РЕФЕРАТ

Дипломна робота: 49 ст., 3 рис., 5 табл., 8 джерел та 3 додатки.

Метою даної роботи є створення адекватної математичної моделі парковки, та отримання математичного результату у вигляді оцінки максимальної кількості автомобілів на парковці.

Результати роботи:

- розглянуто 3 детерміновані моделі поведінки водіїв на парковці, виведено відповідні аналітичні формули;
- розглянуто 2 випадкові моделі поведінки водіїв на парковці, а саме модель рівномірного розподілу по вільному проміжку та модель суміші рівномірного розподілу та розподілу Бернуллі;
- реалізовано додаток консольного типу для моделювання парковки за заданою стратегією поведінки водіїв.

Новизна роботи:

- проведено детальний аналіз результатів Реньї, досліджено і виведено аналітичну формулу для узагальнення класичної моделі парковки Реньї;
- створено додаток для імітаційного моделювання процесу парковки за заданою дисципліною паркування автомобілів.

ВИПАДКОВІ ПРОЦЕСИ, ТЕОРІЯ ЙМОВІРНОСТІ, ТЕОРІЯ ІНТЕГРАЛЬНИХ РІВНЯНЬ ЗІ ЗСУВОМ, ОПЕРАЦІЙНЕ ЧИСЛЕННЯ, ТАУБЕРОВІ ТЕОРЕМИ.

ABSTRACT

The theme of this thesis is ”Stochastic modeling approach to the cars’ parking process research”.

The thesis: 49 p., 3 fig., 5 tabl., 8 sources and 3 appendices.

The theme of this thesis is ”Methods of object recognition in musical notation”.

The purpose of this thesis is to develop adequate mathematical model of parking process, and getting the estimation of maximum count of vehicles on the parking as the result of mathematical inference.

Thesis results:

- three deterministic models of drivers’ behaviour were examined and corresponding analytical formulas were deduced;
- two non-deterministic models of drivers’ behaviour were examined, more precisely – a model with uniform distribution of vehicles along the parking, and a model based on the mixture of uniform and Bernulli distribution;
- all implemented algorithms were assembled into console application for imitational parking process modeling.

Thesis newness:

- a detailed analysis of Rènyi’s results was performed, the classical Rènyi’s parking model was generalised and corresponding analytical formula was deduced;
- a console application for imitational parking process modelling in case of different disciplines was created.

STOCHASTIC PROCESS THEORY, PROBABILITY THEORY, BIASED INTEGRAL EQUATION THEORY, OPERATIONAL CALCULUS, TAUBERIAN THEOREMS.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

МС — математичне сподівання
ММК — метод Монте-Карло
ЕОМ – електронна обчилювальна машина

ВСТУП

Історія першого автомобіля почалася ще в 1768 році разом зі створенням паросилових машин, спроможних перевозити людину. Парові машини працювали на тепловому двигуні зовнішнього згоряння, що перетворював енергію водяної пари в механічну енергію зворотно-поступального руху поршню, а потім в обертальний рух валу. Перша примітивна парова машина було побудована ще у XVII сторіччі Папеном, і являла собою циліндр з поршнем, який підіймався під дією пари, а опускався під тиском атмосфери після згущення відпрацьованої пари. Остаточні удосконалення в паровій машині були зроблені Джеймсом Уаттом в 1769 році.

В 1806 році з'явилися перші машини, які приводилися в рух двигунами внутрішнього згоряння на горючому газу, що призвело до появи в 1885 році повсюди використовуваного газолінового або бензинового двигуна внутрішнього згоряння.

Машини, що працюють на електриці ненадовго з'явилися на початку XX століття, але майже повністю зникли з поля зору аж до початку XXI століття, коли знову виникла зацікавленість в малотоксичному і екологічно чистому транспорті.

На сьогоднішній день автомобілі можна побачити усюди: надворі біля будинків, на автомагістралі чи у невеличкому провулку, в центрі міста та в селі. Автомобілей вже така незліченна кількість, що в Києві скадно знайти куточок, де не видно автотранспорту, де не чутно гулу двигунів. Спираючись на статистику 2011 року, в Україні на 1000 осіб приходилось 158 автомобілів. А от в країнах лідерах за кількістю автомобілей, таких як, наприклад, Монако, нараховувалося порядку 900 авто на 1000 осіб.

Отже, досить гостро постає проблема організації розміщення автомобілів. Для цього створюються парковки. Але при проектуванні парковки постає проблема вибору оптимального розміру. Адже різні водії порізному паркують свої автомобілі: деякі витрачають місце економно, а деякі ставлять авто так, як їм заманеться. Тому, якщо спроектувати парковку, що може вмістити необхідну кількість автомобілів і не більше, то виникне ситуація, що місця на парковці більше немає. Якщо ж парковка буде занадто великою, то ймо-

вірно, що багато місць будуть незайманими, тобто простір був використаний не оптимально.

Саме тому у цій роботі розглядається задача оцінки максимальної кількості автомобілей на парковці. Для вирішення проблеми було виконано наступне:

а) визначено максимальну кількість автомобілей на одновимірній парковці у крайніх, детермінованих випадках, а саме:
   1) коли водії ставлять авто впритул до попереднього;
   2) коли водії ставлять авто рівно так, щоб між автомобілями був пропуск розміру майже з автомобіль, але останній туди все-таки не поміщався;
   3) коли водії ставлять свій автомобіль строго посередині вільного місця.

б) побудовано деякі моделі заповнення парковки з випадковим фактором. Визначено асимптотику функції залежності математичного сподівання максимальної кількості автомобілів на одновимірній парковці від довжини парковки для кожної побудованої моделі;

в) створено невеликий додаток на скриптовій мові Python, що визначає коефіцієнт для асимптотичної оцінки математичного сподівання максимальної кількості автомобілів на одновимірній парковці у одновимірному випадку, а також допомагає встановити залежність від лінійних розмірів парковки у двовимірному випадку.

Об’єктом дослідження є процес паркування автомобілів.

Предметом дослідження є створення алгоритму для визначення максимальної кількості авто на парковці у середньому.

В якості методів дослідження використовуються теорія стохастичних процесів та теорія інтегральних рівнянь зі зсувом.

Наукова новизна роботи: проведено детальний аналіз результатів Реньї щодо проблеми паркування та пакування, та побудовано узагальнення класичної моделі Реньї, в якій поведінка водіїв задається не рівномірним законом розподілу, а сумішшю рівномірного та розподілу Бернуллі..

Практичними результатами роботи є створення додатку для моделювання процесу паркування для більш складних моделей, в тому числі і для двовимірної парковки.

Робота складається з 4 розділів. В першому розділі досліджуються існуючі підходи до вирішення задачі паркування автомобілів, пояснюється можливість застосування лінійної моделі парковки у реальному житті. В другому розділі наводиться математичний апарат для виведення формул у випадку недетермінованих моделей та проведене саме виведення формул. У третьому розділі наведено обґрунтування вибору платформи розробки, аналіз алгоритму модулювання та алгоритму вирахування констант, отриманих у другому розділі. У четвертому розділі проводиться функціонально-вартісний аналіз програмного продукту.

РОЗДІЛ 1 ОГЛЯД ІСНУЮЧОЇ ІТЕРАТУРИ

У цьому розділі наведено інформацію про результат Реньї стосовно проблеми парковки та пакування. Також проведено розбір літератури та опрацьовано існуючі роботи на цю тему.

1.1 Аналіз існуючих підходів

У 1958 році угорський математик Альфред Реньї опублікував працю щодо проблеми заповнення одновимірного обмеженого простору. Ця праця стала підґрунтям для подальшого дослідження в напрямі проблеми парковки та пакування [1].

Задача формулювалася наступним чином: нехай є заданий відрізок $[0;\ x]$, де $x > 1$, і нехай на цей відрізок "паркуються" одновимірні "автомобілі" одиничної довжини, керуючись рівномірним розподілом. У такій ситуації доводилось, що середнє максимальне значення кількості машин на парковці буде $M(x)$ задовільняє наступну систему:

$$M(X) = \begin{cases} 0 & 0 \leq \mathrm{X} < 1 \\ 1 + \frac{2}{X-1} \int\limits_0^{X-1} M(y)dy & \mathrm{X} \geq 1. \end{cases} \tag{1.1}$$

В такому випадку середня щільність автомобілів для великих X виходить

$$m = \lim_{X\to\infty} \frac{M(X)}{X} = \int\limits_0^{\infty} \exp\left(-2\int\limits_0^{x} \frac{1-e^{-y}}{y}dy\right)dx \approx 0.747597. \tag{1.2}$$

Але цей результат потребує узагальнення, адже описана вище класична модель Реньї не дає можливості активно застосовувати результат на практиці. Тому у цій роботі буде створена більш загальна модель, в якій водії мають змішану модель поведінки: з деякою ймовірністю вони ставлять свій автомо-

біль впритул до сусіднього, а в інакшому випадку – керуючись рівномірним розподілом.

1.2 Огляд літератури, в якій досліджено змінну довжину юніту

1.3 Огляд літератури стосовно випадку розподілу з неперервною щільністю з певними обмеженнями

1.4 Огляд літератури стосовно оцінки моментів вищих порядків для проблеми паркування

1.5 Висновки до розділу

У цьому розділі було наведено інформацію про результат Реньї стосовно проблеми парковки та пакування. Також пояснено застосування моделі лінійної парковки у реальному житті.

РОЗДІЛ 2 ВИЗНАЧЕННЯ АСИМПТОТИЧНОЇ ОЦІНКИ МАКСИМАЛЬНОЇ КІЛЬКОСТІ АВТОМОБІЛЕЙ

2.1 Ввідні позначення

Для початку вважатимемо, що у нас одновимірна парковка довжини X, на якій розташовуються автомобілі, довжини 1 кожен. Для спрощення будемо вважати, що водії прибувають на парковку по черзі, і залишають там свій автомобіль. Процес продовжується до того моменту, допоки парковка не заповниться. Тобто, не залишиться вільного відрізку довжини не менше за 1. Через $F(X)$ позначимо максимальну кількість автомобілей. Якщо $F(X)$ – випадкова величина, то через $m(X)$ будемо позначати $\mathbb{E}F(X)$.

2.2 Дослідження крайових випадків

В цьому підрозділі буде розглянуто 3 моделі поведінки водіїв для того, щоб отримати певні оцінки для значень $F(x)$.

Перша модель припускає, що всі водії «чемні» та ставлять свій автомобіль скраю вільної частини парковки, наприклад, зліва. Зрозуміло, що це найбільш оптимальний випадок, тобто така модель дозволить припаркувати рівно стільки автомобілей, скільки взагалі може вміститися на парковці.

Друга модель припускає, що всі водії, навпаки, намагаються зайняти якомога більше місця, і тому відступають від краю вільної зони максимально можливий проміжок, в який не вміститься інший автомобіль.

Третя модель припускає, що водії ставлять свої автомобілі посередині вільного проміжку. Ця модель не є крайнім випадком, але є досить цікавою реалізацією процесу паркування.

Всі 3 моделі схематично зображені на Рис. 2.1.

(а) (б) (в)

Рисунок 2.1 – Схематичне зображення першої (а), другої (б), та третьої (в) невипадкових моделей

Тепер необхідно визначити результати для цих моделей. Для першої моделі відповідь досить очевидна:

$$F(x) = [x] \tag{2.1}$$

Для другої моделі для спрощення вважатимемо, що усі водії стають зліва вільної частини. Тоді у перших $[\frac{x}{2}]$ авто зліва буде відповідний проміжок розміром майже 1 (нехай рівно 1 – границя), а також, якщо виконується нерівність $x - 2 * [\frac{x}{2}] \geq 1$, то можна вмістити ще 1 автомобіль. Остання нерівність виконується тільки якщо $[x]$ – непарне число. Але тоді $[\frac{x+1}{2}] = 2 * [\frac{x}{2}] + 1$. А якщо x – парне, то $[\frac{x}{2}] = [\frac{x+1}{2}]$, і це допоможе уникнути системи у відповіді. Отже, для другої моделі:

$$F(x) = \left[\frac{x+1}{2}\right] \tag{2.2}$$

Для визначення відповіді для тертьої моделі треба навести кілька спостережень:

$$x < 1 \Rightarrow F(x) = 0; \quad 1 \leq x < 3 \Rightarrow F(x) = 1. \tag{2.3}$$

$$F(x) = 2 * F\left(\frac{x-1}{2}\right) + 1, x \geq 1. \tag{2.4}$$

Перше твердження очевидне, а друге випливає з того, що ставлячи автомобіль посередині вільного проміжку довжини x, ми отримуємо два нових вільних проміжка довжини $\frac{x-1}{2}$. Використовуючи наведені факти, спробуємо довести, що

$$F(x) = 2^k - 1, \quad x \in [2^k - 1, 2^{k+1} - 1), k \in \mathbb{N} \cup \{0\} \tag{2.5}$$

Доведення. Скористаємось методом математичної індукції. База індукції доведена, спираючись на спостереження (2.3). Нехай твердження доведено для k, доведемо його для $k + 1$.

$$x \in [2^{k+1} - 1, 2^{k+2} - 1) \Rightarrow \frac{x-1}{2} \in [2^k - 1, 2^{k+1} - 1) \tag{2.6}$$

Оскільки виконується (2.4), і для k виконується (2.5) за припущенням, то маємо:

$$F(x) = 2 * F\left(\frac{x-1}{2}\right) + 1 = 2 * (2^k - 1) + 1 = 2^{k+1} - 2 + 1 = 2^{k+1} - 1 \quad (2.7)$$

Перехід індукції доведено. □

2.3 Опис необхідного математичного апарату для подальшого дослідження

2.3.1 Асимптотична поведінка функції

Нехай f та g – дві функції, визначені в деякому проколотому околі $\dot{U}(x_0)$ точки x_0, причому в цьому околі g не обертається в 0.

Означення 2.3.1: f є "O" великим від g [**?**] при $x \to x_0$, якщо

$$\exists C > 0 \forall x \in \dot{U}(x_0) : |f(x)| < C|g(x)| \quad (2.8)$$

Означення 2.3.2: f є "o" малим від g [**?**] при $x \to x_0$, якщо

$$\forall \varepsilon > 0 \exists \dot{U}_\varepsilon(x_0) \forall x \in \dot{U}_\varepsilon(x_0) : |f(x)| < \varepsilon|g(x)| \quad (2.9)$$

Означення 2.3.3: f є еквівалентним g [**?**] ($f \sim g$) при $x \to x_0$, якщо

$$\lim_{x \to x_0} \frac{f(x)}{g(x)} = 1$$

2.3.2 Теорема Таубера

Означення 2.3.4: Функція $L : [0, +\infty) \to \mathbb{R}$ – слабо мінлива на нескінченності, якщо для $\forall x > 0$

$$\lim_{t \to \infty} \frac{L(tx)}{L(t)} = 1$$

Означення 2.3.5: Функція $L : [0, +\infty) \to \mathbb{R}$ – слабо мінлива в 0, якщо $L(\frac{1}{x})$ – слабо мінлива на нескінченності.

Нехай $u(t)$ – така функція, що має зображення Лапласа. Нехай $U(t) = \int\limits_0^t u(s)ds$ і $\mathcal{L}\{u(t)\} = \omega(\tau)$.

Тоді має місце наступна теорема [5, ст. 445].

Теорема 2.3.1 (Теорема Абеля-Таубера): Нехай L – слабко мінлива на нескінченності і $0 \leq \rho < +\infty$. Тоді наступні два твердження тотожні:

$$\omega(\tau) \sim \tau^{-\rho} L(1/\tau), \qquad \tau \to 0 \tag{2.10}$$

$$U(t) \sim \frac{1}{\Gamma(\rho+1)} t^{\rho} L(t), \qquad t \to +\infty \tag{2.11}$$

Досить цікавим зауваженням до цієї теореми є те, що можна змінити границі на протилежні, тобто $\tau \to \infty,\ t \to 0$ [5, ст. 445].

Теорема 2.3.2: Твердження теореми (2.3.1) залишається вірним, якщо поміняти місцями 0 та ∞, тобто $\tau \to \infty,\ t \to 0$ (і, відповідно, L – слабко мінлива в 0).

2.4 Дослідження моделі з вибором місця для авто за сумішшю рівномірного та розподілу Бернулі

В цій моделі водії вибирають місце для автомобіля, керуючись наступним правилом

- з ймовірністю p водій ставить автомобіль в правому кінці вільного проміжку,
- з ймовірністю $q = 1 - p$ водій вибирає місце керуючись рівномірним розподілом.

2.4.1 Виведення інтегрального рівняння

Нескладно переконатись, що порядок вибору вільних проміжків водіями не впливає на результат, тому будемо вважати, що після паркування одно-

го автомобіля парковка розбивається на 2 частини, і після цього спочатку заповнюється ліва частина, а потім права.

Необхідно визначити $m_p(x) = \mathbb{E}F(x)$. Нехай $\xi \sim Uniform(0, x-1)$ – випадкова величина, що визначає положення лівого краю першого автомобіля на парковці у випадку вибору місця за рівномірним розподілом. Тоді маємо наступну тотожність:

$$m_p(x) = p(1 + m_p(x-1)) + q(1 + \mathbb{E}(\mathbb{E}(F(\xi) + F(x-1-\xi)|\xi))) =$$
$$= 1 + pm_p(x-1) + q\int\limits_0^{x-1} m_p(t)\frac{1}{x-1}dt + q\int\limits_0^{x-1} m_p(x-t-1)\frac{1}{x-1}dt$$

Так як

$$\int\limits_0^{x-1} m_p(x-t-1)dt = \langle u = x-t-1\rangle = -\int\limits_{x-1}^{0} m_p(u)du = \int\limits_0^{x-1} m_p(u)du,$$

то

$$m_p(x) = 1 + pm_p(x-1) + \frac{2q}{x-1}\int\limits_0^{x-1} m_p(t)d \tag{2.12}$$

Для зручності зробимо заміну $x \to x+1$. Отримаємо

$$m_p(x+1) = 1 + pm_p(x) + \frac{2q}{x}\int\limits_0^{x} m_p(t)dt, \quad \forall x > 0 \tag{2.13}$$

Таким чином, отримали інтегральне рівняння. До того ж, відомо, що

$$m_p(x) \equiv 0, \quad x \in [0; 1) \tag{2.14}$$

Спираючись на (2.1) та (2.2), маємо обмеження на $m(x)$:

$$\left[\frac{x+1}{2}\right] \le m_p(x) \le [x] \tag{2.15}$$

З цієї нерівності випливає, що якщо є якась асимптотика у функції $m_p(x)$, то вона порядку x, тобто

$$m_p(x) \sim C_p \cdot x \text{ при } x \to +\infty, \quad C_p \in [0.5; 1] \tag{2.16}$$

2.4.2 Перехід до зображення Лапласа

Спробуємо розв'язати (2.13) за допомогою перетворення Лапласа.

Оскільки виконується (2.15), то зображення Лапласа для $m_p(x)$ існує. До того ж, за властивістю (A.16):

$$\mathcal{L}\left\{m_p(x+1)\right\} = \langle(2.14)\rangle = \mathcal{L}\left\{m_p(x+1)\eta(x+1)\right\} = e^s M_p(s). \tag{2.17}$$

Оскільки $m_p(x) \le x$, то $\int\limits_0^x m_p(t)dt < x^2$, тобто для інтегралу від $m_p(x)$ зображення також існує, за властивістю (A.12):

$$\mathcal{L}\left\{\int\limits_0^x m_p(t)dt\right\} = \frac{M_p(s)}{s}.$$

Аналогічно доводиться, що $\frac{1}{x}\int\limits_0^x m_p(t)dt < x$ при $x > 0$, а тому зображення Лапласа для цього виразу також існує. Тоді за властивістю (A.13):

$$\mathcal{L}\left\{\frac{2}{x}\int\limits_0^x m_p(t)dt\right\} = 2\int\limits_s^\infty \frac{M_p(u)}{u}du.$$

Таким чином, отримали інтегральне рівняння в термінах зображення Лапласа, яке вже можна розв'язати, адже нема зсуву:

$$e^s M_p(s) = pM_p(s) + 2q\int\limits_s^\infty \frac{M_p(u)}{u}du + \frac{1}{s} \tag{2.18}$$

Продиференціюємо обидві частини рівняння відносно s:

$$e^s M_p(s) + e^s \dot{M}_p(s) = p\dot{M}_p(s) - 2q\frac{M_p(s)}{s} - \frac{1}{s^2} \tag{2.19}$$

Виразимо $\dot{M}_p(s)$ з цього рівняння:

$$\dot{M}_p(s) = -M_p(s)\left(\frac{e^s}{e^s - p} + \frac{2q}{s(e^s - p)}\right) - \frac{1}{s^2(e^s - p)} \tag{2.20}$$

Розв'яжемо отримане диференційне рівняння. Спочатку розв'яжемо однорідну частину:

$$\begin{aligned}
\dot{M}_p^h(s) &= -M_p^h(s)\left(\frac{e^s}{e^s - p} + \frac{2q}{s(e^s - p)}\right) \\
\frac{\dot{M}_p^h(s)}{M_p^h(s)} &= -\left(\frac{e^s}{e^s - p} + \frac{2q}{s(e^s - p)}\right) \\
\int_1^s \frac{\dot{M}_p^h(u)}{M_p^h(u)}du &= -\int_1^s \left(\frac{e^u}{e^u - p} + \frac{2q}{u(e^u - p)}\right)du \\
\ln M_p^h(u)\Big|_1^s &= -\int_1^s \frac{e^u}{e^u - p}du - 2\int_1^s \frac{q}{u(e^u - p)}du
\end{aligned}$$

Позначимо

$$Q_p(s) := \int_1^s \frac{q}{u(e^u - p)}du \tag{2.21}$$

Оскільки

$$\begin{aligned}
\int_1^s \frac{e^u}{e^u - p}du &= \langle u = e^u - p,\ du = e^u du = (u + p)du\rangle = \\
&= \int_{e-p}^{e^s - p} \frac{u + p}{u}(u + p)^{-1}du = \int_{e-p}^{e^s - p} \frac{du}{u} = \\
&= \log(e^s - p) - \log(e - p) = \log\frac{e^s - p}{e - p},
\end{aligned}$$

то

$$\ln M_p^h(s) = \ln M_p^h(1) - \log \frac{e^s - p}{e - p} - 2Q_p(s)$$
$$M_p^h(s) = M_p^h(1) \cdot \frac{e - p}{e^s - p} \cdot e^{-2Q_p(s)} \cdot const$$

Оскільки $M(1)$ та $(e - p)$ можна включити в константу, то маємо розв'язок

$$M_p^h(s) = C \cdot \left((e^s - p)e^{2Q_p(s)}\right)^{-1}, \quad \forall C \in \mathbb{R} \tag{2.22}$$

Дійсно, перевіримо цей розв'язок:

$$\begin{aligned}
\dot{M}_p^h(s) &= C \cdot \frac{\partial}{\partial s}\left(\frac{1}{(e^s - p)e^{2Q_p(s)}}\right) = \\
&= -C \cdot \left(\frac{1}{(e^s - p)e^{2Q_p(s)}}\right)^2 \cdot \left(e^s + (e^s - p)\frac{2q}{s(e^s - p)}\right) e^{2Q_p(s)} = \\
&= -C\left((e^s - p)e^{2Q_p(s)}\right)^{-1} \cdot \left(\frac{e^s}{e^s - p} + \frac{2q}{s(e^s - p)}\right) = \\
&= -M_p^h(s) \cdot \left(\frac{e^s}{e^s - p} + \frac{2q}{s(e^s - p)}\right)
\end{aligned}$$

Нескладно помітити, що отримали вихідне рівняння. Тепер застосуємо метод варіації довільних сталих:

$$M_p(s) = C(s) \cdot \left((e^s - p)e^{2Q_p(s)}\right)^{-1}$$

Продиференціювавши відносно s, отримаємо:

$$\begin{aligned}
\dot{M}_p(s) = \dot{C}(s) \cdot \left((e^s - p)e^{2Q_p(s)}\right)^{-1} - C(s)\left((e^s - p)e^{2Q_p(s)}\right)^{-1} \cdot \\
\cdot \left(\frac{e^s}{e^s - p} + \frac{2q}{s(e^s - p)}\right)
\end{aligned}$$

З іншої сторони, з (2.20) маємо

$$\dot{M}_p(s) = -C(s)\left((e^s - p)e^{2Q_p(s)}\right)^{-1}\left(\frac{e^s}{e^s - p} + \frac{2q}{s(e^s - p)}\right) - \frac{1}{s^2(e^s - p)}$$

Тому

$$\dot{C}(s) = -\left((e^s - p)e^{2Q_p(s)}\right) \cdot \frac{1}{s^2(e^s - p)} = -\frac{e^{2Q_p(s)}}{s^2}$$

Тоді простим інтегруванням в межах від 1 до s отримуємо:

$$C(s) = -\int_1^s \frac{e^{2Q_p(u)}}{u^2} du + const \tag{2.23}$$

І тоді отримуємо вираз для $M_p(s)$:

$$\begin{aligned} M_p(s) &= -\left(\int_1^s \frac{e^{2Q_p(u)}}{u^2} du + const\right)\left((e^s - p)e^{2Q_p(s)}\right)^{-1} = \\ &= -\left(\int_1^s \frac{e^{2Q_p(u)}}{u^2} du + K\right)\frac{1}{(e^s - p)e^{2Q_p(s)}}, \quad K \in \mathbb{R} \end{aligned} \tag{2.24}$$

Перевіримо отриманий результат:

$$\dot{M}_p(s) = -\frac{\partial}{\partial s}\left(\int_1^s \frac{e^{2Q_p(u)}}{u^2} du + K\right)\frac{1}{(e^s - p)e^{2Q_p(s)}} - \left(\int_1^s \frac{e^{2Q_p(u)}}{u^2} du + K\right) \cdot$$

$$\cdot \left(\frac{1}{(e^s - p)e^{2Q_p(s)}}\right)' = -\frac{e^{2Q_p(s)}}{s^2}\frac{1}{(e^s - p)e^{2Q_p(s)}} + \left(\int_1^s \frac{e^{2Q_p(u)}}{u^2} du + K\right) \cdot$$

$$\cdot \left((e^s - p)e^{2Q_p(s)}\right)^{-1} \cdot \left(\frac{e^s}{e^s - p} + \frac{2q}{s(e^s - p)}\right) = -\frac{1}{s^2(e^s - p)} -$$

$$- M_p(s)\left(\frac{e^s}{e^s - p} + \frac{2q}{s(e^s - p)}\right)$$

Перевірено. Тоді остаточний результат без вирахування константи:

$$M_p(s) = \left(\int_s^1 \frac{e^{2Q_p(u)}}{u^2} du + K\right)\frac{1}{(e^s - p)e^{2Q_p(s)}} \tag{2.25}$$

Спираючись на (2.17), маємо

$$\mathcal{L}\left\{m_p(x+1)\right\} = \tilde{M}_p(s) = \left(\int\limits_s^1 \frac{e^{2Q_p(u)}}{u^2}du + K\right)\frac{e^s}{e^s - p}e^{-2Q_p(s)}. \quad (2.26)$$

Оскільки зображення Лапласа – аналітична функція в деякій правій півплощині комплексного простору, то $\tilde{M}_p(s) \to 0,\ s \to +\infty$.

Розглянемо $Q_p(s)$ (s розглядаємо на дійсній вісі):

$$\begin{aligned} Q_p(s) = \int\limits_1^s \frac{1-p}{u(e^u - p)}du < \int\limits_1^\infty \frac{1-p}{u(e^u - p)}du < \int\limits_1^\infty \frac{1-p}{e^u - p}du < \\ < \int\limits_1^\infty \frac{1}{e^u}du = \exp(-1) - \exp(-\infty) = \exp(-1) \end{aligned} \quad (2.27)$$

Останній перехід нерівності пояснюється досить просто:

$$\frac{1-p}{u-p} < \frac{1}{p},\ u > 1 \Leftrightarrow u - up = u(1-p) < u - p,\ u > 1$$

Тобто $Q_p(s)$ - обмежена на $[1;\infty]$. Тому обмеженими на цій вісі будуть і $e^{\pm 2Q_p(s)}$. Також зрозуміло, що якщо інтегрувати по дійсній вісі, то $Q_p(s)$ – монотонно зростаюча за s. Тому

$$0 = \tilde{M}_p(\infty) = \lim_{s\to\infty}\tilde{M}_p(s) = \left(\int\limits_\infty^1 \frac{e^{2Q_p(u)}}{u^2}du + K\right)\lim_{s\to\infty}e^{-2Q_p(s)} \quad (2.28)$$

Тут $\lim\limits_{s\to\infty} e^{-2Q_p(s)} = const > 0$, тому маємо, що

$$K = -\int\limits_\infty^1 \frac{e^{2Q_p(u)}}{u^2}du = \int\limits_1^\infty \frac{e^{2Q_p(u)}}{u^2}du. \quad (2.29)$$

Таким чином, отримали нову версію $M_p(s)$:

$$M_p(s) = \left(\int_s^1 \frac{e^{2Q_p(u)}}{u^2} du + K \right) \frac{1}{(e^s - p)e^{2Q_p(s)}} = \\ = \frac{1}{(e^s - p)e^{2Q_p(s)}} \int_s^\infty \frac{e^{2Q_p(u)}}{u^2} du \tag{2.30}$$

2.4.3 Застосування тауберівської теореми

Для знаходження асимптотики $m_p(x)$ на нескінченності, за теоремою Таубера (2.3.1) необхідно визначити асимптотику $M_p(s)$ при $s \to 0$.

Якщо знайти такі $C \in \mathbb{R}$ та $\delta \in \mathbb{R}^+$, що $M_p(s) \sim C \cdot s^{-\delta}, \ s \to 0$, то можна стверджувати, що $\int_0^x m_p(x)dx \sim \frac{1}{\Gamma(\delta+1)} C x^\delta, \ x \to \infty$. Вже зараз зрозуміло, що $\delta = 2$, адже теорема справедлива в обидва боки і виконується (2.16).

Для цього розглянему поведінку в нулі трьох множників, з яких складається $M_p(s)$, а саме:

а) $\frac{1}{e^s - p}$;

б) $e^{-2Q_p(s)}$;

в) $\int_s^\infty \frac{e^{2Q_p(u)}}{u^2} du$,

Щодо першого множнику, то в 0 він, очевидно, прямує до $\frac{1}{1-p}$.

Для наступного аналізу доведемо деякі леми.

Лема 2.4.1: $e^{-2Q_p(s)}$ поводиться як s^{-2} в 0, з точністю до константи, а саме:

$$\lim_{s \to 0} \frac{e^{-2Q_p(s)}}{s^{-2}} = \exp\left(-2 \int_0^1 \frac{e^u - 1}{u(e^u - p)} du \right) \tag{2.31}$$

Доведення. Для знаходження ліміту прологарифмуємо вираз. Отримаємо:

$$2\ln s - 2Q_p(s) = 2\ln s - 2\int\limits_1^s \frac{1-p}{u(e^u-p)}du = 2\int\limits_1^s \frac{1}{u}du -$$
$$-2\int\limits_1^s \frac{1-p}{u(e^u-p)}du = 2\int\limits_1^s \frac{e^u-1}{u(e^u-p)}du = -2\int\limits_s^1 \frac{e^u-1}{u(e^u-p)}du$$

Тепер, підвівши до експоненти обидві частини, отримаємо:

$$\frac{e^{-2Q_p(s)}}{s^{-2}} = \exp\left(-2\int\limits_s^1 \frac{e^u-1}{u(e^u-p)}du\right)$$

Якщо довести, що інтеграл

$$-2\int\limits_0^1 \frac{e^u-1}{u(e^u-p)}du$$

збігається, то лему буде доведено, адже експонента – неперервна функція, і можна переходити до ліміту під експонентою. Зрозуміло, що

$$-2\int\limits_s^1 \frac{e^u-1}{u(e^u-p)}du$$

збігається для $\forall s \in (0;\ 1]$. Дійсно, оскільки $e^u - 1 < e^u - p$, підінтегральна функція $\frac{1-e^{-u}}{u}$ мажорується $\frac{1}{u}$, яка, в свою чергу, має скінченне значення інтегралу:

$$\int\limits_s^1 \frac{1}{u}du = \ln 1 - \ln s = -\ln s, \quad s > 0$$

Невизначеність виникає лише в точці 0. Знайдемо ліміт підінтегральної функції в точці 0:

$$\lim_{s\to 0}\frac{e^u-1}{u(e^u-p)} = \langle\text{правило Лопіталя для невизначенності 0/0}\rangle = \\ = \lim_{s\to 0}\frac{e^u}{ue^u+(e^u-p)} = \frac{1}{1-p}$$

Таким чином, підінтегральна функція обмежена в деякому ε-околі 0, тому інтеграл також збіжний, і лему доведено. □

Лема 2.4.2: Функція

$$Q_p(s) = \int_1^s \frac{1-p}{u(e^u-p)}du$$

– обмежена на $[w;\ \infty],\ w>0$.

Доведення. На проміжку $[1;\ \infty]$ підінтегральна функція мажорується функцією e^{-u} (див. (2.27)), а на проміжку $[w;\ 1]$ – функцією $\frac{1}{u}$, адже $1-p < e^u-p,\ u>0$. Тому, аналогічно доведенню попередньо леми, інтеграл буде збіжний, і:

$$Q_p(s) \le \int_1^\infty e^{-u}du = \exp(-1), \quad s\ge 1$$

$$Q_p(s) \le \int_w^1 \frac{1}{u}du = -\ln w, \quad s\in[w;\ 1]$$

Таким чином, $Q_p(s)\le \max\{-\ln w, \exp(-1)\}$. □

Лема 2.4.3: Інтеграл

$$\int_0^\infty \frac{e^{2Q_p(u)}}{u^2}du$$

– збіжний.

Доведення. Спираючись на лему (2.4.1), маємо, що підінтегральна функція прямує до деякої константи при $u \to 0$, оскільки є обернено пропорційною до функції з тої леми. Тому в деякому проколотому ε-околі точки 0 підінтегральна функція буде обмежена. На інтервалі $[\varepsilon;\ \infty]$ за лемою (2.4.2), $Q_p(u)$ – обмежена, а тому і $\exp(2Q_p(u))$ також. Тому збіжність на інтервалі $[\varepsilon;\ \infty]$ виконується, якщо збігається інтеграл

$$\int\limits_{\varepsilon}^{\infty} \frac{1}{u^2} du.$$

А його збіжність – відомий факт. □

Таким чином, спираючись на доведені леми, маємо при $s \to 0$:

$$M_p(s) \sim \frac{s^{-2}}{1-p} \cdot \exp\left(-2\int\limits_0^1 \frac{e^u-1}{u(e^u-p)}du\right) \int\limits_0^\infty \frac{e^{2Q_p(u)}}{u^2}du =$$

$$= \frac{s^{-2}}{1-p} \cdot \exp\left(-2\int\limits_0^1 \frac{e^u-1}{u(e^u-p)}du\right) \int\limits_0^\infty \exp\left(2\int\limits_1^u \frac{1-p}{\tau(e^\tau-p)}d\tau - 2\ln u\right) du =$$

$$= \frac{s^{-2}}{1-p} \cdot \exp\left(-2\int\limits_0^1 \frac{e^u-1}{u(e^u-p)}du\right) \int\limits_0^\infty \exp\left(2\int\limits_1^u \frac{1-p}{\tau(e^\tau-p)}d\tau - 2\int\limits_1^u \frac{1}{\tau}d\tau\right) du =$$

$$= \frac{s^{-2}}{1-p} \cdot \exp\left(-2\int\limits_0^1 \frac{e^u-1}{u(e^u-p)}du\right) \int\limits_0^\infty \exp\left(-2\int\limits_1^u \frac{e^\tau-1}{\tau(e^\tau-p)}d\tau\right) du =$$

$$= \frac{s^{-2}}{1-p} \cdot \int\limits_0^\infty \exp\left(-2\int\limits_0^1 \frac{e^\tau-1}{\tau(e^\tau-p)}d\tau - 2\int\limits_1^u \frac{e^\tau-1}{\tau(e^\tau-p)}d\tau\right) du$$

Склавши інтеграли під експонентою, отримаємо:

$$M_p(s) \sim s^{-2} \cdot \underbrace{\frac{1}{1-p}\int\limits_0^\infty \exp\left(-2\int\limits_0^u \frac{e^\tau-1}{\tau(e^\tau-p)}d\tau\right) du}_{\text{збігається, не залежить від } s}, \quad s \to 0. \qquad (2.32)$$

Тепер, за теоремою Таубера маємо при $x \to \infty$:

$$\int_0^x m_p(x)dx \sim \frac{1}{(1-p)\Gamma(2+1)} \int_0^\infty \exp\left(-2\int_0^u \frac{e^\tau - 1}{\tau(e^\tau - p)}d\tau\right) du \cdot x^2. \quad (2.33)$$

Або, продиференціювавши обидві частини, отримаємо:

$$m_p(x) \sim \frac{2}{(1-p)\Gamma(2+1)} \int_0^\infty \exp\left(-2\int_0^u \frac{e^\tau - 1}{\tau(e^\tau - p)}d\tau\right) du \cdot x.$$

$$m_p(x) \sim \frac{1}{1-p} \int_0^\infty \exp\left(-2\int_0^u \frac{e^\tau - 1}{\tau(e^\tau - p)}d\tau\right) du \cdot x, \quad x \to \infty. \quad (2.34)$$

Тут ми мали право диференціювати обидві частини за правилом Лопіталя, адже має місце невизначеність ∞/∞.

Таким чином було доведено, що $m_p(x) \sim C_p \cdot x$ при $x \to \infty$, де

$$C_p = \frac{1}{1-p} \int_0^\infty \exp\left(-2\int_0^u \frac{e^\tau - 1}{\tau(e^\tau - p)}d\tau\right) du \quad (2.35)$$

2.5 Покращення асимптотичної оцінки

Нескладно помітити, що $\mu_p(x) = Cx - \frac{1-C}{1-p}$ є розв'язком рівняння (2.13) $\forall C \in \mathbb{R}$:

$$\begin{aligned}
&1 + \frac{2(1-p)}{x}\int_0^x \mu_p(t)dt + p\mu_p(x) = \\
&= 1 + \frac{2(1-p)}{x}\int_0^x \left\{Ct - \frac{1-C}{1-p}\right\}dt + p\left(Cx - \frac{1-C}{1-p}\right) = \\
&= 1 + (1-p)Cx - 2(1-C) + pCx - (\frac{1}{1-p} - 1)(1-C) = \\
&= C(x+1) - \frac{1-C}{1-p} = \mu_p(x+1).
\end{aligned}$$

Тому резонно апроксимувати досліджувану функцію $m_p(x)$ використовуючи функцію μ_p. Далі буде доведено наступне твердження:

$$\lim_{x\to\infty}\left(m(x)-C_p x+\frac{1-C_p}{1-p}\right)=0. \tag{2.36}$$

2.5.1 Застосування тауберівської теореми для покращення оцінки

Нескладно помітити, що

$$\begin{aligned} Q_p(s)=\int\limits_1^s\frac{1-p}{u(e^u-p)}du=\int\limits_1^s\frac{1}{u}du-\int\limits_1^s\frac{e^u-1}{u(e^u-p)}du= \\ =\ln(s)-\int\limits_1^s\frac{e^u-1}{u(e^u-p)}du. \end{aligned} \tag{2.37}$$

Тоді з (2.37) та (2.30) маємо

$$M_p(s)=\frac{e^{-2Q_p(s)}}{(e^s-p)}\int\limits_s^\infty\frac{e^{2Q_p(u)}}{u^2}du=$$

$$\frac{s^{-2}}{e^s-p}\exp\left(2\int\limits_1^s\frac{e^u-1}{u(e^u-p)}du\right)\int\limits_s^\infty\exp\left(-2\int\limits_1^u\frac{e^\tau-1}{\tau(e^\tau-p)}d\tau\right)du.$$

Оскільки $e^u-1\sim u,\ u\to 0$, то інтеграл $\int\limits_0^1\frac{e^u-1}{u(e^u-p)}du$ існує для $\forall p<1$. Отже, можна винести з-під інтегралу константу $\exp\left(\int\limits_0^1\frac{e^u-1}{u(e^u-p)}du\right)$.

$$M_p(s) = \frac{s^{-2}}{e^s - p} \exp\left(2\int\limits_0^s \frac{e^u - 1}{u(e^u - p)} du\right) \int\limits_s^\infty \exp\left(-2\int\limits_0^u \frac{e^\tau - 1}{\tau(e^\tau - p)} d\tau\right) du =$$

$$= \frac{s^{-2}}{e^s - p} \exp\left(2\int\limits_0^s \frac{e^u - 1}{u(e^u - p)} du\right) \left[(1-p)C_p - \right.$$

$$\left. - \int\limits_0^s \exp\left(-2\int\limits_0^u \frac{e^\tau - 1}{\tau(e^\tau - p)} d\tau\right) du\right]$$

Оскільки $s^2 M_p(s)$ — аналітична функція на $\mathfrak{Re}\, s > \sigma$, де $\sigma < 0$ при $p < 1$, то з розкладу в ряд Тейлора випливає

$$M_p(s) = \frac{C_p}{s^2} + \frac{\frac{\partial}{\partial s}(s^2 M_p(s))}{s} + \psi_p(s), \tag{2.38}$$

де $\psi_p(s)$ — аналітична на $\mathfrak{Re}\, s > \sigma$. Позначимо через $R_p(s) = \int\limits_0^s \frac{e^u - 1}{u(e^u - p)} du$, тоді

$$\frac{\partial}{\partial s}(s^2 M_p(s)) = -\frac{e^s}{(e^s - p)^2} \left[(1-p)C_p\ e^{2R_p(s)} - \int\limits_0^s e^{2(R_p(s) - R_p(u))} du\right] +$$

$$+ \frac{1}{e^s - p} \left[\left((1-p)C_p - \int\limits_0^s e^{-R_p(u))} du\right) e^{2R_p(s)} \frac{2(e^s - 1)}{s(e^s - p)} - e^{2(R_p(s) - R_p(s))}\right]$$

Підставивши $s = 0$, отримаємо

$$\frac{\partial(s^2 M_p)}{\partial s}(0) = -\frac{1}{(1-p)^2}(1-p)C_p + \frac{1}{1-p}\left[2C_p - 0 - 1\right] = \frac{C_p - 1}{1-p}.$$

Таким чином,

$$M_p(s) = C_p s^{-2} - \frac{1 - C_p}{1-p} s^{-1} + \psi_p(s). \tag{2.39}$$

Розглянемо перетворення Лапласа від $m_p(x) - C_p x$:

$$\mathcal{L}\left\{m_p(x) - C_p x\right\} = M_p(s) - C_p s^{-2} = \frac{C_p - 1}{1-p} s^{-1} + \psi_p(s),$$
$$\lim_{s\to 0} \frac{\mathcal{L}\left\{m_p(x) - C_p x\right\}}{s^{-1}} = \frac{C_p - 1}{1-p}.$$

Тоді за тауберівською теоремою (2.3.1) можна зробити висновок

$$\lim_{x\to\infty} m_p(x) - C_p x = \frac{C_p - 1}{1-p}, \tag{2.40}$$

тобто (2.36) доведено.

2.5.2 Застосування зворотної формули Фур'є-Мелліна

У цьому параграфі буде виконано уточнення оцінки (2.36) за допомогою зворотної формули Фур'є-Мелліна. У роботі [6] цей результат сформульовано наступним чином.

Теорема 2.5.1 (Формула Фур'є-Мелліна): Нехай $f(t) = 0, t < 0$, $f(t) < Ce^{\alpha t}$, $\mathcal{L}\{f(t)\} = F(s)$. Тоді $\forall \sigma > \alpha$

$$f(t) = \frac{1}{2\pi i} \int\limits_{\sigma - i\infty}^{\sigma + i\infty} e^{st} F(s) ds. \tag{2.41}$$

Наслідок 2.5.1.1: Нехай $0 < f(t) < Ce^{\alpha t},\ \forall \alpha > 0$ і $F(s)$ — аналітична в півплощині $s > \sigma,\ \sigma < 0$. Тоді

$$f(t) = \frac{1}{2\pi i} \int\limits_{-i\infty}^{+i\infty} e^{st} F(s) ds. \tag{2.42}$$

Доведення. Спочатку покажемо, що $|F(s)| < F(\mathfrak{Re}\, s)$.

$$|F(s)| = \left| \int\limits_0^\infty f(t) e^{-st} dt \right| \le \int\limits_0^\infty |f(t) e^{-st}|\, dt = \int\limits_0^\infty f(t) e^{-t\,\mathfrak{Re}\, s} dt = F(\mathfrak{Re}\, s).$$

Тепер, оскільки F — аналітична в правій півплощині відносно σ, то за теоремою Коші [7]

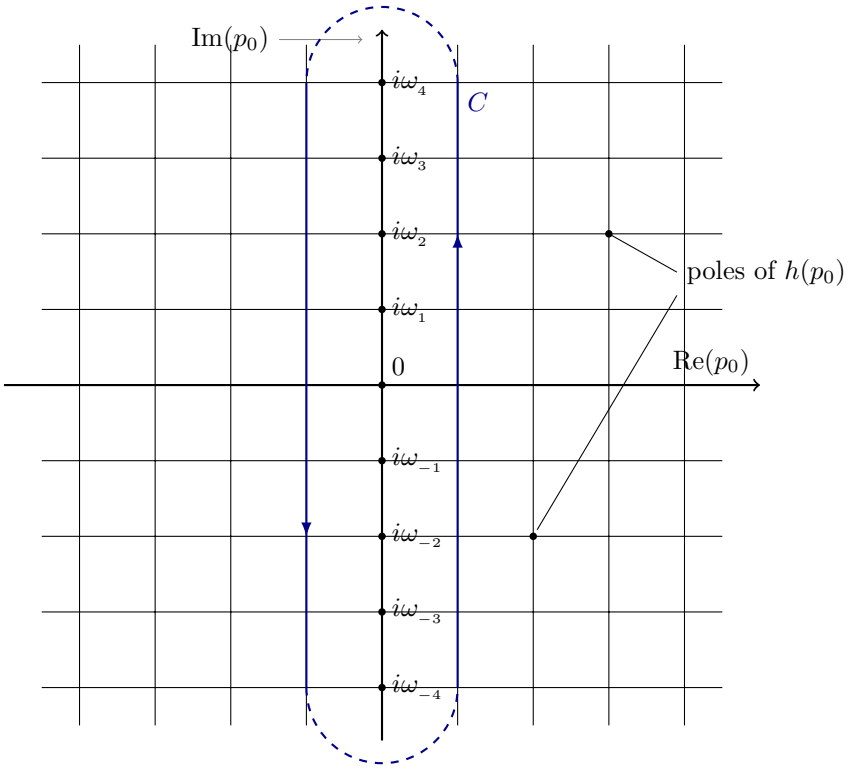

Рисунок 2.2 – Шлях інтегрування аналітичної функції $F(s)$

$$\int\limits_{-i\infty}^{i\infty} F(s)e^{st}ds + \int\limits_{iR}^{iR+\delta} F(s)e^{st}ds + \int\limits_{iR+\delta}^{-iR+\delta} F(s)e^{st}ds + \int\limits_{-iR+\delta}^{-iR} F(s)e^{st}ds = 0.$$

Позначимо $V(R) = \int\limits_{-iR+\delta}^{iR+\delta} F(s)e^{st}ds,\ \forall 0 < \delta < 1$. Нескладно переконатися, що $|\int\limits_{iR}^{iR+\delta} F(s)e^{st}ds| \leq \delta e^{\delta t} \max\limits_{[0,1]} |F(\mathfrak{Re}\, s)| \leq \delta \cdot const$. Тоді $\forall \delta > 0,\ \forall R > 0$

$$\left| \int\limits_{-iR}^{iR} F(s)e^{st}ds - V(R) \right| \leq \delta \cdot const.$$

Зробимо граничний перехід $\delta \to 0$: оскільки через аналітичність $F(s)$ інтеграл $\int\limits_{-iR}^{iR} F(s)e^{st}ds$ існує, то

$$\int\limits_{-iR}^{iR} F(s)e^{st}ds = V(R).$$

Завершується доведення граничним переходом $R \to \infty$:

$$\lim_{R\to\infty} \frac{1}{2\pi i} \int\limits_{-iR}^{iR} F(s)e^{st}ds = \lim_{R\to\infty} \frac{1}{2\pi i} V(R) = f(t).$$

□

Виходячи з рівняння (2.39), аналітичності $\psi_p(s)$ та наслідку (2.5.1.1), маємо наступне твердження:

$$m_p(x) = \frac{1}{2\pi i} \int\limits_{\delta-i\infty}^{\delta+i\infty} M_p(s)e^{sx}ds = C_p x - \frac{1-C_p}{1-p} + \frac{1}{2\pi} \int\limits_{-R}^{R} \psi_p(it)e^{itx}dt. \quad (2.43)$$

Лема 2.5.1: Для $\forall p \in [0;1)$

$$\psi_p(it) \sim O\left(\frac{1}{|t|}\right) \quad (2.44)$$

Доведення. Якщо показати, що $M_p(it) = O\left(\frac{1}{|t|}\right)$, то твердження леми випливає автоматично:

$$\psi_p(it) = M_p(it) + C_p t^{-2} - i\frac{1-C_p}{1-p}t^{-1} = O\left(\frac{1}{|t|}\right)$$

Залишається показати, що для

$$M_p(s) = \frac{s^{-2}}{e^s - p} \exp\left(2\int\limits_0^s \frac{e^u - 1}{u(e^u - p)}du\right) \int\limits_s^\infty \exp\left(-2\int\limits_0^u \frac{e^\tau - 1}{\tau(e^\tau - p)}d\tau\right) du$$

твердження правдиве. Щоб при підрахунку $M_p(it)$ інтегрувати за уявною віссю, необхідно показати, що інтеграл

$$\int_{it}^{i\infty} \exp\left(-2\int_0^u \frac{e^\tau - 1}{\tau(e^\tau - p)} d\tau\right) du$$

існує. Оскільки інтеграл по контуру, зображеному на Рис. 2.3, дорівнює 0 за теоремою Коші, то

$$\int_{it}^{iR} = -\int_{K_t} + \int_t^R + \int_{K_R},$$

де K_r — інтеграл по дузі з радіусом r та зміною кута відносно вісі абсцис в межах $0 \leq \varphi \leq \pi$.

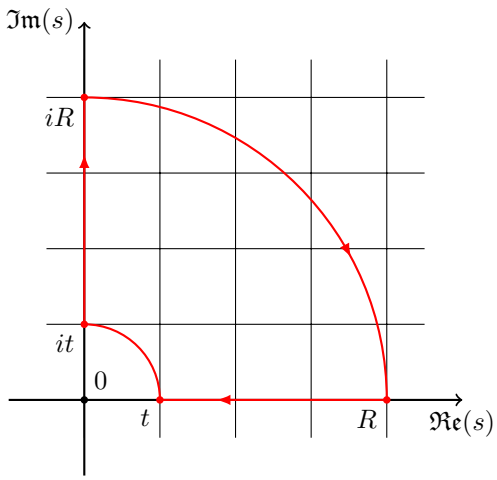

Рисунок 2.3 – Шлях інтегрування функції $\exp\left(-2\int_0^u \frac{e^\tau-1}{\tau(e^\tau-p)} d\tau\right)$

Нескладно помітити, що для доведення існування інтегралу $\int_{it}^{i\infty}$ достатньо показати, що $\lim_{R\to\infty} |\int_{K_R}| = 0$, адже інтеграл $\int_t^\infty$ існує через існування $M_p(t)$. А це вже досить нескладно доводиться.

Отже, при $t > 0$

$$M_p(it) = -\frac{it^{-2}}{e^{it} - p} \exp\left(2 \int_0^t \frac{e^{iu} - 1}{u(e^{iu} - p)} du\right) \cdot \cdot \int_t^\infty \exp\left(-2 \int_0^u \frac{e^{i\tau} - 1}{\tau(e^{i\tau} - p)} d\tau\right) du. \tag{2.45}$$

Враховуючи, що інтеграл $\int_1^\infty \frac{1-p}{u(e^{iu}-p)} du$ існує, з (2.45) негайно випливає

$$M_p(it) = O\left(\frac{1}{|t|}\right) \tag{2.46}$$

для $t > 0$. За властивістю перетворення Лапласа $M_p(-it) = \overline{M_p(it)}$, тому (2.46) виконується і для $t < 0$. □

Отримали, що

$$\lim_{X \to \infty} \left(m(X) - C_p X - \frac{1 - C_p}{1 - p}\right) = O\left(\frac{1}{X^n}\right) \qquad \forall n \in \mathbb{N} \tag{2.47}$$

2.6 Висновки до розділу

У даному розділі було проведено асимптотичний аналіз поведінки математичного сподівання максимальної кількості автомобілів на парковці при достатньо великих лінійних розмірах парковки.

Було розглянуто 3 тривіальних моделі паркування автомобілів, 2 з яких є крайовими, тобто визначають верхню та нижню межу кількості автомобілів на парковці.

Було розглянуто 2 нетривіальні моделі паркування автомобілів. Для них виведено аналітичні формули асимптотичної поведінки математичного сподівання максимальної кількості автомобілів на парковці при достатньо великих розмірах парковки:

а) Узагальнення першої моделі. Водії з ймовірністю p ставлять свій автомобіль скраю вільного проміжку, а з ймовірністю $1 - p$ – рівномірно. Отримано результат (2.34).

Аналітично отримані константи знаходяться чисельно, що буде виконано у наступному розділі.

РОЗДІЛ 3 ОЦІНКА ВИЩИХ МОМЕНТІВ

РОЗДІЛ 4 ПРАКТИЧНА ЧАСТИНА

Створений програмний продукт дозволяє визначати середнє значення максимальної кількості автомобілів одиничних лінійних розмірів на парковці прямокутної фіксованих розімірів, при заданій моделі поведінки водіїв.

4.1 Обґрунтування вибору платформи та мови реалізації програмного продукту

Реалізація програмного продукту проведена за допомогою мови C++ та Python. Вибір такого тандему є цілком виправданим. Мова C++ є мовою середнього рівня - в ній присутні елементи мов програмування як низького (підмножина - C), так і високого рівня, що робить C++ дуже ефективною для розробки складних проектів. При цьому виконання елементарних дій (попіксельна обробка зображень) та алгоритмів вцілому залишається максимально швидкою.

Мова Python використовується для швидких математичних розрахунків, в тому числі, для вирахування чисельних значень інтегралів.

Основною платформою для використання програмного продукту було вибрано сімейство операційних систем на базі UNIX, оскільки процес розробки та використання додатків з консольним інтерфейсом там найбільше спрощений. Також, для швидких математичних розрахунків необхідне середовище виконання Python, що дуже просто налаштовується в системах сімейства Linux та в більшості випадків присутнє в стандартній комплектації операційної системи.

Однак, розробка програмного продукту проводилася з дотриманням правил кросплатформеного програмування, що дає можливість збирати виконувані файли і в інших популярних операційних системах (наприклад, сімейства Windows NT).

4.2 Аналіз архітектури продукту

Продукт ”modeler” побудований таким чином, що він зчитує параметри заданою користувачем моделі у форматі xml, та в циклі моделювати траєкторію процесу паркування автомобілів на одновимірній парковці, використовуючи вбудований в C++ генератор псевдовипадкових чисел. За результат програма бере середнє значення по всім змодельованим траєкторіям.

Продукт ”modeler2d” побудований таким чином, що він зчитує параметри заданою користувачем моделі у форматі xml, та в циклі моделювати траєкторію процесу паркування автомобілів на двовимірній парковці, використовуючи вбудований в C++ генератор псевдовипадкових чисел. За результат програма бере середнє значення по всім змодельованим траєкторіям. Ідея багаторазової ітерації з подальшим взяттям середнього ґрунтується на законі великих чисел.

Продукт ”integral” вираховує подвійні інтеграли, наведені у розділі **??**.

4.3 Керівництво користувача

4.3.1 Основний програмний продукт

Програмний продукт розроблено як додаток консольного типу. Тобто, для оперування роботою додатка використовуються текстові команди, що вводяться в консоль операційної системи.

Запуск програми ”modeler” відбувається за допомогою команди:

```
modeler [<model_name>.xml]
```

Запуска програми ”modeler2d” відбувається аналогічно:

```
modeler2d [<model_name>.xml]
```

4.3.1.1 Основні параметри

Користувач задає необхідну для вивчення модель через xml-файл. Типова структура xml-файлу виглядає наступним чином:

```
<model repeat_count="<count>" parking_length="<length>">
  <behaviour_1 p="<prob>"/>
  ...
  <behaviour_n p="<prob>"/>
</model>
```

Тобто в моделі задаються варіанти поведінки водіїв, а також відповідні ймовірності вибору поведінки. Істотним зауваженням є те, що сума ймовірностей має дорівнювати 1, інакше програма завершує роботу з ненульовим кодом.

Параметри моделі для продукту "modeler" перелічені в таблиці 4.1.

Таблиця 4.1 – Параметри моделі для продукту "modeler"

Параметр	Пояснення
`repeat_count`	Кількість запусків моделювань для вирахування середнього значення
`parking_length`	Довжина парковки для моделювання

Типи дисциплін водіїв, що підтримуються продуктом "modeler" перелічені в таблиці 4.2.

Таблиця 4.2 – Дисципліни водіїв, що підтримуються продуктом "modeler"

Дисципліна	Пояснення
`left`	Водій ставить свій автомобіль зліва вільного проміжку
`right`	Водій ставить свій автомобіль справа вільного проміжку
`center`	Водій ставить свій автомобіль по центру вільного проміжку
`uniform`	Водій ставить свій автомобіль керуючись рівномірним розподілом

Для продукту "modeler2d" було підключено лише поведінку водіїв, аналогічну класичній моделі Реньї у випадку одновимірної моделі, тобто за рівномірним розподілом. Тому для цієї програми задаються лише параметри моделі, перелічені в таблиці 4.3.

Таблиця 4.3 – Параметри моделі для продукту ”modeler2d”

Параметр	Пояснення
`repeat_count`	Кількість запусків моделювань для вирахування середнього значення
`a`	Довжина парковки по вісі $0x$
`b`	Довжина парковки по вісі $0y$

4.4 Аналіз роботи алгоритму

4.4.1 Алгоритм чисельного вирахування інтегралів

У **??** розділу 2 було виведено формулу (**??**). Для визначення середнього значення максимальної кількості автомобілів на парковці, керуючись отриманою формулою, необхідно чисельно визначити наступну константу:

$$\kappa = \int_0^{\infty} \exp\left(-2\int_0^{s} \frac{1-e^{-\tau}}{\tau} d\tau\right) ds \tag{4.1}$$

Оскільки чисельно рахувати інтеграл від експоненти від інтегралу незручно, неоптимально та це дасть досить велику похибку, треба звести інтеграл під експонентою до відомих функцій.

Лема 4.4.1:

$$\int_0^{s} \frac{1-e^{-\tau}}{\tau} d\tau = \ln s + \Gamma(0, s) + \gamma, \tag{4.2}$$

де $\gamma = 0.5772156649$ – константа Ейлера-Маскероні, $\Gamma(0, s)$ – неповна гамма-функція.

Доведення.

$$\int\limits_0^s \frac{1-e^{-\tau}}{\tau}d\tau = \int\limits_0^1 \frac{1-e^{-\tau}}{\tau}d\tau + \int\limits_1^s \frac{1}{\tau}d\tau + \int\limits_s^\infty \frac{e^{-\tau}}{\tau}d\tau - \int\limits_1^\infty \frac{e^{-\tau}}{\tau}d\tau =$$

$$= \int\limits_0^1 \frac{1-e^{-\tau}}{\tau}d\tau + \ln s + \Gamma(0,s) - \int\limits_1^\infty \frac{e^{-\tau}}{\tau}d\tau =$$

$$= \ln s + \Gamma(0,s) + \lim_{a\to 0}(1-e^{-\tau})\ln\tau\Big|_0^1 - \int\limits_0^1 e^{-\tau}\ln\tau d\tau -$$

$$- \underbrace{e^{-\tau}\ln\tau\Big|_1^\infty}_{0} - \int\limits_1^\infty e^{-\tau}\ln\tau d\tau = \ln s + \Gamma(0,s) +$$

$$+ \lim_{a\to 0}(1-e^{-\tau})\ln\tau\Big|_0^1 - \int\limits_0^\infty e^{-\tau}\ln\tau d\tau = \lim_{a\to 0}(1-e^{-\tau})\ln\tau\Big|_0^1 +$$

$$+ \ln s + \Gamma(0,s) + \gamma$$

Останній перехід випливає з тотожності

$$\gamma = -\int\limits_0^\infty e^{-\tau}\ln\tau d\tau$$

Якщо довести, що $\lim\limits_{a\to 0}(1-e^{-\tau})\ln\tau\Big|_0^1 = 0$, то лему доведено.

З озкладу у ряд Тейлора випливає, що

$$\lim_{\tau\to 0}\frac{1-e^{-\tau}}{\tau} = 1.$$

Тому

$$\lim_{\tau\to 0}(1-e^{-\tau})\ln\tau = \lim_{\tau\to 0}\frac{1-e^{-\tau}}{\tau}\cdot\frac{\ln u}{u^{-1}} = \langle\text{правило Лопіталя}\rangle = 1\cdot\lim_{\tau\to 0}\frac{u^{-1}}{-u^{-2}} = 0$$

Тому $\lim\limits_{a\to 0}(1-e^{-\tau})\ln\tau\Big|_0^1 = 0$, а отже, лему доведено. □

Таким чином отримали, що

$$\kappa = e^{-2\gamma} \int\limits_0^\infty \frac{\exp(-2\Gamma(0,s))}{s^2} ds \tag{4.3}$$

Для підрахунку невласного інтегралу можна оцінити його залишок, і знайти такі межі інтегрування, щоб залишок не перевищував деякого ε.

Тобто необхідно знайти таке δ, щоб виконувалось

$$e^{-2\gamma} \int\limits_\delta^\infty \frac{\exp(-2\Gamma(0,s))}{s^2} ds < \varepsilon.$$

Оскільки $\exp(-2\Gamma(0,s)) < 1$, то достатньо знайти таке δ, щоб виконувалось

$$\int\limits_\delta^\infty \frac{1}{s^2} < \varepsilon e^{2\gamma}.$$

Оскілки беремо $\delta \geq 0$, то

$$\begin{aligned} \int\limits_\delta^\infty \frac{1}{s^2} &< \varepsilon e^{2\gamma} \\ -\frac{1}{s}\Big|_\delta^\infty &< \varepsilon e^{2\gamma} \\ \delta^{-1} &< \varepsilon e^{2\gamma} \\ \varepsilon^{-1} e^{-2\gamma} &< \delta. \end{aligned}$$

Тобто, щоб отримати результат з точністю 10^{-6}, треба взяти $\delta > 10^6 e^{-2\gamma} \approx 315236$. Це і реалізовано в додатку "integral".

4.4.2 Алгоритм моделювання на одновимірній парковці

Для моделювання одновимірної парковки виконується наступний алгоритм дій:

а) Створюється вектор довжин вільних проміжків, який ініціалізується однією довжиною – довжиною парковки.

б) Допоки вектор не пустий, виконується:

Вибирається поведінка водія використовуючи рандомізатор.

Дістається остання довжина з вектору, і ділиться на частини відповідно до моделі поведінки водія.

Кожна з частин додається до вектору, якщо її довжина не менша 1, тобто, довжини автомобіля.

Інкрементується поточна кількість автомобілів.

Цей процес повторюється задану кількість разів, і в кінці програма видає середнє значення автомобілів з усіх ітерацій, а також відношення до довжини парковки.

4.4.3 Алгоритм моделювання на двовимірній парковці

Двовимірний алгоритм працює аналогічно одновимірному, тільки розглядаються не проміжки, а прямокутники. А саме, створюється список вільних прямокутників. Під вільним розуміється прямокутник, в якому будь-яка точка може бути потенційним центром автомобіля.

Ініціалізується цей список початковим прямокутником – $(\frac{1}{2}, \frac{1}{2}, a-\frac{1}{2}, b-\frac{1}{2})$, де $a,\ b$ – задані лінійні розміри парковки. Спираючись на площі поточних прямокутників, обирається один з ймовірністю рівною відношенню його площі до сумарної площі усіх вільних прямокутників. Після цього всередині обраного прямокутника обирається за рівномірним розподілом точка, навколо якої будується квадратний окіл розміру 2x2, і перевіряється на перетин з усіма поточними вільними прямокутниками. Якщо перетинається, то вільний прямокутник ділиться на менші прямокутники, при чому один із прямокутників – область перетину. Усі менші прямокутники, окрім перетину, додаються в список.

4.5 Результати роботи програми

4.5.1 Результати підрахунку констант

=

Для формули (2.34) було чисельно пораховано коефіцієнт κ_α, для кожного α від 0 до 0.9 з кроком 0.1. Результати наведені в таблиці 4.4.

Таблиця 4.4 – Результати розрахунку константи для узагальненої моделі

α	0	0.1	0.2	0.3	0.4
κ_α	0.747598	0.76351	0.780574	0.798962	0.818896
α	0.5	0.6	0.7	0.8	0.9
κ_α	0.84066	0.864638	0.891365	0.92165	0.956849

4.5.2 Результати роботи моделера

Для формули (2.34) було промодельовано поведінку водіїв і емпірично визначено коефіцієнт κ_α, для кожного α від 0 до 0.9 з кроком 0.1. Результати наведені в таблиці 4.5.

Таблиця 4.5 – Результати розрахунку константи для узагальненої моделі

α	0	0.1	0.2	0.3	0.4
κ_α	0.747588	0.76352	0.780569	0.798959	0.818891
α	0.5	0.6	0.7	0.8	0.9
κ_α	0.84055	0.863658	0.891214	0.92158	0.95693

Нескладно помітити, що теоретичний результат співпав з результатом моделювання з точністю до 4 знаку.

Для двовимірного випадку існує припущення, що відношення середньої кількості автомобілів до загальної площі парковки прямує до $\kappa^2 \approx 0.56$, але це не доведений факт [1]. Було проведено експеримент на розмірах 50x50, 100x100, 200x200, і результати вийшли відповідно 0.7, 0.6 та 0.58, тобто ймовірно, що припущення правдиве, але для перевірки на дуже великих розмірах парковки необхідні дуже потужні обчислювальні ресурси через велику складність двовимірного алгоритму.

4.6 Висновки до розділу

В даному розділі описано структурну схему розробленого програмного продукту, обґрунтовано вибір цільової ОС, надана коротка інструкція з експлуатації програмного продукту. Програмний продукт є пакетом з декількох програм, які слугують одній цілі – визначення математичного сподівання максимальної кількості автомобілів на парковці, використовуючи аналітичний та алгоритмічний методи. Під час зіставлення результатів виявилося, що аналітичний та алгоритмічний підходи дають майже однакові результати. Перевагою аналітичного методу є швидкість і досить висока точність, в той час як алгоритмічний метод дозволяє більш гнучко налаштувати поведінку водіїв на парковці.

Також була проведена в певному сенсі успішна спроба підтвердити гіпотезу щодо двовимірної парковки.

ВИСНОВКИ ПО РОБОТІ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Одним з досить серйозних пороків людського виду є постійне намагання підлаштувати природу під себе. Дійсно, вже рідко де, окрім заповідників, знайдеш чисту і недоторкану природу, адже куди не глянь – усюди забудови, дороги, машини і трагічні наслідки техногенних катастроф.

В той самий час доситьактивно розвивається медицина, за останні два сторіччя істотно виріс середній вік людини, тож наша популяція стає все більше і більше. А чим більше людей, тим більше потребується житлових місць, харчів, автомобілів. І досить часто все це створюється бездумно, без розрахунку.

Саме тому у 1958 році угорський математик Реньї порушив проблему паркування автомобілів в теоретичному аспекті. І хоча вона називається проблемою паркування, результати, отримані під час її розв'язування можуть бути використані не лише до автомобілів. Їх можна застосувати для оптимального пакування, оптимального розміщення будівель, навіть для оптимальної організації робочого простору.

У даній атестаційній роботі було узагальнено класичну модель паркування Реньї. Було отримано наступні результати.

Проведено дослідження існуючих підходів і результатів щодо проблеми паркування та пакування. Проведено аналіз аналітичного апарату, що дозоляє вирішувати інтегральні рівняння зі зсувом, таким чином, допомагає у вирішенні проблем розглянутого класу. Було створено узагальнену модель паркування, в якій водії паркуються керуючись сумішшю рівномірного розподілу та розподілу Бернулі. Для цього узагальнення було виведено аналітичну формулу математичного сподівання максимальної кількості автомобілів за великих розмірів парковки. Це узагальнення дає змогу використовувати модель Реньї на практиці, адже насправді існують сумлінні водії, які намагаються зайняти якомога менше місця на парковці, а є такі, що ставлять свій автомобіль як заманеться.

Паралельно з виведенням аналітичного результату був створений консольний додаток, що дозволяє моделювати процес паркування імітаційним

шляхом, і визначати математичне сподівання максимальної кількості автомобілів як середнє значення результатів кількох імітацій.

Було перевірено, що обидва методи дають один і той самий результат з точністю до четвертого знаку після коми на класичній моделі Реньї та на узагальненій з 10 різними ймовірностями паркування автомобіля зкраю.

За аналізом отриманих результатів роботи програмного продукту можна зробити висновок, що обрані методи дозволили досягти виконання поставлених цілей з достатньо великою степінню точності. Але у кожного метода є свої плюси та мінуси. Перевагою аналітичного методу є швидкість і досить висока точність, в той час як алгоритмічний метод дозволяє більш гнучко налаштувати поведінку водіїв на парковці.

Подільшими дослідженнями за даним напрямом можуть стати:

- отримання аналітичного результату для асимптотики у випадку двовимірної парковки;
- зменшення складності алгоритму для двовимірної парковки;
- підтримка додатком змінних розмірів автомобілів;
- створення додатку для тривимірного розміщення.

СПИСОК ЛІТЕРАТУРИ

1. Weisstein, Eric W. Rényi's Parking Constants / Eric W. Weisstein. — 2016. [Електронний ресурс]. — Режим доступу: http://mathworld.wolfram.com/RenyisParkingConstants.html.
2. Rényi, A. On a One-Dimensional Problem Concerning Random Space-Filling. / A. Rényi. — Budapest: Math. Inst. Hung. Acad., 1958. — P. 153.
3. Dvoretzky, A. On the Parking Problem. / A. Dvoretzky, H. Robbins. — Budapest: Math. Inst. Hung. Acad., 1964. — P. 275.
4. Blaisdell, B. E. On Random Sequential Packing in the Plane and a Conjecture of Palasti. / B. E. Blaisdell, H. Solomon. — Budapest: J. Appl. Prob., 1970. — P. 834.
5. Feller, William. An Introduction To Probability Theory and Its Applications. / William Feller. — 2 edition. — New-York: John Wiley & Sons, 1971. — Vol. 2. — P. 704.
6. Schiff, J.L. The Laplace Transform: Theory and Applications / J.L. Schiff. Undergraduate Texts in Mathematics. — Springer New York, 1999. [Електронний ресурс]. — Режим доступу: https://books.google.com.ua/books?id=RU-5jSPlKMcC.
7. Лаврентьев, М.А. Методы теории функций комплексного переменного / М.А. Лаврентьев, Б.В. Шабат. — Наука, 1965. [Електронний ресурс]. — Режим доступу: https://books.google.com.ua/books?id=My7vAAAAMAAJ.
8. Кремер, Н.Ш. Теория вероятностей и математическая статистика. / Н.Ш. Кремер. — 2 изд. — М.: ЮНИТИ-ДАНА, 2004. — С. 573.

ДОДАТОК А. **ТЕОРЕТИЧНИЙ МІНІМУМ**

А.1 Поняття випадкової величини

Одним з найважливішим поняттям в теорії ймовірностей є поняття випадкової величини. Під випадковою величиною розуміють змінну, яка в результаті дослідження в залежності від випадку приймає одне з можливої множини своїх значень (яке саме – заздалегідь невідомо).

Означення А.1.1: Випадковою величиною X називається функція, що задана у просторі елементарних подій Ω, тобто $X \in f(\omega)$, де ω – елементарна подія, що належить простору Ω [8].

Для дискретної випадкової величини множина можливих значеньфункції $f(\omega)$ скінчена або злічена, для неперервної – нескінченна або незлічена.

Означення А.1.2: Законом розподілу випадкової величини називається відношення, що встановлює зв'язок між можливим значеннями випадкової величини і відповідними їм ймовірностями [8].

А.1.1 Представлення випадкових величин

Означення А.1.3: Функцією розподілу випадкової величини X називається функція $F(X)$, котра виражає для кожного x ймовірність того, що випадкова величина X прийме значення, яке менше за x [8]:

$$F(x) = \mathsf{P}\{X < x\} \tag{А.1}$$

Означення А.1.4: Випадкова величина називається дискретною, якщо вона набуває скінчену або злічену множину значень.

Означення А.1.5: Випадкова величина називається неперервною, якщо її функція розподілу є неперервною, диференційованою майже скрізь, за винятком, можливо, окремих ізольованих точок [8].

Функцію $F(X)$ іноді називають інтегральною функцією розподілу або інтегральним законом розподілу. Визначення неперервної випадкової величини за допомогою функції розподілу не є єдиним.

Означення А.1.6: Щільністю ймовірності (щільністю розподілу або просто щільністю) $f(x)$ неперервної випадкової величини X називається похідна її функції розподілу $f(x) = F'(x)$ [8].

Щільність ймовірності іноді називають диференціальною функцією розподілу або диференціальним законом розподілу. А графік $f(x)$ – кривою розподілу.

А.1.2 Числові характеристики випадкових величин

Означення А.1.7: Математичним сподіванням, або середнім значенням $\mathbb{E}(X)$ дискретної випадкової величини X називається сума добутків всіх його значень на відповідні їм ймовірності [8]:

$$\mathbb{E}(X) = \sum_{i=1}^{n} x_i \cdot p_i \tag{А.2}$$

Означення А.1.8: Математичне сподівання неперервної випадкової величини визначаться [8]:

$$\mathbb{E}(X) = \int_{-\infty}^{\infty} x f(x) dx \tag{А.3}$$

Означення А.1.9: Дисперсією $\mathbb{D}(X)$ випадкової величини X називається математичне сподівання квадрата її відхилення від математичного сподівання: $\mathbb{D}(X) = \mathbb{E}(X - \mathbb{E}X)^2$. Або $\mathbb{D}(X) = \mathbb{E}(X - a)^2,$ де $a = \mathbb{E}X$.

Якщо X – дискретна, то

$$\mathbb{D}(X) = \sum_{i=1}^{n} (x_i - \mathbb{E}X)^2 \cdot p_i \tag{А.4}$$

Якщо X – неперервна, то

$$\mathbb{D}(X) = \int_{-\infty}^{\infty} (x - \mathbb{E}X)^2 f(x) dx \tag{А.5}$$

Означення А.1.10: Середнім квадратичним відхиленням (стандартним відхиленням або стандартом) σ_X випадкової величини X називається арифметичним значенням кореня квадратного з дисперсії: $\sigma_X = \sqrt{\mathbb{D}X}$

Математичне сподівання $\mathbb{E}X$, або перший початковий момент, характеризує середнє значення або положення розподілу випадкової величини X на числовій осі; дисперсія $\mathbb{D}X$, або другий центральний момент μ_2, – величину розсіювання розподілу X відносно $\mathbb{E}X$ [**?**].

А.2 Перетворення Лапласа

Перетворення Лапласа - інтегральне перетворення, що зв'язує функцію $F(s)$ комплексної змінної (зображення) з функцією $f(t)$ дійсного змінного (оригінал). З його допомогою досліджуються властивості динамічних систем і вирішуються диференціальні і інтегральні рівняння.

Однією з особливостей перетворення Лапласа, які визначили його широке поширення в наукових і інженерних розрахунках, є те, що багатьом співвідношенням і операціям над оригіналами відповідають простіші співвідношення над їх зображеннями. Так, згортка двох функцій зводиться в просторі зображень до операції множення, а лінійні диференціальні рівняння стають алгебраїчними.

Означення А.2.1: Перетворенням Лапласа функції дійсної змінної $f(t)$ називається функція $F(s)$ комплексної змінної $s = \theta + i\omega$ така, що:

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^\infty e^{-st} f(t)dt \tag{А.6}$$

при цьому права частина виразу називається інтегралом Лапласа [**?**].

Перетворення Лапласа існує, якщо $f(t) : \mathbb{R}^+ \cup 0 \to \mathbb{R}$ – інтегровна на будь-якому підінтервалі $[0, +\infty)$ і виконується нерівність $|f(t)| < Ke^{\omega t}$ для деяких фіксованих $K > 0,\ \omega \geq 0$.

А.2.1 Властивості перетворення Лапласа

1. Перетворення Лапласа лінійне, тобто для $\forall \alpha, \beta \in \mathbb{C}$

$$\mathcal{L}\{\alpha f(t) + \beta g(t)\} = \alpha F(s) + \beta G(s) \tag{A.7}$$

2. Теорема подібності. Для $\forall \alpha \in \mathbb{R}^+$

$$\mathcal{L}\{f(\alpha t)\} = \frac{1}{\alpha} F(\alpha s) \tag{A.8}$$

3. Диференціювання оригінала. Якщо перетворення Лапласа існує для $f'(t)$, то

$$\mathcal{L}\{f'(t)\} = sF(s) - f(0) \tag{A.9}$$

4. Диференціювання зображення. Зводиться до домноження оригіналу на $-t$:

$$\mathcal{L}^{-1}\{F'(s)\} = -t \cdot f(t) \tag{A.10}$$

або, взагалі:

$$\mathcal{L}^{-1}\left\{F^{(n)}(s)\right\} = (-t)^n \cdot f(t) \tag{A.11}$$

5. Інтегрування оригіналу. Зводиться до поділу зображення Лапласа на s:

$$\mathcal{L}\left\{\int_0^t f(\tau)d\tau\right\} = \frac{F(s)}{s} \tag{A.12}$$

6. Інтегрування зображення. Якщо інтеграл

$$\int_s^\infty F(p)dp$$

збігається, то він є зображенням для функції $\frac{f(t)}{t}$:

$$\mathcal{L}\left\{\frac{f(t)}{t}\right\} = \int_s^\infty F(p)dp \tag{A.13}$$

7. Теорема зміщення. Для $\forall s_0 \in \mathbb{C}$:

$$\mathcal{L}\{e^{s_0 t} f(t)\} = F(s - s_0) \tag{A.14}$$

8. Теорема запізнення. Для $\forall t_0 > 0$

$$\mathcal{L}\{f(t - t_0)\} = e^{-st_0} F(s) \tag{A.15}$$

або більш загально, $\forall t \in \mathbb{R}$

$$\mathcal{L}\{f(t - t_0)\eta(t - t_0)\} = e^{-st_0} F(s) \tag{A.16}$$

де $\eta(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$ – функція Хевісайда.

Важливою для застосування є наступна теорема:

9. Теорема єдиності. Якщо дві функції $f_1(t)$ та $f_1(t)$ мають одне й те саме зображення Лапласа $F(s)$, то ці функції тотожно рівні.

Означення А.2.2: Згорткою двох функцій $f(t)$ та $g(t)$ називається

$$(f * g)(t) = \int_{-\infty} \infty f(\tau) g(t - \tau) d\tau \tag{A.17}$$

10. Теорема про згортку. Зображенням згортки двох функцій є добуток зображень Лапласа цих функцій:

$$\mathcal{L}\{(f * g)(t)\} = F(s) \cdot G(s) \tag{A.18}$$

Для визначення зв'язку між асимптотичною поведінкою оригіналу та зображення блої використано теорему Таубера.

ДОДАТОК Б. **ЛІСТИНГ КОДУ**

Моделювання процесу парковки

appendicies/cppcode/modeler.cpp

```cpp
#include <boost/property_tree/ptree.hpp>
#include <boost/property_tree/xml_parser.hpp>
//#include <boost/random/uniform_real_distribution.hpp>
//#include <boost/random/normal_distribution.hpp>
#include <random>
#include <string>
#include <vector>
#include <map>
#include <algorithm>
#include <numeric>
#include <cmath>
#include <iostream>
#include <chrono>

struct model
{

  enum behaviour_type {
    place_left,
    place_right,
    place_center,
    place_uniform,
    place_normal
  };

  const static std::map<std::string, behaviour_type> converter;
  const static std::map<behaviour_type, std::string> back_converter;
  const static double eps;
  //typedef boost::random::uniform_real_distribution<double> uni_d_type;
  typedef std::uniform_real_distribution<double> uni_d_type;

  size_t repeat_count = 100;
  double parking_length = 1.0;

  std::vector<std::pair<double, behaviour_type>> strategy;
  void load(const std::string& path);
  void save(const std::string& path);

  int fit();
  inline double fit_avg();
```

```
41    void fit_avg_regularized(std::ostream&);

  private:
    void split_last(behaviour_type, std::vector<double>&);
    uni_d_type uni_d{0,1};
46  unsigned seed = std::chrono::system_clock::now().time_since_epoch().count();
    std::default_random_engine generator{seed};
  };

  const double model::eps = 1e-7;
51 const std::map<std::string, model::behaviour_type> model::converter =
    {
      {"left", model::place_left},
      {"right", model::place_right},
      {"center", model::place_center},
56    {"uniform", model::place_uniform},
      {"normal", model::place_normal}
    };
  const std::map<model::behaviour_type, std::string> model::back_converter =
    {
61    {model::place_left, "left"},
      {model::place_right, "right"},
      {model::place_center, "center"},
      {model::place_uniform, "uniform"},
      {model::place_normal, "normal"}
66  };

  void model::split_last(behaviour_type b, std::vector<double>& v)
  {
    double len = v.back();
71  v.pop_back();
    switch(b) {
      case place_left:
      case place_right:
        if (len >= 2)
76        v.push_back(len - 1);
        break;
      case place_center:
        if (len >= 3)
          v.insert(v.end(), 2, (len - 1) / 2);
81      break;
      case place_uniform:
        if (len >= 2) {
          double pos = uni_d_type(0, len - 1)(generator);
          double residuals[2] = {pos, len - 1 - pos};
86        for (auto resid : residuals)
            if (resid >= 1) {
```

```
            v.push_back(resid);
          }
        }
        break;
      case place_normal:
        throw(std::runtime_error("Not implemented"));
        break;
    }
}

int model::fit()
{
  if (parking_length < 1)
    return 0;
  std::vector<double> lengths{parking_length};
  int counter = 0;
  while(!lengths.empty())
  {
    double acc = 0, p = uni_d(generator);
    for (auto& entry : strategy)
    {
      acc += entry.first;
      if (p < acc) {
        split_last(entry.second, lengths);
        counter++;
        break;
      }
    }
  }
  return counter;
}

inline double model::fit_avg()
{
  double sum = 0.0;
  for (int i = 0; i < repeat_count; i++)
    sum += fit();
  return sum / repeat_count;
}

void model::fit_avg_regularized(std::ostream& out)
{
  out << "Filled ";
  double result = 0;
  if (parking_length < 1)
    out << result;
  else
```

```
    out << (result = fit_avg());
  out << "/" << parking_length << "␣with␣ratio:␣";
  if (parking_length != 0)
    out << result / parking_length;
  else
    out << "inf";
  out << '.' << std::endl;
}

void model::load(const std::string& path)
{
  strategy.clear();
  using boost::property_tree::ptree;
  ptree pt;
  read_xml(path, pt);
  repeat_count = pt.get<size_t>("model.<xmlattr>.repeat_count", repeat_count);
  parking_length = pt.get<double>("model.<xmlattr>.parking_length",
    parking_length);
  std::cout << "Loaded␣parking_length:␣" << parking_length << std::endl;
  std::cout << "Loaded␣repeat_count:␣" << repeat_count << std::endl;
  for (auto& entry : pt.get_child("model"))
  {
    if (entry.first == "<xmlattr>")
      continue;
    strategy.emplace_back(entry.second.get<double>("<xmlattr>.p"),
    converter.at(entry.first));
    std::cout << "Added␣" << entry.first << "␣behaviour␣with␣probability␣" <<
    strategy.back().first << std::endl;
  }
  auto sum = std::accumulate(strategy.begin(), strategy.end(), 0.0, [](double
    acc, std::pair<double, behaviour_type> next) {return acc + next.first;});
  assert(std::abs(sum - 1.0) < eps);
}

void model::save(const std::string& path)
{
  using boost::property_tree::ptree;
  ptree pt;
  pt.put("model.<xmlattr>.repeat_count", repeat_count);
  pt.put("model.<xmlattr>.parking_length", parking_length);
  for (auto& entry : strategy)
  {
    pt.put("model." + back_converter.at(entry.second) + ".<xmlattr>.p",
    entry.first);
  }
  write_xml(path, pt);
}
```

```
int main(int argc, char* argv[]) {
  model m;
  if (argc > 1)
    m.load(argv[1]);
  else
    m.load("base_model.xml");
  m.fit_avg_regularized(std::cout);
  return 0;
}
```

Моделювання процесу двовимірної парковки

appendicies/cppcode/modeler2d.cpp

```
#include <boost/property_tree/ptree.hpp>
#include <boost/property_tree/xml_parser.hpp>
//#include <boost/random/uniform_real_distribution.hpp>
//#include <boost/random/normal_distribution.hpp>
#include <random>
#include <string>
#include <vector>
#include <map>
#include <algorithm>
#include <numeric>
#include <cmath>
#include <iostream>
#include <chrono>
#include <list>
#include <tuple>


struct model2d
{
    double a;
    double b;
    // only uniform distribution is supported here

    struct point
    {
        point(double x, double y) : x(x), y(y) { }
        double x;
        double y;
    };

    typedef std::pair<point, point> rectangle;
```

```
    typedef std::uniform_real_distribution<double> uni_d_type;

    int fit();
    inline double fit_avg();
    void fit_avg_regularized(std::ostream& out);

    model2d(int a, int b, size_t repeat_count = 100) : a(a), b(b),
    repeat_count(repeat_count) { }

private:
    size_t repeat_count;
    double eps = 1e-6;
    double rec_square(const rectangle&);
    void consume(std::list<rectangle>& recs, const rectangle& exclusion);
    bool does_intersect(const rectangle&, const rectangle&);
    double current_square;
    unsigned seed = std::chrono::system_clock::now().time_since_epoch().count();
    std::default_random_engine generator{seed};
};

double model2d::rec_square(const rectangle& rec) {
    return (rec.second.x - rec.first.x) * (rec.second.y - rec.first.y);
}

bool model2d::does_intersect(const rectangle& left, const rectangle& right)
{
    return std::max(left.first.x, right.first.x) < std::min(left.second.x,
    right.second.x) &&
    std::max(left.first.y, right.first.y) < std::min(left.second.y,
    right.second.y);
}

void model2d::consume(std::list<rectangle>& recs, const rectangle& exclusion)
{
    auto g_iterator = recs.begin();
    std::list<rectangle> additions;

    //std::cout << "Exclusion(" << exclusion.first.x << "," << exclusion.first.y
    << "," <<
    //exclusion.second.x << "," << exclusion.second.y << ")" << std::endl;
    while (g_iterator != recs.end())
    {
        auto iterator = g_iterator++;
        if (does_intersect(*iterator, exclusion)) {
            // DO SOME EPIC STUFF HERE
            std::vector<std::pair<double,double>> x_sides, y_sides;
```

```
74          if (exclusion.first.x > iterator->first.x && exclusion.second.x <
    iterator->second.x)
            {
                x_sides.push_back(std::make_pair(iterator->first.x,
    exclusion.first.x));
                x_sides.push_back(std::make_pair(exclusion.first.x,
    exclusion.second.x));
                x_sides.push_back(std::make_pair(exclusion.second.x,
    iterator->second.x));
79          }
            else if (exclusion.first.x > iterator->first.x && exclusion.first.x <
    iterator->second.x)
            {
                x_sides.push_back(std::make_pair(iterator->first.x,
    exclusion.first.x));
                x_sides.push_back(std::make_pair(exclusion.first.x,
    iterator->second.x));
84          }
            else if (exclusion.second.x > iterator->first.x && exclusion.second.x
    < iterator->second.x)
            {
                x_sides.push_back(std::make_pair(iterator->first.x,
    exclusion.second.x));
                x_sides.push_back(std::make_pair(exclusion.second.x,
    iterator->second.x));
89          }
            else
            {
                x_sides.push_back(std::make_pair(iterator->first.x,
    iterator->second.x));
            }
94
            if (exclusion.first.y > iterator->first.y && exclusion.second.y <
    iterator->second.y)
            {
                y_sides.push_back(std::make_pair(iterator->first.y,
    exclusion.first.y));
                y_sides.push_back(std::make_pair(exclusion.first.y,
    exclusion.second.y));
99              y_sides.push_back(std::make_pair(exclusion.second.y,
    iterator->second.y));
            }
            else if (exclusion.first.y > iterator->first.y && exclusion.first.y <
    iterator->second.y)
            {
                y_sides.push_back(std::make_pair(iterator->first.y,
    exclusion.first.y));
```

```
104                 y_sides.push_back(std::make_pair(exclusion.first.y,
    iterator->second.y));
            }
            else if (exclusion.second.y > iterator->first.y && exclusion.second.y
    < iterator->second.y)
            {
                y_sides.push_back(std::make_pair(iterator->first.y,
    exclusion.second.y));
109                 y_sides.push_back(std::make_pair(exclusion.second.y,
    iterator->second.y));
            }
            else
            {
                y_sides.push_back(std::make_pair(iterator->first.y,
    iterator->second.y));
114             }

            for (auto& x_side : x_sides)
                for (auto& y_side : y_sides) {
                    rectangle new_rec = std::make_pair(point(x_side.first,
    y_side.first), point(x_side.second, y_side.second));
119                     if (!does_intersect(new_rec, exclusion))
                        additions.push_back(new_rec);
                    else
                        current_square -= rec_square(new_rec);
                    assert(rec_square(new_rec) >= 0);
124                 }
            recs.erase(iterator);
        }
    }
    recs.splice(recs.end(), additions);
129 }

int model2d::fit()
{
    if (a < 1 || b < 1)
134     return 0;
    std::list<rectangle> rectangles{{{0,0}, {a,b}}};
    current_square = a * b;
    int counter = 0;
    while(!rectangles.empty() && !(current_square <= eps))
139 {
        auto decision = uni_d_type(0, current_square)(generator);
        auto current_rect = std::find_if(rectangles.begin(), rectangles.end(),
    [decision](rectangle& rec) {
            static double sum = 0;
            sum += (rec.second.x - rec.first.x) * (rec.second.y - rec.first.y);
```

```
144             if (sum >= decision)
                    return true;
                return false;
            });
            auto x = uni_d_type(current_rect->first.x,
        current_rect->second.x)(generator);
149         auto y = uni_d_type(current_rect->first.y,
        current_rect->second.y)(generator);
            rectangle exclusion = std::make_pair(point(x - 1, y - 1), point(x + 1, y
        + 1));
            consume(rectangles, exclusion);
            counter++;
        }
154     return counter;
    }

    inline double model2d::fit_avg()
    {
159     double sum = 0.0;
        for (int i = 0; i < repeat_count; i++)
            sum += fit();
        return sum / repeat_count;
    }
164
    void model2d::fit_avg_regularized(std::ostream& out)
    {
        out << "Filled ";
        double result = 0;
169     if (a < 1 || b < 1)
            out << result;
        else
            out << (result = fit_avg());
        out << "/" << a*b << " with ratio: ";
174     if (a * b != 0)
            out << result / (a * b);
        else
            out << "inf";
        out << '.' << std::endl;
179 }


    int main()
    {
184     model2d m {100, 100};
        m.fit_avg_regularized(std::cout);
        return 0;
    }
```

Вирахування константи для класичної моделі Реньї

appendicies/cppcode/integral.cpp

```cpp
#include <boost/math/special_functions/gamma.hpp>
#include <boost/math/constants/constants.hpp>

#include <iostream>
#include <cmath>

using namespace boost::math;

int main() {
    double step = 1e-4;
    long double integral = 0;
    for (int i = 1; i <= 1000000; i++)
        integral += std::exp(- 2 * (tgamma(1e-10, i * step)) - 2 * std::log(i * step)) * step;
    step *= 10;
    for (int i = 100001; i <= 10000000; i++)
        integral += std::exp(- 2 * (tgamma(1e-10, i * step)) - 2 * std::log(i * step)) * step;
    step *= 10;
    for (int i = 1000001; i <= 10000000; i++)
        integral += std::exp(- 2 * (tgamma(1e-10, i * step)) - 2 * std::log(i * step)) * step;
    std::cout << integral / std::exp(2 * constants::euler<double>()) << std::endl;
    return 0;
}
```