Math models in neurobiology

Ben Bolker

Models of neuron excitation

This material will follow Edelstein-Keshet (2005) (E-K) closely: I was able to get the PDF from here, let me know if you want it and don't have access to it. (Much more detail in Ermentrout and Terman (2010).)

Limit cycles

- non-point attractor of deterministic systems; repeated trajectory, periodic orbits
- "any simple oriented closed curve trajectory that does not contain singular points"

Properties

- stable or unstable
- hard to get limit cycles in epidemic systems
  - orbits via stochastic perturbation of weakly stable spirals
  - orbits via seasonl forcing, ditto
  - plenty of math models with limit cycles but **usually weird** (e.g. Wang and Ruan (2004))
- Lotka-Volterra predator-prey system has **neutrally stable** cycles
  - but **Rosenzweig-MacArthur** model = predator-prey + density-dependent prey limitation, nonlinearity in predation rate does have limit cycles: see E-K §8.7

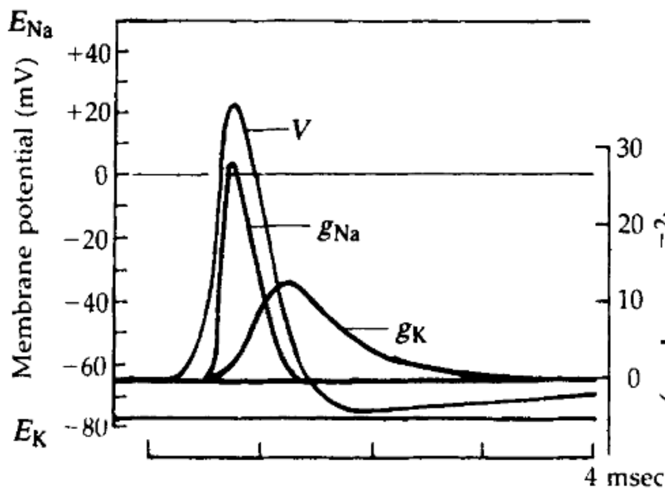
Limit cycles (part 2)

- can occur in any phase space >1D
- easiest to analyze in 2D (**Poincaré-Bendixson theorem**)

Neurons

- dendrites, soma, axon
- balance of ionic dynamics: Na+, K+, Cl-
  - axon -50 mV below environment in resting state
  - maintained by active ion pumping, e.g.
    - Na+: 30 vs 117 millimolar interior/exterior

    * K+: 90 vs 3 mmol
    * Cl-: 4 vs 120 mmol
    * A- ("other"): 0 vs 116 mmol



- sequence:
  - voltage increases
  - Na+ channels open, Na+ in (further ↗ V)
  - K+ channels open, K+ out (V ↘)
  - Na+ channels close
  - changes in V trigger firing at neighbouring site, wave propagates
- experiments:
  - **voltage clamp**: apply/measure spatially homogeneous V dynamics
  - **patch clamp**: measure dynamics of individual pores
- electric circuit analog:
  - Voltage drop (≈ battery) + resistor + capacitor
  - Several **parallel** currents (Na+, K+, etc.)
    * Ohm ($V = IR = I/g$, $g \equiv$ **conductance**)
    * Faraday ($V = q/C$) where $q \equiv$ **charge**)
    * $I = \sum V g_i = q/C$ (typo in E-K eq 4bb??)
    * $dV/dt = (dq/dt)/C = I/C = V/C \sum g_i$

Skipping a few steps:

$$\frac{dv}{dt} = -\frac{1}{C}\Big(g_{\text{Na}}(v)(v - v_{\text{Na}}) + g_{\text{K}}(v)(v - v_{\text{K}}) + g_{\text{L}}(v - v_{\text{L}})\Big)$$

(L= "everything else"; only g_{Na} and g_{K} are concentration-dependent)

- g_{Na} and g_{K} are **strongly** nonlinear functions of v
- $g_{\text{Na}} = \bar{g}_{\text{Na}} m^3 h$; $g_{\text{K}} = \bar{g}_{\text{K}} n^4$

$$
\begin{aligned}
\frac{dn}{dt} &= \alpha_n(v)(1-n) - \beta_n(v)n \\
\frac{dm}{dt} &= \alpha_m(v)(1-m) - \beta_m(v)m \\
\frac{dh}{dt} &= \alpha_h(v)(1-h) - \beta_h(v)h
\end{aligned}
$$

Help from here. (Didn't actually change much; found a typo.
Change in sign convention: $V \to -(V+65)$

```
parms0 <- c(g_bar_Na=120,g_bar_K=36,g_L=0.3, v_Na=-115, v_K=12, v_L=-10.5989,
           C=1,I=0)
alpha <- function(v,type) {
    switch(type,
           m=0.1*(v+25)/(exp((v+25)/10) -1),
           h=0.07*exp(v/20),
           n=0.01*(v+10)/(exp((v+10)/10) -1)
           )
}
beta <- function(v,type) {
    switch(type,
           m=4*exp(v/18),
           h=1/exp((v+30)/10 + 1),
           n=0.125*exp(v/80)
           )
}
HHgrad <- function(t,y,parms) {
    g <- with(as.list(c(y,parms)),
         c(v=-1/C*(-I + g_bar_Na*m^3*h*(v-v_Na) +
                   g_bar_K*n^4*(v-v_K) +
                   g_L*(v-v_L)),
           n=alpha(v,"n")*(1-n) - beta(v,"n")*n,
           m=alpha(v,"m")*(1-m) - beta(v,"m")*m,
           h=alpha(v,"h")*(1-h) - beta(v,"h")*h)
         )
    list(g)
}

y0 <- c(v=0,n=0.3,m=0.05,h=0.6)
HHgrad(0,y0,parms0)

## [[1]]
```

```
##            v            n            m            h
## -0.715470000  0.003238369  0.012385538  0.017010617
```

```
plot_hh <- function(h) {
    op <- par(mfrow=c(1,2),las=1,bty="l")
    plot(h[,1],h[,2], type="l",xlab="time",ylab="V")
    cvec <- c(1,2,4) ## colours
    matplot(h[,1],h[,3:5], type="l",lty=1,xlab="time",ylab="", col=cvec)
    legend("topright",legend=c("n","m","h"),lty=1,col=cvec)
}

library(deSolve)
res0 <- ode(y=y0,times=seq(0,60,by=0.05), func=HHgrad, parms=parms0)
plot_hh(res0)
```

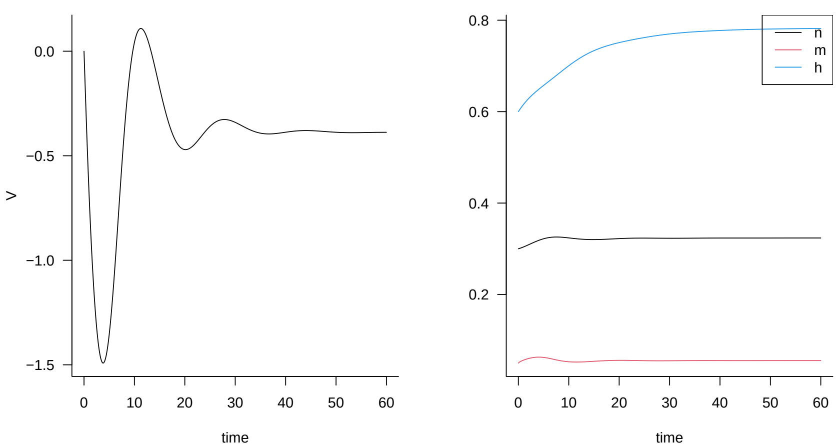


Figure 1: Hodgkin-Huxley (I=0)

```
parms_f <- function(I,parms=parms0) {
    parms[["I"]] <- I
    return(parms)
}
res2 <- ode(y=y0,times=seq(0,100,by=0.05), func=HHgrad, parms=parms_f(-7))
plot_hh(res2)
```

What about a **bifurcation diagram**?

```
get_maxmin <- function(I) {
    res <- ode(y=y0,times=c(0,seq(100,200,by=0.1)),
               func=HHgrad, parms=parms_f(I))
    res <- as.data.frame(res[-1,-1]) ## drop time and first row
    ans <- with(res,
                c(v_min=min(v),v_max=max(v),
```

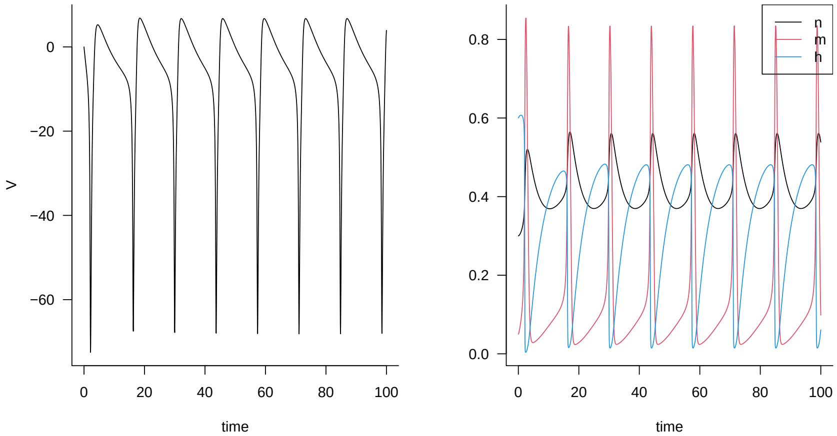


Figure 2: Hodgkin-Huxley (I=-7)

```
                    n_min=min(n),n_max=max(n),
                    m_min=min(m),m_max=max(m),
                    h_min=min(h),h_max=max(h)))
    return(ans)
}
Ivec <- seq(-1,-4,by=-0.05)
res <- t(sapply(Ivec,get_maxmin))
```

what do we now?

- project into lower dimensions
- simplification: separate into **slow** and **fast** components
- (find small values?)
- map changes in nullclines (how?)

```
library(phaseR)
HHeq <- res0[nrow(res0),c("v","n","m","h")]
HHgrad2d <- function(t, y, parms) {
    full_y <- c(v=y[["v"]],m=y[["m"]],n=HHeq[["n"]],h=HHeq[["h"]])
    list(HHgrad(t,full_y,parms)[[1]][c("v","m")])
}
HHgrad2d(0,y0[c("v","m")], parms0)
```

```
## [[1]]
##          v          m
## 0.21059895 0.01238554
```

```
phasePlaneAnalysis(HHgrad2d,xlim=c(-120,10),
                   parameters=parms,
                   state.names=c("v","m"),
                   ylim=c(0,1))
```

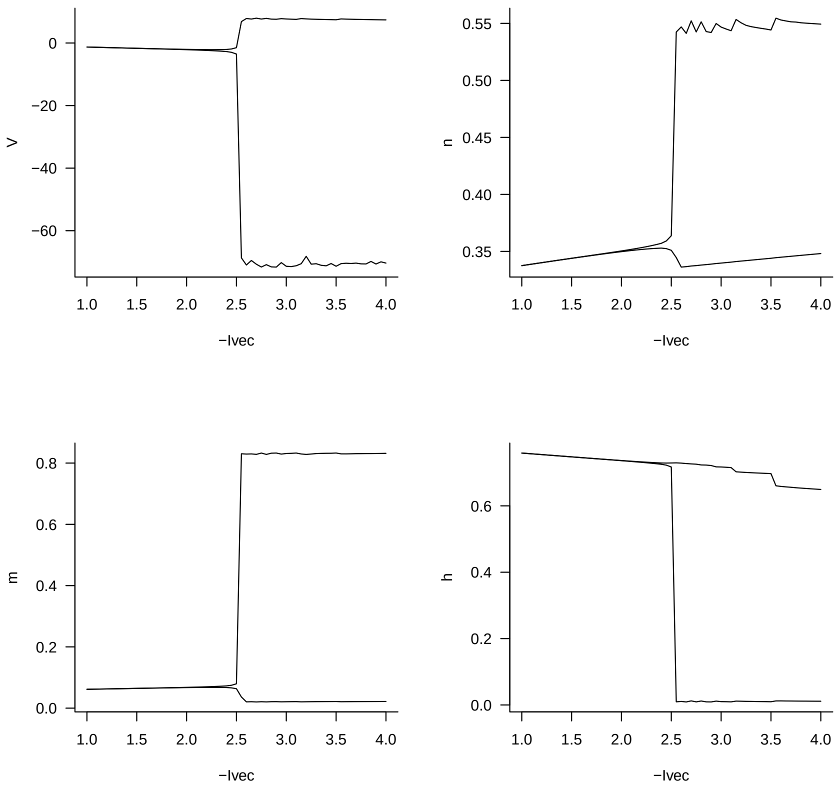


Figure 3: Hodgkin-Huxley bifurcation

Analysis: Poincaré-Bendixson

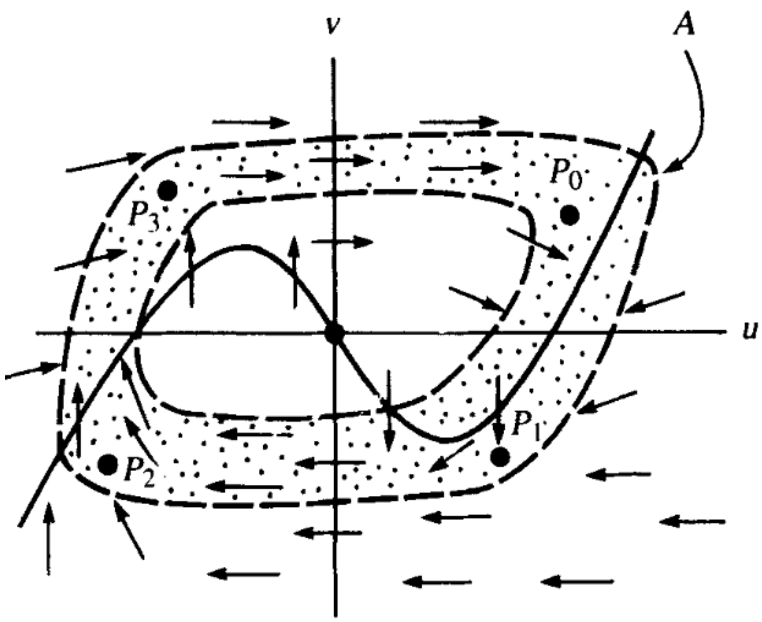
- bounded trajectories that don't approach a singular point are closed & periodic or approach a closed & periodic orbit
  - bounded region D that contains a single repelling (unstable) point, all flow inward
  - bounded annulus A containing no equilibria
- Bendixson: if D is simply connected, $\partial F/\partial x + \partial G/\partial y$ is not identically zero and doesn't change sign, then no closed orbits exist

Cubic nullclines

$$\frac{du}{dt} = v - G(u)$$
$$\frac{dv}{dt} = -u$$

and $G(u) = -(G(-u))$

- nullclines $v = G(u)$, $u = 0$



(*b*)

Fitzhugh-Nagumo model

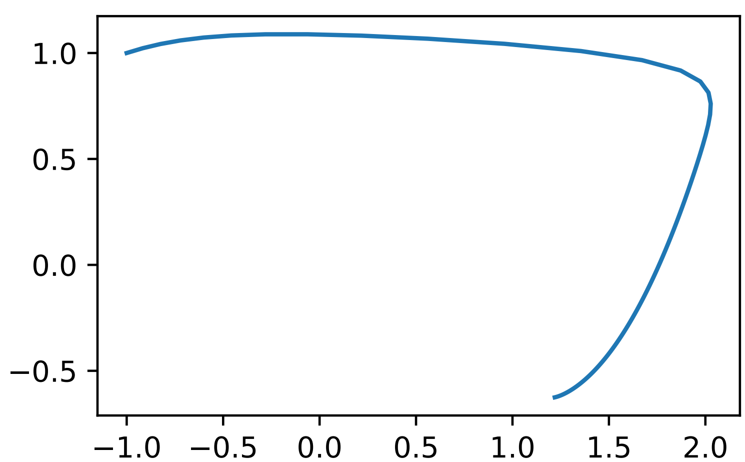
$$\frac{dx}{dt} = c\Big(y + x - x^3/3 + z(t)\Big) \quad \approx \text{Voltage}$$
$$\frac{dy}{dt} = -\frac{x - a + by}{c} \quad \approx \text{recovery}$$

Nullclines:

$$y = x^3 - x - z \quad (x' = 0)$$
$$y = (a - x)/b$$

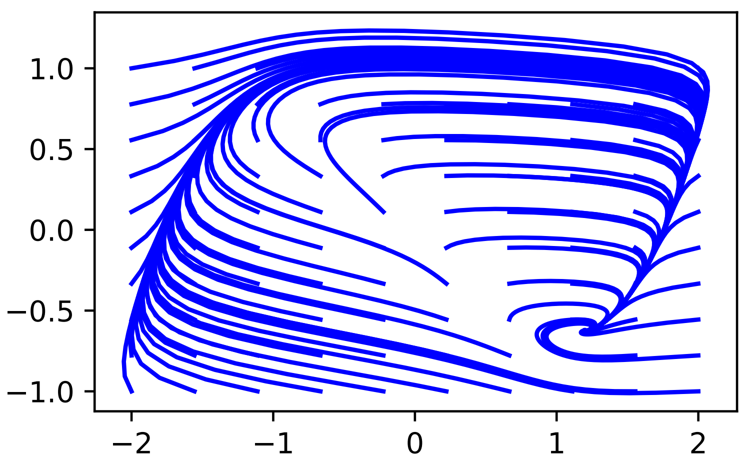
Fitzhugh-Nagumo in Python

```
import numpy as np
import scipy.integrate
def FH_grad(S,t,params):
   """gradient of FH model (autonomous)"""
   a,b,c,z = params   ## unpack parameters
   x,y = S             ## unpack state variables
   return(np.array([c*(y+x-x**3/3 + z),-(x-a+b*y)/c]))

t_vec = np.linspace(0,8,101)
params0 = (0.7,0.8,3,0) ## a,b,c,z
y0 = (-1,1)
FH_sol1 = scipy.integrate.odeint(FH_grad,
                                 y0=y0,
                                 t=t_vec,
                                 args=(params0,))

import matplotlib.pyplot as plt
plt.plot(FH_sol1[:,0], FH_sol1[:,1]);
plt.show()
```



```
def flowfield(params, xlim=(-2,2,10), ylim=(-1,1,10), tvec=np.linspace(0,8,101)):
    for x in np.linspace(xlim[0],xlim[1],xlim[2]):
        for y in np.linspace(ylim[0],ylim[1],ylim[2]):
            FH_sol = scipy.integrate.odeint(FH_grad,
                                            y0=(x,y),
                                            t=t_vec,
                                            args=(params,))
            plt.plot(FH_sol[:,0], FH_sol[:,1], 'b');
    plt.show()
    return None

flowfield(params0)
```

Figure 4: F-H flow field (z=0)




```
params1 = list(params0)
params1[3] = -0.4  ## set new z value
params1 = tuple(params1)
flowfield(params1)
```

van der Pol oscillator:

Rapid intro to chaos for Innocenti et al. (2007)

- ISI = inter-spike interval
- Poincaré map
- Floquet multipliers
- bounded trajectories with *sensitive dependence on initial conditions*
- positive *Lyapunov exponent* ($= \int \lambda(\tau)\, d\tau$)

continuation methods: Blyth, Renson, and Marucci (2020); PyDSTool

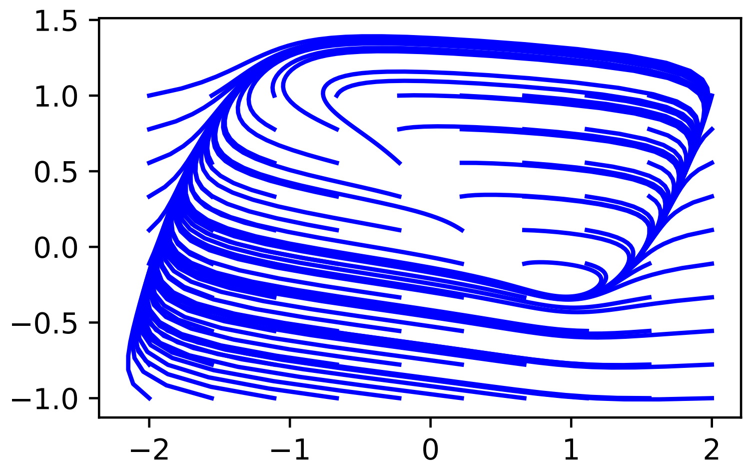


Figure 5: F-H flow field (z=0.4)

References

Blyth, Mark, Ludovic Renson, and Lucia Marucci. 2020. "Tutorial of Numerical Continuation and Bifurcation Theory for Systems and Synthetic Biology." *arXiv:2008.05226 [Q-Bio]*, August. http://arxiv.org/abs/2008.05226.

Edelstein-Keshet, Leah. 2005. "8. Limit Cycles, Oscillations, and Excitable Systems." In *Mathematical Models in Biology*, 311–80. Classics in Applied Mathematics. Society for Industrial; Applied Mathematics. https://doi.org/10.1137/1.9780898719147.ch8.

Ermentrout, G. Bard, and David H. Terman. 2010. *Mathematical Foundations of Neuroscience*. Vol. 35. Interdisciplinary Applied Mathematics. New York, NY: Springer New York. https://doi.org/10.1007/978-0-387-87708-2.

Innocenti, Giacomo, Alice Morelli, Roberto Genesio, and Alessandro Torcini. 2007. "Dynamical Phases of the Hindmarsh-Rose Neuronal Model: Studies of the Transition from Bursting to Spiking Chaos." *Chaos: An Interdisciplinary Journal of Nonlinear Science* 17 (4): 043128. https://doi.org/10.1063/1.2818153.

Wang, Wendi, and Shigui Ruan. 2004. "Bifurcations in an Epidemic Model with Constant Removal Rate of the Infectives." *Journal of Mathematical Analysis and Applications* 291 (2): 775–93. https://doi.org/10.1016/j.jmaa.2003.11.043.