

AMATH 536 Problem Set 1

Cade Ballew #2120804

April 15, 2022

1 Problem 1

1.1 Part a

Consider a birth-death process in which $\lambda = \mu$ which starts with $a > 0$ individuals. Then, for $n \geq 1$, and a small time-interval Δt ,

$$p_n(t + \Delta t) = \lambda \Delta t (n - 1) p_{n-1}(t) + (1 - 2\lambda n \Delta t) p_n(t) + \lambda \Delta t (n + 1) p_{n+1}(t),$$

so

$$\frac{p_n(t + \Delta t) - p_n(t)}{\Delta t} = \lambda(n - 1)p_{n-1}(t) - 2\lambda n p_n(t) + \lambda(n + 1)p_{n+1}(t).$$

Taking the limit of both sides,

$$\frac{dp_n(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{p_n(t + \Delta t) - p_n(t)}{\Delta t} = \lambda(n - 1)p_{n-1}(t) - 2\lambda n p_n(t) + \lambda(n + 1)p_{n+1}(t).$$

For the $n = 0$ case,

$$p_0(t + \Delta t) = 1 \cdot p_0(t) + \lambda \Delta t p_1(t),$$

so

$$\frac{p_0(t + \Delta t) - p_0(t)}{\Delta t} = \lambda p_1(t).$$

Taking the limit of both sides,

$$\frac{dp_0(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{p_0(t + \Delta t) - p_0(t)}{\Delta t} = \lim_{\Delta t \rightarrow 0} \lambda p_1(t) = \lambda p_1(t).$$

Then, by definition,

$$\begin{aligned}
\frac{\partial P(s, t)}{\partial t} &= \sum_{n=0}^{\infty} \frac{dp_n(t)}{dt} s^n = \lambda \sum_{n=2}^{\infty} (n-1) p_{n-1}(t) s^n - 2\lambda \sum_{n=1}^{\infty} n p_n(t) s^n + \lambda \sum_{n=0}^{\infty} (n+1) p_{n+1}(t) s^n \\
&= \lambda s^2 \sum_{n=2}^{\infty} p_{n-1}(t) \frac{ds^{n-1}}{ds} - 2\lambda s \sum_{n=1}^{\infty} p_n(t) \frac{ds^n}{ds} + \lambda \sum_{n=0}^{\infty} p_{n+1}(t) \frac{ds^{n+1}}{ds} \\
&= \lambda s^2 \sum_{n=1}^{\infty} p_n(t) \frac{ds^n}{ds} - 2\lambda s \sum_{n=1}^{\infty} p_n(t) \frac{ds^n}{ds} + \lambda \sum_{n=1}^{\infty} p_n(t) \frac{ds^n}{ds} = \lambda(s-1)^2 \sum_{n=1}^{\infty} p_n(t) \frac{ds^n}{ds} \\
&= \lambda(s-1)^2 \frac{\partial P(s, t)}{\partial s}.
\end{aligned}$$

Note that this has initial condition

$$P(s, 0) = s^a$$

since $p_a(0) = 1$ and $p_b(0) = 0$ for $b \neq a$.

1.2 Part b

Using Mathematica, we find that this PDE has solution

$$P(s, t) = \left(\frac{\lambda t(s-1) - s}{\lambda t(s-1) - 1} \right)^a$$

which defines the PGF. See Appendix A for the Mathematica code utilized in this problem.

1.3 Part c

Now, we assume that $a = 1$. Using Mathematica, we compute for $n \geq 1$

$$p_n(t) = \frac{1}{n!} \frac{\partial^n P(0, t)}{\partial s^n} = \frac{\left(\frac{t\lambda}{1+t\lambda} \right)^{n-1}}{(1+t\lambda)^2} = \frac{t^{n-1} \lambda^{n-1}}{(1+t\lambda)^{n+1}}.$$

For $n = 0$, we simply compute

$$p_0(t) = P(0, t) = \frac{-\lambda t - 0}{-\lambda t - 1} = \frac{\lambda t}{\lambda t + 1}.$$

1.4 Part d

If we first consider $a = 1$, and let $X(t)$ denote the number of individuals at time t , using Mathematica we compute

$$EX = \frac{\partial P(1, t)}{\partial s} = 1.$$

Additionally,

$$\frac{\partial^2 P(1, t)}{\partial s^2} = E[X(t)(X(t) - 1)] = E[X(t)^2] - E[X(t)].$$

With Mathematica, we find that this is $2\lambda t$. Thus,

$$\text{Var } X(t) = E[X(t)^2] - E[X(t)]^2 = 2\lambda t + E[X(t)] - E[X(t)]^2 = 2\lambda t.$$

For general a , we simply consider the r.v. $X_1 + \dots + X_a$ to describe our model where $X_i = X(t)$ for each i . Then,

$$E[X_1 + \dots + X_a] = E[X_1] + \dots + E[X_a] = aE[X(t)] = a$$

and

$$\text{Var}[X_1 + \dots + X_a] = \text{Var}[X_1] + \dots + \text{Var}[X_a] = a \text{Var}[X(t)] = 2a\lambda t.$$

1.5 Part e

To derive the extinction probability by time t , we simply note that we can again consider our a starting individuals as a i.i.d. simulations with 1 starting individual, so the extinction probability is the extinction probability with 1 starting individual multiplied by itself a times. Thus, we can just use our result from part c to get

$$p_0(t) = P(0, t) = \left(\frac{-\lambda t}{-\lambda t - 1} \right)^a = \left(\frac{\lambda t}{\lambda t + 1} \right)^a.$$

1.6 Part f

The extinction probability of the process is given by

$$\lim_{t \rightarrow \infty} p_0(t) = 1.$$

2 Problem 2

Consider a general birth-death process with birth rate λ and death rate μ .

2.1 Part a

To derive the probability distribution for T , the time until the next event in this process, we define $q(t)$ to be the probability that no events occur up to time t . Note that the probability that no event occurs in a small time interval Δt is given by

$$1 - (\lambda + \mu)n\Delta t + o(\Delta t)$$

where n is the number of individuals at the start of the time interval. Neglecting $o(\Delta t)$ terms, this gives that

$$q(t + \Delta t) = q(t)(1 - (\lambda + \mu)n\Delta t).$$

We can then find that

$$\frac{q(t + \Delta t) - q(t)}{\Delta t} = -(\lambda + \mu)nq(t).$$

Taking the limit as $\Delta t \rightarrow 0$ of both sides,

$$\frac{d}{dt}q(t) = -(\lambda + \mu)nq(t).$$

This gives that $q(t) = Ce^{-(\lambda + \mu)nt}$, so imposing the obvious boundary condition $q(0) = 1$ gives that $C = 1$ and

$$q(t) = e^{-(\lambda + \mu)t}.$$

Now, we can use this to find that the CDF of T is given by

$$F_T(t) = \text{Prob}(T \leq t) = 1 - e^{-(\lambda + \mu)nt}$$

for $t \geq 0$. Then, we find that PDF by differentiating and adopting the convention that the PDF is zero for $t < 0$.

$$f_t(t) = \begin{cases} (\lambda + \mu)ne^{-(\lambda + \mu)nt}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

which is the PDF of an exponential distribution with parameter $(\lambda + \mu)n$.

2.2 Part b

Using Julia, we write a simulation for a single run of the birth-death process that starts with $X(0) = a$ cells and collects the number of individuals in the population from time 0 to t . See Appendix B for the Julia code utilized in this problem. We do this by sampling from the distribution we found in part a to determine the time until the next event and noting that if an event occurs, it is a birth with probability $\frac{\lambda}{\lambda + \mu}$ and a death with probability $\frac{\mu}{\lambda + \mu}$. Thus, we determine which event occurs by sampling from a Bernoulli distribution with parameter

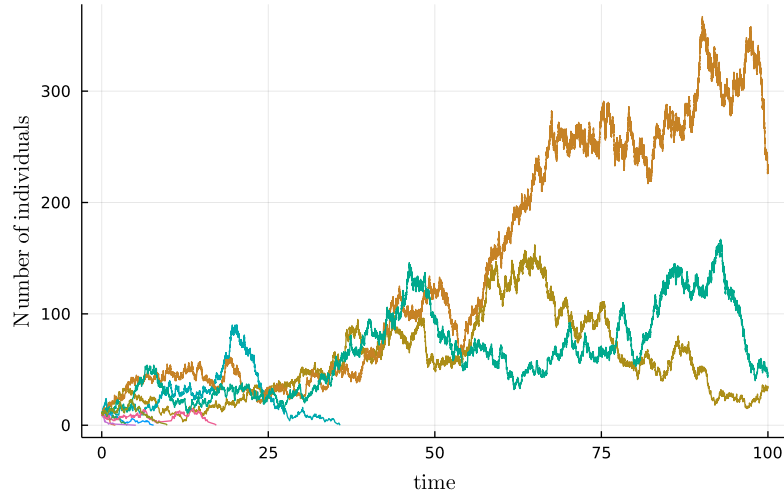
$$\frac{\lambda}{\lambda + \mu}$$

and scaling so that a success corresponds to adding one individual and a failure corresponds to subtracting one individual.

2.3 Part c

Setting $\lambda = \mu = 1$, $a = 10$ and $t = 100$, we plot 10 different runs and observe the following plot.

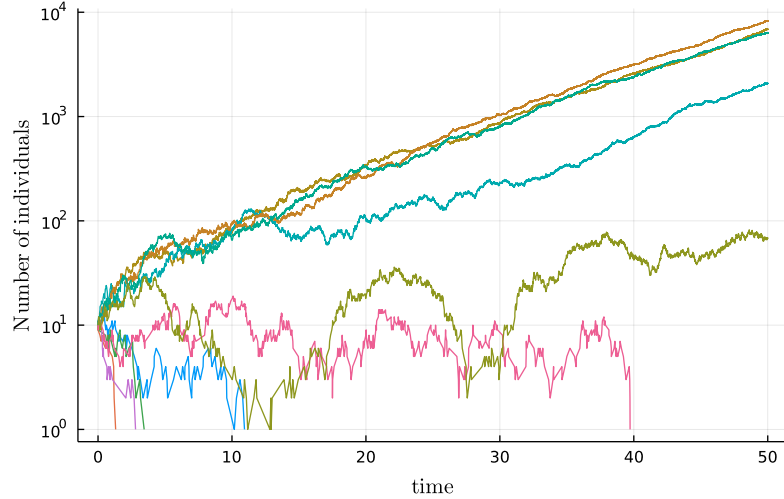
Birth – Death Process with $\lambda = 1$, $\mu = 1$, $a = 10$, $t = 100$



2.4 Part d

Now, we set $\lambda = 1.1$, $\mu = 1$, $a = 10$ and $t = 50$ and 10 different runs on a log scale. Note that we have to remove the data points at which we have zero individuals in order to taking the log of zero. The following plot is produced.

Birth – Death Process with $\lambda = 1.1$, $\mu = 1$, $a = 10$, $t = 50$

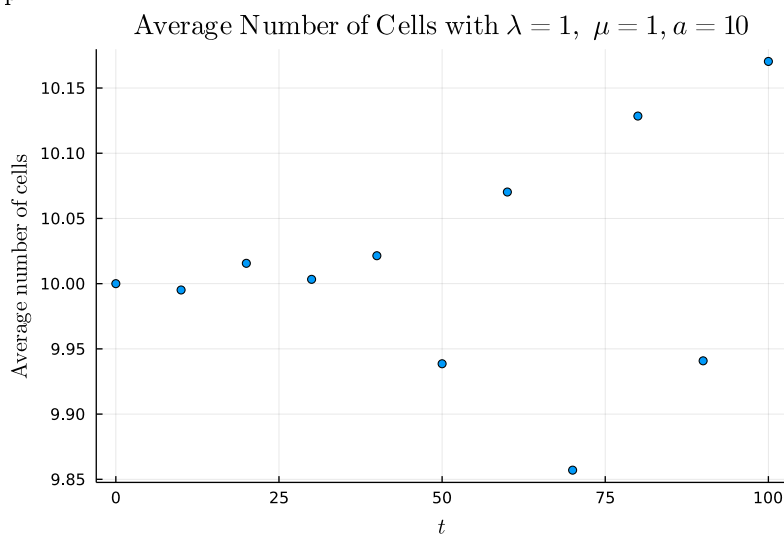


2.5 Part e

In Julia, we write a function `onerun(t)` which calls our simulation from before and returns the number of individuals present at time t . We also write a function `simulation(t, numruns)` which calls `onerun(t)` `numruns` times to compute the average number of individuals present at time t .

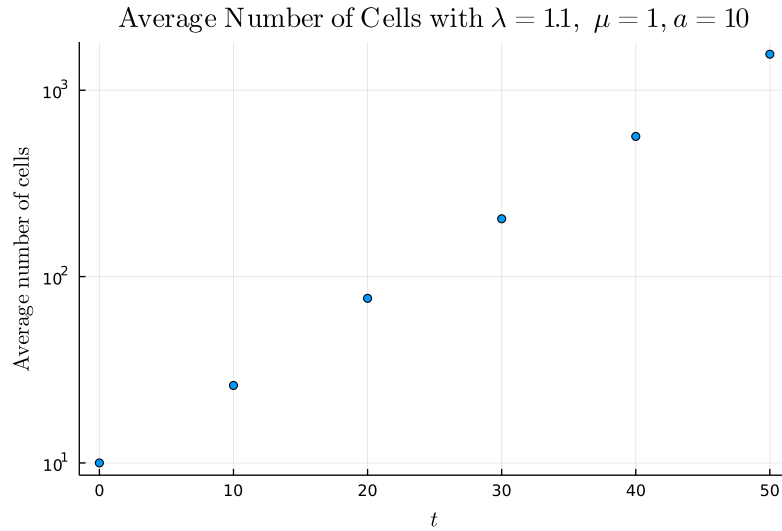
2.6 Part f

We run the function `simulation` from part e first for $\lambda = \mu = 1$, $a = 10$, $t = 0, 10, 20, \dots, 100$. In doing this, we use 50000 runs and obtain the following plot.



While this may look highly random, note that the scale of the y-axis is quite small relative to its values and that the result at each time is quite close to the population mean of 10 that we computed in problem 1.

Now, we run the function `simulation` from part e for $\lambda = 1.1$, $\mu = 1$, $a = 10$, $t = 0, 10, 20, \dots, 50$. Using only 500 runs this time, we obtain the following plot on a log scale.



The points here clearly appear linear which implies that the average grows exponentially in t .

3 Appendix A

The following is the Mathematica code used in problem 1. To compute the solution to the PDE in part b, we use

```
FullSimplify[
DSolve[{D[u[s, t], t] == \[Lambda] (s - 1)^2 D[u[s, t], s],
u[s, 0] == s^a}, u, {s, t}]
```

and

```
FullSimplify[(-1 + \[Lambda] (t +
1/(\[Lambda] - s \[Lambda]))) / (\[Lambda] (t +
1/(\[Lambda] - s \[Lambda])))]
```

to get the form presented. For part c, we use the following commands

```
D[(-s + (-1 + s) t \[Lambda]) / (-1 + (-1 + s) t \[Lambda]), {s, n}]
nderpb[s_,
n_] := -(((t \[Lambda]) / (1 + t (\[Lambda] - s \[Lambda])))^n n!) / (
t \[Lambda] (-1 + (-1 + s) t \[Lambda]))
FullSimplify[nderpb[0, n]]
```

to get the necessary derivatives. Finally, for part d, we use the commands

```
FullSimplify[nderpb[1, 1]]
```

```
FullSimplify[nderpb[1, 2]]
```

4 Appendix B

We use the following Julia code for problem 2.

```
using Random, Plots, Distributions, LaTeXStrings, Printf

#part b
function simul(a, tmax,  $\lambda$ ,  $\mu$ ; seed=[])
    if isempty(seed) == 0
        Random.seed!(seed) #set seed
    end

    t, n = 0., a
    values = [t a]
    while t < tmax
        #time until next event
        dist1 = Exponential(1/(n*( $\lambda$ + $\mu$ ))) #Reciprocal of input sets exponential parameter
        T = rand(dist1)
        t += T

        #determine if event is birth or death
        dist2 = Bernoulli( $\lambda$ /( $\lambda$ + $\mu$ ))
        result = 2*rand(dist2)-1 #rescale to take values -1 and 1
        n += result
        values = [values; [t n]]

        #safegaurd to prevent dividing by 0
        if n == 0
            return values
        end
    end
    values
end

#part c
 $\lambda$ ,  $\mu$ , a, t = 1., 1., 10, 100.
p1 = plot(title=latexstring(@sprintf(
    "\mathrm{Birth-Death-Process-with}~\lambda = %1.0f,~\mu=%1.0f,a=%i,t=%i"
    , $\lambda$ ,  $\mu$ , a, t)),legend=false)
xlabel!(latexstring(@sprintf("\mathrm{time}")))
ylabel!(latexstring(@sprintf("\mathrm{Number-of-individuals}")))
for i=1:10
    results=simul(a, t,  $\lambda$ ,  $\mu$ ; seed=i)
    plot!(results[:,1],results[:,2])
end
savefig(p1, "pc.pdf")
display(p1)

#part d
 $\lambda$ ,  $\mu$ , a, t = 1.1, 1., 10, 50.
p2 = plot(title=latexstring(@sprintf(
    "\mathrm{Birth-Death-Process-with}~\lambda = %1.1f,~\mu=%1.0f,a=%i,t=%i"
    , $\lambda$ ,  $\mu$ , a, t)),legend=false, yaxis=:log)
xlabel!(latexstring(@sprintf("\mathrm{time}")))
ylabel!(latexstring(@sprintf("\mathrm{Number-of-individuals}")))
for i=1:10
    results=simul(a, t,  $\lambda$ ,  $\mu$ ; seed=i)
    # remove zero value from log-scale plot
    if results[end,2]==0
```



```

        results = results[1:(end-1),:]
    end
    plot!(results[:,1],results[:,2])
end
savefig(p2, "pd.pdf")
display(p2)

#part e
function onerun(t, a,  $\lambda$ ,  $\mu$ ; seed=[])
    results = simul(a, t,  $\lambda$ ,  $\mu$ ; seed)
    index = maximum(findall(results[:,1].<=t))
    results[index,2]
end

function simulation(t,numruns, a,  $\lambda$ ,  $\mu$ ; seed=[])
    total = 0
    for i=1:numruns
        total += onerun(t, a,  $\lambda$ ,  $\mu$ ; seed)
    end
    total/numruns
end

#part f1
 $\lambda$ ,  $\mu$ , a = 1., 1., 10
tvec = 0:10:100
avgvec = zeros(size(tvec))
numruns = 50000
for (i,t) in enumerate(tvec)
    avgvec[i] = simulation(t,numruns, a,  $\lambda$ ,  $\mu$ )
end
p3 = plot(tvec,avgvec, seriestype = :scatter,legend=false,
    title=latexstring(@sprintf(
        "\mathrm{Average-Number-of-Cells-with}~\lambda = %1.0f,~\mu=%1.0f,a=%i"
        , $\lambda$ ,  $\mu$ , a)))
xlabel!(latexstring('t'))
ylabel!(latexstring(@sprintf("\mathrm{Average-number-of-cells}")))
savefig(p3, "pf1.pdf")
display(p3)

#part f2
 $\lambda$ ,  $\mu$ , a = 1.1, 1., 10
tvec = 0:10:50
avgvec = zeros(size(tvec))
numruns = 500
for (i,t) in enumerate(tvec)
    avgvec[i] = simulation(t,numruns, a,  $\lambda$ ,  $\mu$ )
end
p4 = plot(tvec,avgvec, seriestype = :scatter,legend=false, yaxis=:log,
    title=latexstring(@sprintf(
        "\mathrm{Average-Number-of-Cells-with}~\lambda = %1.1f,~\mu=%1.0f,a=%i"
        , $\lambda$ ,  $\mu$ , a)))
xlabel!(latexstring('t'))
ylabel!(latexstring(@sprintf("\mathrm{Average-number-of-cells}")))
savefig(p4, "pf2.pdf")
display(p4)

```