

The models used in this book are presented below in the syntax of the authors software.

1 Channels

Hodgkin-Huxley

```
# Hodgkin huxley equations
init v=-65 m=.05 h=0.6 n=.317
par i0=0
par vna=50 vk=-77 vl=-54.4 gna=120 gk=36 gl=0.3 c=1 phi=1
par ip=0 pon=50 poff=150
is(t)=ip*heav(t-pon)*heav(poff-t)
am(v)=phi*.1*(v+40)/(1-exp(-(v+40)/10))
bm(v)=phi*4*exp(-(v+65)/18)
ah(v)=phi*.07*exp(-(v+65)/20)
bh(v)=phi*1/(1+exp(-(v+35)/10))
an(v)=phi*.01*(v+55)/(1-exp(-(v+55)/10))
bn(v)=phi*.125*exp(-(v+65)/80)
v'=(I0+is(t) - gna*h*(v-vna)*m^3-gk*(v-vk)*n^4-gl*(v-vl)-gsyn*s*(v-vsyn))/c
m'=am(v)*(1-m)-bm(v)*m
h'=ah(v)*(1-h)-bh(v)*h
n'=an(v)*(1-n)-bn(v)*n
s'=sinf(v)*(1-s)-s/tausyn
# track the currents
sinf(v)=alpha/(1+exp(-v/vshp))
par alpha=2,vshp=5,tausyn=20,gsyn=0,vsyn=0
aux ina=gna*(v-vna)*h*m^3
aux ik=gk*(v-vk)*n^4
aux il=gl*(v-vl)
# track the stimulus
aux stim=is(t)
@ bound=10000
done
```

HH equiv potentials

```
# hh.ode equivalent potentials
init v=-65 vm=-65,vn=-65,vh=-65
par i0, vna=50 vk=-77 vl=-54.4 gna=120 gk=36 gl=0.3 c=1 phi=1
par eps=.1
am(v)=phi*.1*(v+40)/(1-exp(-(v+40)/10))
bm(v)=phi*4*exp(-(v+65)/18)
ah(v)=phi*.07*exp(-(v+65)/20)
```

```

bh(v)=phi*1/(1+exp(-(v+35)/10))
an(v)=phi*.01*(v+55)/(1-exp(-(v+55)/10))
bn(v)=phi*.125*exp(-(v+65)/80)
minf(v)=am(v)/(am(v)+bm(v))
ninf(v)=an(v)/(an(v)+bn(v))
hinf(v)=ah(v)/(ah(v)+bh(v))
km(v)=am(v)+bm(v)
kn(v)=an(v)+bn(v)
kh(v)=ah(v)+bh(v)
mp(v)=(minf(v+eps)-minf(v-eps))/(2*eps)
np(v)=(ninf(v+eps)-ninf(v-eps))/(2*eps)
hp(v)=(hinf(v+eps)-hinf(v-eps))/(2*eps)
v'=(I0 - gna*hinf(vh)*(v-vna)*minf(vm)^3-gk*(v-vk)*ninf(vn)^4-gl*(v-vl))/c
vm'=km(v)*(minf(v)-minf(vm))/mp(vm)
vn'=kn(v)*(ninf(v)-ninf(vn))/np(vn)
vh'=kh(v)*(hinf(v)-hinf(vh))/hp(vh)
aux n=ninf(vn)
aux h=hinf(vh)
done

```

HH rinzel reduction

```

# reduced HH equations using the rinzel reduction and n
# as the variable
init v=-65 n=.4
par i0=0
par vna=50 vk=-77 vl=-54.4 gna=120 gk=36 gl=0.3 c=1 phi=1
par ip=0 pon=50 poff=150
is(t)=ip*heav(t-pon)*heav(poff-t)
am(v)=phi*.1*(v+40)/(1-exp(-(v+40)/10))
bm(v)=phi*4*exp(-(v+65)/18)
ah(v)=phi*.07*exp(-(v+65)/20)
bh(v)=phi*1/(1+exp(-(v+35)/10))
an(v)=phi*.01*(v+55)/(1-exp(-(v+55)/10))
bn(v)=phi*.125*exp(-(v+65)/80)
v'=(I0+is(t) - gna*h*(v-vna)*m^3-gk*(v-vk)*n^4-gl*(v-vl))/c
m=am(v)/(am(v)+bm(v))
#h'=ah(v)*(1-h)-bh(v)*h
n'=an(v)*(1-n)-bn(v)*n
h=h0-n
par h0=.8
@ bound=10000
done

```

Morris-Lecar

```

# Morris-Lecar model Methods Chapter
dv/dt = ( I - gca*minf(V)*(V-Vca)-gk*w*(V-VK)-gl*(V-Vl)+s(t))/c
dw/dt = phi*(winf(V)-w)/tauw(V)
v(0)=-16
w(0)=0.014915
minf(v)=.5*(1+tanh((v-v1)/v2))
winf(v)=.5*(1+tanh((v-v3)/v4))
tauw(v)=1/cosh((v-v3)/(2*v4))
param vk=-84,vl=-60,vca=120
param i=0,gk=8,gl=2,c=20
param v1=-1.2,v2=18
# Uncomment the ones you like!!
par1-3 v3=2,v4=30,phi=.04,gca=4.4
set hopf {v3=2,v4=30,phi=.04,gca=4.4}
set snic {v3=12,v4=17.4,phi=.06666667,gca=4}
set homo {v3=12,v4=17.4,phi=.23,gca=4}
#par4-6 v3=12,v4=17.4,phi=.06666667,gca=4
#par7-8 v3=12,v4=17.4,phi=.23,gca=4
param s1=0,s2=0,t1=50,t2=55,t3=500,t4=550
# double pulse stimulus
s(t)=s1*heav(t-t1)*heav(t2-t)+s2*heav(t-t3)*heav(t4-t)
@ total=150,dt=.25,xlo=-75,xhi=75,ylo=-.25,yhi=.5,yp=v,yp=w
done

```

Butera-Smith

```

# butera and smith model using NaP
par cm=21,i=0
xinf(v,vt,sig)=1/(1+exp((v-vt)/sig))
taux(v,vt,sig,tau)=tau/cosh((v-vt)/(2*sig))
# leak
il=gl*(v-el)
par gl=2.8,el=-65
# fast sodium -- h=1-n
minf(v)=xinf(v,-34,-5)
ina=gna*minf(v)^3*(1-n)*(v-ena)
par gna=28,ena=50
# delayed rectifier
ninf(v)=xinf(v,-29,-4)
taun(v)=taux(v,-29,-4,10)
ik=gk*n^4*(v-ek)
par gk=11.2,ek=-85
# NaP
mninf(v)=xinf(v,-40,-6)

```

```

hinf(v)=xinf(v,-48,6)
tauh(v)=taux(v,-48,6,taubar)
par gnap=2.8,taubar=10000
inap=gnap*mninf(v)*h*(v-ena)
v' = (i-il-ina-ik-inap)/cm
n'=(ninf(v)-n)/taun(v)
h'=(hinf(v)-h)/tauh(v)
@ total=40000,dt=1,meth=cvode,maxstor=100000
@ tol=1e-8,atol=1e-8
@ xlo=0,xhi=40000,ylo=-80,yhi=20
done

```

Calcium-dependent inactivation

```

# L-type calcium current with calcium-dependent inactivation
# Poirazi P, Brannon T, Mel BW (2003a)
# Arithmetic of subthreshold synaptic summation in a model CA1 pyramidal cell.
# Neuron 37:977-987
# adjusted beta slightly from .028 to .01
# from ModelDB
!faraday=96520
!rgas=8.3134
!temp=273.15+celsius
h=ki/(ki+ca)
xi=v*faraday*2/(rgas*1000*temp)
cfedrive=.002*faraday*xi*(ca-cao*exp(-xi))/(1-exp(-xi))
m=alpm(v)/(alpm(v)+betm(v))
ical=pcal*m*h*cfedrive
par ki=.001,celsius=25,cao=2,pcal=2,cainf=1e-4,taur=200
init ca=1e-4,v=-65
alpm(v) = 0.055*(-27.01 - v)/(exp((-27.01-v)/3.8) - 1)
betm(v) = 0.94*exp((-63.01-v)/17)
# migliore model:
# alpm(v) = 15.69*(-1.0*v+81.5)/(exp((-1.0*v+81.5)/10.0)-1.0)
# betm(v) = 0.29*exp(-v/10.86)
v'=-gl*(v-el)-ical+i0
ca'=-ical*beta-(ca-cainf)/taur
par beta=.01,i0=0,el=-70,gl=.05
aux ica=ical
@ total=1000,meth=qualrk,tol=1e-8,atol=1e-8,dt=.25
@ xp=v,yp=ca,xlo=-80,xhi=-10,ylo=0,yhi=2
done

```

T-type current with rebound

```

# cat-spike.ode
# i=+/- .25 for 25 msec or -2 for rebound + depolarized
# Huguenard and McCormick T-type calcium kinetics
# using CFE with calcium fixed in concentration
# sodium and potassium channels added for spiking
#
i(t)=i0+i1*heav(t-ton)*heav(ton+tdur-t)
!faraday=96520
!rgas=8.3134
!temp=273.15+celsius
xi=v*faraday*2/(rgas*1000*temp)
cfedrive=.002*faraday*xi*(cai-cao*exp(-xi))/(1-exp(-xi))
m=minf(v)
par el=-65,celsius=25,cao=2,pcat=.15,cai=1e-4
par gna=8,gk=4,ena=55,ek=-90
minf(v)=1/(1+exp(-(v+59)/6.2))
hinf(v)=1/(1+exp((v+83)/4))
# tauh(v)=if(v<(-82))then(exp((v+469)/66.6))else(28 + exp(-(v+24)/10.5))
tauh(v)=22.7+.27/(exp((v+48)/4)+exp(-(v+407)/50))
i_cat=pcat*m*m*h*cfedrive
amna=.091*(v+38)/(1-exp(-(v+38)/5))
bmna=-.062*(v+38)/(1-exp((v+38)/5))
ahna=.016*exp((-55-v)/15)
bhna=2.07/(1+exp((17-v)/21))
mna=amna/(amna+bmna)
ank=.01*(-45-v)/(exp((-45-v)/5)-1)
bnk=.17*exp((-50-v)/40)
v'=-gl*(v-el)-i_cat+i(t)-gna*mna^3*hna*(v-ena)-gk*n^4*(v-ek)
h'=(hinf(v)-h)/tauh(v)
hna'=ahna*(1-hna)-bhna*hna
n'=ank*(1-n)-bnk*n
init h=.16,v=-76
par gl=.05,i0=0,i1=0,ton=100,tdur=25
aux icat=i_cat
@ meth=qualrk,dt=.25,total=500,atol=1e-8,rtol=1e-8
@ nmesh=100,xp=v,yp=h,xlo=-90,ylo=-.1,xhi=20,yhi=.8,bound=1000
done

```

Connor stevens

```

# constev.ode
i(t)=i0+i1*heav(t-ton)
par i0,ga=47.7
par gtotal=67.7
!gk=gtotal-ga

```

```

init v=-65
par ek=-72 ena=55 ea=-75 el=-17
par gna=120 gl=0.3
par ms=-5.3 hs=-12 ns=-4.3
par ap=2 ton=100 i1=0
# Hodgkin-Huxley with shifts - 3.8 is temperature factor
am(V)=-.1*(V+35+ms)/(exp(-(V+35+ms)/10)-1)
bm(V)=4*exp(-(V+60+ms)/18)
minf(V)=am(V)/(am(V)+bm(V))
taum(V)=1/(3.8*(am(V)+bm(V)))
ah(V)=.07*exp(-(V+60+hs)/20)
bh(V)=1/(1+exp(-(V+30+hs)/10))
hinf(V)=ah(V)/(ah(V)+bh(V))
tauh(V)=1/(3.8*(ah(V)+bh(V)))
an(V)=-.01*(V+50+ns)/(exp(-(V+50+ns)/10)-1)
bn(V)=.125*exp(-(V+60+ns)/80)
ninf(V)=an(V)/(an(V)+bn(V))
# Taun is doubled
taun(V)=2/(3.8*(an(V)+bn(V)))
# now the A current
ainf(V)=(.0761*exp((V+94.22)/31.84)/(1+exp((V+1.17)/28.93)))^(.3333)
taua(V)=.3632+1.158/(1+exp((V+55.96)/20.12))
binf(V)=1/(1+exp((V+53.3)/14.54))^4
taub(V)=1.24+2.678/(1+exp((V+50)/16.027))
# Finally the equations...
v'=-gl*(v-el)-gna*(v-ena)*h*m*m*m-gk*(v-ek)*n*n*n*n-ga*(v-ea)*b*a*a*a+i(t)
M'=(minf(v)-m)/taum(v)
H'=(hinf(v)-h)/tauh(v)
N'=(ninf(v)-n)/taun(v)
A'=(ainf(v)-a)/taua(v)
B'=(binf(v)-b)/taub(v)
@ total=200,xhi=200,ylo=-70,yhi=20
done

```

Inward rectifier + potassium pump

```

# inward rectifier with potassium pump
v'=-gl*(v-el)-ikir
ek=90*log10(kout)
ikir=gk/(1+exp((v-vth)/vs))*(v-ek)
par gk=.8
par vth=-85,vs=5,el=-60,gl=0.05
init v=-63
par beta=0.04,tau=1000

```

```

init kout=.1
kout'=(ikir*beta-(kout-.1))/tau
aux vk=ek
@ total=2000, meth=qualrk, dt=.5, tol=1e-8, atol=1e-8
done

```

Destexhe & Pare model

```

# currents from destexhe and pare
# J. Neurophys 1999
# sodium
am(v)=-.32*(v-vt-13)/(exp(-(v-vt-13)/4)-1)
par i=0, gkm=2
# shifted to acct for threshold
num vt=-58, vs=-10
bm(v)=.28*(v-vt-40)/(exp((v-vt-40)/5)-1)
ah(v)=.128*exp(-(v-vt-vs-17)/18)
bh(v)=4/(1+exp(-(v-vt-vs-40)/5))
ina(v,m,h)=gna*m^3*h*(v-ena)
par gna=120, ena=55
# delayed rectifier
an(v)=-.032*(v-vt-15)/(exp(-(v-vt-15)/5)-1)
bn(v)=.5*exp(-(v-vt-10)/40)
ikdr(v,n)=gk*n^4*(v-ek)
par gk=100, ek=-85
# slow potassium current
akm(v)=.0001*(v+30)/(1-exp(-(v+30)/9))
bkm(v)=-.0001*(v+30)/(1-exp((v+30)/9))
ikm(v,m)=gkm*m*(v-ek)
#
v'=(I-gl*(v-el)-ikdr(v,n)-ina(v,m,h)-ikm(v,mk))/cm
m'=am(v)*(1-m)-bm(v)*m
h'=ah(v)*(1-h)-bh(v)*h
n'=an(v)*(1-n)-bn(v)*n
mk'=akm(v)*(1-mk)-bkm(v)*mk
init v=-73.87, m=0, h=1, n=.002, mk=.0075
# passive stuff
par gl=.019, el=-65, cm=1
# numerics stuff
@ total=1000, dt=.25, meth=qualrk, xhi=1000, maxstor=10000
@ bound=1000, ylo=-85, yhi=-50
done

```

Sag and inward rectifier

```

# sag + inward rectifier

par i=0
par gl=.025,e1=-70
# sag
# migliore tau0=46,vm=-80,b=23
# migliore vt=-81,k=8
# mccormick tau0=1000,vm=-80,b=13.5
#
ih=gh*(V-eh)*y
par gh=0.25,eh=-43
yinf(v)=1/(1+exp((v-vt)/k))
ty(v)=tau0/cosh((v-vm)/b)
par k=5.5,vt=-75
par tau0=1000,vm=-80,b=13.5
#
# kir
par ek=-85,gk=1
ikir=gk*minf(v)*(v-ek)
minf(v)=1/(1+exp((v-va)/vb))
par va=-80,vb=5
v'=i-gl*(v-e1)-ih-ikir
y'=(yinf(v)-y)/ty(v)
init v=-68
init y=.24
@ total=1000, meth=qualrk, dt=.25
@ xp=v, yp=y, xlo=-90, xhi=-40, ylo=0, yhi=0.6
@ nmesh=100
done

```

CAN current

```

# can.ode
# spiking model plus CAN current

# sodium
am(v)=-.32*(v-vt-13)/(exp(-(v-vt-13)/4)-1)
num vt=-58,vs=-10
bm(v)=.28*(v-vt-40)/(exp((v-vt-40)/5)-1)
ah(v)=.128*exp(-(v-vt-vs-17)/18)
bh(v)=4/(1+exp(-(v-vt-vs-40)/5))
ina(v,m,h)=gna*m^3*h*(v-ena)
par gna=120,ena=55
# delayed rectifier

```

```

an(v)=-.032*(v-vt-15)/(exp(-(v-vt-15)/5)-1)
bn(v)=.5*exp(-(v-vt-10)/40)
ikdr(v,n)=gk*n^4*(v-ek)
par gk=100,ek=-85
# voltage
v'=(I-gl*(v-el)-ikdr(v,n)-ina(v,m,h)-ican)/cm
m'=am(v)*(1-m)-bm(v)*m
h'=ah(v)*(1-h)-bh(v)*h
n'=an(v)*(1-n)-bn(v)*n
# can dynamics
par taumc=4000
ican=gcan*mc*(v-ecan)
par ecan=-20
par gcan=.05,alpha=.005
mc'=alpha*ca^2*(1-mc)-mc/taumc
# pulse function for calcium entry
puls(t)=heav(t)*heav(wid-t)
# here is the calcium
ca=puls(t-t1)+puls(t-t2)+puls(t-t3)
par t1=200,t2=700,t3=1200
par wid=50
# initial data
init v=-64.97,m=0.003,h=.991,n=.01,mc=0
# passive
par gl=.019,el=-65,cm=1,i=0
# keep track of calcium
aux stim=10*ca-100
# XPP stuff
@ total=2000,dt=.05,meth=rk4,xhi=2000,maxstor=100000
@ bound=1000,ylo=-100,yhi=20
@ nplot=2,yp2=stim
done

```

Calcium-dependent AHP

```

# ahp.ode
# uses very simple model of AHP with ca dynamics
# and high threshold Calc
# sodium
am(v)=-.32*(v-vt-13)/(exp(-(v-vt-13)/4)-1)
par i=0,gkm=2
# shifted to acct for threshold
num vt=-58,vs=-10
bm(v)=.28*(v-vt-40)/(exp((v-vt-40)/5)-1)

```

```

ah(v)=.128*exp(-(v-vt-vs-17)/18)
bh(v)=4/(1+exp(-(v-vt-vs-40)/5))
ina(v,m,h)=gna*m^3*h*(v-ena)
par gna=120,ena=55
# delayed rectifier
an(v)=-.032*(v-vt-15)/(exp(-(v-vt-15)/5)-1)
bn(v)=.5*exp(-(v-vt-10)/40)
ikdr(v,n)=gk*n^4*(v-ek)
par gk=100,ek=-85
#
# l-type calcium
ica(v)=gca*(v-eca)/(1+exp(-(v-vlth)/kl))
par vlth=-5,kl=5,gca=.5,eca=120
mahp(ca)=ca^2/(kca^2+ca^2)
iahp(ca)=gahp*mahp(ca)*(v-ek)
par gam=1,tauca=300,kca=2,gahp=1
v'=(I-gl*(v-el)-ikdr(v,n)-ina(v,m,h)-ica(v)-iahp(ca))/cm
m'=am(v)*(1-m)-bm(v)*m
h'=ah(v)*(1-h)-bh(v)*h
n'=an(v)*(1-n)-bn(v)*n
ca'=- (gam*ica(v)+ca)/tauca
#
init v=-73.87,m=0,h=1,n=.002
# passive stuff
par gl=.019,el=-65,cm=1
aux mahpx=mahp(ca)
# numerics stuff
@ total=1000,dt=.25,meth=qualrk,xhi=1000,maxstor=10000
@ bound=1000,ylo=-85,yhi=-50
done

```

HH cable

```

# hhhcable.ode
init v[1..150]=-65 m[j]=.05 h[j]=0.6 n[j]=.317
par L=10,ri=100,d=.1
par vna=50 vk=-77 vl=-54.4 gna=120 gk=36 gl=0.3 c=1 phi=1
# two stimulus protocol
par ip1=0,ip2=0
par wid=2,t1=10,t2=50
# smooth step function
sheav(z)=1/(1+exp(-b*z))
par b=5
# local pulse

```

```

par xwid=5
lpul(t,x)=sheav(xwid-x)*sheav(t)*sheav(wid-t)
is(t,x)=ip1*lpul(t-t1,x)+ip2*lpul(t-t2,x)
am(v)=phi*.1*(v+40)/(1-exp(-(v+40)/10))
bm(v)=phi*4*exp(-(v+65)/18)
ah(v)=phi*.07*exp(-(v+65)/20)
bh(v)=phi*1/(1+exp(-(v+35)/10))
an(v)=phi*.01*(v+55)/(1-exp(-(v+55)/10))
bn(v)=phi*.125*exp(-(v+65)/80)
# boundaries are zero flux
!dd=4*d*150*150/(ri*L*L)
v0=v1
v151=v150
%[1..150]
v[j]’=(is(t,[j]) - gna*h[j]*(v[j]-vna)*m[j]^3-gk*(v[j]-vk)*n[j]^4\
      -gl*(v[j]-v1)+(dd)*(v[j+1]-2*v[j]+v[j-1])))/c
m[j]’=am(v[j])*(1-m[j])-bm(v[j])*m[j]
h[j]’=ah(v[j])*(1-h[j])-bh(v[j])*h[j]
n[j]’=an(v[j])*(1-n[j])-bn(v[j])*n[j]
%
aux stim1=is(t,1)
aux vp50=(is(t,50) - gna*h50*(v50-vna)*m50^3-gk*(v50-vk)*n50^4\
      -gl*(v50-v1)+DD*(v51-2*v50 +v49))/c
@ bound=10000
@ meth=cvode,bandlo=4,bandup=4
@ tol=1e-10,atol=1e-10,dt=.05,total=80
done

```

noisy LIF without reset

```

# noisy LIF without reset
f(v)=I-V
wiener w
V’=f(V)+sig*w
init V=0
par I=0,sig=1
@ meth=euler,total=200
done

```

First passage BVP

```

# first passage set up to compute the firing times
# this is defined on an interval [0,1]
# and split up to get the interior value
#

```

```

par I=-1,sig=1,vreset=-1,vspike=10,a=10
b=(vreset+a)
c=(vspike-vreset)
# ok - here it is
# u is lower and w is upper interval
# s lies between 0 and 1
# u(s=0) = T(-A)
# u(s=1) = w(s=0)=T(V_reset)
# w(s=1) = T(V_spike)
# gotta write it as a system
du/dt=up
dup/dt=-2*b*b/sig-2*f(-a+b*s)*up*b/sig
dw/dt=wp
dwp/dt=-2*c*c/sig-2*f(vreset+c*s)*wp*c/sig
ds/dt=1
# 5 equations - 5 boundary conds
# du/ds=0 at s=0
bndry up
# w=0 at s=1
bndry w'
# du/ds(1)=dw/ds(0)
bndry up'-wp
# u(1)=w(0)
bndry u'-w
# s=t
bndry s
# set up some numerics
@ total=1,dt=.005
# here is f, dont want to forget f
f(x)=x^2+I
done

```

QIFA period as BVP

```

# boundary value problem
# to find period of the QIFA model
v'=p*(v^2+i-u)
u'=p*a*(b*v-u)
p'=0
b v'-1
b v-c
b u-(u'+d)
par I=1
par c=-.25,a=.1,b=1,d=.5
init p=5.6488

```

```

init v=-.25,u=1.211
@ total=1,dt=.005
done

```

Golomb amitai

```

# Golomb amitai
# ionic currents
i_ion(v,h,n,z)=gl*(v-vl)+(gna*minf(v)^3+h*gnap*pinf(v))*(v-vna)+(gk*n^4+gz*z)*(v-vk)
minf(v)=1/(1+exp(-(v-thetam)/sigmam))
pinf(v)=1/(1+exp(-(v-thetap)/sigmap))
GAMMAF(VV,theta,sigma)=1.0/(1.0+exp(-(VV-theta)/sigma))
v'=I-i_ion(v,h,n,z)-gsyn*s*(v-vsyn)
h'=phi*(GAMMAF(V,thetah,sigmah)-h)/(1.0+7.5*GAMMAF(V,t_tauh,-6.0))
n'=phi*(GAMMAF(V,thetan,sigman)-n)/(1.0+5.0*GAMMAF(V,t_taub,-15.0))
z'=(GAMMAF(V,thetaz,sigmaz)-z)/tauZs
s'=alpha*(1-s)/(1+exp(-(v-vsth)/vshp))-beta*s

# synaptic parameters
p gsyn=0.2
p vsth=-10,vshp=5,alpha=.6,beta=.015,vsyn=0

# kinetic parameters/shapes
p phi=2.7
p thetam=-30.0,sigmam=9.5,thetah=-53.0,sigmah=-7.0
p thetan=-30.0,sigman=10.0,thetap=-40.0,sigmap=5.0
p thetaz=-39.0,sigmaz=5.0,tauZs=75.0
# ionic parameters
p VNa=55.0,VK=-90.0,VL=-70.0,t_tauh=-40.5,t_taub=-27.0
p gNa=24.0,gK=3.0,gL=0.02,I=0.0
p gNaP=0.07,gZ=.1
# set gz=0 and gl=.09,vl=-85.5 to compensate
done

```

McCormick Huguenard - many channels

```

# the McCormick-Huguenard channel models -- Mix and match as you like
#
# UNITS: millivolts, milliseconds, nanofarads, nanoamps, microsiemens
# moles
# cell is 29000 micron^2 in area so capacitance is in nanofarads
# all conductances are in microsiemens and current is in nanofarads.
#
par I=0,c=.29

```

```

v'=(I -ina-ik-ileak-ik2-inap-it-iahp-im-ia-ic-il-ih+istep(t))/c
# the current is a step function with amplitude ip
istep(t)=ip*heav(t-t_on)*heav(t_off-t)
par ip=0.0,t_on=100,t_off=200
# passive leaks
par gkleak=.007,gnaleak=.0022
Ileak=gkleak*(v-ek)+gnaleak*(v-ena)
#
aux i_leak=ileak
# INA
par gna=0,Ena=45
Ina=gna*(v-ena)*mna^3*hna
amna=.091*(v+38)/(1-exp(-(v+38)/5))
bmna=-.062*(v+38)/(1-exp((v+38)/5))
ahna=.016*exp((-55-v)/15)
bhna=2.07/(1+exp((17-v)/21))
mna'=amna*(1-mna)-bmna*mna
hna'=ahna*(1-hna)-bhna*hna
#
aux i_na=ina
# Delayed rectifier IK
par gk=0,Ek=-105
Ik=gk*(v-ek)*nk^4
ank=.01*(-45-v)/(exp((-45-v)/5)-1)
bnk=.17*exp((-50-v)/40)
nk'=ank*(1-nk)-bnk*nk
#
aux i_k=ik
# INap same tau as Na but diff activation
par gnep=0
inap=gnep*map^3*(v-ena)
map'=(1/(1+exp((-49-v)/5))-map)/(amna+bmna)
#
aux i_nap=inap
# ia A-type inactivating potassium current
#
ia=ga*(v-ek)*(.6*ha1*ma1^4+.4*ha2*ma2^4)
mainf1=1/(1+exp(-(v+60)/8.5))
mainf2=1/(1+exp(-(v+36)/20))
tma=(1/(exp((v+35.82)/19.69)+exp(-(v+79.69)/12.7))+.37)
ma1'=(mainf1-ma1)/tma
ma2'=(mainf2-ma2)/tma
hainf=1/(1+exp((v+78)/6))
tadef=1/(exp((v+46.05)/5)+exp(-(v+238.4)/37.45))

```

```

tah1=if(v<(-63))then(tadef)else(19)
tah2=if(v<(-73))then(tadef)else(60)
ha1'=(hainf-ha1)/tah1
ha2'=(hainf-ha2)/tah2
par ga=0
aux i_a=ia
#
# Ik2 slow potassium current
par gk2=0,fa=.4,fb=.6
Ik2=gk2*(v-ek)*mk2*(fa*hk2a+fb*hk2b)
minfk2=1/(1+exp(-(v+43)/17))^4
taumk2=1/(exp((v-80.98)/25.64)+exp(-(v+132)/17.953))+9.9
mk2'=(minfk2-mk2)/taumk2
hinfk2=1/(1+exp((v+58)/10.6))
tauhk2a=1/(exp((v-1329)/200)+exp(-(v+129.7)/7.143))+120
tauhk2b=if((v+70)<0)then(8930)else(tauhk2a)
hk2a'=(hinfk2-hk2a)/tauhk2a
hk2b'=(hinfk2-hk2b)/tauhk2b
aux i_k2=ik2
#
# IT and calcium dynamics -- transient low threshold
# permeabilites in 10-6 cm^3/sec
#
par Cao=2e-3,temp=23.5,pt=0,camin=50e-9
number faraday=96485,rgas=8.3147,tabs0=273.15
# CFE stuff
xi=v*faraday*2/(rgas*(tabs0+temp)*1000)
# factor of 1000 for millivolts
cfestuff=2e-3*faraday*xi*(ca-cao*exp(-xi))/(1-exp(-xi))
IT=pt*ht*mt^2*cfestuff
mtinf=1/(1+exp(-(v+52)/7.4))
taumt=.44+.15/(exp((v+27)/10)+exp(-(v+102)/15))
htinf=1/(1+exp((v+80)/5))
tauht=22.7+.27/(exp((v+48)/4)+exp(-(v+407)/50))
mt'=(mtinf-mt)/taumt
ht'=(htinf-ht)/tauht
# il L-type noninactivating calcium current -- high threshold
par pl=0
il=pl*ml^2*cfestuff
aml=1.6/(1+exp(-.072*(V+5)))
bml=.02*(v-1.31)/(exp((v-1.31)/5.36)-1)
ml'=aml*(1-ml)-bml*ml
aux i_l=il
# calcium concentration

```

```

par depth=.1,beta=1,area=29000
ca'=-.00518*(it+il)/(area*depth)-beta*(ca-camin)
ca(0)=50e-9
aux i_t=it
# ic calcium and voltage dependent fast potassium current
ic=gc*(v-ek)*mc
ac=250000*ca*exp(v/24)
bc=.1*exp(-v/24)
mc'=ac*(1-mc)-bc*mc
par gc=0
aux i_c=ic
# ih Sag current -- voltage inactivated inward current
ih=gh*(V-eh)*y
yinf=1/(1+exp((v+75)/5.5))
ty=3900/(exp(-7.68-.086*v)+exp(5.04+.0701*v))
y'=(yinf-y)/ty
par gh=0,eh=-43
# im Muscarinic slow voltage gated potassium current
im=gm*(v-ek)*mm
mminf=1/(1+exp(-(v+35)/10))
taumm=taumm_max/(3.3*(exp((v+35)/20)+exp(-(v+35)/20)))
mm'=(mminf-mm)/taumm
par gm=0,taumm_max=1000
aux i_m=im
# Iahp Calcium dependent potassium current
Iahp=gahp*(v-ek)*mahp^2
par gahp=0,bet_ahp=.001,al_ahp=1.2e9
mahp'=al_ahp*ca*ca*(1-mahp)-bet_ahp*mahp
aux i_ahp=iahp
aux cfe=cfestuff
# set up for 1/2 sec simulation in .5 msec increments
@ total=500,dt=.5,method=qualrk,atoler=1e-4,toler=1e-5,bound=1000
@ xhi=500,ylo=-100,yhi=50
init v=-70,hna=0.5
done

```

Traub fast dynamics with two types of SFA

```

itr(v,m,h,n)=gna*h*m^3*(v-ena)+(gk*n^4)*(v-ek)+gl*(v-el)
v'=(itr(v,m,h,n) -i+i_ca+i_ahp+i_m)/c
m'=am(v)*(1-m)-bm(v)*m
n'=an(v)*(1-n)-bn(v)*n
h'=ah(v)*(1-h)-bh(v)*h
w'=(winf(v)-w)/tw(v)

```

```

s'=alphas*(1-s)/(1+exp(-(v-vthr)/vsshp))-betas*s
# calcium
mlinf=1/(1+exp(-(v-vlth)/vshp))
i_ca=gca*mlinf*(v-eca)
ca'=(-alpha*i_ca-ca/tauca)
# k-ca
i_ahp=gahp*(ca/(ca+kd))*(v-ek)
i_m=gm*w*(v-ek)
#
#
am(v)=.32*(54+v)/(1-exp(-(v+54)/4))
bm(v)=.28*(v+27)/(exp((v+27)/5)-1)
ah(v)=.128*exp(-(50+v)/18)
bh(v)=4/(1+exp(-(v+27)/5))
an(v)=.032*(v+52)/(1-exp(-(v+52)/5))
bn(v)=.5*exp(-(57+v)/40)
#
TW(vs)=tauw/(3.3*EXP((vs-vwt)/20.0)+EXP(-(vs-vwt)/20.0))
WINF(vs)=1.0/(1.0+EXP(-(vs-vwt)/10.0))
#
init v=42.68904,m=.9935,n=.4645,h=.47785,w=.268,s=.2917,ca=.294
par ek=-100,ena=50,el=-67,eca=120
par gl=.2,gk=80,gna=100,gm=0
par c=1,i=0
par gahp=0,gca=1,kd=1,alpha=.002,tauca=80,phi=4
par vshp=2.5,vlth=-25,vsshp=2,vthr=-10
# reyes set vlth=-5,vsshp=10
par betas=.1,alphas=2
par vwt=-35,tauw=100
aux iahp=i_ahp
aux im=i_m
@ meth=qualrk,dt=.1,tol=1e-5,total=25.01,xlo=0,xhi=25,ylo=-85,yhi=50
@ bounds=1000000
done

```

Wang-Buszaki nearest neighbor chain

```

# wang buszaki fsu
p i0=0.5,ip=0,ton=20,toff=60
p phi=5.0
p gL=0.1
p EL=-65.0
p gNa=35.0
p ENa=55.0
p gK=9.0

```

```

p EK=-90.0
p gsyn=0.02,esyn=-80
#
table wr wbfreq.tab
i(x)=i0+i1*wr(x)
par i1=0.0035
v0=v1
v51=v50
s0=s1
s51=s50
V[1..50]'=-gL*(V[j]-EL)-gNa*(Minf(v[j])^3)*h[j]*(V[j]-ENa)-\
gK*(n[j]^4)*(V[j]-EK)+i([j])+gsyn*(esyn-v[j])*(s[j-1]+s[j+1])
h[1..50]'=phi*(alphah(v[j])*(1-h[j])-betah(v[j])*h[j])
n[1..50]'=phi*(alphan(v[j])*(1-n[j])-betan(v[j])*n[j])

s[1..50]'=ai(v[j])*(1-s[j])-s[j]/taui
#
ai(v)=ai0/(1+exp(-(v-vst)/vss))
par ai0=4,taui=6,vst=0,vss=5
#
alphan(v) = 0.1*(V+35.0)/(1.0-exp(-(V+35.0)/10.0))
betam(v) = 4.0*exp(-(V+60.0)/18.0)
Minf(v) = alphan(v)/(alphan(v)+betam(v))
#
alphah(v) = 0.07*exp(-(V+58.0)/20.0)
betah(v) = 1.0/(1.0+exp(-(V+28.0)/10.0))
#
alphan(v) = 0.01*(V+34.0)/(1.0-exp(-(V+34.0)/10.00))
betan(v) = 0.125*exp(-(V+44.0)/80.0)

#
#
V[1..50](0)=-64
h[1..50](0)=0.78
n[1..50](0)=0.09
#
@ XP=T
@ YP=V
@ TOTAL=200
@ DT=0.2,bound=10000
@ METH=qualrk
@ TOLER=0.00001
@ XLO=0.0, XHI=30.0, YLO=-90.0, YHI=30.0
done

```

Amari model - with inhibition

```
# Amari model with dynamic inhibition
# play with tau1
par sige=8,sigi=6
table je % 51 -25 25 exp(-(t/sige)^2)/(sige*sqrt(pi))
table ji % 51 -25 25 exp(-(t/sigi)^2)/(sigi*sqrt(pi))
hue[0..150]=heav(ue[j]-thr)
special ke=conv(even,151,25,je,hue0)
special ki=conv(even,151,25,ji,ui0)
ue[0..150]'=-ue[j]+ae*ke([j])-ki([j])
ui[0..150]'=(-ui[j]+ke([j]))/tau1
par tau1=.1
par thr=.05,ae=1.05
ue[50..75](0)=1
ui[50..75](0)=1
@ dt=.005,nout=20,total=50
done
```

Hansel-Sompolinski model

```
# simple ring model dynamics
# u' = -u + J* F(u)
# J = A + B cos(x-y)
#
par a=2,b=6
table cs % 100 0 99 cos(2*pi*t/100)
table sn % 100 0 99 sin(2*pi*t/100)
f(u)=sqrt(max(u-thr,0))
fu[0..99]=f(c0+c1*cs([j])+d1*sn([j]))
c0'=-c0+a*sum(0,99)of(shift(fu0,i'))*.01+p0
c1'=-c1+b*sum(0,99)of(shift(fu0,i')*cs(i'))*.01+p1*cos(w*t)
d1'=-d1+b*sum(0,99)of(shift(fu0,i')*sn(i'))*.01+p1*sin(w*t)
par thr=1
par p0=0,p1=0,w=0
done
```