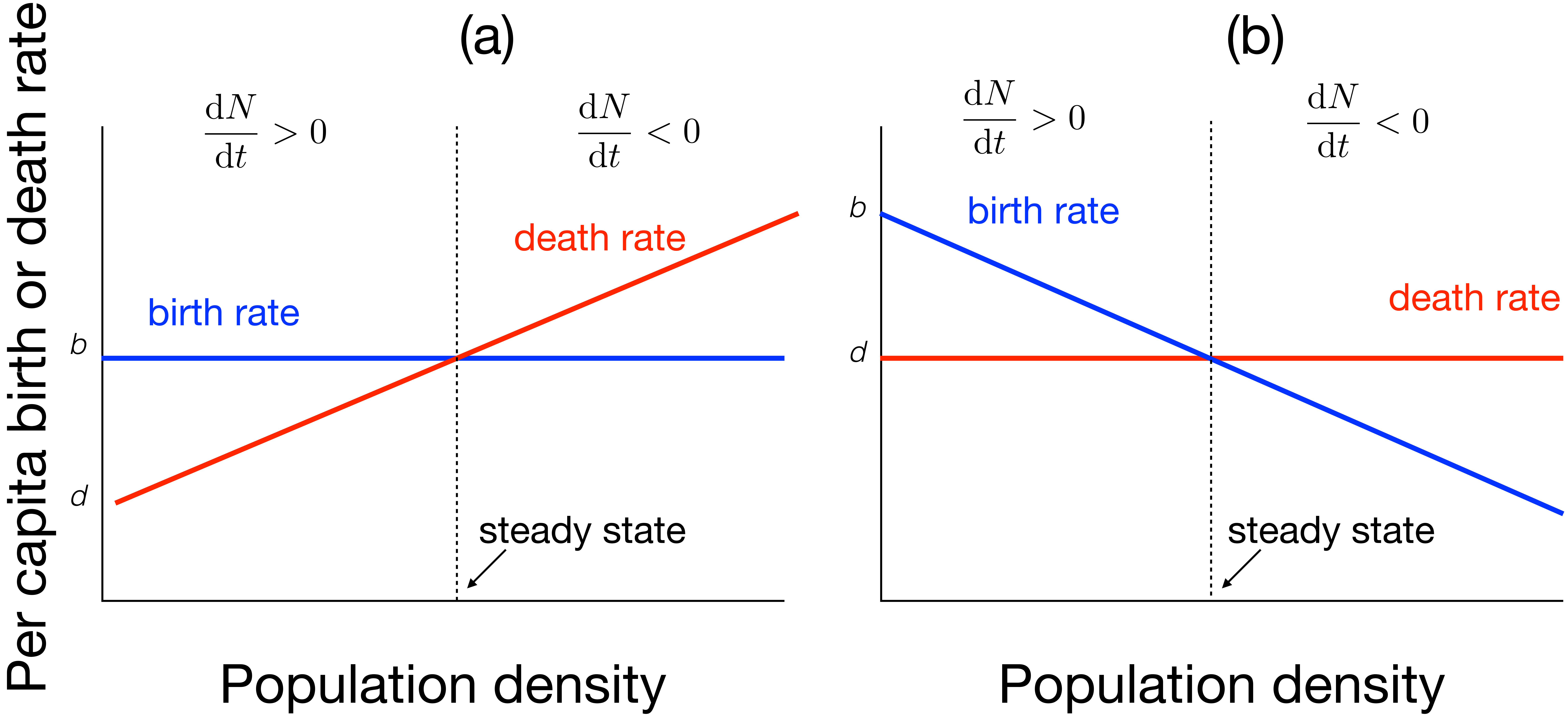
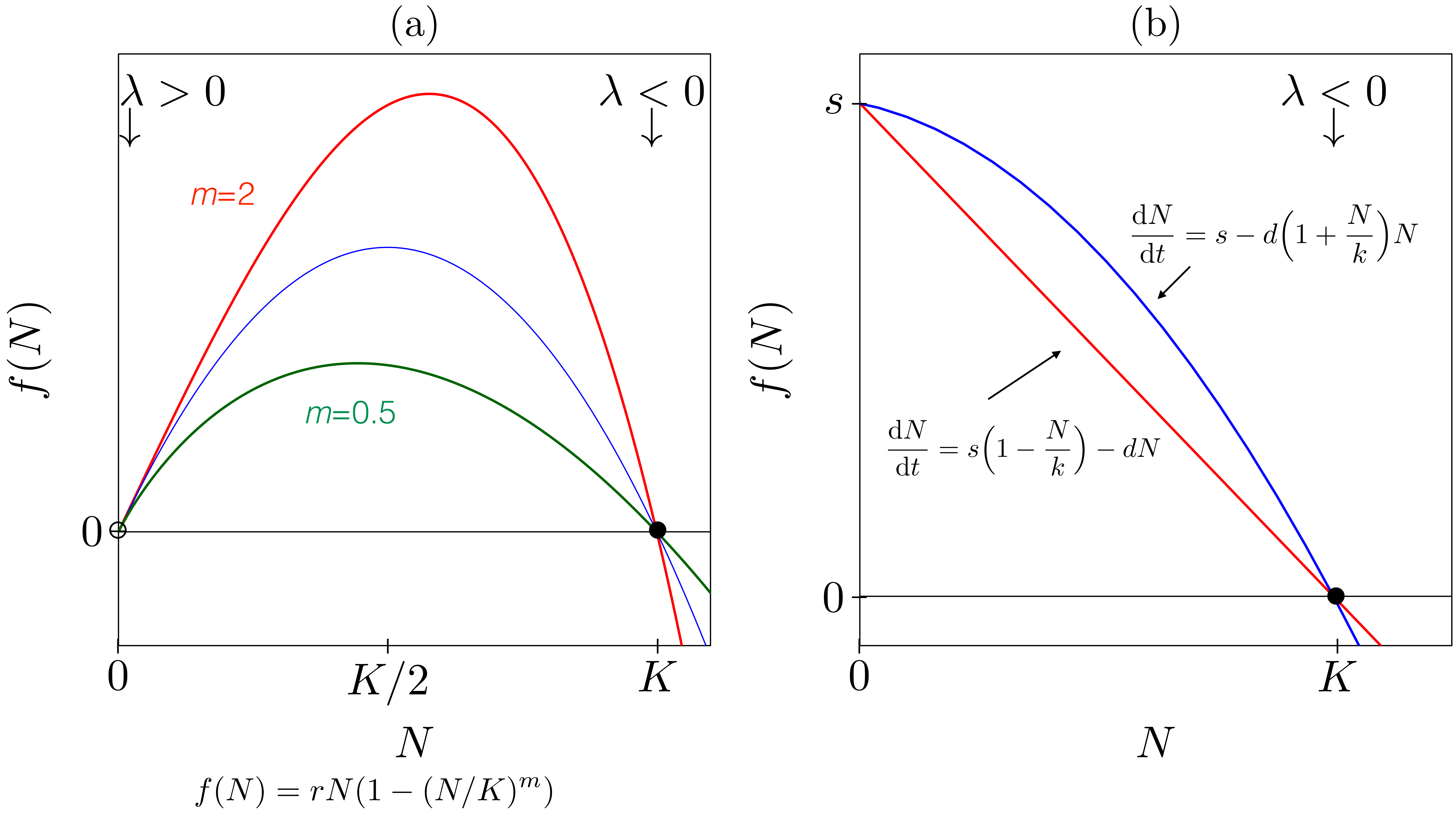
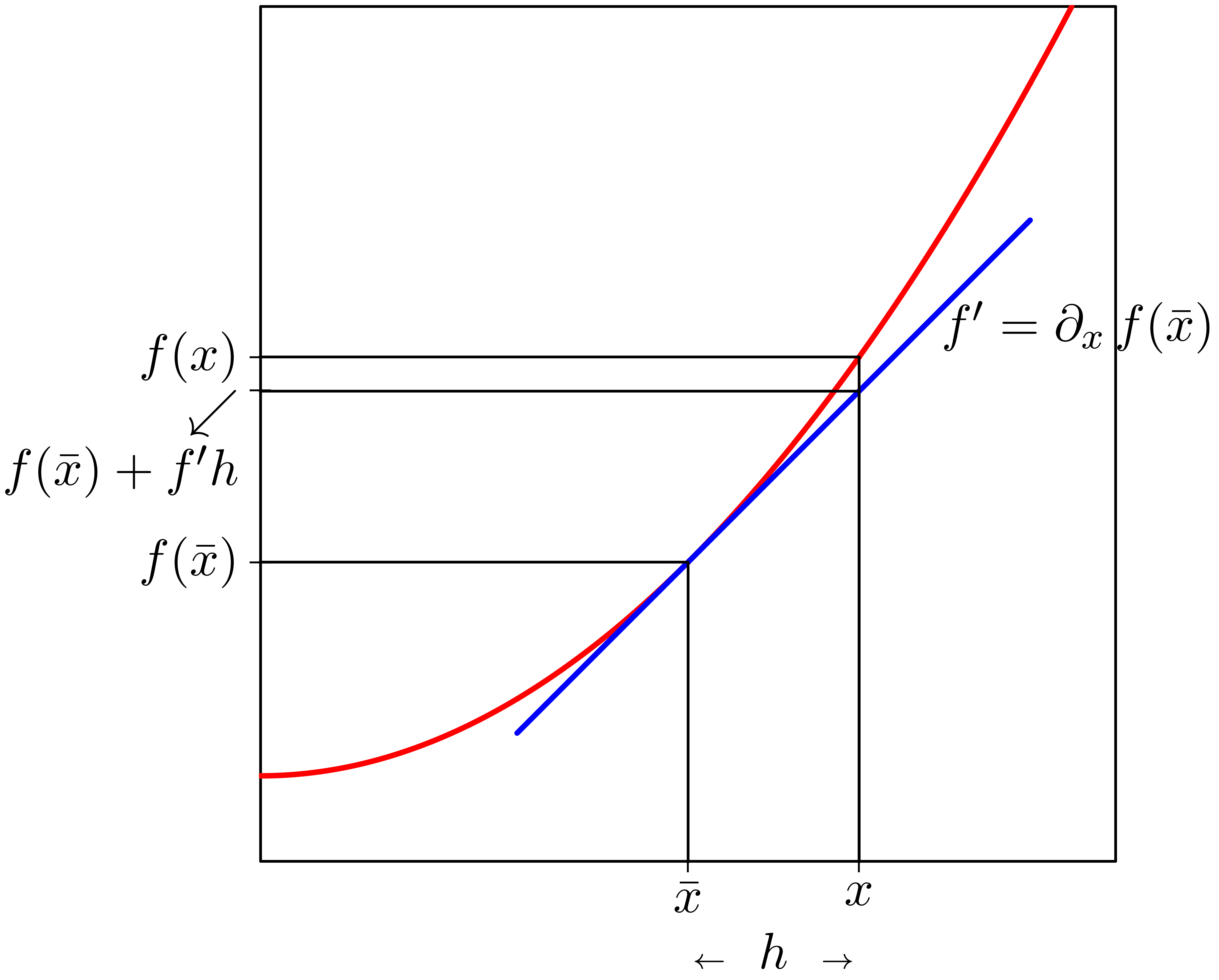
Chapter 4: Stability and return time





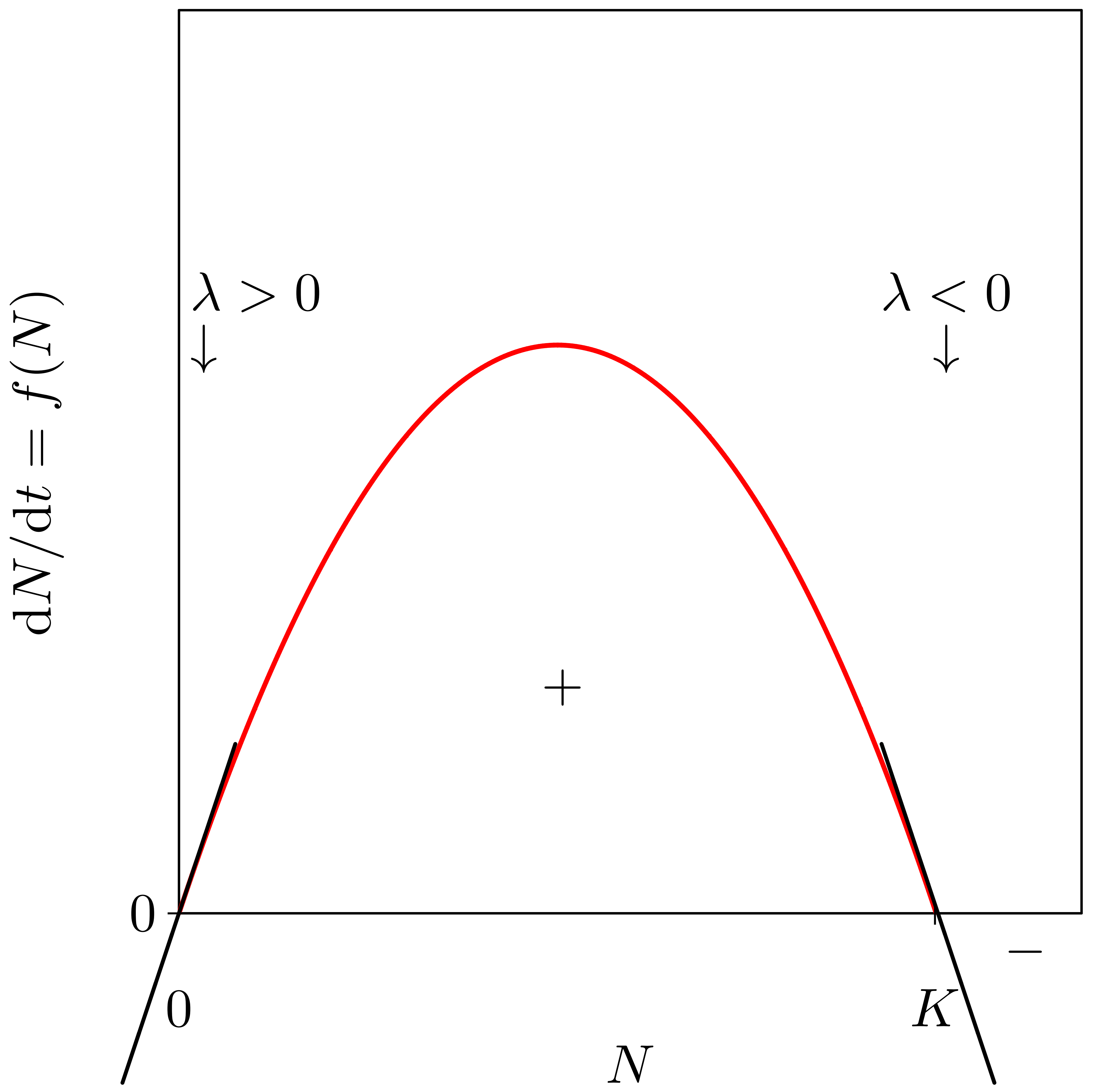
(a)
$\lambda > 0$
$\lambda < 0$
m=2
m=0.5
$f(N)$
0
0
$K/2$
K
N
$f(N) = rN(1-(N/K)^m)$
(b)
$\lambda < 0$
s
$\frac{\mathrm{d}N}{\mathrm{d}t} = s - d\left(1+\frac{N}{k}\right)N$
$\frac{\mathrm{d}N}{\mathrm{d}t} = s\left(1-\frac{N}{k}\right) - dN$
$f(N)$
0
0
K
N

Linearization



$$f(x) \simeq f(\bar{x}) + \partial_x f(\bar{x})\,(x - \bar{x})$$

Stability and Return time



$$h(t) = h(0)e^{\lambda t}$$

$$\frac{dN}{dt} = f(N) \simeq f(\bar{N}) + \partial_N f(\bar{N})\,(N - \bar{N}) = 0 + \lambda h\ ,$$

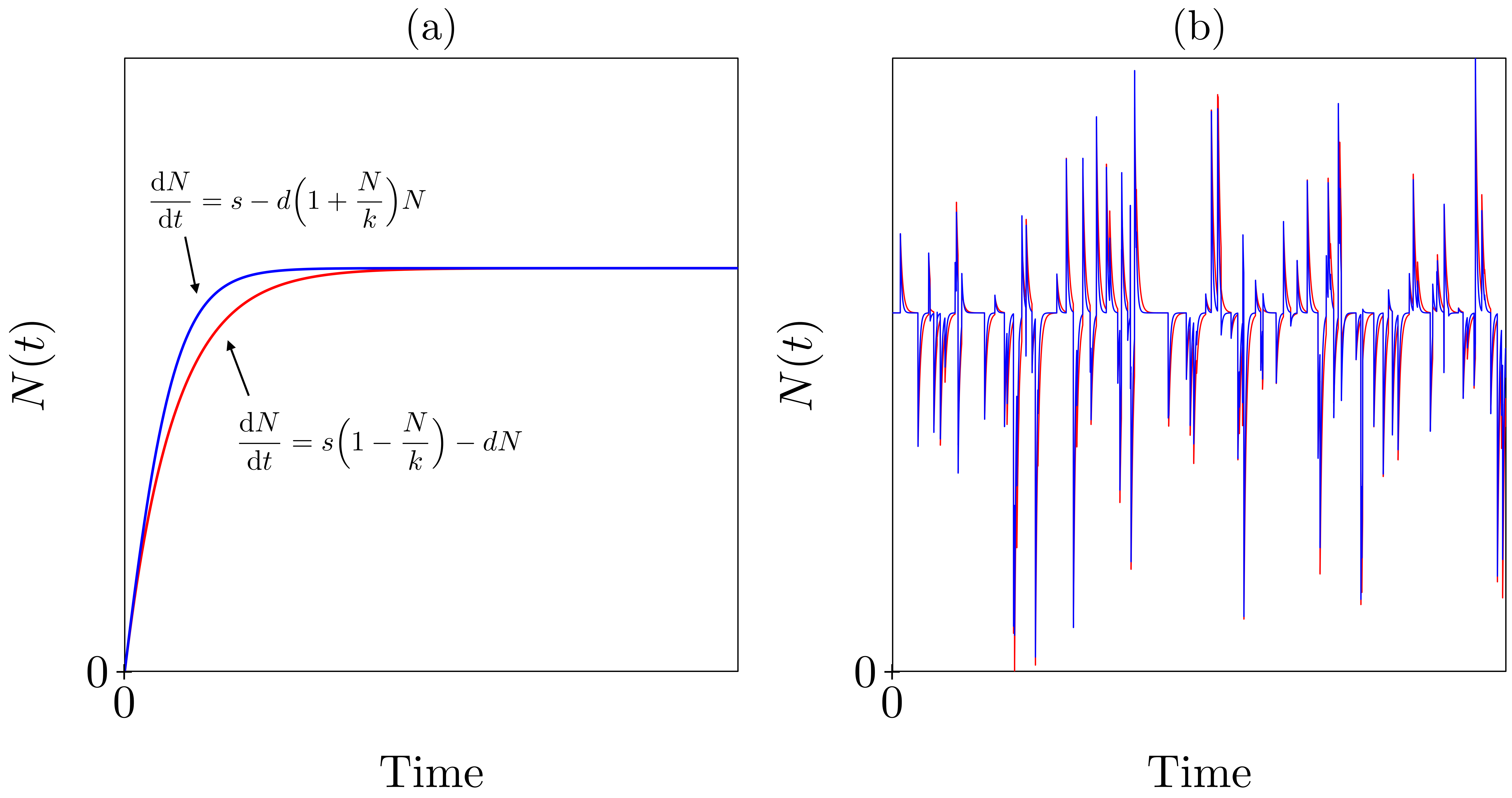
$$\frac{dN}{dt} = \frac{dN}{dt} - \frac{d\bar{N}}{dt} = \frac{d(N - \bar{N})}{dt} = \frac{dh}{dt}\ ,$$

$$\Rightarrow \quad \frac{dh}{dt} = \lambda h$$

Return time $T_R = -\frac{1}{\lambda}$

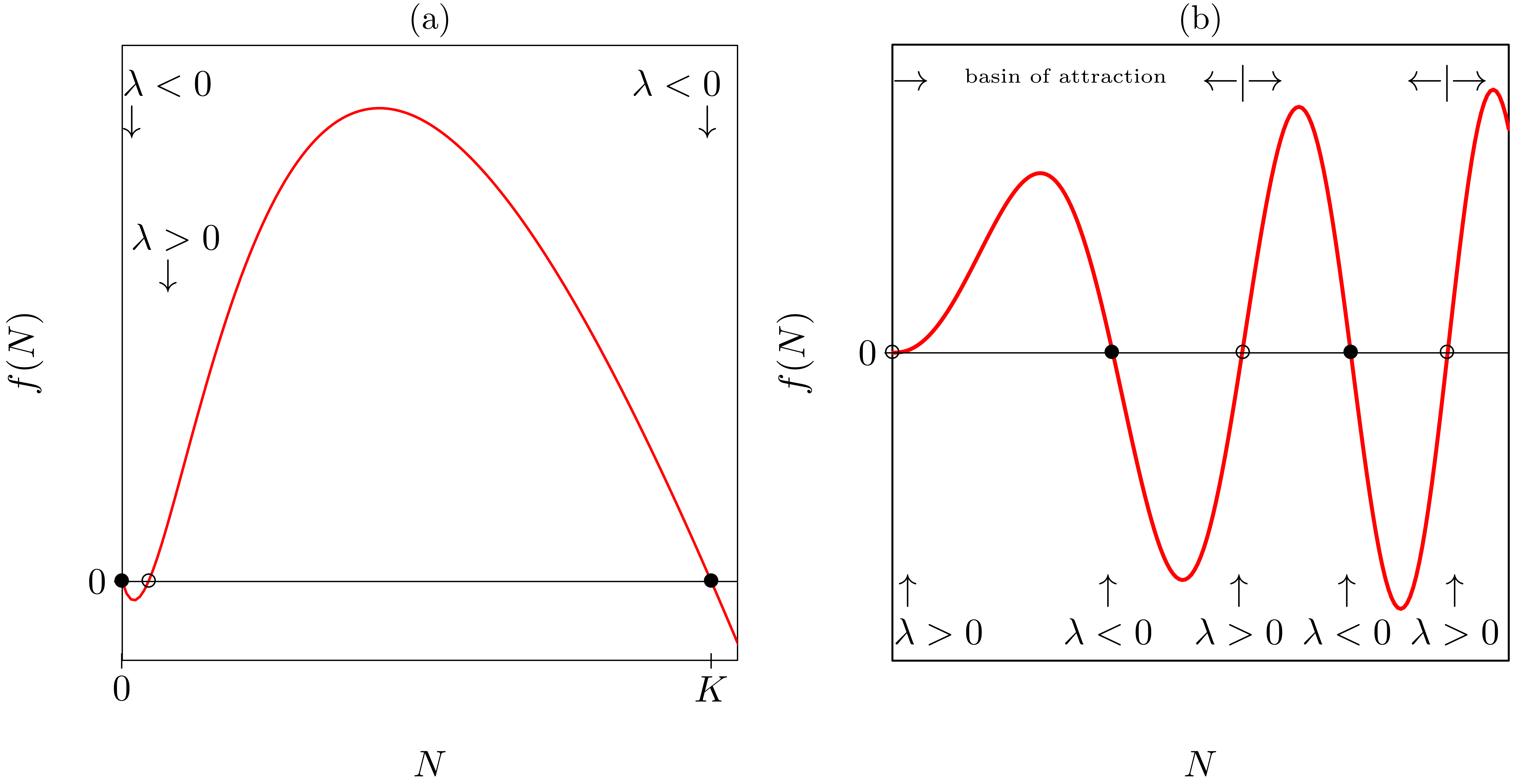
$$\frac{\mathrm{d}N}{\mathrm{d}t} = rN(1 - N/K)$$

$$\frac{\mathrm{d}N}{\mathrm{d}t} = rN(1 - [N/K]^m)$$

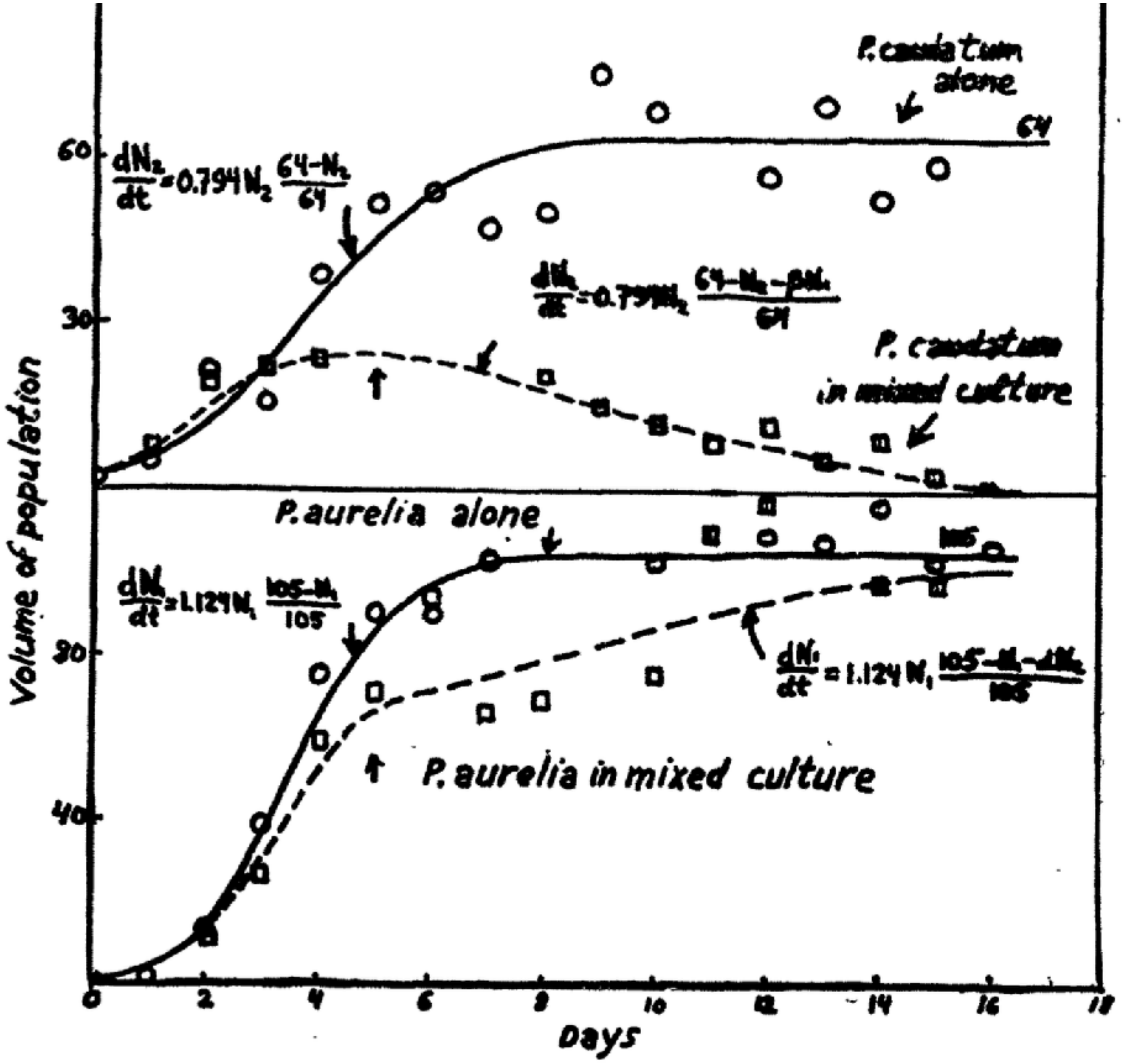


(a)
(b)
$\frac{\mathrm{d}N}{\mathrm{d}t} = s - d\left(1 + \frac{N}{k}\right)N$
$\frac{\mathrm{d}N}{\mathrm{d}t} = s\left(1 - \frac{N}{k}\right) - dN$
$N(t)$
0
Time

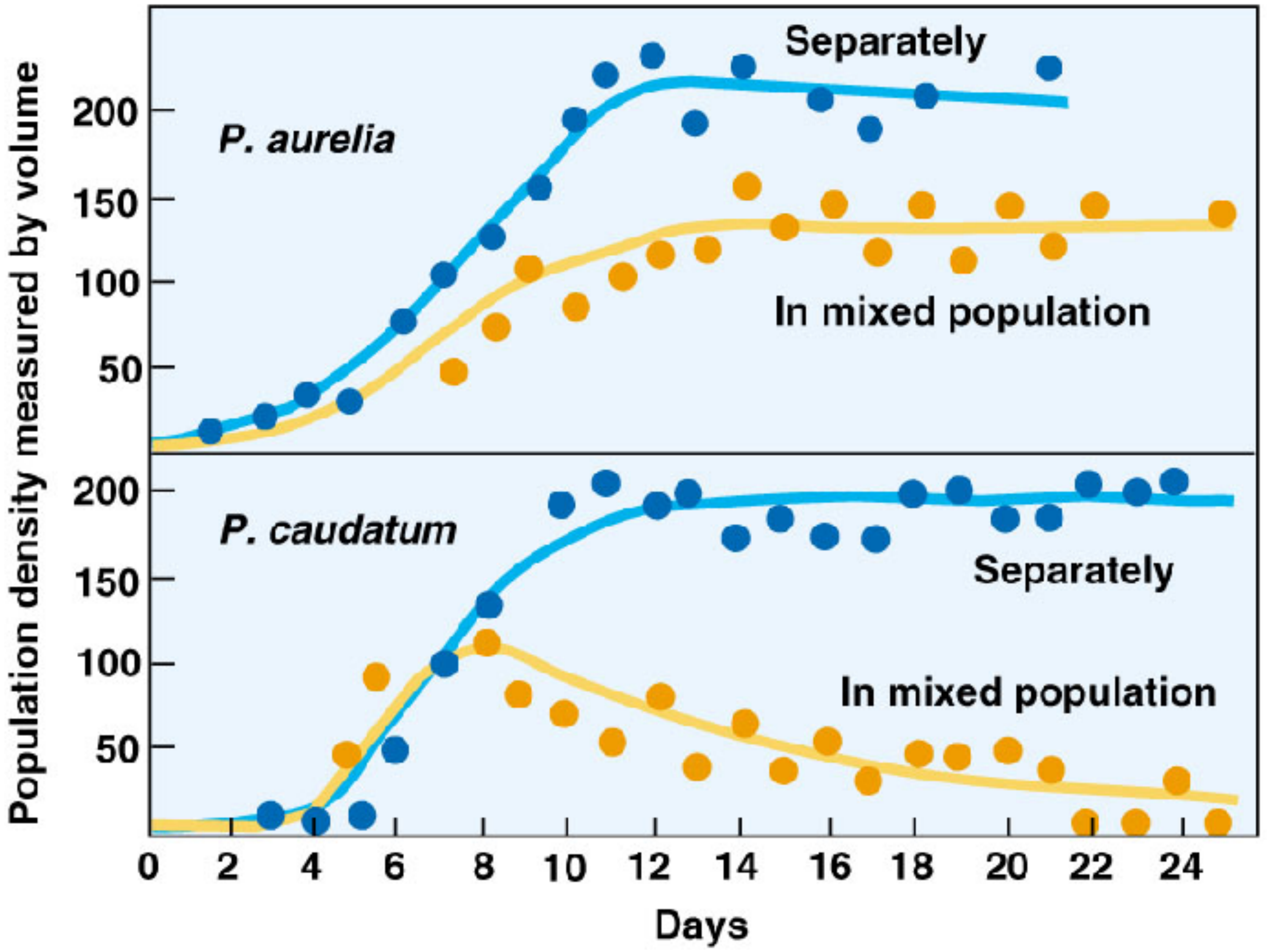
Basins of attraction



Fitting the competitive exclusion data for Gause [1934]



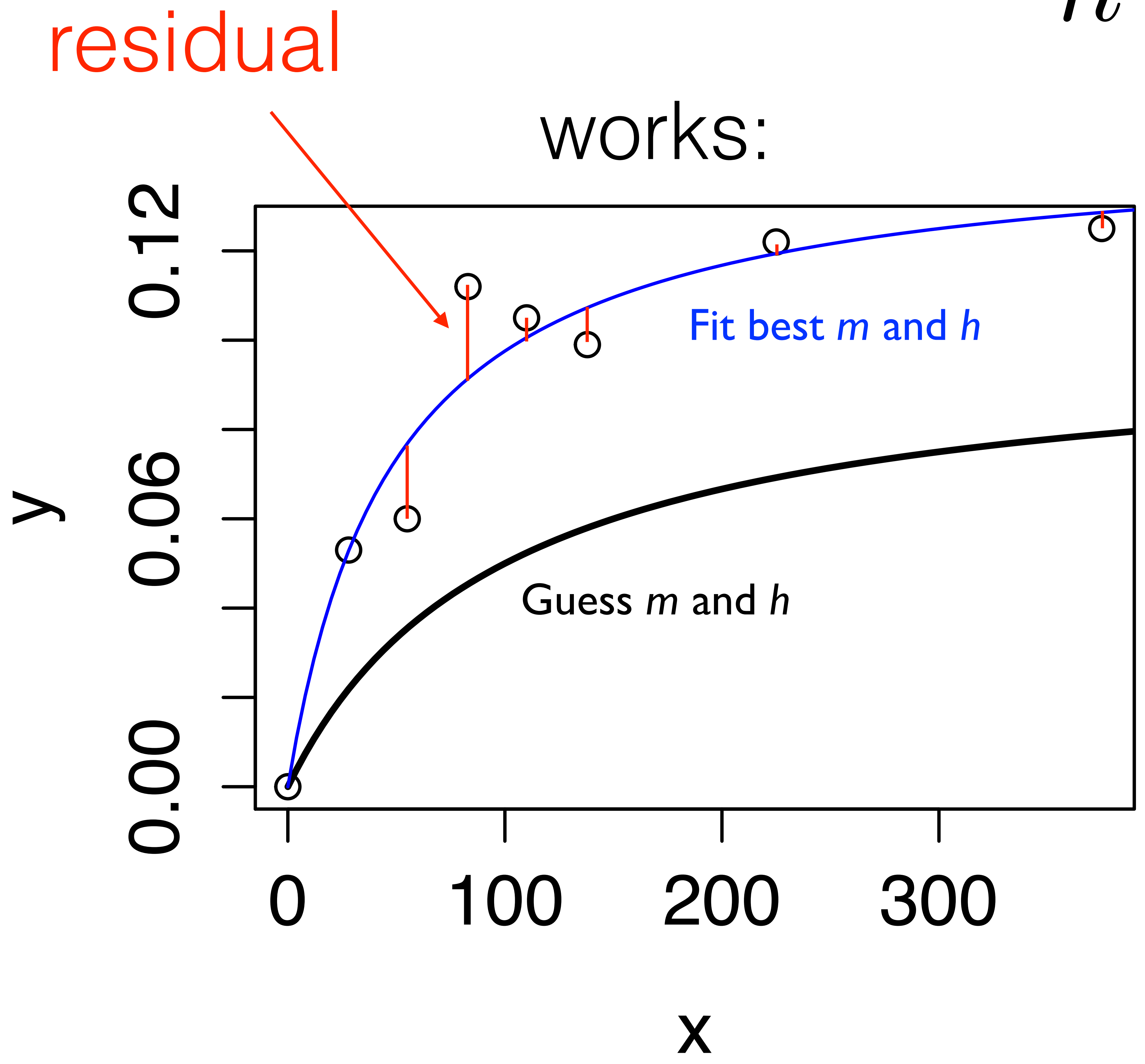
From: Gause 1934

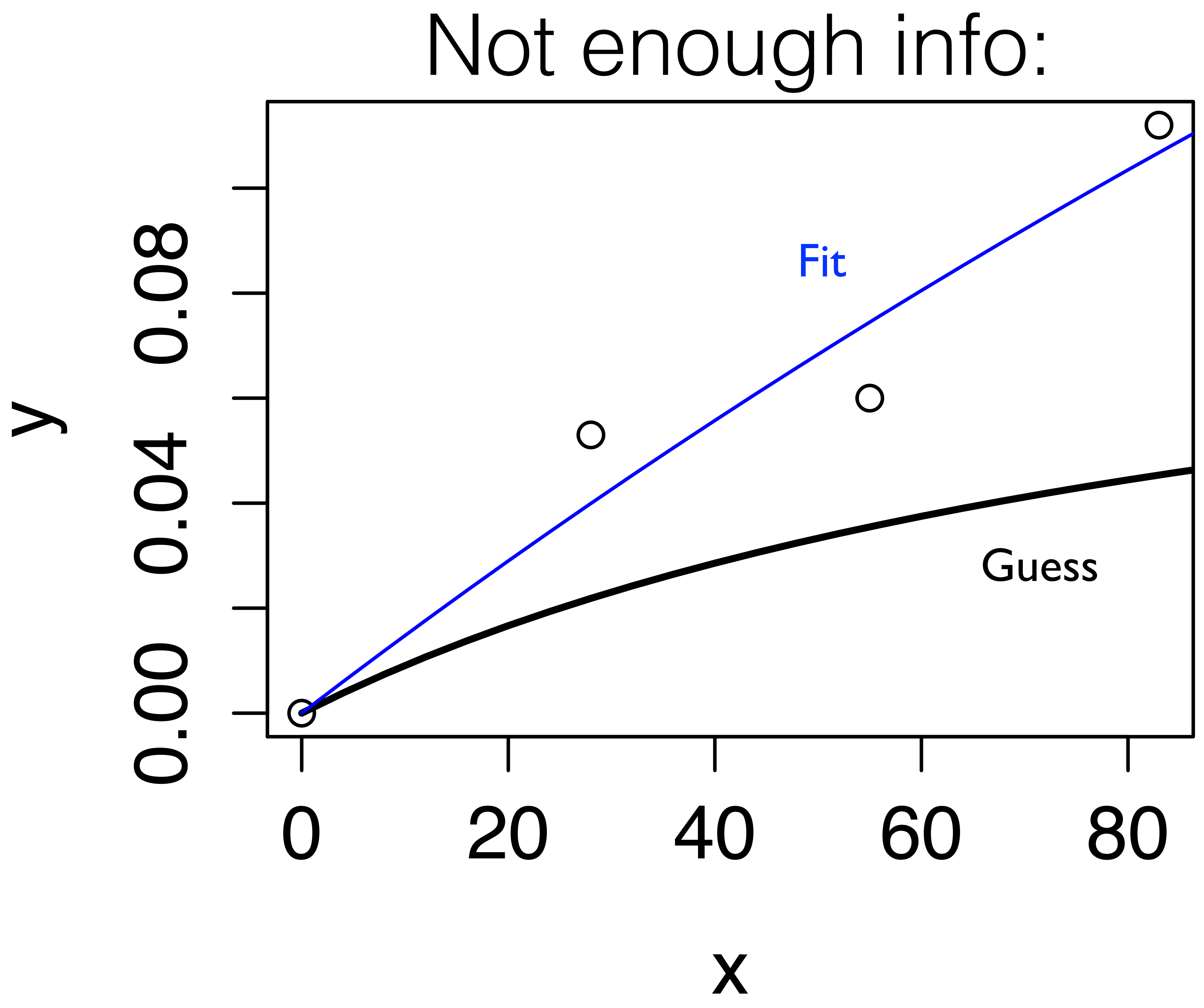


Non linear parameter optimization (Appendix 14.7)

Define a cost function based upon distance of model prediction to the data:

$$y = \frac{mx}{h + x}$$





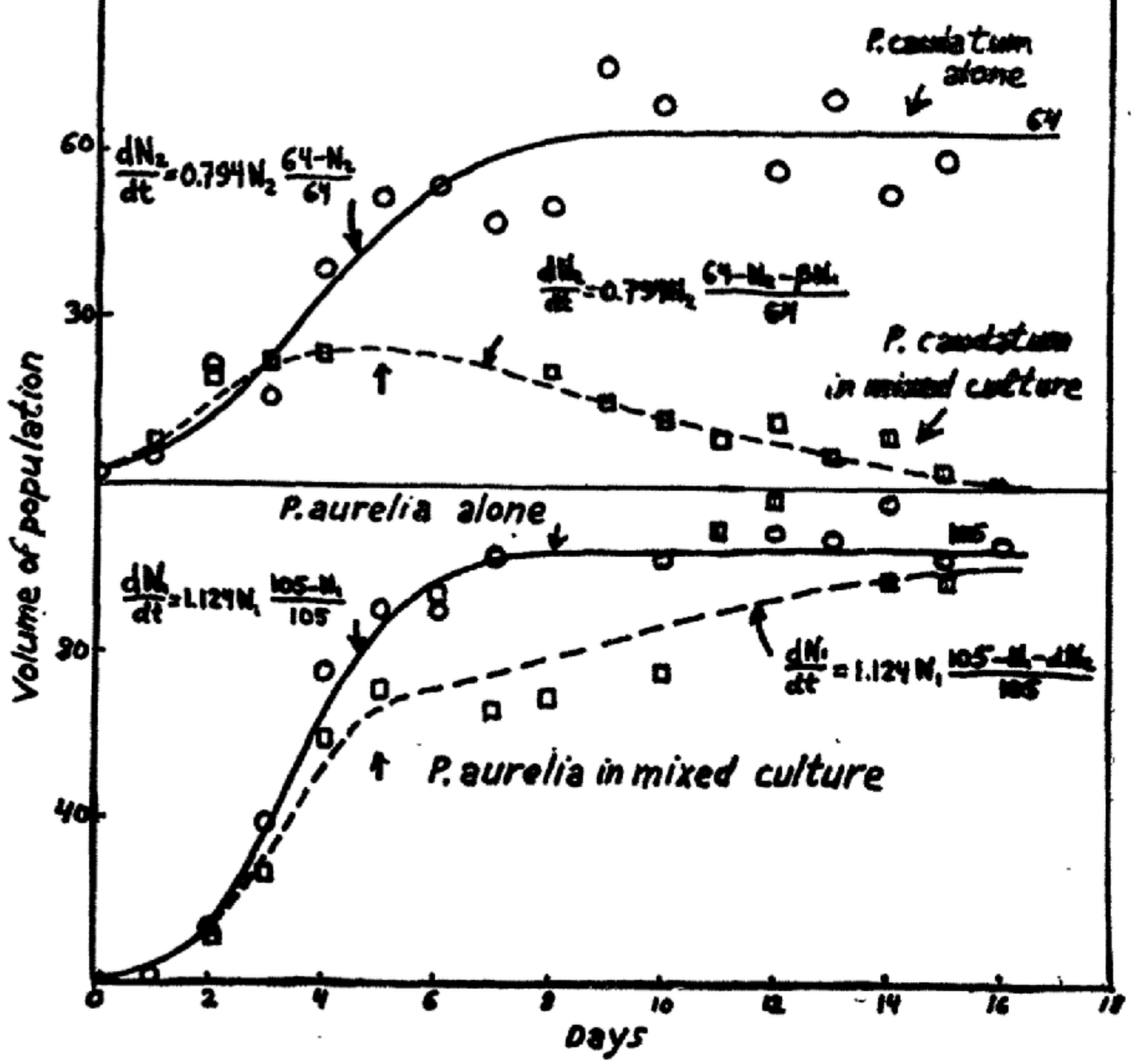
Non linear parameter optimization (Appendix 14.7)

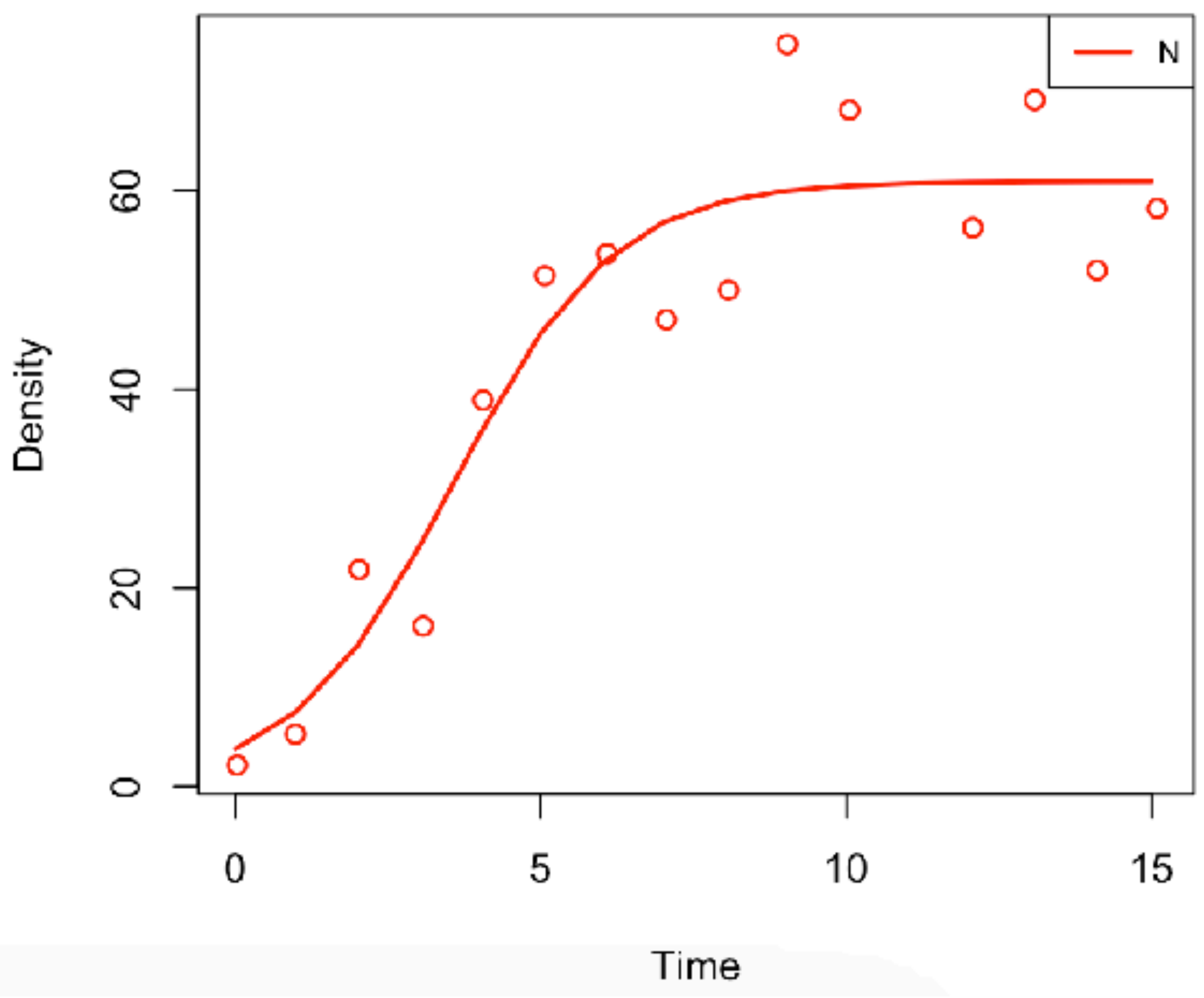
Function fit() in Grind minimizes the Summed Squared Residuals (SSR)

User needs to provide an initial guess for all parameters.
Grind gradually changes the free parameters.
(Gradient descent model: use partial derivatives)

Fitting stops at (local) optimum.

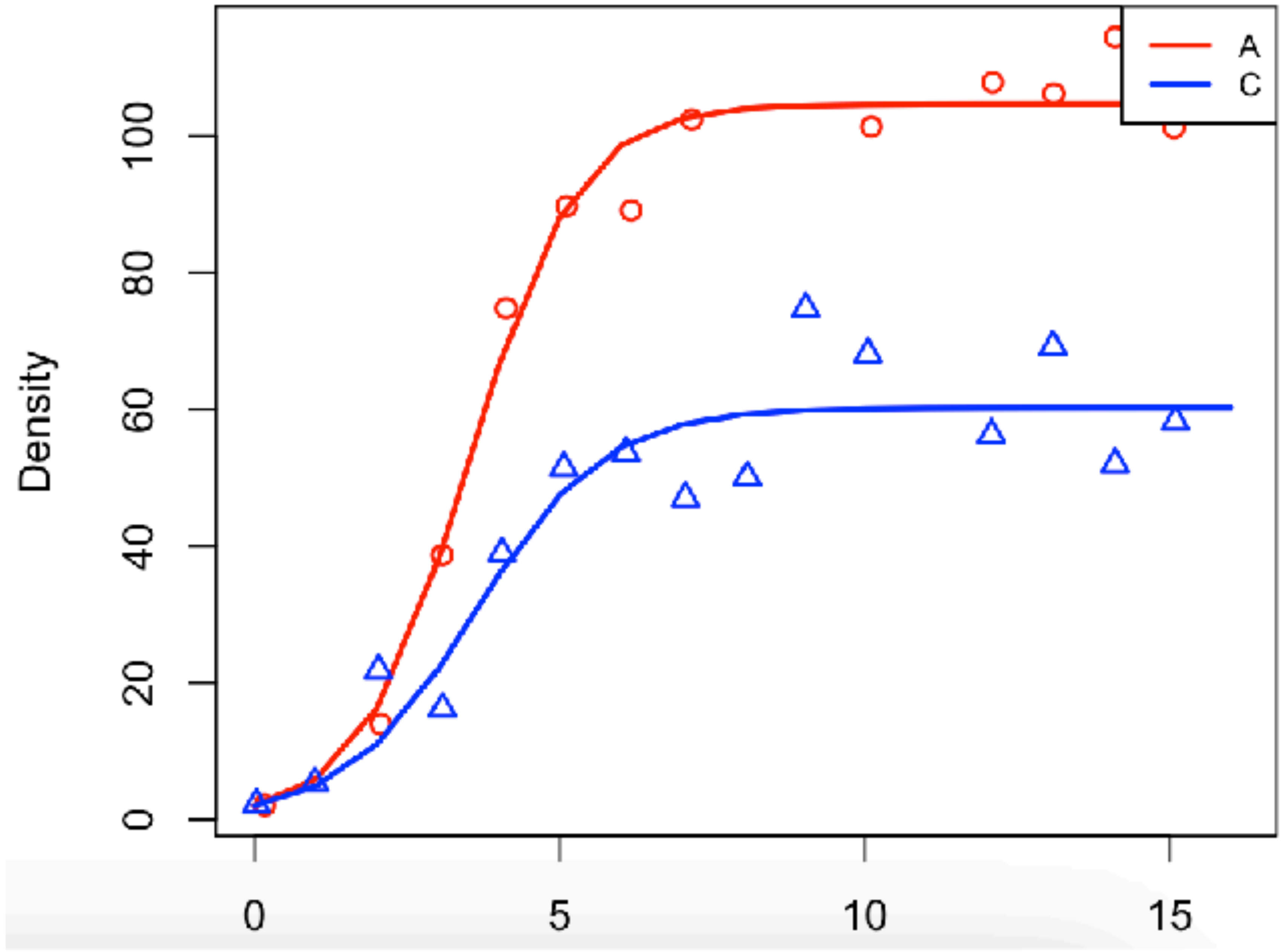
```r
model <- function(t, state, parms) {
  with(as.list(c(state,parms)), {
    dtN <- r*N*(1 - N/K)
    return(list(c(dtN)))
  })
}

p <- c(r=1,K=100)
s <- c(N=2)
run(18)

aurelia <- read.table("data/aurelia1.txt", header=TRUE)
caudatum <- read.table("data/caudatum1.txt", header=TRUE)

names(aurelia) <- c("time", "N")
names(caudatum) <- c("time", "N")

free <- c("r","K","N")
fA <- fit(aurelia,free=free)
```

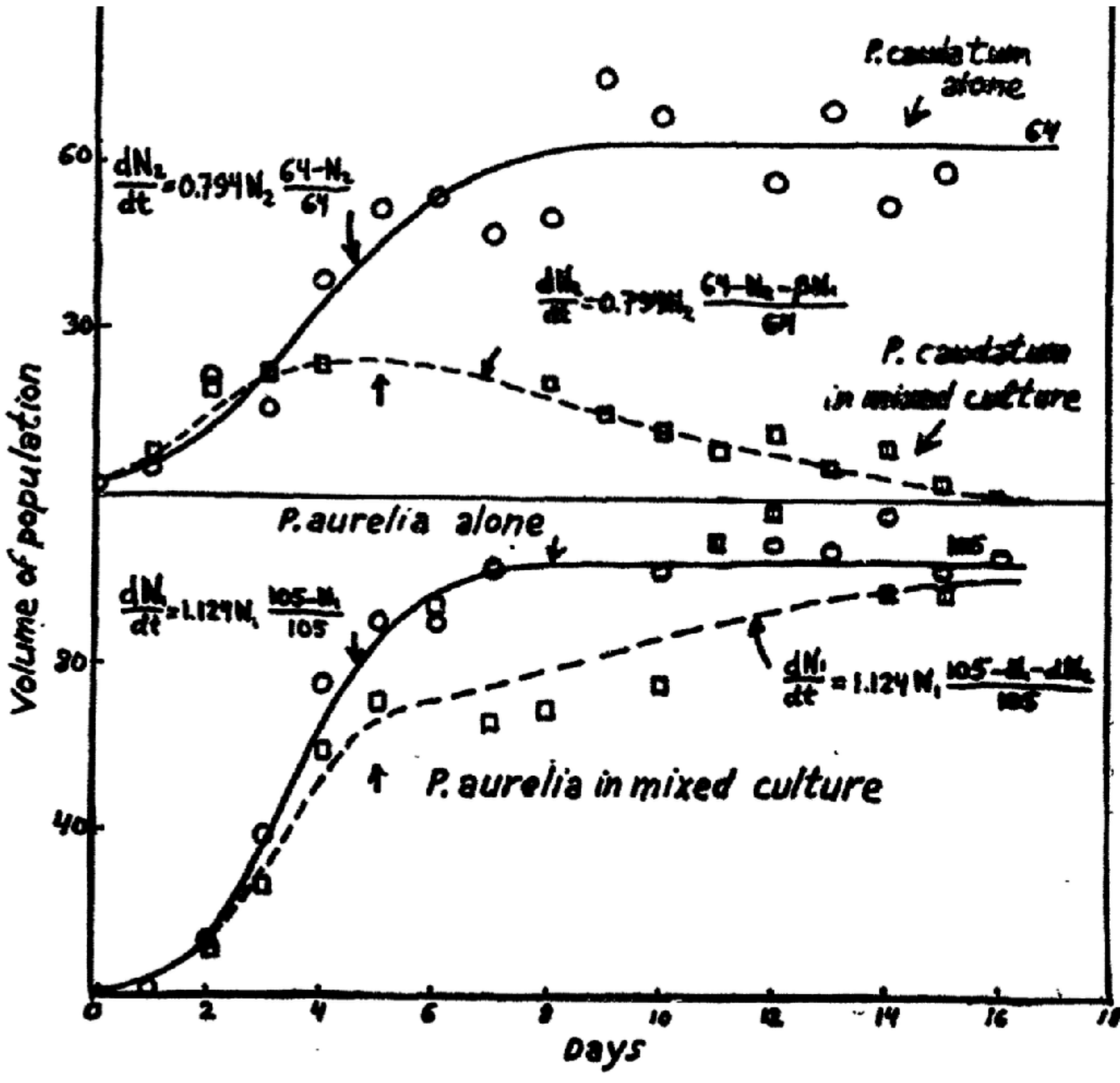




```r
model <- function(t, state, parms) {
  with(as.list(c(state,parms)), {
    dtA <- rA*A*(1 - (A+alpha*C)/kA)
    dtC <- rC*C*(1 - (C+beta*A)/kC)
    return(list(c(dtA,dtC)))
  })
}

p <- c(rA=1.1,rC=0.8,kA=105,kC=64,alpha=0,beta=0)
s <- c(A=2,C=2)
```

```r
free <- c("rA","kA","rC","kC")
f1 <- fit(list(aurelia1,caudatum1),free=free,add=TRUE)
p[free] <- f1$par;p
```





Cell division takes time

Conventional ODE: $$\mathrm{d}N/\mathrm{d}t = (p - d)N$$

Smith-Martin model (first ignoring death):

$$\frac{\mathrm{d}A(t)}{\mathrm{d}t} = 2pA_{t-\Delta} - pA(t) \quad \text{and} \quad \frac{\mathrm{d}B(t)}{\mathrm{d}t} = pA(t) - pA_{t-\Delta}$$

Smith-Martin model with death:

$$\frac{\mathrm{d}A(t)}{\mathrm{d}t} = 2pA_{t-\Delta}\mathrm{e}^{-d\Delta} - (p+d)A(t) \quad \text{and} \quad \frac{\mathrm{d}B(t)}{\mathrm{d}t} = pA(t) - dB(t) - pA_{t-\Delta}\mathrm{e}^{-d\Delta}$$

Cell division takes time

$$\frac{\mathrm{d}A(t)}{\mathrm{d}t} = 2pA_{t-\Delta}\mathrm{e}^{-d\Delta} - (p+d)A(t) \quad \text{and} \quad \frac{\mathrm{d}B(t)}{\mathrm{d}t} = pA(t) - dB(t) - pA_{t-\Delta}\mathrm{e}^{-d\Delta}$$

```r
sm <- function(t, state, parms) {
  with(as.list(c(state,parms)), {
    tlag <- t - Delta
    if (tlag < 0) lags <- 0
    else lags <- lagvalue(tlag,1)     # return lag of A
    dA <- -(p+d)*A + 2*p*lags[1]*exp(-Delta*d)
    dB <- p*A - d*B -  p*lags[1]*exp(-Delta*d)
    return(list(c(dA, dB)))
  })
}
```

Time delays implemented as many small steps

Smith-Martin model with death:

$$\frac{\mathrm{d}A(t)}{\mathrm{d}t} = 2pA_{t-\Delta}\mathrm{e}^{-d\Delta} - (p+d)A(t) \quad \text{and} \quad \frac{\mathrm{d}B(t)}{\mathrm{d}t} = pA(t) - dB(t) - pA_{t-\Delta}\mathrm{e}^{-d\Delta}$$

Smooth the time delay by many (*n*) small steps:

$$\frac{\mathrm{d}A}{\mathrm{d}t} = \frac{2n}{\Delta}B_n - (p+d)A\,, \quad \frac{\mathrm{d}B_1}{\mathrm{d}t} = pA - \left(d + \frac{n}{\Delta}\right)B_1 \quad \text{and} \quad \frac{\mathrm{d}B_i}{\mathrm{d}t} = \frac{n}{\Delta}\left(B_{i-1} - B_i\right) - dB_i$$

Scaling

$$\frac{\mathrm{d}N}{\mathrm{d}T} = rN[1 - N/k] \qquad n = N/k \quad \text{or} \quad N = kn$$

$$\frac{\mathrm{d}kn}{\mathrm{d}T} = k\frac{\mathrm{d}n}{\mathrm{d}T} = rkn[1 - kn/k] \quad \text{or} \quad \frac{\mathrm{d}n}{\mathrm{d}T} = rn[1 - n]$$

$$t = rT \quad \rightarrow \quad r\frac{\mathrm{d}n}{\mathrm{d}t} = rn[1 - n] \quad \text{or} \quad \frac{\mathrm{d}n}{\mathrm{d}t} = n[1 - n]$$

Exercise 5.8