

# openCARP: An Open Sustainable Framework for In-Silico Cardiac Electrophysiology Research

Jorge Sánchez<sup>1</sup>, Mark Nothstein<sup>1</sup>, Aurel Neic<sup>4</sup>, Yung-Lin Huang<sup>2</sup>, Anton J Prassl<sup>3</sup>, Jochen Klar<sup>7</sup>, Robert Ulrich<sup>1</sup>, Felix Bach<sup>1</sup>, Philipp Zschumme<sup>1</sup>, Michael Selzer<sup>1</sup>, Gernot Plank<sup>3</sup>, Edward Vigmond<sup>5,6</sup>, Gunnar Seemann<sup>8</sup>, Axel Loewe<sup>1</sup>

<sup>1</sup>Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

<sup>2</sup>Universitäts-Herzzentrum Freiburg, Freiburg, Germany

<sup>3</sup>Medical University of Graz, Gottfried Schatz Research Center, Division of Biophysics, Graz, Austria

<sup>4</sup>NumeriCor GmbH, Graz, Austria

<sup>5</sup>Univ. Bordeaux, IMB, UMR 5251, Talence, France

<sup>6</sup>LIRYC Electrophysiology and Heart Modeling Institute, Bordeaux, France

<sup>7</sup>Independent researcher, Berlin, Germany

<sup>8</sup>Institute for Experimental Cardiovascular Medicine, University Heart Center Freiburg-Bad Krozingen 2:Faculty of Medicine, Albert-Ludwigs University of Freiburg, Freiburg, Germany

## Abstract

*openCARP is an open cardiac electrophysiology simulator, released to the community to advance the computational cardiology field by making state-of-the-art simulations accessible. It aims to achieve this by supporting self-driven learning. To this end, an online platform is available containing educational video tutorials, user and developer-oriented documentation, detailed examples, and a question-and-answer system. The software is written in C++. We provide binary packages, a Docker container, and a CMake-based compilation workflow, making the installation process simple. The software can fully scale from desktop to high-performance computers. openCARP runs nightly tests to ensure the consistency of the simulator based on predefined reference solutions, keeping a high standard of quality for all of its components. Additionally, sustainability is achieved through automated continuous integration to generate not only software packages, but also documentation and content for the community platform. Furthermore, carputils provides an environment for users to create complex, multi-scale simulations that are shareable and reproducible. In conclusion, openCARP is a tailored software solution for the scientific community in the cardiac electrophysiology field and contributes to increasing use and reproducibility of in-silico experiments.*

## 1. Introduction

Multi-scale computational models of cardiac electrophysiology are used to investigate complex phenomena,

such as cardiac arrhythmias and appropriate therapies. While a couple of software solutions exist, none fully meets the needs of the community. In particular, newcomers to the field often have to go through a very steep learning curve, which could be mitigated by dedicated user interfaces, documentation, and training material. openCARP ([www.opencarp.org](http://www.opencarp.org)) is a cardiac electrophysiology simulator for in-silico experiments. Its source code is public and the software is freely available for academic purposes. openCARP offers single cell as well as multi-scale simulations from ion channel to organ level. The Python-based carputils framework supports users to develop and share simulation pipelines, i.e., automating in-silico experiments including modeling, pre-processing, simulation, and post-processing steps. In this way, openCARP also aims to increase productivity, transparency and reproducibility in our research field. We target a research community interested in solving cardiac electrophysiology problems rather than developing simulation software. openCARP interfaces with standard data formats, which allows importing third-party cellular models in CellML format from established repositories to avoid error-prone and time-demanding manual implementation. Soon, we plan to offer easy ways to upload and persistently share in-silico experiments (based on carputils) with the community. openCARP is supported by software and community infrastructure for sustainable lifecycle management of research software [1]. In this work, we highlight the key features and use cases of openCARP and the ecosystem around it.

## 2. Methods

The openCARP project aims to achieve sustainability by building a community. Therefore, a broad survey and needs analysis among potential users in the cardiac modeling community formed the basis for design choices and implementation priorities, both for the simulator itself, as well as accompanying documentation, training materials, and infrastructure for community interaction.

### 2.1. Software Architecture

openCARP's software design is based on flexibility, modularity, and maintainability. All simulator components are implemented according to the C++ 2011 standard. The software runs under different operating systems using state-of-the-art numerical methods that allow the user to run the simulator on different hardware architectures. To ensure the quality of the software and its modules, different predefined tests including the benchmark problem described by Niederer et al. [2] are run regularly and compared to reference solutions, guaranteeing consistency of the simulation results.

### 2.2. Sustainability

From the beginning, the openCARP project was a joint effort between different institutions (Figure 1). Therefore, infrastructure and processes for distributed development, management, and support are of vital importance. openCARP uses GitLab as a web-based DevOps lifecycle tool supporting planning, creating, building, verifying, and deploying the software.



Figure 1. Closed-loop approach for a sustainable research software with a focus on community interaction.

Sustainability is also based on a high level of automation of routine processes in continuous integration. Deployment pipelines remove the burden from developers and enable optimal use of resources.

Users of the software are provided with detailed support material that addresses different topics including installation and the use of openCARP. Community contributions ranging from bug reports to code and documentation are welcome and encouraged. Structured and commented

source code is released along with a software reference documentation.

## 3. Results and Use cases

### 3.1. Software

openCARP can be installed on a variety of operating systems. Binary packages and Docker containers are provided as production-ready software. Custom compilation and installation from the source code is supported by a CMake workflow simplifying this process for the user. Automated continuous testing is an integral part of the openCARP development process and quality management. For release versions, continuous deployment pipelines provide installation packages and make them available on the web page together with the respective version of the documentation, to minimize the effort for developers and ensure consistency.

openCARP is optimized to solve the cardiac monodomain and bidomain equations as previously described [3]. Parallelization is based on MPI and the use of the state-of-the-art numerical library PETSc [4] makes openCARP well-prepared to be used on a variety of hardware platforms ranging from desktop workstations to high-performance computers. This allows the user to run a wide range of multi-scale simulations (Figure 2) with a reasonable computational time. Additionally, openCARP is fully input and output compatible with its closed source predecessors CARP/CARPentry [5] to allow for an easy transition.

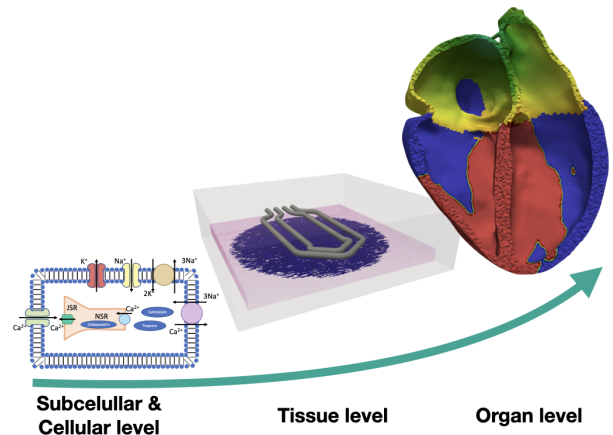


Figure 2. Multi-scale simulation from cellular to organ level using openCARP.

### 3.2. Community

openCARP is a meritocratic, consensus-based community project. Community contributions are highly appreciated and can take many forms like code contributions, bug reports, functional requirements, ideas for improving the software, documentation, or any aspect of the community platform. All code contributions will enter a review process by the software maintainers before merging with the master branch of the repository as a quality control measure.

We provide a long term repository with DOIs for the different releases of the code, a community platform, training material and in the future, a repository to share experiments and make them available with persistent identifiers.

The project web page serves as the central hub and provides, e.g., a variety of support and training materials. The support concept is based on a question-and-answer system that helps the community to interact and leverage crowd resources. Using this method of interaction, questions posted using the available tags will be redirected to the maintainers of the software.

The training materials target beginners in the field of cardiac electrophysiology simulations with introductory video tutorials and examples. For experienced users, we also provide video tutorials that cover more complex simulations.

Online and offline documentation as well as parts of the web page are automatically generated and kept in sync with the code by a continuous integration pipeline to ensure that all information is consistent and up-to-date. User documentation (both interactive web-based and static PDF-based) is automatically generated from the code based on the command line help, which is parsed and translated into markdown files.

Regular user meetings allow to train new users, exchange experience regarding the use of the software and stay in close link with the community as part of the continuous feedback loop.

### 3.3. Single cell experiments

openCARP includes a large number of built-in ionic models. Custom cellular models can be easily modified or created using *EasyML*. Additionally, it offers the possibility to import cellular models from external repositories provided in the CellML format [6] and interoperate with Myokit [7].

*bench* is an interface available for single cell experiments. It can also be called using *carputils*, which allows creating more complex single cell in-silico experiments. *bench* allows running built-in ionic models and dynamically loaded ones allowing the user to tune any parameter of individual channels, perform several pacing protocols,

and track the progression of variables along time (Figure 3). Parameters of the model are modified in a short and easy syntax that allows specifying relative or absolute values for parameters during run time without the need of creating new files or recompiling the software.

Reaching a stable limit cycle (dynamic steady state) is facilitated by pacing the cellular model at a certain basic cycle length for a longer period. This new state can be saved and used to restart a single cell or tissue experiment.

Clamping of any state variable to a prescribed curve can aid both in debugging ionic models and running sophisticated experiments. Additionally, clamping of the transmembrane voltage allows to understand the dynamics of the ionic channels.

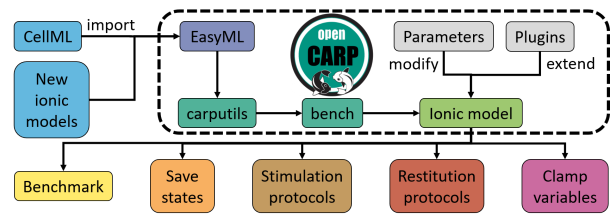


Figure 3. *bench* is a tool for single cell in-silico experiments and allows users to import/create a cell model and run different commonly used or custom electrophysiology protocols.

Restitution curves are an important measure of cellular/tissue behavior concerning the pacing frequency and help to understand how reentry in tissue may evolve. This process has been simplified with an automated function supporting both the basic S1-S2 protocol, which quantifies the propensity to develop an arrhythmia, and the dynamic protocol, which reflects the adaption to different basic cycle lengths and user-defined protocols.

### 3.4. Tissue/Organ experiments

With *carputils*, users can easily create complex tissue/organ in-silico experiments and share them (Figure 4). openCARP offers different tools to easily create simple meshes or import your custom mesh. It allows pre-processing of the meshes using dynamic tagging which creates two or more regions to include heterogeneities in the cardiac tissue. Using regions, users can easily load several cellular models and modify each one of them independently. Furthermore, users can also give different conductivity values to simulate the propagation of the electrical field in the intracellular and extracellular domains. Conduction velocity is an important feature of the cardiac tissue and in simulations is roughly proportional to the square root of the conductivity. openCARP offers a tool called *tuneCV* which helps to determine the conductivity values

to reach the desired conduction velocity and anisotropy ratios in tissue-scale simulations for grids of a given spatial resolution.

The user can also specify stimulation protocols like the number of pulses, basic cycle length, or load a predefined stimulation file. Cardiac tissue can be stimulated either by changing the electrical potential or by the injection of currents in the intra/extra-cellular domain. Additionally, the pulse shape can also be configured regarding duration, waveform (mono/bi-phasic, custom waveforms), and amplitude.

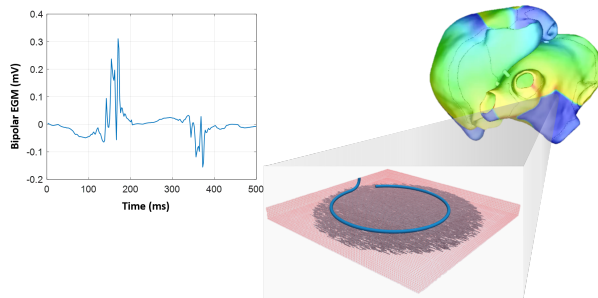


Figure 4. Bidomain in-silico experiment to study intracardiac electrogram signals. A rotational activity was initiated in a 3D volumetric bi-atrial mesh that includes fibrotic tissue. A bipolar fractionated electrogram was acquired by a virtual lasso catheter placed on the surface of the cardiac tissue.

Furthermore, users can also modify numerical solver parameters and switch between monodomain, pseudo-bidomain and bidomain formulations. While running the simulation, checkpoints can be saved to later on restart the simulation from these. This is e.g. useful to test the vulnerability of tissue to arrhythmia with several coupling intervals.

Visualization using *meshalyzer* allows to display time-dependent data on meshes. Additionally, openCARP offers several tools for post-processing. *igbAPD* calculates several features of the action potential like duration and amplitude. *igbextract* and *igbops* extract the potential of specific nodes and perform mathematical operations. Moreover, openCARP can export to standard file formats like VTK to enhance interoperability and facilitate advanced post-processing and visualization.

## 4. Conclusion

openCARP provides a framework to create and share in-silico experiments via a convenient interface requiring little effort and programming experience. It runs under different operating systems and provides an easy installation process. Together with the ecosystem around it, we believe openCARP is a high quality, user-friendly car-

diac electrophysiology simulation software. It supports the full research lifecycle from exploration through to conclusive analysis and publication, to archiving and sharing of data, experimental protocols, and source code. In this way, openCARP could be a suitable software solution for large parts of the cardiac electrophysiology community and contribute to the use, transparency and reproducibility of in-silico experiments.

## Acknowledgments

We gratefully acknowledge support by Deutsche Forschungsgemeinschaft (DFG, project ID 391128822) and the Austrian Science Fund(FWF, SFB F3210-N18 and I2760-B30).

## References

- [1] Loewe A, Seemann G, Wülfers EM, Huang YL, Sánchez J, Bach F, Ulrich R, Selzer M. SuLMaSS - sustainable lifecycle management for scientific software. In E-Science-Tage 2019: Data to Knowledge. 2019; .
- [2] Niederer SA, Kerfoot E, Benson AP, Bernabeu MO, Bernus O, Bradley C, Cherry EM, Clayton R, Fenton FH, Garny A, Heidenreich E, Land S, Maleckar M, Pathmanathan P, Plank G, Rodríguez JF, Roy I, Sachse FB, Seemann G, Skavhaug O, Smith NP. Verification of cardiac tissue electrophysiology simulators using an n-version benchmark. *Philosophical Transactions of the Royal Society A Mathematical Physical and Engineering Sciences* 2011;369(1954):4331–4351.
- [3] Vigmond EJ, Weber dos Santos R, Prassl AJ, Deo M, Plank G. Solvers for the cardiac bidomain equations. *Progress in biophysics and molecular biology* Jan-Apr 2008;96(1-3):3–18.
- [4] Abhyankar S, Brown J, Constantinescu EM, Ghosh D, Smith BF, Zhang H. Petsc/ts: A modern scalable ode/dae solver library. *arXiv preprint arXiv180601437* 2018;.
- [5] Vigmond EJ, Hughes M, Plank G, Leon LJ. Computational tools for modeling electrical activity in cardiac tissue. *J Electrocardiol* 2003;36 Suppl:69–74.
- [6] Yu T, Lloyd CM, Nickerson DP, Cooling MT, Miller AK, Garny A, Terkildsen JR, Lawson J, Britten RD, Hunter PJ, Nielsen PMF. The Physiome Model Repository 2. *Bioinformatics* 01 2011;27(5):743–744. ISSN 1367-4803.
- [7] Clerx M, Collins P, [de Lange] E, Volders PG. Myokit: A simple interface to cardiac cellular electrophysiology. *Progress in Biophysics and Molecular Biology* 2016; 120(1):100 – 114. ISSN 0079-6107. Recent Developments in Biophysics Molecular Biology of Heart Rhythm.

Address for correspondence:

Jorge Sánchez Arciniegas

Institute of Biomedical Engineering, Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany  
publications@ibt.kit.edu