%% function calculate\_APD receives input variables of:

%% locs- vector of the index location of events’ beginnings (index of “time of max dV/dt – constant” for all fluorescent action potentials)

%% t- time vector in seconds

%% signal- fluorescent action potential signal

%% bl\_vec- vector of baseline values for all action potentials

%% amp\_vec- vector of amplitude height values for all action potentials

%% apd\_num- percentage of repolarization (for APD80, apd\_num = 0.8)

%% stop\_vec- the index location of event ending (index of “time of max dV/dt + constant” for all fluorescent action potentials)

%% interval- sampling interval in seconds

%%

%%

%% function calculate\_APD returns as output an array of APDs for each action potential together with the mean

function apd\_array = calculate\_APD(locs,t,signal,bl\_vec,amp\_vec,apd\_num,stop\_vec,interval,apd\_90)

apd\_array = {sprintf('APD\_%d',apd\_num\*100)};%apd title

k =2;

locs\_without\_nan = locs(~isnan(locs));

for i=1:length(locs\_without\_nan)

j = find(locs\_without\_nan(i) == locs);

val = bl\_vec(i) +( (1-apd\_num )\* amp\_vec(i));%distance from base line

start = locs(j);

stop = stop\_vec(i);

signal\_temp = signal(start:stop); %section with one event- from max derivative to post value

max\_ind = find(signal\_temp == max(signal\_temp));%the peak index

signal\_temp = signal(start + max\_ind: stop);%section with the former event starting from the peak

differ = signal\_temp - val;%intersection between the signal to the horizontal line val

differ = abs(differ);

ind = find(min(differ) == differ );%the index of intersection

apd\_array{k,1} = (ind + max\_ind)\*interval\*1000;%indices multiplied with samp inetval, for calculating the time period of APD

k = k+1;

end

apd\_array{k} = mean(cell2mat(apd\_array((2:(k-1)))));