Transient Models

T.J. Clark

2020-08-11

```
library(tidyverse)
library(nimble)
```

1. May (1977)'s alternative stable state model

See NIMBIOS transients code… Also see Boettiger (2018) Ecology Letters - can also be a "stochastic oscillator"

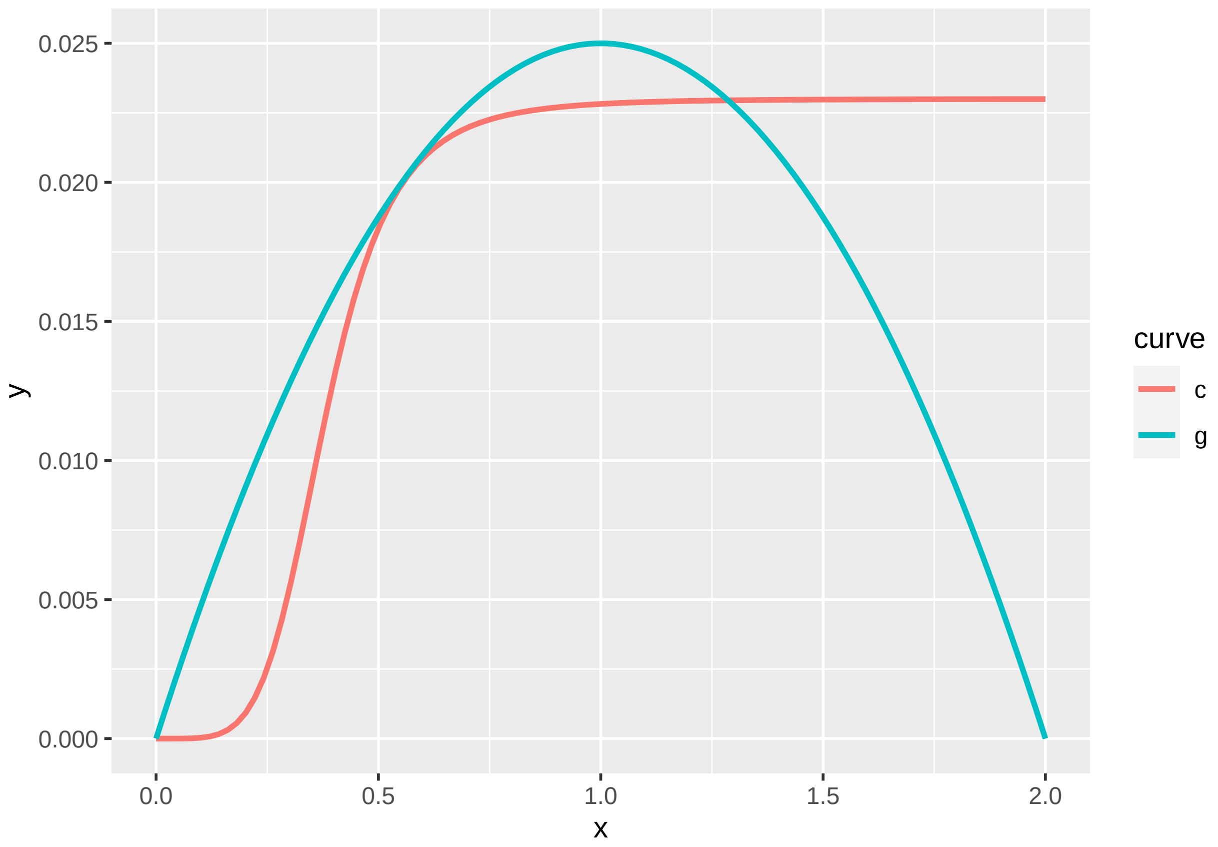
Deterministic Model

```
p <- list(r=0.05, K=2, Q=5, H=0.38, sigma=0.02, a=0.023, N=1000) # ghost params values
#p <- list(r=0.5, K=2, Q=5, H=0.38, sigma=0.04, a=0.245, N=1000) # stochastic oscillator values
growth <- function(x, p) x * p$r * (1 - x / p$K)
consumption <- function(x, p) p$a * x ^ p$Q / (x^p$Q + p$H ^ p$Q)
```

```
theory <-
  tibble(x = seq(0,2, length.out=100)) %>%
  mutate(g = growth(x,p),
         c = consumption(x,p)) %>%
  mutate(potential = - cumsum(g - c)) %>% # compute the potential fxn
  gather(curve, y, -x, -potential)
```

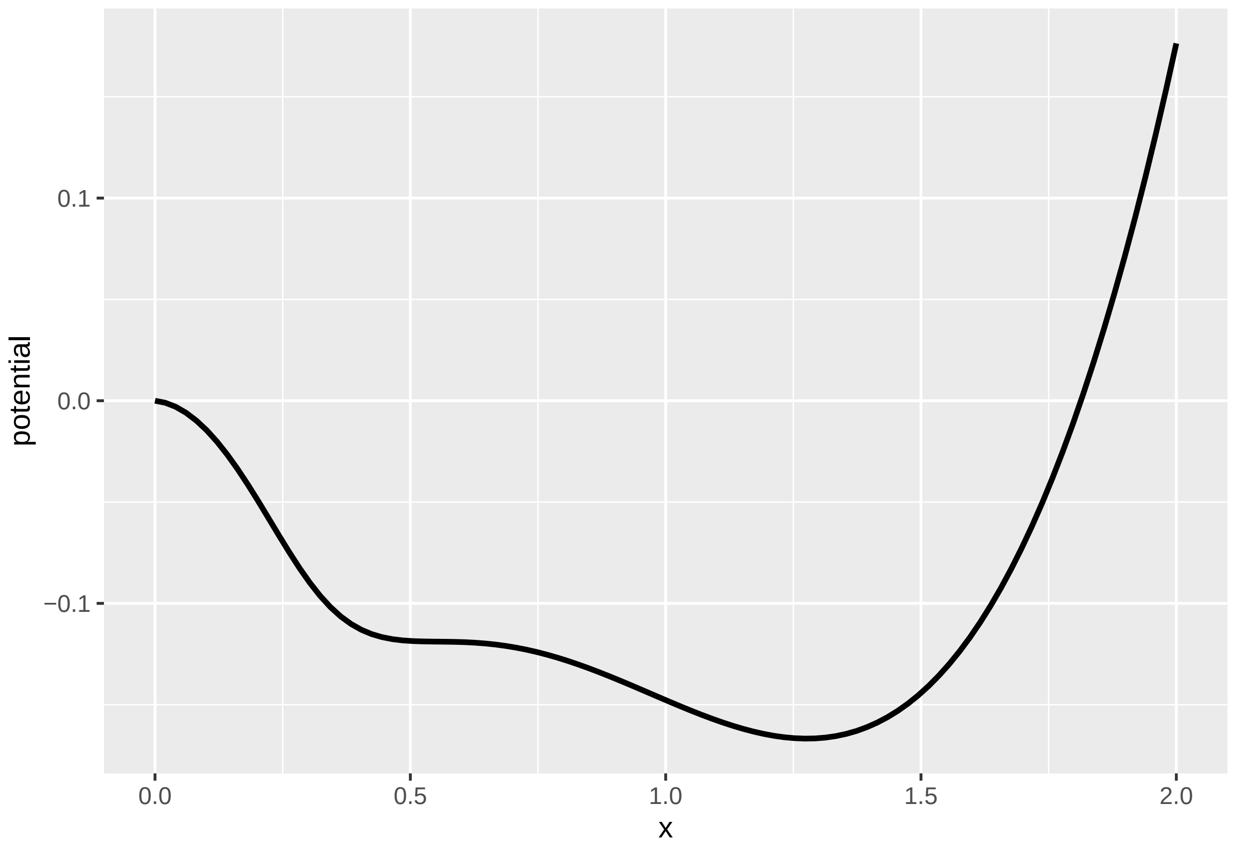
nonlinear birth and death processes intersect at points

```
theory %>%
  ggplot(aes(x, y, col=curve)) +
  geom_line(lwd=1)
```



Potential Function

```
theory %>%
  ggplot(aes(x, potential)) +
  geom_line(lwd=1)
```

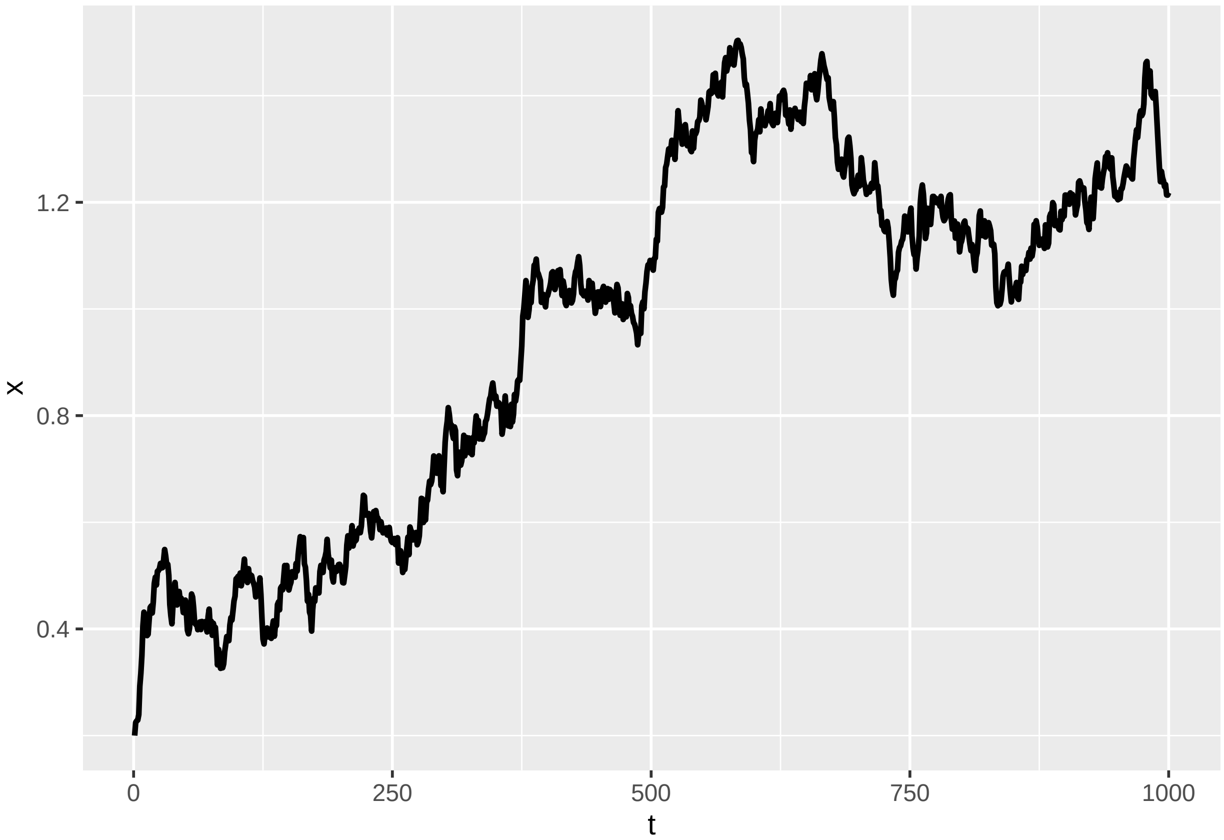


Stochastic Simulations via Nimble (BUGS notation)

```r
may <- nimble::nimbleCode({
  x[1] <- x0
  for(t in 1:(N-1)){
    mu[t] <- x[t] + x[t] * r * (1 - x[t] /K) - a* x[t] ^ Q / (x[t] ^ Q + H ^ Q)
    y[t+1] ~ dnorm(mu[t], sd = sigma)
    x[t+1] <- max(y[t+1],0) # can't go below 0
  }
})

model <- nimbleModel(may, constants = p, inits = list(x0 = 0.2))
cmodel <- model
set.seed(123456)
simulate(cmodel)
df <- tibble(t = seq_along(cmodel$x), x = cmodel$x)
```

```r
# graph
df %>% ggplot(aes(t,x)) + geom_line(lwd=1)
```

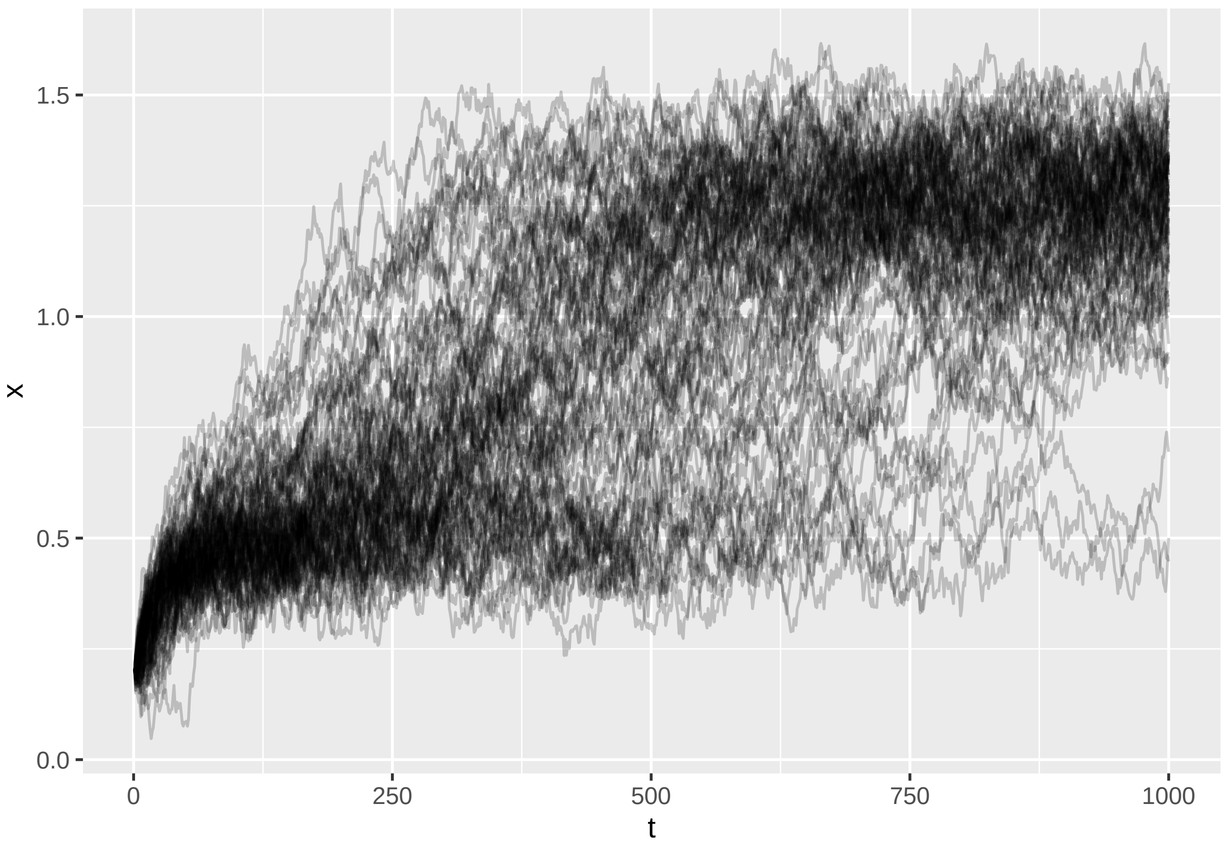


Let's do 100 replicates

```
df <- map_dfr(1:100, # essentially a for loop
              function(i){
                simulate(cmodel)
                tibble(t = seq_along(cmodel$x), x=cmodel$x, reps=i)
              })
```

plot

```
df %>%
  ggplot(aes(t,x, group = reps)) +
  geom_line(alpha=0.2)
```


1.5
1.0
0.5
0.0
x
0
250
500
750
1000
t