The openCARP cardiac electrophysiology simulator

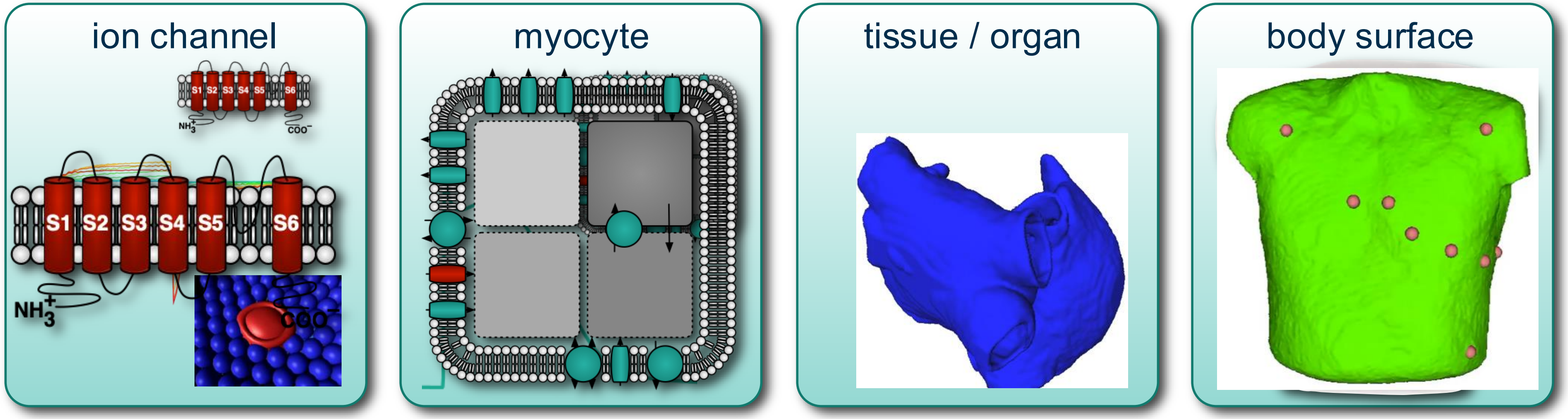
www.openCARP.org
helmholtz.software/software/opencarp

Axel Loewe
Institute of Biomedical Engineering (IBT)

Content

1. How can mathematics and software help address heart diseases?
2. The openCARP ecosystem
3. Research Software Engineering infrastructure

The heart as a multi-scale system



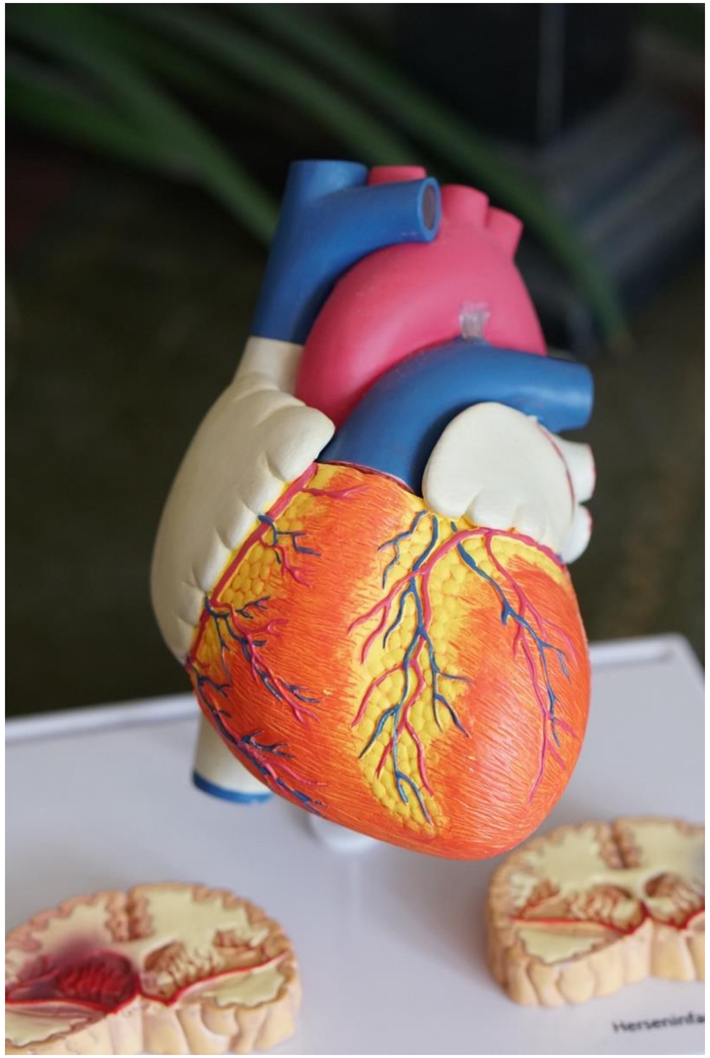
biological level of integration

nanometer

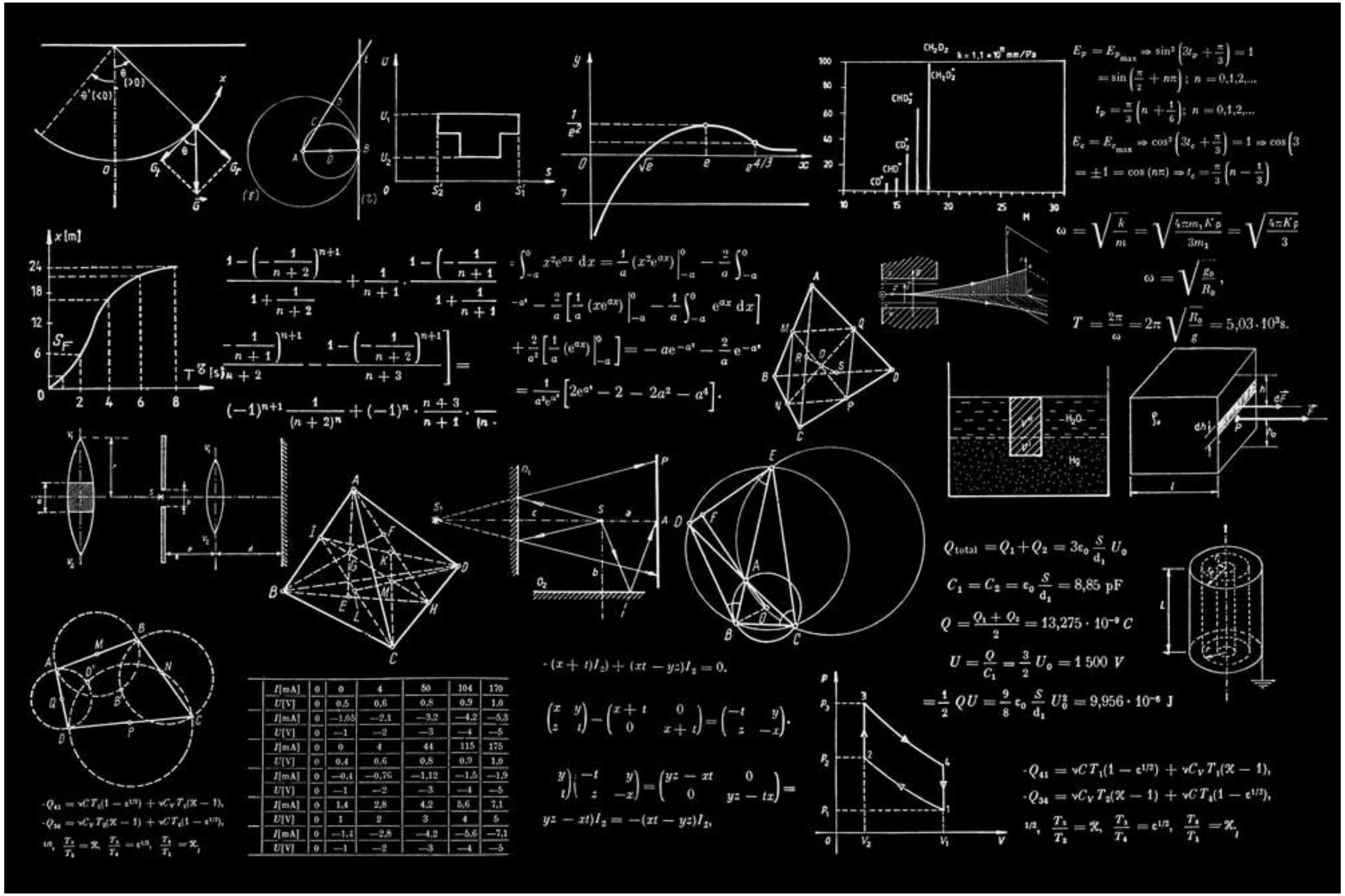
micrometer

centimeter

How can math help?

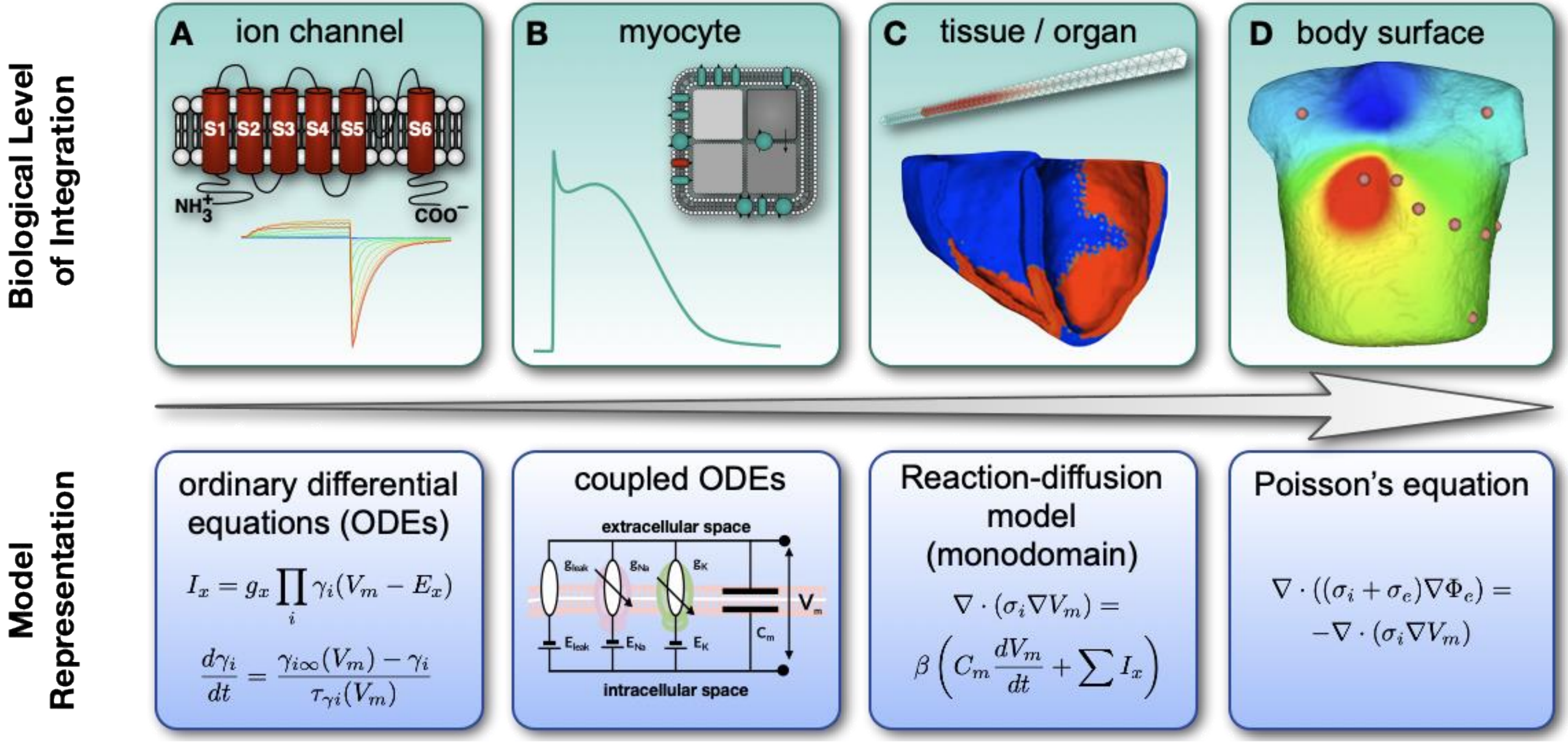


Robina Weermeijer, unsplash.com



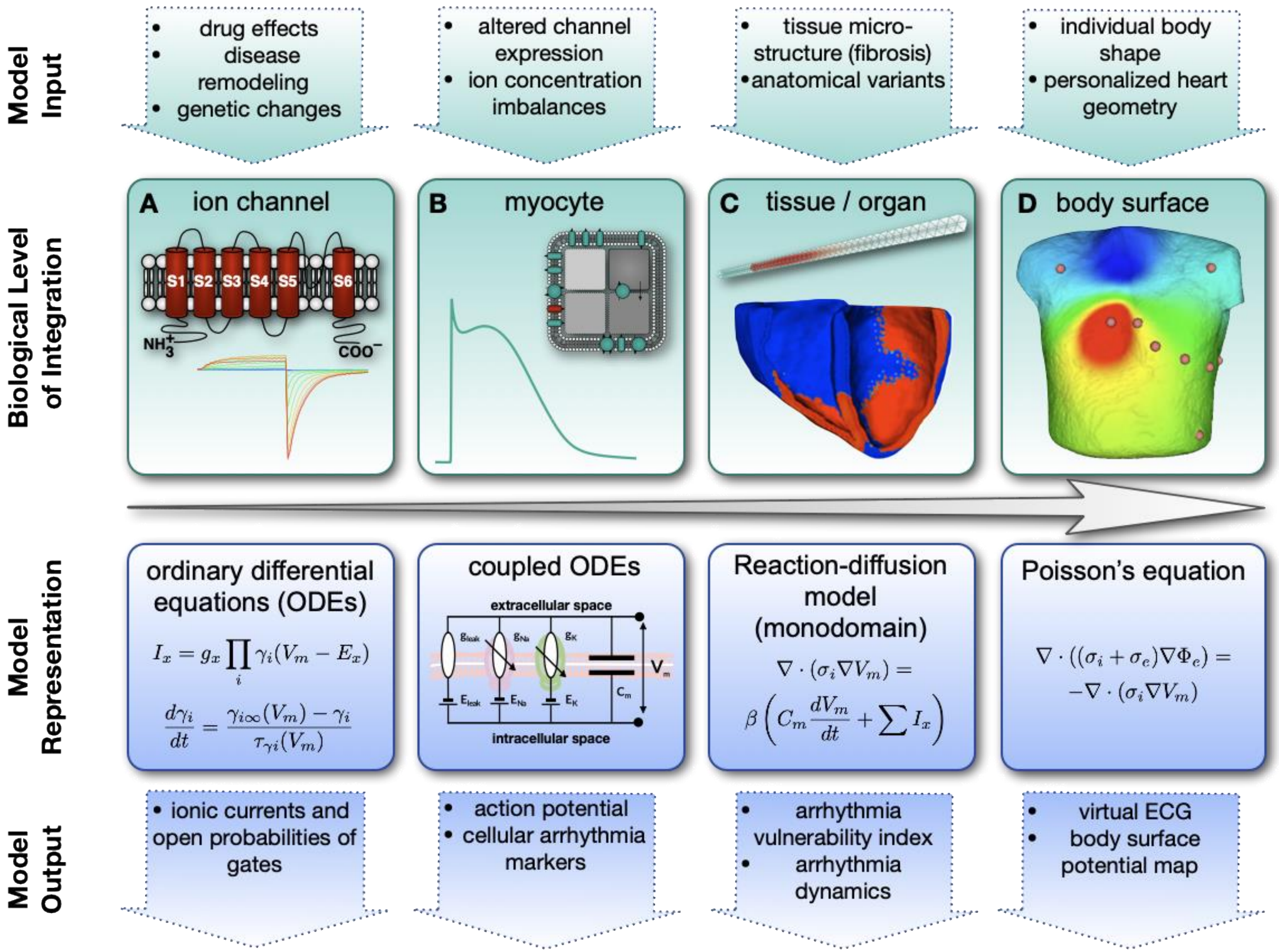
Dan-Cristian Pădureț, unsplash.com

Multi-scale system modeling of cardiac electrophysiology



Loewe et al. "Computational modelling of biological systems now and then: revisiting tools and visions from the beginning of the century" PTRSA 2025;383:20230384

Multi-scale system modeling of cardiac electrophysiology



Loewe et al. "Computational modelling of biological systems now and then: revisiting tools and visions from the beginning of the century" PTRSA 2025;383:20230384

Computer modeling & simulation

Controlled research environment

- to understand fundamental physiology and pathomechanisms
- to evaluate diagnostic and therapeutic approaches (in silico studies)
- to optimize device design

Personalised medicine through digital twins

Loewe et al. “Computational modelling of biological systems now and then: revisiting tools and visions from the beginning of the century“ PTRSA 2025;383:20230384

Kate Miller

DOC RABE Meda / stock.adobe.c

DSL Vorhofflimmern

Deutsche Herzstiftung

Azzolin et al., „Personalized ablation vs. conventional ablation strategies to terminate atrial fibrillation and prevent recurrence“, Europace 2023;25:211-22

Computer modeling & simulation

Controlled research environment

- to understand fundamental physiology and pathomechanisms
- to evaluate diagnostic and therapeutic approaches (in silico studies)
- to optimize device design

Personalised medicine through digital twins

Scalable research environment to generate big, quality-controlled dataset for machine learning

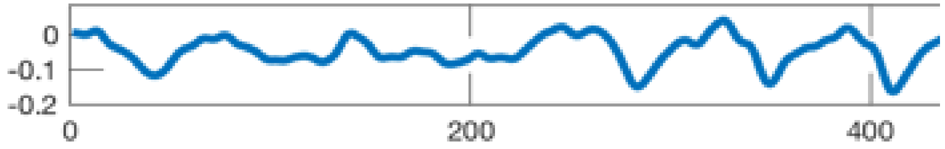
Loewe et al. “Computational modelling of biological systems now and then: revisiting tools and visions from the beginning of the century“ PTRSA 2025;383:20230384

Will pulmonary vein isolation be sucessful?

Universitäts Klinikum Freiburg

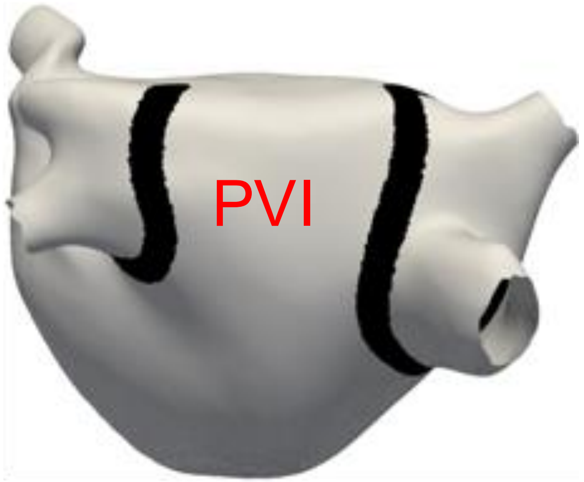
Städtisches Klinikum Karlsruhe

ECG



?

Atrial fibrillation



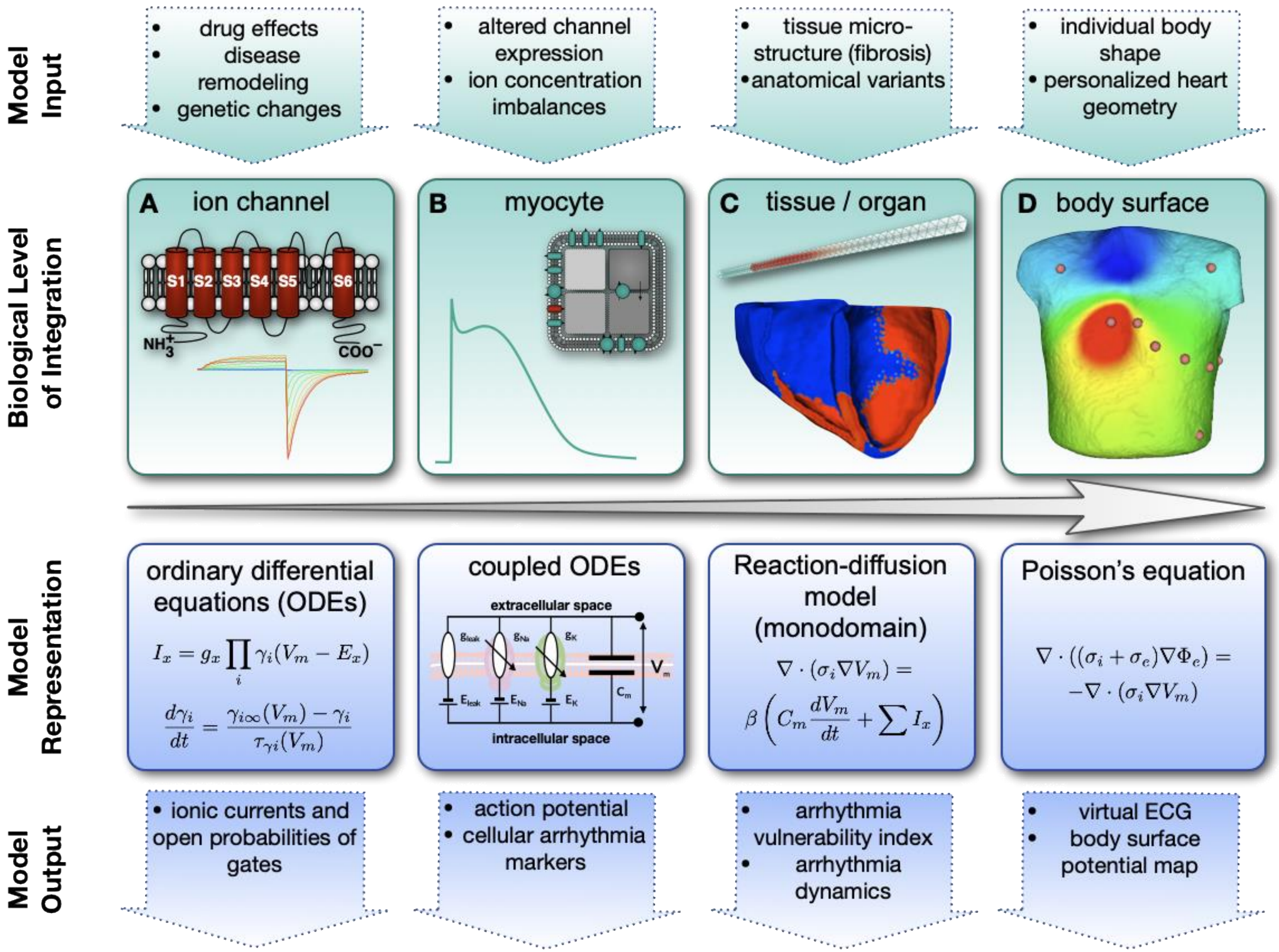
?

Atrial fibrillation

Luongo et al., "Machine learning enables noninvasive prediction of atrial fibrillation driver location and acute PVI success using the 12-lead ECG", J Cardiovasc Digital Health 2021;2:126-136

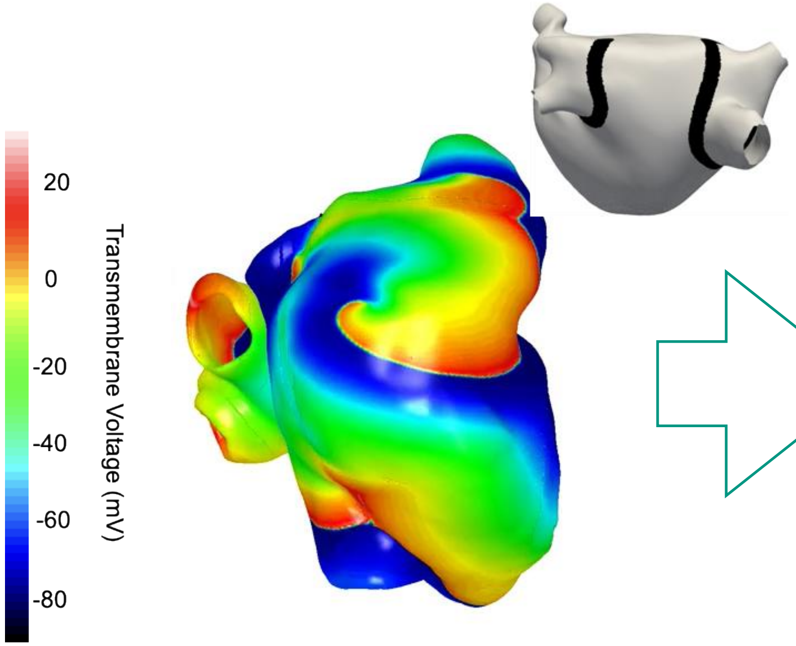
KIT

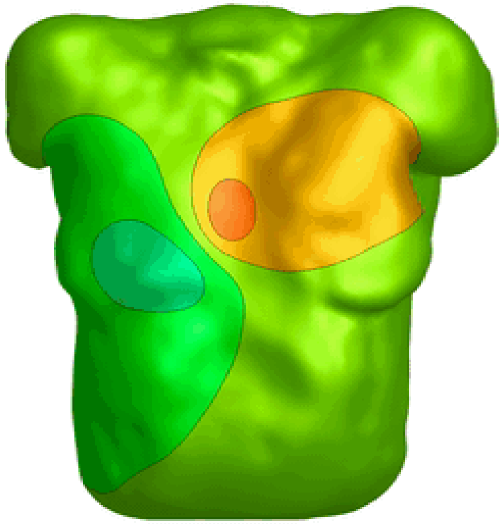
Multi-scale system modeling of cardiac electrophysiology

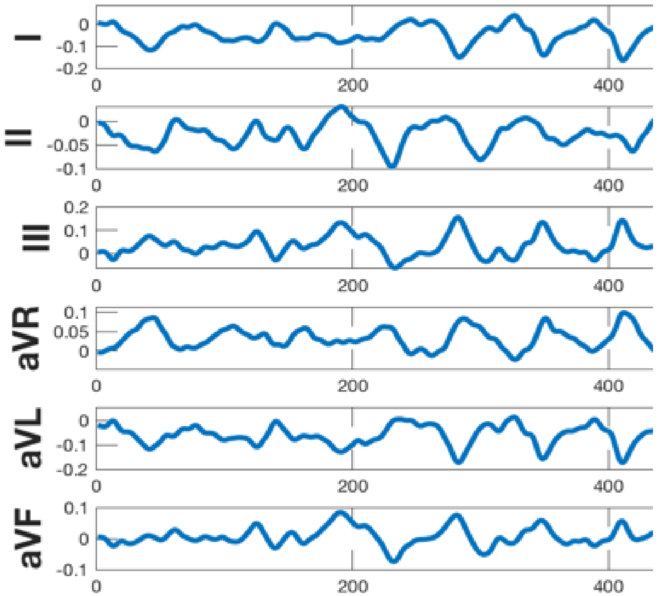


Loewe et al. "Computational modelling of biological systems now and then: revisiting tools and visions from the beginning of the century" PTRSA 2025;383:20230384

ECG simulations







Atrial fibrillation

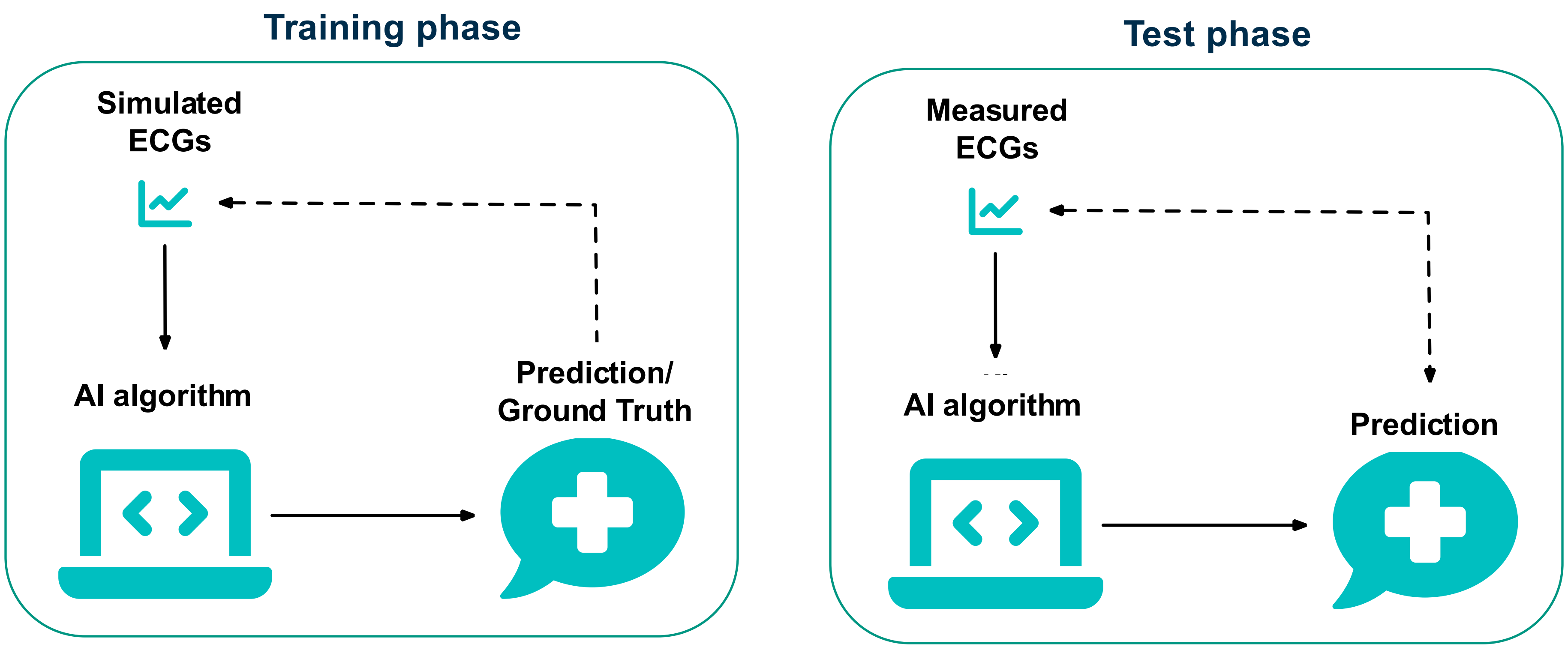
Electric potentials
on the body surface

ECG

> 1000 simulated atrial fibrillation episodes

Luongo et al., "Machine learning enables noninvasive prediction of atrial fibrillation driver location and acute PVI success using the 12-lead ECG", J Cardiovasc Digital Health 2021;2:126-136

Machine learning



46 patients:
82.6% specificity
73.9% sensitivity
93.5% consistency

Luongo et al., "Machine learning enables noninvasive prediction of atrial fibrillation driver location and acute PVI success using the 12-lead ECG", J Cardiovasc Digital Health 2021;2:126-136

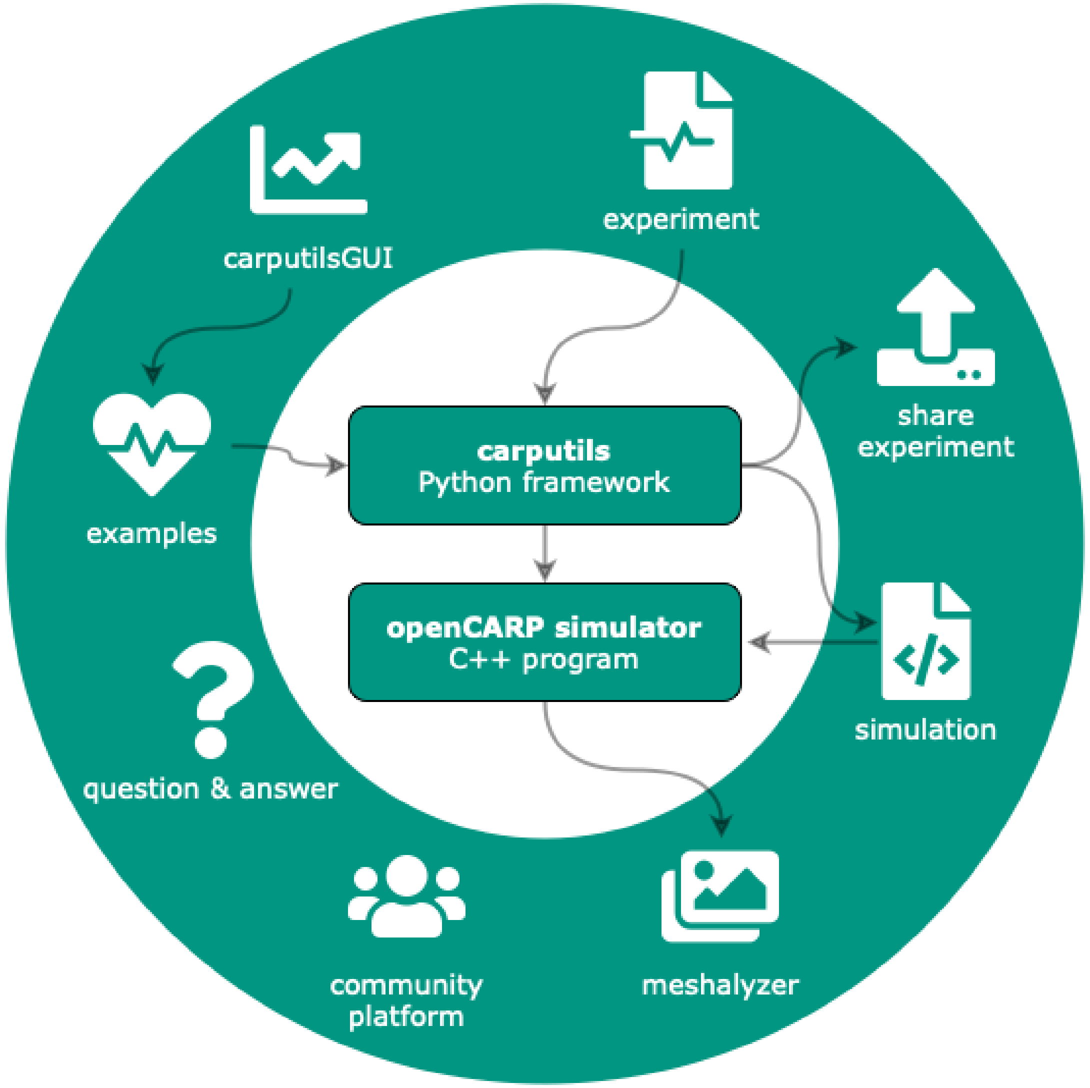
Content

1. How can mathematics and software help address heart diseases?
2. The openCARP ecosystem
3. Research Software Engineering infrastructure

The openCARP ecosystem

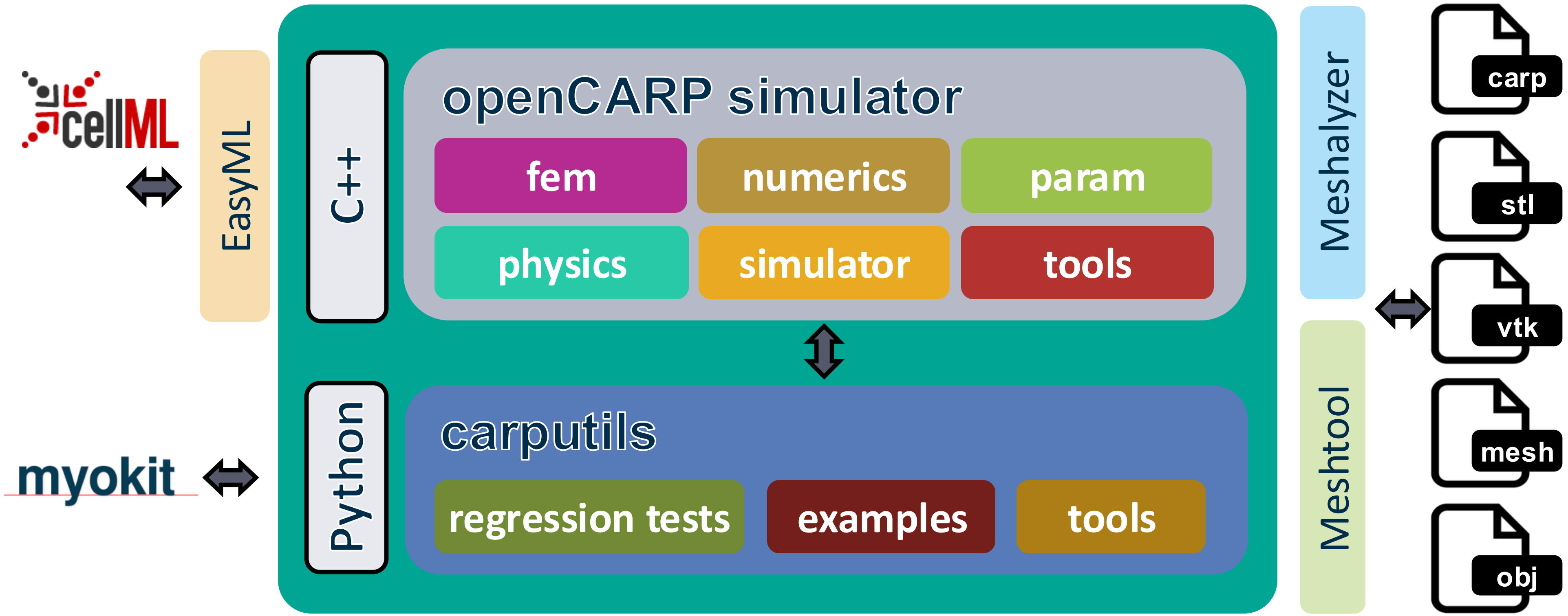
openCARP: C++ cardiac electrophysiology simulator, free for academic, non-commercial use. Simulations from ion channel to organ level.

carputils*:* Python framework to develop complex simulation pipelines, i.e. to automate in silico experiments including all modeling and simulation steps.



Plank et al. „The openCARP Simulation Environment for Cardiac Electrophysiology". Computer Methods and Programs in Biomedicine 2021;208:106223

Global architecture of the openCARP simulation platform



Plank et al. „The openCARP Simulation Environment for Cardiac Electrophysiology“ Computer Methods and Programs in Biomedicine 2021;208:106223

Examples

Mini-experiments coded up in carputils

www.openCARP.org > Documentation > Examples

Basic single cell EP
This example introduces the basic steps of running EP simulations in an isolated myocytes

APD restitution
Action potential duration (APD) restitution example in single cell. As pacing frequency is increased, APD shortens to maintain a one to one stimulus to responses

Voltage clamp
This example explains the basic usage of bench for performing voltage clamp experiments.

EasyML to C code
This tutorial explains the basics of using the code generation tool limpet_fe.py for generating ODE solver code for models of cellular dynamics.

EasyML basics
Here you will get more information about the cell model math format EasyML

Import CellML
Here you learn how to import CellML data into openCARP and what problems might occur during this process

single cell simulations

Basic tissue EP
This example introduces to the basics of using the openCARP executable for simulating EP at the tissue and organ scale

Extracellular stimulation
In this example you learn how to stimulate a tissue from the extracellular space

Tuning Conduction Velocity
This tutorial introduces the background for the relationship between conduction velocity and tissue conductivity

Init tissue from cell
This tutorial demonstrates how to initialize a cardiac tissue with state variables obtained from a single-cell simulation

Adjust parameters
This tutorial demonstrates how to adjust parameters in tissue simulations to match experimental data for conduction velocity, APD, and wavelength

CV restitution
This example demonstrates how to compute conduction velocity restitution in cardiac tissue

tissue simulations

visualization

pre- and postprocessing

IGB utils
This example demonstrates several utilities to manipulate IGB files.

limpetGUI
limpetGUI is a simple tool for visualizing traces output generated by bench

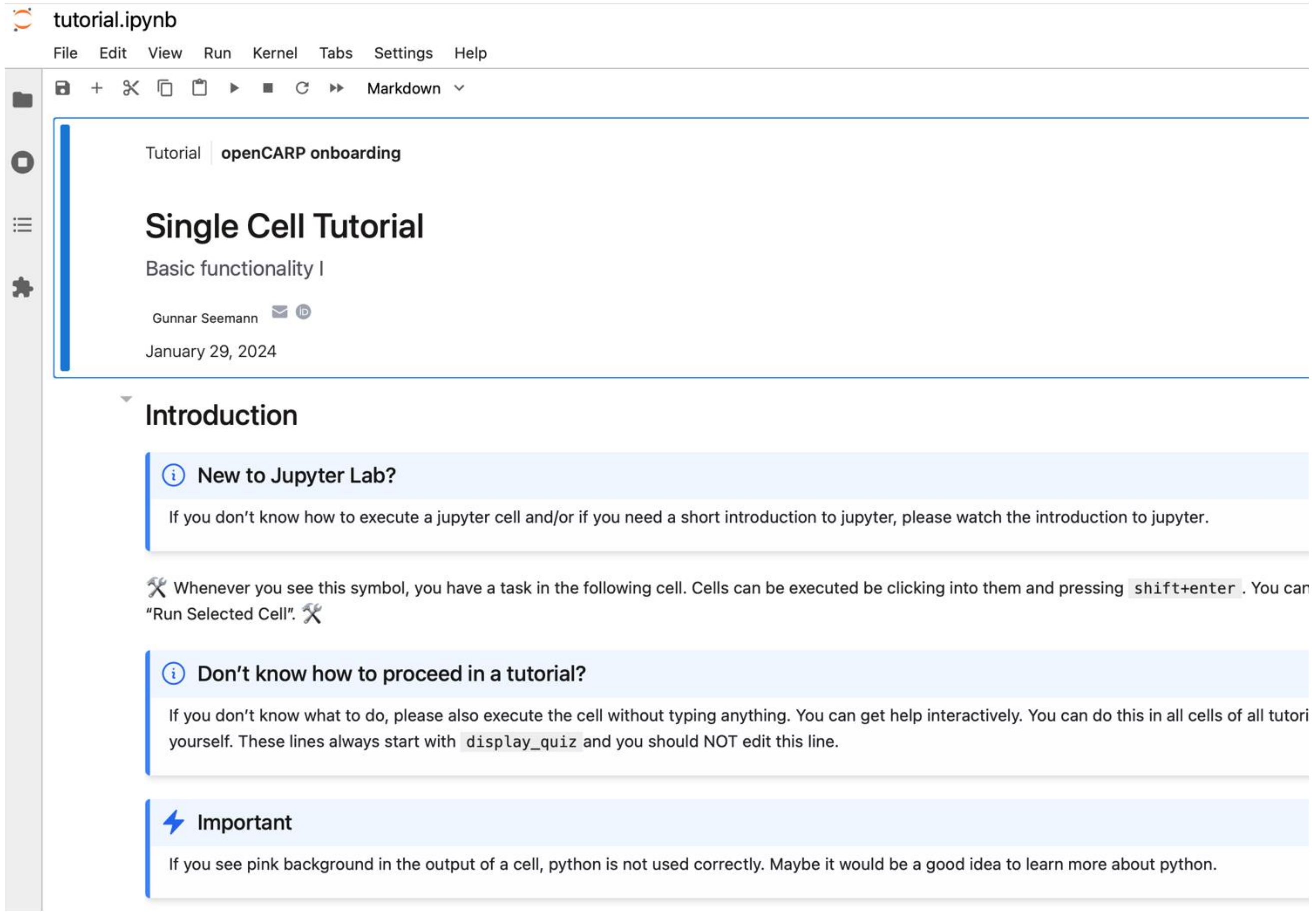
Meshalyzer
meshalyzer is a 4D visualization tool for openCARP allowing interactions to investigate data

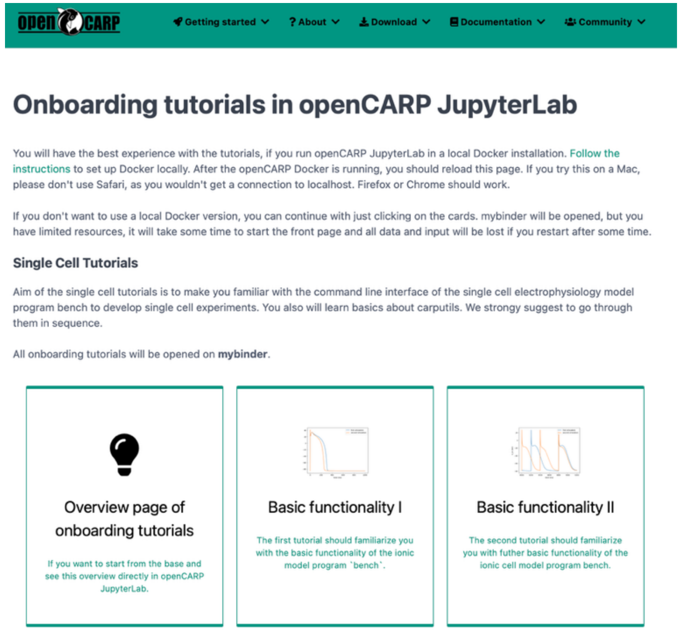
Meshalyzer II
A few advanced features of meshalyzer are explained here to make life simpler

Onboarding Tutorials

Interactive Jupyter notebooks

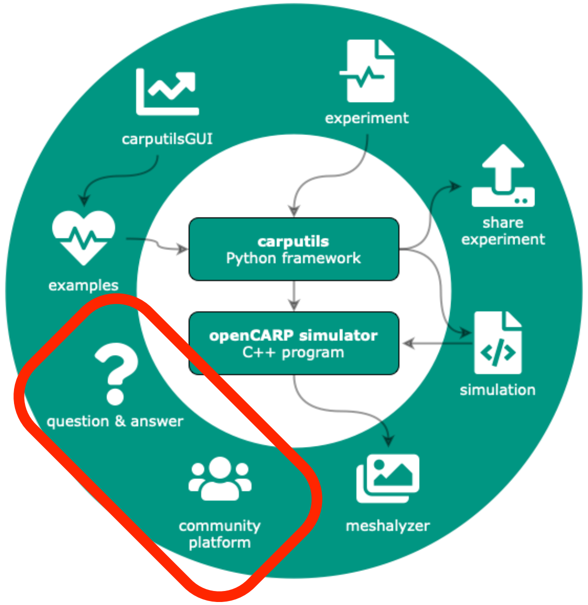
www.openCARP.org > Getting Started > Onboarding Tutorials





Community platform

www.openCARP.org > Community



Newsletter

User & contributor meetings

Q&A forum

Share experiments

Issue tracker

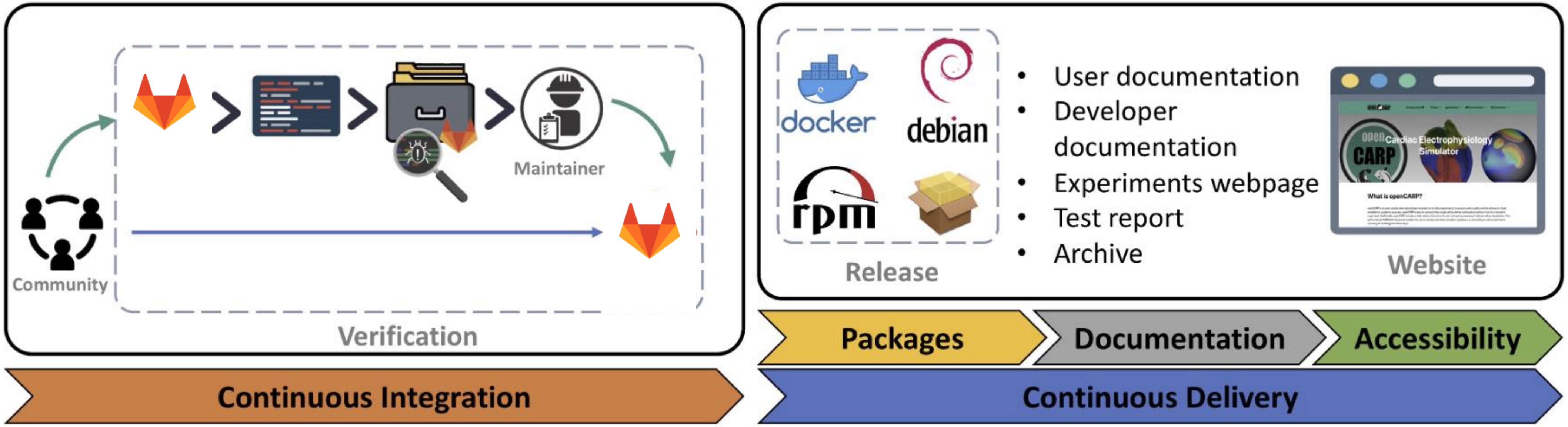
Contributing

Bach et al. "The openCARP CDE – Concept for and implementation of a sustainable collaborative development environment for research software". Bausteine Forschungsdatenmanagement 2022;2022(1):64–84. DOI: 10.17192/bfdm.2022.1.8368.

Content

1. How can mathematics and software help address heart diseases?
2. The openCARP ecosystem
3. Research Software Engineering infrastructure

Infrastructure to support maintainability and sustainability



Collaborative development environment based on Gitlab, automated integration and delivery.

Maintainability: version control, code review, automated testing, automated benchmarking, automated generation of packages, documentation & website content.

Sustainability: automated long-term preservation and citable publishing of software releases along with relevant metadata on persistent research data repository.

Bach et al. "The openCARP CDE – Concept for and implementation of a sustainable collaborative development environment for research software". Bausteine Forschungsdatenmanagement 2022;2022(1):64–84. DOI: 10.17192/bfdm.2022.1.8368.

The FAIR4RS Principles

Findable: Software, and its associated metadata, is easy for both humans and machines to find.

Accessible: Software, and its metadata, is retrievable via standardised protocols.

Interoperable: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.

Reusable: Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software).

Chue Hong et al. "FAIR Principles for Research Software (FAIR4RS Principles)". 2022. DOI:10.15497/RDA00068

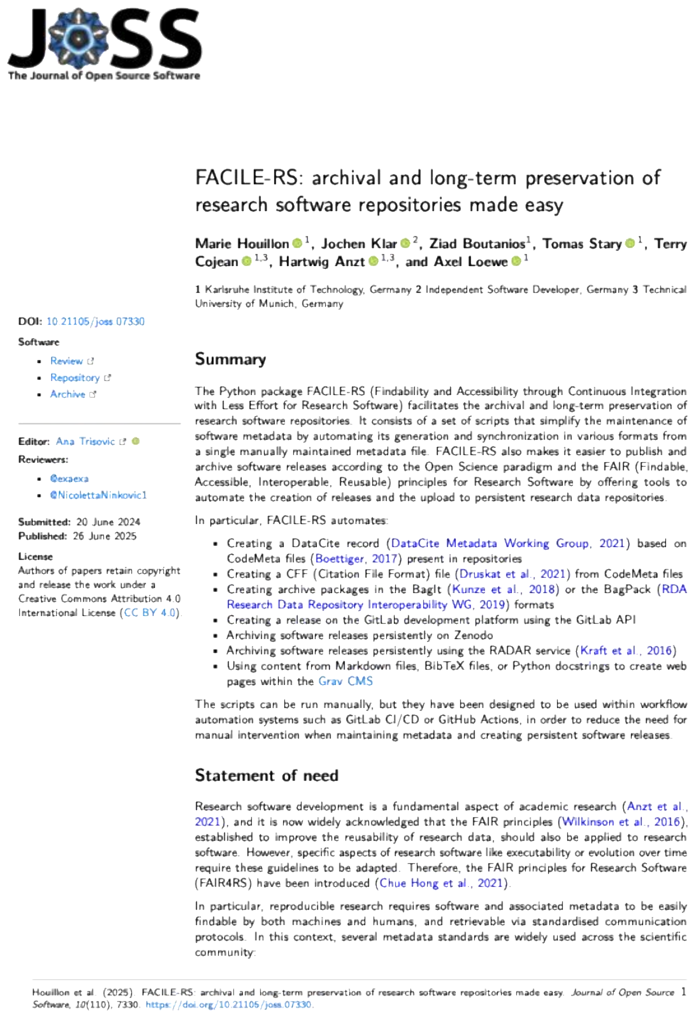
FACILE-RS: Automated Metadata Conversion and Software Publication Based on CodeMeta

Publishing software according to the **FAIR Principles for Research Software (FAIR4RS)** increases transparency, reproducibility, and reusability of research.

Adopting the FAIR4RS principles requires substantial effort from developers, including:

- maintaining software metadata in several standard formats (DataCite, Citation File Format (CFF), CodeMeta, …),
- assigning each software version a unique persistent identifier.

FACILE-RS allows to **generate various metadata formats** from CodeMeta metadata and to **publish software on reputable research data repositories** in an automated way.


JOSS
The Journal of Open Source Software

FACILE-RS: archival and long-term preservation of research software repositories made easy

Marie Houillon [1], Jochen Klar [2], Ziad Boutanios [1], Tomas Stary [1], Terry Cojean [1,3], Hartwig Anzt [1,3], and Axel Loewe [1]

1 Karlsruhe Institute of Technology, Germany 2 Independent Software Developer, Germany 3 Technical University of Munich, Germany

DOI: 10.21105/joss.07330

Software
- Review
- Repository
- Archive

Editor: Ana Trisovic

Reviewers:
- @exaexa
- @NicolettaNinkovic1

Submitted: 20 June 2024
Published: 26 June 2025

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary

The Python package FACILE-RS (Findability and Accessibility through Continuous Integration with Less Effort for Research Software) facilitates the archival and long-term preservation of research software repositories. It consists of a set of scripts that simplify the maintenance of software metadata by automating its generation and synchronization in various formats from a single manually maintained metadata file. FACILE-RS also makes it easier to publish and archive software releases according to the Open Science paradigm and the FAIR (Findable, Accessible, Interoperable, Reusable) principles for Research Software by offering tools to automate the creation of releases and the upload to persistent research data repositories.

In particular, FACILE-RS automates:

- Creating a DataCite record (DataCite Metadata Working Group, 2021) based on CodeMeta files (Boettiger, 2017) present in repositories
- Creating a CFF (Citation File Format) file (Druskat et al., 2021) from CodeMeta files
- Creating archive packages in the BagIt (Kunze et al., 2018) or the BagPack (RDA Research Data Repository Interoperability WG, 2019) formats
- Creating a release on the GitLab development platform using the GitLab API
- Archiving software releases persistently on Zenodo
- Archiving software releases persistently using the RADAR service (Kraft et al., 2016)
- Using content from Markdown files, BibTeX files, or Python docstrings to create web pages within the Grav CMS

The scripts can be run manually, but they have been designed to be used within workflow automation systems such as GitLab CI/CD or GitHub Actions, in order to reduce the need for manual intervention when maintaining metadata and creating persistent software releases.

Statement of need

Research software development is a fundamental aspect of academic research (Anzt et al., 2021), and it is now widely acknowledged that the FAIR principles (Wilkinson et al., 2016), established to improve the reusability of research data, should also be applied to research software. However, specific aspects of research software like executability or evolution over time require these guidelines to be adapted. Therefore, the FAIR principles for Research Software (FAIR4RS) have been introduced (Chue Hong et al., 2021).

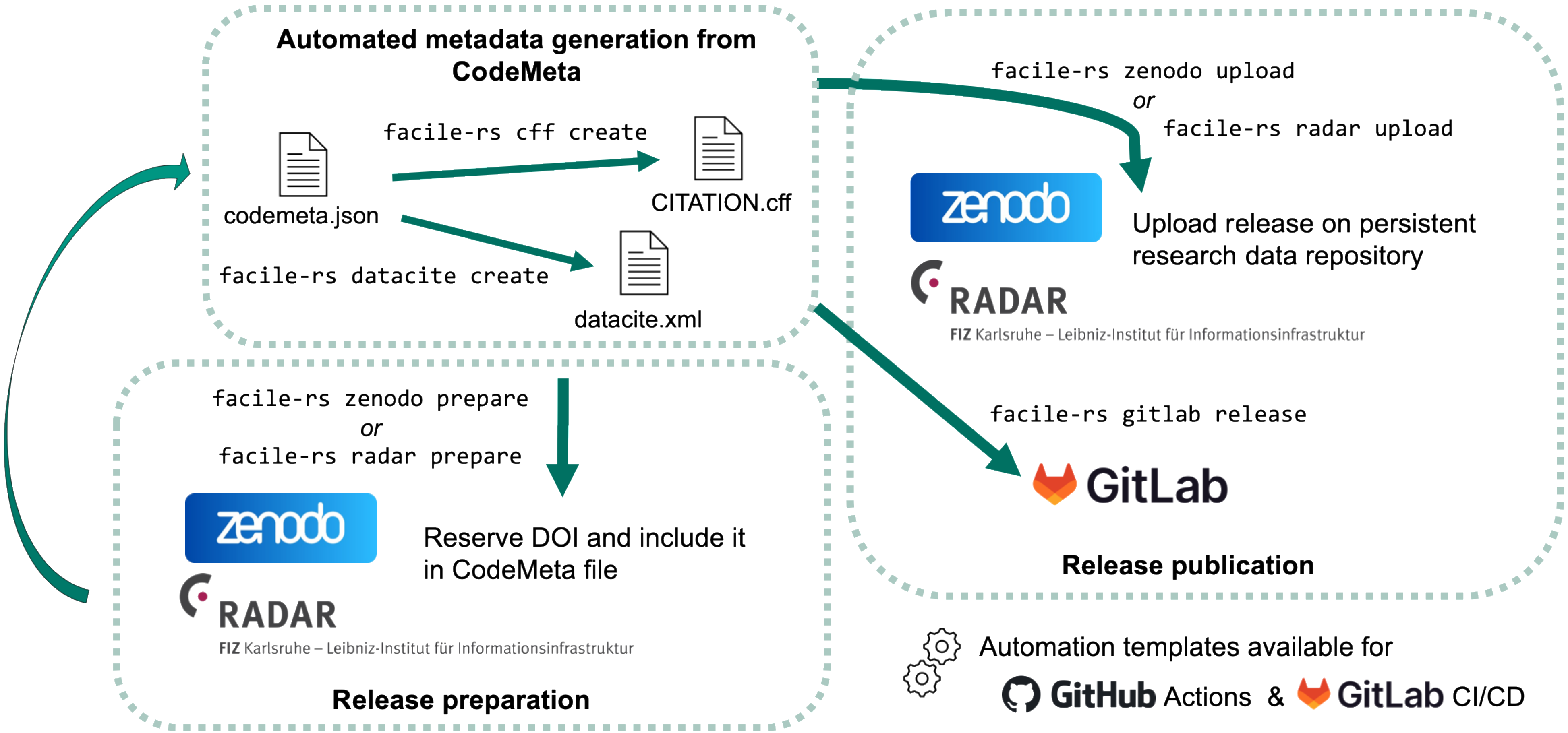
In particular, reproducible research requires software and associated metadata to be easily findable by both machines and humans, and retrievable via standardised communication protocols. In this context, several metadata standards are widely used across the scientific community:

Houillon et al. (2025). FACILE-RS: archival and long-term preservation of research software repositories made easy. *Journal of Open Source Software, 10*(110), 7330. https://doi.org/10.21105/joss.07330. 1

Houillon et al. “FACILE-RS: archival and long-term preservation of research software repositories made easy”. JOSS 2025; accepted

KIT

FACILE-RS: Findability and Accessibility through Continuous Integration with Less Effort for Research Software



Automation templates available for

GitHub Actions & GitLab CI/CD

https://git.opencarp.org/openCARP/FACILE-RS

Houillon et al. "FACILE-RS: archival and long-term preservation of research software repositories made easy". JOSS 2025; accepted

Content

1. How can mathematics and software help address heart diseases?
2. The openCARP ecosystem
3. Research Software Engineering infrastructure

Summary

Lessons learned

- **Collaboration** makes many things easier but some also more complex
- Fostering a **user community** is effort worthwhile
- **Automating** routine tasks makes life easier but also these need to be maintained
- Positive: increasing awareness for **software as research infrastructure**

Open challenges

- Uptake of best practices can be **sluggish**
- Maintaining **know-how** in teams with very few permanent members
- Legal and fiscal **sponsorship**
- **Scaling** support to a growing user base
- Tracking **usage**
- **Maintaining "smaller" software**

MICROCARD
EuroHPC Joint Undertaking
Co-funded by the European Union
DFG
Research Software Engineering (RSE) at KIT