

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу**

До захисту допущено

Завідувач кафедри

Оксана ТИМОЩУК

«__» _____ 20__ р.

Дипломна робота

**на здобуття ступеня бакалавра
за освітньо-професійною програмою «Системний аналіз і управління»
спеціальності 124 «Системний аналіз»
на тему: «Оцінка ризику страхової компанії на основі моделі Крамера -
Лундберга»**

Виконав:

Студент IV курсу, групи КА-71

Крохальов Іван Данилович

Керівник:

доц., канд. фіз.-мат. наук Каніовська І.Ю.

Консультант з економічного розділу:

доцент, к.е.н. Рощина Н.В.

Консультант з нормоконтролю:

доцент, к.т.н., доцент Коваленко А.Є.

Рецензент:

доцент, к. ф.-м. н. Буценко Ю.П.

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2021 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 124 «Системний аналіз»

Освітньо-професійна програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.Л. Тимощук

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

Крохальова Івана Даниловича

1. Тема роботи «Оцінка ризику страхової компанії на основі моделі Крамера - Лундберга», керівник роботи доцент, канд. фіз.-мат. наук Каніовська І.Ю., затверджені наказом по університету від « 26 » травня _____ 2021 р. № 1366-с

2. Термін подання студентом роботи 7 червня 2021р.

3. Вихідні дані до роботи: модель Крамера – Лундберга, відомості про можливі розподіли виплат.

4. Зміст роботи: дослідження процесу страхового ризику, програмна реалізація модуля для простих актуарних розрахунків

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Рощина Н.В. , доцент		

7. Дата видачі завдання: 01.03.2020 _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми ДР	02.03.2021-10.03.2021	виконано
2	Ознайомлення зі структурою БДР згідно з Положенням про державну атестацію студентів НТУУ «КПІ ім. І. Сікорського»	11.03.2021-24.03.2021	виконано
3	Ознайомлення з ДСТУ 3008-95 та стандарти ЄСПД	25.03.2021-31.03.2021	виконано
4	Проведення дослідження за темою БДР під керівництвом керівника	01.04.2021-15.04.2021	виконано
5	Завершення роботи над першим варіантом частини БДР	16.04.2021-29.04.2021	виконано
6	Проведення роботи над експериментальною частиною БДР	05.05.2021-19.05.2021	виконано
7	Проведення роботи над програмним продуктом	30.04.2021-19.05.2021	виконано
8	Оформлення БДР та аналіз отриманих результатів	19.05.2021-26.05.2021	виконано

Студент

І.Д. Крохальов

Керівник

І.Ю. Каніовська

РЕФЕРАТ

Дипломна робота: 87 с. , 12 рис., 17 табл., 2 додат., 19 джерел.

МОДЕЛЬ КРАМЕРА-ЛУНДБЕРГА, ЙМОВІРНІСТЬ БАНКРУТСТВА,
МЕТОДИ АПРОКСИМАЦІЇ, МЕТОДИ МОНТЕ-КАРЛО, МОДЕЛЮВАННЯ.

Об'єкт дослідження: модель Крамера-Лундберга, методи обчислення ймовірності банкрутства.

Мета дослідження: проаналізувати існуючі методи апроксимації ймовірності банкрутства, реалізувати власний програмний модуль для роботи із процесом ризику.

Використані моделі: у програмній реалізації було використано бібліотеки numpy, scipy для генерації випадкових величин, matplotlib для візуалізації та tqdm для відслідковування прогресу циклу.

Отриманні результати: побудовано програмний модуль, на основі якого можна симулювати траєкторії процесу ризику або обчислювати його теоретичну апроксимацію ймовірності банкрутства. Порівняно поведінку процесу при різних розподілах виплат та інших початкових умовах.

В рамках подальшого дослідження пропонується розширити модуль на клас моделей Спарре-Андерсона, на моделі із броунівським рухом. Також, можна попрацювати над швидкодією – симуляції Монте-Карло можна виконувати паралельно, а не послідовно, що може зекономити час виконання програми.

ABSTRACT

Thesis work: 87 pp., 12 fig., 17 tabl., 2 app., 19 sources.

CRAMER-LUNDBERGH MODEL, RUIN PROBABILITY, APPROXIMATION METHODS, MONTE-CARLO METHOD, MODELLING.

Object of study: Cramer-Lundbergh model, ruin probability approximations.

Purpose: to analyze existing ruin probability approximations, to develop own module for actuarial calculations.

Used models: in the software implementation numpy and scipy are used for random variable generation; matplotlib is used for visualization, tqdm – for seeing progress of each loop.

Results: a Python module “tools” built. It includes RiskProcess class, and some several methods for calculation ruin probability from it. Comparison of methods made, using different initial conditions and distributions for claims.

Further research proposes to expand model for Sparre – Anderson models, for models that include Brownian motion. Also, it’s possible to improve working time of program – by using multiprocessing for Monte – Carlo methods.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1	11
ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Коротка історія теорії ризику	11
1.2 Напрями можливих узагальнень процесу ризику	12
1.3 Специфікація дипломної роботи	13
1.4 Висновки до розділу 1:	14
РОЗДІЛ 2	15
ТЕОРЕТИЧНІ ВІДОМОСТІ	15
2.1 Однорідний процес Пуассона	15
2.2 Складний процес Пуассона	16
2.3 Процес страхового ризику	17
2.4 Як обрати функцію прибутку $P(t)$?	19
2.5 Огляд розподілів виплат	20
2.6 Висновки до розділу 2	22
РОЗДІЛ 3	23
ЙМОВІРНІСТЬ БАНКРУТСТВА ТА МЕТОДИ ЇЇ РОЗРАХУНКУ	23
3.1 Модель Крамера-Лундберга	23
3.3 Рівняння для ймовірності банкрутства.	27
3.4 Апроксимація Крамера – Лундберга (легкі виплати)	28
3.5 Дифузійна апроксимація.	29
3.6 Прямий метод Монте – Карло	30
3.7 Метод Монте-Карло на основі формули згортки Бескмана	31
3.8 Висновки до розділу 3:	33
РОЗДІЛ 4	35
МОДЕЛЮВАННЯ	35
4.1 Умова Net-Profit Condition (умова чистого прибутку)	35
4.2 Дослідження процесу із експоненційним розподілом виплат	37
4.3 Дослідження випадку пуассонівських виплат.	39
4.4 Дослідження випадку із гамма-розподіленими виплатами.	40
4.5 Дослідження процесу із Парето-розподіленими виплатами.	42
4.6 Дослідження процесу із Вейбулл-розподіленими виплатами.	45

4.7 Висновки до розділу 4:	47
РОЗДІЛ 5	48
ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ	48
5.1 Постановка задачі проектування	48
5.2 Обґрунтування функцій програмного продукту	49
5.3 Обґрунтування системи параметрів ПП	52
5.4 Аналіз експертного оцінювання параметрів	54
5.5 Аналіз рівня якості варіантів реалізації функцій	58
5.6 Економічний аналіз варіантів розробки ПП	59
5.7 Вибір кращого варіанту ПП техніко-економічного рівня	65
5.8 Висновки до розділу 5	66
ВИСНОВКИ ДО РОБОТИ	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	68
Додаток А Лістинг програмного модулю	70
Додаток Б Презентація	80

ВСТУП

Об'єкт розгляду дипломної роботи – теорія руїн. Це розділ страхової математики, що вивчає подію банкрутства страхової компанії. Зокрема, відповідає на такі питання:

- як формалізувати капітал страхової компанії?
- за яких умов страхова компанія зазнає банкрутства? Із якою ймовірністю це може відбутися?
- якщо банкрутство відбудеться, то який розподіл часу до руїни?

Початком розвитку теорії руїн послужила математична модель капіталу страхової компанії. Вперше модель була представлена Філіпом Лундбергом (1903). Пропонується випадковий процес $U(t)$, що описує капітал страхової компанії у момент часу t :

$$U(t) = u + ct - \sum_{i=1}^{N(t)} X_i$$

де u – початковий капітал, ct – внески клієнтів, $N(t)$ – потік нещасних випадків (пуассонівський однорідний процес), $X_1, X_2, \dots, X_n$ – випадкові величини, що описують виплати клієнтам при нещасних випадках. На їх розподіл накладені деякі припущення – величини незалежні в сукупності, та від $N(t)$, однаково розподілені, мають математичне сподівання та дисперсію (більш детальний опис запропонованої моделі можна знайти в монографіях [1], [3])

Теорія руїн досліджує випадкову подію банкрутства, що має вигляд:

$$[\exists t > 0 : U(t) < 0]$$

Для ймовірності банкрутства розроблено ряд оцінок, нерівностей, теорем, чисельних методів, досліджується асимптотична поведінка.

Добре відомо, що для стандартного процесу Пуассона проміжки часу між подіями розподілені експоненційно, а так як процес Пуассона має незалежні прирости, то і проміжки часу також будуть незалежними випадковими величинами.

Кажучи спрощено, процес Пуассона – це процес відновлення, який побудовано на основі послідовності незалежних, однаково розподілених експоненційних величин. Процеси відновлення можуть мати послідовність із іншим розподілом величин, наприклад рівномірний або геометричний. Результати Крамера та Лундберга в математичній теорії ризику частково узагальнюються на модель ризику із процесом відновлення.

Не будь-який розподіл виплат підходить для моделі Крамера - Лундберга. Теорія Крамера та Лундберга лишилась вірною для виплат із легким хвостом. Для виплат ввели класи розподілів: розподіли із легким хвостом та розподіли із важким хвостом. Для розподілів із важким хвостом деякі результати виявились невірними (зокрема, нерівність Лундберга, що пропонує оцінку зверху для ймовірності банкрутства).

На жаль, аналітично розв'язати рівняння для ймовірностей банкрутства можливо лише у випадку експоненційних виплат. У інших ситуаціях отримати бажану функцію $\psi(u)$, $\psi(u, T)$ не вдається.

Для знаходження ймовірностей банкрутства та часу до руїни розроблено багато аналітичних та чисельних методів. Всі вони різні за природою, деякі не вимагають майже жодних додаткових припущень (апроксимація центральною граничною теоремою), інші використовують дані про моменти величини виплат (дифузійна апроксимація, апроксимація Де-Вілдера та її узагальнення).

Перший розділ роботи присвячено історії проблеми, класичній моделі страхового ризику. В ньому буде коротко подано історію актуарної математики (а саме – математичної теорії ризику, теорії руїн), розглянуто модель Крамера-Лундберга. У другому розділі описано математичний апарат

для теорії ризику – властивості процесу Пуассона, розглянуто процес ризику в загальних термінах, коротко оглянуто класи розподілів виплат. У третьому розділі описано можливі апроксимації ймовірності банкрутства, суміжні із нею поняття (характеристичний коефіцієнт, нерівність Лундберга, інтегральне рівняння відновлення). У четвертому розділі наведено результати комп'ютерного моделювання – для основних розподілів, проаналізовано поведінку методів апроксимації при різних початкових умовах.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Коротка історія теорії ризику

Прориви у фінансовій та актуарній математиці відбулися майже одночасно. Луї Башельє у 1900 році вперше побудував модель ціни опціона на базі процесу броунівського руху. Фактично, він став першим вченим, що застосував складний математичний апарат (теорія ймовірностей, випадкові процеси) до економіки та теорії фінансів, а на базі броунівського руху розроблено чимало фінансових моделей, якими користуються до сих пір.

Відкриття Філіпа Лундберга у 1903 році було дещо схожим. Вчений зрозумів, що пуассонівський процес цілком природно вписується у концепції страхової математики (не пов'язаної із життям). На основі досліджень гомогенного пуассонівського процесу, Лундберг запропонував першу модель актуарних розрахунків (монографії [1], [3]):

$$U(t) = u + ct - \sum_{i=1}^{N(t)} X_i \quad (1.1)$$

де $U(t)$ – капітал компанії у момент часу t , u – початковий капітал компанії, ct – внески клієнтів ($c > 0$), $N(t)$ – потік нещасних випадків (представлений у моделі пуассонівським однорідним процесом), $X_1, X_2, X_3, \dots, X_n$ - випадкові величини, що описують покриття збитків клієнтам при нещасних випадках. Величини незалежні в сукупності, та від $N(t)$, однаково розподілені, мають математичне сподівання.

Вважається, що у страхової компанії є досить великий початковий капітал. Це вельми реалістична умова, навіть у законодавстві України є вимоги щодо статутного капіталу. Відповідно до ст. 1 Закону про страхування : “Мінімальний розмір статутного капіталу повинен становити суму, еквівалентну 1 млн євро (для страхової компанії, яка займатиметься страхуванням життя, – 10 млн євро) за валютним обмінним курсом валюти

України. При цьому статутний капітал має бути сформовано лише грошовими коштами.“

В 1929 році за ініціативою ряду шведських страхових компаній у Стокгольмському університеті була створена “Стокгольмська група” вчених, що займалися актуарною математикою. Її науковим керівником був Гарольд Крамер. У XX ст. був великий ряд результатів шведської групи в області страхової та фінансової математики.

1.2 Напрями можливих узагальнень процесу ризику

Класичну модель ризику можна розширювати, узагальнювати багатьма способами. Зокрема:

- 1) вартість контракту може змінюватися в залежності від результатів страхового бізнесу. Тобто, враховувати залежність процесів $P(t)$, $S(t)$ – процесів загального надходження та загальних витрат;
- 2) включати зовнішні фактори (інфляція, епідемії тощо). Прикладом такої моделі є модель броунівського руху, змішані моделі актуарної та фінансової математики (моделі страхового ризику, що включають в себе випадкові збурення у вигляді броунівського руху).
- 3) розглядати більш загальні процеси кількості виплат ($N(t)$). За своєю природою пуассонівський процес – це процес-лічильник потоку подій, час між якими розподілений за експоненційним законом. Існують моделі на базі процесів відновлення ([3], [13]), що дозволяють довільний розподіл проміжків часу між подіями, але здебільшого припущення про те, що проміжки часу між подіями однаково розподілені – зберігається.
- 4) проводити виплати не миттєво, а вводити затримки, відкладені платежі. Звичайно, в залежності від величини затримки часу, сума виплати може змінюватися. Така постановка моделі є привабливою для представника страхової компанії, оскільки не завжди виплати можуть виконуватися вчасно. Іноді більш оптимальним рішенням буде відсрочити момент виплати,

можливо пообіцявши клієнту за це нагороду. Моделі із затримками прийшли із сусідньої області (цифрова обробка сигналів), а виплати є аналогами електричних імпульсів, що згасають із часом.

1.3 Специфікація дипломної роботи

Об'єктом дослідження дипломної роботи є модель ризику Крамера Лундберга, зокрема класичні результати Лундберга (1930 – 1950) та існуючі апроксимації ймовірності банкрутства (основний інтерес будуть представляти методи Монте-Карло та їх порівняння із іншими апроксимаціями). Для деяких методів наведемо реалізацію мовою програмування Python. Здебільшого розглядаються прості апроксимації – апроксимація Крамера-Лундберга, асимптотика для важких хвостів, дифузійна апроксимація, методи Монте-Карло, формула Беекмана.

Альтернативний варіант відносно теоретичних апроксимацій – методи Монте-Карло. Для процесу страхового ризику можна провести симуляцію у великій кількості, у сучасних умовах комп'ютери на це здатні. Зокрема, так можна оцінити ймовірність банкрутства, середній час до банкрутства. У роботі розглянуто два методи Монте-Карло. Перший – прямолінійний, він полягає у прямій симуляції процесу ризику та підрахунку частки траєкторій, що зазнали банкрутства. Другий – заснований на формулі згортки Беекмана, він має більшу обчислювальну складність, але і вважається більш точним.

Фактично, два методи Монте-Карло оцінюють різні величини. Прямолінійний метод виконує симуляцію процесу на скінченному проміжку часу, тобто оцінюється ймовірність банкрутства на проміжку $[0, T]$. Метод, заснований на формулі Беекмана, пропонує оцінку ймовірності на нескінченному часовому проміжку. У випадку, коли чисельне інтегрування не займає багато часу, краще користуватись формулою Беекмана. Інакше – простіше виконувати пряму симуляцію процесу ризику, користуючись властивостями процесу Пуассона.

1.4 Висновки до розділу 1:

Моделі і методи страхової математики часто переплітаються із фінансовою інженерією. Проблеми ймовірності банкрутства були поставлені у 1930 році, але розрахунки, пов'язані із нею виконуються актуаріями до сих пір.

До методів апроксимації ймовірності банкрутства висунуто очевидні вимоги – точність та обчислювальна складність, і необхідно знайти компроміс між цими двома якостями алгоритму. За різних початкових умов деякі методи працюють краще за інші.

Умовно методи апроксимації можна поділити на два класи – прості і складні. Прості методи не використовують знання про розподіл, а вимагають лише деяких початкових або центральних моментів величини. Складні методи вимагають знання функції розподілу виплат. У роботі наявні методи обох типів, тому буде доречно порівняти їх результати.

РОЗДІЛ 2 ТЕОРЕТИЧНІ ВІДОМОСТІ

2.1 Однорідний процес Пуассона

Випадковий процес N на $T = [0, \infty)$ називають однорідним процесом Пуассона із інтенсивністю λ , якщо виконуються наступні умови:

- 1) $N(0)=0$ (майже напевно)
- 2) Нехай $(s_1, t_1], (s_2, t_2], \dots, (s_n, t_n]$ - проміжки часу, що не перетинаються. Тоді прирости $N(t_1) - N(s_1), N(t_2) - N(s_2), \dots, N(t_n) - N(s_n)$ є незалежними випадковими величинами.
- 3) $P(N(t+h) - N(t) \geq 2) = o(h)$ ординарність. Дві події не настають одночасно.
- 4) Прирости процесу мають розподіл Пуассона (2.1):

$$N(t) - N(s) \sim \text{Pois}(\lambda(t-s)), t \geq s \quad (2.1)$$

Математичне сподівання, коваріаційна функція (2.2):

$$EN(t) = \lambda t, \text{cov}(N(s), N(t)) = \lambda \min\{s, t\} \quad (2.2)$$

Розподіл часу настання подій (теорема 1):

Нехай $T_1, T_2, \dots, T_n$ - моменти часу, в які відбувається стрибок процесу(моменти настання подій). Позначимо:

$W_1 = T_1 - T_0, W_2 = T_2 - T_1, \dots, W_n = T_n - T_{n-1}$ - проміжки часу між подіями.

Теорема 1. $W_1, W_2, \dots, W_n \sim \text{Exp}(\lambda)$ - проміжки часу є незалежними в сукупності, експоненційно розподіленими випадковими величинами;

(доведення можна знайти в підручнику [4], або у монографіях [1], [3])

Асимптотика часу очікування події.

Користуючись посиленням законом великих чисел, можемо приблизно оцінити час настання події (2.3):

$$\frac{T_n}{n} = \frac{W_1 + W_2 + \dots + W_n}{n} \rightarrow \frac{1}{\lambda} \text{ (збіжність майже напевно)} \quad (2.3)$$

Звідси можна сказати $T_n \approx \frac{n}{\lambda}$ – наближене значення часу настання n -ої події. Або, більш точно, розподіл суми незалежних експоненційних величин (приростів часу) має гамма розподіл: $T_n \sim \Gamma(n, \lambda)$.

2.2 Складний процес Пуассона

Випадковий процес $X(t)$ на $T = [0, \infty)$ називають складним процесом Пуассона (2.4):

$$X(t) = \sum_{i=1}^{N(t)} \xi_i \quad (2.4)$$

де $N(t)$ – процес Пуассона із інтенсивністю λ ;

$\xi_1, \xi_2, \dots, \xi_n$ – незалежні, однаково розподілені випадкові величини;

$E\xi_i = \mu, D\xi_i = \sigma^2$ – випадкові величини мають математичне сподівання та дисперсію.

Складний процес Пуассона використовують у страховій математиці для моделювання виплат клієнтам. Припускається, що нещасні випадки відбуваються за потоком подій Пуассона, а виплати – незалежні та однаково розподілені випадкові величини.

Математичне сподівання складного процесу Пуассона (2.5):

$$EX(t) = EN(t)E\xi = \lambda\mu t \quad (2.5)$$

Коваріаційна функція складного процесу Пуассона (2.6):

$$\text{cov}(X(s), X(t)) = (\mu^2 + \sigma^2)\lambda \min\{s, t\} \quad (2.6)$$

(доведення можна знайти у підручнику [4], а також в монографії [3])

Закон великих чисел для складного процесу Пуассона (2.7):

$$\frac{X(t)}{t} \rightarrow \lambda\mu \text{ (збіжність майже напевно)} \quad (2.7)$$

Центральна гранична теорема для складного процесу Пуассона (2.8):

$$\frac{X(t) - EX(t)}{\sqrt{DX(t)}} = \frac{X(t) - \lambda\mu t}{\sqrt{(\sigma^2 + \mu^2)\lambda t}} \rightarrow N(0,1) \text{ (збіжність за розподілом)} \quad (2.8)$$

2.3 Процес страхового ризику

Розглядається випадковий процес, що відображає капітал страхової компанії на момент часу t . Загальна формулу можна описати так (2.9):

$$U(t) = u + P(t) - S(t) \quad (2.9)$$

де $U(t)$ – процес страхового ризику, u – початковий капітал страхової компанії, $P(t)$ – випадковий процес, що відображає грошові надходження в страхову компанію, $S(t)$ – випадковий процес, що відображає виплати клієнтам при нещасних випадках.

Введемо події банкрутства та небанкрутства.

Нескінченний часовий горизонт:

“банкрутство” = $[\exists t > 0: U(t) < 0]$ – банкрутство настає у тому випадку, якщо є хоча б один момент, коли компанія не може покрити збитки.

$\psi(u) = P[\exists t > 0: U(t) < 0]$ – ймовірність банкрутства при початковому капіталі u .

“небанкрутство” = $[\forall t > 0: U(t) > 0]$ – якщо капітал компанії завжди додатний і ніколи не опускається нижче нуля. $\phi(u) = 1 - \psi(u)$ – ймовірність небанкрутства при початковому капіталі u .

Скінченний часовий горизонт:

“банкрутство” = $[\exists t \in [0, T]: U(t) < 0]$.

$\psi(u, T) = P[\exists t \in [0, T]: U(t) < 0]$ – ймовірність банкрутства.

“небанкрутство” = $[\forall t \in [0, T]: U(t) \geq 0]$.

$\phi(u, T) = 1 - \psi(u, T)$ – ймовірність небанкрутства.

час до банкрутства: $\tau(u) = \inf[t: U(t) < 0, \psi(u) = 1]$

(події формально введено у монографіях [1], [3], приклади їх дослідження – [11], [12])

Страхова компанія має прагнути до збагачення. Цілком природньо вимагати, щоб принаймні очікувані значення капіталу компанії зростали:

$$EU(t) = u + (EP(t) - ES(t)) \quad (2.10)$$

Тобто, якщо $EP(t) \leq ES(t)$, то компанія в середньому буде зазнавати збитків. Нескладно зрозуміти, що існує момент часу, при якому навіть математичне сподівання капіталу менше нуля. Це означає, що банкрутство обов’язково відбудеться.

Теорема 2. Net-Profit Condition – умова чистого прибутку.

$EP(t) > ES(t) \rightarrow \psi(u) < 1$ – умова, за якої страхова компанія отримує прибуток принаймні в середньому, може привести до банкрутства, але не із ймовірністю 1.

$EP(t) = ES(t) \rightarrow \psi(u) = 1$ – банкрутство відбудеться із ймовірністю 1, внаслідок випадкових осциляцій процес рано чи пізно все одно опуститься нижче нуля.

$EP(t) < ES(t) \rightarrow \psi(u) = 1$ – умова, за якої компанія в середньому втрачає, обов’язково призведе до банкрутства.

У подальшому цю умову потрібно конкретизувати для кожної окремої моделі (конкретизувати функції $P(t)$ та $S(t)$, уточнити ймовірнісні припущення про їх природу).

2.3 Як обрати функцію прибутку $P(t)$?

Розглянемо декілька класичних випадків як обрати функцію прибутку. Здебільшого це будуть детерміновані функції на основі величини $ES(t)$.

1) принцип еквівалентності: $P(t) = ES(t)$

Цей спосіб можна вважати чесним, оскільки в середньому компанія ані втрачає, ані отримує. Такий спосіб здебільшого приведе до банкрутства (на нескінченному проміжку часу – банкрутство відбудеться майже напевно). Користуючись центральною граничною теоремою, наведемо наївну апроксимацію процесу ризику:

$$U(t) = u + S(t) - ES(t) \approx N(u, DS(t)) \quad (2.11)$$

Дисперсія процесу $S(t)$ буде рости із часом (звичайно, тому що кількість виплат росте із часом). Відповідно, і ймовірність події $[U(t) < 0]$ ненульова і зростає із часом.

2) принцип “запасу міцності”: $P(t) = (1 + \rho)ES(t)$

Коефіцієнт ρ прийнято називати коефіцієнтом безпечного навантаження.

Аналогічно, наведемо апроксимацію за центральною граничною теоремою:

$$U(t) = u + \rho ES(t) + ES(t) - S(t) \approx N(u + \rho ES(t), DS(t)) \quad (2.12)$$

Дисперсія $DS(t)$ звичайно, буде зростати, але і нормальний розподіл вже буде сильно зміщений вправо, за допомогою додаткових надходжень, що гарантуються запасом ρ .

За такого вибору, умова NPC виконується, ймовірність банкрутства вже буде менше одиниці.

ρ – зазвичай набуває невеликих значень (0.02, 0.05, 0.1) – щоб не сильно завищувати вартість страхового контракту. Інакше – менше людей будуть користуватися страховими послугами.

3) принцип дисперсії: $P(t) = ES(t) + \alpha DS(t)$

4) принцип стандартного відхилення: $P(t) = ES(t) + \alpha \sqrt{DS(t)}$

Запропоновані способи є типовими для актуарних розрахунків і розглянуто у [3].

2.4 Огляд розподілів виплат

Цілком природним є питання – який розподіл виплат обрати, щоб він принаймні в деякому наближенні відповідав реальності? На розподіл виплат накладено очевидне обмеження – випадкові величини мають бути невід’ємними, бажано щоб вони мали перший та другий моменти.

Експоненційний розподіл є прикладом розподілу, хвіст якого дуже швидко прямує до нуля. Такою властивістю також володіє усічений нормальний розподіл.

Але є деякі розподіли, в яких хвіст функції розподілу прямує до нуля повільніше. Відповідно, у них ймовірність великих значень вища і з ними доведеться працювати окремо.

Зокрема, стандартна теорія, розроблена Крамером та Лундбергом, використовує припущення про “легкий хвіст” розподілів виплат. Ряд результатів виявляється невірним для виплат із важким хвостом, пізніше і для них розробили деяку теорію, але вона дещо відрізняється від виплат із легким хвостом.

Тому розподіли виплат ми маємо поділити на два класи:

Розподіл виплати назвемо розподілом із легким хвостом, якщо:

$$\exists \beta > 0 : \lim_{x \rightarrow \infty} \frac{1-F(x)}{\exp(-\beta x)} = c < \infty \quad (2.13)$$

Це означає, що хвіст функції розподілу спадає так само швидко, як хвіст експоненційного розподілу ($c > 0$) або навіть швидше ($c = 0$).

До розподілів із легким хвостом відносять експоненційний розподіл, нормальний розподіл, а також довільні розподіли на скінченному відрізку (наприклад, рівномірний). У таблиці 2.1 наведені деякі розподіли із легким хвостом функції розподілу.

Таблиця 2.1 - Типові розподіли із легким хвостом

Назва розподілу	Щільність розподілу	Параметри
Експоненційний	$f(x) = \lambda \exp\{-\lambda x\}$	$\lambda > 0$
Гамма	$f(x) = \frac{\alpha^\beta x^{\beta-1} e^{-\alpha x}}{\Gamma(\beta)}, x > 0$	$\alpha > 0, \beta > 0$
Усічений нормальний	$X \sim \text{abs}(N(a, \sigma^2))$	$a \in R, \sigma > 0$
Довільний розподіл, що має обмеження зверху (рівномірний тощо)		

Розподіл виплати назвемо розподілом із тяжким хвостом, якщо:

$$\forall \beta > 0 : \lim_{x \rightarrow \infty} \frac{1-F(x)}{\exp(-\beta x)} = \infty \quad (2.14)$$

До розподілів із тяжкими хвостами відносяться лог-нормальний розподіл, розподіл Вейбулла при $r < 1$, розподіл Парето тощо. У таблиці 2.2 наведено два

найбільш класичних розподіли із важкими хвостами – розподіли Парето і Вейбулла.

Таблиця 2.2 - Типові розподіли із важкими хвостами

Назва розподілу	Щільність або хвіст розподілу	Параметри
Парето	$f(x) = \frac{\alpha x_m^\alpha}{x^{\alpha+1}}, x > 0$	$\alpha > 0, x_m > 0$
Вейбулла	$1 - F(x) = \exp\left(-\left(\frac{x}{\lambda}\right)^k\right)$	$1 > k > 0, \lambda > 0$

2.5 Висновки до розділу 2

Робота із моделями страхової математики потребує значної кількості теоретичних відомостей – зокрема доскональні знання про конструкцію процесу Пуассона. Необхідно мати уявлення про загальний вигляд процесу ризику, про умову NPC – умову чистого прибутку, при невиконанні якої компанія приречена на банкрутство із ймовірністю 1.

Також необхідно уявляти, який характер має розподіл страхової виплати. Експоненційний, гамма-розподіл та нормальний розподіл відносять до класу виплат із легким хвостом. Розподіли Парето, Вейбулла і логнормальний прийнято відносити до класу розподілів із важким хвостом. Дані типи розподілів поведуть себе на нескінченності по-різному.

Методи апроксимації здебільшого висувають припущення про легкий або важкий хвіст розподілу виплати. Деякі працюють лише із легкими виплатами, деякі – лише із важкими, деякі – універсальні. Важливо розуміти, у чому різниця – адже при невиконанні передположок можна отримати недостовірні результати.

РОЗДІЛ 3 ЙМОВІРНІСТЬ БАНКРУТСТВА ТА МЕТОДИ ЇЇ РОЗРАХУНКУ

3.1 Модель Крамера-Лундберга

Весь цей час ми говорили про модель ризику в найбільш загальних словах. Відповідно до формули (3.1):

$$U(t) = u + P(t) - S(t) \quad (3.1)$$

де $P(t)$ – випадковий процес прибутку та $S(t)$ – процес виплат клієнтам.

Основним об’єктом розгляду є модель Крамера – Лундберга (1930, [1] [3]), представлений у формулі (3.2):

$$U(t) = u + ct - \sum_{i=1}^{N(t)} X_i \quad (3.2)$$

де $U(t)$ – випадковий процес, що відображає капітал страхової компанії на момент часу t , u – початковий капітал страхової компанії, c – інтенсивність грошових внесків від клієнтів, $N(t)$ – потік нещасних випадків (представлений у вигляді однорідного пуассонівського процесу), $X_1, X_2, \dots, X_n$ – випадкові величини, що описують виплати клієнтам при нещасних випадках. Незалежні в сукупності, незалежні від $N(t)$. Однаково розподілені, мають математичне сподівання.

Центральним інтересом будуть ймовірності банкрутства на скінченному та нескінченному проміжку часу $(\psi(u, T), \psi(u))$. Звичайно, з точки зору бізнесу $\psi(u, T)$ представляє собою більший інтерес, оскільки страхування часто діє в обмеженому часовому горизонті, найчастіше від одного до п’яти років. А з іншого боку, зрозуміло що $\psi(u) > \psi(u, T)$. Зменшення ризику на

нескінченному горизонті часу дасть більш гарантований результат і точно знизить ризики на скінченному проміжку часу.

Про $\psi(u)$ можна сказати більше фактів. Інтегральне рівняння для неї простіше, в деяких випадках його можна розв'язати аналітично. Деякі відомості наявні і про час до руїни.

На рисунку 3.1 можемо побачити як виглядають типові траєкторії процесу: лінійно зростають та падають вертикально вниз у момент нещасного випадку.

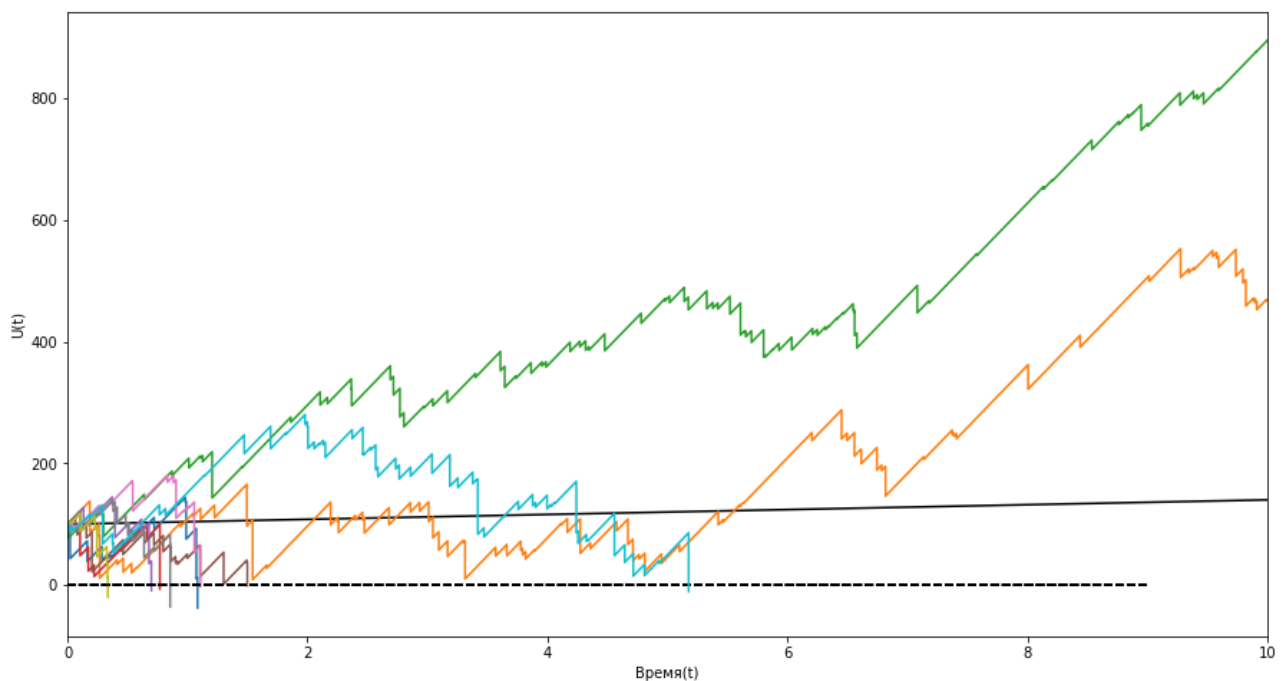


Рисунок 3.1 - Типові траєкторії процесу Крамера-Лундберга.

Пунктирна лінія – нульова межа, межа банкрутства, чорна пряма – лінія математичного сподівання процесу ризику.

3.2 Характеристичний коефіцієнт, нерівність Лундберга

Покладемо зараз і надалі $c = (1 + \rho)\lambda\mu$ – тобто оберемо $P(t)$ за принципом очікуваного значення.

В будь-якій нерівності/твердженні/теоремі вимагається виконання умови NPC ($c > \lambda\mu$), інакше – ймовірність банкрутства дорівнює одиниці і обговорення не має сенсу.

Введемо випадкову величину $Z = X - c\tau$, де $X \sim F(x)$, $\tau \sim \text{Exp}(\lambda)$. Також нехай $\chi_Z(x) = Ee^{xz}$ – твірна функція випадкової величини Z .

Характеристичним коефіцієнтом Лундберга R називається єдиний додатний розв'язок рівняння $\chi_Z(R) = 1$, або у більш зручному вигляді:

$$\chi_Z(R) = 1 \rightarrow \chi_X(R) * \chi_\tau(-cR) = 1 \rightarrow \chi_X(R) = 1 + \frac{cR}{\lambda} \quad (3.3)$$

На рисунку 3.2 можемо бачити наглядно, що розв'язок рівняння існує, це пов'язано із тим, що твірна функція є опуклою.

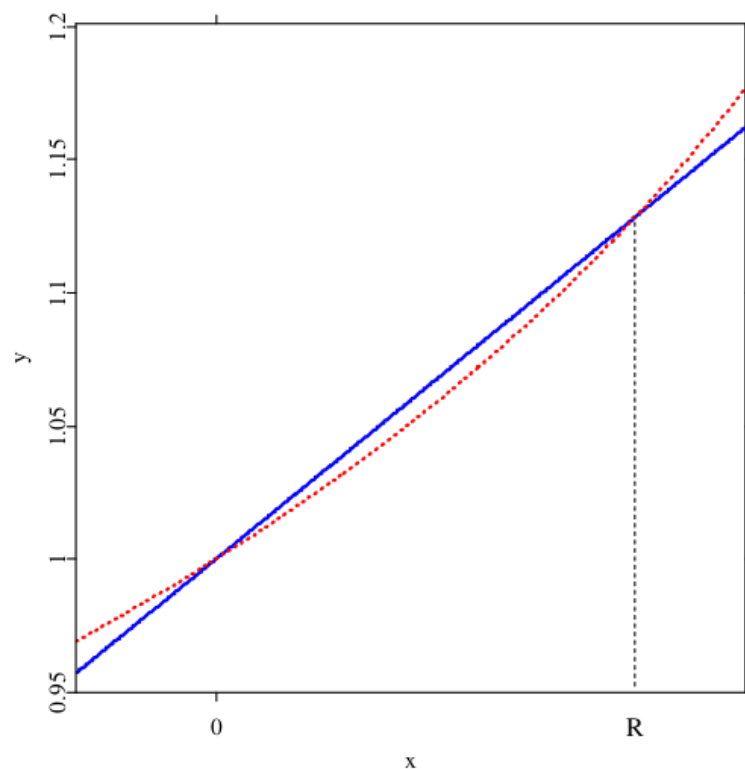


Рисунок 3.2 - Ілюстрація існування характеристичного коефіцієнту. Синя лінія відповідає правій частині, опукла пунктирна крива – лівій.

Якщо твірна функція величини Z існує, то вона є опуклою функцією (наприклад, видно на рисунку 3.2). Відповідно, дане рівняння має принаймні два розв'язки, один з них може бути від'ємним. Нас цікавить більший розв'язок, оскільки далі він буде використовуватись в нерівності Лундберга.

Коефіцієнт R знаходиться аналітично у випадку, якщо X_i розподілено за експоненційним законом, формула (3.4):

$$R = \frac{\rho}{(1+\rho)\mu} \quad (3.4)$$

Для його оцінки існують нерівності (Asmussen, 1986), формула (3.5):

$$R < \frac{2\rho\mu}{EX^2}, (EX^2 < \infty) \quad (3.5)$$

За умови наявності третього початкового моменту у випадкової величини, можна ввести більш жорстку нерівність (Panjer - Willmot, 1992):

$$R < \frac{12\mu\rho}{3EX^2 + \sqrt{9(EX^2)^2 + 24\mu\rho EX^3}}, E|X|^3 < \infty \quad (3.6)$$

Якщо характеристичне рівняння не розв'язується аналітично, то можна застосувати чисельні методи для нелінійних рівнянь (наприклад, метод Ньютона, (3.7)):

$$R_0 = \frac{2\rho\mu}{EX^2}, R_j = R_{j-1} - \frac{\chi_X(R_{j-1}) - 1 - \frac{cR_{j-1}}{\lambda}}{\chi'_X(R_{j-1}) - \frac{c}{\lambda}} \quad (3.7)$$

Нерівність Лундберга (розподіл виплат із легким хвостом).

Якщо розв'язок характеристичного рівняння існує, то ймовірність банкрутства можна оцінити експонентою (3.8):

$$\psi(u) \leq e^{-Ru} \quad (3.8)$$

e^{-Ru} є найкращою експоненційною оцінкою ймовірності банкрутства. Пізніше ця ідея розвилася в апроксимацію Крамера-Лундберга.

Характеристичний коефіцієнт можна знайти лише у випадку виплат із легким хвостом. У випадку виплат із тяжким хвостом твірна функція може просто не існувати (так, наприклад при Парето-розподілених виплатах генератриса визначена лише на проміжку $(-\infty, 0]$, відповідно і рівняння розв'язувати немає сенсу).

3.3 Рівняння для ймовірності банкрутства.

Теорема. Ймовірність виживання на нескінченному часі $\phi(u)$ задовольняє інтегральному рівнянню Вольтерра (3.9):

$$\phi(u) = \phi(0) + \frac{1}{1+\rho} \int_0^u \phi(u-t)(1-F(t))dt \quad (3.9)$$

де $F(y)$ – функція розподілу окремої виплати (доведення факту можна знайти в монографіях [3], [1], [11])

Дане рівняння може бути розв'язане, наприклад, перетворенням Лапласа. У деяких випадках обернене перетворення Лапласа від зображення $\Phi(p)$ можна взяти аналітично. Інакше – потрібно брати обернене перетворення Лапласа чисельними методами.

Особливі випадки:

Експоненційні виплати. Нехай $X_i \sim \text{Exp}\left(\frac{1}{\mu}\right)$. $EX_i = \mu$ Тоді

$$\psi(u) = \frac{1}{1+\rho} e^{-\frac{\rho u}{(1+\rho)\mu}} \quad (3.10)$$

Нульовий початковий капітал (будь-який розподіл виплат):

$$\psi(0) = \frac{1}{1+\rho} \quad (3.11)$$

Нескінченний початковий капітал (будь-який розподіл виплат): $\psi(\infty) = 0$

3.4 Апроксимація Крамера – Лундберга (легкі виплати)

Найбільш відомою та простою апроксимацією ймовірності банкрутства є апроксимація Крамера – Лундберга. Вона заснована на співвідношенні, доведеному в 1945 році (3.12):

$$\lim_{u \rightarrow \infty} e^{Ru} \psi(u) = \frac{\rho u}{\chi'_X(R) - c/\lambda} \quad (3.12)$$

Звідси, можна наближено записати (3.13):

$$\psi(u) \approx e^{-Ru} \frac{\rho u}{\chi'_X(R) - c/\lambda} \quad (3.13)$$

Звичайно, апроксимація підходить лише для легких виплат (гамма, усічений нормальний). Розподіли по типу Парето, Вейбулла автоматично не підходять, для них існує інша асимптотика.

Асимптотика для виплат із важким хвостом.

Введемо перетворення “інтегрованого хвоста”. Знайдемо функцію розподілу нової величини X_I :

$$F_{X_I}(y) = \frac{1}{\mu} \int_0^y (1 - F(t)) dt \quad (3.14)$$

Для розподілів із важкими хвостами типовим є припущення належності інтегрованого розподілу до класу субекспоненційних розподілів:

$$P\{X_{I1} + X_{I2} > u\} \approx 2P\{X_{I1} > u\} \quad (3.15)$$

Зокрема, якщо інтегрований розподіл є субекспоненційним (а це часто виконується, зокрема у випадку розподілів Парето та Вейбулла), тоді ймовірність банкрутства має наступну асимптотику:

$$\lim_{u \rightarrow \infty} \frac{\psi(u)}{1 - F_I(u)} = \frac{1}{\rho} \quad (3.16)$$

$$\psi(u) \approx \frac{1 - F_I(u)}{\rho} \quad (3.17)$$

Апроксимація має меншу швидкість збіжності, ніж апроксимація Крамера - Лундберга. Обидві вимагають повного знання розподілу виплат (у першому випадку – маємо знати твірну функцію моментів величини, у другому – маємо обчислювати інтегральне перетворення для цієї випадкової величини).

Прості апроксимації ймовірності банкрутства коротко наведено у статті [6], та більш детально вивчаються у монографіях [1], [3].

3.5 Дифузійна апроксимація.

Ідея дифузійної апроксимації полягає у граничних властивостях складного процесу Пуассона, яким описується загальна кількість витрат страхової компанії. У [2] можна знайти обґрунтування того, що при збільшенні часу складний процес Пуассона прямує до Вінерівського процесу в слабкому сенсі (за розподілом).

Апроксимація використовує дві ідеї, що згадані раніше – нерівність Лундберга та оцінку характеристичного коефіцієнта.

$$\psi(u) \approx e^{-\frac{2\rho u \mu}{E X^2}} \quad (3.15)$$

Тобто, для її використання вимагається наявність другого моменту величини виплат. Апроксимація має гарну точність при великому початковому капіталі, її прийнято використовувати як гарне перше наближення ймовірності банкрутства.

3.6 Прямий метод Монте – Карло

Альтернативою для граничних властивостей є прямий метод Монте-Карло. Він полягає у тому, щоб провести симуляцію процесу $U(t)$ на деяких часових проміжках (наприклад, рік, 5 років) велику кількість разів.

Таким чином, результат кожного експерименту визначається станом капіталу після останньої виплати. Нехай $p_i = 1$, якщо траєкторія банкрутства, і $p_i = 0$ якщо банкрутства не було для даної траєкторії.

При великій кількості експериментів маємо одержати точну оцінку справжньої ймовірності.

Нехай $p_1, p_2, \dots, p_n$ - результати експерименту (незалежні, однаково розподілені величини Бернуллі). Тоді, за посиленням законом великих чисел:

$$\frac{p_1 + p_2 + \dots + p_n}{n} \rightarrow \psi(u) \text{ (збіжність майже напевно)} \quad (3.16)$$

Наведемо спрощено алгоритм симуляції одної траєкторії процесу ризику на обмеженому часовому проміжку $[0, T]$:

time = 0

capital = u

While ($time < T$):

$\tau_i \sim Exp(\lambda)$ - генеруємо час i -ої виплати

$time = time + \tau_i$ - оновлюємо поточний момент часу

$X_i \sim F(x)$ - генеруємо суму виплати

$capital = capital + c\tau_i - X_i$ - оновлюємо капітал на момент часу $time$.

If ($capital < 0$): break – перевіряємо умову банкрутства.

Даний метод Монте-Карло має переваги:

- 1) алгоритм є досить простим для комп'ютерної реалізації;
- 2) при великій кількості випробувань, можемо отримати точну оцінку ймовірності банкрутства. За умов, що ймовірність досить значна;
- 3) симуляції незалежні одне від одного;

Алгоритм також має недоліки:

- 1) нездатність оцінювати ймовірність банкрутства на нескінченному проміжку часу (неможливо проводити нескінченно довгу симуляцію).
- 2) метод не підходить для оцінювання рідких подій.
(наприклад, коли $\psi(u) < \frac{1}{N}$, де N – кількість симуляцій). В такому випадку необхідна кількість симуляцій є просто недоступною (з точки зору обчислювальних потужностей)

Є багато різних думок з приводу методів Монте-Карло. Зокрема, в [3] рекомендують користуватись ними лише за крайньої необхідності, коли інші можливі способи не працюють.

3.7 Метод Монте-Карло на основі формули згортки Бескмана

Для обчислення формули Бескмана необхідно ввести перетворення інтегруючого хвоста для випадкових величин виплат. За формулою (30), функція розподілу нової (інтегрованої) величини:

$$F_{X_I}(y) = \frac{1}{\mu} \int_0^y (1 - F(t)) dt \quad (3.17)$$

Тоді, фактично, маємо послідовність $X_{I1}, X_{I2}, \dots, X_{In}$ незалежних інтегрованих величин.

За формулою згортки Бескмана, ймовірність небанкрутства $\phi(u)$ можна представити у вигляді геометричної суми:

$$\phi(u) = \left(1 - \frac{\lambda\mu}{c}\right) \left(1 + \sum_{n=1}^{\infty} \left(\frac{\lambda\mu}{c}\right)^n P\{X_{I1} + \dots + X_{In} < u\}\right) \quad (3.18)$$

Зараз формула Бескмана записана у вигляді повного математичного сподівання. Її можна подати за допомогою суми випадкової кількості випадкових величин, не втрачаючи загальності:

$$v \sim \text{Geom}\left(1 - \frac{\lambda\mu}{c}\right) \quad (3.19)$$

$$\phi(u) = P\{X_{I1} + X_{I2} + \dots + X_{Iv} < u\} \quad (3.20)$$

Або, в термінах ймовірності банкрутства:

$$\psi(u) = P\{X_{I1} + X_{I2} + \dots + X_{Iv} > u\} \quad (3.21)$$

Ймовірність банкрутства, записана у такій формі дає змогу застосовувати методи Монте-Карло. Випадкова величина v незалежна від інтегрованих виплат.

Алгоритм досить простий, навіть не потребує псевдокоду. Спочатку генеруємо геометричну величину v . Потім генеруємо вибірку інтегрованих виплат розміром v , перевіряємо умову банкрутства.

Формула згортки Бескмана вважається більш точною, ніж пряме моделювання. Але вона має великий недолік – інтегроване перетворення в більшості випадків потрібно обчислювати чисельними методами.

Приклади, де можна обчислити інтегроване перетворення аналітично: Інтегроване перетворення для експоненційних виплат не змінює випадкову величину:

$$X_i \sim \exp\left(\frac{1}{\mu}\right) \rightarrow 1 - F(x) = e^{-\frac{x}{\mu}} \rightarrow \frac{1}{\mu} \int_0^y e^{-\frac{t}{\mu}} dt = \frac{\mu - \mu e^{-\frac{y}{\mu}}}{\mu} = 1 - e^{-\frac{y}{\mu}} \quad (3.22)$$

Інтегроване перетворення для константної виплати:

$$X = \mu \rightarrow 1 - F(x) = I\{X \leq x\} \rightarrow \frac{1}{\mu} \int_0^y I\{X \leq t\} dt = \begin{cases} 1, & y \geq \mu \\ \frac{y}{\mu}, & 0 < y < \mu \end{cases} \quad (3.23)$$

Таким чином, інтегрованим перетворенням від константи є рівномірно розподілена величина $X_i \sim U[0, \mu]$.

3.8 Висновки до розділу 3:

На даний момент існує досить багато способів обчислити ймовірність банкрутства, звичайно, кожен зі своїми припущеннями на модель:

- дифузійна апроксимація вимагає існування першого та другого моментів;
- апроксимація Крамера-Лундберга вимагає існування характеристичного коефіцієнту, ну і, відповідно, розподіл виплат із легкими хвостами;
- прямий метод Монте-Карло не висуває жодних припущень на модель. Для його ефективної роботи ймовірність банкрутства має бути досить значною;
- формула Бескмана вимагає обчислення інтегрального перетворення. Але, якщо це вдається робити ефективно, отримаємо більш точні оцінки (ймовірності банкрутства);

– корисно порівнювати і прямий метод Монте-Карло, і формулу Бескмана. За своєю природою вони оцінюють різні величини, але на великих часових горизонтах їх результати мають бути схожими.

РОЗДІЛ 4 МОДЕЛЮВАННЯ

4.1 Умова Net-Profit Condition (умова чистого прибутку)

Зафіксуємо початкові умови процесу ризику:

$$\mu = 20, \lambda = 10, X_i \sim \exp\left(\frac{1.0}{20}\right)$$

На рисунках 4.1-4.3 зображено моделювання траєкторій на 5 років.

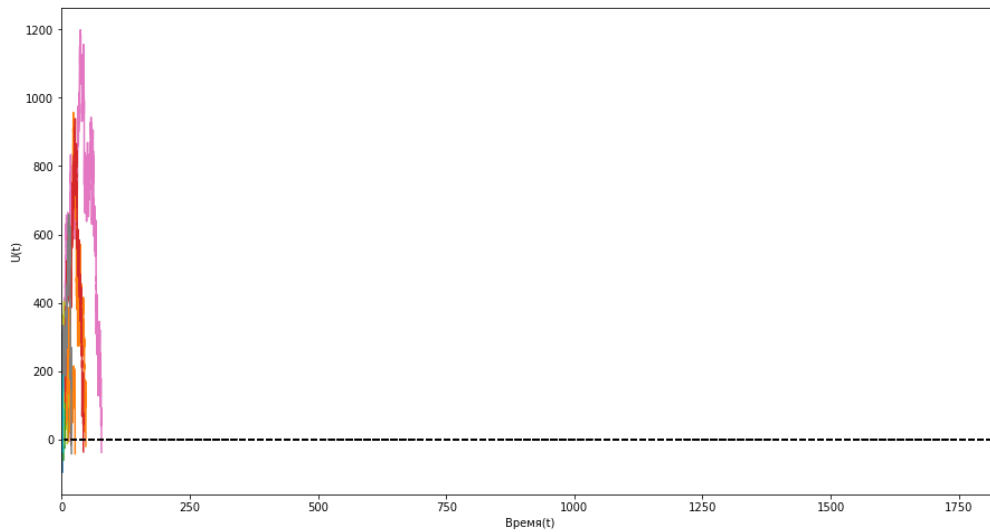


Рисунок 4.1 Траєкторії процесу без NPC. $c = 0.95\lambda\mu$

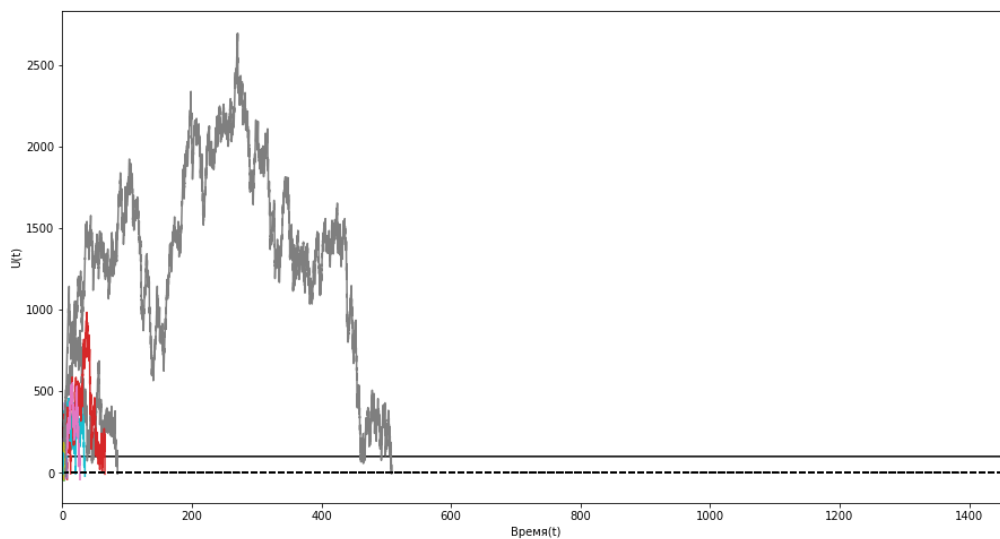


Рисунок 4.2 Граничний випадок NPC. $c = \lambda\mu$

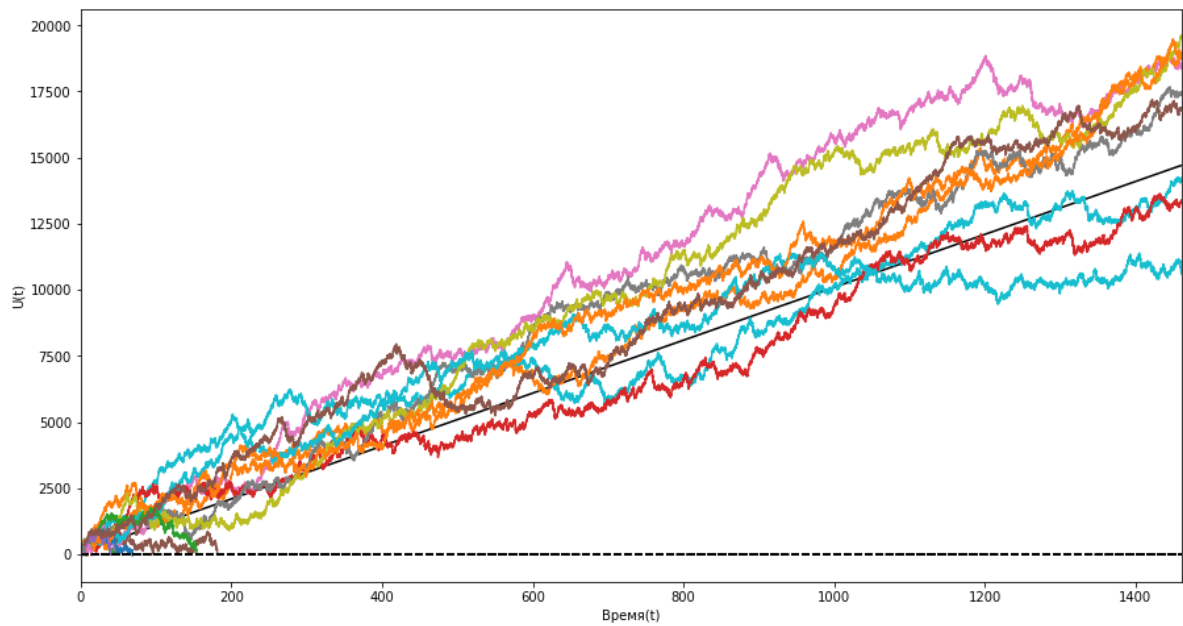


Рисунок 4.3 Траєкторії процесу із виконанням умови NPC. $c = 1.05\lambda\mu$

Можемо бачити, що:

- на першому рисунку жодна траєкторія не “дожила” навіть до кінця першого року;
- на другому рисунку середні втрати = середнім надходженням. Це означає що процес буде розвиватись приблизно навколо початкового капіталу, але внаслідок випадкових осциляцій рано чи пізно опуститься до нуля. Це і видно з рисунку;
- на третьому рисунку умова Net-Profit Condition виконується. Це означає, що в середньому в компанію надходить більше, ніж втрачається. Відповідно, вже значна частина траєкторій впевнено піднімається вгору, і лише деяка мала частина зазнає банкрутства;
- вибір коефіцієнта нульовим або менше є неприпустимим. Відповідно до теореми 2 – такий вибір обов’язково призведе траєкторію до банкрутства.

4.2 Дослідження процесу із експоненційним розподілом виплат

Для експоненційних виплат ми знаємо аналітичну ймовірність, вона обчислена у попередньому розділі (формула (3.10)):

$$\psi(u) = \frac{1}{1 + \rho} e^{-\frac{\rho u}{\mu(1+\rho)}}$$

Відповідно, для початкових умов:

$$\mu = 20, u = 100, \lambda = 10, \rho = 0.05 \rightarrow \psi(u) = 0.7509$$

Застосуємо простий метод Монте-Карло. Він полягає у тому, щоб провести симуляцію процесу $U(t)$ на деяких часових проміжках (наприклад, рік, 5 років – класичні часові горизонти у страхуванні) велику кількість разів, потім обчислити частку траєкторій що зазнали банкрутства. Це і буде емпіричною оцінкою справжньої ймовірності банкрутства.

Результати прямого методу Монте-Карло:

моделювання на 1 рік, 1000 симуляцій – 0.732

моделювання на 5 років, 1000 симуляцій – 0.747

Оцінки методу Монте-Карло досить близько до теоретичної ймовірності. Відхилення у 1-2% незначне, в цілому задовільне.

Формула Беекмана.

Нагадаємо, формула Беекмана – це інший метод Монте-Карло, що полягає у моделюванні інтегрованих випадкових величин (формули (3.19), (3.20), (3.21)):

$$\psi(u) = P\{X_{I1} + X_{I2} + \dots + X_{Iv} > u\}$$

$$v \sim \text{Geom}\left(1 - \frac{\lambda\mu}{c}\right)$$

Для експоненційних виплат чисельне інтегрування проводити не потрібно – інтегроване перетворення можна обчислити аналітично (виконано у (40)).

Провели 1000 симуляцій – вони дали оцінку ймовірності банкрутства 0.748.

Зафіксуємо початкові умови $c = 1.05\lambda\mu$, $\lambda = 10$, $X_i \sim \text{Exp}\left(\frac{1.0}{20}\right)$, $\mu = 20$

Та проведемо дослідження для різного початкового капіталу (логарифмічна шкала). У таблиці 4.1 наведено ймовірності банкрутства для випадку експоненційних виплат.

Таблиця 4.1 Порівняння апроксимацій (експоненційні виплати)

	u=10	u=100	u=1000	u=10000
Аналітичний розв'язок	0.9299	0.7506	0.088	4.35e-11
Дифузійна апроксимація	0.9753	0.7788	0.082	1.38e-11
Метод Монте-Карло (1 рік)	0.942	0.718	0.076	0.0
Метод Монте-Карло (5 років)	0.917	0.721	0.084	0.0
Формула Бескмана (нескінченний час)	0.924	0.743	0.092	0.0

За отриманими оцінками можна зробити зауваження:

- дифузійна апроксимація – найбільш простий метод, який підходить за більшості початкових умов.
- методи Монте-Карло не працюють, якщо ймовірність банкрутства занадто мала (1000 симуляцій – це мало, а $1e10 - 1e12$ – занадто багато для такого роду задачі)
- виглядає так, що дифузійна апроксимація є найменш точною. Це дуже просто пояснюється – дифузійна апроксимація ніяк не використовує знання про розподіл, користується тільки першим та другим моментом величини виплат;

- грубий метод Монте-Карло показує досить точні оцінки істинної ймовірності. Однак, не підходить для ситуацій, де ймовірність банкрутства дуже незначна (наприклад, $u=10000$);
- ймовірності, оцінені за формулою Бескмана та грубим методом Монте-Карло досить близькі за значеннями. Виглядає так, що значення, отримані за формулою Бескмана ближчі до аналітичного розв'язку.

4.3 Дослідження випадку пуассонівських виплат.

Розподіл Пуассона є дискретним розподілом. Можна довести, що розподіл Пуассона є розподілом із легким хвостом. Для нього існує характеристичний коефіцієнт та апроксимація Крамера-Лундберга.

Звичайно, аналітичний розв'язок інтегрального рівняння нам недоступний. Тому ми обмежимося порівнянням апроксимацій, які доступні для процесу із пуассонівськими виплатами.

Зафіксуємо початкові умови: $c = 1.05\lambda\mu$, $\lambda = 10$, $X_i \sim \text{Pois}(\mu = 20)$

Як і минулого разу, розглянемо логарифмічну шкалу початкового капіталу. У таблиці 4.2 наведено ймовірності банкрутства для випадку пуассонівських виплат:

Таблиця 4.2 Порівняння апроксимацій (пуассонівські виплати)

	$u=10$	$u=100$	$u=1000$
Дифузійна апроксимація	0.9534	0.6211	0.00855
Апроксимація Крамера-Лундберга	0.9233	0.6102	0.00969
Метод Монте-Карло(1 рік)	0.924	0.604	0.008
Метод Монте - Карло(5 років)	0.934	0.608	0.007

На основі отриманих оцінок, можна дати коментарі:

- методи Монте-Карло не підходять для моделювання рідких подій. Зокрема, це видно при великому початковому капіталі. 1000 симуляцій просто не вистачає, для того щоб мати хоча б одну траєкторію банкрутства.
- коли ймовірність банкрутства досить значна ($u=10$, $u=100$), методи Монте-Карло гарно підходять, при підвищенні кількості симуляцій оцінку можна уточнювати. Але це займає багато часу!
- дифузійна апроксимація показує себе як гарне перше наближення. Якщо у величини виплат є перший та другий моменти (це не завжди так, але у випадку розподілу Пуассона виконується), то дифузійна апроксимація – гарне перше наближення.
- апроксимація Крамера-Лундберга слугує деяким уточненням дифузійної апроксимації. По-перше, уточнюється характеристичний коефіцієнт (у дифузійній апроксимації використовується його оцінка зверху). По-друге, обчислюється множник для експоненти Лундберга. Розподіл Пуассона має генератрису на $[0, \infty)$, тому обчислення не викликає проблем.
- не вдалося реалізувати формулу Бескмана. Це пояснюється природою функції розподілу величини – вона розривна та потребує постійного обчислення часткових сум. По-перше, займає багато часу обчислення самої функції розподілу, по-друге – час на чисельне інтегрування;

4.4 Дослідження випадку із гамма-розподіленими виплатами.

Для дослідження було обрано такий гамма-розподіл, щоб він збігався із початковими умовами, що були використані раніше. Зокрема:

$$f(x) = \frac{\alpha^\beta x^{\beta-1} e^{-\alpha x}}{\Gamma(\beta)}, x > 0 \quad (4.1)$$

$$EX = \frac{\beta}{\alpha}, \beta = 20, \alpha = 1 \quad (4.2)$$

Гамма-розподіл є розподілом із легким хвостом. До нього можна застосовувати апроксимацію Крамера-Лундберга, а також формулу Бескмана.

Зафіксуємо $c = 1.05\lambda\mu$, $\lambda = 10$, $X_i \sim \Gamma(\alpha = 1, \beta = 20)$

У таблиці 4.3 наведено ймовірності банкрутства на логарифмічній шкалі початкового капіталу (випадок Гамма-розподілу):

Таблиця 4.3 Порівняння апроксимацій (Гамма-розподілені виплати I)

	u=10	u=100	u=1000
Дифузійна апроксимація	0.9535	0.6211	0.0085
Апроксимація Крамера-Лундберга	0.9233	0.6101	0.0097
Метод Монте-Карло (1 рік)	0.930	0.602	0.008
Метод Монте-Карло (5 років)	0.915	0.597	0.007
Формула Бескмана (нескінченний час)	0.95	0.55	0.00

Форма гамма-розподілу сильно міняється в залежності від параметрів розподілу. Не змінюючи математичне сподівання виплати, змінимо параметри гамма - розподілу та оцінимо різницю в оцінках ймовірності:

Зафіксуємо $c = 1.05\lambda\mu$, $\lambda = 10$, $X_i \sim \Gamma(\alpha = 0.25, \beta = 5)$

Та розглянемо поведінку методів апроксимації на логарифмічній шкалі початкового капіталу. У таблиці 4.4 наведено значення ймовірності банкрутства (випадок Гамма-розподілу 2):

Таблиця 4.4 Порівняння апроксимацій (Гамма-розподілені виплати II)

	u=10	u=100	u=1000
Дифузійна апроксимація	0.9592	0.659	0.0155

Апроксимація Крамера-Лундберга	0.9251	0.6447	0.0174
Метод Монте-Карло (1 рік)	0.93	0.623	0.017
Метод Монте-Карло (5 років)	0.913	0.625	0.014
Формула Бескмана (нескінченний час)	0.87	0.62	0.03

Можна дати такі коментарі з приводу отриманих оцінок:

- апроксимація формулою Бескмана займає дуже багато часу(!). Доводиться зменшувати кількість симуляцій до 100 і жертвувати точністю оцінок;
- обидва методи Монте-Карло гарно підходять для випадків $u=10$, $u=100$, але зовсім не підходять для $u=1000$;
- теоретичні оцінки (дифузійна, Крамера-Лундберга) виглядають більш адекватними та точними у випадку великого початкового капіталу(великого – відносно середньої виплати при нещасному випадку).
- зміна параметрів розподілу майже не повпливала, числа в таблиці виглядають приблизно так само.

4.5 Дослідження процесу із Парето-розподіленими виплатами.

Розподіл Парето принципово відрізняється від розглянутих вище – він має функції розподілу із важкими хвостами. Відповідно, характеристичного коефіцієнту для нього не існує.

Нагадаємо, ми використовуємо двохпараметричний розподіл Парето, що починається з нуля:

$$f(x) = \alpha \frac{x_m^\alpha}{x^{\alpha+1}}, x > 0 \quad (4.3)$$

$$EX = \frac{\alpha x_m}{\alpha - 1} \quad (4.4)$$

Залежно від параметрів розподілу, форма щільності та форма хвоста функції розподілу сильно відрізняється. На рисунку 4.4 можемо бачити як змінюється хвіст функції розподілу у Парето-розподіленій величини в залежності від параметрів розподілу.

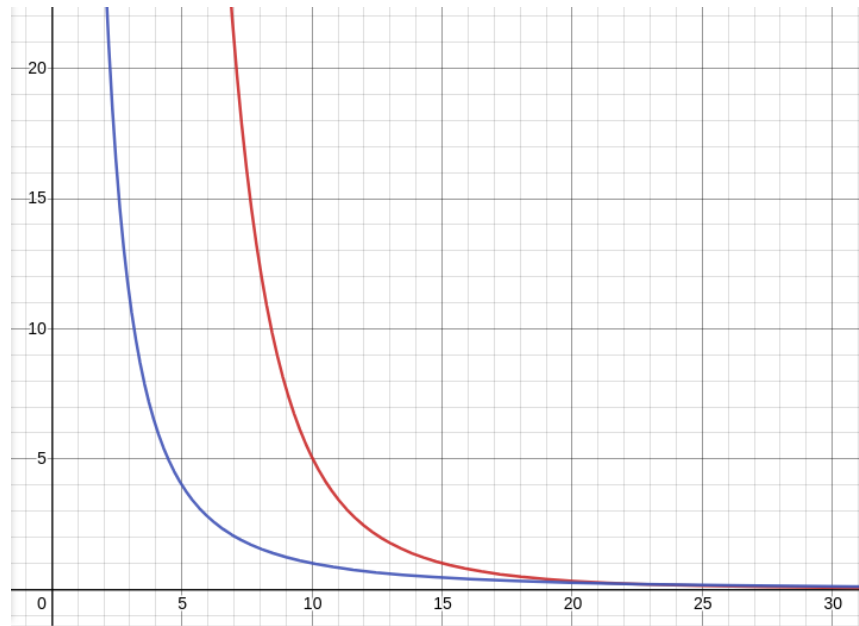


Рисунок 4.4 Порівняння хвостів функції розподілу Парето.

$\alpha = 2, x_m = 10$ – синя крива, $\alpha = 4, x_m = 15$ – червона крива.

Із графіка зрозуміло, що синя крива має важчі хвости розподілу. Це проявляється принаймні у тому, що за параметрів $(2,10)$ випадкова величина не має другого моменту.

Зафіксуємо початкові умови: $c = 1.05\lambda\mu$, $\lambda = 10$, $X_i \sim \text{Pareto}(\alpha = 4, x_m = 15)$.

У таблиці 4.5 наведено результати апроксимацій, які доступні для розподілу Парето $(4,15)$:

Таблиця 4.5 Порівняння апроксимацій (Парето-виплати I)

	u=10	u=100	u=1000
Метод Монте-Карло (1 рік)	0.929	0.634	0.015

Метод Монте-Карло (5 років)	0.924	0.617	0.017
Формула Бєєкмана(нескінченний час)	0.9	0.68	0.01

Розглянемо розподіл Парето із таким же математичним сподіванням виплати ($\mu = 20$), але змінимо його параметри (в бік важчих хвостів розподілу).

Зафіксуємо $c = 1.05\lambda\mu$, $\lambda = 10$, $\mu = 20$, $X_i \sim \text{Pareto}(\alpha = 2, x_m = 10)$

Зауважимо, що при таких параметрах розподілу Парето, випадкова величина не має другого моменту.

У таблиці 4.6 наведено результати апроксимацій, які доступні для розподілу Парето (2,10):

Таблиця 4.6 Порівняння апроксимацій (Парето-виплати II)

	u=10	u=100	u=1000
Метод Монте-Карло (1 рік)	0.916	0.733	0.2
Метод Монте-Карло (5 років)	0.912	0.751	0.232
Формула Бєєкмана (нескінченний час)	0.91	0.73	0.22

Із отриманих значень можна зробити висновки:

- формула Бєєкмана та методи Монте-Карло не підходять для ситуацій із великим початковим капіталом.
- розподіл Парето із важкими хвостами більше підходить для страхування пов'язаного із життям, ніж для страхування майна тощо.
- у випадку $\alpha > 2$ можна використовувати дифузійну апроксимацію.
- інтегровне перетворення не обчислюється аналітично для розподілу Парето, а тому симуляція формули Бєєкмана займає занадто багато часу.

– значення ймовірностей банкрутства сильно змінилися при зміні параметрів розподілу. Звичайно, процес із важкими хвостами зазнає банкрутства частіше, ніж аналогічний процес із легкими хвостами (при рівних характеристиках процесу Пуассону, початковому капіталі та коефіцієнті безпечного навантаження).

4.6 Дослідження процесу із Вейбулл-розподіленими виплатами.

Нагадаємо, щільність розподілу Вейбулла та числові характеристики мають вигляд:

$$f(x) = \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-\left(\frac{x}{\lambda}\right)^k}, x > 0 \quad (4.5)$$

$$EX = \lambda \Gamma\left(1 + \frac{1}{k}\right) \quad (4.6)$$

На рисунку 4.5 зображено хвости функції розподілу Вейбулла за різних параметрів розподілу:

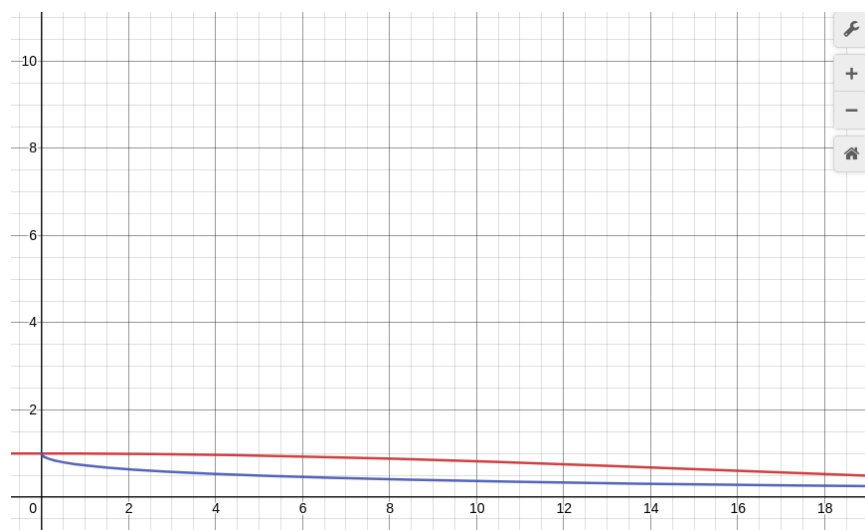


Рисунок 4.5 Порівняння хвостів розподілу Вейбулла.

$Weibull(k = 0.5, \lambda = 10)$ – синя крива, $Weibull\left(k = 2, \lambda = \frac{20}{\Gamma(1.5)}\right)$ – червона.

Із рисунку зрозуміло, що синя крива ($k=0.5$) має тяжчі хвости розподілу.

Випадок $k=1$ породжує експоненційні виплати. Тому звичайно, його не розглядаємо. До речі, випадок $k=2$ називається розподілом Релея.

$k>1$ – розподіл із легким хвостом;

$k<1$ – розподіл із важким хвостом.

Розглянемо два різні випадки розподілу Вейбулла – при $k=2$ та при $k=0.5$.

Зафіксуємо початкові умови:

$$c = 1.05\lambda\mu, \lambda = 10, \mu = 20, X_i \sim Weibull(k = 0.5, \lambda = 10)$$

Розглянемо результати апроксимацій, які доступні для розподілу Вейбулла(10, 0.5). У таблиці 4.7 наведено ймовірності банкрутства:

Таблиця 4.7 Порівняння апроксимацій (Вейбулл-виплати I)

	$u=10$	$u=100$	$u=1000$
Метод Монте-Карло (1 рік)	0.937	0.87	0.401
Метод Монте-Карло (5 років)	0.95	0.877	0.446
Формула Бескмана (нескінченний час)	0.94	0.82	0.41
Дифузійна апроксимація	0.9917	0.9200	0.4346

Зафіксуємо інші початкові умови для розподілу Вейбулла:

$$c = 1.05\lambda\mu, \lambda = 10, \mu = 20, X_i \sim Weibull(k = 2, \lambda \approx 22.5)$$

Розглянемо результати апроксимацій, які доступні для розподілу Вейбулла:

Таблиця 4.8 Порівняння апроксимацій (Вейбулл – виплати II)

	$u=10$	$u=100$	$u=1000$
Метод Монте-Карло	0.929	0.69	0.029

(1 рік)			
Метод Монте-Карло (5 років)	0.934	0.664	0.027
Формула Бескмана (нескінченний час)	0.9	0.66	0.04
Дифузійна апроксимація	0.9615	0.6752	0.0197

4.7 Висновки до розділу 4:

- дифузійна апроксимація досить гарно працює при великому стартовому капіталі ($u=1000$, $u=10000\dots$). Фактично її використовують як гарне перше наближення, зокрема при пошуку характеристичного коефіцієнту.
- якщо порівнювати легкі та важкі розподіли в цілому, то видно що ймовірності банкрутства у важких розподілів вищі. Поясненням цього ефекту є поява виплат великого розміру, яка не характерна для розподілів із легким хвостом.
- у випадку розподілу Вейбулла є чітка ієрархія: $(k=0.5) > (k=1) > (k=2)$. Розподіл Релея дає менші ймовірності банкрутства у порівнянні із експоненційним. Відповідно, розподіл $k=0.5$ дає більші ймовірності банкрутства, ніж експоненційний та розподіл Релея;
- формула Бескмана є дуже точним методом оцінки ймовірності банкрутства на нескінченному проміжку часу. Однак, інтегральне перетворення за допомогою чисельного інтегрування займає багато часу, що робить велику кількість симуляцій просто неможливою.
- прямий метод Монте-Карло доцільно застосовувати, коли відомо що ймовірність банкрутства є досить значною ($u=10$, $u=100$).

РОЗДІЛ 5 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

У даному розділі проводиться оцінка основних характеристик програмного продукту, розробленого для комп'ютерного моделювання страхової компанії, обчислення ймовірності її банкрутства.

Нижче наведено аналіз різних варіантів реалізації модулю з метою вибору оптимальної, з огляду при цьому як на економічні фактори, так і на характеристики продукту, що впливають на продуктивність роботи і на його сумісність з апаратним забезпеченням. Для цього було використано апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) – це технологія, яка дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. ФВА проводиться з метою виявлення резервів зниження витрат за рахунок ефективніших варіантів виробництва, кращого співвідношення між споживчою вартістю виробу та витратами на його виготовлення. Для проведення аналізу використовується економічна, технічна та конструкторська інформація.

Алгоритм функціонально-вартісного аналізу включає в себе визначення послідовності етапів розробки продукту, визначення повних витрат (річних) та кількості робочих часів, визначення джерел витрат та кінцевий розрахунок вартості програмного продукту.

5.1 Постановка задачі проектування

У роботі застосовується метод ФВА для проведення техніко-економічного аналізу розробки системи для комп'ютерного моделювання страхової компанії за різних розподілів виплат. Оскільки рішення стосовно проектування та реалізації компонентів, що розробляється, впливають на всю систему, кожна окрема підсистема має її задовольняти.

Технічні вимоги до програмного продукту є наступні:

- функціонування на персональних комп'ютерах із стандартним набором компонентів;
- зручність та зрозумілість для користувача;
- швидкість обробки даних та доступ до інформації в реальному часі;
- можливість зручного масштабування та обслуговування;
- мінімальні витрати на впровадження програмного продукту.

5.2 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка програмного продукту, який дозволяє аналізувати математичну модель страхової компанії при різних ймовірнісних розподілах виплат. Беручи за основу цю функцію, можна виділити наступні:

F_1 – вибір мови програмування;

F_2 – вибір статистичних бібліотек;

F_3 – вибір середовища розробки.

Кожна з цих функцій має декілька варіантів реалізації:

Функція F_1 :

а) Python;

б) R.

Функція F_2 :

а) numpy, pandas, scipy, matplotlib, tqdm;

б) dplyr, datatable, ggplot, actuar, tidyverse;

Функція F_3 :

а) Jupyter Notebook;

б) PyCharm.

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 5.1).

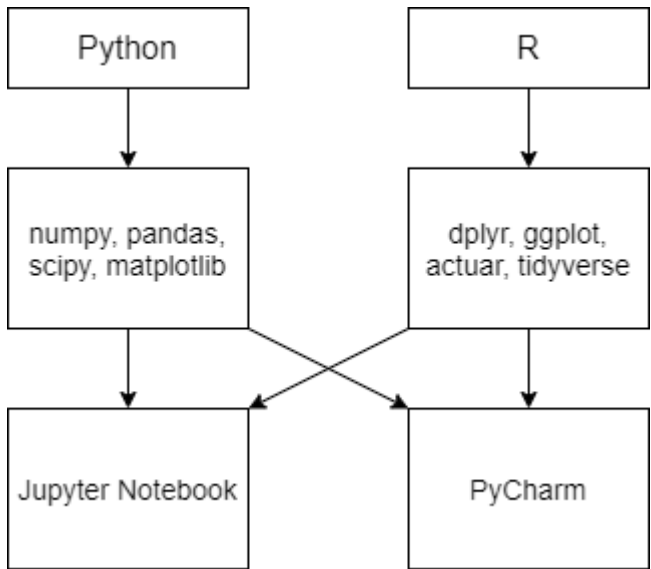


Рисунок 5.1 – Морфологічна карта

Морфологічна карта відображає множину всіх можливих варіанти основних функцій.

Таблиця 5.2 - Позитивно-негативна матриця

Функції	Варіанти реалізації	Переваги	Недоліки
F_1	<i>A</i>	Швидка розробка програми, доступність бібліотек. Універсальність мови програмування	Низька швидкість роботи, особливо, якщо потрібно обробляти велику кількість даних
	<i>B</i>	Доступність бібліотек, статистичне ядро мови	Іде багато часу на розробку програми.
F_2	<i>A</i>	Ефективні обчислення, використання багатопоточності	Підтримується лише мовою Python
	<i>B</i>	Надійність, висока швидкість роботи з великими даними.	Додатковий час на інсталяцію та вивчення

F_3	A	Легко запускається на будь-якому сервері. Комфортна робота із графіками та статистикою.	Не підходить для великих проектів
	B	Багато інструментів. Надійна робота із великими проектами	Підтримує одночасно лише одну мову програмування

На основі цієї карти будемо позитивно-негативну матрицю варіантів основних функцій (Таб.5.2). Робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція F_1 :

Перевагу віддаємо швидкості вивчення, простоті використання та наявності стандартних бібліотек для обчислення. Для мови Python більш доступні інструкції та навчальні матеріали, тому варіант Б має бути відкинутий.

Функція F_2 :

Параметр однозначно визначається мовою програмування, тому при виборі мови Python – варіант А, при виборі мови R – варіант Б.

Функція F_3 :

Віддаємо перевагу варіанту А на користь швидкодії.

Таким чином, будемо розглядати такі варіанти реалізації ПП:

$F_1a - F_2a - F_3a$

$F_1b - F_2b - F_3a$

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

5.3 Обґрунтування системи параметрів ПП

На основі даних, розглянутих вище, визначаються основні параметри вибору, які будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- X1 – швидкодія мови програмування;
- X2 – об’єм пам’яті для обчислень та збереження даних;
- X3 – час моделювання при базових параметрах;
- X4 – потенційний об’єм програмного коду.

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію ПП як показано у таблиці 5.3

Таблиця 5.3 - Основні параметри ПП

Назва Параметра	Умовні позначе ння	Одиниці виміру	Значення параметра		
			Гірші	середні	кращі
Швидкодія мови програмування	X1	оп/мс	9000	15000	20000
Об’єм пам’яті	X2	Мб	500	250	100
Час моделювання при базових параметрах	X3	С	1200	900	600
Потенційний об’єм програмного коду	X4	кількість рядків коду	500	400	300

За даними таблиці 5.3 будуються графічні характеристики параметрів – рис. 5.2 – рис. 5.5.

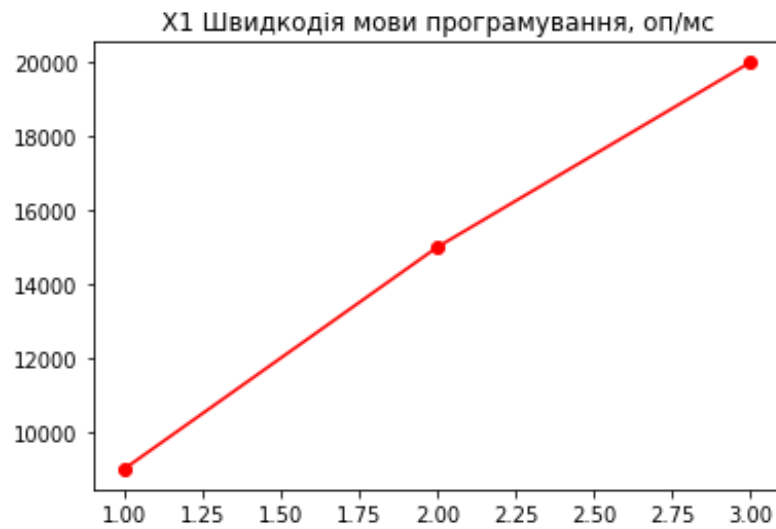


Рисунок 5.2 – X1, швидкодія мови програмування

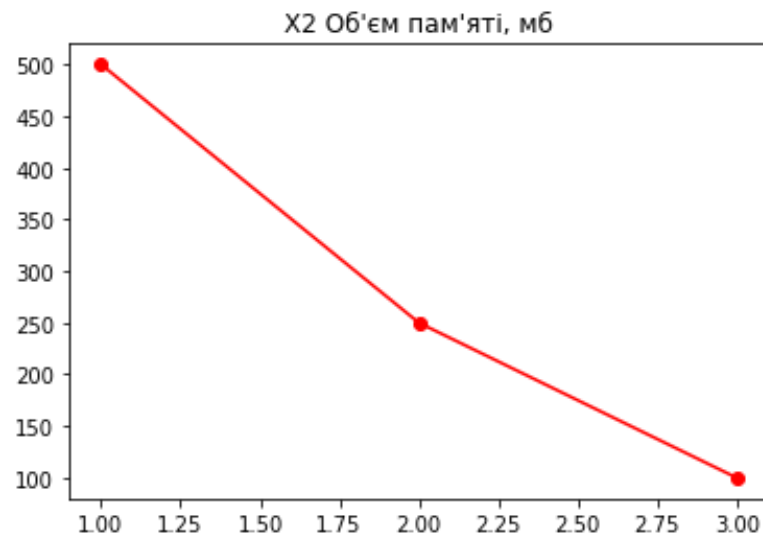


Рисунок 5.3 – X2, об'єм пам'яті

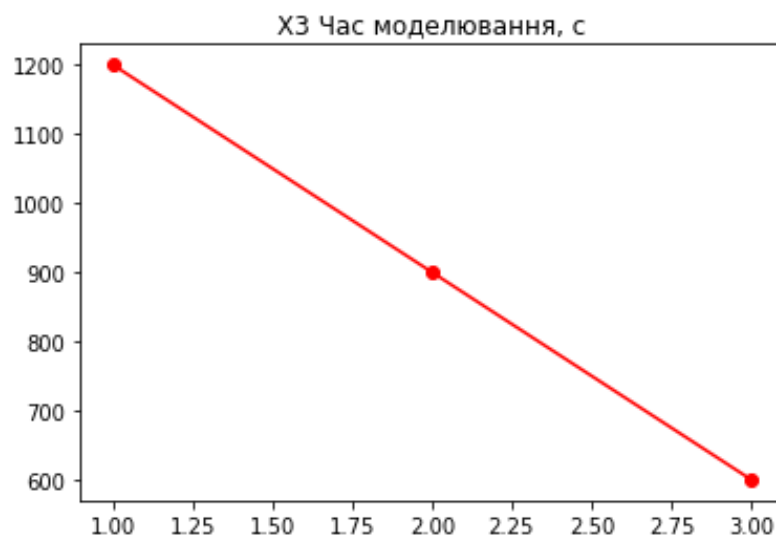


Рисунок 5.4 – X3, час попередньої обробки даних

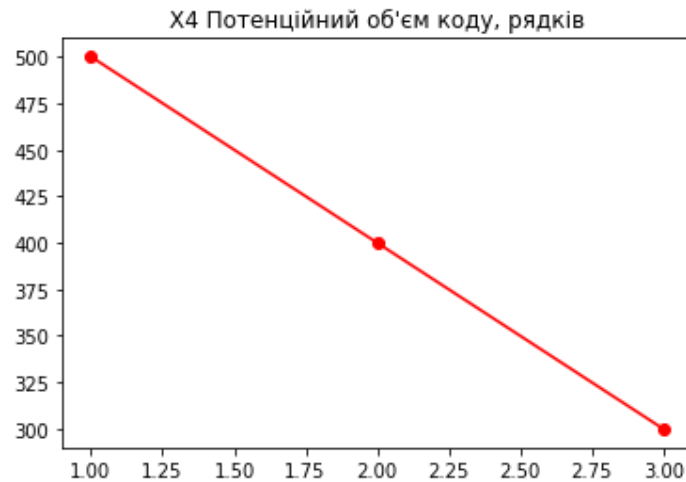


Рисунок 5.5 – X4, потенційний об'єм програмного коду

5.4 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні результати при знаходженні параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 5.4.

Таблиця 5.4 - Результати ранжування параметрів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
$X1$	Швидкодія мови програмування	Оп/мс	3	4	3	4	3	4	3	24	1,5	2,25
$X2$	Об'єм пам'яті	Мб	1	2	2	2	1	2	2	12	-10,5	110,25
$X3$	Час моделювання	С	3	4	2	4	3	4	2	22	0,5	0,25
$X4$	Потенційний об'єм програмного коду	Кількість рядків коду	5	4	5	4	5	5	4	32	9,5	90,25
	Разом		12	14	12	14	12	15	11	90	0	203

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 90, \quad (5.1)$$

де N – число експертів,
 n – кількість параметрів;
 б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 22,5 \quad (5.2)$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T. \quad (5.3)$$

Сума відхилень по всіх параметрах повинна дорівнювати 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 203. \quad (5.4)$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 203}{7^2(4^3 - 4)} = 0,83 > W_k = 0,67. \quad (5.5)$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 5.5.

Таблиця 5.5 - Попарне порівняння параметрів.

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	>	>	>	>	>	>	>	>	1,5
X1 і X3	=	=	<	=	=	=	>	=	1,0
X1 і X4	<	=	<	=	<	<	<	<	0,5
X2 і X3	<	<	=	<	<	<	=	<	0,5
X2 і X4	<	<	<	<	<	<	<	<	0,5
X3 і X4	<	=	<	=	<	<	<	<	0,5

Числове значення, що визначає ступінь переваги i -го параметра над j -тим, a_{ij} визначається по формулі:

$$a_{ij} = \begin{cases} 1.5 \text{ при } X_i > X_j \\ 1.0 \text{ при } X_i = X_j \\ 0.5 \text{ при } X_i < X_j \end{cases} \quad (5.6)$$

З отриманих числових оцінок переваги складемо матрицю $A = \| a_{ij} \|$.

Для кожного параметра зробимо розрахунок вагомості K_{gi} за наступними формулами:

$$K_{Bi} = \frac{b_i}{\sum_{i=1}^n b_i} \quad (5.7)$$

$$b_i = \sum_{j=1}^N a_{ij} \quad (5.8)$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{Bi} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \quad (5.9)$$

$$b'_i = \sum_{j=1}^N a_{ij} b_j \quad (5.10)$$

Як видно з таблиці 5.6, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 5.6 - Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітерація		Друга ітерація		Третя ітерація	
	X1	X2	X3	X4	b_i	K_{bi}	b_i^1	K_{bi}^1	b_i^2	K_{bi}^2
X1	1,0	1,5	1,0	0,5	4,0	0,25	14,5	0,244	53,5	0,243
X2	0,5	1,0	0,5	0,5	2,5	0,156	9,25	0,155	34,375	0,156
X3	1,0	1,5	1,0	0,5	4,0	0,25	14,5	0,244	53,5	0,243
X4	1,5	1,5	1,5	1,0	5,5	0,344	21,25	0,357	78,625	0,357
Всього:					16,0	1	59,25	1	220,0	1

5.5 Аналіз рівня якості варіантів реалізації функцій

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 5.7):

$$K_K(j) = \sum_{i=1}^n K_{ei,j} B_{i,j}, \quad (5.11)$$

де n – кількість параметрів;

K_{ei} – коефіцієнт вагомості i -го параметра;

B_i – оцінка i -го параметра в балах.

Таблиця 5.7 - Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Параметри	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	А	X1	15000	6	0,243	1,458
	Б	X1	9000	4	0,243	0,972
F2	А	X3	500	4	0,243	0,972
	Б	X3	500	4	0,243	0,972

F3	A	X4	400	5	0,357	1,785
----	---	----	-----	---	-------	-------

За даними з таблиці 5.7 за формулою:

$$K_K = K_{\text{ТУ}}[F_{1k}] + K_{\text{ТУ}}[F_{2k}] + \dots + K_{\text{ТУ}}[F_{zk}], \quad (5.12)$$

визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 1,458 + 0,972 + 1,785 = 4,215,$$

$$K_{K2} = 0,972 + 0,972 + 1,785 = 3,729.$$

Як видно з розрахунків, кращим є перший варіант, для якого коефіцієнт технічного рівня має найбільше значення.

5.6 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе два окремих завдання:

1. Розробка проекту програмного продукту;
2. Розробка частини візуалізації;

Обидва завдання відносяться до другої групи складності, а за ступенем новизни – до групи Б.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Загальна трудомісткість обчислюється як

$$T_0 = T_P \cdot K_P \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}, \quad (5.13)$$

де T_P – трудомісткість розробки ПП;

K_{Π} – поправочний коефіцієнт;

$K_{СК}$ – коефіцієнт на складність вхідної інформації;

K_M – коефіцієнт рівня мови програмування;

$K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм;

$K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру степеню новизни Б та групи складності алгоритму 2, трудомісткість дорівнює: $T_P = 15$ людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання: $K_{\Pi} = 1.5$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх семи завдань рівний 1: $K_{СК} = 1$. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{СТ} = 0.8$. Тоді загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 15 \cdot 1.5 \cdot 0.8 = 18 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для подальших завдань.

Для другого завдання $T_P = 15$ людино-днів, $K_{\Pi} = 1.5$, $K_{СК} = 1$. Для першого варіанту реалізації приймемо $K_{СТ} = 1.2$; для другого варіанту - $K_{СТ} = 1.4$

$$T_2^1 = 15 \cdot 1.5 \cdot 0.8 \cdot 1.2 = 21.6 \text{ людино-днів.}$$

$$T_2^2 = 15 \cdot 1.5 \cdot 0.8 \cdot 1.4 = 25.2 \text{ людино-днів.}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = (18 + 21.6) \cdot 8 = 316.8 \text{ людино-годин.}$$

$$T_{II} = (18 + 25.2) \cdot 8 = 345.6 \text{ людино-годин.}$$

Найбільш високу трудомісткість має варіант II.

В розробці беруть участь два програмісти з окладом 10972 грн., один математик в області фінансів з окладом 13715 грн. Визначимо середню зарплату за годину за формулою:

$$C_q = \frac{M}{T_m \cdot t} \text{ грн.,} \quad (5.14)$$

де M – місячний оклад працівників;
 T_m – кількість робочих днів тижень;
 t – кількість робочих годин в день.

$$C_q = \frac{10972 + 10972 + 13715}{3 \cdot 21 \cdot 8} = 70,70 \text{ грн.} \quad (5.15)$$

Тоді, розрахуємо заробітну плату за формулою:

$$C_{зп} = C_q \cdot T_i \cdot K_d, \quad (5.16)$$

де C_q – величина погодинної оплати праці програміста;
 T_i – трудомісткість відповідного завдання;
 K_d – норматив, який враховує додаткову заробітну плату.
 Зарплата розробників за варіантами становить:

$$\text{I. } C_{\text{ЗП}} = 70,7 \cdot 316,8 \cdot 1,2 = 26877,31 \text{ грн.}$$

$$\text{II. } C_{\text{ЗП}} = 70,7 \cdot 345,6 \cdot 1,2 = 29320,70 \text{ грн.}$$

Відрахування на єдиний соціальний внесок становить 22%:

$$\text{I. } C_{\text{ВІД}} = C_{\text{ЗП}} \cdot 0,22 = 26877,31 \cdot 0,22 = 5913,01 \text{ грн.}$$

$$\text{II. } C_{\text{ВІД}} = C_{\text{ЗП}} \cdot 0,22 = 29320,70 \cdot 0,22 = 6450,55 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години. (C_M)

Так як одна ЕОМ обслуговує одного програміста з окладом 10972 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$C_G = 12 \cdot M \cdot K_3 = 12 \cdot 10972 \cdot 0,2 = 26332,80 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{\text{ЗП}} = C_G \cdot (1 + K_3) = 26332,80 \cdot (1 + 0,2) = 31599,36 \text{ грн.}$$

Відрахування на соціальний внесок:

$$C_{\text{ВІД}} = C_{\text{ЗП}} \cdot 0,22 = 31599,36 \cdot 0,22 = 6951,86 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 27400 грн.

$$C_A = K_{\text{ТМ}} \cdot K_A \cdot C_{\text{ПР}} = 1,15 \cdot 0,25 \cdot 27400 = 7877,50 \text{ грн.,}$$

- де K_{TM} — коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;
 K_A — річна норма амортизації;
 $C_{ПР}$ — договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{TM} \cdot C_{ПР} \cdot K_P = 1.15 \cdot 27400 \cdot 0.05 = 1575,50 \text{ грн.},$$

де K_P — відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$T_{\text{ЕФ}} = (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B = (365 - 104 - 12 - 16) \cdot 8 \cdot 0.8 = \\ = 1491.8 \text{ годин},$$

- де D_K — календарна кількість днів у році;
 D_B, D_C — відповідно кількість вихідних та святкових днів;
 D_P — кількість днів планових ремонтів устаткування;
 t — кількість робочих годин в день;
 K_B — коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_C \cdot K_3 \cdot C_{\text{ЕН}} = 1491.8 \cdot 0,2 \cdot 0,2 \cdot 3.52 = 210,05 \text{ грн.},$$

- де N_C — середньо-споживча потужність приладу;
 K_3 — коефіцієнтом зайнятості приладу;

$\Pi_{\text{ЕН}}$ – тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_{\text{Н}} = \Pi_{\text{ГР}} \cdot 0.67 = 27400 \cdot 0,67 = 18358 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{А}} + C_{\text{Р}} + C_{\text{ЕЛ}} + C_{\text{Н}}, \quad (5.17)$$

$$C_{\text{ЕКС}} = 31599,36 + 6951,86 + 7877,50 + 5750 + 1575,50 + 210,05 + 18358 = 72322,27 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 72322,27 / 1491,8 = 48,48 \text{ грн/год.}$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_{\text{М}} = C_{\text{М-Г}} \cdot T, \quad (5.18)$$

$$\text{I.} \quad C_{\text{М}} = 48,48 \cdot 316,8 = 15358,6 \text{ грн.}$$

$$\text{II.} \quad C_{\text{М}} = 48,48 \cdot 345,6 = 16754,70 \text{ грн.}$$

Накладні витрати складають 67% від заробітної плати:

$$C_H = C_{3П} \cdot 0,67, \quad (5.19)$$

$$I. \quad C_H = 26877,31 \cdot 0,67 = 18007,80 \text{ грн.}$$

$$II. \quad C_H = 29320,70 \cdot 0,67 = 19644,87 \text{ грн.}$$

Отже, вартість розробки ПП за варіантами становить:

$$C_{ПП} = C_{3П} + C_{ВІД} + C_M + C_H, \quad (5.20)$$

$$I. \quad C_{ПП} = 26877,31 + 5913,01 + 15358,6 + 18007,80 = 66156,72 \text{ грн.}$$

$$II. \quad C_{ПП} = 29320,70 + 6450,55 + 16754,7 + 19644,87 = 72170,82 \text{ грн.}$$

5.7 Вибір кращого варіанту ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{TEPj} = K_{Kj} / C_{Фj}, \quad (5.21)$$

$$K_{TEP1} = 4,215/66156,72 = 6,37 \cdot 10^{-5},$$

$$K_{TEP2} = 3,74/72170,82 = 5,18 \cdot 10^{-5}.$$

Як бачимо, найбільш ефективним є перший варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{TEP1} = 6,37 \cdot 10^{-5}$.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишилися після першого відбору двох варіантів виконання програмного комплексу оптимальним є перший варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості $K_{\text{ТЕР}} = 6,37 \cdot 10^{-5}$.

Цей варіант реалізації програмного продукту має такі параметри:

- мова програмування – Python;
- використання набору бібліотек numpy, pandas, scipy, matplotlib
- використання середовища Jupyter Notebook, для комфортної роботи із графіками та візуалізацією.

Даний варіант виконання програмного комплексу дає користувачу непогану візуалізацію, швидкодію, можливість розширення проекту.

5.8 Висновки до розділу 5

Проведено повний функціонально-вартісний аналіз програмного продукту. Визначено та проведено оцінку основних функцій програмного продукту. Визначено параметри, які характеризують програмний продукт. Проведено експертне оцінювання параметрів та аналіз якості варіантів реалізації функцій.

Проведено економічний аналіз варіантів розробки – трудомісткість, витрати на заробітну плату та інші витрати.

На основі аналізу вибрано варіант реалізації програмного продукту.

ВИСНОВКИ ДО РОБОТИ

Основи математичної теорії ризику були покладені у 1930 році. Модель Крамера – Лундберга є основною, на її базі будуються будь-які сучасні моделі страхового ризику.

Класичну модель ризику можна узагальнювати довільним чином, зокрема деякі можливі напрями описані в розділі 1. У даній роботі ми провели дослідження саме моделі Крамера – Лундберга, та методів апроксимації ймовірності банкрутства.

На основі комп'ютерного моделювання траєкторій із різними розподілами виплат добре видно, що:

- в цілому ймовірності банкрутства вищі в моделях із важкими хвостами;
- апроксимація Крамера-Лундберга ефективно працює у випадку великого стартового капіталу;
- реалізація формули Бескмана є ускладненою – обчислення інтегрального перетворення часто займає занадто багато часу. Однак, даний метод дає більш точні оцінки до аналітичного розв'язку;
- актуарна література(зокрема, [1] та [3]) вказує на те, що більшість страхових процесів мають все таки розподіли із важким хвостом.
- методи Монте – Карло та формула Бескмана зовсім не підходять для моделювання рідких подій. Для рідких подій розроблена навіть своя теорія(теорія екстремального значення).
- фактично, прямий метод Монте – Карло та формула Бескмана оцінюють різні величини. Але, 1 рік – 5 років можна вважати досить великим часовим горизонтом, тому часто ці величини є дуже близькими за значенням.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Asmussen, S., Albrecher, H. Ruin Probabilities (2-nd edition). World Scientific, Singapore, 2010. 620 p.
- 2) Grandell, J. Aspects of Risk Theory. Springer, New York, 1992. 185 p.
- 3) Mikosch, T. Non-Life insurance mathematics. Springer-Verlag, Berlin, 2009. 432 p.
- 4) Ільєнко, А. Б., Ільєнко, М. К. Елементи теорії випадкових процесів. Київ, 2019. 172 с.
- 5) De Vylder, F. E. A practical solution to the problem of ultimate ruin probability. Scandinavian Actuarial Journal, 1978. 114–119 pp.;
- 6) Grandell, J., Simple approximations of ruin probabilities. Insurance: Mathematics and Economics, 26(2-3):157–173 pp.;
- 7) Beekman, J. A., A series for infinite time ruin probabilities. Insurance: Mathematics and Economics, 4(2):129–134 pp.
- 8) Burneck, K., Mista, P., Veron A. Ruin probabilities for finite and infinite time;
- 9) Dickson, D.C.M. and Waters, H.R. Recursive calculation of survival probabilities. ASTIN Bulletin 21, 199-221 pp.;
- 10) Egidio Dos Reis, A.D. On the moments of ruin and recovery times. Insurance: Mathematics & Economics 27, 331-344 pp.;
- 11) Dickson, D.C.M. and Waters, H.R. The distribution of the time to ruin in the classical risk model. ASTIN Bulletin 32, 299-313 pp.;
- 12) Embrechts, P., Klüppelberg, C., Mikosch, T. Modelling extremal events for Insurance and Finance, Springer-Verlag Berlin Heidelberg;
- 13) Borovkov, K.A., Dickson, D. C. M. On the ruin time distribution for a Sparre Andersen process with exponential claim sizes. Insurance: Mathematics & Economics 42, 1104-1108 pp.;
- 14) Parna, K., Risk theory. Tartu: University of Tartu, 60p.;
- 15) Ширяев, А. Н. Основы стохастической финансовой математики. МЦНМО, 1998, 620 с.;

- 16) Kalashnikov, V., Geometric Sums: Bounds for Rare Events with Applications. Kluwer Academic Publishers, Dordrecht, 1997.
- 17) Beekman, J. A., A ruin function approximation. Transactions of the Society of Actuaries 21, 41–48, 275–279.
- 18) Cramér, H., On the Mathematical Theory of Risk. Skandia Jubilee Volume, Stockholm. In: Martin-Löf, A. (Ed.), Harald Cramér Collected Works, Vol. I. Springer, Berlin, 1994, 601–678pp..
- 19) Cramér, H., 1955. Collective Risk Theory. Skandia Jubilee Volume, Stockholm. In: Martin-Löf, A. (Ed.), Harald Cramér Collected Works, Vol. II. Springer, Berlin, 1994, 1028–1115 pp.

Додаток А. Лістинг програмного модулю

Модуль tools.py:

```
#!/usr/bin/env python
```

```
# coding: utf-8
```

```
# In[1]:
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import scipy.stats
```

```
from scipy.integrate import quad
```

```
from scipy.special import gamma
```

```
from tqdm import tqdm
```

```
# In[2]:
```

```
class Poisson:
```

```
    def __init__(self, mu):
```

```
        self.mu=mu
```

```
    def cdf(self,x):
```

```
        return scipy.stats.poisson.cdf(x, mu=self.mu)
```

```
    def rvs(self,size=1):
```

```
        return np.random.poisson(lam=self.mu, size=size)
```

```

def mean(self):
    return self.mu
def var(self):
    return self.mu
def generatrissa(self,h):
    return np.exp(self.mu*(np.exp(h)-1))

```

In[3]:

```

class Exponential:
    def __init__(self, lambd, x0=0):
        self.lambd=lambd
        self.x0=x0
    def cdf(self,x):
        return scipy.stats.expon.cdf(x-self.x0, scale=1/self.lambd)
    def rvs(self,size=1):
        return self.x0 + np.random.exponential(scale=1/self.lambd)
    def mean(self):
        return self.x0+1.0/self.lambd
    def var(self):
        return (1.0/self.lambd)**2
    def generatrissa(self,h):
        return np.exp(self.x0*h)*(1.0/(1-h/self.lambd))

```

In[4]:

```

class Gamma:
    def __init__(self, alpha, beta):
        self.alpha=alpha
        self.beta=beta
    def cdf(self,x):
        return scipy.stats.gamma.cdf(x,a=self.beta,scale=1.0/self.alpha)
    def rvs(self,size=1):
        return np.random.gamma(shape=self.beta, scale=1.0/self.alpha,size=size)
    def mean(self):
        return self.beta/self.alpha
    def var(self):
        return self.beta/(self.alpha**2)
    def generatrissa(self,h):
        if(h<self.alpha):
            return pow(1-h/self.alpha, -self.beta)
        else:
            return np.inf

```

In[5]:

```

class Pareto:
    def __init__(self,alpha,x_m):
        self.alpha = alpha
        self.x_m = x_m
    def cdf(self, x):
        return scipy.stats.pareto.cdf(x,scale=self.x_m, b = self.alpha)
    def rvs(self, size=1):
        #return np.random.pareto(a=self.alpha, scale = self.x_m)

```

```

    return scipy.stats.pareto.rvs(scale=self.x_m, b = self.alpha)
def mean(self):
    if(self.alpha<=1):
        return np.inf
    else:
        return (self.alpha*self.x_m)/(self.alpha-1.0)
def var(self):
    if(self.alpha<=2):
        return np.inf
    else:
        return (self.alpha*self.x_m**2)/((self.alpha-1)**2)*(self.alpha-2) )
def moment(self,n):
    return (self.alpha*(self.x_m**n))/(self.alpha - n)

```

In[6]:

```

class Weibull:
    def __init__(self, k, lambd):
        self.k = k
        self.lambd = lambd
    def cdf(self,x):
        return scipy.stats.weibull_min.cdf(x,c=self.k, scale = self.lambd)
    def rvs(self,size=1):
        return scipy.stats.weibull_min.rvs(c=self.k,scale=self.lambd,size=size)
    def mean(self):
        return self.lambd*gamma( 1 + 1.0/self.k)
    def var(self):

```

```

    return (self.lambd**2)*gamma(1+2.0/self.k) - (self.lambd*gamma( 1 +
1.0/self.k))**2

```

```

# In[7]:

```

```

class RiskProcess:

```

```

    def __init__(self, u, rho, intensity, claim_rv):

```

```

        self.u=u

```

```

        self.rho=rho

```

```

        self.intensity=intensity

```

```

        self.claim_rv = claim_rv

```

```

        self.time_rv = Exponential(intensity)

```

```

        self.claim_mean = claim_rv.mean()

```

```

        self.c = (1+rho)*intensity*claim_rv.mean()

```

```

        print("Cramer - Lundbergh process:")

```

```

        print(f"u={self.u} - initial capital")

```

```

        print(f"rho={self.rho} - safety loading coefficient")

```

```

        print(f"intensity={self.intensity} - intensity of accidents(Poisson process)")

```

```

        print(f"\nmean function:

```

```

{self.u}+{self.rho*self.intensity*self.claim_mean}*t")

```

```

    def mean_function(self, t):

```

```

        return self.u+self.c*t-self.intensity*t*self.claim_mean

```

```

    #modelling function- returns lists of times and vyplaty

```

```

    def modelling(self, timelimit):

```

```

        time=0

```

```

capital=self.u
time_list=[]
vyplaty_list=[]
while(True):
    tau_i=self.time_rv.rvs()
    time+=tau_i
    if(time>timelimit):
        break
    X_i=self.claim_rv.rvs()
    capital=capital+self.c*tau_i-X_i
    time_list.append(time)
    vyplaty_list.append(X_i)
    if(capital<0):
        break
return (time_list, vyplaty_list, capital)
#drawing function. each vyplata has 2 points(before and after)
def draw(self, timelimit):
    time_list, vyplaty_list, _ = self.modelling(timelimit)
    n=len(time_list)#count of vyplata
    capitals=[self.u]
    suma_vyplat=0
    times=[0]
    for i in range(n):
        capitals.append(self.u+self.c*time_list[i]-suma_vyplat)
        suma_vyplat+=vyplaty_list[i]
        capitals.append(self.u+self.c*time_list[i]-suma_vyplat)
        times.append(time_list[i])
        times.append(time_list[i])
    if(capitals[-1]>0):
        capitals.append(self.u+self.c*timelimit-suma_vyplat)

```

```

        times.append(timelimit)
    plt.xlim(0,int(timelimit)+1)
    plt.xlabel('Время(t)')
    plt.ylabel('U(t)')
    plt.plot(np.arange(int(timelimit)+1),[0]*(int(timelimit)+1),linestyle='--', color
= 'black')
    plt.plot(times,capitals)

```

In[]:

```

def montecarlo_draw(RP, n_paths,timelimit):
    plt.figure(figsize=(15,8))
    plt.plot(np.arange(int(timelimit)+1),
RP.mean_function(np.arange(int(timelimit)+1)), color='black')
    for i in tqdm(range(n_paths)):
        RP.draw(timelimit)

```

In[]:

```

def diffusion_approx(RP):
    return np.exp(-2*RP.rho*RP.claim_mean*RP.u/( RP.claim_mean**2 +
RP.claim_rv.var() ))

```

In[]:

```

def heavytraffic_approx(RP):
    def f_X_hat(z):
        return (1/RP.claim_mean)*quad(lambda t: 1.0-RP.claim_rv.cdf(t) , 0, z)[0]
    class IntegratedDist(scipy.stats.rv_continuous):
        def _cdf(self,z):
            return f_X_hat(z)
    approx = (1.0 - IntegratedDist().cdf(RP.u))/RP.rho
    print(f"Heavy-traffic approx:  $\psi(u) \sim \{ \text{approx} \}$ ")
    return approx

```

In[]:

```

from scipy.optimize import root_scalar
def characteristic_coef(RP):
    R = root_scalar(f=(lambda R: RP.claim_rv.generatrissa(R) - 1 -
RP.c*R/RP.intensity),          fprime=(lambda R: (RP.claim_rv.generatrissa(R+1e-
8)-RP.claim_rv.generatrissa(R-1e-8))/2e-8 -RP.c/RP.intensity),          x0 =
2*RP.rho*RP.claim_mean/( RP.claim_mean**2 + RP.claim_rv.var() ),
        method='newton').root
    print(f"Lundbergh inequality:  $\psi(u) \leq \exp(-Ru) = \{ \text{np.exp}(-R*RP.u) \}$ ")
    return R

```

In[]:

```

def cramer_lundberg_approx(RP):

```

```

R = characteristic_coef(RP)
exp = np.exp(-R*RP.u)
numerator = RP.rho*RP.claim_mean
denominator = (RP.claim_rv.generatrissa(R+1e-8)-RP.claim_rv.generatrissa(R-
1e-8))/2e-8 - RP.c/RP.intensity
approx = exp*numerator/denominator
print(f"Cramer-Lundbergh approx:  $\psi(u) = \{approx\}$ ")
return approx

```

In[]:

```

def analytic_prob_exp(RP):
    approx = (1.0/(1+RP.rho))*np.exp(-RP.rho*RP.u/(RP.claim_mean*(1+RP.rho)))
    return approx

```

In[]:

```

def rough_montecarlo(RP, n_experiments, time_limit):
    #rough Monte-Carlo method
    n_bankrots = 0
    for i in tqdm(np.arange(n_experiments)):
        temp=RP.modelling(time_limit)
        if(temp[2]<=0):
            n_bankrots+=1

    p = n_bankrots/n_experiments

```

$q = 1-p$

```
# p_hat ~ N(p, pq/N). 95% conf. interval:
print(f"psi(u)={p}")
left_bound = p-2*np.sqrt(p*q/n_experiments)
right_bound = p+2*np.sqrt(p*q/n_experiments)
print(f"95% normal conf. interval:
{np.round(left_bound,4)}<psi(u)<{np.round(right_bound,4)}")
return p
```

In[]:

```
def beekman_montecarlo(RP, n_experiments):
    n_bankrots=0
    def f_X_hat(z):
        return (1/RP.claim_mean)*quad(lambda t: 1.0-RP.claim_rv.cdf(t) , 0, z)[0]
    class IntegratedDist(scipy.stats.rv_continuous):
        def _cdf(self,z):
            return f_X_hat(z)

    p=1.0-RP.intensity*RP.claim_mean/RP.c
    for i in tqdm(np.arange(n_experiments)):
        N=np.random.geometric(p)-1
        if(N==0):
            continue
        S=IntegratedDist().rvs(size=N).sum()
        if(S>RP.u):
            n_bankrots+=1
```

```

p = n_bankrots/n_experiments
q = 1-p

# p_hat ~ N(p, pq/N). 95% conf. interval:
print(f"psi(u)={p}")
left_bound = p-2*np.sqrt(p*q/n_experiments)
right_bound = p+2*np.sqrt(p*q/n_experiments)
print(f"95% conf. normal interval:
{np.round(left_bound,4)}<psi(u)<{np.round(right_bound,4)}")
return p

```

Додаток Б. Презентація

Оцінка ризику страхової компанії на основі моделі Крамера-Лундберга

Виконав: Крохальов Іван

Керівник: доцент кафедри ММСА, к. ф.-м. н.

Каніовська Ірина Юріївна

Activate Windows
Go to Settings to activate Windows.



Актуальність дослідження

- Задача обчислення ймовірності банкрутства є провідною у розрахунках страхової компанії;
- Модель Крамера – Лундберга є базовою для всіх подальших розробок;
- Відсутність модулів для актуарних розрахунків у мові програмування Python.

Activate Windows
Go to Settings to activate Windows.

- **Об'єкт дослідження:** математична модель капіталу страхової компанії (модель Крамера – Лундберга);
- **Предмет дослідження:** ймовірність банкрутства при різних розподілах виплат;
- **Мета дослідження:** проаналізувати поведінку процесу ризику, розробити бібліотеку для моделювання процесу ризику.

Activate Windows
Go to Settings to activate Windows.

Постановка задачі

- Проаналізувати поведінку процесу страхового ризику при різних розподілах виплат;
- Проаналізувати теоретичні методи апроксимації ймовірності банкрутства;
- Порівняти їх із методами Монте – Карло (за різних початкових умов процесу страхового ризику);
- Візуалізація результатів.

Activate Windows
Go to Settings to activate Windows.

Модель Крамера - Лундберга

Випадковий процес $U(t)$ описує капітал страхової компанії у момент часу t :

$$U(t) = u + ct - \sum_{i=1}^{N(t)} X_i, t \geq 0$$

u – стартовий капітал компанії;

c – інтенсивність надходжень;

$N(t)$ – потік нещасних випадків. $N(t) \sim Pois(\lambda t)$ - процес Пуассона.

$X_1, X_2, \dots, X_n$ - випадкові величини, що описують виплати. $EX_i = \mu$

Незалежні в сукупності, однаково розподілені.

Activate Windows
Go to Settings to activate Windows.

Типи розподілів виплат

Виплата має розподіл із легким хвостом, якщо:

$$\exists \beta > 0: \lim_{\{x \rightarrow \infty\}} \frac{1 - F(x)}{\exp\{-\beta x\}} = c < \infty$$

До розподілів із легкими виплатами належать експоненційний, гамма-розподіл, усічений нормальний розподіл.

Виплата має розподіл із важким хвостом, якщо:

$$\forall \beta > 0: \lim_{\{x \rightarrow \infty\}} \frac{1 - F(x)}{\exp\{-\beta x\}} = \infty$$

До розподілів із важкими виплатами належать Парето-розподіл, розподіл Вейбулла, лог-гамма та лог-нормальний розподіли.

Activate Windows
Go to Settings to activate Windows.

Огляд методів апроксимації ймовірності банкрутства.

1. Дифузійна апроксимація

$$\psi(u) \approx \exp\left\{-\frac{2\rho\mu u}{EX^2}\right\}$$

- + Вимагає лише знання першого та другого моментів виплат;
- + Обчислюється дуже легко;
- + Висока точність при великих значеннях u ;
- Не підходить для розподілу виплат із важким хвостом;
- Дещо завищені значення.

Activate Windows
Go to Settings to activate Windows.

2. Апроксимація Крамера - Лундберга

Необхідно знайти характеристичний коефіцієнт:

$\chi_X(R) = 1 + \frac{cR}{\lambda}$ - розв'язати характеристичне рівняння відносно R.

Далі: $\psi(u) \approx \frac{\rho u}{\chi'_X(R) - c/\lambda} \exp(-Ru)$

- + Точна апроксимація для виплат із легким хвостом;
- + Ефективність при великому стартовому капіталі;
- Вимагає повного знання розподілу;
- Характеристичне рівняння не розв'язується аналітично.

Activate Windows
Go to Settings to activate Windows.

3. Прямий метод Монте - Карло

Наведемо простий псевдокод симуляції:

```
time=0; capital=u;
while(time<T):
    time = time +  $\tau_i$  (  $\tau_i \sim \text{Exp}(\lambda)$  — генеруємо час виплати )
    capital = capital + c  $\tau_i$  -  $X_i$  (  $X_i \sim F(x)$  — генеруємо сумму )
    if (capital < 0): break
```

- + Легка реалізація будь-якою мовою програмування;
- + При великій кількості симуляцій можна отримати точну оцінку;
- Не підходить, коли ймовірність банкрутства близька до нуля або до одиниці;

Activate Windows
Go to Settings to activate Windows.

4. Формула Беекмана

Введемо перетворення інтегровного хвоста для випадкових величин виплат:

$F_{X_I}(y) = \frac{1}{\mu} \int_0^y (1 - F(t)) dt$ - функція розподілу нової величини.

Тоді, за формулою Беекмана:

$$\psi(u) = P\left\{\sum_{k=1}^v X_{Ik} > u\right\}, \quad v \sim \text{Geom}\left(1 - \frac{\lambda\mu}{c}\right)$$

- + Іноді працює швидше, ніж прямий метод Монте-Карло;
- + Більш точні оцінки при однаковій кількості випробувань;
- Інтегральне перетворення не обчислюється аналітично;
- Не підходить для симуляції рідких подій.

Activate Windows
Go to Settings to activate Windows.

Порівняльний аналіз методів апроксимації ймовірності банкрутства

Початкові умови: $c = 1.05\lambda\mu$, $\lambda = 10$, $X_i \sim \text{Exp}\left(\frac{1.0}{20}\right)$

	u=10	u=100	u=1000	u=10000
Аналітичний розв'язок	0.9299	0.7506	0.088	4.35e-11
Дифузійна апроксимація	0.9753	0.7788	0.082	1.38e-11
Метод Монте-Карло(1 рік)	0.942	0.718	0.076	0.0
Метод Монте-Карло(5 років)	0.917	0.721	0.084	0.0
Формула Беекмана (нескінченний час)	0.924	0.743	0.092	0.0

Activate Windows
Go to Settings to activate Windows.

Порівняння аналіз методів апроксимації ймовірності банкрутства

$$c = 1.05\lambda\mu, \lambda = 10, X_i \sim \text{Pareto}(\alpha = 2, x_m = 10).$$

	u=10	u=100	u=1000
Метод Монте-Карло (1 рік)	0.916	0.733	0.2
Метод Монте-Карло (5 років)	0.912	0.751	0.232
Формула Беекмана (нескінченний час)	0.91	0.73	0.22

Activate Windows
Go to Settings to activate Windows.

Порівняльний аналіз методів апроксимації ймовірності банкрутства

$$c = 1.05\lambda\mu, \lambda = 10, X_i \sim \text{Weibull}(\lambda = 10, k = 0.5).$$

	u=10	u=100	u=1000
Метод Монте-Карло (1 рік)	0.937	0.87	0.401
Метод Монте-Карло (5 років)	0.95	0.877	0.446
Формула Беекмана (нескінченний час)	0.94	0.82	0.41
Дифузійна апроксимація	0.9917	0.9200	0.4346

Activate Windows
Go to Settings to activate Windows.

Порівняльний аналіз методів апроксимації ймовірностей банкрутства

Висновки:

- За точністю відносно аналітичного розв'язку, метод Беєкмана показує найкращий результат;
- За швидкодією, формула Беєкмана та методи Монте-Карло працюють повільно, дифузійна апроксимація та апроксимація Крамера Лундберга обчислюються миттєво;
- Стохастичні методи працюють неадекватно, якщо ймовірність банкрутства близька до нуля або до одиниці;
- Моделювання розподілів Парето і Вейбулла показало ефект важкого хвоста – навіть при великому стартовому капіталі ймовірність банкрутства лишається дуже значною.

Activate Windows
Go to Settings to activate Windows.

Висновки до роботи

- розглянуто основну модель страхової математики – модель Крамера-Лундберга;
- проаналізовано поведінку процесу при різних розподілах виплат, порівняно існуючі методи апроксимації;
- розроблено програмний модуль мовою Python для роботи із процесом ризику.

Activate Windows
Go to Settings to activate Windows.

Подальші дослідження

- розширення модуля на клас моделей Спарре – Андерсона, на моделі із броунівським рухом;
- робота над швидкістю програмного продукту. Імплементация паралельних обчислень для методів Монте – Карло;
- робота із історичними даними. Реалізація оцінювання параметрів модельного процесу за існуючими історичними даними.

Activate Windows
Go to Settings to activate Windows.

Дякую за увагу!

Activate Windows
Go to Settings to activate Windows.