=====================
== ASSOCIATED FILES ==
=====================
gridcell_dynamics.zip

1D gridcell dynamics
— gridcell_dynamics_1D.m
— create_envelope.m
— compute_hardwired_weights.m
— compute_HH_ionic_currents.m
— compute_HH_syn_currents.m
— compute_HH_output.m
— compute_LIF_ionic_currents.m
— compute_LIF_syn_currents.m
— compute_LIF_output.m
— compute_LNP_ionic_currents.m
— compute_LNP_syn_currents.m
— compute_LNP_output.m

1D gridcell development
— gridcell_development_1D.m

2D gridcell dynamics
— gridcell_dynamics_2D.m
— W_2D.mat
— trajectory_2D.mat

==================
== INSTALLATION ==
==================

Unzip gridcell_dynamics.zip with any archive utility (eg. http://www.7-zip.org/download.html)

=================
== DESCRIPTION ==
=================

1D Grid Cell dynamics

The associated code models periodic and aperiodic 1D continuous attractor networks of Grid Cells using either Hodgkin-Huxley (HH) dynamics, Leaky-Integrate-and-Fire (LIF) dynamics, or Linear—Nonlinear-Poisson (LNP) dynamics. The synaptic weights are hardwired in by hand (see function `compute_hardwired_weights.m`). The synaptic profile is inspired from the mature weights that develop in the model by Widloski and Fiete (2014). The code returns matrices in dimensions of neurons x time of 1) spiking outputs (0's,1's,…), membrane potential dynamics, and synaptic activations. (Note that membrane potential dynamics are only relevant for HH and LIF dynamics).

During the simulation, the following are displayed in figures (in this order):
1) Snapshots of synaptic activations of EL and ER populations
2) Temporal evolution of synaptic activations of ER pop.
3) Temporal evolution of subset of synaptic activations of ER pop.
4) Temporal evolution of membrane potentials of ER pop.
5) Temporal evolution of subset of membrane potentials of ER pop.

To run the simulation, call `gridcell_dynamics_1D(dynamics,periodic,temp,in_strength)` from Matlab or Octave where `dynamics = {0,1,2}` for HH, LIF, and LNP dynamics respectively, `periodic = {0,1}` for aperiodic or periodic b.c.s respectively, `temp` is the temperature (set `temp = ''` for default temp of 36 deg. Celsius), and `inh_strength` modulates the strength of inhibition (set `inh_strength = ''` for default strength of 1).

ex. `gridcell_dynamics_1D(0,1,32,2)` simulates a periodic network with HH dynamics, at temperature = 32 deg. Celsius and a inhibition strength set at 2.

1D Grid Cell development

The associated code models the development of an aperiodic Grid Cell network, using LNP dynamics (see Widloski and Fiete, 2014). The code will run uninterrupted for approximately 4 hours of simulated time, and assumes a sinusoidal, back-and-forth motion of the animal across the environment. The code returns the full synaptic weights at the end of development.

During the simulation, the following are displayed in figures:
1) First row, left: profile of I-to-E weight matrices (I-to-EL in blue, I-to-ER in red).
2) First row, right: profile of E-to-I weight matrices (EL-to-I in blue, ER-to-I in red).
3) Second row, left: off-diagonal of I-to-I weight matrix
4) Second row, right: profile of I-to-I weight matrix
5) third row, left: full I-to-EL weight matrix
6) third row, right, top: full EL-to-I weight matrix
7) third row, right, bottom: full I-to-I weight matrix
8) fourth row, left, top: I population activity
9) fourth row, left, bottom: EL (blue) and ER (red) population activities
10) excitatory (blue) and inhibitory (black) synaptic STDP kernels

To run the simulation, simply call `gridcell_dynamics_1D`.

2D Grid Cell dynamics

The associated code models the dynamics of an aperiodic 2D Grid Cell network, using LNP dynamics. The synaptic weights used (G_2D.mat) are from the mature 2D network developed in Widloski and Fiete (2014). The code returns nothing.

During the simulation, the following are displayed in figures:
1) First row: I-to-E weights, concatenated (from left to right: I-to-EL, I-to-ER, I-to-EU, I-to-ED).
2) Second row: I-to-E and I-to-I weights, concatenated (from left to right: EL-to-I, ER-to-I, EU-to-I, ED-to-I, I-to-I).
3) Third row, left: animal trajectory in 2D
4) Third row, right: I population activity

To run the simulation, simply call `gridcell_dynamics_2D`.