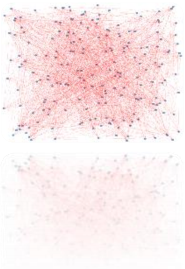
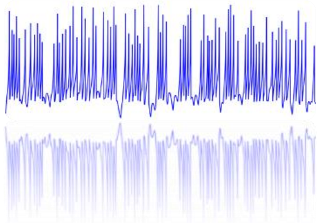




NetPyNE Parameter Overview for Izhikevich Neuron Model and
Network Simulation

Tabel :1 Izhikevich Neuron Model Parameters

Parameter	**Symbol**	**Typical Values**	**Description**
Membrane capacitance	C	1 µF/cm²	Determines how fast the neuron responds to input.
Resting potential	vr	-60 mV	The neuron's stable voltage when no input is given.
Threshold potential	vt	-40 mV	The voltage at which the neuron generates a spike.
Peak voltage	vpeak	35 mV	The highest voltage before reset occurs.
Reset potential	c	-50 mV	Voltage after a spike is completed.
Recovery increment	d	100 pA	The amount by which recovery variable u is increased after a spike.
Adaptation time scale	a	0.03	Controls the speed of recovery variable u.
Sensitivity of u to v	b	-2	Determines how strongly u is coupled to v.
Scaling factor for v	k	0.7 ± 0.05	Modifies the influence of voltage.
Cell type identifier	celltype	1	Defines neuron behavior (e.g., regular spiking, bursting, etc.).

single_izhikevich.py

```python
from netpyne import specs, sim

# Network parameters
netParams = specs.NetParams()  # object of class NetParams to store the network parameters

## Define a single Izhikevich Neuron Model
IZH_Neuron = {'secs': {}}
IZH_Neuron['secs']['soma'] = {'geom': {}, 'pointps': {}}
IZH_Neuron['secs']['soma']['geom'] = {'diam': 30.0, 'L': 10.0, 'cm': 31.831}  # Soma geometry
IZH_Neuron['secs']['soma']['pointps']['Izhi'] = {  # Izhikevich neuron properties
    'mod': 'Izhi2007b',
    'C': 1,
    'k': 'normal(0.7, 0.05)',
    'vr': -60,
    'vt': -40,
    'vpeak': 35,
    'a': 0.03,
    'b': -2,
    'c': -50,
    'd': 100,
    'celltype': 1}
netParams.cellParams['IZH'] = IZH_Neuron  # Add Izhikevich neuron to network

## Step 1: Single Izhikevich Neuron
netParams.popParams['SingleIZH'] = {'cellType': 'IZH', 'numCells': 1}

# Stimulation for single neuron
netParams.stimSourceParams['IClamp'] = {'type': 'IClamp', 'del': 50, 'dur': 1000, 'amp': 10}
netParams.stimTargetParams['IClamp->SingleIZH'] = {'source': 'IClamp', 'conds': {'pop': 'SingleIZH'}, 'sec': 'soma', 'loc': 0.5}

# Simulation Configuration
simConfig = specs.SimConfig()
simConfig.duration = 1000  # Simulate 1 second
simConfig.dt = 0.025  # Internal integration timestep
simConfig.verbose = False
simConfig.recordTraces = {'V_soma': {'sec': 'soma', 'loc': 0.5, 'var': 'v'}}
simConfig.recordStep = 1  # Save data every 1ms
simConfig.filename = 'Single_Izhikevich_Neuron'
simConfig.savePickle = False

# Analysis and plotting
simConfig.analysis['plotTraces'] = {'include': [0], 'saveFig': True}  # Voltage traces
simConfig.analysis['plotRaster'] = {'saveFig': True}  # Raster plot
simConfig.analysis['plot2Dnet'] = {'saveFig': True}  # 2D visualization of neuron location

# Run Simulation
sim.createSimulateAnalyze(netParams=netParams, simConfig=simConfig)
```

Table: 2 NetPyNE Components

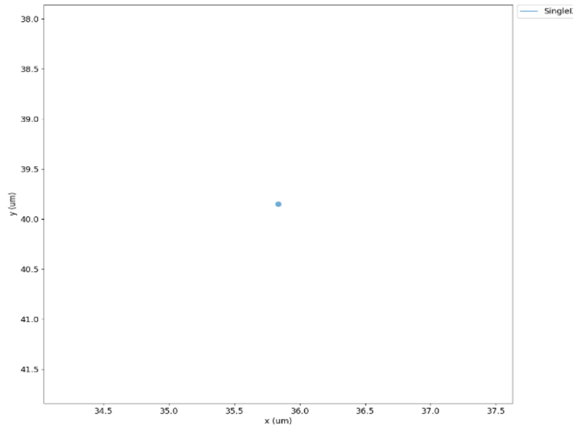
Component	Usage in Code	Purpose
specs	from netpyne import specs	Defines network structure and parameters.
sim	from netpyne import sim	Runs the simulation and analyzes results.
NetParams()	netParams = specs.NetParams()	Creates a network parameter object to store neuron and synapse properties.
cellParams	netParams.cellParams['IZH'] = IZH_Neuron	Stores the definition of neuron models (geometry, dynamics, etc.).
popParams	netParams.popParams['SingleIZH'] = {'cellType': 'IZH', 'numCells': 1}	Defines neuron populations (how many neurons and which type).

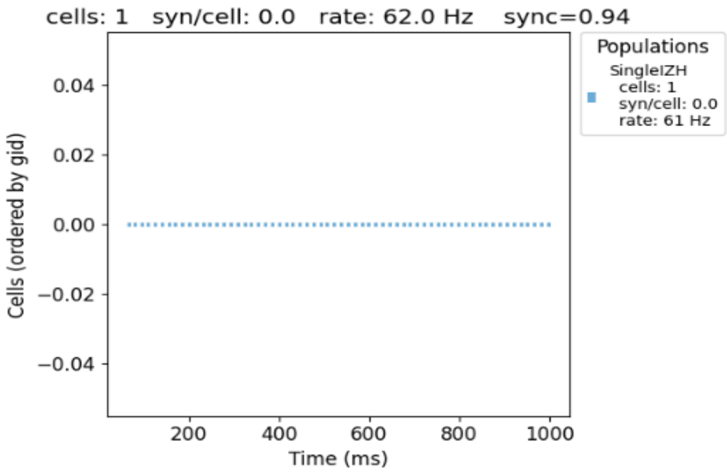
stimSourceParams	netParams.stimSourceParams['IClamp'] = {'type': 'IClamp', 'amp': 10}	Defines external current (e.g., IClamp) or other inputs to neurons.
stimTargetParams	netParams.stimTargetParams['IClamp->SingleIZH'] = {'source': 'IClamp', 'conds': {'pop': 'SingleIZH'}}	Applies external stimuli to a specific neuron population.
connParams	netParams.connParams['IZH->IZH'] = {'preConds': {'pop': 'IZHpop'}, 'postConds': {'pop': 'IZHpop'}, 'probability': 0.1}	Defines connectivity between neurons, probability of connection, and synaptic mechanisms.
synMechParams	netParams.synMechParams['exc'] = {'mod': 'Exp2Syn', 'tau1': 1.0, 'tau2': 5.0, 'e': 0}	Defines synaptic properties (excitation and inhibition).
SimConfig()	simConfig = specs.SimConfig()	Stores all simulation settings (duration, time step, recording options).
recordTraces	simConfig.recordTraces = {'V_soma': {'sec': 'soma', 'loc': 0.5, 'var': 'v'}}	Records membrane potential from specific neuron sections.
plotTraces	simConfig.analysis['plotTraces'] = {'include': [0], 'saveFig': True}	Plots voltage traces of neurons.
plotRaster	simConfig.analysis['plotRaster'] = {'saveFig': True}	Generates a raster plot showing neuron firing times.
plot2Dnet	simConfig.analysis['plot2Dnet'] = {'saveFig': True}	Plots the neuron network layout.
createSimulateAnalyze()	sim.createSimulateAnalyze(netParams=netParams, simConfig=simConfig)	Runs the entire simulation, records data, and generates results.

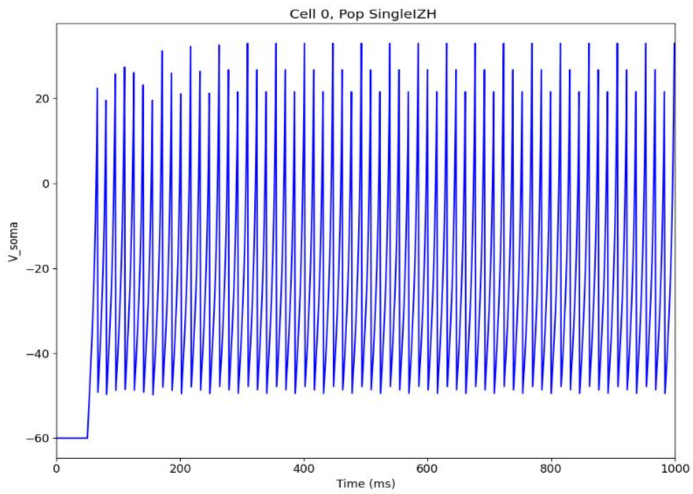
Table: 3 Common Modifications & Their Effect

Modification	**How to Change It?**	**Effect on Simulation**
Increase Excitation	'e': 10 in 'synMechParams'	More neuron firing, more activity.
Increase Inhibition	'e': -80 in 'synMechParams'	Stronger suppression, fewer spikes.
Increase Network Size	numCells: 50 instead of 20	More complex network, longer runtime.
Increase Input Current	'amp': 20 instead of 10	Neuron fires more frequently.
Increase Connection Probability	'probability': 0.5 instead of 0.1	More synaptic connections, more interaction.

Results: single_izhikevich.py







izhikevich_network.py

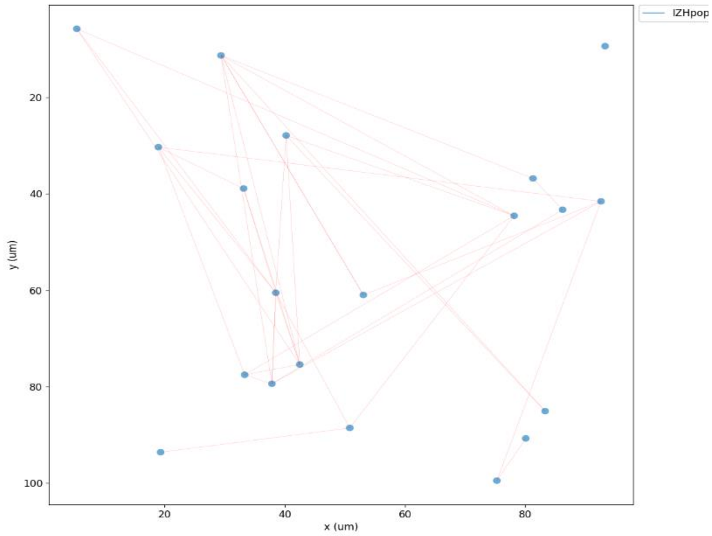
```python
# from netpyne import specs, sim
from netpyne import specs, sim

# Network parameters
netParams = specs.NetParams()  # object of class NetParams to store the network parameters

## Define Izhikevich Neuron Model
IZH_Neuron = {'secs': {}}
IZH_Neuron['secs']['soma'] = {'geom': {}, 'pointps': {}}
IZH_Neuron['secs']['soma']['geom'] = {'diam': 10.0, 'L': 10.0, 'cm': 31.831}  # Soma geometry
IZH_Neuron['secs']['soma']['pointps']['Izhi'] = {  # Izhikevich neuron properties
    'mod': 'Izhi2007b',
    'C': 1,
    'k': 'normal(0.7, 0.05)',
    'vr': -60,
    'vt': -40,
    'vpeak': 35,
    'a': 0.03,
    'b': -2,
    'c': -50,
    'd': 100,
    'celltype': 1}
netParams.cellParams['IZH'] = IZH_Neuron  # Add Izhikevich neuron to network

## Step 2: Network of 20 Izhikevich Neurons
netParams.popParams['IZHpop'] = {'cellType': 'IZH', 'numCells': 20}

## Synaptic Mechanism
netParams.synMechParams['exc'] = {'mod': 'Exp2Syn', 'tau1': 1.0, 'tau2': 5.0, 'e': 0}  # Excitatory synapse

# Stimulation parameters
netParams.stimSourceParams['bkg'] = {'type': 'NetStim', 'rate': 100, 'noise': 0.5}
netParams.stimTargetParams['bkg->IZH'] = {'source': 'bkg', 'conds': {'cellType': 'IZH'}, 'weight': 0.01, 'delay': 5, 'synMech': 'exc'}

# Connectivity rules within Izhikevich population
netParams.connParams['IZH->IZH'] = {
    'preConds': {'pop': 'IZHpop'}, 'postConds': {'pop': 'IZHpop'},
    'probability': 0.1, 'weight': 0.005, 'delay': 5, 'synMech': 'exc'}

# Simulation Configuration
simConfig = specs.SimConfig()
simConfig.duration = 1000  # Simulate 1 second
simConfig.dt = 0.025  # Internal integration timestep
simConfig.verbose = False
simConfig.recordTraces = {'V_soma': {'sec': 'soma', 'loc': 0.5, 'var': 'v'}}
simConfig.recordStep = 1  # Save data every 1ms
simConfig.filename = 'Izhikevich_Network'
simConfig.savePickle = False

# Analysis and plotting
simConfig.analysis['plotRaster'] = {'saveFig': True}  # Raster plot
simConfig.analysis['plotTraces'] = {'include': [1], 'saveFig': True}  # Voltage traces
simConfig.analysis['plot2Dnet'] = {'saveFig': True}  # 2D network visualization

# Run Simulation
sim.createSimulateAnalyze(netParams=netParams, simConfig=simConfig)
```

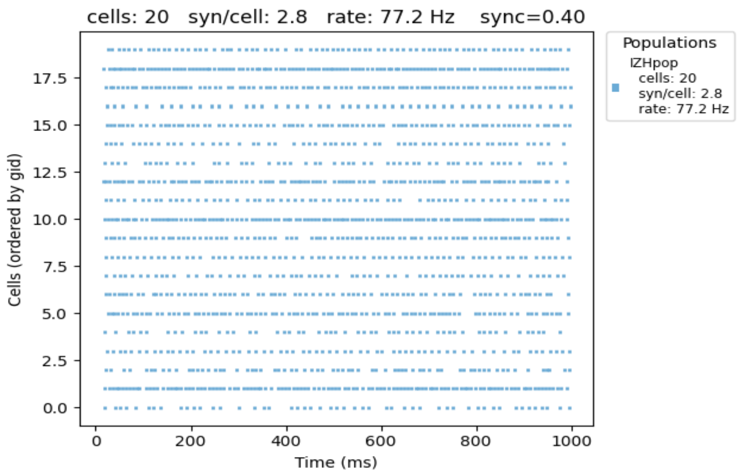
Table 4: Exercise 2

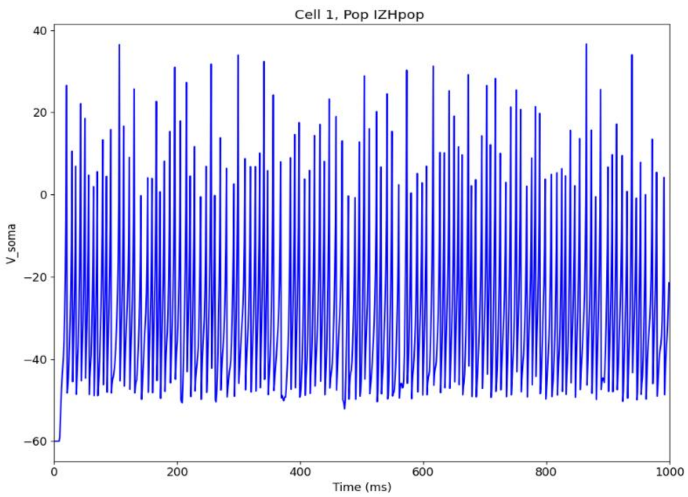
Parameter	Symbol	Typical Value	Description
Number of Neurons	numCells	20	Defines the number of neurons in the network.
Synaptic Mechanism	synMechParams	Exp2Syn	Defines the type of synapse used in the network.
Rise time constant	tau1	1.0 ms	time for the synaptic current to reach its peak.(how quickly the synapse is activated.)

Decay time constant	tau2	5.0 ms	time for the synaptic current to decay (how long the synapse remains active)
Excitatory Reversal Potential	e	0 mV	Determines if the synapse is excitatory (0 mV) or inhibitory (-70 mV).
Synaptic Weight	weight	0.005	Defines the strength of the synapse.
Connection Probability	probability	0.1	Determines how likely two neurons will be connected.
Synaptic Delay	delay	5 ms	Defines the transmission delay of synapses.
Neuron Stimulation	stimSourceParams	NetStim	Defines external stimulation using a Poisson spike generator. (100 Hz neurons receive an average of 100 input spikes per second.)
Stimulation Rate	rate	100 Hz	Frequency of external input stimulation.
Noise in Stimulation	noise	0.5	Adds variability to the external input pattern.
Simulation Duration	duration	1000 ms	Defines how long the network will be simulated.
Integration Time Step	dt	0.025 ms	Defines the resolution of the simulation steps.
Recorded Variable	recordTraces	V_soma	Records the membrane potential of the neurons.
Raster Plot	plotRaster	True	Generates a raster plot to visualize spikes.
Voltage Traces	plotTraces	True	Plots the voltage traces of neurons.
2D Network Visualization	plot2Dnet	True	Plots the network connectivity in 2D.

Results: izhikevich_network.py







izhikevich_inhibitory_network.py

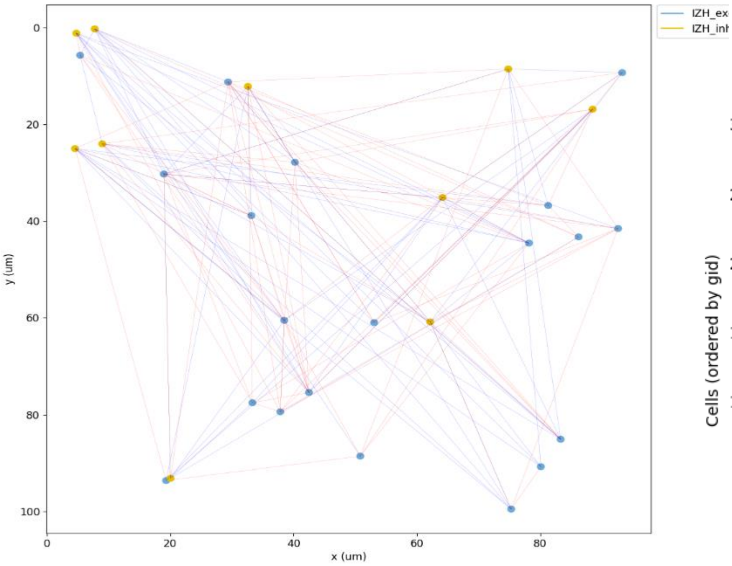
```python
from netpyne import specs, sim

# Network parameters
netParams = specs.NetParams()  # object of class NetParams to store the network parameters

## Define Izhikevich Neuron Model
IZH_Neuron = {'secs': {}}
IZH_Neuron['secs']['soma'] = {'geom': {}, 'pointps': {}}
IZH_Neuron['secs']['soma']['geom'] = {'diam': 10.0, 'L': 10.0, 'cm': 31.831}  # Soma geometry
IZH_Neuron['secs']['soma']['pointps']['Izhi'] = {  # Izhikevich neuron properties
    'mod': 'Izhi2007b',
    'C': 1,
    'k': 'normal(0.7, 0.05)',
    'vr': -60,
    'vt': -40,
    'vpeak': 35,
    'a': 0.03,
    'b': -2,
    'c': -50,
    'd': 100,
    'celltype': 1}
netParams.cellParams['IZH'] = IZH_Neuron  # Add Izhikevich neuron to network

## Step 3: Add Inhibitory Neurons
netParams.popParams['IZH_exc'] = {'cellType': 'IZH', 'numCells': 20}  # Excitatory population
netParams.popParams['IZH_inh'] = {'cellType': 'IZH', 'numCells': 10}  # Inhibitory population

## Synaptic Mechanism
netParams.synMechParams['exc'] = {'mod': 'Exp2Syn', 'tau1': 1.0, 'tau2': 5.0, 'e': 10}  # Excitatory synapse
netParams.synMechParams['inh'] = {'mod': 'Exp2Syn', 'tau1': 2.0, 'tau2': 10.0, 'e': -70}  # Inhibitory synapse

# Stimulation parameters
netParams.stimSourceParams['bkg'] = {'type': 'NetStim', 'rate': 100, 'noise': 0.5}
netParams.stimTargetParams['bkg->IZH_exc'] = {'source': 'bkg', 'conds': {'pop': 'IZH_exc'}, 'weight': 0.01, 'delay': 5, 'synMech': 'exc'}

# Connectivity rules within Izhikevich population
netParams.connParams['IZH_exc->IZH_exc'] = {
    'preConds': {'pop': 'IZH_exc'}, 'postConds': {'pop': 'IZH_exc'},
    'probability': 0.1, 'weight': 0.005, 'delay': 5, 'synMech': 'exc'}

netParams.connParams['IZH_exc->IZH_inh'] = {
    'preConds': {'pop': 'IZH_exc'}, 'postConds': {'pop': 'IZH_inh'},
    'probability': 0.2, 'weight': 0.005, 'delay': 5, 'synMech': 'exc'}

netParams.connParams['IZH_inh->IZH_exc'] = {
    'preConds': {'pop': 'IZH_inh'}, 'postConds': {'pop': 'IZH_exc'},
    'probability': 0.3, 'weight': 0.007, 'delay': 3, 'synMech': 'inh'}

# Simulation Configuration
simConfig = specs.SimConfig()
simConfig.duration = 1000  # Simulate 1 second
simConfig.dt = 0.025  # Internal integration timestep
simConfig.verbose = False
simConfig.recordTraces = {'V_soma': {'sec': 'soma', 'loc': 0.5, 'var': 'v'}}
simConfig.recordStep = 1  # Save data every 1ms
simConfig.filename = 'Izhikevich_Inhibitory_Network'
simConfig.savePickle = False

# Analysis and plotting
simConfig.analysis['plotRaster'] = {'saveFig': True}  # Raster plot
simConfig.analysis['plotTraces'] = {'include': [1], 'saveFig': True}  # Voltage traces
simConfig.analysis['plot2Dnet'] = {'saveFig': True}  # 2D network visualization
```

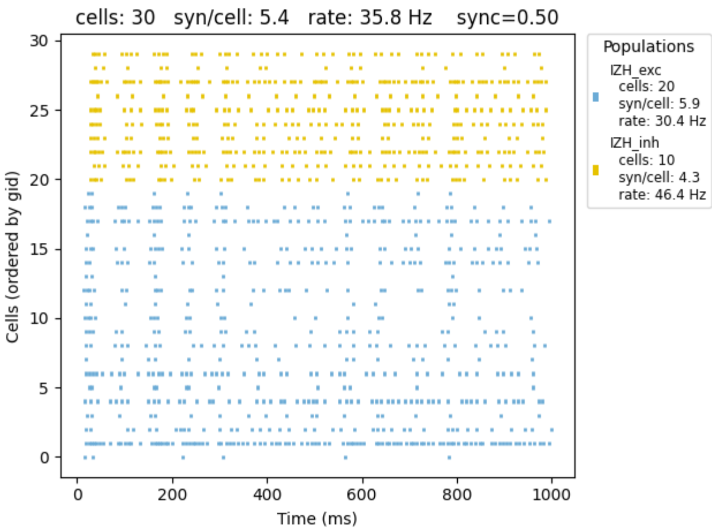
Table 5: Exercise 3

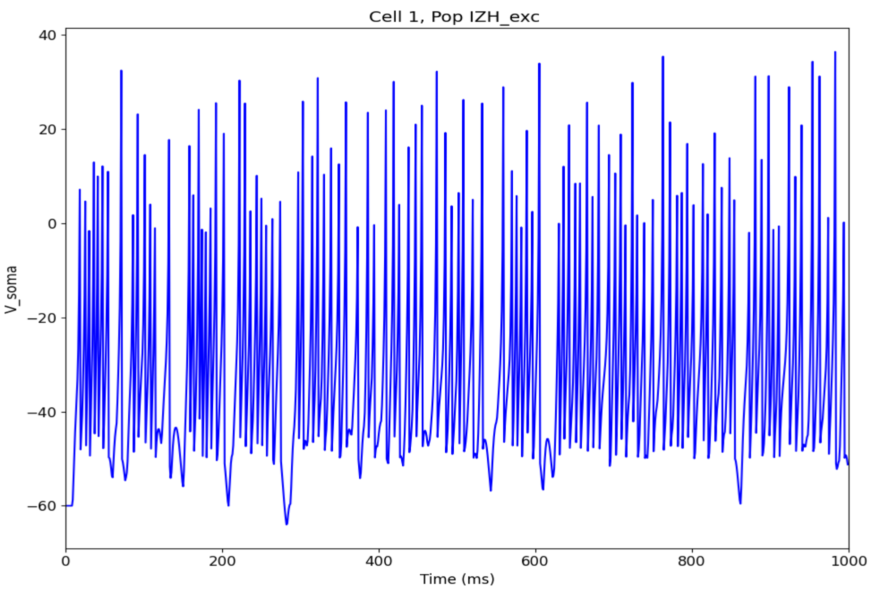
Parameter	Symbol	Value	Description
Number of Excitatory Neurons	numCells (exc)	20	Number of excitatory neurons in the network
Number of Inhibitory Neurons	numCells (inh)	10	Number of inhibitory neurons in the network
Excitatory Synapse Type	mod (exc)	Exp2Syn	Synaptic mechanism for excitation

Excitatory Reversal Potential	e (exc)	10	Reversal potential for excitatory synapses
Inhibitory Synapse Type	mod (inh)	Exp2Syn	Synaptic mechanism for inhibition
Inhibitory Reversal Potential	e (inh)	-70	Reversal potential for inhibitory synapses
Excitatory Synaptic Weight	weight (exc)	0.005	Synaptic weight for excitatory connections
Inhibitory Synaptic Weight	weight (inh)	0.007	Synaptic weight for inhibitory connections
Excitatory Synaptic Delay	delay (exc)	5	Synaptic delay for excitatory connections
Inhibitory Synaptic Delay	delay (inh)	3	Synaptic delay for inhibitory connections
Excitatory Connection Probability	probability (exc)	0.1	Probability of connection for excitatory neurons
Inhibitory Connection Probability	probability (inh)	0.3	Probability of connection for inhibitory neurons
Stimulation Type (Poisson Input)	type	NetStim	Type of external stimulation (Poisson)
Poisson Input Rate	rate	100 Hz	Poisson input firing rate
Poisson Input Noise	noise	0.5	Degree of randomness (0: regular, 1: fully random)
Poisson Input Synaptic Weight	weight	0.01	Synaptic weight for Poisson input
Poisson Input Synaptic Delay	delay	5 ms	Synaptic delay for Poisson input
Simulation Duration	duration	1000 ms	Duration of the simulation
Time Step	dt	0.025 ms	Integration time step
Recorded Variable	recordTraces	V_soma	Variable recorded from neurons (membrane potential)
Raster Plot	plotRaster	True	Whether to generate a raster plot
Voltage Traces	plotTraces	True	Whether to plot voltage traces
2D Network Visualization	plot2Dnet	True	Whether to visualize the network in 2D

Results: izhikevich_inhibitory_network.py







Hands-on Exercise 1 !☺

(For Simulating a Single Neuron)

1. Increase input current: Change `amp` in `IClamp` from 10 to 20 pA → Does the neuron fire more frequently?
2. Modify adaptation: Change `b` in the neuron model from -2 to -1 → Does the firing pattern change (faster or slower adaptation)?
3. Change reset voltage (`c`): Modify `c` from -50 to -55 mV → Does this affect how quickly the neuron recovers after a spike?
4. Increase membrane capacitance (`C`): Change `C` from 1 to 2 → Does a larger capacitance slow down the response of the neuron?

Hands-on Exercise 2 !! ☺☺

(For Building a Network of 20 Neurons)

1. Increase connectivity: Change the connection probability (probability) from 0.1 to 0.5 in IZH->IZH → How does increased connectivity affect firing patterns?
2. Increase synaptic weight: Change 'weight': 0.005 to 'weight': 0.02 in excitatory synapses → Does stronger synaptic input lead to more synchronized activity?
3. Increase number of neurons: Change 'numCells': 20 to 'numCells': 50 → How does increasing network size influence overall activity and spike rate?
4. Modify external input (Poisson stimulation): Change rate in NetStim from 100 to 200 Hz → Does higher Poisson input increase or decrease network synchrony?

Hands-on Exercise 3 !!!☺☺☺

1. Increase inhibition: Change 'e': -80 in the inhibitory synapse → Stronger GABAergic effect.
2. Increase excitation: Change 'e': 10 in the excitatory synapse → More depolarization.

Further Experiments to Try!!!!!!!!!!!☺

1. Change connectivity probability (connParams).
2. Modify Izhikevich neuron parameters (a, b, c, d).
3. Increase inhibition and check its impact on network stability.