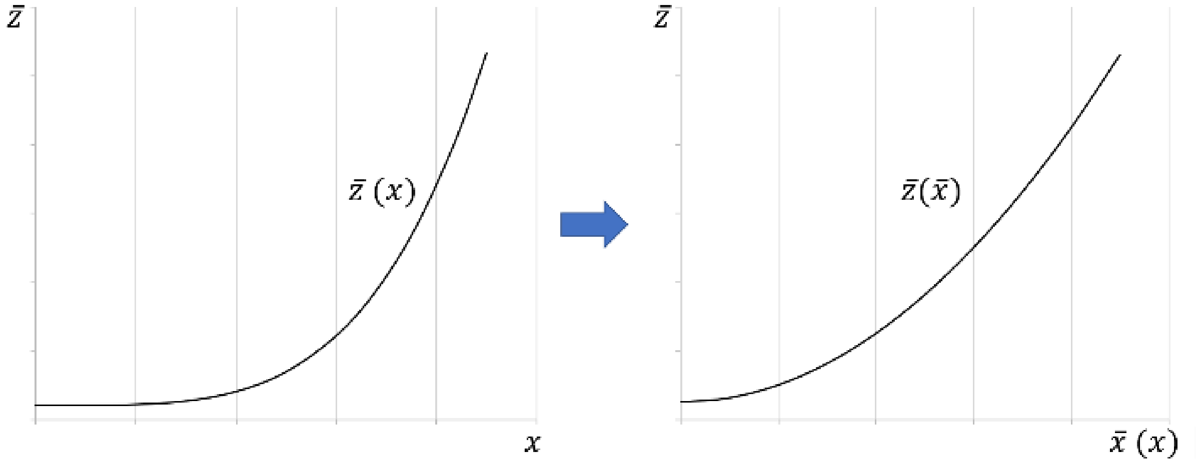
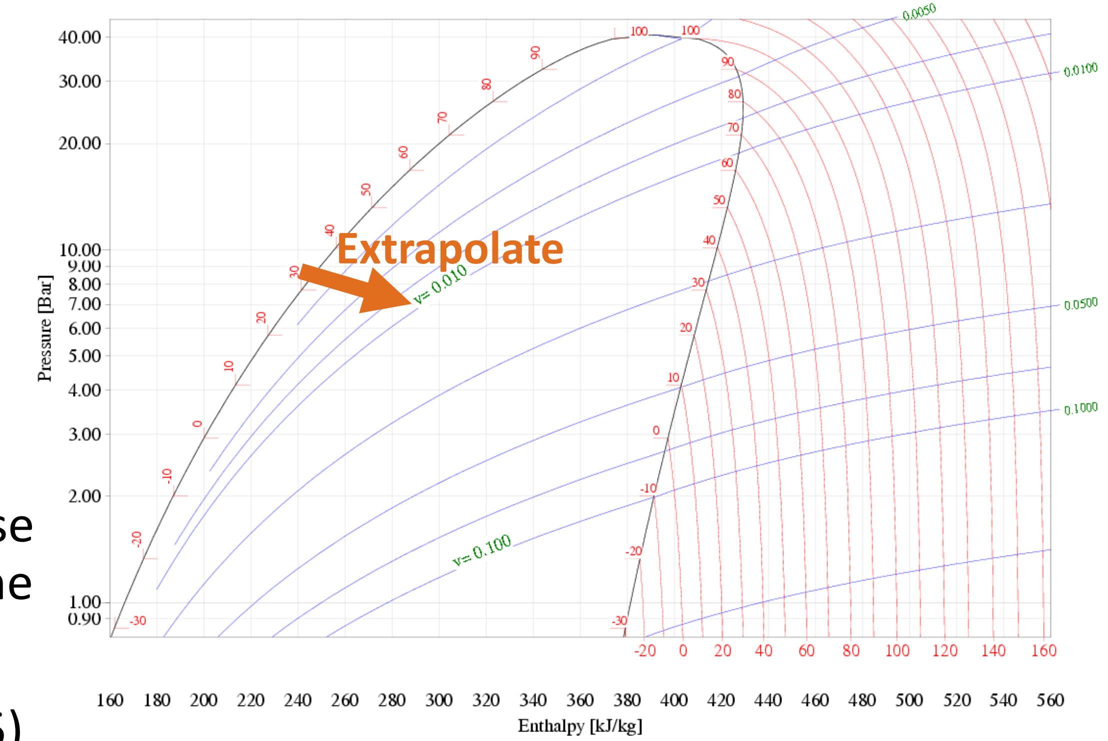


Illustration of coordinate transformation to enhance accuracy with equidistant nodes.

- Use Chain rule for derivatives

IMPLEMENTATION – DATA GENERATION

- Generated using property models in *Modelon's Air Conditioning Library (ACL)*
  - *R134a: Short formulation Helmholtz energy EOS*
  - *R1234yf: Reference Helmholtz energy EOS*
- 1D spline: 100 nodes
- 2D splines: About 120×120 nodes
- Logarithmic scale for pressure
- Extrapolated the raw data from single-phase to two-phase region to ensure accuracy of the resulted splines up to the phase boundary
- Spline coefficients solved (algorithm given in Späth, 1995) and stored in MAT files



Modelon

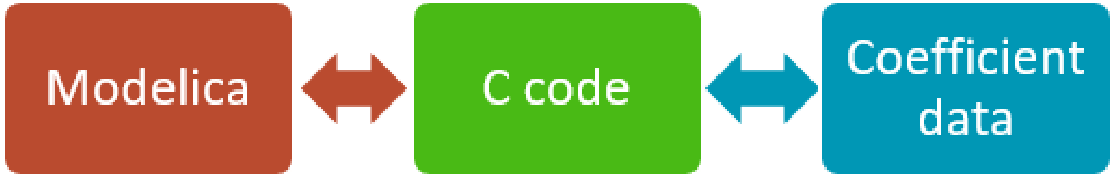
IMPLEMENTATION – THE TWO PHASE PROPERTY MODEL

- Phase boundary described by 1D spline: $T_s(p)$
- Single-phase region:
  - T, ρ, s given by 2D splines: $T(p,h), \rho(p,h), s(p,h)$
  - ${h_v}^{INV}(p, T_s(p))$ and ${h_l}^{INV}(p, T_s(p))$ by 2D inverse functions from $T(p,h)$ spline for consistency

    → Avoid chattering across phase boundary
- Two-phase region: Mass specific properties calculated based on vapor quality $x = \frac{(h - h_l)}{(h_v - h_l)}$
- Other properties like specific heat capacity can derived from the 1D and 2D splines (Tummescheit, 2002; Thorade & Saadat, 2013)

IMPLEMENTATION: STRUCTURE OF MEDIUM MODEL

- External C functions evaluate, invert, and take derivative of the spline
- Property functions are implemented in Modelica
  - Use External object that extracts spline coefficient data from a .mat file
  - Call the C functions and get the value
  - Calculate Derived properties



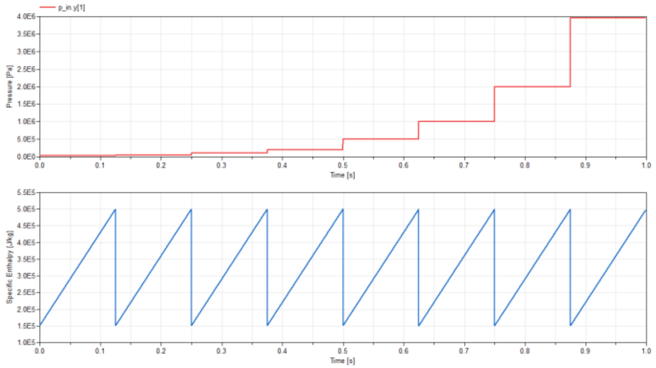
Modelon

V&V – HARDWARE AND SOFTWARE

Model	Dell Precision M2800 Laptop
Processor	Intel® Core™ i7-4810MQ CPU
RAM	16.0 GB
System	64-bit, x64 based, Windows 10 Pro
Software	Dymola 2018
C compiler	Visual Studio 2012 Express Edition
Solver	Euler (functions), Dassl(systems)
Tolerance	1e-6 (to ensure mass conservation)

V&V – FUNCTION TESTS

- Property functions for R134a are tested
- Compared against *Air Conditioning Library*
- 1D spline test for $T_s(p)$: pressure ramped from 0.3 to 39.5 bar
- 2D splines
  - Enthalpy ramped from 150kJ/kg to 500KJ/kg
  - Pressure fixed at 0.3, 0.5, 1, 2, 5, 10, 20 and 39.5 bar in each ramp



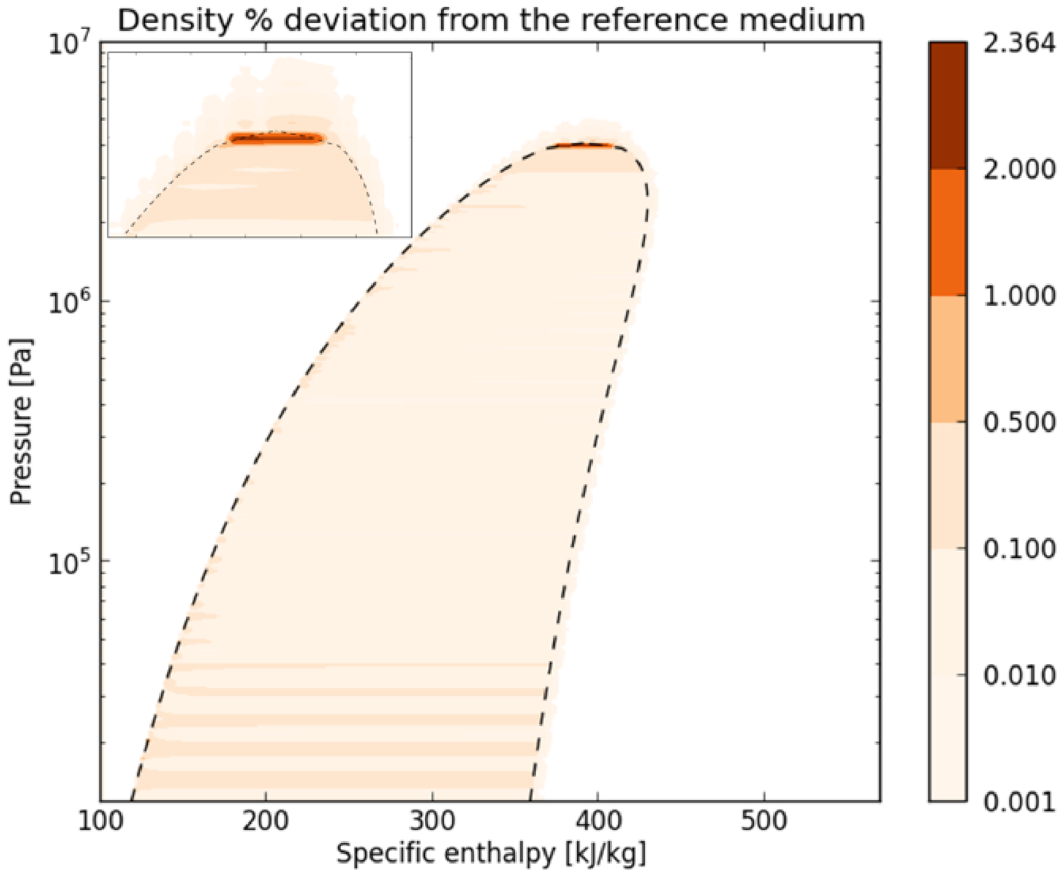
- Ran for 1s with a fixed step size of 1e-4s, resulting in 1e4 evaluations
- CPU time per evaluation was obtained by advanced profiling feature in Dymola

V&V – FUNCTION TESTS

Property	Relative error in %	CPU time short Helmholtz	CPU time SBTL	CPU Time reduced
h_{vap}	<0.5%	1.4e-6s	7e-7s	50%
T	<0.03%	>1.2e-5s	<2.0e-6s	>83.3%
$\left.\frac{\partial \rho}{\partial h}\right\|_p$	N/A[1]	>2.8e-5s	<3.5e-6s	>87.5%

[1]Phase boundary locations are slightly different in the two models, making it hard to compare the results in terms of percentage deviation.

V&V – FUNCTION TESTS

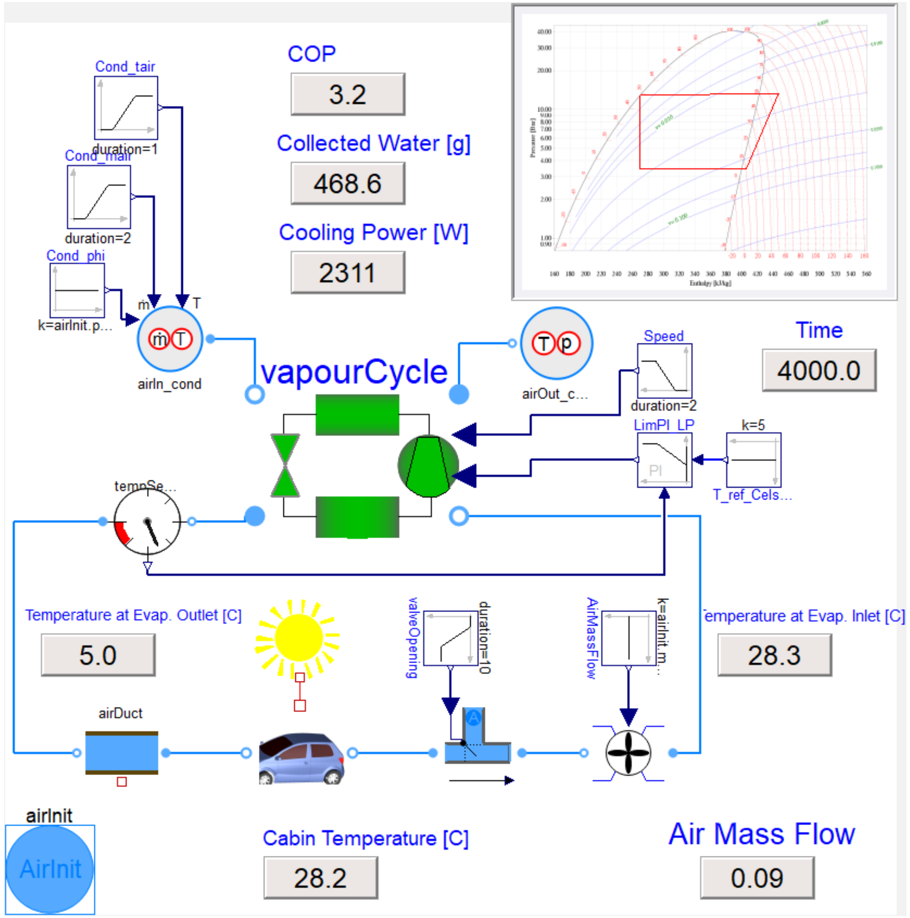


Contour plot of the % deviation of density (300×300 points evaluation).
Enlarged plot in p = 30-51 bar, h = 330-440 kJ/kg.

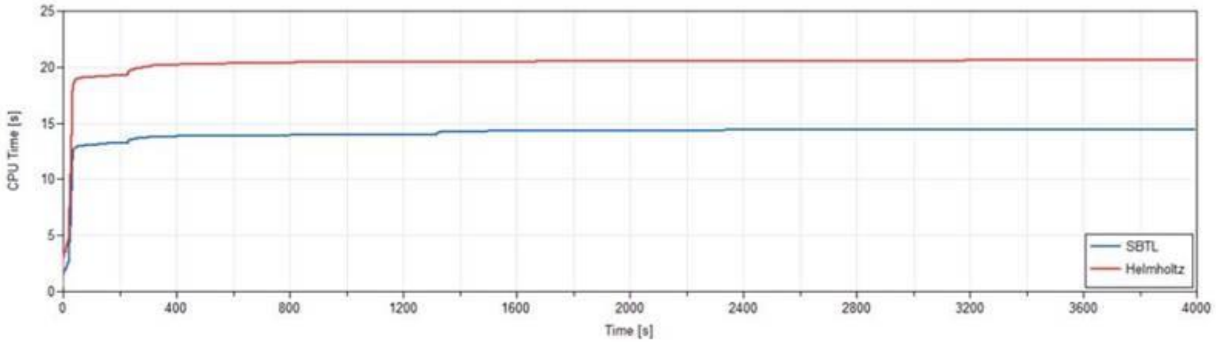
V&V – ACL REGRESSION TESTS

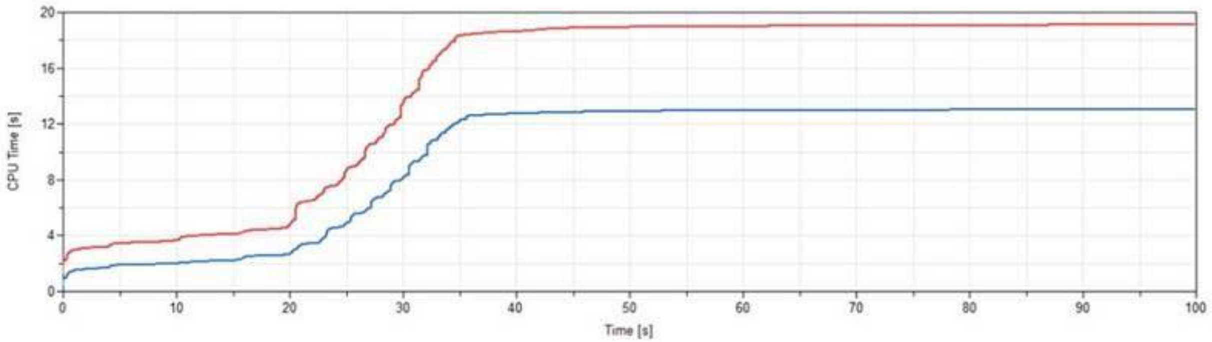
#	Model	Check	Translate	Simulate	Translation Time	Initialization Time	Simulation Time	Test Specification	CPU time plot	Trajectory Check
1	Evaporator	pass	pass [Reference]	pass [Reference]	15	0.6	1.6 (*10.0*)	Open [Reference]	View	pass
2	Compressor	pass	pass [Reference]	pass [Reference]	8	0.0	0.1 (*20.0*)	Open [Reference]	View	pass
3	Condenser	pass	pass [Reference]	pass [Reference]	13	0.2	1.3 (*10.0*)	Open [Reference]	View	pass
4	GasCooler	pass	pass [Reference]	pass [Reference]	13	0.1	1.7 (*60.0*)	Open [Reference]	View	pass
5	PlateEvaporator	pass	pass [Reference]	pass [Reference]	16	0.4	1.6 (*10.0*)	Open [Reference]	View	pass
6_1	IHX_1	pass	pass [Reference]	pass [Reference]	12	0.5	2.7 (*100.0*)	Open [Reference]	View	pass
6_2	IHX_2	pass	pass [Reference]	pass [Reference]	12	0.8	2.8 (*100.0*)	Open [Reference]	View	pass
6_3	IHX_3	pass	pass [Reference]	pass [Reference]	13	0.3	3.0 (*100.0*)	Open [Reference]	View	pass
6_4	IHX_4	pass	pass [Reference]	pass [Reference]	12	0.3	2.8 (*100.0*)	Open [Reference]	View	pass
6_5	IHX_5	pass	pass [Reference]	pass [Reference]	13	0.8	2.8 (*100.0*)	Open [Reference]	View	pass
6_6	IHX_6	pass	pass [Reference]	pass [Reference]	12	0.8	2.9 (*100.0*)	Open [Reference]	View	pass
6_7	IHX_7	pass	pass [Reference]	pass [Reference]	14	0.6	2.8 (*100.0*)	Open [Reference]	View	pass
6_8	IHX_8	pass	pass [Reference]	pass [Reference]	13	1.1	3.0 (*100.0*)	Open [Reference]	View	pass
7	SubcoolerCondenser	pass	pass [Reference]	pass [Reference]	17	1.1	3.0 (*20.0*)	Open [Reference]	View	pass
8	TXVCycle_1	pass	pass [Reference]	pass [Reference]	21	2.0	9.3 (*180.0*)	Open [Reference]	View	pass
8_1	TXVCycle_2	pass	pass [Reference]	pass [Reference]	23	2.0	21.0 (*180.0*)	Open [Reference]	View	pass
9	OrificeCycle	pass	pass [Reference]	pass [Reference]	18	1.2	9.5 (*180.0*)	Open [Reference]	View	pass
10	ChargeEstimation_1	pass	pass [Reference]	pass [Reference]	19	1.1	58.5 (*100.0*)	Open [Reference]	View	pass
10_1	ChargeEstimation_2	pass	pass [Reference]	pass [Reference]	21	1.1	61.2 (*100.0*)	Open [Reference]	View	pass
11	Co2Cycle	pass	pass [Reference]	pass [Reference]	19	2.0	6.9 (*180.0*)	Open [Reference]	View	pass
12	TXVCycleOnOff	pass	pass [Reference]	pass [Reference]	25	0.5	73.7 (*150.0*)	Open [Reference]	View	pass
13	Co2CycleOptimizedCOP	pass	pass [Reference]	pass [Reference]	20	1.8	10.1 (*200.0*)	Open [Reference]	View	pass
14	CondReceiverIHXCycle	pass	pass [Reference]	pass [Reference]	29	3.7	8.9 (*200.0*)	Open [Reference]	View	pass
15	InhomogeneousAirCondenser	pass	pass [Reference]	pass [Reference]	54	0.4	2.5 (*100.0*)	Open [Reference]	View	pass
16	SuperheatControl	pass	pass [Reference]	pass [Reference]	15	2.0	4.8 (*200.0*)	Open [Reference]	View	pass
17	TwinEvaporatorCycle	pass	pass [Reference]	pass [Reference]	29	3.8	10.0 (*180.0*)	Open [Reference]	View	pass
18	SimulinkInterface	pass	pass [Reference]	pass [Reference]	13	0.9	2.4 (*100.0*)	Open [Reference]	View	pass
19	InhomogeneousCSVAirSources	pass	pass [Reference]	pass [Reference]	15	0.5	2.1 (*100.0*)	Open [Reference]	View	pass

V&V – AC PULL-DOWN TEST FROM ACL

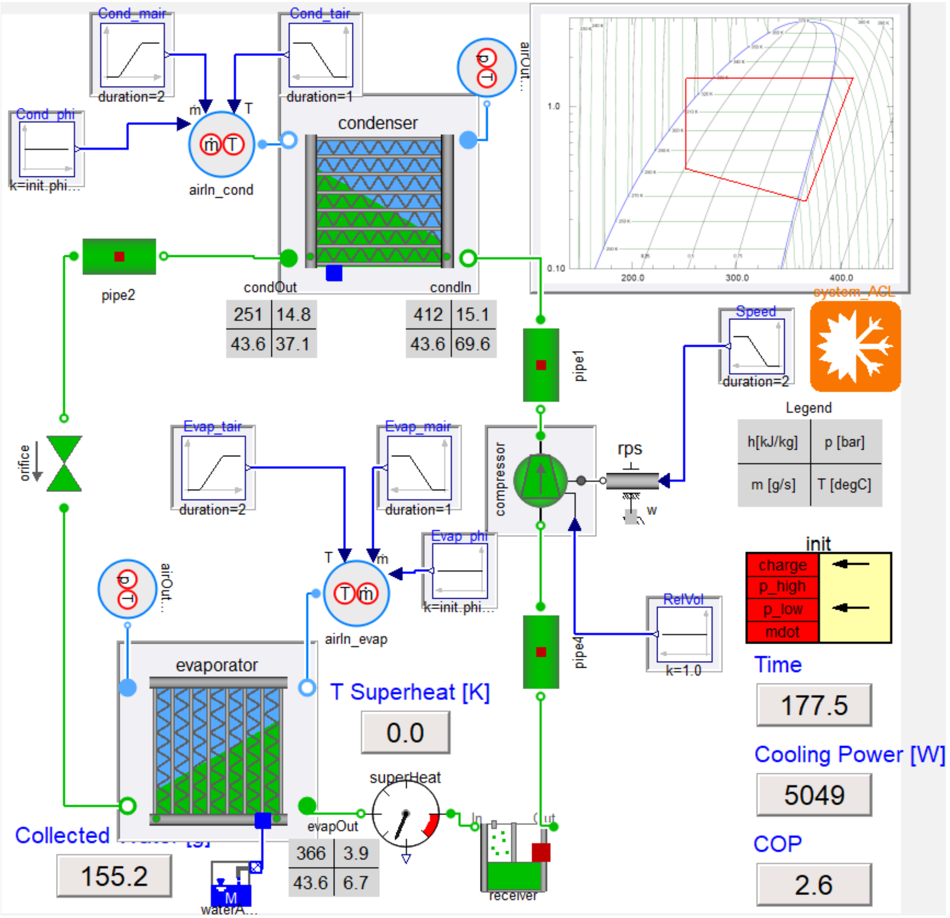


Key results	Deviation in %
Cooling Power	< 0.4
Refrigerant mass flow rate	< 0.2
Cabin temperature	< 0.002

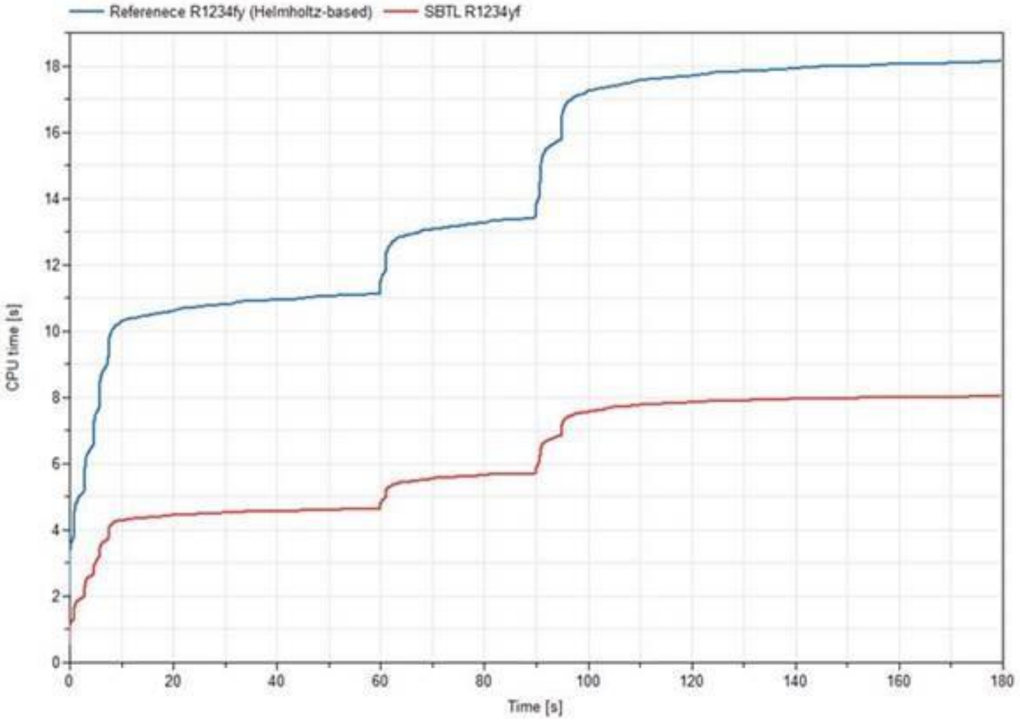




V&V – R1234YF ORIFICE CYCLE FROM ACL

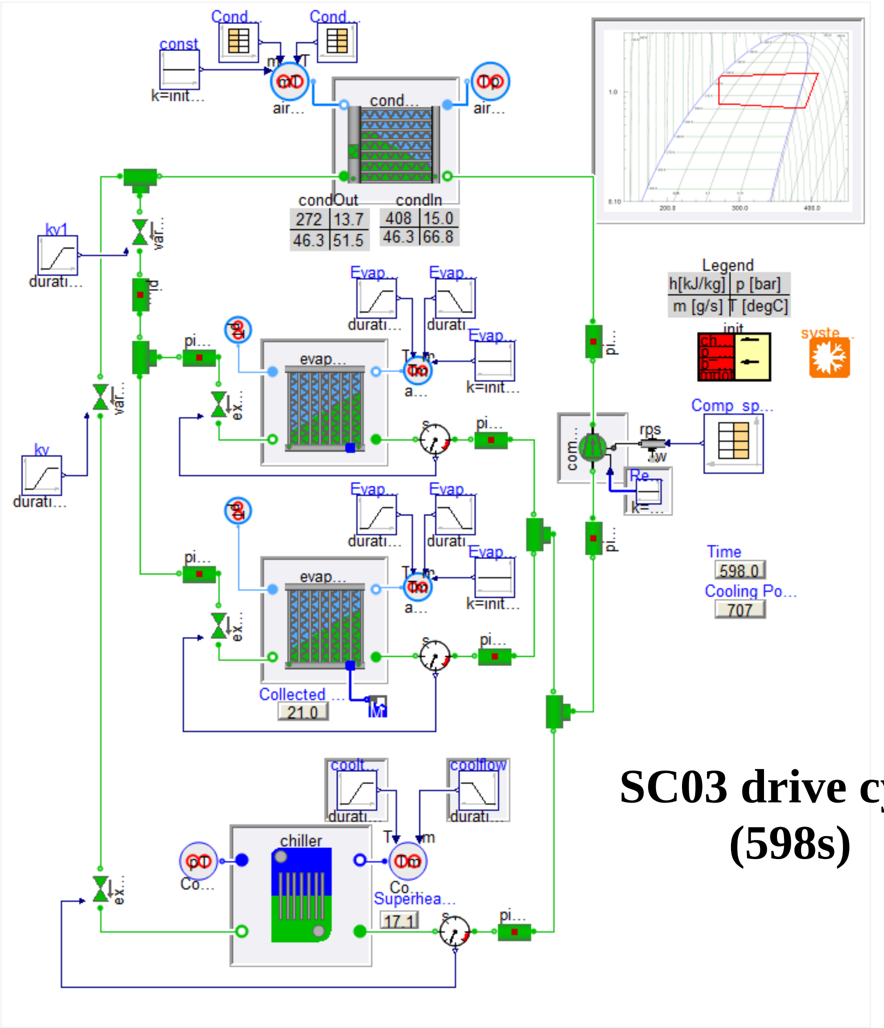


Key results	Deviation in %
Cooling Power	< 0.1
Refrigerant mass flow rate	< 0.02
Evap air outlet temperature	< 0.001

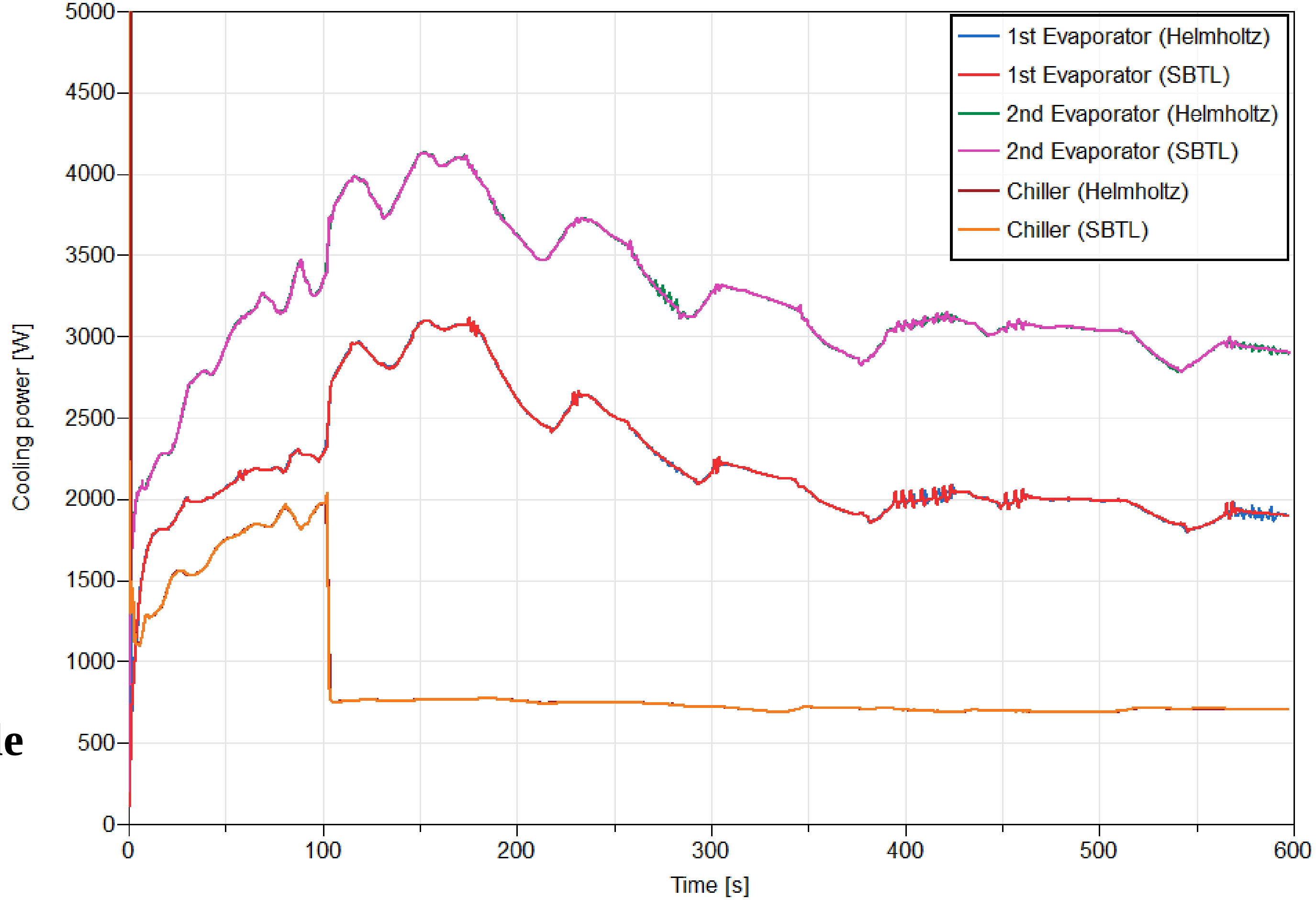


Modelon

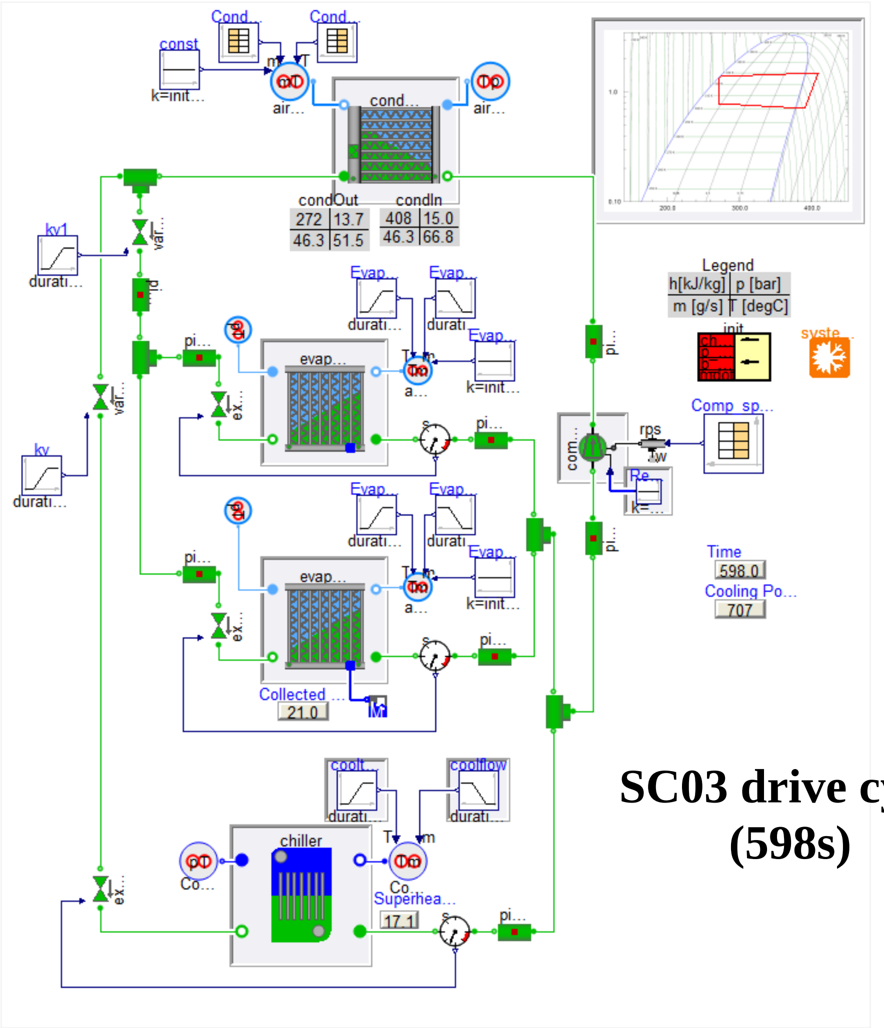
V&V – R1234YF CYCLE FROM FORD MOTOR COMPANY



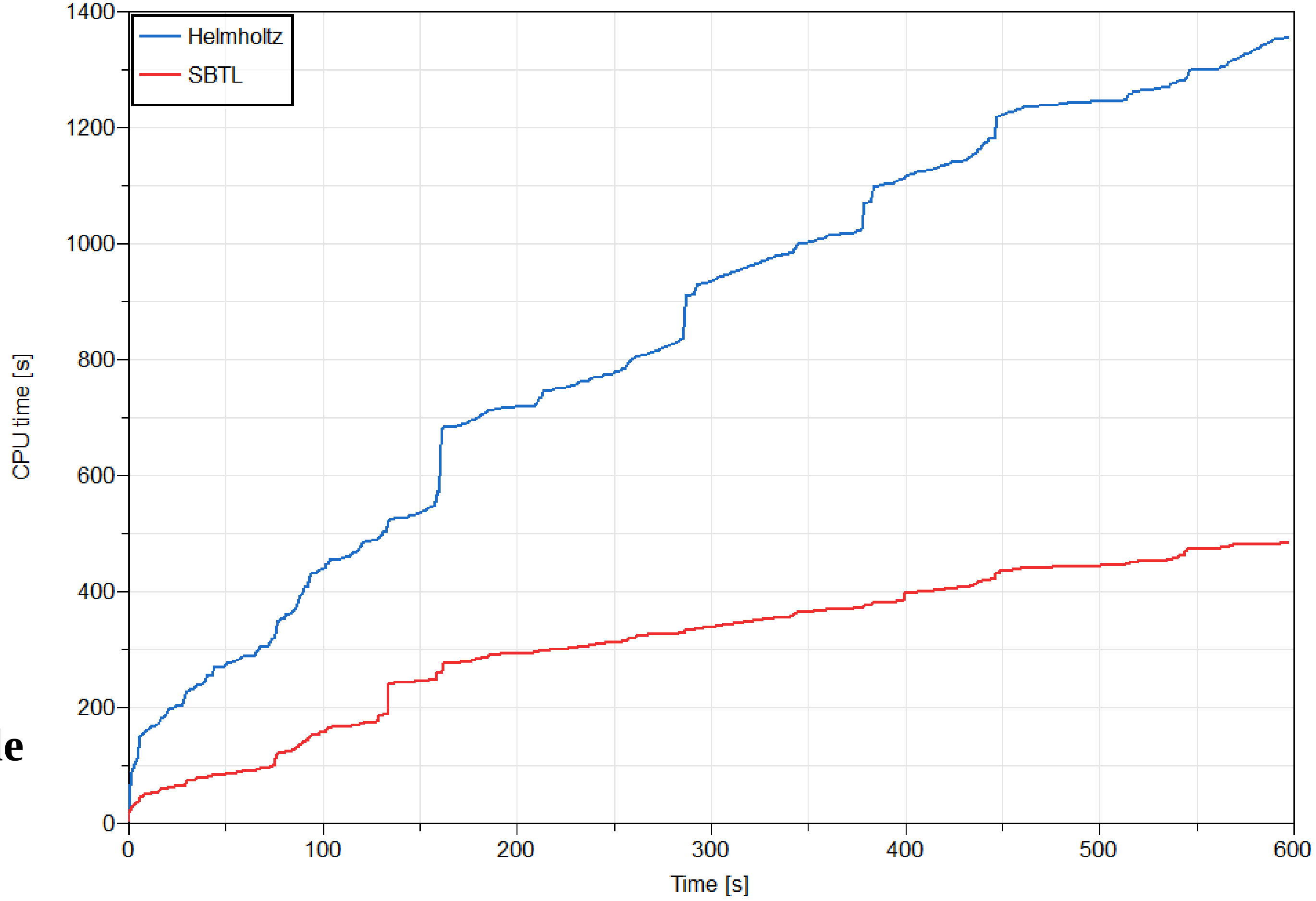
SC03 drive cycle (598s)



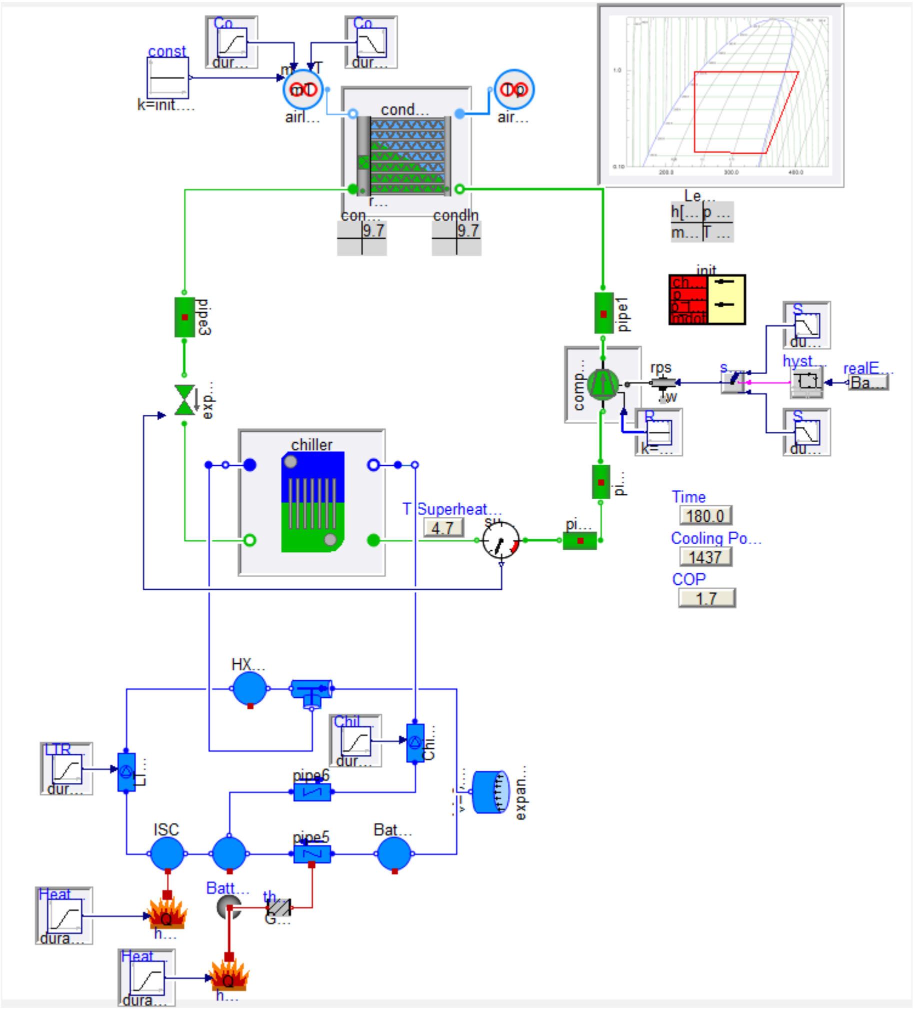
V&V – R1234YF CYCLE FROM FORD MOTOR COMPANY

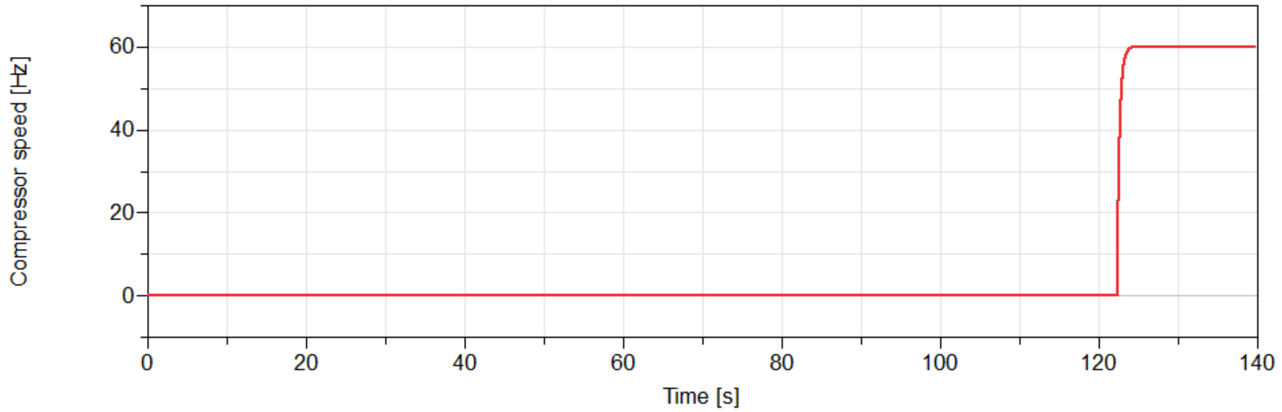


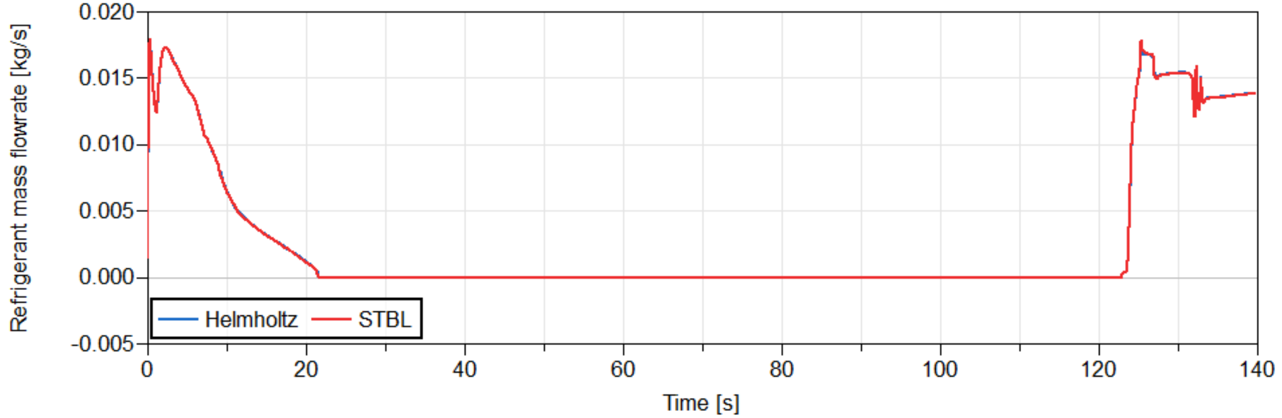
SC03 drive cycle (598s)



V&V – R1234YF CYCLE FROM FORD MOTOR COMPANY

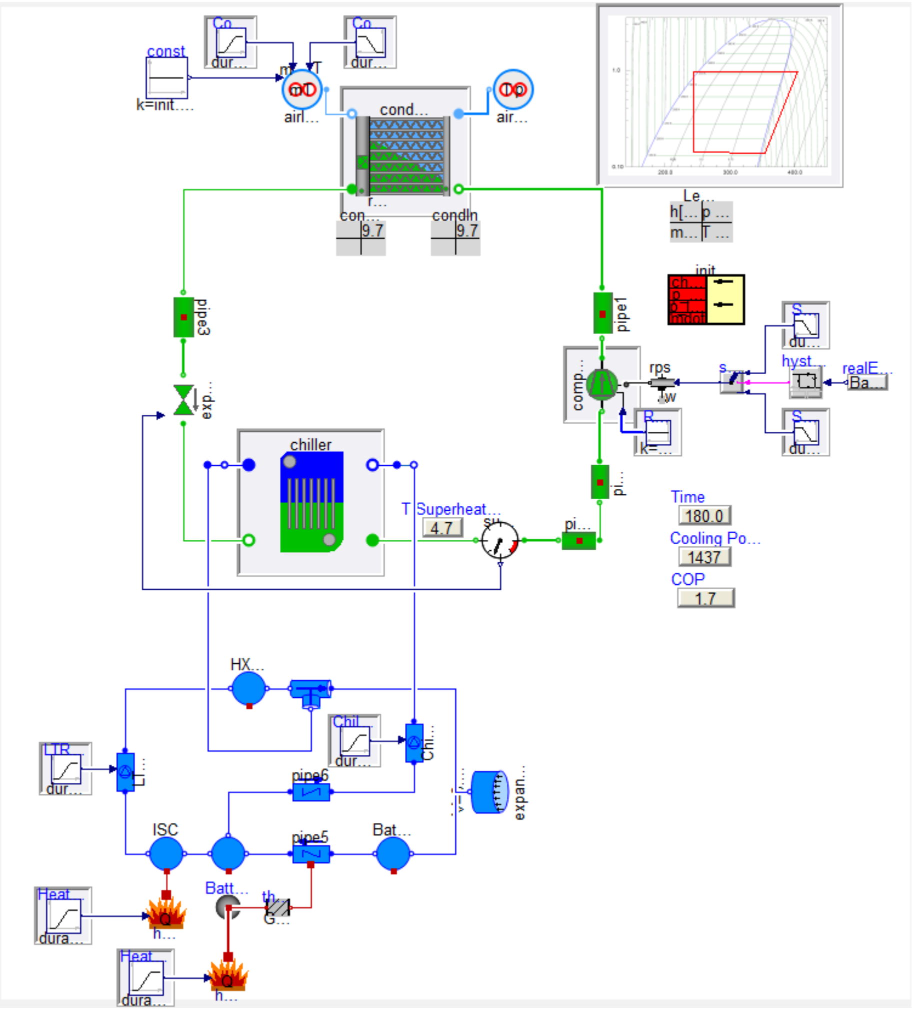


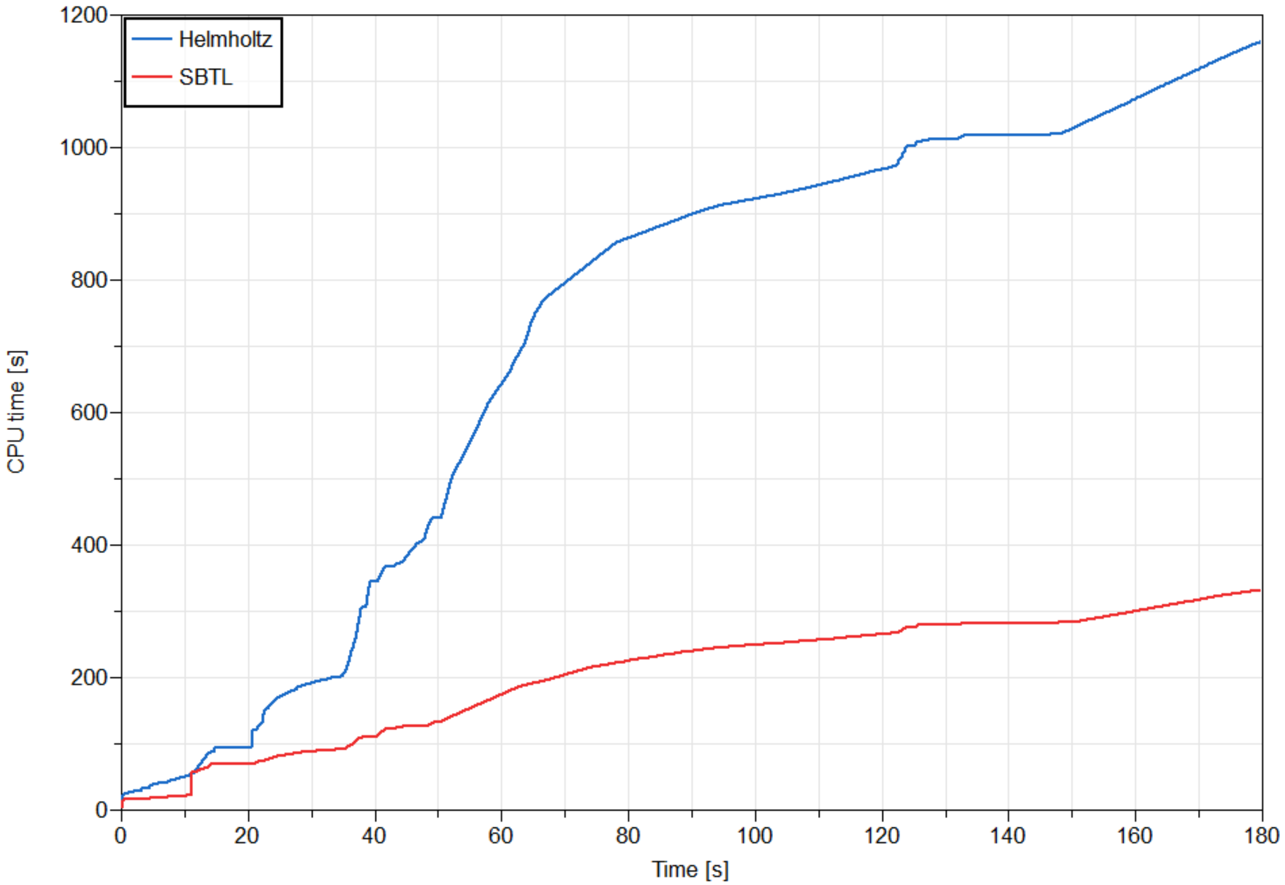




Modelon

V&V – R1234YF CYCLE FROM FORD MOTOR COMPANY





CONCLUSIONS

- An implementation of the SBTL method or fast calculation of refrigerant properties using Modelica language
- Significant improvement in computational speed in single function calls.
- In system simulations of AC cycle with ***R134a***, the SBTL model ***cut the CPU time by 33%*** compared to the short Helmholtz model.
- The complex AC system models from Ford Motor Company run ***more than twice the speed with SBTL model of R1234yf*** than with reference Helmholtz model.

The SBTL models for R134a and R1234yf will be available in the upcoming 2018.2 release (version 1.17) of Modelon's Air Conditioning Library!!!

REFERENCES

[1] Kunick, M., and H. J. Kretzschmar. Guideline on the fast calculation of steam and water properties with the spline-based table look-up method (SBTL). *Technical report, The International Association for the Properties of Water and Steam*, Moscow, Russia, 2015.

[2] Späth, H. One dimensional spline interpolation algorithms. AK Peters/CRC Press, 1995.

[3] Späth, H. *Two dimensional spline interpolation algorithms*. AK Peters, Ltd., 1995.

[4] Tummescheit, H. Design and implementation of object-oriented model libraries using Modelica. *PhD Theses*, 2002

[5] Thorade, M. and Saadat, A Partial derivatives of thermodynamic state properties for dynamic simulation. *Environmental earth sciences*, 70(8), pp.3497-3503, 2013.