

Master of Applied Mathematics Program  
Professional Master's Project Report  
Department of Applied Mathematics  
Illinois Institute of Technology

## **Cramer-Lundberg Model**

Submitted by  
Kin Hang Wong

*In partial fulfillment of the requirements of Master's candidates in  
Master of Applied Mathematics Program*

## Abstract

Ruin theory is motivated by the practical issue of solvency. Back in 1903, Swedish actuary Filip Lundberg introduced the Cramer-Lundberg model to understand the behaviour of the surplus process of insurance contracts. It became the foundation of modern ruin theory after Harald Cramer republished his work in 1930s.

Based on the classic Cramer-Lundberg model, there are different variants/extensions. For example: Sparre Andersen model[2](the waiting time between claims is not restricted to be a Poisson distribution), Classical Cramer-Lundberg Model with Stochastic Investment Return [1].

This project explained the mathematical structure of the Classic Cramer-Lundberg Model which is widely used by insurance industry. The simplest form of the model only consists initial capital, constant premium rate and an assumed claims distribution. Under this simple structure, we will show that the survival probability increases as the initial capital increases. Furthermore, if the surplus process lasts forever, we will show that probability of ruin is 100% if the premium rate does not meet a criteria called "net profit condition".

The simplest form of Cramer-Lundberg model is not enough for practical use for business. Therefore, throughout the derivation of the model, we should seek possibilities to implement additional features. One of the most intuitive and important feature is to implement investment income from bonds. Assuming the investment return is constant throughout the time horizon (from now to forever), i.e. no interest rate risks and no reinvestment risks, the model behaves differently compare with no investments. We will show that the net profit condition is not required for ultimate survival as long as the initial capital is large enough to earn sufficient investment income to compensate the premium shortfall. This model could be a reference when pricing a single premium product.

In this report, we will explore the Cramer-Lundberg model under three common claims distributions: Exponential, Hyper-exponential and Erlang distributions. The theoretical curve will be verified by simulations written in R[3].

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Cramer-Lundberg Model: No investment income</b>	<b>5</b>
2.1	General Claims Distribution . . . . .	6
2.2	Exponential Claims Distribution . . . . .	7
2.3	Hyper-exponential Claims Distribution . . . . .	9
2.4	Erlang Claims Distribution . . . . .	12
<b>3</b>	<b>Cramer-Lundberg Model: Constant investment income</b>	<b>14</b>
3.1	Exponential Claims Distribution . . . . .	14
3.2	Erlang Claims Distribution Simulation . . . . .	18
<b>4</b>	<b>Conclusion</b>	<b>20</b>
<b>5</b>	<b>References</b>	<b>20</b>
<b>6</b>	<b>Appendix</b>	<b>21</b>

# 1 Introduction

Insurers sell insurance contracts to policyholders to protect them from unexpected financial crises. For example a lump sum for medical treatment on critical illnesses, a lump sum death benefit to protect the beneficiaries of the diseased or a series of payment to the policyholder to support his/her retirement. Since the premium required per contract is relatively small compare to the benefit amount, if the product is under-priced, insurer will be at risk due to insufficient fund (insolvency).

Existing capital plus new premiums collected by insurer are responsible for the claims, expense, profit and investment. Therefore, it is utmost important to make sure the cash inflows(e.g. premium, investment income) for the insurer is sufficient to cover cash outflows(e.g. claims, expenses) to keep the business running and run good. For simplicity, this project will focus on the premium and claims. After the model structure being illustrated clearly, we will implement constant investment return and observe how the model behaves.

The Cramer-Lundberg model [1] has been used in insurance pricing for a long period of time. It was first introduced in 1903 by the Swedish actuary Filip Lundberg. The model measures the probability of ruin by assessing the surplus at all time points in terms of initial capital, constant premium rate and aggregate claim. The aggregate claim is a function of two independent random variables: number of claims and claim size. The number of claims follows a Poisson distribution; there is no restrictions on the claim size distribution.

As expected, the probability of ruin depends on the initial capital as well as the premium rate. The model also provides information on the sensitivity of ruin probability by altering the premium rate.

In section 2, we will explore the Cramer-Lundberg model under 3 different claims distribution: Exponential, Hyper-exponential and Erlang, when there is no investment income. We will see the ultimate ruin probability depends on the premium rate, i.e. whether it satisfied the net profit condition. After we derive the analytical solution, we will plot it against the simulation written in R.

In section 3, we will add the constant investment income feature to the classic Cramer-Lundberg model. We can see the ultimate ruin probability no longer depends on the net profit condition. As long as the initial capital is large enough, the ultimate ruin probability is zero.

## 2 Cramer-Lundberg Model: No investment income

The Cramer-Lundberg model involves the surplus  $C_t$  at time  $t$  which is given by[1]:

$$C_t = u + ct - \sum_{i=1}^{N_t} X_i \quad (1)$$

Here  $u$  is the initial capital, i.e.  $C_0 = u$ ,  $c$  is the constant premium rate, the sequence of claims  $\{X_1, X_2, \dots, X_i\}$  are independent and identically distributed(i.i.d) random variables, and  $N_t$  is the number of claims occurred by time  $t$ . The uncertainty comes from both  $X_i$  and  $N_t$ , where  $N_t$  is assumed to follow a Poisson Process with rate  $\lambda$  and  $X_i$  has density function  $f(x)$  with  $E(X_i) = \mu$ . In addition, we can denote the  $i^{th}$  claim occurred at  $T_i$  so that  $T_i - T_{i-1}$  follows exponential distribution with mean  $1/\lambda$ .

Given this model, insurance companies are interested in some basic information:

- **Probability of Ultimate Ruin** given initial capital  $u$  i.e.

$$\psi(u) = Pr\left(\inf_{s>0}\{C_s\} < 0 | C_0 = u\right) \quad (2)$$

where the corresponding survival function is,

$$\phi(u) = 1 - \psi(u) \quad (3)$$

- **Net Profit Condition**, i.e.  $c > \lambda\mu$ , which is a necessary condition to guarantee that the surplus process has zero probability of ultimate ruin given infinite amount of initial capital, i.e.

$$\lim_{u \rightarrow \infty} \psi(u) = \lim_{u \rightarrow \infty} Pr\left(\inf_{s>0}\{C_s\} \leq 0 | C_0 = u\right) = 0 \quad (4)$$

Or equivalently, the certainty of survival given infinite amount of initial capital,

$$\lim_{u \rightarrow \infty} \phi(u) = 1 \quad (5)$$

*Proof.* Let  $Y_i = c(T_i - T_{i-1}) - X_i$  be the net cash inflow between claim times, i.e. total premium received between  $T_i$  and  $T_{i-1}$  minus the amount of the  $i^{th}$  claim occurred at time  $T_i$ . Then, we require every  $Y_i$  to be positive to avoid ruin at all time, i.e.

$$\lim_{u \rightarrow \infty} Pr\left(\inf_{s>0}\{C_s\} \leq 0 | C_0 = u\right) = 0 \iff \{Y_i : Y_i \leq 0\} = \emptyset$$

Equivalently,

$$c(T_i - T_{i-1}) - X_i > 0$$

Since  $N_t$  is a Poisson process with mean  $\lambda$ , the time between each claim is exponential with mean  $1/\lambda$ . By taking expectation on both sides of the inequality, we obtained:

$$\begin{aligned} cE(T_i - T_{i-1}) - E(X_i) &> 0 \\ c/\lambda - \mu &> 0 \\ c &> \lambda\mu = (1 + \theta)\lambda\mu \end{aligned} \quad (6)$$

where  $\theta > 0$  is called the **premium loading**. □

## 2.1 General Claims Distribution

Instead of looking at the ruin probability  $\psi(u)$ , it is more convenient to look at the ultimate survival probability  $\phi(u) = 1 - \psi(u)$ . We now derive an equation for  $\phi(u)$  using the Law of Total Probability. Consider two events: One claim occurred within time interval  $h$ , i.e.  $Pr(\text{One claim occurred within } h) = 1 - e^{-\lambda h}$ , or not, i.e.  $Pr(\text{No claims occurred within } h) = e^{-\lambda h}$ , we obtained the following equation.

$$\phi(u) = e^{-\lambda h} \phi(u + ch) + (1 - e^{-\lambda h}) \int_0^\infty \phi(u + ch - x) f(x) dx$$

where  $f(x)$  be the probability density function of the claims sizes  $X_i$ . We perform some algebra to obtain

$$\frac{\phi(u) - \phi(u + ch)}{ch} = \frac{e^{-\lambda h} - 1}{ch} \phi(u + ch) + \frac{1 - e^{-\lambda h}}{ch} \int_0^\infty \phi(u + ch - x) f(x) dx$$

and taking  $\lim_{h \rightarrow 0}$  on both sides, we obtain the **backward equation**, i.e.

$$-\phi'(u) = \frac{-\lambda}{c} \phi(u) + \frac{\lambda}{c} \int_0^\infty \phi(u - x) f(x) dx \quad (7)$$

Since  $\phi(x) = 0$  for  $x < 0$  (immediate ruin), we have the **Cramer-Lundberg equation**, i.e.

$$c\phi'(u) = \lambda \left[ \phi(u) - \int_0^u \phi(u - x) f(x) dx \right] \quad (8)$$

This is an integro-differential equation. We have an auxiliary condition that survival is sure if the initial reserve is infinite, i.e.  $\phi(\infty) = 1$ . It will be useful to have the condition at  $u = 0$ ,  $\phi(0)$ .

We find  $\phi(0)$  as follows:

$$\begin{aligned} c \int_0^u \phi'(x) dx &= \int_0^u \lambda \phi(x) dx - \int_0^u \int_0^x \lambda \phi(x - y) f(y) dy dx \\ \frac{c}{\lambda} [\phi(u) - \phi(0)] &= \int_0^u \phi(x) dx - \int_0^u \int_0^x \phi(x - y) dF(y) dx \\ &= \int_0^u \phi(x) dx - \int_0^u \int_0^x \phi(x - y) dF(y) dx \\ &= \int_0^u \phi(x) dx - \int_0^u \int_0^{u-x} d[\phi(x) F(y)] dx \\ &= \int_0^u \phi(x) [1 - F(u - x)] dx \\ &= \int_0^u \phi(u - x) [1 - F(x)] dx \end{aligned}$$

where  $F(x)$  is the probability distribution function of the claims. By taking  $\lim_{u \rightarrow \infty}$  on both sides, since  $\lim_{u \rightarrow \infty} \phi(u) = \lim_{u \rightarrow \infty} (1 - \psi(u)) = 1$  and by net profit condition  $c > \lambda\mu$ , we find

$$\begin{aligned}\frac{c}{\lambda}[1 - \phi(0)] &= \int_0^\infty [1 - F(x)]dx \\ \frac{c}{\lambda}[1 - \phi(0)] &= E(ClaimSize) = \mu \\ \phi(0) &= 1 - \frac{\lambda\mu}{c} > 0\end{aligned}$$

So that even if we start with a very small surplus, the ultimate probability of survival is not zero. We then obtained the problem,

$$c\phi'(u) = \lambda \left[ \phi(u) - \int_0^u \phi(u-x)f(x)dx \right] \quad (9)$$

$$\phi(0) = 1 - \frac{\lambda\mu}{c}, \quad \phi(\infty) = 1 \quad (10)$$

We can solve (9) with (10) using Laplace transforms. We define the Laplace transforms as  $\mathcal{L}\{\phi(u)\} = \Phi(s)$  and  $\mathcal{L}\{f(u)\} = \Omega(s)$  and then apply the Laplace transform to (9) with (10). We solve for  $\Phi(s)$  to obtain

$$\Phi(s) = \frac{c(1 - \frac{\lambda\mu}{c})}{cs + \lambda\Omega(s) - \lambda} = \frac{c - \lambda\mu}{cs + \lambda\Omega(s) - \lambda} \quad (11)$$

Here  $\Omega(s)$  depends on the claims distribution, then using inverse Laplace Transformation on  $\Phi(s)$ , we could solve for  $\phi(u)$  explicitly.

## 2.2 Exponential Claims Distribution

If the claims amount follows the exponential distribution, i.e.

$$X_i \sim \exp(\alpha)$$

then we have,

$$\begin{aligned}F(x) &= 1 - e^{-\alpha x}, \quad f(x) = \alpha e^{-\alpha x}, \quad \mu = \frac{1}{\alpha} \\ \Omega(s) &= \mathcal{L}\{\alpha e^{-\alpha x}\} = \frac{\alpha}{\alpha + s}\end{aligned}$$

Equation (11) becomes,

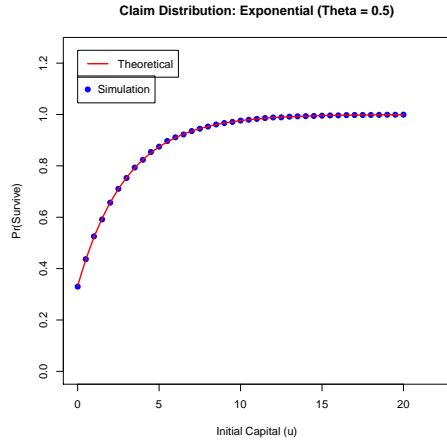
$$\Phi(s) = \frac{c - \frac{\lambda}{\alpha}}{cs + \lambda(\frac{\alpha}{\alpha+s}) - \lambda} = \frac{(c\alpha - \lambda)(s + \alpha)}{s\alpha(sc + c\alpha - \lambda)} = \frac{1}{s} - \left(\frac{\lambda}{c\alpha}\right) \frac{1}{s + \frac{c\alpha - \lambda}{c}}$$

then by inverse Laplace transformation,

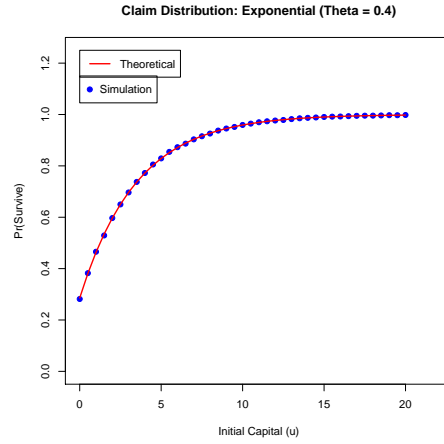
$$\phi(u) = 1 - \frac{\lambda}{c\alpha} e^{-(\alpha - \frac{\lambda}{c})u} \quad (12)$$

$$\psi(u) = \frac{\lambda}{c\alpha} e^{-(\alpha - \frac{\lambda}{c})u} \quad (13)$$

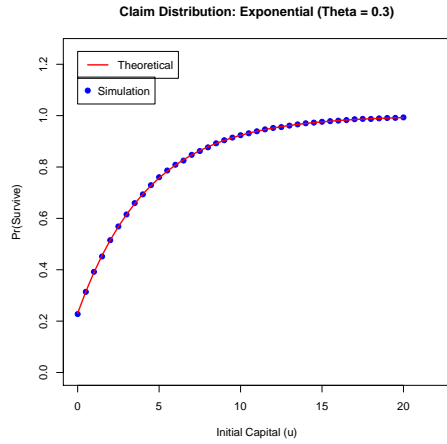
This theoretical survival function can be verified by simulation in R[3], see Figure (1) and Figure (2). For each initial capital amount  $u_i$ ,  $0 \leq u_i \leq 20$  with increment 0.5, we generate 400 claims amount (4400 for  $u = 0$  for better illustration) for each of the 10000 simulations and see how many ruined cases among the 10000 simulations, this fraction is an approximated value of  $\phi(u_i)$ . Plot the 41 points  $(u_i, \phi(u_i))$ , the survival function can be estimated. The survival probability depends on the premium loading  $\theta$ , i.e.  $c = (1 + \theta)\lambda\mu = (1 + \theta)\lambda/\alpha$ . If  $\theta$  is positive, net profit condition is met, otherwise net profit condition is not met. Graphs below compared the theoretical survival function and the simulated survival function for  $\theta \in \{0.5, 0.4, 0.3, 0.2, 0.1, 0\}$ . For cases that the net profit condition is not satisfied, i.e.  $\phi(u) = 0$  for all  $u$ . The simulations failed to illustrate this nicely because finite amount of claims cannot mimic infinite time very well.



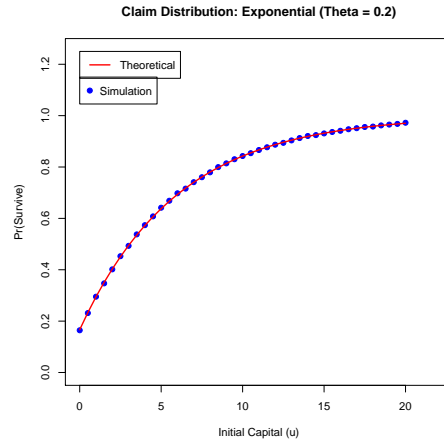
(a)  $\alpha = 1, \theta = 0.5$



(b)  $\alpha = 1, \theta = 0.4$



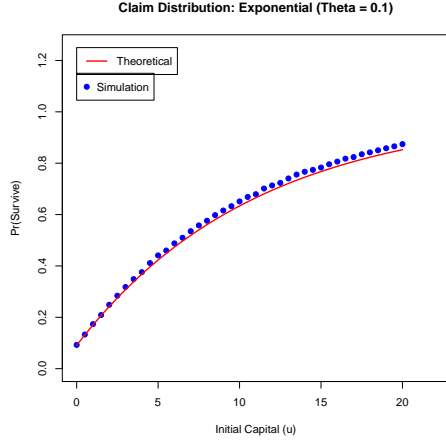
(c)  $\alpha = 1, \theta = 0.3$



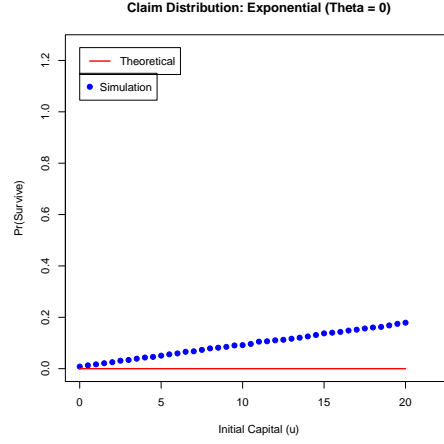
(d)  $\alpha = 1, \theta = 0.2$

Figure 1: Survival probability under exponential claims distribution without investment





(a)  $\alpha = 1, \theta = 0.1$



(b)  $\alpha = 1, \theta = 0$

Figure 2: When  $\theta = 0$ , the required no. of claims to ruin the process is way larger than the 4400 claims we used in the simulation as  $u$  increases. Therefore, we can see the discrepancy is larger when  $u$  is larger.

## 2.3 Hyper-exponential Claims Distribution

A hyper-exponential distribution is a mixture of  $n$  exponential distributions. If the claims distribution follows the hyper-exponential distribution, for simplicity we let  $n = 2$  and  $\beta > \alpha$ , i.e.

$$X_i \sim \text{hyperexp} \left( \begin{matrix} p, 1-p \\ \alpha, \beta \end{matrix} \right)$$

then we have,

$$F(x) = 1 - pe^{-\alpha x} - (1-p)e^{-\beta x}, \quad f(x) = p\alpha e^{-\alpha x} + (1-p)\beta e^{-\beta x}$$

$$\mu = \frac{p}{\alpha} + \frac{1-p}{\beta}$$

$$\Omega(s) = \mathcal{L}\{f(x)\} = \frac{p\alpha}{\alpha + s} + \frac{(1-p)\beta}{\beta + s}$$

equation (11) becomes,

$$\Phi(s) = \frac{c - \lambda\mu}{cs + \lambda\left(\frac{p\alpha}{\alpha+s} + \frac{(1-p)\beta}{\beta+s}\right) - \lambda} = \frac{\left(1 - \frac{\lambda\mu}{c}\right)(\alpha + s)(\beta + s)}{s\left[s^2 + \left(\alpha + \beta - \frac{\lambda}{c}\right)s + \alpha\beta\left(1 - \frac{\lambda\mu}{c}\right)\right]}$$

by looking at  $D(s) = s^2 + \left(\alpha + \beta - \frac{\lambda}{c}\right)s + \alpha\beta\left(1 - \frac{\lambda\mu}{c}\right)$ ,

$$\Delta = \left(\alpha + \beta - \frac{\lambda}{c}\right)^2 - 4\alpha\beta\left(1 - \frac{\lambda\mu}{c}\right) = \left(\beta - \alpha - \frac{\lambda}{c}\right)^2 + \frac{4\lambda p(\beta - \alpha)}{c} > 0.$$

This implies we could perform partial fraction on  $\Phi(s)$  and get,

$$\begin{aligned}\Phi(s) &= \frac{A}{s} + \frac{B}{s-r_1} + \frac{C}{s-r_2} \\ \phi(u) &= A + Be^{r_1 u} + Ce^{r_2 u} \\ \psi(u) &= 1 - A - Be^{r_1 u} - Ce^{r_2 u}\end{aligned}$$

where  $r_1$  and  $r_2$  are distinct real roots of  $D(s) = 0$ .

Now we have to find  $r_1, r_2, A, B$  and  $C$  to complete  $\psi(u)$ . From the partial fraction,

$$\begin{aligned}\left(1 - \frac{\lambda\mu}{c}\right)(\alpha + s)(\beta + s) &= A(s-r_1)(s-r_2) + Bs(s-r_2) + Cs(s-r_1) \\ s=0 &\implies A=1 \\ s=r_1 &\implies B = \frac{\left(1 - \frac{\lambda\mu}{c}\right)(r_1 + \alpha)(r_1 + \beta)}{r_1(r_1 - r_2)} \\ s=r_2 &\implies C = \frac{\left(1 - \frac{\lambda\mu}{c}\right)(r_2 + \alpha)(r_2 + \beta)}{r_2(r_2 - r_1)} \\ (r_1, r_2) &= \left(\frac{-\left(\alpha + \beta - \frac{\lambda}{c}\right) + \sqrt{\Delta}}{2}, \frac{-\left(\alpha + \beta - \frac{\lambda}{c}\right) - \sqrt{\Delta}}{2}\right) \\ \Delta &= \left(\alpha + \beta - \frac{\lambda}{c}\right)^2 - 4\alpha\beta\left(1 - \frac{\lambda\mu}{c}\right)\end{aligned}$$

By product of roots and net profit condition, i.e.  $1 - \frac{\lambda\mu}{c} > 0$ , we have

$$\begin{aligned}r_1 &= \frac{-\left(\alpha + \beta - \frac{\lambda}{c}\right) + \sqrt{\left(\alpha + \beta - \frac{\lambda}{c}\right)^2 - 4\alpha\beta\left(1 - \frac{\lambda\mu}{c}\right)}}{2} < 0 \\ r_1 r_2 &= \alpha\beta\left(1 - \frac{\lambda\mu}{c}\right) > 0 \implies r_2 < 0\end{aligned}$$

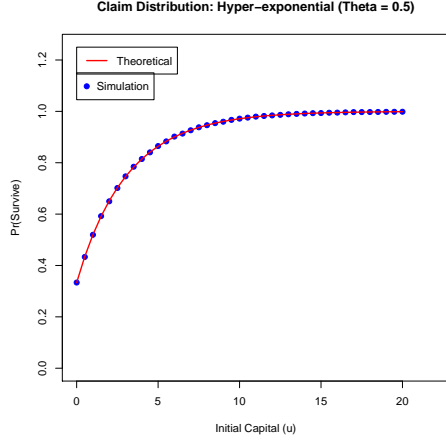
After all,  $\phi(u)$  and  $\psi(u)$  could be found by gathering all information below:

$$\begin{aligned}\phi(u) &= 1 + \left[\frac{\left(1 - \frac{\lambda\mu}{c}\right)(r_1 + \alpha)(r_1 + \beta)}{r_1\sqrt{\Delta}}\right]e^{r_1 u} + \left[\frac{\left(1 - \frac{\lambda\mu}{c}\right)(r_2 + \alpha)(r_2 + \beta)}{-r_2\sqrt{\Delta}}\right]e^{r_2 u} \quad (14) \\ \psi(u) &= 1 - \phi(u) \\ \mu &= \frac{p}{\alpha} + \frac{1-p}{\beta}, \quad \Delta = \left(\alpha + \beta - \frac{\lambda}{c}\right)^2 - 4\alpha\beta\left(1 - \frac{\lambda\mu}{c}\right) \\ r_1 &= \frac{-\left(\alpha + \beta - \frac{\lambda}{c}\right) + \sqrt{\Delta}}{2}, \quad r_2 = \frac{-\left(\alpha + \beta - \frac{\lambda}{c}\right) - \sqrt{\Delta}}{2}\end{aligned}$$

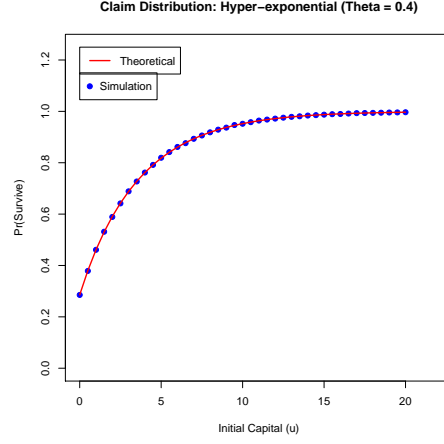
Simulation vs theoretical plots are shown in Figure (3). In the simulations,  $c = (1 + \theta)\lambda\left(\frac{p}{\alpha} + \frac{1-p}{\beta}\right)$ .

By putting  $p = 1$ , hyper-exponential distribution will be reduced to exponential distribution, i.e.

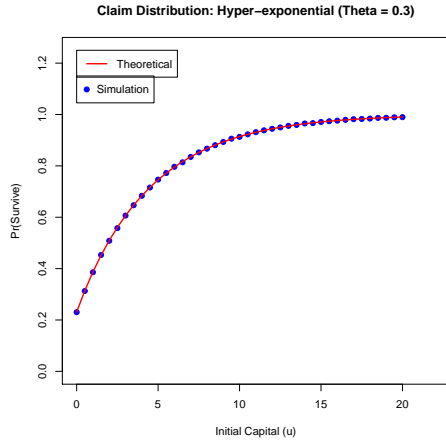
$$\begin{aligned}\mu &= \frac{1}{\alpha}, \quad \Delta = \left(\alpha - \frac{\lambda}{c}\right)^2 \\ r_1 &= 0, \quad r_2 = -\alpha + \frac{\lambda}{c}\end{aligned}$$



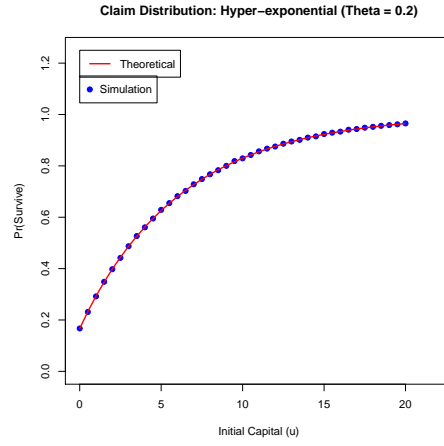
(a)  $\alpha = 1, \theta = 0.5$



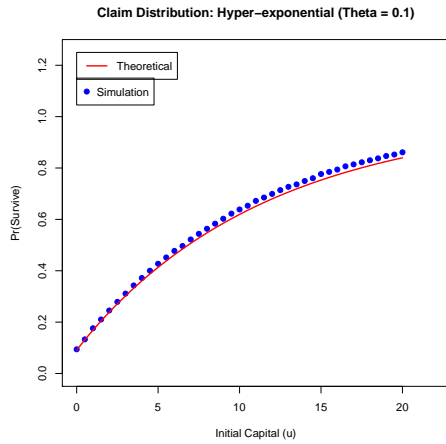
(b)  $\alpha = 1, \theta = 0.4$



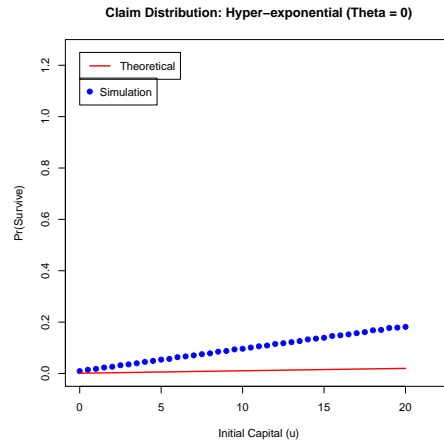
(c)  $\alpha = 1, \theta = 0.3$



(d)  $\alpha = 1, \theta = 0.2$



(e)  $\alpha = 1, \theta = 0.1$



(f)  $\alpha = 1, \theta = 0.001$

Figure 3: Survival probability under Hyper-exponential claims distribution without investment. Note that for illustration purpose,  $\theta = 0.001$  is used to approximate  $\theta = 0$ . When  $\theta = 0$ , the required no. of claims to ruin the process is way larger than the 4400 claims we used in the simulation as  $u$  increases. Therefore, we can see the discrepancy is larger when  $u$  is larger.

Therefore,

$$\phi(u) = 1 - \frac{\lambda}{c\alpha} \exp^{-(\alpha - \frac{\lambda}{c})u} \quad (15)$$

$$\psi(u) = \frac{\lambda}{c\alpha} \exp^{-(\alpha - \frac{\lambda}{c})u} \quad (16)$$

which agrees with the previous result (12) and (13).

## 2.4 Erlang Claims Distribution

An Erlang distribution is a special case of Gamma distributions where the shape parameter  $k$  is an integer. If the claims distribution follows Erlang distribution, for simplicity we let  $shape = k = 2$  and  $rate = \beta = 2$  such that the mean of the distribution is 1, i.e.

$$X_i \sim \text{Gamma}(k = 2, \beta)$$

Then we have,

$$\begin{aligned} F(x) &= \gamma(2, \beta x) = 1 - e^{-\beta x} - \beta x e^{-\beta x}, \quad f(x) = \beta^2 x e^{-\beta x} \\ \mu &= \frac{2}{\beta} \\ \Omega(s) &= \mathcal{L}\{\beta^2 x e^{-\beta x}\} = \frac{\beta^2}{(s + \beta)^2} \end{aligned}$$

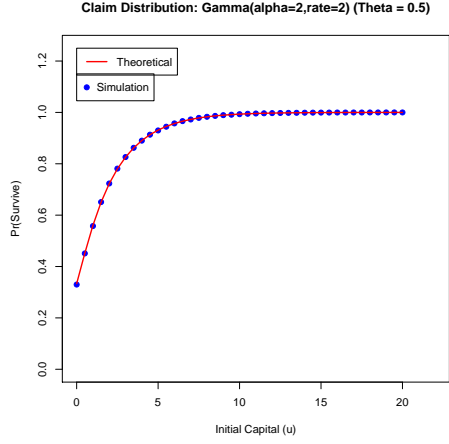
Equation (11) becomes,

$$\begin{aligned} \Phi(s) &= \frac{c - \frac{2\lambda}{\beta}}{cs + \frac{\beta^2 \lambda}{(s + \beta)^2} - \lambda} \\ &= \frac{(c - \frac{2\lambda}{\beta})(s + \beta)^2}{cs[s^2 + (2\beta - \lambda/c)s + \beta(\beta - 2\lambda/c)]} \\ &= \frac{(c - \frac{2\lambda}{\beta})(s + \beta)^2}{cs(s - r_1)(s - r_2)} \\ r_1 &= \frac{-(2\beta - \lambda/c) + \sqrt{(2\beta - \lambda/c)^2 - 4\beta(\beta - 2\lambda/c)}}{2} < 0, \quad c > \lambda\mu = \frac{2\lambda}{\beta} \\ r_2 &= \frac{-(2\beta - \lambda/c) - \sqrt{(2\beta - \lambda/c)^2 - 4\beta(\beta - 2\lambda/c)}}{2} < 0, \quad c > \frac{2\lambda}{\beta} \end{aligned} \quad (17)$$

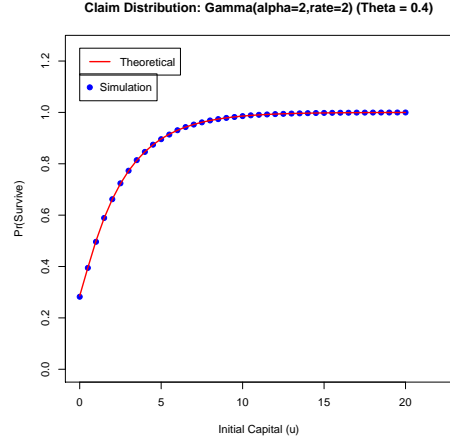
By partial fraction, (17) becomes,

$$\begin{aligned} \Phi(s) &= \frac{1}{s} + \frac{(c - \lambda)(r_1 + \beta)^2}{cr_1(r_1 - r_2)} \frac{1}{s - r_1} + \frac{(c - \lambda)(r_2 + \beta)^2}{cr_2(r_2 - r_1)} \frac{1}{s - r_2} \\ \phi(u) &= 1 + \frac{(c - \frac{2\lambda}{\beta})(r_1 + \beta)^2}{cr_1(r_1 - r_2)} e^{r_1 u} + \frac{(c - \frac{2\lambda}{\beta})(r_2 + \beta)^2}{cr_2(r_2 - r_1)} e^{r_2 u} \end{aligned} \quad (18)$$

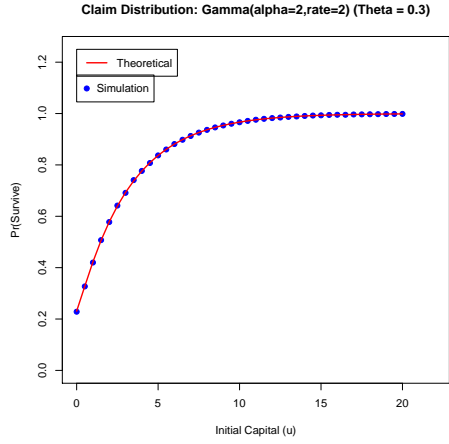
The simulation results are shown in Figure (4). Note that in the simulations,  $c = (1 + \theta) \frac{2\lambda}{\beta}$ .



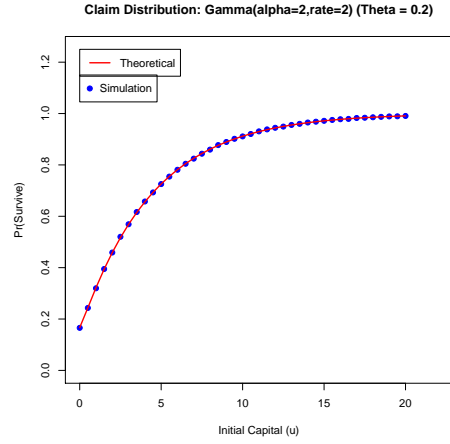
(a)  $\alpha = 2, \beta = 2, \theta = 0.5$



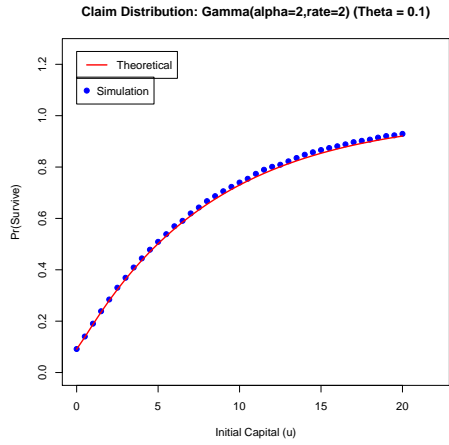
(b)  $\alpha = 2, \beta = 2, \theta = 0.4$



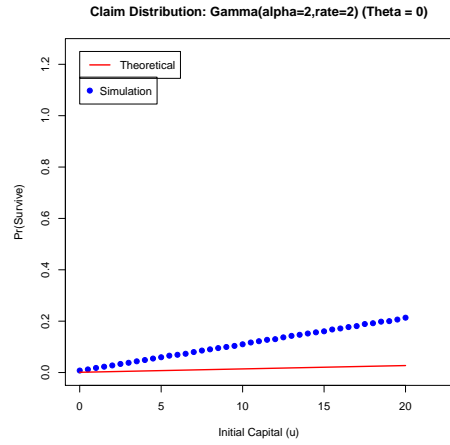
(c)  $\alpha = 2, \beta = 2, \theta = 0.3$



(d)  $\alpha = 2, \beta = 2, \theta = 0.2$



(e)  $\alpha = 2, \beta = 2, \theta = 0.1$



(f)  $\alpha = 2, \beta = 2, \theta = 0.001$

Figure 4: Survival probability under Erlang claims distribution without investment. Note that for illustration purpose,  $\theta = 0.001$  is used to approximate  $\theta = 0$ . When  $\theta = 0$ , the required no. of claims to ruin the process is way larger than the 4400 claims we used in the simulation as  $u$  increases. Therefore, we can see the discrepancy is larger when  $u$  is larger.

### 3 Cramer-Lundberg Model: Constant investment income

#### 3.1 Exponential Claims Distribution

Recall equation (8), the left side indicating the constant cash-inflow from premiums. A similar equation consisting stochastic investment is stated by Asmussen as equation (6.1) in his book[1]. To use the same equation for constant investment, there would be no variance and no Brownian motion, i.e.

$$(c + ru)\phi'(u) = \lambda \left[ \phi(u) - \int_0^u \phi(u-x)f(x)dx \right] \quad (19)$$

when the claim distribution is exponential,  $f(x) = \alpha e^{-\alpha x}$ , then (19) becomes,

$$(c + ru)\phi'(u) = \lambda \left[ \phi(u) - \int_0^u \phi(u-x)\alpha e^{-\alpha x}dx \right] \quad (20)$$

where

$$\phi(\infty) = 1, \quad c\phi'(0) - \lambda\phi(0) = 0. \quad (21)$$

Equation (20) cannot be solved easily using Laplace Transformation. Instead, we convert (20) into an ODE. First we let  $y = u - x$ ,

$$\int_0^u \phi(u-x)\alpha e^{-\alpha x}dx = \alpha e^{-\alpha u} \int_0^u \phi(y)e^{\alpha y}dy$$

now (20) becomes,

$$(c + ru)[e^{\alpha u}\phi'(u)] = \lambda \left[ e^{\alpha u}\phi(u) - \alpha \int_0^u \phi(y)e^{\alpha y}dy \right] \quad (22)$$

we now differentiate both sides,

$$re^{\alpha u}\phi'(u) + (c + ru)[\alpha e^{\alpha u}\phi'(u) + e^{\alpha u}\phi''(u)] = \lambda \left[ e^{\alpha u}\phi'(u) + \alpha e^{\alpha u}\phi(u) - \alpha\phi(u)e^{\alpha u} \right]$$

by grouping terms,

$$\frac{\phi''(u)}{\phi'(u)} = \frac{\lambda - r - \alpha(c + ru)}{c + ru} \quad (23)$$

by integrating both sides,

$$\begin{aligned} \ln[\phi'(u)] &= \frac{\lambda - r}{r} \ln(c + ru) - \alpha u + K \\ \phi'(u) &= K_1(c + ru)^{\frac{\lambda}{r}-1} e^{-\alpha u} \end{aligned} \quad (24)$$

by integrating again,

$$\phi(u) = K_2 - K_1 \int_u^\infty (c + r\hat{u})^{\frac{\lambda}{r}-1} e^{-\alpha \hat{u}} d\hat{u} \quad (25)$$

now we let  $w = c + r\hat{u}$ ,

$$\begin{aligned}
\phi(u) &= K_2 - \frac{K_1}{r} \int_{c+ru}^{\infty} w^{\frac{\lambda}{r}-1} e^{-\alpha(\frac{w-c}{r})} dw \\
&= K_2 - K_1 \left(\frac{r}{\alpha}\right)^{\frac{\lambda}{r}-1} \frac{e^{\frac{\alpha c}{r}}}{\alpha} \int_{c+ru}^{\infty} \left(\frac{\alpha w}{r}\right)^{\frac{\lambda}{r}-1} e^{-(\frac{\alpha w}{r})} d\left(\frac{\alpha w}{r}\right) \\
&= K_2 - K_1 \left(\frac{r}{\alpha}\right)^{\frac{\lambda}{r}-1} \frac{e^{\frac{\alpha c}{r}}}{\alpha} \Gamma\left(\frac{\lambda}{r}, \frac{\alpha c}{r} + \alpha u\right)
\end{aligned} \tag{26}$$

where  $\Gamma$  is the upper incomplete gamma function. By (21) and taking  $\lim_{u \rightarrow \infty}$  on both sides, we can get  $K_2 = 1$  Therefore, we have a boundary value problem,

$$\phi(u) = 1 - K_1 \left(\frac{r}{\alpha}\right)^{\frac{\lambda}{r}-1} \frac{e^{\frac{\alpha c}{r}}}{\alpha} \Gamma\left(\frac{\lambda}{r}, \frac{\alpha c}{r} + \alpha u\right) \tag{27}$$

$$\phi'(u) = K_1 (c + ru)^{\frac{\lambda}{r}-1} e^{-\alpha u} \tag{28}$$

$$c\phi'(0) - \lambda\phi(0) = 0 \tag{29}$$

$$\phi(\infty) = 1 \tag{30}$$

by substitution,  $K_1$  could be found as below,

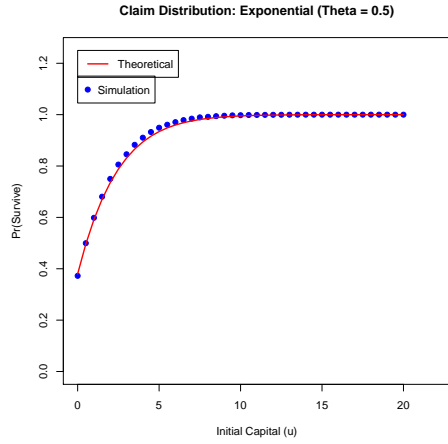
$$K_1 = \frac{\lambda}{c^{\frac{\lambda}{r}} + \frac{\lambda}{\alpha} \left(\frac{r}{\alpha}\right)^{\frac{\lambda}{r}-1} e^{\frac{\alpha c}{r}} \alpha \Gamma\left(\frac{\lambda}{r}, \frac{\alpha c}{r}\right)}$$

therefore,

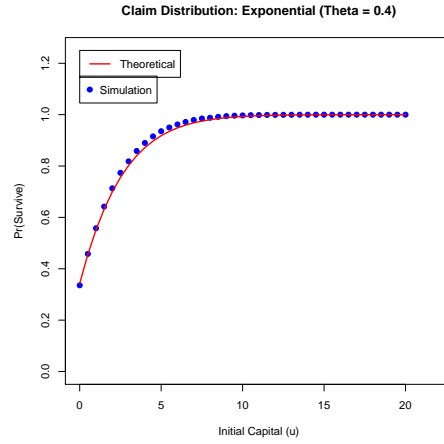
$$\phi(u) = 1 - \frac{\lambda e^{\frac{\alpha c}{r}} \Gamma\left(\frac{\lambda}{r}, \frac{\alpha c}{r} + \alpha u\right)}{r \left(\frac{\alpha c}{r}\right)^{\frac{\lambda}{r}} + \lambda e^{\frac{\alpha c}{r}} \Gamma\left(\frac{\lambda}{r}, \frac{\alpha c}{r}\right)} \tag{31}$$

The same result can be found by Mathematica using the code in Appendix.

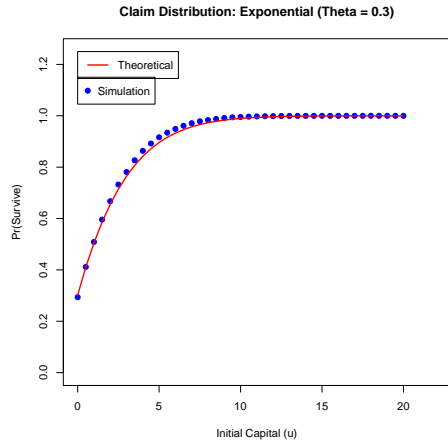
Using  $c = (1 + \theta)\lambda/\alpha$ , The simulation results are shown in Figure (5) and Figure (6). Note that  $\theta$  does not necessary to be positive, means that net profit condition is not a must for survival. Note that if  $\theta = -1$ ,  $c = 0$  means no premiums.



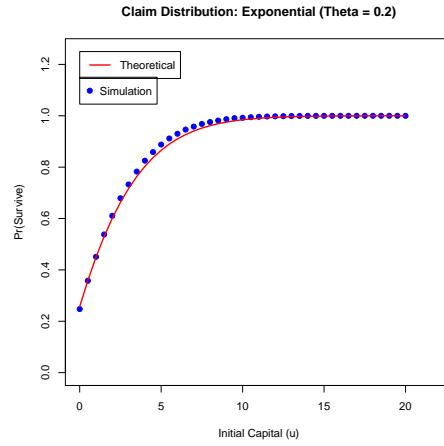
(a)  $\alpha = 1, \theta = 0.5, r = 0.05$



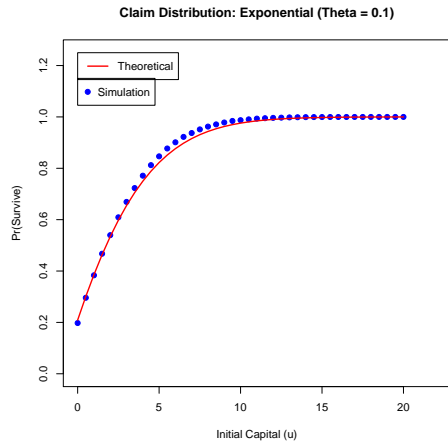
(b)  $\alpha = 1, \theta = 0.4, r = 0.05$



(c)  $\alpha = 1, \theta = 0.3, r = 0.05$



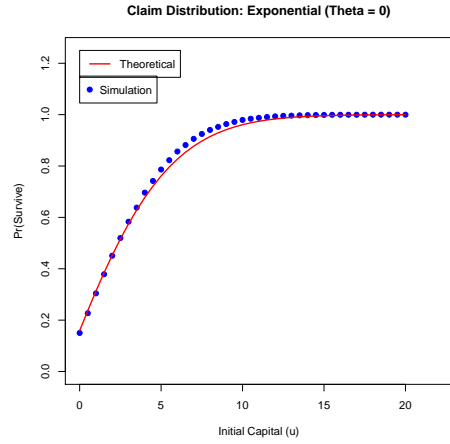
(d)  $\alpha = 1, \theta = 0.2, r = 0.05$



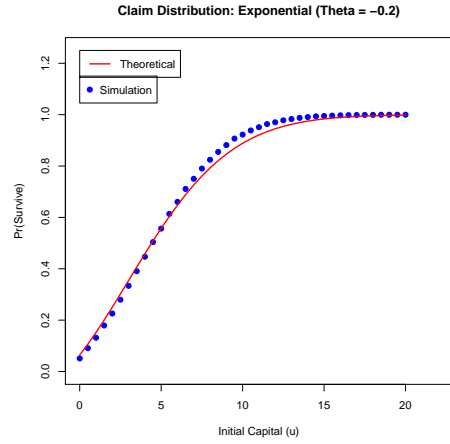
(e)  $\alpha = 1, \theta = 0.1, r = 0.05$

Figure 5:  
Survival probability under exponential claims distribution with investment under positive  $\theta$

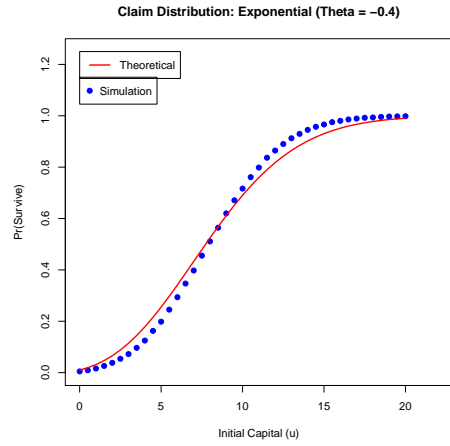




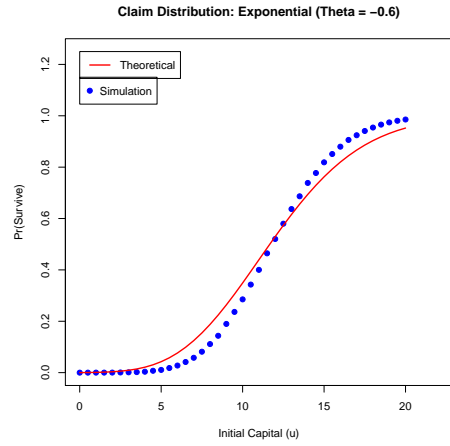
(a)  $\alpha = 1, \theta = 0, r = 0.05$



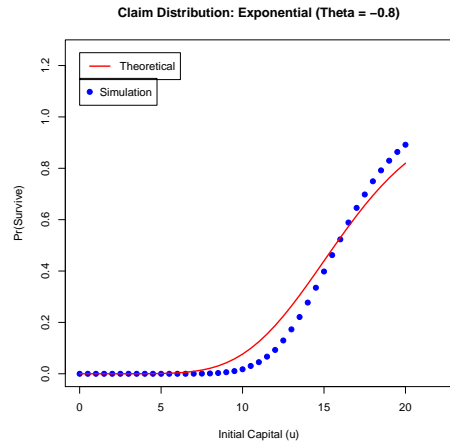
(b)  $\alpha = 1, \theta = -0.2, r = 0.05$



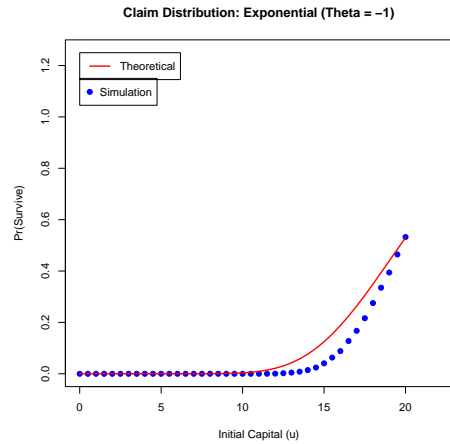
(c)  $\alpha = 1, \theta = -0.4, r = 0.05$



(d)  $\alpha = 1, \theta = -0.6, r = 0.05$



(e)  $\alpha = 1, \theta = -0.8, r = 0.05$

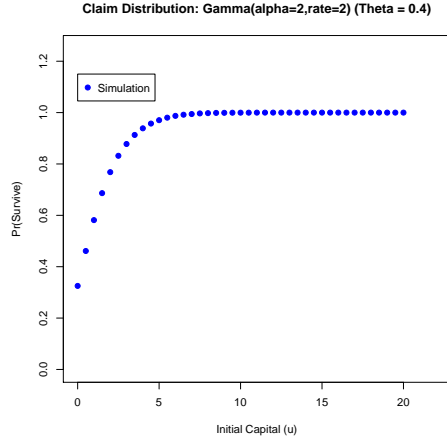


(f)  $\alpha = 1, \theta = -1, r = 0.05$

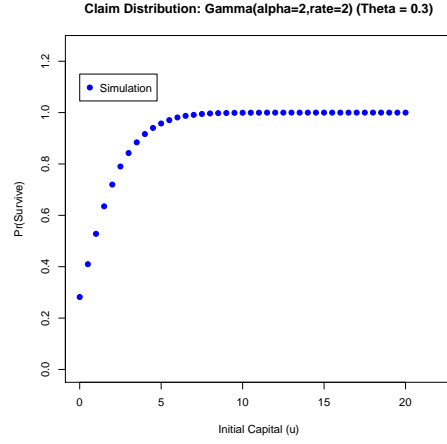
Figure 6:  
Survival probability under exponential claims distribution with investment under negative  $\theta$

### 3.2 Erlang Claims Distribution Simulation

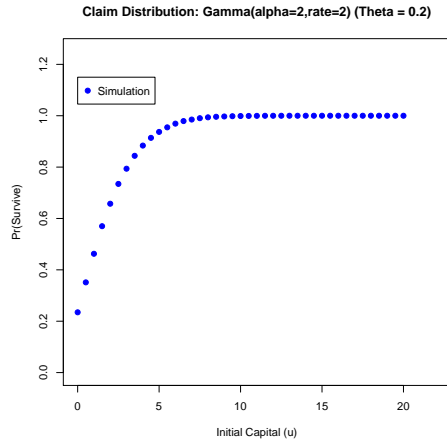
If the claim amount follows an Erlang distribution, the survival probability is similar to the exponential distribution case. Using  $c = (1 + \theta) \frac{2\lambda}{\beta}$ , the simulations are shown in Figure (7).



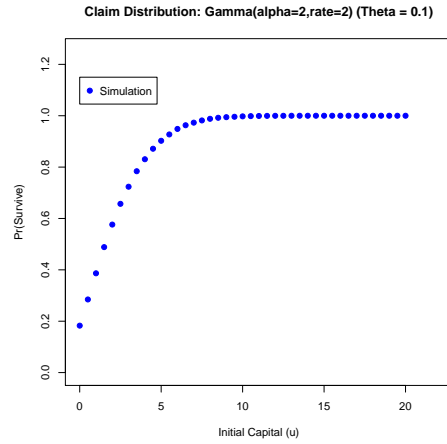
(a)  $\alpha = 1, \theta = 0.4, r = 0.05$



(b)  $\alpha = 1, \theta = 0.3, r = 0.05$

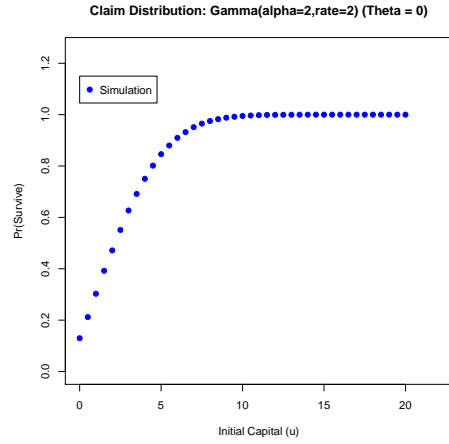


(c)  $\alpha = 1, \theta = 0.2, r = 0.05$

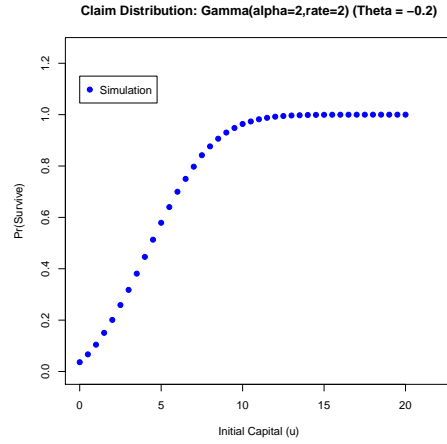


(d)  $\alpha = 1, \theta = 0.1, r = 0.05$

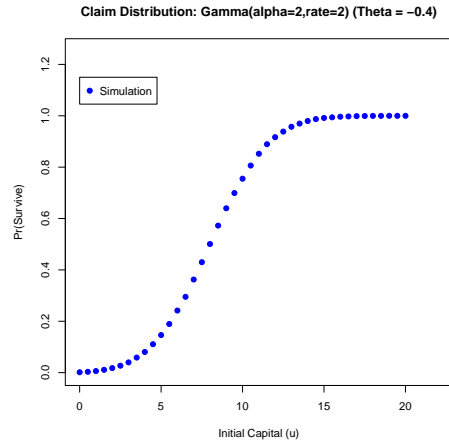
Figure 7:  
Survival probability under Erlang claims distribution with investment under positive  $\theta$



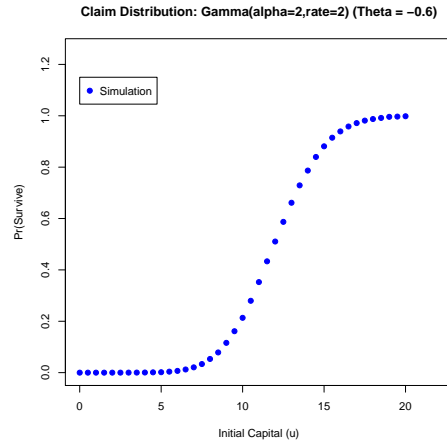
(a)  $\alpha = 1, \theta = 0.001, r = 0.05$



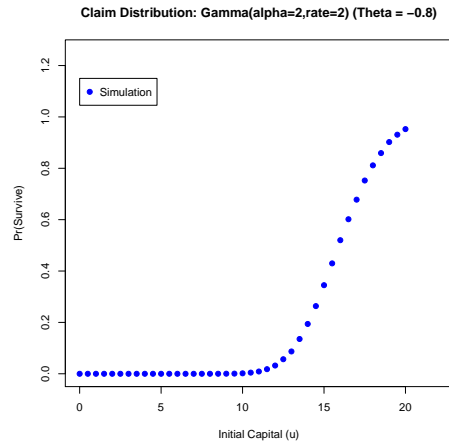
(b)  $\alpha = 1, \theta = -0.2, r = 0.05$



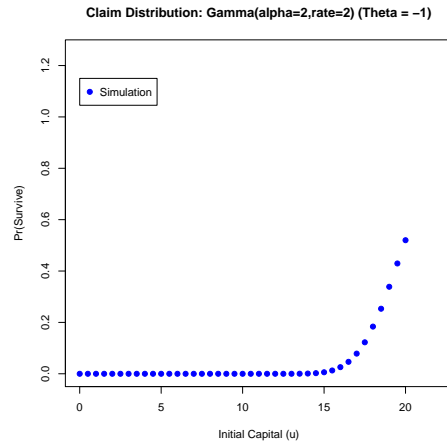
(c)  $\alpha = 1, \theta = -0.4, r = 0.05$



(d)  $\alpha = 1, \theta = -0.6, r = 0.05$



(e)  $\alpha = 1, \theta = -0.8, r = 0.05$



(f)  $\alpha = 1, \theta = -1, r = 0.05$

Figure 8:  
Survival probability under Erlang claims distribution with investment under negative  $\theta$

## 4 Conclusion

In the simplest form of Cramer-Lundberg model, we can conclude that if the claim distribution has a probability distribution function  $f(x)$  (p.d.f.) such that  $\mathcal{L}\{f(u)\} = \Omega(s)$  exists and if we can perform inverse transformation on (11), then we can find the analytical solution for  $\phi(u) = \mathcal{L}^{-1}\{\Phi(s)\}$ .

To attain the goal such that survival probability given  $u$ , i.e.  $\phi(u)$ , is 1 as  $u$  tends to  $\infty$  (infinite initial capital guarantees survival of the surplus process as time  $t$  goes to  $\infty$ ), the net profit condition  $c > \lambda\mu$  is the utmost critical, where  $c$  is the premium rate,  $\lambda$  is the mean of claims frequency per unit time and  $\mu$  is the mean of the claim size.

Furthermore, if we implement the constant investment income as a feature to the classic Cramer-Lundberg model, we see that the same strategy does not work to solve for  $\phi(u)$  because Laplace Transformation is not applicable on (19). Instead, we convert (19) to an ODE and solve it. We can see that the net profit condition is no longer needed to guarantee  $\phi(u)$  to be 1 as  $u$  tends to  $\infty$ . It is because as long as the initial capital is large enough, the investment income can compensate the premium shortfall.

Large initial capital can also interpret as the single premium receivable at  $t = 0$ . Therefore, using appropriate assumptions on the claims distribution and the interest rate  $r$ , the Cramer-Lundberg could be used as a reference to price single premium products.

## 5 References

- [1] S Asmussen, H Albrecher. (2010). *Ruin Probabilities*. Sweden: World Scientific Publishing Co. Pte. Ltd.
- [2] Andersen, E. Sparre. *On the collective theory of risk in case of contagion between claims*. Transactions of the XVth International Congress of Actuaries. Vol. 2. No. 6. 1957.
- [3] *R*, version 3.5.2. (2018-12-20).
- [4] *Wolfram Mathematica 10 (Student Edition)*, version 10.4.1.0. (2016-04-11).

## 6 Appendix

Written by Professor Charles Tier, equation (31) can be found by Mathematica[4]:

```
ode = y'''[x] + (b + (r - l) / (c + r x)) y'[x] == 0
(b + (-l + r) / (c + r x)) y'[x] + y''[x] == 0

soln = DSolve[ode, y[x], x]

{{y[x] -> C[2] - (c + r x)^{lr} (b/(c + r x))^{-1/r} C[1] Gamma[1/r, b/(c + r x)] / r}}
```

$$y[x] = \text{PowerExpand}[\text{soln}[[1, 1, 2]]]$$

$$C[2] - b^{-\frac{1}{r}} r^{-1+\frac{1}{r}} C[1] \text{Gamma}\left[\frac{1}{r}, \frac{b(c+rx)}{r}\right]$$

```
$Assumptions = {r > 0, b > 0, c > 0, l > 0}
{r > 0, b > 0, c > 0, l > 0}

ter = y[x] /. {C[1] -> c1, C[2] -> c2}
c2 - b^{-1/r} c1 r^{-1+1/r} Gamma[1/r, b(c+rx)/r]
```

$$tt = \text{Solve}[\text{Limit}[\text{ter}, x \rightarrow \text{Infinity}] == 1, c2]$$

$$\{\{c2 \rightarrow 1\}\}$$

```
yy[x_] = ter /. tt
{1 - b^{-1/r} c1 r^{-1+1/r} Gamma[1/r, b(c+rx)/r]}
```

$$yy[x]$$

$$\{1 - b^{-\frac{1}{r}} c1 r^{-1+\frac{1}{r}} \text{Gamma}\left[\frac{1}{r}, \frac{b(c+rx)}{r}\right]\}$$

```
bc = c*(D[yy[x], x] /. x -> 0) - l*yy[0]
{b^{1-1/r} c c1 e^{-b/c} (b/c)^{-1+1/r} r^{-1+1/r} - l(1 - b^{-1/r} c1 r^{-1+1/r} Gamma[1/r, b/c])}
```

$$ttt = \text{Solve}[bc == 0, c1]$$

$$\left\{\left\{c1 \rightarrow \frac{b^{lr} e^{\frac{bc}{r}} l r^{1-\frac{1}{r}}}{\left(\frac{bc}{r}\right)^{lr} r + e^{\frac{bc}{r}} l \text{Gamma}\left[\frac{1}{r}, \frac{bc}{r}\right]}\right\}\right\}$$

```
yyy[x_] = yy[x] /. ttt
{{1 - (e^{bc/r} l Gamma[1/r, b(c+rx)/r]) / ((bc/r)^{lr} r + e^{bc/r} l Gamma[1/r, bc/r])}}
```

$$yyy[0]$$

$$\left\{\left\{1 - \frac{e^{\frac{bc}{r}} l \text{Gamma}\left[\frac{1}{r}, \frac{bc}{r}\right]}{\left(\frac{bc}{r}\right)^{lr} r + e^{\frac{bc}{r}} l \text{Gamma}\left[\frac{1}{r}, \frac{bc}{r}\right]}\right\}\right\}$$

The R code below simulates exponential claims without investment.

```
# Theta loop
for (th in seq(0,0.5,0.1)){

  graph = c();

  theta <- round(th,1);
  # loading such that Net Profit Condition holds
  lab <- 1;
  # lambda of the poisson distribution
  EX <- 1;
  # expected value of claims distribution
  alpha <- 1/EX;
  # parameter of the exponential distribution
  r <- 0;
  # constant investment income
  n <- 400+4000*(1-round(th/(th+0.000001),0));
  # each simulation generates 400 claims except 4400 claims for u = 0
  nSim <- 10000;
  # number of simulations, each simulation is a U process
  c. <- (1+theta)*lab*EX
  # premium rate with loading

  matrix = c();

  # seed loop
  for (seed in seq(1,10,1)){

    set.seed(seed)
    tobeplot = c();
    case = 1;

    # capital loop
    for (y in seq(0, 20, 0.5)){
      u <- y
      # run different initial capital u
      N <- rep(Inf, nSim)

      # simulation loop
      for (k in 1:nSim){
        Wi <- rexp(n)/lab;
        # /lab is for standardize
        # Wi is a vector in R^n

        Xi <- rexp(n, rate=1/EX)
        # Xi storing the generated n claims
        ## severity has mean EX=1
```

```

Ui <- rep(0,n)
Ui[1] = u;
for (j in c(2:n)){
  Ui[j] = Ui[j-1]*(1+r) + Wi[j-1]*c. - Xi[j-1];
}
# Ui storing capital amount deposite whether it hits zero at
# some claim time Ti

ruin <- !all(Ui>=0)
# ruin = 0 or 1 depends on whether ruin occurred in the process

if (ruin) N[k] <- min(which(Ui<0))
# N is a vector storing the index of the first negative Ui in
# each simulation
# Value is "inf" if no negative Ui
}

N <- N[N<Inf];
# A vector which only stores the index of ruin time of each
# simulation

# show progress during the run
print("-")
print("-")
print("-")
print("-")
print(paste("Theta_=",th))
print(paste("Seed_=",seed))
print(paste("u_=",u))
print(paste("Ruined_=",length(N),"/",nSim))
print("-----")

tobeplot[case] = (nSim - length(N))/nSim
case = case + 1
}
matrix = cbind(matrix,tobeplot);
}

average = c();
for (j in seq(0, 41, 1)){
  average[j] = mean(matrix[j,]);
}

graph = cbind(graph,average);

```

```
#####
##          Plots comparison          ##
#####

# Export plots as pdf
pdf(paste0("C:/Users/Woodrow/Documents/MATH_594/R_Codes/No_Investment
/exponential/plots_theta_",gsub("\\.", "", toString(theta)), ".pdf"))

# Simulation plot
plot(seq(0, 20, 0.5), graph[,1],
      xlim=c(0, 22), ylim=c(0, 1.25), col="blue",
      xlab="Initial_Capital_(u)", ylab="Pr(Survive)",
      main=paste0("Claim_Distribution:_Exponential_(Theta=_", theta, "
      ")),
      type="p", pch=19)
# Theoretical plot
points(seq(0, 20, 0.5), 1-lab/(alpha*c.)*exp(-seq(0, 20, 0.5)*(alpha-
lab/c.)), col="red", type="l", lwd=2)
legend(0, 1.15, c("Simulation"), col=c("blue"), pch=19);
legend(0, 1.25, c("Theoretical"), col=c("red"), lwd=2, lty=c(1));

dev.off()

#rm(list=ls())
}
```



The R code below simulates hyper-exponential claims without investment.

```
# Theta loop
for (th in seq(0,0.5,0.1)){

  graph = c();

  theta    <- round(th,1)+ifelse(th==0,0.001,0);
  # loading such that Net Profit Condition holds
  lab      <- 1;
  # lambda of the poisson distribution
  p.       <- 0.2;
  alpha    <- 0.7;
  bet      <- 1.12;
  # parameter of the hyper-exponential distribution such that EX=1
  EX       <- p./alpha+(1-p.)/bet;
  # expected value of claims distribution EX=1
  r        <- 0;
  # constant investment income
  n        <- 400+4000*(1-round(th/(th+0.000001),0));
  # each simulation generates 400 claims except 4400 claims for u = 0
  nSim     <- 10000;
  # number of simulations, each simulation is a U process
  c.       <- (1+theta)*lab*EX
  # premium rate with loading

  matrix = c();

  # seed loop
  for (seed in seq(1,10,1)){

    set.seed(seed)
    tobeplot = c();
    case = 1;

    # capital loop
    for (y in seq(0, 20, 0.5)){
      u <- y
      # run different initial capital u
      N <- rep(Inf, nSim)

      # simulation loop
      for (k in 1:nSim){
        Wi <- rexp(n)/lab;
        # /lab is for standardize
        # Wi is a vector in R^n
        bern = rbinom(n,1,p.);
        Xi <- bern*rexp(n,rate=alpha)+(1-bern)*rexp(n,rate=bet);
```

```

# Xi storing the generated n claims
## severity has mean EX=1
Ui <- rep(0,n)
Ui[1] = u;
for (j in c(2:n)){
  Ui[j] = Ui[j-1]*(1+r) + Wi[j-1]*c. - Xi[j-1];
}
# Ui storing capital amount deposite whether it hits zero at
  some claim time Ti

ruin <- !all(Ui>=0)
# ruin = 0 or 1 depends on whether ruin occurred in the process

if (ruin) N[k] <- min(which(Ui<0))
# N is a vector storing the index of the first negative Ui in
  each simulation
# Value is "inf" if no negative Ui
}

N <- N[N<Inf];
# A vector which only stores the index of ruin time of each
  simulation

# show progress during the run
print("-")
print("-")
print("-")
print("-")
print(paste("Theta_=",th))
print(paste("Seed_=",seed))
print(paste("u_=",u))
print(paste("Ruined_=",length(N),"/",nSim))
print("-----")

tobeplot[case] = (nSim - length(N))/nSim
case = case + 1
}
matrix = cbind(matrix,tobeplot);
}

average = c();
for (j in seq(0, 41, 1)){
  average[j] = mean(matrix[j,]);
}

graph = cbind(graph,average);

```

```
#####
##              Plots comparison              ##
#####

# Export plots as pdf
pdf(paste0("C:/Users/Woodrow/Documents/MATH_594/R_Codes/No_Investment
        /hyperexpo/plots_theta_",gsub("\\.", "", toString(theta)), ".pdf"))

# Simulation plot
plot(seq(0, 20, 0.5), graph[,1],
      xlim=c(0,22), ylim=c(0,1.25), col="blue",
      xlab="Initial_Capital_(u)", ylab="Pr(Survive)",
      main=paste0("Claim_Distribution:_Hyper-exponential_(Theta_=",
        round(theta,1), ")"),
      type="p", pch=19)
# Theoretical plot
tri = (alpha+bet-lab/c.)^2-4*alpha*bet*(1-lab*EX/c.);
r1 = (-(alpha+bet-lab/c.)+tri^0.5)/2;
r2 = (-(alpha+bet-lab/c.)-tri^0.5)/2;
A1 = 1;
A2 = (1-lab*EX/c.)*(r1+alpha)*(r1+bet)/(r1*(tri^0.5));
A3 = (1-lab*EX/c.)*(r2+alpha)*(r2+bet)/(-r2*(tri^0.5));
points(seq(0, 20, 0.5), A1+A2*exp(r1*seq(0, 20, 0.5))+A3*exp(r2*seq(0,
    20, 0.5)), col="red", type="l", lwd=2)
legend(0, 1.15, c("Simulation"), col=c("blue"), pch=19);
legend(0, 1.25, c("Theoretical"), col=c("red"), lwd=2, lty=c(1));

dev.off()

#rm(list=ls())
}
```

The R code below simulates Erlang claims without investment.

```
# Theta loop
for (th in seq(0,0.5,0.1)){

  graph = c();

  theta <- round(th,1)+ifelse(th==0,0.001,0);
  # loading such that Net Profit Condition holds, use 0.001 to approx 0
  # and show better plot
  lab <- 1;
  # lambda of the poisson distribution
  EX <- 1;
  # expected value of claims distribution
  r <- 0;
  # investment income
  n <- 400+4000*(1-round(th/(th+0.000001),0));
  # each simulation generates 400 claims except 4400 claims for u = 0
  nSim <- 10000;
  # number of simulations, each simulation is a U process
  c. <- (1+theta)*lab*EX
  # premium rate with loading
  alpha <- 2;
  # shape of gamma (k)
  rate <- alpha/EX;
  # rate (beta)

  matrix = c();

  # seed loop
  for (seed in seq(1,10,1)){

    set.seed(seed)
    tobeplot = c();
    case = 1;

    # capital loop
    for (y in seq(0, 20, 0.5)){
      u <- y
      # run different initial capital
      N <- rep(Inf, nSim)

      # simulation loop
      for (k in 1:nSim){
        Wi <- rexp(n)/lab;
        # /lab is for standardize
        # Wi is a vector in R^n
```

```

Xi <- rgamma(n, shape=alpha, scale=1/rate)
# Xi storing the generated n claims
## severity has mean EX=1

Ui <- rep(0,n)
Ui[1] = u;
for (j in c(2:n)){
  Ui[j] = Ui[j-1]*(1+r) + Wi[j-1]*c. - Xi[j-1];
}
# Ui storing capital amount deposite whether it hits zero at
  some claim time Ti

ruin <- !all(Ui>=0)
# ruin = 0 or 1 depends on whether ruin occurred in the process

if (ruin) N[k] <- min(which(Ui<0))
# N is a vector storing the index of the first negative Ui in
  each simulation
# Value is "inf" if no negative Ui
}

N <- N[N<Inf];
# A vector which only stores the index of ruin time of each
  simulation

# show progress during the run
print("-")
print("-")
print("-")
print("-")
print(paste("Theta_=",th))
print(paste("Seed_=",seed))
print(paste("u_=",u))
print(paste("Ruined_=",length(N),"/",nSim))
print("-----")

tobeplot[case] = (nSim - length(N))/nSim
case = case + 1
}
matrix = cbind(matrix,tobeplot);
}

average = c();
for (j in seq(0, 41, 1)){
  average[j] = mean(matrix[j,]);
}

```

```

graph = cbind(graph,average);

#####
### Plots comparison
#####

# Export plots as pdf
pdf(paste0("C:/Users/Woodrow/Documents/MATH_594/R_Codes/No_Investment
/gamma/plots_theta_",gsub("\\.", "", toString(theta)), ".pdf"))

# Simulation plot
plot(seq(0, 20, 0.5),graph[,1],
      xlim=c(0,22),ylim=c(0,1.25),col="blue",
      xlab="Initial_Capital_(u)",ylab="Pr(Survive)",
      main=paste0("Claim_Distribution:_Gamma(alpha=2,rate=2)_(Theta=_
",round(theta,1),")"),
      type="p",pch=19)
# Theoretical plot
delta = (2*rate-lab/c.)^2-4*rate*(rate-2*lab/c.)
r1 = (- (2*rate-lab/c.)+delta^0.5)/2
r2 = (- (2*rate-lab/c.)-delta^0.5)/2
A1 = 1
A2 = ((c.-2*lab/rate)*(r1+rate)^2)/(c.*r1*(r1-r2))
A3 = ((c.-2*lab/rate)*(r2+rate)^2)/(c.*r2*(r2-r1))
points(seq(0, 20, 0.5),A1+A2*exp(r1*seq(0, 20, 0.5))+A3*exp(r2*seq(0,
20, 0.5)),col="red",type="l",lwd=2)
legend(0, 1.15, c("Simulation"), col=c("blue"), pch=19);
legend(0, 1.25, c("Theoretical"), col=c("red"), lwd=2, lty=c(1));

dev.off()

#rm(list=ls())
}

```

The R code below simulates exponential claims with investment.

```
library(pracma)
# Theta loop
for (th in seq(-1,0.5,0.1)){

  graph = c();

  theta <- round(th,1);
  # loading such that Net Profit Condition holds
  lab <- 1;
  # lambda of the poisson distribution
  EX <- 1;
  # expected value of claims distribution
  alpha <- 1/EX;
  # parameter of the exponential distribution
  r <- 0.05;
  # constant investment income
  n <- 400+4000*(1-round(th/(th+0.000001),0));
  # each simulation generates 400 claims except 4400 claims for u = 0
  nSim <- 10000;
  # number of simulations, each simulation is a U process
  c. <- (1+theta)*lab*EX
  # premium rate with loading

  matrix = c();

  # seed loop
  for (seed in seq(1,10,1)){

    set.seed(seed)
    tobeplot = c();
    case = 1;

    # capital loop
    for (y in seq(0, 20, 0.5)){
      u <- y
      # run different initial capital u
      N <- rep(Inf, nSim)

      # simulation loop
      for (k in 1:nSim){
        Wi <- rexp(n)/lab;
        # /lab is for standardize
        # Wi is a vector in  $R^n$ 

        Xi <- rexp(n, rate=1/EX)
        # Xi storing the generated n claims
```

```

## severity has mean EX=1

Ui <- rep(0,n)
Ui[1] = u;
for (j in c(2:n)){
  Ui[j] = Ui[j-1]*(1+r) + Wi[j-1]*c. - Xi[j-1];
}
# Ui storing capital amount deposite whether it hits zero at
  some claim time Ti

ruin <- !all(Ui>=0)
# ruin = 0 or 1 depends on whether ruin occurred in the process

if (ruin) N[k] <- min(which(Ui<0))
# N is a vector storing the index of the first negative Ui in
  each simulation
# Value is "inf" if no negative Ui
}

N <- N[N<Inf];
# A vector which only stores the index of ruin time of each
  simulation

# show progress during the run
print("-")
print("-")
print("-")
print("-")
print(paste("Theta_=",th))
print(paste("Seed_=",seed))
print(paste("u_=",u))
print(paste("Ruined_=",length(N),"/",nSim))
print("-----")

tobeplot[case] = (nSim - length(N))/nSim
case = case + 1
}
matrix = cbind(matrix,tobeplot);
}

average = c();
for (j in seq(0, 41, 1)){
average[j] = mean(matrix[j,]);
}

graph = cbind(graph,average);

```



```
#####
##          Plots comparison          ##
#####

# Export plots as pdf
pdf(paste0("C:/Users/Woodrow/Documents/MATH_594/R_Codes/Investment/
    exponential/plots_theta_",gsub("\\.", "", toString(theta)), ".pdf"))

# Simulation plot
plot(seq(0, 20, 0.5), graph[,1],
     xlim=c(0,22), ylim=c(0,1.25), col="blue",
     xlab="Initial_Capital_(u)", ylab="Pr(Survive)",
     main=paste0("Claim_Distribution:_Exponential_(Theta=_", theta, "
    ")),
     type="p", pch=19)
# Theoretical plot
A1 <- lab*exp(alpha*c./r);
A2 <- r*(alpha*c./r)^(lab/r);
G1 <- rep(0, length(seq(0, 20, 0.5)));
for (i in c(1:length(G1))){
  G1[i] = incgam(alpha*c./r+alpha*seq(0, 20, 0.5)[i], lab/r);
}
points(seq(0, 20, 0.5), 1-(A1*G1)/(A2+A1*incgam(alpha*c./r, lab/r)), col
    ="red", type="l", lwd=2)
legend(0, 1.15, c("Simulation"), col=c("blue"), pch=19);
legend(0, 1.25, c("Theoretical"), col=c("red"), lwd=2, lty=c(1));

dev.off()

#rm(list=ls())
}
```

The R code below simulates Erlang claims with investment.

```
library(pracma)
# Theta loop
for (th in seq(-1,0.5,0.1)){

  graph = c();

  theta <- round(th,1)+ifelse(th==0,0.001,0);
  # loading such that Net Profit Condition holds
  lab <- 1;
  # lambda of the poisson distribution
  EX <- 1;
  # expected value of claims distribution
  r <- 0.05;
  # investment income
  n <- 400+4000*(1-round(th/(th+0.000001),0));
  # each simulation generates 400 claims except 4400 claims for u = 0
  nSim <- 10000;
  # number of simulations, each simulation is a U process
  c. <- (1+theta)*lab*EX
  # premium rate with loading
  alpha <- 2;
  # shape of gamma (k)
  rate <- alpha/EX;
  # rate (beta)

  matrix = c();

  # seed loop
  for (seed in seq(1,10,1)){

    set.seed(seed)
    tobeplot = c();
    case = 1;

    # capital loop
    for (y in seq(0, 20, 0.5)){
      u <- y
      # run different initial capital
      N <- rep(Inf, nSim)

      # simulation loop
      for (k in 1:nSim){
        Wi <- rexp(n)/lab;
        # /lab is for standardize
        # Wi is a vector in R^n
```

```

Xi <- rgamma(n, shape=alpha, scale=1/rate)
# Xi storing the generated n claims
## severity has mean EX=1

Ui <- rep(0,n)
Ui[1] = u;
for (j in c(2:n)){
  Ui[j] = Ui[j-1]*(1+r) + Wi[j-1]*c. - Xi[j-1];
}
# Ui storing capital amount deposite whether it hits zero at
  some claim time Ti

ruin <- !all(Ui>=0)
# ruin = 0 or 1 depends on whether ruin occurred in the process

if (ruin) N[k] <- min(which(Ui<0))
# N is a vector storing the index of the first negative Ui in
  each simulation
# Value is "inf" if no negative Ui
}

N <- N[N<Inf];
# A vector which only stores the index of ruin time of each
  simulation

# show progress during the run
print("-")
print("-")
print("-")
print("-")
print(paste("Theta_=",th))
print(paste("Seed_=",seed))
print(paste("u_=",u))
print(paste("Ruined_=",length(N),"/",nSim))
print("-----")

tobeplot[case] = (nSim - length(N))/nSim
case = case + 1
}
matrix = cbind(matrix,tobeplot);
}

average = c();
for (j in seq(0, 41, 1)){
  average[j] = mean(matrix[j,]);
}

```

```

graph = cbind(graph,average);

#####
### Plots comparison
#####

# Export plots as pdf
pdf(paste0("C:/Users/Woodrow/Documents/MATH_594/R_Codes/Investment/
gamma/plots_theta_",gsub("\\.", "", toString(theta)), ".pdf"))

# Simulation plot
plot(seq(0, 20, 0.5), graph[,1],
      xlim=c(0, 22), ylim=c(0, 1.25), col="blue",
      xlab="Initial_Capital_(u)", ylab="Pr(Survive)",
      main=paste0("Claim_Distribution:_Gamma(alpha=2,rate=2)_(Theta=_
", round(theta, 1), ")"),
      type="p", pch=19)
legend(0, 1.15, c("Simulation"), col=c("blue"), pch=19);

dev.off()

#rm(list=ls())
}

```