REFERENCES

Blitzstein, J. K., & Hwang, J. (2015). *Introduction to probability*. California: CRC Press.

Burden, R. L., & Faires, J. D. (2010). *Numerical analysis*. Brooks/Cole Cengage Learning.

Chen, Y., Yi, C., Xie, X., Hou, M., & Cheng, Y. (2020). Solution of ruin probability for continuous time model based on block trigonometric exponential neural network. *Symmetry*, *12*(6), 876. https://doi.org/10.3390/sym12060876

Cocagne, A. (2017). Applications of Laplace transform. *Senior Honors Theses and Projects*. 541. https://commons.emich.edu/honors/541

Cocco, D., & Giona, M. (2021). Generalized counting processes in a stochastic environment. *Mathematics*, *9*(20), 2573. https://doi.org/10.3390/math9202573

Das, J., & Nath, D. C. (2019). Weighted quantile regression theory and its application Weibull distribution as an actuarial risk model: Computation of its probability of ultimate ruin and the moments of the time to ruin, deficit at ruin and surplus prior to ruin. *Journal of Data Science*, *17*(1), 161–194. https://doi.org/10.6339/jds.201901_17(1).0008

Das, K. K., & Roy, T. D. (2011). On Some Length-Biased Weighted Weibull Distribution. *Advances in Applied Science Research*, *2*(5), 465–475.

Dickson, D. C. M. (2017). *Insurance risk and ruin* (2nd ed.). Cambridge University Press.

Febrianti, R. (2022). Calculation Of the Ultimate Ruin Probability Using Laplace Transform. Thesis. Faculty of Business. President University: Cikarang.

Grimmett, G., & Welsh, D. (2014). *Probability: An Introduction* (2nd ed.). Oxford University Press.

James, G., Burley, D., Clements, D., Dyke, P., Searl, J., & Wright, J. (2015). *Modern Engineering Mathematics* (5th ed.). Pearson Education Limited.

Klugman, S. A., Panjer, H. H., & Willmot, G. E. (2019). *Loss models: From data to decisions* (5th ed.). JohnWiley & Sons, Inc.

Kabosu, P. Y. L. (2021). Perhitungan Peluang Kebangkrutan Dengan Asumsi Besar Klaim Berdistribusi Eksponensial dan Berdistribusi Weibull. Skripsi. Universitas Katolik Parahyangan.

Lawler, G. F. (2006). *Introduction to Stochastic Processes* (2nd ed.). Taylor and Francis/CRC Press.

Loke, S.-H., & Thomann, E. (2018). Numerical ruin probability in the dual risk model with risk-free investments. *Risks*, *6*(4), 110. https://doi.org/10.3390/risks6040110

Maulina, H. (2022). Lower and Upper Bound Method for Approximating the Probability of Ruin. Thesis. Faculty of Business. President University: Cikarang.

Otaya, L. (2016). *Distribusi Probabilitas Weibull Dan Aplikasinya (Pada Persoalan Keandalan (Reliability) dan Analisis Rawatan (Mantainability)*. TADBIR: Jurnal Manajemen Pendidikan Islam.

Pishro-Nik, H. (2014). *Introduction to Probability, Statistics, and Random Processes*. Kappa Research, LLC.

Septia, W. (2019). Perbandingan Pemodelan Kecepatan Angin Untuk Hijriyah dan Masehi Dengan Menggunakan Distribusi *Weibull* dan *Gamma*. Tugas Akhir. Universitas Islam Negeri Sultan Syarif Kasim Riau.

Varberg, D. E., Purcell, E. J., & Rigdon, S. E. (2014). *Calculus early transcendentals* (1st ed.). Pearson.

Walpole, R. E. et al. (2016). *Probability & Statistics for Engineers & Scientists* (9th ed.). Pearson.

Zhou, T., Liu, X., Hou, M., & Liu, C. (2018). Numerical solution for ruin probability of continuous time model based on neural network algorithm. *Neurocomputing*, *331*, 67–76. https://doi.org/10.1016/j.neucom.2018.08.020

APPENDIX

Appendix 1: Python Code for Survival vs Ruin Probability Graph for Case 1

```
import numpy as np

from math import *

import matplotlib.pyplot as plt

%matplotlib inline


plt.figure(figsize=(5,3))


beta = 2 # beta>0

theta = 0.5 # theta>0

u = np.arange(0,20,0.01)

phi = 1-((np.exp(-u*theta/((1+theta)*beta)))/(1+theta))

psi = 1-phi


plt.plot(u, phi, label='$\phi(u)$')

plt.plot(u, psi, label='$\psi(u)$')

plt.grid(True)

plt.xlabel(r'$u$')

plt.ylabel(r'$Probability$')

plt.title(r"Survival vs Ruin Probability with $\theta =0.5, \beta =$
{}".format(beta))
```

```
plt.legend()


plt.show()
np.set_printoptions(threshold=np.inf)
phi
```

Appendix 2: Python Code for Case 2

```
import numpy as np
from math import *
import matplotlib.pyplot as plt
%matplotlib inline
```

The graph of PDF:

```
plt.rcParams['figure.dpi'] = 75
alpha = 2
beta = 3
x = np.arange(0,10,0.01)
y = (alpha/(beta**alpha))*x**(alpha-1)*np.exp(-(x/beta)**alpha)
plt.plot(x,y)
plt.grid(True)
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.title(r"The graph of Weibull p.d.f with $\alpha = 2, \beta = 3$")
```

Euler's method & the looping:

```
m1 = beta*gamma(1+(1/alpha))

cc = 1/3 # lambda/c

Du = 0.02

L = 50

N = int(L/Du)

u = np.arange(0,L+Du,Du)

x = np.arange(0,L+Du,Du)

phin = np.zeros(N+1)

phin[0] = 1-cc*m1 # initial condition

integ = np.zeros(N+1)

f = lambda x: (alpha/(beta**alpha))*x**(alpha-1)*np.exp(-(x/beta)**alpha)

phin[1] = phin[0]+Du*cc*phin[0]


k = 1

while k < N:

   y = 0

   for i in range(0,k+1):

      y = y + phin[i]*f(u[k]-x[i])+phin[i+1]*f(u[k]-x[i+1])

   integ[k] = y


   phin[k+1] = phin[k]+(Du*cc*(phin[k]-Du*0.5*integ[k]))

   k = k+1
```

```
psi = 1-phin
```

The survival vs ruin probability graph:

```
plt.rcParams['savefig.dpi'] = 100
plt.plot(u, phin, label='$\phi(u)$')
plt.plot(u, psi, label='$\psi(u)$')
plt.grid(True)
plt.xlabel(r'$u$')
plt.ylabel(r'$Probability$')
plt.savefig('Weibull32-1.eps', format='eps')
plt.title(r"Survival vs Ruin Probability")
plt.legend()
plt.show()
```