In [1]:
```
import nest
import time
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from matplotlib.gridspec import GridSpec
import numpy as np
import scipy.special as sp
import os

from pynestml.codegeneration.nest_code_generator_utils import NESTCodeGeneratorU
```

```
             -- N E S T --
 Copyright (C) 2004 The NEST Initiative

Version: 3.7.0-post0.dev0
Built: Jun 10 2024 22:16:13

This program is provided AS IS and comes with
NO WARRANTY. See the file LICENSE for details.

Problems or suggestions?
  Visit https://www.nest-simulator.org

Type 'nest.help()' to find out more about NEST.
```

```
/usr/local/lib/python3.10/dist-packages/matplotlib/projections/__init__.py:63: U
serWarning:Unable to import Axes3D. This may be due to multiple versions of Matp
lotlib being installed (e.g. as a system package and as a pip package). As a res
ult, the 3D projection is not available.
```

In [2]:
```
nestml_hawkes_model = '''
model hawkes_neuron:
    state:
        V_m mV = 0 mV                # Instantaneous rate trace (Hz)

    equations:
        kernel K_delta = delta(t)
        V_m' = -V_m / tau_m + convolve(K_delta, spikes) * (mV / ms)

    parameters:
        tau_m ms = 10 ms             # Time constant for rate trace evolution
        rbar mV = 20 mV              # Baseline rate (Hz)

    input:
        spikes <- spike

    output:
        spike

    update:
        rate mV = V_m + rbar

        # Ensure the rate is positive
        if rate > 0 mV:
            n_spikes_in_this_timestep integer = random_poisson(rate * resolution
            if n_spikes_in_this_timestep > 0:
                emit_spike()
```

```
        integrate_odes()
'''
```

In [3]:

```
# generate and build code
hawkes_module, hawkes_neuron = NESTCodeGeneratorUtils.generate_code_for(nestml_h
                                                                        #logging
                                                                        )
```

```
WARNING:root:PyGSL is not available. The stiffness test will be skipped.
WARNING:root:Error when importing: No module named 'pygsl'
               -- N E S T --
  Copyright (C) 2004 The NEST Initiative

 Version: 3.7.0-post0.dev0
 Built: Jun 10 2024 22:16:13

 This program is provided AS IS and comes with
 NO WARRANTY. See the file LICENSE for details.

 Problems or suggestions?
   Visit https://www.nest-simulator.org

 Type 'nest.help()' to find out more about NEST.

WARNING:root:Not preserving expression for variable "V_m" as it is solved by pro
pagator solver
CMake Warning (dev) at CMakeLists.txt:93 (project):
  cmake_minimum_required() should be called prior to this top-level project()
  call.  Please see the cmake-commands(7) manual for usage documentation of
  both commands.
This warning is for project developers.  Use -Wno-dev to suppress it.

-- The CXX compiler identification is GNU 11.4.0
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done

------------------------------------------------------
nestml_d8e7c71301444c7abe75b24a0fa45374_module Configuration Summary
------------------------------------------------------

C++ compiler         : /usr/bin/c++
Build static libs    : OFF
C++ compiler flags   :
NEST compiler flags  :  -std=c++17 -Wall -fopenmp  -O2 -fdiagnostics-color=auto
NEST include dirs    :  -I/opt/nest/include/nest -I/usr/include -I/usr/include -
I/usr/include -I/usr/lib/x86_64-linux-gnu/openmpi/include -I/usr/lib/x86_64-linu
x-gnu/openmpi/include/openmpi -I/usr/include
NEST libraries flags : -L/opt/nest/lib/nest -lnest -lsli /usr/lib/x86_64-linux-g
nu/libltdl.so /usr/lib/x86_64-linux-gnu/libreadline.so /usr/lib/x86_64-linux-gn
u/libncurses.so /usr/lib/x86_64-linux-gnu/libgsl.so /usr/lib/x86_64-linux-gnu/li
bgslcblas.so   /usr/lib/x86_64-linux-gnu/openmpi/lib/libmpi_cxx.so /usr/lib/x86_
64-linux-gnu/openmpi/lib/libmpi.so /usr/lib/gcc/x86_64-linux-gnu/11/libgomp.so /
usr/lib/x86_64-linux-gnu/libpthread.a
```

```
-- ------------------------------------------------------

You can now build and install 'nestml_d8e7c71301444c7abe75b24a0fa45374_module' u
sing
  make
  make install

The library file libnestml_d8e7c71301444c7abe75b24a0fa45374_module.so will be in
stalled to
  /tmp/nestml_target_d8cl16sm
The module can be loaded into NEST using
  (nestml_d8e7c71301444c7abe75b24a0fa45374_module) Install       (in SLI)
  nest.Install(nestml_d8e7c71301444c7abe75b24a0fa45374_module)   (in PyNEST)

CMake Warning (dev) in CMakeLists.txt:
  No cmake_minimum_required command is present.  A line of code such as

    cmake_minimum_required(VERSION 3.29)

  should be added at the top of the file.  The version specified may be lower
  if you wish to support older CMake versions for this project.  For more
  information run "cmake --help-policy CMP0000".
This warning is for project developers.  Use -Wno-dev to suppress it.

-- Configuring done (1.1s)
-- Generating done (0.0s)
-- Build files have been written to: /opt/data/tutorials/target
[ 66%] Building CXX object CMakeFiles/nestml_d8e7c71301444c7abe75b24a0fa45374_mo
dule_module.dir/nestml_d8e7c71301444c7abe75b24a0fa45374_module.o
[ 66%] Building CXX object CMakeFiles/nestml_d8e7c71301444c7abe75b24a0fa45374_mo
dule_module.dir/hawkes_neuron_nestml.o
/opt/data/tutorials/target/hawkes_neuron_nestml.cpp: In member function ‘void ha
wkes_neuron_nestml::init_state_internal_()’:
/opt/data/tutorials/target/hawkes_neuron_nestml.cpp:163:16: warning: unused vari
able ‘__resolution’ [-Wunused-variable]
  163 |   const double __resolution = nest::Time::get_resolution().get_ms();  //
do not remove, this is necessary for the resolution() function
      |                ^~~~~~~~~~~~
/opt/data/tutorials/target/hawkes_neuron_nestml.cpp: In member function ‘virtual
void hawkes_neuron_nestml::update(const nest::Time&, long int, long int)’:
/opt/data/tutorials/target/hawkes_neuron_nestml.cpp:237:24: warning: comparison
of integer expressions of different signedness: ‘long int’ and ‘const size_t’ {a
ka ‘const long unsigned int’} [-Wsign-compare]
  237 |     for (long i = 0; i < NUM_SPIKE_RECEPTORS; ++i)
      |                      ~~^~~~~~~~~~~~~~~~~~~~~~
/opt/data/tutorials/target/hawkes_neuron_nestml.cpp:232:10: warning: variable ‘g
et_t’ set but not used [-Wunused-but-set-variable]
  232 |     auto get_t = [origin, lag](){ return nest::Time( nest::Time::step( o
rigin.get_steps() + lag + 1) ).get_ms(); };
      |          ^~~~~
[100%] Linking CXX shared module nestml_d8e7c71301444c7abe75b24a0fa45374_module.
so
[100%] Built target nestml_d8e7c71301444c7abe75b24a0fa45374_module_module
[100%] Built target nestml_d8e7c71301444c7abe75b24a0fa45374_module_module
Install the project...
-- Install configuration: ""
-- Installing: /tmp/nestml_target_d8cl16sm/nestml_d8e7c71301444c7abe75b24a0fa453
74_module.so
```

In [4]:
```
# Get the number of logical processors
num_logical_processors = os.cpu_count()
print("Number of logical processors:", num_logical_processors)

desired_num_threads = 6
```

```
Number of logical processors: 12
```

In [5]:
```
nest.ResetKernel()

nest.SetKernelStatus({"local_num_threads": desired_num_threads})
# nest.SetKernelStatus({"verbosity": "M_DEBUG"})

nest.Install(hawkes_module)
```

```
Jun 17 04:13:57 Install [Info]:
    loaded module nestml_d8e7c71301444c7abe75b24a0fa45374_module
```

In [6]:
```
dt = 0.1  # the resolution in ms
simtime = 120.0  # Simulation time in ms
delay = dt  # synaptic delay in ms
```

In [7]:
```
g = 5.0  # ratio inhibitory weight/excitatory weight
h = 1.0  # ration of synapses from external (LGN) to excitatory/inhibitory
eta = 2.0  # external rate relative to threshold rate
epsilon = 0.1  # connection probability
```

In [8]:
```
order = 20
NE = 4 * order  # number of excitatory neurons
NI = 1 * order  # number of inhibitory neurons
Next = order
N_neurons = NE + NI  # number of neurons in total
N_rec = 20  # record from 50 neurons
```

In [9]:
```
rb_E = 25.0  # Baseline firing rate for excitatory (I)
rb_I = 25.0  # Baseline firing rate for neuron 1 (E)
rb_ext = 500

tau_m = 5.0  # Synaptic Kernel Time Constant (s)
```

In [10]:
```
excitatory_params = {
    "tau_m": tau_m,           # Time constant for rate trace evolution
    "rbar": rb_E         # Baseline rate (kHz)
}

inhibitory_params = {
    "tau_m": tau_m,           # Time constant for rate trace evolution
    "rbar": rb_I         # Baseline rate (kHz)
}
```

In [11]:
```
J = 0.1  # postsynaptic amplitude in mV
J_ex = J / N_neurons  # amplitude of excitatory postsynaptic current
J_in = -g * J_ex  # amplitude of inhibitory postsynaptic current
```

In [12]:
```
Jacobian = np.array([[-1/tau_m + J_ex, J_in, J_ex],
                     [J_ex, -1/tau_m + J_in, h*J_ex],
                     [0, 0, -1/tau_m]])

eigenvalues = np.linalg.eigvals(Jacobian)

print("Eigenvalues: ", eigenvalues)
```

```
Eigenvalues:  [-0.2   -0.204 -0.2  ]
```

In [13]:
```
nest.resolution = dt
nest.print_time = True
nest.overwrite_files = True
```

```
Jun 17 04:13:57 hawkes_neuron_nestml [Warning]:
    Simulation resolution has changed. Internal state and parameters of the
    model have been reset!

Jun 17 04:13:57 SimulationManager::set_status [Info]:
    Temporal resolution changed from 0.1 to 0.1 ms.
```

In [14]:
```
# Create neurons with specified parameters
nest.CopyModel(hawkes_neuron,
                   hawkes_neuron + "_excitatory",
                   excitatory_params
                  )

nest.CopyModel(hawkes_neuron,
                   hawkes_neuron + "_inhibitory",
                   inhibitory_params
                  )
```

In [15]:
```
nodes_ex = nest.Create(hawkes_neuron + "_excitatory", NE)
nodes_in = nest.Create(hawkes_neuron + "_inhibitory", NI)

noise_ex = nest.Create("poisson_generator", params={"rate": rb_ext})
noise_in = nest.Create("poisson_generator", params={"rate": rb_ext})

espikes = nest.Create("spike_recorder")
ispikes = nest.Create("spike_recorder")
```

In [16]:
```
nest.CopyModel("static_synapse", "excitatory", {"weight": J_ex, "delay": delay})
nest.CopyModel("static_synapse", "inhibitory", {"weight": J_in, "delay": delay})
nest.CopyModel("static_synapse", "external_to_excitatory", {"weight": J_ex, "del
nest.CopyModel("static_synapse", "external_to_inhibitory", {"weight": h*J_ex, "d
```

In [17]:
```
nest.Connect(noise_ex, nodes_ex, syn_spec="external_to_excitatory")
nest.Connect(noise_in, nodes_in, syn_spec="external_to_inhibitory")
```

In [18]:
```
nest.Connect(nodes_ex[:N_rec], espikes)
nest.Connect(nodes_in[:N_rec], ispikes)
```

In [19]:
```
conn_params_ex_to_ex = {'rule': 'pairwise_bernoulli', 'p': 0.15}
conn_params_ex_to_in = {'rule': 'pairwise_bernoulli', 'p': 0.55}
conn_params_in_to_ex = {'rule': 'pairwise_bernoulli', 'p': 0.45}
conn_params_in_to_in = {'rule': 'pairwise_bernoulli', 'p': 0.20}
```

In [20]:
```
nest.Connect(nodes_ex, nodes_ex, conn_params_ex_to_ex, "excitatory")
nest.Connect(nodes_ex, nodes_in, conn_params_ex_to_in, "excitatory")
nest.Connect(nodes_in, nodes_ex, conn_params_in_to_ex, "inhibitory")
nest.Connect(nodes_in, nodes_in, conn_params_in_to_in, "inhibitory")
```

In [21]:
```
nest.Simulate(100)
# nest.Simulate(100)
```

```
Jun 17 04:13:57 NodeManager::prepare_nodes [Info]:
    Preparing 124 nodes for simulation.

Jun 17 04:13:57 SimulationManager::start_updating_ [Info]:
    Number of local nodes: 124
    Simulation time (ms): 100
    Number of OpenMP threads: 6
    Number of MPI processes: 1

[ 100% ] Model time: 100.0 ms, Real-time factor: 0.0944 Model time: 18.0 ms, Rea
l-time factor: 0.3957

Jun 17 04:13:57 SimulationManager::run [Info]:
    Simulation finished.
```

In [22]:
```
nest.Simulate(100)
```

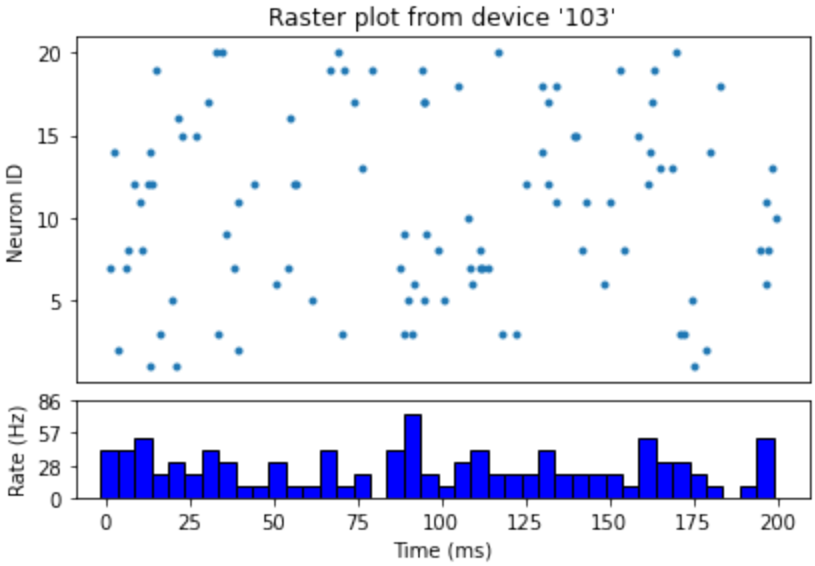
```
Jun 17 04:13:57 NodeManager::prepare_nodes [Info]:
    Preparing 124 nodes for simulation.

Jun 17 04:13:57 SimulationManager::start_updating_ [Info]:
    Number of local nodes: 124
    Simulation time (ms): 100
    Number of OpenMP threads: 6
    Number of MPI processes: 1

[ 100% ] Model time: 200.0 ms, Real-time factor: 0.0303ime factor: 0.0301

Jun 17 04:13:57 SimulationManager::run [Info]:
    Simulation finished.
```

In [23]:
```
nest.raster_plot.from_device(espikes, hist=True)
plt.show()
```


Raster plot from device '103'
Neuron ID
20
15
10
5
Rate (Hz)
86
57
28
0
0
25
50
75
100
125
150
175
200
Time (ms)

In [ ]: