
de*simdocumentation*

Release 1.0.5

Karr Lab

Nov 16, 2020

1	Contents	3
1.1	Getting started	3
1.1.1	Examples	3
1.1.2	Interactive tutorials	3
1.1.3	Template for models and simulations	3
1.2	Installation	4
1.2.1	Prerequisites	4
1.2.2	Latest release from PyPI	4
1.2.3	Latest release from DockerHub	4
1.2.4	Latest revision from GitHub	4
1.3	Performance	4
1.4	Comparison to other DES tools	4
1.5	About	4
1.5.1	License	4
1.5.2	Citing <i>DE-Sim</i>	5
1.5.3	Contributing to <i>DE-Sim</i>	5
1.5.4	Development team	5
1.5.5	Acknowledgements	5
1.5.6	Questions and comments	5

DE-Sim is an open-source, Python-based, object-oriented discrete-event simulation tool that helps modelers model complex systems. First, *DE-Sim* enables them to use Python's powerful object-oriented features to manage multiple types of components in a complex system and multiple types of interactions between these components. Second, by building upon Python, *DE-Sim* makes it easy for modelers to use Python's powerful data science tools, such as [NumPy](#), [Scipy](#), [pandas](#), and [SQLAlchemy](#), to incorporate large, heterogeneous datasets into comprehensive and detailed models. We anticipate that *DE-Sim* will enable a new generation of models that capture systems with unprecedented breadth and depth.

DE-Sim provides the following features to help users build and simulate complex, data-intensive models:

- **High-level, object-oriented modeling:** *DE-Sim* facilitates model designs that use classes of *simulation objects* to encapsulate the complex logic required to represent each *model component*, and use classes of *event messages* to encapsulate the logic required to describe the *interactions* between components.
- **Powerful stop conditions:** *DE-Sim* makes it easy to terminate simulations when specific criteria are reached. Modelers can specify stop conditions as functions that return true when the simulation should conclude.
- **Results checkpointing:** Models that use *DE-Sim* can record the results of simulations, and metadata such as the start and run time of each simulation, by simply configuring a checkpointing module.
- **Space-time visualizations:** *DE-Sim* can generate space-time visualizations of simulation objects and the event messages that they exchange. These diagrams can help modelers understand and debug simulations.
- **Reproducible simulations:** To help modelers debug simulations and analyze their results, repeated executions of a simulation with the same configuration and random number generator seed produce the same results.

We have used *DE-Sim* to develop [WC-Sim](#), a multi-algorithmic simulator for [whole-cell models](#).

For more information, see the [interactive DE-Sim Jupyter notebooks](#) that contain a *DE-Sim* tutorial and several example *DE-Sim* models.

1.1 Getting started

The following examples and tutorials illustrate how to use *DE-Sim* to build and simulate models.

1.1.1 Examples

- [Minimal simulation](#): a minimal example of a simulation
- [Random walk](#): a one-dimensional random walk model, with random times between steps
- [Parallel hold \(PHOLD\)](#): a model developed by Richard Fujimoto to benchmark parallel discrete-event simulators
- [Epidemic](#): two SIR models of an infectious disease epidemic

These examples have corresponding unit tests which run them in the *DE-Sim*'s [directory of unit tests of examples](#).

1.1.2 Interactive tutorials

Please see sandbox.karrlab.org for interactive Jupyter notebook tutorials about designing, building and simulating models with *DE-Sim*. It includes tutorials that use the random walk, PHOLD, and epidemic models listed above.

1.1.3 Template for models and simulations

The minimal simulation, located at [de_sim/examples/minimal_simulation.py](#), can be used as a template for implementing and simulating a model with *DE-Sim*.

1.2 Installation

1.2.1 Prerequisites

- Python ≥ 3.7
- Pip ≥ 19

1.2.2 Latest release from PyPI

Run the following command to install the latest release from PyPI:

```
pip install de_sim
```

1.2.3 Latest release from DockerHub

Run the following command to install a Docker image with the latest release from DockerHub:

```
docker pull karrlab/de_sim
```

1.2.4 Latest revision from GitHub

Run the following command to install the latest version from GitHub:

```
pip install git+https://github.com/KarrLab/de_sim.git#egg=de_sim
```

1.3 Performance

Please see Arthur P. Goldberg & Jonathan Karr. (2020). *DE-Sim: an object-oriented, discrete-event simulation tool for data-intensive modeling of complex systems in Python*. *Journal of Open Source Software*, 5(55), 2685. for information about the performance of *DE-Sim*.

1.4 Comparison to other DES tools

Please see Arthur P. Goldberg & Jonathan Karr. (2020). *DE-Sim: an object-oriented, discrete-event simulation tool for data-intensive modeling of complex systems in Python*. *Journal of Open Source Software*, 5(55), 2685. for a comparison of *DE-Sim* with other DES tools.

1.5 About

1.5.1 License

The software is released under the MIT license:

The MIT License (MIT)

Copyright (c) 2019–2020 Karr Lab

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.5.2 Citing *DE-Sim*

Please use the following reference to cite *DE-Sim*:

Arthur P. Goldberg & Jonathan Karr. (2020). *DE-Sim: an object-oriented, discrete-event simulation tool for data-intensive modeling of complex systems in Python*. *Journal of Open Source Software*, 5(55), 2685.

1.5.3 Contributing to *DE-Sim*

We enthusiastically welcome contributions to *DE-Sim*! Please see the [guide to contributing](#) and the [developer's code of conduct](#).

1.5.4 Development team

This package was developed by the [Karr Lab](#) at the Icahn School of Medicine at Mount Sinai in New York, USA by the following individuals:

- [Arthur Goldberg](#)
- [Jonathan Karr](#)

1.5.5 Acknowledgements

This work was supported by National Science Foundation award 1649014, National Institutes of Health award R35GM119771, and the Icahn Institute for Data Science and Genomic Technology.

1.5.6 Questions and comments

Please submit questions and issues to [GitHub](#) or contact the [Karr Lab](#).