For reference to the code

```
function dy = output(t,y)
%Output Variables  (Eric)
%Soma's matrix column
Vm_soma = y(1); %set matrix column to Vm_soma
m = y(2); % set matrix column to n
h = y(3); % set matrix column to m
n = y(4); % set matrix column to h
%Dendrite's matrix columns
Vm_dend = y(5); % set matrix column to Vm_dendrite
mca = y(6); % set matrix column to m_ca
mkca = y(7); % set matrix column to m_kca
mkahp = y(8); % set matrix column to m_kahp
% Calcium concentration
ca_conc = y(9);

%Defining rate constants (Eric)
[alpha_m, beta_m, alpha_h, beta_h, alpha_n, beta_n ] = PR_soma_gating(Vm_soma);
[alpha_mca, beta_mca, alpha_kca, beta_kca, alpha_kahp, beta_kahp ] = PR_dend_gating(Vm_dend,ca_

% Define constants for the equation (Eric)
AS = 1/3; % fractional area of soma
AD = 2/3; % fractional area of dendrite
GS_leak = AS*5*(10^-9); % somatic leak conductance (S)
GD_leak = AD*5*(10^-9); % dendrite leak conductance (S)
GNa_max = AS * 3 * (10^-6); % maximum sodium conductance (S)
GK_max = AS * 2 * (10^-6); % maximum potassium conductance (S)
GCa_max = AD * 2 * (10^-6); % maximum calcium conductance (S)
GKCa_max = AD * 2.5 * (10^-6); % max calcium-dependent potassium conductance (S)
GKAHP_max = AD * 40 * (10^-9); % max after-hyperpolarization conductance (S)
GLink = 50*10^-9; % link conductance (S)
ENa = 60 * (10^-3); % sodium reversal potential (V)
ECa = 80 * (10^-3); % calcium reversal potential (V)
EK = -75 * (10^-3); % potassium reversal potential (V)
EL = -60 * (10^-3); % leak reversal potential (V)
CS = AS * 100 * (10^-12); % capacitance of soma (F)
CD = AD * 100 * (10^-12); % capacitance of dendrite (F)
tau_Ca = 50 * (10^-3); % calcium delay time constant (s)
k = (5 * 10^6) / AD; % conversion charge->concentration (MC^-1)

%Applied Currents (Eric)
IS_app = 0; % soma applied current (A)
ID_app = 0; % dendrite applied current (A)

%Steady state for all gating variables (Eric and Anthony)
mca_infi = alpha_mca/(alpha_mca + beta_mca);
mkca_infi = alpha_kca/(alpha_kca+beta_kca);
mkahp_infi = alpha_kahp/(alpha_kahp+beta_kahp);
h_infi = alpha_h/(alpha_h+beta_h);
```

```
m_infi = alpha_m/(alpha_m+beta_m);
n_infi = alpha_n/(alpha_n+beta_n);

%Time constant for all gating variables (Eric and Anthony)
mca_tau = 1/(alpha_mca + beta_mca);
mkca_tau = 1/(alpha_kca+beta_kca);
mkahp_tau = 1/(alpha_kahp+beta_kahp);
m_tau = 1/(alpha_m + beta_m);
h_tau= 1/(alpha_h + beta_h);
n_tau = 1/(alpha_n + beta_n);

%ODE for soma gating (Eric and Anthony)
dm_over_dt = (m_infi-m)/m_tau;
dh_over_dt = (h_infi-h)/h_tau;
dn_over_dt = (n_infi-n)/n_tau;
%ODE for dendrite gating (Eric and Anthony)
dmca_over_dt = (mca_infi-mca)/mca_tau;
dmkca_over_dt = (mkca_infi-mkca)/mkca_tau;
dmkahp_over_dt = (mkahp_infi-mkahp)/mkahp_tau;
%ODE for Ca concentration (Eric and Anthony)
Ica = GCa_max*mca^2*(ECa-Vm_dend);
dca_over_dt = (-ca_conc/tau_Ca) + k * Ica;
%X as function of Ca_conc (Eric)
if (4000*ca_conc < 1)
    X_val = 4000*ca_conc;
else
    X_val = 1;
end
%ODE for Soma (Mauricio)
I_leak_soma = GS_leak*(EL-Vm_soma);
Na_leak = GNa_max*m^2*h*(ENa-Vm_soma);
K_leak = GK_max*n^2*(EK-Vm_soma);
link_soma = GLink*(Vm_dend-Vm_soma);
Vms_over_dt = (I_leak_soma + Na_leak + K_leak + link_soma + IS_app) / CS;

%ODE for Dendrite (Mauricio)
I_leak_den = GD_leak*(EL-Vm_dend);
Ca_leak = GCa_max*mca^2*(ECa-Vm_dend);
Ca_dep_K_I = GKCa_max*mkca*X_val*(EK-Vm_dend);
Kahp_I = GKAHP_max * mkahp * (EK-Vm_dend);
link_den = -1*(GLink * (Vm_dend-Vm_soma));
Vmd_over_dt = (I_leak_den + Ca_leak + Ca_dep_K_I + Kahp_I + link_den + ID_app) / CD;

%Wrapper function
dy = [Vms_over_dt;dm_over_dt;dh_over_dt;dn_over_dt;Vmd_over_dt;dmca_over_dt;dmkca_over_dt;dmkah

end %function end
```