

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

Кафедра прикладної математики та моделювання складних систем

Допущено до захисту  
Завідувач кафедри ПМ та МСС  
\_\_\_\_\_ Коплик І.В.  
(підпис)  
« \_\_\_\_ » \_\_\_\_\_ 2021р.

**КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття освітнього ступеня «бакалавр»

спеціальність 113 «Прикладна математика»  
освітньо-професійна програма «Прикладна математика»

тема роботи «**ЧИСЕЛЬНЕ ДОСЛІДЖЕННЯ ДИНАМІКИ  
БРІЗЕРА НЕЛІНІЙНОГО ХВИЛЬОВОГО РІВНЯННЯ,  
ЗА ДОПОМОГОЮ ЗАСОБІВ MATLAB**»

**Виконавець**

студент факультету ЕЛІТ

Лютий Антон Тарасович \_\_\_\_\_  
(прізвище, ім'я, по батькові) (підпис)

**Науковий керівник**

Ph.D., Adjunct Professor  
(науковий ступінь, вчене звання)

Kilchuk, Viktor \_\_\_\_\_  
(прізвище, ім'я, по батькові) (підпис)

# СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Факультет **електроніки та інформаційних технологій**  
Кафедра **прикладної математики та моделювання складних систем**

Рівень вищої освіти Перший (бакалавр)  
(перший (бакалавр) або другий (магістр))

Галузь знань **11 Математика та статистика**  
Спеціальність **113 Прикладна математика**  
Освітня програма **освітньо-професійна «Прикладна математика»**

ЗАТВЕРДЖУЮ  
Завідувач кафедри ПМ та МСС  
Коплик І.В. \_\_\_\_\_  
«\_\_» \_\_\_\_\_ 2021р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Лютий Антон Тарасович

(прізвище, ім'я, по батькові)

1. Тема роботи Чисельне дослідження динаміки брізера  
нелінійного хвильового рівняння, за допомогою засобів MatLab

Керівник роботи Kilchyk, Viktor, Ph.D., Adjunct Professor  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання, посада)

затверджено наказом факультету ЕЛІТ від «14» квітня 2021р. № 0180-VI

2. Термін подання роботи студентом «31» травня 2021р.

3. Вихідні дані до роботи Хвильове рівняння, явні чисельні методи,  
неявні чисельні методи, штучна в'язкість.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) Вступ. Аналітичний огляд. Методика досліджень.

Реалізація комп'ютерного експерименту. Висновки. Перелік використаних джерел. Додатки.



## РЕФЕРАТ

**Кваліфікаційна робота:** 53 с., 105 рисунків, 0 таблиць, 22 джерела.

**Мета роботи:** дослідити одновимірне хвильове рівняння, створити модель області простору з штучно введеною в'язкістю за допомогою засобів програмування, автоматизувати процес підбору коефіцієнтів в'язкості, та дослідити хвильові процеси в даній області.

**Об'єкт дослідження:** явні та неявні чисельні методи, і хвильові процеси у області простору зі штучно введеною в'язкістю.

**Предмет дослідження:** математична модель одновимірного хвильового рівняння зі штучно введеною в'язкістю за допомогою дисперсійного або дифузійного доданку з коефіцієнтом.

**Методи навчання:** рівняння математичної фізики, чисельні методи, моделювання хвильових процесів, засоби програмування в MatLab.

**Ключові слова:** хвильове рівняння, явні чисельні методи, неявні чисельні методи, різницеві схеми, штучна в'язкість, машинна оптимізація.

## ЗМІСТ

ВСТУП.....	4
1 АНАЛІТИЧНИЙ ОГЛЯД .....	5
1.1 Хвильове рівняння .....	5
1.1.1 Одновимірний випадок.....	6
1.1.2 Рівняння синус – Гордона.....	7
1.2 Граничні умови.....	7
1.3.1 Граничні умови Діріхле.....	8
1.3.2 Граничні умови Неймана .....	9
1.3.3 Граничні умови Робена .....	10
1.3 Аналіз стабільності фон Неймана .....	11
2 МЕТОДИКА ДОСЛІДЖЕНЬ .....	12
2.1 Методи скінченних різниць .....	12
2.2 Явні чисельні методи.....	13
2.2.1 Прямий метод Ейлера.....	14
2.2.2 Методи Рунге – Кутти .....	15
2.3 Неявні чисельні методи .....	17
2.3.1 Непрямий метод Ейлера.....	18
2.3.2 Метод Кранка – Ніколсона .....	19
2.4 Розрахунок енергії для біжучої хвилі .....	20
2.5 Штучна в'язкість .....	21
2.6 Машинна оптимізація .....	22
3 РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОГО ЕКСПЕРИМЕНТУ .....	24
3.1 Дослідження за допомогою прямого методу Ейлера .....	24

3.2 Дослідження за допомогою методу Рунге – Кутта 4 порядку.....	27
3.3 Дослідження за допомогою непрямого методу Ейлера .....	30
3.4 Дослідження за допомогою методу Кранка – Ніколсона .....	33
3.5 Дослідження результатів машинної оптимізації .....	36
3.5.1 Лінійна функція.....	37
3.5.2 Степенева функція .....	38
3.5.3 Коренева функція.....	40
3.5.4 Кубічна крива Безьє .....	41
ВИСНОВКИ.....	43
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	45
ДОДАТОК А .....	47
ДОДАТОК Б .....	48
ДОДАТОК В .....	49
ДОДАТОК Г .....	50
ДОДАТОК Ґ.....	51

## ВСТУП

При моделюванні хвильових процесів іноді виникає необхідність поглинання/розсіювання хвиль, коли не потрібне їх розповсюдження, або необхідно спостерігати за ними тільки в певній області. Тому, метою даної кваліфікаційної роботи було дослідження одновимірного хвильового рівняння зі штучною в'язкістю та написання програми, яка для вхідних хвиль різних типів з різними параметрами буде підбирати закон розподілення коефіцієнтів в'язкості та давати на виході масив коефіцієнтів, що приводитимуть до отримання оптимального результату.

На першому етапі одновимірне хвильове рівняння було досліджене різними чисельними методами, явними та неявними, як-от прямий і непрямий методи Ейлера, метод Рунге – Кутти 4 порядку, метод Кранка – Ніколсона. Дослідження проводилися з метою оволодіння даними методами та набуття навичок та досвіду для кращого орієнтування у темі.

Другим етапом був вибір способів задання штучної в'язкості, їх аналіз та аналіз чисельних розв'язків одновимірного хвильового рівняння, які були отримані з використанням досліджуваних способів. Було обрано два способи, які придатні до подальшого застосування.

Фінальним етапом кваліфікаційної роботи була розробка алгоритму для автоматизації підбору коефіцієнтів штучної в'язкості за допомогою засобів програмування, та його програмна реалізація.

# 1 АНАЛІТИЧНИЙ ОГЛЯД

## 1.1 Хвильове рівняння

Одним із основних рівнянь математичної фізики є хвильове рівняння – лінійне гіперболічне диференціальне рівняння у часткових похідних другого порядку, що використовується для опису хвиль в класичній фізиці – таких як механічні хвилі (наприклад, хвилі води, звукові хвилі та сейсмічні хвилі) або світлові хвилі. Знаходить застосування і в інших областях, наприклад при описі гравітаційних хвиль.

Історично проблему вібрації струн, таких як в музичних інструментах, вивчали Жан ле Рон Д'Аламбер, Леонард Ейлер, Даніель Бернуллі та Джозеф-Луї Лагранж [1-2]. У 1746 р. Д'Аламбер відкрив одновимірне хвильове рівняння, а протягом десяти років Ейлер відкрив тривимірне хвильове рівняння [3].

Розв'язки рівняння мають застосування в таких областях, як акустика, електромагнетизм та динаміка рідини. Особливо важливими є рішення перетворення Фур'є хвильового рівняння, що визначають ряди Фур'є, сферичні гармоніки та їх узагальнення.

У загальному вигляді для  $n$ -мірного випадку має такий запис:

$$u_{tt} = c^2 \Delta u + f(x, t) \quad (0.1)$$

де  $u = u(x_1, x_2, \dots, x_n; t)$  – шукана функція від змінної  $t$  та однієї або багатьох просторових змінних  $x_1, x_2, \dots, x_n$ ,  $\Delta = \nabla^2 = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \dots + \frac{\partial^2}{\partial x_n^2}$  – оператор Лапласа,  $c$  – фазова швидкість, та  $f(x, t)$  – деяка функція, що описує вплив зовнішніх сил.

Рішення цього рівняння може бути досить складним, але його можна проаналізувати як лінійну комбінацію простих рішень, які представляють собою синусоїдальні плоскі хвилі з різними напрямками поширення та довжинами



хвиль, але з однаковою швидкістю поширення  $c$ . Цей аналіз можливий, оскільки хвильове рівняння є лінійним – будь-який кратний розв’язок також є розв’язком, а сума будь-яких двох розв’язків знову є розв’язком. Ця властивість називається принципом суперпозиції у фізиці.

### 1.1.1 Одновимірний випадок

В одновимірному випадку хвильове рівняння називається також рівнянням коливання струни або рівнянням поздовжніх коливань стрижня, та записується у вигляді:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} + f(x, t) \quad (0.2)$$

Лінійний випадок:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \quad (0.3)$$

Стаціонарним варіантом хвильового рівняння є рівняння Лапласа, або рівняння Пуассона в неоднорідному випадку.

Задача знаходження нормальних коливань системи, що описується хвильовим рівнянням, призводить до задачі на власні значення для рівняння Лапласа, тобто до знаходження рішень рівняння Гельмгольца, що отримується підстановкою  $u(x, t) = v(x)e^{i\omega t}$  або  $u(x, t) = v(x)\cos(\omega t)$ .

Для гіперболічного рівняння в часткових похідних існує аналітичне рішення. Частинний випадок для одновимірного простору називається формулою Д’Аламбера, названою на честь математика Жана ле Рона д’Аламбера, який вивів її в 1747 р., як рішення проблеми вібраційної струни [4].

$$u(x, t) = \frac{\varphi(x + at) + \varphi(x - at)}{2} + \frac{1}{2a} \int_{x-at}^{x+at} \psi(\alpha) d\alpha + \frac{1}{2a} \int_0^t \int_{x-a(t-\tau)}^{x+a(t-\tau)} f(s, \tau) ds d\tau \quad (0.4)$$

або для однорідного випадку:

$$u(x,t) = \frac{\varphi(x+at) + \varphi(x-at)}{2} + \frac{1}{2a} \int_{x-at}^{x+at} \psi(\alpha) d\alpha \quad (0.5)$$

### 1.1.2 Рівняння синус – Гордона

Рівняння синус – Гордона – це гіперболічне диференціальне рівняння в частинних похідних, яке є частинним випадком одновимірного хвильового рівняння [5-7]. Спочатку його було розглянуто в 1862 році Едвардом Буром в зв'язку з вивченням поверхонь постійної від'ємної кривизни. У 1970-х роках рівняння знову привернуло увагу вчених через наявність у нього солітонних розв'язків.

Загальний запис рівняння:

$$u_{tt} = u_{xx} - \sin u, \quad u = u(x,t) \quad (0.6)$$

або

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} - \sin u \quad (0.7)$$

Одна із основних властивостей рівняння синус – Гордона – існування солітонних та багатосолітонних розв'язків. Наприклад, базовий односолітонний розв'язок:

$$u(x,t) = 4 \arctan \left[ \exp \left( \pm \frac{x - vt}{\sqrt{1 - v^2}} \right) \right] \quad (0.8)$$

який описує солітон, що рухається зі швидкістю  $0 < v < 1$ , та змінює фазу від 0 до  $2\pi$  (кінк), або від  $2\pi$  до 0 (антикінк).

### 1.2 Граничні умови

У математиці, в області диференціальних рівнянь, крайовою задачею є диференціальне рівняння разом із набором додаткових обмежень, які

називаються граничними умовами [8]. Рішення крайової задачі – це рішення диференціального рівняння, яке також задовольняє граничним умовам.

Крайові проблеми надзвичайно важливі, оскільки вони моделюють величезну кількість явищ: від механіки твердого тіла до теплообміну, від механіки рідини до акустичної дифузії [9]. Вони виникають природним чином у кожній задачі на основі диференціального рівняння, що розв'язується у просторі.

Крайові проблеми широко досліджували Жак Шарль Франсуа Штурм (1803-1855) та Жозеф Ліувілл (1809-1882), які вивчали власні значення лінійного диференціального рівняння другого порядку [10]. Вони вивчали умови, які гарантують існування та унікальність розв'язку диференціальної задачі та те, як на неї впливають граничні умови [11]. Теорія Штурма-Ліувілля надзвичайно важлива для будь-якої обчислювальної задачі, оскільки вона дозволяє зрозуміти, чи проблема правильно поставлена, і як можна отримати її рішення.

### 1.3.1 Граничні умови Діріхле

Граничні умови Діріхле, або граничні умови першого роду – межові умови звичайного диференційного рівняння або диференційного рівняння в часткових похідних, названі на честь Йоганна Петера Густава Лежена Діріхле.

Ці умови визначають значення, яке невідома функція повинна прийняти вздовж межі домену. У випадку рівняння в часткових похідних межові умови можуть задаватися на якомусь контурі або поверхні, а тому можуть бути функцією, визначеною на цьому контурі чи поверхні. Якщо взяти, наприклад, рівняння Лапласа, крайову задачу з умовами Діріхле можна записати так:

$$\begin{aligned} \Delta\varphi(x) &= 0 & \forall x \in \Omega \\ \varphi(x) &= f(x) & \forall x \in \partial\Omega \end{aligned} \tag{0.9}$$

де  $\varphi$  – невідома функція,  $x$  – незалежна змінна,  $\Omega$  – область визначення функцій,  $\partial\Omega$  – межа області, та  $f$  – задана функція, визначена на  $\partial\Omega$ .

У рамках чисельного моделювання зазвичай задачі зводяться безпосередньо до алгебраїчних систем, які необхідно вирішити. Розглянемо наступну алгебраїчну систему, отриману з числового алгоритму:

$$\begin{bmatrix} k_{1,1} & k_{1,2} & \cdot & k_{1,m-1} & k_{1,m} \\ k_{2,1} & k_{2,2} & \cdot & k_{2,m-1} & k_{2,m} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ k_{m-1,1} & k_{m-1,2} & \cdot & k_{m-1,m-1} & k_{m-1,m} \\ k_{m,1} & k_{m,2} & \cdot & k_{m,m-1} & k_{m,m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ x_{m-1} \\ x_m \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ a_{m-1} \\ a_m \end{bmatrix} \quad (0.10)$$

де  $k_{i,j}$  – це елементи алгебраїчного оператора,  $x_i$  – невідомі (тобто ступені свободи),  $a_i$  – відомі значення. Найпростіший спосіб встановити граничну умову Діріхле на  $n$ -му ступені свободи – це модифікувати систему наступним чином:

$$\begin{bmatrix} k_{1,1} & \cdot & \cdot & \cdot & k_{1,m} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ k_{m,1} & \cdot & \cdot & \cdot & k_{m,m} \end{bmatrix} \begin{bmatrix} x_1 \\ \cdot \\ x_n \\ \cdot \\ x_m \end{bmatrix} = \begin{bmatrix} a_1 \\ \cdot \\ f \\ \cdot \\ a_m \end{bmatrix} \quad (0.11)$$

де  $f$  – значення, яке приймає ступінь свободи  $n$ .

### 1.3.2 Граничні умови Неймана

Граничні умови Неймана або граничні умови другого роду – граничні умови звичайного диференційного рівняння або диференційних рівнянь в часткових похідних, які визначають на границі області похідну від шуканої функції. Названі на честь Карла Неймана.

Коли гранична умова накладається на звичайне диференціальне рівняння або на рівняння з частковими похідними, вона вказує значення, які буде приймати похідна на межі області. Враховуючи, наприклад, рівняння Лапласа, крайова задача з граничними умовами Неймана запишеться як:

$$\begin{aligned}\Delta\varphi(x) &= 0 \quad \forall x \in \Omega \\ \frac{\partial\varphi(x)}{\partial n} &= f(x) \quad \forall x \in \partial\Omega\end{aligned}\tag{0.12}$$

де  $\varphi$  – невідома функція,  $x$  – незалежна змінна,  $\Omega$  – область визначення функцій,  $\partial\Omega$  – межа області,  $n$  – це нормаль до межі  $\partial\Omega$  та  $f$  – задана функція, визначена на  $\partial\Omega$ .

Нормальна похідна, яка присутня на лівій стороні, визначається як

$$\frac{\partial\varphi(x)}{\partial n} = \nabla\varphi(x) \cdot \hat{n}(x)\tag{0.13}$$

де  $\nabla = \sum_i \frac{\partial}{\partial x_i} e_i$  – оператор градієнта,  $\hat{n}$  – одиничний вектор нормалі.

Стає зрозумілим, що межа повинна бути достатньо гладкою, щоб нормальна похідна могла існувати, оскільки, наприклад, у кутових точках на межі вектор нормалі не є чітко визначеним.

### 1.3.3 Граничні умови Робена

Граничні умови Робена або граничні умови третього роду – граничні умови звичайного диференційного рівняння або диференційних рівнянь в часткових похідних, які складаються з лінійної комбінації значень поля та його похідних на межі, тобто комбінація граничних умов Діріхле та граничних умов Неймана. Названі на честь Віктора Гюстава Робена.

Наприклад, якщо взяти рівняння Лапласа, крайову задачу з граничними умовами Робена можна записати як:

$$\begin{aligned}\Delta\varphi(x) &= 0 \quad \forall x \in \Omega \\ a\varphi(x) + b\frac{\partial\varphi(x)}{\partial n} &= f(x) \quad \forall x \in \partial\Omega\end{aligned}\tag{0.14}$$

тут  $a$  та  $b$  – реальні параметри,  $\varphi$  – шукана функція,  $x$  – незалежна змінна,  $\Omega$  – область визначення функцій,  $\partial\Omega$  – межа області,  $n$  – це нормаль до межі  $\partial\Omega$  та  $f$  – задана функція, визначена на  $\partial\Omega$ .

Граничні умови Робена зазвичай використовуються при вирішенні задач Штурма – Ліувілля, які виникають у багатьох контекстах у науці та техніці.

### 1.3 Аналіз стабільності фон Неймана

Аналіз стійкості фон Неймана – це процедура, що використовується для перевірки стійкості кінцевих різницевих схем, які застосовуються для розв’язання лінійних диференціальних рівнянь з частковими похідними [12]. Аналіз заснований на розкладі за допомогою рядів Фур'є, як от:

$$e(x) = \sum_m A_m e^{ik_m x} \quad (0.15)$$

і був розроблений у Національній лабораторії Лос-Аламоса після короткого опису у статті британських дослідників Кранка та Ніколсона 1947 року [13].

Стійкість числових схем тісно пов'язана з числовою помилкою. Схема скінченних різниць стабільна, якщо помилки, допущені за один крок обчислення, не призводять до збільшення помилок під час продовження обчислень, тобто працює метод стискаючих відображень. Нейтрально стабільною схемою є схема, в якій похибки залишаються незмінними під час подальшого обчислення.

$$\left| \frac{e_i^{n+1}}{e_i^n} \right| = 1 \quad (0.16)$$

Якщо помилки згасають і з часом згасають, числова схема є стабільною.

$$\left| \frac{e_i^{n+1}}{e_i^n} \right| < 1 \quad (0.17)$$

Якщо, помилки зростають з часом, числова схема вважається нестабільною.

$$\left| \frac{e_i^{n+1}}{e_i^n} \right| > 1 \quad (0.18)$$

## 2 МЕТОДИКА ДОСЛІДЖЕНЬ

### 2.1 Методи скінченних різниць

Під час чисельного розв'язання диференціальних рівнянь в часткових похідних виникає необхідність апроксимації диференціальних операторів, які присутні в подібних рівняннях. Одним із рішень для даної проблеми є застосування методів скінченних різниць та різницевих схем.

Методи скінченних різниць являють собою клас методів чисельного аналізу для розв'язування диференціальних рівнянь шляхом апроксимації похідних скінченними різницями – математичними виразами вигляду  $f(x + b) - f(x + a)$ . При необхідності простір та/або часовий інтервал дискретизуються, тобто розбиваються на скінченну кількість кроків, і розв'язок в отриманих точках апроксимується шляхом розв'язання алгебраїчних рівнянь, що містять в собі скінченні різниці. Термін "скінченна різниця" на сьогодні часто приймається як синонім наближень похідних, особливо в контексті чисельних методів [14-15].

Методи скінченних різниць перетворюють звичайні диференціальні рівняння або диференціальні рівняння з частковими похідними у кінцеву систему алгебраїчних лінійних рівнянь, в якій диференціальні оператори апроксимуються скінчено-різницеви операторами. Дана система називається різницевою схемою.

Для опису властивостей різницевих схем введемо наступні позначення:

- $u(t)$  – точне рішення диференціального рівняння.
- $u_h(t)$  – точне рішення різницевої схеми.
- $\tilde{u}_h(t)$  – отримане чисельне рішення різницевої схеми.

Тоді, чисельне рішення отриманої задачі має наступні характеристики:

- $\|u_h(t) - u(t)\|$  – відповідає за апроксимацію задачі різницевою схемою.

- $\|\tilde{u}_h(t) - u_h(t)\|$  – відповідає за стабільність різницевої схеми при чисельному рішенні.
- $\|\tilde{u}_h(t) - u(t)\|$  – відповідає за збіжність чисельного рішення.

Приклади апроксимації похідних скінчено-різницеви операторами:

- $u_x = \frac{u_{i+1} - u_i}{h}$ ,  $u_i = u(x_i)$ ,  $x_{i+1} = x_i + h$  – апроксимація першої похідної.
- $u_{xx} = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$ ,  $u_i = u(x_i)$ ,  $x_{i+1} = x_i + h$  – апроксимація другої похідної.

На сьогодні, сучасні комп'ютери здатні ефективно проводити подібні обчислення, що, поряд із відносною легкістю їх реалізації, призвело до широкого використання методів скінченних різниць у сучасному чисельному аналізі [16].

## 2.2 Явні чисельні методи

Явні чисельні методи являються застосовуються в чисельному аналізі для отримання чисельних наближень до рішень як звичайних диференціальних рівнянь, так і диференціальних рівнянь у часткових похідних. Явні методи обчислюють стан системи в більш пізній момент часу через стан системи в поточний момент часу.

Якщо прийняти  $Y(t)$  – значення поточного стану системи,  $Y(t + \Delta t)$  – значення наступного стану системи ( $\Delta t$  – крок по часу), то для будь-якого явного методу справедливим буде наступне:

$$Y(t + \Delta t) = F(Y(t)) \quad (0.19)$$

Явні методи не потребують додаткових обчислень, тому досить прості в реалізації. Але при цьому область стабільності явних методів невелика, що призводить до труднощів під час їх застосування до жорстких рівнянь [17]. В деяких випадках стає можливо отримати наближене рішення тільки зменшивши



крок по часу  $\Delta t$ , але це значно збільшує час розв'язання, та не завжди може привести до необхідного результату.

### 2.2.1 Прямий метод Ейлера

Прямий метод Ейлера – це один із базових явних методів розв'язання та чисельного інтегрування диференціальних рівнянь, диференціальних рівнянь в часткових похідних, а також є найпростішим з класу методів Рунге – Кутти. Метод Ейлера названий на честь видатного швейцарського математика та фізика Леонарда Ейлера [18].

Метод Ейлера є методом першого порядку, а це означає, що локальна похибка пропорційна квадрату розміру кроку, а загальна похибка пропорційна. Дана похибка слугує обмеженням для застосування методу на практиці, оскільки численний розв'язок збігається до точного рішення дуже повільно при зменшенні кроку. Це не дозволяє використовувати метод Ейлера у складних розрахунках. але із-за своєї простоти він досить часто служить основою для побудови більш складних методів.

Різницева схема методу Ейлера є найпростішою, та для одновимірного хвильового рівняння (0.2) має вигляд:

$$\frac{u_i^{k+1} - 2u_i^k + u_i^{k-1}}{\Delta t^2} = c^2 \frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{\Delta x^2} + f(x, t) \quad (0.20)$$

або

$$u_i^{k+1} = \frac{c^2 \Delta t^2}{\Delta x^2} (u_{i+1}^k - 2u_i^k + u_{i-1}^k) + 2u_i^k - u_i^{k-1} + \Delta t^2 f(x, t) \quad (0.21)$$

Алгоритм знаходження чисельного розв'язку диференціального рівняння, побудований на даній різницевій схемі, буде обчислювати значення функції  $u$  на наступному часовому кроці, використовуючи при цьому значення функції з двох попередніх часових кроків [19-20].

### 2.2.2 Методи Рунге – Кутти

Методи Рунге – Кутти являють собою досить широкий клас чисельних методів розв'язання задачі Коші для звичайних диференціальних рівнянь та їх систем, а також диференціальних рівнянь в часткових похідних та їх систем. Перші методи даного класу були запропоновані у 1900 році німецькими математиками К. Рунге і М. В. Кутта.

До класу методів Рунге – Кутти входять явні та неявні методи. Формули явних методів мають наступний вигляд:

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i k_i \quad (0.22)$$

де  $y = y(x)$  – шукана функція,  $h$  – крок по часу або простору, та обчислення нового значення  $y_{n+1}$  відбувається в  $s$  етапів:

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + c_2 h, y_n + a_{2,1} k_1 h) \\ k_3 &= f(x_n + c_3 h, y_n + a_{3,1} k_1 h + a_{3,2} k_2 h) \\ &\dots \\ k_n &= f(x_n + c_n h, y_n + a_{s,1} k_1 h + a_{s,2} k_2 h + \dots + a_{s,s-1} k_{s-1} h) \end{aligned} \quad (0.23)$$

Кожен окремий метод можна подати у вигляді таблиці Бутчера, де визначені коефіцієнти  $a_{i,j}$ ,  $b_i$ ,  $c_i$ , що задають сам метод:

$$\begin{array}{c|cccc} 0 & & & & \\ c_2 & a_{2,1} & & & \\ c_3 & a_{3,1} & a_{3,2} & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s,1} & a_{s,2} & \dots & a_{s,s-1} \\ & b_1 & b_2 & \dots & b_{s-1} & b_s \end{array} \quad (0.24)$$

Для даних коефіцієнтів необхідне виконання умови:

$$\sum_{j=1}^{i-1} a_{i,j} = c_i \quad \text{для } i = 2, \dots, s \quad (0.25)$$

Для забезпечення необхідного порядку точності  $p$ , необхідно виконання додаткової умови:

$$|y(x_0 + h) - \tilde{y}(x_0 + h)| = O(h^{p+1}) \quad (0.26)$$

де  $\tilde{y}(x_0 + h)$  – наближення, обчислене за методом Рунге – Кутти.

Найбільш відомим та розповсюдженим методом Рунге – Кутти є класичний метод Рунге – Кутти 4 порядку точності.

Наближене значення наступного кроку знаходиться наступним чином:

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (0.27)$$

Безпосередньо, обчислення значень  $k_i$  для нового значення функції  $y_{n+1}$  відбувається в чотири етапи:

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{k_1 h}{2}\right) \\ k_3 &= f\left(x_n + \frac{h}{2}, y_n + \frac{k_2 h}{2}\right) \\ k_4 &= f(x_n + h, y_n + k_3 h) \end{aligned} \quad (0.28)$$

що можна подати у вигляді таблиці Бутчера:

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array} \quad (0.29)$$

Для того, щоб отримати численне рішення одновимірного хвильового рівняння (0.2) необхідно зробити наступне перетворення:

$$\begin{cases} \frac{\partial u}{\partial t} = z(x,t) \\ \frac{\partial z}{\partial t} = \frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} + f(x,t) \end{cases} \quad (0.30)$$

а також зробити заміну похідних скінчено-різницевиими операторами:

$$\begin{cases} \frac{u^{n+1} - u^n}{\Delta t} = z(x,t) \\ \frac{z^{n+1} - z^n}{\Delta t} = c^2 \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + f(x,t) \end{cases} \quad (0.31)$$

або

$$\begin{cases} u^{n+1} = u^n + \Delta t \cdot z(x,t) \\ z^{n+1} = z^n + \frac{\Delta t \cdot c^2}{\Delta x^2} (u_{i+1}^n - 2u_i^n + u_{i-1}^n) + \Delta t \cdot f(x,t) \end{cases} \quad (0.32)$$

### 2.3 Неявні чисельні методи

Вище розглянуті методи є явними методами. Але на жаль, явні методи, як правило, непридатні для вирішення жорстких диференціальних рівнянь через досить малу область їх стійкості [17]. Нестійкість явних методів створює досить серйозні проблеми і тому почали активно розроблятися неявні методи.

Якщо прийняти  $Y(t)$  – значення поточного стану системи,  $Y(t + \Delta t)$  – значення наступного стану системи ( $\Delta t$  – крок по часу), то на кожному кроці неявного методу маємо справу з рівнянням:

$$G(Y(t), Y(t + \Delta t)) = 0 \quad (0.33)$$

з якого необхідно знайти  $Y(t + \Delta t)$ .

Стає очевидно, що неявні методи знаходять рішення, вирішуючи рівняння, що включає в собі як поточний стан системи, так і більш пізній стан системи.

Неявні методи потребують додаткових обчислень на кожному кроці, тому загальний час знаходження розв'язку неявним методом буде більшим, ніж при застосуванні явного методу, при однакових значеннях  $\Delta t$ . При цьому висока стійкість дозволяє застосовувати неявні методи, оперуючи більшим значенням  $\Delta t$ , ніж в явних методах. Дана особливість може значно зменшити час обчислень.

### 2.3.1 Непрямий метод Ейлера

Неявний, або непрямий метод Ейлера – найпростіший з неявних методів. Різницева схема неявного методу Ейлера для одновимірного хвильового рівняння (0.2):

$$\frac{u_i^{k+1} - 2u_i^k + u_i^{k-1}}{\Delta t^2} = c^2 \frac{u_{i+1}^{k+1} - 2u_i^{k+1} + u_{i-1}^{k+1}}{\Delta x^2} + f(x, t) \quad (0.34)$$

або

$$-bu_{i+1}^{k+1} + (1 + 2b)u_i^{k+1} - bu_{i-1}^{k+1} = 2u_i^k - u_i^{k-1} + df(x, t) \quad (0.35)$$

де  $b = \frac{c^2 \Delta t^2}{\Delta x^2}$ , та  $d = \Delta t^2$ .

Для отримання чисельного розв'язку хвильового рівняння за допомогою даного метода, на кожному з  $(k+1)$  часовому кроці необхідно розв'язувати наступну систему лінійних алгебраїчних рівнянь:

$$\begin{bmatrix} 1+2b & -b & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ -b & 1+2b & -b & & & & & \cdot \\ 0 & -b & 1+2b & -b & & & & \cdot \\ \cdot & & \ddots & \ddots & \ddots & & & \cdot \\ \cdot & & & \ddots & \ddots & \ddots & & \cdot \\ \cdot & & & & \ddots & \ddots & \ddots & 0 \\ \cdot & & & & & -b & 1+2b & -b \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & -b & 1+2b \end{bmatrix} \begin{bmatrix} u_1^{k+1} \\ u_2^{k+1} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ u_{m-1}^{k+1} \\ u_m^{k+1} \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ C_{m-1} \\ C_m \end{bmatrix} \quad (0.36)$$

де

$$\begin{aligned} C_i &= 2u_i^k - u_i^{k-1} + df(x, t) \\ C_1 &= 2u_1^k - u_1^{k-1} + df(x, t) - (-bu_0^{k+1}) \\ C_m &= 2u_m^k - u_m^{k-1} + df(x, t) - (-bu_{m+1}^{k+1}) \end{aligned} \quad (0.37)$$

де  $u_0^{k+1}$  та  $u_{m+1}^{k+1}$  – граничні умови [9].

Розв’язок системи дає значення функції  $u$  на наступному кроці в усіх точках простору [19-20].

### 2.3.2 Метод Кранка – Ніколсона

Метод Кранка – Ніколсона – це метод кінцевих різниць з чисельного аналізу, що використовується для чисельного розв’язування диференціальних рівнянь в часткових похідних, таких як хвильове рівняння або рівняння теплопровідності. Цей метод відноситься до неявних методів, має другий порядок точності та високу стабільність. Метод винайшли в середині 20-го століття англійські вчені Джон Кранк та Филліс Ніколсон [13].

Для багатьох диференціальних рівнянь, зокрема рівнянь дифузії, можна зазначити, що метод Кранка – Ніколсона є безумовно стабільним [21]. Проте, при використанні великого кроку сітки розв’язок може мати досить велику похибку, що обумовлюється виникненням сильних коливань. У таких випадках необхідно зменшувати крок сітки, або застосовувати інші методи, які будуть стабільні та стійкі до коливань водночас.

Неявна схема, яку винайшли Джон Кранк та Филліс Ніколсон ґрунтується на наближенні похідних через усереднення значень поточного та наступного часових станів. Для одновимірного хвильового рівняння (0.2) різницева схема матиме наступний вигляд:

$$\frac{u_i^{k+1} - 2u_i^k + u_i^{k-1}}{\Delta t^2} = c^2 \frac{u_{i+1}^{k+1} - 2u_i^{k+1} + u_{i-1}^{k+1} + u_{i+1}^k - 2u_i^k + u_{i-1}^k}{2\Delta x^2} + f(x, t) \quad (0.38)$$

або

$$-bu_{i+1}^{k+1} + (1 + 2b)u_i^{k+1} - bu_{i-1}^{k+1} = 2u_i^k - u_i^{k-1} + b(u_{i+1}^k - 2u_i^k + u_{i-1}^k) + df(x, t) \quad (0.39)$$

приймавши  $b = \frac{c^2 \Delta t^2}{2\Delta x^2}$ , та  $d = \Delta t^2$ .

Для отримання чисельного розв'язку хвильового рівняння за допомогою даного метода, на кожному з  $(k + 1)$  часовому кроці необхідно розв'язувати наступну систему алгебраїчних рівнянь, як (0.36).

де

$$\begin{aligned} C_i &= 2u_i^k - u_i^{k-1} + b(u_{i+1}^k - 2u_i^k + u_{i-1}^k) + df(x, t) \\ C_1 &= 2u_1^k - u_1^{k-1} + b(u_2^k - 2u_1^k + u_0^k) + df(x, t) - (-bu_0^{k+1}) \\ C_m &= 2u_m^k - u_m^{k-1} + b(u_{m+1}^k - 2u_m^k + u_{m-1}^k) + df(x, t) - (-bu_{m+1}^{k+1}) \end{aligned} \quad (0.40)$$

де  $u_0^{k+1}$  та  $u_{m+1}^{k+1}$  – граничні умови [9].

Розв'язок системи дає значення функції  $u$  на наступному кроці в усіх точках простору [19-20].

#### 2.4 Розрахунок енергії для біжучої хвилі

При розповсюдженні хвилі в просторі від якого-небудь джерела відбувається також поширення енергії вздовж хвилі. Для опису даного явища розглянемо випадок коливання струни (0.2).

Під час поширення енергії вздовж струни, кожен елемент маси струни рухається вгору і вниз з тією ж частотою, що і хвиля. Оскільки струна має постійну лінійну щільність  $\mu = \frac{\Delta m}{\Delta x}$ , то кожен подібний елемент маси струни має

масу  $\Delta m = \mu \Delta x$ . Повна механічна енергія хвилі складається має такі складові як кінетичну енергію та потенційну енергію. В загальному випадку кінетична

енергія дорівнює  $E_k = \frac{mv^2}{2}$ .

Для кожного елемента струни з масою  $\Delta m$  справедливим буде наступне:

$$\Delta E_k = \frac{1}{2}(\Delta m)v^2 = \frac{1}{2}(\mu\Delta x)\frac{\partial^2 u}{\partial t^2} \quad (2.23)$$

В загальному випадку кожен елемент струни є нахиленим та його довжина дорівнює  $\sqrt{\partial x^2 + \partial u^2}$ . Виконавши певні математичні перетворення, отримаємо:

$$\sqrt{\partial x^2 + \partial u^2} = \partial x \sqrt{1 + \left(\frac{\partial u}{\partial x}\right)^2} \approx \partial x + \frac{\partial x}{2} \left(\frac{\partial u}{\partial x}\right)^2 \quad (2.24)$$

де було використано розкладання у серію Тейлора  $\sqrt{1+e} \approx 1 + \frac{e}{2}$ .

З рівняння можна бачити, що кожен елемент струни має розтяг  $\frac{\partial x}{2} \left(\frac{\partial u}{\partial x}\right)^2$ .

Оскільки потенційна енергія залежить від розтягу струни, а розтягнення зумовлене наявністю дії зовнішніх сил натягу на двох кінцях, які виконують деяку кількість роботи  $T \frac{\partial x}{2} \left(\frac{\partial u}{\partial x}\right)^2$  (де  $T$  – жорсткість, або натяг), то саме ця робота і позиціонує як потенціальна енергія. Тобто, потенціальна енергія має вигляд:

$$E_p = \frac{1}{2}(T\Delta x)\frac{\partial^2 u}{\partial x^2} \quad (2.25)$$

Тоді повну енергію можна записати:

$$E = E_k + E_p = \frac{1}{2}(\mu\Delta x)\frac{\partial^2 u}{\partial t^2} + \frac{1}{2}(T\Delta x)\frac{\partial^2 u}{\partial x^2} \quad (2.26)$$

Даний вираз є справедливим для будь-якої хвилі.

## 2.5 Штучна в'язкість

При моделюванні хвильових процесів іноді виникає задача створення спеціальної області, в якій вхідні хвилі починають затухати або розсіюватися.



Дана задача може з'являтися коли існує необхідність зменшити або зупинити просування хвилі, відсутністю частини граничних умов, дослідженням хвилі на певному проміжку, тощо.

За допомогою власних досліджень вдалося перевірити деякі припущення та виділити два способи задання штучної в'язкості для одновимірного хвильового рівняння (0.2), які дають змогу отримати позитивні результати:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} + f(x,t) + v \frac{\partial u}{\partial x} \quad (0.41)$$

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} + f(x,t) - v \frac{\partial^2 u}{\partial x^2} \quad (0.42)$$

де  $+v \frac{\partial u}{\partial x}$  та  $-v \frac{\partial^2 u}{\partial x^2}$  – вплив зовнішньої сили, що задає в'язкість,  $v$  – коефіцієнт в'язкості.

Метод (2.27) використовує дисперсійний доданок, а метод (2.28) використовує дифузійний доданок.

## 2.6 Машинна оптимізація

На етапі роботи з штучною в'язкістю одним із найбільш важливих моментів став пошук значень коефіцієнта в'язкості. Оскільки результати при різних значеннях можуть суттєво відрізнятись один від одного, виникла необхідність пошуку таких значень коефіцієнта в'язкості, які при застосуванні давали оптимальний результат. Безпосередній підбір коефіцієнтів вручну, або навіть закону розподілення, за яким можна задавати ці коефіцієнти, є неефективними способами вирішення насущної проблеми. Тобто, виникає задача знайти аналітично, або за допомогою експериментів такий спосіб задання коефіцієнтів в'язкості, щоб отримати на виході оптимальний чисельний розв'язок хвильового рівняння.

В кваліфікаційній роботі було обрано такий підхід для розв'язання поставленої задачі, як машинна оптимізація. Для проведення серії експериментів було обрано другий спосіб задання в'язкості (2.28), оскільки в ході експериментів було визначено межі значень коефіцієнтів –  $[0, c^2)$ , а з наведених вище чисельних методів використовувався явний метод Ейлера (2.3), оскільки він має найпростішу реалізацію та є найшвидшим з усіх.

Алгоритм, розроблений для реалізації машинної оптимізації, має наступний вигляд:

1. Визначення типу та параметрів хвилі.
2. Визначення параметрів, необхідних для отримання чисельного розв'язку, таких як: кількість точок по простору, крок по простору, кількість інтервалів по часу, крок по часу.
3. Запуск програми з обраними параметрами.
4. Програма використовує відомі їй закони розподілення, щоб задати коефіцієнти в'язкості, та, змінюючи коефіцієнти самих законів розподілення, запускає пошук чисельного розв'язку за допомогою явного методу Ейлера.
5. Після отримання кожного з чисельних розв'язків по певному закону розподілення, програма обирає оптимальний по принципу мінімальної вихідної амплітуди хвилі, та запам'ятовує значення використаних коефіцієнтів в'язкості.
6. Після тестування усіх відомих програмі законів розподілення, програма порівнює отримані оптимальні результати по кожному, та обирає один з мінімальною вихідною амплітудою.
7. На вихід програма видає мінімальну амплітуду та масив коефіцієнтів в'язкості, за використання яких біла отримана мінімальна амплітуда.

### 3 РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОГО ЕКСПЕРИМЕНТУ

#### 3.1 Дослідження за допомогою прямого методу Ейлера

На рис. 3.1-3.8 зображені різні моменти часу для чисельного розв'язку рівняння (0.3), отриманого за допомогою прямого методу Ейлера (0.21), з параметрами  $c = 1$ ,  $\Delta x = 0,001$ ,  $\Delta t = 0,001$ , 3000 точок по  $x$ , початковою умовою  $u(x,0) = 0$  та граничною умовою  $u(0,t) = \sin 10\pi(x-t)$ .

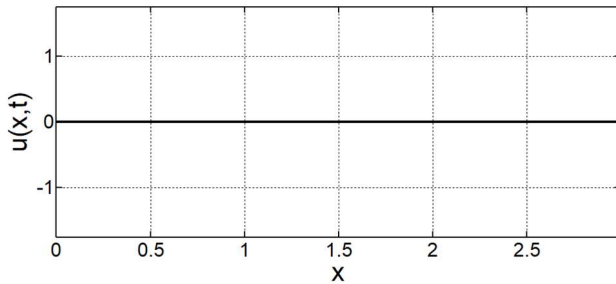


Рис. 3.1 – хвиля в момент часу  $t = 0$ .

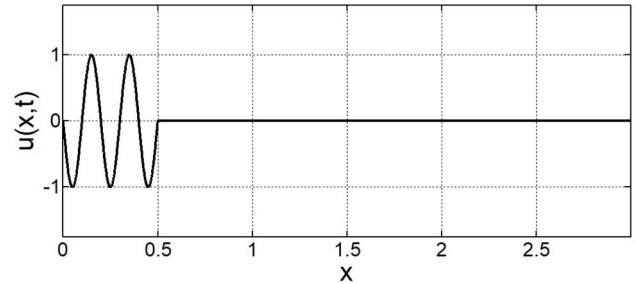


Рис. 3.2 – хвиля в момент часу  $t = 500$ .

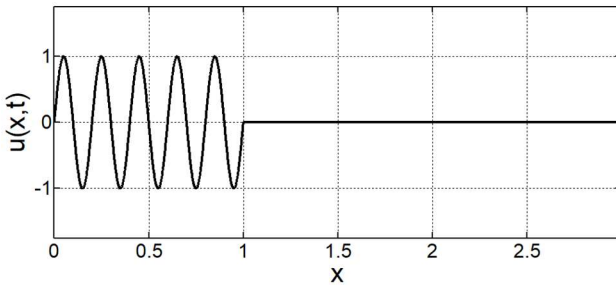


Рис. 3.3 – хвиля в момент часу  $t = 1000$ .

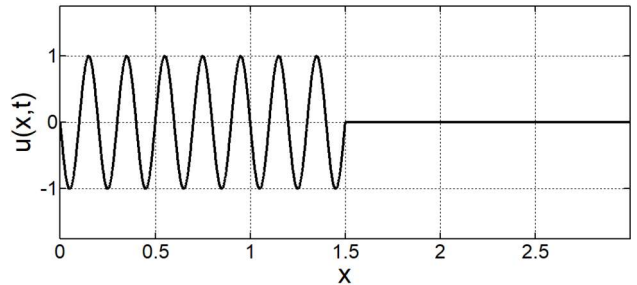


Рис. 3.4 – хвиля в момент часу  $t = 1500$ .

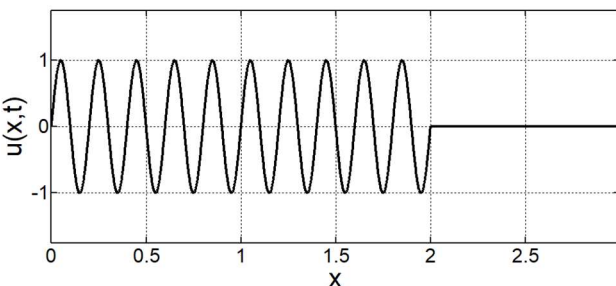


Рис. 3.5 – хвиля в момент часу  $t = 2000$ .

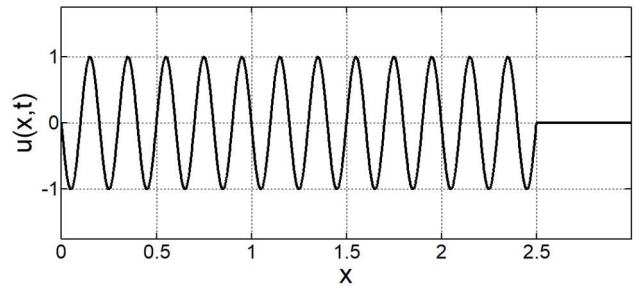


Рис. 3.6 – хвиля в момент часу  $t = 2500$ .

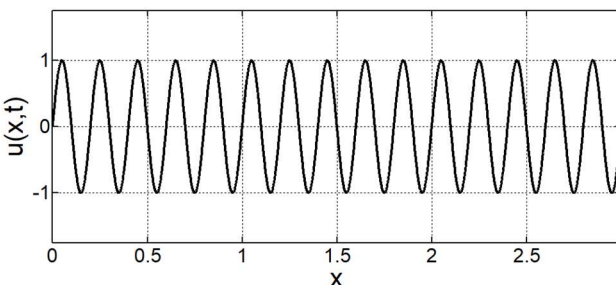


Рис. 3.7 – хвиля в момент часу  $t = 3000$ .

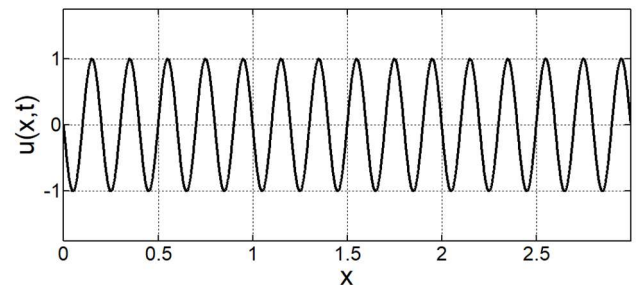
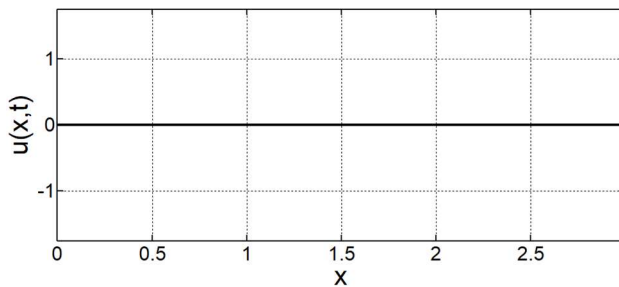
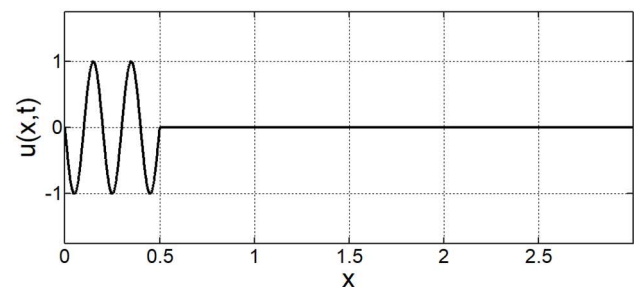
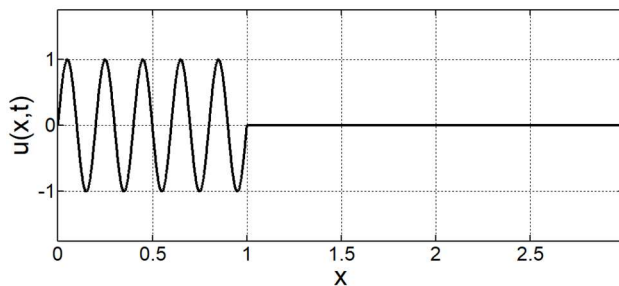
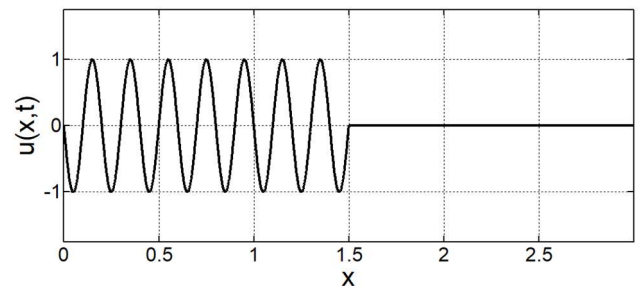
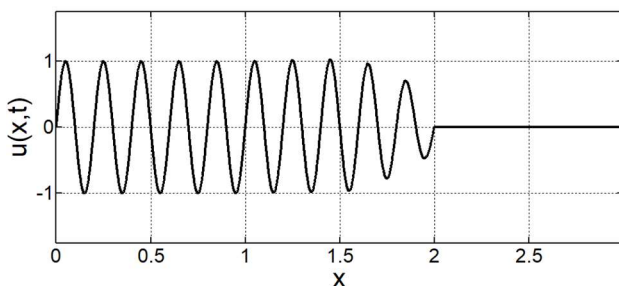
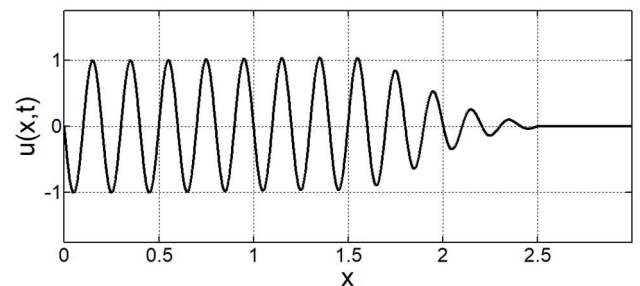
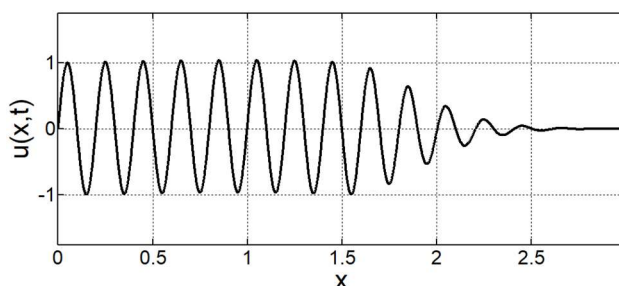
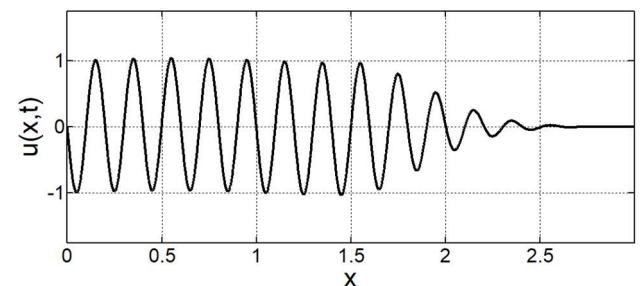
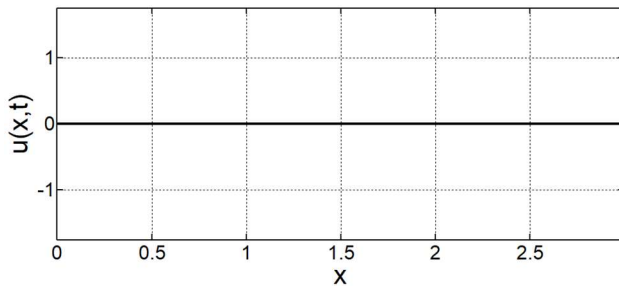
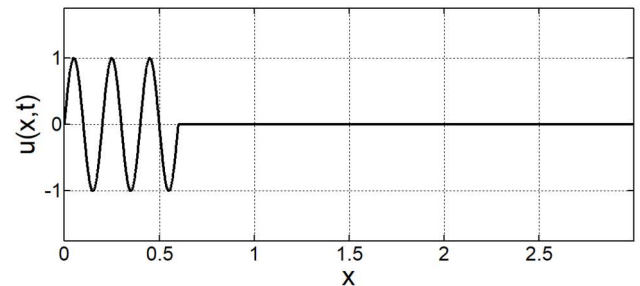
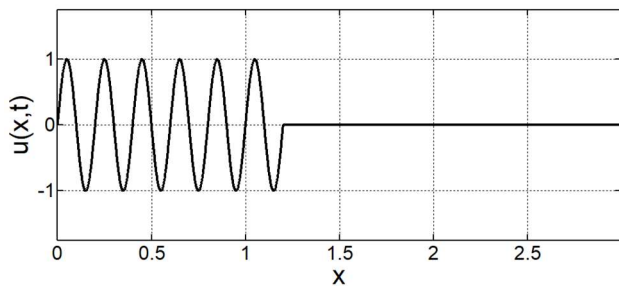
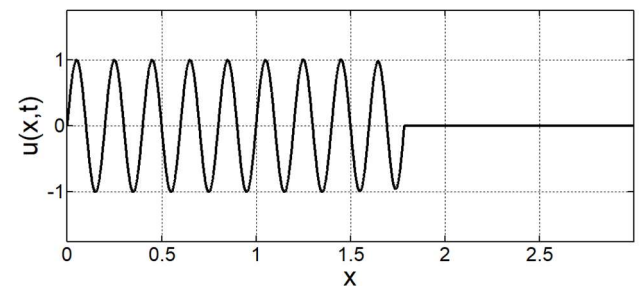
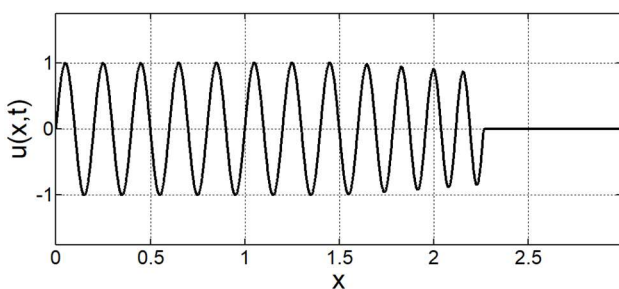
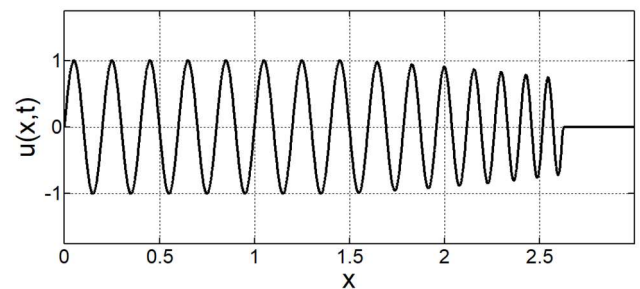
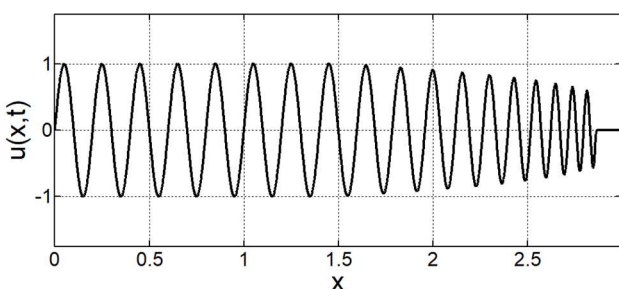
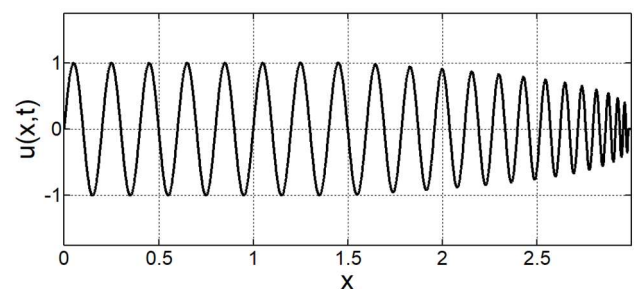


Рис. 3.8 – хвиля в момент часу  $t = 3500$ .

На рис. 3.9-3.16 зображені різні моменти часу для чисельного розв'язку рівняння (0.3), отриманого за допомогою прямого методу Ейлера (0.21), з використанням першого способу додання штучної в'язкості (0.41), та з параметрами  $c = 1$ ,  $\Delta x = 0,001$ ,  $\Delta t = 0,001$ , 3000 точок по  $x$ , початковою умовою  $u(x,0) = 0$  та граничною умовою  $u(0,t) = \sin 10\pi(x-t)$ . Коефіцієнт в'язкості  $\nu$  дорівнює 0 для перших 50% точок, та лінійно збільшується на інтервалі  $[0; 20]$  для останніх 50% точок.

Рис. 3.9 – хвиля в момент часу  $t = 0$ .Рис. 3.10 – хвиля в момент часу  $t = 500$ .Рис. 3.11 – хвиля в момент часу  $t = 1000$ .Рис. 3.12 – хвиля в момент часу  $t = 1500$ .Рис. 3.13 – хвиля в момент часу  $t = 2000$ .Рис. 3.14 – хвиля в момент часу  $t = 2500$ .Рис. 3.15 – хвиля в момент часу  $t = 3000$ .Рис. 3.16 – хвиля в момент часу  $t = 3500$ .

На рис. 3.17-3.24 зображені різні моменти часу для чисельного розв'язку рівняння (0.3), отриманого за допомогою прямого методу Ейлера (0.21), з використанням другого способу додання штучної в'язкості (0.42), та з параметрами  $c = 1$ ,  $\Delta x = 0,001$ ,  $\Delta t = 0,001$ , 3000 точок по  $x$ , початковою умовою  $u(x,0) = 0$  та граничною умовою  $u(0,t) = \sin 10\pi(x-t)$ . Коефіцієнт в'язкості  $\nu$  дорівнює 0 для перших 50% точок, та лінійно збільшується на інтервалі  $[0; 0,999]$  для останніх 50% точок.

Рис. 3.17 – хвиля в момент часу  $t = 0$ .Рис. 3.18 – хвиля в момент часу  $t = 600$ .Рис. 3.19 – хвиля в момент часу  $t = 1200$ .Рис. 3.20 – хвиля в момент часу  $t = 1800$ .Рис. 3.21 – хвиля в момент часу  $t = 2400$ .Рис. 3.22 – хвиля в момент часу  $t = 3000$ .Рис. 3.23 – хвиля в момент часу  $t = 3600$ .Рис. 3.24 – хвиля в момент часу  $t = 4200$ .

### 3.2 Дослідження за допомогою методу Рунге – Кутта 4 порядку

На рис. 3.25-3.32 зображені різні моменти часу для чисельного розв'язку рівняння (0.3), отриманого з допомогою методу Рунге – Кутти 4 порядку (0.32), з параметрами  $c=1$ ,  $\Delta x=0,001$ ,  $\Delta t=0,001$ , 3000 точок по  $x$ , початковою умовою  $u(x,0)=0$  та граничною умовою  $u(0,t)=\sin 10\pi(x-t)$ .

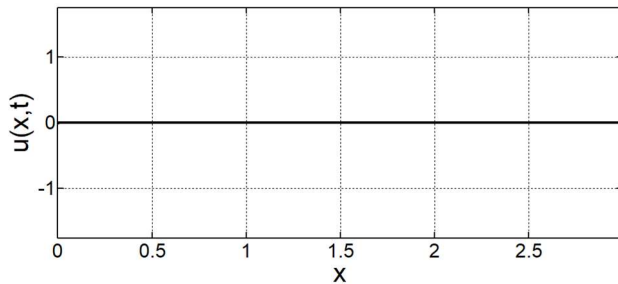


Рис. 3.25 – хвиля в момент часу  $t=0$ .

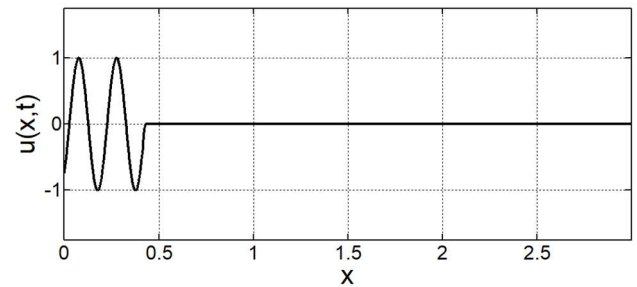


Рис. 3.26 – хвиля в момент часу  $t=425$ .

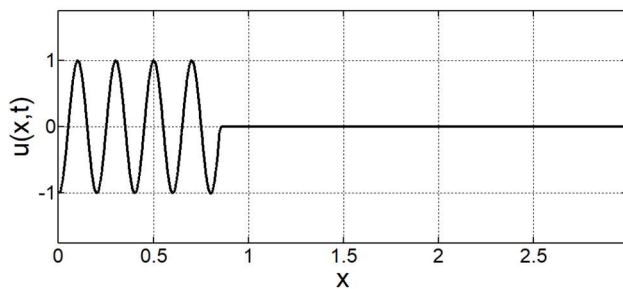


Рис. 3.27 – хвиля в момент часу  $t=850$ .

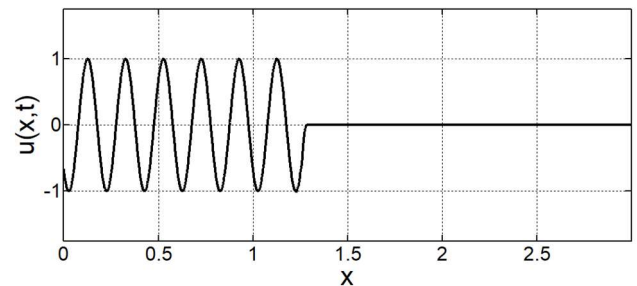


Рис. 3.28 – хвиля в момент часу  $t=1275$ .

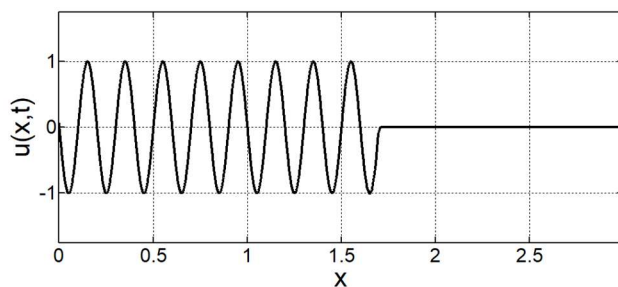


Рис. 3.29 – хвиля в момент часу  $t=1700$ .

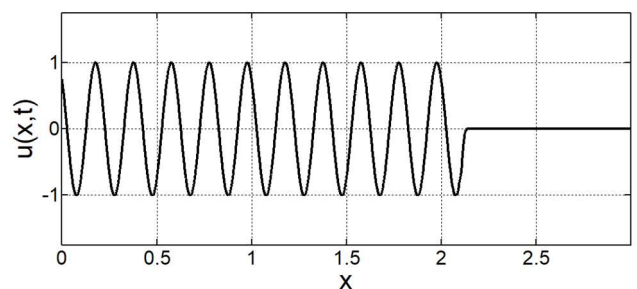


Рис. 3.30 – хвиля в момент часу  $t=2125$ .

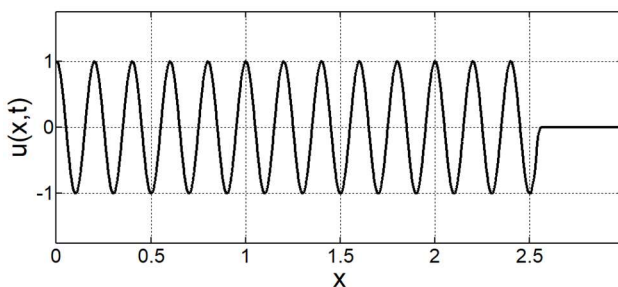


Рис. 3.31 – хвиля в момент часу  $t=2550$ .

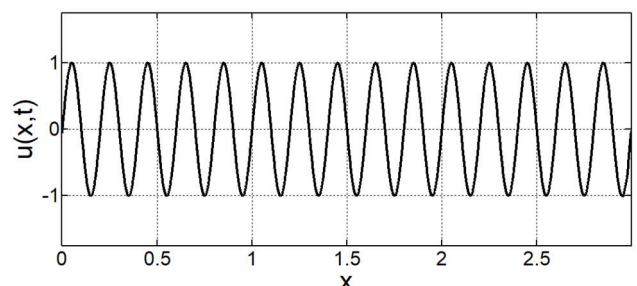
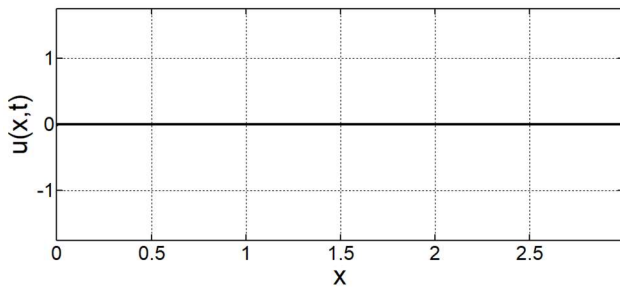
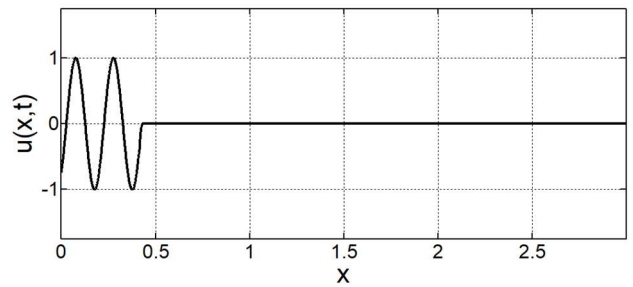
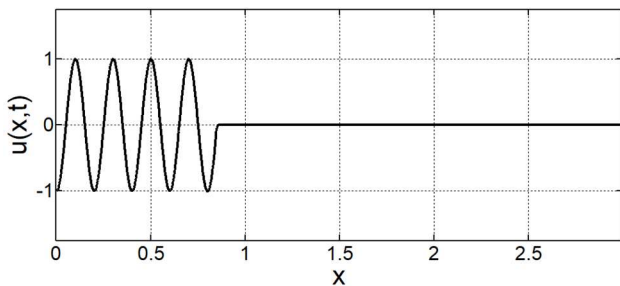
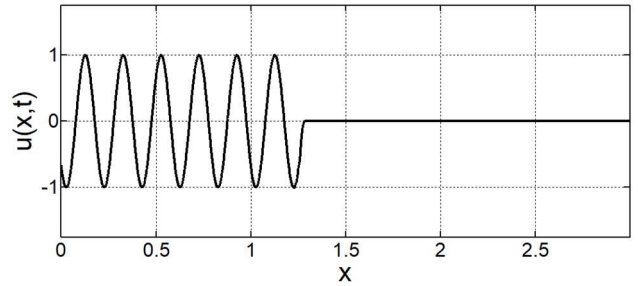
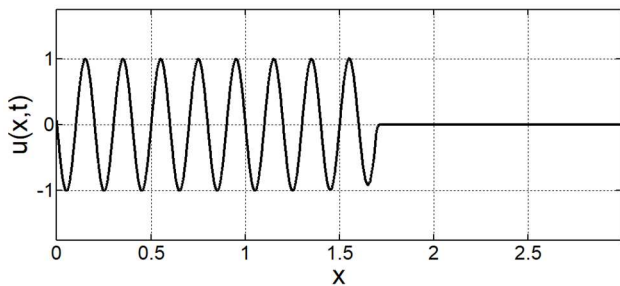
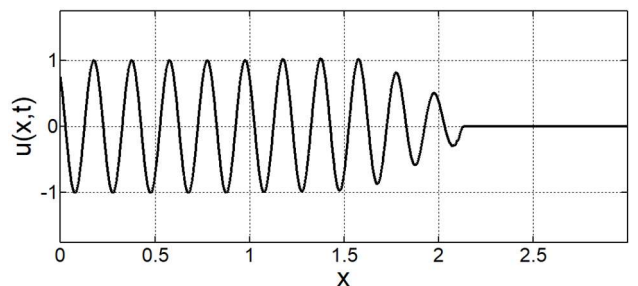
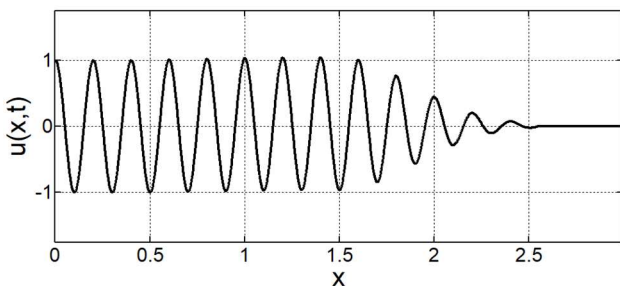
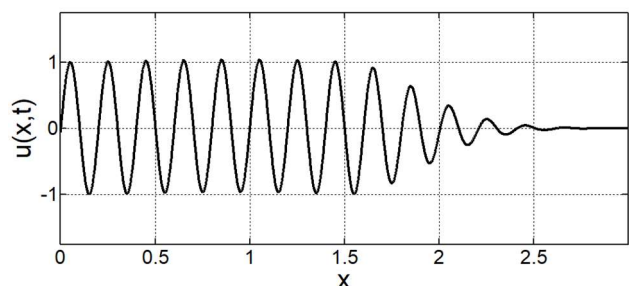


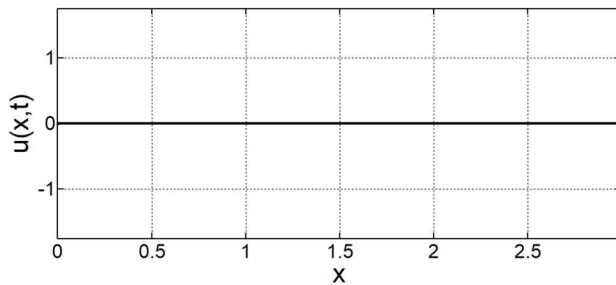
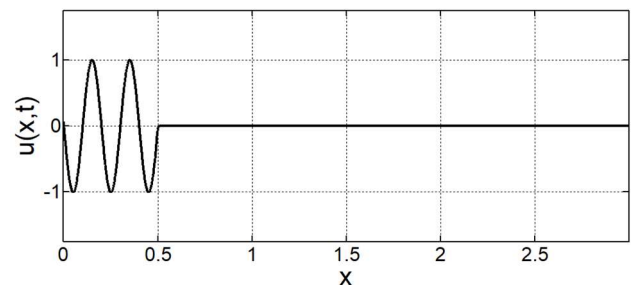
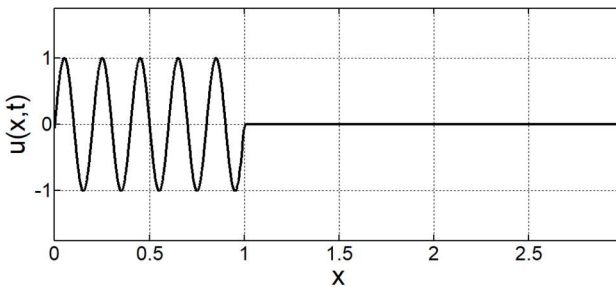
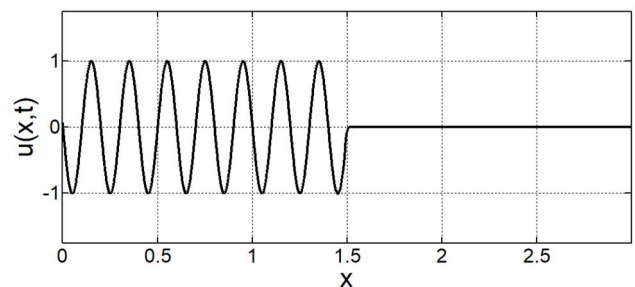
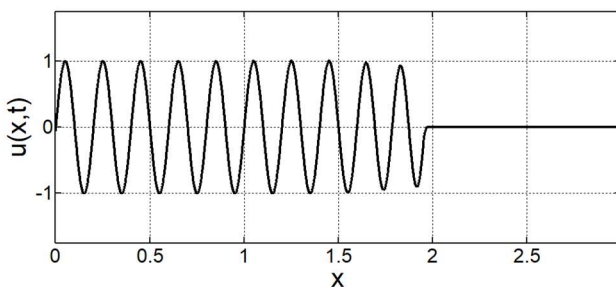
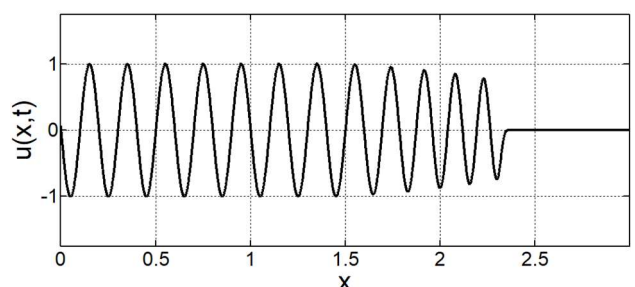
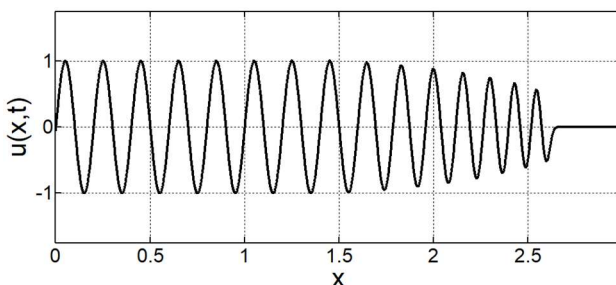
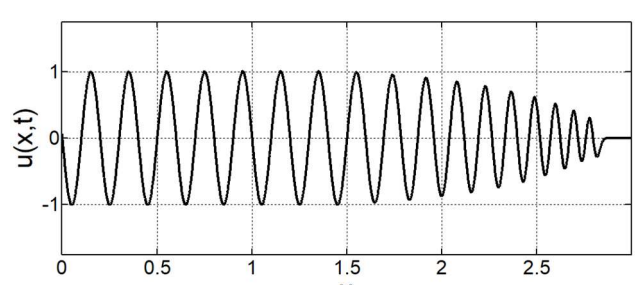
Рис. 3.32 – хвиля в момент часу  $t=3000$ .



На рис. 3.33-3.40 зображені різні моменти часу для чисельного розв'язку рівняння (0.3), отриманого з допомогою методу Рунге – Кутти 4 порядку (0.32), з використанням першого способу додання штучної в'язкості (0.41), та з параметрами  $c = 1$ ,  $\Delta x = 0,001$ ,  $\Delta t = 0,001$ , 3000 точок по  $x$ , початковою умовою  $u(x,0) = 0$  та граничною умовою  $u(0,t) = \sin 10\pi(x-t)$ . Коефіцієнт в'язкості  $\nu$  дорівнює 0 для перших 50% точок, та лінійно збільшується на інтервалі  $[0; 20]$  для останніх 50% точок.

Рис. 3.33 – хвиля в момент часу  $t = 0$ .Рис. 3.34 – хвиля в момент часу  $t = 425$ .Рис. 3.35 – хвиля в момент часу  $t = 850$ .Рис. 3.36 – хвиля в момент часу  $t = 1275$ .Рис. 3.37 – хвиля в момент часу  $t = 1700$ .Рис. 3.38 – хвиля в момент часу  $t = 2125$ .Рис. 3.39 – хвиля в момент часу  $t = 2550$ .Рис. 3.40 – хвиля в момент часу  $t = 3000$ .

На рис. 3.41-3.48 зображені різні моменти часу для чисельного розв'язку рівняння (0.3), отриманого з допомогою методу Рунге – Кутти 4 порядку (0.32), з використанням другого способу додання штучної в'язкості (0.42), та з параметрами  $c = 1$ ,  $\Delta x = 0,001$ ,  $\Delta t = 0,001$ , 3000 точок по  $x$ , початковою умовою  $u(x,0) = 0$  та граничною умовою  $u(0,t) = \sin 10\pi(x-t)$ . Коефіцієнт в'язкості  $\nu$  дорівнює 0 для перших 50% точок, та лінійно збільшується на інтервалі  $[0; 0,999]$  для останніх 50% точок.

Рис. 3.41 – хвиля в момент часу  $t = 0$ .Рис. 3.42 – хвиля в момент часу  $t = 425$ .Рис. 3.43 – хвиля в момент часу  $t = 850$ .Рис. 3.44 – хвиля в момент часу  $t = 1275$ .Рис. 3.45 – хвиля в момент часу  $t = 1700$ .Рис. 3.46 – хвиля в момент часу  $t = 2125$ .Рис. 3.47 – хвиля в момент часу  $t = 2550$ .Рис. 3.48 – хвиля в момент часу  $t = 3000$ .



### 3.3 Дослідження за допомогою непрямого методу Ейлера

На рис. 3.49-3.56 зображені різні моменти часу для чисельного розв'язку рівняння (0.3), отриманого за допомогою непрямого методу Ейлера (0.35), з параметрами  $c = 1$ ,  $\Delta x = 0,001$ ,  $\Delta t = 0,001$ , 1500 точок по  $x$ , початковою умовою  $u(x,0) = 0$  та граничною умовою  $u(0,t) = \sin 10\pi(x-t)$ .

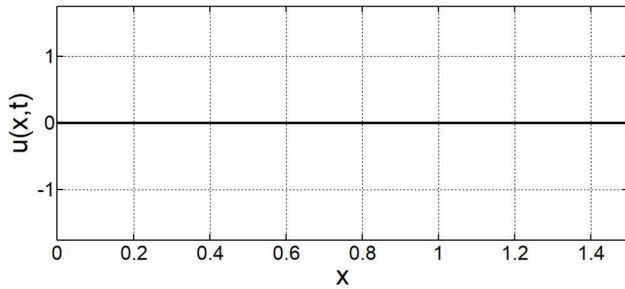


Рис. 3.49 – хвиля в момент часу  $t = 0$ .

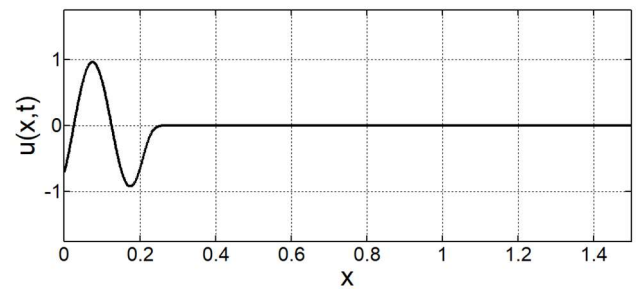


Рис. 3.50 – хвиля в момент часу  $t = 225$ .

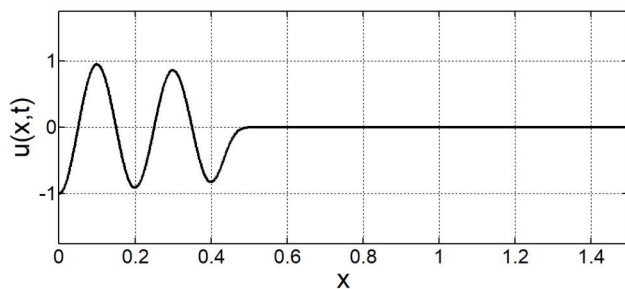


Рис. 3.51 – хвиля в момент часу  $t = 450$ .

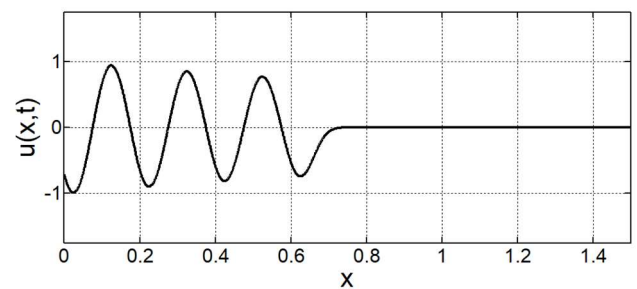


Рис. 3.52 – хвиля в момент часу  $t = 675$ .

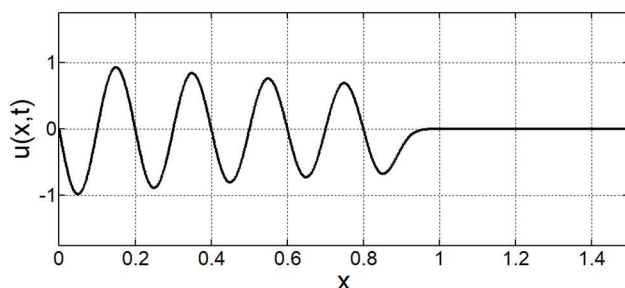


Рис. 3.53 – хвиля в момент часу  $t = 900$ .

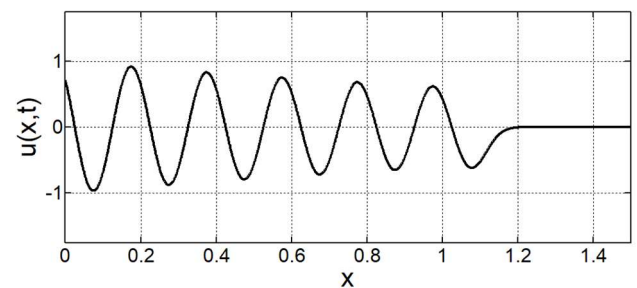


Рис. 3.54 – хвиля в момент часу  $t = 1125$ .

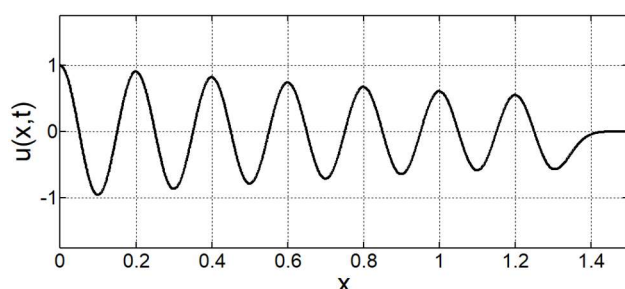


Рис. 3.55 – хвиля в момент часу  $t = 1350$ .

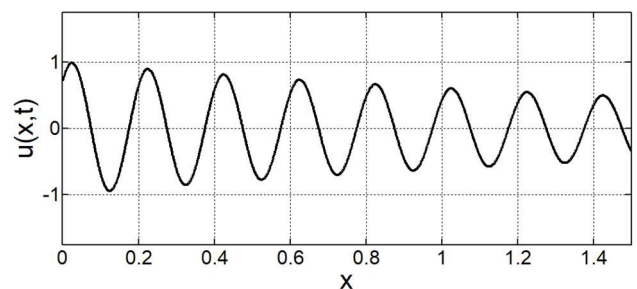
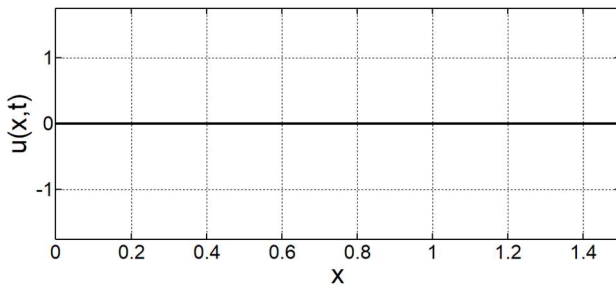
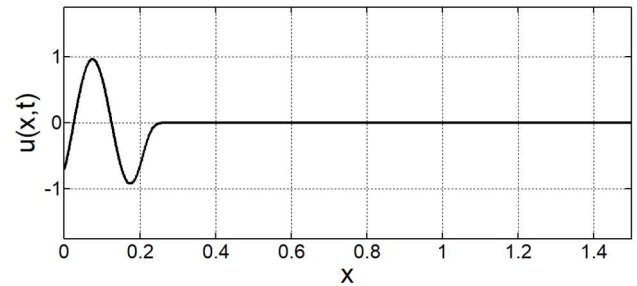
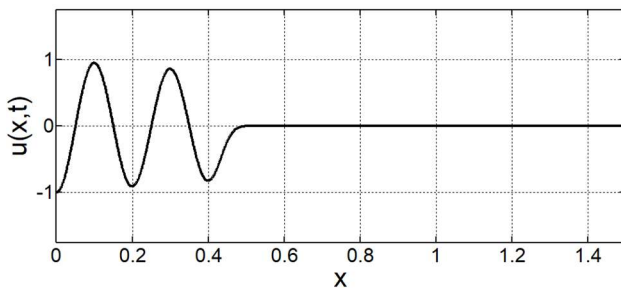
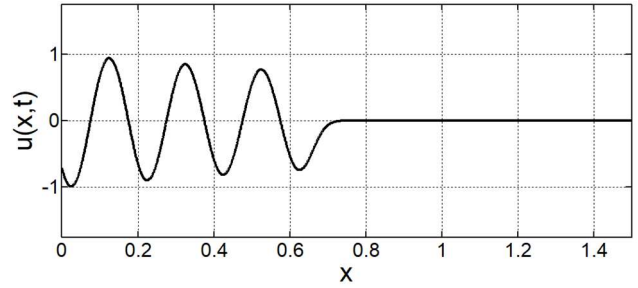
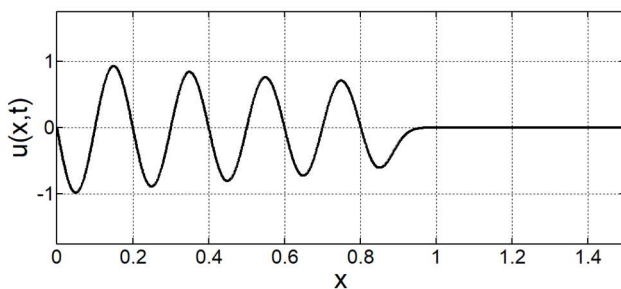
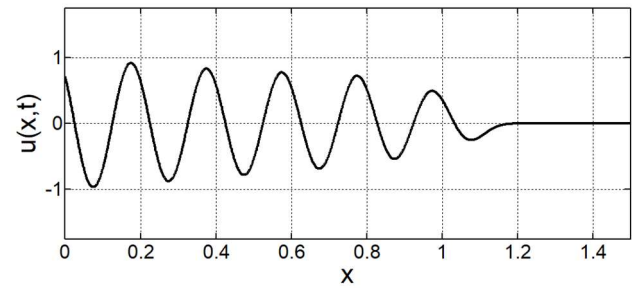
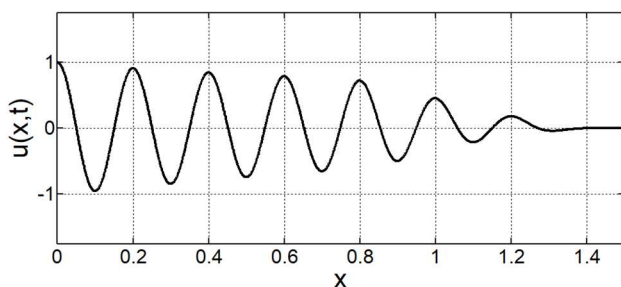
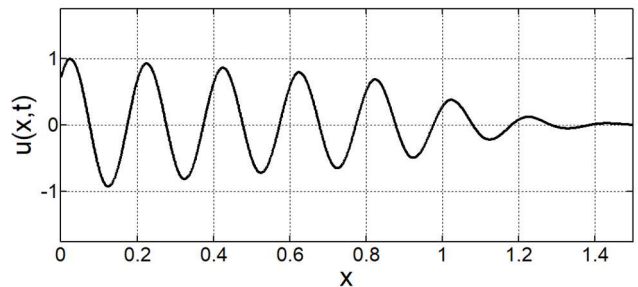
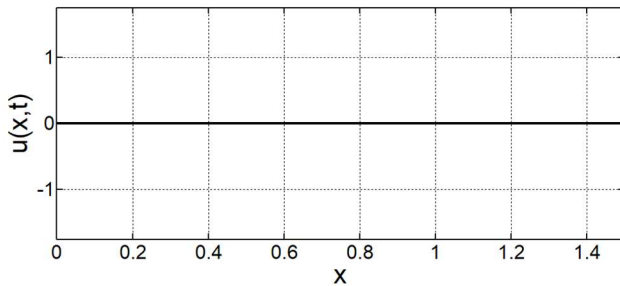
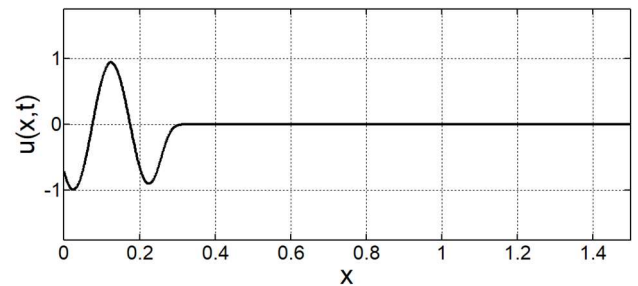
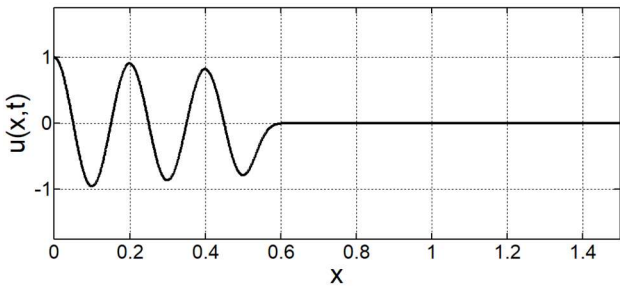
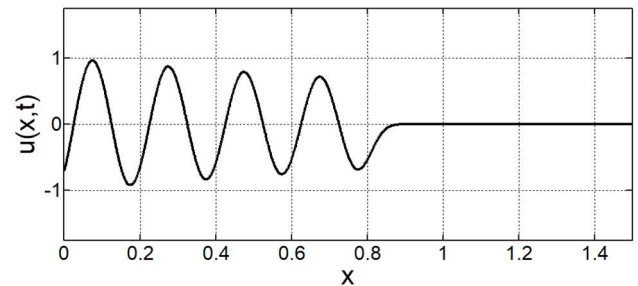
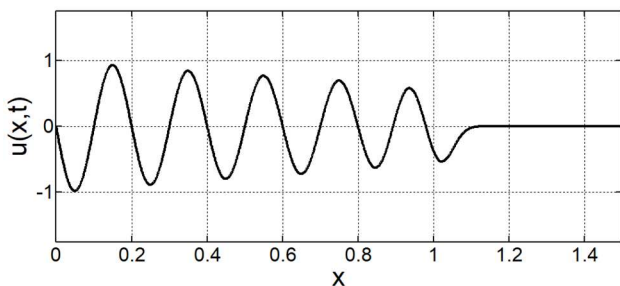
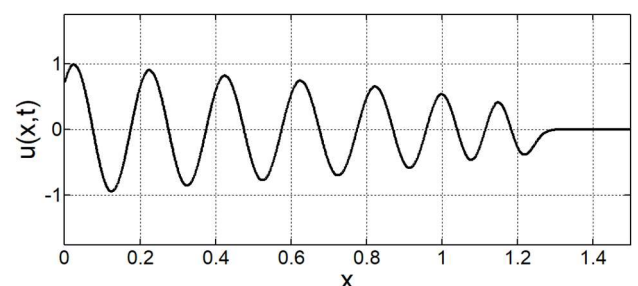
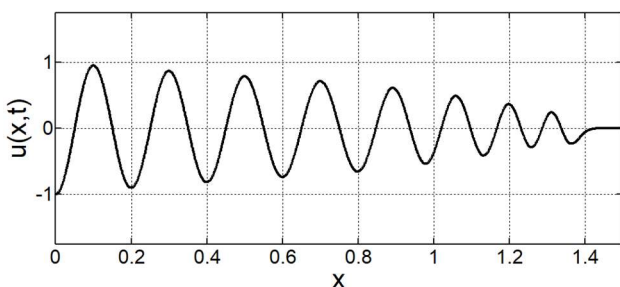
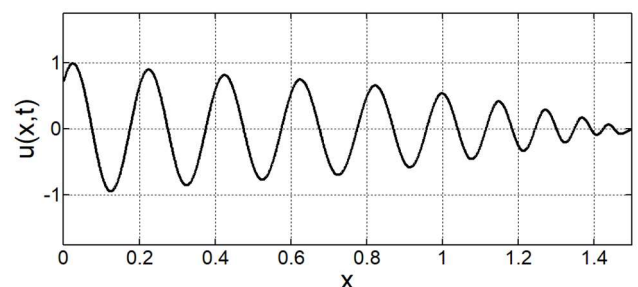


Рис. 3.56 – хвиля в момент часу  $t = 1575$ .

На рис. 3.57-3.64 зображені різні моменти часу для чисельного розв'язку рівняння (0.3), отриманого за допомогою непрямого методу Ейлера (0.35), з використанням першого способу додання штучної в'язкості (0.41), та з параметрами  $c = 1$ ,  $\Delta x = 0,001$ ,  $\Delta t = 0,001$ , 1500 точок по  $x$ , початковою умовою  $u(x,0) = 0$  та граничною умовою  $u(0,t) = \sin 10\pi(x-t)$ . Коефіцієнт в'язкості  $\nu$  дорівнює 0 для перших 50% точок, та лінійно збільшується на інтервалі  $[0; 20]$  для останніх 50% точок.

Рис. 3.57 – хвиля в момент часу  $t = 0$ .Рис. 3.58 – хвиля в момент часу  $t = 225$ .Рис. 3.59 – хвиля в момент часу  $t = 450$ .Рис. 3.60 – хвиля в момент часу  $t = 675$ .Рис. 3.61 – хвиля в момент часу  $t = 900$ .Рис. 3.62 – хвиля в момент часу  $t = 1125$ .Рис. 3.63 – хвиля в момент часу  $t = 1350$ .Рис. 3.64 – хвиля в момент часу  $t = 1575$ .

На рис. 3.65-3.72 зображені різні моменти часу для чисельного розв'язку рівняння (0.3), отриманого за допомогою непрямого методу Ейлера (0.35), з використанням другого способу додання штучної в'язкості (0.42), та з параметрами  $c = 1$ ,  $\Delta x = 0,001$ ,  $\Delta t = 0,001$ , 1500 точок по  $x$ , початковою умовою  $u(x,0) = 0$  та граничною умовою  $u(0,t) = \sin 10\pi(x-t)$ . Коефіцієнт в'язкості  $\nu$  дорівнює 0 для перших 50% точок, та лінійно збільшується на інтервалі  $[0; 0,999]$  для останніх 50% точок.

Рис. 3.65 – хвиля в момент часу  $t=0$ .Рис. 3.66 – хвиля в момент часу  $t=275$ .Рис. 3.67 – хвиля в момент часу  $t=550$ .Рис. 3.68 – хвиля в момент часу  $t=825$ .Рис. 3.69 – хвиля в момент часу  $t=1100$ .Рис. 3.70 – хвиля в момент часу  $t=1375$ .Рис. 3.71 – хвиля в момент часу  $t=1650$ .Рис. 3.72 – хвиля в момент часу  $t=1925$ .

### 3.4 Дослідження за допомогою методу Кранка – Ніколсона

На рис. 3.73-3.80 зображені різні моменти часу для чисельного розв'язку рівняння (0.3), отриманого за допомогою методу Кранка – Ніколсона (0.39), з параметрами  $c = 1$ ,  $\Delta x = 0,001$ ,  $\Delta t = 0,001$ , 1500 точок по  $x$ , початковою умовою  $u(x,0) = 0$  та граничною умовою  $u(0,t) = \sin 10\pi(x-t)$ .

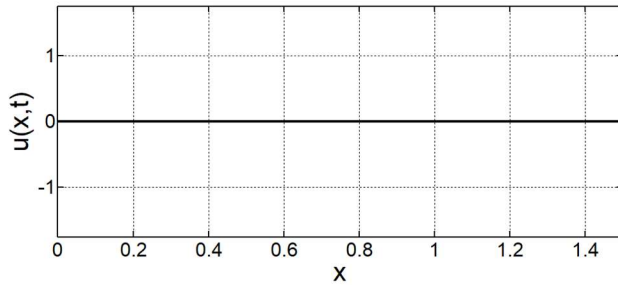


Рис. 3.73 – хвиля в момент часу  $t=0$ .

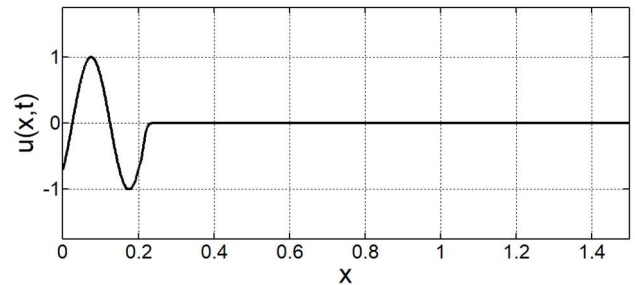


Рис. 3.74 – хвиля в момент часу  $t = 225$ .

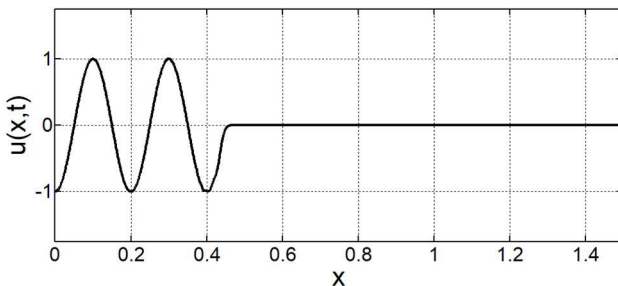


Рис. 3.75 – хвиля в момент часу  $t = 450$ .

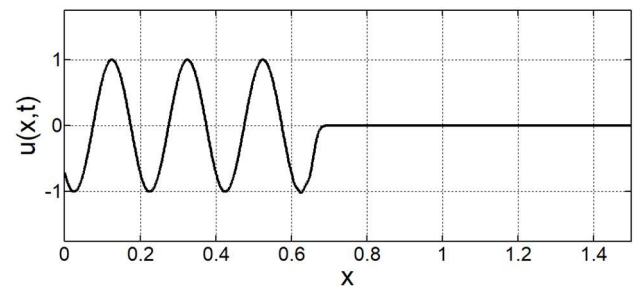


Рис. 3.76 – хвиля в момент часу  $t = 675$ .

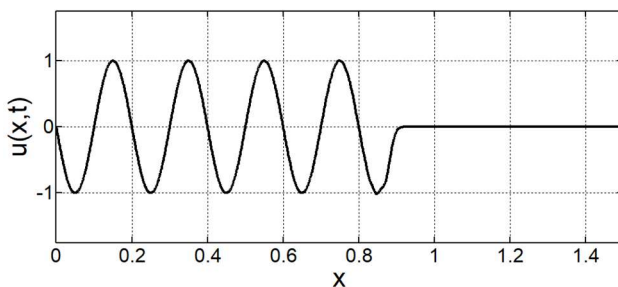


Рис. 3.77 – хвиля в момент часу  $t = 900$ .

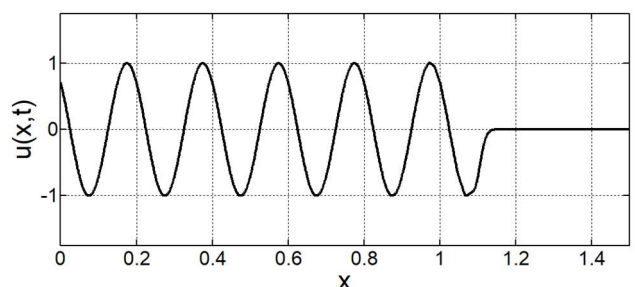


Рис. 3.78 – хвиля в момент часу  $t = 1125$ .

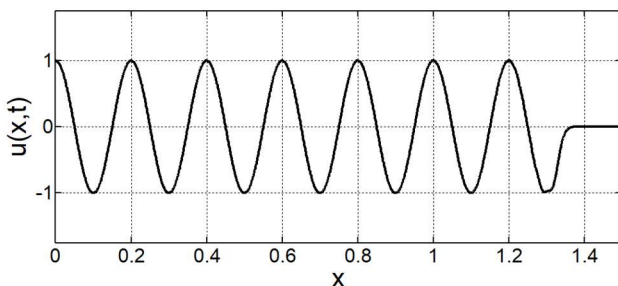


Рис. 3.79 – хвиля в момент часу  $t = 1350$ .

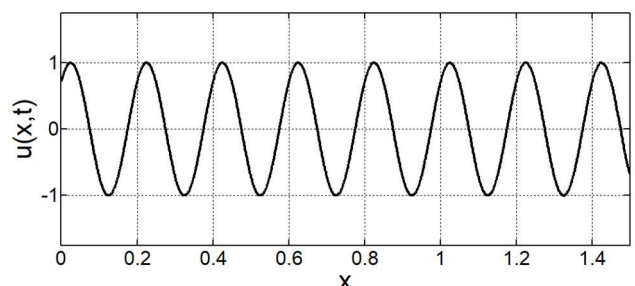
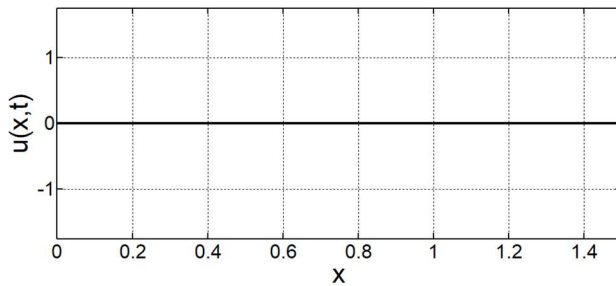
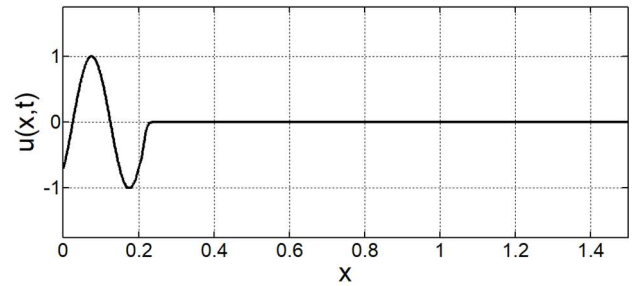
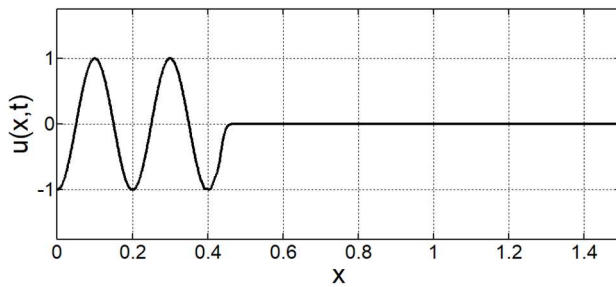
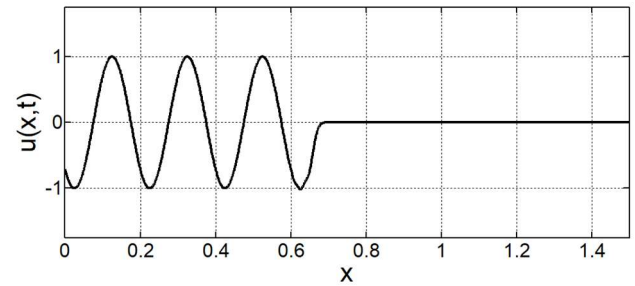
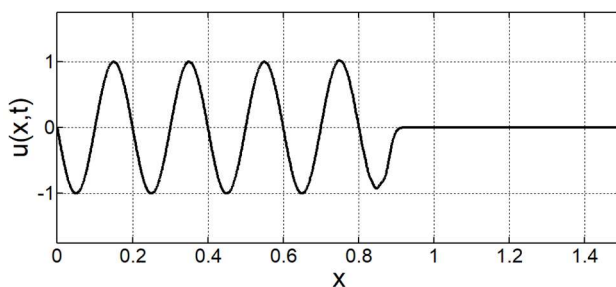
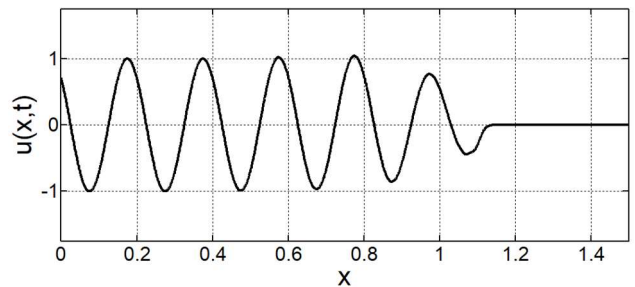
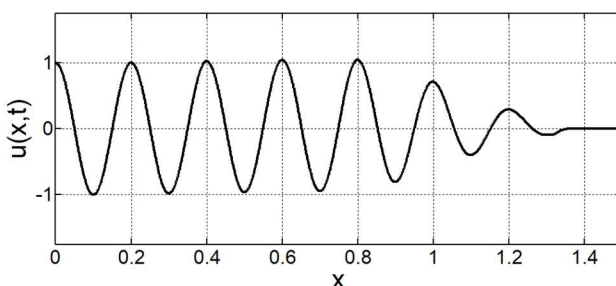
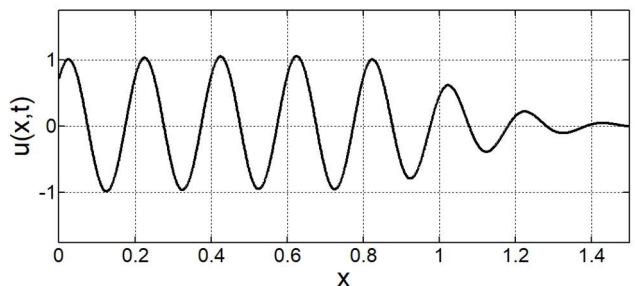


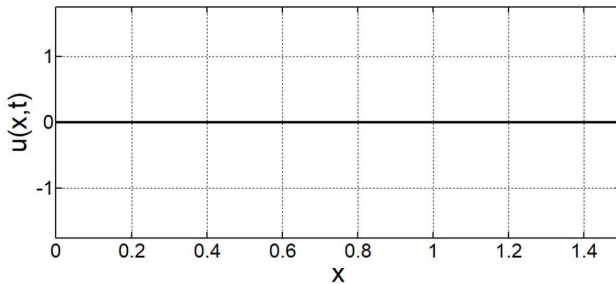
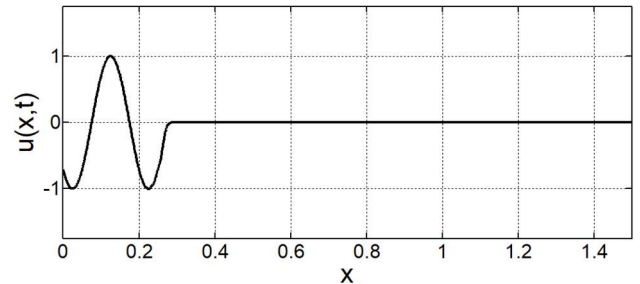
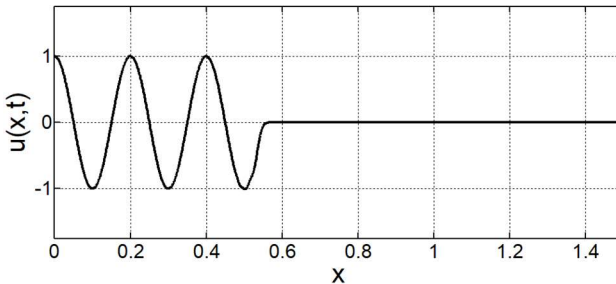
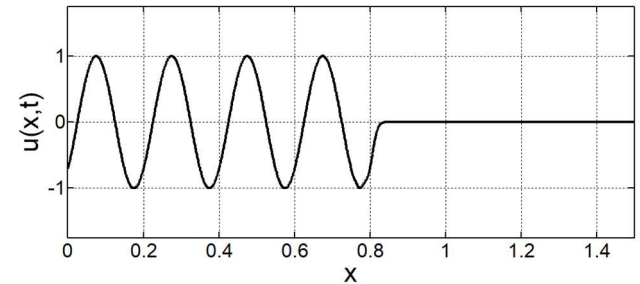
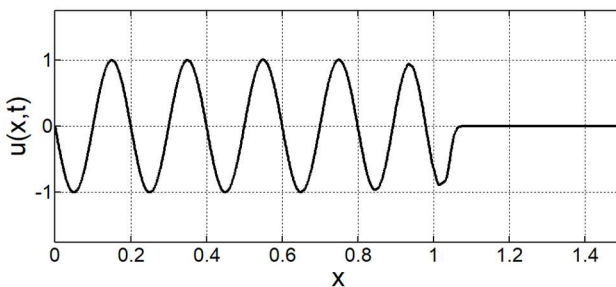
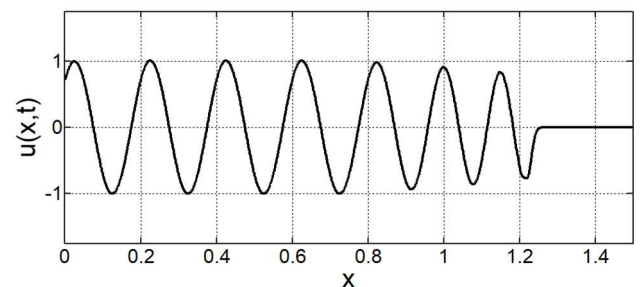
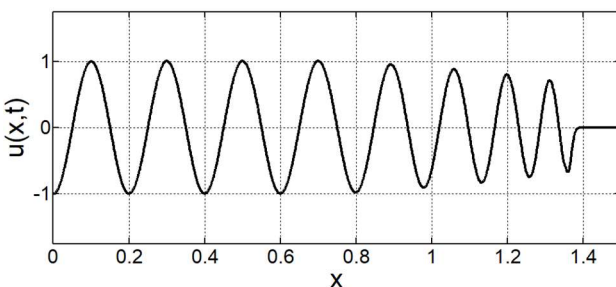
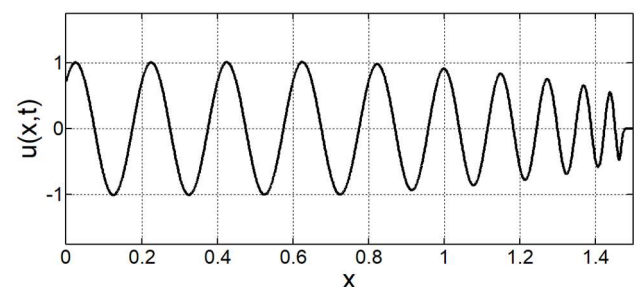
Рис. 3.80 – хвиля в момент часу  $t = 1575$ .

На рис. 3.81-3.88 зображені різні моменти часу для чисельного розв'язку рівняння (0.3), отриманого за допомогою методу Кранка – Ніколсона (0.39), з використанням першого способу додання штучної в'язкості (0.41), та з параметрами  $c = 1$ ,  $\Delta x = 0,001$ ,  $\Delta t = 0,001$ , 1500 точок по  $x$ , початковою умовою  $u(x,0) = 0$  та граничною умовою  $u(0,t) = \sin 10\pi(x-t)$ . Коефіцієнт в'язкості  $\nu$  дорівнює 0 для перших 50% точок, та лінійно збільшується на інтервалі  $[0; 20]$  для останніх 50% точок.

Рис. 3.81 – хвиля в момент часу  $t = 0$ .Рис. 3.82 – хвиля в момент часу  $t = 225$ .Рис. 3.83 – хвиля в момент часу  $t = 450$ .Рис. 3.84 – хвиля в момент часу  $t = 675$ .Рис. 3.85 – хвиля в момент часу  $t = 900$ .Рис. 3.86 – хвиля в момент часу  $t = 1125$ .Рис. 3.87 – хвиля в момент часу  $t = 1350$ .Рис. 3.88 – хвиля в момент часу  $t = 1575$ .



На рис. 3.89-3.96 зображені різні моменти часу для чисельного розв'язку рівняння (0.3), отриманого за допомогою методу Кранка – Ніколсона (0.39), з використанням другого способу додання штучної в'язкості (0.42), та з параметрами  $c = 1$ ,  $\Delta x = 0,001$ ,  $\Delta t = 0,001$ , 1500 точок по  $x$ , початковою умовою  $u(x,0) = 0$  та граничною умовою  $u(0,t) = \sin 10\pi(x-t)$ . Коефіцієнт в'язкості  $\nu$  дорівнює 0 для перших 50% точок, та лінійно збільшується на інтервалі  $[0; 0,999]$  для останніх 50% точок.

Рис. 3.89 – хвиля в момент часу  $t=0$ .Рис. 3.90 – хвиля в момент часу  $t = 275$ .Рис. 3.91 – хвиля в момент часу  $t = 550$ .Рис. 3.92 – хвиля в момент часу  $t = 825$ .Рис. 3.93 – хвиля в момент часу  $t=1100$ .Рис. 3.94 – хвиля в момент часу  $t=1375$ .Рис. 3.95 – хвиля в момент часу  $t=1650$ .Рис. 3.96 – хвиля в момент часу  $t=1925$ .

### 3.5 Дослідження результатів машинної оптимізації

Під час проведення серії експериментів за розробленим алгоритмом було розглянуто наступні закони, або функції, за якими відбувався вибір коефіцієнтів штучної в'язкості:

- Константне значення  $v(x) = const$ .
- Задання за допомогою генератора випадкових чисел на проміжку  $[0, c^2)$  та їх сортуванням за зростанням.
- Лінійна функція  $v(x) = kx$
- Степенева функція  $v(x) = x^k$
- Коренева функція  $v(x) = x^{1/k}$ , як частинний випадок степеневої функції
- Кубічна крива Безьє, яка задається чотирьома точками  $P_0, P_1, P_2, P_3$ , та має вигляд  $v(x) = (1-x)^3 P_0 + 3x(1-x)^2 P_1 + 3x^2(1-x) P_2 + x^3 P_3$  [22].

В процесі тестування алгоритму було виявлено, що задання коефіцієнтів в'язкості константним значенням створює велику кількість похибок, що виникають внаслідок відбиття хвилі від області з штучною в'язкістю. Оскільки похибки мають досить високий вплив на отримуваний результат, тому варіант з константними значеннями було відхилено. Додатково було висунуто припущення, що значення коефіцієнтів мають зростати від мінімуму до максимуму неперервно, без розривів, тобто в якості закону розподілення має бути неперервно диференційована функція.

Також під час тестування було з'ясовано, що використання випадкових чисел у відсортованому порядку майже не має сенсу, адже подібне розподілення наближається до лінійної функції (рис. 3.97).

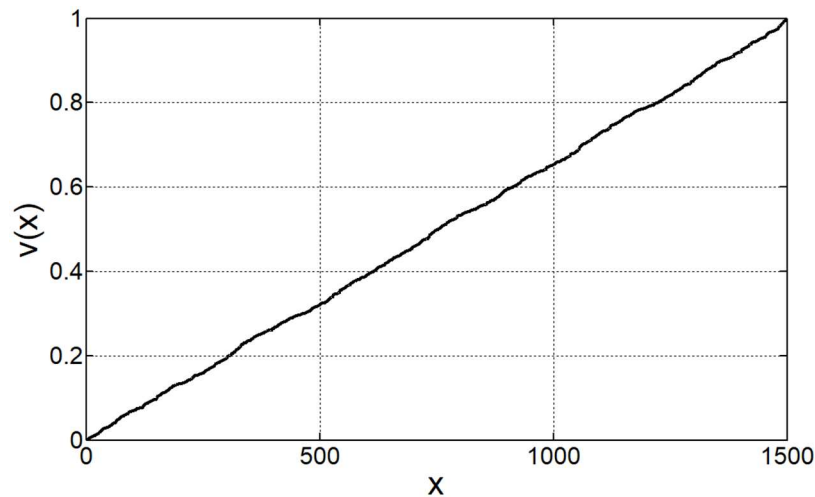


Рис. 3.97 – відсортовані у порядку зростання 1500 випадкових значень на проміжку  $[0,1]$

Ґрунтуючись на схожості з лінійною функцією, біло прийняте рішення не розглядати далі даний розподіл.

В якості об'єкта дослідження, під час тестування машинної оптимізації, було використано лінійне хвильове рівняння (0.3), з параметрами  $c=1$ ,  $\Delta x=0,001$ ,  $\Delta t=0,001$ , 3000 точок по  $x$ , початковою умовою  $u(x,0)=0$  та граничною умовою  $u(0,t)=\sin 10\pi(x-t)$ . Коефіцієнт в'язкості  $\nu$  дорівнює 0 для перших 50% точок, та лінійно збільшується за одним із законів розподілення для останніх 50% точок.

### 3.5.1 Лінійна функція

Лінійна функція – функція вигляду  $y=kx+b$ . Принцип роботи алгоритму полягає в знаходженні чисельного розв'язку при постійній зміні параметру  $k$ , а параметр  $b$  завжди рівний нулю.

На рис. 3.98-3.99 наведені результати роботи програми.



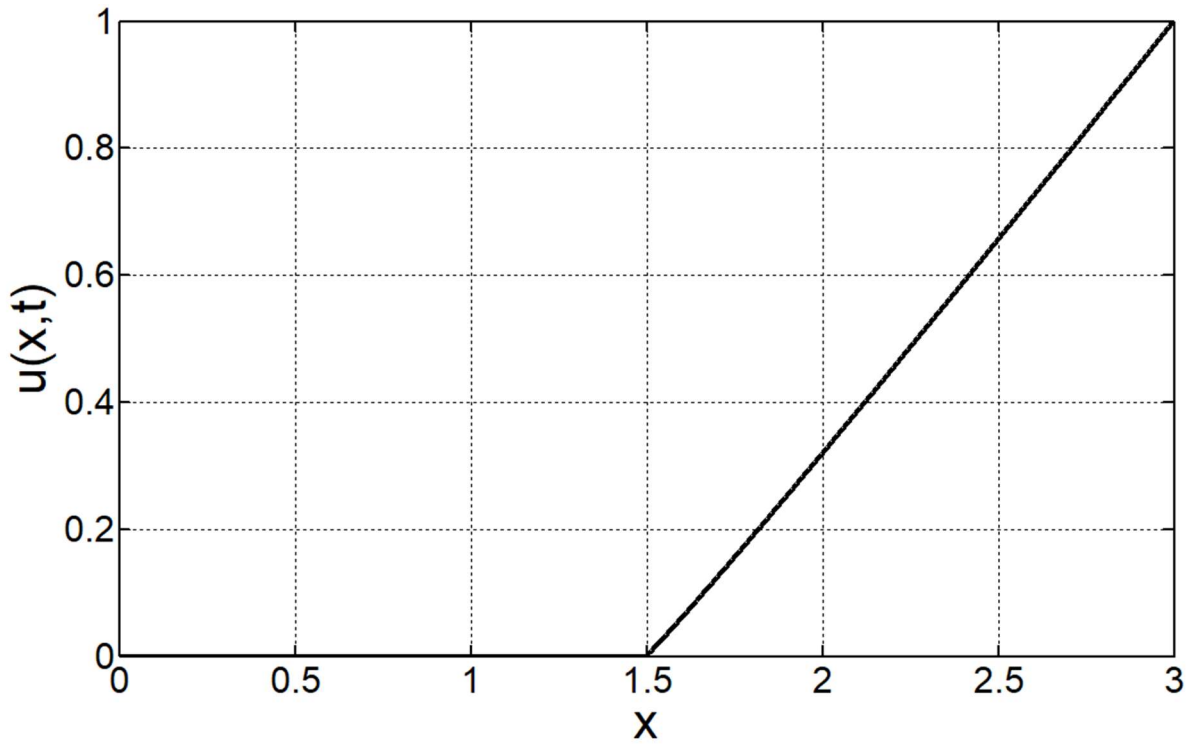
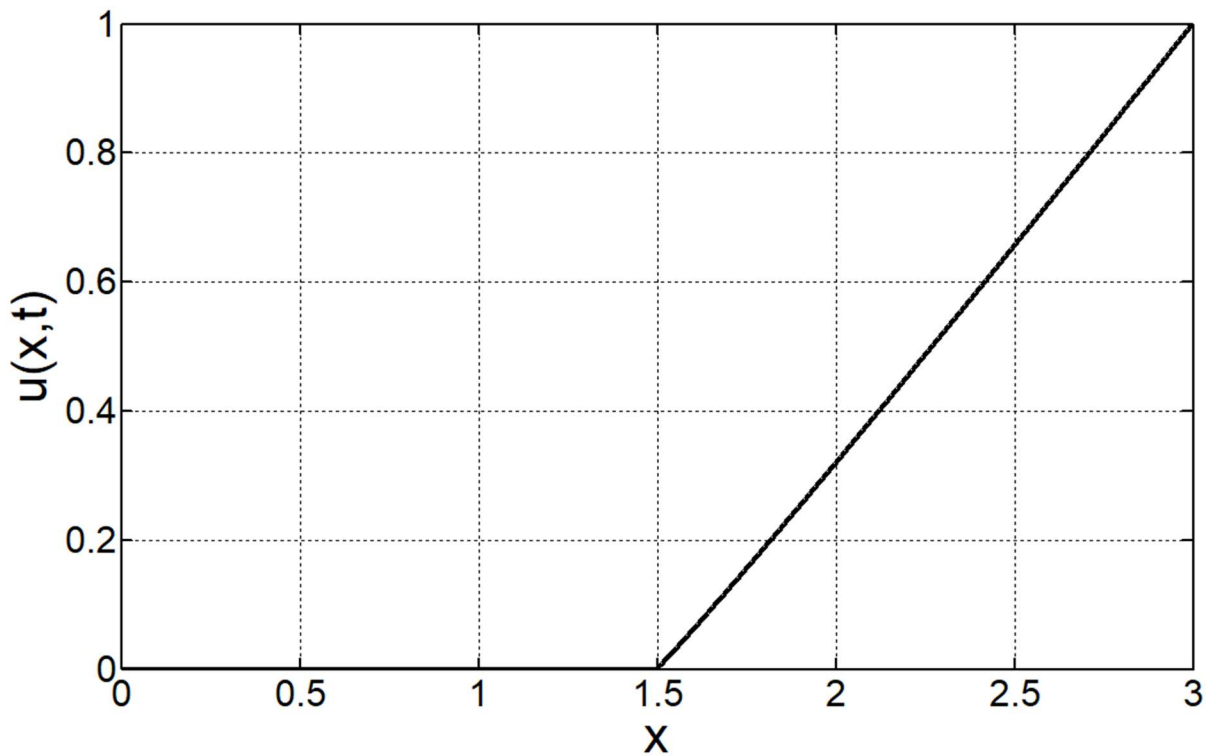


Рис. 3.98 – значення коефіцієнтів в'язкості.

Рис. 3.99 – хвиля в момент часу  $t=10000$ .

### 3.5.2 Степенева функція

Степенева функція – функція вигляду  $y = x^k$ . Принцип роботи алгоритму полягає в знаходженні чисельного розв'язку при постійній зміні параметру  $k$ .

На рис. 3.100-3.101 наведені результати роботи програми.

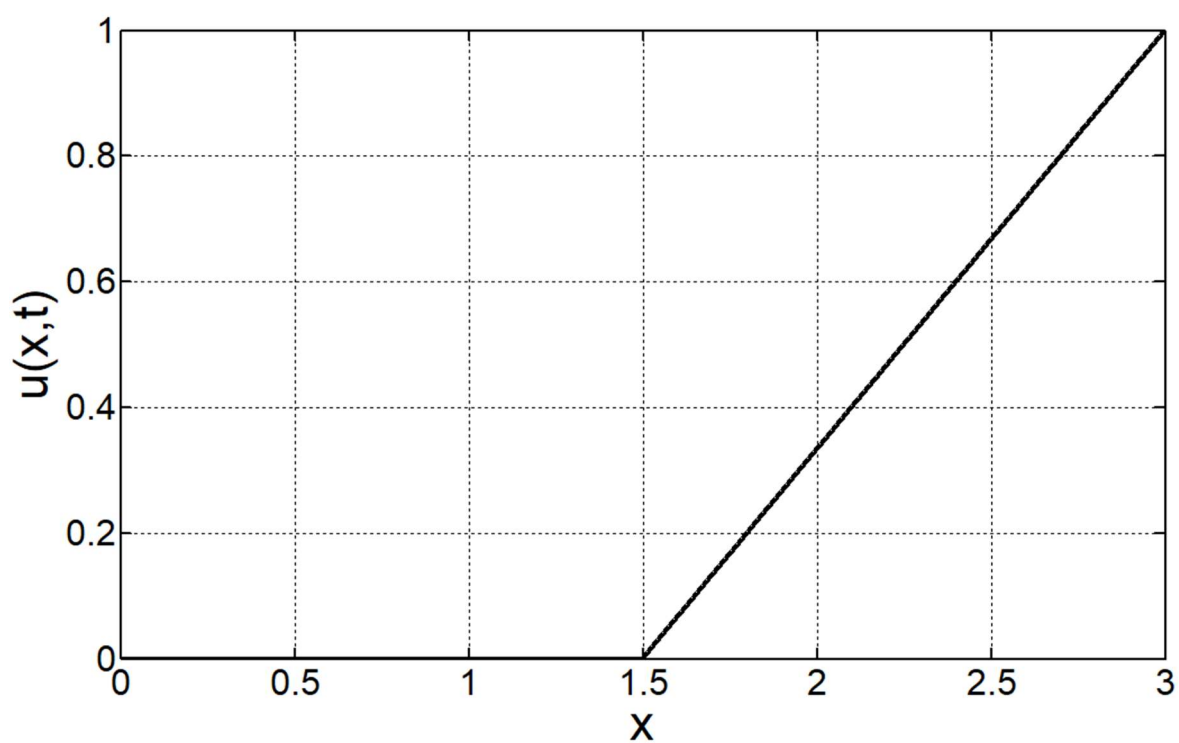


Рис. 3.100 – значення коефіцієнтів в'язкості.

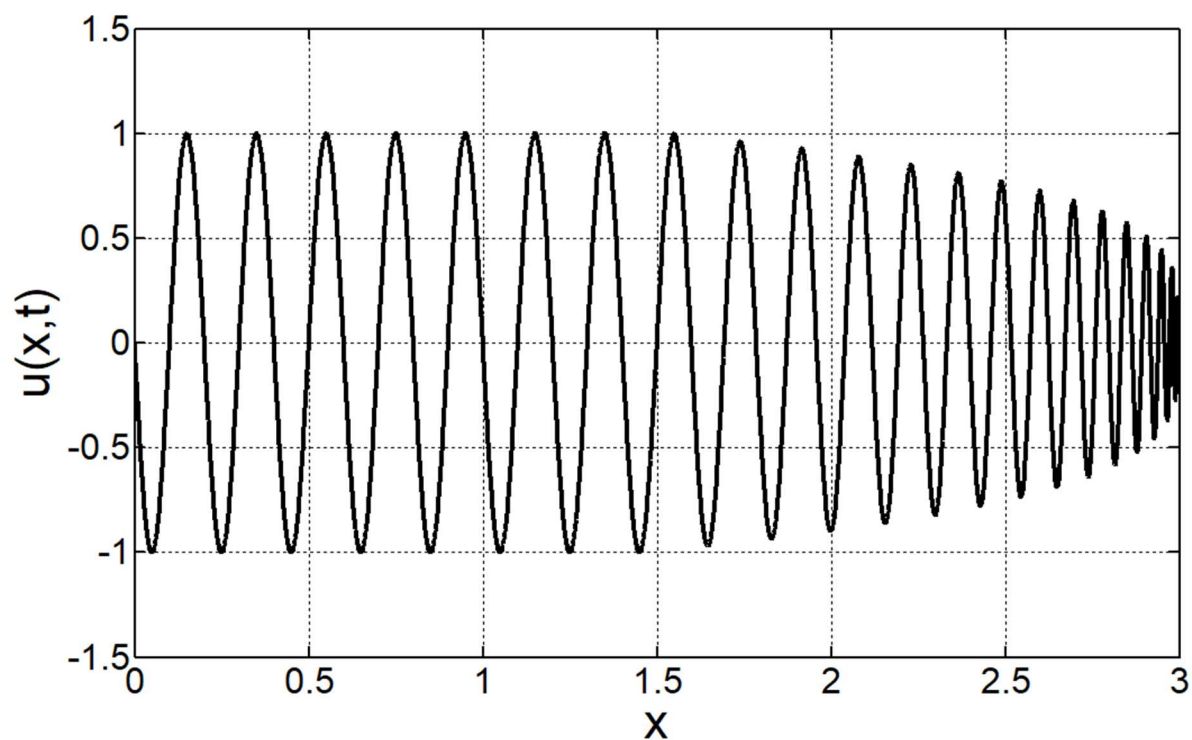


Рис. 3.101 – хвиля в момент часу  $t=4500$ .

### 3.5.3 Коренева функція

Коренева функція – функція вигляду  $y = x^{1/k}$ , частинний випадок степеневі функції. Принцип роботи алгоритму полягає в знаходженні чисельного розв'язку при постійній зміні параметру  $k$ .

На рис. 3.102-3.103 наведені результати роботи програми.

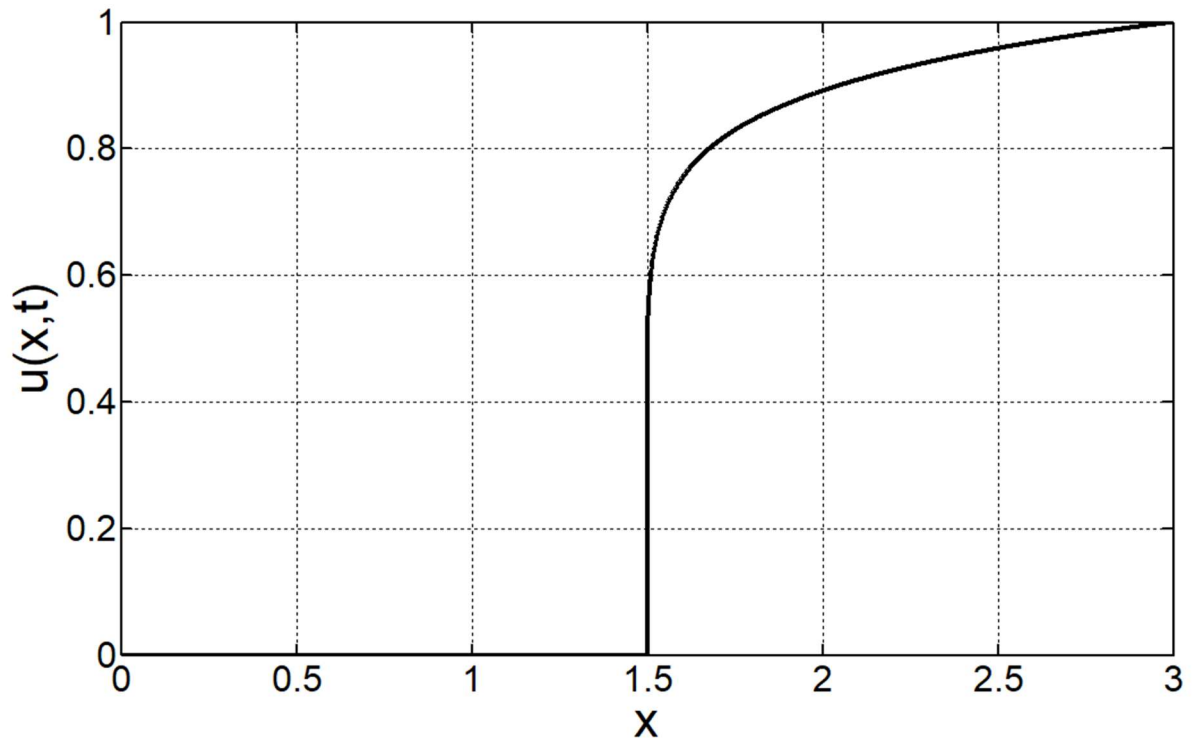


Рис. 3.102 – значення коефіцієнтів в'язкості.

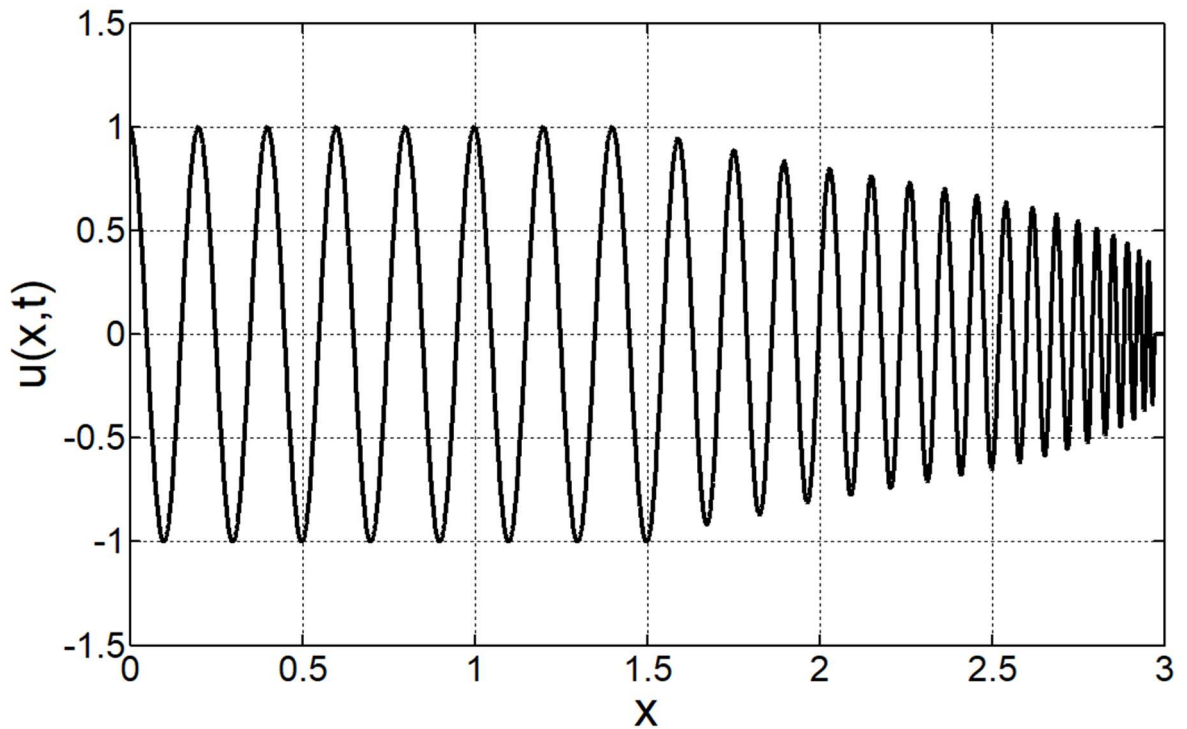


Рис. 3.103 – хвиля в момент часу  $t=5000$ .

### 3.5.4 Кубічна крива Безьє

Кубічна крива Безьє задається за допомогою чотирьох точок  $P_0, P_1, P_2, P_3$ . Точка  $P_0$  відповідає за початок кривої, а точка  $P_3$  – за її кінець. Відштовхуючись від умови другого методу задання в'язкості  $\nu \in [0, c^2)$  та обраних параметрів, отримуємо  $P_0 = (0, 0)$ , та  $P_3 = (1, 0.999)$ . Для точок  $P_1 = (x_1, y_1)$  та  $P_2 = (x_2, y_2)$  планувалося використовувати чотири цикли, щоб змінювати значення  $x_1, y_1, x_2, y_2$  від мінімальних до максимальних. Але, оскільки середа MatLab дуже повільно обробляє вкладені цикли, було прийняте рішення замінити чотири вкладених цикли на два, які будуть змінювати значення  $y_1, y_2$ , а значення  $x_1, x_2$  брати декілька раз фіксовані, такі як  $(0.15, 0.85)$ ,  $(0.25, 0.75)$  та  $(0.35, 0.65)$ .

На рис. 3.104-3.105 наведені результати роботи програми.

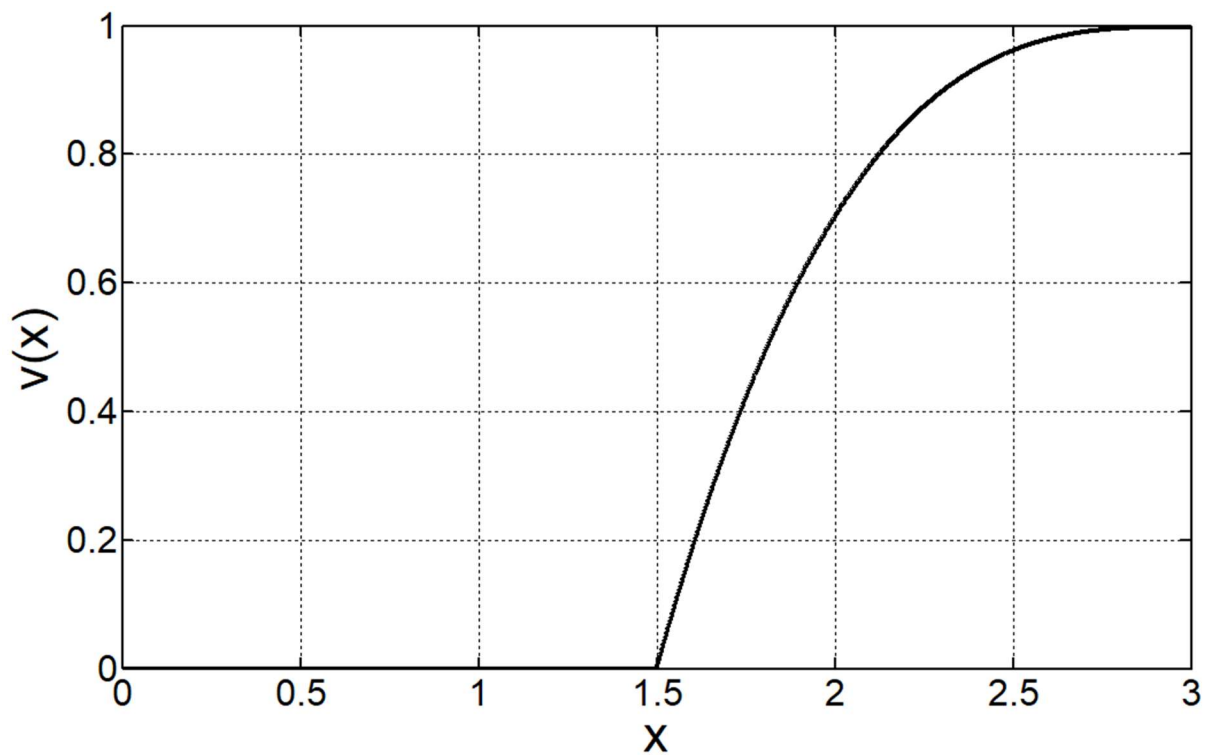
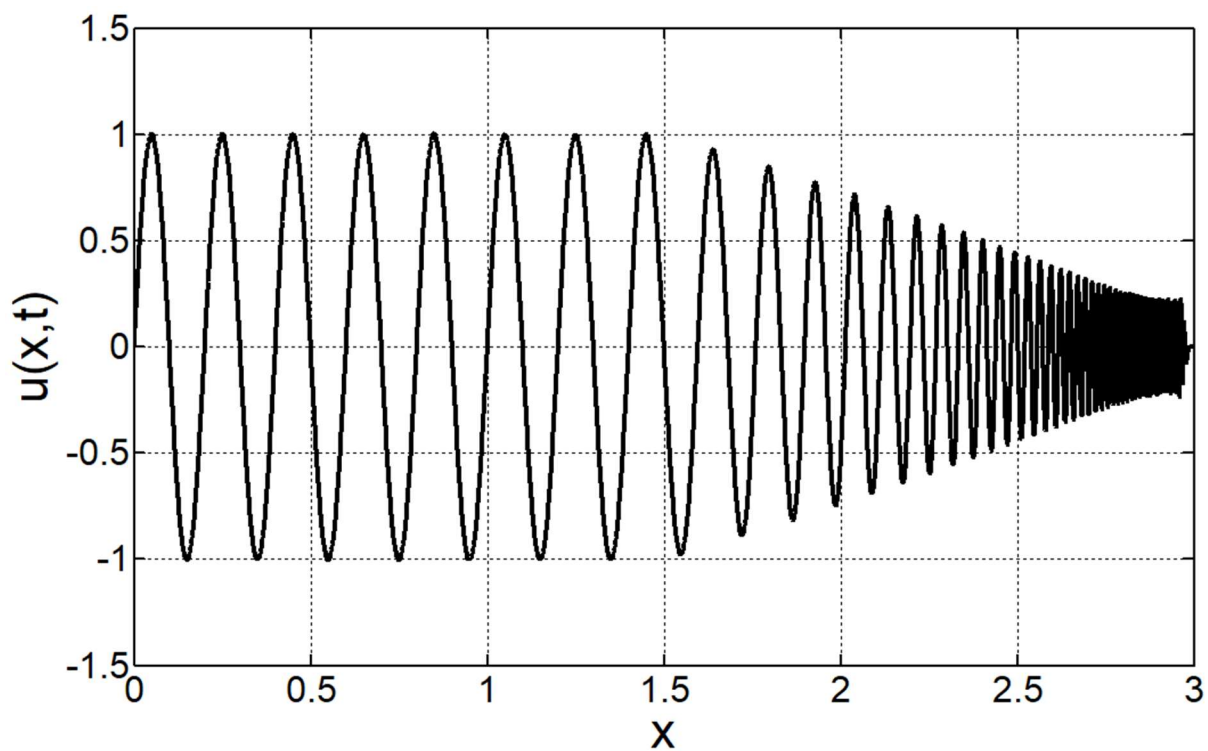


Рис. 3.104 – значення коефіцієнтів в'язкості.

Рис. 3.105 – хвиля в момент часу  $t=10000$ .

## ВИСНОВКИ

Було розглянуто декілька чисельних методів, які дозволяють отримати чисельний розв'язок для хвильового рівняння, та було обрано ті методи, які дозволяють отримати результат, максимально наближений до аналітичного, а також дозволяють зробити це якомога швидше.

Явні методи дають майже точний розв'язок для лінійного хвильового рівняння, на відміну від неявних методів, під час застосування яких накопичується похибка.

Чисельний розв'язок, отриманий за допомогою неявного методу Ейлера, має досить похибку відносно точного розв'язку, яка залежить від кроку по простору.

Перший спосіб задання штучної в'язкості за допомогою дисперсійного доданку (2.27) дозволяє досить швидко розсіяти біжучу хвилю і знизити її амплітуду майже до нульового значення. Але при доходженні хвилі до межі області з штучною в'язкістю виникає відбиття, яке накопичується з часом та через деяку кількість кроків по часу стає видимим для неозброєного ока та має вплив на результат. Також, оскільки не було знайдено необхідної літератури, цей спосіб не використовувався при проведенні експерименті з машинною оптимізацією.

Другий спосіб задання штучної в'язкості за допомогою дифузійного доданку (2.28) не дає можливість знизити амплітуду біжучої хвилі майже до нуля, але досить сильно впливає на швидкість поширення хвилі в області з штучною в'язкістю. Швидкість хвилі може знижуватися в декілька разів, тому для подальшого проходження хвилі знадобиться набагато більше часу.

Під час тестування алгоритму машинної оптимізації було використано чотири функції розподілу для задання коефіцієнтів штучної в'язкості. Судячи по отриманим результатам, можна побачити, що оптимальне значення вихідної

амплітуди досягається при використанні випуклої функції. В досліджуваному випадку такі функції є коренева та кубічна криве Безье, які дають на виході майже однакове мінімальне значення амплітуди. Але при цьому, хвиля, що рухається в області з штучною в'язкістю при використанні кривої Безье, має набагато меншу швидкість.

Після проведення всіх дослідів з машинною оптимізацією, можна зробити висновок, що було розроблене таке середовище, яке затримує просування хвилі в ньому, або розсіює її. Дане середовище можна використовувати в подальшому для дослідження динаміки брізера нелінійного хвильового рівняння.

Також, необхідно зазначити, що для повного розкриття потенціалу алгоритму необхідні досить потужні обчислювальні ресурси, бо для перебору параметрів функцій розподілень на великому проміжку з досить малим кроком, необхідно дуже багато ітерацій, тобто часу.

Наступним кроком модернізації алгоритму є використання більшої кількості законів та функцій розподілу, а також тестування алгоритму на великій кількості хвиль різного типу з різними параметрами.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. John T. Cannon, *The Evolution of Dynamics: Vibration Theory from 1687 to 1742*. – Springer-Verlag New York, 1981. – p. 192.
2. Allan D. Pierce, *Acoustics. An Introduction to Its Physical Principles and Applications*. – Springer, 2019. – p. 797.
3. David Speiser, *Discovering the Principles of Mechanics 1600-1800: Essays by David Speiser*. – Birkhäuser, 2008. – p. 344.
4. Jean le Rond d'Alembert, *Recherches sur la courbe que forme une corde tendue mise en vibration*. – Berlin, 1749. – p. 219.
5. Jesús Cuevas-Maraver, Panayotis G. Kevrekidis, Floyd Williams (eds.), *The sine-Gordon Model and its Applications: From Pendula and Josephson Junctions to Gravity and High-Energy Physics*. – Springer International Publishing, 2014. – p. 271.
6. Vladimir G. Ivancevic, Tijana T. Ivancevic, “Sine--Gordon Solitons, Kinks and Breathers as Physical Models of Nonlinear Excitations in Living Cellular Structures”, *J. Geom. Symmetry Phys.* 31(none), pp. 1-56, 2013.
7. Sine-Gordon equation [Electronic resource]. – Mode of access: [https://en.wikipedia.org/wiki/Sine-Gordon\\_equation/](https://en.wikipedia.org/wiki/Sine-Gordon_equation/)
8. Andrei D. Polyanin, Valentin F. Zaitsev, *Handbook of Ordinary Differential Equations: Exact Solutions, Methods, and Problems*. – Chapman and Hall/CRC, 2018. – p. 1496.
9. What Are Boundary Conditions? [Electronic resource]. – Mode of access: <https://www.simscale.com/docs/simwiki/numerics-background/what-are-boundary-conditions/>
10. Sturm-Liouville problem [Electronic resource]. – Mode of access: [https://www.encyclopediaofmath.org/index.php/Sturm-Liouville\\_problem/](https://www.encyclopediaofmath.org/index.php/Sturm-Liouville_problem/)
11. Sturm-Liouville theory [Electronic resource]. – Mode of access: [https://www.encyclopediaofmath.org/index.php/Sturm-Liouville\\_theory/](https://www.encyclopediaofmath.org/index.php/Sturm-Liouville_theory/)



12. Eugene Isaacson, Herbert Bishop Keller, *Analysis of Numerical Methods*. – Dover Publications, 1994. – p. 558.
13. Crank, J.; Nicolson, P., "A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of Heat Conduction Type", *Proc. Camb. Phil. Soc.*, 43: pp. 50–67, 1947.
14. Peter J. Olver, *Introduction to Partial Differential Equations*. – Springer International Publishing, 2014. – p. 652.
15. Paul Wilmott, Sam Howison, Jeff Dewynne, *The Mathematics of Financial Derivatives: A Student Introduction*. – Cambridge University Press, 1995. – p. 339.
16. Christian Grossmann, Hans-Görg Roos, Martin Stynes, *Numerical treatment of partial differential equations*. – Springer, 2007. – p. 601.
17. Nicholas J. Higham, *Accuracy and Stability of Numerical Algorithms*. – Society for Industrial and Applied Mathematics, 1996. – p. 718.
18. J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. – Wiley, 2016. – p. 540.
19. John C. Tannehill, Dale A. Anderson, Richard H. Pletcher, *Computational Fluid Mechanics and Heat Transfer (3<sup>rd</sup> Edition)*. – Boca Raton, FL, CRC Press Taylor & Francis Group, 2012. – p. 763.
20. Vladimir N Kukudzhanov, Alexei Zhurov, *Numerical Continuum Mechanics*. – De Gruyter, 2012. – p. 442.
21. J. W. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*. – Springer, 1999. – p. 573.
22. Bézier curve [Electronic resource]. – Mode of access: [https://en.wikipedia.org/wiki/B%C3%A9zier\\_curve/](https://en.wikipedia.org/wiki/B%C3%A9zier_curve/)

## ДОДАТОК А

Програмний код для реалізації прямого методу Ейлера:

```
function result = explicit_Euler_with_viscosity(X, steps_t, h_t, wave_func, c, visc_type,
visc_filling, visc_max)
points_x = length(X);
h_x = X(2) - X(1);
result = zeros(steps_t+2, points_x);
result(1, :) = [wave_func(X(1), -h_t), zeros(1, points_x-1)];
result(2, :) = [wave_func(X(1), 0), zeros(1, points_x-1)];
visc_coef = viscosity(points_x-2, visc_filling, visc_max);
t = 0;
for i = 3:steps_t+2
    t = t + h_t;
    u_x = (result(i-1, 3:end) - result(i-1, 1:end-2)) / (2 * h_x);
    u_xx = (result(i-1, 1:end-2) - 2 * result(i-1, 2:end-1) + result(i-1, 3:end)) /
h_x^2;
    visc = 0;
    if visc_type == 1
        visc = visc_coef .* u_x;
    elseif visc_type == 2
        visc = -visc_coef .* u_xx;
    end
    result(i, 2:end-1) = h_t^2 * (c^2 * u_xx + visc) + 2 * result(i-1, 2:end-1) -
result(i-2, 2:end-1);
    result(i, 1) = wave_func(X(1), t);
    result(i, end) = result(i-1, end-1);
end
end

function result = viscosity(len, filling_percent, max_coef)
start = round(len * (100 - filling_percent) / 100);
h = max_coef / (len - start);
result = [zeros(1, start - 1), 0:h:max_coef];
end
```

## ДОДАТОК Б

Програмний код для реалізації методу Рунге – Кутта 4 порядку:

```
function result_u = runge_kutta(X, steps_t, h_t, wave_func, c, visc_type, visc_filling,
visc_max)
points_x = length(X);
h_x = X(2) - X(1);
result_u = zeros(steps_t+1, points_x);
result_z = zeros(steps_t+1, points_x);
visc_coef = viscosity(points_x-2, visc_filling, visc_max);
t = 0;
for i = 2:steps_t+1
    t = t + h_t;
    u_x = (result_u(i-1, 3:end) - result_u(i-1, 1:end-2)) / (2 * h_x);
    u_xx = (result_u(i-1, 1:end-2) - 2 * result_u(i-1, 2:end-1) + result_u(i-1, 3:end)) /
h_x^2;
    visc = zeros(1, points_x-2);
    if visc_type == 1
        visc = visc_coef .* u_x;
    elseif visc_type == 2
        visc = -visc_coef .* u_xx;
    end
    func_u = @(u, z, t, h_x) z;
    func_z = @(u, z, t, h_x) c^2 * ([wave_func(X(1)-h_x, t), u(1:end-1)] - 2 * u +
[u(2:end), 0]) / h_x^2 + [0, visc, 0];
    [result_u(i, :), result_z(i, :)] = next_step(func_u, func_z, result_u(i-1, :),
result_z(i-1, :), t, h_t, h_x);
end
end

function [u_next, z_next] = next_step(func_u, func_z, u, z, t, h_t, h_x)
k1 = h_t * func_u(u, z, t, h_x);
m1 = h_t * func_z(u, z, t, h_x);
k2 = h_t * func_u(u + k1 / 2, z + m1 / 2, t + h_t / 2, h_x);
m2 = h_t * func_z(u + k1 / 2, z + m1 / 2, t + h_t / 2, h_x);
k3 = h_t * func_u(u + k2 / 2, z + m2 / 2, t + h_t / 2, h_x);
m3 = h_t * func_z(u + k2 / 2, z + m2 / 2, t + h_t / 2, h_x);
k4 = h_t * func_u(u + k3, z + m3, t + h_t, h_x);
m4 = h_t * func_z(u + k3, z + m3, t + h_t, h_x);
u_next = u + (k1 + 2 * k2 + 2 * k3 + k4) / 6;
z_next = z + (m1 + 2 * m2 + 2 * m3 + m4) / 6;
end

function result = viscosity(len, filling_percent, max_coef)
start = round(len * (100 - filling_percent) / 100);
h = max_coef / (len - start);
result = [zeros(1, start - 1), 0:h:max_coef];
end
```

## ДОДАТОК В

Програмний код для реалізації непрямого методу Ейлера:

```
function result = implicit_Euler(X, steps_t, h_t, wave_func, c, visc_type, visc_filling,
visc_max)
points_x = length(X);
h_x = X(2) - X(1);
result = zeros(steps_t+2, points_x);
result(1, :) = [wave_func(X(1), -h_t), zeros(1, points_x-1)];
result(2, :) = [wave_func(X(1), 0), zeros(1, points_x-1)];
b = c^2 * h_t^2 / h_x^2;
A = eye(points_x);
for i = 2:points_x-1
    A(i, i) = 1 + 2 * b;
    A(i, i-1) = -b;
    A(i, i+1) = -b;
end
visc_coef = viscosity(points_x-2, visc_filling, visc_max);
t = 0;
for i = 3:steps_t+2
    t = t + h_t;
    u_x = (result(i-1, 3:end) - result(i-1, 1:end-2)) / (2 * h_x);
    u_xx = (result(i-1, 1:end-2) - 2 * result(i-1, 2:end-1) + result(i-1, 3:end)) /
h_x^2;
    visc = 0;
    if visc_type == 1
        visc = visc_coef .* u_x;
    elseif visc_type == 2
        visc = -visc_coef .* u_xx;
    end
    R = 2 * result(i-1, :) - result(i-2, :);
    R(2:end-1) = R(2:end-1) + h_t^2 * visc;
    R(1) = wave_func(X(1), t);
    R(end) = result(i-1, end-1);
    res = A \ R;
    result(i, :) = res;
end
end

function result = viscosity(len, filling_percent, max_coef)
start = round(len * (100 - filling_percent) / 100);
h = max_coef / (len - start);
result = [zeros(1, start - 1), 0:h:max_coef];
end
```

## ДОДАТОК Г

Програмний код для реалізації методу Кранка – Ніколсона:

```
function result = crank_nicolson(X, steps_t, h_t, wave_func, c, visc_type, visc_filling,
visc_max)
points_x = length(X);
h_x = X(2) - X(1);
result = zeros(steps_t+2, points_x);
result(1, :) = [wave_func(X(1), -h_t), zeros(1, points_x-1)];
result(2, :) = [wave_func(X(1), 0), zeros(1, points_x-1)];
b = c^2 * h_t^2 / h_x^2 / 2;
A = eye(points_x);
for i = 2:points_x-1
    A(i, i) = 1 + 2 * b;
    A(i, i-1) = -b;
    A(i, i+1) = -b;
end
visc_coef = viscosity(points_x-2, visc_filling, visc_max);
t = 0;
for i = 3:steps_t+2
    t = t + h_t;
    u_x = (result(i-1, 3:end) - result(i-1, 1:end-2)) / (2 * h_x);
    u_xx = (result(i-1, 1:end-2) - 2 * result(i-1, 2:end-1) + result(i-1, 3:end)) /
h_x^2;
    visc = 0;
    if visc_type == 1
        visc = visc_coef .* u_x;
    elseif visc_type == 2
        visc = -visc_coef .* u_xx;
    end
    R = 2 * result(i-1, :) - result(i-2, :);
    R(2:end-1) = R(2:end-1) + h_t^2 * visc' + ...
        b * (result(i-2, 1:end-2)' - 2 * result(i-2, 2:end-1)' + result(i-2, 3:end)');
    R(1) = wave_func(X(1), t);
    R(end) = result(i-1, end-1);
    res = A \ R;
    result(i, :) = res';
end
end

function result = viscosity(len, filling_percent, max_coef)
start = round(len * (100 - filling_percent) / 100);
h = max_coef / (len - start);
result = [zeros(1, start - 1), 0:h:max_coef];
end
```

## ДОДАТОК Г

Програмний код для машинної оптимізації знаходження коефіцієнтів штучної в'язкості:

```
function [viscosity_final, A_final] = viscosity_generator(X, h_t, wave_properties)
[wave_type, A_0, k, w, c] = get_wave_parameters(wave_properties);
wave_function = get_wave_function(wave_type, A_0, k, w);
viscosity_distribution_types = get_viscosity_distribution_types();
A_final = A_0;
viscosity_final = zeros(1, length(X));
for i = 1:length(viscosity_distribution_types)
    [viscosity, A] = test_viscosity_distribution_type(viscosity_distribution_types{i},
    wave_function, c, X, h_t);
    if A < A_final
        A_final = A;
        viscosity_final = viscosity;
    end
end
end

function [type, A, k, w, c] = get_wave_parameters(wave_properties)
for i = 1:2:length(wave_properties)
    switch wave_properties{i}
        case 'type'
            type = wave_properties{i + 1};
        case 'A'
            A = wave_properties{i + 1};
        case 'k'
            k = wave_properties{i + 1};
        case 'w'
            w = wave_properties{i + 1};
        case 'c'
            c = wave_properties{i + 1};
        otherwise
            error(strcat('Invalid input parameter name at position', 32, num2str(i),
            '.'))
    end
end
end

function wave_function = get_wave_function(type, A_0, k, w)
switch type
    case 'sin'
        wave_function = @(x, t) A_0 * sin(k * x - w * t);
    case 'cos'
        wave_function = @(x, t) A_0 * cos(k * x - w * t);
    otherwise
        error('Invalid wave type.')
end
end

function types = get_viscosity_distribution_types()
types = {'line', 'power', 'nth-root', 'cubic-bezier'};
end

function [viscosity_best, A_best] = test_viscosity_distribution_type(type, wave_function,
c, X, h_t)
iterations = 4;
A_best = inf;
viscosity_best = zeros(1, length(X));
switch type
    case 'line'
        for v = 0.001 : (c-0.001)/iterations : c-0.001
```

```

        step = v / (length(X)/2-1);
        visc_tmp = 0:step:v;
        visc_coef = [zeros(1, length(X)/2-1), visc_tmp, max(visc_tmp) * ones(1,
length(X)/2-1)];
        [A_min, A_max] = get_A_from_numerical_solution(wave_function, X, h_t, c,
visc_coef);
        if min(abs(A_min), abs(A_max)) < A_best
            A_best = min(abs(A_min), abs(A_max));
            viscosity_best = visc_coef(1 : length(X));
        end
    end
    case 'nth-root'
        for v = 1.001 : 9/iterations : 10.001
            step = (c-0.001) / (length(X)/2-1);
            visc_tmp = (0:step:(c-0.001)).^(1/v);
            visc_coef = [zeros(1, length(X)/2-1), visc_tmp, max(visc_tmp) * ones(1,
length(X)/2-1)];
            [A_min, A_max] = get_A_from_numerical_solution(wave_function, X, h_t, c,
visc_coef);
            if min(abs(A_min), abs(A_max)) < A_best
                A_best = min(abs(A_min), abs(A_max));
                viscosity_best = visc_coef(1 : length(X));
            end
        end
    case 'power'
        for v = 1.001 : 9/iterations : 10.001
            step = (c-0.001) / (length(X)/2-1);
            visc_tmp = (0:step:(c-0.001)).^v;
            visc_coef = [zeros(1, length(X)/2-1), visc_tmp, max(visc_tmp) * ones(1,
length(X)/2-1)];
            [A_min, A_max] = get_A_from_numerical_solution(wave_function, X, h_t, c,
visc_coef);
            if min(abs(A_min), abs(A_max)) < A_best
                A_best = min(abs(A_min), abs(A_max));
                viscosity_best = visc_coef(1 : length(X));
            end
        end
    case 'cubic-bezier'
        x_min = 0; y_min = 0; x_max = 1; y_max = c - 0.002;
        step = 5;
        x1 = 0.35; x2 = 0.65;
        for y1 = y_min:y_max/step:y_max
            for y2 = y_min:y_max/step:y_max
                [~, visc_tmp] = cubic_Bezier([x_min, x1, x2, x_max], [y_min, y1, y2,
y_max], length(X)/2);
                visc_coef = [zeros(1, length(X)/2-1), visc_tmp, max(visc_tmp) * ones(1,
length(X)/2-1)];
                [A_min, A_max] = get_A_from_numerical_solution(wave_function, X, h_t, c,
visc_coef);
                if min(abs(A_min), abs(A_max)) < A_best
                    A_best = min(abs(A_min), abs(A_max));
                    viscosity_best = visc_coef(1 : length(X));
                end
            end
        end
        x1 = 0.25; x2 = 0.75;
        for y1 = y_min:y_max/step:y_max
            for y2 = y_min:y_max/step:y_max
                [~, visc_tmp] = cubic_Bezier([x_min, x1, x2, x_max], [y_min, y1, y2,
y_max], length(X)/2);
                visc_coef = [zeros(1, length(X)/2-1), visc_tmp, max(visc_tmp) * ones(1,
length(X)/2-1)];
                [A_min, A_max] = get_A_from_numerical_solution(wave_function, X, h_t, c,
visc_coef);
                if min(abs(A_min), abs(A_max)) < A_best
                    A_best = min(abs(A_min), abs(A_max));
                    viscosity_best = visc_coef(1 : length(X));
                end
            end
        end
    end
end
end

```

```

        x1 = 0.15; x2 = 0.85;
        for y1 = y_min:y_max/step:y_max
            for y2 = y_min:y_max/step:y_max
                [~, visc_tmp] = cubic_Bezier([x_min, x1, x2, x_max], [y_min, y1, y2,
y_max], length(X)/2);
                visc_coef = [zeros(1, length(X)/2-1), visc_tmp, max(visc_tmp) * ones(1,
length(X)/2-1)];
                [A_min, A_max] = get_A_from_numerical_solution(wave_function, X, h_t, c,
visc_coef);
                if min(abs(A_min), abs(A_max)) < A_best
                    A_best = min(abs(A_min), abs(A_max));
                    viscosity_best = visc_coef(1 : length(X));
                end
            end
        end
    otherwise
        error('Invalid viscosity distribution type.')
    end
end
end

```

```

function [A_min, A_max] = get_A_from_numerical_solution(wave_function, X, h_t, c,
visc_coef)
h_x = X(2) - X(1);
XX = [X, X + h_x * 1:length(X)/2];
points_x = length(XX);
result = zeros(points_x*3+2, points_x);
result(1, :) = [wave_function(XX(1), -h_t), zeros(1, points_x-1)];
result(2, :) = [wave_function(XX(1), 0), zeros(1, points_x-1)];
t = 0; A_max = 0; A_min = 0;
for i = 3:points_x*3+2
    t = t + h_t;
    u_xx = (result(i-1, 1:end-2) - 2 * result(i-1, 2:end-1) + result(i-1, 3:end)) /
h_x^2;
    visc = -visc_coef .* u_xx;
    result(i, 2:end-1) = h_t^2 * (c^2 * u_xx + visc) + 2 * result(i-1, 2:end-1) -
result(i-2, 2:end-1);
    result(i, 1) = wave_function(XX(1), t);
    result(i, end) = result(i-1, end-1);
    A_max = max(A_max, max(result(i, length(X)-50:length(X))));
    A_min = min(A_min, min(result(i, length(X)-50:length(X))));
end
end

```

```

function [Qx, Qy] = cubic_Bezier(Px, Py, n)
Px0 = Px(1); Py0 = Py(1);
Px1 = Px(2); Py1 = Py(2);
Px2 = Px(3); Py2 = Py(3);
Px3 = Px(4); Py3 = Py(4);
cx3 = -Px0 + 3 * (Px1 - Px2) + Px3;
cy3 = -Py0 + 3 * (Py1 - Py2) + Py3;
cx2 = 3 * (Px0 - 2 * Px1 + Px2);
cy2 = 3 * (Py0 - 2 * Py1 + Py2);
cx1 = 3 * (Px1 - Px0);
cy1 = 3 * (Py1 - Py0);
cx0 = Px0; cy0 = Py0;
dt = 1 / (n - 1);
t = 0;
Qx = [Px0, zeros(1, n - 1)];
Qy = [Py0, zeros(1, n - 1)];
for i = 2:n
    t = t + dt;
    Qx(i) = ((cx3 * t + cx2) * t + cx1) * t + cx0;
    Qy(i) = ((cy3 * t + cy2) * t + cy1) * t + cy0;
end
end

```