Fast Simulations of Air Conditioning Systems Using Spline-Based Table Look-Up Method (SBTL) with Analytic Jacobians

Lixiang Li[1] Jesse Gohl[1] John Batteh[1] Christopher Greiner[2] Kai Wang[2]

[1]Modelon Inc, USA, {lixiang.li, jesse.gohl, john.batteh}@modelon.com
[2]Ford Motor Company, USA, {cgreiner, kwang37}@ford.com

Abstract

Refrigerant property calculation has a significant impact on the computational performance of vapor compression cycle simulations. In a previous publication, the authors have described the Modelica implementation of a Spline-Based Table Look-Up Method (SBTL) for fast calculation of refrigerant properties. This implementation demonstrated significant improvement in computational speed for a range of complex air conditioning system models. This paper describes further development of the SBTL method to allow the generation of analytic Jacobians. The new implementation with analytic Jacobian capability is tested on a range of air conditioning system models and demonstrates significant further improvement of computational speed when compared to the original SBTL model.

Keywords: Refrigerant Properties, Equation of State (EOS), Analytic Jacobians, Thermodynamic Modeling, Vapor Compression Cycle, Air Conditioning, Spline Interpolation, Computational Performance

1 Introduction

Dynamic simulations of vapor compression cycles typically require significant numbers of function evaluations to calculate properties of the working fluid. There are a number of different approaches for working fluid calculations. These calculations are typically performed using reference Helmholtz energy (multi-parameter) equation of state (EOS) (Tillner-Roth et al, 1994; Richter et al, 2011) to achieve high accuracy. Short formulation (Span et al, 2003) of Helmholtz energy EOS improves the computational performance, but it does not cover all popular refrigerants, in particular the commonly used R1234yf. In Modelon's Air Conditioning Library (Modelon, 2019), both the reference Helmholtz EOS and short Helmholtz EOS are implemented for a wide range of refrigerants.

While typically accurate, these Helmholtz approaches have a complicated multi-parameter functional form that is very computationally expensive. In addition, Helmholtz energy EOS uses density and temperature as thermodynamic states, but system models typically use pressure and enthalpy as dynamic states. Thus, internal iteration is needed when property calculations are needed in vapor compression cycle system simulations.

To address performance requirements for complex vapor cycle system simulations, a new property model based on the Spline-Based Table Look-Up Method (Kunick et al, 2015) was implemented in Modelica (Li et al., 2018). This implementation uses external C functions for fast spline evaluations and inversion. A full overview of the SBTL method and implementation is provided in the previous publication (Li et al., 2018) from the authors and thus not repeated here. The SBTL property model was validated against the existing highly accurate Helmholtz energy EOS models. Accuracy and computational performance were tested starting from single function calls and then increasing in system complexity to heat exchanger tests and then full system models of vapor compression cycles using Modelon's Air Conditioning Library and the associated library regression testing suite. The SBTL property model was also tested with a series of system models from Ford Motor Company including drive cycle simulations, air conditioning pulldown tests with air loop, and a shutdown-startup test. Testing demonstrated significant improvements in computational speed without sacrificing accuracy over the range of tests conducted. Complex system models demonstrated 2x speedups using the new SBTL property model as compared to those using the Helmholtz energy EOS. The SBTL property model and implementations for R134a and R1234yf were added to the 2018.2 release (version 1.17) of Modelon's Air Conditioning Library.

While the SBTL method has demonstrated significant improvements in computational efficiency for vapor cycle models, there are additional opportunities for efficiency improvement. In particular, the SBTL property model as initially implemented in Modelica and used with Dymola (Dassault Systemes, 2019) results in numerical Jacobians when used in system simulations. The computational benefits of analytic Jacobians with Modelica models has been well-documented (Braun et al., 2011; Jorissen et al., 2015). In particular, Modelon has experienced

speedups of up to 100x in very complex thermal power models after the development of a high precision water property model in Thermal Power Library (Modelon, 2019) that is capable of generating analytic Jacobians with Dymola.

This paper describes the extension of the SBTL method to allow the generation of analytic Jacobians. A short overview of the use of analytic Jacobians in system simulation is provided in Section 2. A brief overview of the SBTL method and implementation is provided in Section 3. Section 4 provides an overview of the development and implementation effort to enhance the SBTL method to allow analytic Jacobian generation. Section 5 shows a series of different models that were tested to validate the accuracy and computational efficiency improvement of the SBTL method with analytic Jacobian capability.

2 Analytic Jacobians in System Simulation

One key advantage of Modelica modeling is that the equations are typically accessible to the Modelica compiler. Depending on the sophistication of the Modelica compiler, it is often possible for the compiler to utilize the differential-algebraic equations to provide additional information required for the numerical solver. In particular, there are two critical places where the equations and their derivatives can be leveraged to improve the computational efficiency of the numerical solver:

- Nonlinear equation solution using Newton-Raphson techniques
- Numerical integration scheme

In both these cases, the Jacobian of the system of equations typically needs to be calculated.

While many Modelica compilers apply automatic differentiation to construct Jacobians, there are many ways in which automatic differentiation can fail to generate an analytic Jacobian. Common issues include external code without derivatives provided via the appropriate annotations, functions without appropriate information for the tool to perform automatic differentiation at the level required to generate an analytic Jacobian, or even equations and equation structure which make it difficult for the tool to perform automatic differentiation. In cases where an analytic Jacobian cannot be generated, numerical Jacobians are calculated. While the analytic Jacobian is not an approximation but is the true symbolic derivatives of the equation, a numerical Jacobian is an approximation and typically calculated via finite differences.

2.1 Nonlinear Equation Solution

It is common for the differential-algebraic equations that result from a Modelica model to involve the solution of a nonlinear system of equations. Consider the following representation of a nonlinear system of equations:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} \tag{1}$$

Newton's method is a typical iterative technique for the solution of non-linear equation systems. The iteration scheme for Newton's method is as follows:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \boldsymbol{J}(\boldsymbol{x}_n)^{-1}\, \boldsymbol{f}(\boldsymbol{x}_n) \tag{2}$$

where the Jacobian is as follows:

$$\boldsymbol{J}(\boldsymbol{x}) = \nabla \boldsymbol{f}(\boldsymbol{x}) \tag{3}$$

Thus, it can be seen that the Jacobian is fundamental for solving nonlinear equation systems. The use of analytic Jacobians increases the accuracy of the derivatives used in the iterative solution of nonlinear equation systems, leading to a reduction in the number of iterations required for convergence and potentially also increase the robustness of the solution of the nonlinear equation system.

2.2 Numerical Integration

Many numerical integration schemes require the calculation of the Jacobian of the system of equations (Braun et al., 2011; Jorissen et al., 2015). For example, implicit integration schemes require Jacobian computation and require iterations for convergence. These iterations lead to multiple function evaluations for the calculation of the Jacobian when numerical Jacobians are used. Increased accuracy from analytic Jacobians results in fewer iterations to convergence.

The Jacobian is related to the number of states in the model. Complex models with more states lead to a larger Jacobian as the number of elements in the Jacobian matrix is equal to the square of the number of states. Thus, it is expected that complex models with a larger number of states will exhibit greater computational improvement with analytic Jacobians by elimination of the large number of finite differences that would be required for the approximate of the Jacobian numerically.

3 Spline-Based Table Look-Up Method (SBTL) Overview

In recent years, different kinds of interpolation-based methods have been explored for fast calculation of refrigerant properties (Laughman et al, 2012; Schulze, 2013; Aute et al, 2014). SBTL method is one of them with proven accuracy, performance, and robustness. A detailed description of the SBTL method can be found in the reference (Kunick et al, 2015). This section focuses on its distinct features and derivative derivation.

3.1 Key features of SBTL method

By using a specific type of quadratic/biquadratic spline (Späth, 1995), the SBTL method (Kunick et al, 2015) possesses the following distinct features:

- Equidistant grid
- Continuous first derivatives
- Analytic inverse
- Consistent phase boundary definition

These features make the method a good fit for system simulation of vapor compressor cycle, where both speed of function evaluations and consistency are key requirements. To further tailor the implementation for modeling vapor compression cycle in Modelica, the authors made several design choices in the previous work (Li et al., 2018):

- Performed an overall fit to cover the whole domain instead of fitting over several sub-divided domains, balancing accuracy and data loading time at model initialization.
- Limited the use of grid transformation, trading-off between accuracy and the cost of derivative evaluation and inversion.
- Used external C functions for the spline evaluation, inversion, and derivatives to maximize the speed.

The benefit of the above choices carries over to the implementation of analytic Jacobians with a fine balance of speed and simplicity.

3.2 Derivative of the splines with grid transformation

To illustrate the impact of grid transformation on derivative calculation, we will look at the example of 1D saturation temperature splines. Derivatives of 2D spline follow the same principle.

The SBTL method uses equidistant grid to avoid searching. To enhance the accuracy of the interpolation, particularly for fitting highly non-linear functions, grid transformation can be used on both the independent and dependent variables. The chain rule must be applied properly when calculating derivatives for the transformed variables. Take the 1D saturation temperature spline as an example:

$$\bar{p} = \log(p) \tag{4}$$

$$T_{s\{i\}} = a_{i1} + a_{i2}\left(\bar{p} - \bar{p}_i\right) + a_{i3}\left(\bar{p} - \bar{p}_i\right)^2 \tag{5}$$

where $\bar{p}_i$ is the i^{th} transformed pressure node and a_{ik} are the spline coefficients in the i^{th} interval of the spline. The derivative in the i^{th} interval is given as:

$$\frac{dT_{s\{i\}}}{dp} = \frac{dT_{s\{i\}}}{d\bar{p}}\frac{d\bar{p}}{dp} = \frac{dT_{s\{i\}}}{d\bar{p}}\frac{1}{p} \tag{6}$$

where

$$\frac{dT_{s\{i\}}}{d\bar{p}} = a_{i2} + 2a_{i3}\left(\bar{p} - \bar{p}_i\right) \tag{7}$$

The 2nd-order derivative can be written as:

$$\begin{aligned}\frac{d^2T_{s\{i\}}}{dp^2} &= \frac{d}{dp}\left(\frac{dT_{s\{i\}}}{d\bar{p}}\frac{1}{p}\right) \\ &= \left(\frac{d^2T_{s\{i\}}}{d\bar{p}^2} - \frac{dT_{s\{i\}}}{d\bar{p}}\right)\frac{1}{p^2}\end{aligned} \tag{8}$$

where

$$\frac{d^2T_{s\{i\}}}{d\bar{p}^2} = 2a_{i3} \tag{9}$$

As we can see, more advanced grid transformation can result in complicated expression for derivative function due to chain rule. Therefore, model developers need to be careful when balancing the cost of derivative calculation and model accuracy.

4 Implementation of Analytic Jacobians with SBTL

In this section, we discuss the implementation of analytic Jacobians in SBTL medium models. Our first step towards analytic Jacobians was to identify what derivative functions were needed by the compiler to construct the Jacobians. We then derived the derivative functions analytically based 1D and 2D splines. In the SBTL method, dew and bubble enthalpy functions were implemented as inverse function of the 2D spline of temperature (Kunick et al, 2015). Therefore, we had to pay close attention to consistency between function evaluation and derivative calculation when deriving derivatives of the saturation property functions. Additional derivatives might also be needed in some component models to obtain analytic Jacobians for the full system model.

4.1 Diagnostics of required derivative functions

Two advanced flags in Dymola were used to identify the derivatives required for the analytic Jacobians:

- Advanced.GenerateAnalyticJacobian
- Advanced.PrintFailureToDifferentiate

With the first flag set to true, Dymola will try to construct analytic Jacobians during compilation of the model. When Dymola fails to do so, the user can set the second flag to true to find out what derivative functions are missing.

Consider the twin evaporator cycle model (see Figure 4) from the Air Conditioning Library as an example, Using the flags above, we were able to find out that we needed to provide derivatives (with respect to time) of the following functions:

- Partial derivatives of density $\left.\frac{\partial\rho}{\partial h}\right|_p, \left.\frac{\partial\rho}{\partial p}\right|_h$
- Saturation properties: $\frac{dh_l}{dp}$, $\frac{dh_v}{dp}$, $\frac{d\rho_l}{dp}$, $\frac{d\rho_v}{dp}$
- Isentropic enthalpy $h_{isen}(p, s, X)$

- Spline functions for compressor efficiencies

In this example, not only derivatives of the medium properties are required, some specific component models (e.g. compressor) also need derivatives. We suggest model developers perform this kind of diagnostics on selected system models that can well represent the causality and complexity of their typical use cases.

4.2 Derivatives of saturation properties

While it is difficult to derive analytic derivatives for fluid property models based on Helmholtz energy EOS, it is relatively straight forward to do so for spline-based models, especially in the single-phase region. SBTL method, despite its advantage in speed and consistency, does require additional work when it comes to derivative functions of saturation properties. First, grid transformation needs to be taken care of as discussed in Section 3.2. More importantly, we need to keep the derivatives consistent with the spline fit, i.e. all derivatives are obtained from the 1D and 2D spline and no extra information or fit shall be used.

Dew and bubble enthalpy functions are implemented as inverse of the 2D spline for temperature in SBTL method. They can be written as ${h_l}^{inv}(p, T_s(p))$ and ${h_v}^{inv}(p, T_s(p))$. This feature ensures consistent definition of the phase boundary, but it is more complicated to get derivatives of saturation properties in SBTL method than an implementation of h_l and h_v as 1D splines of pressure. As summarized in (Thorade and Saadat, 2013), we can write the first-order derivatives (with respect to pressure) as

$$\frac{\mathrm{d}h}{\mathrm{d}p} = \left.\frac{\partial h}{\partial p}\right|_T + \left.\frac{\partial h}{\partial T}\right|_p \frac{\mathrm{d}T_s}{\mathrm{d}p} \tag{10}$$

where

$$\left.\frac{\partial h}{\partial p}\right|_T = -\frac{\partial T/\partial p|_h}{\partial T/\partial h|_p} \tag{11}$$

$$\left.\frac{\partial h}{\partial T}\right|_p = \frac{1}{\partial T/\partial h|_p} \tag{12}$$

Similarly, we can get

$$\frac{\mathrm{d}\rho}{\mathrm{d}p} = \left.\frac{\partial \rho}{\partial p}\right|_T + \left.\frac{\partial \rho}{\partial T}\right|_p \frac{\mathrm{d}T_s}{\mathrm{d}p} \tag{13}$$

where

$$\left.\frac{\partial \rho}{\partial p}\right|_T = \left.\frac{\partial \rho}{\partial p}\right|_h + \left.\frac{\partial \rho}{\partial h}\right|_p \left.\frac{\partial h}{\partial p}\right|_T \tag{14}$$

$$\left.\frac{\partial \rho}{\partial T}\right|_p = \frac{\partial \rho/\partial h|_p}{\partial T/\partial h|_p} \tag{15}$$

The partial derivatives $\left.\frac{\partial T}{\partial p}\right|_h$, $\left.\frac{\partial T}{\partial h}\right|_p$, $\left.\frac{\partial \rho}{\partial p}\right|_h$, $\left.\frac{\partial \rho}{\partial h}\right|_p$ can be evaluated as derivatives of 2D splines $T(p, h)$ and $\rho(p, h)$ along the dew and bubble lines.

Second-order derivatives of saturation properties are also needed when calculating the derivative of $\left.\frac{\partial \rho}{\partial h}\right|_p$ and $\left.\frac{\partial \rho}{\partial p}\right|_h$ in the two-phase region. Derivation details are outlined in a Jupyter Notebook on GitHub of CoolProp, hence not to repeat in this paper.

5 System simulations using SBTL method with analytic Jacobians

In this section, we demonstrate the speedup of system simulations using SBTL method with analytic Jacobians by comparing with those using Helmholtz energy EOS or SBTL method without analytic Jacobians. We first look at a simple heat exchanger test and a system model of vapor compression cycle in the Air Conditioning Library. We then move on to a complex three-branch air conditioning (AC) system and an AC system coupled with a cooling loop, both developed at the Ford Motor Company, to evaluate their computational performance in drive cycle simulations. Lastly, we study the scalability of the SBTL method with analytic Jacobians by using higher discretization for the heat exchangers in the three-branch model. All the simulations are performed with the same computer configuration shown in Table 1.

Table 1. Configuration of the computer used for testing

Model	Dell XPS 8700 Desktop
Processor	Intel® Core™ i7-4770 CPU
RAM	16.0 GB
System	64-bit, x64 based, Windows 10 Pro
Software	Dymola 2018 FD01
C compiler	Visual Studio 2015 Express Edition
Solver	Dassl
Tolerance	1e-6 (to ensure mass conservation)

5.1 Comparison of heat exchanger test results and performance

Heat exchangers modeled by finite volume method usually have the most property function calls in system simulations of vapor compressor cycles. Therefore, a heat exchanger test is representative of how well the property model is going to perform in full system simulations. We used an evaporator test bench from the Air Conditioning Library, shown in Figure 1, to test the SBTL property model with analytic Jacobians for R134a. The two-layer evaporator had 18 discretized volumes and 56 dynamic states (pressure and enthalpy in each refrigerant volume and temperature in each wall element). The test was run for 20 s with constant boundary conditions except for the refrigerant mass flow rate which ramps from 0.028 to 0.038 kg/s during 5s to 7s.

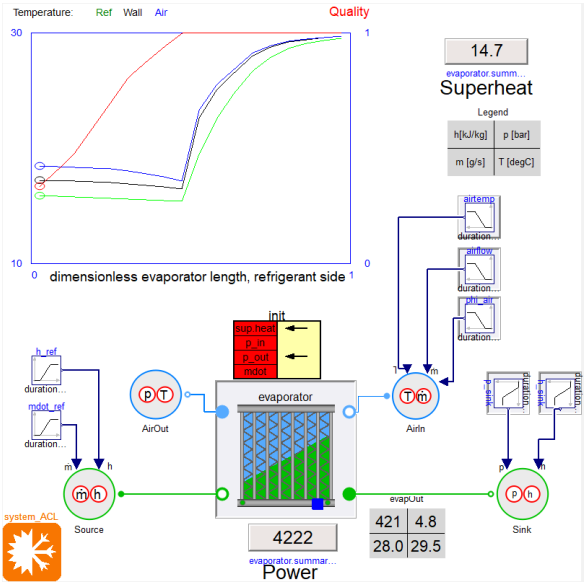

Figure 1. Evaporator test bench in Modelon's Air Conditioning Library.

Trajectories of evaporator cooling power and air outlet temperature matched very well between the SBTL model with analytic Jacobian and its data source, the short formulation of Helmholtz energy EOS, as shown in Figure 2. The maximum deviation in cooling power is less than 0.03%. The computation performance of different property models is vastly different. CPU time of the three cases can be found in Figure 3. The use of analytic Jacobians cuts the time by 17% compared with the original SBTL model. The heat exchange tests provide a preview of the great potential of using the SBTL method with analytic Jacobians to speed up system simulations.

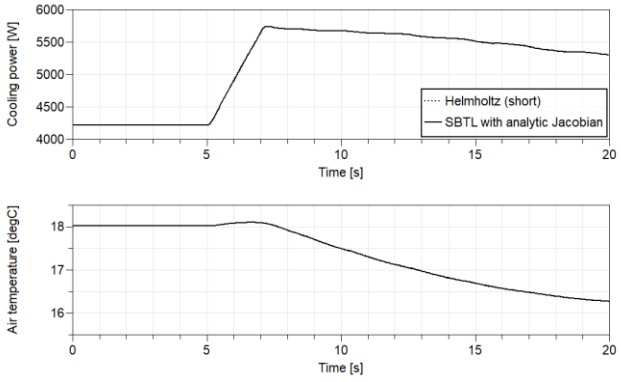

Figure 2. Comparison of cooling power and air outlet temperature.

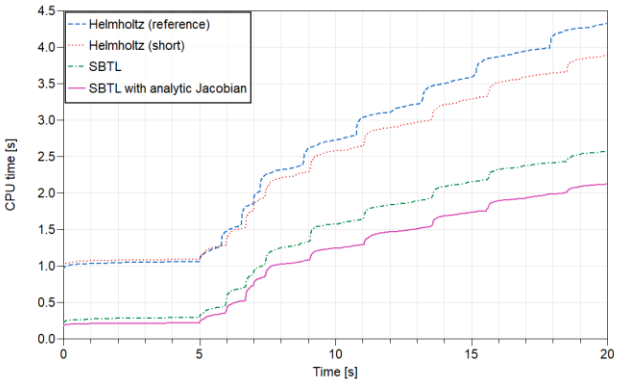

Figure 3. Comparison of CPU time running the evaporator test bench.

5.2 Comparison of system models in the Air Conditioning Library results and performance

The twin evaporator model from the Air Conditioning Library (Figure 4) was simulated for 180s to further evaluate the speedup of a complex system model brought by analytic Jacobians. The model has 131 dynamic states and has a 10°C ramp in the incoming air temperature at the condenser after 90 s.

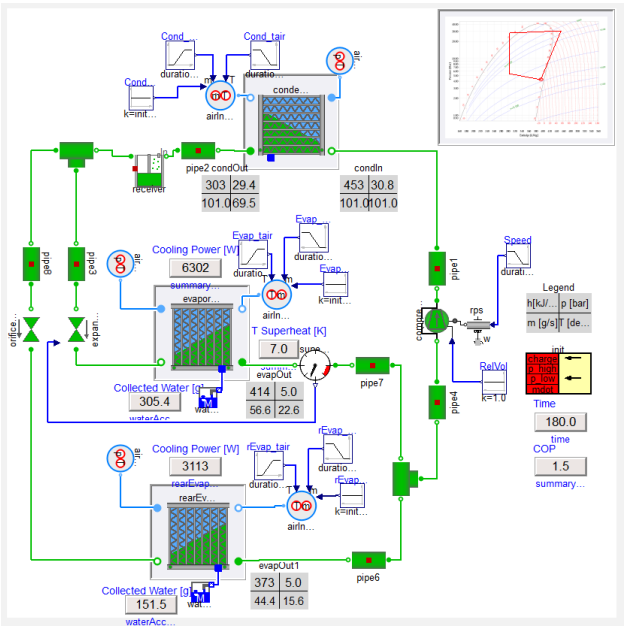

Figure 4. Twin evaporator cycle model in Modelon's Air Conditioning Library.

As shown in Figure 5, simulation with analytic Jacobians ran at 1.7x the speed of the one without them. The speedup can mostly be explained by a decrease in the number of function evaluations from 2987 to 1000 when analytic Jacobians were used. This agrees with findings in (Jorissen et al., 2015).

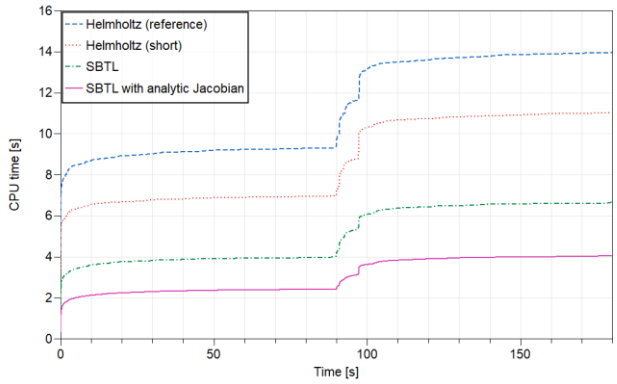

Figure 5. CPU time comparison of the twin evaporator cycle simulations. From top to bottom: Blue – Reference Helmholtz EOS, Red – Short formulation of Helmholtz EOS, Green – SBTL, Magenta – SBTL with analytic Jacobians.

We then moved on to a pull-down test example with 91 dynamic states in the Air Conditioning Library, as depicted in Figure 6. A comparison of the CPU time can is shown in Figure 7. The simulation ran 2x faster with analytic Jacobians than without.

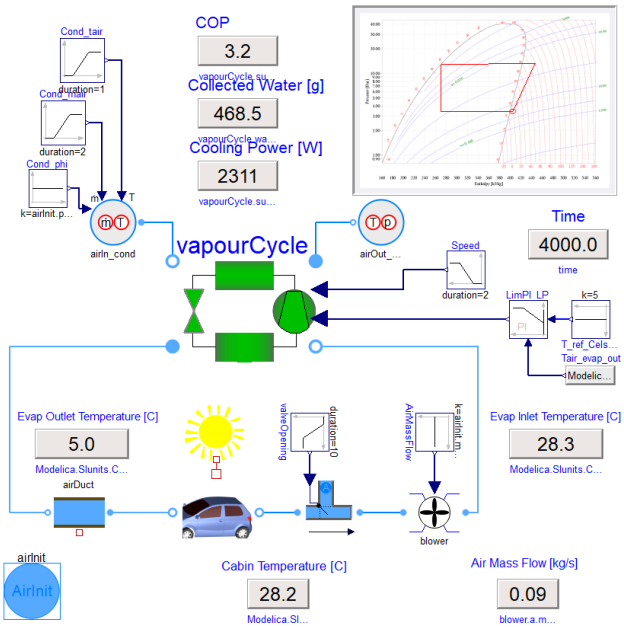

Figure 6. Pull-down test example in Modelon's Air Conditioning Library.

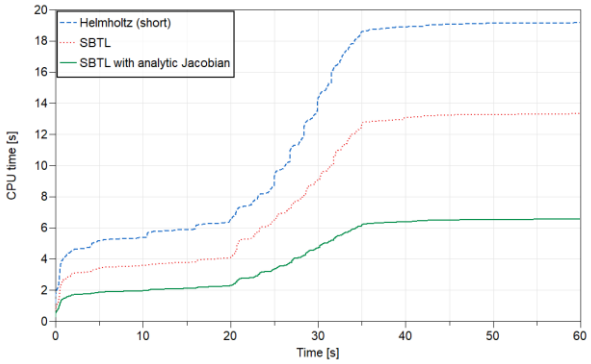

Figure 7. CPU time comparison of the pull-down simulations.

5.3 Comparison of Ford air conditioning system models results and performance

In this section, we show the results from two complex automotive air conditioning system models, developed at Ford Motor Company, to illustrate the speedup by the SBTL method with analytic Jacobians in drive cycle simulations. There are three distinct features in these simulations compared with the ones in previous sections:

- The working fluid in both systems is R1234yf. Short formulation of Helmholtz energy EOS was not available for it and reference state Helmholtz EOS had to be used. The original SBTL method offered an even greater speed-up in these cases (Li et al., 2018).
- Both models run into low flow conditions in the chiller, which was known to challenge the performance and robustness of the model.
- Tabulated boundary conditions or compressor shut down are imposed on the model, inducing more fast dynamics.

By running these simulations, we can get a better understanding of the performance and robustness of the SBTL method with analytic Jacobians in complex system models.

Figure 8 depicts a three-branch air conditioning system. Two evaporators and one chiller are connected in parallel, and all of them are superheat controlled by the thermal expansion valves (TXVs). The condenser model consists of a condenser section, a subcool section, and an internal receiver. The system model has 183 dynamic states. Apart from the tabulated boundary conditions, we deliberately introduced fast dynamics by rapidly closing the valve upstream of the chiller TXV and introducing low flow rate condition in the chiller.

Drive cycle (SC03) simulations were performed for 598s using different R1234yf medium models. Figure 9 shows a good match in results. As seen in Figure 10, the simulation finished in 112.6s when using the SBTL medium with analytic Jacobians, just 19% of the real-time (598s). This is more than 4x the speed of the

original SBTL medium and 10x the speed of the reference state Helmholtz energy EOS.

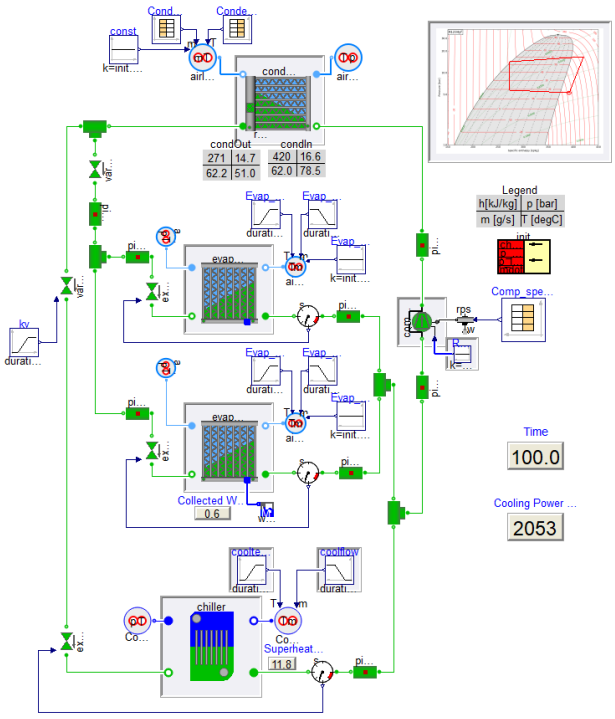

Figure 8. Air conditioning system with two evaporators and one chiller connected in parallel in the loop.

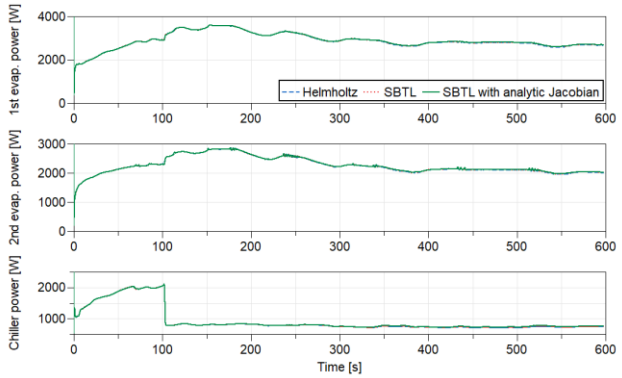

Figure 9. Cooling power of the evaporators and the chiller using different property models.

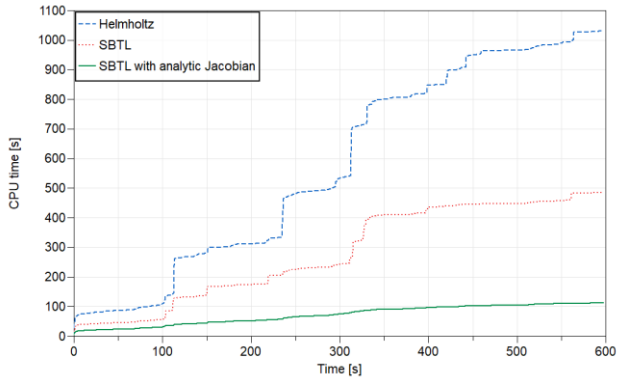

Figure 10. CPU time comparison of the three-branch air conditioning system model.

Figure 11 is a diagram of a vapor compression cycle coupled with a battery cooling system. The refrigerant loop was initialized at certain operating conditions (defined by pressures, chiller superheat and condenser) The compressor was off from 0s, simulating cycle shutdown. The temperature of the battery kept rising afterward and hit a certain threshold. The compressor was then turned back on (at about 122s) to cool the coolant and eventually drive the battery temperature down. Fast dynamics introduced by the shutdown and startup of the compressor posed serious challenges to the performance and robustness of the model.

Compressor speed and refrigerant mass flow rate are plotted in Figure 12. Simulations using different R1234yf medium models predicted the same mass flow trajectory during the fast transients. As shown in Figure 13, the SBTL medium with analytic Jacobians offered significant speedup (14x) compared to the original SBTL medium. If we focus on the zero-flow period from 22s to 122s, it only took 1.34s of CPU time using analytic Jacobians, a 125x speedup compared to the simulation using the SBTL medium without analytic Jacobians.

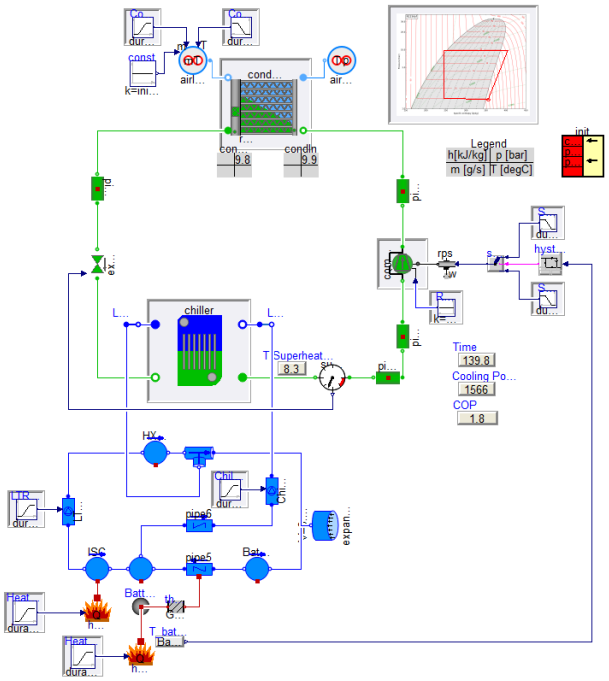

Figure 11. R1234yf vapor compression cycle with a chiller connected to a battery cooling loop.

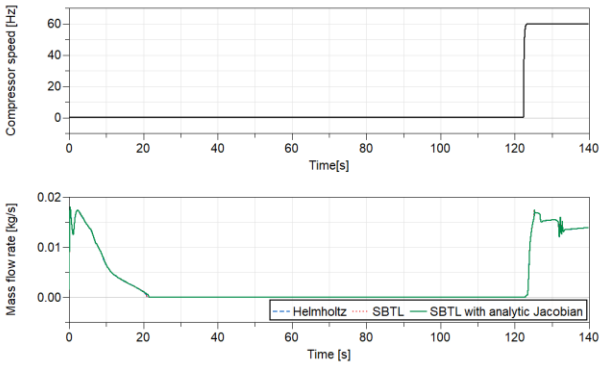

Figure 12. Compressor speed and refrigerant mass flow rate.

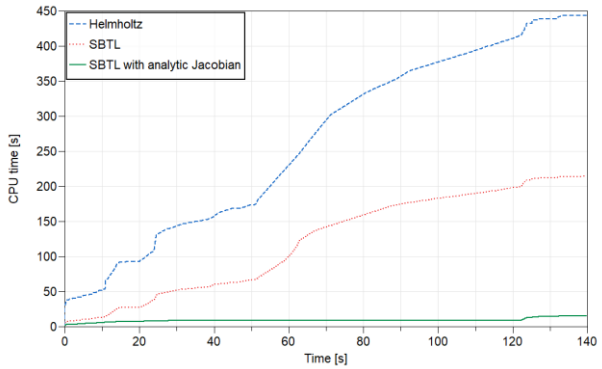

Figure 13. CPU time comparison for the model shown in Figure 11.

5.4 Scalability of SBTL method with analytic Jacobians

To better understand how the performance of the SBTL method with analytic Jacobians scales with the number of states, we simulated the three-branch air conditioning system, shown in Figure 8, with higher discretization on the refrigerant side of the heat exchanger. Non-sparse DASSL solver was used for the study. Simulation setup and CPU time are summarized in Table 2. As the total number of states increases, the CPU time grows as $O(N^{1.6})$.

Table 2. Setup and CPU time of 598s drive cycle simulations of the three-branch air conditioning system with different discretizations.

Number of finite volumes			Total number of states	CPU time [s]
Condenser	Each evaporator	Chiller		
12	12	5	183	112.65
24	24	10	311	327.01
36	36	15	439	472.23
48	48	20	567	730.11
60	60	25	695	1014.87

6 Conclusions

This paper summarizes the further development of the SBTL method for fast calculation of refrigerant properties in Modelica to allow the generation of analytic Jacobians. By expanding the SBTL implementation to provide the additional derivatives needed for analytic Jacobians, Dymola can generate analytic Jacobians for complex vapor cycle system models with Air Conditioning Library. The new SBTL model capable of analytic Jacobians was tested on a suite of models ranging from a heat exchanger bench test to full closed vapor cycle loop systems, including several complex system models from Ford Motor Company.

The computational improvements from the SBTL models with analytic Jacobians are significant. While the original implementation of the SBTL method provides improvements in computational speed on the order of 33% - 50% for complex system models when compared to the baseline Helmholtz property models, the computational improvements from SBTL with analytic Jacobians are even greater when compared to the original SBTL implementation without analytic Jacobians. Reductions in computational speed ranging from to 2-4x are obtained on complex system models for SBTL with analytic Jacobians versus SBTL without analytic Jacobians. Even greater improvement was observed in models under low or zero flow conditions. In addition, the computational improvement could be even larger for more complex models with even more states.

The new SBTL models with analytic Jacobians capability will be available in an upcoming release of Modelon's Air Conditioning Library.

References

Tillner-Roth, R. and Baehr, H.D. An international standard formulation of the thermodynamic properties of 1,1,1,2-tetrafluoroethane (HFC-134a) for temperatures from 170 K to 455 K at pressures up to 70 MPa. *Journal of Physical and Chemical Reference Data*, 23(5), 657-729, 1994. DOI: https://doi.org/10.1063/1.555958

Span, R. and Wagner, W. Equations of state for technical applications. I. Simultaneously optimized functional forms for nonpolar and polar fluids. *International Journal of Thermophysics*, 24(1), 1-39, 2003. DOI: https://doi.org/10.1023/A:1022390430888

Modelon AB, *Air Conditioning Library*, https://www.modelon.com/library/air-conditioning-library/, Lund, Sweden, 2019.

Richter, M., McLinden, M.O., and Lemmon, E. W. Thermodynamic Properties of 2, 3, 3, 3-Tetrafluoroprop-1-ene (R1234yf): Vapor Pressure and p–ρ–T Measurements and an Equation of State. *Journal of Chemical & Engineering Data*, 56(7), 3254-3264, 2011. DOI: https://doi.org/10.1021/je200369m

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. URL: http://www.iapws.org/relguide/SBTL.html

Li, L., Gohl, J., Batteh, J., Greiner, C., and Wang, K., Fast Calculation of Refrigerant Properties in Vapor Compression Cycles Using Spline-Based Table Look-Up Method (SBTL). *Proceedings of the 1st American Modelica Conference*, p. 77-84, 2018. DOI: http://dx.doi.org/10.3384/ecp1815477

Dassault Systemes, Dymola, https://www.3ds.com/products-services/catia/products/dymola/, 2019.

Braun, W., Ochel, L., and Bachmann, B., Symbolically Derived Jacobians Using Automatic Differentiation – Enhancement of the OpenModelica Compiler. *Proceedings of the 8th International Modelica Conference*, DOI: 10.3384/ecp11063495, p. 495-501, 2011.

Jorissen, F., Wetter, M., and Helsen, L., Simulation Speed Analysis and Improvements of Modelica Models for Building Energy Simulation. *Proceedings of the 11th International Modelica Conference*, p. 59-69, 2015. DOI: http://dx.doi.org/10.3384/ecp1511859

Modelon AB, *Thermal Power Library*, https://www.modelon.com/library/thermal-power-library/, Lund, Sweden, 2019.

Laughman, C., Zhao, Y., and Nikovski, D. Fast Refrigerant Property Calculations Using Interpolation-Based Methods. *International Refrigeration and Air Conditioning Conference*. Paper 1344, 2012. URL: https://docs.lib.purdue.edu/iracc/1344/

Schulze, C. W. A contribution to numerically efficient modeling of thermodynamic systems. *PhD Thesis*, 2013. URL: http://www.digibib.tu-bs.de/?docid=00057492

Aute, V. and Radermacher, R. Standardized Polynomials for Fast Evaluation of Refrigerant Thermophysical Properties. *International Refrigeration and Air Conditioning Conference*. Paper 1499, 2014. URL: https://docs.lib.purdue.edu/iracc/1499/

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

Späth, H. Two dimensional spline interpolation algorithms. AK Peters, Ltd., 1995.

Tummescheit, H. Design and implementation of object-oriented model libraries using Modelica. *PhD Thesis*, 2002. URL: http://lup.lub.lu.se/record/20836

Thorade, M. and Saadat, A Partial derivatives of thermodynamic state properties for dynamic simulation. *Environmental earth sciences*, 70(8), pp.3497-3503, 2013. DOI: 10.1007/s12665-013-2394-z

Jupyter Notebook on the GitHub of CoolProp, URL: https://github.com/CoolProp/CoolProp/blob/master/doc/notebooks/Saturation.ipynb