Course: Epidemic Models

Professor: Demiris

Assignment: Exercise 1: Calculate Final Size Probabilities of standard SIR epidemic

Student: Andre Ehrlich

Date: 2023.04.05

Demiris Instruction

“This will be an extended version of exercise 2.1 from the Andersson & Britton book for λ=0.5, 1, 2 and 5 and for n=10, 50, 100 and 200.”

Exercise 2.1 (Andresson & Britton)

Compute P_0^n, P_1^n, and P_2^n numerically using the recursive formula given by (2.4) assuming n = 10, m = 1, and λ = 2 and that the infectious period I is:

A) Exponentially distributed (Markovian case) with mean 1 time unit

B) Γ(2,2)-distributed (i.e. with mean 1)

C) Constant and equal to 1

Equation 2.4 from Andersson & Britton

$$\sum_{k=0}^{l} \binom{n-k}{l-k} P_k^n / [\varphi(\lambda(n - l)/n]^{k+m} = \binom{n}{l},\ 0 \le l \le n$$

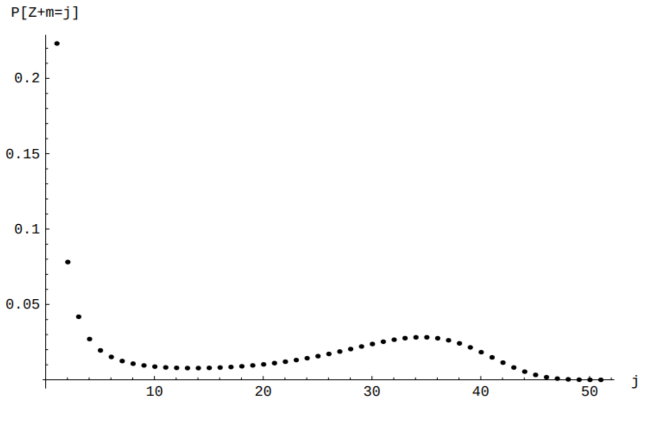


Figure 2.2: The exact distribution of $Z + m$ for $m = 1$, $n = 50$, $\lambda = 1.5$ and $I \equiv 1$, i.e. the infectious period is constant and equal to 1.

Φ(t) function: Laplace Transform of I r.v.

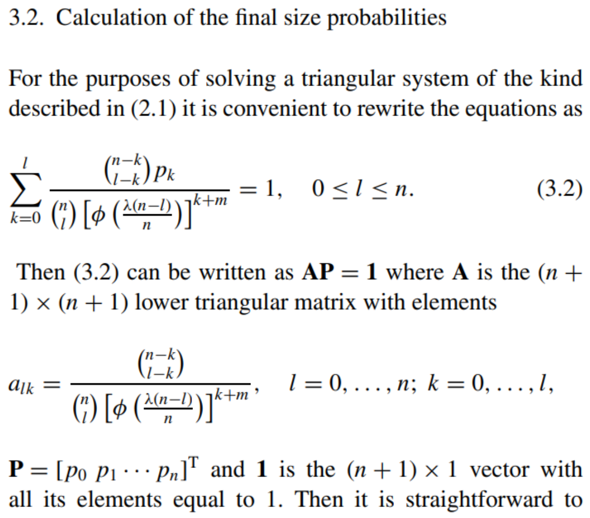
The Laplace transform of I is closely related to the moment generating function, but it is with the flipped sign of the variable t. Calculations of the laplace transform for the given distributions can be found in appendix.

Scenario	Distribution	Φ(t) - Laplace Transform	Note
a	Exp(1) (mean = 1)	$\phi(t) \;=\; \frac{1}{1+t}$	
b	Γ(2,2) (mean = 1)	$\phi(t) \;=\; (\frac{2}{2+t})^2$	Follows general form of a
c	constant =1	$\phi(t) \;=\; e^{-t}$	Degenerate r.v. final size equiv. Reed-Frost

Table 1: The laplace transforms for each distribution

Solving for Final Size Probabilities

The A matrix is the infectious pressure, a key mathematical construct that is dependent on the distribution of the infectious period. Build the A matrix with equation from Demiris 2006 section 3.2. And we can solve for the final size probabilities easily with the triangular system of equations. We implemented Forward Substitution with a few lines of R code (see appendix for code).



3.2. Calculation of the final size probabilities

For the purposes of solving a triangular system of the kind described in (2.1) it is convenient to rewrite the equations as

$$\sum_{k=0}^{l} \frac{\binom{n-k}{l-k} p_k}{\binom{n}{l}\left[\phi\left(\frac{\lambda(n-l)}{n}\right)\right]^{k+m}} = 1, \quad 0 \le l \le n. \qquad (3.2)$$

Then (3.2) can be written as $\mathbf{AP} = \mathbf{1}$ where $\mathbf{A}$ is the $(n+1) \times (n+1)$ lower triangular matrix with elements

$$a_{lk} = \frac{\binom{n-k}{l-k}}{\binom{n}{l}\left[\phi\left(\frac{\lambda(n-l)}{n}\right)\right]^{k+m}}, \quad l = 0, \ldots, n;\; k = 0, \ldots, l,$$

$\mathbf{P} = [p_0\ p_1 \cdots p_n]^{\mathrm{T}}$ and $\mathbf{1}$ is the $(n+1) \times 1$ vector with all its elements equal to 1. Then it is straightforward to

Numerical Instability

The graphs below show that there is numerical instability when doing calculations even on the scale of lambda 0.05 - 5 and n = 10 - 200. Nonsensical numbers for probabilities are returned from certain configurations, typically when n is > 100, or the lambda parameter is approaching 0. The non-corrupted results reveal the expected “bathtub” bimodal plot that has a mode at small values of k, where a disease transmission dies out without reaching many people, followed by a long range of miniscule probabilities before reaching the second mode that would be considered pandemic outbreak.

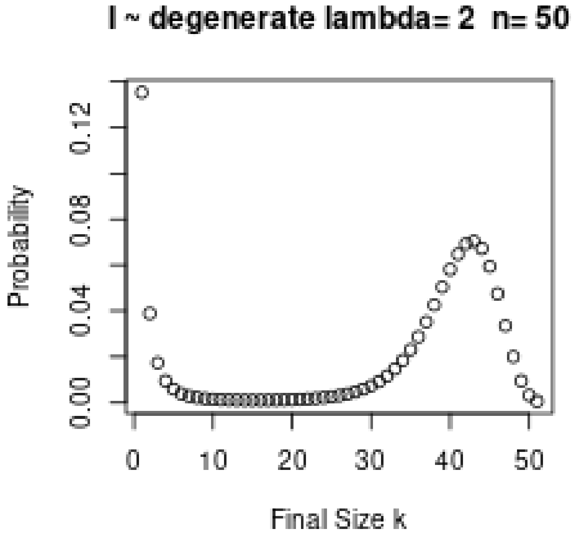


Figure 1(a): Typical Bathtub-shaped GSE final-size distribution

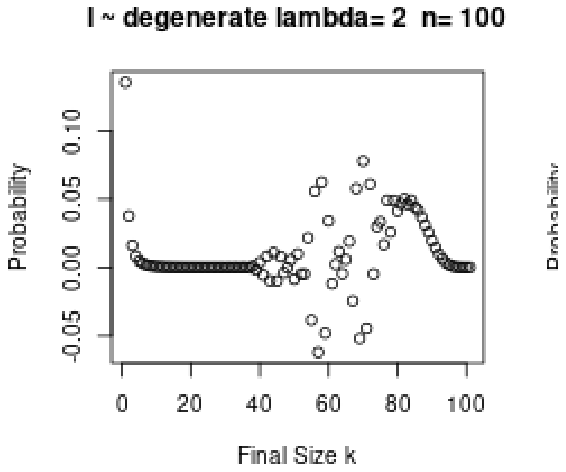


Figure 1(b): Corrupted result due to numerical instability

But how to handle the computational difficulties like negative probabilities and rounding errors? R Package Rmpfr allows us to specify arbitrary length of precision to numbers when making calculations, allowing for more precise results than R's default 17-digit representation. With 40 precisionbits, running the range of values of the assignment takes several minutes. Unfortunately, numerical instability is still present. **More time is needed to find the requisite precision and r code implementation to yield precise, correct final size probabilities.**

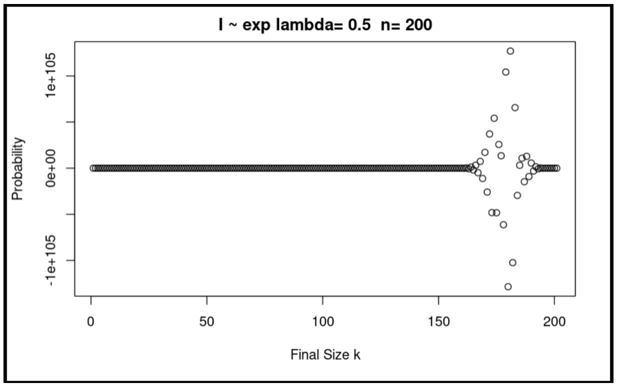


Figure 2(a): Rmpfr computed final size probabilities with PrecBit = 40

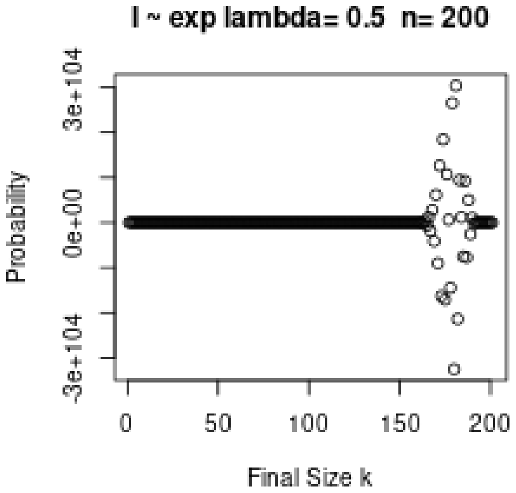


Figure 2(a): Rmpfr computed final size probabilities with PrecBit = 400

I ~ Exp(1) Final Size Probabilities

The First three final size probabilities for the given parameters

λ	n	P(k=0)	P(k=1)	P(k=2)
0.5	10	0.666666666666667	0.1585414189457	0.0740996245619005
0.5	50	0.666666666666667	0.150143386934523	0.0674021254311841
0.5	100	0.666666666666667	0.149140762780435	0.0666173095669721
0.5	200	0.666666666666667	0.148643212625768	0.0662290841329194
1	10	0.5	0.138504155124654	0.0790841626483373
1	50	0.5	0.127538006325885	0.0654024032856085
1	100	0.5	0.12625943789298	0.0639284216452738
1	200	0.5	0.125627351586985	0.0632086305997762
2	10	0.333333333333333	0.0850340136054423	0.0470260930954078
2	50	0.333333333333333	0.0760896031166306	0.0352216876928904
2	100	0.333333333333333	0.0750716934672617	0.0340449511106494
2	200	0.333333333333333	0.0745703813902173	0.0334768803126956
5	10	0.166666666666667	0.0275482093663912	0.0104132231404958
5	50	0.166666666666667	0.0239394809920518	0.00703416727456479
5	100	0.166666666666667	0.0235388272956237	0.0067228332112541
5	200	0.166666666666667	0.0233422617484533	0.00657420905663573

Table 2: Calculated Final Size Probabilities using of Exp (R default precision)

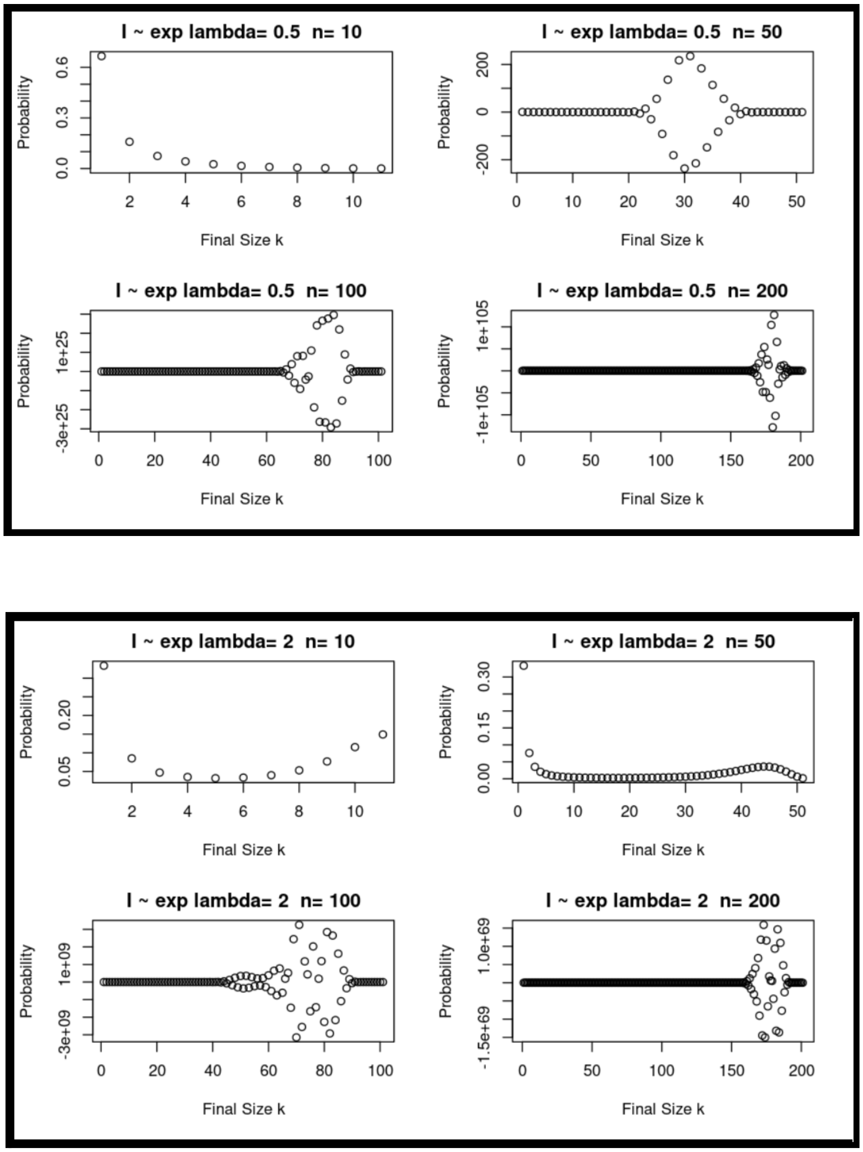


Figure 3: Selected plots of calculated final size distributions, showing numerical instability

I ~ Gamma(2,2) Final Size Probabilities

λ	n	P(k=0)	P(k=1)	P(k=2)
0.5	10	0.64	0.175853424949099	0.0824361557965245
0.5	50	0.64	0.166154889861461	0.0751377362825645
0.5	100	0.64	0.16499214260652	0.0742645945964782
0.5	200	0.64	0.164414753181673	0.0738313472120276
1	10	0.444444444444444	0.148298750975764	0.0893361376137326
1	50	0.444444444444444	0.134807359108215	0.0719735667399046
1	100	0.444444444444444	0.133235373065145	0.0700989226899341
1	200	0.444444444444444	0.132458353934926	0.0691834797883618
2	10	0.25	0.0748152638484971	0.043560032299451
2	50	0.25	0.0647384533691749	0.0299018438933388
2	100	0.25	0.0636063681716361	0.0285880100388588
2	200	0.25	0.0630500147097087	0.0279574603983232
5	10	0.0816326530612245	0.0123473639306919	0.00402955720813403
5	50	0.0816326530612245	0.0100118041241411	0.00222827046049357
5	100	0.0816326530612245	0.00976177731219632	0.00207961809557118
5	200	0.0816326530612245	0.00963981891559658	0.00200981074878791

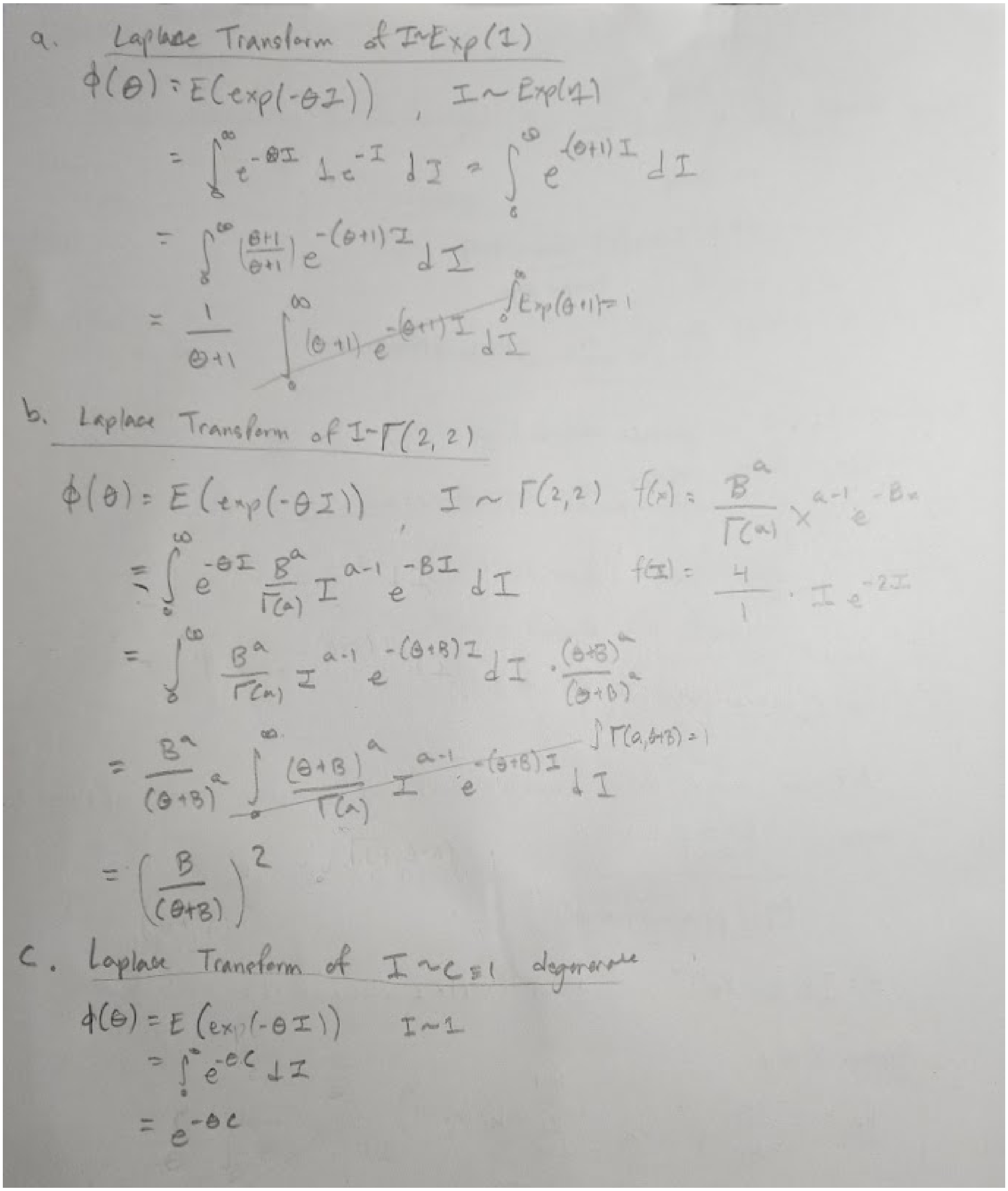
Table 3: Calculated Final Size Probabilities using of Gamma (R default precision)

I ~ 1 Final Size Probabilities

λ	n	P(k=0)	P(k=1)	P(k=2)
0.5	10	0.606530659712633	0.19828636286098	0.0935709345876281
0.5	50	0.606530659712633	0.186720391467694	0.0856334612088385
0.5	100	0.606530659712633	0.185324647798671	0.0846518717047663
0.5	200	0.606530659712633	0.184630837563202	0.0841623804099641
1	10	0.367879441171442	0.157302689989515	0.103870985665967
1	50	0.367879441171442	0.13945918050761	0.0798190519638378
1	100	0.367879441171442	0.137381186536896	0.0772095258549547
1	200	0.367879441171442	0.136354255734942	0.0759351263104722
2	10	0.135335283236613	0.0495295059112696	0.0320947052383371
2	50	0.135335283236613	0.0388990226379325	0.0173389820734622
2	100	0.135335283236613	0.0377474953878865	0.0160580286988139
2	200	0.135335283236613	0.0371850435827047	0.0154538010613523
5	10	0.00673794699908547	0.00048557974198979	9.47314037318947e-05
5	50	0.00673794699908547	0.000263845868800083	1.73733690393225e-05
5	100	0.00673794699908547	0.000244704811937336	1.41110115293794e-05
5	200	0.00673794699908547	0.00023567989611584	1.27215066934712e-05

Table 4: Calculated Final Size Probabilities using of Cons I (R default precision)

Laplace Transformation



a. Laplace Transform of $I \sim Exp(1)$

$$\phi(\theta) = E(\exp(-\theta I)) \quad , \quad I \sim Exp(1)$$

$$= \int_0^\infty e^{-\theta I} \, 1 e^{-I} \, dI = \int_0^\infty e^{-(\theta+1) I} \, dI$$

$$= \int_0^\infty \left(\frac{\theta+1}{\theta+1}\right) e^{-(\theta+1) I} \, dI$$

$$= \frac{1}{\theta+1} \cancel{\int_0^\infty (\theta+1) e^{-(\theta+1) I} \, dI} \quad \int_0^\infty Exp(\theta+1) = 1$$

b. Laplace Transform of $I \sim \Gamma(2, 2)$

$$\phi(\theta) = E(\exp(-\theta I)) \quad , \quad I \sim \Gamma(2,2) \quad f(x) = \frac{B^a}{\Gamma(a)} x^{a-1} e^{-Bx}$$

$$= \int_0^\infty e^{-\theta I} \frac{B^a}{\Gamma(a)} I^{a-1} e^{-BI} \, dI \qquad f(I) = \frac{4}{1} \cdot I e^{-2I}$$

$$= \int_0^\infty \frac{B^a}{\Gamma(a)} I^{a-1} e^{-(\theta+B) I} \, dI \cdot \frac{(\theta+B)^a}{(\theta+B)^a}$$

$$= \frac{B^a}{(\theta+B)^a} \cancel{\int_0^\infty \frac{(\theta+B)^a}{\Gamma(a)} I^{a-1} e^{-(\theta+B) I} \, dI} \quad \int \Gamma(a, \theta+B) = 1$$

$$= \left(\frac{B}{(\theta+B)}\right)^2$$

c. Laplace Transform of $I \sim c \equiv 1$ degenerate

$$\phi(\theta) = E(\exp(-\theta I)) \qquad I \sim 1$$

$$= \int_0^\infty e^{-\theta c} \, dI$$

$$= e^{-\theta c}$$

SIR Notes

- **Definitions**
  - m - infectious individuals (aka infectives)
  - n - susceptible individuals
  - I - infectious period random variable (i.i.d across infectives)
  - Z - final number of infected [0,n]
  - R_0- Basic reproduction number
  - P_k^n - Probability that k of the initial susceptibles are ultimately infected.
- **Intensity:** Infective makes contact with a given individual at time points of a homogeneous Poisson process with intensity λ/N.
  - Set s.t. the rate at which a given infective makes contact with other (susceptible) individuals is constant (= λ) , independently of population size.
  - If a contacted individual is susceptible, they immediately become infectious and can infect others.
- **Reed-Frost:** When the infectious period is constant, $I \equiv c,$ the distribution of the final size equivalent to the Reed-Frost model defined for discrete time dynamics.
- **Infection pressure:** Mathematical construct, useful. Each susceptible individual is associated with 'exposure to infection', beyond a certain threshold will become infected.

$$A(t) = \lambda/n \int_0^t Y(u)\, du$$

$A(t) - \ pressure\ on\ given\ susceptible\ at\ time\ t$

$Y(t) - \ number\ of\ infectives\ at\ time\ t$

$Q_i \quad - r.v.\ iid\ ,\ individual\ thresholds,\ exponential$

$I_i \quad - r.v.\ iid\ ,\ individual\ infectious\ periods$

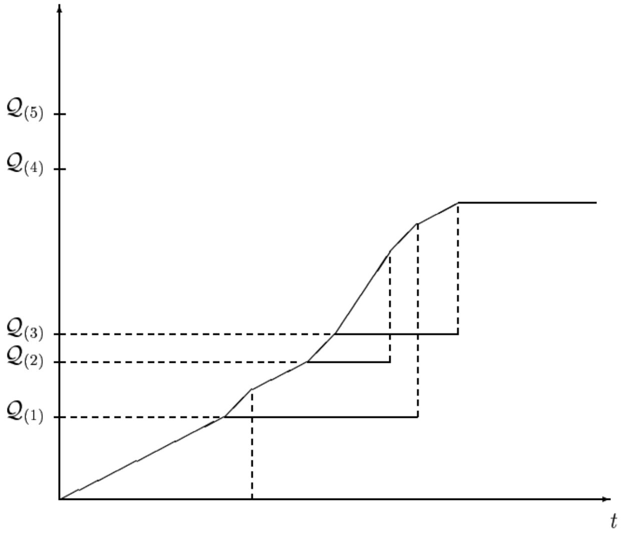


Figure 2.1: A typical realisation of the total infection pressure with $m = 1$ initially infectious individual. Note that the infection pressure never reaches $Q_{(4)}$ so the epidemic stops and the final size is $Z = 3$.

Plot: Infectious Pressure A(t), m=1.

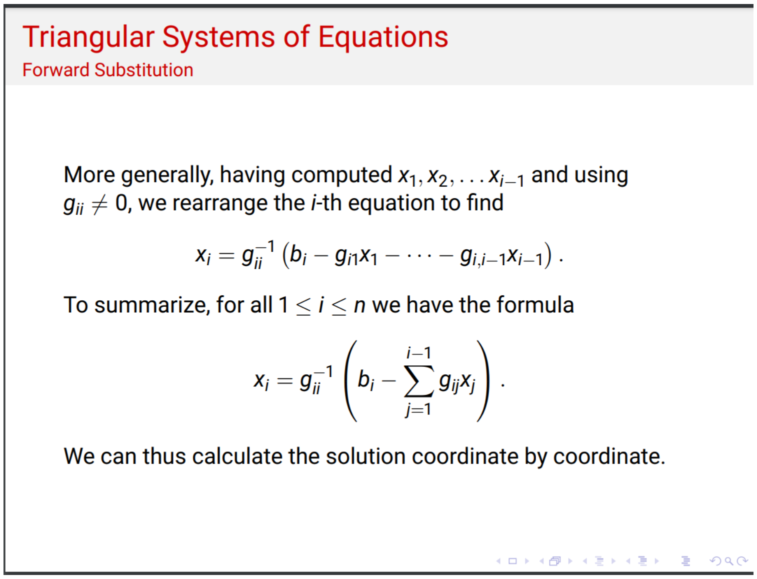
Note: slope proportional to number of infectious periods covering time point t, i.e. Y(t)

- **Sub-Epidemics**: Now fix k and choose l such that $0 \leq k \leq l D \leq n$, implying that $K \subseteq L \subseteq N$. We use the notion of infection pressure to compare an epidemie within N with a sub-epidemie within L. The event that an epidemie within N infects precisely the set K is the same as the event that a sub-epidemic within L infects precisely K, and that these k new infectives, together with the m, initial infectives, fail to infect any of the individuals in the set N \ L
- **Selke (1983) Construction**: probability of avoiding infection = $exp(-a)$ given that sub-epidemic has generated infection pressure $A^l = a$

  => $P^n_K = P^l_K E[exp(- A^l(n - l)|Z^l = k]$
- **Markovian Case:**

- **Triangular system of equations** solved recursively

$$
\begin{aligned}
&g_{11}x_1 &&= b_1,\\
&g_{21}x_1 + g_{22}x_2 &&= b_2,\\
&g_{31}x_1 + g_{32}x_2 + g_{33}x_3 &&= b_3,\\
&\vdots && \vdots\\
&g_{n1}x_1 + g_{n2}x_2 + \cdots + g_{nn}x_n &&= b_n.
\end{aligned}
$$



Triangular Systems of Equations

Forward Substitution

More generally, having computed $x_1, x_2, \dots x_{i-1}$ and using $g_{ii} \neq 0$, we rearrange the i-th equation to find

$$x_i = g_{ii}^{-1}\left(b_i - g_{i1}x_1 - \cdots - g_{i,i-1}x_{i-1}\right).$$

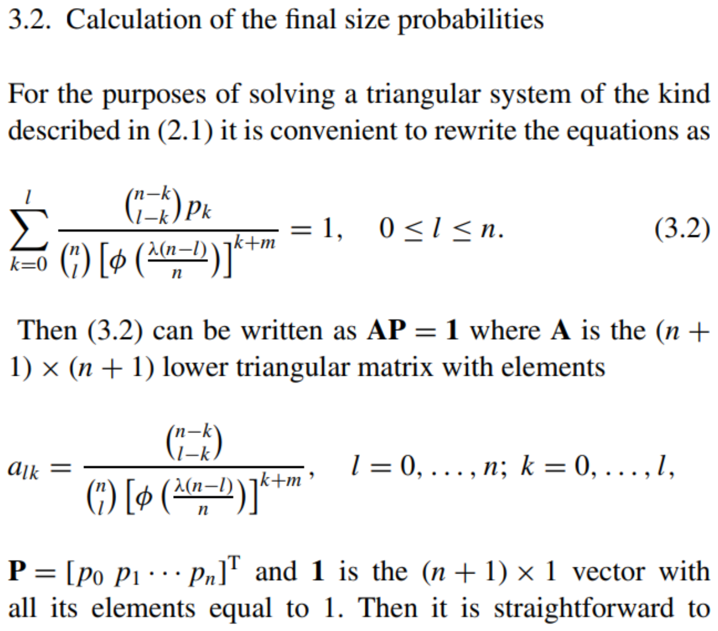
To summarize, for all $1 \leq i \leq n$ we have the formula

$$x_i = g_{ii}^{-1}\left(b_i - \sum_{j=1}^{i-1} g_{ij}x_j\right).$$

We can thus calculate the solution coordinate by coordinate.

Computation of final outcome probabilities for the generalised stochastic epidemic | Demiris, O'Neil

- Used Fixed-Point Precision Arithmetic implementations that allow the program to hold more memory for each number, at the expense of computational time, they can avoid problematic numerical issues that result in unfeasible results like negative probabilities.
- *"The bimodal nature of the final size distribution lies at the heart of the numerical instabilities encountered when solving (2.1). Specifically, calculation of the probabilities corresponding to the epidemic taking off requires calculation of the intermediate probabilities. The latter are typically very small in value, and numerical problems arise when they reach the limits of machine accuracy."*
    - Negative probabilities
    - handled with fine precision software.
    - Special case with markovian assumption has nice exploit
- How to compute the final probabilities:



3.2. Calculation of the final size probabilities

For the purposes of solving a triangular system of the kind described in (2.1) it is convenient to rewrite the equations as

$$\sum_{k=0}^{l} \frac{\binom{n-k}{l-k} p_k}{\binom{n}{l}\left[\phi\left(\frac{\lambda(n-l)}{n}\right)\right]^{k+m}} = 1, \quad 0 \le l \le n. \tag{3.2}$$

Then (3.2) can be written as $\mathbf{AP} = \mathbf{1}$ where $\mathbf{A}$ is the $(n+1) \times (n+1)$ lower triangular matrix with elements

$$a_{lk} = \frac{\binom{n-k}{l-k}}{\binom{n}{l}\left[\phi\left(\frac{\lambda(n-l)}{n}\right)\right]^{k+m}}, \quad l = 0, \ldots, n; \; k = 0, \ldots, l,$$

$\mathbf{P} = [p_0 \; p_1 \cdots p_n]^{\mathrm{T}}$ and $\mathbf{1}$ is the $(n+1) \times 1$ vector with all its elements equal to 1. Then it is straightforward to

- MCMC & Gaussian Approximation are pretty good in practice.

Bibliography

- Epidemic Models, Anderson & Britton, 2000
- Calculate Exact Final Size Probabilities, Demiris, O'Neil, 2006
- The shape of the size distribution of an epidemic in a finite population, Frank Ball, Ingemar Nåsell, 1994
- Computation of epidemic final size distributions, Andrew J. Black, J. V. Ross, 2015
- Arbitrarily Accurate Computation with R: The Rmpfr Package, Martin Mächler

I cannot find a freely available pdf of this paper:

- A unified approach to the distribution of total size and total area under the trajectory of infectives in epidemic models, F G Ball, 1986

Code

Unset

~~~
---
title: "Epidemic GSE Exercise"
author: "Andre Ehrlich"
date: "`r Sys.Date()`"
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

library(Rmpfr)
```

## Standard SIR Epidemic: Probability of final size
Compute following probabilities:
$$P_0^n,P_1^n,P_2^n,$$

Given following recursive formula:
$$ \sum_{k=0}^{n}  \frac{{{n-k}\choose{l-k}}P_k^n }{
[\phi(\lambda(n-l)/n)]^{k+m} } = {n \choose l}, 0 \le l \le n  $$

```{r phi}
# phi(θ) = E[exp(-θ*I)]
# Phi: laplace transform of r.v. I, depends on distribution
# we are given several to try for this exercise.

# scenario a: I ~ Exp(1) (mean = 1)
phi_mgf_exp <- function(t, lambda=1){
  lambda/(lambda+t)
}

# scenario b: I ~ Γ(2,2) (mean = 1)
phi_mgf_gamma <- function(t, alpha=2, beta=2){
~~~

~~~
  (beta / (t + beta)) ^ alpha
}

# scenario c: I ~ Constant = 1   (Reed-Frost model)
phi_mgf_degenerate <- function(t,a=1)  exp(-a*t)


phi_func <- function(I_dist, t){
  if (I_dist == "exp"){
    phi_mgf_exp(t=t)
  } else if (I_dist == "gamma") {
    phi_mgf_gamma(t)
  } else if (I_dist == "degenerate") {
    phi_mgf_degenerate(t)
  }
}
```



```{r, calc}
# Solve triangular system of equations recursively
# AP = 1 # A is (n+1)x(n+1) # P is (n+1) * 1 # 1 is (n+1) * 1 of
1's
calculate_final_size_probabilities <- function(I, lambda, n, m){
  # Compute A
  A = matrix(rep(0, (n+1)*(n+1) ), nrow=n+1, ncol=n+1)
  for (l in seq(0,n)){
    for (k in seq(0,l)){
      A[l+1,k+1] = choose(n-k, l-k) / (
        choose(n,l) * phi_func(I_dist=I, t=lambda*(n-l)/n)^(k+m)
      )
    }
  }
  # Solve Triangular system of equations with forward
substitution.
  # AP = 1
~~~

```
  # vector of ones
  b = rep(1,n+1)

  # Final Size Probabilities
  # Forward Substitution solution to triangular system of
equations.
  x <- backsolve(A, b, upper.tri=FALSE)
  # x = matrix(0,nrow=n+1, ncol=1)
  # for (i in seq(1,n+1)){
  #   x[i] = b[i]
  #   # print(paste("i", i))
  #   if (i > 1){
  #     for (j in seq(1,i-1)){
  #       # print(paste("j",j))
  #       x[i] = x[i] - A[i,j]*x[j]
  #     }
  #   }
  #   x[i] =  x[i] / A[i,i]
  # }

  x
}


### Traverse Vectors of Parameters
results = data.frame()
m = 1
for (I in c("exp", "gamma", "degenerate")) {
  for (lambda in c(0.5,1,2,5)) {
    fname <-  file.path(getwd(),
paste0("I",I,"-lambda",lambda,".png"))
    print(fname)
    png(fname)
    par(mfrow=c(2,2))
    for ( n in c(10,50,100,200) ) {
      x = calculate_final_size_probabilities(I, lambda, n, m)
      cres = c(I, lambda, n, m, x[1],x[2],x[3])
      results = rbind(results, cres)
      plot(x, main=paste("I ~", I, "lambda=",lambda, " n=",n),
xlab="Final Size k", ylab="Probability")
    }
    dev.off()
  }
}

colnames(results) <- c("I", "lambda", "n","m", "P(k=0)",
"P(k=1)", "P(k=2)")
results
# png("result_table_exp.png")
subset(results, results$I == "degenerate") %>% gt() #%>%
gtsave(filename="result_table_exp.png")
# dev.off()
```

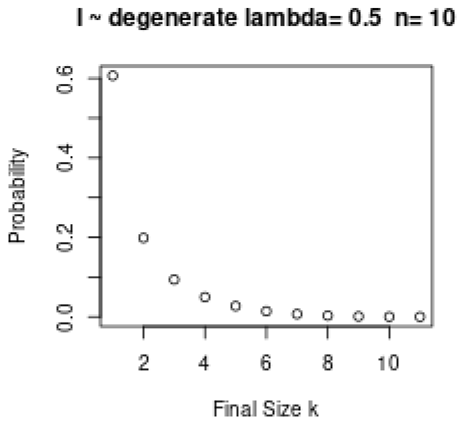
```{r, calc_rmpfr}
# Solve triangular system of equations recursively
# AP = 1 # A is (n+1)x(n+1) # P is (n+1) * 1 # 1 is (n+1) * 1 of
1's
calculate_final_size_probabilities_mpfr <- function(I, lambda, n,
m){
  # I = "exp"
  # lambda = 1
  # n = 10
  # m = 1
  # Compute A
  A = mpfrArray(0, precBits = 400, dim = c(n+1, n+1))
  A
  # A = matrix(rep(0, (n+1)*(n+1) ), nrow=n+1, ncol=n+1)
  for (l in seq(0,n)){
    for (k in seq(0,l)){
      A[l+1,k+1] = choose(n-k, l-k) / (
        choose(n,l) * phi_func(I_dist=I, t=lambda*(n-l)/n)^(k+m)
      )
    }
  }
  A
  # Solve Triangular system of equations with forward
substitution.
  # AP = 1
  # vector of ones
  b =  mpfrArray(1, precBits=400, dim = c(1, n+1))
  b
  # Final Size Probabilities
  # Forward Substitution solution to triangular system of
equations.
  # x <- backsolve(A, b, upper.tri=FALSE)
  # x = matrix(0,nrow=n+1, ncol=1)
  x = mpfrArray(0, precBits = 400, dim = c(n+1,1))
  x
  for (i in seq(1,n+1)){
    x[i] = b[i]
    # print(paste("i", i))
    if (i > 1){
      for (j in seq(1,i-1)){
        # print(paste("j",j))
        x[i] = x[i] - A[i,j]*x[j]
      }
    }
    x[i] =  x[i] / A[i,i]
  }

  x
}
```

```
### Traverse Vectors of Parameters
results = data.frame()
m = 1
for (I in c("exp")){ #}, "gamma", "degenerate")) {
  print(paste("I", I))
  for (lambda in c(0.5)){#},1,2,5)) {
    print(paste("lambda", lambda))


    fname <-  file.path(getwd(),
paste0("I",I,"-lambda",lambda,"_mpfr.png"))
    png(fname)
    par(mfrow=c(2,2))
    # for ( n in c(10,50,100,200) ) {
    for ( n in c(200) ) {

      print(paste("n", n))
      x = calculate_final_size_probabilities_mpfr(I, lambda, n,
m)
      print("HAHAHAHA")

      # cres = c(I, lambda, n, m, x[1,1],x[2,1],x[3,1])
      # cres
      # x[,1]

      # results = rbind(results, cres)
      plot(seq(1,length(x[,1])), x[,1], main=paste("I ~", I,
"lambda=",lambda, " n=",n), xlab="Final Size k",
ylab="Probability")
    }
    dev.off()
  }
}
```

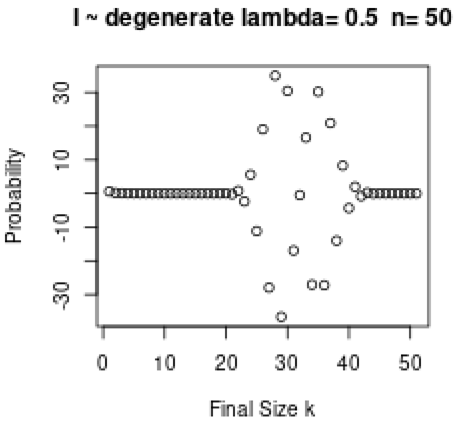
~~~
colnames(results) <- c("I", "lambda", "n","m", "P(k=0)",
"P(k=1)", "P(k=2)")
results
# png("result_table_exp.png")
subset(results, results$I == "exp") %>% gt() %>%
gtsave(filename="result_table_exp.png")
# dev.off()
```
~~~

Appendix

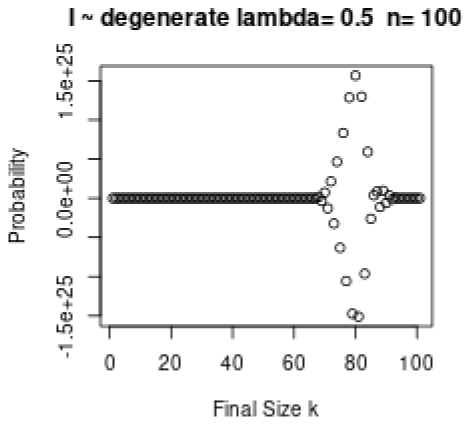
All plots of results calculated with default R numerical precision



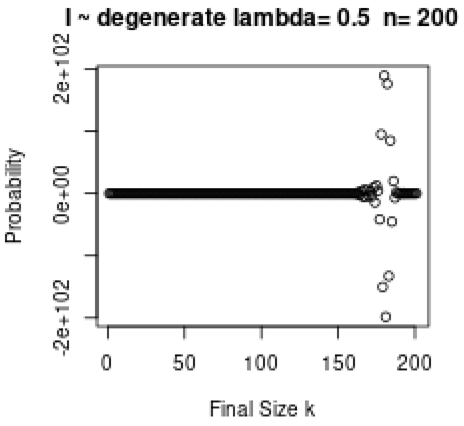
I ~ degenerate lambda= 0.5 n= 10



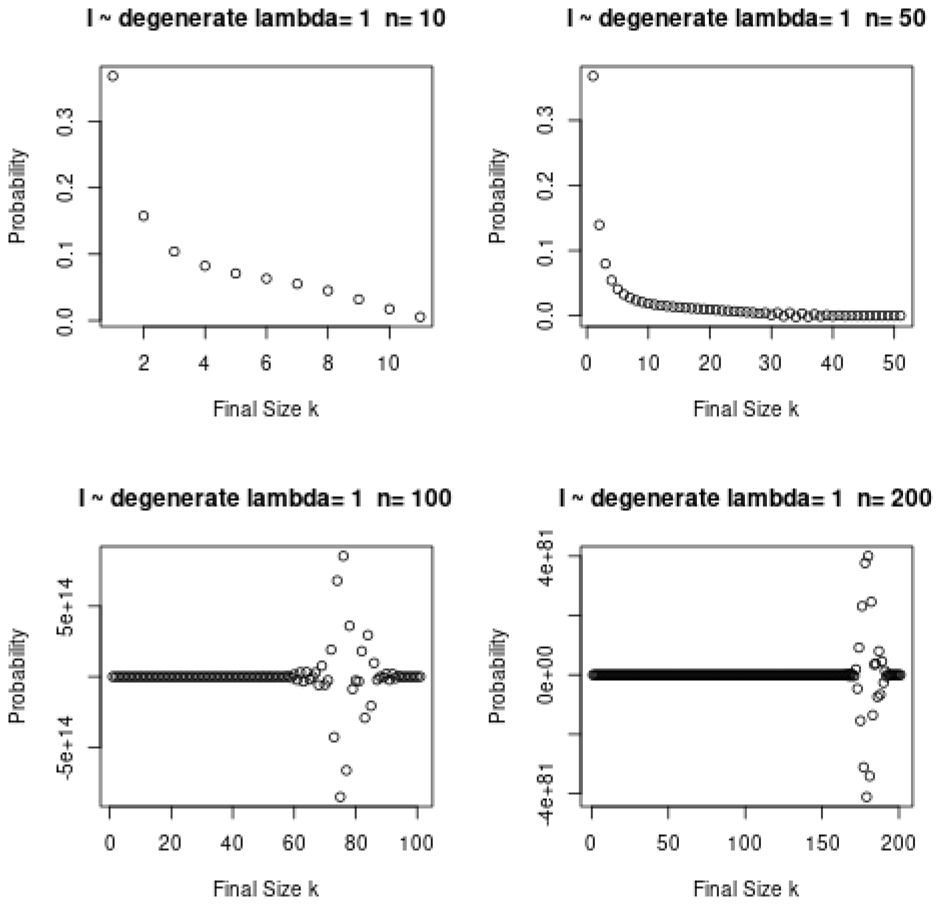
I ~ degenerate lambda= 0.5 n= 50

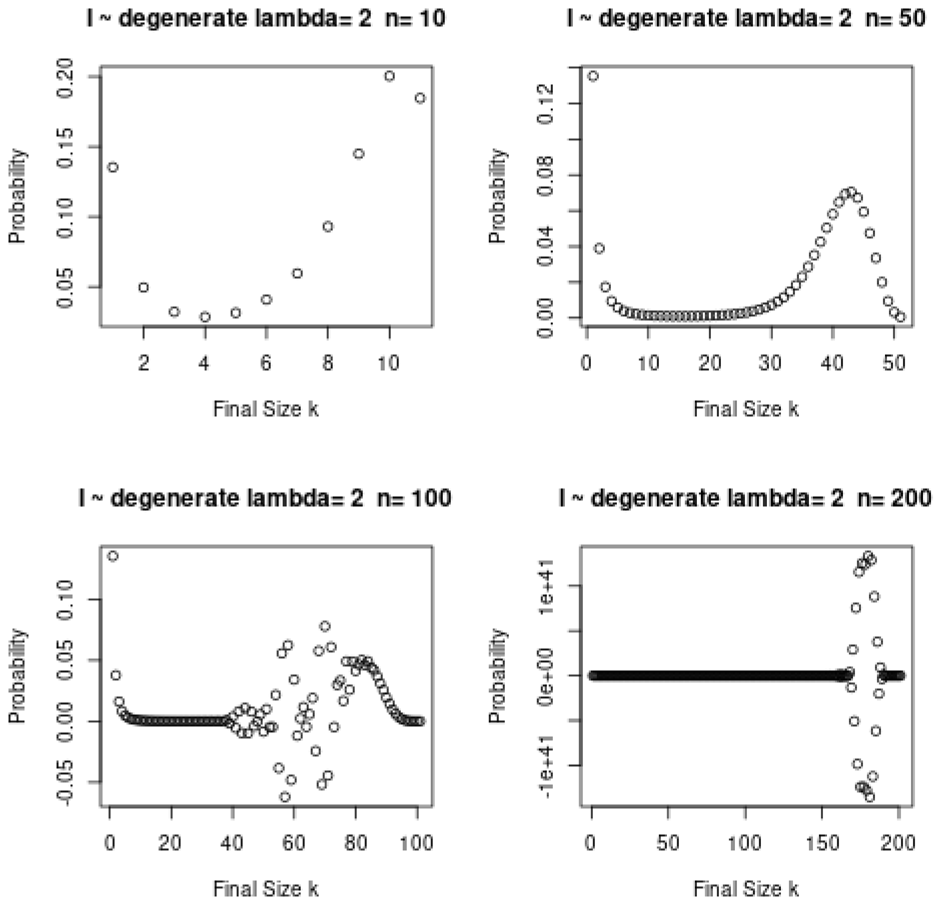


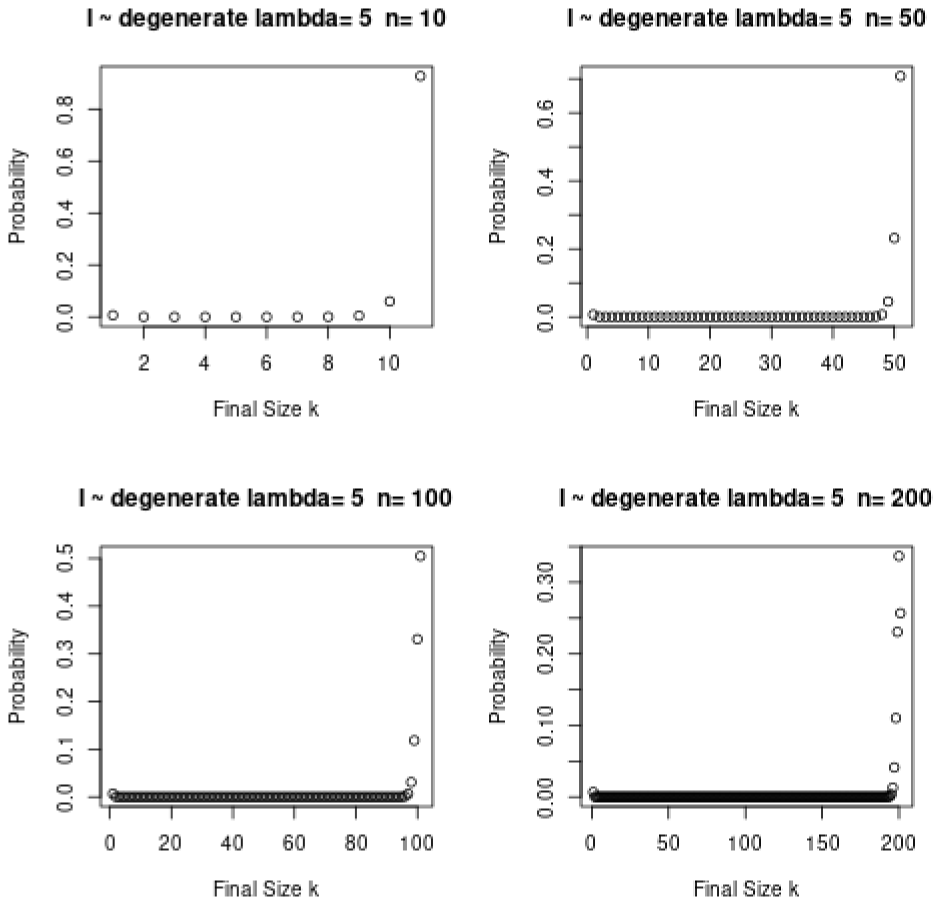
I ~ degenerate lambda= 0.5 n= 100



I ~ degenerate lambda= 0.5 n= 200


I ~ degenerate lambda= 1 n= 10
Probability
0.0
0.1
0.2
0.3
2
4
6
8
10
Final Size k
I ~ degenerate lambda= 1 n= 50
Probability
0.0
0.1
0.2
0.3
0
10
20
30
40
50
Final Size k
I ~ degenerate lambda= 1 n= 100
Probability
-5e+14
5e+14
0
20
40
60
80
100
Final Size k
I ~ degenerate lambda= 1 n= 200
Probability
-4e+81
0e+00
4e+81
0
50
100
150
200
Final Size k


I ~ degenerate lambda= 2 n= 10
Probability
0.05
0.10
0.15
0.20
2
4
6
8
10
Final Size k
I ~ degenerate lambda= 2 n= 50
Probability
0.00
0.04
0.08
0.12
0
10
20
30
40
50
Final Size k
I ~ degenerate lambda= 2 n= 100
Probability
-0.05
0.00
0.05
0.10
0
20
40
60
80
100
Final Size k
I ~ degenerate lambda= 2 n= 200
Probability
-1e+41
0e+00
1e+41
0
50
100
150
200
Final Size k

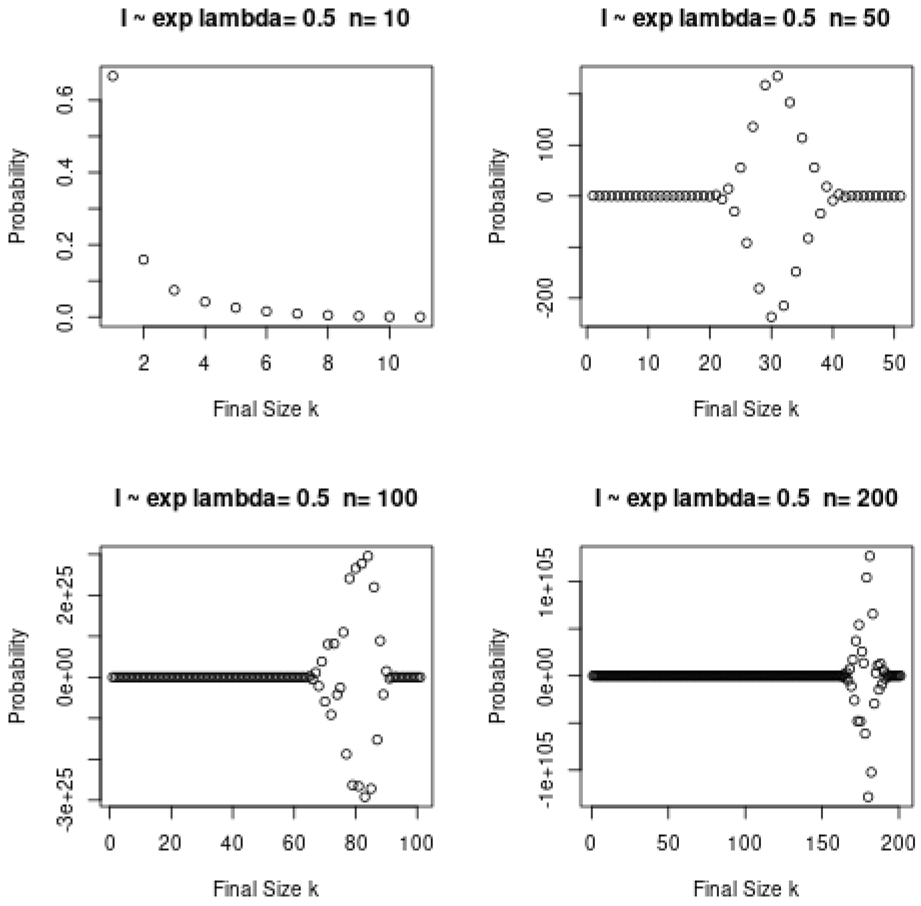


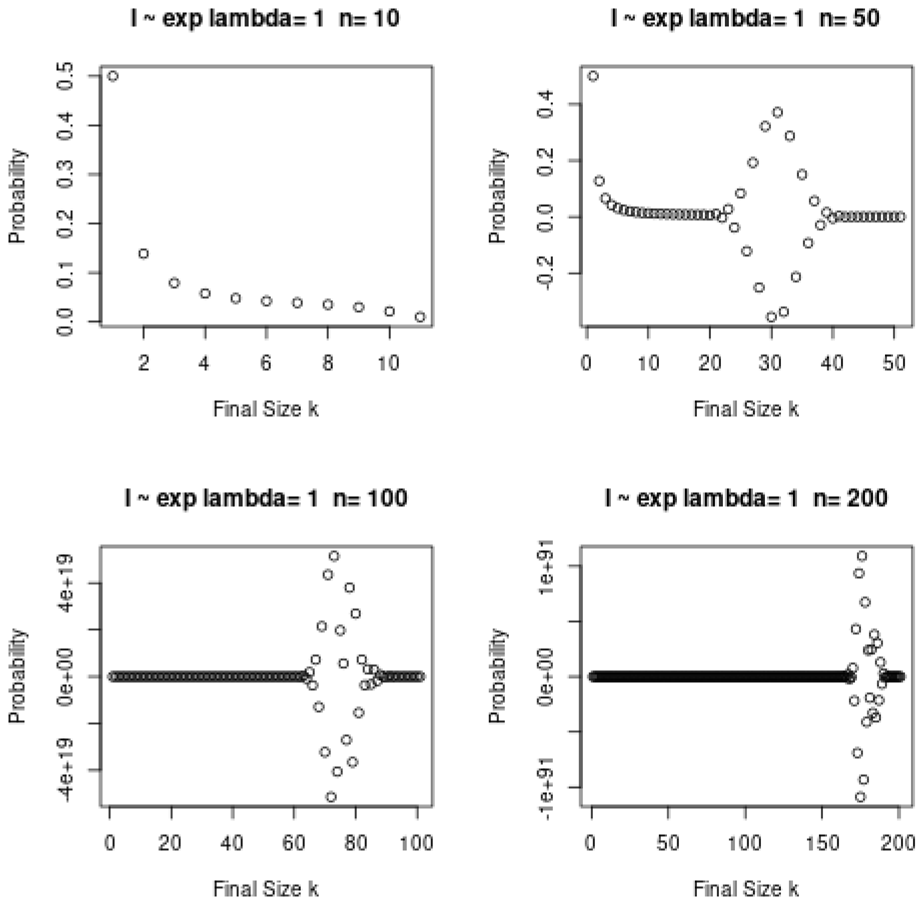
I ~ degenerate lambda= 5 n= 10

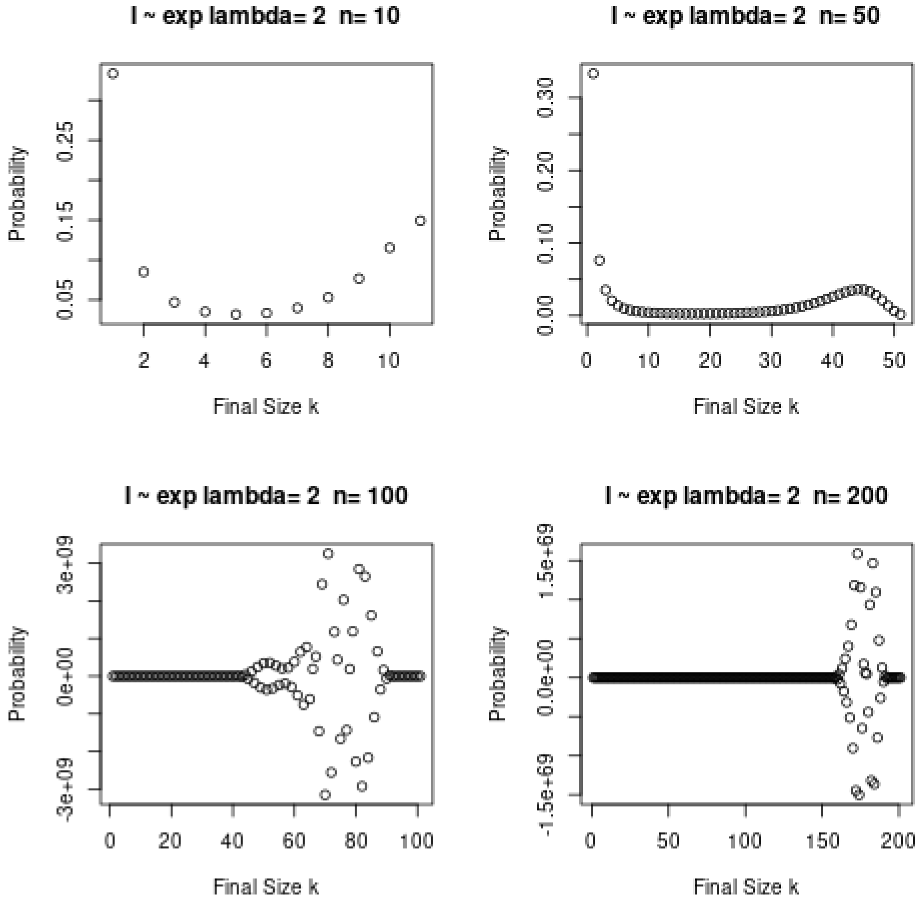
I ~ degenerate lambda= 5 n= 50

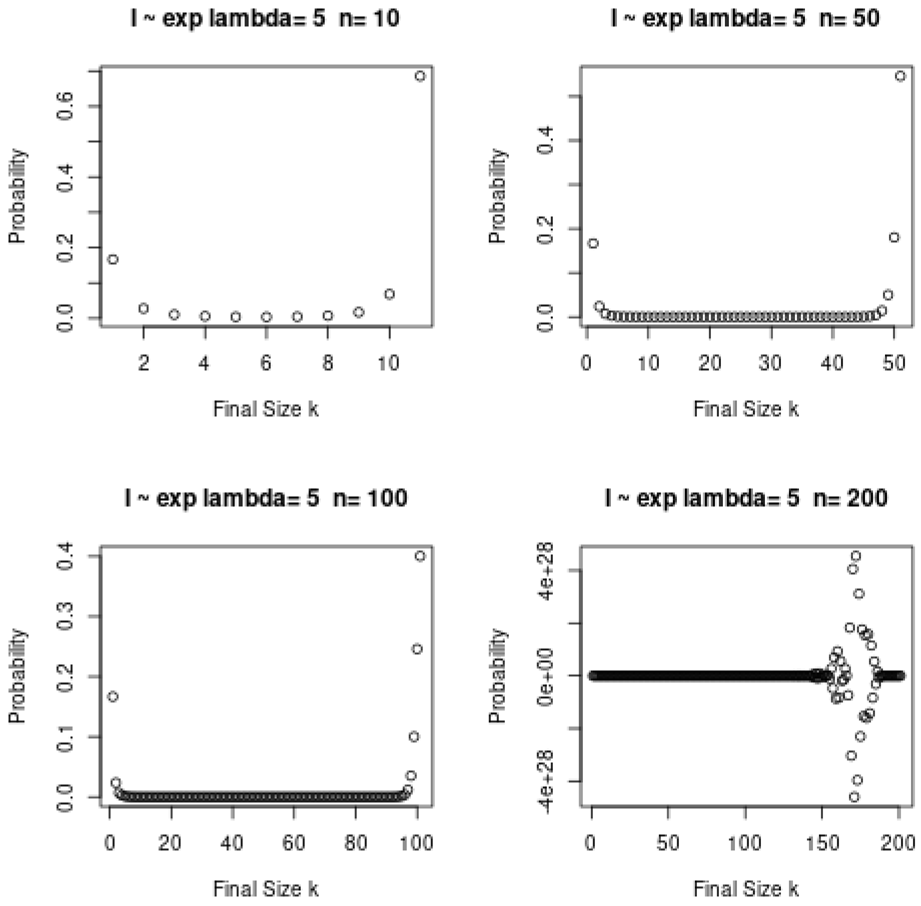
I ~ degenerate lambda= 5 n= 100

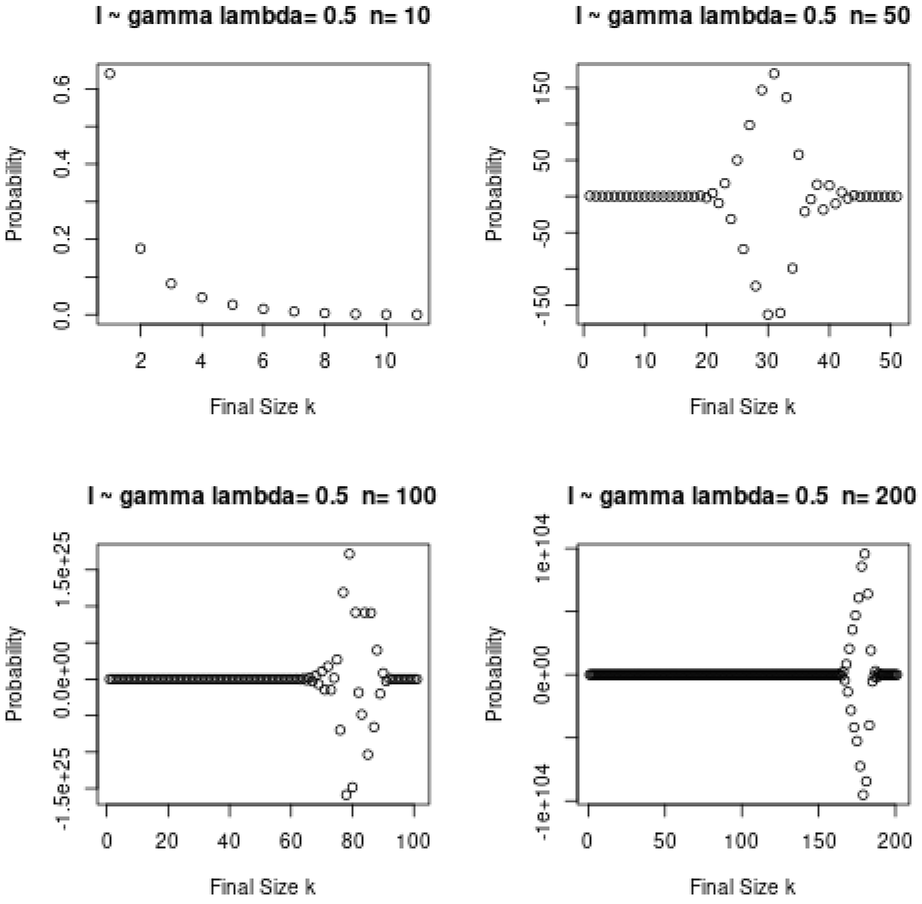
I ~ degenerate lambda= 5 n= 200

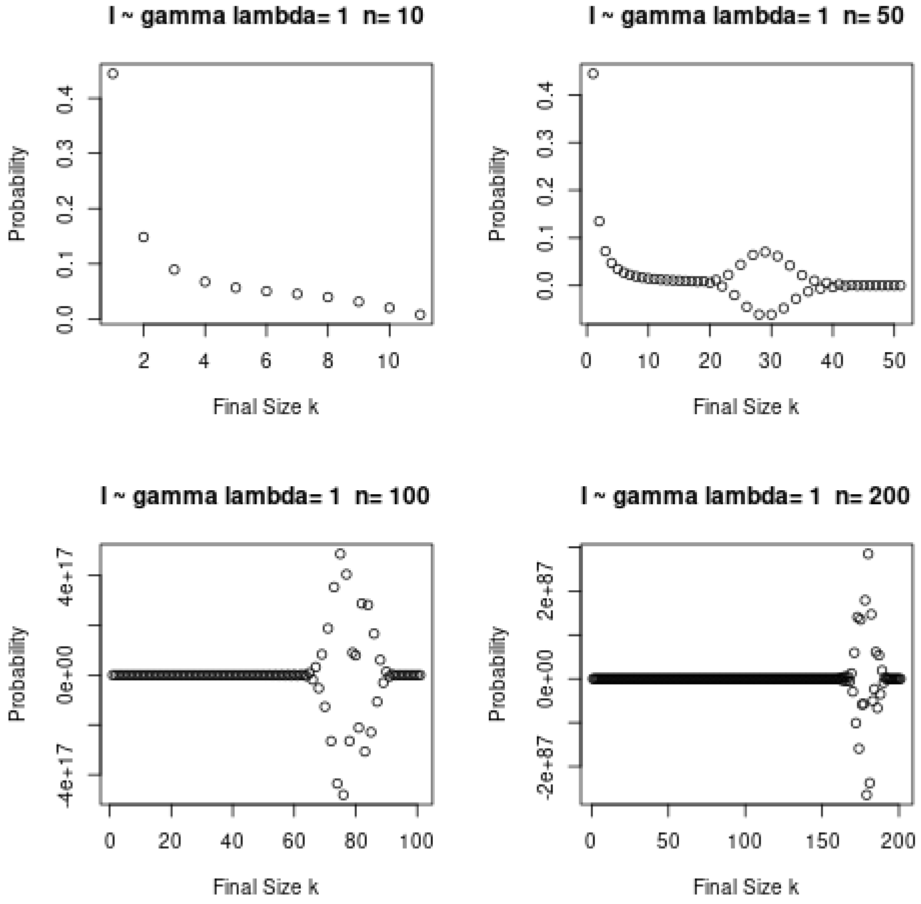

I ~ exp lambda= 0.5 n= 10
Probability
0.0
0.2
0.4
0.6
Final Size k
2
4
6
8
10
I ~ exp lambda= 0.5 n= 50
Probability
-200
0
100
Final Size k
0
10
20
30
40
50
I ~ exp lambda= 0.5 n= 100
Probability
-3e+25
0e+00
2e+25
Final Size k
0
20
40
60
80
100
I ~ exp lambda= 0.5 n= 200
Probability
-1e+105
0e+00
1e+105
Final Size k
0
50
100
150
200

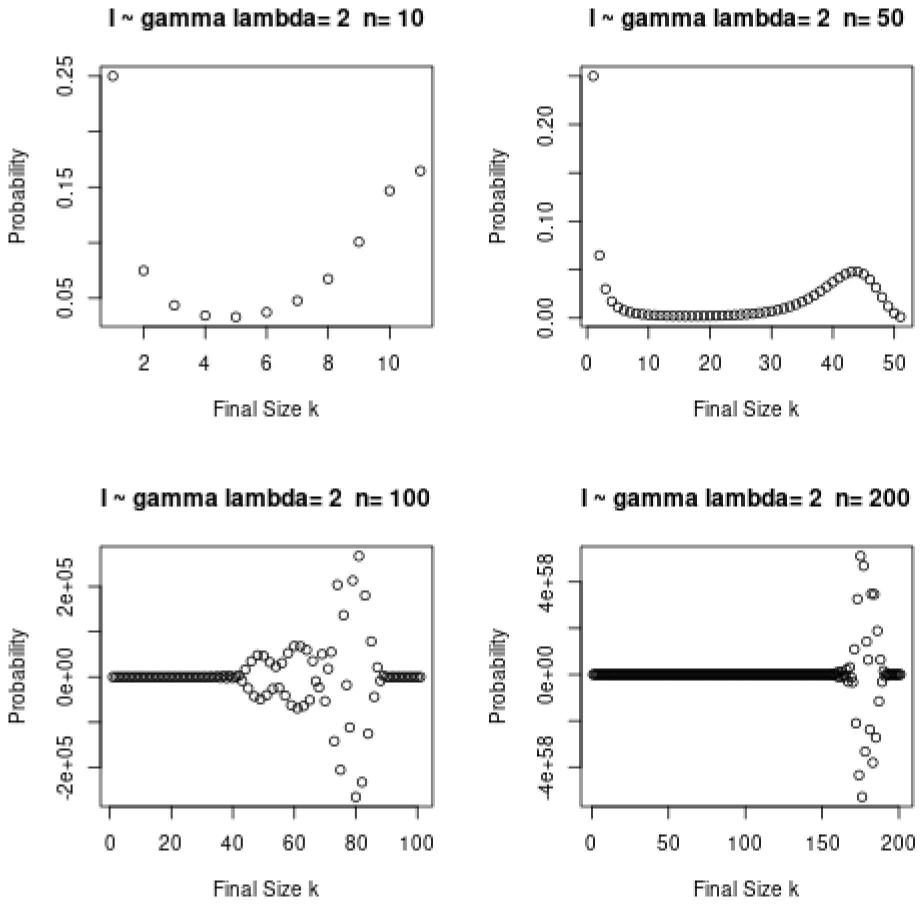

I ~ exp lambda= 1 n= 10
Probability
0.0
0.1
0.2
0.3
0.4
0.5
2
4
6
8
10
Final Size k
I ~ exp lambda= 1 n= 50
Probability
-0.2
0.0
0.2
0.4
0
10
20
30
40
50
Final Size k
I ~ exp lambda= 1 n= 100
Probability
-4e+19
0e+00
4e+19
0
20
40
60
80
100
Final Size k
I ~ exp lambda= 1 n= 200
Probability
-1e+91
0e+00
1e+91
0
50
100
150
200
Final Size k

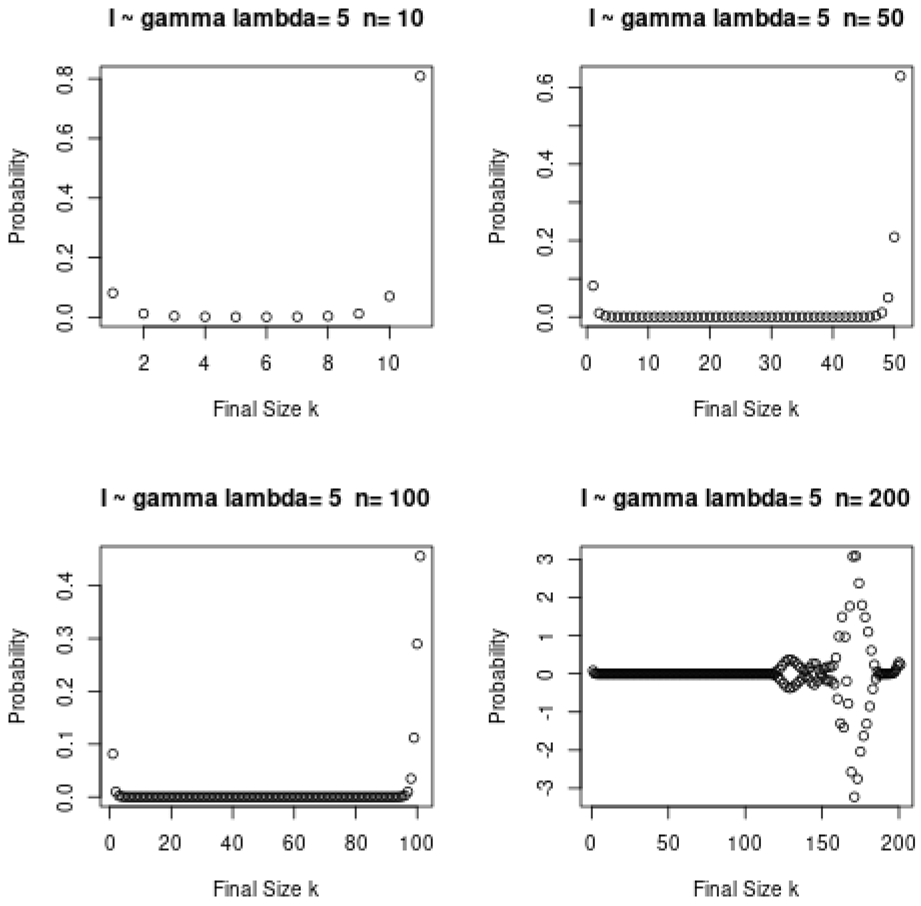

I ~ exp lambda= 2 n= 10
Probability
0.05
0.15
0.25
2
4
6
8
10
Final Size k
I ~ exp lambda= 2 n= 50
Probability
0.00
0.10
0.20
0.30
0
10
20
30
40
50
Final Size k
I ~ exp lambda= 2 n= 100
Probability
-3e+09
0e+00
3e+09
0
20
40
60
80
100
Final Size k
I ~ exp lambda= 2 n= 200
Probability
-1.5e+69
0.0e+00
1.5e+69
0
50
100
150
200
Final Size k


I ~ exp lambda= 5 n= 10
Probability
0.0
0.2
0.4
0.6
2
4
6
8
10
Final Size k
I ~ exp lambda= 5 n= 50
Probability
0.0
0.2
0.4
0
10
20
30
40
50
Final Size k
I ~ exp lambda= 5 n= 100
Probability
0.0
0.1
0.2
0.3
0.4
0
20
40
60
80
100
Final Size k
I ~ exp lambda= 5 n= 200
Probability
-4e+28
0e+00
4e+28
0
50
100
150
200
Final Size k


I ~ gamma lambda= 0.5 n= 10
Probability
0.0
0.2
0.4
0.6
2
4
6
8
10
Final Size k
I ~ gamma lambda= 0.5 n= 50
Probability
-150
-50
50
150
0
10
20
30
40
50
Final Size k
I ~ gamma lambda= 0.5 n= 100
Probability
-1.5e+25
0.0e+00
1.5e+25
0
20
40
60
80
100
Final Size k
I ~ gamma lambda= 0.5 n= 200
Probability
-1e+104
0e+00
1e+104
0
50
100
150
200
Final Size k


I ~ gamma lambda= 1 n= 10
Probability
0.0
0.1
0.2
0.3
0.4
2
4
6
8
10
Final Size k
I ~ gamma lambda= 1 n= 50
Probability
0.0
0.1
0.2
0.3
0.4
0
10
20
30
40
50
Final Size k
I ~ gamma lambda= 1 n= 100
Probability
-4e+17
0e+00
4e+17
0
20
40
60
80
100
Final Size k
I ~ gamma lambda= 1 n= 200
Probability
-2e+87
0e+00
2e+87
0
50
100
150
200
Final Size k


I ~ gamma lambda= 2 n= 10
Probability
0.05
0.15
0.25
2
4
6
8
10
Final Size k
I ~ gamma lambda= 2 n= 50
Probability
0.00
0.10
0.20
0
10
20
30
40
50
Final Size k
I ~ gamma lambda= 2 n= 100
Probability
-2e+05
0e+00
2e+05
0
20
40
60
80
100
Final Size k
I ~ gamma lambda= 2 n= 200
Probability
-4e+58
0e+00
4e+58
0
50
100
150
200
Final Size k


I ~ gamma lambda= 5 n= 10
Probability
0.0
0.2
0.4
0.6
0.8
2
4
6
8
10
Final Size k
I ~ gamma lambda= 5 n= 50
Probability
0.0
0.2
0.4
0.6
0
10
20
30
40
50
Final Size k
I ~ gamma lambda= 5 n= 100
Probability
0.0
0.1
0.2
0.3
0.4
0
20
40
60
80
100
Final Size k
I ~ gamma lambda= 5 n= 200
Probability
-3
-2
-1
0
1
2
3
0
50
100
150
200
Final Size k