

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА**

**Факультет математики та інформатики  
Кафедра математичного моделювання**

**Побудова різницевих схем Адамса-Башфорта, Адамса-Маултона,  
Гіра та дослідження їх стійкості**

**Кваліфікаційна робота  
Рівень вищої освіти – перший (бакалаврський)**

***Виконав:***

студент 4 курсу, 401 групи

**Диренко Віктор Вікторович**

***Керівник:***

доктор фізико-математичних наук,

професор Черевко І.М.

*До захисту допущено*

*на засіданні кафедри математичного моделювання*

*протокол № 16 від 30 травня 2023 р.*

*Зав. кафедрою \_\_\_\_\_ проф. Черевко І.М.*

## АНОТАЦІЯ

У роботі досліджено умови коректності багатокрокових різницевих схем у вигляді системи лінійних алгебраїчних рівнянь. Одержано коефіцієнти різницевих схем Адамса-Башфорта, Адамса-Маултона та Гіра і запропоновано алгоритм побудови їх областей абсолютної стійкості, який можна ефективно реалізувати на ЕОМ.

В роботі розроблена прикладна програма, яка дозволяє знаходити коефіцієнти різницевих схем Адамса-Башфорта, Адамса-Маултона та Гіра довільного порядку та будувати області їх абсолютної стійкості.

**Ключові слова:** Різницеві схеми Адамса-Башфорта, різницеві схеми Адамса-Маултона, різницеві схеми Гіра, область абсолютної стійкості, умови коректності, жорстко стійкі різницеві схеми.

## ANNOTATION

The work examines the correctness conditions of multi-step difference schemes in the form of a system of linear algebraic equations. The coefficients of the Adams-Bashforth, Adams-Moulton, and Gear difference schemes were obtained and an algorithm for constructing their regions of absolute stability was proposed, which can be effectively implemented on a computer.

In the work, an application program was developed that allows you to find the coefficients of the difference schemes of Adams-Bashforth, Adams-Moulton, and Gear of arbitrary order and to build regions of their absolute stability

**Keywords:** Adams-Bashforth difference schemes, Adams-Moulton difference schemes, Gear difference schemes, region of absolute stability, correctness conditions, rigidly stable difference schemes.

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів наукових досліджень інших авторів мають посилання на відповідне джерело.

\_\_\_\_\_ В.В. Диренко

## ЗМІСТ

<b>ВСТУП.....</b>	<b>5</b>
<b>1. Багатокрокові різницеві схеми.....</b>	<b>7</b>
1.1 Умови коректності для багатокрокових методів.....	7
1.2 Методи Адамса-Башфорта.....	8
1.3 Методи Адамса-Маултона.....	10
<b>2. Стійкість багатокрокових різницевих схем.....</b>	<b>13</b>
2.1 Необхідна умова стійкості.....	13
2.2 Критерій стійкості багатокрокових різницевих схем.....	14
<b>3. Абсолютна стійкість багатокрокових різницевих схем.....</b>	<b>15</b>
3.1 Алгоритм побудови області абсолютної стійкості.....	15
3.2 Області абсолютної стійкості різницевих схем Адамса-Башфорта та Адамса-Маултона.....	17
<b>4. Жорстко стійкі різницеві схеми.....</b>	<b>21</b>
4.1 Жорсткі системи диференціальних рівнянь.....	21
4.2 Жорстко стійкі різницеві схеми.....	22
<b>5. Різницеві схеми Гіра.....</b>	<b>24</b>
5.1 Існування різницевих схем Гіра.....	24
5.2 Області абсолютної стійкості різницевих схем Гіра.....	26
<b>6. Опис використаного програмного забезпечення та розробленої прикладної програми.....</b>	<b>28</b>
6.1 Мова програмування Python та її бібліотеки.....	28
6.2 Опис розробленої прикладної програми.....	29
<b>ВИСНОВКИ.....</b>	<b>31</b>
<b>СПИСОК ЛІТЕРАТУРИ.....</b>	<b>32</b>

**ДОДАТОК 1.....34**

**ДОДАТОК 2.....53**

## Вступ

Прикладна математика сформувала новий науковий метод дослідження – обчислювальний експеримент, основними етапами якого є формування математичної моделі та застосування числових методів.

Математичні моделі прикладних процесів фізичної і хімічної кінетики, електроніки, біології, теорії керування та інших наук описуються системами диференціальних рівнянь, що містять різномасштабні змінні. Їх відносять до класу “жорстких задач”, які вимагають спеціальних числових методів. У вітчизняній науковій літературі цій тематиці не приділяється достатньо уваги.

У даній роботі здійснено аналіз багатокрокових різницевих схем, які є основним апаратом для наближеного розв’язання систем диференціальних рівнянь. Жорсткі задачі вимагають введення нових понять стійкості для різницевих схем. Тому в роботі розглядається поняття абсолютної стійкості різницевих схем і обговорюється необхідна область абсолютної стійкості, щоб можна застосувати різницеву схему до наближеного розв’язання жорстких задач.

Основна увага в даній роботі приділяється різницеvim схемам Адамса-Башфорта, Адамса-Маултона та Гіра. Досліджено які з них є жорстко стійкими та можуть бути застосовані для розв’язку жорстких задач.

**Мета роботи** - побудова явних виразів для коефіцієнтів різницевих схем Адамса-Башфорта, Адамса-Маултона та Гіра довільного високого порядку та розробка алгоритму автоматизації побудови областей їх абсолютної стійкості і його реалізація у вигляді прикладної програми.

У роботі одержано систему лінійних алгебраїчних рівнянь для знаходження коефіцієнтів різницевих схем Адамса-Башфорта, Адамса-Маултона та Гіра. Запропоновано алгоритм побудови областей абсолютної стійкості різницевих схем, який можна ефективно реалізувати на ЕОМ. За допомогою мови програмування Python в роботі розроблена прикладна програма, яка дозволяє знаходити коефіцієнти різницевих схем Адамса-

Башфорта, Адамса-Маултона та Гіра досить високого порядку (Додаток 2) та будувати області їх абсолютної стійкості (Рис. 3, Рис. 4, Рис. 6).

Результати роботи доповідалися на студентській науковій конференції Чернівецького національного університету (12–14 квітня 2022 року), міжнародній науковій конференції «Прикладна математика та інформаційні технології», присвяченій 60-річчю кафедри прикладної математики та інформаційних технологій, 22–24 вересня 2022 р. та подавались на II тур Всеукраїнського конкурсу студентських наукових робіт за напрямком «Математика та статистика. Прикладна математика (механіка)» у 2022 році.

# 1. Багатокрокові різницеві схеми

## 1.1 Умови коректності для багатокрокових методів

Розглянемо загальний вигляд багатокрокового методу порядку  $k$  для знаходження розв'язку задачі Коші [1-2].

$$x_{n+1} = \sum_{i=0}^p a_i x_{n-i} + h \sum_{i=-1}^p b_i f(x_{n-i}, t_{n-i}). \quad (1)$$

Коефіцієнти  $\{a_0, a_1, \dots, b_{-1}, b_0, \dots, b_p\}$  будемо вибрати так, щоб різницева схема (1) була точною для всіх многочленних розв'язків задачі Коші для деякого степеня  $k$ . Поліном першого степеня (пряма) визначається двома параметрами, поліном другого степеня (парабола) задається трьома параметрами, тоді поліном  $k$ -того степеня має  $k+1$  параметрів. Очевидно, що у випадку полінома  $k$ -того степеня, для різницевої схеми (1) потрібно знайти  $(k+1)$  коефіцієнтів. Звичайно, що всі ці коефіцієнти повинні задовольняти певні умови. Розглянемо ці умови на прикладі задачі Коші вигляду:

$$\dot{x} = f(x, t), \quad x(t_0) = x_0. \quad (2)$$

1) Нехай  $k = 0$  (поліном нульового степеня). Розв'язок задачі Коші у цьому випадку буде  $\hat{x}(t) = \alpha_0$ , а сама задача Коші задається у вигляді  $\dot{x} = f(x, t) = 0$ . Умова коректності вимагає, щоб

$$x_{n+1} = \alpha_0, x_{n-i} = \alpha_0, f(x_{n-i}, t_{n-i}) = 0. \quad (3)$$

Підставивши (3) в (1), отримаємо  $\alpha_0 = \sum_{i=0}^p a_i \alpha_0$ . Поділимо на  $\alpha_0$  і отримаємо

$$\sum_{i=0}^p a_i = 1. \quad (4)$$

2) Нехай  $k = 1$  (поліном першого степеня). Розв'язок задачі Коші у цьому випадку буде  $\hat{x}(t) = \alpha_0 + \alpha_1 t$ , а сама задача Коші задається у вигляді  $\dot{x} = f(x, t) = \alpha_1$ . Розглянемо умову коректності для цього випадку. Нехай  $t_n = 0$ . Тоді  $t_{n+1} = h, t_{n-1} = -h, t_{n-2} = -2h, t_{n-p} = -ph$ . Умова коректності матиме наступний вигляд:

$$x_{n+1} = \alpha_0 + \alpha_1 h, x_{n-i} = \alpha_0 - \alpha_1(ih), f(x_{n-i}, t_{n-i}) = \alpha_1. \quad (5)$$

Підставимо рівняння (5) в (1) та отримаємо

$$\alpha_0 + \alpha_1 h = \sum_{i=0}^p a_i (\alpha_0 - \alpha_1(ih)) + h \sum_{i=-1}^p b_i \alpha_1. \quad (6)$$

Враховуючи, умову (4) та поділивши ліву та праву частини на  $\alpha_1 h$ , отримаємо

$$\sum_{i=0}^p (-i) a_i + \sum_{i=-1}^p b_i = 1. \quad (7)$$

3) Нехай  $k = 2$  (поліном другого степеня). Розв'язок задачі Коші у цьому випадку буде  $\hat{x}(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2$ , а сама задача Коші задається у вигляді  $\dot{x} = f(x, t) = 2\alpha_2 t + \alpha_1$ . Нехай  $t_n = 0$ . Тоді  $t_{n+1} = h$ ,  $t_{n-1} = -h$ ,  $t_{n-2} = -2h$ ,  $t_{n-p} = -ph$ . Умова коректності матиме наступний вигляд:

$$x_{n+1} = \alpha_0 + \alpha_1 h + \alpha_2 h^2, x_{n-i} = \alpha_0 - \alpha_1(ih) + \alpha_2(ih)^2, f(x_{n-i}, t_{n-i}) = 2\alpha_2(-ih) + \alpha_1. \quad (8)$$

Підставивши (8) в (1) отримаємо

$$\alpha_0 + \alpha_1 h + \alpha_2 h^2 = \sum_{i=0}^p a_i(\alpha_0 - \alpha_1(ih) + \alpha_2(ih)^2) + h \sum_{i=-1}^p b_i(2\alpha_2(-ih) + \alpha_1). \quad (9)$$

Врахуємо умови коректності (4) та (7) і поділимо обидві частини на  $\alpha_2 h^2$

$$\sum_{i=1}^p i^2 a_i + 2 \sum_{i=-1}^p (-i) b_i = 1. \quad (10)$$

В загальному випадку, коли задачі Коші з розв'язком

$\hat{x}(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \dots + \alpha_k t^k$  задаються у вигляді

$\dot{x} = k\alpha_k t^{k-1} + (k-1)\alpha_{k-1} t^{k-2} + \dots + 2\alpha_2 t + \alpha_1$  умови коректності матимуть

такий вигляд [1, 3-4]:

$$\sum_{i=1}^p (-i)^k a_i + k \sum_{i=-1}^p (-i)^{k-1} b_i = 1. \quad (11)$$

Врахувавши умови коректності (4), (7), (10) та (11), випишемо їх у вигляді системи

$$\begin{cases} \sum_{i=0}^p a_i = 1, \\ \sum_{i=1}^p (-i)^j a_i + j \sum_{i=-1}^p (-i)^{j-1} b_i = 1, j = 1, 2, \dots, k. \end{cases} \quad (12)$$

Всі коефіцієнти різницевої схеми (1) мають задовільняти рівняння у системі (12).

## 1.2 Методи Адамса-Башфорта

Нехай

$$p = k - 1, a_1 = a_2 = \dots = a_{k-1} = 0, b_{-1} = 0. \quad (13)$$

Підставивши (13) в рівняння (1), отримаємо різницеву схему Адамса-Башфорта  $k$ -того порядку [1-2]

$$x_{n+1} = a_0 x_n + h(b_0 f(x_n, t_n) + b_1 f(x_{n-1}, t_{n-1}) + \dots + b_{k-1} f(x_{n-k+1}, t_{n-k+1})). \quad (14)$$

Зрозуміло, що всі  $(k+1)$  коефіцієнти повинні бути підібрані таким чином, щоб схема (13) була точною для всіх поліноміальних розв'язків степеня  $k$ . Також врахуємо умови коректності (12) і замінимо коефіцієнти  $b_i$ ,  $i = 0, 2, \dots, k-1$  на відповідні функції  $\beta_i(k)$ ,  $i = 0, 2, \dots, k-1$  і отримаємо



$$x_{n+1} = x_n + h \sum_{i=0}^{k-1} \beta_i(k) f(x_{n-i}, t_{n-i}). \quad (15)$$

Щоб знайти невідомі  $\beta_i(k)$ ,  $i = 0, 2, \dots, k-1$  підставимо  $a_0 = 1$ ,  $b_i = \beta_i(k)$  і коефіцієнти (13) в рівняння (12) та отримаємо

$$\sum_{i=0}^{k-1} (-i)^{j-1} \beta_j(k) = \frac{1}{j}, j = 1, 2, \dots, k. \quad (16)$$

Співвідношення (16) - це система лінійних алгебраїчних рівнянь із  $k$  рівнянь та  $k$  невідомих, яку ми можемо записати у матричному вигляді.

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & -1 & -2 & \dots & -(k-1) \\ 0 & 1 & 4 & \dots & (k-1)^2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & (-1)^{k-1} & (-2)^{k-1} & \dots & (-(k-1))^{k-1} \end{pmatrix} \begin{pmatrix} \beta_0(k) \\ \beta_1(k) \\ \cdot \\ \cdot \\ \beta_{k-1}(k) \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \cdot \\ \cdot \\ \frac{1}{k} \end{pmatrix}. \quad (17)$$

Визначник системи відмінний від нуля [1,4], отже (17) має єдиний розв'язок.

1) Нехай  $k = 1$ . Тоді розв'язком рівняння (16) буде  $\beta_0(1) = 1$  і підставивши розв'язок в (15), отримаємо різницеву схему Адамса-Башфорта першого порядку

$$x_{n+1} = x_n + hf(x_n, t_n).$$

Як бачимо, різницева схема Адамса-Башфорта першого порядку це явний метод Ейлера.

2) Нехай  $k = 2$ . Тоді ми отримаємо систему із двох рівнянь:

$$\begin{cases} \beta_0(2) + \beta_1(2) = 1, \\ -\beta_1(2) = \frac{1}{2}. \end{cases}$$

Розв'язок даної системи

$$\begin{cases} \beta_0(2) = \frac{3}{2}, \\ \beta_1(2) = -\frac{1}{2}. \end{cases}$$

Підставимо розв'язок в (15) і отримаємо різницеву схему Адамса-Башфорта другого порядку

$$x_{n+1} = x_n + h\left(\frac{3}{2}f(x_n, t_n) - \frac{1}{2}f(x_{n-1}, t_{n-1})\right).$$

3) Нехай  $k = 3$ . Тоді одержимо систему із трьох рівнянь

$$\begin{cases} \beta_0(3) + \beta_1(3) + \beta_2(3) = 1, \\ -\beta_1(3) - 2\beta_2(3) = \frac{1}{2}, \\ \beta_1(3) + 4\beta_2(3) = \frac{1}{3}. \end{cases}$$

Розв'язок даної системи

$$\begin{cases} \beta_0(3) = \frac{23}{12}, \\ \beta_1(3) = \frac{16}{12}, \\ \beta_2(3) = -\frac{5}{12}. \end{cases}$$

Підставимо розв'язок в (15) і отримаємо різницеву схему Адамса-Башфорта третього порядку

$$x_{n+1} = x_n + h\left(\frac{23}{12}f(x_n, t_n) - \frac{16}{12}f(x_{n-1}, t_{n-1}) + \frac{5}{12}f(x_{n-2}, t_{n-2})\right).$$

У даній роботі [10] розроблена прикладна програма, яка дозволяє знайти коефіцієнти різницевих схем Адамса-Башфорта досить високого порядку. Таблиця цих коефіцієнтів наведена у додатку 2.

### 1.3 Методи Адамса-Маултона

Нехай

$$p = k - 2, a_1 = a_2 = \dots = a_{k-2} = 0. \quad (18)$$

Підставивши (18) в (1) отримаємо різницеву схему Адамса-Маултона [1-2]

$$x_{n+1} = a_0 x_n + h(b_{-1}f(x_{n+1}, t_{n+1}) + b_0 f(x_n, t_n) + \dots + b_{k-2} f(x_{n-k+2}, t_{n-k+2})) \quad (19)$$

Знову ж  $(k+1)$  коефіцієнтів повинні бути такими, щоб рівняння (19) було точним для всіх поліноміальних розв'язків степеня  $k$ . Врахувавши умови коректності (12) і замінивши коефіцієнти  $b_i$  на функції  $\mu_i(k)$ ,  $i = -1, 0, \dots, k-2$ , запишемо рівняння (19) у такому вигляді

$$x_{n+1} = x_n + h \sum_{i=-1}^{k-2} \mu_i(k) f(x_{n-i}, t_{n-i}). \quad (20)$$

Щоб визначити  $\mu_i(k)$ ,  $i = -1, 0, \dots, k-2$  підставимо  $a_0 = 1$ ,  $b_i = \mu_i(k)$  і коефіцієнти (18) в рівняння (12) та отримаємо

$$\sum_{i=-1}^{k-2} (-i)^{j-1} \mu_i(k) = \frac{1}{j}, j = 1, 2, \dots, k. \quad (21)$$

Співвідношення (21) - це система лінійних алгебраїчних рівнянь із  $k$  рівнянь та

$k$  невідомих, яку ми можемо записати у матричному вигляді.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & -1 & -2 & \dots & -(k-2) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & (-1)^{k-1} & (-2)^{k-1} & \dots & -(k-2)^{k-1} \end{pmatrix} \begin{pmatrix} \mu_{-1}(k) \\ \mu_0(k) \\ \vdots \\ \mu_{k-2}(k) \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \vdots \\ \frac{1}{k} \end{pmatrix}. \quad (22)$$

Визначник системи відмінний від нуля [1-3], отже (22) має єдиний розв'язок

1) Нехай  $k = 1$ . Тоді розв'язком рівняння (21) буде  $\mu_{-1}(1) = 1$  і підставивши розв'язок в (20), отримаємо різницеву схему Адамса-Маултона першого порядку

$$x_{n+1} = x_n + hf(x_{n+1}, t_{n+1}).$$

Як бачимо, різницєва схема Адамса-Маултона першого порядку це неявний метод Ейлера.

2) Нехай  $k = 2$ . Тоді ми отримаємо систему із двох рівнянь:

$$\begin{cases} \mu_{-1}(2) + \mu_0(2) = 1, \\ \mu_{-1}(2) = \frac{1}{2}. \end{cases}$$

Розв'язок даної системи

$$\begin{cases} \mu_{-1}(2) = \frac{1}{2}, \\ \mu_0(2) = \frac{1}{2}. \end{cases}$$

Підставимо розв'язок в (20) і отримаємо різницєву схему Адамса-Маултона другого порядку

$$x_{n+1} = x_n + h\left(\frac{1}{2}f(x_{n+1}, t_{n+1}) + \frac{1}{2}f(x_n, t_n)\right).$$

3) Нехай  $k = 3$ . Тоді одержимо систему із трьох рівнянь

$$\begin{cases} \mu_{-1}(3) + \mu_0(3) + \mu_1(3) = 1, \\ \mu_{-1}(3) - \mu_1(3) = \frac{1}{2}, \\ \mu_{-1}(3) + \mu_1(3) = \frac{1}{3}. \end{cases}$$

Розв'язок даної системи

$$\begin{cases} \mu_{-1}(3) = \frac{5}{12}, \\ \mu_0(3) = \frac{8}{12}, \\ \mu_1(3) = -\frac{1}{12}. \end{cases}$$

Підставимо розв'язок в (20) і отримаємо різницеву схему Адамса-Маултона третього порядку

$$x_{n+1} = x_n + h\left(\frac{5}{12}f(x_{n+1}, t_{n+1}) + \frac{8}{12}f(x_n, t_n) - \frac{1}{12}f(x_{n-1}, t_{n-1})\right).$$

У даній роботі розроблена прикладна програма [10], яка дозволяє знайти коефіцієнти різницевих схем Адамса-Маултона досить високого порядку.

Таблиця цих коефіцієнтів наведена у додатку 2.

## 2. Стійкість багатокрокових різницевих схем

### 2.1 Необхідна умова стійкості

Для дослідження стійкості багатокрокових різницевих схем, як правило, беруть модельний приклад задачі Коші

$$\dot{x} = -\lambda x, x(0) = 1, \lambda = \text{const} > 0. \quad (23)$$

Розв'язок задачі (23) має вигляд  $x(t) = x_0 e^{-\lambda t}$ . При цьому  $\lim_{t \rightarrow \infty} u(t) = 0$ . Оскільки поведінка розв'язку різницевої схеми повинна відповідати поведінці розв'язку апроксимуючого рівняння, то для нього повинна виконуватись аналогічна властивість.

Застосуємо різницеву схему (1) до задачі (23) і одержимо наступне:

$$x_{n+1} = \sum_{i=0}^p a_i x_{n-i} + h \sum_{i=-1}^p b_i (-\lambda x_{n-i}). \quad (24)$$

Замінивши  $\lambda h$  на  $\sigma$  перепишемо (24) у наступному вигляді:

$$(1 + \sigma b_{-1})x_{n+1} - (a_0 - \sigma b_0)x_n - \dots - (a_p - \sigma b_p)x_{n-p} = 0. \quad (25)$$

Розв'язком рівняння (25) буде

$$x_n = c_1 \rho_1^n + c_2 \rho_2^n + \dots + c_{p+1} \rho_{p+1}^n, \quad (26)$$

де  $\rho_i, i = 1, 2, \dots, p + 1$  - це корені характеристичного поліноміального рівняння

$$P(\rho) = (1 + \sigma b_{-1})\rho^{p+1} - (a_0 - \sigma b_0)\rho^p - \dots - (a_p - \sigma b_p) = 0. \quad (27)$$

Зауважимо, що у випадку, коли корінь  $\rho_i$  є коренем кратності  $m > 1$ , то йому відповідає частинний розв'язок різницевого рівняння (25) вигляду

$$(c_0 + c_1 n + c_2 n^2 + \dots + c_{m-1} n^{m-1}) \rho_i^n.$$

Отже, якщо серед коренів характеристичного рівняння (25) існують корені  $|\rho_i| > 1$  або  $|\rho_i| = 1$ , але їх кратність більша за одиницю, то розв'язок різницевого рівняння (24)  $|x_n| \rightarrow \infty$  при  $n \rightarrow \infty$ . Це означає, що різницєва схема (1) є нестійкою.

Таким чином маємо таке твердження.

**Теорема 1[1] (Необхідна умова стійкості).**

*Для того щоб багатокрокова різницева схема (1) була стійкою необхідно, щоб корені характеристичного рівняння (25) задовольняли умову  $|\rho_i| \leq 1$ , а серед тих, що  $|\rho_i| = 1$  не було кратних.*

**2.2 Критерій стійкості багатокрокових різницевих схем [1]**

Багатокрокова різницева схема при виконанні умови  $\sum_{i=0}^p b_i \neq 0$  стійка в тому розумінні, що її локальна похибка є обмеженою при достатньо малому кроці  $h$  тоді і тільки тоді, коли всі корені характеристичного рівняння (25) задовольняють умову  $|\rho_i| < 1$  за винятком того кореня, що при  $\lambda h = 0$  перетворюється в 1.

### 3. Абсолютна стійкість багатокрокових різницевих схем

#### 3.1 Алгоритм побудови області абсолютної стійкості

Стійкість різницевої схеми забезпечує обмеженість локальних похибок при достатньо малому кроці  $h$ . Однак, якщо крок  $h$  малий, то потрібно велике число кроків, щоб знайти розв'язок задачі на заданому інтервалі, що збільшує похибки округлень. Крім того, перевірка умов стійкості ефективна тільки при  $h \rightarrow 0$ . Розглянемо дещо інше означення стійкості, яке при  $h \neq 0$  можна автоматизувати на комп'ютері.

#### Означення (Абсолютна стійкість) [1, 3].

Багатокрокова різницева схема (1) називається абсолютно стійкою для тих значень  $\sigma = \lambda h$ , для яких усі корені характеристичного рівняння (25) лежать всередині одиничного кола  $|\rho| \leq 1$ .

Множина значень  $\sigma = \lambda h$  для яких різницева схема (1) є абсолютно стійкою називається областю абсолютної стійкості. Так як  $\lambda$ , як правило, комплексне число, то область абсолютної стійкості багатокрокової різницевої схеми представляє область комплексної  $\sigma$ -площини.

#### Алгоритм побудови області абсолютної стійкості

1) Застосувати відповідну різницеву схему до тестової модельної задачі та дістати відповідне характеристичне рівняння і виразити основне співвідношення для  $\sigma$ .

Перепишемо рівняння (27) наступним чином:

$$P(\rho) = (\rho^{p+1} - a_0\rho^p - \dots - a_p) + \sigma(b_{-1}\rho^{p+1} + b_0\rho^p + \dots + b_p) = 0. \quad (28)$$

Нехай  $P_a(\rho) = (\rho^{p+1} - a_0\rho^p - \dots - a_p)$ ,  $P_b(\rho) = (b_{-1}\rho^{p+1} + b_0\rho^p + \dots + b_p)$ , тоді (28) можна зобразити у вигляді:

$$\sigma = \frac{-P_a(\rho)}{P_b(\rho)}. \quad (29)$$

Співвідношення (29) задає в площині  $\sigma$  певну криву  $\Gamma$  (замкнене коло  $|\rho| = 1$  переходить в замкнену криву  $\Gamma$ ), яка є симетричною відносно осі  $\text{Re } \sigma$  [1].

2) Підставити замість  $\rho^p = e^{pi\theta}$  в (29) і одержати наступне співвідношення

$$\begin{aligned} \sigma(\theta) &= \frac{-e^{i(p+1)\theta} + a_0 e^{ip\theta} + \dots + a_{p-1} e^{i\theta} + a_p}{b_{-1} e^{i(p+1)\theta} + b_0 e^{ip\theta} + \dots + b_{p-1} e^{i\theta} + b_p} = \frac{\varphi_1(\theta) + i\varphi_2(\theta)}{\varphi_3(\theta) + i\varphi_4(\theta)} = \\ &= -\frac{\varphi_1(\theta)\varphi_3(\theta) - \varphi_2(\theta)\varphi_4(\theta)}{\varphi_3^2(\theta) + \varphi_4^2(\theta)} - i\frac{\varphi_3(\theta)\varphi_4(\theta) + \varphi_1(\theta)\varphi_2(\theta)}{\varphi_3^2(\theta) + \varphi_4^2(\theta)} = \operatorname{Re}\sigma + i\operatorname{Im}\sigma. \end{aligned} \quad (30)$$

3) За співвідношенням (30) побудувати таблицю значень функцій  $\operatorname{Re} \sigma$  та  $\operatorname{Im} \sigma$

$\sigma$  для  $\theta_j = \frac{2\pi}{m}j, j = \overline{0, m}, m \in \mathbb{N}$ .

$\Theta$	$\theta_0$	$\theta_1$	$\theta_2$	...	$\theta_m$
$\operatorname{Re} \sigma = u$	$u_0$	$u_1$	$u_2$	...	$u_m$
$\operatorname{Im} \sigma = v$	$v_0$	$v_1$	$v_2$	...	$v_m$

4) Побудувати криву  $\Gamma$  за точками  $(u_j, v_j), j = \overline{0, m}$ : у площині  $\sigma$  потрібно відставити точки  $u_j + iv_j, j = \overline{0, m}$  і послідовно з'єднати прямими.

5) Побудувати область абсолютної стійкості: вибираємо  $|\rho_0| < 1$ , знайти відповідне  $\sigma_0 = \sigma(\rho_0)$  та замалювати ту частину  $\sigma$ -площини куди попадає точка  $\sigma_0$ .



### 3.2 Области абсолютної стійкості різницевих схем Адамса-Башфорта та Адамса-Маултона

#### А) Різницеві схеми Адамса-Башфорта

Розглянемо метод Адамса Башфорта першого порядку

$$x_{n+1} = x_n + hf(x_n, t_n).$$

Врахувавши, що  $a_0 = 1$ ,  $b_0 = 1$  та коефіцієнти для різницевої схеми Адамса-Башфорта (13), підставимо їх в рівняння (30) і отримаємо

$$\sigma(\theta) = \frac{-e^{i\theta} + 1}{1} = -e^{i\theta} + 1 = -\cos(\theta) - i\sin(\theta) + 1, 0 \leq \theta \leq 2\pi.$$

$\text{Re } \sigma = -\cos(\theta) + 1$ ,  $\text{Im } \sigma = -\sin(\theta)$ . Оскільки  $(\text{Re } \sigma - 1)^2 + (\text{Im } \sigma)^2 = 1$ , то дане співвідношення задає криву  $\Gamma$ , яка є одиничним колом, центр якого розташований в точці  $\sigma = 1 + 0 \cdot i$  комплексної  $\sigma$ -площини. абсолютної стійкості буде всередині кола так, як зображено на рис. 1.

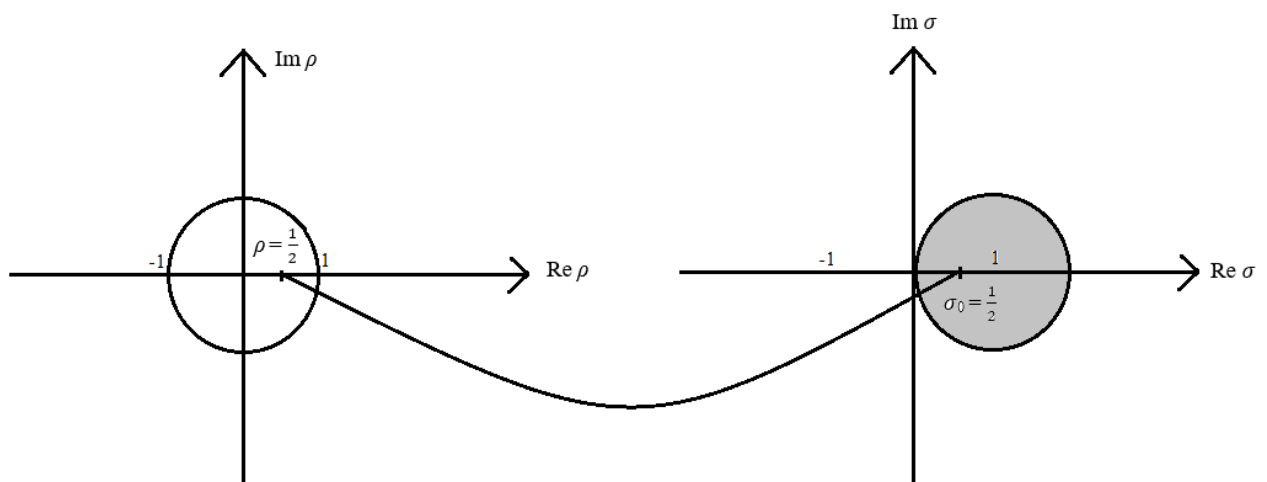


Рис. 1

Врахуємо коефіцієнти (13) та те, що  $a_0 = 1$  та запишемо співвідношення (30) для різницевої схеми Адамса-Башфорта  $k$ -того порядку

$$\sigma = \frac{-e^{ik\theta} + e^{i(k-1)\theta}}{b_0 e^{i(k-1)\theta} + b_1 e^{i(k-2)\theta} + \dots + b_{k-2} e^{i\theta} + b_{k-1}}.$$

Замінивши  $e^{ik\theta}$  на  $\cos(k\theta) + i \sin(k\theta)$  отримаємо наступні співвідношення для  $\varphi_1(\theta)$ ,  $\varphi_2(\theta)$ ,  $\varphi_3(\theta)$ ,  $\varphi_4(\theta)$ :

$$\varphi_1(\theta) = \cos((k-1)\theta) - \cos(k\theta),$$

$$\varphi_2(\theta) = \sin((k-1)\theta) - \sin(k\theta),$$

$$\varphi_3(\theta) = b_0 \cos((k-1)\theta) + b_1 \cos((k-2)\theta) + \dots + b_{k-2} \cos(\theta) + b_{k-1},$$

$$\varphi_4(\theta) = b_0 \sin((k-1)\theta) + b_1 \sin((k-2)\theta) + \dots + b_{k-2} \sin(\theta).$$

Для знаходження точки  $\sigma_0$  зробимо заміну  $e^{ik\theta} = \rho^k$  і отримаємо:

$$\sigma_0 = \sigma(\rho_0) = \frac{-\rho_0^k + \rho_0^{(k-1)}}{b_0 \rho_0^{(k-1)} + b_1 \rho_0^{(k-2)} + \dots + b_{k-2} \rho_0 + b_{k-1}}.$$

## Б) Різницеві схеми Адамса-Маултона

Метод Адамса-Маултона першого порядку

$$x_{n+1} = x_n + hf(x_{n+1}, t_{n+1}).$$

Врахувавши, що  $a_0 = 1$ ,  $b_{-1} = 1$  та коефіцієнти для різницевої схеми Адамса-Маултона (18), підставимо їх в рівняння (30) і отримаємо

$$\sigma(\theta) = \frac{e^{-i\theta} - 1}{1} = e^{-i\theta} - 1 = \cos(-\theta) + i\sin(-\theta) - 1 = \cos(\theta) - i\sin(\theta) - 1, \quad 0 \leq \theta \leq 2\pi.$$

$\text{Re } \sigma = \cos(\theta) - 1$ ,  $\text{Im } \sigma = -\sin(\theta)$ . Оскільки  $(\text{Re } \sigma + 1)^2 + (\text{Im } \sigma)^2 = 1$ , то дане співвідношення задає криву  $\Gamma$ , яка є одиничним колом, центр якого розташований в точці  $\sigma = -1 + 0 \cdot i$  комплексної  $\sigma$ -площини.

Оскільки  $e^{i\theta} = \rho$ , то  $\sigma(\theta) = \rho - 1$ . Нехай  $\rho_0 = \frac{1}{2}$ , тоді  $\sigma_0 = 1 - \frac{1}{2} = \frac{1}{2}$ . Отже, область абсолютної стійкості буде зовні кола так, як зображено на рис. 2.

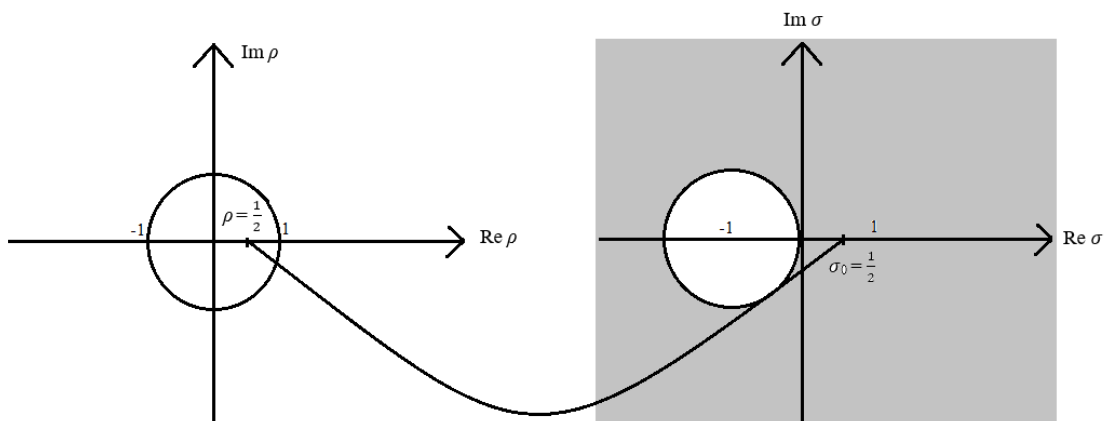


Рис. 2

Врахуємо коефіцієнти (18) та те, що  $a_0 = 1$  та запишемо співвідношення (30) для різницевої схеми Адамса-Маултона  $k$ -того порядку

$$\sigma = \frac{-e^{i(k-1)\theta} + e^{i(k-2)\theta}}{b_{-1}e^{i(k-1)\theta} + b_0e^{i(k-2)\theta} + \dots + b_{k-3}e^{i\theta} + b_{k-2}}$$

Замінивши  $e^{ik\theta}$  на  $\cos(k\theta) + i \sin(k\theta)$  отримаємо наступні співвідношення для  $\varphi_1(\theta), \varphi_2(\theta), \varphi_3(\theta), \varphi_4(\theta)$ :

$$\varphi_1(\theta) = \cos((k-2)\theta) - \cos((k-1)\theta),$$

$$\varphi_2(\theta) = \sin((k-2)\theta) - \sin((k-1)\theta),$$

$$\varphi_3(\theta) = b_{-1}\cos((k-1)\theta) + b_0\cos((k-2)\theta) + \dots + b_{k-3}\cos(\theta) + b_{k-2},$$

$$\varphi_4(\theta) = b_{-1}\sin((k-1)\theta) + b_0\sin((k-2)\theta) + \dots + b_{k-3}\sin(\theta).$$

Для знаходження точки  $\sigma_0$  зробимо заміну  $e^{ik\theta} = \rho^k$  і отримаємо:

$$\sigma_0 = \sigma(\rho_0) = \frac{-\rho_0^{(k-1)} + \rho_0^{(k-2)}}{b_{-1}\rho^{(k-1)} + b_0\rho^{(k-2)} + \dots + b_{k-3}\rho + b_{k-2}}$$

В роботі розроблена прикладна програма, яка за допомогою вище наведених формул та алгоритму, будує області абсолютної стійкості. На рисунках 3 та 4 відповідно наведено області абсолютної стійкості різницевих схем Адамса-Башфорта та Адамса-Маултона до шостого порядку [10]

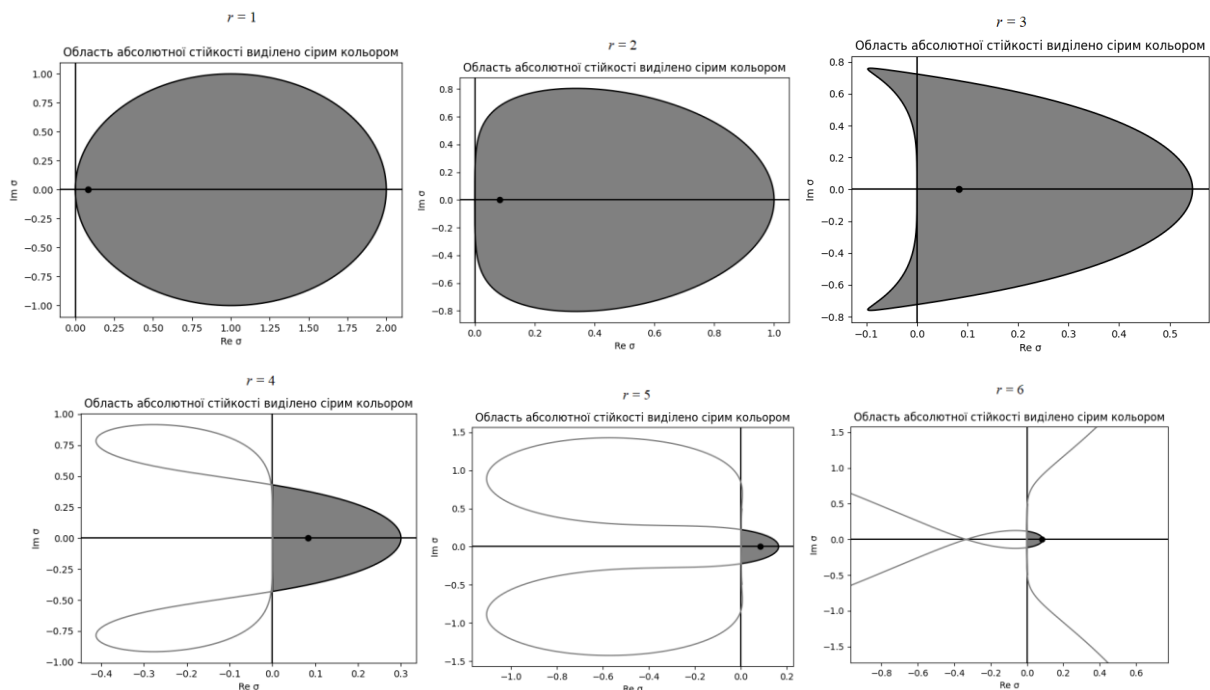


Рис. 3

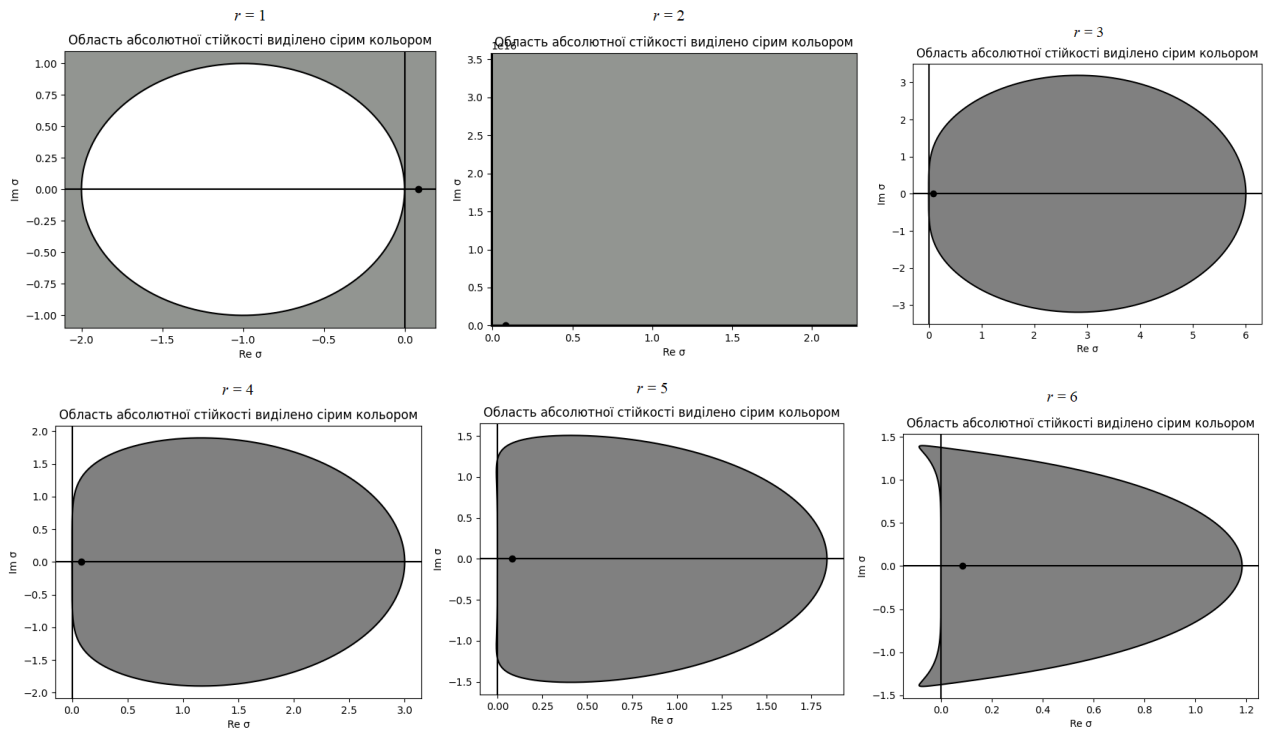


Рис. 4

## 4. Жорстко стійкі різницеві схеми

### 4.1 Жорсткі системи диференціальних рівнянь

Системи диференціальних рівнянь є математичними моделями багатьох прикладних задач у різних галузях науки. Знайти точний розв'язок для таких систем не завжди просто, тому наближені методи для побудови їх розв'язків є вкрай важливими.

Проте, для деяких типів диференціальних задач класичні різницеві схеми не працюють. Такі задачі називають жорсткими [5-7]. Розглянемо поняття жорсткості на системі лінійних диференціальних рівнянь зі сталими коефіцієнтами

$$\dot{u} = Au, \quad (31)$$

де  $A$  –  $n \times n$  стала матриця,  $u \in \mathbb{R}^n$ ,  $\lambda_i$ ,  $i = \overline{1, n}$  – власні значення матриці  $A$ .

**Означення [4-5].** Система диференціальних рівнянь (31) називається жорсткою, якщо виконуються такі умови:

- 1)  $\operatorname{Re} \lambda_i < 0$ ,  $i = \overline{1, n}$ ,
- 2)  $k = \frac{\min \operatorname{Re} \lambda_i}{\max \operatorname{Re} \lambda_i} \gg 1$ .

Величина  $k$  називається коефіцієнтом жорсткості. Система вже вважається жорсткою, якщо  $k > 10$ . При  $10 < k < 100$  жорстка система називається м'якою, а при  $k > 100$  – твердою.

**Приклад [1].** Розглянемо задачу Коші

$$\begin{aligned} u' &= 998u + 1998v, & u(0) &= 1, \\ v' &= -999u - 1999v, & v(0) &= 1. \end{aligned} \quad (32)$$

Точний розв'язок має вигляд:

$$\begin{aligned} u_T &= 4e^{-t} - 3e^{-1000t}, \\ v_T &= -2e^{-t} + 3e^{-1000t}. \end{aligned}$$

У даному випадку:  $A = \begin{pmatrix} 998 & 1998 \\ -999 & -1999 \end{pmatrix}$ ,  $\lambda_1 = -1$ ,  $\lambda_2 = -1000$ .

Отже,  $k = \frac{1000}{1} = 1000$ , це означає, що задача (32) є твердою жорсткою задачею. Застосувавши для її розв'язання класичні різницеві схеми Рунге-Кути, за кілька кроків обчислення одержимо розбіжний ітераційний процес. Це пов'язано з тим, що розв'язок містить швидко змінну складову  $e^{-1000t}$ , яка істотна тільки в малому околі початкової точки  $t = 0$ . Для врахування цієї складової потрібно вибрати дуже малий крок інтегрування і його не можна збільшити в області, де ця складова вже неістотна.

## 4.2 Жорстко стійкі різницеві схеми

Розглядаючи жорстку систему диференціальних рівнянь, потрібно врахувати, що її розв'язок містить «дуже швидку» та «повільну» складову. Для лінійних систем це проявляється, коли власні значення матриці системи сильно відрізняються.

Для ефективного розв'язання жорстких задач потрібно вибрати таку різницеву схему, яка дозволяє змінювати крок інтегрування, зберігаючи при цьому стійкість.

Для врахування швидкої складової розв'язку потрібно починати інтегрування з достатньо малого кроку  $h$ , а потім крок потрібно збільшувати як тільки швидка складова стане неістотною, зберігаючи при цьому стійкість різницевої схеми. Такий підхід дозволяє визначити необхідну область абсолютної стійкості в комплексній  $\sigma = \lambda h$  площині різницевої схеми, яку можна застосувати для розв'язання жорстких задач.

Розглянемо частину  $\sigma$  площини, що визначається співвідношенням

$$\operatorname{Re} \sigma \geq \delta > 0,$$

де  $\delta$  таке дійсне число, що швидка складова при  $t \geq \delta$  стає неістотною. Тут крок потрібно задавати достатньо великий, зберігаючи при цьому стійкість.

В області

$$0 < \operatorname{Re} \sigma < \delta$$

швидку компоненту потрібно враховувати, використовуючи достатньо малий крок  $h$ . При цьому наявність синусоїдальних складових можна обмежити співвідношенням [1]

$$|\operatorname{Im} \sigma| < \theta, \quad \theta < \frac{\pi}{4}.$$

При  $-\mu < \operatorname{Re} \sigma < 0$  розв'язок жорсткої задачі складається із зростаючих компонент, тому для їх врахування необхідний малий крок на невеликому часовому проміжку. Зазначені вимоги до різницевої схеми представлені на наступному рисунку:

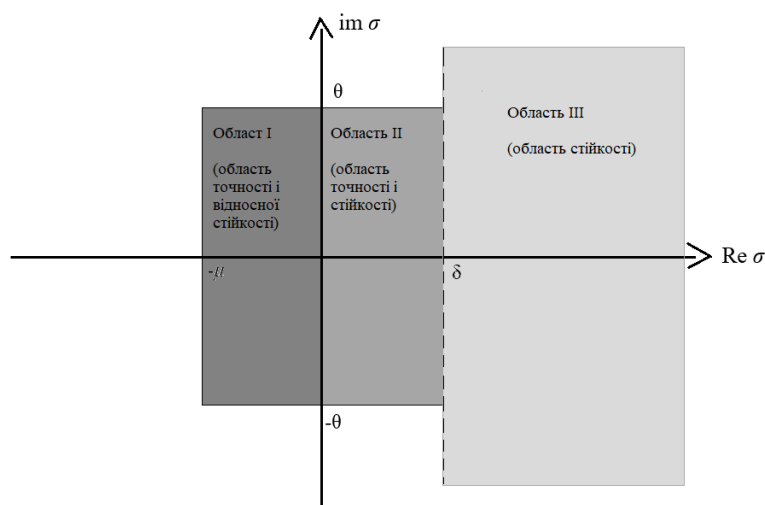


Рис. 5. Область абсолютної стійкості жорстко стійкої різницевої схеми

Багатокрокова різницева схема, що володіє абсолютною областю стійкості, яка зображена на рис. 5 називається жорстко стійкою.

Аналізуючи області абсолютної стійкості різницевих схем Адамса-Башфорта та Адамса-Маултона, які зображені на рисунках 3 та 4 відповідно, можемо зробити висновок, що серед явних різницевих схем Адамса-Башфорта немає жорстко стійких, а серед неявних різницевих схем Адамса-Маултона жорстко стійкими є лише різницеві схеми першого та другого порядків.

**Теорема 3 [8] (Далквіста).** *Різницева схема, яка абсолютно стійка в області  $\operatorname{Re} \sigma \geq 0$  не може мати порядок вище 2.*

Теорема Далквіста виключає існування жорстко стійких різницевих схем високих порядків з  $\delta > 0$ .

## 5. Різницеві схеми Гіра

### 5.1 Існування різницевих схем Гіра [1, 6]

Аналізуючи необхідну область стійкості жорстко стійкої різницевої схеми, бачимо, що вона повинна бути стійкою при  $\delta > 0$  і  $\sigma = \lambda h \rightarrow \infty$ . Розглянемо характеристичне рівняння багатокрокової різницевої схеми (1) у такому вигляді (28)

$$P(\rho) = (\rho^{p+1} - a_0\rho^p - \dots - a_p) + \sigma(b_{-1}\rho^{p+1} + b_0\rho^p + \dots + b_p) = 0.$$

Оскільки різницєва є стійкою при  $\sigma \rightarrow \infty$ , то другий доданок у рівності (28) є домінуючим, тобто

$$P(\rho) \approx \sigma(b_{-1}\rho^{p+1} + b_0\rho^p + \dots + b_0), \quad \sigma \rightarrow \infty.$$

Необхідно забезпечити стійкість різницевої схеми, тобто всі корені характеристичного рівняння (28) повинні задовольняти умову

$$|\rho_i| \leq 1. \quad (33)$$

Цього можна досягти, якщо  $b_0 = b_1 = \dots = b_p = 0$ . В цьому випадку

$$P(\rho) \approx \sigma b_{-1}\rho^{p+1}$$

і є єдиним коренем поліноміального рівняння буде  $\rho = 0$  кратності  $r$ , що задовольняє умову (33).

Така ідея побудови різницєвих схем була запропонована С. Гіром [4], тому називатимемо їх різницєвими схемами Гіра, хоча в роботі [3] їх називають методами формули диференціювання назад (ФДН).

Нехай

$$p = k - 1, \quad b_0 = b_1 = \dots = b_{k-1} = 0. \quad (34)$$

Тоді підставивши ці коефіцієнти в (1) ми отримаємо

$$x_{n+1} = a_0(k)x_n + a_1(k)x_{n-1} + \dots + a_{k-1}(k)x_{n-k+1} + h(b_{-1}(k)f(x_{n+1}, t_{n+1})), \quad (35)$$

де  $a_i(k)$ ,  $b_{-1}(k)$  – це коефіцієнти  $a_i$ ,  $i = 0, 1, \dots, k - 1$  та  $b_{-1}$  відповідно. Ці  $k+1$  коефіцієнти повинні бути такими, що різницєва схема (35) була точною для всіх



поліноміальних розв'язків степеня  $k$ . Щоб визначити коефіцієнти  $a_i(k)$ ,  $b_{-1}(k)$ , підставимо коефіцієнти (34) в умову коректності (12) і отримаємо

$$\sum_{i=0}^{k-1} (-i)^j a_i(k) + j b_{-1}(k) = 1, j = 0, 1, 2, \dots, k. \quad (36)$$

Співвідношення (36) - це система лінійних алгебраїчних рівнянь із  $k+1$  невідомими та  $k+1$  рівнянням, яка у матричному вигляді виглядатиме наступним чином:

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 0 \\ 0 & -1 & -2 & \dots & (k-1) & 1 \\ 0 & 1 & 4 & \dots & (k-1)^2 & 2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & (-1)^k (-2)^k & \dots & \dots & -(k-1)^k & k \end{pmatrix} \begin{pmatrix} a_0(k) \\ a_1(k) \\ a_2(k) \\ \vdots \\ \vdots \\ b_{-1}(k) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ \vdots \\ 1 \end{pmatrix}. \quad (37)$$

Визначник системи (37) відмінний від нуля [1], тому вона має розв'язок.

1) Нехай  $k = 1$ , тоді розв'язком (36) буде:

$$\begin{cases} a_0(1) = 1, \\ b_{-1}(1) = 1. \end{cases}$$

Підставивши розв'язок в (35) отримаємо різницеву схему Гіра першого порядку

$$x_{n+1} = x_n + h(f(x_{n+1}, t_{n+1})).$$

Як бачимо, дана різницєва схема співпадає з різницєвою схемою Адамса-Маултона першого порядку та неявним методом Ейлера.

2) Нехай  $k = 2$ , тоді ми отримаємо систему із трьох рівнянь

$$\begin{cases} a_0(2) + a_1(2) = 1, \\ -a_1(2) + b_{-1}(2) = 1, \\ a_1(2) + 2b_{-1}(2) = 1. \end{cases}$$

Розв'язком даної системи буде:

$$\begin{cases} a_0(2) = \frac{4}{3}, \\ a_1(2) = -\frac{1}{3}, \\ b_{-1}(2) = \frac{2}{3}. \end{cases}$$

Підставивши розв'язок в (35) отримаємо різницеву схему Гіра другого порядку

$$x_{n+1} = \frac{4}{3}x_n - \frac{1}{3}x_{n-1} + h \left( \frac{2}{3}f(x_{n+1}, t_{n+1}) \right).$$

Різницеві схеми Гіра до шостого порядку наведені в роботах [1, 3], а до 15 порядку в роботі [9] і в додатку 2.

## 5.2 Області абсолютної стійкості різницевих схем Гіра

Оскільки різницєва схема Гіра першого порядку співпадає з різницевою схемою Адамса-Маултона першого порядку, то область абсолютної стійкості для різницевої схеми Гіра першого порядку показано на рисунку 2.

У випадку різницевої схеми Гіра  $k$ -того порядку, врахувавши коефіцієнти (34), запишемо співвідношення (30)

$$\sigma(\theta) = \frac{-e^{ik\theta} + a_0 e^{i(k-1)\theta} + \dots + a_{k-2} e^{i\theta} + a_{k-1}}{b_{-1} e^{ik\theta}}.$$

Замінивши  $e^{ik\theta}$  на  $\cos(k\theta) + i \sin(k\theta)$  отримаємо наступні співвідношення для  $\varphi_1(\theta), \varphi_2(\theta), \varphi_3(\theta), \varphi_4(\theta)$ :

$$\varphi_1(\theta) = -\cos(k\theta) + a_0 \cos((k-1)\theta) + \dots + a_{k-2} \cos(\theta) + a_{k-1},$$

$$\varphi_2(\theta) = -\sin(k\theta) + a_0 \sin((k-1)\theta) + \dots + a_{k-2} \sin(\theta),$$

$$\varphi_3(\theta) = b_{-1} \cos(k\theta),$$

$$\varphi_4(\theta) = b_{-1} \sin(k\theta).$$

Для знаходження точки  $\sigma_0$  зробимо заміну  $e^{ik\theta} = \rho^k$  і отримаємо:

$$\sigma_0 = \sigma(\rho_0) = \frac{-\rho_0^k + a_0 \rho_0^{(k-1)} + \dots + a_{k-2} \rho_0 + a_{k-1}}{b_{-1} \rho_0^k}.$$

Області абсолютної стійкості різницевих схем Гіра 1-6 порядків.

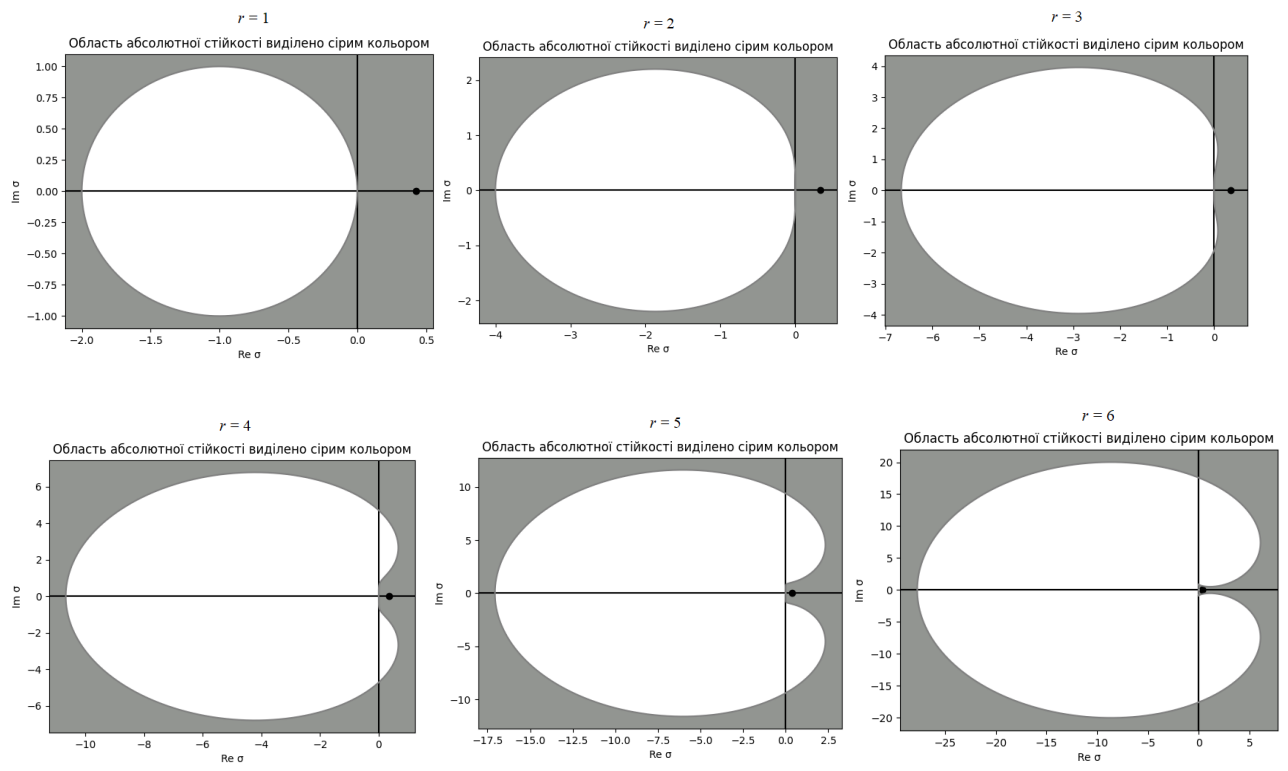


Рис. 6

## **6. Опис використаного програмного забезпечення та розробленої прикладної програми**

### **6.1 Мова програмування Python та її бібліотеки [11]**

Для реалізації алгоритмів знаходження коефіцієнтів різницевих схем Гіра та побудови областей абсолютної стійкості було використано одну з найпопулярніших на даний час мов програмування – Python.

Python – це об'єктно-орієнтована, інтерпретована високорівнена мова програмування з динамічною семантикою. Її високорівневі вбудовані структури даних у поєднанні з динамічною типізацією та динамічним зв'язуванням роблять її дуже привабливою для швидкої розробки додатків, а також для використання в якості мови сценаріїв або склеювання для з'єднання існуючих компонентів. Python підтримує модулі та пакети, що сприяє модульності програм і повторному використанню коду.

NumPy (Numeric Python) – це open-source модуль для Python, який надає загальні математичні і числові операції у вигляді пре-скомпільованих, швидких функцій. Вони об'єднуються у високорівневі пакети. NumPy надає базові методи для маніпулювання з великими масивами і матрицями.

Matplotlib – бібліотека на мові програмування Python для візуалізації даних двовимірною 2D графікою (3D графіка також підтримується). Отримані зображення можуть бути використані як ілюстрації в публікаціях.

Matplotlib є гнучким, легко конфігурованим пакетом, який разом з NumPy, SciPy і IPython надає можливості, подібні до MATLAB. В даний час пакет працює з декількома графічними бібліотеками, включаючи wxWindows і PyGTK.

Пакет підтримує багато видів графіків і діаграм:

- Графіки (line plot)
- Діаграми розсіювання (scatter plot)
- Стовпчасті діаграми (bar chart) і гістограми (histogram)

- Секторні діаграми (pie chart)

## 6.2 Опис розробленої прикладної програми

Розроблена прикладна програма (додаток 1) має наступну структуру:

- **class Fraction**

Клас, який дозволяє працювати із дробовими числами у вигляді звичайного дроби. Клас містить два поля: чисельник та знаменник і перевизначені операції: множення двох звичайних дробів, віднімання двох звичайних дробів, ділення звичайного дроби на ціле число та вивід звичайного дроби на екран. Конструктор класу приймає два аргументи: чисельник дроби та знаменник.

- Функція **build\_matrix(n)**

Дана функція приймає один параметр: порядок різницевої схеми **n** та будує матрицю системи заданого порядку і рядок вільних членів для подальшої обробки

- Функція **make\_zero\_under(A)**

Дана функція приймає один параметр: матрицю **A** і утворює методом Гауса нулі під головною діагоналлю

- Функція **make\_zero\_up(A)**

Дана функція робить аналогічні дії над головною діагоналлю і ділить діагональні елементи і матриці на самих себе. Таким чином після виконання функцій **make\_zero\_under(A)** та **make\_zero\_up(A)** задана матриця перетворюється на одиничну матрицю. Крім того, ті дії, що виконуються над матрицею системи, виконуються і над стовпцем вільних членів. І в результаті, ми одержуємо розв'язок системи, який міститься у стовпчику вільних членів.

- Функція **NSD(a, b)**

Це рекурсивна функція, яка в якості аргументів приймає два числа та шукає найбільший спільний дільник. Він необхідний, що спростувати

дроби, які отримуються в результаті виконання функцій **make\_zero\_under(A)** та **make\_zero\_up(A)**

- Функція **coefficient()**  
Функція виводить коефіцієнти на екран
- Функції **fi1(theta)**, **fi2(theta)**, **fi3(theta)**, **fi4(theta)** потрібні, щоб обчислювати величини для масивів *u* та *v*. В якості аргумента функція приймає кут  $\theta$
- Функція **build\_graphic(ro, n)**  
Функція, яка на основі значень у масивах *u* та *v* будує криву  $\Gamma$ , і за точкою *ro* визначає область абсолютної стійкості різницевої схеми відповідного порядку.

Також у програмі використані ще такі модулі та бібліотеки мови Python:

- Модуль **Decimal**, який необхідний для того щоб із потрібною точністю обчислювати великі числа та уникати заокруглень.
- Бібліотека **math**, з якої використовуються функції  $\cos$ , та  $\sin$  і константа  $\pi$
- Бібліотека **matplotlib**, яка потрібна, щоб відобразити графічно область абсолютної стійкості різницевої схеми.

## Висновки

1. Для багатокрокової різницевої схеми  $k$ -того порядку наведено умову коректності
2. Побудовано загальний вигляд різницевих схем Адамса-Башфорта, Адамса-Маултона та Гіра  $k$ -того порядку, а також одержано систему лінійних алгебраїчних рівнянь, за допомогою якої можна знайти коефіцієнти для різницевої схеми відповідного порядку
3. Розглянуто поняття абсолютної стійкості багатокрокових різницевих схем та проаналізовано необхідний вигляд області абсолютної стійкості для того, щоб багатокрокова різницева схема була жорстко стійкою.
4. Побудовано прикладну програму, що дозволяє знаходити коефіцієнти різницевих схем Адамса-Башфорта, Адамса-Маултона та Гіра досить високого порядку.
5. Реалізовано алгоритм побудови областей абсолютної стійкості різницевих схем Адамса-Башфорта, Адамса-Маултона та Гіра за допомогою прикладної програми. В роботі наведено області абсолютної стійкості різницевих схема до шостого порядку і встановлено, що жорстко стійкими є лише різницеві схеми Адамса-Маултона першого та другого порядків та різницеві схеми Гіра від першого до шостого порядку.

## СПИСОК ЛІТЕРАТУРИ

1. Leon O. Chua, Pen-Min Lin. [Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques](#). Englewood Cliffs, N.J : Prentice Hall, 1975. – 737 с.
2. Бігун Я. Й. Числові методи.– Чернівці: Чернівецький національний університет імені Юрія Федьковича. 2019 – 436 с.
3. Hairer E., Norsett S. P., Wanner G. Solving Ordinary Differential Equations I: Nonstiff and Differential-Algebraic Problems. Berlin: Springer-Verlag, 1993. – 528 с.
4. Butcher, J.C. *Numerical Methods for Ordinary Differential Equations*; John Wiley & Sons: Hoboken, NJ, USA, 2016.- 484 p.
5. Ernst Hairer, Gerhard Wanner Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. Springer, 2010 p. - 614 p.
6. Gear C. W. Numerical initial value problems in ordinary differential equations. – New Jersey : Printice, 1971. – 253 с.
7. Gurtiss C. F., Hirschfelder J. O. Integration of stiff equations. Proc. Nat. Acad. Sci. of U.S. 1952, v.38, p. 235-243.
8. Dahlquist G. 33 years of numerical instability. Part 1. ВІТ. 1985, v.25, p. 188-204.
9. Диренко В. Побудова різницевих схем Гіра та дослідження їх стійкості .  
Матеріали студентської наукової конференції Чернівецького національного університету (12–14 квітня 2022 року). Факультет математики та інформатики. Чернівці : Чернівец. нац. ун-т ім. Ю. Федьковича, 2022. С. 31-32.
10. Диренко Віктор. Побудова та дослідження стійкості різницевих схем Гіра  
// Матеріали міжнародної наукової конференції «Прикладна математика та інформаційні технології», присвяченої 60-річчю кафедри прикладної математики та інформаційних технологій, 22–24 вересня 2022 р. Чернівці: Чернівецький нац. ун-т, 2022. – С. 258-259.



11. Python - Overview [Электронный ресурс]. – Режим доступа до ресурсу:  
<https://docs.python.org/3/tutorial/>

## Додаток 1. Коди розроблених програм

### Різницеві схеми Адамса-Башфорта

```
from decimal import Decimal

from math import cos, sin, pi

import matplotlib.pyplot as plt

class Fraction:

    def __init__(self, a, b):

        self.chis = a

        self.znam = b

    def __mul__(self, ob):

        newob = Fraction(1, 1)

        newob.chis = int(Decimal(self.chis * ob.chis))

        newob.znam = int(Decimal(self.znam * ob.znam))

        return newob

    def __sub__(self, ob):

        newob = Fraction(1, 1)

        newob.chis = int(Decimal(self.chis * ob.znam - self.znam * ob.chis))

        newob.znam = int(Decimal(self.znam * ob.znam))

        return newob

    def __truediv__(self, num):

        newob = Fraction(1, 1)

        newob.chis = int(self.chis / num)

        newob.znam = int(se

lf.znam / num)

        return newob

    def __str__(self):

        return str(self.chis) + "/" + str(self.znam)

def build_matrix(n):

    A = []

    free_row = []

    for j in range(1, n + 1):
```

```

temp = []
for i in range(n):
    temp.append(int(Decimal((-i)**(j-1))))
A.append(temp)
free_row.append(Fraction(1,j))

return A,free_row

def make_zero_under(A):
    n = len(A)
    if(n==1):
        return 1
    elif(n==2):
        return -1
    else:
        i = 2
        j = 1
        k = 2
        m = 1
        while(k <= n-1):
            while(i < n):
                koef = Decimal(A[i][j]/A[i-m][j])
                KOEF = Fraction(koef,1)
                free_row[i] = free_row[i]-KOEF*free_row[i-m]
                nsd = NSD(free_row[i].chis,free_row[i].znam)
                free_row[i] = free_row[i]/nsd
            while(j < n):
                str = Decimal(A[i][j])-Decimal(koef*A[i-m][j])
                A[i][j] = int(str)
                j+=1
            i+=1
            j = k-1
            m+=1
            k+=1

```

```

    i=k

    j = i - 1

    m = 1

    return A

def make_zero_up(A):

    n = len(A)

    i = 0

    j = n - 1

    k = 1

    m = n - 1

    while(k <= n-1):

        KOEF = Fraction(1,A[j][j])

        A[j][j]=Decimal(A[j][j]/A[j][j])

        free_row[j] = KOEF*free_row[j]

        while(i < m):

            koef = Decimal(A[i][j]/A[m][j])

            KOEF = Fraction(koef,1)

            free_row[i] = free_row[i]-KOEF*free_row[m]

            nsd = NSD(free_row[i].chis,free_row[i].znam)

            free_row[i] = free_row[i]/nsd

            while(j > 0):

                str = Decimal(A[i][j])-Decimal(koef*A[m][j])

                A[i][j] = int(str)

                j-=1

            i+=1

            j = m

        k+=1

        m-=1

        i = 0

        j = m

    return A

def NSD(a, b): # Функція знаходження найбільшого спільного дільника

```

```

if a % b == 0:
    return b
else:
    return NSD(b, a%b)
def coefficient():
    i=0
    for myobj in free_row:
        if((myobj.chis > 0 and myobj.znam < 0) or (myobj.chis < 0 and myobj.znam > 0)):
            print('a{:s} = {:d} / {:d}'.format(str(i),-abs(myobj.chis),abs(myobj.znam)))
            print()
        else:
            print('a{:s} = {:d} / {:d}'.format(str(i),abs(myobj.chis),abs(myobj.znam)))
            print()
        i+=1
def fi1(theta):
    return 1 - cos(theta)
def fi2(theta):
    return -sin(theta)
def fi3(theta):
    i = 0
    suma=0
    for obj in free_row:
        suma = suma+(obj.chis/obj.znam)*cos(i*theta)
        i-=1
    return suma
def fi4(theta):
    i = 0
    suma = 0
    for obj in free_row:
        suma = suma+(obj.chis/obj.znam)*sin(i*theta)
        i-=1
    return suma

```

```

def build_graphic(ro,n):

    fig,axes = plt.subplots()

    angle_pi=[]

    k = 5000

    u = []

    v = []

    u_copy = []

    v_copy=[]

    for i in range(k+1):

        angle_pi.append(pi*i/(k/2))

        u_temp = (fi1(angle_pi[i])*fi3(angle_pi[i])+fi2(angle_pi[i])*fi4(angle_pi[i]))/(fi3(angle_pi[i])*fi3(angle_pi[i])+fi4(angle_pi[i])*fi4(angle_pi[i]))

        v_temp = (fi2(angle_pi[i])*fi3(angle_pi[i])-fi1(angle_pi[i])*fi4(angle_pi[i]))/(fi3(angle_pi[i])*fi3(angle_pi[i])+fi4(angle_pi[i])*fi4(angle_pi[i]))

        if n == 5 or n == 4 or n == 6:

            if u_temp <= 0 or v_temp < -0.5 or v_temp > 0.5:

                u_copy.append(u_temp)

                v_copy.append(v_temp)

            else:

                u.append(u_temp)

                v.append(v_temp)

        else:

            u.append(u_temp)

            v.append(v_temp)

    suma=0

    i = 0

    for obj in free_row:

        suma+=(obj.chis/obj.znam)*ro**i

        i-=1

    a = (1 - ro)/suma

    if(a>= min(u) and a <= max(u)):

        plt.fill(u,v, facecolor = "grey")

```

```

else:

    plt.gca().set_facecolor('xkcd:grey')

    plt.fill(u,v,facecolor='white')

axes.set_title("Область абсолютної стійкості виділено сірим кольором")

axes.set_xlabel("Re  $\sigma$ ")

axes.set_ylabel("Im  $\sigma$ ")

axes.axhline(y=0, color='k')

axes.axvline(x=0, color='k')

plt.plot(a,0,'ko')

plt.plot(u,v,'black')

if(len(u_copy)!= 0 and len(v_copy)!=0):

    plt.plot(u_copy,v_copy,'grey')

plt.show()

while True:

    try:

        n=int(input("Введіть порядок різницевої схеми Адамса-Башфорда: "))

        if(n < 1):

            print("Порядок різницевої схеми повинен бути більшим за 0")

        else:

            A,free_row = build_matrix(n)

            if(n==1):

                free_row[0] = Fraction(1,1)

            elif(n==2):

                A = make_zero_up(A)

            else:

                A = make_zero_under(A)

                A = make_zero_up(A)

            coefficient()

            build_graphic(0.92,n)

            break

    except ValueError:

        print("Порядок різницевої схеми повинен бути числом")

```

## Різницеві схеми Адамса-Маултона

```
from decimal import Decimal

import matplotlib.pyplot as plt

from math import cos, sin, pi

class Fraction:

    def __init__(self,a,b):

        self.chis = a

        self.znam = b

    def __mul__(self,ob):

        newob = Fraction(1,1)

        newob.chis = self.chis*ob.chis

        newob.znam = self.znam*ob.znam

        return newob

    def __sub__(self, ob):

        newob = Fraction(1,1)

        newob.chis = self.chis*ob.znam-self.znam*ob.chis

        newob.znam = self.znam*ob.znam

        return newob

    def __truediv__(self,num):

        newob = Fraction(1,1)

        newob.chis = int(self.chis/num)

        newob.znam = int(self.znam/num)

        return newob

    def __str__(self):

        return str(self.chis)+"/"+str(self.znam)

def build_matrix(n):

    A = []

    free_row = []

    for j in range(1,n+1):

        temp = []

        for i in range(-1,n-1):

            temp.append(int(Decimal((-i)**(j-1))))
```



```

A.append(temp)

free_row.append(Fraction(1,j))

return A,free_row

def make_zero_under(A):

n = len(A)

if(n==1):

    return 1

elif(n==2):

    temp_row = A[0]

    A[0] = A[1]

    A[1] = temp_row

    temp_val = free_row[0]

    free_row[0] = free_row[1]

    free_row[1] = temp_val

    i = 1

    j = 0

    m = 1

    while(i < n):

        koef = Decimal(A[i][j]/A[i-m][j])

        KOEF = Fraction(koef,1)

        free_row[i] = free_row[i]-KOEF*free_row[i-m]

        nsd = NSD(free_row[i].chis,free_row[i].znam)

        free_row[i] = free_row[i]/nsd

        while(j < n):

            str = Decimal(A[i][j])-Decimal(koef*A[i-m][j])

            A[i][j] = int(str)

            j+=1

        i+=1

        j = 0

        m+=1

    return -1

else:

```

```

temp_row = A[0]
A[0] = A[1]
A[1] = temp_row
temp_val = free_row[0]
free_row[0] = free_row[1]
free_row[1] = temp_val

i = 1
j = 0
k = 3
m = 1
while(i < n):
    koef = Decimal(A[i][j]/A[i-m][j])
    KOEF = Fraction(koef,1)
    free_row[i] = free_row[i]-KOEF*free_row[i-m]
    nsd = NSD(free_row[i].chis,free_row[i].znam)
    free_row[i] = free_row[i]/nsd
    while(j < n):
        str = Decimal(A[i][j])-Decimal(koef*A[i-m][j])
        A[i][j] = int(str)
        j+=1
    i+=1
    j = 0
    m+=1
i = 3
j = 2
m = 1
while(k <= n-1):
    while(i < n):
        koef = Decimal(A[i][j]/A[i-m][j])
        KOEF = Fraction(koef,1)
        free_row[i] = free_row[i]-KOEF*free_row[i-m]
        nsd = NSD(free_row[i].chis,free_row[i].znam)

```

```

    free_row[i] = free_row[i]/nsd

    while(j < n):

        str = Decimal(A[i][j])-Decimal(koef*A[i-m][j])

        A[i][j] = int(str)

        j+=1

    i+=1

    j = k-1

    m+=1

    k+=1

    i=k

    j = i - 1

    m = 1

    return A

def make_zero_up(A):

    n = len(A)

    i = 0

    j = n - 1

    k = 1

    m = n - 1

    while(k <= n-2):

        KOEF = Fraction(1,A[j][j])

        A[j][j]=Decimal(A[j][j]/A[j][j])

        free_row[j] = KOEF*free_row[j]

        while(i < m):

            koef = Decimal(A[i][j]/A[m][j])

            KOEF = Fraction(koef,1)

            free_row[i] = free_row[i]-KOEF*free_row[m]

            nsd = NSD(free_row[i].chis,free_row[i].znam)

            free_row[i] = free_row[i]/nsd

        while(j > 0):

            str = Decimal(A[i][j])-Decimal(koef*A[m][j])

            A[i][j] = int(str)

```

```

        j-=1

        i+=1

        j = m

        k+=1

        m-=1

        i = 0

        j = m

    return A

def NSD(a, b): # Функція знаходження найбільшого спільного дільника

    if a % b == 0:

        return b

    else:

        return NSD(b, a%b)

def coefficient():

    i=0

    for obj in free_row:

        nsd = NSD(abs(obj.chis),abs(obj.znam))

        myobj = obj/nsd

        if((myobj.chis > 0 and myobj.znam < 0) or (myobj.chis < 0 and myobj.znam > 0)):

            print('a{:s} = {:d} / {:d}'.format(str(i),-abs(myobj.chis),abs(myobj.znam)))

            print()

        else:

            print('a{:s} = {:d} / {:d}'.format(str(i),abs(myobj.chis),abs(myobj.znam)))

            print()

        i+=1

def fi1(theta):

    return 1 - cos(theta)

def fi2(theta):

    return -sin(theta)

def fi3(theta):

    i = 1

    suma=0

```

```

for obj in free_row:

    suma = suma+(obj.chis/obj.znam)*cos(i*theta)

    i-=1

return suma

def fi4(theta):

    i = 1

    suma = 0

    for obj in free_row:

        suma = suma+(obj.chis/obj.znam)*sin(i*theta)

        i-=1

    return suma

def build_graphic(ro,n):

    fig, axes = plt.subplots()

    angle_pi=[]

    k = 5000

    u = []

    v = []

    for i in range(k+1):

        angle_pi.append(pi*i/(k/2))

        u_temp = (fi1(angle_pi[i])*fi3(angle_pi[i])+fi2(angle_pi[i])*fi4(angle_pi[i]))/(fi3(angle_pi[i])*fi3(angle_pi[i])+fi4(angle_pi[i])*fi4(angle_pi[i]))

        v_temp = (fi2(angle_pi[i])*fi3(angle_pi[i])-fi1(angle_pi[i])*fi4(angle_pi[i]))/(fi3(angle_pi[i])*fi3(angle_pi[i])+fi4(angle_pi[i])*fi4(angle_pi[i]))

        u.append(u_temp)

        v.append(v_temp)

    suma=0

    i = 1

    for obj in free_row:

        suma+=(obj.chis/obj.znam)*ro**i

        i-=1

    a = (1-ro)/suma

    if(a>= min(u) and a <= max(u)):

```

```

plt.fill(u,v,facecolor = "grey")
else:
    plt.gca().set_facecolor('xkcd:grey')
    plt.fill(u,v,facecolor='white')
axes.set_title("Область абсолютної стійкості виділено сірим кольором")
axes.set_xlabel("Re  $\sigma$ ")
axes.set_ylabel("Im  $\sigma$ ")
axes.axhline(y=0, color='k')
axes.axvline(x=0, color='k')
plt.plot(a,0,'ko')
plt.plot(u,v,'black')
plt.show()
while True:
    try:
        n=int(input("Введіть порядок різницевої схеми Адамса-Маултона: "))
        if(n < 1):
            print("Порядок різницевої схеми повинен бути більшим за 0")
        else:
            A,free_row = build_matrix(n)
            if(n==1):
                free_row[0] = Fraction(1,1)
            elif(n==2):
                A = make_zero_under(A)
            else:
                A = make_zero_under(A)
                A = make_zero_up(A)
            coefficient()
            build_graphic(0.92,n)
            break
    except ValueError:
        print("Порядок різницевої схеми повинен бути числом")

```

## Різницеві схеми Гіра

```
from decimal import Decimal

import numpy as np

import matplotlib.pyplot as plt

from math import *

def u(teta, free_delta, main_delta):

    res = 0

    for i in range(len(free_delta)-1):

        temp = float(free_delta[i]/main_delta)*float(main_delta/free_delta[len(free_delta)-1])*cos((i+1)*teta)

        res+=temp

    res-=float(main_delta/free_delta[len(free_delta)-1])

    return res

def v(teta, free_delta, main_delta):

    res = 0

    for i in range(len(free_delta)-1):

        temp = float((free_delta[i]/main_delta))*float(main_delta/free_delta[len(free_delta)-1])*sin((i+1)*teta)*(-1)

        res+=temp

    return res

def build_graphic(main_delta, free_delta,ro):

    fig, axes = plt.subplots()

    angle_pi=[]

    k=5000

    for i in range (k+1):

        angle_pi.append(pi*i/(k/2))

    U = np.zeros(len(angle_pi))

    V = np.zeros(len(angle_pi))

    for i in range(len(angle_pi)):

        U[i] = u(angle_pi[i],free_delta,main_delta)

        V[i] = v(angle_pi[i],free_delta,main_delta)

        if(i == k/2):

            b = U[i]

    a=0
```

```

#print("δ = ",round(max(U),1))

for i in range(len(free_delta)-1):

    temp = float(free_delta[i]/main_delta)*float(main_delta/free_delta[len(free_delta)-1])*ro**(-(i+1))

    a+=temp

a-=float(main_delta/free_delta[len(free_delta)-1])

axes.set_title("Область абсолютної стійкості виділено сірим кольором")

axes.set_xlabel("Re σ")

axes.set_ylabel("Im σ")

axes.axhline(y=0, color='k')

axes.axvline(x=0, color='k')

if((a < U[0] and a > b) or n>6):

    plt.fill(U,V, facecolor = "grey")

else:

    plt.gca().set_facecolor('xkcd:grey')

    plt.fill(U,V,facecolor='white')

plt.plot(a,0,'ko')

plt.plot(U,V,'grey')

plt.show()

def build_matrix(n):

    A = []

    for j in range(n+1):

        temp = []

        for k in range(n):

            temp.append(int(Decimal((-k)**j)))

        temp.append(j)

        A.append(temp)

    return A

def find_determinant(A,n):

    if(n == 1):

        return 1

    elif(n == 2):

        return -1

```



else:

```
if(A[1][0] == 0):
```

```
    i = 2
```

```
    i_const = 2
```

```
    j = 1
```

```
    k = 1
```

```
    m = 1
```

```
    z = 1
```

```
while(z < n-1):
```

```
    while(i < n):
```

```
        koef = Decimal(A[i][j])/Decimal(A[i-m][j])
```

```
        while(k < n):
```

```
            str =Decimal(A[i][k]-int(koef)*A[i-m][k])
```

```
            a=int(str)
```

```
            A[i][k] = a
```

```
            k+=1
```

```
        m+=1
```

```
        i+=1
```

```
        k=1
```

```
    i_const+=1
```

```
    i=i_const
```

```
    j+=1
```

```
    z+=1
```

```
    m=1
```

```
determinant = 1
```

```
i = 0
```

```
j = 0
```

```
while(i < n):
```

```
    determinant = determinant*A[i][j]
```

```
    i+=1
```

```
    j+=1
```

```
return determinant
```

```

else:
    i = 1
    i_const = 1
    j = 0
    k = 0
    m = 1
    z = 0
    while(z < n-1):
        while(i < n):
            koef = Decimal(A[i][j])/Decimal(A[i-m][j])
            while(k < n):
                str =Decimal(A[i][k]-int(koef)*A[i-m][k])
                a = int(str)
                A[i][k] = a
                k+=1
            m+=1
            i+=1
            k=1
        i_const+=1
        i=i_const
        j+=1
        z+=1
        m=1
    determinant = 1
    i = 0
    j = 0
    while(i < n):
        determinant = determinant*A[i][j]
        i+=1
        j+=1
    return determinant

```

```

def NSD(a, b): # Функція знаходження найбільшого спільного дільника

    if a % b == 0:

        return b

    else:

        return NSD(b, a%b)

def koef_Gear(n):

    print("Коефіцієнти для різницевої схеми Гіра {:s} порядку\n".format(str(n)))

    D = build_matrix(n)

    C = build_matrix(n)

    main_delta = find_determinant(D,n+1)

    free_delta=[]

    for i in range (n+1):

        C = build_matrix(n)

        B = C

        for j in range (n+1):

            B[j][i] = 1

        free_delta.append(find_determinant(B,n+1))

    for i in range(len(free_delta)):

        nsd = NSD(abs(free_delta[i]),abs(main_delta))

        if(i != len(free_delta)-1):

            if(free_delta[i] > 0 and main_delta < 0):

                print('a{:s} = {:d} / {:d}'.format(str(i),int(-free_delta[i]/nsd),abs(int(main_delta/nsd))))

                print()

            elif(free_delta[i] < 0 and main_delta > 0):

                print('a{:s} = {:d} / {:d}'.format(str(i),int(free_delta[i]/nsd),int(main_delta/nsd)))

                print()

            else:

                print('a{:s} = {:d} / {:d}'.format(str(i),abs(int(free_delta[i]/nsd)),abs(int(main_delta/nsd))))

                print()

        else:

            if(free_delta[i] > 0 and main_delta < 0):

                print('b0 = {:d} / {:d}'.format(int(-free_delta[i]/nsd),abs(int(main_delta/nsd))))

```

```

    print()

elif(free_delta[i] < 0 and main_delta > 0):

    print('b0 = {:d} / {:d}'.format(int(free_delta[i]/nsd),int(main_delta/nsd)))

    print()

else:

    print('b0 = {:d} / {:d}'.format(abs(int(free_delta[i]/nsd)),abs(int(main_delta/nsd))))

    print()

build_graphic(main_delta,free_delta,0.7)

while True:

    try:

        n = int(input("Введіть порядок різницевої схеми: "))

        if(n < 1):

            print("Порядок різницевої схеми повинен бути більшим за 0")

        else:

            koef_Gear(n)

            break

    except ValueError:

        print("Порядок різницевої схеми повинен бути числом")

```

Додаток 2

Таблиця коефіцієнтів різницевих схем Адамса-Башфорта до п'ятнадцятого порядку

№ порядку	Коефіцієнти												
1	$b_0 = 1$												
2	$b_0 = \frac{3}{2}$	$b_1 = \frac{-1}{2}$											
3	$b_0 = \frac{23}{12}$	$b_1 = \frac{-4}{3}$	$b_2 = \frac{5}{12}$										
4	$b_0 = \frac{55}{24}$	$b_1 = \frac{-59}{24}$	$b_2 = \frac{37}{24}$	$b_3 = \frac{-3}{8}$									
5	$b_0 = \frac{1901}{720}$	$b_1 = \frac{-1387}{360}$	$b_2 = \frac{109}{30}$	$b_3 = \frac{-637}{360}$	$b_4 = \frac{251}{720}$								
6	$b_0 = \frac{4277}{1440}$	$b_1 = \frac{-2641}{480}$	$b_2 = \frac{4991}{720}$	$b_3 = \frac{-3649}{720}$	$b_4 = \frac{959}{480}$	$b_5 = \frac{-95}{288}$							
7	$b_0 = \frac{198721}{60480}$	$b_1 = \frac{-18637}{2520}$	$b_2 = \frac{235183}{20160}$	$b_3 = \frac{-10754}{945}$	$b_4 = \frac{135713}{20160}$	$b_5 = \frac{-5603}{2520}$	$b_6 = \frac{19087}{60480}$						
8	$b_0 = \frac{16083}{4480}$	$b_1 = \frac{-1152169}{120960}$	$b_2 = \frac{242653}{13440}$	$b_3 = \frac{-296053}{13440}$	$b_4 = \frac{2102243}{120960}$	$b_5 = \frac{-115747}{13440}$	$b_6 = \frac{32863}{13440}$	$b_7 = \frac{-5257}{17280}$					
9	$b_0 = \frac{14097247}{3628800}$	$b_1 = \frac{-21562603}{1814400}$	$b_2 = \frac{47738393}{1814400}$	$b_3 = \frac{-69927631}{1814400}$	$b_4 = \frac{862303}{22680}$	$b_5 = \frac{-45586321}{1814400}$	$b_6 = \frac{19416743}{1814400}$	$b_7 = \frac{-4832053}{1814400}$	$b_8 = \frac{1070017}{3628800}$				
10	$b_0 = \frac{4325321}{1036800}$	$b_1 = \frac{-104995189}{7257600}$	$b_2 = \frac{6648317}{181440}$	$b_3 = \frac{-28416361}{453600}$	$b_4 = \frac{269181919}{3628800}$	$b_5 = \frac{-222386081}{3628800}$	$b_6 = \frac{15788639}{453600}$	$b_7 = \frac{-2357683}{181440}$	$b_8 = \frac{20884811}{7257600}$	$b_9 = \frac{-25713}{89600}$			
11	$b_0 = \frac{2132509567}{479001600}$	$b_1 = \frac{-2067948781}{119750400}$	$b_2 = \frac{1572737587}{31933440}$	$b_3 = \frac{-1921376209}{19958400}$	$b_4 = \frac{3539798831}{26611200}$	$b_5 = \frac{-82260679}{623700}$	$b_6 = \frac{2492064913}{26611200}$	$b_7 = \frac{-186080291}{3991680}$	$b_8 = \frac{2472634817}{159667200}$	$b_9 = \frac{-52841941}{17107200}$	$b_{10} = \frac{26842253}{95800320}$		
12	$b_0 = \frac{4527766399}{958003200}$	$b_1 = \frac{-6477936721}{319334400}$	$b_2 = \frac{12326645437}{191600640}$	$b_3 = \frac{-15064372973}{106444800}$	$b_4 = \frac{35689892561}{159667200}$	$b_5 = \frac{-41290273229}{159667200}$	$b_6 = \frac{35183928883}{159667200}$	$b_7 = \frac{-625551749}{4561920}$	$b_8 = \frac{923636629}{15206400}$	$b_9 = \frac{-17410248271}{958003200}$	$b_{10} = \frac{30082309}{9123840}$	$b_{11} = \frac{-4777223}{17418240}$	
13	$b_0 = \frac{13064406523627}{2615348736000}$	$b_1 = \frac{-931781102989}{39626496000}$	$b_2 = \frac{5963794194517}{72648576000}$	$b_3 = \frac{-10498491598103}{52306974720}$	$b_4 = \frac{20730767690131}{58118860800}$	$b_5 = \frac{-34266367915049}{72648576000}$	$b_6 = \frac{228133014533}{486486000}$	$b_7 = \frac{-2826800577631}{8072064000}$	$b_8 = \frac{2253957198793}{11623772160}$	$b_9 = \frac{-20232291373837}{261534873600}$	$b_{10} = \frac{4588414555201}{217945728000}$	$b_{11} = \frac{-169639834921}{48432384000}$	$b_{12} = \frac{703604254357}{2615348736000}$

<b>14</b>	$b_0 = \frac{905730205}{172204032}$	$b_1 = \frac{-140970750679621}{5230697472000}$	$b_2 = \frac{89541175419277}{871782912000}$	$b_3 = \frac{-34412222659093}{124540416000}$	$b_4 = \frac{570885914358161}{1046139494400}$	$b_5 = \frac{-31457535950413}{1046139494400}$	$b_6 = \frac{134046425652457}{145297152000}$	$b_7 = \frac{-350379327127877}{435891456000}$	$b_8 = \frac{310429955875453}{581188608000}$	$b_9 = \frac{-10320787460413}{38745907200}$	$b_{10} = \frac{7222659159949}{74724249600}$	$b_{11} = \frac{-21029162113651}{871782912000}$	$b_{12} = \frac{6460951197929}{1743565824000}$	$b_{13} = \frac{-106364763817}{402361344000}$	
<b>15</b>	$b_0 = \frac{86616749299607}{15692092416000}$	$b_1 = \frac{-120015358300723}{3923023104000}$	$b_2 = \frac{24790135438867}{196151155200}$	$b_3 = \frac{-132314062847509}{356638464000}$	$b_4 = \frac{486517506482369}{603542016000}$	$b_5 = \frac{-5228158741563491}{3923023104000}$	$b_6 = \frac{185663287013681}{108972864000}$	$b_7 = \frac{-3286835719793}{1937295360}$	$b_8 = \frac{458669070403157}{348713164800}$	$b_9 = \frac{-950173217523373}{1207084032000}$	$b_{10} = \frac{1400731219189397}{3923023104000}$	$b_{11} = \frac{-58262613384023}{490377888000}$	$b_{12} = \frac{859236476684231}{31384184832000}$	$b_{13} = \frac{-696561442637}{178319232000}$	$b_{14} = \frac{1166309819657}{4483454976000}$

**Таблиця коефіцієнтів різницевих схем Адамса-Маултона до п'ятнадцятого порядку**

№ порядку	Коефіцієнти													
	$b_{-1}$	$b_0$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$b_6$	$b_7$	$b_8$	$b_9$	$b_{10}$	$b_{11}$	$b_{12}$
<b>1</b>	$b_{-1} = 1$													
<b>2</b>	$b_{-1} = \frac{1}{2}$	$b_0 = \frac{1}{2}$												
<b>3</b>	$b_{-1} = \frac{5}{12}$	$b_0 = \frac{2}{3}$	$b_1 = \frac{-1}{12}$											
<b>4</b>	$b_{-1} = \frac{3}{8}$	$b_0 = \frac{19}{24}$	$b_1 = \frac{-5}{24}$	$b_2 = \frac{1}{24}$										
<b>5</b>	$b_{-1} = \frac{251}{720}$	$b_0 = \frac{323}{360}$	$b_1 = \frac{-11}{30}$	$b_2 = \frac{53}{360}$	$b_3 = \frac{-19}{720}$									
<b>6</b>	$b_{-1} = \frac{95}{288}$	$b_0 = \frac{1427}{1440}$	$b_1 = \frac{-133}{240}$	$b_2 = \frac{241}{720}$	$b_3 = \frac{-173}{1440}$	$b_4 = \frac{3}{160}$								
<b>7</b>	$b_{-1} = \frac{19087}{60480}$	$b_0 = \frac{2713}{2520}$	$b_1 = \frac{-15487}{20160}$	$b_2 = \frac{586}{945}$	$b_3 = \frac{-6737}{20160}$	$b_4 = \frac{263}{2520}$	$b_5 = \frac{-863}{60480}$							
<b>8</b>	$b_{-1} = \frac{5257}{17280}$	$b_0 = \frac{139849}{120960}$	$b_1 = \frac{-4511}{4480}$	$b_2 = \frac{123133}{120960}$	$b_3 = \frac{-88547}{120960}$	$b_4 = \frac{1537}{4480}$	$b_5 = \frac{-11351}{120960}$	$b_6 = \frac{275}{24192}$						
<b>9</b>	$b_{-1} = \frac{1070017}{3628800}$	$b_0 = \frac{2233547}{1814400}$	$b_1 = \frac{-2302297}{1814400}$	$b_2 = \frac{2797679}{1814400}$	$b_3 = \frac{-31457}{22680}$	$b_4 = \frac{1573169}{1814400}$	$b_5 = \frac{-645607}{1814400}$	$b_6 = \frac{156437}{1814400}$	$b_7 = \frac{-33953}{3628800}$					
<b>10</b>	$b_{-1} = \frac{25713}{89600}$	$b_0 = \frac{9449717}{7257600}$	$b_1 = \frac{-1408913}{907200}$	$b_2 = \frac{200029}{90720}$	$b_3 = \frac{-8641823}{3628800}$	$b_4 = \frac{6755041}{3628800}$	$b_5 = \frac{-462127}{453600}$	$b_6 = \frac{335983}{907200}$	$b_7 = \frac{-116687}{1451520}$	$b_8 = \frac{8183}{1036800}$				
<b>11</b>	$b_{-1} = \frac{26842253}{95800320}$	$b_0 = \frac{164046413}{119750400}$	$b_1 = \frac{-296725183}{159667200}$	$b_2 = \frac{12051709}{3991680}$	$b_3 = \frac{-33765029}{8870400}$	$b_4 = \frac{2227571}{623700}$	$b_5 = \frac{-21677723}{8870400}$	$b_6 = \frac{23643791}{19958400}$	$b_7 = \frac{-12318413}{31933440}$	$b_8 = \frac{9071219}{119750400}$	$b_9 = \frac{-3250433}{479001600}$			

<b>12</b>	$b_{-1} = \frac{4777223}{17418240}$	$b_0 = \frac{1374799219}{958003200}$	$b_1 = \frac{-99642413}{45619200}$	$b_2 = \frac{36465037}{9123840}$	$b_3 = \frac{-102212233}{17740800}$	$b_4 = \frac{1007253581}{159667200}$	$b_5 = \frac{-91910491}{17740800}$	$b_6 = \frac{501289903}{159667200}$	$b_7 = \frac{-87064741}{63866880}$	$b_8 = \frac{384709327}{958003200}$	$b_9 = \frac{-68928781}{958003200}$	$b_{10} = \frac{4671}{788480}$			
<b>13</b>	$b_{-1} = \frac{703604254357}{2615348736000}$	$b_0 = \frac{6595204069}{4402944000}$	$b_1 = \frac{-551368413119}{217945728000}$	$b_2 = \frac{1346577425651}{261534873600}$	$b_3 = \frac{-485500845331}{58118860800}$	$b_4 = \frac{84400835489}{8072064000}$	$b_5 = \frac{-4874320027}{486486000}$	$b_6 = \frac{529394045911}{72648576000}$	$b_7 = \frac{-229882484333}{58118860800}$	$b_8 = \frac{406332786317}{261534873600}$	$b_9 = \frac{-30336027563}{72648576000}$	$b_{10} = \frac{2724891251}{39626496000}$	$b_{11} = \frac{-13695779093}{2615348736000}$		
<b>14</b>	$b_{-1} = \frac{106364763817}{402361344000}$	$b_0 = \frac{741197087471}{475517952000}$	$b_1 = \frac{-168235945379}{58118860800}$	$b_2 = \frac{16964495066809}{2615348736000}$	$b_3 = \frac{-1748248003751}{149448499200}$	$b_4 = \frac{9575580965507}{581188608000}$	$b_5 = \frac{-7866111554491}{435891456000}$	$b_6 = \frac{1335017017153}{87178291200}$	$b_7 = \frac{-5797545653629}{581188608000}$	$b_8 = \frac{5124051955567}{1046139494400}$	$b_9 = \frac{-4590817802567}{2615348736000}$	$b_{10} = \frac{1636420501}{3773952000}$	$b_{11} = \frac{-69091417279}{1046139494400}$	$b_{12} = \frac{2224234463}{475517952000}$	
<b>15</b>	$b_{-1} = \frac{1166309819657}{4483454976000}$	$b_0 = \frac{3173185470929}{1961511552000}$	$b_1 = \frac{-102885148956217}{31384184832000}$	$b_2 = \frac{3933201478249}{490377888000}$	$b_3 = \frac{-71363886250691}{4483454976000}$	$b_4 = \frac{48869476129477}{1961511552000}$	$b_5 = \frac{-321201800274911}{10461394944000}$	$b_6 = \frac{38029005269}{1277025750}$	$b_7 = \frac{-236770944732449}{10461394944000}$	$b_8 = \frac{26159487787579}{1961511552000}$	$b_9 = \frac{-187504936597931}{31384184832000}$	$b_{10} = \frac{137855863153}{70053984000}$	$b_{11} = \frac{-14110480969927}{31384184832000}$	$b_{12} = \frac{124922452271}{1961511552000}$	$b_{13} = \frac{-132282840127}{31384184832000}$

Таблиця коефіцієнтів різницьових схем Гіра до п'ятнадцятого порядку

№ порядку	Коефіцієнти																
1	$a_0 = 1$	$b_0 = 1$															
2	$a_0 = \frac{4}{3}$	$a_1 = -\frac{1}{3}$	$b_0 = \frac{2}{3}$														
3	$a_0 = \frac{18}{11}$	$a_1 = -\frac{9}{11}$	$a_2 = \frac{2}{11}$	$b_0 = \frac{6}{11}$													
4	$a_0 = \frac{48}{25}$	$a_1 = -\frac{36}{25}$	$a_2 = \frac{16}{25}$	$a_3 = -\frac{3}{25}$	$b_0 = \frac{12}{25}$												
5	$a_0 = \frac{300}{137}$	$a_1 = -\frac{300}{137}$	$a_2 = \frac{200}{137}$	$a_3 = -\frac{75}{137}$	$a_4 = \frac{12}{137}$	$b_0 = \frac{60}{137}$											
6	$a_0 = \frac{120}{49}$	$a_1 = -\frac{150}{49}$	$a_2 = \frac{400}{147}$	$a_3 = -\frac{75}{49}$	$a_4 = \frac{24}{49}$	$a_5 = -\frac{10}{147}$	$b_0 = \frac{20}{49}$										
7	$a_0 = \frac{980}{363}$	$a_1 = -\frac{490}{121}$	$a_2 = \frac{4900}{1089}$	$a_3 = -\frac{1225}{363}$	$a_4 = \frac{196}{121}$	$a_5 = -\frac{490}{1089}$	$a_6 = \frac{20}{363}$	$b_0 = \frac{140}{363}$									
8	$a_0 = \frac{2240}{761}$	$a_1 = -\frac{3920}{761}$	$a_2 = \frac{15680}{2283}$	$a_3 = -\frac{4900}{761}$	$a_4 = \frac{3136}{761}$	$a_5 = -\frac{3920}{2283}$	$a_6 = \frac{320}{761}$	$a_7 = -\frac{35}{761}$	$b_0 = \frac{280}{761}$								
9	$a_0 = \frac{22680}{7129}$	$a_1 = -\frac{45360}{7129}$	$a_2 = \frac{70560}{7129}$	$a_3 = -\frac{79380}{7129}$	$a_4 = \frac{63504}{7129}$	$a_5 = -\frac{35280}{7129}$	$a_6 = \frac{12960}{7129}$	$a_7 = -\frac{2835}{7129}$	$a_8 = \frac{280}{7129}$	$b_0 = \frac{2520}{7129}$							
10	$a_0 = \frac{25200}{7381}$	$a_1 = -\frac{56700}{7381}$	$a_2 = \frac{100800}{7381}$	$a_3 = -\frac{132300}{7381}$	$a_4 = \frac{127008}{7381}$	$a_5 = -\frac{88200}{7381}$	$a_6 = \frac{43200}{7381}$	$a_7 = -\frac{14175}{7381}$	$a_8 = \frac{2800}{7381}$	$a_9 = -\frac{252}{7381}$	$b_0 = \frac{2520}{7381}$						
11	$a_0 = \frac{304920}{83711}$	$a_1 = -\frac{762300}{83711}$	$a_2 = \frac{1524600}{83711}$	$a_3 = -\frac{2286900}{83711}$	$a_4 = \frac{2561328}{83711}$	$a_5 = -\frac{2134440}{83711}$	$a_6 = \frac{1306800}{83711}$	$a_7 = -\frac{571725}{83711}$	$a_8 = \frac{169400}{83711}$	$a_9 = -\frac{30492}{83711}$	$a_{10} = \frac{2520}{83711}$	$b_0 = \frac{27720}{83711}$					
12	$a_0 = \frac{332640}{86021}$	$a_1 = -\frac{914760}{86021}$	$a_2 = \frac{2032800}{86021}$	$a_3 = -\frac{3430350}{86021}$	$a_4 = \frac{4390848}{86021}$	$a_5 = -\frac{4268880}{86021}$	$a_6 = \frac{3136320}{86021}$	$a_7 = -\frac{1715175}{86021}$	$a_8 = \frac{677600}{86021}$	$a_9 = -\frac{182952}{86021}$	$a_{10} = \frac{30240}{86021}$	$a_{11} = -\frac{2310}{86021}$	$b_0 = \frac{27720}{86021}$				
13	$a_0 = \frac{4684680}{1145993}$	$a_1 = -\frac{14054040}{1145993}$	$a_2 = \frac{34354320}{1145993}$	$a_3 = -\frac{64414350}{1145993}$	$a_4 = \frac{92756664}{1145993}$	$a_5 = -\frac{103062960}{1145993}$	$a_6 = \frac{88339680}{1145993}$	$a_7 = -\frac{57972915}{1145993}$	$a_8 = \frac{28628600}{1145993}$	$a_9 = -\frac{10306296}{1145993}$	$a_{10} = \frac{2555280}{1145993}$	$a_{11} = -\frac{390390}{1145993}$	$a_{12} = \frac{27720}{1145993}$	$b_0 = \frac{360360}{1145993}$			
14	$a_0 = \frac{5045040}{1171733}$	$a_1 = -\frac{16396380}{1171733}$	$a_2 = \frac{43723680}{1171733}$	$a_3 = -\frac{90180090}{1171733}$	$a_4 = \frac{144288144}{1171733}$	$a_5 = -\frac{180360180}{1171733}$	$a_6 = \frac{176679360}{1171733}$	$a_7 = -\frac{135270135}{1171733}$	$a_8 = \frac{80160080}{1171733}$	$a_9 = -\frac{36072036}{1171733}$	$a_{10} = \frac{11924640}{1171733}$	$a_{11} = -\frac{2732730}{1171733}$	$a_{12} = \frac{388080}{1171733}$	$a_{13} = -\frac{25740}{1171733}$	$b_0 = \frac{360360}{1171733}$		
15	$a_0 = \frac{50450400}{1195757}$	$a_1 = -\frac{18918900}{1195757}$	$a_2 = \frac{54654600}{1195757}$	$a_3 = -\frac{122972850}{1195757}$	$a_4 = \frac{216432216}{1195757}$	$a_5 = -\frac{300600300}{1195757}$	$a_6 = \frac{331273800}{1195757}$	$a_7 = -\frac{289864575}{1195757}$	$a_8 = \frac{200400200}{1195757}$	$a_9 = -\frac{108216108}{1195757}$	$a_{10} = \frac{44717400}{1195757}$	$a_{11} = -\frac{13663650}{1195757}$	$a_{12} = \frac{2910600}{1195757}$	$a_{13} = -\frac{386100}{1195757}$	$a_{14} = \frac{24024}{1195757}$	$b_0 = \frac{360360}{1195757}$	