



Spiking Simulation Models and Tools

A brief overview over the HBP/EBRAINS tools around spiking neural
network simulations with point-neuron resolution

by many contributors and Dennis Terhorst

SfN Global Connectome

2021-01-12

Network Level Spiking Simulations

EBRAINS Context

neuron
models

synapse
models

connectivity

simulation

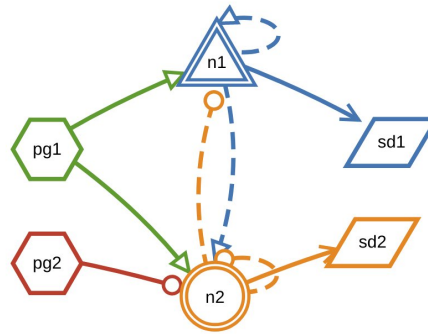
analysis &
validation

nest::ml

$$C_m \frac{dV}{dt} = -g_L(V - E_L) + g_L \Delta_T \exp\left(\frac{V - V_{th}}{\Delta_T}\right) - g_e(t)(V - E_e) - g_i(t)(V - E_i) - w + I_e$$



PyNN



Brainscales
Scales

nest::



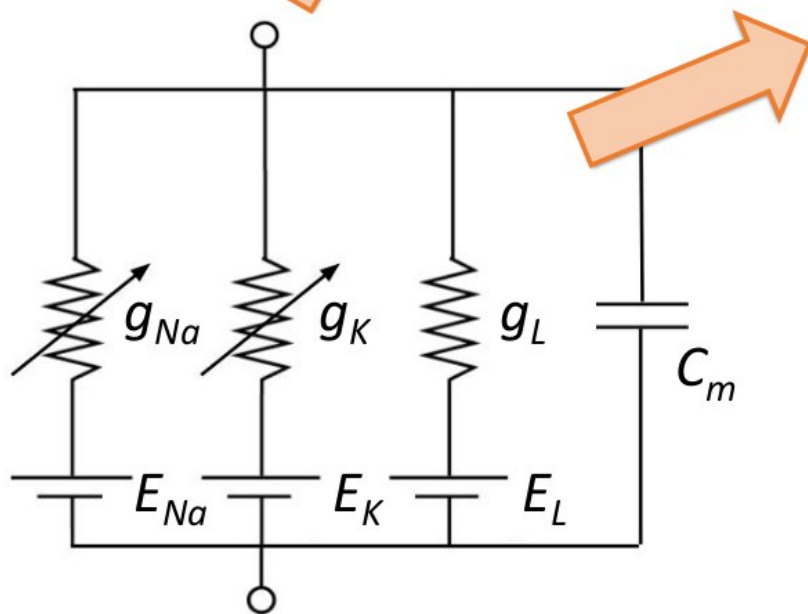
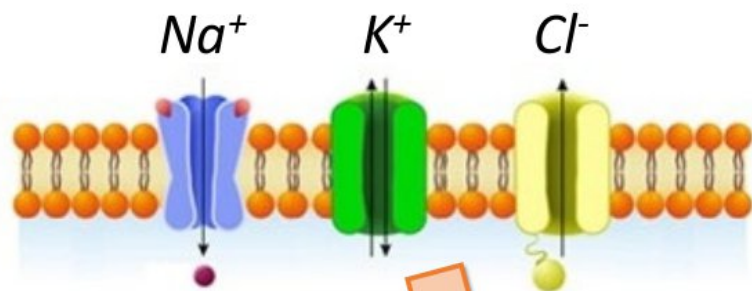
EBRAINS



NESTML is a **domain-specific modeling language** for the dynamical simulation of

- **point neurons** (spiking and rate-based), as well as
- **synapses** and
- **synaptic plasticity rules** (in alpha version).

- Low on boilerplate; concise yet expressive syntax
- Direct language support for dynamical equations
- Imperative programming style specification of event
- handling and generation



neuron hodgkin_huxley:

state:

V_m mV = -65 mV

Act_m, Act_n, Inact_h ...

end

equations:

shape syn_psc_kernel = exp(-t / tau_syn)

function I_Na pA = $g_{Na} * \text{Act}_m^{**3} * \text{Inact}_h * (V_m - E_{Na})$

function I_K pA = ...

function I_L pA = $g_L * (V_m - E_L)$

$V_m' = -(I_{Na} + I_K + I_L) / C_m$

+ convolve(syn_psc_kernel, spikes)

$\text{Act}_n' = (\alpha_n * (1 - \text{Act}_n) - \beta_n * \text{Act}_n) / \text{ms}$

$\text{Act}_m' = \dots$

$\text{Inact}_h' = \dots$

end

parameters:

C_m pF = 250 pF

$V_{\text{threshold}}$ mV = 40 mV

...

end

update:

integrate_odes()

if $V_m \geq V_{\text{threshold}}$:

emit_spike()

end

end

end

nest::ml



Code
generation



nest::

NESTML software development uses best practices in software engineering.

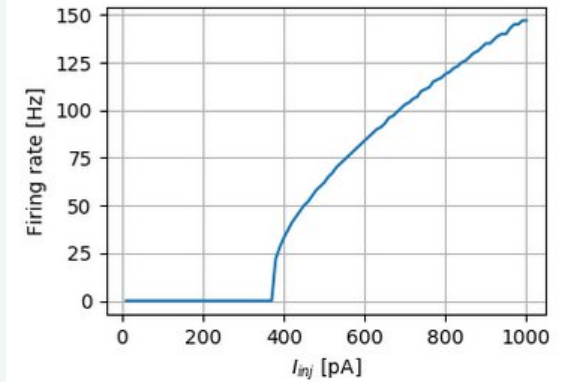
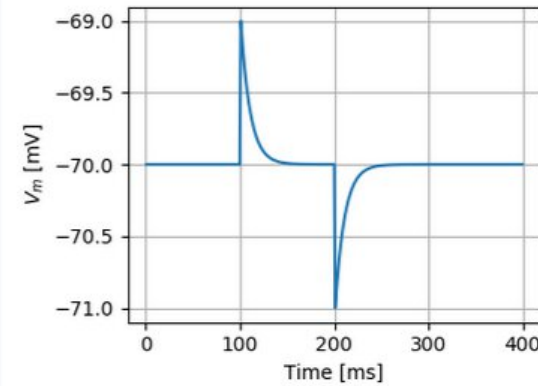
- Unit tests: language feature tests; physical units consistency; etc.
- Integration tests: models are behaviorally validated in one or more simulation runs
- Extensive documentation and automated HTML documentation generation for models: <https://nestml.readthedocs.org/>
- Open development: <https://github.com/nest/nestml>
- GNU GPL v2.0 licensed

Models library

iaf_psc_delta

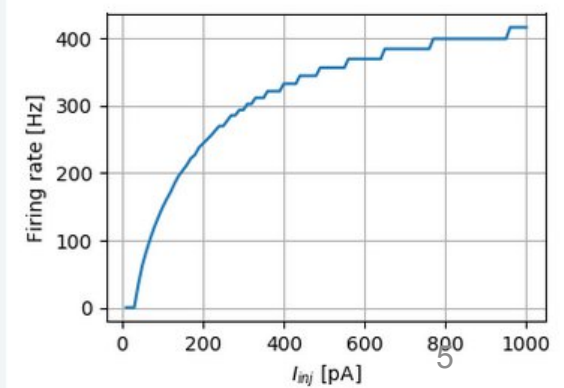
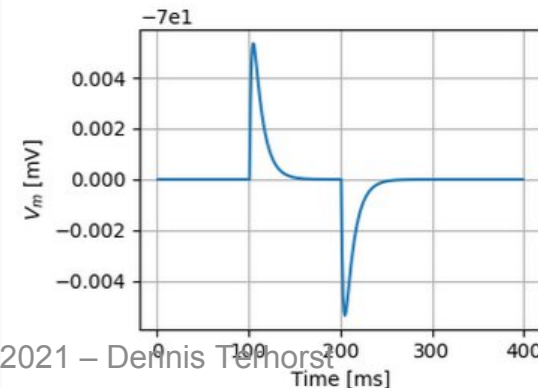
Source file: `iaf_psc_delta.nestml`

nest::ml



iaf_psc_exp

Source file: `iaf_psc_exp.nestml`



Network Level Spiking Simulations

EBRAINS Context

neuron
models

synapse
models

connectivity

simulation

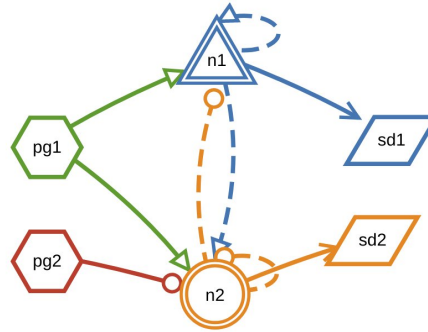
analysis &
validation

nest::ml

$$C_m \frac{dV}{dt} = -g_L(V - E_L) + g_L \Delta_T \exp\left(\frac{V - V_{th}}{\Delta_T}\right) - g_e(t)(V - E_e) - g_i(t)(V - E_i) - w + I_e$$



PyNN



Brainscales
Scales

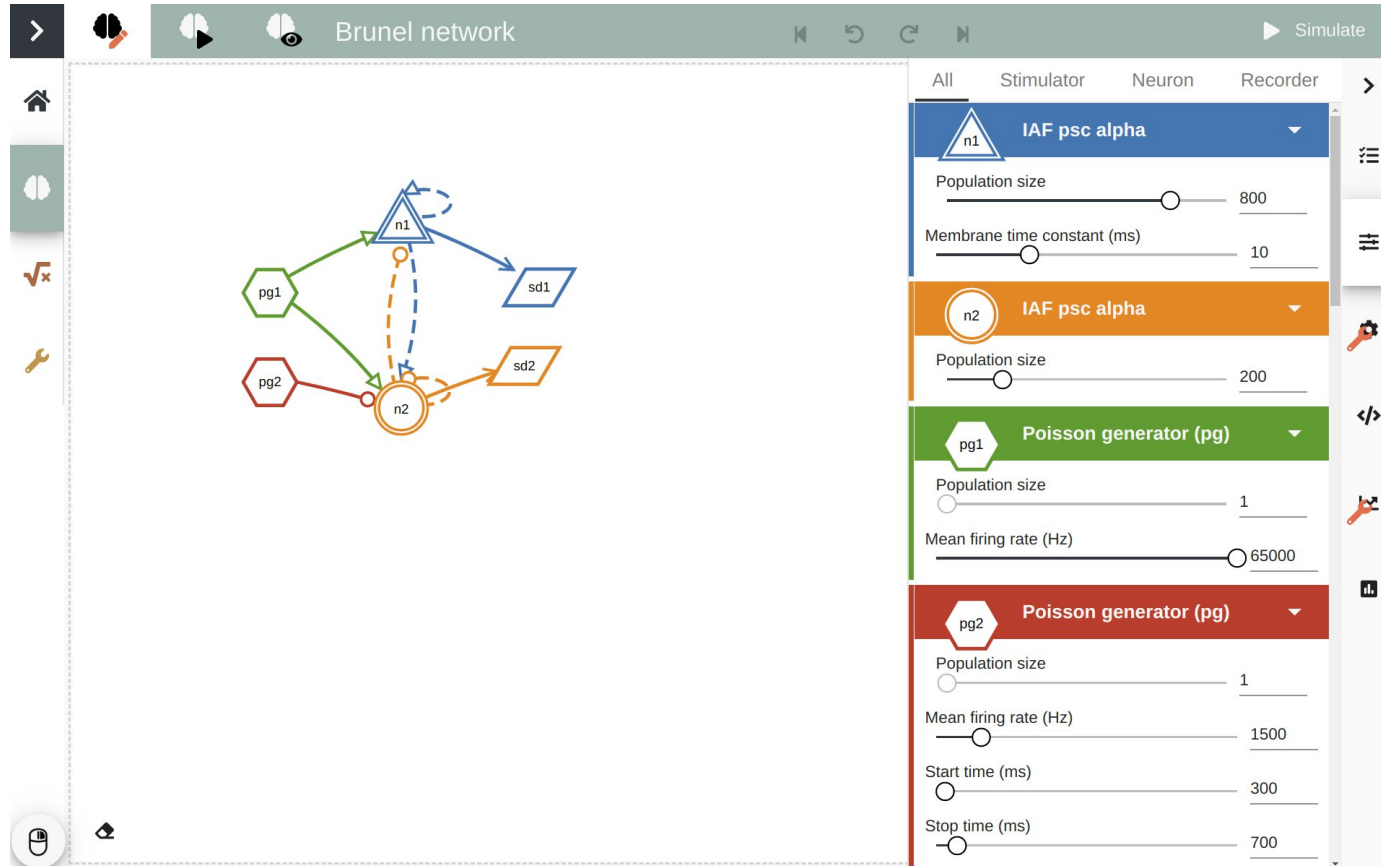
nest::



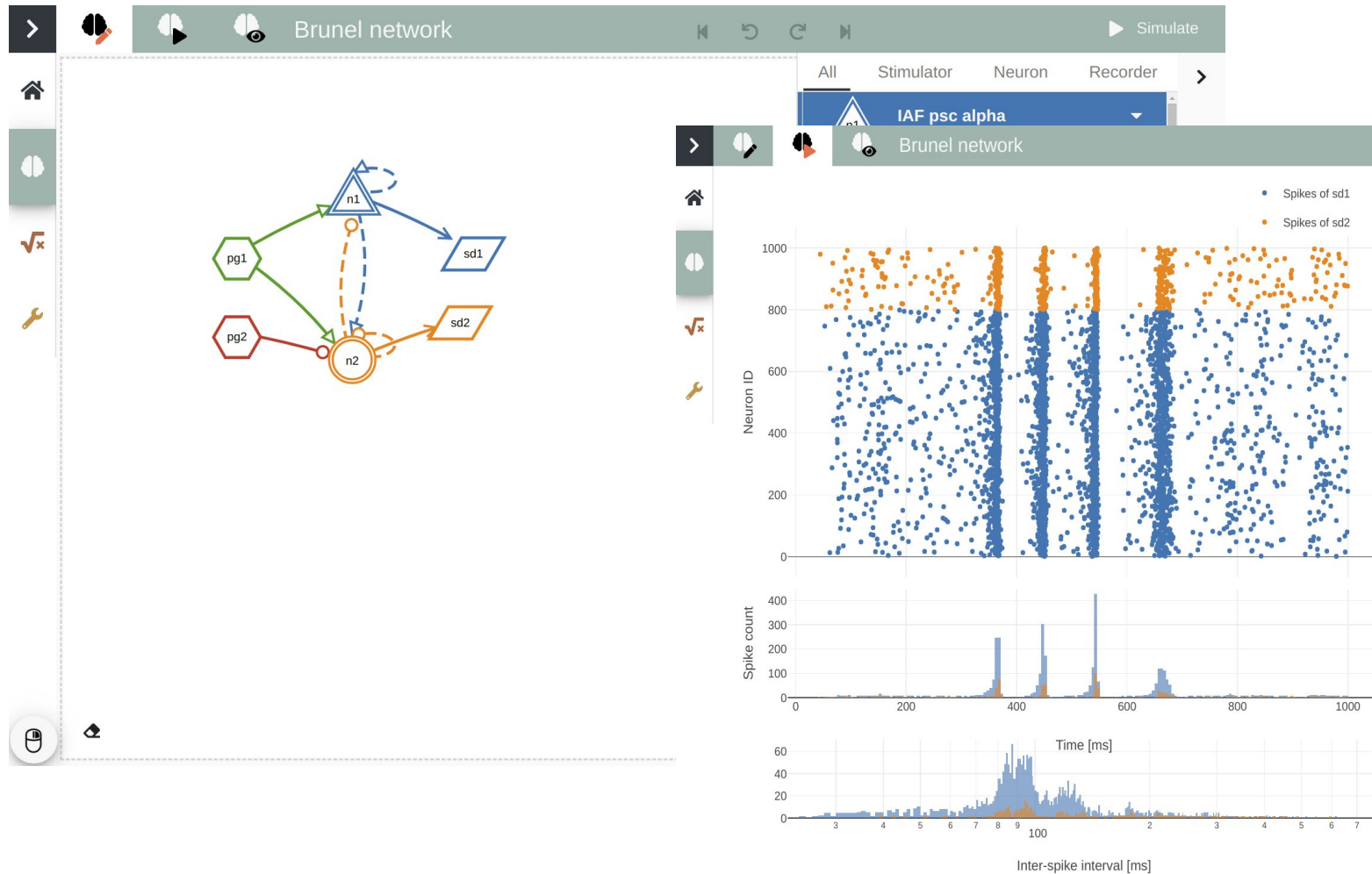
EBRAINS



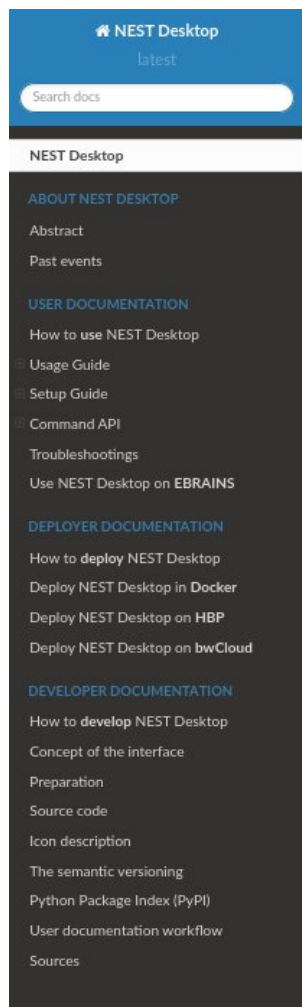
NEST Desktop



NEST Desktop



8	3	279.1	274.7	1.0
9	2	0	0	0
10	4	101.5	26.3	0.3
11	5	122.2	22.5	0.2
12	4	101.7	24.3	0.2
		$\Sigma = 3431\lambda = 117.7$	$\lambda = 49.7$	$\lambda = 0.4$



Docs » NEST Desktop

[Edit on GitHub](#)

NEST Desktop



Hello there!

NEST Desktop is a web-based GUI application for NEST Simulator, an advanced simulation tool for the computational neuroscience.

The app enables the rapid construction, parametrization, and instrumentation of neuronal network models.

It's so great that you want to use NEST Desktop.

Let's get started.

How the documentation is organized

The documentation is organized into three sections. Select the appropriate section that fits your needs.



The User

... learns how to build networks, parameterize nodes and links, and perform simulations on the graphical



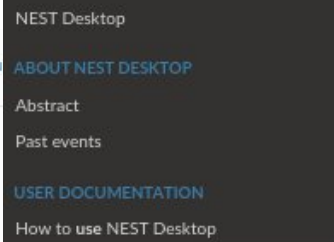
The Deployer

... learns how to set up NEST Desktop on a machine via the Python Package, Docker or Singularity installation.



The Developer

... learns the source code architecture of NEST Desktop and how to contribute code or enhancements to the



Usage Guide

Explore neuron models and devices
Manage projects
Construct neuronal networks
Simulate neuronal networks
Explore network activity

Setup Guide

Command API
Troubleshootings
Use NEST Desktop on EBRAINS

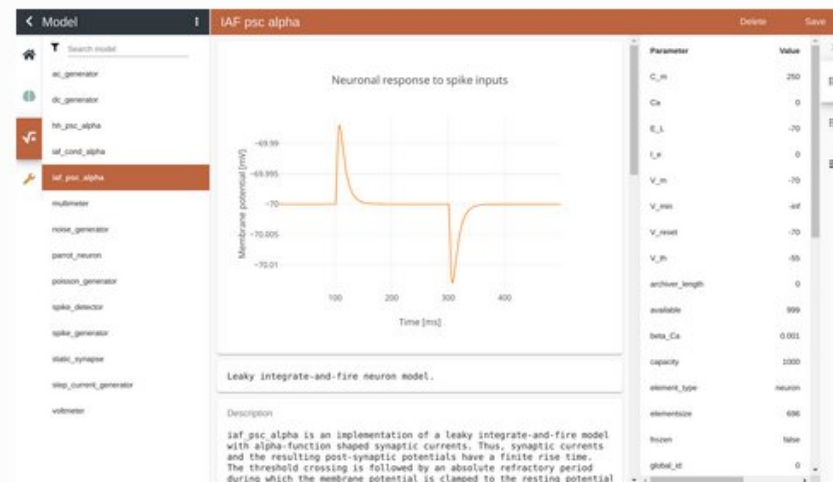
DEPLOYER DOCUMENTATION

How to deploy NEST Desktop
Deploy NEST Desktop in Docker
Deploy NEST Desktop on HBP
Deploy NEST Desktop on bwCloud

DEVELOPER DOCUMENTATION

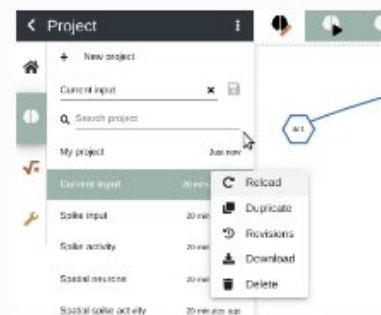
How to develop NEST Desktop
Concept of the interface
Preparation
Source code
Icon description
The semantic versioning
Python Package Index (PyPI)
User documentation workflow
Sources

Explore neuron models and devices



The model page provides you detailed documentation of models. When a selected model is a neuron, it also shows activity graph of neuronal response to excitatory and inhibitory spike inputs at 100 ms and 300 ms, respectively.

Manage projects



NEST Desktop has project management helping you to organize your networks and network activity. An important remark is that it stores only neuronal networks in database and activity will be lost after page reload. If you want to explore activity of the project, you have to start the simulation (See [Simulate neuronal networks](#)).

Clicking on a [New project](#) creates a new project where you can construct network from the scratch (See [Construct neuronal networks](#)). It is useful to give project a proper name that you can

Check out
<https://nest-desktop.readthedocs.org>

Network Level Spiking Simulations

EBRAINS Context

neuron
models

synapse
models

connectivity

simulation

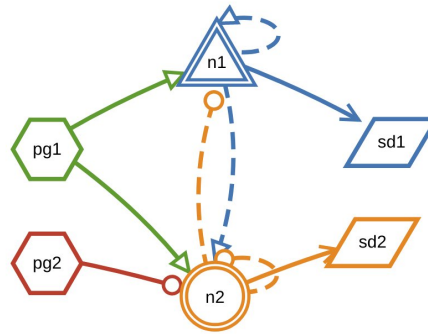
analysis &
validation

nest::ml

$$C_m \frac{dV}{dt} = -g_L(V - E_L) + g_L \Delta_T \exp\left(\frac{V - V_{th}}{\Delta_T}\right) - g_e(t)(V - E_e) - g_i(t)(V - E_i) - w + I_e$$



PyNN

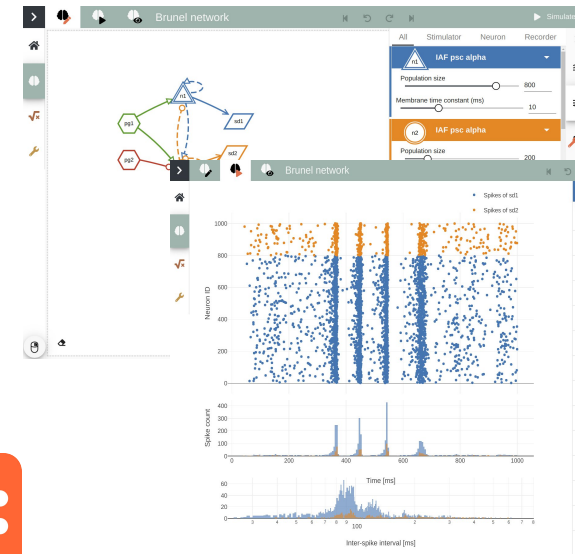


Brainscales
Scales

nest::



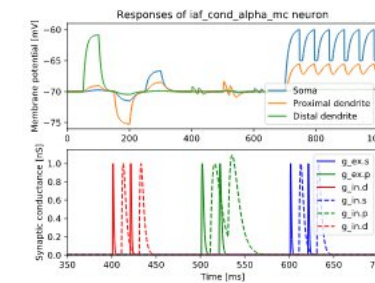
EBRAINS



NEST Simulator is a simulator for spiking neural network models that focuses on the

- dynamics,
 - size and
 - structure of neural systems
- rather than on the exact morphology of individual neurons.

multi-compartment neuron example
[...]



Download NEST

<https://nest-simulator.org/download>

NEST for your research, modify and improve it

Current Release [NEST 2.20.0 \(January 31 2020\)](#)

Tools for modern computational neuroscience

Synaptic plasticity
Topological network definition
Precise spike timing
MUSIC interface

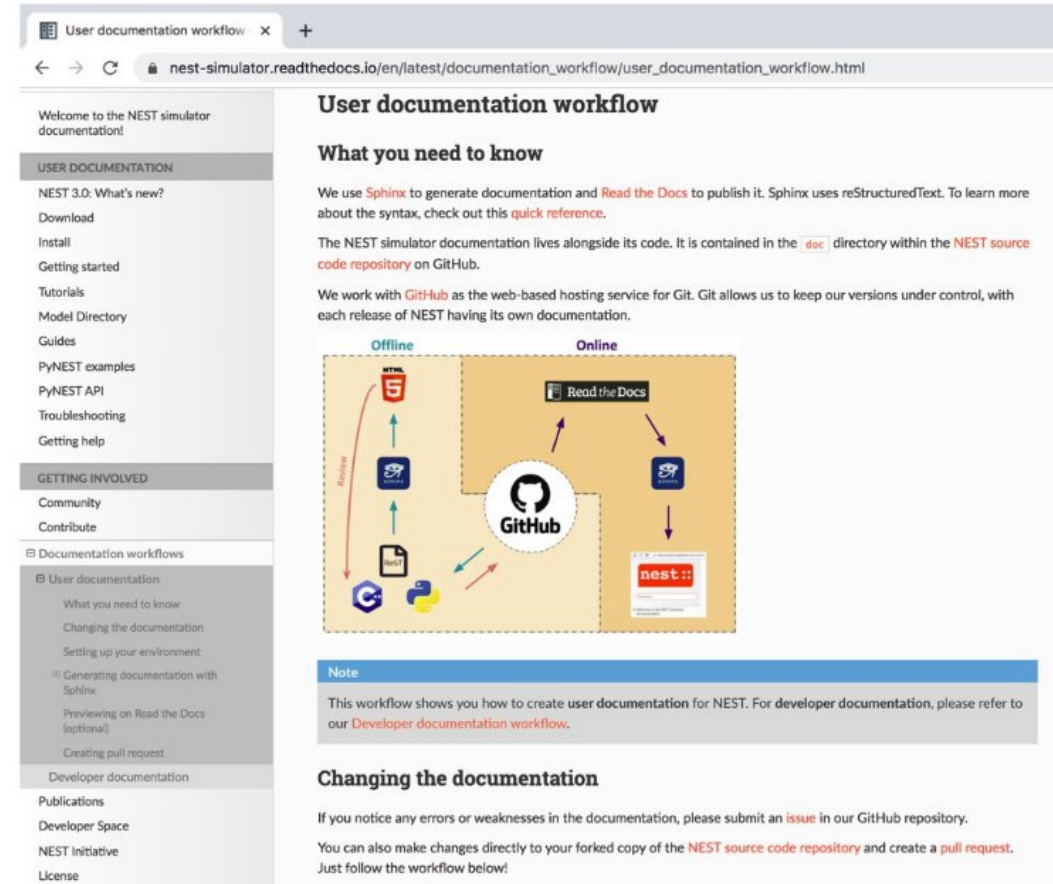
Correctness and release stability

A battery of unit tests
Continuous integration (CI) techniques
Regular open source releases under the terms of the GPL

DOCUMENTATION REQUIREMENTS AND CONCEPTS



- For the community, documentation should be easy to:
 - Find
 - Use
 - Understand
 - Write
 - Maintain



nest ::

Welcome to the NEST simulator documentation!

Introducing NEST 3.0

USER DOCUMENTATION

NEST 3.0: What's new?

Download

Install

Configure

Getting started

Tutorials

Guides

PyNEST examples

Model Directory

PyNEST API

Troubleshooting

Getting help

GETTING INVOLVED

Community

Contribute

Check out
<https://nest-simulator.readthedocs.org>

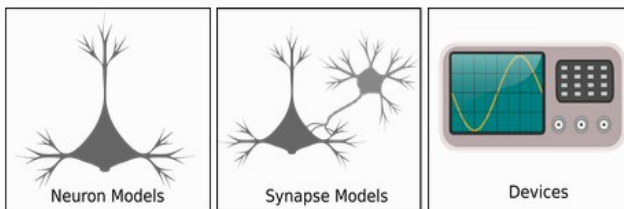
1. Models of information processing e.g., in the visual or auditory cortex of mammals,
2. Models of network activity dynamics, e.g., laminar cortical networks or balanced random networks,
3. Models of learning and plasticity.

New to NEST?

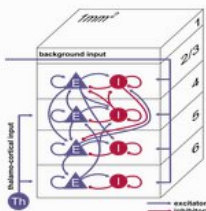
Start here at our [Getting Started](#) page

Have an idea of the type of model you need?

Click on one of the images to access our [model directory](#):



Create complex networks using the Microcircuit Model:



Need a different model?

Check out how you can create you own model here.

Have a question or issue with NEST?

See our [Getting Help](#) page.

nest ::

Welcome to the NEST simulator documentation!

USER DOCUMENTATION

NEST 3.0: What's new?

Download

Install

Configure

Getting started

Tutorials

Guides

NEST 3.0: What's new?

NEST 3.0: Detailed transition guide

Connection Management

Running simulations

Stimulating the network

Recording from simulations

multimeter – Sampling continuous quantities from neurons

spike_recorder – Collecting spikes from neurons

weight_recorder – Recording weights from synapses

Guide to parallel computing

nest ::

www.nest-simulator.org

entry in the mask dictionary. If you want to rotate the box of ellipsoidal masks, you can add an `'azimuth_angle'` entry in the specific mask dictionary for rotation from the x-axis towards the y-axis about the z-axis, or a `'polar_angle'` entry, specifying the rotation angle in degrees from the z-axis about the (possibly rotated) x axis, from the (possibly rotated) y-axis. You can specify both at once of course. If both are specified, we first rotate about the z-axis and then about the new x-axis. NEST currently does not support rotation in all three directions, the rotation from the y-axis about the (possibly rotated) z-axis, from the (possibly rotated) x-axis is missing.

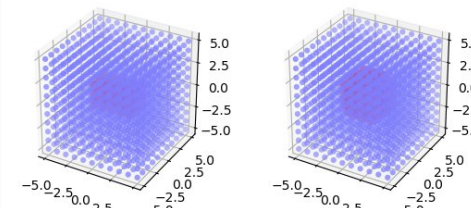


Figure 33 Masks for 3D NodeCollections. For all mask types, the driver node is marked by a wide light-red circle, the selected pool nodes by red dots and the masks are red. From left to right: box and spherical masks centered about the driver node.

Masks for grid-based layers

Grid-based layers can be connected using rectangular *grid masks*. For these, you specify the size of the mask not by lower left and upper right corner coordinates, but give their size in x and y direction, as in this example:

```
conndict = {'rule': 'pairwise_bernoulli',  
            'p': 1.0,  
            'mask': {'grid': {'shape': [5, 3]}}
```

The resulting connections are shown in Figure 34. By default the top-left corner of a grid mask, i.e., the grid mask element with grid index $[0, 0]^2$, is aligned with the driver node. You can change this alignment by specifying an *anchor* for the mask:

```
conndict = {'rule': 'pairwise_bernoulli',  
            'p': 1.0,  
            'mask': {'grid': {'shape': [5, 3],  
                              'anchor': [2, 1]}}
```

References

Visit

<https://nestml.readthedocs.org>

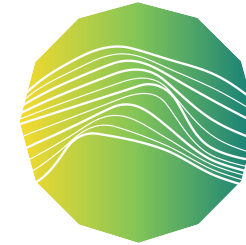
<https://nest-desktop.readthedocs.org>

<https://pynn.readthedocs.org>

<https://nest-simulator.readthedocs.org>

<https://nest-simulator.org>

Discover everything at



EBRAINS

<https://ebrains.eu>

Slides (or parts) where provided by Charl Linssen, Sara Konradi, Sebastian Spreitzer, Steffen Graber and others.