

Міністерство освіти і науки України
Вінницький національний технічний університет

І. В. БОГАЧ, О. Ю. КРАКОВЕЦЬКИЙ, Л. В. КРИЛИК

**ЧИСЕЛЬНІ МЕТОДИ РОЗВ'ЯЗАННЯ
ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ
ЗАСОБАМИ MATHCAD**

Навчальний посібник

Вінниця
ВНТУ
2020

УДК 517.9:004.4](075.8)

Б73

Рекомендовано до друку Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 11 від 28.05.2020 р.)

Рецензенти:

Н. І. Заболотна, доктор технічних наук, професор

Т. Б. Мартинюк, доктор технічних наук, професор

А. Я. Кулик, доктор технічних наук, професор

Богач, І. В.

Б73 Чисельні методи розв'язання диференціальних рівнянь засобами MathCAD : навчальний посібник / І. В. Богач, О. Ю. Краковецький, Л. В. Крилик. – Вінниця : ВНТУ, 2020. – 106 с.

ISBN 978-966-641-802-2

В навчальному посібнику розглянуто основні теоретичні підходи до розв'язання різних типів диференціальних рівнянь засобами MathCAD та подано понад 60 прикладів розв'язання прикладних математичних задач.

Навчальний посібник призначений для студентів комп'ютерних спеціальностей та може бути корисний аспірантам, науковцям, інженерам та програмістам-практикам, діяльність яких пов'язана з розв'язанням різних типів задач з диференціальними рівняннями.

УДК 517.9:004.4](075.8)

ISBN 978-966-641-802-2

© ВНТУ, 2020

ЗМІСТ

ВСТУП.....	5
1 ЗАСТОСУВАННЯ ОСНОВНИХ ЧИСЕЛЬНИХ МЕТОДІВ ДЛЯ РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ В МАТЕМАТИЧНОМУ СЕРЕДОВИЩІ MATHCAD	6
1.1 Однокрокові методи розв'язання задачі Коші	7
1.2 Багатокрокові методи розв'язання задачі Коші	15
1.3 Методи прогнозу і корекції	17
1.4 Розв'язання рівнянь вищих порядків і систем диференціальних рівнянь	19
1.5 Оцінення похибки за правилом Рунге.....	23
1.6 Розв'язання крайової задачі метод кінцевих різниць	25
Питання для самоконтролю	32
Задачі для самоконтролю	32
2 РОЗВ'ЯЗАННЯ ЗВИЧАЙНИХ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ	34
2.1 Обчислювальний блок Given ... Odesolve	34
2.2 Функції rkfixed, Rkadapt і Bulstoer.....	42
2.2.1 Диференціальні рівняння першого порядку.....	43
2.2.2 Диференціальні рівняння n-го порядку	44
2.2.3 Системи диференціальних рівнянь	46
2.3 Функція statespace	48
2.4 «Жорсткі» системи диференціальних рівнянь	49
2.5 Знаходження розв'язку в кінцевій точці	52
2.6 Розв'язання крайових задач	54
Питання для самоконтролю	59
Задачі для самоконтролю	59
3 РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ В ЧАСТИННИХ ПОХІДНИХ.....	61
3.1 Розв'язання параболічних і гіперболічних рівнянь	61
3.1.1 Функції Pdesolve і numol	61
3.1.2 Розв'язання параболічного рівняння теплопровідності за допомогою функції Pdesolve.....	65
3.1.3 Розв'язання гіперболічного хвильового рівняння за допомогою функції Pdesolve	68
3.1.4 Розв'язання хвильового рівняння за допомогою функції numol.....	70
3.2 Еліптичні рівняння	71
3.2.1 Рівняння Пуассона з нульовими граничними умовами	72
3.2.2 Рівняння Пуассона з довільними граничними умовами	74
Питання для самоконтролю	78
Задачі для самоконтролю	78

4 ДОСЛІДЖЕННЯ СИСТЕМ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ.....	79
4.1 Побудова інтегральної кривої задачі Коші.....	79
4.2 Побудова векторного поля автономної системи.....	80
4.3 Поведінка розв'язків в околі стійкого вузла	81
4.4 Поведінка розв'язків в околі нестійкого вузла	82
4.5 Поведінка розв'язків в околі сідла	84
4.6 Поведінка розв'язків в околі центра.....	85
4.7 Поведінка розв'язків в околі стійкого фокусу	87
4.8 Поведінка розв'язків в околі нестійкого фокусу	88
4.9 Поведінка розв'язків в околі діакритичного вузла	90
Питання для самоконтролю	91
Задачі для самоконтролю	92
5 РОЗВ'ЯЗАННЯ ЗАДАЧ, ЯКІ ОПИСУЮТЬСЯ ДИФЕРЕНЦІАЛЬНИМИ РІВНЯННЯМИ	93
5.1 Розв'язання системи рівнянь Ван дер Поля	93
5.2 Рівняння Лотки-Вольтерра.....	96
5.3 Періодичні коливання матеріальної точки.....	99
5.4 Зміна об'єму виробництва в економічній системі.....	101
5.5 Затухаючі коливання.....	102
Питання для самоконтролю	103
ГЛОСАРІЙ.....	104
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	105

ВСТУП

При математичному описі різноманітних процесів, явищ та залежностей, що містять елементи «руху», користуються математичними моделями у вигляді рівнянь, до яких, крім незалежних величин і залежних від них шуканих функцій, входять також похідні (диференціали) певного порядку від шуканих функцій. Такі рівняння називають диференціальними.

Кожному інженеру під час своєї практичної діяльності доводиться розв'язувати такі задачі. Це і не дивно, оскільки будь-яке моделювання складних систем пов'язане з розв'язанням диференціальних рівнянь (ДР). Економічні та екологічні теорії, задачі механіки, фізики, математичної фізики так чи інакше пов'язані з розв'язанням ДР.

В навчальному посібнику пропонується увазі читачів теорія та приклади розв'язання диференціальних рівнянь та систем з використанням математичного пакета MathCAD.

Структура посібника містить п'ять розділів.

Перший розділ присвячено реалізації чисельних методів за відомими алгоритмами в пакеті MathCAD без використання вбудованих функцій та процедур пакета.

Другий та третій розділи присвячені опису вбудованих засобів математичного пакета MathCAD, наведено рекомендації щодо застосування тих чи інших функцій та приклади, які демонструють їх роботу.

В четвертому розділі описано застосування пакета MathCAD для розв'язання задач з теорії стійкості автоматичних систем.

В останньому розділі наведено ряд задач з різноманітних галузей науки, які зводяться до розв'язання ДР.

В посібнику зустрічаються такі позначки:



Коментар



Приклад

1 ЗАСТОСУВАННЯ ОСНОВНИХ ЧИСЕЛЬНИХ МЕТОДІВ ДЛЯ РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ В МАТЕМАТИЧНОМУ СЕРЕДОВИЩІ MATHCAD

Диференціальне рівняння – це рівняння, в якому невідомими є функції однієї або декількох змінних і яке містить похідні функції за цими змінними. Розв'язати диференціальне рівняння – знайти невідому функцію на певному інтервалі її змінних.

Рівняння, в які входять похідні тільки за однією змінною, *називаються звичайними диференціальними рівняннями (ЗДР)*, в протилежному випадку їх називають *рівнянням в частинних похідних*.

MathCAD дозволяє розв'язати диференціальні рівняння різних типів:

- *задачі Коші* – для яких визначені початкові умови, тобто задані значення цих функції в початковій точці інтервалу інтегрування (початкові умови);

- *крайові задачі* – для яких визначені значення функції чи її похідних відразу на обох границях інтервалу (граничні умови).

Задача Коші для диференціального рівняння першого порядку задається таким чином: знайти функцію $y(x)$, яка задовольняє диференціальне рівняння

$$y' = f(x, y) \quad (1.1)$$

і початкову умову

$$y(x_0) = y_0. \quad (1.2)$$

Розв'язання задачі Коші полягає в знаходженні наближеного розв'язку цієї задачі на кінцевій множині точок інтервалу $[a, b]$, який називається сіткою:

$$\begin{aligned} x_i &= x_0 + i \cdot h, \quad x_0 = a, \quad x_n = b, \\ h &= \frac{(b-a)}{n}, \quad i = 0, 1, 2, \dots, n. \end{aligned} \quad (1.3)$$

В основі чисельних методів розв'язання диференціальних рівнянь лежить розкладання функції y в ряд Тейлора навколо вихідної точки x_0 :

$$y(x_0 + h) = y(x_0) + hy'(x_0) + \frac{1}{2}h^2 y''(x_0) + \dots + \frac{1}{i!}h^i y^{(i)}(x_0). \quad (1.4)$$

Наближеним розв'язком задачі буде деяка сіткова функція $y = y(x)$. Для того щоб отримати значення сіткової функції використовують різні методи, в основі яких є заміна похідної будь-яким різницеvim рівнянням.

1.1 Однокрокові методи розв'язання задачі Коші

Метод Ейлера є найпростішим чисельним методом для розв'язання диференціальних рівнянь.

Розглянемо основну ідею методу Ейлера. Замінімо похідну y' в кожному i -у вузлі сітки правою різницею:

$$y'_i = \frac{(y_{i+1} - y_i)}{h}$$

і, підставивши її в рівняння (1.1), отримаємо:

$$\frac{(y_{i+1} - y_i)}{h} = f(x, y).$$

Розв'язавши це рівняння відносно y_{i+1} , отримаємо формулу Ейлера:

$$y_{i+1} = y_i + h \cdot f(x_i, y_i). \quad (1.5)$$

Оскільки значення y_0 відомо з початкових умов, то за формулою (1.5) можна послідовно отримати значення y_i в усіх вузлах сітки.

Метод Ейлера дуже простий в реалізації, але має невисоку точність. Він є методом першого порядку точності відносно кроку h , а його глобальна похибка оцінюється виразом:

$$\left| y_i - y_i^* \right| \leq O(h), \quad (1.6)$$

де y_i^* – точне значення задачі (1.1) – (1.2) в точці x_i ;

$O(h)$ – похибка на одному кроці h .

Оскільки похибка при розв'язанні рівнянь звичайним методом Ейлера досить суттєва, то цей метод на практиці використовується рідко. Натомість пропонуються *модифікований* та *вдосконалений (виправлений)* методи Ейлера, які є методами другого порядку, і для обчислення значення функції використовують два члени ряду Тейлора (1.4).

Похибка цих методів пропорційна h^2 , тобто $\left| y_i - y_i^* \right| < O(h^2)$.

Для модифікованого методу Ейлера розрахункова формула має вигляд:

$$y_{i+1} = y_i + hf \left(x_i + \frac{h}{2}, y_i + h \frac{f(x_i, y_i)}{2} \right). \quad (1.7)$$

У вдосконаленому методі для обчислення функції $y(x)$ в одній точці потрібно двічі обчислити функцію $f(x, y)$:

$$y_{i+1} = y_i + \frac{h}{2} \left(f(x_i, y_i) + f(x_i + h, y_i + hf(x_i, y_i)) \right). \quad (1.8)$$

Крім цих методів розрізняють так званий *неявний (обернений) метод Ейлера*, розрахункова формула якого має такий вигляд:

$$y_{i+1} = y_i + hf(x_i, y_{i+1}). \quad (1.9)$$

Практична реалізація неявної схеми Ейлера можлива, якщо вираз (1.9) вдається звести до вигляду:

$$y_{i+1} = g(h, x_i, y_i). \quad (1.10)$$

Методи Ейлера відносяться до групи з загальною назвою методи Рунге–Кутта, до цієї ж групи належить метод, який має назву класичний метод Рунге–Кутта четвертого порядку. Згідно з цим методом для обчислення значення функції $y(x)$ необхідно обчислити функцію $f(x, y)$ в чотирьох точках:

$$y_{i+1} = y_i + h \cdot \frac{(k1_i + 2k2_i + 2k3_i + k4_i)}{6}, \quad (1.11)$$

де коефіцієнти $k1, k2, k3, k4$ визначаються із співвідношень:

$$\begin{aligned} k1_i &= f(x_i, y_i), \\ k2_i &= f\left(x_i + \frac{h}{2}, y_i + h \frac{k1_i}{2}\right), \\ k3_i &= f\left(x_i + \frac{h}{2}, y_i + h \frac{k2_i}{2}\right), \\ k4_i &= f(x_i + h, y_i + hk3_i). \end{aligned}$$

Похибка цього методу пропорційна h^4 , тобто

$$\left| y_i - y_i^* \right| < O(h^4).$$

Методи Рунге–Кута відносяться до однокрокових методів, оскільки для обчислення значення функції $y(x)$ в точці x_{i+1} необхідно знати тільки значення функції $y(x)$ в одній попередній точці x_i .

Цей метод реалізований у вигляді вбудованих функцій *rkfixed* і *odesolve*, які будуть розглянуті в другому розділі.



Реалізуємо усі однокрокові методи розв'язання диференціальних рівнянь і розв'яжемо таку задачу Коші:

$$y' = xy, y(0) = 1 \text{ на інтервалі } [0, 1] \text{ з кроком } h = 0.1.$$

Для розв'язання такої задачі необхідно визначити функцію $f(x, y)$, початкове та кінцеве значення інтервалу інтегрування a і b , крок інтегрування h та кількість кроків інтегрування N .

Основним блоком є запис різницевої схеми для кожного з методів розв'язання рівняння. Знаходження розв'язків диференціального рівняння реалізовано у вигляді підпрограм.

Програму розв'язання задачі трьома методами Ейлера зображено на рисунку 1.1. Якщо порівняти графічні розв'язки задачі різними методами, то можна побачити, що розв'язки, отримані за модифікованим та виправленим методами, майже збігаються.

Неявний метод Ейлера є першим з групи неявних методів, що є основними для розв'язання «жорстких» систем диференціальних рівнянь. «Жорсткі» задачі мають місце тільки для рівнянь другого та вищих порядків, тому для рівняння першого порядку приклад застосування неявного методу має ілюстративний характер.

Для розв'язання задачі неявним методом Ейлера зробимо такі перетворення:

$$y_{i+1} = y_i + hx_i y_{i+1},$$

$$y_{i+1} - hx_i y_{i+1} = y_i,$$

$$y_{i+1}(1 - hx_i) = y_i,$$

$$y_{i+1} = \frac{y_i}{1 - hx_i},$$

$$g(h, x_i, y_i) = \frac{y_i}{1 - hx_i}.$$

Отже, для нашої задачі розрахункова формула має вигляд:

$$y_{i+1} = g(h, x_i, y_i).$$

Реалізація неявного методу Ейлера зображена на рисунку 1.2.

Реалізацію методу Рунге-Кутта четвертого порядку зображено на рисунку 1.3, реалізація у вигляді підпрограми – на рисунку 1.4.

Початкові умови

$$f(x, y) := x \cdot y \quad \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$a := 0 \quad b := 1 \quad N := 10 \quad h := \frac{b-a}{N} \quad i := 1..N$$

Звичайний метод Ейлера

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} := \begin{pmatrix} x_{i-1} + h \\ y_{i-1} + h \cdot f(x_{i-1}, y_{i-1}) \end{pmatrix} \quad Y := \begin{array}{l} Y_0 \leftarrow y_0 \\ \text{for } i \in 1..N \\ \quad Y_i \leftarrow Y_{i-1} + h \cdot f(x_{i-1}, Y_{i-1}) \\ Y \end{array}$$

$$y^T =$$

	0	1	2	3	4	5	6	7	8	9	10
0	1	1	1.01	1.03	1.061	1.104	1.159	1.228	1.314	1.419	1.547

$$Y^T =$$

	0	1	2	3	4	5	6	7	8	9	10
0	1	1	1.01	1.03	1.061	1.104	1.159	1.228	1.314	1.419	1.547

$$Y_{\text{zvichainiy}} := Y$$

Модифікований метод Ейлера

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} := \begin{pmatrix} x_{i-1} + h \\ y_{i-1} + h \cdot f\left(x_{i-1} + \frac{h}{2}, y_{i-1} + h \cdot \frac{f(x_{i-1}, y_{i-1})}{2}\right) \end{pmatrix}$$

$$Y := \begin{array}{l} Y_0 \leftarrow y_0 \\ \text{for } i \in 1..N \\ \quad Y_i \leftarrow Y_{i-1} + h \cdot f\left(x_{i-1} + \frac{h}{2}, Y_{i-1} + h \cdot \frac{f(x_{i-1}, Y_{i-1})}{2}\right) \\ Y \end{array}$$

$$y^T =$$

	0	1	2	3	4	5	6	7	8	9	10
0	1	1.005	1.02	1.046	1.083	1.133	1.197	1.277	1.376	1.497	1.646

$$Y^T =$$

	0	1	2	3	4	5	6	7	8	9	10
0	1	1.005	1.02	1.046	1.083	1.133	1.197	1.277	1.376	1.497	1.646

$$Y_{\text{mod}} := Y$$

Рисунок 1.1 – Розв’язання задачі Коші методами Ейлера

Вдосконалений метод Ейлера

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} := \begin{pmatrix} x_{i-1} + h \\ y_{i-1} + \frac{h}{2} \cdot (f(x_{i-1}, y_{i-1}) + f(x_{i-1} + h, y_{i-1} + h \cdot f(x_{i-1}, y_{i-1}))) \end{pmatrix}$$

$$Y := \begin{cases} Y_0 \leftarrow y_0 \\ \text{for } i \in 1..N \\ Y_i \leftarrow Y_{i-1} + \frac{h}{2} \cdot (f(x_{i-1}, Y_{i-1}) + f(x_{i-1} + h, Y_{i-1} + h \cdot f(x_{i-1}, Y_{i-1}))) \\ Y \end{cases}$$

$$y^T =$$

	0	1	2	3	4	5	6	7	8	9	10
0	1	1.005	1.02	1.046	1.083	1.133	1.197	1.277	1.377	1.499	1.648

$$Y^T =$$

	0	1	2	3	4	5	6	7	8	9	10
0	1	1.005	1.02	1.046	1.083	1.133	1.197	1.277	1.377	1.499	1.648

$$Y_{vip} := Y$$

Подамо результат графічно:

i := 0..10

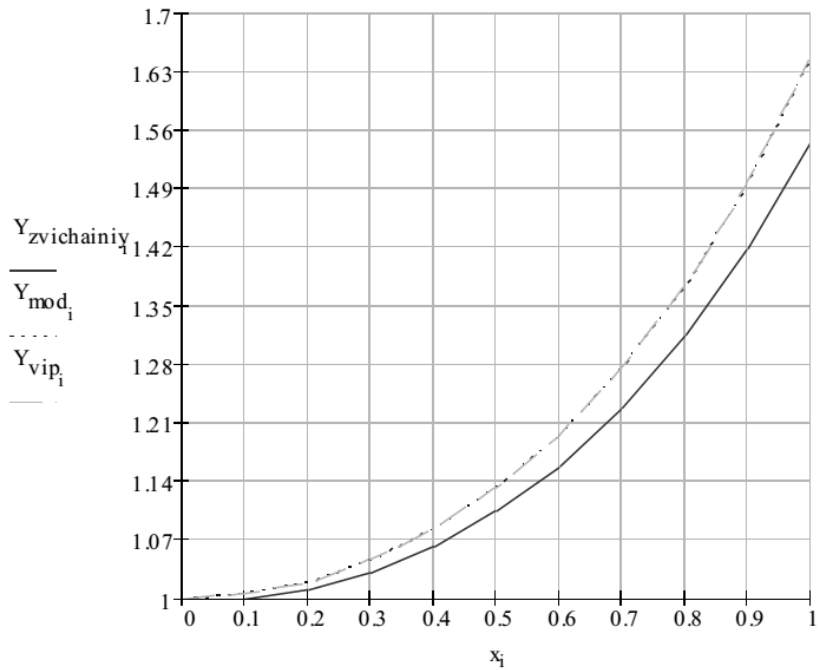


Рисунок 1.1, аркуш 2

Неявний метод Ейлера

$$g(h, x, y) := \frac{y}{1 - h \cdot x}$$

$$Y_0 := y_0$$

$$Y := \begin{cases} \text{for } i \in 0..N-1 \\ y_{i+1} \leftarrow g(h, x_i, y_i) \\ y \end{cases}$$

$$Y^T =$$

	0	1	2	3	4	5	6	7	8	9	10
0	1	1	1.01	1.031	1.063	1.107	1.165	1.239	1.333	1.449	1.592

$$Y_{\text{neyavniy}} := Y \quad i := 0..10$$

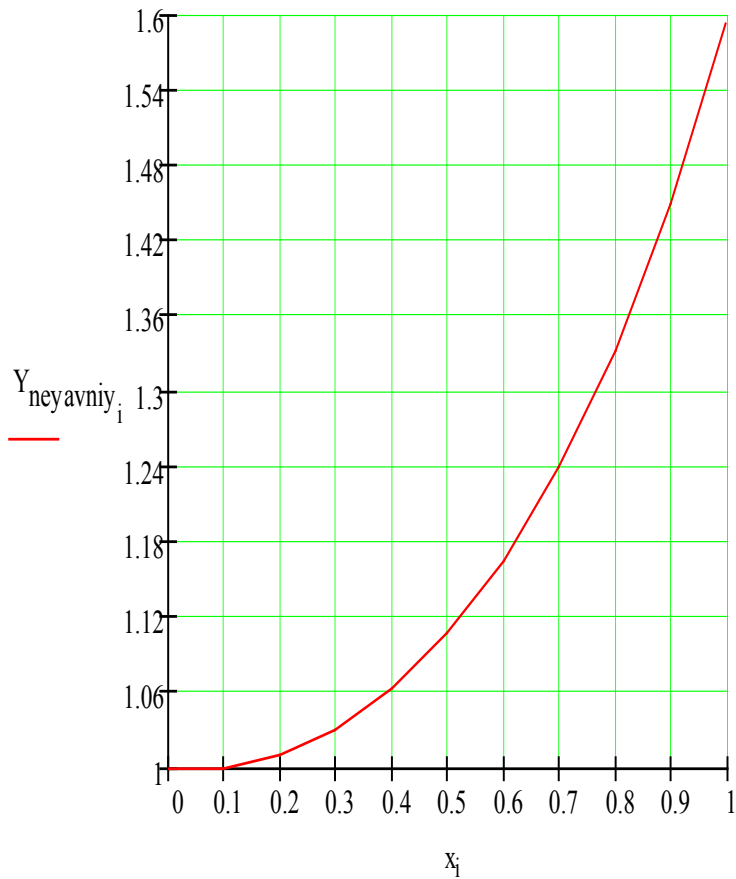


Рисунок 1.2 – Розв’язання задачі Коші неявним методом Ейлера

Початкові умови:

$$f(x, y) := xy \quad a := 0 \quad b := 1 \quad N := 10 \quad h := \frac{b - a}{N}$$

Формули для коефіцієнтів Рунге-Кутта:

$$k1(x, y) := h \cdot f(x, y)$$

$$k2(x, y) := h \cdot f\left(x + \frac{h}{2}, y + \frac{k1(x, y)}{2}\right)$$

$$k3(x, y) := h \cdot f\left(x + \frac{h}{2}, y + \frac{k2(x, y)}{2}\right)$$

$$k4(x, y) := h \cdot f(x + h, y + k3(x, y))$$

$$k(x, y) := k1(x, y) + 2 \cdot k2(x, y) + 2 \cdot k3(x, y) + k4(x, y)$$

Початкові значення для x та y:

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Ітераційні формули методу Рунге-Кутта в векторній формі

$$i := 1..N$$

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} := \begin{pmatrix} x_{i-1} + h \\ y_{i-1} + \frac{k(x_{i-1}, y_{i-1})}{6} \end{pmatrix}$$

$$x^T =$$

	0	1	2	3	4	5	6	7	8	9	10	
	0	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

$$y^T =$$

	0	1	2	3	4	5	6	7	8	9	
	0	1	1.005	1.02	1.046	1.083	1.133	1.197	1.278	1.377	1.499

Подамо результат графічно:

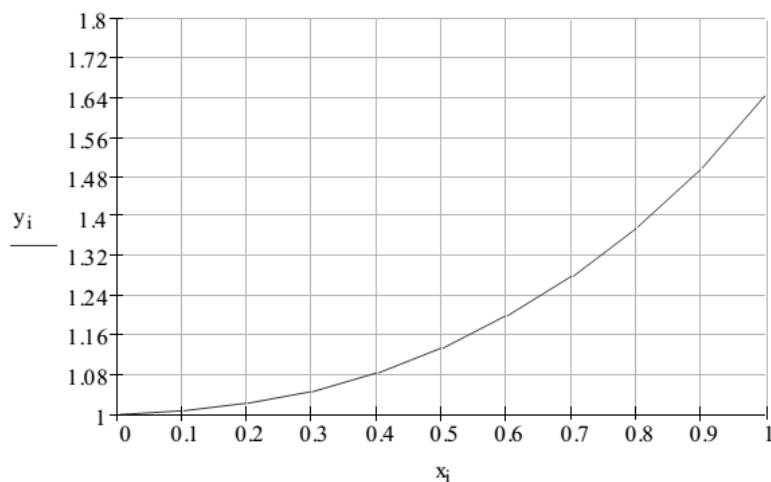


Рисунок 1.3 – Розв’язання задачі Коші методом Рунге-Кутта четвертого порядку

Початкові дані:

$$f(x, y) := x \cdot y \quad a := 0 \quad b := 1 \quad y_0 := 1 \quad n(k) := k \quad h(k) := \frac{b-a}{k} \quad Y_0 := y_0$$

Розбиття сітки:

$$X(k) := \begin{cases} X_0 \leftarrow a \\ \text{for } i \in 1..n(k) \\ X_i \leftarrow a + h(k) \cdot i \\ X \end{cases}$$

Формули Рунге-Кутта:

$$Y(k) := \begin{cases} Y_0 \leftarrow y_0 \\ \text{for } i \in 1..k \\ \begin{cases} F1 \leftarrow f(X(k)_{i-1}, Y_{i-1}) \\ F2 \leftarrow f\left(X(k)_{i-1} + \frac{h(k)}{2}, Y_{i-1} + \frac{h(k)}{2} \cdot F1\right) \\ F3 \leftarrow f\left(X(k)_{i-1} + \frac{h(k)}{2}, Y_{i-1} + \frac{h(k)}{2} \cdot F2\right) \\ F4 \leftarrow f(X(k)_{i-1} + h(k), Y_{i-1} + h(k) \cdot F3) \\ Y_i \leftarrow Y_{i-1} + \frac{h(k)}{6} \cdot (F1 + 2 \cdot F2 + 2 \cdot F3 + F4) \end{cases} \\ Y \end{cases}$$

$$Y(10)^T =$$

	0	1	2	3	4	5	6	7	8	9	10
0	1	1.005	1.02	1.046	1.083	1.133	1.197	1.278	1.377	1.499	1.649

Графік отриманих розв'язків:

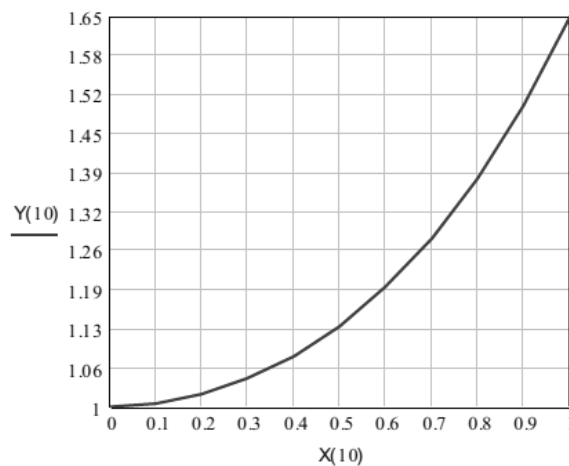


Рисунок 1.4 – Розв'язання задачі Коші методом Рунге-Кутта четвертого порядку



Нагадаємо, що для використання неявного методу Ейлера необхідно виконати перетворення заданого рівняння до вигляду: $y_{i+1} = g(h, x_i, y_i)$.

Для функції $f(x, y) = x \cdot y$ формула для використання неявного методу Ейлера набуде такого вигляду:

$$y_{n+1} = y_n + h \cdot x_n y_{n+1},$$

звідки

$$y_{n+1} = \frac{y_n}{1 - h \cdot x_n}.$$

1.2 Багатокрокові методи розв'язання задачі Коші

Багатокрокові методи, які часто називаються *методами Адамса*, побудовані шляхом інтерполяції за декількома сусідніми точками, для їх використання необхідно знати значення функції $y(x)$ в декількох попередніх точках. Це дещо ускладнює процедуру їх застосування порівняно з однокроковими, які мають так звану властивість «самостартування» («selfstarting»).

Метод Адамса другого порядку записується так:

$$y_{i+1} = y_i + \frac{h(3y'_i - y'_{i-1})}{2}, \quad (1.12)$$

де

$$y'_i = f(x_i, y_i). \quad (1.13)$$

Ітераційне рівняння для методу Адамса третього і четвертого порядків мають вигляд:

$$y_{i+1} = y_i + \frac{h(23y'_i - 16y'_{i-1} + 5y'_{i-2})}{12}, \quad (1.14)$$

$$y_{i+1} = y_i + \frac{h(55y'_i - 59y'_{i-1} + 37y'_{i-2} - 9y'_{i-3})}{24}. \quad (1.15)$$

Похибка розв'язку знайденого багатокроковим методом оцінюється з виразу:

$$|y_i - y_i^*| < O(h^m),$$

де m – порядок методу.

Таким чином, метод Рунге-Кутта четвертого порядку і метод Адамса четвертого порядку мають однакову оцінку похибки, але метод Адамса потребує приблизно в 4 рази менших обчислень.



Реалізуємо багатокроковий метод Адамса на прикладі такої задачі Коші:

$$y' = xy, y(0) = 1 \text{ на відрізку } [0,1] \text{ з кроком } h = 0.1.$$

Особливістю використання цих методів є те, що для обчислення значення в наступній точці необхідно знати розв'язок в m попередніх, де m – порядок методу. На рисунку 1.5 подано програму розв'язання задачі багатокроковими методами Адамса.

Початкові умови

$$f(x,y) := x \cdot y \quad \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$a := 0 \quad b := 1 \quad N := 10 \quad h := \frac{b-a}{N} \quad i := 1..10$$

Знаходження значень в перших чотирьох точках вдосконаленим методом Ейлера

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} := \begin{pmatrix} x_{i-1} + h \\ y_{i-1} + \frac{h}{2} \cdot (f(x_{i-1}, y_{i-1}) + f(x_{i-1} + h, y_{i-1} + h \cdot f(x_{i-1}, y_{i-1}))) \end{pmatrix}$$

$$y^T = \begin{array}{c|cccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline 0 & & 1 & 1.005 & 1.02 & 1.046 & 1.083 & 1.133 & 1.197 & 1.277 & 1.377 & 1.499 & 1.648 \end{array}$$

$$Y_{\text{Euler}} := y$$

Метод Адамса другого порядку

$$i := 2..10 \quad y := Y_{\text{Euler}}$$

$$y_i := y_{i-1} + \frac{h}{2} \cdot (3 \cdot f(x_{i-1}, y_{i-1}) - f(x_{i-2}, y_{i-2}))$$

$$y^T = \begin{array}{c|cccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline 0 & & 1 & 1.005 & 1.02 & 1.046 & 1.083 & 1.132 & 1.195 & 1.274 & 1.372 & 1.492 & 1.639 \end{array}$$

$$Y_{\text{Adams2}} := y$$

Метод Адамса третього порядку

$$i := 3..10 \quad y := Y_{\text{Euler}}$$

$$y_i := y_{i-1} + \frac{h}{12} \cdot (23 \cdot f(x_{i-1}, y_{i-1}) - 16 \cdot f(x_{i-2}, y_{i-2}) + 5 \cdot f(x_{i-3}, y_{i-3}))$$

$$y^T = \begin{array}{c|cccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline 0 & & 1 & 1.005 & 1.02 & 1.046 & 1.083 & 1.133 & 1.197 & 1.277 & 1.376 & 1.498 & 1.646 \end{array}$$

$$Y_{\text{Adams3}} := y$$

Рисунок 1.5 – Розв'язання задачі Коші за допомогою багатокрокових методів

Метод Адамса четвертого порядка

$$i := 4..10 \quad y := Y_{\text{Euler}}$$

$$y_i := y_{i-1} + \frac{h}{24} \cdot (55 \cdot f(x_{i-1}, y_{i-1}) - 59 \cdot f(x_{i-2}, y_{i-2}) + 37 \cdot f(x_{i-3}, y_{i-3}) - 9 \cdot f(x_{i-4}, y_{i-4}))$$

$y^T =$	0	1	2	3	4	5	6	7	8	9	10	
	0	1	1.005	1.02	1.046	1.083	1.133	1.197	1.277	1.377	1.499	1.648

$$Y_{\text{Adams4}} := y$$

Подамо результат графічно:

$$i := 0..10$$

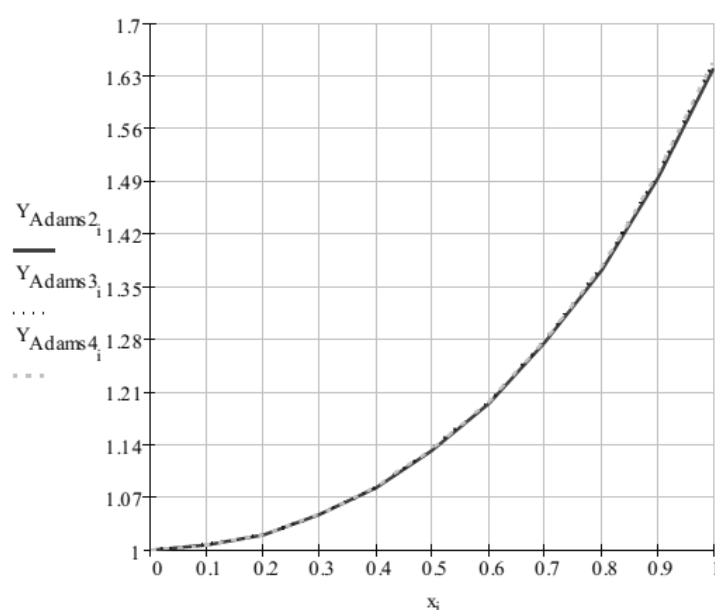


Рисунок 1.5, аркуш 2

Для знаходження розв'язку рівняння в перших чотирьох точках було використано вдосконалений метод Ейлера.

Як бачимо з графічного подання результатів розв'язання задачі Коші (рисунок 1.5), багатокрокові методи є більш точними за однокрокові, проте потребують обчислення розв'язку в попередніх точках.

1.3 Методи прогнозу і корекції

При отриманні формули Ейлера (1.5) було використано заміну похідної правою кінцевою різницею:

$$y'_i = \frac{(y_{i+1} - y_i)}{h},$$

якщо використати ліву кінцеву різницю

$$y_i' = \frac{(y_i - y_{i-1})}{h},$$

то отримаємо рівняння:

$$y_i - y_{i-1} - hf(x_i, y_i) = 0,$$

в яке величина y_i входить нелінійним чином.

Для розв'язання такого рівняння можна використовувати ітераційні методи.

Якщо задати початкове наближення:

$$y_i^{(0)} = y_{i-1}$$

і послідовно уточнювати значення y_i за формулою:

$$y_i^{(k+1)} = y_{i-1} + hf(x_i, y_i^{(k)}),$$

отримуємо неявний метод Ейлера.

Ітерації виконують до тих пір, поки не виконається умова:

$$\left| y_{i+1}^{(k+1)} - y_{i+1}^{(k)} \right| < \varepsilon.$$

Якщо після трьох або чотирьох ітерацій необхідна точність не досягається, то потрібно зменшити крок.

Реалізовані в такий спосіб багатокрокові методи називають *методами прогнозу і корекції*.

Методи прогнозу і корекції реалізуються так:

метод Адамса другого порядку:

$$y_{i+1}^{(0)} = y_i + \frac{h(3y_i' - y_{i-1}')}{2}, \quad (1.16, \text{a})$$

$$y_{i+1}^{(k+1)} = y_i + \frac{h(y_{i+1}'^{(k)} + y_i')}{2}, \quad (1.16, \text{б})$$

метод Адамса третього порядку:

$$y_{i+1}^{(0)} = y_i + \frac{h(23y_i' - 16y_{i-1}' + 5y_{i-2}')}{12}, \quad (1.17, \text{a})$$

$$y_{i+1}^{(k+1)} = y_i + \frac{h(5y_{i+1}'^{(k)} + 8y_i' - y_{i-1}')}{12}, \quad (1.17, б)$$

метод Адамса четвертого порядку:

$$y_{i+1}^{(0)} = y_i + \frac{h(55y_i' - 59y_{i-1}' + 37y_{i-2}' - 9y_{i-3}')}{24}, \quad (1.18, а)$$

$$y_{i+1}^{(k+1)} = y_i + \frac{h(9y_{i+1}'^{(k)} + 19y_i' - 5y_{i-1}' + y_{i-2}')}{24}. \quad (1.18, б)$$

В усіх цих формулах для обчислення $y_{i+1}'^{(k)}$ використовується значення y_{i+1} , яке отримане в поточному наближенні $y_{i+1}'^{(k)} = f(x_{i+1}, y_{i+1}^{(k)})$.

В методі Мілна використовуються формули прогнозу і корекції, які побудовані за різними точками:

$$y_i^{(0)} = y_{i-4} + \frac{4h(2y_{i-3}' - y_{i-2}' + 5y_{i-1}')}{3}, \quad (1.19, а)$$

$$y_{i+1}^{(k+1)} = y_{i-2} + \frac{h(y_{i-2}' + 4y_{i-1}' - y_i')^{(k)}}{3}. \quad (1.19, б)$$

1.4 Розв'язання рівнянь вищих порядків і систем диференціальних рівнянь

Задача Коші для звичайного диференціального рівняння вищого порядку задається в такий спосіб: знайти розв'язок рівняння

$$y^{(n)} = f(x, y^{(n-1)}, y^{(n-2)}, \dots, y', y) \quad (1.20)$$

яке задовольняє початкові умови:

$$\begin{aligned} y^{(n-1)}(x_0) &= y_0^{(n-1)}, \\ y^{(n-2)}(x_0) &= y_0^{(n-2)}, \\ &\dots \\ y'(x_0) &= y_0', \\ y(x_0) &= y_0. \end{aligned} \quad (1.21)$$

Шляхом введення таких додаткових змінних:

$$z = y, \quad z_1 = y', \quad z_2 = y^{(2)}, \quad z_3 = y^{(3)}, \dots, z_{n-1} = y^{(n-1)}, \quad (1.22)$$

можна перетворити рівняння (1.20) в систему рівнянь першого порядку:

$$\begin{aligned}
z'_{n-1} &= f(x, z_{n-1}, z_{n-2}, z_{n-3}, \dots, z', z), \\
z'_{n-2} &= z_{n-1}, \\
z'_{n-3} &= z_{n-2}, \\
&\dots \\
z'_1 &= z_2, \\
z' &= z_1,
\end{aligned}
\tag{1.23}$$

з початковими умовами:

$$\begin{aligned}
z_{n-1}(x_0) &= y_0^{(n-1)}, \\
z_{n-2}(x_0) &= y_0^{(n-2)}, \\
&\dots \\
z_1(x_0) &= y'_0, \\
z(x_0) &= y_0.
\end{aligned}
\tag{1.24}$$

Таким чином, задача Коші (1.20), (1.21) для диференціального рівняння n -го порядку еквівалентна задачі Коші (1.23), (1.24) для системи n диференціальних рівнянь першого порядку. Тому далі будемо розглядати тільки методи розв'язання систем диференціальних рівнянь.

Для розв'язання задачі Коші для системи диференціальних рівнянь можна використовувати ті ж методи, що і для одного рівняння, але всі етапи розв'язання потрібно виконувати за всіма змінними одночасно. Простіше подати формули, які реалізують відповідний метод для одного рівняння в векторній формі, а потім замінити векторну форму на алгебраїчну. Розглянемо це на прикладі задачі Коші для системи двох рівнянь:

$$\begin{aligned}
t' &= v(x, t, s), \quad t(x_0) = t_0, \\
s' &= w(x, t, s), \quad s(x_0) = s_0.
\end{aligned}
\tag{1.25}$$

Метод Рунге-Кутта другого порядку для одного рівняння можна записати у вигляді:

$$K_i = h \cdot f(x_i, y_i); \quad y_{i+1} = y_i + h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{K_i}{2}\right).$$

Запишемо його в векторній формі:

$$K_i = h \cdot f(x_i, y_i); \quad y_{i+1} = y_i + h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{K_i}{2}\right).$$

Відповідно до задачі Коші (1.25) вектори будуть складатись із компонентів:

$$y_i = \{t_i, s_i\}; \quad K_i = \{kt_i, ks_i\}; \quad f = \{v, w\}.$$

Якщо замінити векторну форму на алгебраїчну, отримаємо формули:

$$\begin{aligned}
kt_i &= h \cdot v(x_i, t_i, s_i), \\
ks_i &= h \cdot w(x_i, t_i, s_i), \\
k_{i+1} &= t_i + h \cdot v\left(x_i + \frac{h}{2}, t_i + \frac{kt_i}{2}, s_i + \frac{ks_i}{2}\right), \\
s_{i+1} &= s_i + h \cdot w\left(x_i + \frac{h}{2}, t_i + \frac{kt_i}{2}, s_i + \frac{ks_i}{2}\right).
\end{aligned}
\tag{1.26}$$

Метод Адамса другого порядку для задачі (1.25) реалізується в такий спосіб:

$$\begin{aligned}
t_{i+1} &= t_i + \frac{h \cdot (v(x_i, t_i, s_i) + v(x_{i-1}, t_{i-1}, s_{i-1}))}{2}, \\
s_{i+1} &= s_i + \frac{h \cdot (w(x_i, t_i, s_i) + w(x_{i-1}, t_{i-1}, s_{i-1}))}{2}.
\end{aligned}
\tag{1.27}$$

Всі етапи обчислень потрібно повністю провести для всіх змінних, і тільки після цього переходити до наступного етапу.

Розв'яжемо диференціальне рівняння другого порядку:



$$\begin{aligned}
x'' + w^2 x &= 0, \\
w = 5, \quad x(0) = 0, \quad x'(0) &= w
\end{aligned}
\tag{1.28}$$

на інтервалі $[0, 5]$ з кроком $h = 0.005$.

Перетворимо рівняння (1.28) на систему диференціальних рівнянь. Введемо заміну:

$$\begin{aligned}
z_0 &= x, \\
z_1 &= x'.
\end{aligned}
\tag{1.29}$$

Перепишемо рівняння (1.28) з врахуванням (1.29):

$$z_1' = -w^2 z_0$$

або у вигляді системи:

$$z_0' = z_1,$$

$$z_1' = -w^2 z_0$$

з початковими умовами:

$$z_0(0) = 0,$$

$$z_1(0) = w.$$

Розв'яжемо цю задачу за допомогою вдосконаленого методу Ейлера, а також за допомогою вбудованої функції *rkfixed* для перевірки отриманих рішень.

Програму розв'язання задачі (1.28) подано на рисунку 1.6.

```
t0 := 0    t1 := 5    n := 1000    h := (t1 - t0) / n    w := 5
f(x, y) := -w^2 * x    xn := 0    yn := w    h = 0.005
```

```
T := [ T0 ← xn
      for i ∈ 1..n
      Ti ← t0 + h·i
      T
      (X1) := [ X0 ← xn
              Y0 ← yn
              for i ∈ 1..n
              (Xi) ← [ Xi-1 + h·Yi-1
                    Yi-1 + h/2·(f(Xi-1, Yi-1) + f(Xi-1 + h, Yi-1 + h·f(Xi-1, Yi-1)))
              (X)
              (Y)
```

```
z := ( xn
      yn )    D(t, z) := ( z1
                        -w^2·z0 )    U := rkfixed(z, t0, t1, n, D)
```

```
k := 0..n
```

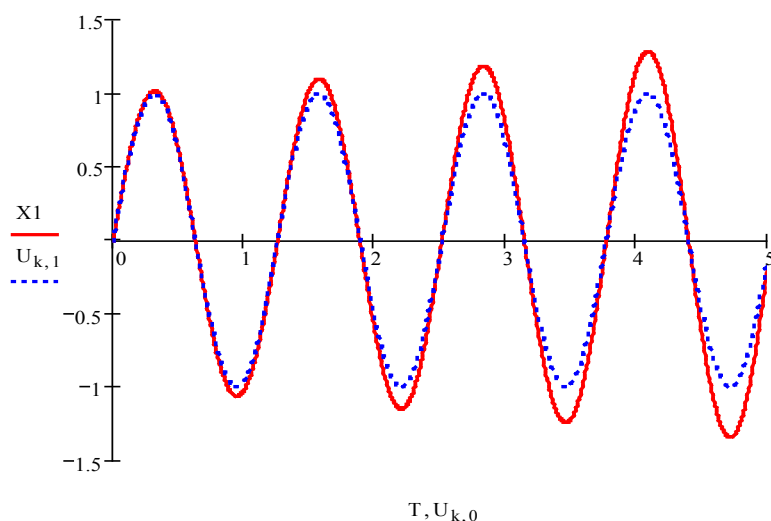


Рисунок 1.6 – Розв'язання системи диференціальних рівнянь вдосконаленим методом Ейлера

Як видно з рисунка 1.6, в останніх точках інтервалу інтегрування похибка збільшується. Це так звана *глобальна похибка поширення*, яка накопичується на кожному кроці і наприкінці інтервалу ця похибка може досягати значної величини.

Щоб продемонструвати послідовне накопичення похибки, змінимо значення параметра $w=10$. На рисунку 1.7 показано графічний розв'язок задачі в такому випадку. Добре видно, що похибка становить майже 100%. Це ще раз доводить, що використовувати метод Ейлера для точних розрахунків треба дуже обережно.

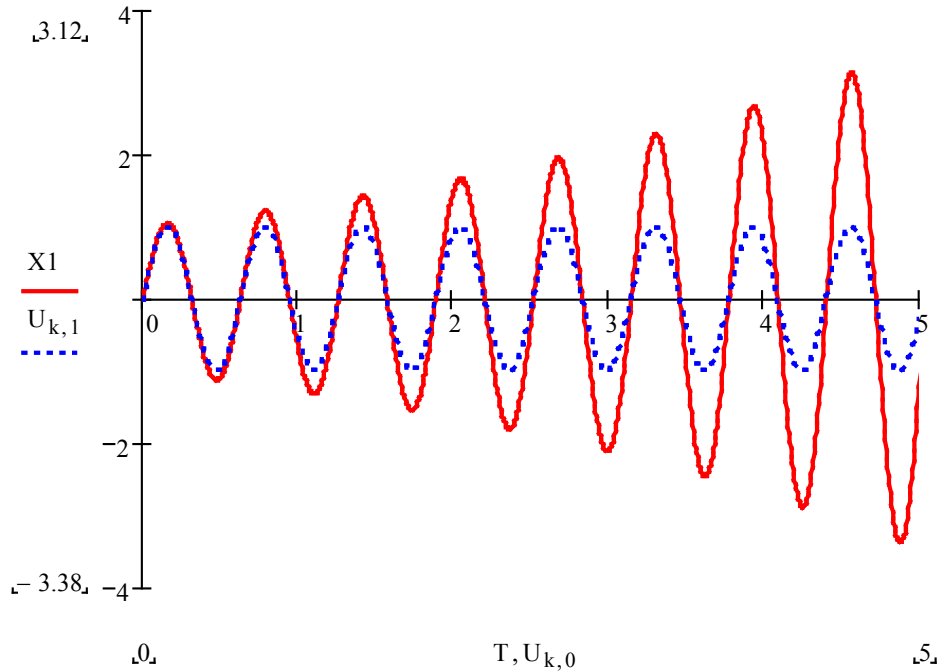


Рисунок 1.7 – Графічний результат розв’язання системи диференціальних рівнянь методом Ейлера ($w = 10$)

1.5 Оцінення похибки за правилом Рунге

Практично оцінити похибку чисельних методів дозволяє правило Рунге. Спочатку обчислюють наближений розв’язок з кроком h , потім – з кроком $\frac{h}{2}$. Тоді для метода Рунге-Кутта четвертого порядку точності справедлива наближена рівність:

$$y(x_{2i}) - (y_{2i})^{(\frac{h}{2})} = \frac{(y_{2i})^{(\frac{h}{2})} - (y_i)^{(h)}}{2^p - 1}, \quad (1.30)$$

де p – порядок методу;

$(y_i)^{(h)}$ – наближений розв’язок з кроком h ;

$(y_{2i})^{(\frac{h}{2})}$ – наближений розв’язок з кроком $\frac{h}{2}$.

Розв’яжемо цю ж саму задачу, яка наведена у п.1.4, методом Рунге-Кутта четвертого порядку. Програму розв’язання задачі подано на рисунку 1.8.

Початкові дані:

$$t_0 := 0 \quad t_1 := 5 \quad n := 1000 \quad h := \frac{t_1 - t_0}{n} \quad w := 5$$

$$f(x, y, z) := -w^2 \cdot y \quad x_n := 0 \quad y_n := w \quad h = 0.005$$

Обчислення коефіцієнтів:

$$K_1(x, y, z) := h \cdot f(x, y, z)$$

$$K_2(x, y, z) := h \cdot f\left(x + \frac{h}{2}, y + h \cdot \frac{z}{2} + h \cdot \frac{K_1(x, y, h)}{8}, z + \frac{K_1(x, y, z)}{2}\right)$$

$$K_3(x, y, z) := h \cdot f\left(x + \frac{h}{2}, y + h \cdot \frac{z}{2} + h \cdot \frac{K_1(x, y, z)}{8}, z + \frac{K_2(x, y, z)}{2}\right)$$

$$K_4(x, y, z) := h \cdot f\left(x + h, y + h \cdot z + h \cdot \frac{K_3(x, y, z)}{2}, z + K_3(x, y, z)\right)$$

$$S(x, y, z) := \frac{K_1(x, y, z) + 2 \cdot K_2(x, y, z) + 2 \cdot K_3(x, y, z) + K_4(x, y, z)}{6}$$

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} := \begin{pmatrix} 0 \\ 1 \\ w \end{pmatrix} \quad i := 1..n \quad \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} := \begin{pmatrix} x_{i-1} + h \\ y_{i-1} + h \cdot z_{i-1} \\ z_{i-1} + S(x_{i-1}, y_{i-1}, z_{i-1}) \end{pmatrix}$$

$$z := \begin{pmatrix} x_n \\ y_n \end{pmatrix} \quad D(t, z) := \begin{pmatrix} z_1 \\ -w^2 \cdot z_0 \end{pmatrix} \quad U := \text{rkfixed}(z, t_0, t_1, n, D)$$

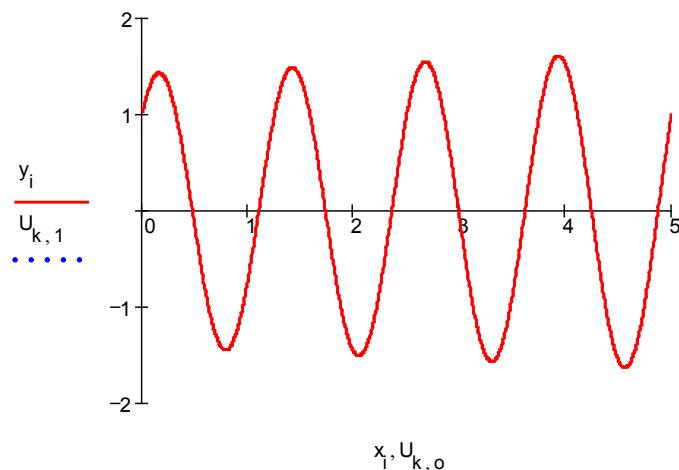


Рисунок 1.8 – Розв’язання системи диференціальних рівнянь методом Рунге-Кутта

Як видно з рисунка 1.8, графіки, отримані при реалізації методу Рунге-Кутта і за допомогою вбудованої функції *rkfixed*, майже збігаються.

За оцінку похибки розв’язку, обчисленого з кроком $\frac{h}{2}$, беруть величину:

$$\varepsilon = \frac{\left| \max [y(x_{2i}) - (y_{2i})^{\left(\frac{h}{2}\right)}] \right|}{2^p - 1}.$$



Розв'яжемо задачу Коші:

$$y' = xy, y(0) = 1$$

на інтервалі $[0,1]$ з кроками $h = 0.2$ і $h = 0.1$ і обчислимо похибку за правилом Рунге. Програму обчислення подано на рисунку 1.9.

Розв'язання з кроком $h=0.1$:

`ORIGIN := 1` `D(x,y) := x*y` `y1 := 1`

`Y1 := rkfixed(y, 0, 1, 10, D)`

$Y1^T =$		1	2	3	4	5	6	7	8	9	10
	1	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	2	1	1.005	1.02	1.046	1.083	1.133	1.197	1.278	1.377	1.499

Розв'язання з кроком $h=0.05$:

`Y2 := rkfixed(y, 0, 1, 20, D)`

$Y2^T =$		1	2	3	4	5	6	7	8	9	10	11
	1	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
	2	1	1.001	1.005	1.011	1.02	1.032	1.046	1.063	1.083	1.107	1.133

$$i := 1..10 \quad \varepsilon_i := \left| (Y1^{(2)})_i - (Y2^{(2)})_{2 \cdot i - 1} \right| \quad \max(\varepsilon) = 1.298 \times 10^{-7}$$

Рисунок 1.9 – Обчислення похибки за правилом Рунге

В нашому випадку похибка при обчисленні з кроком $h = 0.05$ не перевищує 0.071.

1.6 Розв'язання крайової задачі метод кінцевих різниць

Нехай дано лінійне диференціальне рівняння другого порядку:

$$y'' + p(x)y' + g(x)y = f(x), \quad (1.31)$$

де $p(x), g(x), f(x)$ – відомі неперервні на відрізку $[a, b]$ функції з граничними умовами:

$$\begin{aligned} a_0 y(a) + a_1 y'(a) &= A, \\ b_0 y(a) + b_1 y'(a) &= B, \end{aligned} \quad (1.32)$$

де a_0, a_1, b_0, b_1 – задані постійні коефіцієнти, причому:

$$|a_0| + |a_1| \neq 0, |b_0| + |b_1| \neq 0.$$

Одним із найпростіших методів розв'язання лінійної крайової задачі (1.31) та (1.32) є зведення її до системи різницевих рівнянь.

Суть цього методу в тому, що похідні замінюють деякими різницями.

Розіб'ємо відрізок $[a, b]$ на n рівних частин з кроком h ,

де

$$h = \frac{b-a}{n}.$$

Позначимо точки відрізка $[a, b]$ як:

$$x_0 = a, x_n = b, x_i = x_0 + ih \quad (i = 1..n-1),$$

$$p_i = p(x_i), g_i = g(x_i), f_i = f(x_i),$$

$$y_i = y(x_i), y'_i = y'(x_i), y''_i = y''(x_i).$$

Замінімо похідні в кожній внутрішній точці x_i інтервалу $[a, b]$ y'_i і y''_i різницевиими формулами:

$$\begin{aligned} y'_i &= \frac{y_{i-1} - y_i}{h}, \\ y''_i &= \frac{y_{i-1} - 2y_{i+1} + y_i}{h^2}. \end{aligned} \quad (1.33)$$

Для граничних точок $x_0 = a, x_n = b$ задамо додаткові умови:

$$\begin{aligned} y'_0 &= \frac{y_1 - y_0}{h}, \\ y'_n &= \frac{y_n - y_{n-1}}{h}. \end{aligned} \quad (1.34)$$

З використанням формул (1.33) і (1.34), рівняння (1.31) і крайові умови (1.32) запишуться у вигляді системи із $n+1$ лінійних алгебраїчних рівнянь з $n+1$ невідомими $y_0, y_1, y_2, \dots, y_n$, які відповідають значенням шуканої функції $y = y(x)$ в точках $x_0, x_1, x_2, \dots, x_n$:

$$\begin{aligned} \frac{y_{i+2} - 2y_{i+1} + y_i}{h^2} + \frac{p_i(y_{i+1} - y_i)}{h} + g_i y_i &= f_i, \\ i &= 0, \dots, n-2, \\ a_0 y_0 + a_1 \frac{(y_i - y_0)}{h} &= A, \\ b_0 y_n + b_1 \frac{(y_n - y_{n-1})}{h} &= B. \end{aligned} \tag{1.35}$$

Розв'язання цієї системи дає змогу одержати таблицю приблизних значень функції $y = y(x)$.

На практиці часто похідні y'_i і y''_i у внутрішніх точках x_i інтервалу $[a, b]$ замінюють центральними різницями:

$$\begin{aligned} y'_i &= \frac{y_{i+1} - y_{i-1}}{2h}, \\ y''_i &= \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}. \end{aligned} \tag{1.36}$$

Тоді система рівнянь для знаходження $y_0, y_1, y_2, \dots, y_n$ набуває вигляду:

$$\begin{aligned} \frac{y_{i+2} - 2y_{i+1} + y_i}{h^2} + \frac{p_i(y_{i+1} - y_i)}{2h} + g_i y_i &= f_i, \\ i &= 0..n-2, \\ a_0 y_0 + a_1 \frac{(y_i - y_0)}{h} &= A, \\ b_0 y_n + b_1 \frac{(y_n - y_{n-1})}{h} &= B. \end{aligned} \tag{1.37}$$

При великому n системи (1.35) або (1.37) стають дуже громіздкими. Для розв'язання систем такого виду розроблено спеціальний метод, що отримав назву *методу прогонки*, який використовується для тридіагональних систем вигляду (1.37).

Нехай маємо систему (1.35). Перетворивши її, отримуємо:

$$y_{i+2} + (-2 + hp_i)y_{i+1} + (1 - hp_i + hg_i)y_i = hf_i. \tag{1.38}$$

Введемо позначення:

$$\begin{aligned} M_i &= -2 + hp_i, \\ K_i &= 1 - hp_i + h^2 g_i, \\ i &= 0..n-2. \end{aligned} \tag{1.39}$$

Тоді (1.38) запишеться в вигляді:

$$y_{i+2} + M_i y_{i+1} + K_i y_i = h f_i. \quad (1.40)$$

Розв'язавши (1.40) відносно y_{i+1} , одержимо:

$$y_{i+1} = \frac{h^2 f_i}{M_i} - \frac{y_{i+2}}{M_i} - \frac{K_i y_i}{M_i}. \quad (1.41)$$

З врахуванням (1.35) рівняння можна записати у вигляді:

$$\begin{aligned} y_{i+1} &= C_i (d_i - y_{i+2}), \\ i &= 0..n-2, \end{aligned} \quad (1.42)$$

де C_i, d_i – деякі коефіцієнти.

Нехай $i=0$; тоді (1.41) набуде вигляду:

$$y_1 = h \frac{f_0}{M_0} - \frac{y_2}{M_0} - \frac{K_0 y_0}{M_0}. \quad (1.43)$$

Із крайової умови $a_0 y_0 + a_1 \frac{(y_i - y_0)}{h} = A$ знайдемо y_0 :

$$y_0 = \frac{Ah}{a_0 - h} - \frac{a_1 y_1}{a_0 h - a_1}$$

і підставимо його в (1.43). Після перетворень отримаємо:

$$y_1 = \frac{a_1 - a_0 h}{M_0 (a_1 - a_0 h) + K_0 D_1} \left[\left(\frac{K_0 Ah}{a_1 - a_0 h} + h^2 f_0 \right) - y_2 \right].$$

Позначимо:

$$\begin{aligned} C_0 &= \frac{a_1 - a_0 h}{M_0 (a_1 - a_0 h) + K_0 D_1}, \\ D_0 &= \frac{K_0 Ah}{a_1 - a_0 h} + h^2 f_0. \end{aligned} \quad (1.44)$$

Із (1.42) можна записати:

$$Y_i = C_{i-1} (d_{i-1} - Y_{i+1}).$$

Якщо підставити цей вираз в (1.39), одержимо:

$$Y_{i+2} + M_i \cdot Y_{i+1} + K_i \cdot C_{i-1} (d_{i-1} - Y_{i+1}) = h \cdot F_i.$$

Звідки:

$$y_{i+1} = \frac{[(h \cdot f_i - K_i \cdot C_{i-1} \cdot D_{i-1})]}{(M_i - K_i C_{i-1})}. \quad (1.45)$$

З (1.42) і (1.45), одержимо рекурентні формули для знаходження C_i і D_i :

$$\begin{aligned} C_i &= \frac{1}{(M_i - K_i C_{i-1})}, \\ D_i &= h f_i - K_i C_{i-1} D_{i-1}, \\ i &= 1, 2, \dots, n-2, \end{aligned}$$

де C_i, D_i – прогоночні коефіцієнти.

Метод прогонки складається з двох етапів: прямого і зворотного ходу. На першому етапі (прямий хід) на основі (1.43) знаходять коефіцієнти C_0 і D_0 . Після цього послідовно з використанням рекурентних формул (1.45) одержують значення C_i і D_i ($i=1, 2, \dots, n-2$). Другий етап (зворотний хід) починається зі знаходження y_n . Якщо використовувати другу крайову умову (1.35) і формулу (1.42) при $i=n-2$, отримаємо систему другорядних рівнянь:

$$\begin{aligned} B_0 y_n + B_1 (y_n - y_{n-1}) &= B; \\ y_{n-1} &= C_{n-2} (d_{n-2} - y_n). \end{aligned} \quad (1.46)$$

Розв'язання цієї системи відносно y_n , має такий вигляд:

$$y_{n-1} = \frac{(B_1 \cdot C_{n-2} \cdot D_{n-2} + B \cdot h)}{[B_1(1 + C_{n-2}) + B_0 h]}. \quad (1.47)$$

Підставивши в (1.47) уже знайдені прямим ходом C_{n-2}, D_{n-2} , отримуємо y_n . Після цього обчислюють $y_{n-1}, y_{n-2}, y_{n-3}, \dots, y_1$, послідовно використовуючи рекурентну формулу (1.42):

$$\begin{aligned} y_{n-1} &= C_{n-2} (D_{n-2} - y_n); \\ y_{n-2} &= C_{n-3} (D_{n-3} - y_{n-1}); \\ &\dots\dots\dots \\ y_1 &= C_0 (D_0 - y_2). \end{aligned} \quad (1.48)$$

Значення y_0 знаходимо за формулою, яка була одержана з першої крайової умови (1.35):

$$y_0 = \frac{(A_1 y_1 - A h)}{(A_1 - A_0 h)}. \quad (1.49)$$

Розв'яжемо лінійну крайову задачу:

$$p(x)y''(x) + q(x)y'(x) + r(x)y(x) = f(x),$$



де $p(x) = 1, q(x) = 0, r(x) = 0, f(x) = \frac{x}{2}, y' = xy, y(0) = 1$ на інтервалі $[0,4]$. Програма для розв'язання задачі подана на рисунку 1.10.

Початкові дані:

$$p(x) := 1 \quad q(x) := 0 \quad r(x) := 0 \quad f(x) := \frac{x}{2}$$

Границі інтервалу: $a := 0 \quad b := 4$

Граничні умови: $y_a := 0 \quad y_b := 0$

Розрахункові формули:

$$n(k) := k \quad h(k) := \frac{b-a}{n(k)}$$

Розбиття сітки:

$$X(k) := \begin{cases} X_0 \leftarrow a \\ \text{for } i \in 1..n(k) \\ X_i \leftarrow a + h(k) \cdot i \\ X \end{cases}$$

Формування тридіагональної матриці системи рівнянь:

$$A(k) := \begin{cases} \text{for } i \in 1..n(k) - 1 \\ A_i \leftarrow p(X(k)_i) - \frac{q(X(k)_i) \cdot h(k)}{2} \\ A \end{cases}$$

$$B(k) := \begin{cases} \text{for } i \in 1..n(k) - 1 \\ B_i \leftarrow -2 \cdot p(X(k)_i) + r(X(k)_i) \cdot h(k) \cdot h(k) \\ B \end{cases}$$

$$C(k) := \begin{cases} \text{for } i \in 1..n(k) - 1 \\ C_i \leftarrow p(X(k)_i) + \frac{q(X(k)_i) \cdot h(k)}{2} \\ C \end{cases}$$

$$D(k) := \begin{cases} \text{for } i \in 1..n(k) - 1 \\ D_i \leftarrow f(X(k)_i) \cdot h(k) \cdot h(k) \\ D_1 \leftarrow D_1 - A(k)_1 \cdot y_a \\ D_{n(k)-1} \leftarrow D_{n(k)-1} - C(k)_{n(k)-1} \cdot y_b \\ D \end{cases}$$

Рисунок 1.10 – Розв'язання крайової задачі методом кінцевих різниць

Розв'язання системи рівнянь методом прогонки:

$$\begin{aligned}
 Y(k) := & \left\{ \begin{array}{l} AA_1 \leftarrow -\frac{C(k)_1}{B(k)_1} \\ BB_1 \leftarrow \frac{D(k)_1}{B(k)_1} \\ \text{for } i \in 2..n(k)-2 \\ \left\{ \begin{array}{l} AA_i \leftarrow \frac{-C(k)_i}{B(k)_i + A(k)_i \cdot AA_{i-1}} \\ BB_i \leftarrow \frac{D(k)_i - A(k)_i \cdot BB_{i-1}}{B(k)_i + A(k)_i \cdot AA_{i-1}} \end{array} \right. \\ Y_{n(k)-1} \leftarrow \frac{D(k)_{n(k)-1} - A(k)_{n(k)-1} \cdot BB_{n(k)-2}}{A(k)_{n(k)-1} \cdot AA_{n(k)-2} + B(k)_{n(k)-1}} \\ \text{for } i \in n(k)-2..1 \\ Y_i \leftarrow AA_i \cdot Y_{i+1} + BB_i \\ Y_0 \leftarrow ya \\ Y_{n(k)} \leftarrow yb \\ Y \end{array} \right.
 \end{aligned}$$

Розв'язок задачі:

$$Y(10)^T =$$

	0	1	2	3	4	5	6	7	8	9	10
0	0	-0.528	-1.024	-1.456	-1.792	-2	-2.048	-1.904	-1.536	-0.912	0

Графічний розв'язок:

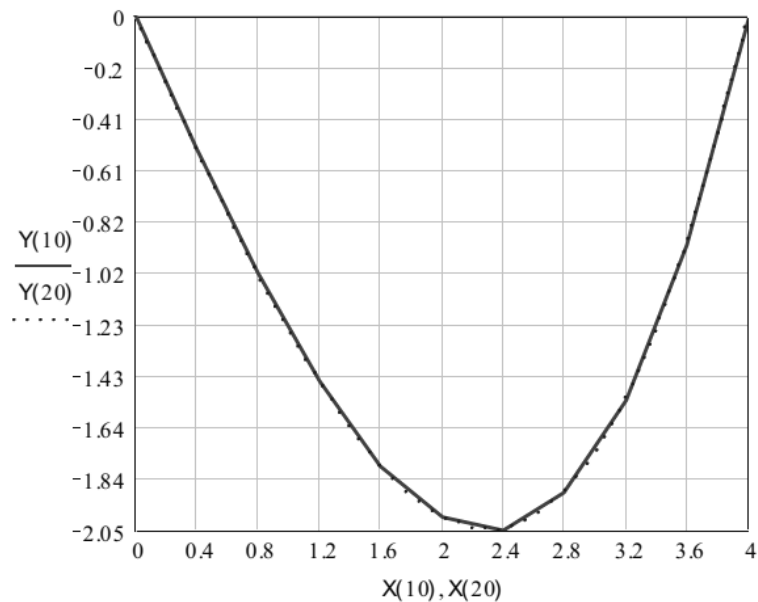


Рисунок 1.10, аркуш 2

Питання для самоконтролю

1. В чому суть задачі Коші?
2. Який метод є найпростішим чисельним методом для розв'язання ДР?
3. Назвіть назву групи методів, які найчастіше використовуються для розв'язання ДР.
4. Який однокроковий метод використовується для розв'язання «жорстких» ДР?
5. Чи доцільно використовувати метод Ейлера для точних розрахунків?
6. Які методи називаються методами Адамса?
7. Які методи називаються методами прогнозу і корекції?
8. Що таке глобальна похибка поширення?
9. За яким правилом проводиться оцінювання похибки розв'язку ДР?
10. Для чого використовують метод прогонки і з яких етапів він складається?

Задачі для самоконтролю:

1. Розв'язати задачі Коші за допомогою явних методів Ейлера:
 - a) $(2x + 1)y' = 4x + 2y$, $y(0) = 1$ на інтервалі $[0, 1]$;
 - b) $x^2y' + xy + 1 = 0$, $y(0) = 0$ на інтервалі $[0, 10]$.
2. Розв'язати задачі Коші за допомогою неявного методу Ейлера:
 - a) $y' = x + y^2$, $y(0) = 1$ на інтервалі $[0, 1]$;
 - b) $y' - xy + 1 = 0$, $y(0) = 0$ на інтервалі $[0, 2]$.
3. Розв'язати задачі Коші за допомогою методу Рунге-Кутта четвертого порядку:
 - a) $(x + y^2)dy = ydx$, $y(0) = 0$ на відрізку $[0, 1]$;
 - b) $y' + 2y = y^2e^x$, $y(0) = 5$ на відрізку $[0, 1]$.
4. Розв'язати задачі Коші за допомогою модифікованого методу Ейлера і методу Рунге-Кутта четвертого порядку. Порівняти отримані результати:
 - a) $x^2y' + xy + x^2y^2 = 4$, $y(0) = 0$ на відрізку $[0, 1]$;
 - b) $y' - 2xy + y^2 = 5 - x^2$, $y(0) = 3$ на відрізку $[0, 1]$.
5. Розв'язати задачі Коші за допомогою багатокрокових методів:
 - a) $y' = y^3 + 3x^2 - 1$, $y(1) = 1$ на відрізку $[1, 2]$;
 - b) $y' = y + e^{x-1}$, $y(0) = 1$ на відрізку $[0, 1]$.
6. Розв'язати задачі Коші за допомогою методів прогнозу і корекції:

a) $y' = 1 + x \sin(y)$, $y(\pi) = 1$ на відрізку $[\pi, 2\pi]$;

b) $y' - x = y^3$, $y(0) = 1$ на відрізку $[0, 2]$.

7. Розв'язати диференціальні рівняння другого порядку:

a) $y'' + y' - 2y = 0$, $y(0) = 0$, $y'(0) = 5$ на інтервалі $[0, 5]$;

b) $y'' - 2y' = 0$, $y(1) = 0$, $y(0) = 0$ на інтервалі $[0, 6]$.

8. Розв'язати системи диференціальних рівнянь:

a)
$$\begin{cases} x' = y, \\ y' = 2x + y, \end{cases} \quad y(0) = 0, \quad y'(0) = 1 \text{ на інтервалі } [0, 2];$$

b)
$$\begin{cases} x' = x + y, \\ y' = 2x - y^2, \end{cases} \quad y(0) = 0, \quad y'(0) = 1 \text{ на інтервалі } [0, 5].$$

9. Обчислити похибку за правилом Рунге:

a) $y' = 6x + y^2 + \sin(x)$, $y(0) = 1$ на інтервалі $[0, 5]$ з кроками $h = 0.2$ і $h = 0.1$;

b) $4y = x^2 + y'$, $y(0) = 1$ на інтервалі $[0, 1]$ з кроками $h = 0.05$ і $h = 0.1$;

10. Розв'язати крайові задачі методом кінцевих різниць:

a) $y'' - y = 2x$, $y(0) = 0$, $y(1) = -1$ на інтервалі $[0, 4]$;

b) $y'' + y' = 1$, $y'(0) = 0$, $y(1) = 1$ на інтервалі $[0, 7]$.

2 РОЗВ'ЯЗАННЯ ЗВИЧАЙНИХ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

В цьому розділі розглядаються основні вбудовані засоби MathCAD для розв'язання звичайних диференціальних рівнянь та їх систем, «жорстких» рівнянь. Розглядаються як задачі Коші, так і крайові задачі. Наведено опис функцій, рекомендації щодо використання та ряд прикладів.

Важливо підкреслити, що MathCAD вміє розв'язувати тільки такі системи диференціальних рівнянь, які можуть бути подані в стандартній формі (формі Коші).

В розділі розглядаються такі функції: *Odesolve*, *rkfixed*, *Rkapapt*, *rkadapt*, *Bulstoer*, *bulstoer*, *Stiffrr*, *stiffrr*, *Stiffb*, *stiffb*, *Radau*, *radau*, *statespace*, *Adams*, *BDF*, *AdamsBDF*.

2.1 Обчислювальний блок Given ... Odesolve

В бібліотеці вбудованих функцій MathCAD є функція *Odesolve*, що призначена для розв'язання лінійних диференціальних рівнянь.

За допомогою функції *Odesolve* розв'язують задачу Коші з початковими умовами:

$$y(x_0) = y_0, y'(x_0) = y_{0,1}, \dots, y^{(n-1)}(x_0) = y_{0,n-1} \quad (2.1)$$

та найпростішу крайову задачу, в якій задані n граничних умов у вигляді:

$$y^{(k)}(a) = y_{a,k}, y^{(m)}(b) = y_{b,m}, 0 \leq k \leq n-1, 0 \leq m \leq n-1. \quad (2.2)$$

Опис параметрів функції *Odesolve* подано в таблиці 2.1.

Таблиця 2.1 – Опис параметрів функції *Odesolve*

Формат виклику функції: <i>Given ... Odesolve(x,b,step)</i>	
Параметри	Опис параметра
b	Ім'я змінної інтегрування (аргументу шуканої функції).
b	Права границя інтервалу інтегрування.
$step$	Кількість кроків, що використовуються при інтегруванні (необов'язковий параметр, за його відсутності програма підбирає його автоматично). Чим більше значення $step$, тим точнішим буде отриманий результат, але тим більше часу буде витрачено на його пошук.

Використання функції *Odesolve* вимагає запису обчислювального блоку, який містить три частини:

1. Ключове слово *Given*;
2. Диференціальне рівняння та умови (початкові або граничні);
3. Безпосередньо виклик функції *Odesolve*.



При розв'язанні крайової задачі заборонено лінійні граничні умови, наприклад, $\alpha y(a) + \beta y'(a) = \gamma$, що визначають значення шуканої функції $y(x)$ і її похідних на кінцях інтервалу $[a, b]$.



При введенні рівняння і умов задачі використовується знак символічної рівності (*Ctrl*+*=*), а для запису похідних можна використовувати як оператор диференціювання, так і знак похідної. Наприклад, другу похідну можна ввести звичайним способом у вигляді $\frac{d^2}{dx^2}y(x)$ або у вигляді $y''(x)$, використовуючи комбінацію клавіш *Ctrl* + *F7*.



Розв'яжемо задачу Коші для лінійного диференціального рівняння $y''(x) + \cos(x)y'(x) = \frac{\pi}{x+1}$ з початковими умовами $y(0) = 0$, $y'(0) = 1$ за допомогою обчислювального блока *Given...Odesolve* на відріжку $[0, 2\pi]$. Розв'язання цієї задачі подано на рисунку 2.1.

Розв'язання лінійного диференціального рівняння

Вихідне рівняння: $y''(x) + \cos(x) \cdot y'(x) = \frac{\pi}{x+1}$

Початкові умови: $y(0) = 0$ $y'(0) = 1$

Given

$$y''(x) + \cos(x) \cdot y'(x) = \frac{\pi}{x+1}$$

$$y(0) = 0 \quad y'(0) = 1 \quad y := \text{Odesolve}(x, 2 \cdot \pi)$$

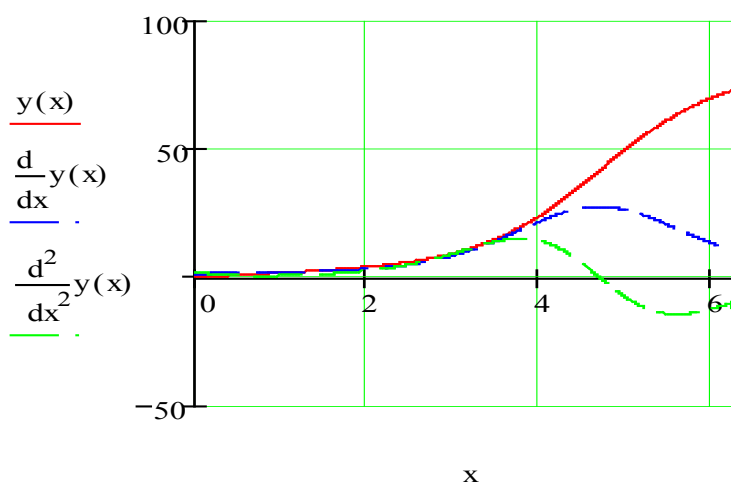


Рисунок 2.1 – Розв'язання лінійного диференціального рівняння за допомогою функції *Odesolve*



Розв'яжемо крайову задачу $y'' = y' \sin(x)$ з граничними умовами $y(0) = 0$, $y'(0) = 3$ за допомогою блока *Given ... Odesolve*. Розв'язання цієї задачі подано на рисунку 2.2.

Розв'язання диференціального рівняння з граничними умовами

Вихідне рівняння: $y'' = y' \sin(x)$

Початкові умови: $y(0) = 0$ $y'(0) = 3$

Given

$y''(x) = y'(x) \cdot \sin(x)$ $y(0) = 0$ $y(1) = 3$ $Y := \text{Odesolve}(x, 6)$

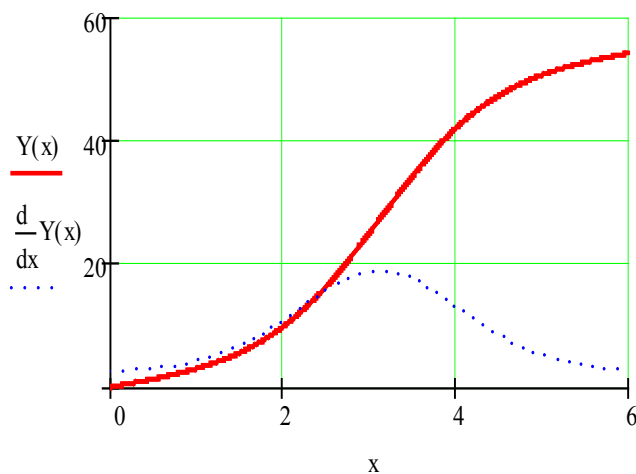


Рисунок 2.2 – Розв'язання крайової задачі за допомогою функції *Odesolve*

Якщо вигляд диференціального рівняння змінюється залежно від аргументу, то таке рівняння можна записати з відповідною умовою, тобто об'єднати декілька рівнянь в одне.



Розв'яжемо задачу Коші:

$$\begin{cases} y'' = \cos(y) - y'x^2, & 0 \leq x < 1, \\ y'' = \cos(y) + y'x^2, & 1 \leq x \leq 2, \end{cases}$$

з початковими умовами $y(0) = 1$, $y'(0) = 10$. Розв'язання задачі подано на рисунку 2.3.

Вихідне рівняння: $y'' = \cos(y) - y' \cdot x^2 \quad 0 \leq x < 1$

$y'' = \cos(y) + y' \cdot x^2 \quad 1 \leq x \leq 2$

Початкові умови: $y(0) = 1 \quad y'(0) = 10$

Given $y''(x) + y'(x) \cdot x^2 \cdot (x < 1) - y'(x) \cdot x^2 \cdot (x \geq 1) = \cos(y(x))$

$y(0) = 1 \quad y(2) = 10 \quad y := \text{Odesolve}(x, 2)$

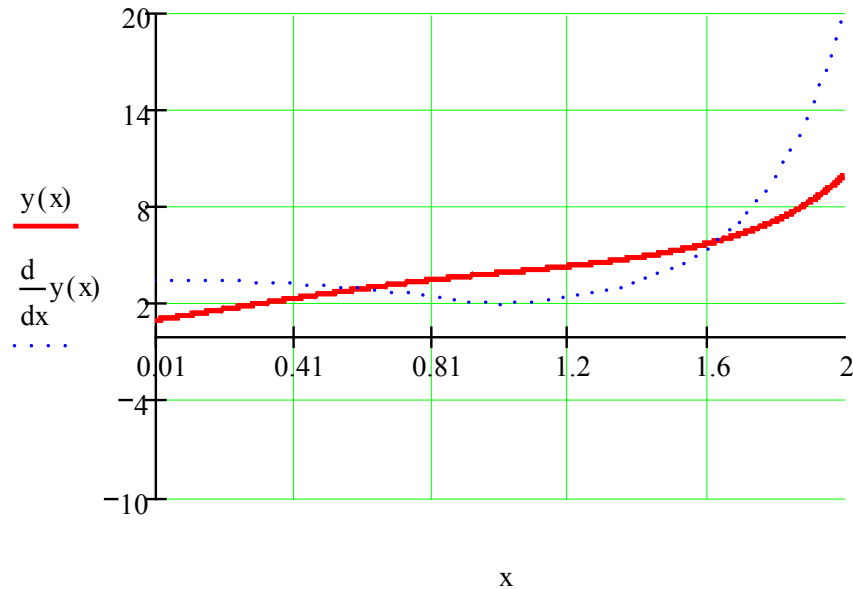


Рисунок 2.3 – Розв’язання двох диференціальних рівнянь, об’єднаних в одне рівняння

Якщо диференціальне рівняння (ДР) є складною функцією відносно вищої похідної, то для його розв’язання необхідно спочатку знайти розв’язок відносно цієї функції в символьній формі.

Для цього головну змінну необхідно виділити і в головному меню вибрати *Symbolics* → *Variable* → *Solve* (Символіка → Змінна → Рішення). Знайдений таким чином результат необхідно вставити як диференціальне рівняння.



Розглянемо задачу Коші $y'(x) + y'(x)(x+1) = 0$ з початковою умовою $y(0) = 0$ на відрізку $[0; 2]$. На рисунку 2.4 подано знаходження символьного розв’язку рівняння відносно $y'(x)$ і розв’язання задачі Коші.

Розв'язання диференціального рівняння відносно $y'(x)$

Вихідне рівняння: $y'(x) + y'(x) \cdot (x + 1) = 0$

Початкові умови: $y(0) = 0$

Розв'яжемо вихідне рівняння символно:

$$y'(x) + y'(x) \cdot (x + 1) = 0 \quad a + a \cdot (x + 1) = 1 \quad \frac{1}{2 + x}$$

Знайдений вираз підставимо в обчислювальний блок **Given...Odesolve**:

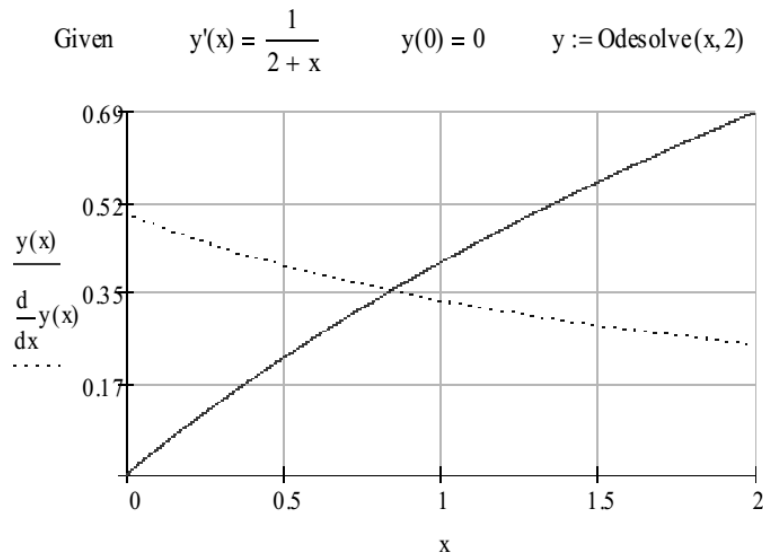


Рисунок 2.4 – Розв'язання диференціального рівняння символно відносно $y'(x)$

Починаючи із версії MathCAD 2001i за допомогою функції *Odesolve* можна розв'язувати системи диференціальних рівнянь. При цьому звернення до функції здійснюється у вигляді:

$$\text{Odesolve}(\text{вектор невідомих функцій}), x, b, [\text{step}]).$$



Розв'яжемо систему з двох диференціальних рівнянь першого порядку:

$$\begin{cases} x'(t) = x(t)^2 - y(t), \\ y'(t) = x(t)^2 + y(t). \end{cases}$$

з початковими умовами $x(1) = 0$, $y(1) = 5$ на відрізку $[0; 2]$.

В цьому випадку вектор невідомих значень містить дві змінні – x , y , які ми подаємо на вхід функції. Все інше записується аналогічно попереднім прикладам. Розв'язання цієї задачі подано на рисунку 2.5.

Розв'язання системи диференціальних рівнянь

Вихідна система: $x'(t) = x(t)^2 - y(t)$
 $y'(t) = x(t)^2 + y(t)$

Початкові умови: $x(1) = 0$ $y(1) = 5$

Given $x'(t) = x(t)^2 - y(t)$ $y'(t) = x(t)^2 + y(t)$
 $x(1) = 0$ $y(1) = 5$

$$\begin{pmatrix} x \\ y \end{pmatrix} := \text{Odesolve}\left[\begin{pmatrix} x \\ y \end{pmatrix}, t, 2\right]$$

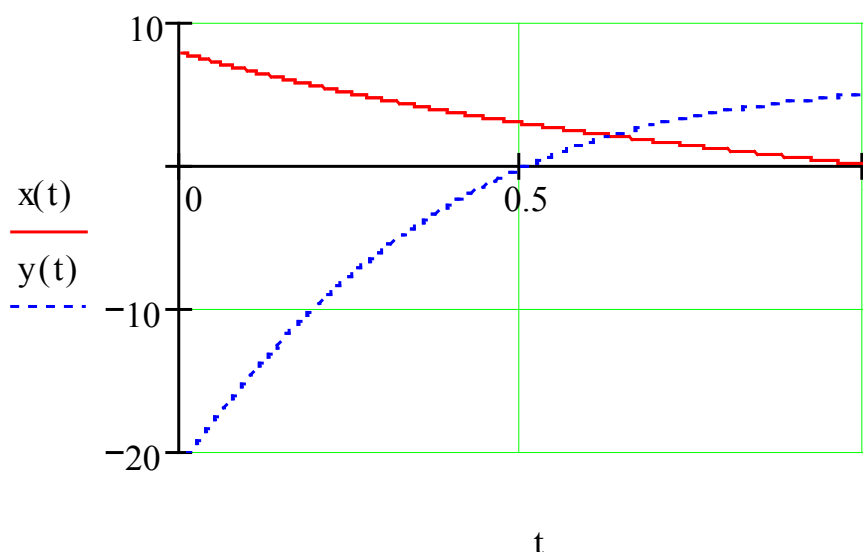


Рисунок 2.5 – Розв'язання системи ДР за допомогою функції *Odesolve*

В MathCAD 2001i як обмеження функції *Odesolve* можна вказувати алгебраїчні рівняння, проте лише в тому випадку, якщо обраний спосіб розв'язання «жорстких» систем ДР (для цього в контекстному меню функції необхідно активізувати команду *Stiff*) і в розрахунок взято рівняння Лагранжа (рисунок 2.6).

Розв'язання системи диференціальних рівнянь з обмеженням у вигляді алгебраїчних рівнянь

$s := 10$ $m := 1$ $g := 9.8$ $end := 30$ $t := 0, 0.1..end$

Given $m \cdot \frac{d^2}{dt^2} x(t) = -2 \cdot x(t) \cdot \lambda(t)$ $m \cdot \frac{d^2}{dt^2} y(t) = -m \cdot g - 2 \cdot y(t) \cdot \lambda(t)$

$x(t)^2 + y(t)^2 = s^2$ $x(0) = 0$ $x'(0) = 1$ $y(0) = -s$ $y'(0) = 0$ $\lambda(0) = \frac{m(1 + s \cdot g)}{2 \cdot s^2}$

Для функції `Odesolve` має бути активований режим `Stiff`

$\begin{pmatrix} x \\ y \\ \lambda \end{pmatrix} := \text{Odesolve} \left[\begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, t, end \right]$

$g(t) := x(t)^2 + y(t)^2 - s^2$

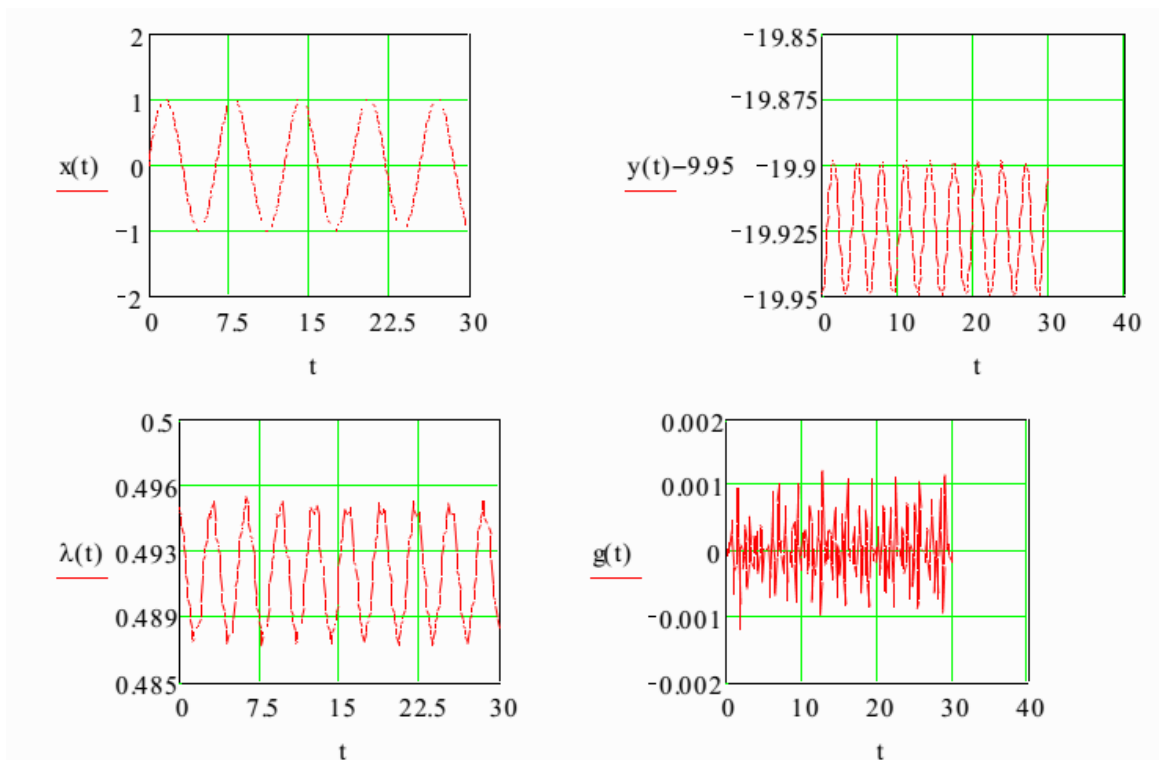


Рисунок 2.6 – Розв'язання системи диференціальних рівнянь з обмеженнями у вигляді алгебраїчних рівнянь

В процесі розв'язання функція `Odesolve` перетворює задані диференціальні рівняння до загального вигляду і розв'язує їх за допомогою функцій `rkfixed` і `Rkadapt` (ці функції буде розглянуто нижче). В результаті отримують масив значень функції та її похідних.

Далі використовуються функції інтерполяції `lspline` і `interp`, які перетворюють масив значень розв'язку системи звичайних диференціальних рівнянь на функцію, яку потім можна диференціювати або інтегрувати.

Необов'язковий параметр *step* задає кількість точок інтерполяції. Чим більше значення *step*, тим більша точність інтерполяції, та, відповідно, більший час розв'язання задачі.

При розв'язанні системи звичайних диференціальних рівнянь є можливість вибирати метод розв'язання (рисунок 2.7, а). Для цього необхідно натиснути правою кнопкою миші на імені функції *Odesolve* і в контекстному меню вибрати команду *Fixed* (розв'язання з фіксованим кроком за допомогою функції *rkfixed*), *Adaptive* (розв'язання з автоматичним вибором кроку за допомогою функції *Rkadapt*) або *Stiff* (розв'язання «жорсткої» системи за допомогою функції *Radau*). В MathCAD 14 назву *Stiff* було замінено на *Radau*.

В останній версії (MathCAD 14) список алгоритмів (рисунок 2.7, б), які використовують при реалізації функції *Odesolve*, поповнився методами Адамса (*Adams*) та *BDF* (*backward differentiation formula*). Причому, при виборі цих методів MathCAD буде сам динамічно визначати, який метод чисельного інтегрування використовувати для розв'язання рівнянь – *Adams* (для звичайних рівнянь) чи *BDF* (для «жорстких»).

Методи *Adams*, *BDF* і *Adams/BDF* (комбінація цих методів) оформлені також і у вигляді нових вбудованих функцій, які описані нижче.

Таким чином, функція *Odesolve* об'єднала всі основні функції розв'язання звичайних диференціальних рівнянь, що існують на цей момент.

Якщо граничні умови задані в різних точках, то на початку розв'язання функція *Odesolve* зводить задачу з граничними умовами до задачі з початковими умовами. Для цього використовується функція *sbval* (див. розділ 1.6). Далі розв'язується задача залежно від обраного користувачем методу (*Fixed*, *Adaptive* або *Radau*).

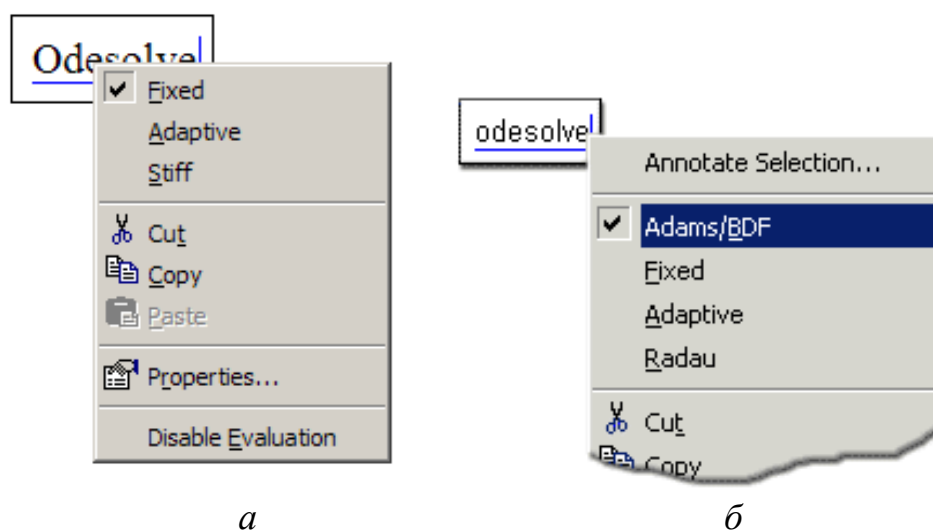


Рисунок 2.7 – Контекстне меню функції *Odesolve*: а – MathCAD 200i; б – MathCAD 14

2.2 Функції *rkfixed*, *Rkadapt* і *Bulstoer*

Функції *rkfixed*, *Rkadapt* і *Bulstoer* призначені для розв'язання диференціальних рівнянь та їх систем. Вони використовуються для розв'язання задачі Коші і не призначені для розв'язання крайової задачі.

Усі три функції мають однакові вхідні параметри, опис яких подано в таблиці 2.2.

Таблиця 2.2 – Опис параметрів функцій *rkfixed*, *Rkadapt* і *Bulstoer*

Формат виклику функцій: <i>rkfixed</i> ($y, x1, x2, npoints, D$) <i>Rkadapt</i> ($y, x1, x2, npoints, D$) <i>Bulstoer</i> ($y, x1, x2, npoints, D$)	
Параметри	Опис параметра
y	Початкова умова або вектор початкових умов (для систем рівнянь).
$x1, x2$	Відповідно початкова і кінцева точки інтервалу інтегрування системи.
$npoints$	Кількість вузлів на інтервалі $[x_1, x_2]$, при розв'язанні задачі результат містить $(npoints + 1)$ рядок.
$D(x, y)$	Функція, що містить у вигляді вектора перші похідні невідомих функцій. Результат роботи функції – матриця, що містить $n + 1$ стовпець; її перший стовпець містить координати вузлів сітки, другий стовпець – знайдені наближені значення розв'язку $y_1(x)$ у вузлах сітки, $(k + 1)$ -й – значення розв'язку $y_k(x)$.

При розв'язанні задачі Коші для диференціального рівняння першого порядку результати обчислень всіх наведених вище функцій – матриця, в першому стовпці якої містяться координати вузлів сітки x_0, x_1, \dots, x_n , а в другому – значення наближеного розв'язку у відповідних вузлах.

Функція *rkfixed* реалізує метод Рунге-Кутта четвертого порядку з фіксованим кроком інтегрування. Цей метод не завжди є найшвидший, проте майже завжди знаходить розв'язок задачі.

Функція *Rkadapt* реалізує метод Рунге-Кутта з автоматичним вибором кроку. Вона перевіряє, як швидко змінюється розв'язок і адаптує відповідно крок інтегрування. Це дає можливість функції обчислювати наближене значення рівняння з більшим кроком там, де розв'язок змінюється повільно, і з меншим кроком там, де розв'язок змінюється швидко. Як правило, ця функція використовується при розв'язанні «жорстких» рівнянь і систем рівнянь.

Якщо відомо, що вихідна функція є гладкою, доцільно використовувати функцію *Bulstoer*, яка реалізує метод Булірша-Штера. Розв'язок, знайдений для такої функції, буде точнішим за розв'язки, які отримані за допомогою інших функцій.

2.2.1 Диференціальні рівняння першого порядку

При розв'язанні задачі Коші для рівняння першого порядку вектор розв'язку має єдину компоненту y_1 . Вектор правих частин також містить одну компоненту. Перед зверненням до функції розв'язання диференціального рівняння необхідно присвоїти змінній y_1 значення початкової умови, а змінній $D(x, y)$ – вираз для правої частини рівняння.



Для того, щоб нумерація елементів в масиві y починалася з одиниці, на початку документа необхідно вказати `ORIGIN:=1`, в іншому випадку нумерація почнеться з нуля і тоді необхідно буде використовувати запис y_0 .



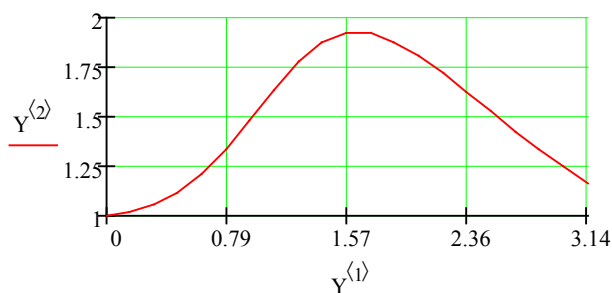
Розглянемо задачу Коші $y' = \cos(xy)$ з початковою умовою $y(0) = 1$ на відрізку $[0, \pi]$. Результати обчислень функції *rkfixed* присвоїмо матриці Y . Перший стовпець матриці Y містить координати 20 вузлів рівномірної сітки на відрізку $[0, \pi]$, а другий – наближені значення розв'язку в цих вузлах. Розв'язання цієї задачі подано на рисунку 2.8.

Розв'язання диференціального рівняння першого порядку

Початкове рівняння: $y' = \cos(xy)$

Початкові умови: $y(0) = 1$

`ORIGIN := 1` $y' = \sin(xy)$ $y_1 := 1$ $D(x, y) := \sin(xy_1)$
`Y := rkfixed(y, 0, pi, 20, D)`



$Y =$

	1	2
1	0	1
2	0.157	1.012
3	0.314	1.05
4	0.471	1.115
5	0.628	1.208
6	0.785	1.331
7	0.942	1.477
8	1.1	1.633
9	1.257	1.774
10	1.414	1.874
11	1.571	1.921

Рисунок 2.8 – Розв'язання диференціального рівняння першого порядку за допомогою функції *rkfixed*

Якщо рівняння записане в нелінійному вигляді, то його потрібно спочатку розв'язати алгебраїчно відносно вищої похідної (в цьому випадку відносно першої). Це можна зробити вручну або в символному вигляді і підставити результат в функцію $D(x, y)$.



Принцип використання функцій `Rkadapt` і `Bulstoer` аналогічний використанню функції `rkfixed`, тому далі всі приклади будемо наводити для неї.

2.2.2 Диференціальні рівняння n -го порядку

В загальному випадку диференціальне рівняння n -го порядку має вигляд:

$$y^{(n)} = f(x, y^{(n-1)}, y^{(n-2)}, \dots, y', y). \quad (2.3)$$

Виконаємо заміну:

$$\begin{aligned} y_1(x) &= y(x), \\ y_2(x) &= y'(x), \\ &\dots \\ y_n(x) &= y^{(n-1)}(x). \end{aligned} \quad (2.4)$$

Після заміни диференціальне рівняння може бути записане у вигляді системи диференціальних рівнянь:

$$\begin{aligned} y_1'(x) &= y_2(x), \\ y_2'(x) &= y_3(x), \\ &\dots \\ y_n'(x) &= f(x, y^{(n-1)}, y^{(n-2)}, \dots, y', y). \end{aligned} \quad (2.5)$$

Для того, щоб розв'язати диференціальне рівняння n -го порядку, необхідно задати вектор з n елементів, що визначають початкові умови для шуканої функції і її похідних (вектор початкових умов):

$$y = \begin{pmatrix} y_2(a) \\ y_3(a) \\ \dots \\ y_n^{(n-1)}(a) \end{pmatrix},$$

де a – початкова точка інтегрування,

y_i – праві частини системи (2.5).

Далі необхідно ввести функцію $D(x, y)$ (яка є вектором), що містить n елементів:

$$D(x, y) = \begin{pmatrix} y_2(x) \\ y_3(x) \\ \dots \\ f(x, y^{(n-1)}, y^{(n-2)}, \dots, y', y) \end{pmatrix}$$

і ввести вираз, що містить функцію *rkfixed*, *Rkadapt* або *Bulstoer*.

В результаті розв'язання отримуємо матрицю, яка містить $n + 1$ стовпців: в першому містяться значення аргументу x , в другому – значення функції $y(x)$, а в тих, що залишилися – значення $n - 1$ похідних функції $y(x)$.



Розв'яжемо диференціальне рівняння другого порядку $y'' = y \sin(x)$ з початковими умовами $y(0) = y'(0) = 1$ на відрізку $[0, 1]$ за допомогою функції *rkfixed*. Розв'язання задачі подано на рисунку 2.9.

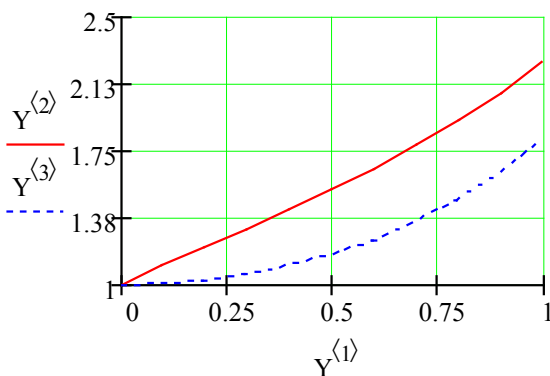
Розв'язання диференціального рівняння другого порядку

Початкове рівняння: $y'' = y \cdot \sin(x)$

Початкові умови: $y(0) = 1 \quad y'(0) = 1$

ORIGIN := 1 $D(x, y) := \begin{pmatrix} y_2 \\ y_1 \cdot \sin(x) \end{pmatrix}$

$y := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $Y := \text{rkfixed}(y, 0, 1, 10, D)$



	1	2	3
1	0	1	1
2	0.1	1.1	1.005
3	0.2	1.201	1.023
4	0.3	1.305	1.054
5	0.4	1.413	1.1
6	0.5	1.526	1.164
7	0.6	1.646	1.247
8	0.7	1.776	1.351
9	0.8	1.917	1.477
10	0.9	2.072	1.626
11	1	2.243	1.802

Рисунок 2.9 – Розв'язання диференціального рівняння другого порядку

2.2.3 Системи диференціальних рівнянь

Методика розв'язання систем звичайних диференціальних рівнянь така ж сама, як і одного диференціального рівняння n -го порядку. Відмінністю є те, що система може складатися з декількох диференціальних рівнянь n -го порядку. В такому випадку необхідно для кожного рівняння, що входить в систему, виконати перетворення (2.3) – (2.5), в результаті чого отримаємо систему звичайних диференціальних рівнянь з великою кількістю невідомих функцій, проте з похідними лише першого порядку. Якщо ж система містить лише диференціальні рівняння першого порядку, то перетворення для них робити не потрібно.



Розв'яжемо систему диференціальних рівнянь першого порядку

$$\begin{cases} x' = x + y, \\ y' = x - y. \end{cases}$$

з початковими умовами $x(0) = 2, y(0) = 1$ на інтервалі $[0,1]$.

Розв'язання задачі подано на рисунку 2.10.

Розв'язання системи диференціальних рівнянь першого порядку

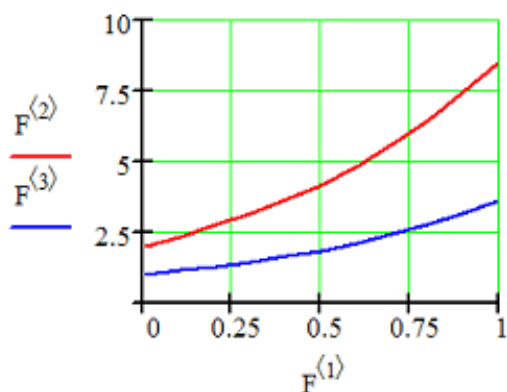
Початкова система: $x' = x + y$

$$y' = x - y$$

Початкові умови: $x(0) = 2 \quad y(0) = 1$

$$\text{ORIGIN} := 1 \quad x := \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad D(t,x) := \begin{pmatrix} x_1 + x_2 \\ x_1 - x_2 \end{pmatrix}$$

$F := \text{rkfixed}(x, 0, 1, 10, D)$



	1	2	3
1	0	2	1
2	0.1	2.321	1.11
3	0.2	2.689	1.243
4	0.3	3.11	1.4
5	0.4	3.594	1.586
6	0.5	4.149	1.803
7	0.6	4.788	2.057
8	0.7	5.523	2.351
9	0.8	6.368	2.693
10	0.9	7.341	3.089
11	1	8.461	3.546

Рисунок 2.10 – Розв'язання системи диференціальних рівнянь першого порядку



Розв'яжемо систему диференціальних рівнянь другого порядку

$$\begin{cases} u'' = 2v, \\ v'' = 2u \end{cases}$$

з початковими умовами $u(0) = u'(0) = 1, v(0) = v'(0) = 0$ на відрізку $[0, 2]$. Розв'язання задачі подано на рисунку 2.11.

Розв'язання системи диференціальних рівнянь другого порядку

Початкова система: $u''' = 2v$

$$v'' = 2u$$

Початкові умови: $u(0) = 1 \quad u'(0) = 1$

$$v(0) = 0 \quad v'(0) = 0$$

$$y := \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{matrix} u(0) \\ u'(0) \\ v(0) \\ v'(0) \end{matrix} \quad D(x, y) := \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ 2 \cdot y_0 \end{pmatrix} \begin{matrix} u' \\ u'' \\ v' \\ v'' \end{matrix} \quad F := \text{rkfixed}(y, 0, 2, 20, D)$$

$$x := F^{(0)} \quad u := F^{(1)} \quad u' := F^{(2)} \quad v := F^{(3)} \quad v' := F^{(4)} \quad i := 0..20$$

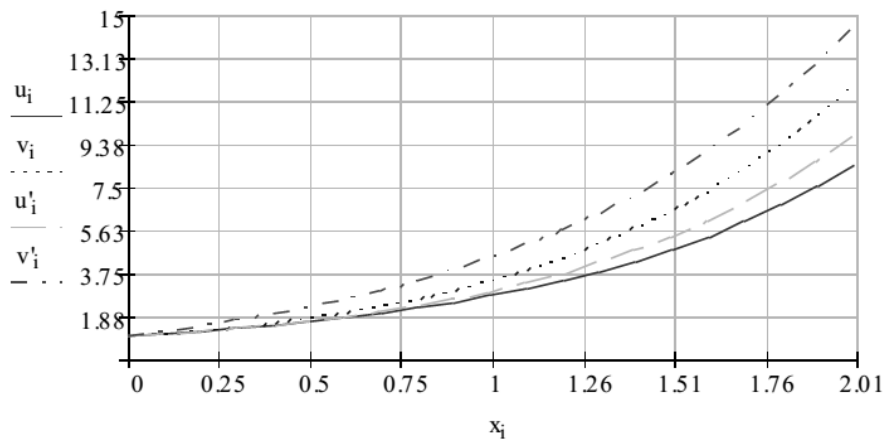


Рисунок 2.11 – Розв'язання системи диференціальних рівнянь другого порядку

Всі чисельні методи основані на апроксимації диференціальних рівнянь з використанням різницевої формули. Залежно від конкретної форми апроксимації отримують алгоритми різної точності та швидкодії. В MathCAD використано найбільш відомий алгоритм Рунге-Кутта четвертого порядку. Він забезпечує маленьку похибку для широкого класу систем звичайних диференціальних рівнянь за виключенням «жорстких» систем. Тому в більшості випадків використовується функція *rkfixed*.

Функція *Rkadapt* використовується у випадках, коли відомо, що розв’язок на заданому інтервалі змінюється нерівномірно, тобто існують інтервали повільних і різких змін. Метод Рунге-Кутта зі змінним кроком розбиває інтервал не на рівномірні кроки, а більш оптимальним способом. Там, де функція змінюється слабо, крок інтегрування вибирають більший, а в місцях його сильних змін – менший. Як результат, для дослідження однакової точності потрібно менше число кроків, ніж для *rkfixed*. Метод Булірша–Штера *Bulstoer* часто є більш ефективним для пошуку розв’язків, якщо функція є гладкою.

2.3 Функція *statespace*

В MathCAD 14 введена нова функція *statespace* (поряд з новими функціями Бесселя – *Dai*, *DBi*, *DAI.sc*, *Jacob* у *DBi.sc*), яка дозволяє розв’язувати диференціальні рівняння, записані в матричній формі.

Ця функція дозволяє вводити лінійні системи звичайних диференціальних рівнянь як матриці. Вона буде корисною для користувачів, які розробляють і аналізують автоматизовані інженерні проекти, а також тим, кому необхідно розв’язувати лінійні системи звичайних ДР.

Приклад використання функції *statespace* подано на рисунку 2.12.

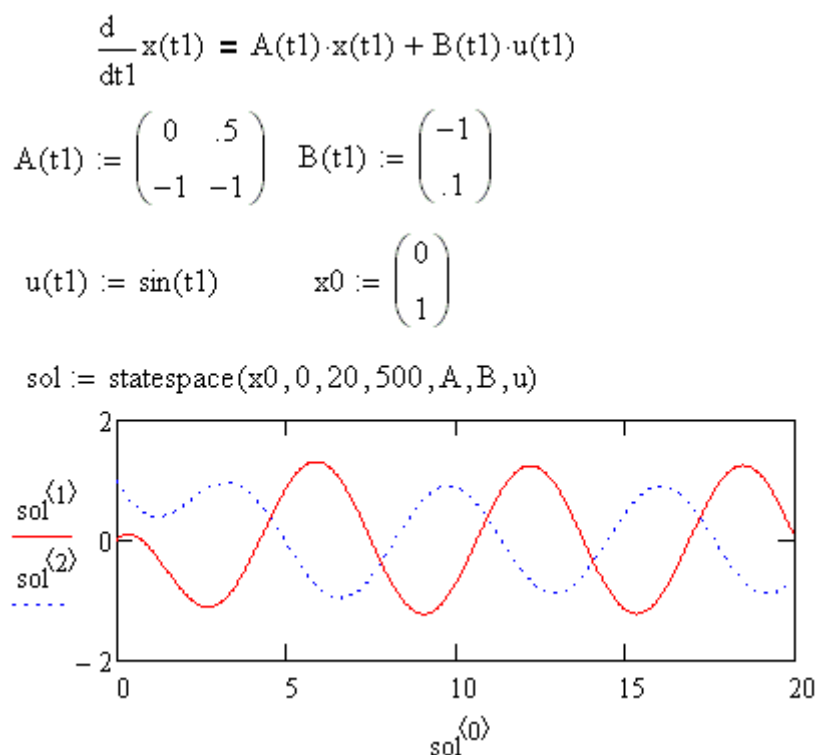


Рисунок 2.12 – Розв’язання системи лінійних диференціальних рівнянь за допомогою функції *statespace*

2.4 «Жорсткі» системи диференціальних рівнянь

В інженерній практиці трапляються випадки, коли шукана функція змінюється дуже динамічно та нерівномірно, повільні інтервали змінюються на швидкі, і її значення на інтервалі інтегрування можуть відрізнятись у велику кількість разів (на практиці в 100 і більше). Такі системи диференціальних рівнянь називаються «жорсткими».

Для розв'язання таких задач необхідно використовувати дуже малий крок інтегрування, що, звісно, призводить до утворення великої кількості обчислювальних операцій, а це, й собі, призводить до значного збільшення часу розв'язання задачі.

Крім цього, через те що похибки округлення накопичуються зі збільшенням кількості обчислень приблизно пропорційно цій кількості, на практиці вже після мільйона операцій похибки округлення можуть невпізнанно спотворити інтегральну криву. Якщо врахувати, що кожний крок інтегрування може містити тисячі операцій, то вже через тисячі кроків інтегрування відносна похибка внаслідок округлення збільшується до 100%. Подальше інтегрування стає недоцільним.

Тому для розв'язання «жорстких» систем рівнянь розроблено додаткові функції – *Stiff* і *Stiffb*, що реалізують відповідно методи *Розенброка* і *Булірша-Шмера*. Функції *Stiff* і *Stiffb* стали доступними, починаючи з версії MathCAD 2001.

Опис їх параметрів подано в таблиці 2.3.

В результаті розв'язання диференціального рівняння за допомогою цих функцій отримують матрицю, яка містить $n + 1$ стовпців: в першому містяться значення аргументу x , в другому – значення функції $y(x)$, а в тих, що залишилися, – значення $n - 1$ похідних функції $y(x)$.



Розв'яжемо жорстку систему диференціальних рівнянь першого порядку:

$$\begin{cases} x' = -0.1x + 100yz, \\ y' = 0.1x - 100yz - 1000y, \\ z' = 1000y \end{cases}$$

з початковими умовами $x(0) = 1$, $y(0) = 0$, $z(0) = 0$ на інтервалі $[0, 50]$ за допомогою функції *Stiffb*. Розв'язання цієї задачі подано на рисунку 2.13.

Таблиця 2.3 – Опис параметрів функції *Stiffr* і *Stiffb*

Формат виклику функції: <i>Stiffb</i> ($y, x1, x2, N, D, J$) <i>Stiffr</i> ($y, x1, x2, N, D, J$)	
Параметри	Опис параметра
y	Вектор початкових умов $\bar{Y}_0(x)$, $y_i = (\bar{Y}_0)_i$.
$x1, x2$	Відповідно початкова і кінцева точки інтервалу інтегрування системи; для функцій, що обчислюють розв'язок в заданій точці, $x1$ – початкова точка, $x2$ – задана точка.
N	Кількість вузлів на інтервалі $[x1; x2]$; при розв'язанні задачі на відрізку результат містить $N + 1$ рядок.
D	Ім'я вектор-функції $D(x, y)$ правих частин $\bar{F}(x, \bar{Y})$, $D_i(x, y) = f_i(x, y_1, \dots, y_n)$ (ім'я D від <i>Derivative</i> – похідна).
J	Ім'я матриці-функції $J(x, y)$ розмірності $n(n + 1)$, в першому стовпці якої зберігаються вирази частинних похідних за x правих частин системи, а в інших n стовпцях міститься матриця Якобі правих частин: $J(x, y) = \begin{pmatrix} \frac{\partial f_1(x, y)}{\partial x} & \frac{\partial f_1(x, y)}{\partial y_1} & \dots & \frac{\partial f_1(x, y)}{\partial y_n} \\ \frac{\partial f_2(x, y)}{\partial x} & \frac{\partial f_2(x, y)}{\partial y_1} & \dots & \frac{\partial f_2(x, y)}{\partial y_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n(x, y)}{\partial x} & \frac{\partial f_n(x, y)}{\partial y_1} & \dots & \frac{\partial f_n(x, y)}{\partial y_n} \end{pmatrix}.$

Починаючи з версії MathCAD 2001i, доступна ще одна функція – *Radau*, що реалізує метод *RADAU5*.

Нова функція *Radau* має такий самий список параметрів, що й функції *rkfixed*, *Rkadapt* і *Bulstoer*. Відмінність і перевага цієї функції перед функціями *Stiffr* і *Stiffb* в тому, що вона не потребує матриці Якобі як параметра.

В MathCAD 14 функція *Radau* отримала три нових додаткових, але необов'язкових аргументи, які дозволяють впливати на точність і швидкість її роботи.

Порівняння результатів розв'язання жорсткої системи звичайних диференціальних рівнянь за допомогою функцій *Radau* і *Stiffr* подано на рисунку 2.14. Як видно, результати роботи збігаються, проте використовувати функцію *Radau* значно простіше.

Розв'язання "жорсткої" системи диференціальних рівнянь

Початкова система:

$$\begin{aligned}x' &= -0.1 \cdot x + 100 \cdot y \cdot z \\y' &= 0.1 \cdot x - 100 \cdot y \cdot z - 1000 \cdot y \\z' &= 1000 \cdot y\end{aligned}$$

Початкові умови: $x(0) = 1$ $y(0) = 0$ $z(0) = 0$

$$y_0 := \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \underline{\underline{F}}(t, y) := \begin{pmatrix} -0.1 \cdot y_0 + 100 \cdot y_1 \cdot y_2 \\ 0.1 \cdot y_0 - 100 \cdot y_1 \cdot y_2 - 1000 \cdot y_1 \\ 1000 \cdot y_1 \end{pmatrix}$$

$$\underline{\underline{J}}(t, y) := \begin{pmatrix} 0 & -0.1 & 100 \cdot y_2 & 100 \cdot y_1 \\ 0 & 0.1 & -100 \cdot y_2 - 1000 & -100 \cdot y_1 \\ 0 & 0 & 1000 & 0 \end{pmatrix} \quad D := \text{Stiffb}(y_0, 0, 50, 50, F, J)$$

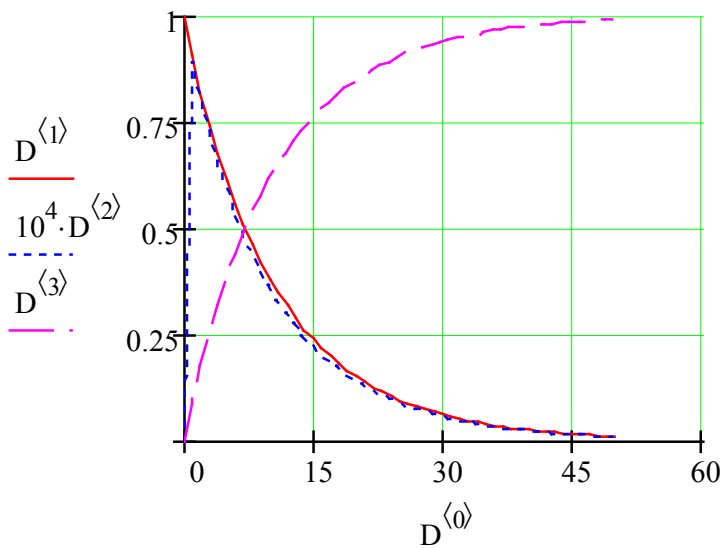


Рисунок 2.13 – Розв'язання «жорсткої» системи диференціальних рівнянь

Розв'язання "жорсткої" системи диференціальних рівнянь за допомогою функцій Radau та Stiff

Початкова система: $x' = 22 \cdot x + 42 \cdot y$
 $y' = -21 \cdot x - 41 \cdot y$

Початкові умови: $x(0) = 3$ $y(0) = -2$

$y := \begin{pmatrix} 3 \\ -2 \end{pmatrix}$ $x1 := 0$ $x2 := 3$ $acc := 10^{-5}$ $npts := 3$ $kmax := 50$ $step := 10^{-4}$

$D(x, Y) := \begin{pmatrix} 22 \cdot Y_0 + 42 \cdot Y_1 \\ -21 \cdot Y_0 - 41 \cdot Y_1 \end{pmatrix}$

$J(x, Y) := \begin{pmatrix} 0 & 22 & 42 \\ 0 & -21 & -41 \end{pmatrix}$

$A := \text{stiff}(y, x1, x2, acc, D, J, kmax, step)$

$B := \text{radau}(y, x1, x2, acc, D, kmax, step)$

A =

	0	1	2
0	0	3	-2
1	$7.5 \cdot 10^{-3}$	2.876	-1.868
2	0.015	2.775	-1.761
3	0.022	2.689	-1.667
4	0.029	2.615	-1.585
5	0.037	2.552	-1.515
6	0.045	2.499	-1.454
7	0.053	2.456	-1.401
8	0.061	2.42	-1.357

B =

	0	1	2
0	0	3	-2
1	$5.85 \cdot 10^{-4}$	2.99	-1.989
2	$4.681 \cdot 10^{-3}$	2.92	-1.915
3	0.024	2.663	-1.638
4	0.044	2.503	-1.458
5	0.064	2.41	-1.344
6	0.084	2.362	-1.275
7	0.111	2.343	-1.226
8	0.139	2.36	-1.211

Таблиця з 50 точок

Рисунок 2.14 – Розв'язання «жорсткої» системи диференціальних рівнянь за допомогою функцій *Radau* і *Stiff*



«Жорсткі» системи можна розв'язувати за допомогою функцій *rkfixed*, *Rkadapt* і *Bulstoer*, проте для цього потрібно вказувати дуже малий крок інтегрування. Зрозуміло, що час пошуку розв'язку буде значно більшим, ніж при використанні функцій *Stiff*, *Stiffb* чи *Radau*.



Якщо система не дуже «жорстка», то за допомогою функції *Radau* вдається отримати розв'язок тільки для першої половини інтервалу, приблизно до $t_1=20$. Для великих інтервалів вона дає збій і виводить повідомлення про помилку.

2.5 Знаходження розв'язку в кінцевій точці

Дуже часто при математичних розрахунках не потрібно знаходити розв'язок на всьому інтервалі інтегрування, а лише в кінцевій точці. Для цього розробники MathCAD передбачили ряд функцій, що мають такі ж назви, що і розглянуті вище функції, проте вони пишуться з малої літери. За допомогою цих функцій можна знаходити розв'язки рівняння або системи рівнянь лише в кінцевій точці.

Опис додаткових параметрів, що відсутні в аналогах цих функцій, подано в таблиці 2.4.

Таблиця 2.4 – Опис параметрів функцій *bulstoer*, *stiffb*, *stiffr*, *radau*

Формат виклику функції: $bulstoer(y, a, b, acc, D, k, s)$ $rkadapt(y, a, b, acc, D, k, s)$ $stiffb(y, a, b, acc, D, J, k, s)$ $stiffr(y, a, b, acc, D, J, k, s)$ $radau(y, a, b, acc, F, k, s)$	
Параметри	Опис параметра
$Y, x1, x2,$ D, J	Такі ж, як і в попередніх функціях.
acc	Параметр, який контролює похибку розв'язання при автоматичному виборі кроку інтегрування (якщо похибка розв'язання більше acc , то крок сітки зменшується; крок зменшується до тих пір, поки його значення не стане меншим s , рекомендоване значення 0.001).
k	Максимальне число вузлів сітки, в яких може бути обчислений розв'язок задачі на інтервалі інтегрування, дорівнює максимальному числу рядків в результаті.
s	Найменше допустиме значення кроку нерівномірної сітки.

Видно, що замість числа кроків на інтервалі інтегрування в цих функціях потрібно задати точність розрахунку чисельним методом значення функції в кінцевій точці. Кількість кроків та їх розташування визначається чисельним методом автоматично, щоб забезпечити цю точність. Два останні параметри потрібні для того, щоб користувач міг штучно вплинути на розбиття інтервалу на кроки. Параметр k слугує для того, щоб кроків не було занадто багато, причому k не може бути більшим за 1000. Параметр s слугує для того, щоб жоден крок не був дуже малим для появи великих похибок при різницевій апроксимації диференціальних рівнянь всередині алгоритму. Ці параметри необхідно задавати явно, виходячи з властивостей конкретної системи звичайних диференціальних рівнянь. Як правило, провівши ряд тестових розрахунків, можна підібрати їх оптимальний набір для кожного конкретного випадку.



Розв'яжемо диференціальне рівняння другого порядку $y''(t) + 0.1y'(t) + y(t) = 0$ з початковими умовами $y(0) = 0.1$, $y'(0) = 0$ в кінцевій точці методом Булірша–Штера (функція *bulstoer*). Розв'язання подано на рисунку 2.15.



Ці функції не призначені для знаходження розв'язку в внутрішніх точках інтервалу, хоча вони і видають їх в матриці-результаті. На рисунку 2.16 подано фазові портрети цієї системи звичайних диференціальних рівнянь, одержані за допомогою функції *bulstoer* (рисунок 2.15) і за допомогою функції *rkadapt* (при відповідній

заміні третього рядка). Видно, що не дивлячись на високу точність і правильний результат на кінці інтервалу, лівий графік мало нагадує правильний фазовий портрет. Тому для знаходження значень в кожній точці інтервалу ці функції використовувати недоцільно.

$$D(t,y) := \begin{pmatrix} y_1 \\ -y_0 - 0.1 \cdot y_1 \end{pmatrix} \quad y_0 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$u := \text{bulstoer}(y_0, 0, 50, 10^{-5}, D, 300, 0.0001)$$

$$M := \text{length}(u^{(0)}) - 1 \quad M = 22$$

$$(u^T)^{(M)} = \begin{pmatrix} 50 \\ 0.05 \\ 0.106 \end{pmatrix} \quad u_{M,1} = 0.05$$

Рисунок 2.15 – Знаходження розв’язку рівняння в кінцевій точці за допомогою функції *bulst*

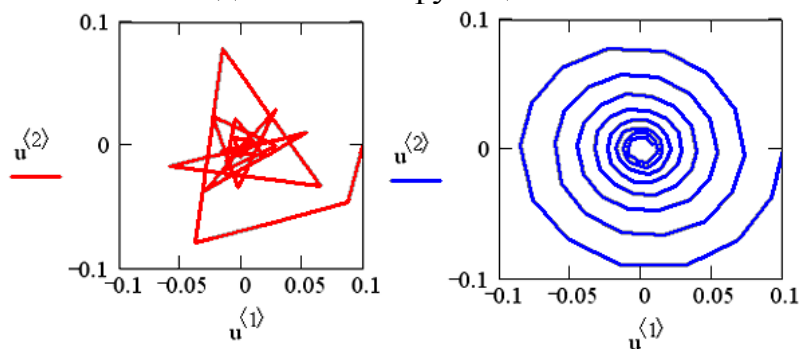


Рисунок 2.16 – Фазові портрети, отримані при розв’язанні системи за допомогою функцій *bulstoer* і *rkadapt*

2.6 Розв’язання крайових задач

При розв’язанні диференціальних рівнянь кількість граничних умов має дорівнювати порядку рівняння. Причому, ці умови можуть бути задані в одній точці, обов’язково в початковій, або в двох точках, одна з яких обов’язково має бути початком інтегрування. Друга точка може бути будь-якою, проте обов’язково правіше початку інтегрування.

Припустимо, що:

- є диференціальне рівняння або система звичайних диференціальних рівнянь;
- в початковій точці x_1 інтервалу, на якому відшукується розв’язок, задана частина граничних умов;
- всередині або в кінці інтервалу в точці x_2 задані інші граничні умови;
- загальна кількість граничних умов дорівнює порядку системи.

В такому випадку маємо крайову задачу, для розв'язання якої потрібно використовувати функцію *sbval*, щоб знайти граничні умови, яких не вистачає в початковій точці x_1 . Зауважимо, що функція *sbval* не розв'язує диференціальне рівняння в цілому, а лише знаходить початкові умови, яких не вистачає. Функція *sbval* реалізує алгоритм «стрілянини». Суть його полягає в пробному запису початкових умов, яких не вистачає, і розв'язанні потім задачі Коші відомими методами.

Опис параметрів функції *sbval* подано в таблиці 2.5.

Таблиця 2.5 – Опис параметрів функції *sbval*

Формат виклику функції: <i>sbval(v, x1, x2, D, load, score)</i> .	
Параметри	Опис параметра
v	Вектор початкових наближень для початкових умов, яких не вистачає в точці x_1 ;
x_1, x_2	Точки інтервалу, в яких задані граничні умови;
$D(x, y)$	Функція, що містить у вигляді вектора перші похідні невідомих функцій;
$load(x_1, y)$	Векторна функція, що містить значення початкових умов в точці x_1 (вектор складається з n елементів, які відповідають вектору початкових умов в точці x_1 , деякі з цих елементів є константами, які визначаються початковими умовами, інші невідомі і мають визначатися функцією <i>sbval</i>);
$score(x_2, y)$	Векторна функція, що повертає вектор, кількість елементів якого дорівнює порядку рівняння n . Кожний елемент вектора містить різницю між початковими умовами, заданими в точці x_2 , і значенням шуканого розв'язку рівняння в цій точці. Вектор <i>score</i> показує, наскільки значення знайденого розв'язку в точці x_2 близькі до значень, що задані в точці x_2 . Значення 0 для будь-якого елемента вказує на повний збіг між початковими умовами і тим значенням, яке повертає функція <i>sbval</i> .



Розв'язання крайових задач в MathCAD, на відміну від більшості інших операцій, реалізовано не зовсім очевидним чином. Зокрема, необхідно пам'ятати, що кількість елементів векторів D і $load$ дорівнює кількості рівнянь N , а кількість векторів v , $score$ і результату дії функції *sbval* – кількості правих граничних умов L . Відповідно, лівих граничних умов в завданні має бути $(N-L)$.

Функція *sbval* призначена не для пошуку власного розв'язку, тобто невідомих функцій $y_i(x)$, а для визначення початкових умов, яких не вистачає в першій точці інтервалу, тобто $y_i(x_0)$. Щоб обчислити $y_i(x)$ на всьому інтервалі, потрібно додатково розв'язати задачу Коші.



Розв'яжемо крайову задачу $y^{(V)} + y = 0$ з граничними умовами $y(0) = 0, y(1) = 1, y'(0) = 7, y'(1) = 10, y''(1) = 5$ на інтервалі $[0, 1]$ за допомогою функції *sbval*. Розв'язання задачі подано на рисунку 2.17.

$$\begin{aligned}
 & y^{(V)} + y = 0 & x1 := 0 & \quad x2 := 1 \\
 & y(0) = 0 & y'(0) = 7 \\
 & y(1) = 1 & y'(1) = 10 & \quad y''(1) = 5
 \end{aligned}$$

Вектор наближених значень

$$v := \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{matrix} y''(0) \\ y'''(0) \\ y''(0) \end{matrix}$$

$$Ll(x1, v) := \begin{pmatrix} 0 \\ 7 \\ v_0 \\ v_1 \\ v_2 \end{pmatrix} \quad \text{score}(x2, y) := \begin{pmatrix} y_0 - 1 \\ y_1 - 10 \\ y_2 - 5 \end{pmatrix} \quad F(x, y) := \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ -y_0 \end{pmatrix} \quad S := \text{sbval}(v, 0, 1, F, Ll, \text{score})$$

Початкові умови, яких не вистачає

$$S = \begin{pmatrix} -85.014 \\ 348.107 \\ -516.257 \end{pmatrix} \begin{matrix} y''(0) \\ y'''(0) \\ y''(0) \end{matrix} \quad y := \begin{pmatrix} 0 \\ 7 \\ S_0 \\ S_1 \\ S_2 \end{pmatrix} \quad Y := \text{rkfixed}(y, x1, x2, 50, F) \quad n := 0..50$$

	0	1	2	3	4	5
0	0	0	7	-85.014	348.107	-516.257
1	0.02	0.123	5.369	-78.155	337.782	-516.259
2	0.04	0.216	3.872	-71.503	327.457	-516.262
3	0.06	0.279	2.507	-65.057	317.131	-516.267
4	0.08	0.317	1.269	-58.818	306.806	-516.273
5	0.1	0.331	0.153	-52.785	296.48	-516.279
6	0.12	0.324	-0.844	-46.958	286.155	-516.286
7	0.14	0.298	-1.727	-41.339	275.829	-516.292
8	0.16	0.255	-2.499	-35.925	265.503	-516.298
9	0.18	0.199	-3.165	-30.718	255.177	-516.302
10	0.2	0.129	-3.729	-25.718	244.851	-516.306

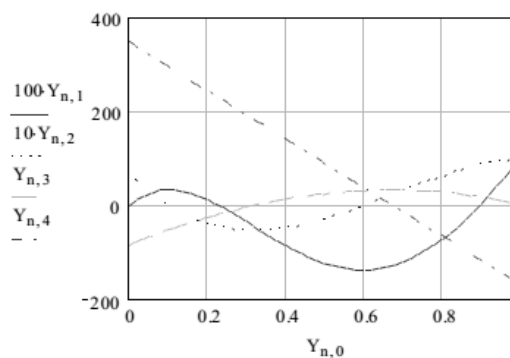


Рисунок 2.17 – Розв'язання крайової задачі за допомогою функції *sbval*

Вектор S містить початкові умови, яких не вистачає. Після цього розв'язується звичайна задача Коші за допомогою функції *rkfixed*.

Після перетворення рівняння п'ятого порядку до системи диференціальних рівнянь першого порядку крайова задача складається з системи п'яти рівнянь ($N=5$), двох лівих ($L=2$) і трьох правих ($N-L=5-2=3$) граничних умов.



Початкові значення, які визначаються вектором v , фактично є параметрами чисельного методу і тому можуть вирішальним чином вплинути на розв'язок крайової задачі.

Векторній функції $load(x, v)$ присвоюють ліві граничні умови. Ця функція аналогічна векторній змінній, що визначає початкові умови для вбудованих функцій при розв'язанні задачі Коші. Відмінність полягає у записі умов, яких не вистачає. Замість конкретних чисел на відповідних місцях пишуться імена шуканих елементів вектора v .

Перший аргумент функції $load$ – це ліва гранична умова. Його конкретне значення визначається безпосередньо в списку аргументів функції $sbval$. Аргумент x функції $score$ аналогічний функції $load$ і потрібен для тих випадків, коли гранична умова явно залежить від координати x . Вектор $score$ має складатися з такої ж кількості елементів, що і вектор v .

Реалізований у функції $sbval$ алгоритм «стрілянини» шукає початкові умови, яких не вистачає так, щоб розв'язок одержаної задачі Коші наближав функцію $score(x,y)$ до нуля.



Метод «стрілянини» не можна використовувати для розв'язання «жорстких» та нелінійних крайових задач. Алгоритми розв'язання таких задач в MathCAD необхідно реалізовувати самостійно.

В деяких випадках буває, що крім граничних умов в двох точках задана умова в деякій третій точці всередині інтервалу інтегрування. В цьому випадку необхідно використовувати функцію $bvalfit$, яка як і функція $sbval$ знаходить вектор початкових умов, що не вистачає для розв'язання задачі Коші.

Опис параметрів функції $bvalfit$ подано в таблиці 2.6.

Таблиця 2.6 – Опис параметрів функції $bvalfit$

Формат виклику функції: $bvalfit(v1, v2, x1, x2, xf, D, load1, load2, score)$	
Параметри	Опис параметра
$v1$	Вектор, який містить початкове наближення для початкових умов, що не задані в точці $x1$.
$v2$	Вектор, який містить початкове наближення для початкових умов, що не задані в точці $x2$.
$x1$ і $x2$	Точки інтервалу, в яких задані граничні умови.
xf	Деяка точка між $x1$ і $x2$. В цій точці розв'язки мають бути однаковими.
$D(x,y)$	Функція, що містить у вигляді вектора перші похідні невідомих функцій.

Продовження таблиці 2.6

Формат виклику функції: $bvalfit(v1, v2, x1, x2, xf, D, load1, load2, score)$	
Параметри	Опис параметра
$load1(x1, y)$	Векторна функція, що містить значення початкових умов в точці $x1$ (вектор складається з n елементів, які відповідають вектору початкових умов в точці $x1$, деякі з цих елементів є константами, які визначаються початковими умовами, інші невідомі і мають визначатися функцією $bvalfit$)
$load2(x2, y)$	Функція, аналогічна $load1$, але для значень в точці $x2$.
$score(xf, y)$	Векторна функція, яка повертає вектор із n елементів, що визначає, наскільки значення розв'язку, які починаються з точок $x1$ і $x2$, відповідають один одному в точці xf . Якщо необхідно, щоб розв'язки мали збіг, потрібно визначити $score(xf, y) = y$.



Функцію $bvalfit$ доцільно використовувати, якщо похідні мають розрив всередині інтервалу інтегрування.



Розв'яжемо крайову задачу $y'' = y$ з крайовими умовами $y(-1) = 1, y(3) = 2$ на інтервалі $[-1, 3]$ за допомогою функції $bvalfit$. Розв'язання цієї задачі подано на рисунку 2.18.

Загалом, цю задачу можна розв'язати і за допомогою функції $sbval$. Проте в певних випадках (зокрема з міркувань швидкості розрахунків) зручніше використовувати функцію $bvalfit$, тобто вести «стрілянину» з обох меж інтервалу.

$$y'' = y \quad y(-1) = 1 \quad y(3) = 2 \quad x1 := -1$$

$$xf := 0 \quad v1_0 := 1 \quad x2 := 3 \quad v2_0 := 1$$

$$F(x, y) := \begin{pmatrix} y_1 \\ y_0 \end{pmatrix}$$

$$load1(x1, v1) := \begin{pmatrix} 1 \\ v1_0 \end{pmatrix} \quad load2(x2, v2) := \begin{pmatrix} 2 \\ v2_0 \end{pmatrix}$$

$$score(xf, y) := y \quad S1 := bvalfit(v1, v2, x1, x2, xf, F, load1, load2, score)$$

Рисунок 2.18 – Розв'язання крайової задачі за допомогою функції $bvalfit$

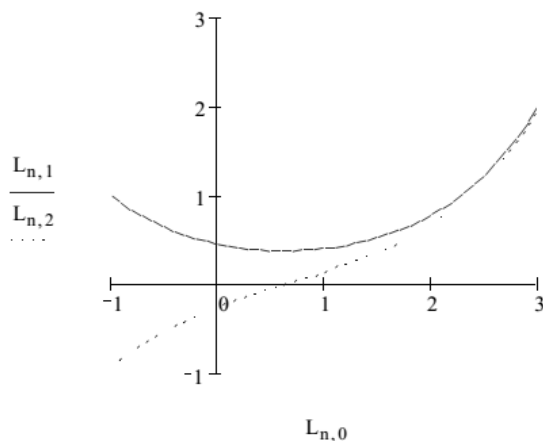
Початкові умови, яких не вистачає – значення похідних в точках $x=-1$ і $x=3$, відповідно

$$S1 = (-0.927 \ 1.965)$$

$$y := \begin{pmatrix} 1 \\ -0.927 \end{pmatrix}$$

Складаємо вектор початкових умов $L := rkfixed(y, x1, x2, 40, F)$

$$n := 0..40$$



	0	1	2
0	-1	1	-0.927
1	-0.9	0.912	-0.831
2	-0.8	0.833	-0.744
3	-0.7	0.763	-0.665
4	-0.6	0.7	-0.591
5	-0.5	0.645	-0.524
6	-0.4	0.595	-0.462
7	-0.3	0.552	-0.405
8	-0.2	0.514	-0.352
9	-0.1	0.482	-0.302
10	0	0.454	-0.255
11	0.1	0.43	-0.211
12	0.2	0.411	-0.169
13	0.3	0.397	-0.129
14	0.4	0.386	-0.09
15	0.5	0.379	-0.051

Рисунок 2.18, аркуш 2

Питання для самоконтролю:

1. Що таке диференціальне рівняння (ДР)?
2. Які задачі можна обчислювати за допомогою функції Odesolve?
3. Що означає необов'язковий параметр step функції Odesolve?
4. Чи можна за допомогою функції Odesolve розв'язувати системи ДР?
5. Яке призначення функції sbval?
6. Які нові методи розв'язання ДР стали доступними в MathCAD 14?
7. Які методи реалізують функції *rkfixed*, *Rkadapt* і *Bulstoer*?
8. Що означає запис: `ORIGIN:=1`? Чи є він обов'язковим?
9. В чому переваги алгоритму Рунге-Кутта четвертого порядку над іншими методами?
10. Яка функція дозволяє розв'язувати ДР, записані в матричній формі?

Задачі для самоконтролю:

1. Розв'язати задачі Коші за допомогою обчислювального блока Odesolve:

- а) $y'' + 2y' + 5y = e^{-x}(\cos^2 x + \operatorname{tg} x)$ з початковими умовами $y(0) = 1$, $y'(0) = 0$ на інтервалі $[0, \frac{\pi}{2}]$;

b) $y = x(y' - x \cos x)$ з початковою умовою $y(\pi) = \pi$ на інтервалі $[\pi, 2\pi]$.

2. Розв'язати задачі Коші за допомогою функцій *rkfixed*, *Rkadart* і *Bulstoer*. Порівняти отримані значення:

a) $(2e^y - x)y' = 1$ з початковою умовою $y(0) = 1$ на інтервалі $[0, 2]$;

b) $y' = \frac{y}{3x - y^2}$ з початковою умовою $y(5) = 10$ на інтервалі $[5, 10]$.

3. Розв'язати системи ДР за допомогою функцій *Odesolve* і *rkfixed*:

a)
$$\begin{cases} x' = y + z, \\ y' = x + z, \\ z' = y + z, \end{cases}$$

з початковими умовами $x(0) = 1, y(0) = 5, z(0) = 2$ на інтервалі $[0, 10]$;

b)
$$\begin{cases} x' = x + y^2 + 2, \\ y' = 6x - \ln y, \end{cases}$$

з початковими умовами $x(5) = 1, y(5) = 2$ на інтервалі $[5, 10]$.

4. Розв'язати «жорсткі» системи ДР за допомогою функцій *StiffR*, *StiffB*, *Radau*. Порівняти отримані результати.

a)
$$\begin{cases} x' = 10000x + 0.0001y, \\ y' = 100x - 0.01y, \end{cases}$$

з початковими умовами $x(0) = 0, y(0) = 0$ на інтервалі $[0, 0.5]$;

b)
$$\begin{cases} x' = 999y^2, \\ y' = 0.999x^2 \end{cases}$$

з початковими умовами $x(0) = 1, y(0) = 2$ на інтервалі $[0, 1]$.

5. Знайти розв'язок ДР в кінцевій точці інтервалу інтегрування:

a) $y' = y - x^2$ з початковою умовою $y(0) = 0$ на інтервалі $[0, 1]$;

b) $y' = \frac{x^2 + y^2}{2} - 1$ з початковими умовами $y(1) = 10$ на інтервалі $[10, 15]$.

6. Розв'язати крайові задачі за допомогою функцій *sbval*, *bvalfit*:

a) $y'' + y = 2x - \pi$ з умовами $y(0) = 0, y(\pi) = 0$ на інтервалі $[0, \pi]$;

b) $y'' - y' - 2y = 0$ з умовами $y'(0) = 0, y(1) = 0$ на інтервалі $[0, 1]$.

3 РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ В ЧАСТИННИХ ПОХІДНИХ

В цьому розділі розглянуто основні принципи і підходи щодо розв'язання диференціальних рівнянь другого порядку в частинних похідних вбудованими засобами MathCAD (функціями *pdesolve*, *numol*, *multigrid*, *relax*). Наведено особливості застосування тих чи інших функцій, а також подано загальні рекомендації щодо їх використання.

В загальному вигляді диференціальне рівняння другого порядку в частинних похідних можна записати у вигляді:

$$A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + E(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}) = 0, \quad (3.1)$$

де коефіцієнти A , B , C , E можуть залежати від x , y , а в окремих випадках можуть бути константами.

Дискримінантом рівняння називається функція $D = B^2 - AC$.

Залежно від значення дискримінанта всі рівняння поділяють на:

– $D > 0$: параболічні – що містять першу похідну за однією змінною і другу – за іншою, причому всі ці похідні входять в рівняння з однаковим знаком;

– $D = 0$: гіперболічні – що містять першу похідну за однією змінною і другу – за іншою, що входять в рівняння з різними знаками;

– $D < 0$: еліптичні – що містять тільки другі похідні, причому одного знака.

Деякі більш складні рівняння, які неможна однозначно підігнати під наведену класифікацію, називають гібридними типами рівнянь.

Постановка задачі для рівнянь в частинних похідних містить визначення самого рівняння (або системи декількох рівнянь), а також необхідної кількості крайових умов (число і характер яких визначається специфікою рівняння). Рівняння мають містити частинні похідні невідомої функції (або декількох функцій, якщо рівнянь декілька) за різними аргументами, наприклад, просторовою змінною x і часом t . Відповідно, для розв'язання задачі необхідно обчислити функцію декількох змінних, наприклад $u(x, t)$ в деякій області визначення аргументів $0 < x < L$ і $0 < t < T$.

3.1 Розв'язання параболічних і гіперболічних рівнянь

3.1.1 Функції *Pdesolve* і *numol*

Починаючи з версії MathCAD 11, для розв'язання рівнянь в частинних похідних доступна вбудована функція *Pdesolve*. Ця функція застосовується в рамках обчислювального блока, що починається ключо-

вим словом *Given* і використовується для розв'язання гіперболічних і параболічних рівнянь.

Ця функція використовується для розв'язання одновимірних рівнянь (або системи рівнянь) в частинних похідних, що залежать від часу t і просторової координати x . Функція повертає скалярну (для єдиного початкового рівняння) або векторну (для системи рівнянь) функцію двох аргументів (x, t) , що є розв'язком диференціального рівняння (або системи рівнянь) в частинних похідних. Кінцевий результат отримують у вигляді функції шляхом інтерполяції сітчастої функції, що обчислюється відповідно до різницевої схеми.

Параметри функції *Pdesolve* та їх опис подано в таблиці 3.1.

Таблиця 3.1 – Опис параметрів функції *Pdesolve*

Формат виклику функції: <i>Pdesolve</i> ($u, x, xrange, t, trange, [xpts], [tpts]$)	
Параметри	Опис параметра
u	Явно заданий вектор імен функцій (без імен аргументів), що підлягають обчисленню. Ці функції, а також граничні умови (у формі Діріхле або Неймана), мають бути визначені користувачем в рамках обчислювального блока <i>Given...Pdesolve</i> . Якщо розв'язується не система рівнянь в частинних похідних, а єдине рівняння, то вектор має містити тільки одне ім'я функції, відповідно.
x	Просторова координата (ім'я аргументу невідомої функції).
$xrange$	Просторовий інтервал, тобто вектор значень аргументу x для граничних умов. Цей вектор має складатися з двох дійсних чисел, що визначають ліву і праву межі інтервалу.
t	Час (ім'я аргументу невідомої функції)
$trange$	Розрахункова область за часом: вектор значень аргументу t , який має складатися з двох дійсних чисел, що визначають ліву і праву межі інтервалу за часом.
$xpts$	Кількість просторових точок дискретизації (необов'язковий параметр, якщо його не вказувати, то в такому разі він буде підібраний програмою автоматично).
$tpts$	Кількість тимчасових шарів, тобто інтервалів дискретизації за часом (необов'язковий параметр, також якщо його не вказувати, то він буде підібраний програмою автоматично).

Для коректного використання функції *Pdesolve* після ключового слова *Given* необхідно записати рівняння і граничні умови за допомогою логічних операторів, які розміщені на панелі *Boolean* (рисунок 3.1).

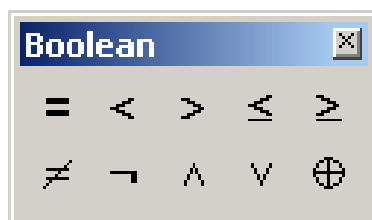


Рисунок 3.1 – Панель логічних операторів *Boolean*

Рівняння має містити ім'я невідомої функції $u(x,t)$ разом з іменами аргументів. Для ідентифікації частинних похідних в межах обчислювального блоку необхідно використовувати нижні індекси, наприклад, $u_{xx}(x,t)$ для позначення другої похідної функції по просторовій координаті x .



Для введення частинних похідних в цьому випадку не можна використовувати звичайний оператор похідної. Необхідно користуватися нижнім індексом, як при записі рівнянь в частинних похідних. Причому нижній індекс необхідно вводити не як елемент масиву (клавіша « [»), а як текстовий нижній індекс (клавіша « . »). Для того, щоб ввести другу похідну від функції u за x , необхідно набрати: **u. xx**.

Використання вбудованої функції *Pdesolve* пов'язане з досить громіздкими обчисленнями, які можуть займати велику кількість часу. Тому для зменшення часу обчислень, необхідно підібрати оптимальні значення параметрів *xpts* і *tpts* або дозволити програмі зробити це автоматично.

Починаючи з версії MathCAD 11, також з'являється можливість розв'язувати рівняння і системи параболічного і гіперболічного типів без створення групи рішень. В цьому випадку використовують функцію *numol*. Вона має більшу кількість аргументів і дозволяє управляти додатковими параметрами різницевого методу.



Рівняння гіперболічного типу спочатку мають бути зведені до системи рівнянь першого порядку за часом.

Параметри функції *numol* та їх опис подано в таблиці 3.2.

Таблиця 3.2 – Опис параметрів функції *numol*

Формат виклику функції: $numol(xrange, xpts, trange, tpts, Npde, Nae, rhs, init, bc)$	
Параметри	Опис параметра
$xrange$	Просторовий інтервал, тобто вектор значень аргументу x для граничних умов (два дійсних числа – ліва і права границі просторового інтервалу).
$xpts$	Кількість просторових точок дискретизації (можна не вказувати явно, в такому випадку програма підбере значення параметра автоматично).

Продовження таблиці 3.2

Формат виклику функції: $numol(xrange, xpts, trange, tpts, Npde, Npae, rhs, init, bc)$	
Параметри	Опис параметра
$trange$	Розрахункова часова область: вектор значень аргументу t (два дійсних числа – ліва і права границі розрахункового часового інтервалу).
$tpts$	Кількість часових шарів, тобто інтервалів дискретизації за часом (також можна не вказувати явно, в такому випадку програма підбере значення параметру автоматично).
$Npde$	Загальна кількість диференціальних рівнянь в частинних похідних в системі.
$Npae$	Загальна кількість додаткових алгебраїчних рівнянь, які також можуть входити до системи.
rhs	Векторна функція, що визначає систему диференціальних і алгебраїчних рівнянь.
$init(x)$	Векторна функція, що визначає початкові умови для кожної невідомої функції.
$bc(t)$	Функціональна матриця граничних умов.

Для розв'язання задачі за допомогою функції $numol$ потрібно:

1. Визначити векторну функцію $pdef(x, t, u, u_x, u_{xx})$. Ця функція є аналогом функції $D(x, y)$, яку ми раніше визначали при розв'язанні систем звичайних диференціальних рівнянь. Кількість елементів цього вектора дорівнює сумі $Npde + Npae$. Перші $npde$ елементів вектора $pdef$ задають праві частини диференціальних рівнянь (в лівій частині знаходяться перенесені похідні за часом). Інші елементи задають алгебраїчні рівняння, які пов'язують між собою невідомі функції (в цих елементах ліва частина дорівнює нулю).

2. Задати початкові умови для невідомих функцій у вигляді векторної функції $init(x)$. Кожний елемент цього вектора має містити початкову умову для відповідної функції.

3. Задати граничні умови у вигляді матричної функції $bc(t)$. Ця матриця має містити три стовпці і стільки рядків, скільки функцій в системі. В перших двох стовпцях цієї матриці задаються умови, що накладаються на ту чи іншу функцію відповідно в початковій і кінцевій точках відрізка інтегрування. Третій стовпець визначає тип умов. Якщо для функції задаються умови Діріхле, то в третьому стовпці відповідного рядка потрібно ввести «D», а якщо умови Неймана – то необхідно ввести «N». Якщо в

системі відсутня друга похідна тієї чи іншої функції за просторовою змінною, то для неї потрібно задавати тільки одну граничну умову, а замість другої ввести «NA» (при цьому в третьому стовпці обов'язково має стояти «D»). Якщо ж для деякої функції в системі зовсім відсутні похідні за просторовою змінною, то граничні умови для неї не потрібні (відповідний рядок в матриці bc має виглядати так: «NA», «NA», «D»).

4. Викликати функцію *numol* з усіма параметрами.



В функції *numol* не дозволяється використовувати змішані граничні умови, якщо на одній границі інтервалу задається значення самої функції, а на іншій – її похідної. Для таких задач необхідно використовувати тільки функцію *Pdesolve*.

Функція *numol* транспонує матрицю розв'язку диференціального рівняння в частинних похідних в кожній точці за просторовою (по рядках) і часовою координатою (по стовпцях). Якщо розв'язується не одне рівняння, а їх система, то результуюча матриця буде утворена шляхом сполучення зліва направо матриць із значеннями кожної шуканої сітчастої функції.

3.1.2 Розв'язання параболічного рівняння теплопровідності за допомогою функції *Pdesolve*

В літературі можна зустріти іншу назву рівняння теплопровідності – рівняння Фур'є.

Температуру фізичного тіла в довільній точці з координатами (x, y, z) в момент часу t можна подати у вигляді функції:

$$u = u(x, y, z).$$

Розглянемо диференціальне рівняння в частинних похідних:

$$\frac{\partial u}{\partial t} = a^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right). \quad (3.2)$$

Рівняння (3.2) можна записати у вигляді:

$$\frac{\partial u}{\partial t} = a^2 \Delta, \quad (3.3)$$

де Δ – оператор Лапласа, що дорівнює:

$$\Delta = \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right). \quad (3.4)$$

Рівняння (3.3) називається *рівнянням теплопровідності в просторі* або *рівнянням Фур'є*.

Це рівняння вважають найпростішим рівнянням *параболічного* типу. Воно описує процеси теплопровідності, фільтрації рідин і газів, а також деякі задачі в теорії ймовірності та статичній радіотехніці.

Можуть мати місце такі випадки:

- $\frac{\partial u}{\partial t} = a^2 \left(\frac{\partial^2 u}{\partial x^2} \right)$ – рівняння теплопровідності в стержні;
- $\frac{\partial u}{\partial t} = a^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$ – рівняння теплопровідності на площині.

Розглянемо приклад розв'язання рівняння теплопровідності в стержні за допомогою функції *Pdesolve*.

Розв'яжемо рівняння теплопровідності в стержні

$$\frac{\partial u}{\partial t} = a^2 \left(\frac{\partial^2 u}{\partial x^2} \right),$$



де $a = 1$ з такими граничними умовами:

$$u(0, t) = t,$$

$$u(L, t) = 0,$$

$$u(x, 0) = \sin(x),$$

$$L = 0..10, \quad T = 0..1.$$

Для розв'язання цієї задачі необхідно:

1. Записати ключове слово *Given*.
2. Записати рівняння, яке необхідно розв'язання. Формат запису має бути таким:

$$u_t(x, t) = f(x, t, u, u_x, u_{xx}).$$

3. Записати граничні умови для функції $u(x, t)$. Якщо рівняння другого порядку за x , то і граничних умов має бути дві. Записувати можна як умови Діріхле $u(x_0, t) = w(t)$, так і умови Неймана $u_x(x_0, t) = w(t)$ або їх комбінацію.

4. Ввести початкове значення для невідомої функції $u(x, 0)$.

Програма для розв'язання цієї задачі подана на рисунку 3.2, графічна інтерпретація розв'язку – на рисунку 3.3.

$$c := 1 \quad L := 10 \quad T := 1$$

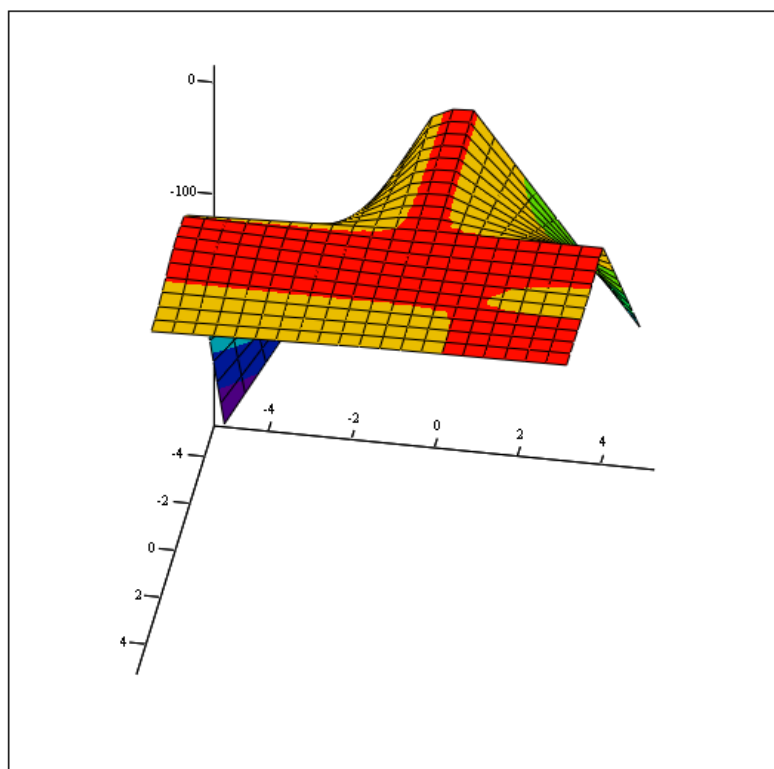
Given

$$u_t(x, t) = c^2 \cdot u_{xx}(x, t) \quad u_x(L, t) = 0$$

$$u(x, 0) = \sin(x) \quad u(0, t) = t$$

$$Y := \text{Pdesolve} \left[u, x, \begin{pmatrix} 0 \\ L \end{pmatrix}, t, \begin{pmatrix} 0 \\ T \end{pmatrix} \right]$$

Рисунок 3.2 – Розв'язання рівняння теплопровідності за допомогою функції Pdesolve



у

Рисунок 3.3 – Графічний розв'язок рівняння теплопровідності

Таким чином, спочатку необхідно покрити розрахункову область (x, t) сіткою і використовувати вузли цієї сітки для різницевої апроксимації рівняння. Як результат, замість пошуку неперервних залежностей $u(x, t)$ досить буде відшукати значення функції у вузлах сітки, а її поведінка в інтервалах між вузлами може бути описана за допомогою побудови будь-

якої інтерполяційної кривої. З цієї причини дискретне подання функції часто називають сітчастою функцією.

Оскільки рівняння в частинних похідних за означенням залежать від похідних невідомих функцій за декількома змінними, то способів дискретизації цих рівнянь може бути, як правило, декілька. Конфігурацію вузлів, що використовується для різницевого запису рівнянь в частинних похідних на сітці, називають *шаблоном*, а одержану систему різницевих рівнянь – *різницевою схемою*.

3.1.3 Розв’язання гіперболічного хвильового рівняння за допомогою функції `Pdesolve`

Функція `Pdesolve` також дозволяє розв’язувати системи диференціальних рівнянь в частинних похідних першого порядку за часом. Така можливість може бути використана для розв’язання задач з диференціальними рівняннями *гіперболічного* типу.

Оскільки рівняння гіперболічного типу містять другу похідну за часом, то вони не можуть бути безпосередньо введені для розв’язання функцією `Pdesolve`. Диференціальне рівняння гіперболічного типу має бути зведене до системи з двох рівнянь першого порядку за часом (як це робилося раніше для звичайних диференціальних рівнянь високих порядків). А далі одержану задачу необхідно розв’язувати як систему рівнянь.

Для кожної невідомої функції має бути задане початкове значення. Кількість граничних умов для кожної функції може бути різною: якщо в системі зустрічається друга похідна від цієї функції за просторовою змінною, то потрібно задати дві граничні умови; якщо тільки перша похідна – одну умову; якщо похідних за просторовою змінною від цієї функції в системі немає, то і граничних умов для неї не потрібно.

Розглянемо приклад застосування функції `Pdesolve` для розв’язання одновимірного хвильового рівняння, яке описує вільні коливання струни музичного інструмента:

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2}. \quad (3.5)$$

Тут невідома функція $u(x,t)$ описує динаміку зсуву профілю струни відносно прямолінійного положення, а параметр c характеризує матеріал, з якого виготовлено струну.

Рівняння (3.5) містить похідні другого порядку як за просторовою координатою, так і за часом. Для того, щоб розв’язати це рівняння за допомогою функції `Pdesolve`, необхідно переписати хвильове рівняння у вигляді системи двох рівнянь в частинних похідних та ввести другу невідому функцію $u = v_t$.

Розв'яжемо хвильове рівняння:

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2}, \quad (3.6)$$



де $c = 1$ з такими граничними умовами:

$$u(0,t) = 0, u(L,t) = 0, u(x,0) = \sin\left(\frac{x}{2}\right),$$

$$L = 0..2\pi, T = 0..10.$$

Для розв'язання цієї задачі необхідно:

1. Записати ключове слово *Given*.
2. Записати систему рівнянь, яку необхідно розв'язати. Формат запису після проведення заміни має бути таким:

$$v_t(x,t) = f(x,t,u,u_x,u_{xx}),$$

$$u_t(x,t) = v(x,t).$$

3. Записати граничні умови для функції $u(x,t)$. Записувати можна як умови Діріхле $u(x_0,t) = w(t)$, так і умови Неймана $u_x(x_0,t) = w(t)$ або їх комбінацію.

4. Ввести початкове значення (або функцію $f(x)$) для невідомих функцій $u(x,0)$ і $v(x,0)$.

Програму розв'язання хвильового рівняння подано на рисунку 3.4, а графічний розв'язок рівняння – на рисунку 3.5.

$$\underline{c} := 1 \quad \underline{L} := 2 \cdot \pi \quad \underline{T} := 10$$

Given

$$v_t(x,t) = c^2 \cdot u_{xx}(x,t) \quad u_t(x,t) = v(x,t) \quad v(x,0) = 0$$

$$u(x,0) = \sin\left(\frac{x}{2}\right) \quad u(0,t) = 0 \quad u(L,t) = 0$$

$$\begin{pmatrix} U \\ \underline{V} \end{pmatrix} := \text{Pdsolve} \left[\begin{pmatrix} u \\ v \end{pmatrix}, x, \begin{pmatrix} 0 \\ L \end{pmatrix}, t, \begin{pmatrix} 0 \\ T \end{pmatrix} \right]$$

Рисунок 3.4 – Розв'язання хвильового рівняння за допомогою функції *Pdsolve*

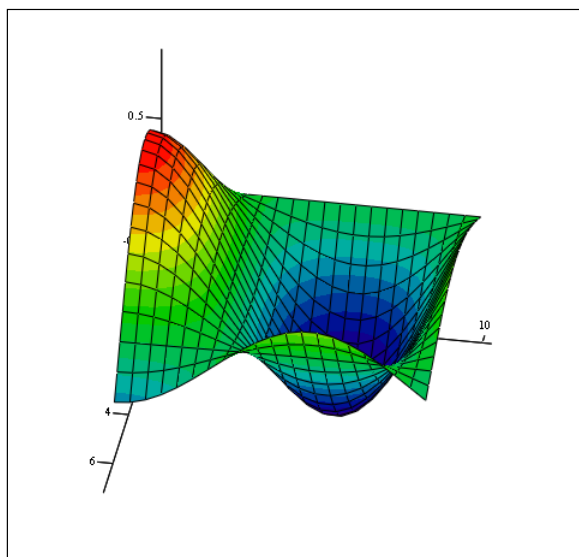


Рисунок 3.5 – Графічний розв’язок хвильового рівняння за допомогою функції *Pdesolve*

3.1.4 Розв’язання хвильового рівняння за допомогою функції *numol*

Розглянемо приклад розв’язання хвильового рівняння (рисунок 3.6) за допомогою функції *numol*.

При розв’язанні рівняння особливу увагу необхідно звернути на формат подання векторів *rhs*, *init* і *bc*, а також принципу виділення окремих сіткових рішень із результувальної матриці.

Програму розв’язання хвильового рівняння за допомогою функції подано на рисунку 3.6, а графічний розв’язок рівняння – на рисунку 3.7.

```

c := 1      L := 2*pi      T := 10
pdef(x, t, u, u_x, u_xx) := (
  u_1
  c^2 * u_xx_0
)
init(x) := (
  sin(x/2)
  0
)          bc(t) := (
  init(0)_0  init(L)_0  "D"
  "NA"      "NA"      "D"
)
sol := numol(
  [(0), (L)], 40, [(0), (T)], 30, 2, 0, pdef, init, bc
)
U := submatrix(sol, 0, 39, 0, 29)

```

Рисунок 3.6 – Розв’язання хвильового рівняння за допомогою функції *numol*

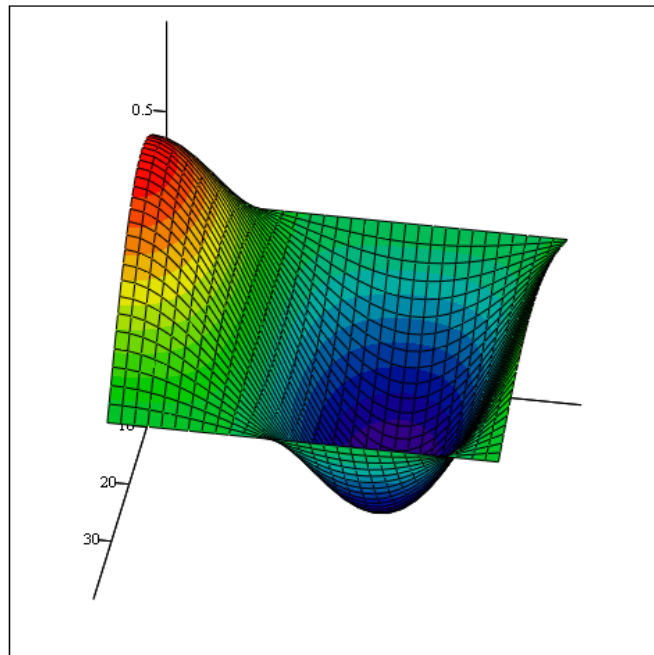


Рисунок 3.7 – Графічний розв’язок хвильового рівняння за допомогою функції *numol*

Якщо порівняти графіки розв’язків рисунок 3.5 та рисунок 3.7, можна побачити їх повну відповідність. Проте використовувати функцію *numol* досить складно, оскільки вона має велику кількість параметрів, а початкові і граничні умови необхідно записати в спеціальному форматі.

Використання функції *numol* виправдане, коли необхідно внести розв’язання диференціальних рівнянь в частинних похідних в більш складні обчислення як підпрограму та коли необхідно організувати серію розрахунків із змінними параметрами чи підготувати анімацію графіків тощо.

3.2 Еліптичні рівняння

Розв’язання еліптичних рівнянь в частинних похідних в MathCAD реалізоване тільки для єдиного типу задач – для двовимірного рівняння Пуассона. За допомогою рівняння Пуассона можна описати, наприклад, розподіл електростатичного поля $u(x, y)$ у двовимірній області з щільністю заряду $f(x, y)$ стаціонарний розподіл температури $u(x, y)$ на площині, в якій є джерела (або споживачі) тепла з інтенсивністю $f(x, y)$.

Рівняння Пуассона містить другі похідні функції $u(x, y)$ за двома просторовими змінними:

$$\frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} = -f(x, y). \quad (3.7)$$



Не дивлячись на те, що застосування вбудованих функцій, описаних в цьому розділі, анонсовано розробниками MathCAD тільки для рівняння Пуассона, їх можна застосовувати і для розв'язання інших рівнянь (необов'язково еліптичного типу).

3.2.1 Рівняння Пуассона з нульовими граничними умовами

Коректна постановка крайової задачі для рівняння Пуассона потребує запису граничних умов. В MathCAD розв'язок відшуковується на плоскій квадратній області, що складається з $(M + 1)(M + 1)$ точок. Тому граничні умови мають бути визначені користувачем для всіх чотирьох сторін такої області.

У випадку нульових граничних умов (тобто температура по всьому периметру розрахункової області постійна) необхідно використовувати вбудовану функцію *multigrd*.

Рівняння Пуассона з нульовими граничними умовами в науковій літературі також називають *рівнянням Лапласа*.

Параметри функції *multigrd* та їх опис подано в таблиці 3.3.

Таблиця 3.3 – Опис параметрів функції *multigrd*

Формат виклику функції: <i>multigrd(F, ncycle)</i>	
Параметри	Опис параметра
<i>F</i>	Матриця розміру $(m + 1)(m + 1)$, що визначає праву частину рівняння Лапласа.
<i>ncycle</i>	Параметр чисельного алгоритму (кількість циклів в межах кожної ітерації). В більшості випадків цей параметр досить взяти 2.

Функція *multigrd* розв'язує рівняння Пуассона розміром $(M + 1)(M + 1)$ на квадратній області з нульовими граничними умовами.

Результатом функції *multigrd* є матриця розміру M , що містить розв'язок рівняння в усіх точках області. Точність обчислень в цьому випадку ніяк не контролюється і визначається тільки розмірами матриці M – чим дрібніше буде розбита область інтегрування, тим точнішим буде результат.



Сторона квадрата розрахункової області має містити точно $m = 2^n$ кроків, тобто $2^n + 1$ вузол, де n – ціле число.

Розглянемо приклад використання функції *multigrid* для розрахунку крайової задачі на області розміром 33×33 точки і точковим джерелом тепла в місці, що задається координатами (15, 20) всередині цієї області.

Програму розрахунку подано на рисунку 3.8.

$M := 32$ $F_{M,M} := 0$
 $F_{15,20} := 10^4$ $G := \text{multigrid}(-F, 2)$

G =

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	$7.512 \cdot 10^{-3}$	0.015	0.023	0.031	0.038
2	0	0.015	0.03	0.045	0.061	0.077
3	0	0.022	0.045	0.068	0.091	0.115
4	0	0.03	0.059	0.09	0.12	0.152
5	0	0.037	0.073	0.111	0.149	0.188
6	0	0.043	0.087	0.131	0.177	0.223
7	0	0.05	0.1	0.151	0.203	0.257
8	0	0.056	0.112	0.169	0.228	0.289
9	0	0.061	0.123	0.186	0.251	0.318
10	0	0.066	0.133	0.201	0.271	0.344
11	0	0.07	0.142	0.214	0.289	0.367
12	0	0.074	0.149	0.225	0.304	0.386
13	0	0.077	0.154	0.233	0.315	0.401
14	0	0.078	0.158	0.239	0.323	0.411
15	0	0.079	0.16	0.242	0.327	0.416

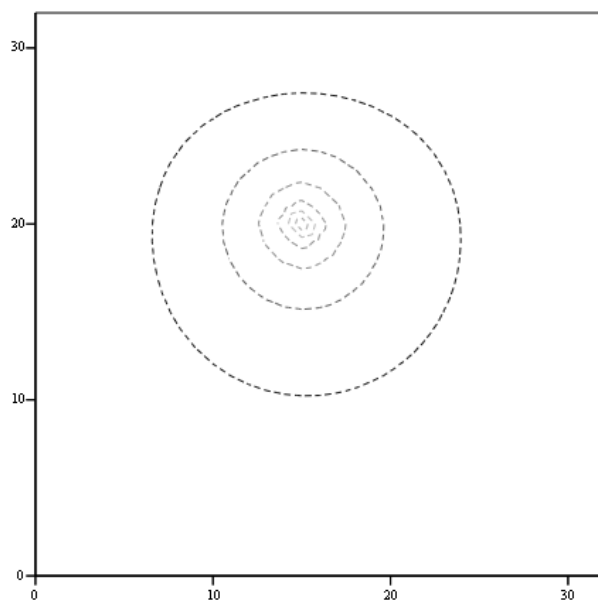
Рисунок 3.8 – Розв’язок рівняння Пуассона з нульовими граничними умовами

У першому рядку задається значення $M = 32$ (тобто $n = 5$), у двох наступних рядках створюється матриця правої частини рівняння Пуассона, що складається зі всіх нульових елементів, за винятком одного, що задає розташування джерела. В останньому рядку матриці G присвоюється результат дії функції *multigrid*.



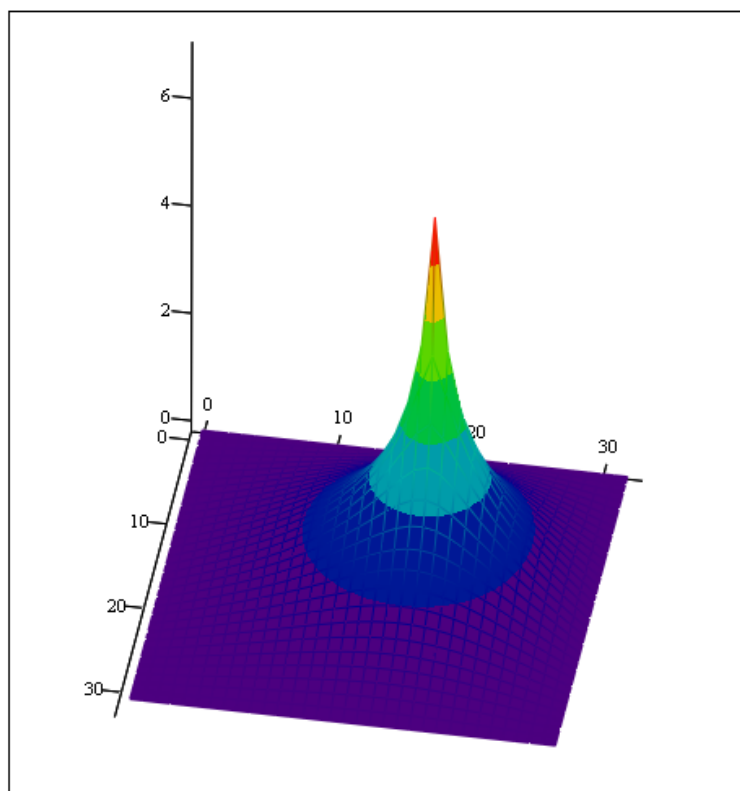
Перший аргумент функції *multigrid* записується зі знаком «мінус», що відповідає запису правої частини рівняння Пуассона (3.6).

На рисунках 3.9 і 3.10 подано розв’язок рівняння відповідно у вигляді ліній рівня і тривимірної поверхні.



G

Рисунок 3.9 – Розв’язок рівняння Пуассона у вигляді ліній рівня



G

Рисунок 3.10 – Розв’язок рівняння Пуассона у вигляді поверхні

3.2.2 Рівняння Пуассона з довільними граничними умовами

У більш складних випадках, коли хоча б одна умова відмінна від нуля, для розв’язання крайової задачі необхідно використовувати функцію *relax*. Параметри функції *relax* та їх описи подано в таблиці 3.4.

Таблиця 3.4 – Параметри функції *relax*

Формат виклику функції: <i>multigrid(F, ncycle)</i>	
Параметри	Опис параметра
a, b, c, d, e	Квадратні матриці коефіцієнтів різницевої схеми, що апроксимує диференціальне рівняння.
F	Квадратна матриця, яка задає праву частину диференціального рівняння.
ν	Квадратна матриця граничних умов і початкового наближення.
$rjac$	Параметр чисельного алгоритму (спектральний радіус ітерацій Якобі). Він характеризує швидкість збіжності ітерацій і має бути дійсним числом від 0 до 1.

Результатом роботи функції *relax* є матриця розв'язку диференціального рівняння в частинних похідних на квадратній області, яка одержана за допомогою алгоритму релаксації для методу сіток.

В матриці граничних умов ν необхідно задати тільки граничні елементи, виходячи зі значення крайових умов по периметру розрахункової області. Інші (внутрішні) елементи цієї матриці слугують для запису початкового наближення до розв'язку.

Суть алгоритму релаксації зводиться до того, що в ході ітерацій відбувається перевірка рівнянь і відповідна корекція значень функції в кожній точці. Якщо початкове наближення вибране вдало, то можна сподіватися, що алгоритм буде швидко збігатися.

Всі матриці, що задаються як коефіцієнти різницевої схеми a, b, c, d, e , граничні умови ν , так і сам розв'язок F , мають бути однакового розміру $(M + 1)(M + 1)$, відповідного до розміру розрахункової області. При цьому ціле число M обов'язково має бути степенем двійки: $M = 2^n$.

Для розв'язання диференціальних рівнянь за допомогою функції *relax* необхідно:

1. Задати п'ять квадратних матриць a, b, c, d, e . Ці матриці є коефіцієнтами в формулі наближеного обчислення лапласіана (оператора Лапласа):

$$(\Delta u)_{i,j} = a_{i,j}u_{i+1,j} + b_{i,j}u_{i-1,j} + c_{i,j}u_{i,j+1} + d_{i,j}u_{i,j-1} + e_{i,j}u_{i,j}.$$

2. Стандартні значення для елементів цих матриць:

$$a_{i,j} = b_{i,j} = c_{i,j} = d_{i,j} = 1, \quad e_{i,j} = -4.$$

Розмір цих матриць може бути будь-яким.

3. Задати матрицю s , яка визначає інтенсивність джерела в кожній точці заданої області. Якщо всі елементи цієї матриці будуть дорівнювати нулю, то отриманий результат буде розв'язком рівняння Лапласа.

4. Задати матрицю f . Перший і останній стовпці та перший і останній рядки цієї матриці задають граничні умови.

5. Безпосередній виклик функції $relax$ із відповідними параметрами.

Розв'язок рівняння Пуассона з трьома джерелами різної інтенсивності за допомогою функції $relax$ подано на рисунку 3.11. Розв'язок рівняння у вигляді тривимірної поверхні і ліній рівня подано відповідно на рисунках 3.12 і 3.13.

$M := 32$

$F_{M,M} := 0$

$F_{25,20} := 50$ $F_{10,10} := -5$

$i := 0..M$ $k := 0..M$

$a_{i,k} := 1$

$b := a$

$c := a$

$d := a$

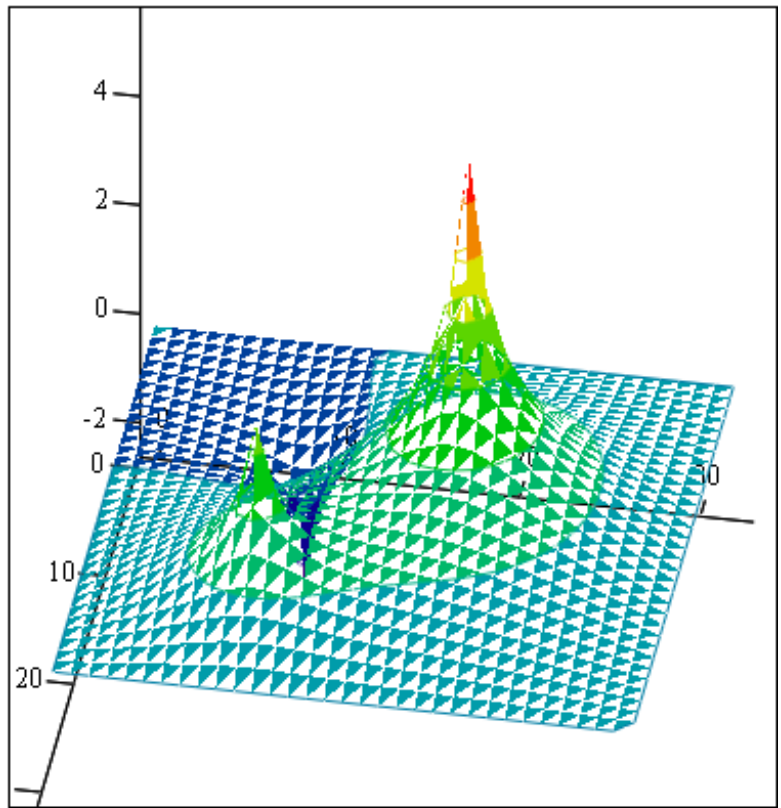
$e := -4 \cdot a$

$v_{i,k} := 0$

$G := relax(a, b, c, d, e, -F, v, 0.95)$

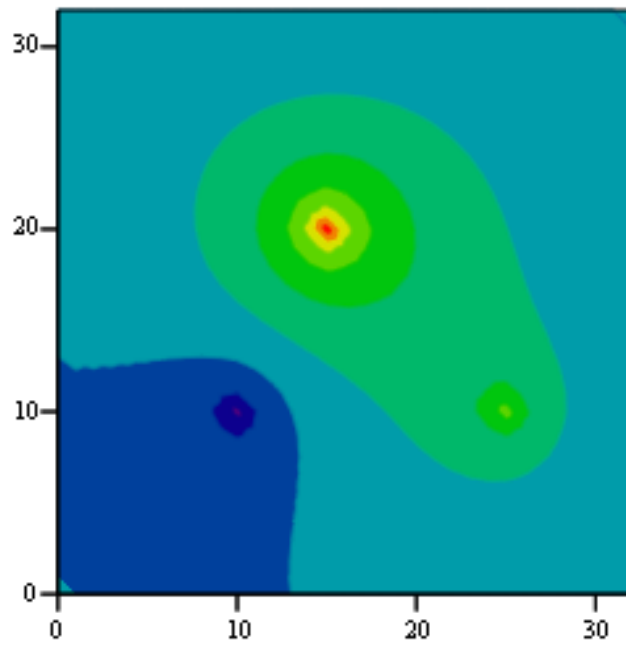
	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	$-5.72 \cdot 10^{-3}$	-0.011	-0.017	-0.022	-0.027
2	0	-0.011	-0.023	-0.034	-0.045	-0.056
3	0	-0.017	-0.034	-0.052	-0.069	-0.086
4	0	-0.023	-0.046	-0.07	-0.094	-0.118
5	0	-0.028	-0.057	-0.087	-0.119	-0.152
6	0	-0.033	-0.066	-0.103	-0.143	-0.186
7	0	-0.036	-0.073	-0.115	-0.163	-0.218
8	0	-0.037	-0.076	-0.121	-0.175	-0.241
9	0	-0.035	-0.074	-0.119	-0.175	-0.248
10	0	-0.031	-0.065	-0.106	-0.159	-0.23
11	0	-0.023	-0.048	-0.081	-0.123	-0.182
12	0	-0.012	-0.026	-0.045	-0.072	-0.11
13	0	$1.801 \cdot 10^{-3}$	$1.96 \cdot 10^{-3}$	$-1.162 \cdot 10^{-3}$	$-9.133 \cdot 10^{-3}$	-0.023
14	0	0.017	0.033	0.047	0.06	0.07
15	0	0.033	0.066	0.098	0.131	0.164

Рисунок 3.11 – Розв'язок рівняння Пуассона за допомогою функції $relax$



G

Рисунок 3.12 – Розв’язок рівняння Пуассона у вигляді поверхні



G

Рисунок 3.13 – Розв’язок рівняння Пуассона у вигляді ліній рівня

Питання для самоконтролю

1. На які типи поділяються ДР в частинних похідних?
2. Які функції використовуються для розв'язання параболічних і гіперболічних рівнянь?
3. В яких випадках необхідно використовувати функцію *numol*? В чому її переваги і недоліки?
4. Чи можна використовувати функцію *numol*, якщо задано змішані граничні умови?
5. Який алгоритм розв'язання рівняння теплопровідності за допомогою функції *Pdesolve*?
6. Які функції використовуються для розв'язання еліптичних рівнянь?
7. Що таке різницева схема?
8. Яка функція використовується для розв'язання рівняння Пуассона з нульовими граничними умовами?
9. Яка функція використовується для розв'язання рівняння Пуассона з довільними граничними умовами?
10. Чи можна використовувати функції *relax* і *multigrid* для розв'язання рівнянь нееліптичного типу?

Задачі для самоконтролю

Розв'язати рівняння в частинних похідних:

$$\text{a) } \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \text{ з умовами } \begin{cases} u(0,t) = t, \\ u(L,t) = 1, \\ u(x,0) = \cos(x^2), \\ L = 0..1, \quad T = 0..1. \end{cases}$$

$$\text{b) } \frac{\partial^2 u(x,t)}{\partial t^2} = \frac{\partial^2 u(x,t)}{\partial x^2} \text{ з умовами } \begin{cases} u(0,t) = 10, \\ u(L,t) = t, \\ u(0,t) = t - 1, \\ u(x,0) = x^2, \\ L = 0..2, \quad T = 0..10. \end{cases}$$

4 ДОСЛІДЖЕННЯ СИСТЕМ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

4.1 Побудова інтегральної кривої задачі Коші

Інтегральна крива – це лінія в тривимірному просторі (y_1, y_2, x) , задана в параметричній формі рівняннями $y_1 = y_1(x)$, $y_2 = y_2(x)$, $x = x$, що проходить через точку з координатами $(1, 0, 0)$ (функції $y_1 = y_1(x)$, $y_2 = y_2(x)$ – розв’язки задачі Коші).

Побудуємо інтегральну криву для розв’язання задачі Коші:

$$y_1(0) = 1,$$

$$y_2(0) = 0$$

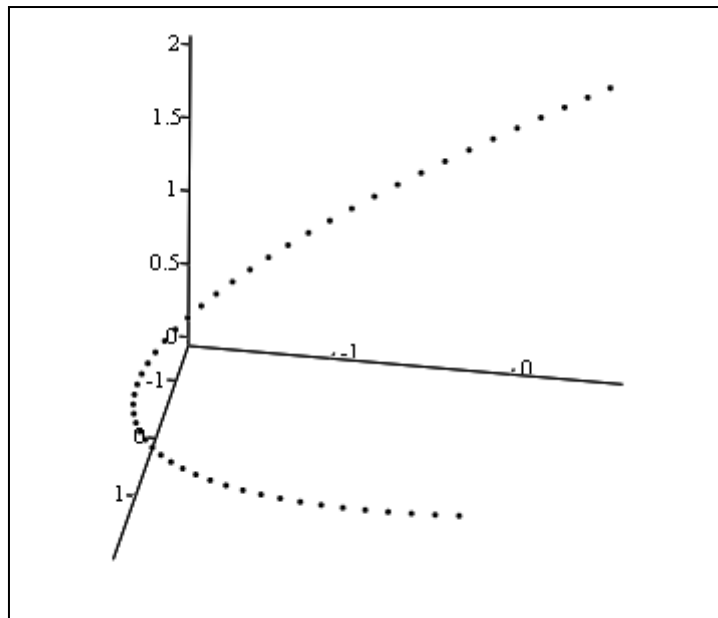
на інтервалі $[0, 2]$ з кроком $h=0,04$. Для цього визначимо початкову умову і праву частину системи та знайдемо наближений розв’язок:

ORIGIN := 1

$$y := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad D(x, y) := \begin{pmatrix} 3 \cdot y_1 + 4y_2 \\ -3y_1 - 3y_2 \end{pmatrix}$$

Y := rkfixed(y, 0, 2, 50, D)

Побудуємо інтегральну криву (рисунок 4.1).



$(Y^{(2)}, Y^{(3)}, Y^{(1)})$

Рисунок 4.1 – Інтегральна крива для розв’язання задачі Коші

4.2 Побудова векторного поля автономної системи

Для побудови векторного поля системи диференціальних рівнянь другого порядку необхідно спочатку визначити праві частини системи, потім на прямокутнику визначити квадратну сітку і обчислити праві частини у вузлах цієї сітки.

Для того, щоб побудувати векторне поле, потрібно натиснути в панелі *Graph* на піктограмі векторного поля (*Vector Field Plot*) і ввести в поміченій позиції імена матриць, що містять компоненти векторного поля.

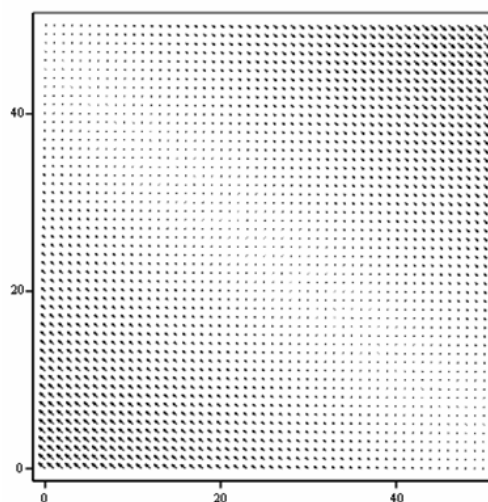
Побудуємо векторне поле системи другого порядку:

$$\begin{cases} y_1' = 3y_1 + 4y_2, \\ y_2' = -3y_1 - 3y_2 \end{cases}$$

в прямокутній області $[-2,2][[-2,2]$.

Програма розрахунку подана на рисунку 4.2.

```
F1(x,y) := 3x + 4y
F2(x,y) := -3x - 3y
i := 0..50      j := 0..50
Xi := -2 + i· $\frac{4}{50}$     Yj := -2 + j· $\frac{4}{50}$ 
MF1i,j := F1(Xi, Yj)  MF2i,j := F2(Xi, Yj)
```



(MF1, MF2)

Рисунок 4.2 – Побудова векторного поля системи

4.3 Поведінка розв'язків в околі стійкого вузла

Дослідження поведінки розв'язків автономної системи в околі стійкого вузла проведемо на прикладі такої системи звичайних диференціальних рівнянь:

$$\begin{cases} x' = -4x - 4y, \\ y' = 1.5x + y. \end{cases}$$

Для знаходження власних значень системи, яка записана у вигляді матриці, використовується функція *eigenvals*. Для побудови фазових портретів (родини фазових кривих) необхідно розв'язати задачу Коші з різними початковими умовами. Для цього використаємо функцію *rkfixed*.

Запишемо матрицю системи і знайдемо її власні значення:

$$\begin{aligned} \text{ORIGIN} &:= 1 \\ A &:= \begin{pmatrix} -4 & -4 \\ 1.5 & 1 \end{pmatrix} \quad \text{eigenvals}(A) = \begin{pmatrix} -2 \\ -1 \end{pmatrix} \end{aligned}$$

Значення розв'язків – від'ємні дійсні числа. Точка спокою – *стійкий вузол*.

Визначимо праву частину системи $D(t, Y)$:

$$D(t, Y) := \begin{pmatrix} -4Y_1 - 4Y_2 \\ 1.5Y_1 + Y_2 \end{pmatrix}$$

Визначимо декілька початкових умов, розв'яжемо відповідні задачі Коші і побудуємо фазові траєкторії (рисунок 4.3):

$Y1 := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$Y1 := \text{rkfixed}(Y1, 0, 10, 100, D)$	$Y6 := \begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$Y6 := \text{rkfixed}(Y6, 0, 10, 100, D)$
$Y2 := \begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$Y2 := \text{rkfixed}(Y2, 0, 10, 100, D)$	$Y7 := \begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$Y7 := \text{rkfixed}(Y7, 0, 10, 100, D)$
$Y3 := \begin{pmatrix} -1 \\ 0 \end{pmatrix}$	$Y3 := \text{rkfixed}(Y3, 0, 10, 100, D)$	$Y8 := \begin{pmatrix} 1 \\ -1 \end{pmatrix}$	$Y8 := \text{rkfixed}(Y8, 0, 10, 100, D)$
$Y4 := \begin{pmatrix} 0 \\ -1 \end{pmatrix}$	$Y4 := \text{rkfixed}(Y4, 0, 10, 100, D)$	$Y9 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$Y9 := \text{rkfixed}(Y9, 0, 10, 100, D)$
$Y5 := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$Y5 := \text{rkfixed}(Y5, 0, 10, 100, D)$	$Y10 := \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$	$Y10 := \text{rkfixed}(Y10, 0, 10, 100, D)$

Рисунок 4.3 – Розв'язки в околі стійкого вузла та фазові траєкторії

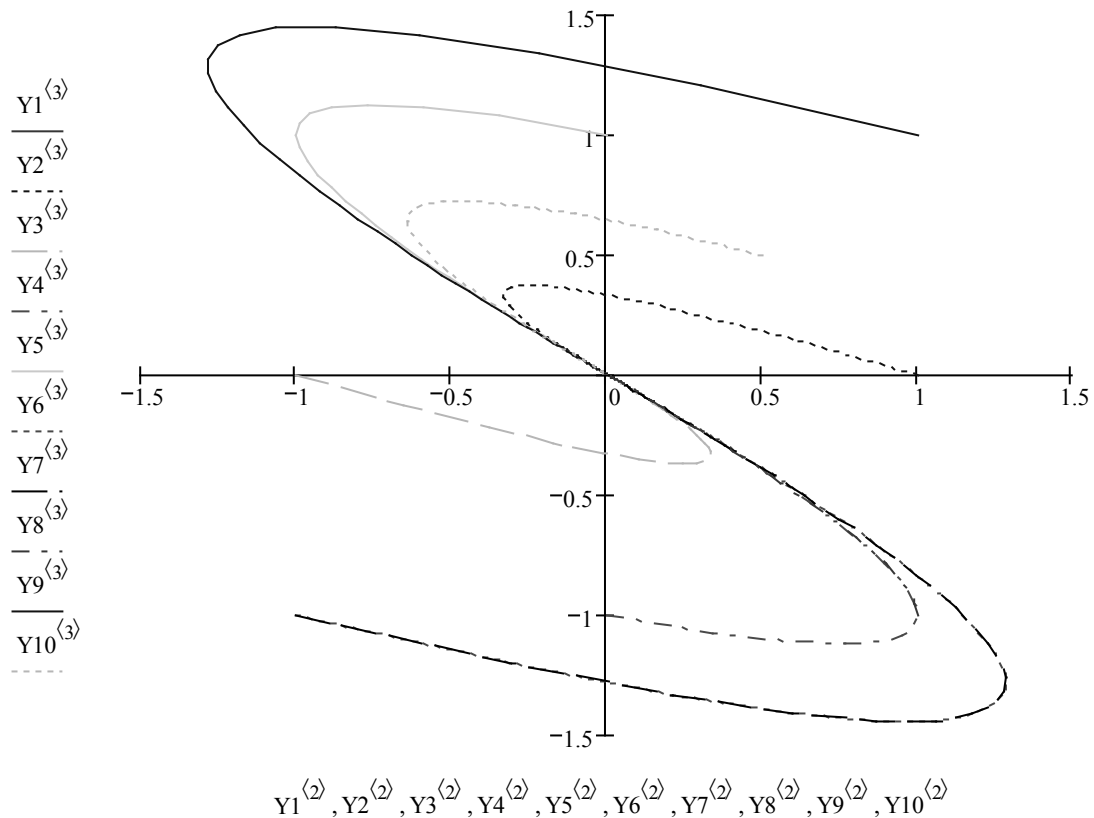


Рисунок 4.3, аркуш 2

4.4 Поведінка розв'язків в околі нестійкого вузла

Аналогічно попередньому прикладу проведемо дослідження поведінки розв'язків автономної системи в околі нестійкого вузла.

Система має вигляд:

$$\begin{cases} x' = 5x + y, \\ y' = x + 5y. \end{cases}$$

Запишемо матрицю системи і знайдемо її власні значення:

$$\text{ORIGIN} := 1$$

$$A := \begin{pmatrix} 5 & 1 \\ 1 & 5 \end{pmatrix} \quad \text{eigenvals}(A) = \begin{pmatrix} 4 \\ 6 \end{pmatrix}$$

Значення розв'язків – додатні дійсні числа. Точка спокою – *нестійкий вузол*.

Визначимо праву частину системи $D(t, Y)$:

$$D(t, Y) := \begin{pmatrix} 5Y_1 - Y_2 \\ Y_1 + 5Y_2 \end{pmatrix}$$

Визначимо декілька початкових умов, розв'яжемо відповідні задачі Коші і побудуємо фазові траєкторії (рисунок 4.4):

$Y1 := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$Y1 := \text{rkfixed}(Y1, 0, 10, 100, D)$	$Y6 := \begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$Y6 := \text{rkfixed}(Y6, 0, 10, 100, D)$
$Y2 := \begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$Y2 := \text{rkfixed}(Y2, 0, 10, 100, D)$	$Y7 := \begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$Y7 := \text{rkfixed}(Y7, 0, 10, 100, D)$
$Y3 := \begin{pmatrix} -1 \\ 0 \end{pmatrix}$	$Y3 := \text{rkfixed}(Y3, 0, 10, 100, D)$	$Y8 := \begin{pmatrix} 1 \\ -1 \end{pmatrix}$	$Y8 := \text{rkfixed}(Y8, 0, 10, 100, D)$
$Y4 := \begin{pmatrix} 0 \\ -1 \end{pmatrix}$	$Y4 := \text{rkfixed}(Y4, 0, 10, 100, D)$	$Y9 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$Y9 := \text{rkfixed}(Y9, 0, 10, 100, D)$
$Y5 := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$Y5 := \text{rkfixed}(Y5, 0, 10, 100, D)$	$Y10 := \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$	$Y10 := \text{rkfixed}(Y10, 0, 10, 100, D)$

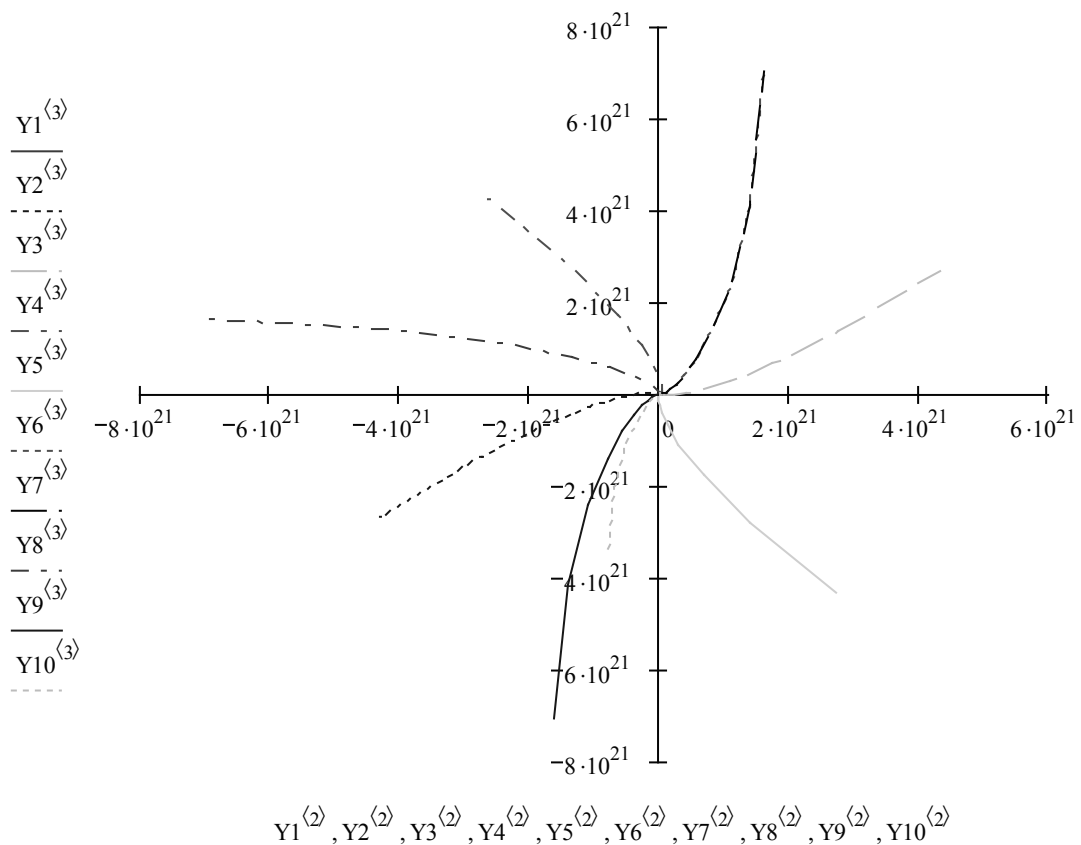


Рисунок 4.4 – Розв'язки в околі нестійкого вузла та фазові траєкторії

4.5 Поведінка розв'язків в околі сідла

Проведемо дослідження поведінки розв'язків автономної системи в околі сідла.

Система має вигляд:

$$\begin{cases} x' = 2x + y, \\ y' = 0.5x - 4y. \end{cases}$$

Запишемо матрицю системи і знайдемо її власні значення:

$$\text{ORIGIN} := 1$$

$$A := \begin{pmatrix} 2 & 1 \\ 0.5 & -4 \end{pmatrix} \quad \text{eigenvals}(A) = \begin{pmatrix} 2.082 \\ -4.082 \end{pmatrix}$$

Значення розв'язків – дійсні числа різних знаків. Точка спокою – *нестійка, сідло*.

Визначимо праву частину системи $D(t, Y)$:

$$D(t, Y) := \begin{pmatrix} 2Y_1 + Y_2 \\ 0.5Y_1 - 4Y_2 \end{pmatrix}$$

Визначимо декілька початкових умов, розв'яжемо відповідні задачі Коші і побудуємо фазові траєкторії (рисунок 4.5):

$Y1 := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$Y1 := \text{rkfixed}(Y1, 0, 1, 100, D)$	$Y6 := \begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$Y6 := \text{rkfixed}(Y6, 0, 1, 100, D)$
$Y2 := \begin{pmatrix} 1 \\ 2 \end{pmatrix}$	$Y2 := \text{rkfixed}(Y2, 0, 1, 100, D)$	$Y7 := \begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$Y7 := \text{rkfixed}(Y7, 0, 1, 100, D)$
$Y3 := \begin{pmatrix} -1 \\ -0.6 \end{pmatrix}$	$Y3 := \text{rkfixed}(Y3, 0, 1, 100, D)$	$Y8 := \begin{pmatrix} 1 \\ -1 \end{pmatrix}$	$Y8 := \text{rkfixed}(Y8, 0, 1, 100, D)$
$Y4 := \begin{pmatrix} 0 \\ -1 \end{pmatrix}$	$Y4 := \text{rkfixed}(Y4, 0, 1, 100, D)$	$Y9 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$Y9 := \text{rkfixed}(Y9, 0, 1, 100, D)$
$Y5 := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$Y5 := \text{rkfixed}(Y5, 0, 1, 100, D)$	$Y10 := \begin{pmatrix} 0.5 \\ 2 \end{pmatrix}$	$Y10 := \text{rkfixed}(Y10, 0, 1, 100, D)$

Рисунок 4.5 – Розв'язки в околі сідла та фазові траєкторії

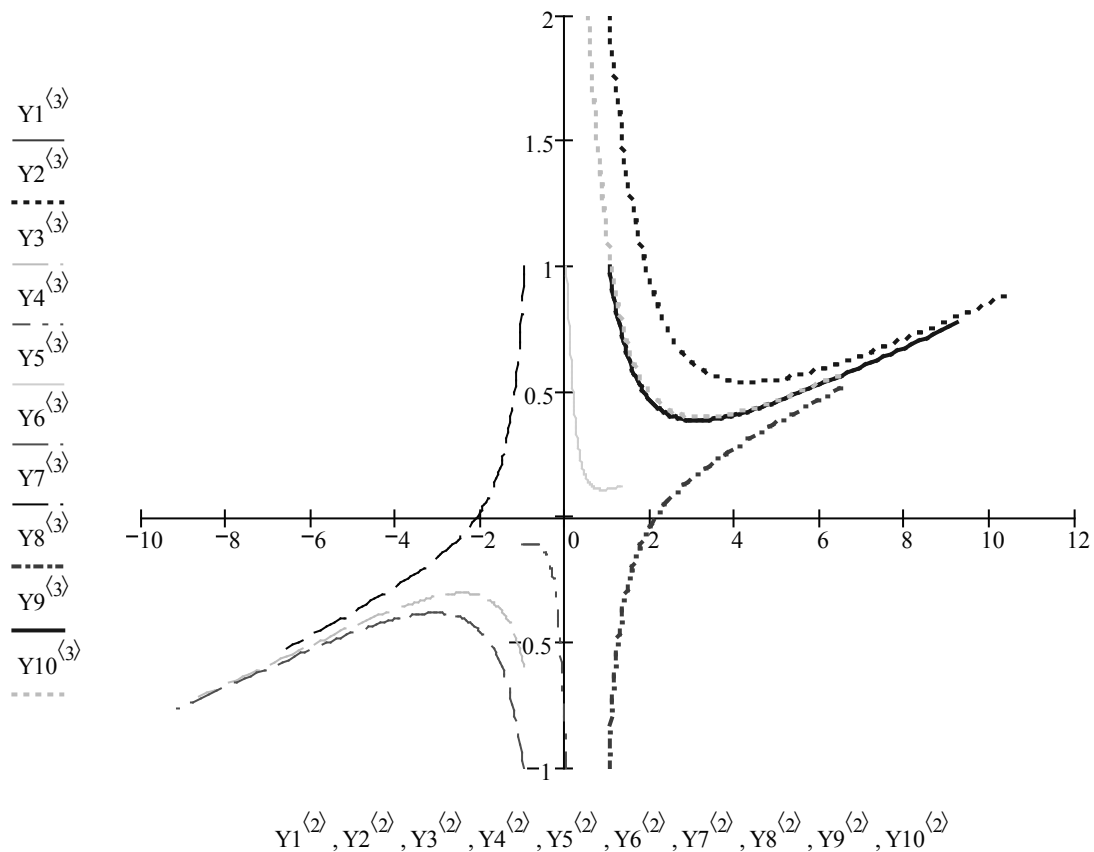


Рисунок 4.5, аркуш 2

4.6 Поведінка розв'язків в околі центра

Проведемо дослідження поведінки розв'язків автономної системи в околі центра.

Система має вигляд:

$$\begin{cases} x' = x - 2y, \\ y' = x - y. \end{cases}$$

Запишемо матрицю системи і знайдемо її власні значення:

$$\text{ORIGIN} := 1$$

$$A := \begin{pmatrix} 1 & -2 \\ 1 & -1 \end{pmatrix} \quad \text{eigenvals}(A) = \begin{pmatrix} i \\ -i \end{pmatrix}$$

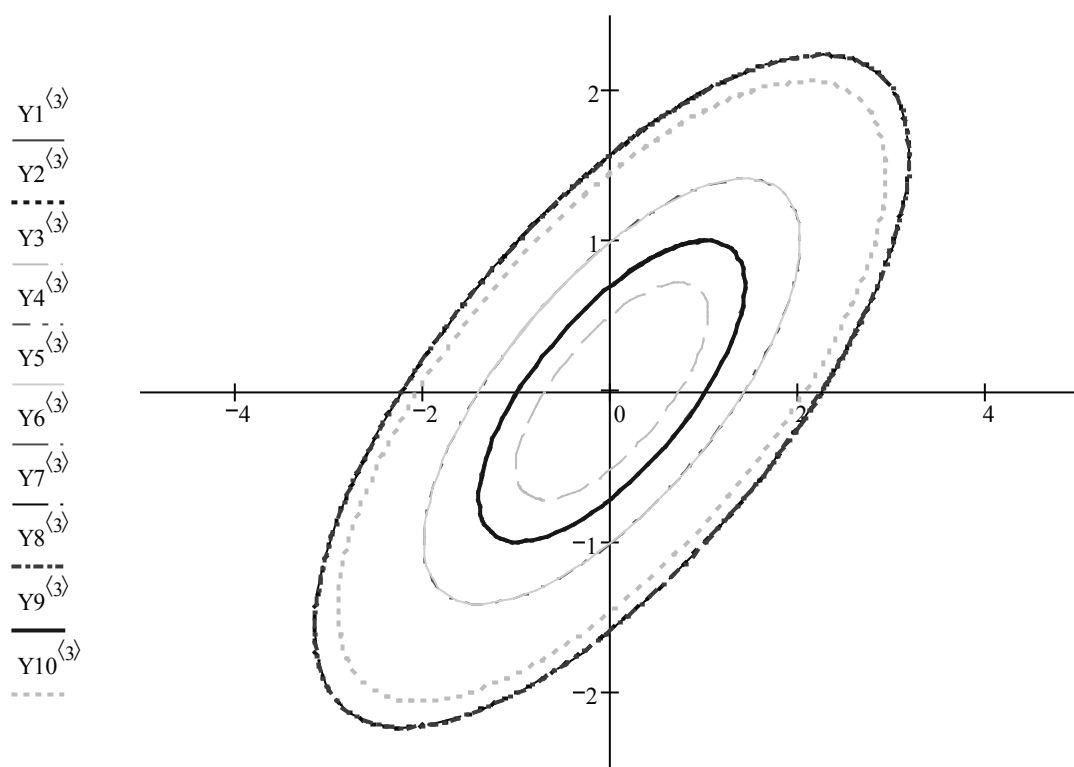
Значення розв'язків – комплексні числа. Точка спокою – *стійка, центр* (але не асимптотично стійка точка спокою).

Визначимо праву частину системи $D(t, Y)$:

$$D(t, Y) := \begin{pmatrix} Y_1 - 2Y_2 \\ Y_1 - Y_2 \end{pmatrix}$$

Визначимо декілька початкових умов, розв'яжемо відповідні задачі Коші і побудуємо фазові траєкторії (рисунок 4.6):

$Y1 := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$Y1 := \text{rkfixed}(Y1, 0, 7, 100, D)$	$Y6 := \begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$Y6 := \text{rkfixed}(Y6, 0, 7, 100, D)$
$Y2 := \begin{pmatrix} 1 \\ 2 \end{pmatrix}$	$Y2 := \text{rkfixed}(Y2, 0, 7, 100, D)$	$Y7 := \begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$Y7 := \text{rkfixed}(Y7, 0, 7, 100, D)$
$Y3 := \begin{pmatrix} -1 \\ -0.6 \end{pmatrix}$	$Y3 := \text{rkfixed}(Y3, 0, 7, 100, D)$	$Y8 := \begin{pmatrix} 1 \\ -1 \end{pmatrix}$	$Y8 := \text{rkfixed}(Y8, 0, 7, 100, D)$
$Y4 := \begin{pmatrix} 0 \\ -1 \end{pmatrix}$	$Y4 := \text{rkfixed}(Y4, 0, 7, 100, D)$	$Y9 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$Y9 := \text{rkfixed}(Y9, 0, 7, 100, D)$
$Y5 := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$Y5 := \text{rkfixed}(Y5, 0, 7, 100, D)$	$Y10 := \begin{pmatrix} 1.5 \\ 2 \end{pmatrix}$	$Y10 := \text{rkfixed}(Y10, 0, 7, 100, D)$



$Y1^{(2)}, Y2^{(2)}, Y3^{(2)}, Y4^{(2)}, Y5^{(2)}, Y6^{(2)}, Y7^{(2)}, Y8^{(2)}, Y9^{(2)}, Y10^{(2)}$

Рисунок 4.6 – Розв'язки в околі центра та фазові траєкторії

4.7 Поведінка розв'язків в околі стійкого фокуса

Проведемо дослідження поведінки рішень автономної системи в околі стійкого фокусу.

Система має вигляд:

$$\begin{cases} x' = 2y, \\ y' = -3x - 0.8y. \end{cases}$$

Запишемо матрицю системи і знайдемо її власні значення:

$$\text{ORIGIN} := 1$$

$$A := \begin{pmatrix} 0 & 2 \\ -3 & -0.8 \end{pmatrix} \quad \text{eigenvals}(A) = \begin{pmatrix} -0.4 + 2.417i \\ -0.4 - 2.417i \end{pmatrix}$$

Значення розв'язків – комплексні числа. Точка спокою – *стійка, фокус* (але не асимптотично стійка точка спокою).

Визначимо праву частину системи $D(t, Y)$:

$$D(t, Y) := \begin{pmatrix} 0Y_1 + 2Y_2 \\ -3Y_1 - 0.8Y_2 \end{pmatrix}$$

Визначимо декілька початкових умов, розв'яжемо відповідні задачі Коші і побудуємо фазові траєкторії (рисунок 4.7):

$Y1 := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$Y1 := \text{rkfixed}(Y1, 0, 10, 100, D)$	$Y6 := \begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$Y6 := \text{rkfixed}(Y6, 0, 10, 100, D)$
$Y2 := \begin{pmatrix} 1 \\ 2 \end{pmatrix}$	$Y2 := \text{rkfixed}(Y2, 0, 10, 100, D)$	$Y7 := \begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$Y7 := \text{rkfixed}(Y7, 0, 10, 100, D)$
$Y3 := \begin{pmatrix} -1 \\ -0.6 \end{pmatrix}$	$Y3 := \text{rkfixed}(Y3, 0, 10, 100, D)$	$Y8 := \begin{pmatrix} 1 \\ -1 \end{pmatrix}$	$Y8 := \text{rkfixed}(Y8, 0, 10, 100, D)$
$Y4 := \begin{pmatrix} 0 \\ -1 \end{pmatrix}$	$Y4 := \text{rkfixed}(Y4, 0, 10, 100, D)$	$Y9 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$Y9 := \text{rkfixed}(Y9, 0, 10, 100, D)$
$Y5 := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$Y5 := \text{rkfixed}(Y5, 0, 10, 100, D)$	$Y10 := \begin{pmatrix} 1.5 \\ 2 \end{pmatrix}$	$Y10 := \text{rkfixed}(Y10, 0, 10, 100, D)$

Рисунок 4.7 – Розв'язки в околі стійкого фокуса та фазові траєкторії

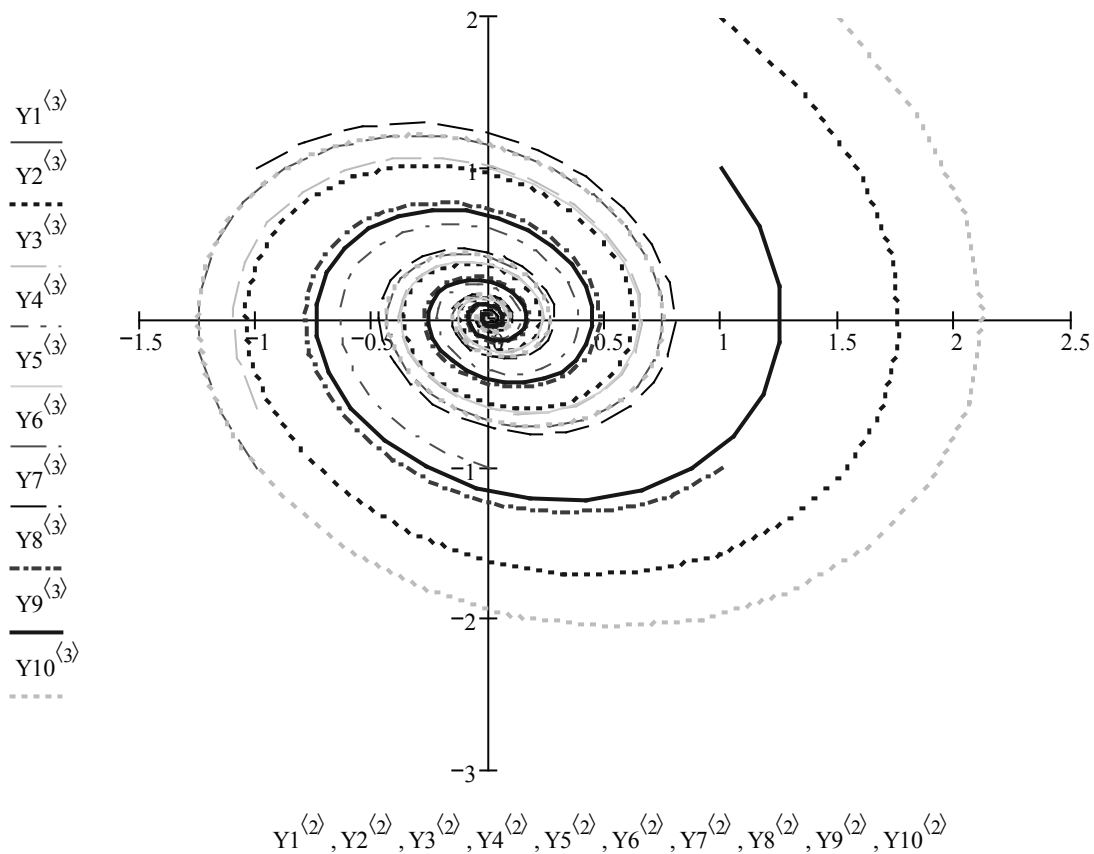


Рисунок 4.7, аркуш 2

4.8 Поведінка розв'язків в околі нестійкого фокуса

Проведемо дослідження поведінки розв'язків автономної системи в околі нестійкого фокуса.

Система має вигляд:

$$\begin{cases} x' = 0.1x - 2y, \\ y' = 2x + 0.1y. \end{cases}$$

Запишемо матрицю системи і знайдемо її власні значення:

$$\text{ORIGIN} := 1$$

$$A := \begin{pmatrix} 0.1 & -2 \\ 2 & 0.1 \end{pmatrix} \quad \text{eigenvals}(A) = \begin{pmatrix} 0.1 + 2i \\ 0.1 - 2i \end{pmatrix}$$

Значення розв'язків – комплексні числа з додатними дійсними частинами. Точка спокою – *нестійкий фокус*.

Визначимо праву частину системи $D(t, Y)$:

$$D(t, Y) := \begin{pmatrix} 0.1 Y_1 - 2 Y_2 \\ 2 Y_1 + 0.1 Y_2 \end{pmatrix}$$

Визначимо декілька початкових умов, розв'яжемо відповідні задачі Коші і побудуємо фазові траєкторії (рисунок 4.8):

$Y1 := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$Y1 := \text{rkfixed}(Y1, 0, 10, 100, D)$	$Y6 := \begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$Y6 := \text{rkfixed}(Y6, 0, 10, 100, D)$
$Y2 := \begin{pmatrix} 1 \\ 2 \end{pmatrix}$	$Y2 := \text{rkfixed}(Y2, 0, 10, 100, D)$	$Y7 := \begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$Y7 := \text{rkfixed}(Y7, 0, 10, 100, D)$
$Y3 := \begin{pmatrix} -1 \\ -0.6 \end{pmatrix}$	$Y3 := \text{rkfixed}(Y3, 0, 10, 100, D)$	$Y8 := \begin{pmatrix} 1 \\ -1 \end{pmatrix}$	$Y8 := \text{rkfixed}(Y8, 0, 10, 100, D)$
$Y4 := \begin{pmatrix} 0 \\ -1 \end{pmatrix}$	$Y4 := \text{rkfixed}(Y4, 0, 10, 100, D)$	$Y9 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$Y9 := \text{rkfixed}(Y9, 0, 10, 100, D)$
$Y5 := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$Y5 := \text{rkfixed}(Y5, 0, 10, 100, D)$	$Y10 := \begin{pmatrix} 1.5 \\ 2 \end{pmatrix}$	$Y10 := \text{rkfixed}(Y10, 0, 10, 100, D)$

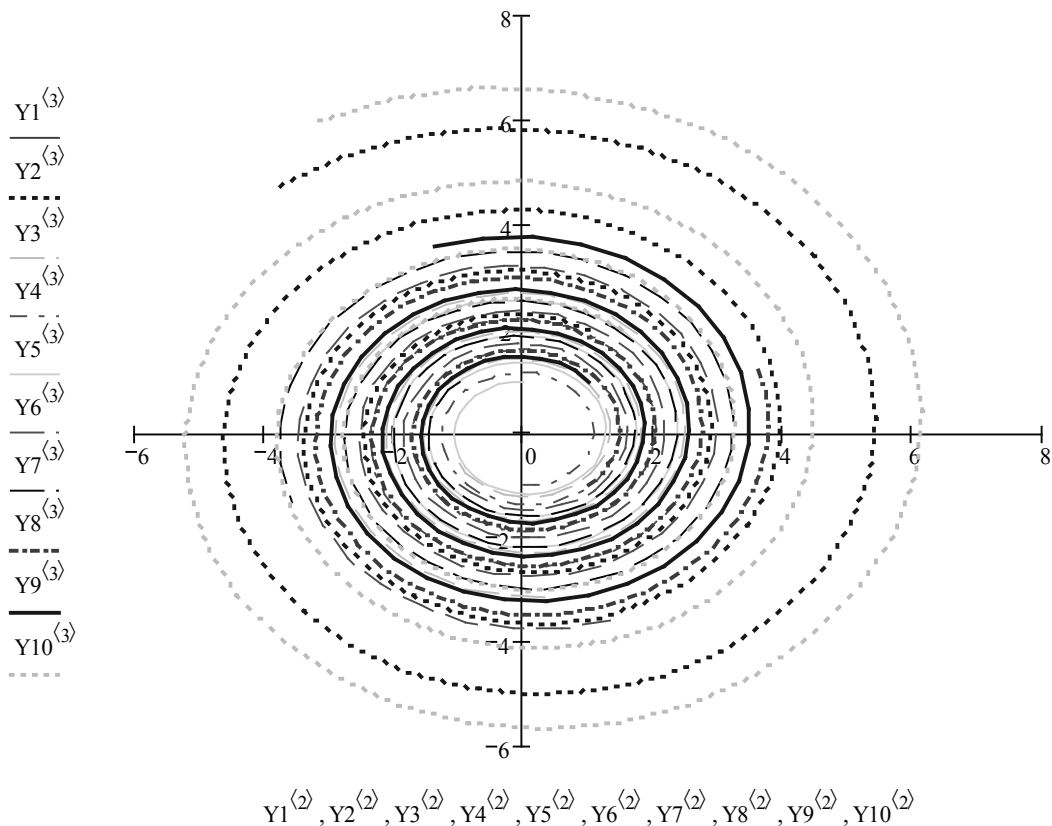


Рисунок 4.8 – Розв'язки в околі нестійкого фокуса та фазові траєкторії

4.9 Поведінка розв'язків в околі діакритичного вузла

Проведемо дослідження поведінки розв'язків автономної системи в околі діакритичного вузла.

Система має вигляд:

$$\begin{cases} x' = 0.1x - 2y, \\ y' = 2x + 0.1y. \end{cases}$$

Запишемо матрицю системи і знайдемо її власні значення:

$$\text{ORIGIN} := 1 \\ A := \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \quad \text{eigenvals}(A) = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

Значення розв'язків – однакові дійсні числа. Точка спокою – *стійкий діакритичний вузол*.

Визначимо праву частину системи $D(t, Y)$:

$$D(t, Y) := \begin{pmatrix} 2Y_1 + 0Y_2 \\ 0Y_1 + 2Y_2 \end{pmatrix}$$

Визначимо декілька початкових умов, розв'яжемо відповідні задачі Коші і побудуємо фазові траєкторії (рисунок 4.9):

$Y1 := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$Y1 := \text{rkfixed}(Y1, 0, 10, 100, D)$	$Y6 := \begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$Y6 := \text{rkfixed}(Y6, 0, 10, 100, D)$
$Y2 := \begin{pmatrix} 1 \\ 2 \end{pmatrix}$	$Y2 := \text{rkfixed}(Y2, 0, 10, 100, D)$	$Y7 := \begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$Y7 := \text{rkfixed}(Y7, 0, 10, 100, D)$
$Y3 := \begin{pmatrix} -1 \\ -0.6 \end{pmatrix}$	$Y3 := \text{rkfixed}(Y3, 0, 10, 100, D)$	$Y8 := \begin{pmatrix} 1 \\ -1 \end{pmatrix}$	$Y8 := \text{rkfixed}(Y8, 0, 10, 100, D)$
$Y4 := \begin{pmatrix} 0 \\ -1 \end{pmatrix}$	$Y4 := \text{rkfixed}(Y4, 0, 10, 100, D)$	$Y9 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$Y9 := \text{rkfixed}(Y9, 0, 10, 100, D)$
$Y5 := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$Y5 := \text{rkfixed}(Y5, 0, 10, 100, D)$	$Y10 := \begin{pmatrix} 1.5 \\ 2 \end{pmatrix}$	$Y10 := \text{rkfixed}(Y10, 0, 10, 100, D)$

Рисунок 4.9 – Розв'язки в околі діакритичного вузла та фазові траєкторії

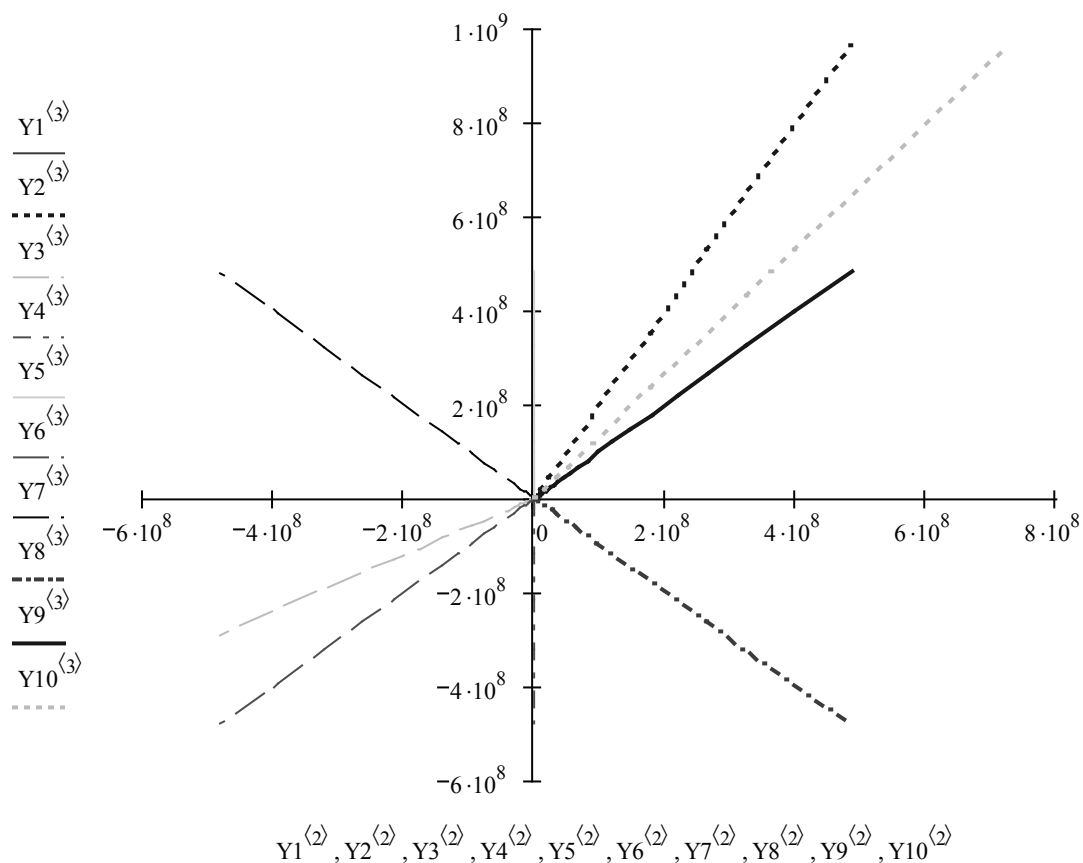


Рисунок 4.9, аркуш 2

Питання для самоконтролю

1. Що таке інтегральна крива?
2. Як будується векторне поле автономної системи?
3. Які значення розв'язків системи, якщо точка спокою – стійкий вузол?
4. Які значення розв'язків системи, якщо точка спокою – нестійкий вузол?
5. Які значення розв'язків системи, якщо точка спокою – сідло?
6. Які значення розв'язків системи, якщо точка спокою – центр?
7. Які значення розв'язків системи, якщо точка спокою – стійкий фокус?
8. Які значення розв'язків системи, якщо точка спокою – нестійкий фокус?
9. Які значення розв'язків системи, якщо точка спокою – діакритичний вузол?
10. За допомогою якої функції знаходять власні значення автономної системи?

Задачі для самоконтролю

1. Побудувати інтегральну криву задачі Коші:

a)
$$\begin{cases} x' = 2x + 5y, \\ y' = 3x - 4y \end{cases}$$

з початковими умовами $x(0) = y(0) = 0$ на інтервалі $[0, 1]$;

b)
$$\begin{cases} x' = 7x - 3y, \\ y' = 10x + y \end{cases}$$

з початковими умовами $x(1) = 1, y(0) = 0$ на інтервалі $[1, 5]$.

2. Побудувати векторне поле задачі Коші:

a)
$$\begin{cases} x' = 3x - 2y, \\ y' = 3x + 2y \end{cases}$$

на прямокутній області $[-2, 2][[-2, 2]$;

b)
$$\begin{cases} x' = 8x + y, \\ y' = 5x + y^2 \end{cases}$$

на прямокутній області $[-1, 1][[-1, 1]$.

3. Дослідити поведінку розв'язків систем і визначити тип точок спокою:

a)
$$\begin{cases} x' = 1.5x - 3y, \\ y' = 2.7x + 10y, \end{cases}$$

b)
$$\begin{cases} x' = 3.3x + 5y, \\ y' = 4x - y, \end{cases}$$

c)
$$\begin{cases} x' = 3x + 10y, \\ y' = x + 10y, \end{cases}$$

d)
$$\begin{cases} x' = 1.4x + 1.5y, \\ y' = 6.7x + 2.6y, \end{cases}$$

e)
$$\begin{cases} x' = 21x + 13y, \\ y' = 3x - 10y, \end{cases}$$

f)
$$\begin{cases} x' = 0.5x + 1.5y, \\ y' = 1.6x - 2.5y, \end{cases}$$

g)
$$\begin{cases} x' = 5x - 3y, \\ y' = 3x + 5y, \end{cases}$$

h)
$$\begin{cases} x' = 11x - 0.1y, \\ y' = 1.1x + 11y, \end{cases}$$

j)
$$\begin{cases} x' = x - y, \\ y' = x + y, \end{cases}$$

k)
$$\begin{cases} x' = -0.1x + 0.2y, \\ y' = -0.5x + 0.7y. \end{cases}$$

5 РОЗВ'ЯЗАННЯ ЗАДАЧ, ЯКІ ОПИСУЮТЬСЯ ДИФЕРЕНЦІАЛЬНИМИ РІВНЯННЯМИ

В цьому розділі подано приклади задач, які зводяться до розв'язання диференціальних рівнянь. Серед них задачі екології, теоретичної електротехніки, поширення хвиль, кінематика тощо.

5.1 Розв'язання системи рівнянь Ван дер Поля

Класичною в теорії коливань є схема автогенератора Ван дер Поля, досліджена в 1920 – 1930 роках Ван дер Полем та О. О. Андроном (рисунок 5.1).

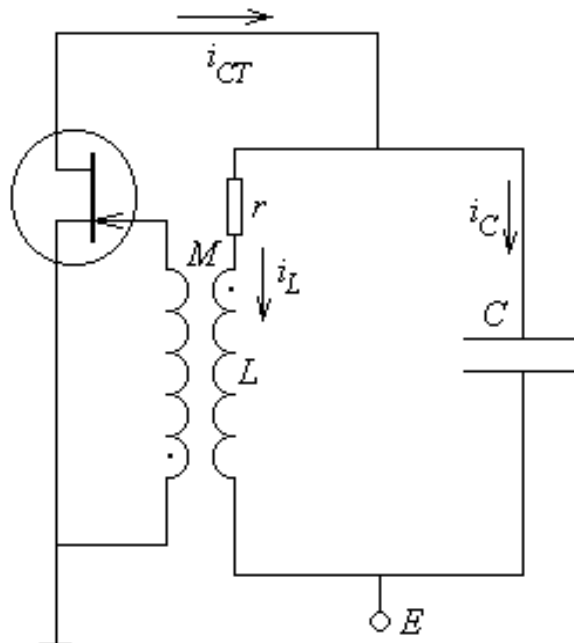


Рисунок 5.1 – Схема автогенератора Ван дер Поля

Вона записується у вигляді рівняння:

$$x'' - (\lambda - x^2)x' + a_0^2x = 0.$$

Система рівнянь Ван дер Поля описує нелінійні коливання при малих значеннях параметрів системи: $\lambda = 0.002$.

Розв'язання системи Ван дер Поля подано на рисунку 5.2.

Розглянемо розв'язання цієї ж системи при великих значеннях параметрів системи: $\lambda = 0.002$.

$$\lambda := 0.0025 \quad a_0 := 1$$

Система диференціальних рівнянь:

$$f1(y1, y2) := y1 \cdot (\lambda - y2^2) - a_0^2 \cdot y2$$

$$f2(y1, y2) := y1$$

Розрахункові формули:

$$fd1(y1, y2, h) := f1\left[y1 + \frac{h}{2} \cdot f1(y1, y2), y2 + \frac{h}{2} \cdot f2(y1, y2)\right]$$

$$fd2(y1, y2, h) := f2\left[y1 + \frac{h}{2} \cdot f1(y1, y2), y2 + \frac{h}{2} \cdot f2(y1, y2)\right]$$

Початкові умови:

$$y1_0 := 1.25 \quad y2_0 := 1.25$$

$$h := 0.05 \quad \text{крок за часом}$$

$$imax := 5000 \quad \text{число кроків}$$

Побудова розв'язку системи:

$$i := 0..imax$$

$$\begin{pmatrix} y1_{i+1} \\ y2_{i+1} \end{pmatrix} := \begin{pmatrix} y1_i + h \cdot fd1(y1_i, y2_i, h) \\ y2_i + h \cdot fd2(y1_i, y2_i, h) \end{pmatrix}$$

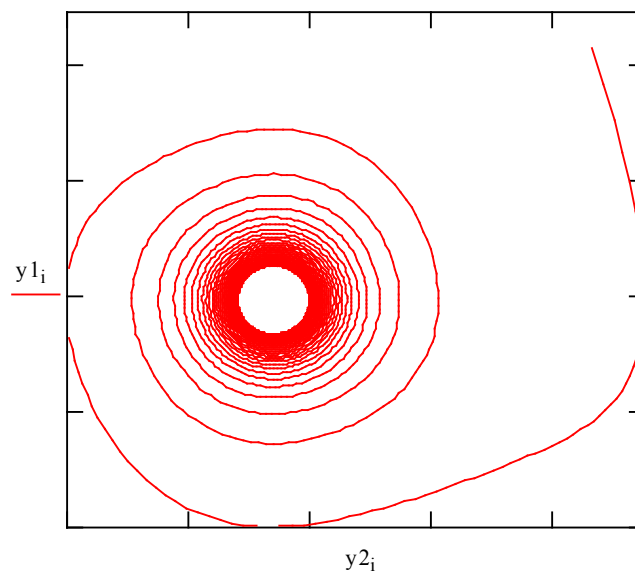


Рисунок 5.2 – Розв'язання системи Ван дер Поля (малі значення параметрів)

Розв'язання системи Ван дер Поля методом Рунге-Кутта четвертого порядку для великих значень параметрів подано на рисунку 5.3.

$$\lambda := 0.25 \quad a_0 := 1$$

Система диференціальних рівнянь:

$$fd01(y1, y2) := y1 \cdot (\lambda - y2^2) - a_0^2 \cdot y2$$

$$fd02(y1, y2) := y1$$

Розрахункові формули:

$$fd11(y1, y2, h) := fd01\left[\left(y1 + \frac{h}{2} \cdot fd01(y1, y2)\right), \left(y2 + \frac{h}{2} \cdot fd02(y1, y2)\right)\right]$$

$$fd12(y1, y2, h) := fd02\left[\left(y1 + \frac{h}{2} \cdot fd01(y1, y2)\right), \left(y2 + \frac{h}{2} \cdot fd02(y1, y2)\right)\right]$$

$$fd21(y1, y2, h) := fd11\left[\left(y1 + \frac{h}{2} \cdot fd11(y1, y2, h)\right), \left(y2 + \frac{h}{2} \cdot fd12(y1, y2, h)\right), h\right]$$

$$fd22(y1, y2, h) := fd12\left[\left(y1 + \frac{h}{2} \cdot fd11(y1, y2, h)\right), \left(y2 + \frac{h}{2} \cdot fd12(y1, y2, h)\right), h\right]$$

$$fd31(y1, y2, h) := fd11[(y1 + h \cdot fd21(y1, y2, h)), (y2 + h \cdot fd22(y1, y2, h)), h]$$

$$fd32(y1, y2, h) := fd12[(y1 + h \cdot fd21(y1, y2, h)), (y2 + h \cdot fd22(y1, y2, h)), h]$$

$$fd41(y1, y2, h) := \frac{fd01(y1, y2) + 2 \cdot fd11(y1, y2, h) + 2 \cdot fd21(y1, y2, h) + fd31(y1, y2, h)}{6}$$

$$fd42(y1, y2, h) := \frac{fd02(y1, y2) + 2 \cdot fd12(y1, y2, h) + 2 \cdot fd22(y1, y2, h) + fd32(y1, y2, h)}{6}$$

Початкові умови:

$$y1_0 := 0 \quad y2_0 := 1$$

$$h := 0.0025 \quad \text{крокок за часом}$$

$$imax := 25000 \quad \text{число кроків}$$

Рисунок 5.3 – Розв'язання системи Ван дер Поля при великих значеннях параметрів

Початкові умови:

$$\begin{aligned} y1_0 &:= 0 & y2_0 &:= 1 \\ h &:= 0.0025 & & \text{крокок за часом} \\ imax &:= 25000 & & \text{число кроків} \end{aligned}$$

Побудова розв'язку системи:

$$i := 0..imax$$
$$\begin{pmatrix} y1_{i+1} \\ y2_{i+1} \end{pmatrix} := \begin{pmatrix} y1_i + h \cdot fd41(y1_i, y2_i, h) \\ y2_i + h \cdot fd42(y1_i, y2_i, h) \end{pmatrix}$$

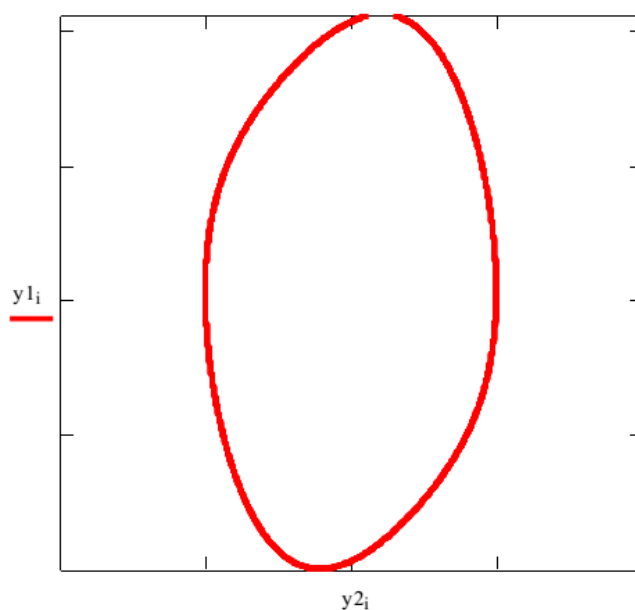


Рисунок 5.3, аркуш 2

5.2 Рівняння Лотки-Вольтерра

Рівняння Лотки-Вольтерра також має назву *система «хижак-жертва»*. Воно показує залежність між чисельністю «хижаків» та «жертв».

Вперше вона була одержана А. Лоткою в 1925 році для опису динаміки біологічних популяцій, що взаємодіють між собою. Трохи пізніше і незалежно від Лотки аналогічні (і більш складні) моделі були розроблені італійським математиком В. Вольтерром (1926 р.), глибокі дослідження якого в галузі екологічних проблем заклали фундамент математичної теорії біологічних співтовариств або так званої математичної екології.

Нехай є два біологічні види, які спільно мешкають в ізольованому середовищі. Середовище стаціонарне і забезпечує в необмеженій кількості

всім необхідним для життя один з видів, який називатимемо «жертвою». Інший вид – «хижак» також знаходиться в стаціонарних умовах, але харчується лише особинами першого вигляду.

Це можуть бути карасі і щуки, зайці і вовки, миші і лисиці, мікроби і антитіла і т.д. ... Будемо для визначеності називати їх карасями і щуками. Карасі і щуки живуть в деякому ізольованому ставку. Середовище надає карасям живлення в необмеженій кількості, а щуки харчуються лише карасями.

Позначимо y – кількість щук, x – кількість число карасів.

З часом кількість карасів і щук змінюється, але оскільки риби в ставку багато, тому вважатимемо x і y неперервними функціями часу t .

Пару чисел (x, y) має назву *стан моделі*. Необхідно визначити, як стан змінюється з часом.

Рівняння Лотки-Вольтерра має вигляд:

$$\begin{cases} x' = (a - by)x, \\ y' = (-c + dx)y, \end{cases}$$

де коефіцієнти a, b, c, d називаються *параметрами моделі*.

Зміна параметрів і розв'язання системи рівнянь моделі дає можливість досліджувати закономірності зміни стану екологічної системи.

Розв'яжемо рівняння Лотки–Вольтерра з такими параметрами системи: $a = 4$, $b = 2.5$, $c = 2$, $d = 1$ і початковою умовою: $x_1(0) = 3$, $x_2(0) = 1$.

Програму розв'язання системи подано на рисунку 5.4.

Розв'яжемо також вдосконалену модель Лотки–Вольтерра з «логістичною поправкою», яка визначає ступінь впливу «хижаків» і «жертв» один на одного. Програму розв'язання системи з логістичною поправкою подано на рисунку 5.5.

Як приклад, на рисунку 5.5 побудовані криві зміни чисельності карасів і щук y залежно від часу t . Максимуми кривих чергуються, причому максимуми щук відстають від максимумів карасів. Це відставання різне для різних екосистем типу «хижак–жертва», але, як правило, набагато менше за період коливань.

Не дивлячись на те, що розглянута модель є спрощеною і реально все відбувається набагато складніше, вона дозволила пояснити деякі процеси, що відбуваються у природі.

Одержала пояснення періодичність в протіканні хронічних захворювань, стало частково ясно, чому перебіг хвороби залежить від фази і інтенсивності лікування, що проводиться. Дійсно, як протікає хронічне захворювання? Загострення змінюється поліпшенням і знову все повторюється. Хвороба пов'язана з наявністю «хижака» (мікроб, вірус), який поїдає щось в організмі «жертви».

Загострення буває, коли «хижаків» багато (про це свідчать верхні ділянки кривих на рисунку), поліпшення самопочуття відповідає спадним ділянкам (тобто нижні ділянки на рисунку) і знову настає погіршення (зростаючі ділянки кривої). Загострення тим сильніше, чим більша амплітуда кривої.

Зараз подібні екологічні моделі будуються при лікуванні різних хронічних захворювань, зокрема, при боротьбі з хронічними інфекціями. Будується екологічна модель хвороби з урахуванням всіх імунних чинників і лікування проводиться відповідно до цієї моделі.

```

ORIGIN := 1
a := 4    b := 2.5  c := 2    d := 1
D(t, Y) := [ (a - b·Y2)·Y1 ]
            [ (-c + d·Y1)·Y2 ]
Y := [ 3 ]
      [ 1 ]
Y := rkfixed(Y, 0, 10, 500, D)

```

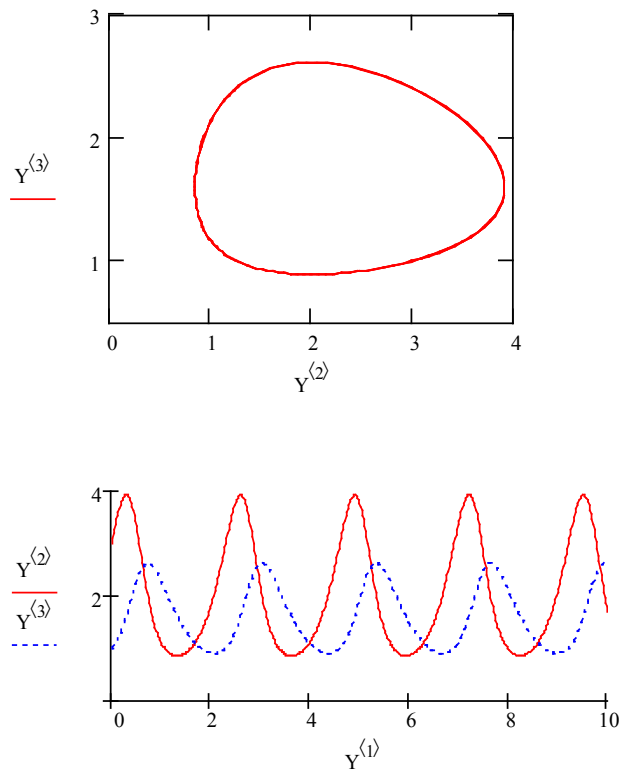


Рисунок 5.4 – Програма розв’язання системи Лотки–Вольтерра

ORIGIN := 1

a := 4 b := 2.5 c := 2 d := 1 α := 0.1

$$D(t, Y) := \begin{bmatrix} (a - b \cdot Y_2) \cdot Y_1 - \alpha \cdot (Y_1)^2 \\ (-c + d \cdot Y_1) \cdot Y_2 - \alpha \cdot (Y_2)^2 \end{bmatrix}$$

$$Y := \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

Y := rkfixed(Y, 0, 10, 500, D)

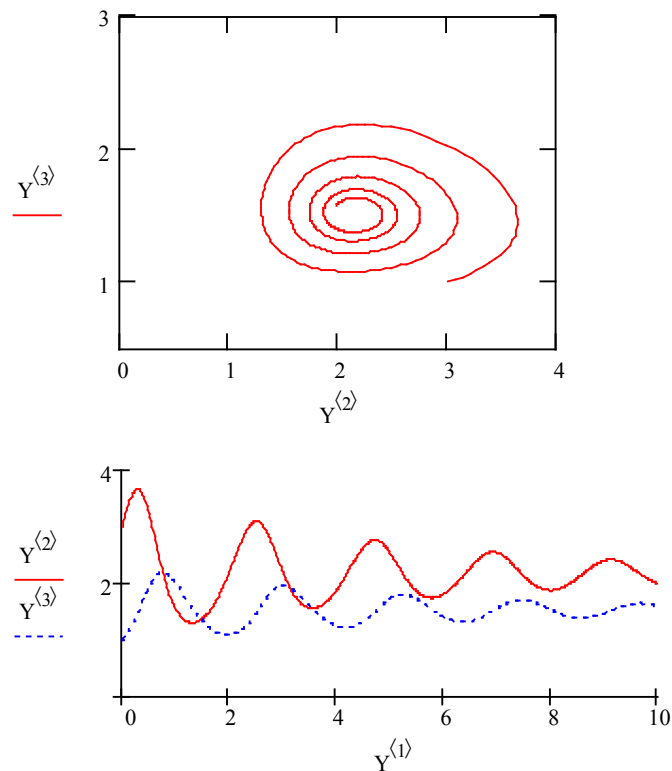


Рисунок 5.5 – Програма розв’язання системи Лотки–Вольтерра з логістичною поправкою

5.3 Періодичні коливання матеріальної точки

Рух матеріальної точки маси m під дією зовнішніх сил F описується другим законом Ньютона $ma = F$. Нехай точка рухається по осі Ox , тоді функція $x = x(t)$ – абсциса точки у момент часу t , задовольняє звичайне диференціальне рівняння другого порядку $mx'' = F(t, x, x')$.

Наприклад, рівняння:

$$x'' + w^2 x = 0$$

називається *рівняння гармонійного осцилятора* і описує періодичні коливання матеріальної точки з періодом:

$$T = \frac{2\pi}{\omega}$$

Задача про гармонічний осцилятор відіграє центральну роль як у класичній, так і у квантовій фізиці.

Велика кількість фізичних систем поведуть себе як гармонічні осцилятори при малому відхиленні від рівноваги. До них належать математичний і фізичний маятники, коливання атомів у молекулах і твердих тілах, електричні коливальні контури і багато інших.

Побудуємо декілька інтегральних кривих рівняння гармонічного осцилятора з частотою $\omega = 2$.

Програму розв'язання задачі подано на рисунку 5.6.

З наведених графіків добре видно, що всі одержані розв'язки періодичні з однаковим періодом.

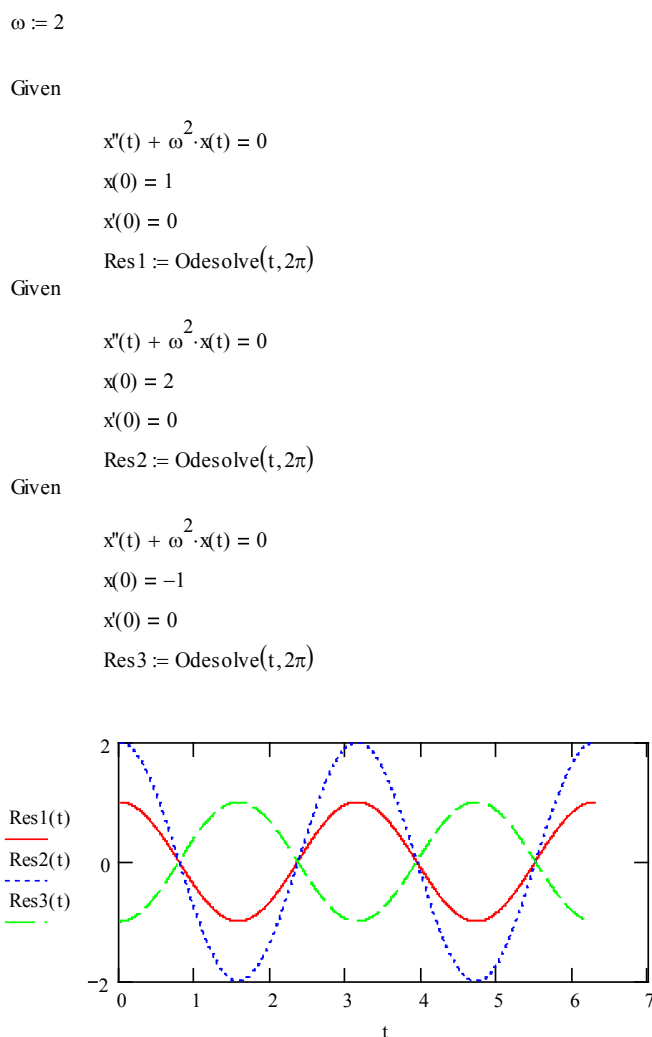


Рисунок 5.6 – Програма розв'язання рівняння гармонічного осцилятора

Легко перевірити, що $x = c1 \cos(\omega t) + c2 \sin(\omega t)$ є розв'язком рівняння гармонічного осцилятора при будь-яких значеннях констант $c1$ і $c2$ (рисунок 5.7).

$$x(t, \omega, C1, C2) := C1 \cdot \cos(\omega \cdot t) + C2 \cdot \sin(\omega \cdot t)$$

$$\frac{d^2}{dt^2} x(t, \omega, C1, C2) + \omega^2 \cdot x(t, \omega, C1, C2) \text{ simplify } \rightarrow 0$$

Рисунок 5.7 – Перевірка розв'язків рівняння гармонічного осцилятора

Отже, функція $x(t) = c1 \cos(\omega t) + c2 \sin(\omega t)$ при підстановці в рівняння перетворює його на тотожність. Цим доведено, що вона є розв'язком рівняння.

5.4 Зміна обсягу виробництва в економічній системі

Зміну обсягу виробництва в деякій замкнутій економічній системі описує диференціальне рівняння другого порядку:

$$y'' + 2|k|y' + \omega^2 y = 0.$$

Це досить поширена економічна модель.

У замкнутій економічній системі немає експорту, імпорту і припливу капіталу ззовні. Рівняння описує поведінку різниці:

$$y(x) = Y(x) - G / s$$

між обсягом виробництва $Y(x)$ і фіксованою величиною G/s відношення урядових витрат до граничної схильності населення до заощадження.

Розв'яжемо рівняння зміни обсягу виробництва в замкнутій економічній системі, параметри якої $k = 0.25$ і $\omega = 0.5$ (рисунок 5.8). З поданого графіка добре видно періоди економічного спаду і підйому.

k := 0.25 ω := 0.5

Given

$$y''(t) + 2|k| \cdot y'(t) + \omega^2 \cdot y(t) = 0$$

$$y(0) = 1$$

$$y'(0) = 0.1$$

Res := Odesolve(t, 20)

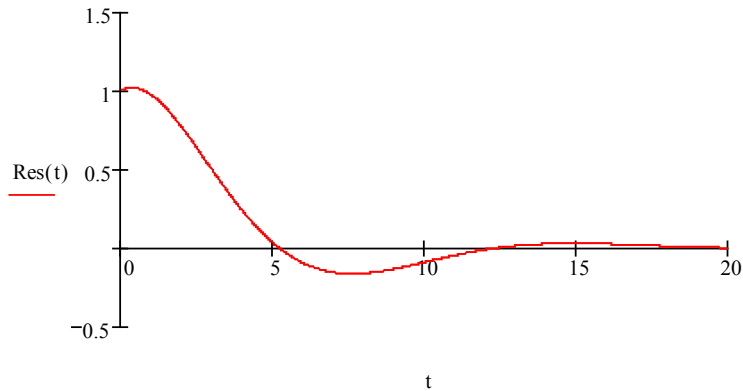


Рисунок 5.8 – Розв’язання рівняння зміни обсягу виробництва

5.5 Затухаючі коливання

Розв’яжемо рівняння другого порядку

$$x'' + 2ax' + bx = 0, \quad a^2 < b$$

яке описує *затухаючі (ангармонічні) коливання*.

Програму розв’язання задачі для параметрів $a = 0.1$ і $b = 1$ подано на рисунку 5.9.

a := 0.2 b := 0.5

Given

$$x''(t) + 2 \cdot a \cdot x'(t) + b \cdot x(t) = 0$$

$$x(0) = 1 \quad x'(0) = 0.5$$

Res := Odesolve(t, 30)

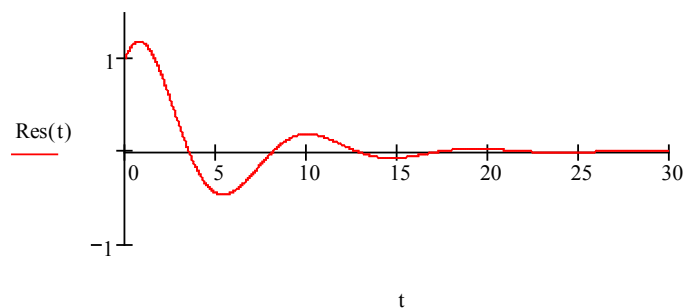


Рисунок 5.9 – Розв’язання рівняння затухаючих коливань

Питання для самоконтролю

1. Що описує система рівнянь Ван дер Поля?
2. Яка модель характеризує залежність чисельності конкуруючих видів? Які модифікації моделі ви знаєте?
3. Назвіть приклади економічних задач, які розв'язуються за допомогою ДР?
4. Наведіть власні приклади задач, які розв'язуються за допомогою ДР.

ГЛОСАРІЙ

Чисельні методи	Numerical solution
Диференціальне рівняння	Differential equation
Звичайне диференціальне рівняння	Ordinary differential equation
Диференціальне рівняння в частинних похідних	Partial differential equation
Задача Коші	Cauchy problem
Метод Ейлера	Euler method
Метод Рунге-Кутта	Runge-Kutta method
Початкові умови	Initial conditions
Правило Рунге	Runge rule
Крайова задача	Boundary value problem
Граничні умови	Boundary conditions
Однокрокові методи	One-step methods
Багатокрокові методи	Multistep methods
Метод прогнозу та корекції	Forecasting and correction method
Похибка	Measure of inaccuracy
Наближений розв'язок	Approaching solution
Метод кінцевих різниць	The method of finite differences
Кінцева точка	Finishing point
Диференціальне параболічне рівняння	Parabolic partial differential equation
Диференціальне рівняння гіперболічного типу	Hyperbolic partial differential equation
Рівняння Пуассона	Poisson's equation
Система	System
Векторне поле	Field of vectors
Автономна система	single-remote system
Стійкий вузол	Stable node, stable nodal point
Нестійкий вузол	Unstable node, unstable nodal point
Стійкий фокус	Stable focus
Нестійкий фокус	Unstable focus
Діактричний вузол	The dielectric node
Системи рівнянь Ван дер Поля	Van der Pol equation systems
Рівняння Лотки–Вольтерра	Lotka-Volterra equations,
Періодичні коливання	Periodical fluctuations
Матеріальна точка	Material point
Затухаючі коливання	Underdamping oscillations
Функція	Function

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 1 : навчальний посібник / [Р. Н. Кветний, І. В. Богач, О. Р. Бойко та інші]; за заг. ред. Р. Н. Кветного. – Вінниця : ВНТУ, 2013. – 191 с.
2. Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 2 : навчальний посібник / [Р. Н. Кветний, І. В. Богач, О. Р. Бойко та інші]; за заг. ред. Р. Н. Кветного. – Вінниця : ВНТУ, 2013. – 235 с.
3. Демидович Б. П. Численные методы анализа. Приближение функций, дифференциальные и интегральные уравнения / Демидович Б. П., Марон И. А., Шувалова Э. З. – СПб. : Лань, 2010. – 400 с.
4. Kvyetnyu R. Basics of Modelling and Computational Methods / Kvyetnyu R. – Вінниця : ВДТУ, 2007. – 147 с.
5. Колесницький О. К. Чисельні методи : навчальний посібник / Колесницький О. К, Арсенюк І. Р., Месюра В. І. – Вінниця : ВНТУ, 2017. – 130 с.
6. Бахвалов Н. С. Численные методы. Решения задач и упражнения: учебное пособие / Бахвалов Н. С., Корнев А. А., Чижонков Е. В. – М. : Бинном, 2016. – 352 с.
7. Савенкова Н. П. Численные методы в математическом моделировании: Учебное пособие / Савенкова Н. П., Проворова О. Г., Мокин А. Ю. – М. : Инфра-М, 2018. – 256 с.
8. Зализняк В. Е. Численные методы. Основы научных вычислений: учебное пособие для бакалавров / Зализняк В. Е. – М. : Юрайт, 2012. – 356 с.
9. Турчак Л. И. Основы численных методов : учебное пособие / Л. И. Турчак, П. В. Плотников. – М. : Физматлит, 2005. – 304 с.
10. Киреев В. И. Численные методы в примерах и задачах : учебн. пособие / В. И. Киреев, А. В. Пантелеев. – М. : Высшая школа, 2006. – 480 с.
11. Шахов Ю. Н. Численные методы : учебное пособие / Ю. Н. Шахов, Е. И. Деза. – М. : Книжный дом «Либроком», 2012. – 248 с.
12. Копченова Н. В. Вычислительная математика в примерах и задачах : учебное пособие / Н. В. Копченова, И. А. Марон. – М. : Лань, 2008. – 368 с.
13. Учебное пособие: Пособие MathCAD [Електронний ресурс]. – Режим доступу : http://prodcp.ru/referaty_po_matematike/uchebnoe_posobie_posobie_mathcad.html/. – Назва з екрана.
14. Тарасевич Ю. Ю. Численные методы на Mathcad'e [Електронний ресурс]. – Режим доступу: <http://www.exponenta.ru/educat/systemat/tarasevich/> - Назва з екрана.

Навчальне видання

**Богач Ілона Віталіївна
Краковецький Олександр Юрійович
Крилик Людмила Вікторівна**

**ЧИСЕЛЬНІ МЕТОДИ РОЗВ'ЯЗАННЯ
ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ
ЗАСОБАМИ MATHECAD**

Навчальний посібник

Рукопис оформлено *О. Краковецьким, Л. Крилик*

Редактор *Т. Старічек*

Оригінал-макет виготовив *О. Ткачук*

Підписано до друку 15.06.2020 р.
Формат 29,7×42¼. Папір офсетний.
Гарнітура Times New Roman.
Друк різнографічний. Ум. друк. арк 6,36.
Наклад 50 (1–21) пр. Зам. № 2020-067.

Видавець та виготовлювач
інформаційний редакційно-видавничий центр.
ВНТУ, ГНК, к. 114.
Хмельницьке шосе, 95,
м. Вінниця, 21021.
Тел. (0432) 65-18-06.
press.vntu.edu.ua;
E-mail: kivc.vntu@gmail.com.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.