2021

Quantitative and Qualitative Evaluation of Various Positive-Displacement Compressor Modeling Platforms

Mohsin Mohsin Tanveer
Building and Environmental Thermal Systems Research Group, Oklahoma State University, Stillwater, OK 74078, mtanvee@okstate.edu

Craig R. Bradshaw

Quantitative and Qualitative Evaluation of Various Positive-Displacement Compressor Modeling Platforms

M. Mohsin Tanveer[1*], Craig R. Bradshaw[1]

[1] Center for Integrated Building Systems, Oklahoma State University, Stillwater, OK 74078
Contact Information (mohsin.tanveer@okstate.edu)

* Corresponding Author

ABSTRACT

Several regulatory measures aimed to mitigate climate change are forcing compressor manufacturers to replace current refrigerants to those with low Global Warming Potential (GWP). New refrigerants need to be evaluated to ensure adequate efficiency for use in modern products. Evaluation can be done heuristically, which is expensive and time-consuming, while a carefully designed simulation model can provide similar outcomes for a significantly reduced cost. This paper presents a comparison between various user-developed and existing reciprocating compressor models to assist in the selection of a suitable modeling platform for a wide-ranging study. The reciprocating compressor is selected because of the simplicity of the model to ensure consistency across different platforms. The user-developed models are developed in MATLAB™ and Modelica™ for the reciprocating compressor. The same compressor is also modeled using existing compressor modeling platforms, PDSim and GT-Suite™. The compressor model includes three main components; geometry, compression process and frictional losses. Other sub-models, like valve model and heat transfer model, are also part of the compression process. These platforms are evaluated based on both quantitative and qualitative criteria. Modelica™ is found to be computationally efficient while GT-Suite™ took maximum time for simulation among the compared platforms. On a qualitative basis, PDSim is potentially a better platform for compressor optimization; which is also readily available to end user due to its open-source nature and prospects for future model development.

1. INTRODUCTION

A 2016 amendment to Montreal protocol directs a shift towards the low-GWP refrigerants, which will result in phasing down the use of HFC refrigerants in HVAC&R systems Consequently, HFC refrigerants will need to be replaced with low-GWP alternatives or natural refrigerants. The effect of this transition on the performance of existing compressor technology is a critical evaluation step in developing next-generation products.

The development of prolific and powerful computer technology has enabled detailed compressor models to become a more viable substitute for heuristic compressor design. The statistical correlations are the simplest among various compressor models and can only be applied within the ranges of the training data for the model. The extrapolation of these models often entails very high uncertainties and can not be used as predictive tools.

The fidelity of the models increases as they incorporate the physics involved in the compression process, and their reliability depends on the accuracy of the individual sub-models. The mechanistic chamber model extends the fidelity beyond a statistical model and is based on a control volume (or chamber). It can be implemented for various positive displacement compressors and can also be used as a predictive tool. Chen et al. (2002) suggested that a comprehensive simulation model for the scroll compressor can predict the real behavior of the compressor and can be used for compressor design and optimization purposes.

Fully distributed (or 3D) models typically manifest as CFD-based. Though CFD provides a detailed analysis of fluid behavior in the compressor, often such detail is not required for performance evaluation. Moreover, computer technology developed and readily available in the past decade but still, CFD is computationally impracticable for holistic performance optimization of compressors. The same outcomes can be accomplished in significantly reduced time and cost by utilizing properly tuned and validated 0D or 1D mechanistic model (Pereira et al., 2008).

While the physics of the model is important for reliable results, numerical solution techniques are also vital for dynamic models. Dynamic behavior might not be modeled properly, if proper solution technique is not selected, due to the stiff nature of compression process equations. The selection of suitable solver technique is important to reduce the computational time and fulfill the model requirement. Moreover, several proprietary, closed-source and open-source compressor modeling platforms are available. A qualitative comparison of these available platforms is required to explore their capabilities for compressor modeling. Additionally, simulation time is very important for complex physics-based models and a comparison of available platforms should be conducted to help in platform selection for future model development.

In summary, a mechanistic chamber model can be used to predict the compressor performance at conditions different than for which it had been validated. This makes this type of model well-suited to explore the future of compressor technologies amid the transition of working fluids. This work presents the first step in the process of model development by the selection of the modeling platform that will allow the most consistent, and rapid, development of a plurality of compressor models to support this effort. To this end, four different compressor modeling platforms are evaluated based on qualitative and quantitative parameters applied to the modeling of a reciprocating compressor. The first two platforms are user-developed mechanistic chamber models, developed in MATLAB™ (MATLAB, 2019) and Modelica™. The third platform is PDSim (Bell et al., 2020a), which is a Python-based package and utilizes the same mechanistic chamber modeling approach, developed to model positive displacement compressors. The fourth platform is GT-Suite™ (GT-Suite™, 2019) which is based on 1-D CFD. Model templates for certain compressors (i.e. reciprocating compressors, scroll compressors) are available in GT-Suite™ and has been employed for various applications by number of authors.

2. MECHANISTIC CHAMBER MODEL OF A RECIPROCATING COMPRESSOR

This section presents the physics of the model developed in all four modeling platforms (MATLAB™, Modelica™, PDSim, and GT-Suite™). This model is constructed using the mechanistic chamber model approach and is similar to the models presented by Bell et al. (2020a). This approach models the compressor by applying the mass and energy conservation to the control volume (defined inside the compression chamber) and predicting the refrigerant properties on each step of the crankshaft rotation. The general approach of the model is presented in Figure 1. The modeling process starts by defining the guess values for the initial thermodynamic state in the control volume. The volume model is used to calculate the instantaneous volume of the control volume. Then, mass and energy conservation equations are used to calculate the next thermodynamic state which requires inputs from sub-models of volume, mass flow and heat transfer. The process is assumed to be quasi-steady and thermophysical properties are considered constant for each time step. The process is repeated for one crank rotation. The compressor wall temperature is considered constant for each cycle and re-evaluated at the end of each iteration. At the end of the cycle, residuals are calculated for instantaneous temperature, density and time-averaged wall temperature. If the residuals are greater than tolerance than final states are used as an initial guess and the process will be repeated until convergence is achieved. The details of each sub-model is described in the following sections.

2.1 Compression Process Equations

The compression process is modeled by applying the mass-energy balance to a general control volume. The general mass conservation equation of a control volume can be re-cast to calculate the instantaneous density of a generic control volume, written as,

$$\frac{d\rho}{dt} = \frac{1}{V}\left[-\rho\frac{dV}{dt} + \left(\sum \dot{m}_{in} - \sum \dot{m}_{out}\right)\right]. \tag{1}$$

Similarly, by ignoring kinetic and potential energy changes, the conservation of energy can be represented as,

$$\frac{dE_{CV}}{dt} = \frac{du}{dt} = \frac{\partial u}{\partial \rho}\frac{d\rho}{dt} + \frac{\partial u}{\partial T}\frac{dT}{dt}. \tag{2}$$

These expressions can be combined and in then applying the definition of enthalpy and boundary work to generate the

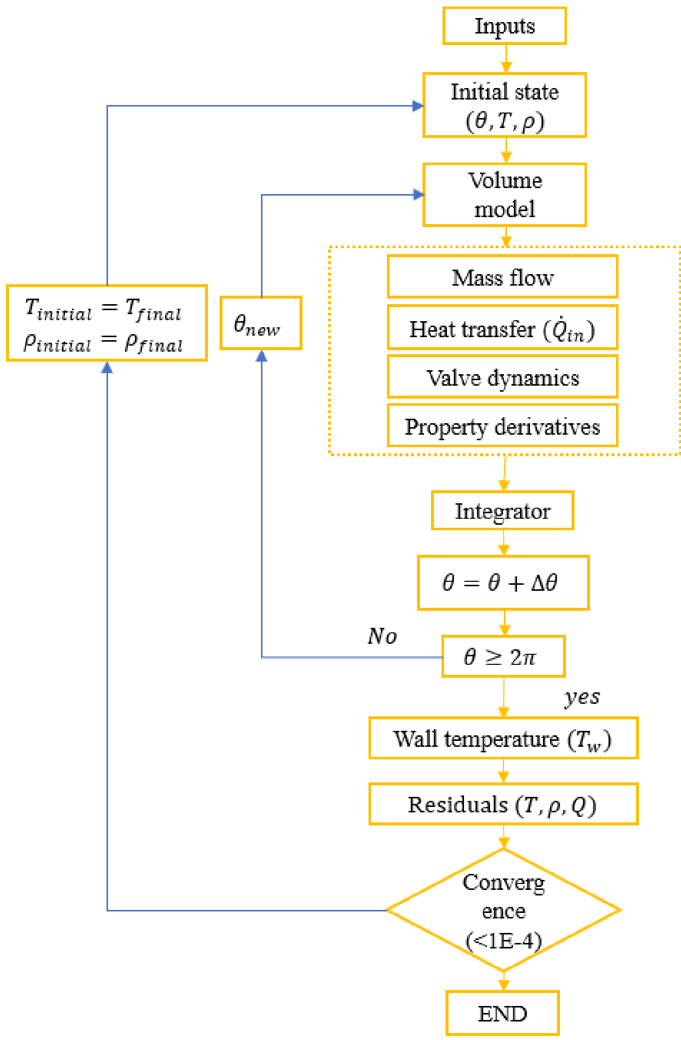


Figure 1: Flow chart for mechanistic chamber model.

final form of the energy balance equation,

$$\frac{dT}{dt} = \frac{-1}{\rho V \frac{\partial u}{\partial T}} \left[\left(\rho V \frac{\partial u}{\partial \rho} + Vu \right) \frac{d\rho}{dt} - (P + \rho u) \frac{dV}{dt} + \sum \dot{m}_{in} h_{in} - \sum \dot{m}_{out} h_{out} + \dot{Q}_{in} \right]. \tag{3}$$

This general form of the compression process equations is generic for all positive displacement types and requires, as inputs, information for the various terms in the expressions that define the specific type of compressor being analyzed. In this case, the volume, heat transfer, and mass flow models developed are specific for a reciprocating compressor.

2.2 Volume Modeling

The mass and energy balance equations require as an input, the instantaneous volume in the cylinder to define the behavior of the control volume. For a reciprocating compressor, this can be calculated by using two different approaches, either kinematic expressions or assuming a sinusoidal volume change. The model selected depends on the availability of information. As current work is focused on the capabilities of the platform, a hypothetical compressor is modeled by assuming displaced volume because detailed compressor geometry is not available. Therefore, for the current application, the second approach is more suitable that assumes a sinusoidal piston movement,

$$V = V_0 + \frac{V_{disp}}{2(1 - cos(\theta))}. \tag{4}$$

The clearance volume is assumed to be 1% of the displaced volume.

2.3 Thermo-Physical Properties

Thermophysical properties are a vital part of the model and are required in almost all the sub-models. Consequently, model accuracy is strongly dependent on the accuracy of the method used for property calculation. Moreover, these property calculations take around 90% of the total simulation time. It is important to discuss different available methods for property calculations. The thermophysical properties are calculated using pre-coded Equations of State (EOS) from various libraries. The REFPROP is used in the model developed in MATLAB™ and GT-Suite™. TIL media library is used for the model developed in Modelica ™ which is developed by TLK-Thermo GmbH and allows a user to access external database i.e. REFPROP. Being an open-source platform, PDSim utilizes an open-source library (CoolProp) for property calculation (Bell et al., 2014). Density and temperature are selected as independent properties to fix the thermodynamic state of the refrigerant at each each step.

2.4 Instantaneous Heat Transfer in Cylinder

Heat transfer from the refrigerant to the cylinder wall is calculated initially by assuming a constant cylinder wall temperature over a cycle because instantaneous cylinder wall temperature changes very little as compared to the time-averaged wall temperature over a cycle. The new wall temperature is computed at the end of the cycle which is used for the next iteration, convergence is achieved. Instantaneous heat transfer in the cylinder is calculated using Newton's law of cooling as,

$$\dot{Q}_{in} = h_{ref} A_p (T_w - T), \tag{5}$$

where convection heat transfer coefficient is calculated as given below,

$$h_{ref} = 0.053 \frac{k}{d_p} \mathrm{Pr}^{0.6} \mathrm{Re}^{0.8}. \tag{6}$$

For Reynolds number calculation, the velocity of refrigerant is assumed to be equal to the velocity of the piston. The net heat transfer between the refrigerant and the wall is then calculated for each compression cycle by taking the average of the instantaneous heat transfer over the entire cycle.

2.4.1 Ambient Heat Balance

The average cylinder heat transfer is coupled to the ambient through an ambient heat balance. The cylinder wall temperature is calculated at the end of each iteration by setting up heat transfer from the cylinder wall to the ambient as given below,

$$T_w = \frac{\dot{Q}_{amb}}{h_a A_{cyl}} + T_{amb} \tag{7}$$

The ambient heat transfer $\dot{Q}_{amb}$ is computed as,

$$\dot{Q}_{amb} = h_a A_{cyl} (T_w - T_{amb}), \tag{8}$$

and heat transfer coefficient for ambient air is calculated by following correlation

$$\mathrm{Nu} = 0.683 \mathrm{Re}^{0.466} \mathrm{Pr}^{\frac{1}{3}}, \tag{9}$$

where, Reynolds number is calculated by assuming fixed wind velocity.

2.5 Mass Flow and Valve Model

Mass flow through suction and the discharge port is calculated by assuming an isentropic compressible flow assumption and calculated using the following expression,

$$\dot{m} = \rho\sqrt{\gamma RT}A_v\sqrt{\left(\frac{P_{high}}{P_{low}}\right)^{\left(\frac{\gamma-1}{\gamma}\right)}\frac{2}{\gamma-1}}. \tag{10}$$

The model is designed to work with or without valve dynamics to observe the effect of the valve model on the stiffness of compression process equations and computational time. Valves in reciprocating compressors are typically a reed valve, which is a flat, constrained, beam that behaves like a spring when moved in response to the pressure differences. The flow area available to the mass flow model requires the valve opening, which is used to calculate of flow area. This information is collected by modeling the dynamics of the reeds assuming they behave as one degree of freedom mass-spring system. Valve dynamics is modeled using the method presented by Kim & Groll (2007).

3. NUMERICAL INTEGRATORS

Three numerical integration techniques are compared in this study using the various software platforms. This comparison will add an additional dimension to the platform comparison as some platforms allow precise control over numerical integration (*e.g.* MATLAB™, PDSim, Python) while others offer very little (*e.g.* Modelica™, GTSuite™). As a basis of comparison, this study will compare the most basic numerical techniques, Euler and Modified Euler, against more advanced solvers like the Adaptive-RK and the built-in solvers in the commercial packages. Additional information on each numerical integrator can be found in (Bell et al., 2020b).

4. MODELING PLATFORMS

Four different platforms are selected for quantitative and qualitative evaluation to help the selection of a suitable platform for future compressor performance evaluation. A user-developed program is modeled in two of these platforms. The rest of the platforms have already developed model components or libraries which are utilized for compressor modeling. A detailed description of the model developed in each platform is presented below. The code for each of the four models is also provided as an electronic annex (available at doi:10.5281/zenodo.3923415).

4.1 User Developed Code

4.1.1 MATLAB ™, Code is developed in MATLAB™ and program flow is presented in Figure 1. The model physics is described in Section 2. The sub-models are developed in the form of functions and integrated for compressor modeling. Three different integration approaches, as discussed in Section 3, are used to demonstrate their capability for compressor performance and their efficiency in terms of computational time.

4.1.2 Modelica ™, The same model is replicated in Modelica™ using available Euler and RK45 solvers. Only suction and discharge mass flow rate models are provided in the form of functions while rest of the program is structured in a single Modelica™ model. Contrary to most programming software that interprets the program as a set of instructions, Modelica™ tries to simulate the program w.r.t time so there is no required flow for programming. Besides, equations are not required to be manipulated for the calculated variable to be on the left side. The only criteria that need to be fulfilled, is the number of provided equations should be equal to the number of variables. Also, variables used in the model need to be declared at the top along with their type.

4.2 Pre-Developed Model Bases

4.2.1 PDSim, PDSim is a Python-based, object-oriented, positive-displacement compressor and expander modeling platform described in (Bell et al., 2020a) and (Ziviani et al., 2020). It utilizes the high-level features of python and incorporates low-level code using Cython for performance optimization purposes. It has some libraries, developed in an object-oriented fashion, to simplify the modeling of positive displacement compressors. It allows users to use a text-based programming approach to model the compressor which provides adequate flexibility for modification or its Graphical User Interface (GUI) can be used by simply providing the compressor parameters. Different pre-developed example compressor models are provided in the package, including rolling piston compressor, scroll compressor and reciprocating compressor.

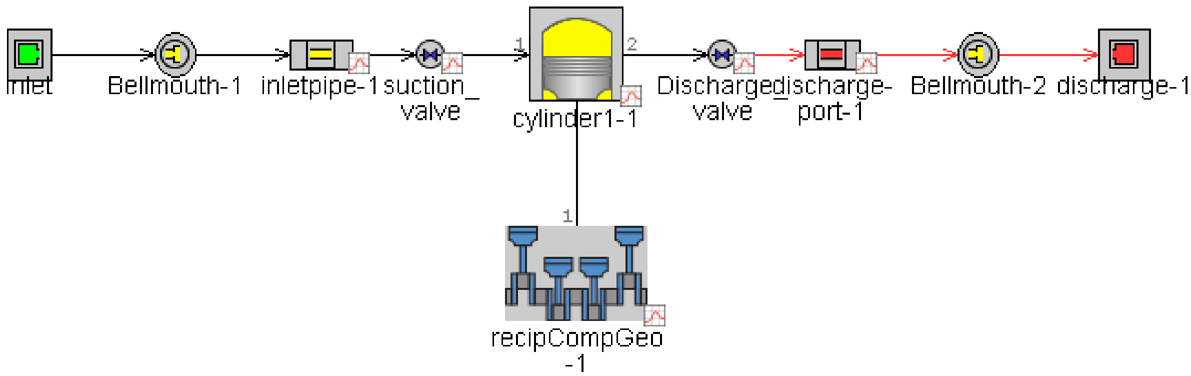


Figure 2: Schematic of compressor model in GT-Suite™.

4.2.2 GT-Suite™, GT-suite™ is a software package that utilizes a 1-D computational fluid dynamics (CFD) engine to do the dynamic analysis of a system of components. It has built-in physics-based models of components that can be connected to model a system, including compressors. A schematic of the reciprocating compressor model is shown in figure 2. The cylinder is connected to the inlet and discharge side by pressure-driven valves and ports. Cylinder volume is calculated using crank kinematics and is capable of modeling a multi-cylinder compressor. Detailed list of parameters used for setting up the compressor model in GT-Suite is presented in (Tanveer & Bradshaw, 2020) and can also found in the electronic annex.

5. RESULTS AND QUANTITATIVE COMPARISON

5.1 Model Prediction Comparison

The compressor with the same dimensions and parameters is modeled across four platforms to compare the simulation time. Table 1 and Figure 3 shows that models developed across platforms are similar enough to be considered identical for this study. The calculated mass flow rate, power and volumetric efficiency are reasonably close to each other for four developed models. The pressure-volume curve indicates that the results for all four platforms present very similar thermodynamic behavior as well. The difference in the indicator curve of GT-Suite™ can be explained by the dynamics of the valve component. The valve component in GT-Suite™ is very detailed which considers the non-linear behavior of the valve (i.e. damping etc.) while the rest of the models use a 1-D spring-mass-damper model. This difference in the valve modeling approach results in mass flow fluctuations on the discharge side. Additionally, the Modelica™ model does not have a valve sub-model because Modelica™ requires variables to be a continuous function of time. In the valve model, the integration of maximum valve lift in terms of stopper height makes it a piece-wise function. The valve model can be integrated by using the "$Algorithm$" feature in Modelica™ but it will limit the use of Modelica's other features. Although there are minor differences in modeling approaches across different platforms, overall the compressor outputs are identical as evident from Table 1. Small differences are found in the indicated power which is caused by slight differences in valve behavior and discharge temperature. For MATLAB™, it is also observed that reducing the solver tolerance gives better agreement with PDSim and GT-Suite™ results. The results, presented in Table 1, are obtained at same solver tolerance to make a comparison between platforms at consistent settings. Similarly, as mentioned earlier, the model in Modelica™ does not account for valve losses and is slightly under-predicting the indicated power as a result. Despite the small differences in predicted outputs, these results are considered sufficiently close to compare the qualitative and quantitative metrics associated with the model development process.

Table 1: Compressor performance parameters calculated from four different platforms

parameter	MATLAB™	PDSim	Modelica™	GT-Suite™
Mass flow rate [g/s]	11.12	11.15	11.15	11.20
Indicated Power [W]	261.16	247.7	235.05	244.59
Volumetric efficiency [%]	97.63	97.83	97.85	98.42

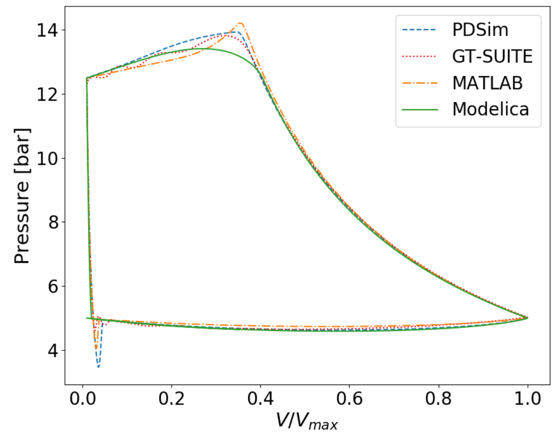
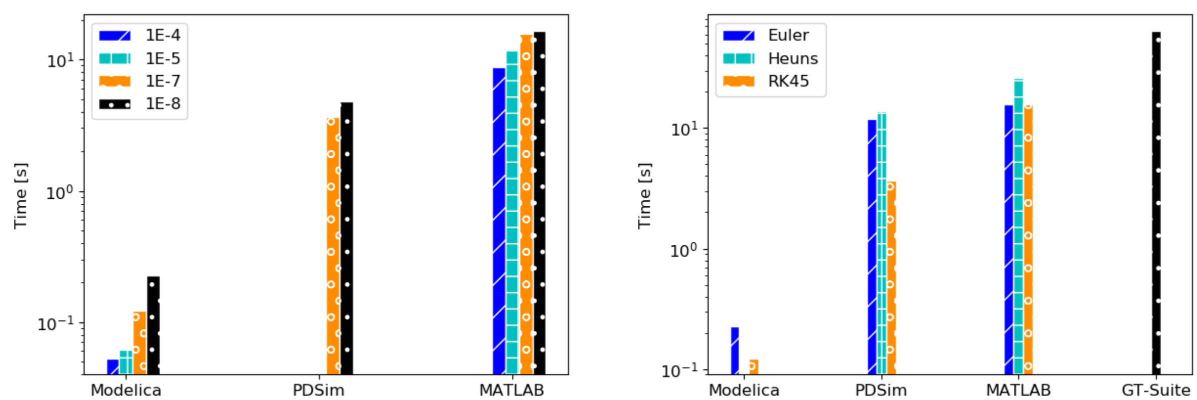


Figure 3: Indicator diagram across different platforms

5.2 Temporal Results for Four Platforms and Three Integrators

Figure 4a [1] shows the simulation time required by the four platforms for compressor modeling using the RK45 method at various solver error tolerances. The simulation time increases with reduced error tolerance due to the reduced step size requirement necessitating smaller steps. It was found that a significantly reduced minimum step size limit is required for large error tolerances. Often, choosing large error tolerance cause stability issues with the model.



(a) Effect of error tolerance in RK45

(b) Simulation time for different numerical integrators

Figure 4: Simulation time comparison

Figure 4a also provides a comparison of solution time for the compared platforms. Modelica™ is found to be significantly faster than other platforms compared in this article. One possible reason is the utilization of its built-in solver in the model which is optimized computationally for dynamic modeling. PDSim is found to be slightly faster than MATLAB™ with a couple of possible explanations being the use of Cython in PDSim for a reduction in interpreted code steps or the use of REFPROP in the MATLAB™ model and CoolProp in PDSim. Although there is a difference in simulation time for different platforms, It is not significant enough for a user to change modeling platform.

[1] The simulation time results for Modelica™, presented in Figure 4a and 4b, are CPU time instead of actual simulation time. A detailed explanation is presented in section 5.2.

Figure 4b presents the performance of different numerical integrators for compressor modeling, showing the integrators used on each platform and the impact on simulation speed. Although the RK45 is the most complex of the compared solvers, it takes much less simulation time as compared to Euler and Heun's method. For simulation time comparison across different platforms(excluding GT-Suite™), the compressor is simulated using the Euler method with a fixed number of steps. The Euler method is used with 7000 steps due to computational limitations and it should be noted that it would take a significantly higher than 7000 steps to match the same level of error tolerance for a direct comparison to RK45. Despite this, RK45 was still significantly faster than the Euler and Heun's integrator. It was found that Modelica™ takes the least amount of time for a solution while PDSim is slightly faster than the MATLAB™ based model. Contrary to the other platforms, the simulation time for Modelica™ is the CPU time as it is the only profiling variable that can be measured in Modelica™ with reasonable accuracy. It is observed that there is some overhead time (few seconds) associated with some simulation initialization that can only be recorded using a stopwatch. A stopwatch analysis shows that the actual simulation time for Modelica™is higher than CPU time presented in Figure 4a and 4b but with high levels of variability. The actual simulation time are similar orders of magnitudes as PDSim and MATLAB ™ but due to uncertainties in time measurement, these results are not reported in the text.

6. QUALITATIVE ANALYSIS OF FOUR PLATFORMS

All the aforementioned platforms are found to be capable of compressor modeling, but each one of them has some pros and cons. These platforms are evaluated on the basis of four criteria 1) User friendliness, 2)Ease for model development and 3) Future prospects.

6.1 User-Friendliness

A user-friendly simulation model should require minimum knowledge about the model from the user; but, at the same time, should be flexible enough to cater to the diverse needs of the different users. A GUI based platform is often very easy to use, with minimal knowledge requirement from a user, as it shadows most of the strenuous physics and programming. Among different platforms, compared in the text, GT-Suite™ is completely GUI based while others have some kind of GUI development tools available. GT-Suite™ has a drag and drop configuration, where components can be connected to model a system. It also allows users to define inputs in the form of parametric equations but does not provide a user the option to customize or change any background physics for a component.

Modelica™ also provides component development tools in the form of blocks. A block can be developed for each sub-model and, later on, these blocks can be connected in different orientations for system modeling. In Modelica™, the model should be a continuous function of time for successful simulation. The number of equations and variables should be equal because Modelica™ solves the equations simultaneously. These features are very useful for dynamic simulation and reduce the number of lines required for programming. It is very convenient for systems that involve less complex components but as the component complexity increases, its simultaneous solution structure provides less control and often causes instability to the simulations.

PDSim also has a GUI interface available, where a user can enter compressor parameter (*i.e.* dimensions, flow parameters, etc.) to get results and it requires minimum compressor knowledge. It also provides the user with an option to modify the individual models by adding Python functions and/or model the entire compressor using a script to call pre-coded software objects. The GUI interface of PDSim is not as robust as it is in GT-Suite ™. It does not allow flexibility for connecting components in different orientations for example modeling of multi-cylinder compressor. But its object oriented model provides full control to the user for model modification and development activities.

6.2 Model Troubleshooting

Another most important feature specific to a platform is easiness for troubleshooting. Most of the effort goes into troubleshooting by a user when using previously developed models. A platform must be able to provide sufficient information for troubleshooting. Apart from syntax or platform-specific errors, errors could be related to the model and can be anticipated by the developer. Flags should be added to the program to raise those errors to ease up the troubleshooting process. In GT-Suite™, if a simulation fails, it most often refers to the individual component and in some cases to the parameter which is causing a problem. It often proves useful in identifying the problem. In Modelica™, as variables can be assigned types and units, it makes it easy to find unit errors but sometimes, due to its simulation equation solution structure, troubleshooting a code becomes very difficult. For example, if a model is over-defined or under-defined, then it will only raise an error mentioning the model is structurally singular. Often, it becomes very hard to find which variable is missing or which equation needs to be added to resolve the issue. Contrary, as PDsim is based on Python, it treats the code as a set of instructions to execute. If something causes a simulation error, it returns the last instance which causes the error. Additionally, models related problems, for example specifying the wrong values to a parameter, are anticipated and flags are added to point out the problem.

6.3 Future Prospects

Finally, the last comparison is for prospects of compressor modeling across each platform. For GT-Suite™, being commercial software, it is not individual discretion to improve or add new compressor models to the software. Modelica™ has a fast computational time and easy object-oriented based syntax but its simulation focused solution structure is not convenient for complex compressor model development. Components in Modelica™ can be developed easily but it could be very time consuming for a user to troubleshoot it. As the purpose of such model development is to ease the efforts at a later stage, Modelica™ might not be a suitable option for compressor modeling. MATLAB™ is a powerful tool for model development and Simulink could be very helpful in defining drag and drop configuration but it is a closed source platform and may limit the researchers from contributing towards model development. Additionally, the availability of the model to the user will be limited. In contrast, PDSim is an open-source package. So, anyone can contribute to its development. Moreover, it is based on Python which also an open-source high-level programming language and is widely used among the research community. While sufficiently good results can be obtained from PDSim's GUI interface, it involves a steep learning curve associated with its text-based model development.

7. CONCLUSION

Platforms for compressor modeling are compared by modeling the same reciprocating compressor on both quantitative and qualitative bases. The model developed on each platform was found to be similar enough for direct comparison. Each platform was found to have pros and cons associated with it and depending on the modeling application, some platforms may be better suited to specific circumstances for compressor modeling. For performance evaluation of low-GWP refrigerants, PDSim possesses the better combinations of features. Although its simulation time is higher than Modelica™, the simulation time difference is not significant. Additionally, the availability of pre-coded libraries for mechanistic chamber model can be helpful in reducing the time associated with model development. While GT-Suite is more user friendly than PDSim, PDSim offers more flexibility for model improvement and customization. Moreover, it is an open-source platform that allows researchers to contribute to its development.

NOMENCLATURE

$\dot{m}$	Mass flow rate	[kg/s]
$\dot{m}_{out}$	Total mass leaving the control volume	[kg/s]
$\dot{Q}_{amb}$	Average heat transfer rate from cylinder wall to ambient	[W]
$\dot{Q}_{in}$	Instantaneous heat transfer rate from cylinder wall to the refrigerant	[W]
$\dot{W}$	Rate of work done	[W]
γ	Heat capacity ratio	[-]
ρ	Density	[kg/m^3]
Re	Reynolds number	[-]
θ	Crank angle	[rad]
A_p	Cylinder cross-sectional area	[m^2]
A_v	Area of the valve	[m^2]
A_{cyl}	Outer surface area of the cylinder	[m^2]
E_{CV}	Energy inside the control volume	[J]
h_a	Ambient air heat transfer coefficient	[J/kg]
h_{in}	Specific enthalpy of the refrigerant at suction	[J/kg]
h_{out}	Specific enthalpy of refrigerant at discharge	[J/kg]
h_{ref}	Refrigerant convection heat transfer	[J/kg]
P	Pressure	[kPa]
P_{high}	Pressure on high pressure side of the valve	[kPa]
P_{low}	Pressure on low pressure side of the valve	[kPa]
R	Specific gas constant	[J/kg.K]
T	Instantaneous temperature in control volume	[K]
t	Time	[s]
T_w	Cylinder wall temperature	[K]
T_{amb}	Ambient air temperature	[K]
u	Specific internal energy	[J/kg]

V	Instantaneous volume of the control volume	[m^3]	Nu	Nusselt number	[-]
V_0	Clearance volume	[m^3]			
V_{disp}	Total displaced volume	[m^3]	Pr	Prandtl number	[-]

REFERENCES

Bell, I. H., Wronski, J., Quoilin, S., & Lemort, V. (2014). Pure and Pseudo-Pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp. *Industrial & Engineering Chemistry Research*, *53*, 2498–2508.

Bell, I. H., Ziviani, D., Lemort, V., Bradshaw, C. R., Mathison, M., Horton, W. T., … Groll, E. A. (2020a). PDSim: A General Quasi-Steady Modeling Approach for Positive Displacement Compressors and Expanders. *International Journal of Refrigeration*, *110*, 310–322. doi: 10.1016/j.ijrefrig.2019.09.002

Bell, I. H., Ziviani, D., Lemort, V., Bradshaw, C. R., Mathison, M., Horton, W. T., … Groll, E. A. (2020b). Pdsim: A general quasi-steady modeling approach for positive displacement compressors and expanders. *International Journal of Refrigeration*, *110*, 310–322.

Chen, Y., Halm, N. P., Braun, J. E., & Groll, E. A. (2002). Mathematical Modeling of Scroll Compressors- part II: Overall Scroll Compressor Modeling. *International Journal of Refrigeration*, *25*(6), 751–764. doi: 10.1016/S0140-7007(01)00072-X

GT-Suite™. (2019). *v2019*. Westmont, Illinois: Gamma Technologies.

Kim, J. H., & Groll, E. A. (2007). Feasibility Study of a Bowtie Compressor with Novel Capacity Modulation. *International Journal of Refrigeration*, *30*(8), 1427–1438.

MATLAB. (2019). *version 9.7.0 (R2019b)*. Natick, Massachusetts: The Mathworks Inc.

Pereira, E. L. L., Deschamps, C. J., & Ribas, F. A. (2008). A Comparative Analysis of Numerical Simulation Approaches for Ring Valve Dynamics. In *International compressor engineering conference.*

Tanveer, M. M., & Bradshaw, C. R. (2020). Quantitative and qualitative evaluation of various positive-displacement compressor modeling platforms. *International Journal of Refrigeration*, *119*, 48–63.

Ziviani, D., Bell, I. H., Zhang, X., Lemort, V., Paepe, M. D., Braun, J. E., & Groll, E. A. (2020). PDSim: Demonstrating the Capabilities of an Open-Source Simulation Framework for Positive Displacement Compressors and Expanders. *International Journal of Refrigeration*, *110*, 323–339. doi: 10.1016/j.ijrefrig.2019.10.015