The Journal of Open Source Software

CirculationModels.jl - A ModelingToolkit Library for 0D-Lumped-Parameter Models of the Cardiovascular Circulation

Torsten Schenkel [1,2*] **and Harry Saxton** [2*]

1 Department of Engineering and Mathematics, Sheffield Hallam University, UK **2** Materials and Engineering Research Institute MERI, Sheffield Hallam University, UK ***** These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 05 December 2022
Published: unpublished

Summary

Within the realm of circulatory mechanics, lumped parameter (0D) modelling (Shi et al., 2011) offers the unique ability to examine both cardiac function and global hemodynamics within the context of a single model. Due to the interconnected nature of the cardiovascular system, being able to quantify both is crucial for the effective prediction and diagnosis of cardiovascular diseases. Lumped parameter modelling derives one of its main strengths from the minimal computation time required to solve ODEs and algebraic equations. Furthermore, the relatively simple structure of the model allows most personalized simulations to be automated. Meaning the ability to embed these lumped parameter models into a clinical workflow could one day become trivial (Bozkurt et al., 2022; Holmes & Lumens, 2018).

CirculationModels.jl is a Julia (Bezanson et al., 2017) package, built on the acausal modelling framework provided by *ModelingToolkit.jl* (Ma et al., 2021), containing all the common elements plus more needed for effective and realistic lumped parameter modelling. Currently *CirculationModels.jl* supports common elements such as a capacitor, resistor, inductance and diodes (Nicolaas Westerhof et al., 2010), which act as simple valve functions. We also make extensions to the common elements to include constant compliance chambers, non-linear and Poiseuille resistances (Pfitzner, 1976). Plus the Double-Hill, and Shi activation functions which are used as the cardiac driving chamber elastance's (Korakianitis & Shi, 2006; Stergiopulos et al., 1996). We also include non-linear valve functions from Shi, and Mynard (Korakianitis & Shi, 2006; Mynard et al., 2012). Alongisde individual components we also have created a collection of sub compartments including, full circulatory, systemic, pulmonary and heart models. We then also break down these full systems into collections of elements such as the famous Windkessel models (Nico Westerhof et al., 2009) to give the user full control over their modelling.

Users can easily add new elements to *CirculationModels.jl* using *ModelingToolkit.jl* functions.

Statement of need

Lumped parameter modelling has become an essential part of contributing strongly to our understanding of circulatory physiology. Lumped parameter models have been used in many different contexts across cardiovascular medicine such as aortic valve stenosis, neonatal physiology and detection of coronary artery disease (Dash et al., 2022; Laubscher et al., 2022; Sepúlveda Oviedo et al., 2022). There already exists some popular packages within both Simulink and Modelica such as the "Cardiovascular library for lumped-parameter modeling" (Rosalia et al., 2021) and "Physiolibary" (Mateják et al., 2014) respectively. These languages operate a block orientated "drag and drop" approach with Modelica been the common choice due to it's acausal modelling approach (Schweiger et al., 2020). Other languages, based on XML, exist for lumped parameter modelling such as CellML (Cuellar et al., 2003) and SBML (Hucka et al., 2003), while these are great for exchanging models they are often difficult to implement and model analysis is limited. A common theme within all current lumped parameter modelling software is the systems inability to deal with complex event handling and non-linear components. Being based on *ModelingToolkit.jl*, *CirculationModels.jl* overcomes these limitations by leveraging the wider *SciML* framework.

CirculationModels.jl provides the Julia community with a quick and effective way to perform lumped parameter modelling, being the first within the field to leverage both multiple dispatch and JIT compilation. As a result of Julia's architecture, as the complexity of the model increases the model analysis time does not become unreasonable.

Other packages exist which allow users to import models from other frameworks, CellMLToolkit.jl (Iravanian & others, 2021) and OpenModelica.jl (Tinnerholm et al., 2022). *CirculationModels.jl* goes beyond these by providing a lumped parameter modelling library with seamless integration with the SciML framework (Dixit & Rackauckas, 2022; Rackauckas et al., 2020; Rackauckas & Nie, 2017) which allows for extensive and efficient model analysis. Since both the modelling library and the framework it is built on are pure Julia, new components can be developed in a transparent and consistent manner.

Using Julia, *CirculationModels.jl* models compute significantly faster than models implemented in Matlab or Python, and at the same speed as specialised C-code (Table 1). This allows the models to run in real-time, and opens the possibility of global parameter optimisation, global sensitivity analysis.

The modular, acausal approach also allows quick and straightforward changes to the topology of the model, which can be automated using meta-programming techniques.

Example

Validation and benchmarking was performed on a full, 4-chamber, model of the circulation system proposed by (Korakianitis & Shi, 2006) (Figure 1). This model was previously implemented in CellML (Shi et al., 2018), which makes it an ideal candidate for validation of the new modeling library[1].

Model results from *CirculationModels.jl* model (Figure 2) are a perfect match for the CellML model. The CellML model was run in three versions: (1) imported into *ModelingToolkit.jl* using *CellMLToolKit.jl*, (2) Matlab code exported from CellML, (3) Python/SciPy code exported from CellML. Speedup against Matlab and Python is 2 and 3 orders of magnitude, respectively (Table 1).

[1]Note that CellML does not allow the callbacks which are required for the extended valve model, so only the simplified model can be compared. The *CirculationModels.jl* implementation of (Korakianitis & Shi, 2006) includes the extended model as well.

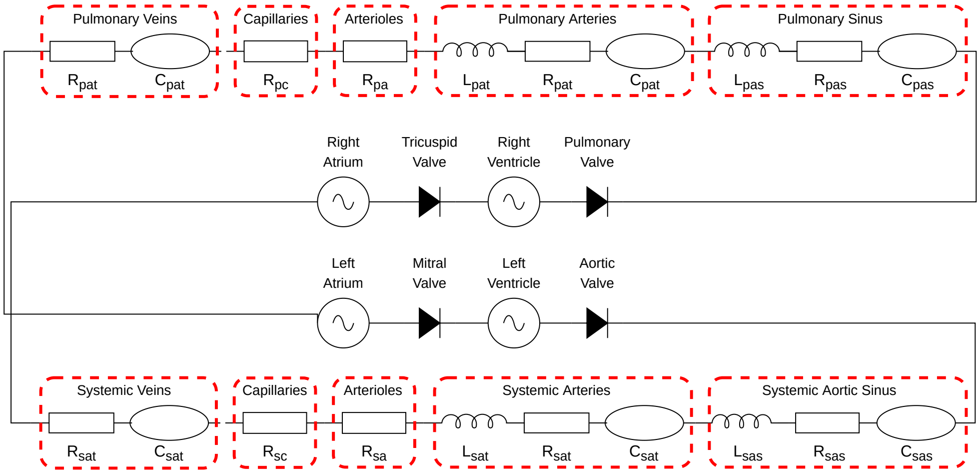

Figure 1: 4-chamber, full-circulationmodel from (Korakianitis & Shi, 2006). Groupings in dashed rectangles are implemented as compound subsystems, which in turn have been composed from individual resistor, compliance, and inertance elements. Ventricles and Atria implemented as time-variable elastances. Both simplified and dynamic, non-linear valve models are implemented.

Table 1: Simulation time comparison for a single run of (Korakianitis & Shi, 2006). CirculationModels.jl model was implemented from scratch. CellML model was imported into ModelingToolkit.jl using CellMLToolkit.jl, Matlab and Python models were created from the CellML code and downloaded from the CellML Model Repository.

CirculationModels.jl	CellMLToolkit.jl	Matlab (ode45)	Python (scipy.solve)
1x	1.6x	272x	963x

The Journal of Open Source Software

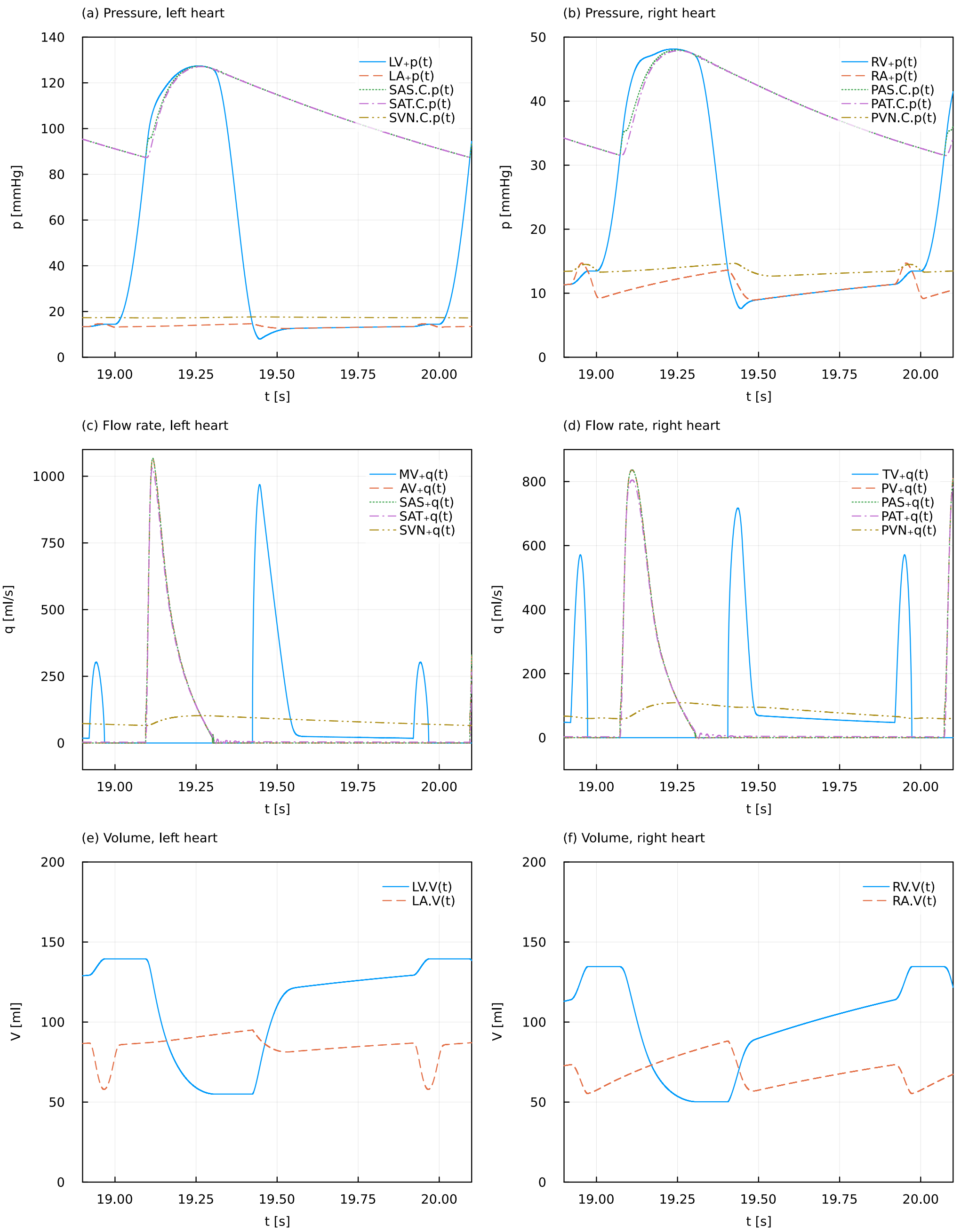

Figure 2: (a-f) Results for simplified model (Korakianitis & Shi, 2006) implemented in *CirculationModels.jl*. These results match the results from the CellML models (not shown).

Acknowledgements

Harry Saxton is supported by a Sheffield Hallam University Graduate Teaching Assistant PhD scholarship.

References

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98.

Bozkurt, S., Paracha, W., Bakaya, K., & Schievano, S. (2022). Patient-specific modelling and parameter optimisation to simulate dilated cardiomyopathy in children. *Cardiovascular Engineering and Technology*, 1–13.

Cuellar, A. A., Lloyd, C. M., Nielsen, P. F., Bullivant, D. P., Nickerson, D. P., & Hunter, P. J. (2003). An overview of CellML 1.1, a biological model description language. *Simulation*, *79*(12), 740–747.

Dash, A., Jain, K., Ghosh, N., & Patra, A. (2022). Non-invasive detection of coronary artery disease from photoplethysmograph using lumped parameter modelling. *Biomedical Signal Processing and Control*, *77*, 103781.

Dixit, V. K., & Rackauckas, C. (2022). GlobalSensitivity.jl: Performant and parallel global sensitivity analysis with julia. *Journal of Open Source Software*, *7*(76), 4561. https://doi.org/10.21105/joss.04561

Holmes, J. W., & Lumens, J. (2018). Clinical applications of patient-specific models: The case for a simple approach. *Journal of Cardiovascular Translational Research*, *11*(2), 71–79.

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., & others. (2003). The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, *19*(4), 524–531.

Iravanian, S., & others. (2021). *CellMLToolkit.jl: A Julia library that connects CellML models to SciML*. https://github.com/SciML/CellMLToolkit.jl.

Korakianitis, T., & Shi, Y. (2006). Numerical simulation of cardiovascular dynamics with healthy and diseased heart valves. *Journal of Biomechanics*, *39*(11), 1964–1982.

Laubscher, R., Merwe, J. van der, Liebenberg, J., & Herbst, P. (2022). Dynamic simulation of aortic valve stenosis using a lumped parameter cardiovascular system model with flow regime dependent valve pressure loss characteristics. *Medical Engineering & Physics*, *106*, 103838.

Ma, Y., Gowda, S., Anantharaman, R., Laughman, C., Shah, V., & Rackauckas, C. (2021). *ModelingToolkit: A composable graph transformation system for equation-based modeling*. https://arxiv.org/abs/2103.05244

Mateják, M., Kulhánek, T., Šilar, J., Privitzer, P., Ježek, F., & Kofránek, J. (2014). Physiolibrary-modelica library for physiology. *Proceedings of the 10th International Modelica Conference*, *96*, 499–505.

Mynard, J., Davidson, M., Penny, D., & Smolich, J. (2012). A simple, versatile valve model for use in lumped parameter and one-dimensional cardiovascular models. *International Journal for Numerical Methods in Biomedical Engineering*, *28*(6-7), 626–641.

Pfitzner, J. (1976). Poiseuille and his law. *Anaesthesia*, *31*(2), 273–275.

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., & Edelman, A. (2020). Universal differential equations for scientific machine learning. *arXiv Preprint arXiv:2001.04385*.

Rackauckas, C., & Nie, Q. (2017). Differentialequations. Jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, *5*(1).

Rosalia, L., Ozturk, C., Van Story, D., Horvath, M. A., & Roche, E. T. (2021). Object-oriented lumped-parameter modeling of the cardiovascular system for physiological and pathophysiological conditions. *Advanced Theory and Simulations*, *4*(3), 2000216.

Schweiger, G., Nilsson, H., Schoeggl, J., Birk, W., & Posch, A. (2020). Modeling and simulation of large-scale systems: A systematic comparison of modeling paradigms. *Applied Mathematics and Computation*, *365*, 124713.

Sepúlveda Oviedo, E. H., Bermeo Clavijo, L. E., & Méndez Córdoba, L. C. (2022). OpenModelica-based virtual simulator for the cardiovascular and respiratory physiology of a neonate. *Journal of Medical Engineering & Technology*, *46*(3), 179–197.

Shi, Y., Lawford, P., & Hose, D. R. (2018). Construction of lumped-parameter cardiovascular models using the CellML language. *Journal of Medical Engineering & Technology*, *42*(7), 525–531. https://doi.org/10.1080/03091902.2019.1576792

Shi, Y., Lawford, P., & Hose, R. (2011). Review of zero-d and 1-d models of blood flow in the cardiovascular system. *Biomedical Engineering Online*, *10*(1), 1–38.

Stergiopulos, N., Meister, J.-J., & Westerhof, N. (1996). Determinants of stroke volume and systolic and diastolic aortic pressure. *American Journal of Physiology-Heart and Circulatory Physiology*, *270*(6), H2050–H2059.

Tinnerholm, J., Pop, A., & Sjölund, M. (2022). A modular, extensible, and modelica-standard-compliant OpenModelica compiler framework in julia supporting structural variability. *Electronics*, *11*(11), 1772.

Westerhof, Nico, Lankhaar, J.-W., & Westerhof, B. E. (2009). The arterial windkessel. *Medical & Biological Engineering & Computing*, *47*(2), 131–141.

Westerhof, Nicolaas, Stergiopulos, N., Noble, M. I., & Westerhof, B. E. (2010). *Snapshots of hemodynamics: An aid for clinical research and graduate education* (Vol. 7). Springer.