

```

from brian2 import *

defaultclock.dt = 0.01*ms

E_L = -68*mV; E_Na = 20*mV; E_K = -80*mV; E_Ca = 120*mV; E_proc = -10*mV
C_s = 0.2*nF; C_a = 0.02*nF; g_E = 10*nS; g_La = 7.5*nS; g_Na = 300*nS;
g_Kd = 4*uS; G_Ca = 0.2*uS; G_K = 16*uS; tau_h_Ca = 150*ms; tau_m_A = 0.1*ms;
tau_h_A = 50*ms; tau_m_proc = 6*ms; tau_m_Na = 0.025*ms; tau_z = 5*second

eqs = '''# somatic compartment
dV_s/dt = (-I_syn - I_L - I_Ca - I_K - I_A - I_proc - g_E*(V_s - V_a))/C_s : volt
I_L = g_Ls*(V_s - E_L) : amp
I_K = g_K*m_K**4*(V_s - E_K) : amp
I_A = g_A*m_A**3*h_A*(V_s - E_K) : amp
I_proc = g_proc*m_proc*(V_s - E_proc) : amp
I_syn = I_fast + I_slow: amp
I_fast : amp
I_slow : amp
I_Ca = g_Ca*m_Ca**3*h_Ca*(V_s - E_Ca) : amp
dm_Ca/dt = (m_Ca_inf - m_Ca)/tau_m_Ca : 1
m_Ca_inf = 1/(1 + exp(0.205/mV*(-61.2*mV - V_s))): 1
tau_m_Ca = 30*ms - 5*ms/(1 + exp(0.2/mV*(-65*mV - V_s))) : second
dh_Ca/dt = (h_Ca_inf - h_Ca)/tau_h_Ca : 1
h_Ca_inf = 1/(1 + exp(-0.15/mV*(-75*mV - V_s))) : 1
dm_K/dt = (m_K_inf - m_K)/tau_m_K : 1
m_K_inf = 1/(1 + exp(0.1/mV*(-35*mV - V_s))) : 1
tau_m_K = 2*ms + 55*ms/(1 + exp(-0.125/mV*(-54*mV - V_s))) : second
dm_A/dt = (m_A_inf - m_A)/tau_m_A : 1
m_A_inf = 1/(1 + exp(0.2/mV*(-60*mV - V_s))) : 1
dh_A/dt = (h_A_inf - h_A)/tau_h_A : 1
h_A_inf = 1/(1 + exp(-0.18/mV*(-68*mV - V_s))) : 1
dm_proc/dt = (m_proc_inf - m_proc)/tau_m_proc : 1
m_proc_inf = 1/(1 + exp(0.2/mV*(-55*mV - V_s))) : 1
# axonal compartment
dV_a/dt = (-g_La*(V_a - E_L) - g_Na*m_Na**3*h_Na*(V_a - E_Na)
          - g_Kd*m_Kd**4*(V_a - E_K) - g_E*(V_a - V_s))/C_a : volt
dm_Na/dt = (m_Na_inf - m_Na)/tau_m_Na : 1
m_Na_inf = 1/(1 + exp(0.1/mV*(-42.5*mV - V_a))) : 1
dh_Na/dt = (h_Na_inf - h_Na)/tau_h_Na : 1
h_Na_inf = 1/(1 + exp(-0.13/mV*(-50*mV - V_a))) : 1
tau_h_Na = 10*ms/(1 + exp(0.12/mV*(-77*mV - V_a))) : second
dm_Kd/dt = (m_Kd_inf - m_Kd)/tau_m_Kd : 1
m_Kd_inf = 1/(1 + exp(0.2/mV*(-41*mV - V_a))) : 1
tau_m_Kd = 12.2*ms + 10.5*ms/(1 + exp(-0.05/mV*(58*mV - V_a))) : second
# class-specific fixed maximal conductances
g_Ls : siemens (constant)
g_A : siemens (constant)
g_proc : siemens (constant)
# Adaptive conductances
g_Ca = G_Ca/2*(1 + tanh(z)) : siemens
g_K = G_K/2*(1 - tanh(z)) : siemens
I_diff = (I_target + I_Ca) : amp
dz/dt = tanh(I_diff/nA)/tau_z : 1
I_target : amp (constant)
# Neuron class
label : integer (constant)'''

circuit = NeuronGroup(3, eqs, method='rk2',
                      threshold='m_Na > 0.5', refractory='m_Na > 0.5')
ABPD, LP, PY = 0, 1, 2
# class-specific constants
circuit.label = [ABPD, LP, PY]
circuit.I_target = [0.4, 0.3, 0.5]*nA; circuit.g_Ls = [30, 25, 15]*nS
circuit.g_A = [450, 100, 250]*nS; circuit.g_proc = [6, 8, 0]*nS
# Initial conditions
circuit.V_s = E_L; circuit.V_a = E_L
circuit.m_Ca = 'm_Ca_inf'; circuit.h_Ca = 'h_Ca_inf'; circuit.m_K = 'm_K_inf';
circuit.m_A = 'm_A_inf'; circuit.h_A = 'h_A_inf'; circuit.m_proc = 'm_proc_inf'
circuit.m_Na = 'm_Na_inf'; circuit.h_Na = 'h_Na_inf'; circuit.m_Kd = 'm_Kd_inf'

```