МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Ю. Г. Лега
В. В. Мельник
О. М. Папуша

ПРИКЛАДНІ МЕТОДИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ В СЕРЕДОВИЩІ MATHEMATICA

ЧЕРКАСИ 2011

УДК 004.942 (075/8)
ББК 32.973я7
Л-38

Рекомендовано до друку Вченою радою Черкаського державного технологічного університету, протокол № 6 від 15.03.2010 р.

Р е ц е н з е н т и :

Вовк В.М., *д.е.н., професор, зав. кафедри економічної кібернетики Львівського національного університету ім. Івана Франка,*

Соловйов В.М., *д.ф.-м.н., професор, зав. кафедри економічної кібернетики Черкаського національного університету ім. Богдана Хмельницького,*

Хомяков В.І., *д.т.н., професор, декан факультету економіки та управління, зав. кафедри економіки та управління Черкаського державного технологічного університету*

Лега, Ю. Г. Прикладні методи комп'ютерного моделювання в середовищі Mathematica [Електронний ресурс] / Ю. Г. Лега, В. В. Мельник, О. М. Папуша ; М-во освіти і науки України, Черкас. держ. технол. ун-т. – Черкаси : ЧДТУ, 2011. – 188 с.
Л-38
ISBN 978-966-402-087-6

У посібнику викладено основи теорії комп’ютерного моделювання в середовищі Mathematica, що передбачає використання методів символьних обчислень і комп'ютерної алгебри для ряду комп'ютерних програм і модулів, які потім будуть використані в навчальному процесі й наукових дослідженнях.

Для студентів економічних спеціальностей, аспірантів, здобувачів, викладачів ВНЗ, підприємців і керівників.

УДК 004.942 (075/8)
ББК 32.973я7

Н а в ч а л ь н е в и д а н н я

ЛЕГА Юрій Григорович,
МЕЛЬНИК Веніамін Васильович,
ПАПУША Олександр Миколайович

ПРИКЛАДНІ МЕТОДИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ
В СЕРЕДОВИЩІ MATHEMATICA

В авторській редакції

Комп’ютерна верстка *Костенко Т.В.*
Коректор *Пепчук С.М.*
Дизайн обкладинки *Трохименко Н.К.*

ISBN 978-966-402-087-6

Формат 60х84 1/8. Папір офс. Гарн. Times New Roman. Друк оперативний.
Ум. друк. арк. 21,86. Обл.-вид. арк. 16,27. Зам. № 11-е225.

Черкаський державний технологічний університет

Свідоцтво про державну реєстрацію ДК № 896 від 16.04.2002 р.

Надруковано в редакційно-видавничому центрі ЧДТУ
бульвар Шевченка, 460, м. Черкаси, 18006.

ЗМІСТ

ВСТУП

Посібник підготовлено в Черкаському державному технологічному університеті, де нині активно просуваються сучасні комп'ютерні технології навчання студентів, аспірантів і фахівців інженерних, технологічних і фінансово-економічних напрямів для їхнього подальшого практичного використання. Крім оволодіння студентами традиційних методів розрахунків майбутнім фахівцям необхідне також освоєння сучасного комп'ютерного середовища Mathematica.

Цей посібник присвячений універсальним і масовим системам комп'ютерної математики Mathematica – світовий лідер серед програм символьної математики для ПК. Вони створені фірмою Wolfram Research, Inc на чолі з її президентом і головним розробником програм Стівеном Вольфрамом (Stephen Wolfram)

Як відомо, розроблювачі системи Mathematica проводять багато навчальних семінарів по її застосуванню в різних галузях науки й технологій, а також по навчанню студентів сучасним методам дослідження. Але набагато важливішим, з погляду розроблювачів, є залучення як користувачів-початківців, так і фахівців різних напрямків до тих нових можливостей символьних обчислень і символьного моделювання, які надає комп'ютерне середовище Mathematica. Нова і корисна інформація про застосування методів символьної алгебри в навчальних програмах розроблювачів системи знаходиться на сайті www.wolfram.com, а також на сайтах організаторів International Mathematica Simposium, де знаходяться як легко розповсюджувані продукти системи Mathematica, так і нові програмні засоби, виконані в C++ й Mathematica.

Здавалося б, є достатня інформація про застосування комп’ютерного моделювання в середовищі Mathematica. Проте, досвід застосування Mathematica в університетському класі показує, що після вступного курсу в комп'ютерну алгебру середовища Mathematica студенти починають вільно оперувати з досить складними комп'ютерними проектами, виконання яких без символьних обчислень було б для них досить важким.

Пропонований посібник охоплює значний комплекс навчально-методичного забезпечення, яке необхідне для вивчення курсу «Прикладні методи комп’ютерного моделювання в середовищі Mathematica». Вивчення дисципліни передбачає використання методів символьних обчислень і комп'ютерної алгебри для ряду комп'ютерних програм і модулів, які потім будуть використані в

навчальному процесі й наукових дослідженнях, наприклад, при проектуванні сучасних інженерних систем у техніці й технологіях.

Курс по символьних обчисленнях і чисельних методах впроваджується в навчальний процес і читається як спецкурс студентам кількох факультетів ЧДТУ, зокрема на кафедрах прикладної математики й економічної кібернетики.

Метою посібника є теоретичне навчання сучасним комп'ютерним технологіям в області символьних і чисельних методів, а також забезпечення студентів фактичним матеріалом для практичної підготовки. Застосовувані у різних інженерних й економіко-математичних курсах, включаючи курс прикладної математики, ці методи є основою освіти по математичному циклу в підготовці фахівців інженерних і природничо-наукових спеціальностей.

Автори вважають, що читач віддає перевагу інтерактивним комп'ютерним середовищам у порівнянні зі звичайними текстами електронних і традиційних книг. Тому тексти даного курсу одночасно можна використати як електронні версії широко використовуваних математичних текстів і як "шаблони" розрахункових програм у середовищі Mathematica. На думку авторів, такий виклад є найбільш придатним для навчання, оскільки запобігає непродуктивному опрацюванню надлишкового матеріалу в середовищі програмування і вивченню суто математичних (включаючи й чисельні) методів, теорія яких викладена в спеціальних підручниках і монографіях.

Основною відмінністю пропонованого викладу матеріалу є подання тем і методики практичних занять в інтерактивному середовищі Mathematica, опис якого українською мовою обмежений або в деяких аспектах просто відсутній. Найбільш важливі теоретичні положення підкріплені типовими прикладами.

Викладений матеріал можна використати як посібник для вивчення теоретичних засад, а також для виконання практичних завдань і самопідготовки студентів. Для роботи необхідний сучасний комп'ютер із завантаженими у нього Windows і Mathematica.

Досвід роботи зі студентами показав, що з використанням методів символьної алгебри в середовищі Mathematica вони успішно виконують усі домашні й розрахунково-графічні завдання.

Посібник розрахований для студентів економічних і технічних спеціальностей, а також може бути корисним для працівників фінансових установ.

Тема 1

РОЗВ'ЯЗАННЯ СИСТЕМ ЛІНІЙНИХ РІВНЯНЬ

1.1. Вектори

1.1.1. Визначення і операції над векторами

Вектор є одновимірним масивом і зображується у вигляді

```
ln[1] := a = {1,2,4}
b = {p, q, s}
Out[1] = {1,2,4}
Out[2] = {p, q, s}
```

або у вигляді таблиці

```
ln[3] := TableForm[a]
TableForm[b]
Out[3]//TableForm =
1
2
4
Out[4]//TableForm =
p
q
s
```

Операції над векторами в комп'ютерній математиці дещо відрізняються від звичних дій над ними при "ручних" обчисленнях:

• лінійна комбінація

```
ln[5] := 5a + 4b
Out[5] = {4p + 5,4q + 10,4s + 20}
```

• піднесення до степеня

```
a^2 + b^2
{1 + p^2,4 + q^2,16 + s^2}
```

• складання скалярного вектора

```
a + 1
{2,3,5}
```

Дві останні операції прийняті в символьній алгебрі системи *Mathematica*.

Скалярний і векторний добуток

$\ln[6] := a.b$

$Out[6] = p + 2q + 4s$

або в еквівалентній формі

$\ln[7] := Inner[Times, a, b]$

$Out[7] = p + 2q + 4s$

Якщо бажаємо знайти добуток векторів так, щоб у результаті одержати матрицю, то використаємо операцію зовнішнього добутку

$\ln[8] := Outer[Times, a, b]$

$$Out[8] = \begin{pmatrix} p & q & s \\ 2p & 2q & 2s \\ 4p & 4q & 4s \end{pmatrix}$$

$\ln[9] = TableForm[\%]$

$out[9] // TableForm =$

$$\begin{array}{ccc} p & q & s \\ 2p & 2q & 2s \\ 4p & 4q & 4s \end{array}$$

Крім скалярного добутку, вводиться й векторний добуток. Звичайна операція векторного добутку представлена нижче:

$\ln[10] := Cross[a, b]$

$Out[10] = \{2s - 4q, 4p - s, q - 2p\}$

Пакет *«LinearAlgebra CrossProduct* дозволяє по іншому обчислювати векторний добуток векторів, причому останні повинні бути тривимірними. Нижче наведений приклад ілюструє застосування пакета:

$\ln[11] := f = \{x, y, z\}$

$F = \{fx, fy. fz\}$

$Out[11] = \{x, y, z\}$

$Out[12] = \{fx, fy, fz\}$

$\ln[13] := Cross[f, F]$

$Out[13] = \{fzy - fyz, fxz - fzx, fyx - fxy\}.$

1.1.2. Ортогоналізація

Якщо система векторів лінійно незалежна, то її можна ортогоналізувати, тобто представити у вигляді системи взаємно перпендикулярних векторів. У математиці розроблена процедура Грамма-Шмідта що дозволяє зробити це:

ln[14] := << *LinearAlgebra'Orthogonalization'*

ln[15] := $v1 = \{4,6,1\}; v2 = \{1,0,-1\};$

ln[16] := *GramSchmidt*[{*v*1, *v*2}]

$$Out[16] = \begin{pmatrix} \frac{4}{\sqrt{53}} & \frac{6}{\sqrt{53}} & \frac{1}{\sqrt{53}} \\ \frac{41}{\sqrt{5141}} & -\frac{18}{\sqrt{5141}} & -\frac{56}{\sqrt{5141}} \end{pmatrix}$$

Проекція вектора *v1* на вектор *v2*

ln[17] := *Projection*[*v*1, *v*2]

$$Out[17] = \{\frac{3}{2}, 0, -\frac{3}{2}\}.$$

1.1.3. Ортогоналізація багаточленів

Операцію ортогоналізації можна застосувати для лінійно незалежних багаточленів у такому вигляді:

ln[18] := *poly* = {1, *x*, *x*^2, *x*^3, *x*^4}

$Out[18] = \{1, x, x^2, x^3, x^4\}$

ln[19] := *GramSchmidt*[*poly*, *Normalized* → *False*,

InnerProduct → (*Integrate*[#1#2 / *Sqrt*[1 − *x*^2], {*x*, −1, 1}]&)] // *Simplify*

$$Out[19] = \{1, x, x^2 - \frac{1}{2}, x^3 - \frac{3x}{4}, x^4 - x^2 + \frac{1}{8}\}$$

1.2. МАТРИЦІ

1.2.1. Матриця у вигляді таблиці

Матриця зображується як двовимірний масив або як система векторів. Задамо матрицю як систему векторів. Матриця має такий вигляд:

ln[21] := *Quit*[];

ln[1] := $A = \{\{a_{11}, a_{12}, a_{13}\},$

$\{a_{21}, a_{22}, a_{23}\},$

$\{a_{31}, a_{32}, a_{33}\}\}$

$$Out[1] = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

або матриця в табличному вигляді

$\ln[2] := TableForm[A]$

$Out[2]//TableForm =$

$a_{11}\quad a_{12}\quad a_{13}$

$a_{21}\quad a_{22}\quad a_{23}$

$a_{31}\quad a_{32}\quad a_{33}$

Елемент матриці й рядок матриці задаються у вигляді

$\ln[3] := A[[2,3]]$

$Out[3] = a_{23}$

$\ln[4] := A[[3]]$

$Out[4] = \{a_{31}, a_{32}, a_{33}\}$

Транспонована матриця записується таким способом у табличному вигляді:

$\ln[5] := Transpose[A]//TableForm$

$Out[5]//TableForm =$

$a_{11}\quad a_{21}\quad a_{31}$

$a_{12}\quad a_{22}\quad a_{32}$

$a_{13}\quad a_{23}\quad a_{33}$

і той же самий результат у матричному вигляді:

$\ln[6] := Transpose[A]//MatrixForm$

$Out[6]//MatrixForm =$

$a_{11}\quad a_{21}\quad a_{31}$

$a_{12}\quad a_{22}\quad a_{32}$

$a_{13}\quad a_{23}\quad a_{33}$

1.2.2. Генерація матриць

Зазначимо деякі шляхи генерації матриць

$\ln[7] = matriceRandom = Table[Random[], \{3\}, \{3\}]$

$$Out[7] = \begin{pmatrix} 0.634904 & 0.492875 & 0.604098 \\ 0.351572 & 0.518241 & 0.936993 \\ 0.217584 & 0.0734664 & 0.453426 \end{pmatrix}$$

$\ln[8] := TableForm[\%]$

$Out[8]//TableForm =$

0.634904 0.492875 0.604098

0.351572 0.518241 0.936993

0.217584 0.0734664 0.453426

Одинична матриця має такий вигляд:

$\ln[9] := IdentiMatrix[4]$

$$Out[9] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Діагональна матриця:

$\ln[10] := DiagonalMatrix[\{2,3,4,5\}]$

$$Out[10] = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 5 \end{pmatrix}$$

Як приклад генерації матриці оператором *Table[...]* розглянемо генерацію матриці Гілберта:

$\ln[11] := Table[1/(i + j + 1), \{i,3\}, \{j,3\}]$

$$Out[11] = \begin{matrix} \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{matrix}$$

1.2.3. Операції над матрицями

Основні операції над матрицями дуже прості й звичайно відомі з курсу вищої математики. Проілюструємо це на прикладах матриць розмірності 2×2:

$\ln[12] := \{\{1,2\},(7,8)\}; G = \{\{4,2\},\{1,1\}\};$

$\ln[13] := 3H$

$$Out[13] = \begin{pmatrix} 3 & 6 \\ 21 & 24 \end{pmatrix}$$

$\ln[14] := G + 1$

$$Out[14] := \begin{pmatrix} 5 & 3 \\ 2 & 2 \end{pmatrix}$$

$\ln[15] := H + G$

$Out[15] = \begin{pmatrix} 5 & 4 \\ 8 & 9 \end{pmatrix}$

$\ln[16] := TableForm[\%]$

$Out[16] // TableForm =$

5 4

8 9

Наведемо найбільш уживані команди над матрицями:

*Transpose[A] (*операція транспонування*);*

*Det[A] (*визначник матриці*);*

*Minor[A,k] (*матриця мінору k*k*);*

*Sum[A[[i,i]],{i,n}] (*слід матриці*);*

*A,B (*похідні матриці*);*

*MatrixPower[A,n] (*n-ий степінь матриці*);*

*MatrixExp[A] *матрична експонента E ^A*);*

*Inverse[A] (*обернена матриця*);*

*PseudoInverse[A] (*псевдо обернена трикутна матриця*);*

*NullSpace[A] (*множин векторів задовольняючі рівність A.x?0*);*

*RowReduce[A] (*спрощена форма матриці A*].*

Розглянемо деякі приклади:

ln[17] := *Quit*[];

ln[1] := $A = Table[1/(i + j + 1), (i,3), (j,3)]$

$$Out[1] = \begin{pmatrix} \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{pmatrix}$$

ln[2] := *Transpose*[*A*]

$$Out[2] = \begin{pmatrix} \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{pmatrix}$$

ln[3] := *Det*[*A*]

$$Out[3] = \frac{1}{378000}$$

ln[4] := *Minors*[*A*,2]

$$Out[4] = \begin{pmatrix} \frac{1}{240} & \frac{1}{180} & \frac{1}{600} \\ \frac{1}{180} & \frac{4}{525} & \frac{1}{420} \\ \frac{1}{600} & \frac{1}{420} & \frac{1}{1260} \end{pmatrix}$$

ln[5] := *Inverse*[*A*]

$$Out[5] = \begin{pmatrix} 300 & -900 & 630 \\ -900 & 2880 & -2100 \\ 630 & -2100 & 1575 \end{pmatrix}$$

Перевіримо останню операцію множенням матриці А на обернену A^{-1}.

ln[6] := *A*.*Inverse*[*A*]

$$Out[6] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Як бачимо з останнього результату, відповідь вірна.

Операція матричної експоненти має такий символьний вигляд:

$\ln[7] := MatrixExp[A]$

$$Out[7] = \begin{cases} 20\,RootSum[378000\#1^3 - 255600\#1^2 + 4755\#1 - 1\&, \dfrac{1260e^{\#1}\#1^2 - 432e^{\#1}\#1 + e^{\#1}}{75600\#1^2 - 34080\#1 + 317}\&] \\ 60\,RootSum[378000\#1^3 - 255600\#1^2 + 4755\#1 - 1\&, \dfrac{105e^{\#1}\#1 - e^{\#1}}{75600\#1^2 - 34080\#1 + 317}\&] \\ 42\,RootSum[378000\#1^3 - 255600\#1^2 + 4755\#1^2 + 4755\#1 - 1\&, \dfrac{120e^{\#1}\#1 + e^{\#1}}{75600\#1^2 - 34080\#1 + 317}\&] \end{cases}$$

Далі введемо матрицю *A*

$\ln[8] := a = \{\{0,1,-1\},\{1,-2,1\},\{-2,2,0\}\}$

$$Out[8] = \begin{pmatrix} 0 & 1 & -1 \\ 1 & -2 & 1 \\ -2 & 2 & 0 \end{pmatrix}$$

і введемо для неї нуль-простір, тобто множину векторів, яка формує нуль-простір цієї матриці.

$\ln[9] := NullSpace[a]$

$Out[9] = (111)$.

1.3. РІШЕННЯ В *MATHEMATICA*

1.3.1. Формулювання подвійного підходу для рішення системи лінійних рівнянь

Система лінійних рівнянь розмірності n×m у матричному вигляді представлена нижче:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} * \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = (b_1 \quad b_2 \quad \dots \quad b_m) \qquad (1.1)$$

Рішення системи лінійних рівнянь, представленої в матричному вигляді, також зображується, як рішення, записане у вигляді вектора. Позначення прийняті такі ж, як у курсі вищої математики.

Оператор для знаходження вектора рішення має такий вигляд:

LinearSolve[*A*,*b*].

Приклад розв'язування й перевірки розв'язку для лінійної системи рівнянь 4×4, рішення якої "вручну" знайти не просто через значні обчислення, наведені нижче:

$\ln[10] := A = \{\{1,3,5,7\},\{3,5,7,1\},\{5,7,1,3\},\{7,1,3,5\}\};$

$b = \{12,0,4,16\};$

ln[12] := *TableForm*[*A*]

Out[12] // *TableForm* =

1	3	5	7
3	5	7	1
5	7	1	3
7	1	3	5

ln[13] := *TableForm*[*b*]

Out[13] // *TableForm* =

12
0
4
16

Рішення системи лінійних рівнянь

$A.\vec{x} = \vec{b}.$

має вигляд

ln[14] := *solutionX* = *LinearSolve*[*A*, *b*]

Out[14] = {1,−1,0,2}.

Потім перевіримо отримане рішення

ln[15] := *A.solutionX* − *b*

Out[15] = {0,0,0,0}.

Інший спосіб одержання рішення полягає в поданні невідомого вектора рішення $\vec{x}$ у такому вигляді:

$\vec{x} = A^{-1}.\vec{b}.$

Рішення, записане через A^{-1} обернену матрицю, має вигляд

ln[16] := *solutionX* = *Inverse*[*A*].*b*

Out[16] = {1,−1,0,2}.

Обидва розв’язки, отримані різними операторами, збігаються.

1.3.2. Рішення системи лінійних рівнянь у традиційному вигляді

Якщо система лінійних рівнянь задана в природному вигляді

$$a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n = b_1;$$
$$a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n = b_2;$$
$$\ldots$$
$$a_{m,1}x_1 + a_{m,2}x_2 + \ldots + a_{m,n}x_n = b_m,$$

то відповідне рішення знаходимо за допомогою операторів

Solve[*equations*]

(*система задається у вихідному вигляді*);

Solve[*equations*, var *iables*]

(*система задається у вихідному вигляді й вказується набір шуканих змінних*).

Розглянемо приклади застосування оператора *Solve:*

ln[17] := *equations* = {2x − 3y + 7z = 3, −x + 3y − 2z = 4, 3x + 5y + z = 1}

Out[17] = {2x − 3y + 7z = 3, −x + 3y − 2z = 4, 3x + 5y + z = −1}

ln[18] = *equations* // *TableForm*

Out[18] // *TableForm* =

$2x - 3y + 7z = 3$

$-x + 3y - 2z = 4$

$3x + 5y + z = -1$

ln[19] := $\{x \to -\frac{206}{57}, y \to \frac{88}{57}, z \to \frac{121}{57}\}$

Перевіримо отриманий розв'язок шляхом підстановки розв'язку у вихідну систему:

ln[20] := *chekEquation* = *equations* /. *solutionEquation*

Out[20] = {*True*, *True*, *True*}.

Якщо заздалегідь не зазначені змінні, то необхідно використати іншу форму застосування оператора *Solve*, не забуваючи вказувати змінні підпорядковані визначенню.

Приклади застосування символьних обчислень із використанням оператора *Solve[]* наведені нижче:

ln[22] := *Quit*[];

ln[2] := *Solve*[{$2x_1 - y_1 == a - b, x_1 + 7y_1 == a + 3b$}, {$x_1, y_1$}] // *Flatten*

Out[2] = $\{x_1 \to \frac{4}{15}(2a - b), y_1 \to \frac{1}{15}(a + 7b)\}$

ln[3] := *Solve*[{$2x_1 - y_1 == a - b, x_1 + 7y_1 == a + 3b$}, {$a, b$}]

Out[3] = $\{\{a \to \frac{1}{4}(7x_1 + 4y_1), b \to \frac{1}{4}(8y_1 - x_1)\}\}$

ln[4] := *Solve*[{$2x_1 - y_1 == a - b, x_1 + 7y_1 == a + 3b$}, {$a, x_1$}]

Out[4] = $\{\{a \to 15y_1 - 7b, x_1 \to -4(b - 2y_1)\}\}$

ln[5] := *Solve*[{$2x_1 - y_1 == a - b, x_1 + 7y_1 == a + 3b$}, {$a, y_1$}]

Out[5] = $\{\{a \to \frac{1}{8}(4b + 15x_1), y_1 \to \frac{1}{8}(4b + x_1)\}\}$.

Далі наведемо приклади, коли система не має рішення або рішень існує нескінченна множина:

$\ln[6] := Solve[\{3x + y = 9, 6x + 2y = 4\}, \{x, y\}]$

$Out[6] = \{\}$

$\ln[7] := Solve[\{3x + y = 9, 6x + 2y = 18\}, \{x, y\}]$

Solve:: svars: Equations may give solutions for all "solve" variables More…

$Out[7] = \{\{x \to 3 - \frac{y}{3}\}\}.$

1.3.3. Перевизначення системи лінійних рівнянь

Якщо кількість рівнянь більше як невідомих, то такі системи називаються перевизначеними, а їхні розв'язки необхідні при розв'язанні ряду економічних задач [4]. Для їхнього розв'язку використовується оператор

PseudoInverse[*matrix*].

Нижче наведений приклад рішення перевизначеної системи:

$\ln[8] := matrix = \{\{3,1\}, \{2,5\}, \{8,1\}\}; b = \{1, -1, -2\};$

$\ln[9] := solutionApprox = PseudoInverse[matrix].b // N$

$Out[9] = \{-0.17033, -0.0897436\}.$

1.3.4. Недовизначені системи лінійних рівнянь

Якщо кількість невідомих у системі більша за кількість рівнянь у тій же системі, то така система називається недовизначеною.

Для розв'язку недовизначених систем також використовують зазначені вище оператори *Solve і PseudoInverse.*

Приведемо приклади застосування цих операторів для розв'язку.

Розпочнемо із системи, що має декілька розв'язків, застосування оператора *LinearSolve* дає одне з можливих розв'язків.

$\ln[10] := A = \{\{4,3,1\}, \{3,2,5\}\};$

$b = \{9,4\};$

$\ln[12] := solutionOne = LinearSolve[A, b]$

$Out[12] = (-6, 11, 0).$

Далі розглянемо інші розв'язки:

$\ln[13 := Solve[\{4x + 3y + z = 9, 3x + 2y + 5z = 4\}, \{x, y, z\}].$

Solve:: svars: Equations may give solutions for all "solve" variables. More…

$Out[13] = \{\{x \to -13z - 6, y \to 17z + 11\}\}$

Застосовуючи оператор *PseudoInverse*, знову знаходимо одне з розв'язків

$\ln[14] := N[PseudoInverse[A].b]$

$Out[14] = \{1.50545, 1.18519, -0.577342\}.$

Тема 2

РОЗВ'ЯЗАННЯ ПОЛІНОМІАЛЬНИХ І ТРАНСЦЕНДЕНТНИХ РІВНЯНЬ

2.1. Рівняння з однією змінною

2.1.1. Введення і оператор розв'язку

Для розв'язання рівнянь із однією змінною використовуються оператори

Solve[*equation*];

Solve[*equation*, *x*].

2.1.2. Квадратне рівняння

Розглянемо розв'язання квадратного рівняння:

ln[15] := *Quit*[];

ln[1] := *equation* = *ax*^2 + *bx* + *c* = 0

$Out[1] = ax^2 + bx + c = 0$

ln[2] := *solutionSquar* = *Solve*[*equation*, *x*]

$$Out[2] = \{\{x \to \frac{-b-\sqrt{b^2-4ac}}{2a}\}, \{x \to \frac{\sqrt{b^2-4ac}-b}{2a}\}\}.$$

Перевірка розв'язку має вигляд:

ln[3] := *equation* /. *solutionSquar* // *Simplify*

$Out[3] = \{True, True\}$.

Далі приведемо символьний розв'язок кубічного рівняння, заданого у загальному вигляді:

ln[3] := *equation*3 = $x^3 + ax^2 + bx + c = 0$

$Out[3] = x^3 + ax^2 + bx + c = 0.$

Символьний розв'язок набуває вигляду:

ln[4] := *solutionCubic* = *Solve*[*equation*3, *x*] // *Simplify*

$$Out[4] = \{\{x \to \frac{1}{6}(-2a + 2^{2/3}(-2a^3 + 9ba - 27c + 3\sqrt{3}\sqrt{4ca^3 - b^2a^2 - 18bca + 4b^3 + 27c^2})\text{^}(1/3) +$$
$$(2\sqrt[3]{2}(a^2 - 3b))/$$
$$((-2a^3 + 9ba - 27c + 3\sqrt{3}\sqrt{4ca^3 - b^2a^2 - 18bca + 4b^3 + 27c^2})\text{^}(1/3)))\},$$
$$\{x \to \frac{1}{12}(-4a + i2^{2/3}(i + \sqrt{3})(-2a^3 + 9ba - 27c + 3\sqrt{3}\sqrt{4ca^3 - b^2a^2 - 18bca + 4b^3 + 27c^2})\text{^}$$
$$(1/3) - (2i\sqrt[3]{2}(-i + \sqrt{3})(a^2 - 3b))/$$
$$((-2a^3 + 9ba - 27c + 3\sqrt{3}\sqrt{4ca^3 - b^2a^2 - 18bca + 4b^3 + 27c^2})\text{^}(1/9)))\}\}.$$

Із загального символьного розв'язку кубічного рівняння одержимо числові розв'язки, шляхом задання коефіцієнтів a, b, c.

$$\ln[5] := solutionCubic/.\{a \to 2, b \to 1, c \to 3\}$$

$$Out[5] = \{\{x \to \frac{1}{6}\left(-4 - 2(-1)^{2/3}\sqrt[3]{\frac{2}{79 - 9\sqrt{77}}} + 2^{2/3}\sqrt[3]{-79 + 9\sqrt{77}}\right)\},$$
$$\{x \to \frac{1}{12}\left(-8 - \frac{2\sqrt[6]{-1}\sqrt[3]{2}(-i + \sqrt{3})}{\sqrt[3]{79 - 9\sqrt{77}}} + i2^{2/3}(i + \sqrt{3})\sqrt[3]{-79 + 9\sqrt{77}}\right)\},$$
$$\{x \to \frac{1}{12}\left(-8 + \frac{2\sqrt[6]{-1}\sqrt[3]{2}(i + \sqrt{3})}{\sqrt[3]{79 - 9\sqrt{77}}} - 2^{2/3}(1 + i\sqrt{3})\sqrt[3]{-79 + 9\sqrt{77}}\right)\}\}.$$

Очевидно, що формат виведення рівняння зберігся. Числові розв'язки кубічного рівняння із заданою точністю наведені нижче:

$$\ln[6] := N[solutionCubic/.\{a \to 2\}, b \to 1, c \to 3]\}$$
$$Out[6] = \{\{x \to 0.0872797 - 1.17131i\}, \{x \to -2.17456 + 1.68036 \times 10^{-16} i\}, \{x \to 0.0872797 + 1.17131i\}\}$$
$$\ln[8] := N[solutionCubic/.\{a \to 2, b \to 1, c \to 3\}, 25]$$
$$Out[8] = \{\{x \to 0.0872797051464900371011595 - 1.1713121110008787349637439i\},$$
$$\{x \to -2.1745594102929800742023 19 + 0. \times 10^{-25} i\},$$
$$\{x \to 0.0872797051464900371011595 + 1.1713121110008787349637439i\}\}$$
$$\ln[9] := N[solutionCubic/.\{a \to 2, b \to 1, c \to 3\}]Chop$$
$$Out[9] = \{\{x \to 0.0872797 - 1.17131i\}, \{x \to -2.17456\}, \{x \to 0.0872797 + 1.17131i\}\}$$
$$\ln[10] := N[solutionCubic/.\{a \to 2, b \to 1, c \to 3\}]Chop$$
$$Out[10] = \{\{x \to 0.0872797051464900371011595 - 1.1713121110008787349637439i\},$$
$$\{x \to -2.1745594102929800742023 19\},$$
$$\{x \to 0.0872797051464900371011595 + 1.1713121110008787349637439i\}\}.$$

Далі наведемо символьний розв'язок поліноміального рівняння 4-го степеня, який має такий вигляд:

$$\ln[11] := equationFourth = x^4 + bx^3 + c = 0$$
$$Out[11] = x^4 + bx^3 + c = 0.$$

Символьний розв’язок набуває вигляду:

$$\text{In}[12] := solution4 = Solve[equationFourth, x] // Simplify$$

$$Out[12] = \{\{x \to -\frac{b}{4} + \frac{1}{2}\sqrt{\frac{b^2}{4} + \frac{8\sqrt[3]{3}c + \sqrt[3]{2}\left(9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}\right)^{2/3}}{6^{2/3}\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}} -$$

$$\frac{1}{2}\sqrt{\left(\frac{b^3}{\sqrt[4]{\frac{b^2}{4} + \frac{8\sqrt[3]{3}c + \sqrt[3]{2}(9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2})^{2/3}}{6^{2/3}\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}}} + \frac{b^2}{2} - \right.}$$

$$\left.\frac{\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}{\sqrt[3]{2}3^{2/3}} - \frac{4\sqrt[3]{\frac{2}{3}}c}{\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}\right)}\}\}$$

$$\text{In}[12] := solution4 = Solve[equationFourth, x] // Simplify$$

$$Out[12] = \{\{x \to -\frac{b}{4} + \frac{1}{2}\sqrt{\frac{b^2}{4} + \frac{8\sqrt[3]{3}c + \sqrt[3]{2}\left(9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}\right)^{2/3}}{6^{2/3}\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}} -$$

$$\frac{1}{2}\sqrt{\left(\frac{b^3}{\sqrt[4]{\frac{b^2}{4} + \frac{8\sqrt[3]{3}c + \sqrt[3]{2}(9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2})^{2/3}}{6^{2/3}\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}}} + \frac{b^2}{2} - \right.}$$

$$\left.\frac{\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}{\sqrt[3]{2}3^{2/3}} - \frac{4\sqrt[3]{\frac{2}{3}}c}{\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}\right)}\},$$

$$\{x \to -\frac{b}{4} + \frac{1}{2}\sqrt{\frac{b^2}{4} + \frac{8\sqrt[3]{3}c + \sqrt[3]{2(9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2})^{2/3}}}{6^{2/3}\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}} +$$

$$\frac{1}{2}\sqrt{\left(\frac{b^3}{\sqrt[4]{\frac{b^2}{4} + \frac{8\sqrt[3]{3}c + \sqrt[3]{2}(9cb^2 + \sqrt{3})\sqrt{(27b^4 - 256c^2)^{2/3}}}{6^{2/3}\sqrt[3]{9cb^2 + \sqrt{3}}\sqrt{(27b^4 - 256c)c^2}}}} + \frac{b^2}{2} - \right.}$$

$$\left.\frac{\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}{\sqrt[3]{2}3^{2/3}} - \frac{4\sqrt[3]{\frac{2}{3}}c}{\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}\right)}\},$$

$$\{x \to -\frac{b}{4} - \frac{1}{2}\sqrt{\frac{b^2}{4} + \frac{8\sqrt[3]{3}c + \sqrt[3]{2}(9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2})^{2/3}}{6^{2/3}\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}} -$$

$$\frac{1}{2}\sqrt{\left(\frac{b^3}{4\sqrt{\frac{b^2}{4} + \frac{8\sqrt[3]{3}c + \sqrt[3]{2}(9cu^2 + \sqrt{3}\sqrt{27b^4 - 256c^2})^{2/3}}{6^{2/3}\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}}} + \frac{b^2}{2} - \frac{\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}{\sqrt[3]{2}3^{2/3}} - \frac{4\sqrt[3]{\frac{2}{3}}c}{\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}\right)}\},$$

$$\{x \to -\frac{b}{4} - \frac{1}{2}\sqrt{\frac{b^2}{4} + \frac{8\sqrt[3]{3}c + \sqrt[3]{2}(9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2})^{2/3}}{6^{2/3}\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}} +$$

$$\frac{1}{2}\sqrt{\left(\frac{b^3}{4\sqrt{\frac{b^2}{4} + \frac{8\sqrt[3]{3}c + \sqrt[3]{2}(9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2})^{2/3}}{6^{2/3}\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}}} + \frac{b^2}{2} - \frac{\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}{\sqrt[3]{2}3^{2/3}} - \frac{4\sqrt[3]{\frac{2}{3}}c}{\sqrt[3]{9cb^2 + \sqrt{3}\sqrt{(27b^4 - 256c)c^2}}}\right)}\}\}.$$

Як і раніше, із загального символьного розв'язку рівняння 4-го степеня нескладно одержати числові розв'язки шляхом задання числових значень коефіцієнтам b, c:

$\ln[13] := solution4 /. \{b \to 1, c \to 3\}$

$$Out[13] = \{\{x \to -\frac{1}{4} + \frac{1}{2}\sqrt{\frac{1}{4} + \frac{24\sqrt[3]{3} + \sqrt[3]{2}(27 + 9i\sqrt{247})^{2/3}}{6^{2/3}\sqrt{27 + 9i\sqrt{247}}}} -$$

$$\frac{1}{2}\sqrt{\frac{1}{2} - \frac{4 \cdot 3^{2/3}}{\sqrt[3]{\frac{1}{2}(27 + 9i\sqrt{247})}} - \frac{\sqrt[3]{\frac{1}{2}(27 + 9i\sqrt{247})}}{3^{2/3}} - \frac{1}{\sqrt[4]{\frac{1}{4} + \frac{24\sqrt[3]{3} + \sqrt[3]{2}(27 + 9i\sqrt{247})^{2/3}}{6^{2/3}\sqrt[3]{27 + 9i\sqrt{247}}}}}}\},$$

$$(x \to -\frac{1}{4} + \frac{1}{2}\sqrt{\frac{1}{4} + \frac{24\sqrt[3]{3} + \sqrt[3]{2}(27 + 9i\sqrt{247})^{2/3}}{6^{2/3}\sqrt[3]{27 + 9i\sqrt{247}}}} +$$

$$\frac{1}{2}\sqrt{\frac{1}{2} - \frac{4 \cdot 3^{2/3}}{\sqrt[3]{\frac{1}{2}(27 + 9i\sqrt{247})}} - \frac{\sqrt[3]{\frac{1}{2}(27 + 9i\sqrt{247})}}{3^{2/3}} - \frac{1}{\sqrt[4]{\frac{1}{4} + \frac{24\sqrt[3]{3} + \sqrt[3]{2}(27 + 9i\sqrt{247})^{2/3}}{6^{2/3}\sqrt[3]{27 + 9i\sqrt{247}}}}}}),$$

$$x \to -\frac{1}{4} - \frac{1}{2}\sqrt{\frac{1}{4} + \frac{24\sqrt[3]{3} + \sqrt[3]{2}(27 + 9i\sqrt{247})^{2/3}}{6^{2/3}\sqrt[3]{27 + 9i\sqrt{247}}}} -$$

$$\frac{1}{2}\sqrt{\frac{1}{2} - \frac{4 \cdot 3^{2/3}}{\sqrt[3]{\frac{1}{2}(27 + 9i\sqrt{247})}} - \frac{\sqrt[3]{\frac{1}{2}(27 + 9i\sqrt{247})}}{3^{2/3}} + \frac{1}{\sqrt[4]{\frac{1}{4} + \frac{24\sqrt[3]{3} + \sqrt[3]{2}(27 + 9i\sqrt{247})^{2/3}}{6^{2/3}\sqrt[3]{27 + 9i\sqrt{247}}}}}}\},$$

$$\{x \to -\frac{1}{4} - \frac{1}{2}\sqrt{\frac{1}{4} + \frac{24\sqrt[3]{3} + \sqrt[3]{2}(27 + 9i\sqrt{247})^{2/3}}{6^{2/3}\sqrt[3]{27 + 9i\sqrt{247}}}} +$$

$$\frac{1}{2}\sqrt{\frac{1}{2} - \frac{4 \cdot 3^{2/3}}{\sqrt[3]{\frac{1}{2}(27 + 9i\sqrt{247})}} - \frac{\sqrt[3]{\frac{1}{2}(27 + 9i\sqrt{247})}}{3^{2/3}} + \frac{1}{\sqrt[4]{\frac{1}{4} + \frac{24\sqrt[3]{3} + \sqrt[3]{2}(27 + 9i\sqrt{247})^{2/3}}{6^{2/3}\sqrt[3]{27 + 9i\sqrt{247}}}}}}\}\}$$

Із представлених коренів рівняння видно, що формат висновку коренів розв'язку зберігається. Числові розв'язку рівняння 4-го степеня із заданою точністю наведені нижче:

$In[14] := N[solution4 /. \{b \to 1, c \to 3\}]$

$Out[14] = \{\{x \to 0.728892 - 0.895909i\}, \{x \to 0.728892 + 0.895909i\},$
$\{x \to -1.22889 - 0.859538i\}, \{x \to -1.22889 + 0.859538i\}\}$

$In[15] := N[solution4, \{b \to 1, c \to 3\}, 25]$

$Out[15] = \{\{x \to 0.7288920612024940952334696 - 0.8959093224682316151363582i\},$
$\{x \to 0.7288920612024940952334696 + 0.8959093224682316151363582i\},$
$\{x \to -1.122889206120249409523347 0 - 0.8595381439383880356571 45i\},$
$\{x \to -1.228892061202494095233470 + 0.8595381439383880356571 45i\}\}$

$In[16] := N[solution4 /. \{b \to 10, c \to 3\}] // Chop$

$Out[16] = \{\{x \to 0.341232 - 0.566877i\}, \{x \to 0.341232 + 0.566877i\}, \{x \to -9.997\}, \{x \to -0.685467\}\}$

$In[18] := N[solution4 /. \{b \to 1, c \to 3\}, 25] // Chop$

$Out[18] = \{\{x \to 0.7288920612024940952334696 - 0.8959093224682316151363582i\},$
$\{x \to 0.7288920612024940952334696 + 0.8959093224682316151363582i\},$
$\{x \to -1.228892061202494095233470 - 0.8595381439383880356571 45i\},$
$\{x \to -1.228892061202494095233470 + 0.8595381439383880356571 45i\}\}.$

2.1.3. Рівняння вищого порядку. Розв'язок у квадратурах

Оператор *Solve* можна також застосувати й для знаходження розв'язку для рівняння більшого порядку чим 2. Однак точні розв'язки насправді одержати не завжди вдається. Приведемо приклади, коли це можливо:

$\ln[4] := equationFourth = x\text{^}4 + (3-a)x\text{^}3 + (1-3a)x\text{^}2 + (3-a)x = 3a$

$\ln[6] := solutionFourth = Solve[equationFourth, x]$

$Out[6] = \{\{x \to 3\}, \{x \to -i\}, \{x \to i\}, \{x \to a\}\}$

$\ln[8] := x /. solutionFourth$

$Out[8] = \{-3, -i.i, a\}.$

Також легко можна знайти квадрати коренів

$\ln[9] := x\text{^}2 /. solutionFourth$

$Out[9] = \{9, -1, -1, a^2\}$

або суму квадратів коренів

$\ln[11] := Sum[x\text{^}2 /. solutionFourth[[i]]\{i,4\}]$

$Out[11] = a^2 + 7.$

Знайдемо розв’язки для рівняння п'ятого степеня

$\ln[12] := Solve[x\text{^}5 - 8x\text{^}4 + 24x\text{^}3 - 34x\text{^}2 + 23x - 6 = 0, x]$

$Out[12] = \{\{x \to 1\}, \{x \to 1\}, \{x \to 1\}, \{x \to 2\}, \{X \to 3\}\}.$

Однак у більшості випадків знайти точне рішення не вдається. Такий приклад наведений нижче:

$\ln[13] := Solve[x\text{^}5 - x + 1 = 0]$

$Out[13] = \{\{x \to Root[\#1^5 - \#1 + 1\&, 1]\}, \{x \to Root[\#1^5 - \#1 + 1\& \, 2]\},$

$\{x \to Root[\#1^5 - \#1 + 1\&, 3]\}, \{x \to Root[\#1^5 - \#1 + 1\&, 4]\}, \{x \to Root[\#1^5 - \#1 + 1\&, 5]\}\}.$

2.2. Числові розв’язки для рівнянь з однією змінною

2.2.1. Числові розв’язки

Якщо точні розв’язки одержати не вдається, то зручно скористатися числовим розв’язком, як це показано нижче для рівняння п'ятого степеня

$\ln[15] := Quit[].$

Символьний вигляд розв’язку поліноміального рівняння п'ятого степеня має вигляд:

$\ln[1] := rootsNumerical = Solve[x\text{^}5 - x + 1 = 0]$

$Out[13] = \{\{x \to Root[\#1^5 - \#1 + 1\&, 1]\}, \{x \to Root[\#1^5 - \#1 + 1\& \, 2]\},$

$\{x \to Root[\#1^5 - \#1 + 1\&, 3]\}, \{x \to Root[\#1^5 - \#1 + 1\&, 4]\}, \{x \to Root[\#1^5 - \#1 + 1\&, 5]\}\}.$

Крім того, *Mathematica* дозволяє визначити числові значення коренів попереднього рівняння, використовуючи введене раніше ім'я розв’язку

$\ln[2] := rootsNumerical // N$

$Out[2] = \{\{x \to -1.1673\}, \{x \to -0.181232 - 1.08395i\}, \{x \to -0.181232 + 1.08395i\},$

$\{x \to 0.764884 - 0.352472i\}, \{x \to 0.764884 + 0.352472i\}\}.$

Якщо точність розв'язку недостатня, то це уточнення можливо вказати в операторі як додатковий параметр розв'язку.

Приклад знаходження кореня із двадцятьма значущими цифрами після коми наведений нижче:

$$\text{In[3]} := rootsNumerical = NSolve[x\text{^}5 - x + 1 = 0, x, 20]$$

$$Out[3] = \{\{x \to -1.673039782614186843\}, \{x \to -0.18123244446987538390 - 1.083954101317710668 43i\},$$
$$\{x \to -0.18123244446987538390 + 1.083954101317710668 43i\},$$
$$\{x \to 0.7648844336005847260 - 0.352471546031726249 3i\},$$
$$\{x \to 0.7648844336005847260 + 0.352471546031726249 3i\}\}.$$

2.3. Декілька рівнянь. Символьні і числові розв'язки

2.3.1. Система рівнянь із двома змінними

Спочатку розглянемо кілька прикладів, коли для системи алгебраїчних рівнянь знаходяться символьні розв'язки.

$$\text{In[1]} := system = \{a^{-2}x^2 + b^{-2}y^2 = 1, cx + y = d\};$$

$$\text{In[2]} := solution[system, \{x, y\}] // Flatten // Simplify$$

$$Out[2] = \{y \to \frac{b^2 d - c\sqrt{a^2 b^2 (b^2 + a^2 c^2 - d^2)}}{b^2 + a^2 c^2}, x \to \frac{cda^2 + \sqrt{a^2 b^2 (b^2 + a^2 c^2 - d^2)}}{b^2 + a^2 c^2},$$
$$y \to \frac{db^2 + c\sqrt{a^2 b^2 (a^2 b^2 (b^2 + a^2 c^2 - d^2)}}{b^2 + a^2 c^2}, x \to \frac{a^2 cd - \sqrt{a^2 b^2 (b^2 + a^2 c^2 - d^2)}}{b^2 + a^2 c^2}\}.$$

Інший приклад, коли одна із шуканих змінних ступенів вище 2.

$$\text{In[5]} := system2 = \{a^2 x + b^2 y^3 = 1, cx + y = d];$$

$$\text{In[6]} := solutionSystem = Solve[system2, \{x, y\}] // Flatten // Simplify$$

$$Out[6] = \{x \to -\frac{1}{18a^2}\left(\begin{array}{l} \frac{12b^2 a^6}{9c^3 b^4 - 9a^2 c^2 db^4 + \sqrt{3}\sqrt{b^6 c^3 (27b^2 c(c - a^2 d)^2 - 4a^6)}} + \\ \frac{6\sqrt[3]{2}\cdot 3^{2/3} a^4}{c\sqrt[3]{9c^3 b^4 - 9a^2 c^2 db^4 + \sqrt{3}\sqrt{b^6 c^3 (27b^2 c(c - a^2 d)^2 - 4a^6)}}} - \frac{9da^2}{c} \\ \frac{3\cdot 2^{2/3}\sqrt[3]{3}\sqrt[3]{9c^3 b^4 - 9a^2 c^2 db^4 + \sqrt{3}\sqrt{b^6 c^3 (27b^2 c(c - a^2 d)^2 - 4a^6)}}\, a^2}{b^2 c^2} + \\ \frac{\sqrt{3}\sqrt{b^6 c^3 (27b^2 c(c - a^2 d)^2 - 4a6)}}{b^4 c^3} - 9 \end{array}\right)\}$$

$$y \to \frac{2\sqrt[3]{3}a^2 + \frac{\sqrt[3]{2}(9c^3b^4 - 9a^2c^2db^4 + \sqrt{3}\sqrt{b^6c^3(27b^2c(c-a^2d)^2 - 4a^6)}\,)^{2/3}}{b^2c}}{6^{2/3}\sqrt[3]{9c^3b^4 - 9a^2c^2db^4 + \sqrt{3}\sqrt{b^6c^3(27b^2c(c-a^2d)^2 - 4a^6)}}},$$

$$x \to \frac{1}{36a^2}\left(\begin{array}{l} -\frac{24b^2a^6}{9c^2(c-a^2d)b^4 + \sqrt{3}\sqrt{b^6c^3(-4a^6 + 27b^2cd^2a^4 - 54b^2c^2da^2 + 27b^2c^3)}} + \\ \frac{6\sqrt[3]{2}\sqrt[6]{3}(3i+\sqrt{3})a^4}{c\sqrt[3]{9c^2(c-a^2d)b^4 + \sqrt{3}\sqrt{b^6c^3(-4a^6 + 27b^2cd^2a^4 - 54b^2c^2da^2 + 27b^2c^3)}}} + \\ \frac{1}{b^2c^2}\left(3\left(\frac{6cdb^2 + 2^{2/3}\sqrt[3]{3}(1-i\sqrt{3})\cdot}{\sqrt[3]{9c^2(c-a^2d)b^4 + \sqrt{3}\sqrt{b^6c^3(-4a^6 + 27b^2cd^2a^4 - 54b^2c^2da^2 + 27b^2c^3)}}}\right)a^2\right) - \\ \frac{2\sqrt{3}\sqrt{b^6c^3(-4a^6 + 27b^2cd^2a^4 - 54b^2c^2da^2 + 27b^2c^3)}}{b^4c^3} + 18 \end{array}\right),$$

$$y \to -\frac{(1+i\sqrt{3})a^2}{2^{2/3}\sqrt[3]{27c^3b^4 - 27a^2c^2db^4 + \sqrt{(27b^4c^3 - 27a^2b^4c^2d)^2 - 108a^6b^6c^3}}} - \frac{(1-i\sqrt{3})\sqrt[3]{27c^3b^4 - 27a^2c^2db^4 + \sqrt{(27b^4c^3 - 27a^2b^4c^2d)^2 - 108a^6b^6c^3}}}{6\sqrt[3]{2}b^2c},$$

$$x \to \frac{1}{36a^2}\left(\begin{array}{l} -\frac{24b^2a^6}{9c^2(c-a^2d)b^4 + \sqrt{3}\sqrt{b^6c^3(-4a^6 + 27b^2cd^2a^4 - 54b^2c^2da^2 + 27b^2c^3)}} + \\ \frac{6\sqrt[3]{2}\sqrt[6]{3}(-3i+\sqrt{3})a^4}{c\sqrt[3]{9c^2(c-a^2d)b^4 + \sqrt{3}\sqrt{b^6c^3(-4a^6 + 27b^2cd^2a^4 - 54b^2c^2da^2 + 27b^2c^3)}}} + \\ \frac{1}{b^2c^2}\left(3\left(\frac{6cdb^2 + 2^{2/3}\sqrt[3]{3}(1+i\sqrt{3})\cdot}{\sqrt[3]{9c^2(c-a^2d)b^4 + \sqrt{3}\sqrt{b^6c^3(-4a^6 + 27b^2cd^2a^4 - 54b^2c^2da^2 + 27b^2c^3)}}}\right)a^2\right) - \\ \frac{2\sqrt{3}\sqrt{b^6c^3(-4a^6 + 27b^2cd^2a^4 - 54b^2c^2da^2 + 27b^2c^3)}}{b^4c^3} + 18 \end{array}\right),$$

$$y \to -\frac{(1-i\sqrt{3})a^2}{2^{2/3}\sqrt[3]{27c^3b^4 - 27a^2c^2db^4 + \sqrt{(27b^4c^3 - 27a^2b^4c^2d)^2 - 108a^6b^6c^3}}} - \frac{(1+i\sqrt{3})\sqrt[3]{27c^3b^4 - 27a^2c^2db^4 + \sqrt{(27b^4c^3 - 27a^2b^4c^2d)^2 - 108a^6b^6c^3}}}{6\sqrt[3]{2}b^2c},$$

Система трансцендентних рівнянь

```
In[7]:= system3 = {Log[x, y] = 1, Tan[x + y] = d};
In[8]:= solutionSystem = Solve[system3, {x, y}] // Flatten // Simplify
```

Solve::ifun: Inverse functions are being used by Solve, so some

solutions may not be found; use Reduce for complete solution information. More…

$$Out[8] = \{x \to \frac{1}{2}\left(\tan^{-1}(d) - \sqrt{\tan^{-1}(d)^2 - 4e}\right), y \to \frac{1}{2}\left(\tan^{-1}(d) + \sqrt{\tan^{-1}(d)^2 - 4e}\right),$$

$$x \to \frac{1}{2}\left(\tan^{-1}(d) + \sqrt{\tan^{-1}(d)^2 - 4e}\right), y \to \frac{1}{2}\left(\tan^{-1}(d) - \sqrt{\tan^{-1}(d)^2 - 4e}\right)\}.$$

ln[9] := *solutionSystem* = Re *duce*[*system*3,{*x*, *y*}] // *Flatten* // *Simplify*

$$Out[9] = d^2 + 1 = 0 \Lambda x + y = \tan^{-1}(d) + pc_1 \Lambda$$

$$\left(2x + \sqrt{(\tan^{-1}(d) + pc_1)^2 - 4e} = \tan^{-1}(d) + pc_1 \vee \tan^{-1}(d) + pc_1 + \sqrt{(\tan^{-1}(d) + pc_1) - 4e} = 2x\right) \Lambda c_1 \in Z.$$

2.3.2. Числові розв'язки для системи

Для системи рівнянь застосуємо також загальний метод, розглянутий у попередньому розділі. Нехай система рівнянь представлена у вигляді списку

ln[10] := *Quit*[];

ln[1] := *system* = {*x*^2 + *y*^2 = 5, *x* + *y* = 1};

Рішення системи має вигляд

ln[2] := *solutionSustem* = *Solve*[*system*]

$$Out[2] = \{\{x \to -1, y \to 2\}, \{x \to 2, y \to -1\}\}.$$

Перевірка розв'язку записується у вигляді

ln[3] := *system* /.*solutionSustem*

$$Out[3] = \begin{pmatrix} True & True \\ True & True \end{pmatrix}$$

Числові розв'язки для системи, що має вищі степені, чим у попередньому прикладі, наведені нижче.

ln[4] := Re *duce*[$x^3 + 4y^2 = 8$ && $4x^2 - 3y^2 + xy == 1, \{x, y\}$, *Modulus* → 9]

$$Out[4] = (x = 2 \Lambda y = 6) \vee (x = 5 \Lambda y = 0) \vee (x = 8 \Lambda y = 3).$$

Можливо й інше подання для розв'язку системи рівнянь, що представлена в іншому вигляді, а саме як векторна рівність, символьний вигляд якого представлений нижче:

$$\vec{f}(\vec{x}) = \vec{g}(\vec{x}). \qquad (2.1)$$

Умови накладання на функції *f, g* у цьому підручнику не обговорюються. Приклад розв'язку, коли системи алгебраїчних рівнянь задана в явному вигляді:

$$\begin{aligned} x^2 + y^3 &= xy, \\ x + y + x^3 &= 2, \end{aligned} \qquad (2.2)$$

представлено нижче.

Background → *RGBColor*[0.909804,0.937255,0.729412]

ln[4] := *sold*2 = *NSolve*[{*x*^2 + *y*^3, *x* + *y* + *x*^3} = {*x*, *y*,2}]

Out[4] = {{*y* → −1.55329, *x* → 1.30922},{*y* → −0.483769 − 1.53681*i*, *x* → −0.77774 + 1.26471*i*},
{*y* → −0.483769 + 1.53681*i*, *x* → −0.77774 − 1.26471*i*},
{*y* → 1.14965 + 0.514626*i*, *x* → −0.232175 + 1.25385*i*],

{*y* → 1.14965 + 0.514626*i*, *x* → −0.232175 + 1.25385*i*],

{*y* → −0.553608 + 0.841964*i*, *x* → −0.468533 + 1.49106*i*},

{*y* → −0.553608 − 0.841964*i*, *x* → −0.468533 − 1.49106*i*},

{*y* → 0.664376 − 0.41764*i*, *x* → 0.823839 + 0.13843*i*},

{*y* → 0.664376 + 0.41764*i*, *x* → 0.823839 − 0.13843*i*},

Знайдені розв'язки можна виділяти із загального набору в такий спосіб:

ln[5] := *Part*[*sol*2,1,1]

Out[5] = *y* → −1.55329

2.3.3. Метод виключення

Іноді перш ніж вирішувати систему рівнянь зручно виключити яку-небудь змінну. Із цією метою скористаємося оператором

Eliminate[system, variable].

Приклад застосування цього оператора наведений нижче:

ln[6] := *Quit*[];

ln[1] := *system* = {*x*^2 + *y*^3 = *xy*, *x* + *y* + *x*^2 = 3};

ln[2] := *equation* = *E* lim *inate*[*system*, *x*]

Out[2] = $y^6 - y^4 + 8y^3 - y^2 - 9y = -9.$

Потім знаходимо числовий розв'язок знайденого рівняння:

ln[3] := *solution* = *NSolve*[*equation*, *y*]

Out[3] = {{*y* → −1.85177},{*y* → −1.47451},{*y* → 0.776945 − 0.539264*i*},
{*y* → 0.776945 + 0.539264*i*},{*y* → 0.886194 − 1.70288*i*},{*y* → 0.886194 + 1.70288*i*}}

Один із коренів виділяється так:

ln[4] := *Part*[*solution*,1]

Out[4] = {*y* → 1.85177}.

2.3.4. Числові розв'язки для системи

Для системи алгебраїчних рівнянь і для її розв'язку також застосовуються числові розв'язки з використанням оператора *NSolve.*

Приклад, наведений нижче, показує техніку застосування зазначеної процедури

ln[5] := *solution* = *NSolve*[{*x*^2 + *y*^3 = *x*, *x* + *y* + *x*^2 = 2}]

Out[5] = {{*x* → 1.48008, *y* → −1.67071}, {*x* → −1.81375 + 0.581791*i*, *y* → 0.862535 + 1.52866},

{*x* → −1.81375 − 0.581791*i*, *y* → 0/862535 − 1.52866*i*}, {*x* → −2.39927, *y* → −135723},

{*x* → 0.773349 − 0.151175*i*, *y* → 0.651437 + 0.384998*i*},

{*x* → 0.773349 + 0.151175*i*, *y* → 0.651437 − 0.384998*i*}}

2.3.5. Уточнення числових розв'язків

Як уже зазначалося, оператор *Solve* є генератором розв'язків. Разом з тим він не дає відповідь на умови, що накладають на параметри при яких ці розв'язки існують. Для уточнення одержуваних розв'язків використається оператор, представлений нижче:

Reduce [equation, variable]

Reduce [equations, variables]

Розглянемо найпростіший приклад квадратного рівняння, заданого в символьному вигляді:

ln[11] := *Quit*[];

ln[1] := *equation* = *ax*^2 + *bx* + *c* = 0

ln[2] := *solution* = Re *duce*[*equation*, *x*]

$$Out[2] = \left(a \neq 0 \Lambda \left(x = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \vee x = \frac{\sqrt{b^2 - 4ac} - b}{2a}\right)\right) \vee$$

$$\left(a = 0 \Lambda b \neq 0 \wedge x = -\frac{c}{b}\right) \vee \left(c = 0 \Lambda b \neq 0 \Lambda x = -\frac{c}{b}\right) \vee (c = 0 \Lambda b = 0 \Lambda a = 0).$$

Для кубічного рівняння маємо

ln[3] = *equation*3 = $ax^3 + bx^2 + cx + d = 0$

Out[3] = $ax^3 + bx^2 + cx + d = 0$

ln[4] := *solutionCubic* = Re *duce*[*equation*3, *x*] // *Simplify*

Out[4] = (*a* ≠ 0Λ(*Root*[*a*#1^3 + *b*#1^2 + *c*#1 + *d* &1] = *x* ∨

Root[*a*#1^3 + *b*#1^2 + *c*#1 + *d* &,2] = *x* ∨ *Root*[*a*#1^3 + *b*#1^2 + *c*#1 + *d* &,3] = *x*)) ∨

$$\left(a = 0 \Lambda \left(\begin{array}{l} \left(b \neq 0 \Lambda \left(\dfrac{c + 2bx + \sqrt{c^2 - 4bd}}{b} = 0 \vee x = \dfrac{\sqrt{c^2 - 4bd} - c}{2b}\right)\right) \vee \\ b = 0 \Lambda \left((c = 0 \Lambda d = 0) \vee \left(\dfrac{d}{c} + x = 0 \Lambda c \neq 0\right)\right) \end{array}\right)\right)$$

2.4. Трансцендентні рівняння

2.4.1. Загальні відомості

Оператор *Solve* можна застосовувати для розв'язку деяких не поліноміальних рівнянь. Прикладом може бути такий розв'язок:

$\ln[5] := Quit[];$

$\ln[1] := Solve[Sqrt[x+1]+1 == a, x]$

$Out[1] = \{\{x \to a^2 - 2a\}\}.$

Зазначимо, що останній розв'язок справедливо не для всіх a.

$\ln[2] := Solve[Sqrt[x+1] == 0.x]$

$Out[2] = \{\}.$

Разом з тим можна одержати деякий розв'язок трансцендентних рівнянь:

$\ln[3] := Solve[Log[x] == a, x]$

$Out[3] = \{\{x \to e^a\}\}$

$\ln[4] := Solve[Cos[x] == a, x]$

Solve::ifun: Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. More…

$Out[4] = \{\{x \to -\cos^{-1}(a)\}, \{x \to \cos^{-1}(a)\}\}.$

Зазначимо, що рівняння *Cos[x]==a* має нескінченну множину коренів, тому *Mathematica* дає одну із множин відповідно попереджень.

У загальному випадку оператор *Solve* не можна застосовувати для розв'язку трансцендентних рівнянь. Наприклад, таке рівняння не можна розв'язати за допомогою оператора *Solve:*

$\ln[5] := Solve[Sin[x] == x, x]$

Solve::tdep:

The equations apper to involve the variables to be solved for in an essentially non – algebraic way More…

$Out[5] = Solve[\sin(x) = x, x].$

Проте очевидно, що рівняння вигляду *Sin[x]==x* має очевидний розв'язок *x==0.* Отже, для знаходження коренів трансцендентних рівнянь необхідно застосовувати чисові методи.

$\ln[8] := NSolve[-(1+x) + 2\sqrt{1+x^2} - 3(1+x^3)^{1/3} + 5(1+x^5)^{1/5} - 4 == 0, x]$

$Out[8] = \{\{x \to 1.5420843295974531088310402145842984914045152704443465733911 16\},$
$\{x \to 0.9994783429736008164054863920302272787518935584337543198356266\},$
$\{x \to 0.4372960533241537668377494119464693868219855655171474309204049 +$
$0.8387130516562431017585412993546099999266947512452096436074 09i\},$
$\{x \to 0.4372960533241537668377494119464693868219855655171474309204049 -$
$0.8387130516562431017585412993546099999266947512452096436074 09i\},$
$\{x \to -0.5667119642006044530029721606692563498513414700117613723870637\}\}$

2.4.2. Метод Ньютона

Відомо, що для знаходження коренів рівняння виду

$f(x) = 0,$

застосовується метод Ньютона (метод дотичних). Корені трансцендентного рівняння знаходяться так:

$$x_{n+1} = x_n + \frac{f(x_n)}{f'(x_n)}.$$

Так само можна розв’язати за допомогою операторів

FindRoot[f ,{ x, $x0$}];

FindRoot[f ,{ x, $x0$, $x\min$, $x\max$}].

Останній оператор дуже корисний у випадку, якщо є ризик, тому процедура розбіжна.

Розглянемо приклад знаходження кореня за допомогою оператора *FindRoot[...]* для наступного алгебраїчного рівняння:

$$e^{-x} - x^3 = 0 \qquad (2.3)$$

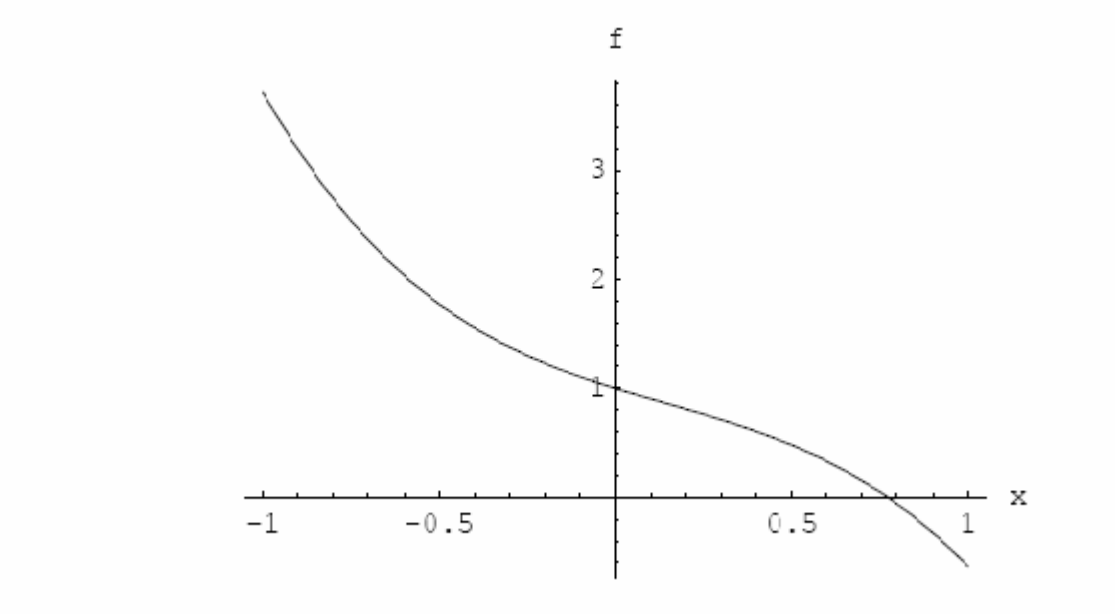
праву частину якого представимо у вигляді такої функції:

ln[9]:= *Quit*[];

ln[2]:= f = *Exp*[$-x$] – x^3

$Out[2] = e^{-x} - x^3$

ln[3]:= gr = *Plot*[f ,{ x,–1,1}, *AxesLabel* → {"x"," f"}];



Далі застосовуємо метод Ньютона зі стартовою точкою *x = 0.*

$\ln[4] := FindRoot[f, \{x, 0\}]$

$Out[4] = \{x \to 0.772883\}.$

Також проводимо перевірку даного розв'язку:

$\ln[5] := f /. \%$

$Out[5] = 0.$

У наступному коді ми зазначимо границі, в яких на нашу думку, лежить корінь, хоча насправді його там немає, і подивимося за розв'язком, який надасть нам *Mathematica*

$\ln[6] := FindRoot[f, \{x, 0., 0., 0.5\}]$

FindRoot:: reged : The point {0.5} is at the edge of the search region {0.5}

In coordinate I and the computed search direction points outside the region. More…

$Out[6] = \{x \to 0.5\}.$

Крім того, ми також можемо відшукати й комплексні корені, беручи за приклад початкове значення комплексного числа:

$\ln[7] := z_0 = -1 + i;$

$\ln[8] := FindRoot[f, \{x, z_0\}]$

$Out[8] = \{x \to -0.184626 + 1.04733i\}.$

2.4.3. Метод Брентона і метод січних

Метод Ньютона потребує, щоб функція була диференційованою, а оператор *FindRoot* надасть відповідний розв'язок. Разом з тим метод Брентона не потребує диференціювання функції, а використовує метод січних для знаходження коренів рівнянь.

Використаємо метод Брентона для попереднього рівняння із вказівкою проміжку, в якому знаходиться корінь:

$\ln[9] := FindRoot[f, \{x, 0., 1.\}]$

$Out[9] = \{x \to 0.772883\}$

і теж саме для січних

$\ln[10] := FindRoot[f, \{x, 0., 0.3\}]$

$Out[10] = \{x \to 0.772883\}.$

2.4.4. Параметри для розв'язку

Оператор дозволяє застосовувати наступні параметри розв'язку:

WokingPrecision, AccuracyGoal,MaxIterations, DampingFactor

Нижче наведені деякі приклади застосування зазначених вище параметрів:

$\ln[11] := Quit[];$

$\ln[1] := eq = Sin(x) - x^{0.5};$

$\ln[2] := FindRoot[eq, \{x, 0.7\}, AccuracyGoal \to 10]$

$Out[2] = \{x \to 1.12961 \times 10^{-15}\}$

далі продовжимо обчислення кореня з останньої точки:

$\ln[3] := FindRoot[eq, \{x, x/.\%\}, AccuracuGoal \to 10]$

$Out[3] = \{x \to 1.12961 \times 10^{-15}\}.$

Дещо змінимо початкове рівняння й простежимо різницю в розв'язку:

$\ln[4] := eq = Sin(x) - x$

$\ln[5] := FindRoot[eq, \{x, 0.7\}, AccuracyGoal \to 10]$

$Out[5] = \{x \to 1.60949 \times 10^{-8}\},$

далі продовжимо обчислення кореня з останньої точки:

$\ln[6] := FindRoot[eq, \{x, x/.\%\}, AccuracyGoal \to 10]$

$Out[6] = \{x \to 1.60949 \times 10^{-8}\}.$

Тема 3

ІНТЕРПОЛЯЦІЙНА ЗАДАЧА

3.1. Вступна частина

У *Mathematica* передбачені чотири види інтерполяцій, оператори яких наведені нижче:

InterpolatingPolynominal[] (“інтерполяційний поліном”);
Interpolation[] (“часткова інтерполяція”);
*SplineFit[] (*сплайн*);*
RationalInterpolation[] (∗ інтерполяція функцією двох поліномів).

Вибір методу інтерполяції числових (іноді й символьних) даних найчастіше залежить від фізичного (у деяких випадках економіко-математичного) змісту розв'язуваної задачі.

3.2. Інтерполяція поліномами

3.2.1. Задання функції у вигляді набору точок

Приведемо коди команд для інтерполяції поліномами, коли задані значення функції в деяких вузлових точках

InterpolatingPolynomial[data,x].

Значення інтерполяційного полінома задані набором точок *data,* причому точки можуть бути задані у такому вигляді:

$\{f_1, f_2, \ldots f_{n+1}\}$;
$\{\{x_1, f_1\}, \{x_2, f_2\}, \ldots \{x_{n+1}, f_{n+1}\}\}$.

Для першого набору справедлива рівномірна інтерполяція, тобто інтерполяція вигляду

$\{\{1, f_1\}, \{2, f_2\}, \ldots \{n+1, f_{n+1}\}\}$,

тоді як для другого набору-нерівномірна.

Приведемо найпростіші приклади. Спочатку розглянемо прості приклади інтерполяції даних записаних у символьному вигляді.

Нехай функція задана трьома точками *(a,f), (b,g), (c,h).* Побудуємо інтерполяційний поліном другого степеня в символьному вигляді:

$\ln[7] := Quit[];$

$\ln[1] := poly = InterpolatingPolynomial[\{\{a, f\},\{b, g\},\{c, h\}\}, x]$

$$Out[1] = f + (x-a)\left(\frac{g-f}{b-a} + \frac{\left(\frac{h-g}{c-b} - \frac{g-f}{b-a}\right)(x-b)}{c-a}\right)$$

$\ln[2] := poly // Expand // Simplify$

$$Out[2] = \frac{1}{(a-b)(a-c)(b-c)}(-(cg+(h-g)x)a^2 + (gc^2 + (h-g)x^2)a +$$

$$c(f-g)(c-x)x + b^2(cf - ah + (h-f)x) + b(ha^2 + (f-h)x^2 - c^2 f))$$

Як видно з останнього результату, отриманий поліном другого степеня.

3.2.2. Числові розв'язки. Графічні подання розв'язків

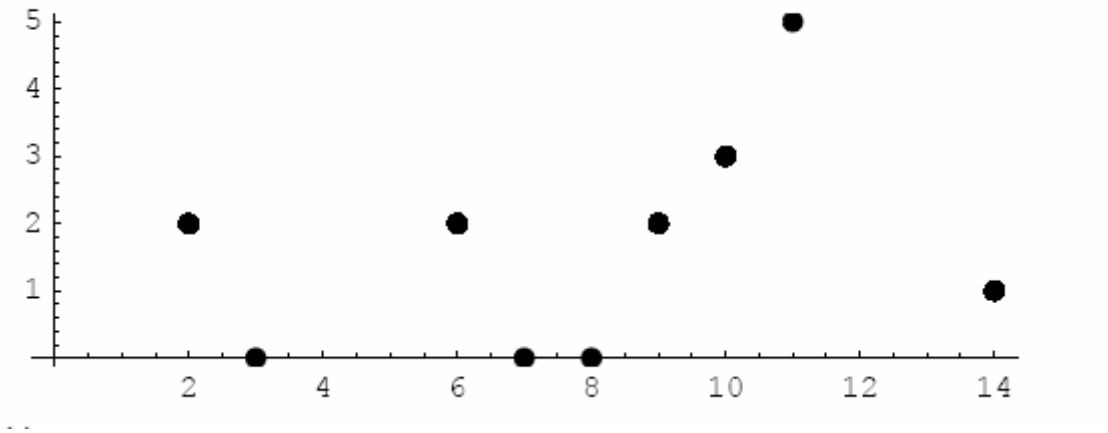
Нехай задані наступні дані:

$\ln[3] := data1 = \{1,2,0,2,2,2,0,0,2,3,5,4,3,1\};$

Наведемо візуалізацію цих даних:

$\ln[4] := graficData =$

$ListPlot[data1, PlotStyle \to Point Size[0,0.2], AspectRatio \to Automatic, AxesOrigin \to \{0,0\}]$



$Out[4] = -Graphics-$

Далі знаходимо інтерполяційний поліном

$\ln[5] := int erpolationPol = InterpolatingPolynomial[data1, x];$

Наведемо таблицю проміжних значень

$\ln[6] := Table = [\{x, int erpolationPol\},\{x,2,4,0.2\}] // TableForm$

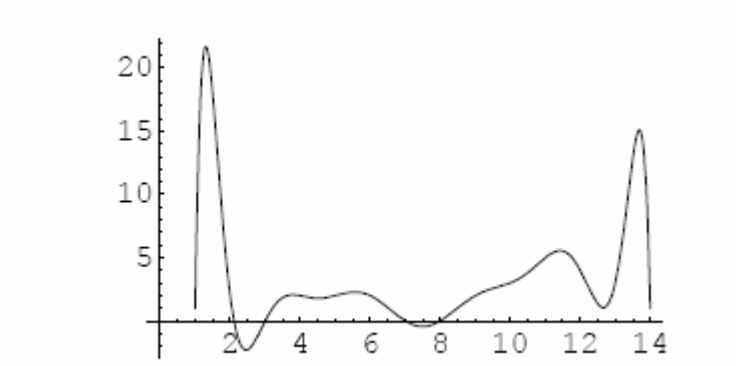
$Out[6] // TableForm =$

2	2
2.2	-1.1265
2.4	-2.25856

2.6	-2.03382
2.8	-1.10846
3	0
3.2	0.962184
3.4	1.6249
3.6	1.96935
3.8	2.06046
4	2

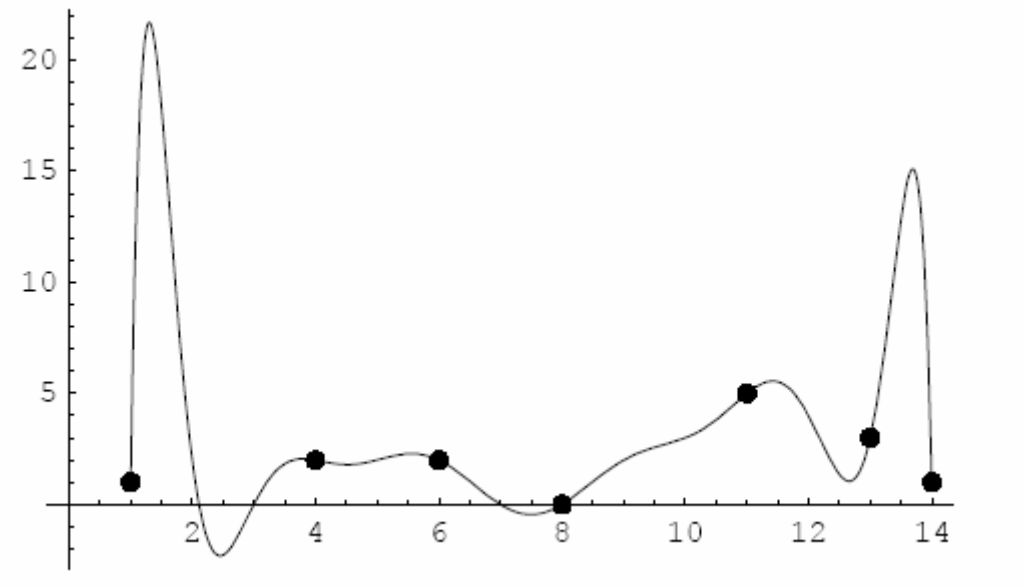
Наведемо графічне зображення інтерполяції

ln[7]:= *jo*int *Graphics* = *Plot*[int *erpolationPol*, {*x*,1,14}];



спільний графік інтерполяційного полінома й точок інтерполяції.

ln[8]:= *Show*[*jo*int *Graphics*, *graphicData*]



Out[8] = –*Graphics* –

3.2.3. Інтерполяційні форми Лагранжа і Ньютона

Наведемо програмні модулі для побудови інтерполяційних поліномів Лагранжа і Ньютона. Інтерполяційний поліном Лагранжа представлений таким програмним модулем:

```
In[9]:= Quit[];
In[1]:= lagrange[dx_List, dy_List, x_]:=
Sum[dy[[i]]Apply[Times, (x - Drop[dx, {i}])/(dx[[i]] - Drop[dx, {i}])], {i, Length[dx]}]
```

Наведемо приклад застосування інтерполяційної форми Лагранжа

```
In[2]:= data = {{a, f}, {b, g}, {c, h}};
In[3]:= {dx, dy} = Transpose[data];
In[4]:= int Lagrange = lagrange[dx, dy, x]
```

$$Out[4] = \frac{h(x-a)(x-b)}{(c-a)(c-b)} + \frac{f(x-c)(x-b)}{(a-b)(a-c)} + \frac{g(x-a)(x-c)}{(b-a)(b-c)}$$

Інтерполяційний поліном Ньютона представлений таким модулем:

```
In[5]:= newtonInterpolation[dx_, dy_, x]:= Module[{n = Length[dx], div},
Do[div[{dx[[i]]}] = dy[[i]], {i, n}];
div[{p_, q_, r_}]:= (div[{q, r}] - div[{p, q}])/(r - p);
Sum[div[Take[dx, i]] Product[x - dx[[j]], {j, i - 1}], {i, n}]]
```

Наведемо приклад застосування інтерполяційної форми Ньютона:

```
In[6]:= dataNewton = {{a, f}, {b, g}, {c, h}};
In[7]:= {dx, dy} = Transpose[dataNewton];
In[8]:= int Newton = newtonInterpolation[dx, dy, x]
```

$$Out[8] = f + \frac{(g-f)(x-a)}{b-a} + \frac{\left(\frac{h-g}{c-b} - \frac{g-f}{b-a}\right)(x-a)(x-b)}{c-a}$$

3.3. Частково-неперервна інтерполяція

3.3.1. Частково-лінійна інтерполяція

Частково-лінійна інтерполяція представляється кодом

```
Interpolation[data, InterpolationOrder → 1].
```

Розглянемо приклад частково-лінійної інтерполяції

```
In[9]:= Quit[].
```

Нехай набір даних представлений нижче:

```
In[1]:= data1 = {1,2,0,2,2,2,0,0,2,3,5,4,3,1};
```

частково-лінійна інтерполяція запишеться наступним оператором

```
In[2]:= int1 = Interpolation[data1, InterpolationOrder → 1]
Out[2]:= InterpolatingFunction[(1 _ 14), <>].
```

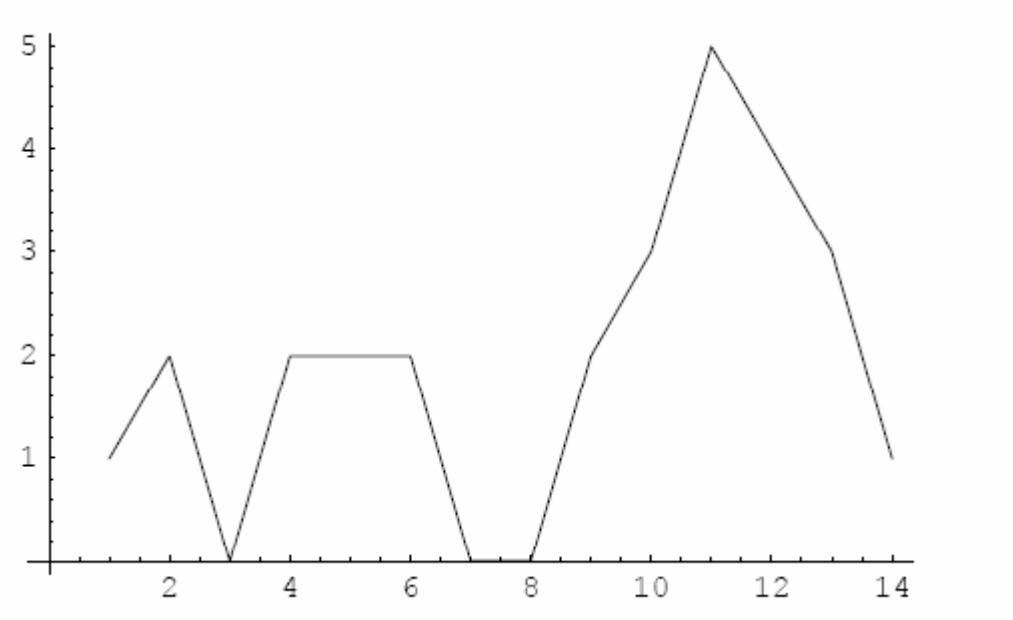
Далі приведемо приклад набору інтерполяційних даних:

```
In[3]:= Table[{x, int1[x]}, {x,2,5,0.2}] // TableForm
Out[3]// TableForm =
```

2	2
2.2	1.6
2.4	1.2
2.6	0.8
2.8	0.4
3.	0.
3.2	0.4
3.4	0.8
3.6	1.2
3.8	1.6
4.	2.
4.2	2
4.4	2
4.6	2
4.8	2
5.	2

Нижче наведемо графік лінійної інтерполяції

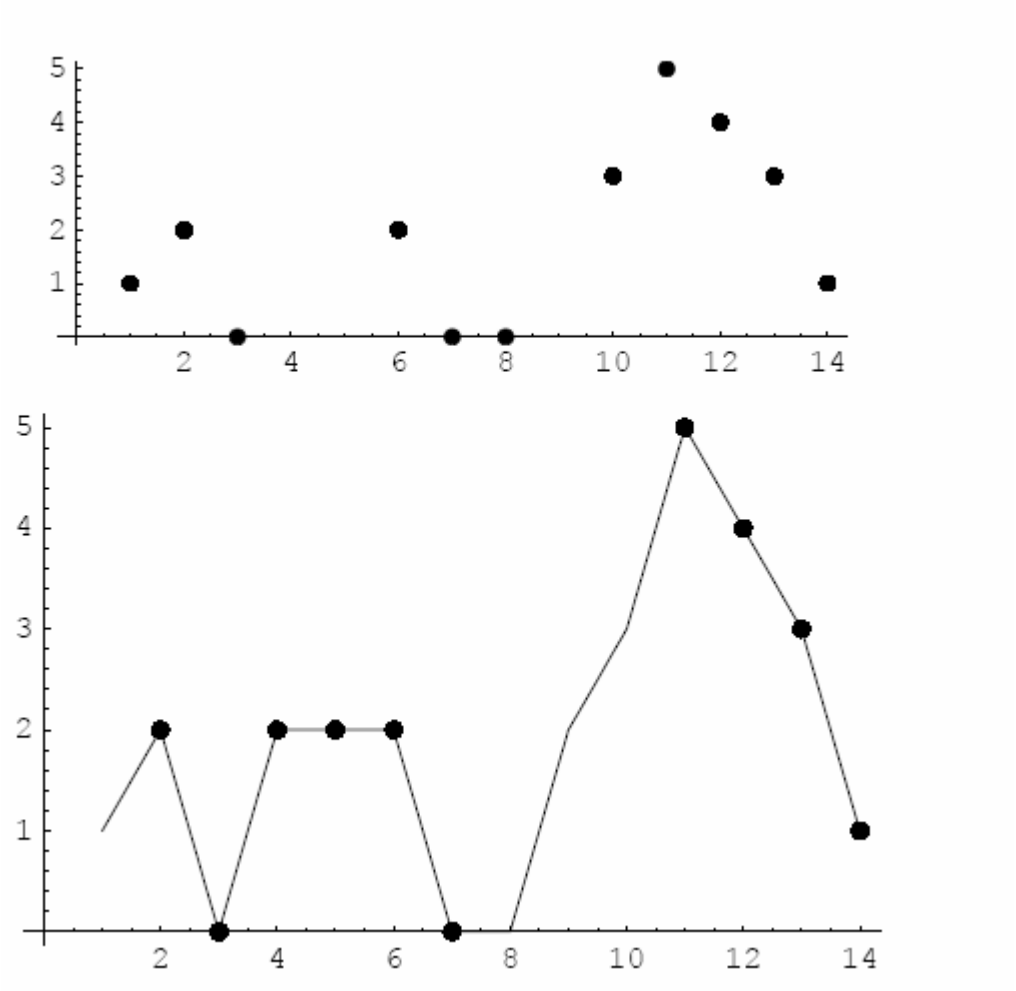
```
In[4]:= graphicInt1 = Plot[int1[x], {x,1,14}]
```



Out[4]=-Graphics-

а також частково-лінійна інтерполяція і точки

ln[6] := *Show*[*graphicInt*1,
ListPlot[*data*1, *PlotStyle* → *Po*int *Size*[0.02], *AspectRatio* → *Automatic*, *AxesOrigin* → {0,0}]]



Out[6] = –*Graphic* –

3.3.2. Частково-кубічна інтерполяція

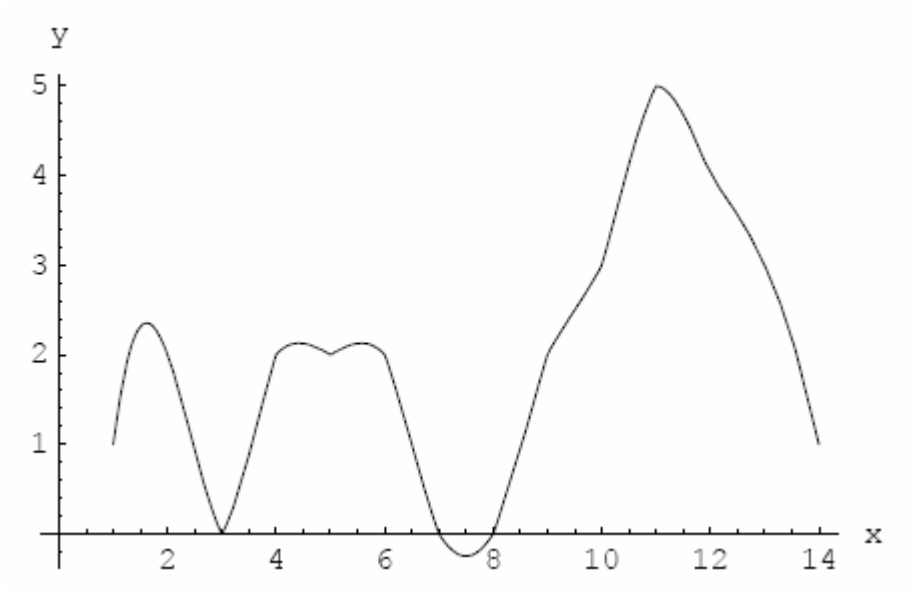
Розглянемо частково-кубічну інтерполяцію, що задається таким оператором:

ln[7] := int 3 = *Interpolation*[*data*1]

Out[7] = *InterpolatingFunction*[(1_14),<>]

Графік інтерполяції має вигляд

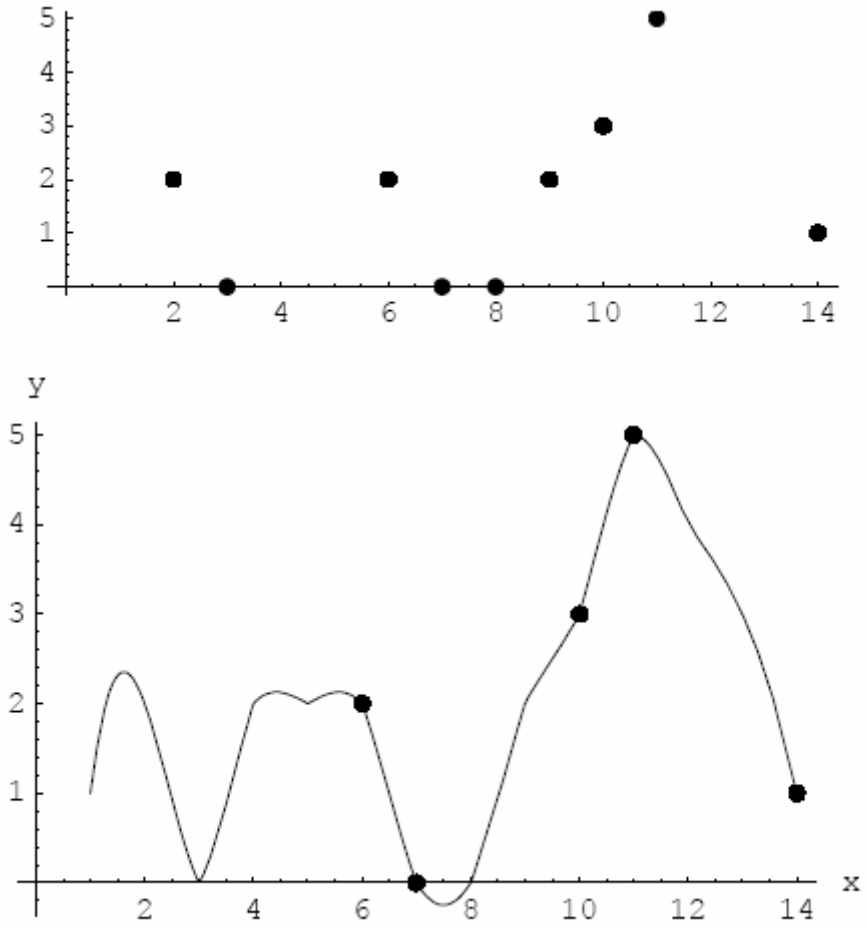
ln[8] := *graphicInt*3 = *Plot*[int 3[*x*], {*x*,1,14}, *AxesLabel* → {"*x*","*y*"}]



Out[8]=-Graphics-

а з'єднаний графік інтерполяції і точок інтерполяції наведений нижче.

ln[9]:= *Show*[*graphicInt*3,
ListPlot[*data*1, *PlotStyle* → *Po*int *Size*[0.02], *AspectRatio* → *Automatic*, *AxesOrigin* → {0,0}]];



3.4. Сплайн–інтерполяція

3.4.1. Поняття про сплайн

Під сплайном розуміється крива, що проходить через задані три точки. Це визначення розраховане тільки на інтуїтивне сприйняття читача явища, що відбувається.

3.4.2. Сплайн-інтерполяція

У *Mathematica* сплайн-інтерполяція задається операторами

SplineFit[data, type];
SplineFit[data, Cubic].

Розглянемо спочатку кубічний сплайн:

нехай дані задані у вигляді точок

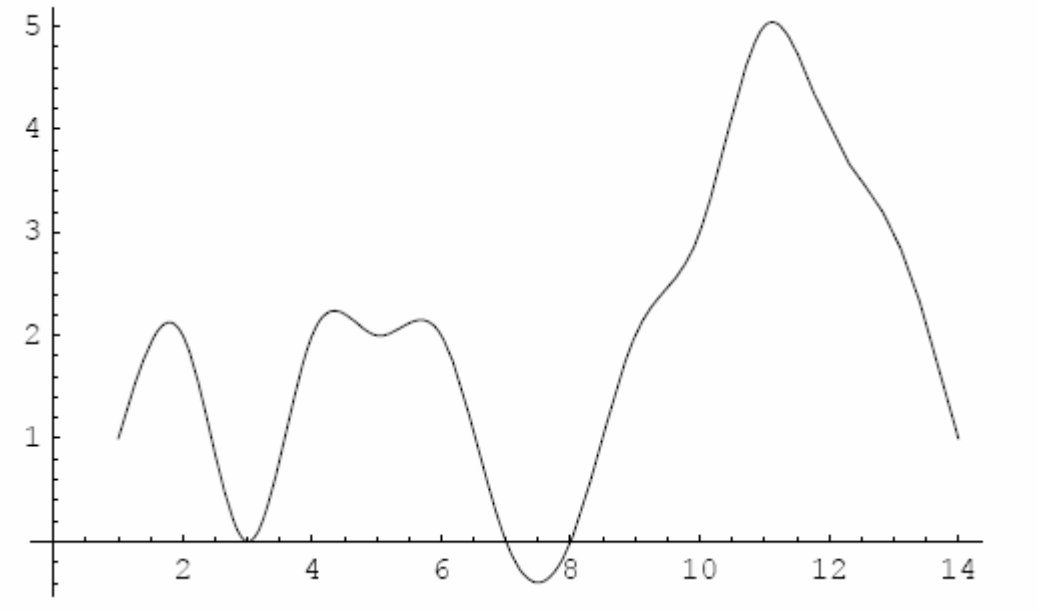
```
In[10]:= Quit[];
In[1]:= data3 = {{1,1},{2,2},{3,0},{4,2},{5,2},
{6,2},{7,0},{8,0},{9,2}{10,3},{11,5},{12,4},{13,3},{14,1}}.
```

Перш ніж застосовувати код операції спочатку завантажуємо пакет

```
In[2]:=<< NumericalMath'SplineFit'
In[3]:= cubFit = SplineFit[data3, Cubic]
Out[3] = SplineFunction[Cubic, {0,13}, <>].
```

Далі будуємо графік кубічної сплайн-інтерполяції:

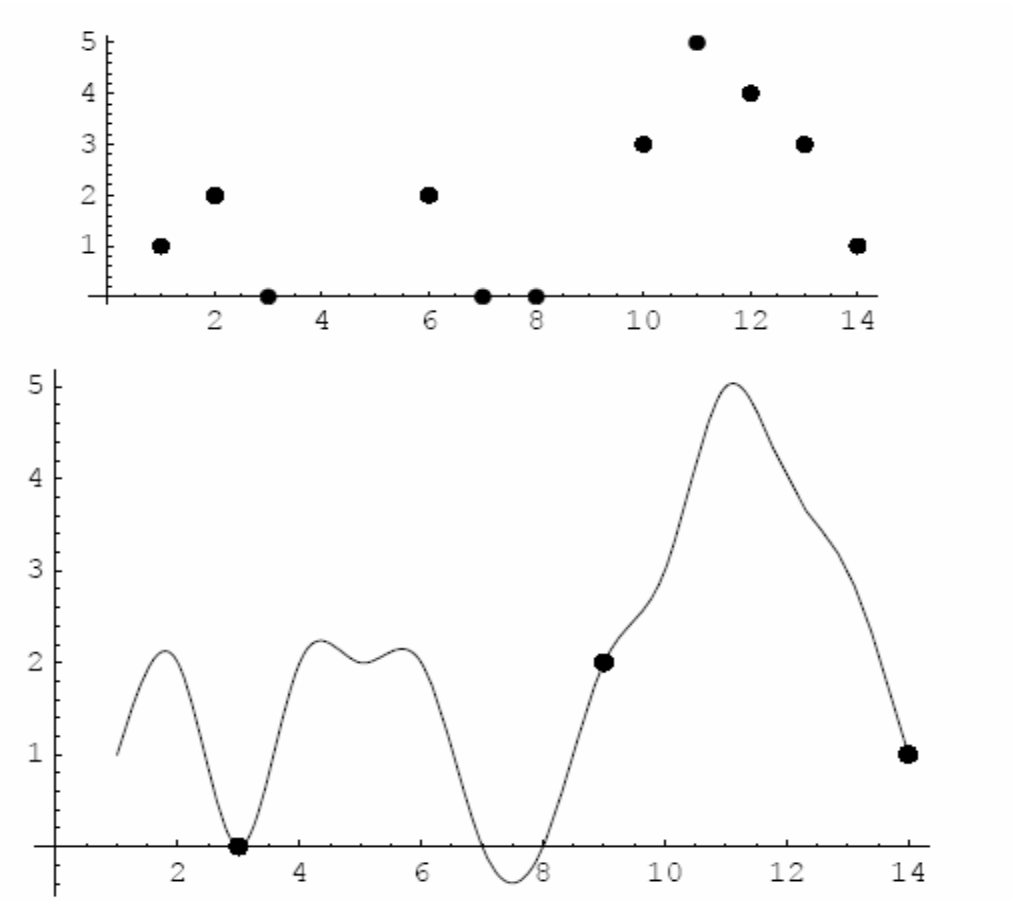
```
In[4]:= graphicCubicFit = ParametricPlot[cubFit[t], {t,0,13}, Complited → False]
```



Out[4]:=-Graphics-

і з'єднанні графіки сплайна і точок інтерполяції

ln[6]:= *Show*[*graphicCubicFit*,
ListPlot[*data*3, *PlotStyle* → *Po*int *Size*[0.02], *AspectRatio* → *Automatic*, *AxesOrigin* → {0,0}]]



Out[6]:=-Graphics-

3.4.3. Сплайн на площині

Розглянемо інтерполяцію даних на площині *(X,Y).*

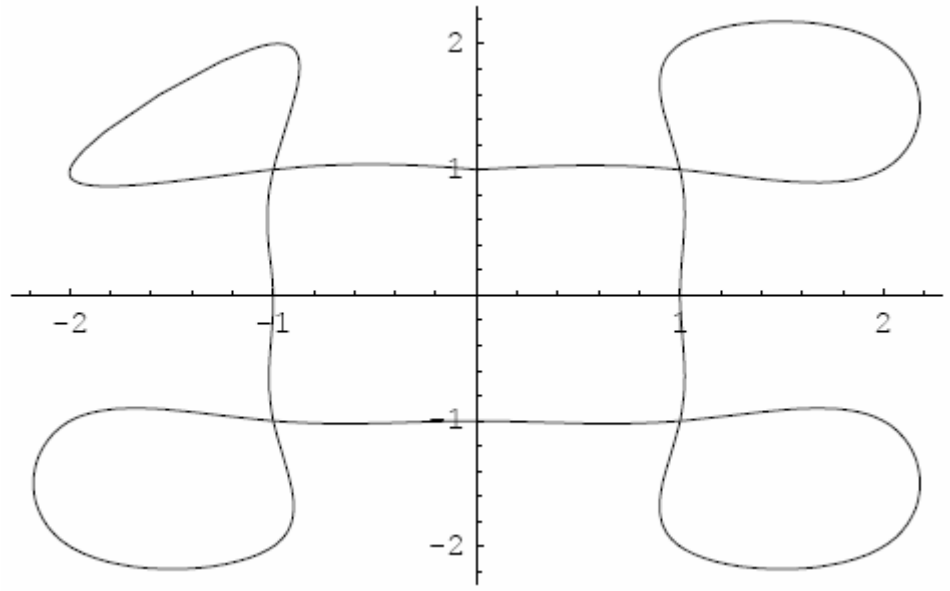
ln[7]:= *data*4 = {{0,1},{1,1},{2,1},{2,2},{1,2},{1,1},{1,0},{1,−1},
{1,−2},{2,−2},{2,−1},{1,−1},{0,−1],{−1,−1},{−2,−1},{−2,−2},
{−1,−2},{−1,−1},{−1,0},{−1,−1},{−1,2},{−2,1},{−1,1},{0,1}};

ln[8]:= *cubPlan* = *SplineFit*[*data*4, *Cubic*]

Out[8] = *SplineFunction*[*Cubic*,{0.,23},<>]

Будуємо графік сплайн-інтерполяції на площині

ln[9]:= *graphicPlan* = *ParametricPlot*[*cubPlan*[*t*],{*t*,0,23}, *Complited* → *False*]



Out[6]:=-Graphics-

Як і в попередніх прикладах можна обчислити значення в точці

ln[10] := *cubPlan*[4.41]

Out[10] = {0.942715,1.32369}.

3.4.4. Bezier-сплайн

Розглянемо Bezier-сплайн, що проходить через крайні точки, а від інших знаходиться на раціональній з погляду сплайна відстані.

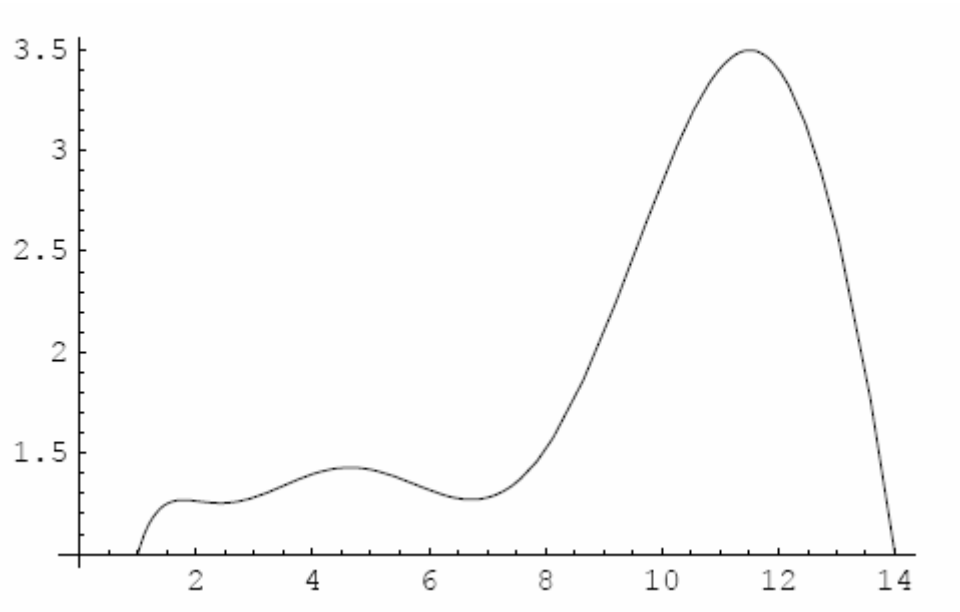
ln[11] := *data*3 = {{1,1},{2,2},{3,0},{4,2},{5,2},
{6,2},{7,0},{8,0},{9,2},{10,3},{11,5},{12,4},{13,3},{14,1}};

ln[12] = *bezierFit* = *SplineFit*[*data*3, *Bezier*]

Out[12] = *SplineFunction*[*Bezier*,{0.,13.},<>]

Графік Bezier-сплайна, що побудований по точках, заданих вище, представлений на графіку

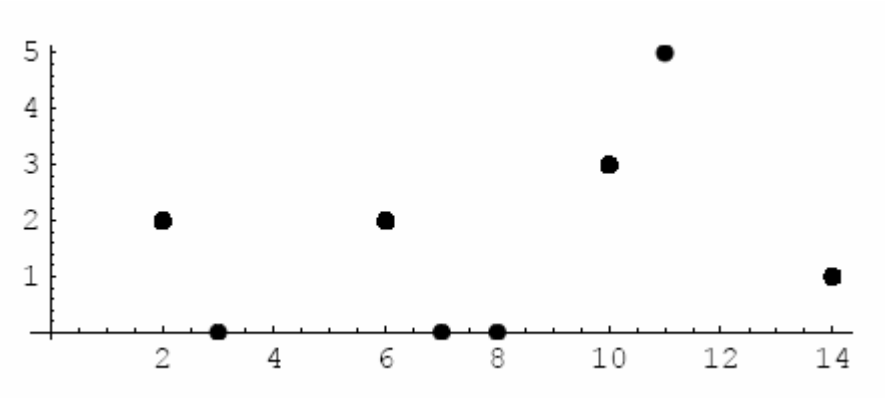
ln[13] := *graphicBezierFit* = *ParametricPlot*[*bezierFit*[*t*],{*t*,0,13}, *Compiled* → *False*]

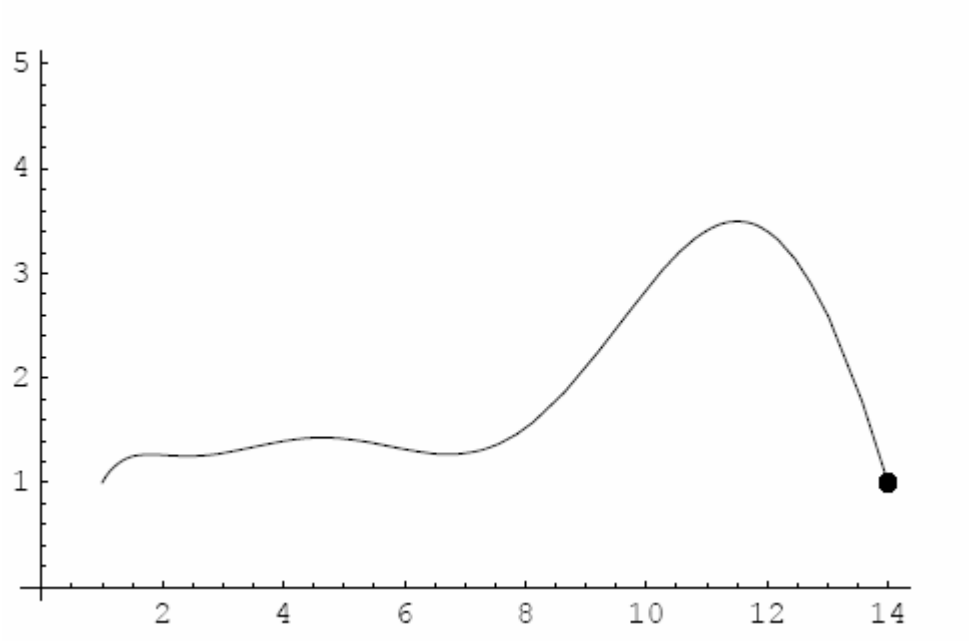


Out[6]:=-Graphics-

Відповідно, Bezier-сплайн і розташування вихідних точок інтерполяції представлені нижче на спільному графіку.

ln[14]:= *Show*[*graphicBezierFit*, *ListPlot*[*data*3, *PlotStyle* → *Po*int *Size*[0.02], *AspectRatio* → *Automatic*, *AxesOr*]]





Out[6]:=-Graphics-

Із представленого спільного графіка видно, що Bezier-сплайн проходить через крайні точки і вільно "прогинається" всередині точок інтерполяції.

3.4.5. BezierComposite-сплайн

Розглянемо BezierComposite-сплайн, що проходить через першу, третю, п'яту і т.п. точки, а від інших стоїть на "керованій" з погляду сплайна відстані.

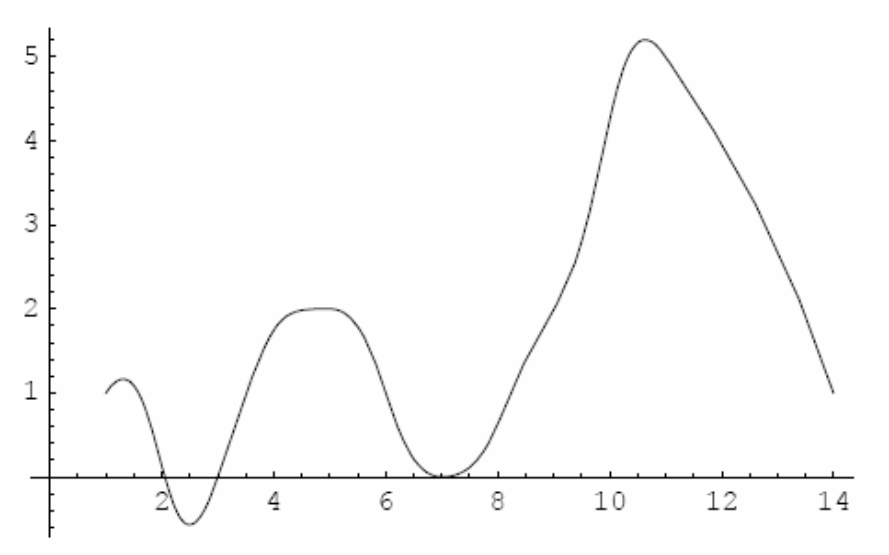
```
In[15]:= data3 = {{1,1},{2,2},{3,0},{4,2},{5,2},
{6,2},{7,0},{8,0},{9,2},{10,3},{11,5},{12,4},{13,3},{14,1}}.
```

BezierComposite-сплайн визначений відповідним сплайном (параметром інтерполяції) в операторі *SplineFit[...].*

```
In[16]:= bezierCompositeFit = SplineFit[data3, CompositeBezier]
Out[16] = SplineFunction[CompositeBezier, {0.,13.}, <>]
```

Графік BezierComposite-сплайна має такий вигляд:

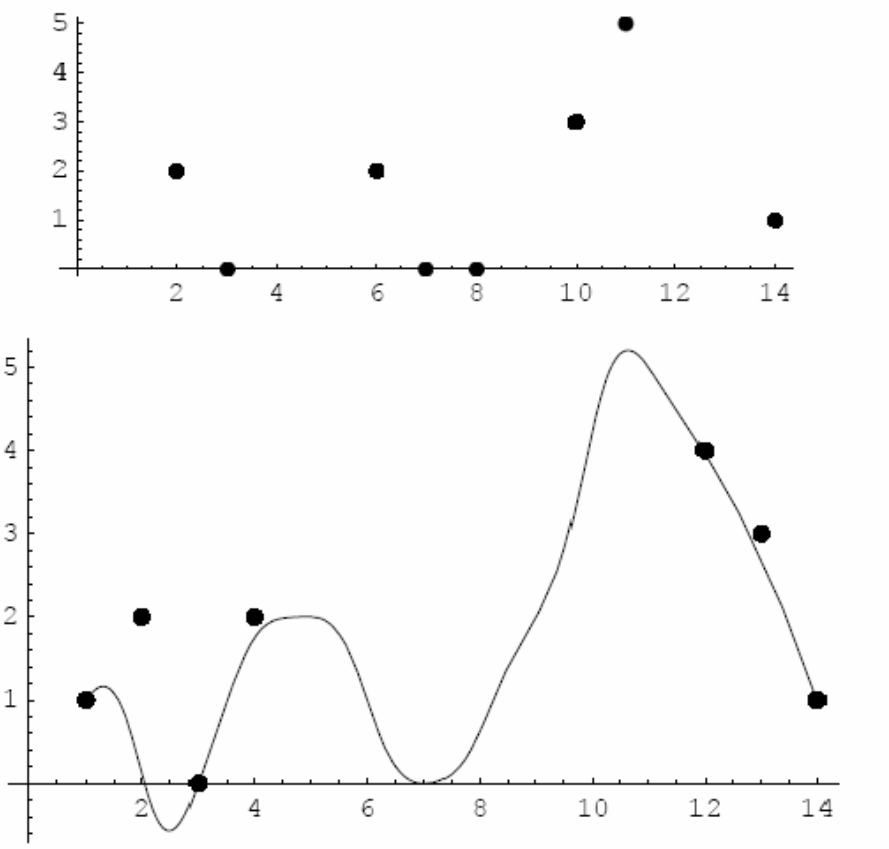
```
In[17]:= graphicCompositeBezierFit = ParametricPlot[bezierCompositeFit[t], {t,0,13},
Complited → False]
```



Out[6]:=-Graphics-

а спільний графік BezierComposite-сплайна і точок інтерполяції, частина з яких лежить на BezierComposite-сплайні, представлений нижче.

ln[18]:= *Show*[*graphicCompositeBezierFit*, *ListPlot*[*data*3, *PlotStyle* → *Po*int *Size*[0.02], *AspectRatio* → *Automat*]]



Out[6]:=-Graphics-

Таким чином, у цьому розділі розглянуто різні методи інтерполяції даних, які можуть бути використані при розв’язання практичних задач.

Тема 4

АПРОКСИМАЦІЯ ЧИСЛОВИХ ДАНИХ

4.1. Вступна частина

До апроксимації чисел завжди необхідно ставитися обережно. Знаменита фраза А.Ейнштейна не втратить своєї актуальності доти, поки числові дані (переважно це дані експериментальних досліджень) будуть мати практичний інтерес у повсякденній діяльності людини, а прогноз базуватиметься на розрахунку, а не на вгадуванні.

4.1.1. Метод найменших квадратів

Основний оператор, що використовується при апроксимації даних, має формат *Fit[data, basis, variable].*

Застосування зазначеного оператора зобразимо на таких прикладах.

Оскільки вихідні дані можна згенерувати в *Mathematica*, то скористаємося тими засобами, які надає комп'ютер.

Спочатку треба згенерувати дані, що підлягають апроксимації. Для цього завантажимо пакет

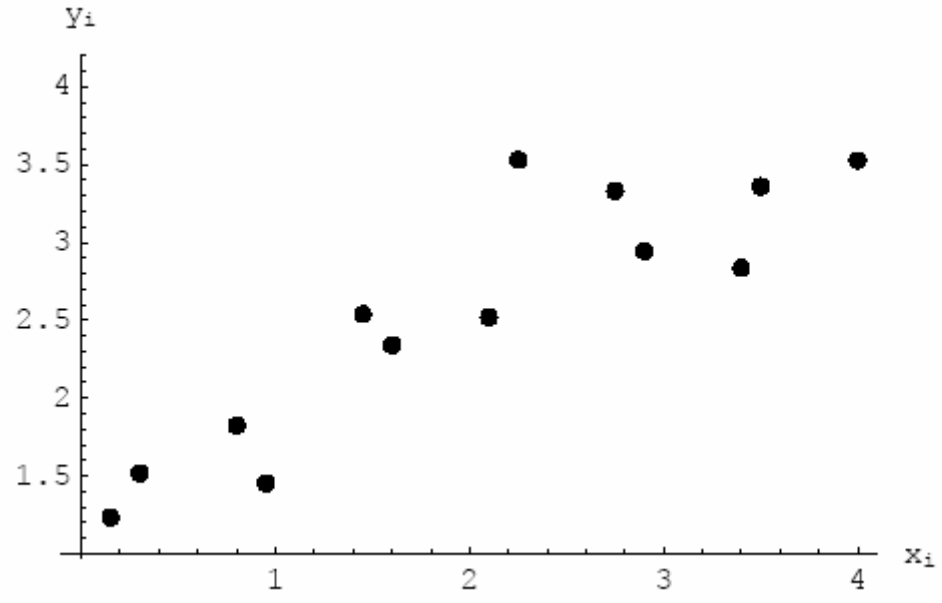
```
In[1]:=<< Statistic`NormalDistribution`
```

і генеруємо набір "експериментальних" даних підлягаючих апроксимації.

```
In[2]:= dataError= Table[{x,1+ x – 0.1x^2 + 0.3Random[NormalDistribution[0,1]]},{x,0,4,0.05}];
```

Потім переглянемо здобуті дані

```
In[3]:= graphicData =
ListPlot[dataError, PlotRange → All, PlotStyle → PointSize[0.02], AxesLabel → {"x_i","y_i"}]
```



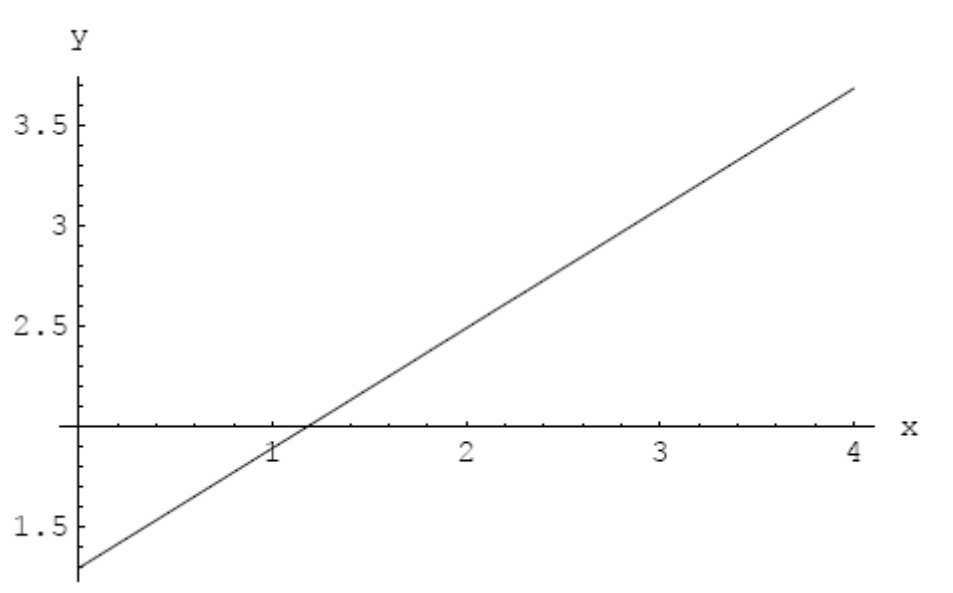
Out[3]:=-Graphics-

Далі спробуємо апроксимувати ці дані за допомогою полінома першого степеня

```
In[4]:= dataFit = Fit[dataError, {1, x}, x]
Out[4]= 0.595998x + 1.29807.
```

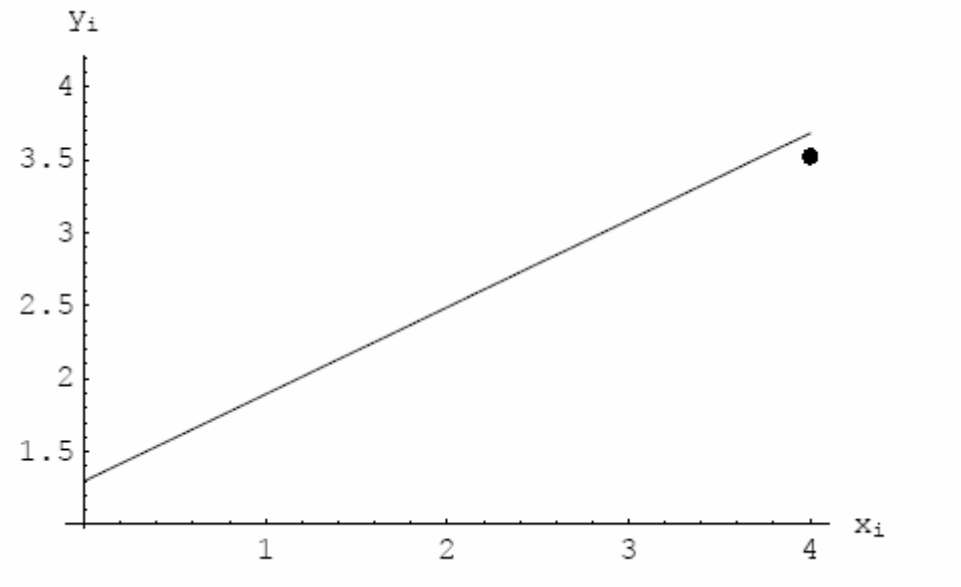
Побудуємо графік апроксимації

```
In[6]:= grFit = Plot[dataFit, {x,0,4}, AxesLabel → {"x","y"}];
```



і сполучимо функцію апроксимації з вихідними "експериментальними" даними.

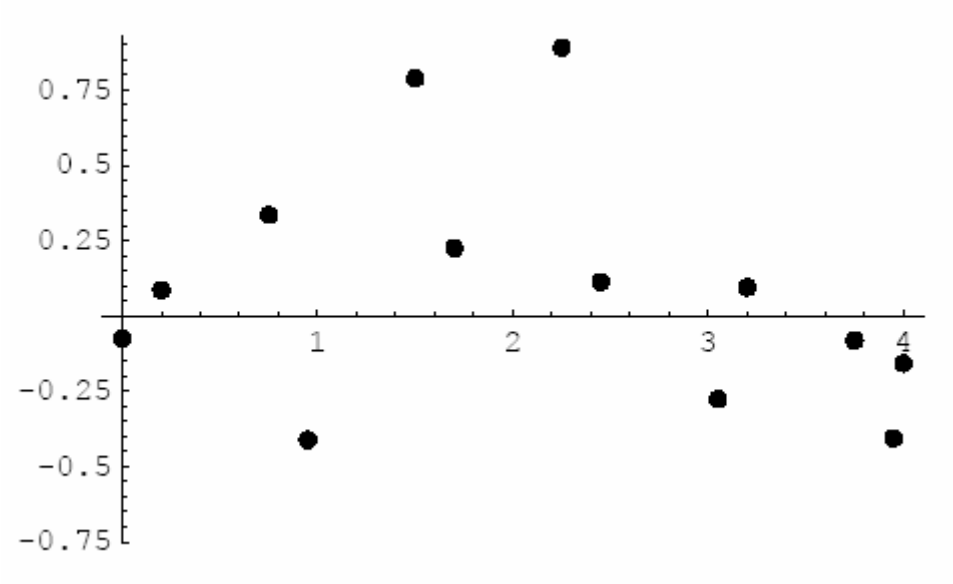
```
In[7]:= Show[graphicData, grFit];
```



Відповідність даних видно із графіка.

Далі проведемо різницевий аналіз між вихідними даними і апроксимацією. Аналіз виконаємо графічно й числовим методом.

```
In[8] := {x1, f1} = Transpose[dataError];
In[9] := pred1 = dataFit /. x → x1;
In[10] := res1 = Transpose[{x1, f1 − pred1}];
In[11] := graphicRes = ListPlot[res1, PlotRange → All, PlotStyle → PointSize[0.02]];
```

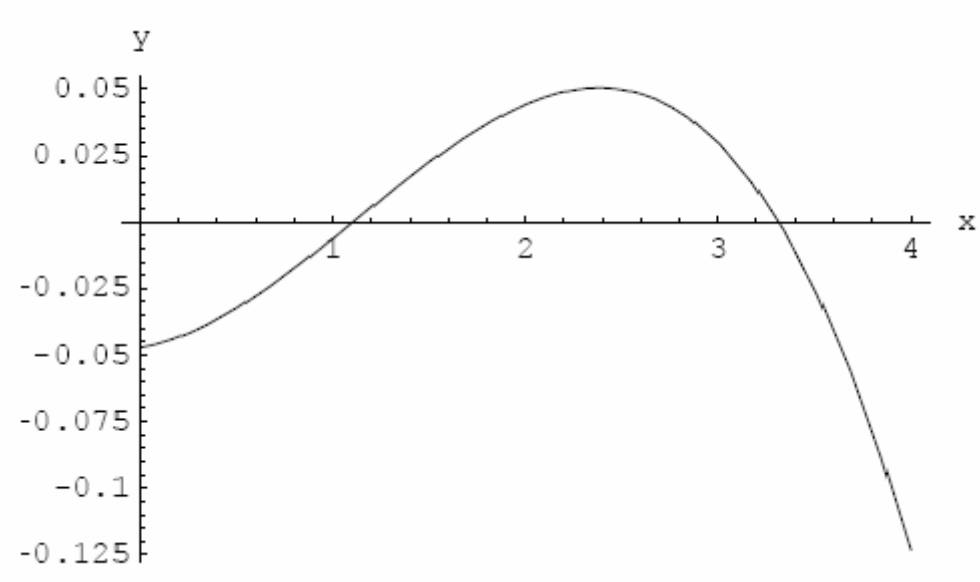


Як видно з останнього графіка, різниця між даними й апроксимацією задається випадковою величиною. Тому досліджуємо останні дані за допомогою кубічної апроксимації.

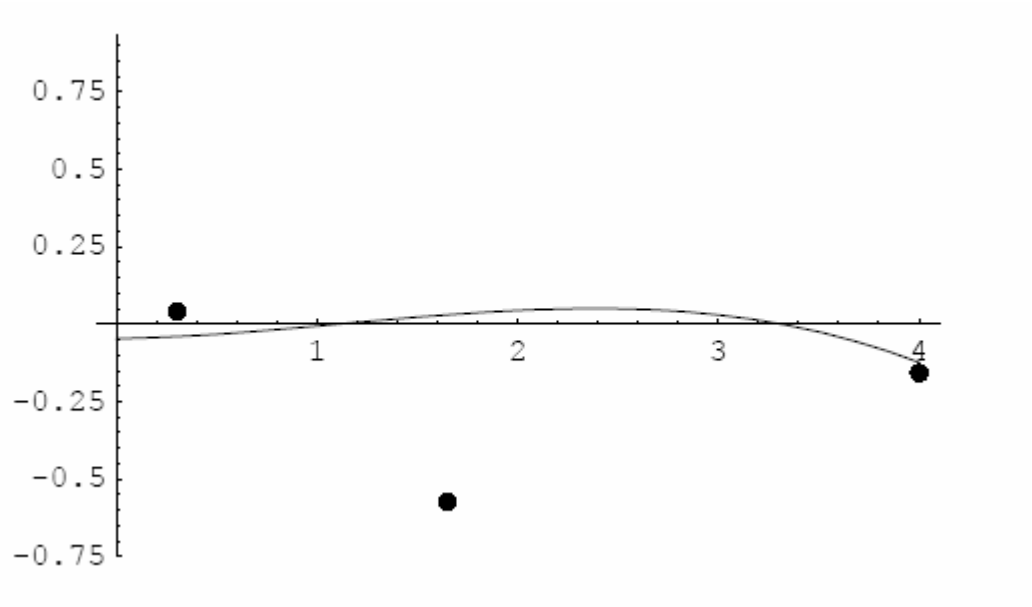
```
In[12] := resFit[res1, {1, x, x^2, x^3}, x]
Out[12] = −0.0123531x^3 + 0.0417769x^2 + 0.0115123x − 0.0471068
```

Потім будуємо графік кубічної апроксимації і різницевих даних.

```
In[13] := graphicResFit = Plot[resFit, {x,0,4}, AxesLabel → {"x","y"}];
```




```
In[14] := Show[graphicResFit];
```



Останній графік показує, що лінійна апроксимація не адекватна вихідним даним. Це видно з того, що різниця містить регулярну складову. Обчислимо суму квадратів відмінностей між даними і апроксимацією (по суті, це міра відстані між функціями).

```
In[15]:= Apply[Plus,(f1-pred1)^2]
Out[15]= 8.60615
```

Оскільки лінійне наближення неадекватне вихідним даним, то процедуру наближення доводиться повторювати. Для того щоб автоматизувати цей процес, приведемо модуль, що дає змогу це робити відразу.

```
In[16]:= Off[General::spell]
In[17]:= showfit[data_,datafit_,var_,deg_]:=
Module[{x,f,pred,gdata,gdatafit,gdatasum},{x,f}=Transpose[data];
pred=datafit/.var→x;
res=Transpose[{x,f-pred}];
resfit=Fit[res,Table[var^i,{i,0,deg}],var];
Print["Sum of squared residuals:",Apply[Plus,(f-pred)^2]];
Block[{SdisplayFunction=Identity,
opt={PlotRange→All,AxesOrigin→{0,0}},a=data[[1,1]],b=Last[data][[1]]},
gdata=ListPlot[data,opt,PlotStyle→PointSize[0.02]];
gdatafit=Plot[datafit,{var,a,b}];
gdatasum=Show[gdata,gdatafit];
gres=ListPlot[res,opt,PlotStyle→PointSize[0.02]];
gresfit=Plot[resfit,{var,a,b}];
gressum=Show[gres,gresfit]];
Show[GraphicsArray[{gdatasum,gressum}]];]
```

4.1.2. Логарифмічне перетворення

Якщо вихідні дані, на перший погляд, ілюструють експонентне зростання, то кандидатом на апроксимацію буде функція вигляду

$$f(x) = e^{(a+bx)} \tag{4.1}$$

але, як видно з співвідношення (4.1), функція апроксимації є істотно нелінійною за параметрами *a* і *b*. Тільки тому прямо й автоматично оператор *Fit[...]* застосовувати неможна. Однак якщо до функції (4.1) спочатку застосувати операцію логарифмування в такому вигляді

$$Log[\, f(x)] = (a + bx), \tag{4.2}$$

тоді права частина стає лінійною функцією змінної *x*, до якої можна застосувати оператор *Fit[...]*, що дозволить виконати апроксимацію вихідних даних.

Далі розглянемо приклад

```
In[18]:= Quit[];
In[1]:=<< Statistics`NormalDistribution`
In[2]:= data4 = Table[{x,1.2Exp[0.2x] + 0.4Random[NormalDistribution[0,1]]},{x,0,10,0.1}];
```

візьмемо логарифм від даних, а потім застосуємо процедуру апроксимації

```
In[3]:= log data4 = Map[{#[[1]], Log[#[[2]]]}&, data4];
In[4]:= log data4 fit = Fit[log data4, {1, x}, x].
```

Функція логарифмічної апроксимації має вигляд

```
In[5]:= data4 fit = Exp[log data4 fit]
Out[5]= e^(0.216005 x+0.0442202)
```

що дозволяє виконати обчислення апроксимуючої функції у довільній точці, наприклад, яка належить до вихідних даних.

```
In[6]:= data4 fit /. x → 5
Out[6]= 3.07789
```

Нарешті застосуємо модуль автоматичного вибору апроксимуючої функції.

```
In[8]:= Off[General :: spell]
In[9]:= showfit[data _, datafit _, var_, deg_] :=
Module[{x, f, pred, gdata, gdatafit, gdatasum}, {x, f} = Transpose[data];
 pred = datafit /. var → x;
 res = Transpose[{x, f − pred}];
 resfit = Fit[res, Table[var^i, {i, 0, deg}], var];
Print["Sum of squared residuals:", Apply[Plus, (f-pred)^2]];
Block[{SdisplayFunction = Identity,
opt = {PlotRange → All, AxesOrigin → {0, 0}}, a = data[[1, 1]], b = Last[data][[1]]},
```

gdata = ListPlot[data, opt, PlotStyle → PointSize[0.02]];

gdatafit = Plot[datafit,{var, a, b}];

gdatasum = Show[gdata, gdatafit];

gres = ListPlot[res, opt, PlotStyle → PointSize[0.02]];

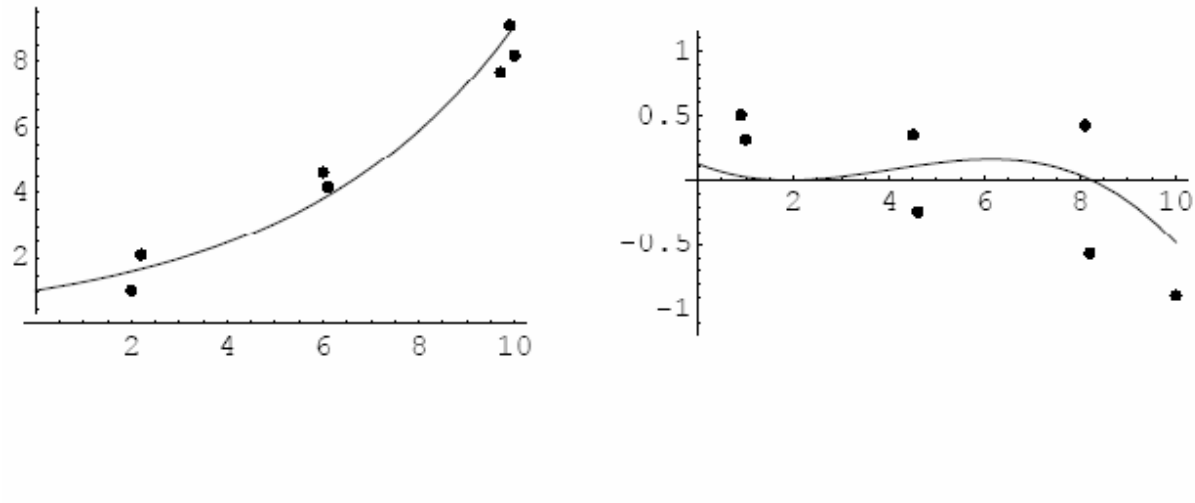
gresfit = Plot[resfit, {var, a, b}];

gressum=Show[gres, gresfit]];

Show[GraphicsArray[{gdatasum, gressum}]];]

$\ln[10] := showfit[data4, data4\,fit, x, 3]$

Sum of squared residuals: 195417



4.1.3. Нелінійне наближення

4.1.3.1. Експонентне зростання. Застосуємо останній оператор до попередньої апроксимації експонентного зростання функції

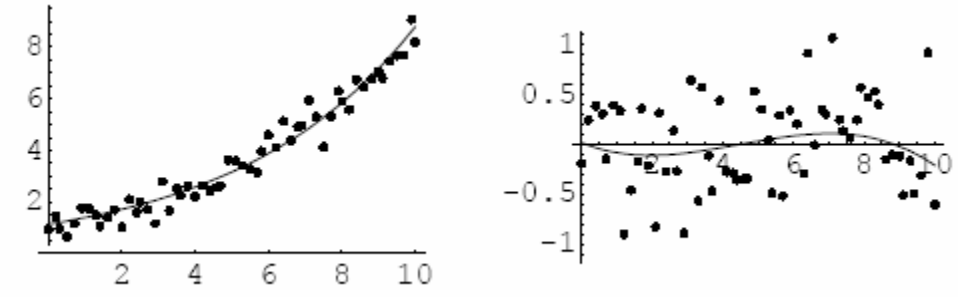
$\ln[11] := << Statistics'NonlinearFit'$

$\ln[12] := data4TrueFit = NonlinearFit[data4, Exp[a + bx], x, \{\{a, 0.3\}, \{b, 0.3\}\}]$

$Out[12] = e^{0.203927\,x + 0.133534}$

$\ln[13] := showfit[data4, data4TrueFit, x, 3]$

Sum of squared residuals: 18.3095



Як видно з останніх графіків результат апроксимації цілком задовільний і зрівнюється з попереднім.

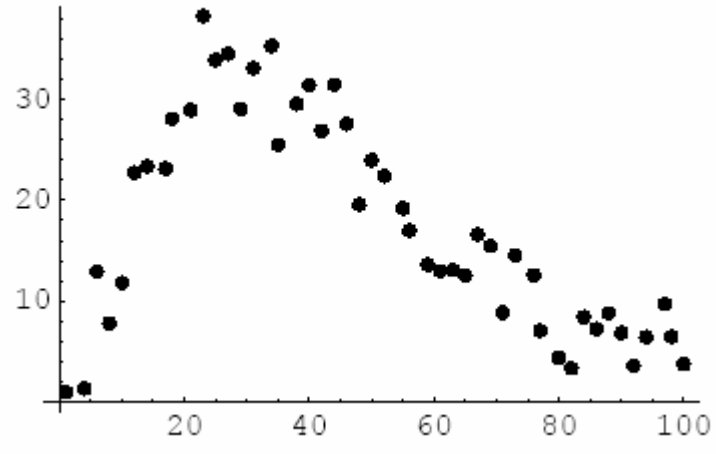
4.1.3.2. Зростання і спадання. Далі розглянемо приклад одночасного зростання і спадання даних. Для цього розглянемо таку функцію

$$f(x) = ax^b Exp[cx] \tag{4.3}$$

яка задовольняє поставлені задачі.

В водимо дані по такому оператору

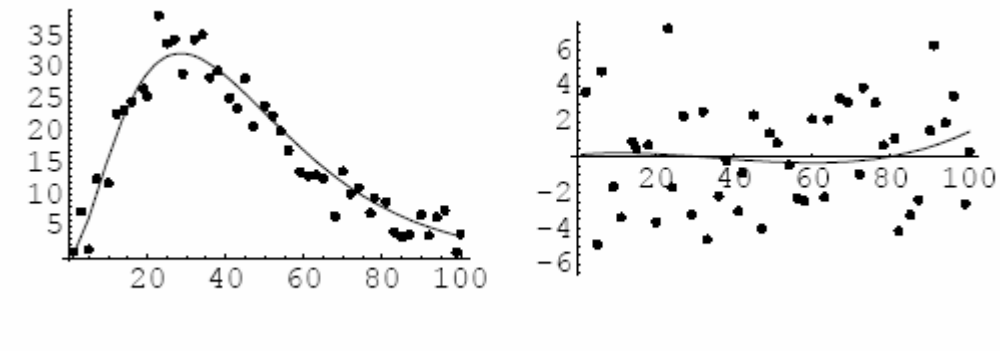
```
In[14]:= data5 = Table[{x,0.3x^2Exp[-0.07x] + 3Random[NormalDistribution[0,1]]},{x,100}];
In[15]:= Do[If[data5[[i,2]] < 1, data5[[i,2]] = 1, Continue],{i,100}]
gdata5 = ListPlot[data5, PlotStyle → PointSize[0.02]];
```



Далі застосуємо метод LM

```
In[17]:= f = ax^bExp[cx]
Out[17] = ae^(cx) x^b
In[18]:= nonLinearFit = NonlinearFit[data5, f, x, {{a,1},{b,1},{c,0}}]
Out[18] = 0.47599e^(-0.0627803 x) x^1.79274
In[19]:= showfit[data5, nonLinearFit, x,3]
```

Sum of squared residuals: 880.285



Тема 5

ЧИСЛОВІ РІШЕННЯ В ЗАДАЧАХ ОПТИМІЗАЦІЇ

5.1. Вступна частина

У задачах оптимізації розглянемо лінійні, класичні й нелінійні задачі оптимізації з їхньою числовою реалізацією.

У задачах лінійної оптимізації, або лінійного програмування функція мети повинна бути мінімізована або максимізована, а зв'язки повинні бути лінійними.

Коди операцій для задач лінійної оптимізації такі:

LinearProgramming[c,m,b]
(∗ мінімізувати величину c.x при зв'язках m.x > 0, x > 0∗)
Minimize[obj, cons, var]
Maximze[obj, cons, var].

5.1.1. Задачі лінійного програмування

Ми можемо сформулювати задачу лінійного програмування або в матрично-векторному вигляді, або як функціональну задачу у вихідних змінних. Відповідні коди команд наведені вище.

5.1.1.1. Матрично-векторний спосіб. Задача формулюється в такий спосіб: необхідно максимізувати величину *C.x* (або мінімізувати - *C.x*), де C - відомий вектор , а x шуканий вектор. Як приклад розглянемо задачу:

• мінімізувати величину

$$(x + 2y) \to Min \qquad (5.1)$$

• 'здержувану' зв'язками

$$\begin{aligned} &-2x - y \ge -2;\\ &x - y \ge -1/2;\\ &-x - 2y \ge 0;\\ &x \ge 0;\\ &y \ge 0. \end{aligned} \qquad (5.2)$$

Для розв'язання запропонованої задачі (5.1), (5.2) скористаємося оператором *LinearProgramming[c, m, b].* Задамо вихідні дані для векторів і матриць

$\ln[23] := Quit[];$
$\ln[1] := c = \{-1,-2\};$
$m = \{\{-2,-1\},\{1,-1\},\{-1,-2\}\};$
$b = \{-2,-1/2,-2\};$

Оптимальний розв’язок перебуває в точці

$\ln[4] := opt = LinearPogram\min g[c, m, b]$

$Out[4] = \{\frac{1}{3}, \frac{5}{6}\}$,

а максимальне значення функції мети дорівнює

$\ln[5] := aim\,\mathrm{Pr}\,ise = -c.opt$

$Out[5] = 2$

5.1.1.2. Візуалізація отриманого розв’язання. Лінійна задача оптимізації (5.1), як класична задача із двома змінними, наочно ілюструє розв’язок на площині.

Для цього побудуємо відповідний розв’язок, використовуючи наступні коди:

$\ln[2] := constr =$

$Plot[\{2-2x, x+1/2, 1-x/2\}, \{x, 0, 1.1\}, AspectRatio \to Automatic, PlotRange \to \{0, 1.1\}];$

$obj = ContourPlot[x+2y, \{x, 0, 1.1\}, \{y, 0, 1.1], Contours \to Range[1/3, 2, 1/3],$

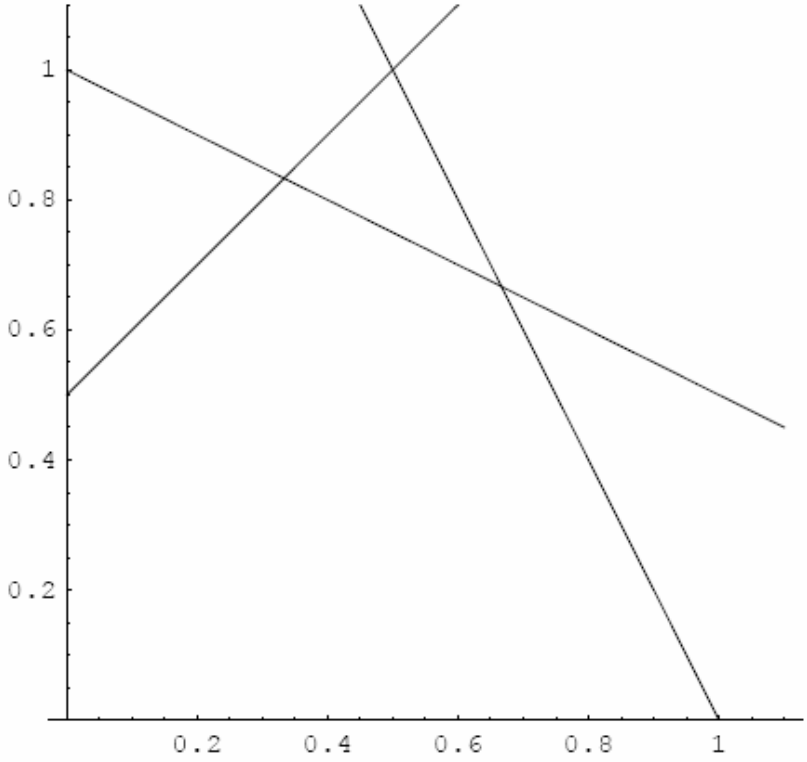
$ContourShading \to False, ContourStyle \to \{Dashing[\{0.02\}]\}];$

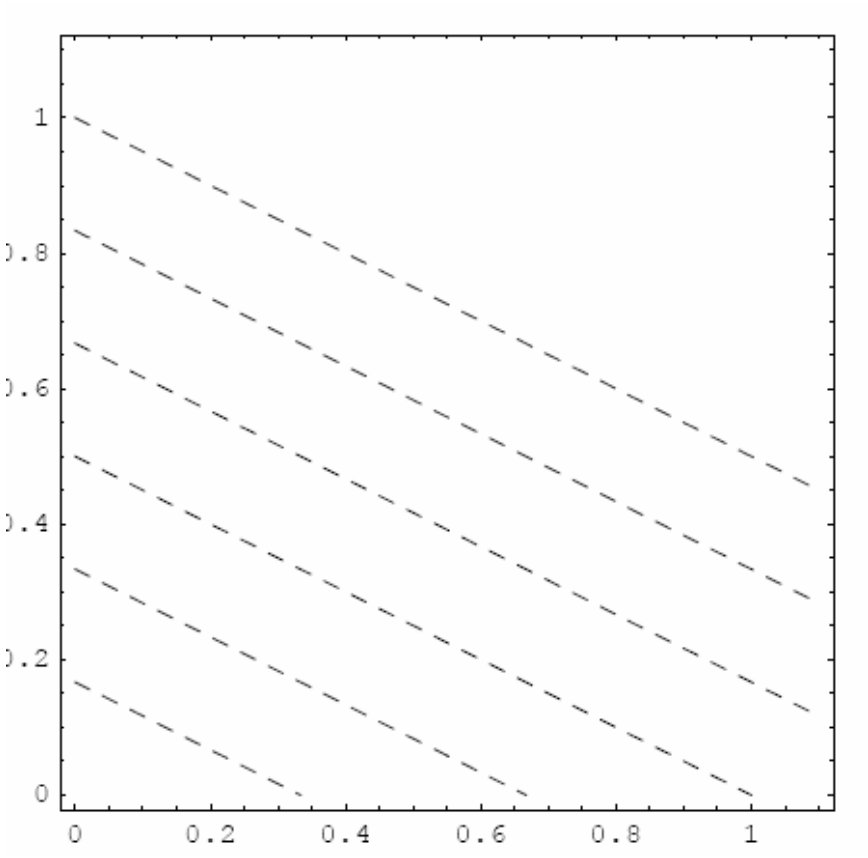
$po\,\mathrm{int}\,s = \{\{0,0\}, \{1,0\}, \{2/3, 2/3\}, \{1/3, 5/6\}, \{0, 1/2\}, \{0,0\}\};$

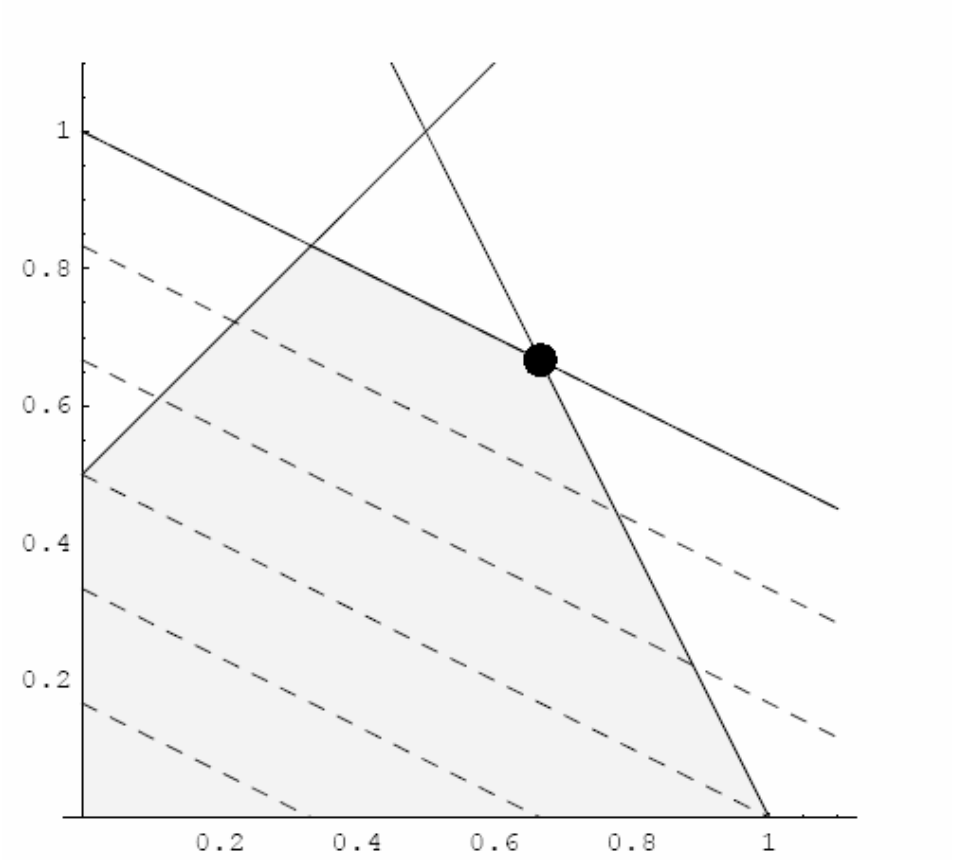
$feasible = Graphics[\{GrayLevel[0.95], Polygon[po\,\mathrm{int}\,s],$

$GrayLevel[0], Line[po\,\mathrm{int}\,s], Po\,\mathrm{int}\,Size[0.04], Po\,\mathrm{int}[\{2/3, 2/3\}]\}];$

$Show[constr, feasible, obj]$







Out[6]:=-Graphics-

Як видно з останнього малюнка, областю допустимих розв’язків є сірий багатокутник. Оптимальний розв’язок це фактично точка на вершині побудованого багатокутника.

На графіку точка оптимуму знаходиться в одній з вершин побудованого багатокутника.

5.1.1.3. Теорія ігор. Приклад оптимального вибору стратегії розвитку.

Розглянемо приклад з теорії ігор [3].

Дві галузі можуть здійснювати капітальні вкладення в чотири об'єкти. Стратегії галузей: і-та стратегія полягає у фінансуванні і-го об'єкта. Зважаючи на особливості внесків і місцеві умови прибутку першої галузі виражаються такою матрицею:

$$A = \begin{pmatrix} 0 & 1 & -1 & 2 \\ -1 & 0 & 3 & 2 \\ 0 & 1 & 2 & -1 \\ 2 & 0 & 0 & 0 \end{pmatrix} \tag{5.3}$$

У цій ситуації величина прибутку першої галузі вважається такою ж як величина збитку для другої галузі. Тому представлена гра може бути матричною грою двох гравців з нульовою сумою.

Для рішення цієї гри до кожного елемента матриці А додамо 2 і одержимо

$$A^* = A + 2 = \begin{pmatrix} 2 & 3 & 1 & 4 \\ 1 & 2 & 5 & 4 \\ 2 & 3 & 4 & 1 \\ 4 & 2 & 2 & 2 \end{pmatrix} \tag{5.4}$$

Задача оптимізації набуває вигляду:

• мінімізувати функцію

$$\sum_i p_i \to \min, \tag{5.5}$$

• при зв'язках

$$A^* \mathbf{p} \geq \mathbf{1} \tag{5.6}$$

або в явному виді

$$\begin{pmatrix} 2 & 3 & 1 & 4 \\ 1 & 2 & 5 & 4 \\ 2 & 3 & 4 & 1 \\ 4 & 2 & 2 & 2 \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} \geq (1 \quad 1 \quad 1 \quad 1). \tag{5.7}$$

Розв’язок задачі лінійного програмування для першого гравця має вигляд

$\ln[1] := c = \{1,1,1,1\};$

$m = \{\{2,3,1,4\},\{1,2,5,4\},\{2,3,4,1\},\{4,2,2,2\}\};$

$b = \{1,1,1,1\};$

$\ln[4] := m // MatrixForm$

$Out[4] // NatrixForm =$

$$\begin{pmatrix} 2 & 3 & 1 & 4 \\ 1 & 2 & 5 & 4 \\ 2 & 3 & 4 & 1 \\ 4 & 2 & 2 & 2 \end{pmatrix}$$

Оптимальне рішення перебуває в "точці"

$\ln[5] := opt = Linear\,\mathrm{Pr}\,ogram\,\min\, g[c, Transpose[m], b]$

$Out[5] = \{\frac{8}{69}, \frac{1}{23}, \frac{7}{69}, \frac{3}{23}\}$

Сума "гри" дорівнює

$\ln[6] := priseSum = Sum[opt[[i]], \{i,4\}]$

$Out[6] = \frac{9}{23}$

$\ln[7] := v = 1/\ priseSum$

$Out[7] = \frac{23}{9}$.

Оптимальна вартість дорівнює

$\ln[8] := aim\,\mathrm{Pr}\,ise = c.opt$

$Out[8] = \frac{9}{23}$.

Оскільки задача формулюється як ігрова, то змішані стратегії для першого гравця рівні:

$\ln[9] := strategy = vopt$

$Out[9] = \{\frac{8}{27}, \frac{1}{9}, \frac{7}{27}, \frac{1}{3}\}$.

Ціна вихідної гри дорівнює

$\ln[10] := valueGame = v - 2$

$Out[10] = \frac{5}{9}$.

Знаходимо оптимальні змішані стратегії першого "гравця"

$\ln[11] := x = vopt$

$Out[11] = \{\frac{8}{27}, \frac{1}{9}, \frac{7}{27}, \frac{1}{3}\}$,

які можна виразити в частках розподілу вкладення інвестицій по об'єктах, наприклад, числові значення для перших двох наведені нижче:

$\ln[12] := N[x[[1]]]$

$Out[12] = 0.296296$

$\ln[13] := N[x[[3]]]$

$Out[13] = 0.259259$

Отримані оптимальні стратегії перший гравець може використати так: всі свої капіталовкладення може розподіляти по об'єктах у частках отриманого розв'язку.

Наприклад, на перший об'єкт виділити 29,63 %, а на третій 25,92 % капіталовкладень і т.д., при ціні гри 5/9 (~0,5) грошової одиниці.

Далі представимо рішення ігрової задачі у вихідних змінних задачі.

Для першого гравця знаходимо мінімум

$$\sum_i p_i \to \min \quad , \tag{5.8}$$

а рішення оптимізаційної задачі у вихідних змінних знаходимо оператором *ConstrainedMin[...]*

$\ln[14] := Minimize[p1 + p2 + p3 + p4, \{2p1 + p2 + 2p3 + 4p4 \geq 1, 3p1 + 2p2 + 3p3 + 2p4 \geq 1,$
$p1 + 5p2 + 4p3 + 2p4 \geq 1, 4p1 + 4p2 + p3 + 2p4 \geq 1\}, \{p1, p2, p3, p4\}]$.

$Out[14] = \{\frac{9}{23}, \{p1 \to \frac{8}{69}, p2 \to \frac{1}{23}, p3 \to \frac{7}{69}, p4 \to \frac{3}{23}\}\}$

Для другого гравця знаходимо максимум

$$\sum_i q_i \to \max , \tag{5.9}$$

$\ln[15] := sol = Maximize[q1 + q2 + q3 + q4, \{2q1 + 3q2 + q3 + 4q4 \leq 1, q1 + 2q2 + 5q3 + 4q4 \leq 1,$
$2q1 + 3q2 + 4q3 + q4 \leq 1, 4q1 + 2q2 + 2q3 + 2q4 \leq 1\}, \{q1, q2, q3, q4\}]$.

$Out[15] = \{\frac{9}{23}, \{q1 \to \frac{5}{46}, q2 \to \frac{7}{46}, q3 \to \frac{3}{46}, q4 \to \frac{3}{46}\}\}$

Відповідні стратегії для другого гравця рівні

$\ln[16] := strY = vTable[sol[[2, i, 2]], \{i, 4\}]$

$Out[16] = \{\frac{5}{18}, \frac{7}{18}, \frac{1}{6}, \frac{1}{6}\}$.

Отримані оптимальні стратегії другий гравець може використати так: всі свої капіталовкладення може розподіляти по об'єктах у частках отриманого розв'язку *strY*.

$\ln[17] := N[strY[[1]]]$

$Out[17] = 0.277778$

$\ln[18] := N[strY[[2]]]$

$Out[18] = 0.388889$

Наприклад на перший об'єкт виділити 27,78 %, а на другий 38,9 % капіталовкладень при ціні гри 5/9 (~0,5) грошової одиниці.

5.1.1.4. Формулювання у вихідних змінних. Отже представимо рішення попередньої задачі (5.1) у вихідних змінних

$\ln[20] := Quit[];$

$\ln[1] := optim = Maximize[x + 2y, \{2x + y \le 2, x - y \ge -1/2, x + 2y \le 2\}, \{x, y\}]$

$Out[1] := \{2, \{x \to \frac{1}{3}, y \to \frac{5}{6}\}\}.$

Далі представимо рішення ігрової задачі вибору оптимальних стратегій завоювання ринку для першого гравця в задачах (5.3) - (5.7).

$\ln[22] := Quit[];$

$$\ln[1] := A = \begin{pmatrix} 0 & 1 & -1 & 2 \\ -1 & 0 & 3 & 2 \\ 0 & 1 & 2 & -1 \\ 2 & 0 & 0 & 0 \end{pmatrix};$$

$$\ln[2] := \mathbf{r}p \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix}$$

$\ln[3] := \mathbf{r}b = \{1, \quad 1, \quad 1, \quad 1\};$

$\ln[4] := system = Thread[((A+2)^T . \mathbf{r}p \,//\, Flatten) \ge \mathbf{r}b]$

$Out[4] = \{2p_1 + p_2 + 2p_3 + 4p_4 \ge 1, 3p_1 + 2p_2 + 3p_3 + 2p_4 \ge 1,$

$p_1 + 5p_2 + 4p_3 + 2p_4 \ge 1, 4p_1 + 4p_2 + p_3 + 2p_4 \ge 1\}$

$\ln[5] := solMin = Minimize[Sum[p_i, \{i,1,4\}], system, Table[p_i, \{i,1,4\}]] // Flatten$

$Out[5] = \{\frac{9}{23}, p_1 \to \frac{8}{69}, p_2 \to \frac{1}{23}, p_3 \to \frac{7}{69}, p_4 \to \frac{3}{23}\}.$

Порівнюючи два рішення, отримані різними операторами з’ясовуємо, що "точка" оптимальних стратегій для першого гравця находиться в =, знайдена оператором *LinearProgramming[...]* і останнє розв’язання знайдене оператором *Minimize[...]*, збігаються.

Однак в останньому випадку знаходимо значення мінімуму в точці оптимуму, чого немає в попередньому розв’язанні.

5.1.2. Задача класичної оптимізації

Під задачею класичної оптимізації розуміється задача визначення *MinMax* для функції однієї або декількох змінних. Послідовно розглянемо розв’язання для функції однієї, а потім і декількох змінних.

5.1.2.1. Функція однієї змінної. Розглянемо найпростіший приклад: поліном п'ятого степеня

```
In[7]:= Quit[];
In[1]:= f = 12x^5 - 45x^4 + 40x^3 + 5;
```

Знайдемо критичні точки

```
In[2]:= erit = Select[Union[Solve[D[f, x] = 0]], FreeQ[#, Complex]&]
Out[2]= {{x → 0}, {x → 1}, {x → 2}}
```

Далі перевіримо значення других похідних у критичних точках

```
In[3]:= secondDer = D[f, x, x] /. crit
Out[3]= {0, -60, 240}
```

і відповідно, третя похідна в точці x = 0 дорівнює

```
In[4]:= D[f, x, x, x] /. x → 0
Out[4]= 240
```

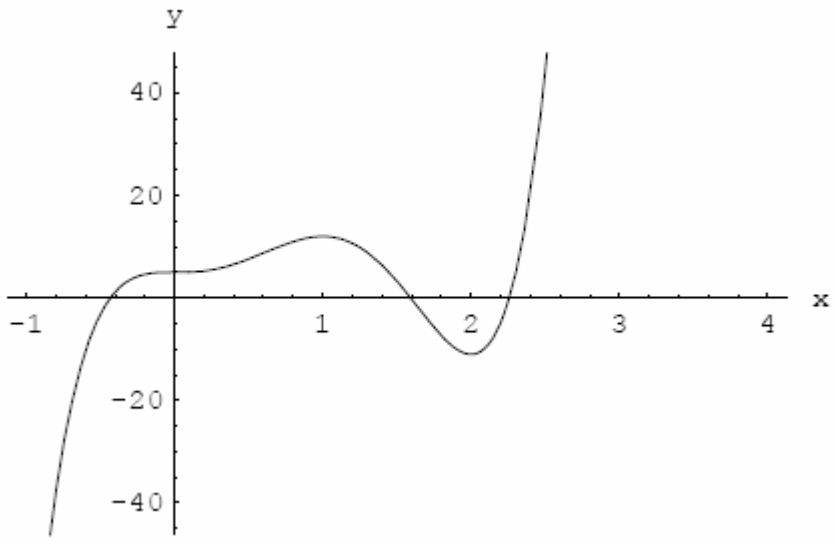
Знаходимо координати критичних крапок

```
In[5]:= point s = {x, f} /. crit
```

$$Out[5] = \begin{pmatrix} 0 & 5 \\ 1 & 12 \\ 2 & -11 \end{pmatrix}$$

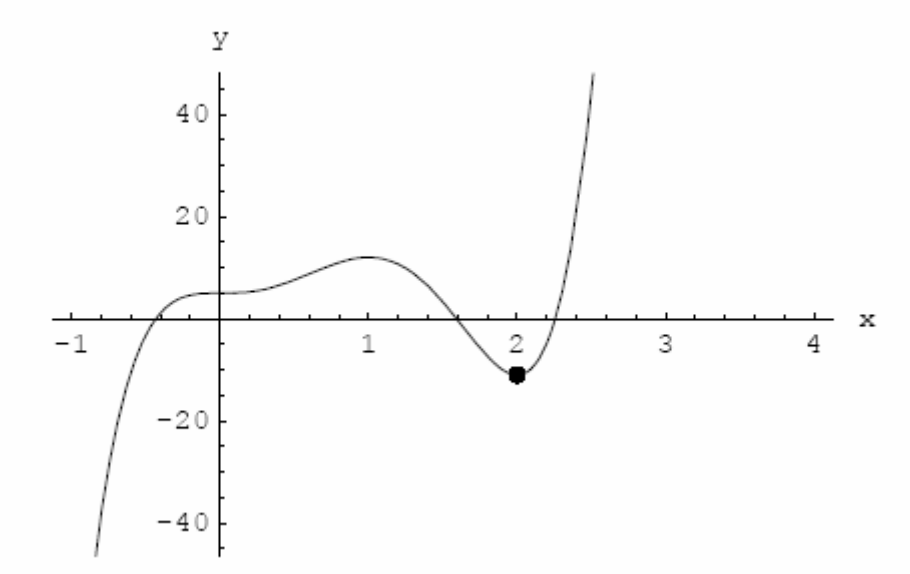
Потім будуємо відповідний графік вихідної функції

```
In[6]:= graphicFunction = Plot[f, {x, -1, 4}, AxesLabel → {"x", "y"}];
```



а також спільний графік вихідної функції і критичних точок

```
In[7]:= graphicPoint s = Graphics[{PointSize[0.02], Point[#]}] & /@ point s;
Show[graphicFunction, graphicPoint s];
```



Графічне зображення наочно ілюструє точки локального максимуму і мінімуму.

5.1.2.2. Функція багатьох змінних. Для знаходження критичних точок функції багатьох змінних, ми повинні знайти множину часток похідних функції, потім прирівняти їх до нуля і розв’язати отриману систему рівнянь. Як відомо серед множини критичних точок перебувають точки максимуму, мінімуму і сідлоподібні точки.

Суттєвою умовою мінімуму (максимуму) є позитивне (негативне) значення Гесіана, обчисленого в критичній точці.

Набір кодів для реалізації процедури знаходження критичних точок функції багатьох змінних такий:

```
grad [f_,x] := D[f,#] & / @ x;

crit [f_,x_] :=
   Select [Union [Solve [grad [f, x]=0, x]], FreeQ [#, Complex] &];

hess [f_, x_] :=Outer [D, qrad [f, x], x];

mini [f_, x_] :=
   Select [crit [f, x], And @@ Positive [Eigenvalues [hess [f, x]/.#]] &];

maxi [f_, x_] :=
   Select [crit [f, x], And @@ Negative [Eigenvalues [hess [f, x]/.#]] &];

Sadd [f_, x_] := Complement [crit [f, x], mini [f,x], maxi [f, x]];
```

Розглянемо приклад застосування описаних вище кодів.

$\ln[7] := f = x\text{^}3 + y\text{^}3 + 2x\text{^}2 + 4y\text{^}2 + 6$

$Out[7] = x^3 + 2x^2 + y^3 + 4y^2 + 6$

$\ln[8] := criticalPo\,\text{int}\,s = crit[f, \{x, y\}]$

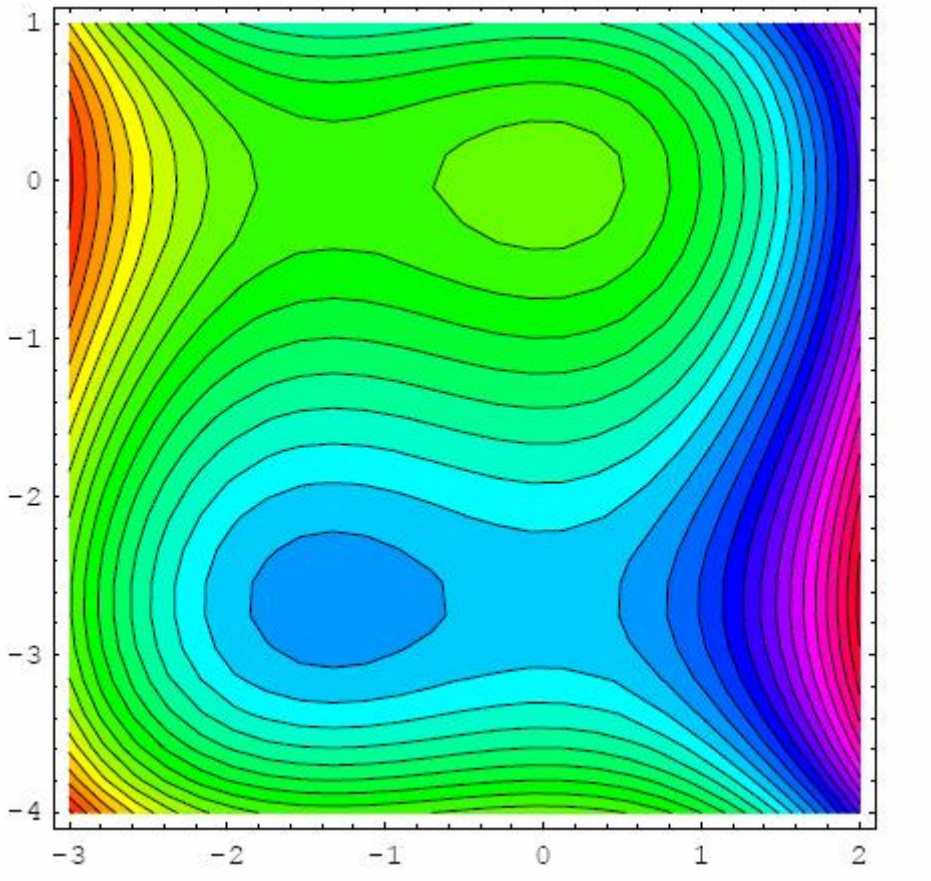
$Out[8] = \{\{x \to -\frac{4}{3}, y \to -\frac{8}{3}\}, \{x \to -\frac{4}{3}, y \to 0\}, \{x \to 0, y \to 0\}\}$

Для наочності побудуємо контурний графік функції разом із критичними точками

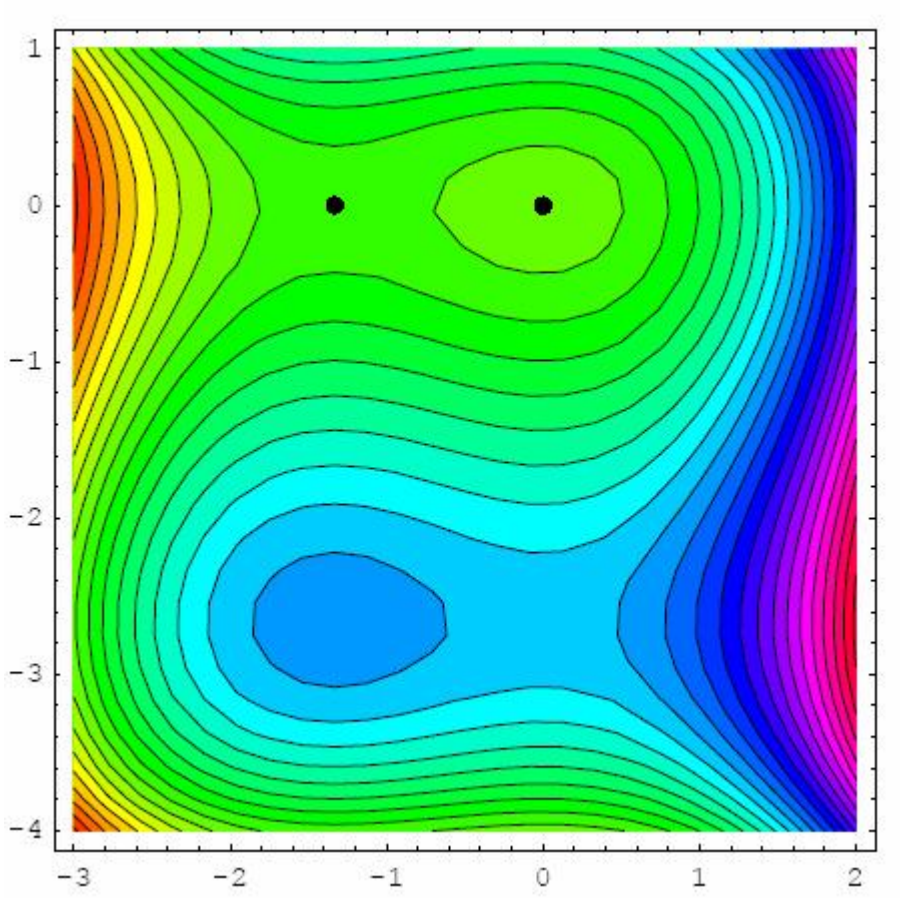
$\ln[12] := cont = ContourPlot[f, \{x, -3, 2\}, \{y, -4, 1\}, Contours \to 30, ColorFunction \to Hue];$

$po\,\text{int}\,s = Graphics[\{Po\,\text{int}\,Size[0.02], Po\,\text{int}[\#]\}]\,\&\,/@(\{x, y\}\,/.criticalPo\,\text{int}\,s);$

$Show[cont, po\,\text{int}\,s];$



які на контурному графіку представлені окремо візуальними точками.



Величина функції в критичних точках дорівнює

$\ln[15] := f /. criticalPoint s$

$Out[15] = \{\frac{50}{3}, \frac{194}{27}, \frac{418}{27}, 6\}$

Далі ми можемо записати таблицю, яка має критичні крапки, значення функції Гесіана і власних значень:

$\ln[16] := pr = Table[\{x, y, f, Eigenvalues[hess[f, \{x, y\}]]\}] /. N[criticalPoint s]];$

$PaddedForm[TableForm[pr, TableHeadings \to \{None, \{"x", "y", "f", "Hessian"\}\}, TableSpacing \to \{0,1\}, TableDepth]]$

$Out[17] // PaddedForm =$

x	y	f	Hessian
-1.333333	-2.666667	16.666670	{-4.000000, -8.000000}
-1.333333	0.000000	7.185185	{-4.000000, 8.000000}
0.000000	-2.666667	15.481480	{4.000000, -8.000000}
0.000000	0.000000	6.000000	{4.000000, 8.000000}

Отже визначимо координати критичних точок:

$\ln[18] := \min i[f, \{x, y\}]$

$Out[18] = \{\{x \to 0, y \to 0\}\}$.

Точка максимуму

$\ln[19] := \max i[\, f, \{x, y\}]$

$Out[19] = \{\{x \to -\frac{4}{3}, y \to -\frac{8}{3}\}\}$.

Сідлова точка

$\ln[20] := sadd[\, f, \{x, y\}]$

$Out[20] = \{\{x \to -\frac{4}{3}, y \to 0\}, \{x \to 0, y \to -\frac{8}{3}\}\}$.

5.1.2.3. Функція багатьох змінних з одним зв'язком. Розглянемо знаходження оптимальних значень функції багатьох змінних при наявності зв'язків. Зв'язок являє собою функцію у вигляді

$$g\{x_1, x_2, ...x_n\} = 0 \qquad (5.10)$$

Як відомо, така задача вирішується методом Лагранжа.

$\ln[21] := f = x\text{^}3 - xy + y\text{^}2;$

$g = x\text{^}2 + y\text{^}2 - 1;$

$\ln[23] := lagrangeFunction = f + ag;$

$\text{var}\, s = \{x, y, a\};$

$\ln[25] := derivetives = D[lagrangeFunction, \#]\ \&\ /@\ \text{var}\, s$

$Out[25] = \{3x^2 + 2ax - y, -x + 2ay + 2y, x^2 + y^2 - 1\}$

$\ln[26] := point\ sCrit = Select[Union[Solve[derivetives = 0, \text{var}\, s]], FreeQ[\#, Complex]\&]\ //\ N$

$Out[26] = \{\{a \to 1.56968, y \to -0.90995, x \to -0.981591\}, \{a \to -1.15599, y \to 0.954623,$
$x \to -0.297816\}, \{a \to -0.519366, y \to 0.72093, x \to 0.693008\}, \{a \to -1.56586, y \to -0.662151,$
$x \to 0.74937\}, \{a \to -1.16423 + 0.341111i, y \to -1.07787 - 0.338806i, x \to 0.585181 -$
$0.624061i\}, \{a \to -1.16423 - 0.341111i, y \to -1.07787 + 0.338806i, x \to 0.585181 + 0.624061i\}\}$

Покажемо отримані рішення в табличній формі

$\ln[27] := PaddedForm[$

$TableForm[\{x, y, f\}\ /.\ point\ sCrit, TableHeadings \to \{None\{"x", "y", "f"\}\}], \{6,6\}]$

$Out[27]\ //\ PaddedForm =$

x	y	f
-0.981591	-0.190995	-1.096780
-0.297816	0.954623	1.169190
0.693008	0.720930	0.352954
0.749370	-0.662151	1.355450
0.585181 – 0.624061i	-1.077870 – 0.338806i	1.405890 – 0.142079i
0.585181 + 0.624061i	-1.077870 + 0.338806i	1.405890 + 0.142079i

5.1.2.4. Функція багатьох змінних із множиною зв'язків. Розв’яжемо задачу знаходження оптимуму функції багатьох змінних при наявності декількох зв'язків. Нижче наведений модуль, за допомогою якого розв’язуємо дану задачу

```
lagrangeOptimizing[f_,c_List, x_List] :=Module
  {n = Length[x], m = Length[c], p, xl, eq, sol}
  (*локальні змінні користувальницького пакета*),
   l = Array[lambda, m]; {*множники Лагранжа*}
   p = f – Lc; (*функція Лагранжа*)
  x1 = Join[x, l] (*змінні*)
  eq = Map[D[p, #]&, xl]?0; (*рівняння необхідних умов*)
  sol = Select[Solve[eq, xl], FreeQ[#, Complex]&]//Simplify;
  Flatten[{x, f}]./sol//Simplify
      (*оптимальні точки й значення функції в точках оптимуму*)
```

Вирішимо задачу із двома зв'язками, але розв’язок представимо в символьному вигляді

$$\ln[29]:= f = x\text{^}2 + y\text{^}2 + z\text{^}2 + w\text{^}2;$$

$$g = x + 3y - z + w + a;$$

$$q = 2x - y + z + 2w + b$$

$$\ln[32]:= optimum = lagrangeOptimizing[\,f\,,\{g,q\},\{x,y,z,w\}\,]\,//\,Flatten$$

$$Out[32] = \{\frac{1}{60}(-5a-12b), \frac{1}{20}(2b-5a), \frac{1}{60}(5a-6b), \frac{1}{60}(-5a-12b), \frac{1}{60}(5a^2+6b^2)\}$$

Тут перші чотири значення є координати точки оптимуму, а останнє значення є оптимальне значення функції.

Оптимум функції дорівнює

$$\ln[33]:= optimum[[5]]$$

$$Out[33] = \frac{1}{60}(5a^2+6b^2).$$

5.1.3. Нелінійне програмування

Під нелінійним програмуванням розуміється знаходження *MinMax* без залучення знаходження похідної функції. Цей метод зручний у випадку, коли функціональні залежності задані неявно або рішення системи визначальних рівнянь знайти не вдається.

5.1.3.1. Функція однієї змінної. Код операторів для знаходження мінімуму функції однієї змінної представлені нижче.

```
FindMinimum [f, {x, x0}]
```

FindMinimum [f, {x, x0, xmin, xmax}]

Максимум функції перебуває, якщо покласти замість *f- > -f.*

Розглянемо приклад.

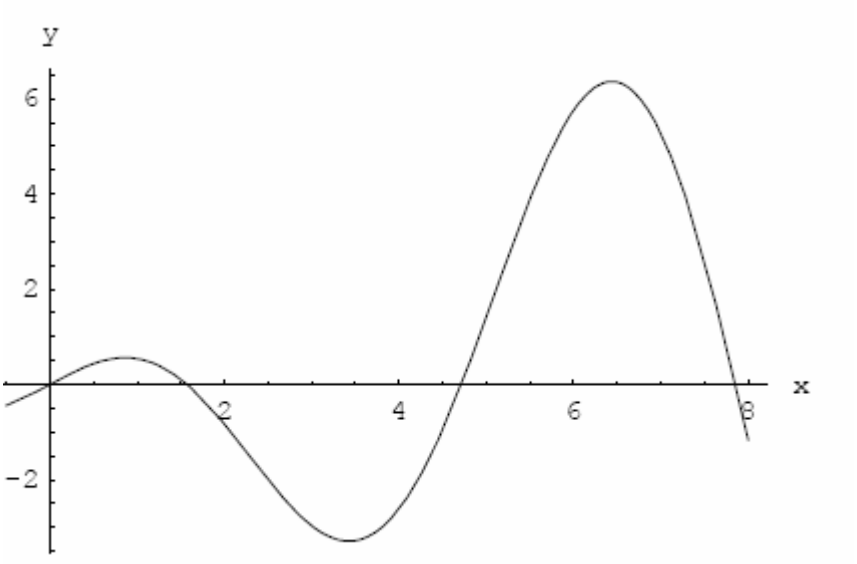
```
In[34]:= Quit[];
```

Нехай функція однієї змінної дорівнює

```
In[1]:= f = xCos[x];
```

Графік функції *f(x)* зображено нижче.

```
In[2]:= grn = Plot[f, {x, –0.5, 8}, AxesLabel → {"x", "y"}];
```



Очевидно, що функція *x Cos[x]* має локальні максимуми s мінімуми, видимі на графіку. Тому скористаємось оператором *FindMinimum[]* для знаходження локального мінімуму на замкнутому інтервалі [0,5]

```
In[3]:= FindMinimum[f, {x, 5}]
Out[3]= {–3.28837, {3.42562}}
```

Як видно з останнього результату мінімум функції дорівнює 3.28837 і досягається він у точці x = 3.42562. Далі знайдемо точку локального максимуму

```
In[14]:= FindMinimum[–f, {x, 8}]
Out[14]= {–6.361, {x → 6.4373}}
```

5.1.3.2. Функція декількох змінних. Для функції декількох змінних ми будемо використовувати два методи: градієнтний метод сполучення або метод головного напрямку.

Градієнтний метод

У випадку градієнтного методу сполучення ми повинні дати стартове значення кожної зі змінних. Оператори для цього методу мають вигляд

FindMinimum[f, {x, x0}, {y, y0}, …]

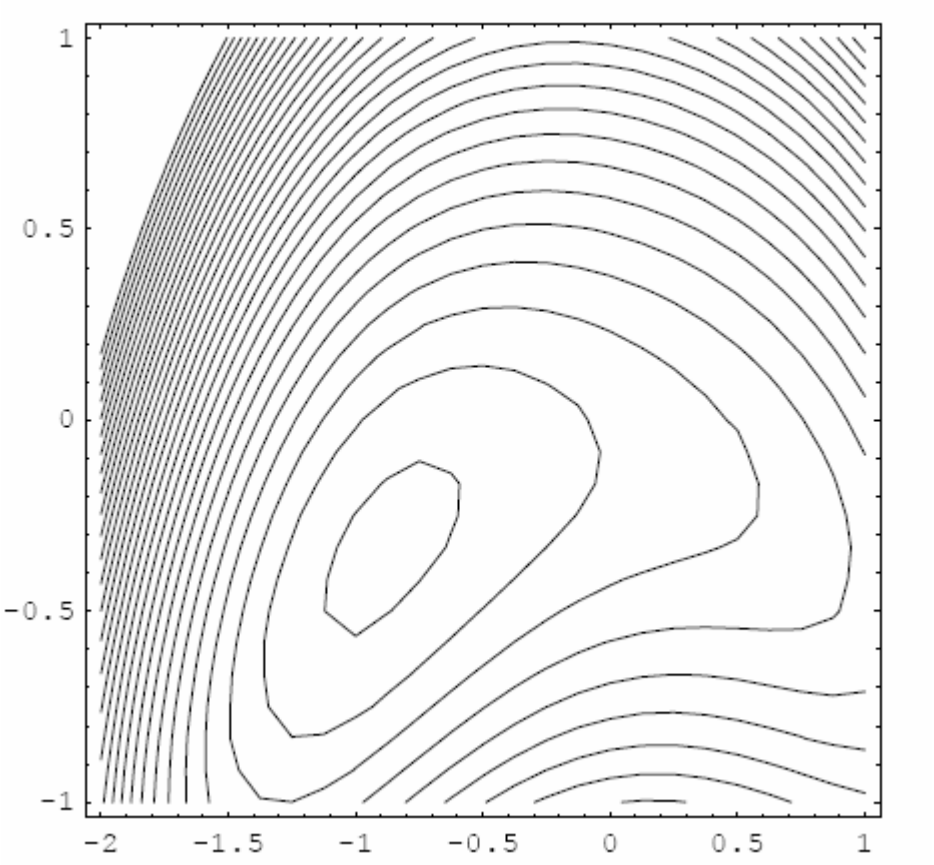
FindMinimum[f, {x, x0, xmin, xmax}, {y, y0, ymin, ymax}, …]

Розглянемо приклад знаходження мінімуму такої функції

```
In[5]:= Quit[];
In[1]:= f = x^4 + 3x^2y + 5y^2 + x + y .
```

Далі розглянемо контурний графік цієї функції

```
In[2]:= cont = ContourPlot[ f ,{x,–2,1},{y,–1,1}, Contours → 30, ContourShading → False,
ColorFunction]
```



Знайдемо рішення задачі зі стартової точки (0.5, 0.4)

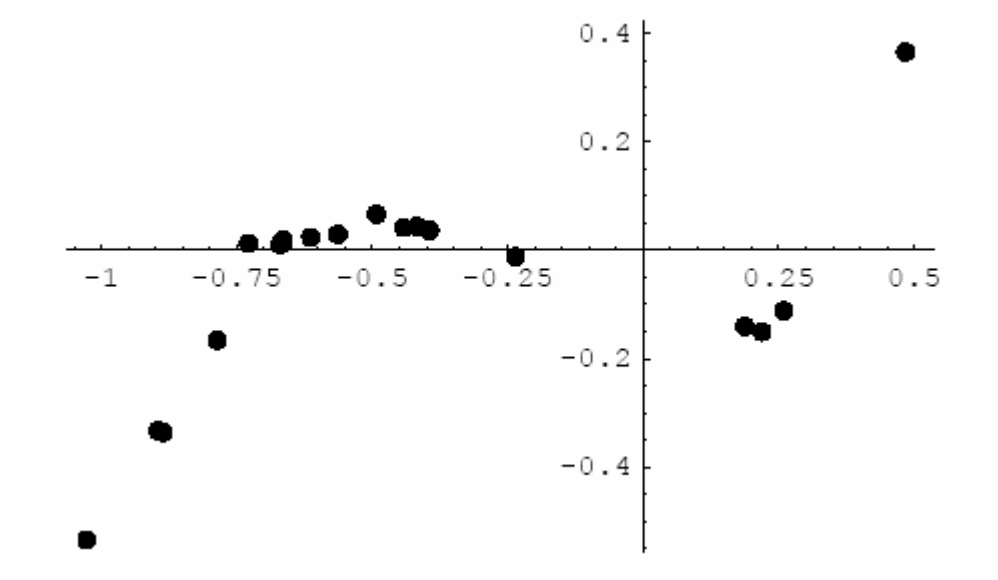
```
In[3]:= sol = FindMinimum[ f ,{x,0.5},{y,0.4}]
Out[3]= {–0.832579,{x → –0.886324, y → –0.335671}}
```

Тепер розглянемо візуалізацію послідовних кроків ітерацій до оптимального розв’язання:

```
In[8]:= Quit[];
In[1]:= fune[x_, y_] := x^4 + 3x^2y + 5y^2 + x + y;
In[7]:= v = {};
FindMinimum[AppendTo[v,{x, y}]; func[x, y],{x,0.5},{y,0.4}, EvaluationMonitor → True]
```

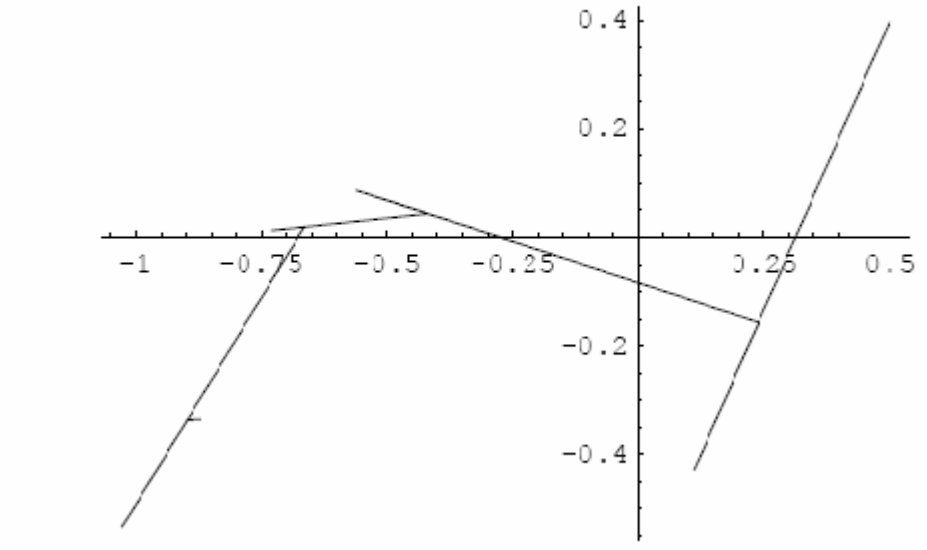
$Out[8] = \{-0.832579, \{x \to -0.886324, y \to -0.335671\}\}$

$\ln[9] := point\ ListPlot[v, PlotStyle \to Point\ Size[0.02], PlotRange \to All, AspectRatio \to Automatic;]$



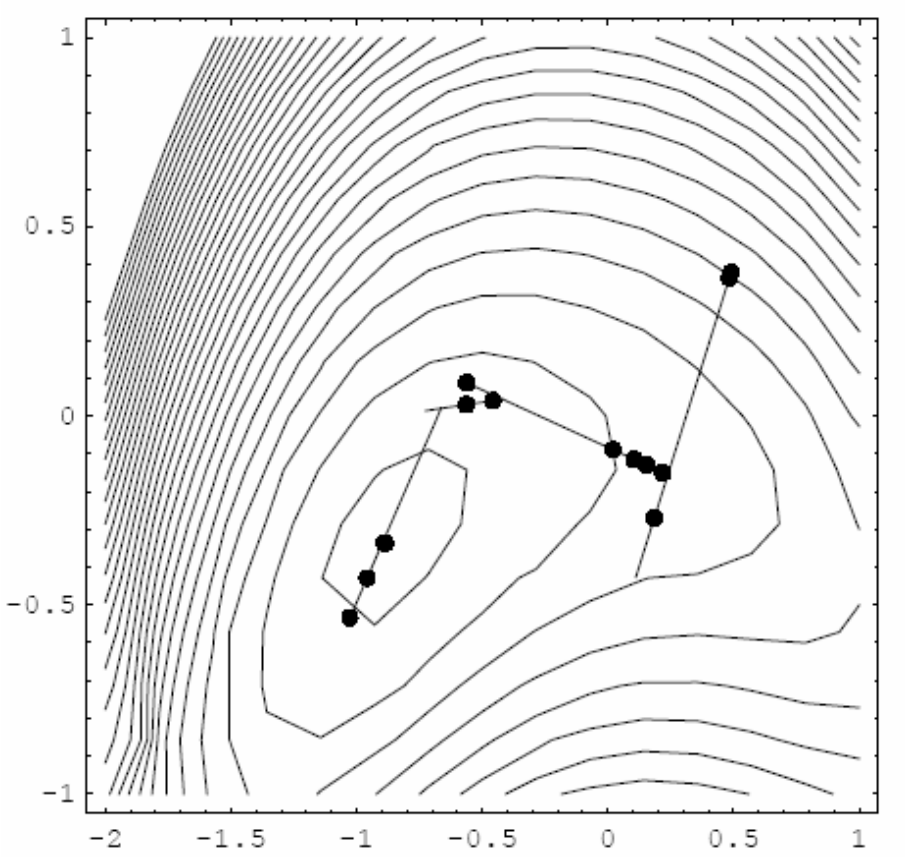
Далі з'єднаємо отримані точки відрізками прямих

$iter = ListPlot[v, PlotJoined \to True, PlotRange \to All, AspectRatio \to Automatic]$



тепер розташуємо всі графіки на контурі

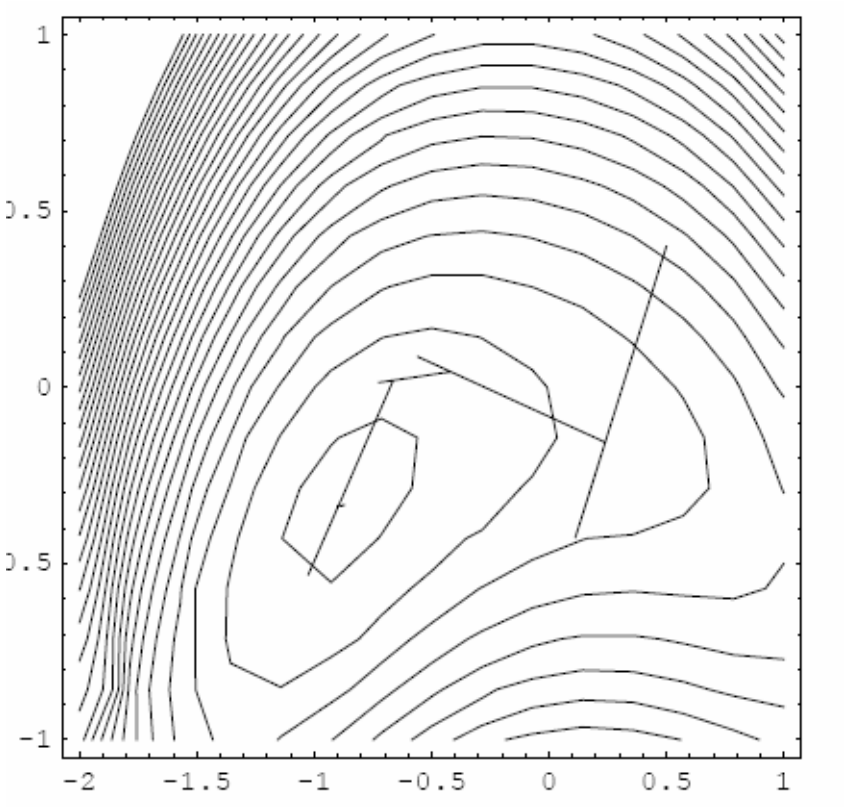
$Show[cont, iter, point\ s]$



- Graphics -

розмістимо тільки відрізки на контурі

Show[*cont*, *iter*]



Метод головного напрямку

Розглянемо метод головного напрямку, що не використовує функції градієнта при знаходженні напрямку до оптимальної точки. Оператори для рішення такі ж як і у попередній задачі. На відміну від попереднього прикладу даємо початкові значення для кожної змінної.

Розглянемо приклад наведений у пункті градієнтного методу для функції

$$z = x^4 + 3x^2y + 5y^2 + x + y; \qquad (5.11)$$

код уведення для якої наведений нижче.

```
In[1]:= z = x^4 + 3x^2y + 5y^2 + x + y .
```

Очевидно, що оператор для визначення локального мінімуму приводить рішення до вигляду

```
In[2]:= FindMinimum[z,{x,0.5,0},{y,0.5,0}]
Out[2]= {-0.832579,{x → -0.886324, y → -0.335671}}
```

Далі розглянемо візуалізацію отриманого розв'язку, введемо допоміжні модулі

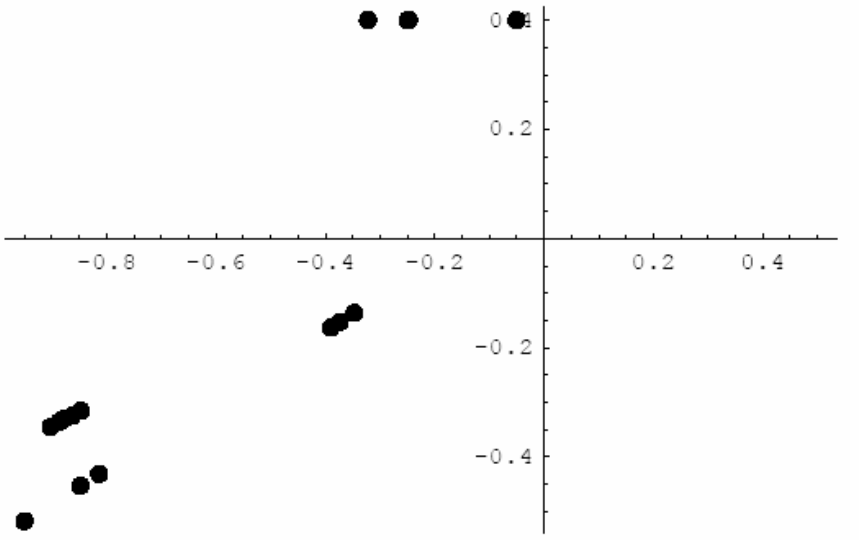
```
In[3]:= f1[x_, y_] := z;
dfx[x_, y_] := Evaluate[D[z, x]];
dfy[x_, y_] := Evaluate[D[z, y]];
```

застосуємо процедуру формування точок наближення до мінімуму функції.

```
In[6]:= v = {};
FindMinimum[AppendTo[v,[x, y]]; f1[x, y],{x,0.5,0},{y,0.4,0}].
Out[7]= {-0.832579,{x → -0.886314, y → -0.335667}}
```

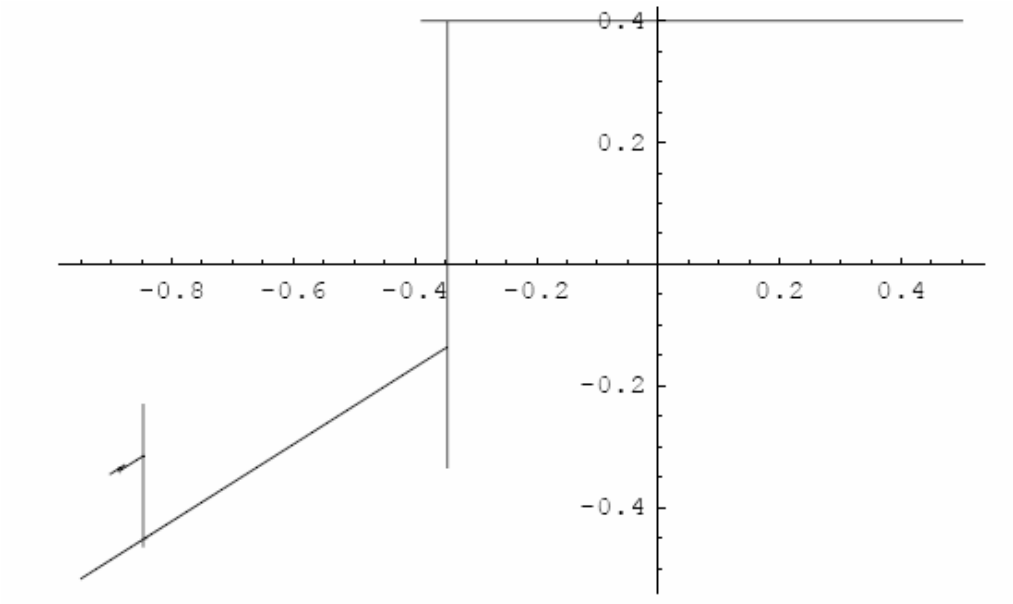
Візуалізація точок наближення до мінімуму функції наведена на площині

```
In[8]:= point sNew = ListPlot[v, PlotStyle → Point Size[0.02], PlotRange → All, AspectRatio
→ Automatic]
```



Графік шляхів наближення має вигляд

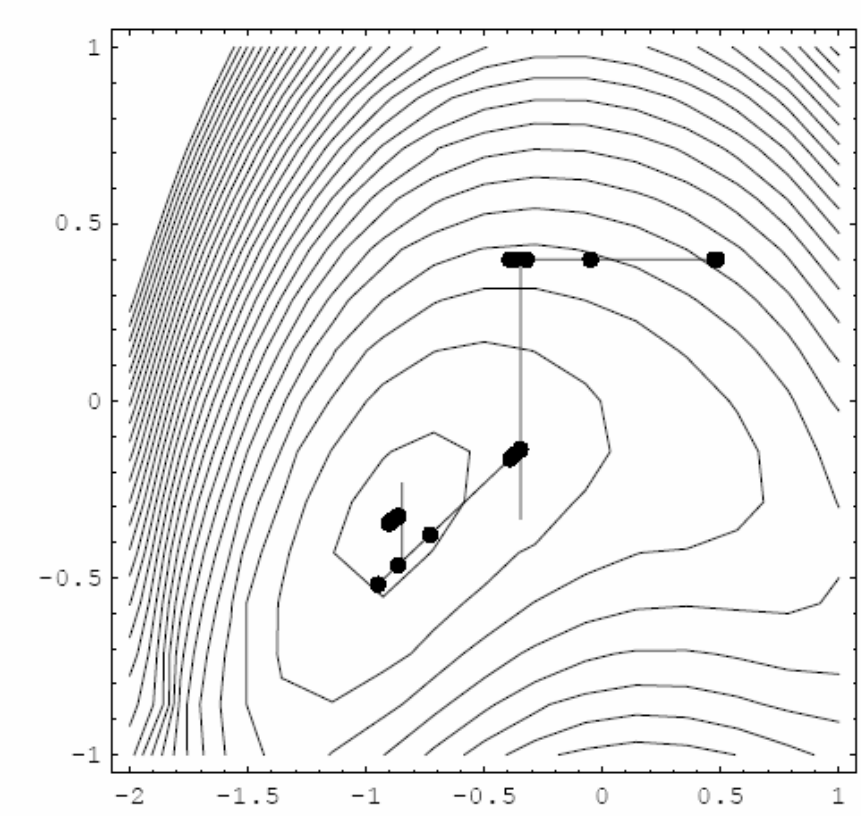
$iterNew = ListPlot[v, PlotJoined \rightarrow True, PlotRange \rightarrow All, AspectRatio \rightarrow Automatic]$



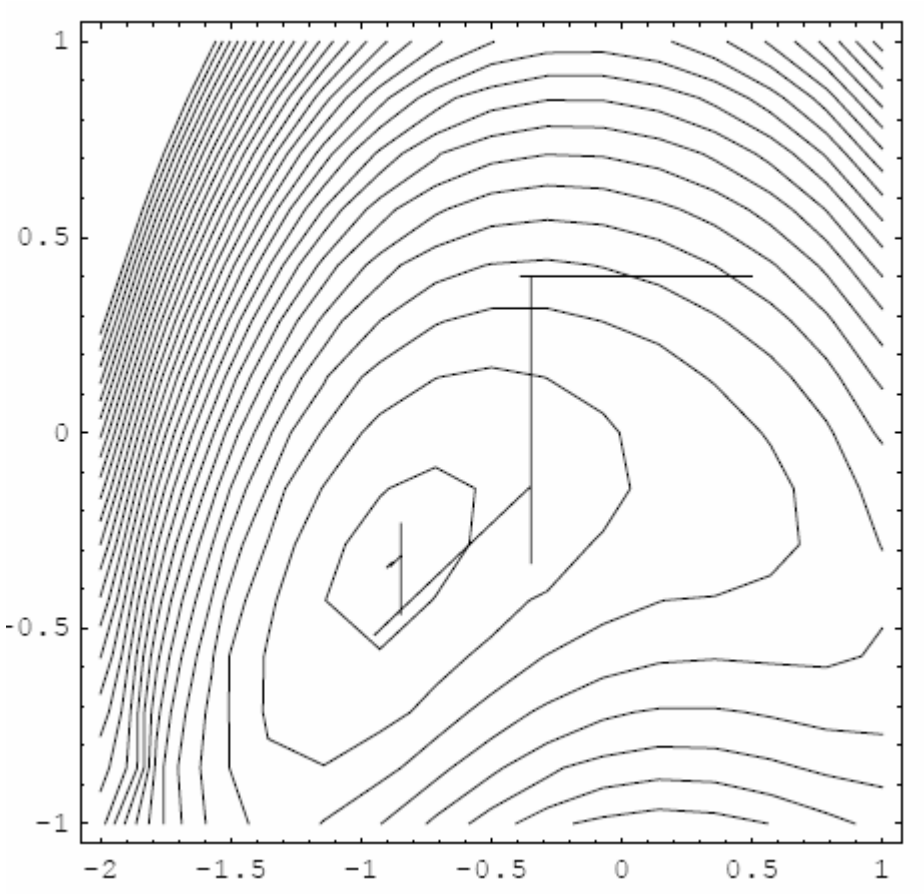
- Graphics -

Які потім з’єднуємо із точками наближення на спільному графіку.

$Show[cont, point sNew, iterNew]$



Остаточний графік функції і шлях "руху" до точки локального мінімуму зображений нижче. *Show*[*cont*, *iterNew*]



- Graphics -

Необхідно зазначити, що представлений тут алгоритм і його програмний модуль розроблений у ранніх версіях системи *Mathematica* й успішно працює в її ранніх версіях (починаючи в v.2.2).

Однак в останніх версіях *Mathematica* (v.5.2, v.6.0) для рішення задач визначення локальних екстремумів розроблений зручний пакет, завантаження якого може бути здійснено наступним оператором.

ln[1] := *Needs*["*Optimization*'*Uneonstrained* Pr*oblems*'"].

Далі розглянемо кілька прикладів застосування убудованого пакета

Optimization'*Uneonstrained* Pr*oblems*'

для рішення задач оптимізації.

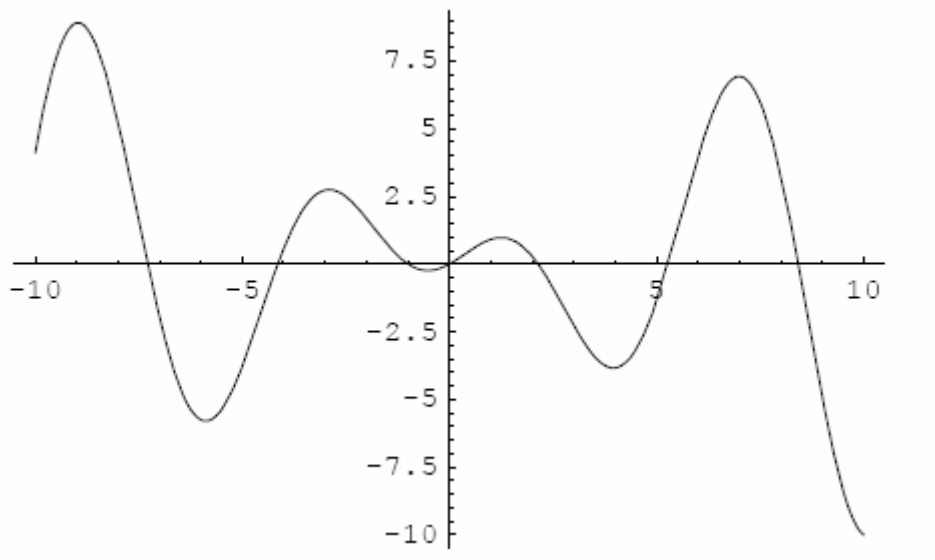
Mathematica має ряд команд для рішення задач оптимізації з не утримуючими зв'язками (див. наприклад команди ***FindMinimum and FindMaximum***) і для рішення задач оптимізації для нелінійних рівнянь (оператор ***FindRoot***), рішення задач нелінійного програмування (оператор ***FindFit***).

Всі ці функціональні оператори працюють, у загальному випадку за методом випадкового пошуку, начинаючи з певної "точки" у просторі визначення функції, крок за кроком зменшуючи початкову величину розв'язку (або збільшуючи для оператора *FindMaximum*).

Наприклад, для функції *x sin(x + 1),* яка є необмеженою в області визначення, але не має глобального екстремуму, але маюча локальні максимуми й мінімуми.

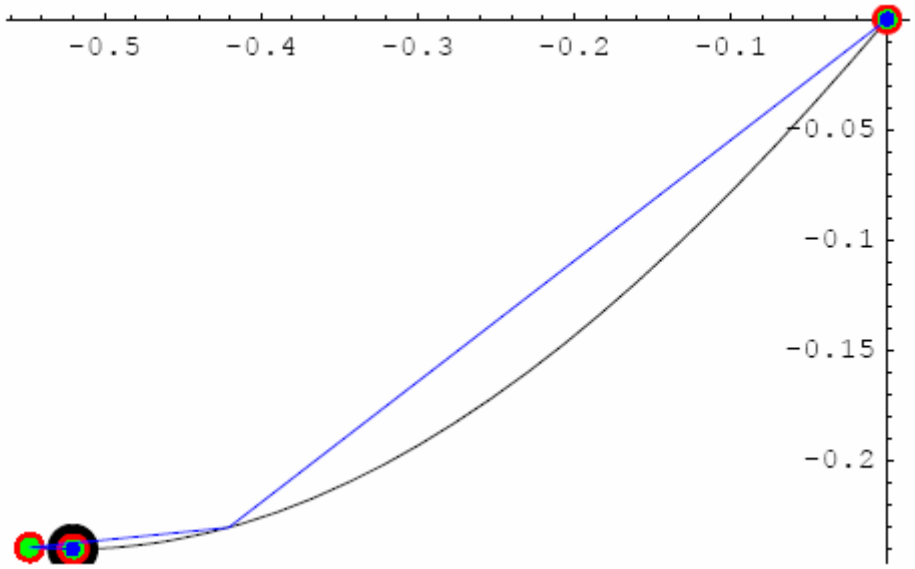
Графік функції представлений нижче

```
In[11]:= Plot[xSin[x+1],{x,-10,10}];
```



Графік нижче показує наближення до точки локального мінімуму, якщо стартуємо із точки *x = 0.*

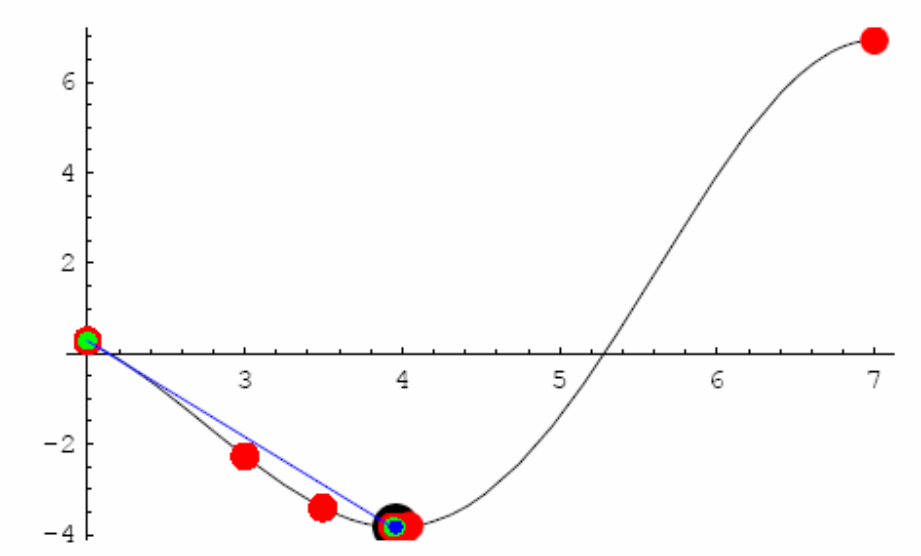
```
In[12]:= FindMinimumPlot[xSin[x+1],{x,0}]
```




```
Out[12]= {{-0.240125,{x → -0.520269}},{Steps → 5,Function → 6},-Graphics-}
```

У той же час, починаючи в точці x = 2, приходимо в іншу точку локального мінімуму.

$\ln[13] := FindMinimumPlot[xSin[x+1],\{x,2\}]$

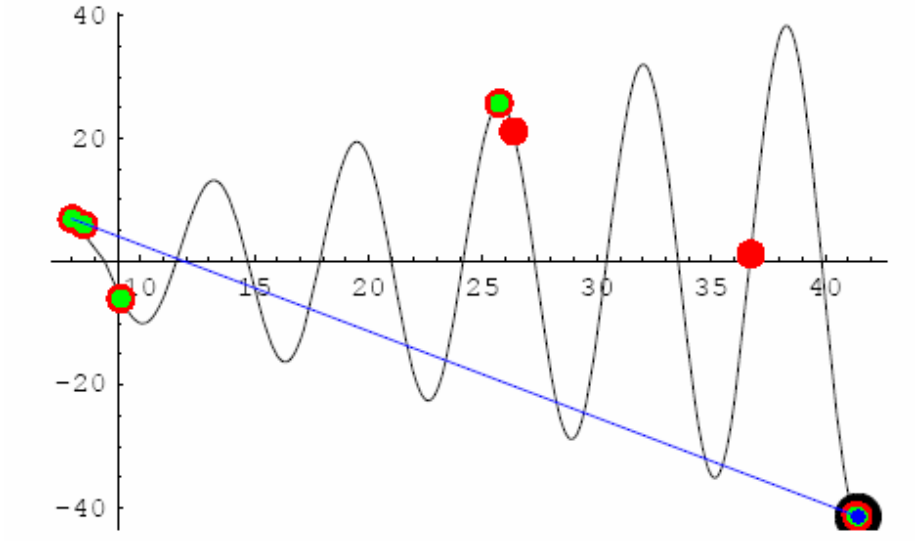


$Out[13] = \{\{-3.83922,\{x \to 3.95976\}\},\{Steps \to 4, Function \to 9\},-Graphics-\}$

Із цих двох графіків можна зробити помилковий висновок, що якщо функція “іде вниз”, то завжди досягнемо мінімуму, рухаючись у “потрібному” напрямку. Проте це не завжди так, хоча функціональний оператор *FindMinimumtakes*, застосовуваний для розв’язання задачі оптимізації приводить до шуканого розв’язку. Так, якщо похідна в точках наближення є досить малою величиною, то оператор *FindMinimum* може не приводити до розв’язку навіть за досить великий час обчислень.

Графік нижче показує типову ситуацію з оператором знаходження локального мінімуму, коли точка старту дорівнює x = 7,

$\ln[14] := FindMinimumPlot[xSin[x+1],\{x,7\}]$

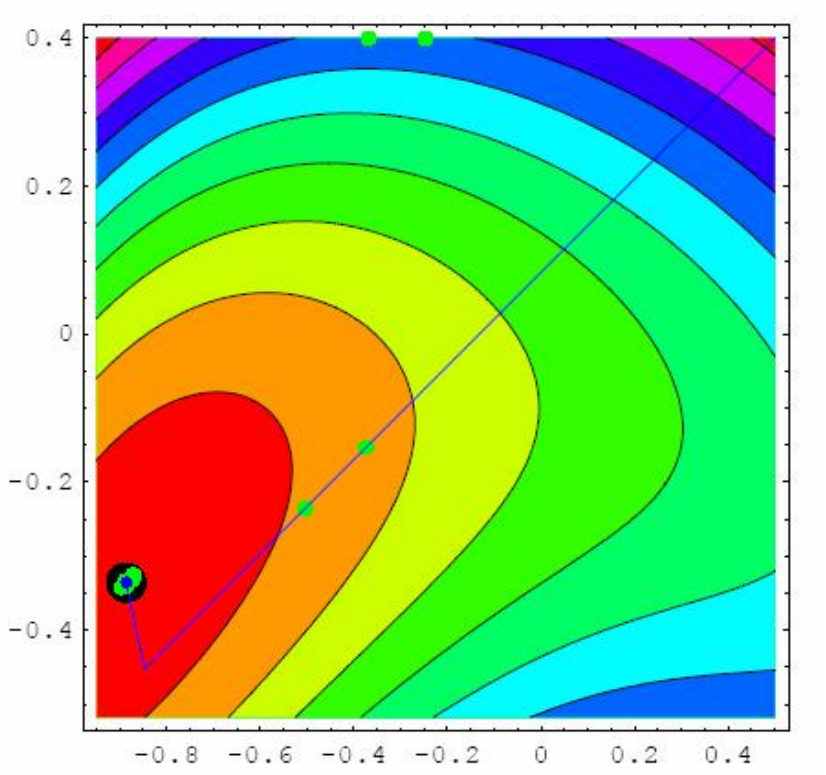


$Out[14] = \{\{-41.4236, \{x \to 41.4356\}\}, \{Steps \to 3, Function \to 14, Gradient \to 14\}, -Graphics-\}$

хоча точка старту перебуває поблизу локального максимуму функції. Точками подзначені кроки до мінімуму. Така поведінка в наближенні обумовлена величиною кута нахилу дотичної до кривого графіка функції в стартовій точці.

Нарешті приведемо розв'язок задачі по визначенню локального мінімуму для функції (5.11), застосовуючи оператор *FindMinimumPlot[...]* і вбудований пакет, що природно повинен бути завантажений на початку процедури визначення мінімуму.

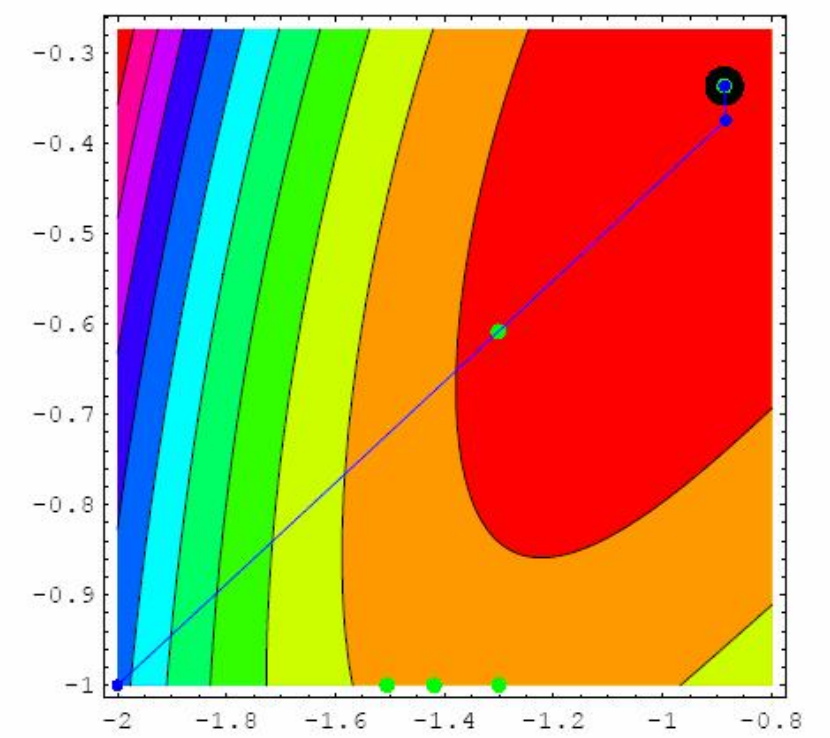
$In[29] := FindMinimumPlot[z, \{\{x, 0.5, 0\}, \{y, 0.4, 0\}\}, ColorFunction \to Hue]$



$Out[29] = \{\{-0.832579, \{x \to -0.886314, y \to -0.335667\}\}, \{Steps \to 4, Function \to 125\},$
$-ContourGraphics-\}$

Те ж саме рішення при старті з іншої точки.

$In[30] := FindMinimumPlot[z, \{\{x, -2, 1\}, \{y, -1, 1\}\}, ColorFunction \to Hue]$



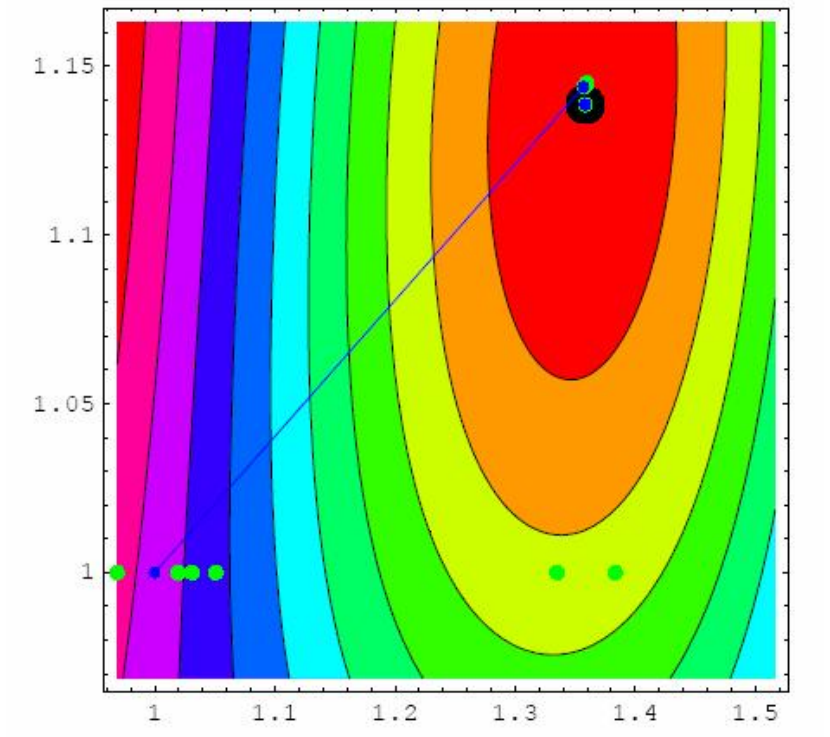
$Out[30] = \{\{-0.832579, \{x \to -0.886324, y \to -0.335671\}\}, \{Steps \to 3, Function \to 87\},$
$-ContourGraphics-\}$

Два приклади для функції вигляду

$$z(x, y) = Cos[x^2 - 3y]^2 + Sin[x^2 + y^2]^2; \qquad (5.12)$$

наведено нижче, як ілюстрація можливостей методу

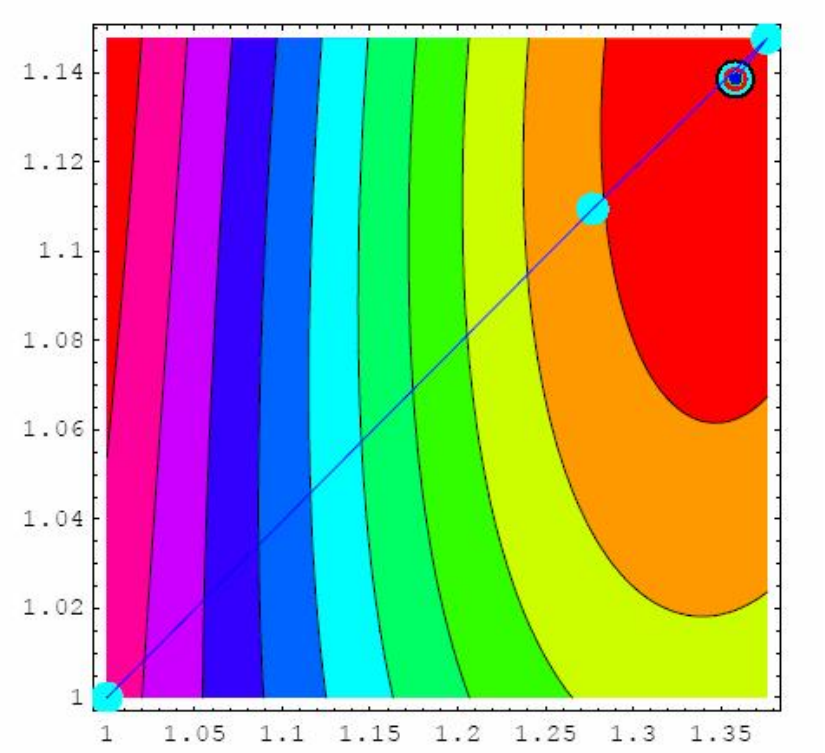
$In[25] := FindMinimumPlot[Cos[x\text{^}2 - 3y]^2 + Sin[x\text{^}2 + y\text{^}2]^2,$
$\{\{x,1,0\}, \{y,1,0\}\}, ColorFunction \to Hue]$



$Out[25] = \{\{1.14494 \times 10^{-13}, \{x \to 1.35835, y \to 1.13863\}\}, \{Steps \to 3, Function \to 97\},$
$-ContourGraphics-\}$

і застосування опції $Method \to Newton$ в операторі *FindMinimumPlot[...].*

$\ln[28] := FindMinimumPlot[Cos[x\text{^}2 - 3y]^2 + Sin[x\text{^}2 + y\text{^}2]^2,$
$\{\{x,1\},\{y,1\}\}, Method \to Newton, ColorFunction \to Hue]$



$Out[28] = \{\{7.99633 \times 10^{-28}, \{x \to 1.35835, y \to 1.13863\}\},$
$\{Steps \to 5, Function \to 6, Gradient \to 6, Hessian \to 6\}, -ContourGraphics-\}$

В останньому випадку для відшукання оптимуму був використаний метод Ньютона.

Тема 6

ЗАДАЧІ ОПТИМІЗАЦІЇ ЗІ ЗВ'ЯЗКАМИ

6.1. Вступна частина

При рішенні задач оптимізації зі зв'язками звичайно використовується метод підстановки. Суть методу полягає в тому, що з рівняння зв'язку виключається / знаходиться одна зі змінних, а потім вона використовується у функції мети.

6.1.1. Метод підстановки в задачі оптимізації

Як найпростіший приклад рішення класичної задачі оптимізації, розглянемо задачу максимізації об'єму куба в сфері одиничного радіуса. Якщо, наприклад, сторони куба *2x, 2y, 2z*, то ми ставимо задачу максимізації функції *f = 2x2y2z* (за геометричним змістом об'єму)

```
In[10]:= Quit[];
In[1]:= f = 8xyz;
```

залежного зв'язку

```
In[2]:= g = x^2 + y^2 + z^2 - 1.
```

Рішення поставленої задачі оптимізації подано нижче такими кодами.

Знайдемо, наприклад, z з рівняння зв'язку

```
In[3]:= zSolve[g = 0, z]
Out[3] = {{x → -√(-x^2 - y^2 + 1)}, {z → √(-x^2 - y^2 + 1)}}.
```

З отриманих рішень використаємо останнє. Підставимо його у функцію мети

```
In[4]:= funetSol = f /. zSol[[2]]
Out[4] = 8xy√(-x^2 - y^2 + 1)
```

.

Для знаходження критичних точок скористаємося прийомом використовуваним у попередньому розділі

```
In[5]:= grad[f_, x_] := D[f, #] & /@ x;
crit[f_, x_] := Seleet[Union[Solve[grad[f, x] = 0, x]], FreeQ[#, Complex]&].
```

Використовуючи наведені вище оператори, знайдемо критичні точки

```
In[7]:= crit[funetSol, {x, y}]
```

$$Out[7] = \{\{x \to 0, y \to 0\}, \{x \to -\frac{1}{\sqrt{3}}, y \to -\frac{1}{\sqrt{3}}\},$$

$$\{x \to -\frac{1}{\sqrt{3}}, y \to \frac{1}{\sqrt{3}}\}, \{x \to \frac{1}{\sqrt{3}}, y \to -\frac{1}{\sqrt{3}}\}, \{x \to \frac{1}{\sqrt{3}}, y \to \frac{1}{\sqrt{3}}\}\}$$

Очевидно, за геометричним змістом, що тільки останнє рішення відповідає рішенню поставленої задачі

```
In[8]:= zSol[[2]]/.Last[%]
Out[8]= {z → 1/Sqrt[3]}
```

.

Найбільший об'єм вписаного в одиничну сферу куба дорівнює

```
In[9]:= f/.%/.Last[%%]
Out[9]= 8/(3 Sqrt[3])
In[10]:= N[%]
Out[10]= 1.5396
```

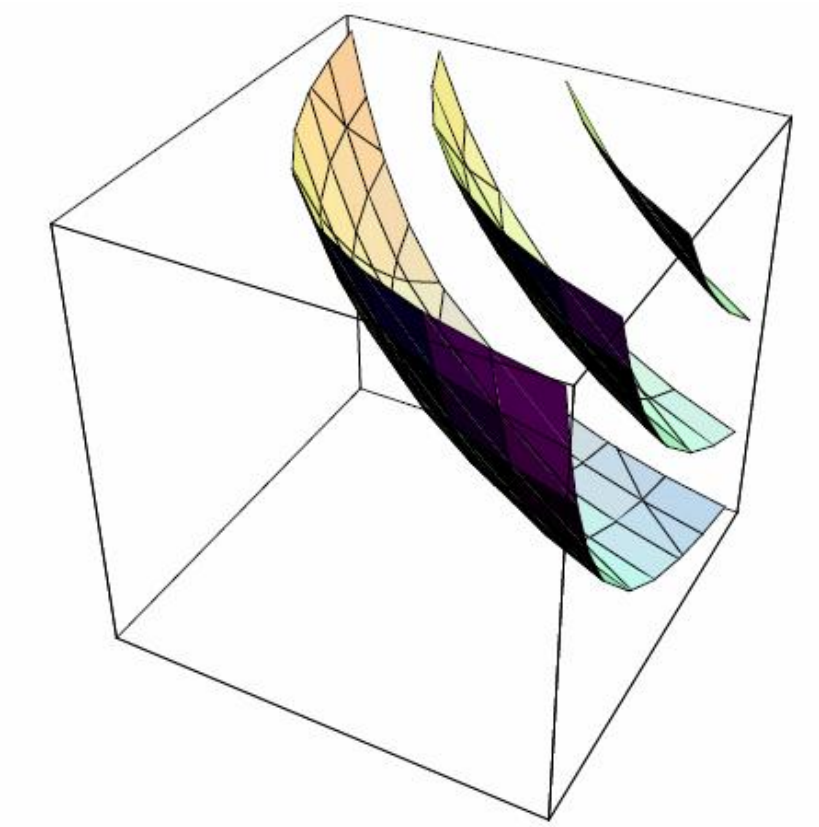
.

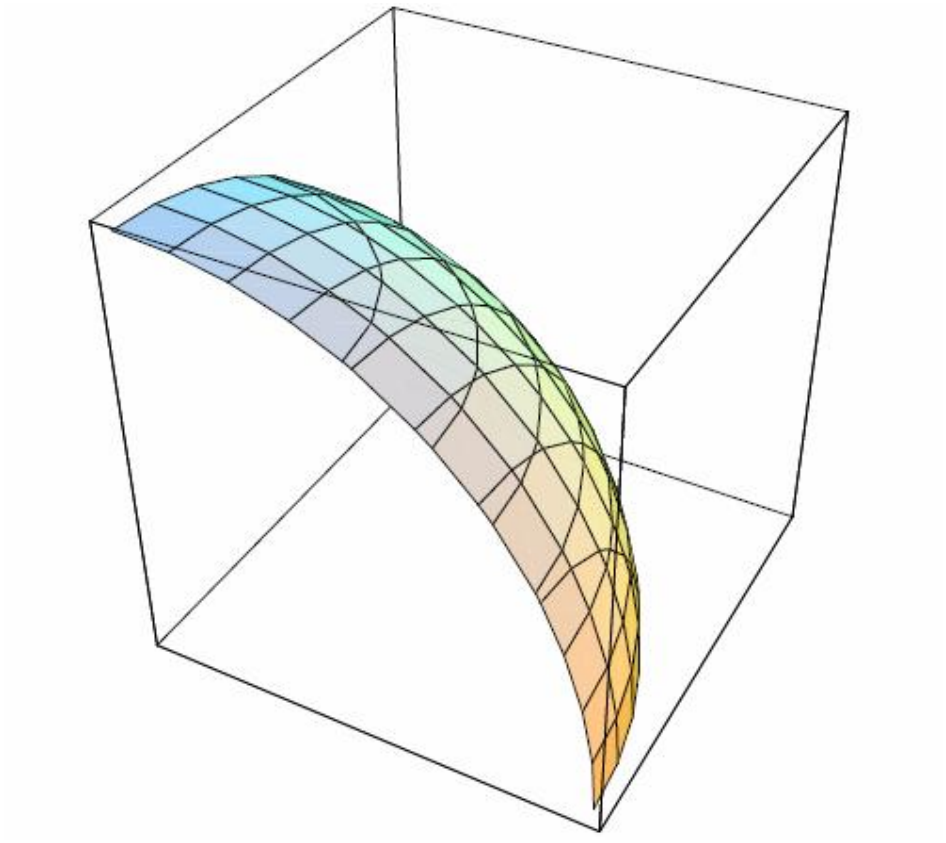
6.1.2. Візуалізація оптимального рішення

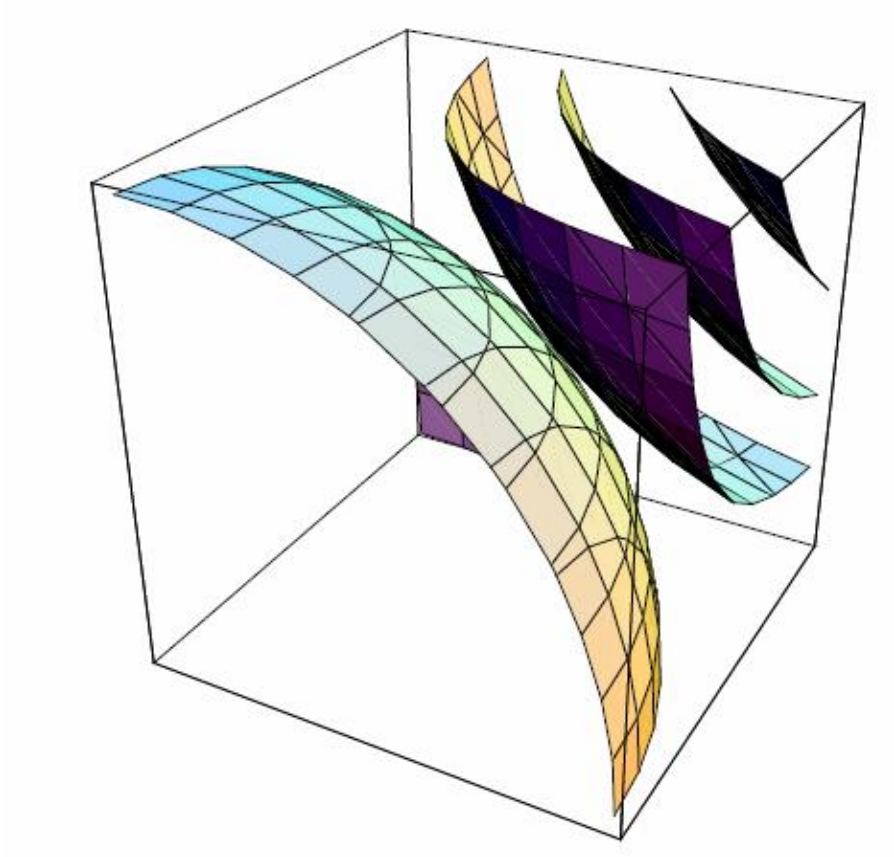
Як відомо, графічне подання функції трьох змінних – це поверхня в просторі. Причому на цій поверхні функція набуває деякого постійного значення, як і на площині, геометричним образом є крива, де функція набуває постійного значення.

Тому для візуалізації рішення побудуємо відповідні контурні графіки поверхонь. Ми побудуємо три контурні поверхні для функції мети *f*, яку описали раніше. Значення функції в точці оптимуму дорівнює *8/(3 Sqrt[3])=1.54,* а величини 3 і 5 обмежують оптимум зверху і знизу. Тому саме ці значення використаємо при побудові відповідних графіків.

```
In[11]:=<< Graphics'ContourPlot3D'
In[12]:= obj = ContourPlot3D[f,{x,0,1},{y,0,.1},{z,0,1},Contours → {8/(3Sqrt[3]),3,5}];
con = ContourPlot3D[x^2 + y^2 + z^2 - 1,{x,0,1},{y,0,1},{z,0,1}];
Show[obj,con,ViewPoint → {1.3,-2.4,1.5}]
```







Out[14] = –*Graphics*3*D* –

Оптимальне рішення знаходиться у точці зіткнення.

Тема 7

СИМВОЛЬНІ І ЧИСЛОВІ РІШЕННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

7.1. Звичайні диференціальні рівняння

Символьні рішення звичайних диференціальних рівнянь (ЗДР) і диференціальних рівнянь у частинних похідних розглядаються і розв'язуються методами символьної алгебри системи *Mathematica*.

7.1.1. Вступ у частинні диференціальні рівняння

Рішення частинних диференціальних рівнянь у середовищі *Mathematica* виробляється за допомогою одного із двох операторів

Dsolve[]

NDSolve[]

причому перший з них використовується як для символьних рішень, так і для числових реалізацій, а другий тільки для числових рішень. Пакет *<<Calculus'DSolve'* дозволяє вирішити і ряд нелінійних задач включаючи рішення через спеціальні функції.

Крім того використовуючи убудований пакет *<<Calculus 'VariationalMethods'* користувач може вирішувати задачі варіаційного обчислення, знаходити перші інтеграли і т. ін.

Інший пакет *<<Calculus'VectorAnalysis'* вирішує диференціальні рівняння записані в криволінійних системах координат.

За допомогою зазначених операторів, крім загальних рішень, можна одержати й особові рішення, які перебувають і записуються через початкові і крайові умови.

7.1.2. Символьні рішення ЗДР

7.1.2.1. Одне рівняння. За допомогою наступного оператора можна одержати розв'язок одного звичайного диференціального рівняння першого порядку

*Dsolve[eq, y, x] (*рішення рівняння eq,*

y - залежна змінна,

x - незалежна змінна).*

Розглянемо приклад

$\ln[1] := eq1 = y'[x] = ay[x] + bx + c;$

Рішення у вигляді функціонального символьного вираження у комп'ютерних кодах має вигляд

$\ln[2] := soll = DSolve[eq1, y, x] // Flatten$

$Out[2] = \{y \to Function[\{x\}, e^{ax} c_1 - \frac{axb + b + ac}{a^2}]\}$.

Інше подання того ж самого символьного рішення, але у вигляді явної функції незалежної змінної x, представлено нижче.

$\ln[3] := soll = DSolve[eq1, y, x] // Flatten$

$Out[3] = \{y(x) \to e^{ax} c_1 - \frac{axb + b + ac}{a^2}]\}$.

Далі розглянемо застосування вбудованих пакетів

$Quit[];$

$\ln[2] :=<< Calculus'VariationalMethods'$.

Знайдемо варіаційну похідну від такого функціонала:

$$F = \int_{x_{\min}}^{x_{\max}} y(x)\sqrt{1 + y'(x)^2}\, dx \ . \qquad (7.1)$$

Код оператора варіаційної похідної наведений нижче:

$\ln[6] := VariationalD[\left(y[x]\sqrt{1 + y'[x]^2}\right), y[x], x]$

$Out[6] = \frac{y'(x)^2 - y(x)y''(x) + 1}{(y'(x)^2 + 1)^{3/2}}$.

Застосування убудованого пакета *Calculus'VariationalMethods'* дозволяє не тільки оперувати з функціоналами, але й формувати математичні постановки задач для різних додатків, що допускають варіаційне формулювання.

Наприклад, у задачі про рух математичного маятника маємо вигляд всієї енергії маятника:

$$e = \frac{ml^2 O'(t)^2}{2} + mg / Cos(O(t)), \qquad (7.2)$$

що дозволяє виводити диференціальне рівняння руху

$\ln[7] := e = \frac{ml^2 O'(t)^2}{2} + mg / Cos(O(t));$

$\ln[8] := EulerEquations[e, O(t), t]$

$Out[8] = -lm(g \sin(O(t)) + lO''(t)) = 0$

яке повністю збігається із класичним результатом.

Нарешті застосовуючи оператор рішення ЗДР приходимо до відомого загального розв'язку для математичного маятника.

$\ln[10] := solPendulum = DSolve[EulerEquations[e, O(t), t], O(t), t] // Flatten$

Solve::ifun : Inverse functions are being used by Solve, so some

Solutions may not be found; use Reduce for complete solution information. More…

$$Out[10] = \{O(t) \to 2am\left(-\frac{\sqrt{2gt^2 + lc_1t^2 + 4gc_2t + 2lc_1c_2t + 2gc_2^2 + lc_1c_2^2}}{2\sqrt{l}}\middle| \frac{4g}{2g + lc_1}\right),$$

$$O(t) \to 2am\left(\frac{\sqrt{2gt^2 + lc_1t^2 + 4gc_2t + 2lc_1c_2t + 2gc_2^2 + lc_1c_2^2}}{2\sqrt{l}}\middle| \frac{4g}{2g + lc_1}\right)\}$$

7.1.2.2. Задача з початковими умовами. Ми можемо ввести початкові умови безпосередньо в оператор розв'язку

Dsolve[{eq, y[x0]?y0}, y, x] («рішення рівняння eq,

y - залежна змінна,

x - незалежна змінна,

y0, x0 початкові умови)*

Запишемо розв'язок попереднього рівняння з початковими умовами

$$In[14] = Quit[];$$

$$In[1] := eq1 = y'[x] = ay[x] + bx + c;$$

$$In[3] := soll = DSolve[\{eq1, y[x0] = yO\}, y[x], x] // Flatten // Simplify$$

$$Out[3] = \{y(x) \to \frac{e^{-ax0}(b(e^{ax}(ax0+1) - e^{ax0}(ax+1)) + a(c(e^{ax} - e^{ax0}) + ae^{ax}y0))}{a^2}\}$$

Рішення із привласненим числовим значенням параметра *a* має вигляд

$$In[4] := soll /. a \to 1$$

$$Out[4] = \{y(x) \to e^{-x0}(c(e^x - e^{x0}) + b(e^x(x0+1) - e^{x0}(x+1)) + e^x y0)\}.$$

Представимо рішення з нелінійними початковими умовами

$$In[6] := sol2 = DSolve[\{eq1, y[0]\text{^}2 - y[0] - 2 = 0\}, y[x], x] // Flatten // Simplify$$

$$Out[6] = \{y(x) \to \frac{a(2e^{ax}a + c(-1 + e^{ax})) + b(-ax + e^{ax} - 1)}{a^2},$$

$$y(x) \to \frac{b(-ax + e^{ax} - 1) - a(e^{ax}a - ce^{ax} + c)}{a^2}\}$$

Отже з отриманого рішення маємо два різні розв'язки, що відповідають двом початковим умовам.

Інший приклад одержання символьного розв'язку для математичного маятника, рух якого проходить у малій площині положення рівноваги, подано нижче:

$$In[1] :=<< Calculus'VariationalMethods'$$

Розкладання потенційної енергії маятника в ряд за ступенями кута відхилення *O(t)* виконано також у символьному вигляді.

$$\text{In}[6] := e = \frac{ml^2 O'(t)^2}{2} + mglSeries[Cos(O(t)),\{O(t),0,2\}]\,//\,Normal\ .$$

Потім нижче представлений висновок лінійного ЗДР руху математичного маятника,

$$\text{In}[7] := EulerEquations[e, O(t), t]$$

$$Out[7] = -lm(gO(t) + lO''(t)) = 0$$

Відповідно часткове символьне рішення, що відповідає гармонійним коливанням маятника, має вигляд

$$\text{In}[9] := solPendulum = DSolve[\{EulerEquations[e, O(t), t], O(0) = v_0, O'(0) = 0\}, t]\,//\,Flatten$$

$$Out[9] = \{O(t) \to \cos\left(\frac{\sqrt{g}t}{\sqrt{l}}\right)v_0\}\ ,$$

який збігається із класичним рішенням незатухаючих коливань маятника.

7.1.2.3. Табуляція і побудова графіків розв'язків. Якщо необхідно табулювати рішення ЗДР або побудувати графік розв'язків, то, як відомо, ми повинні задати всі числові значення параметрів ЗДР, знайти його частинний розв'язок, що зображено нижче

$$\text{In}[7] := sol4 = DSolve[\{y'[x] = y[x] + x + 1, -y[-2] - 2 = 0\}, y[x], x]$$

$$Out[7] = \{\{y(x) \to -x - 2e^{x+2} - 2\}\}\ ,$$

а потім виконати табуляцію отриманого рішення.

$$\text{In}[8] := Table[\{x, y[x]/.sol4\}, \{x,0,4,0.2\}]\,//\,TableForm\,//\,PaddedForm\,//\,N$$

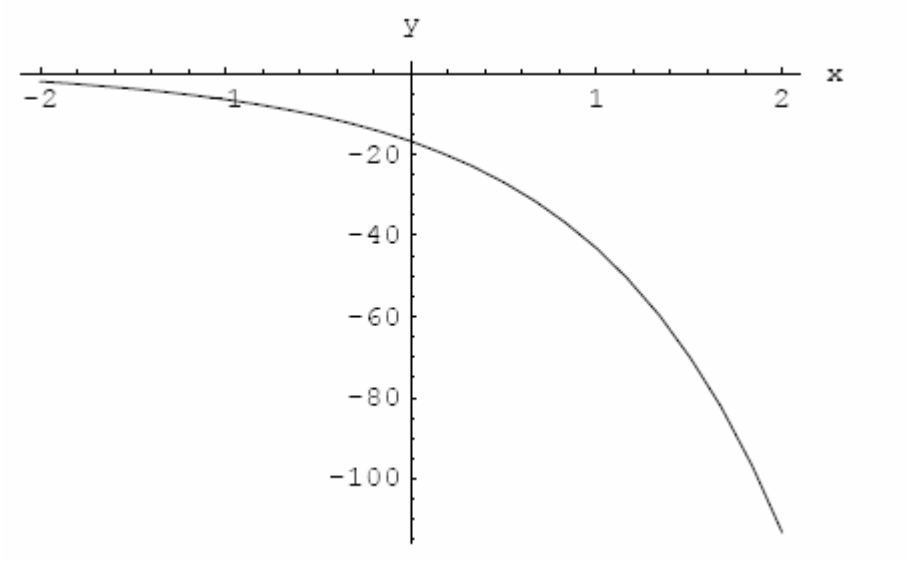
$$Out[8]\,//\,PaddedForm =$$

0.	*-16.7781*
0.2	*-20.25*
0.4	*-24.4464*
0.6	*-29.5275*
0.8	*-35.6893*
1.	*-43.1711*
1.2	*-52.2651*
1.4	*-63.3282*
1.6	*-76.7965*
1.8	*-93.2024*
2.	*-113.196*
2.2	*-137.573*

2.4	*-167.302*
2.6	*-203.569*
2.8	*-247.821*
3.	*-301.826*
3.2	*-367.744*
3.4	*-448.213*
3.6	*-546.453*
3.8	*-666.399*
4.	*-812.858*

Далі будуємо графік отриманого розв'язку

```
In[10]:= p1 = Plot[Evaluate[y[x] /. sol 4], {x, -2, 2}, AxesLabel → {"x", "y"}];
```



Очевидно, що отриманий розв'язок можна використати на довільному замкнутому проміжку незалежної змінної.

Далі представимо частковий розв'язок і його числова реалізація для математичного маятника. Повторюючи без коментарів висновок особового рішення для математичного маятника

```
In[10]:= Quit[];
In[1]:= <<Calculus`VariationalMethods`
In[2]:= e = m l^2 O'[t]^2 / 2 + m g l Series[Cos[O[t]], {O[t], 0, 2}] // Normal;
In[3]= EulerEquations[e, O[t], t]
Out[3]= -l m (g O[t] + l O''[t]) = 0
In[4]:= solPendulum = DSolve[{EulerEquations[e, O[t], t] /. O[0] == v0, O'[0] == 0}, O[t], t] // Flatten
Out[4]= {O[t] → Cos[Sqrt[g] t / Sqrt[l]] v0},
```

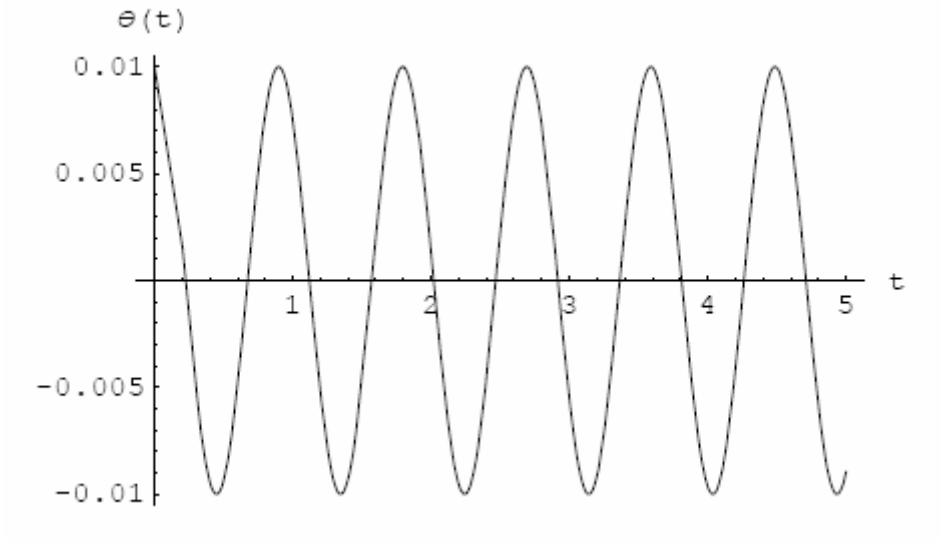
складемо таблицю табульованих значень кута відхилення маятника від вертикалі

```
In[7]:= Table[{t, O(t) /. solPendulum /. {v0 → .01, g → 9.81, l → .2}}, {t, 0, 2, 0.2}] // TableForm //
 PaddedForm // N
Out[7] // PaddedForm =
```

0.	0.01
0.2	0.00169263
0.4	-0.009427
0.6	-0.00488393
0.8	0.00777366
1.	0.00751552
1.2	-0.00522946
1.4	-0.00928583
1.6	0.00208595
1.8	0.00999198
2.	0.0012966

а потім зобразимо графік руху математичного маятника.

```
In[8]:= grPendulum =
 Plot[O(t) /. solPendulum /. {v0 → .01, g → 9.81, l → .2}, {t, 0, 5}, AxesLabel → {"t", "O(t)"}];
```



7.1.3. Звичайне диференціальне рівняння вищого порядку. Розв'язання крайової задачі

Спочатку розглянемо розв'язок звичайного диференціального рівняння другого порядку, яке подано нижче.

$$y'' - y = 1 - x. \tag{7.3}$$

```
In[11] := Quit[];
In[1] := eg2 = y''[x] - y[x] = 1 - x;
```

Загальне розв'язання ЗДР з ім'ям *eq2* подано нижче.

```
In[2] := sol5 = DSolve[eg2, y[x], x] // Flatten // Simplify
Out[2] = {y(x) → x + e^x c_1 + e^(-x) c_2 - 1}
```

Розглянемо розв'язок крайової задачі вигляду:

$$\begin{aligned} &y'' - y = 1 - x; \\ &y(0) = 1; \\ &y(2) = 0, \end{aligned} \tag{7.4}$$

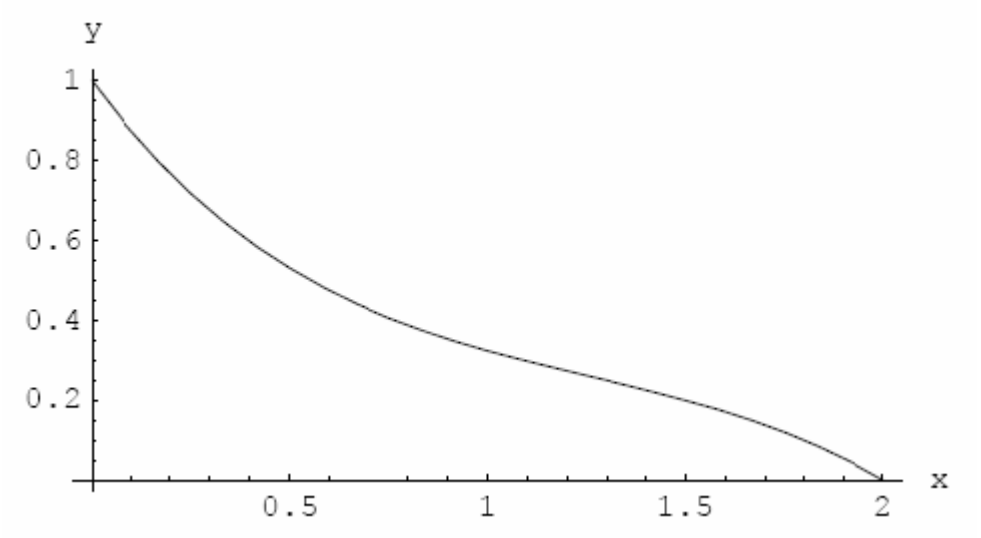
яка розв'язана кодом, наведеним нижче.

```
In[3] := sol5 = DSolve[{eg2, y[0] = 1, y[2] = 0}, y[x], x] // Flatten // Simplify
```

$$Out[3] = \{y(x) \to -\frac{e^4(x-1) + e^{2-x} + 2e^{4-x} - 2e^x - e^{x+2} - x + 1}{1 - e^4}\}$$

Графік розв'язку крайової задачі (7.4) має вигляд

```
In[4] := gr5 = Plot[Evaluate[y[x] /. sol5], {x,0,2},
AspectRatio → Automatic, PlotRange → All, AxesLabel → {"x", "y"}];
```



Потім розглянемо розв'язок крайової задачі з коефіцієнтами залежними від змінної *x*. Нехай крайова задача має вигляд

$$y'' + (1 - 2px)\, y = 1 - x; \qquad (7.5)$$
$$y(0) = 1;$$
$$y(2) = 0,$$

У кодах рівняння (7.5) запишеться в такий спосіб:

```
In[5]:= eq3 = y''[x] + (1 - 2px) y[x] == 1 - x .
```

Система *Mathematica* дозволяє визначити загальне символьне розв'язання ЗДР (7.5), використовуючи той же самий формат оператора, що й раніше,

```
In[6]:= sol6 = DSolve[eq3, y[x], x] //Flatten // Simplify
```

$$Out[6] = \{y(x) \to \frac{1}{216p^2\Gamma(\frac{2}{3})\Gamma(\frac{4}{3})\Gamma(\frac{5}{3})}$$

$$\left(\Gamma\left(\frac{2}{3}\right)\left(-\sqrt[6]{3}(-1+2p)\left(3Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)+\sqrt{3}Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\right)\Gamma\left(\frac{2}{3}\right){}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(2px-1)^3}{36p^2}\right)\right.\right.$$

$$(1-2px)^2 - 108p^2\left(Ai'\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right) - 2c_2 Bi\left(\frac{2px-1}{(2p)^{2/3}}\right) -\right.$$

$$\left.Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)\left(Bi'\left(\frac{2px-1}{(2p)^{2/3}}\right)+2c_1\right)\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)\Big) -$$

$$3\sqrt[3]{3}(2p)^{2/3}(-1+2p)(2px-1)\left(\sqrt{3}Ai\left(\frac{2px-1}{(2p)^{2/3}}\right) - Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\right)\Gamma\left(\frac{1}{3}\right)$$

$$\left.\left.\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(2px-1)^3}{36p^2}\right)\right)\right\}$$

і крім того, тим же самим оператором можна вирішити і крайову задачу (7.5). Символьне розв'язання лінійної крайової задачі (7.5) має вигляд

```
In[7]:= sol7 = DSolve[{eq3, y[0] == 1, y[2] == 0}, y[x], x] // Flatten // Simplify
```

$$Out[7] = \{y(x) \to \left(-\sqrt[6]{3}(-1+2p)\left(Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right) - Ai\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\right)\right.$$

$$\left(\sqrt{3}Ai\left(\frac{2px-1}{(2p)^{2/3}}\right) + Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(2px-1)^3}{36p^2}\right)(1-2px)^2 +$$

$$3\sqrt[3]{3}(2p)^{2/3}(-1+2p)(2px-1)\left(Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right) - Ai\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\right)$$

$$\left(\sqrt{3}Bi\left(\frac{2px-1}{(2p)^{2/3}}\right) - 3Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(2px-1)^3}{36p^2}\right) -$$

$$32\sqrt[6]{3}p^3 Ai\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) +$$

$$32\sqrt[6]{3}p^2 Ai\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) -$$

$$10\sqrt[6]{3}pAi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) +$$

$$\sqrt[6]{3}Ai\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) -$$

$$323^{2/3}p^3 Ai\left(-\frac{1}{(2p)^{2/3}}\right)Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) +$$

$$323^{2/3}p^2 Ai\left(-\frac{1}{(2p)^{2/3}}\right)Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) -$$

$$103^{2/3}pAi\left(-\frac{1}{(2p)^{2/3}}\right)Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) +$$

$$3^{2/3}Ai\left(-\frac{1}{(2p)^{2/3}}\right)Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) +$$

$$32\sqrt[6]{3}p^3 Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) -$$

$$32\sqrt[6]{3}p^2 Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) +$$

$$10\sqrt[6]{3}pAi\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) -$$

$$\sqrt[6]{3}Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) +$$

$$323^{2/3}p^3 Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) -$$

$$323^{2/3}p^2 Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)^2 {}_1F_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) +$$

$$
\begin{aligned}
&103^{2/3} p Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right) Ai\left(\frac{2px-1}{(2p)^{2/3}}\right) Bi\left(-\frac{1}{(2p)^{2/3}}\right) \Gamma\left(\frac{2}{3}\right)^2 {}_1\mathrm{F}_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) - \\
&3^{2/3} Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right) Ai\left(\frac{2px-1}{(2p)^{2/3}}\right) Bi\left(-\frac{1}{(2p)^{2/3}}\right) \Gamma\left(\frac{2}{3}\right)^2 {}_1\mathrm{F}_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};\frac{(-1+4p)^3}{36p^2}\right) + \\
&2\sqrt[6]{3} p Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right) Bi\left(-\frac{1}{(2p)^{2/3}}\right) Bi\left(\frac{2px-1}{(2p)^{2/3}}\right) \Gamma\left(\frac{2}{3}\right)^2 {}_1\mathrm{F}_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};-\frac{1}{36p^2}\right) - \\
&\sqrt[6]{3} Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right) Bi\left(-\frac{1}{(2p)^{2/3}}\right) Bi\left(\frac{2px-1}{(2p)^{2/3}}\right) \Gamma\left(\frac{2}{3}\right)^2 {}_1\mathrm{F}_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};-\frac{1}{36p^2}\right) + \\
&23^{2/3} p Ai\left(-\frac{1}{(2p)^{2/3}}\right) Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right) Bi\left(\frac{2px-1}{(2p)^{2/3}}\right) \Gamma\left(\frac{2}{3}\right)^2 {}_1\mathrm{F}_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};-\frac{1}{36p^2}\right) - \\
&3^{2/3} Ai\left(-\frac{1}{(2p)^{2/3}}\right) Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right) Bi\left(\frac{2px-1}{(2p)^{2/3}}\right) \Gamma\left(\frac{2}{3}\right)^2 {}_1\mathrm{F}_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};-\frac{1}{36p^2}\right) - \\
&23^{2/3} p Ai\left(-\frac{1}{(2p)^{2/3}}\right) Ai\left(\frac{2px-1}{(2p)^{2/3}}\right) Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right) \Gamma\left(\frac{2}{3}\right)^2 {}_1\mathrm{F}_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};-\frac{1}{36p^2}\right) + \\
&3^{2/3} Ai\left(-\frac{1}{(2p)^{2/3}}\right) Ai\left(\frac{2px-1}{(2p)^{2/3}}\right) Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right) \Gamma\left(\frac{2}{3}\right)^2 {}_1\mathrm{F}_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};-\frac{1}{36p^2}\right) - \\
&2\sqrt[6]{3} p Ai\left(\frac{2px-1}{(2p)^{2/3}}\right) Bi\left(-\frac{1}{(2p)^{2/3}}\right) Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right) \Gamma\left(\frac{2}{3}\right)^2 {}_1\mathrm{F}_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};-\frac{1}{36p^2}\right) + \\
&\sqrt[6]{3} Ai\left(\frac{2px-1}{(2p)^{2/3}}\right) Bi\left(-\frac{1}{(2p)^{2/3}}\right) Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right) \Gamma\left(\frac{2}{3}\right)^2 {}_1\mathrm{F}_2\left(\frac{2}{3};\frac{4}{3},\frac{5}{3};-\frac{1}{36p^2}\right) + \\
&33^{5/6} (2p)^{2/3} Ai\left(-\frac{1}{(2p)^{2/3}}\right) Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right) Bi\left(\frac{2px-1}{(2p)^{2/3}}\right) \Gamma\left(\frac{4}{3}\right) \Gamma\left(\frac{5}{3}\right) \\
&{}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(-1+4p)^3}{36p^2}\right) + 242^{2/3} 3^{5/6} p^{8/3} Ai\left(-\frac{1}{(2p)^{2/3}}\right) Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right) \\
&Bi\left(\frac{2px-1}{(2p)^{2/3}}\right) \Gamma\left(\frac{4}{3}\right) \Gamma\left(\frac{5}{3}\right) {}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(-1+4p)^3}{36p^2}\right) - 182^{2/3} 3^{5/6} p^{5/3} Ai\left(-\frac{1}{(2p)^{2/3}}\right) \\
&Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right) Bi\left(\frac{2px-1}{(2p)^{2/3}}\right) \Gamma\left(\frac{4}{3}\right) \Gamma\left(\frac{5}{3}\right) {}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(-1+4p)^3}{36p^2}\right) - 9\sqrt[3]{3} (2p)^{2/3} \\
&Ai\left(-\frac{1}{(2p)^{2/3}}\right) Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right) Bi\left(\frac{2px-1}{(2p)^{2/3}}\right) \Gamma\left(\frac{4}{3}\right) \Gamma\left(\frac{5}{3}\right) {}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(-1+4p)^3}{36p^2}\right) - \\
&722^{2/3} \sqrt[3]{3} p^{8/3} Ai\left(-\frac{1}{(2p)^{2/3}}\right) Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right) Bi\left(\frac{2px-1}{(2p)^{2/3}}\right) \Gamma\left(\frac{4}{3}\right) \Gamma\left(\frac{5}{3}\right) \\
&{}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(-1+4p)^3}{36p^2}\right) + 542^{2/3} \sqrt[3]{3} p^{5/3} Ai\left(-\frac{1}{(2p)^{2/3}}\right) Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right) \\
&Bi\left(\frac{2px-1}{(2p)^{2/3}}\right) \Gamma\left(\frac{4}{3}\right) \Gamma\left(\frac{5}{3}\right) {}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(-1+4p)^3}{36p^2}\right) - 33^{5/6} (2p)^{2/3} Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)
\end{aligned}
$$

$$Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(-1+4p)^3}{36p^2}\right)-242^{2/3}3^{5/6}p^{8/3}$$

$$Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(-1+4p)^3}{36p^2}\right)+$$

$$182^{2/3}3^{5/6}p^{5/3}Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)$$

$${}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(-1+4p)^3}{36p^2}\right)+9\sqrt[3]{3}(2p)^{2/3}Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)$$

$$\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(-1+4p)^3}{36p^2}\right)+722^{2/3}\sqrt[3]{3}p^{8/3}Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)$$

$$Bi\left(-\frac{1}{(2p)^{2/3}}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(-1+4p)^3}{36p^2}\right)-542^{2/3}\sqrt[3]{3}p^{5/3}$$

$$Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};\frac{(-1+4p)^3}{36p^2}\right)-$$

$$33^{5/6}(2p)^{2/3}Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};-\frac{1}{36p^2}\right)+$$

$$62^{2/3}3^{5/6}p^{5/3}Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};-\frac{1}{36p^2}\right)+$$

$$9\sqrt[3]{3}(2p)^{2/3}Ai\left(-\frac{1}{(2p)^{2/3}}\right)Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};-\frac{1}{36p^2}\right)-$$

$$182^{2/3}\sqrt[3]{3}p^{5/3}Ai\left(-\frac{1}{(2p)^{2/3}}\right)Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};-\frac{1}{36p^2}\right)-$$

$$9\sqrt[3]{3}(2p)^{2/3}Ai\left(-\frac{1}{(2p)^{2/3}}\right)Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};-\frac{1}{36p^2}\right)+$$

$$182^{2/3}\sqrt[3]{3}p^{5/3}Ai\left(-\frac{1}{(2p)^{2/3}}\right)Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};-\frac{1}{36p^2}\right)+$$

$$33^{5/6}(2p)^{2/3}Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};-\frac{1}{36p^2}\right)-$$

$$62^{2/3}3^{5/6}p^{5/3}Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right){}_1F_2\left(\frac{1}{3};\frac{2}{3},\frac{4}{3};-\frac{1}{36p^2}\right)+$$

$$48p^3Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{5}{3}\right)+$$

$$36\sqrt{3}p^2Ai\left(-\frac{1}{(2p)^{2/3}}\right)Ai'\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)-$$

$$36\sqrt{3}p^2Ai\left(-\frac{1}{(2p)^{2/3}}\right)Ai'\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)-$$

$$36\sqrt{3}p^2Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Ai'\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)+$$

$$36\sqrt{3}p^{2}Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Ai'\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)-$$

$$36\sqrt{3}p^{2}Ai\left(-\frac{1}{(2p)^{2/3}}\right)Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi'\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)+$$

$$36\sqrt{3}p^{2}Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi'\left(\frac{2px-1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)+$$

$$36\sqrt{3}p^{2}Ai\left(-\frac{1}{(2p)^{2/3}}\right)Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi'\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)-$$

$$36\sqrt{3}p^{2}Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi'\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)-$$

$$36\sqrt{3}p^{2}Ai\left(-\frac{1}{(2p)^{2/3}}\right)Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi'\left(-\frac{1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)+$$

$$36\sqrt{3}p^{2}Ai\left(-\frac{1}{(2p)^{2/3}}\right)Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi'\left(-\frac{1}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)+$$

$$36\sqrt{3}p^{2}Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Ai'\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)-$$

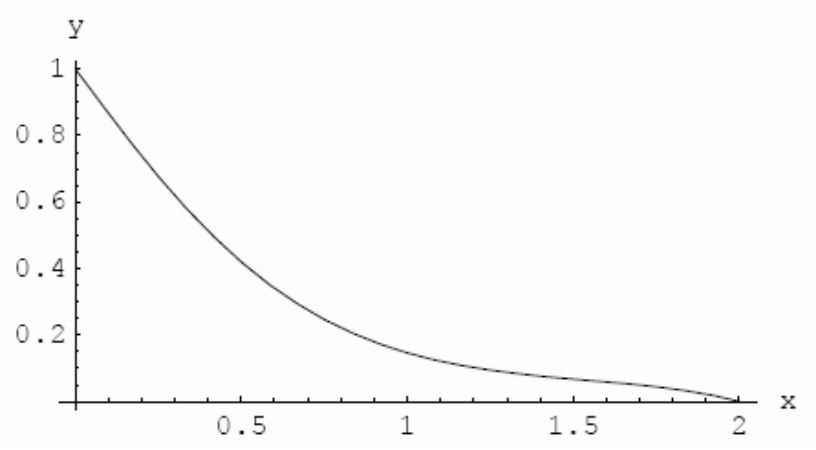
$$36\sqrt{3}p^{2}Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Ai'\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{2}{3}\right)\Gamma\left(\frac{4}{3}\right)\Gamma\left(\frac{5}{3}\right)-$$

$$48p^{3}Ai\left(\frac{2px-1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\Gamma\left(\frac{5}{3}\right)\Bigg)/$$

$$\left(48p^{3}\left(Ai\left(\frac{-1+4p}{(2p)^{2/3}}\right)Bi\left(-\frac{1}{(2p)^{2/3}}\right)-Ai\left(-\frac{1}{(2p)^{2/3}}\right)Bi\left(\frac{-1+4p}{(2p)^{2/3}}\right)\right)\Gamma\left(\frac{5}{3}\right)\right)\}$$

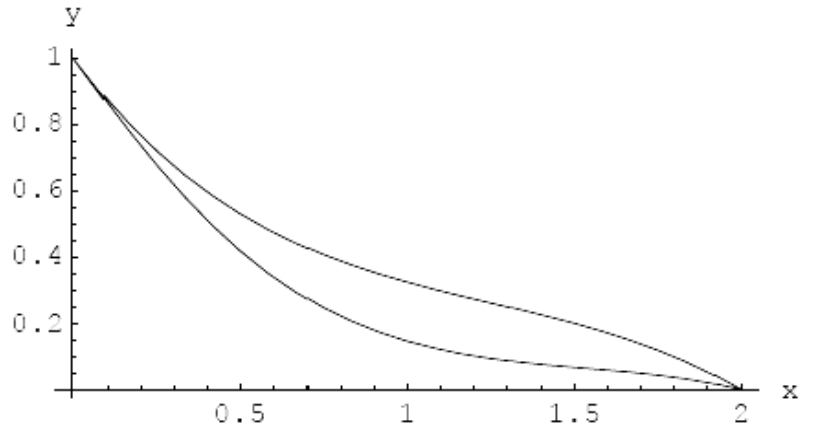
Очевидно, що використовуючи знайдений вище символьний розв’язок крайової задачі (7.5), можна побудувати графік цього розв’язку, як це зроблено нижче через оператор *Plot[ ].*

```
In[8]:= gr7 = Plot[Evaluate[y[x]/.sol7],{x,0,2},
AspectRatio → Automatic, PlotRange → All, AxesLabel → {"x","y"}];
```



Порівняльні розв’язки двох крайових задач представлені нижче на одному графіку.

In[9] := *Show*[*gr*5, *gr*7];



Із представлених графіків наочно видно вплив змінного коефіцієнта на кінцеве розв’язання крайової задачі.

7.1.4. Система звичайних диференціальних рівнянь. Перший інтеграл. Однорідна система ЗДР

У довільній системі звичайних диференціальних рівнянь ми маємо декілька залежних змінних.

Так, загальний вид системи ЗДР n-го порядку представлений нижче.

$$\begin{aligned} y_1' &= f_1(y_1, y_2, \ldots y_n, x); \\ y_2' &= f_2(y_1, y_2, \ldots y_n, x); \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ y_n' &= f_n(y_1, y_2, \ldots y_n, x). \end{aligned} \qquad (7.6)$$

Функції являють собою, у загальному вигляді, аналітичні, однозначні функції своїх змінних.

Оператор, що дозволяє розв'язувати системи, має вигляд

$DSolve[\{eg1, eg2,...\},\{y1, y2,...\}, x]$

Оператор *DSolve[...]* може розв'язувати системи лінійних диференціальних рівнянь довільного (але кінцевого порядку) з постійними коефіцієнтами. Також можливий розв'язок системи ЗДР з постійними і змінними коефіцієнтами, але в загальному випадку одержати загальне розв'язання вдається не завжди.

У результаті розв'язку за вищезазначеним оператором одержуємо загальний розв'язок.

$In[10] := Quit[];$

$In[1] := DSolve[\{u'[x] = v[x], v'[x] = 2u[x] + v[x]\},\{u[x], v[x]\}, x]$

$$Out[1] = \{\{u(x) \to \frac{1}{3}e^{-x}(2+e^{3x})c_1 + \frac{1}{3}e^{-x}(-1+e^{3x})c_2, v(x) \to \frac{2}{3}e^{-x}(-1+e^{3x})c_1 + \frac{1}{3}e^{-x}(1+2e^{3x})c_2\}\}.$$

Розв'язок може бути представлений також через матричну експоненту.

$In[2] := MatrixExp[\{\{0,1\},\{1,0\}\}x].\{c1, c2\}$

$$Out[2] = \{\frac{1}{2}c2e^{-x}(-1+e^{2x}) + \frac{1}{2}c1e^{-x}(1+e^{2x}), \frac{1}{2}c1e^{-x}(-1+e^{2x}) + \frac{1}{2}c2e^{-x}(1+e^{2x})\}$$

Далі знайдемо розв'язок при відповідних початкових умовах.

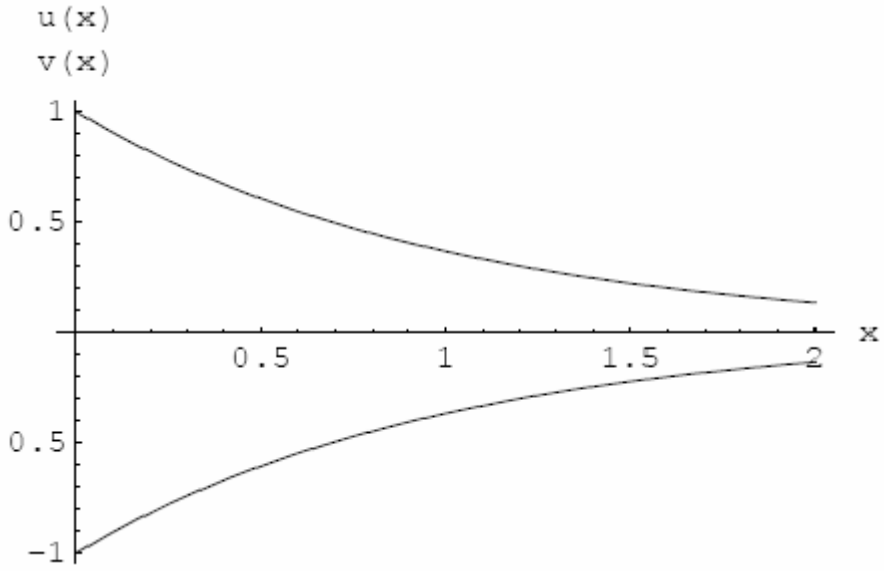
$In[3] := sol = DSolve[\{u'[x] = v[x], v'[x] = 2u[x] + v[x],$

$u[0] = 1, v[0] = -1\},\{u[x], v[x]\}, x]$

$Out[3] = \{\{u(x) \to e^{-x}, v(x) \to -e^{-x}\}\}$

Побудуємо графіки розв'язків системи

$In[5] := Plot[Evaluate[\{u[x], v[x]\}/.sol],\{x,0,2\}, AxesLabel \to \{"x","u(x) \setminus nv(x)"\}]$



$Out[5] = -Graphics-$

Розглянемо приклад розв'язку задачі класичної фізики про рух точкової маси в центральному полі сил. Для розв'язку запропонованої задачі скористаємося вбудованими в *Mathematica* пакетами.

```
In[24]:= Quit[] .
```

Спочатку завантажуємо пакети для розв'язання задачі в полярних координатах *(r, φ)*.

```
In[1]:=<< Calculus`VariationalMethods`
In[2]:=<< Calculus`VectorAnalysis`
```

Далі складаємо вираз для повної енергії руху у вигляді

```
In[3]:= e = m(r'[t]^2 + r[t]j'[t]^2)/2 - U[r[t]];
```

Тут *U(r(t))* потенціал центрального поля сил, що у цьому випадку задаємо в загальному вигляді. Очевидно, що рівняння руху частки в полярній площині має вигляд

```
In[4]:= EulerEguations[e,{j(t),r(t)},t]// Simplify
Out[4]= {m(r'(t)j'(t) + r(t)j''(t)) = 0, 2(U'(r(t)) + mr''(t)) = m j'(t)^2}
```

Система перших інтегралів руху частки в центральному потенційному полі сил має вигляд:

```
In[23]:= FirstIntegrals[e,{j(t),r(t)},t]
Out[23]= {FirstIntegral(j) → -mr(t)j'(t), FirstIntegral(t) → 1/2 (2U(r(t)) + m(r'(t)^2 + r(t)j'(t)^2))}
```

Фізичний зміст знайдених перших інтегралів можна знайти в традиційних курсах класичної фізики.

Якщо рух частки відбувається в центральному полі сил всесвітнього тяжіння, то, повторюючи попередні символьні обчислення, знаходимо:

- рівняння руху частки

```
In[5]:= e = m(r'[t]^2 + r[t]j'[t]^2)/2 - ymM / r[t];
In[6]:= EulerEguations[e,{j(t),r(t)},t]// Simplify
Out[6]= {m(r'(t)j'(t) + r(t)j''(t)) = 0, m(j'(t)^2 - 2r''(t) + 2My/r(t)^2) = 0}
```

- перші інтеграли руху

```
In[7]:= FirstIntegrals[e,{j(t),r(t)},t]
Out[7]= {FirstIntegral(j) → -mr(t)j'(t), FirstIntegral(t) → 1/2 m(r'(t)^2 + r(t)j'(t)^2 + 2My/r(t))}
```

який повністю збігається з відомими виразами з курсу класичної фізики.

7.1.5. Нелінійні диференціальні рівняння

За допомогою оператора *DSolve* можна розв'язувати і деякі нелінійні диференціальні рівняння. Наведемо деякі приклади, коли одержати такий розв'язок можливо.

$In[6] := Quit[]$.

Перший приклад, коли вдається одержати загальний розв'язок в символьному вигляді, наведений нижче,

$In[1] := DSolve[y'[x] = 1/y[x], y[x], x]$

$Out[1] = \{\{y(x) \to -\sqrt{2}\sqrt{x + c_1}\}, \{y(x) \to \sqrt{2}\sqrt{x + c_1}\}\}$,

а другий приклад має такий вигляд:

$In[2] := DSolve[y'[x] = y[x]\text{^}2 + a\text{^}2, y[x], x]$

$Out[2] = \{\{y(x) \to a\tan(a(x + c_1))\}\}$.

Наведемо приклад, коли за допомогою оператора *DSolve* вдається розв'язати рівняння в спеціальних функціях:

$In[5] := sol = DSolve[y''[x] = -Sin[y[x]], y[x], x]$

Solve::ifun : Inverse functions are being used by Solve, so some

solutions may not be found; use Reduce for complete solution information. More…

$$Out[5] = \{\{y(x) \to 2am(-\frac{1}{2}\sqrt{(c_1 + 2)(x + c_2)^2} \mid \frac{4}{c_1 + 2})\}, \{y(x) \to 2am(\frac{1}{2}\sqrt{(c_1 + 2)(x + c_2)^2} \mid \frac{4}{c_1 + 2})\}\}$$

Обчислимо значення отриманого розв'язку в точці x = 1.

$In[6] := Evaluate[sol /. x \to 1]$

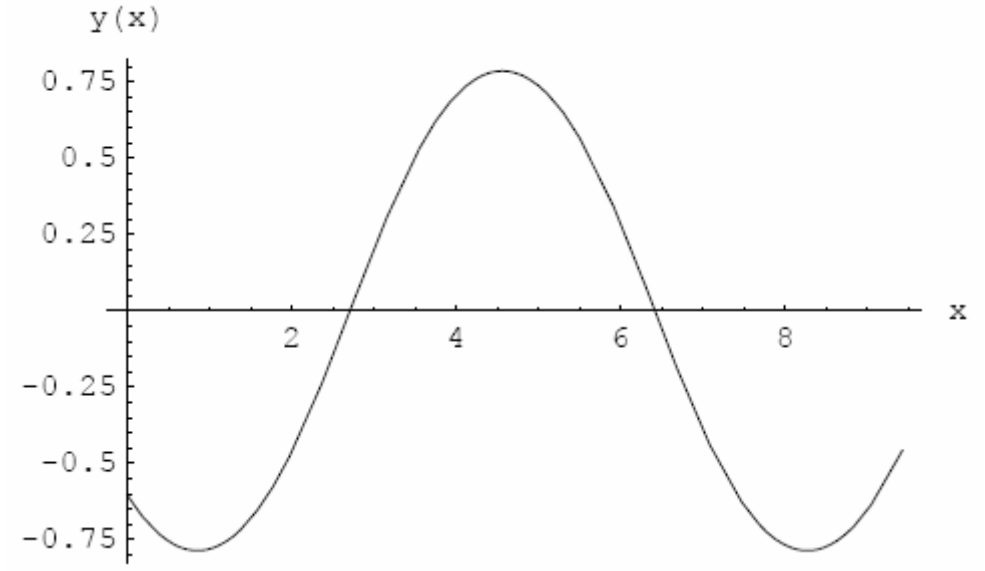
$$Out[6] = \{\{y(1) \to 2am(-\frac{1}{2}\sqrt{(c_1 + 2)(c_2 + 1)^2} \mid \frac{4}{c_1 + 2})\}, \{y(1) \to 2am(\frac{1}{2}\sqrt{(c_1 + 2)(c_2 + 1)^2} \mid \frac{4}{c_1 + 2})\}\}$$

Далі побудуємо графік отриманого раніше розв'язку, який представлений вище через функцію:

$In[9] := Part[y[x] /. sol, 1, 2]$

$$Out[9] = am(-\frac{1}{2}\sqrt{(c_1 + 2)(x + c_2)^2} \mid \frac{4}{c_1 + 2})$$

$In[11] := Plot[Evaluate[Part[y[x]/.sol,1,2]/.\{c_1 \to 0, c_2 \to 1\}], \{x,0,3Pi\}, AxesLabel \to \{"x","y(x)"\}]$



$Out[11] = -Graphics-$

7.2. Диференціальні рівняння в частинних похідних

Символьні і числові розв'язки диференціальних рівнянь у частинних похідних виконуються методами символьної алгебри.

7.2.1. Диференціальні рівняння в частинних похідних першого порядку

Розв'язування диференціальних рівнянь у частинних похідних у середовищі *Mathematica* обробляється за допомогою оператора

DSolve[*eg*, *u*[*x*, *y*], {*x*, *y*}]

(* розв'язок *eg*-квазілінійного диференціального рівняння в частинних похідних;

u - залежна змінна;

x,*y* - незалежна змінна*).

Пакет *«Calculus`PDSolve1»* дозволяє розв'язати ряд диференціальних рівнянь у частинних похідних першого порядку з постійними коефіцієнтами.

За допомогою названого оператора і пакета можна одержати, крім загальних розв'язків, і частинні розв'язки, записані через початкові і крайові умови.

Розглянемо найпростіші розв'язки.

$In[12] := Quit[];$

$In[1] :=<< Calculus`PDSolve1`$

$In[1] := eg1 = aD[u[x, y], x] + bD[u[x, y], y] + cu[x, y] = 0$

$Out[1] = cu(x, y) + bu^{(0,1)}(x, y) + au^{(1,0)}(x, y) = 0$

$In[2] := sol1 = DSolve[eg1, u[x, y], \{x, y\}] // Flatten // Simplify$

$Out[2] = \{u(x, y) \to e^{-\frac{cx}{a}} c_1[y - \frac{bx}{a}]\}$

Перевірка розв'язку представлена нижче.

$In[3] := (eg1 /. sol1 /. \{u^{(0,1)}(x, y) \to D[u(x, y) /. sol1, y], u^{(1,0)}(x, y) \to D[u(x, y) /. sol1, x]\}) // Simplify$

$Out[3] = True$

Далі розглянемо диференціальне рівняння в частинній змінній зі змінними коефіцієнтами, причому змінні коефіцієнти двох незалежних змінних (x, y).

$In[4] := eg2 = x\text{^}2D[u[x, y], x] - xyD[u[x, y], y] = -yu[x, y]$

$Out[4] = x^2 u^{(1,0)}(x, y) - xyu^{(0,1)}(x, y) = -yu(x, y)$

Розв'язок диференціального рівняння в частинних похідних *eg2*, має вигляд

$In[5] := sol2 = DSolve[eg2, u[x, y], \{x, y\}] // Flatten // Simplify$

$Out[5] = \{u(x, y) \to e^{\frac{y}{2x}} c_1[xy]\}$

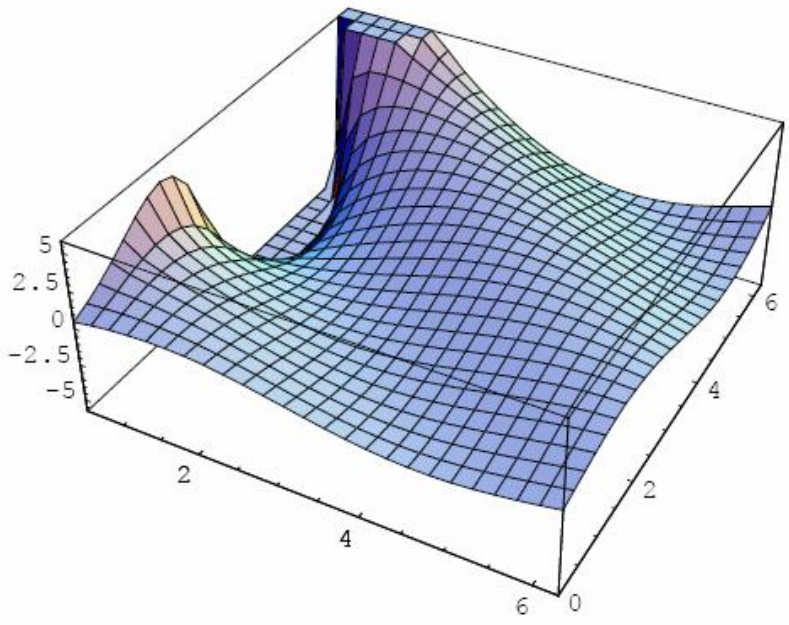
Частинний розв'язок запишемо в такий спосіб:

$In[6] := p = sol2 /. \{C[1][xy] \to Sin[x + y]\}$

$Out[6] = \{u(x, y) \to e^{\frac{y}{2x}} \sin(x + y)\}$

Графічне зображення частинного розв'язку на 3D плоті представлено нижче

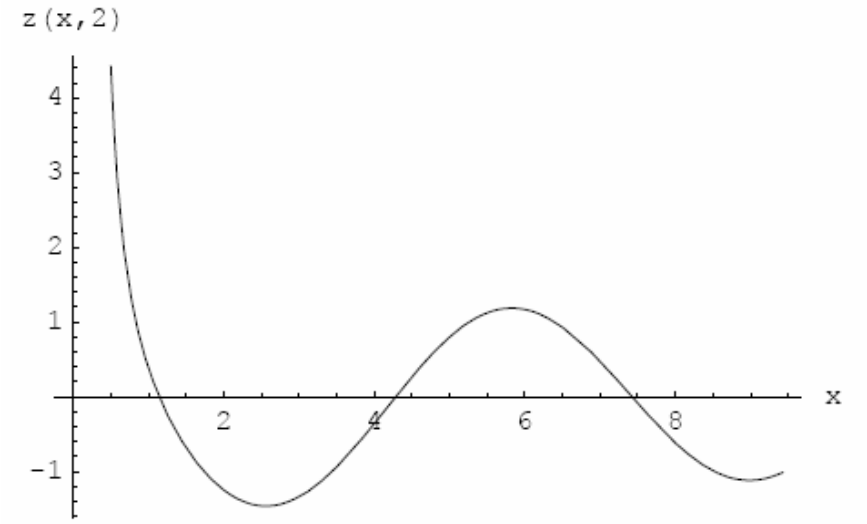
$In[7] := Plot3D[Evaluate[u[x, y] /. p], \{x, 0.5, 2Pi\}, \{y, 0, 2Pi\}]$



Out[7] = –*SurfaceGraphics* –

Перетин, або проекція частинного розв'язку в площині *(z, x)* має вигляд

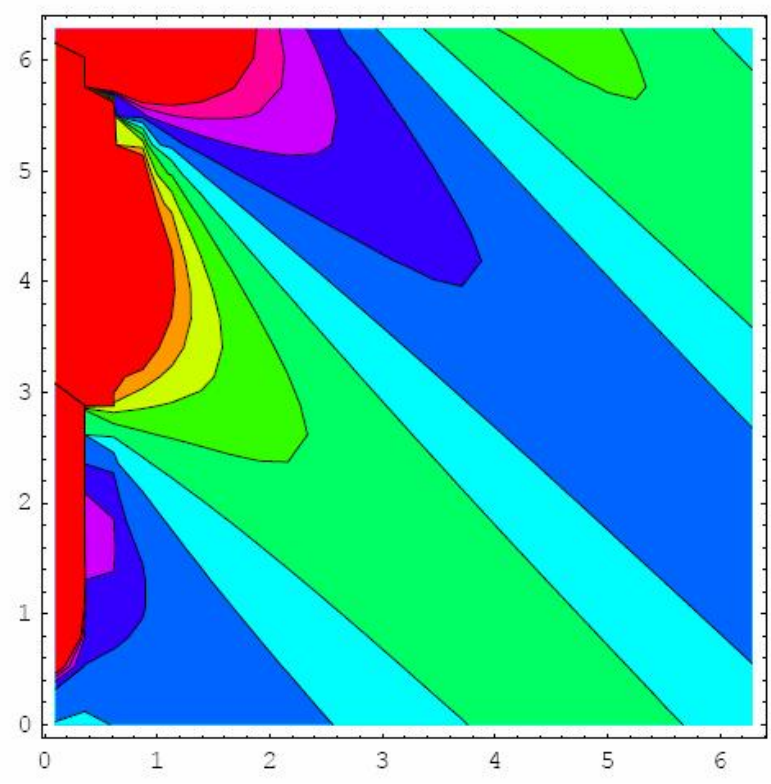
In[8] := *Plot*[*Evaluate*[*u*[*x*, *y*]/.*p*/.*y* → 2],{*x*,0.5,3*Pi*}, *AxesLabel* → {"*x*,""*z*(*x*,2)"}]



Out[8] = –*Graphics* –

Разом з тим, іноді зручно користуватися візуалізацією функцій двох змінних на контурному графіку. Приклад такої побудови функції двох незалежних змінних представлений нижче.

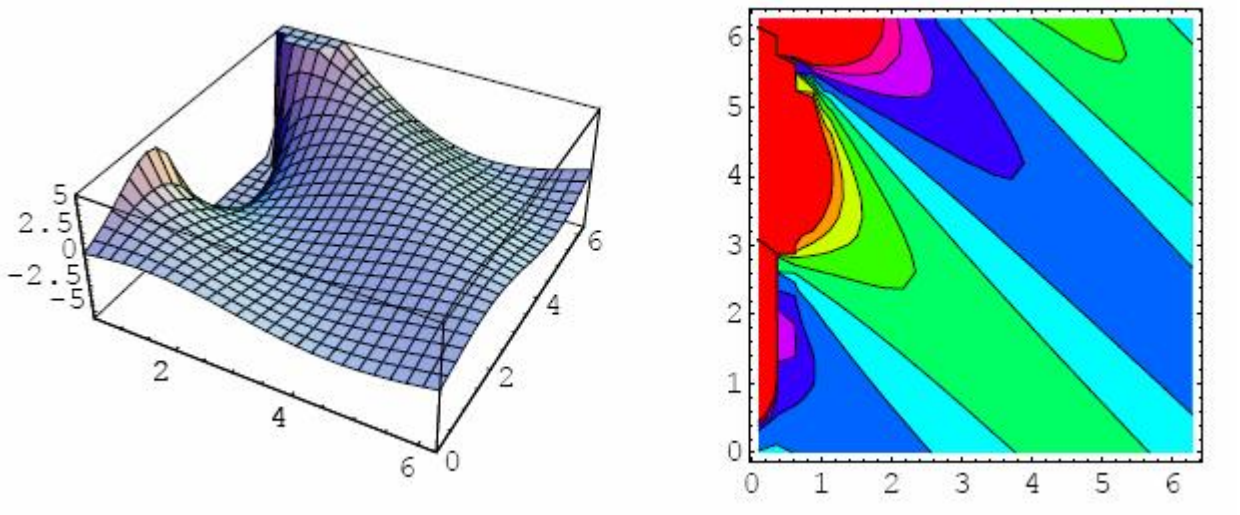
In[9] := *ContourPlot*[*Evaluate*[*u*[*x*, *y*]/.*p*],{*x*,0.1,2*Pi*},{*y*,0,2*Pi*}, *ColorFunction* → *Hue*]



Out[9] = –*ContourGraphics* –

Графіки розв’язків на спільному плоті представлені нижче.

```
In[11]:= Show[GraphicsArray[{Plot3D[Evaluate[u[x, y]/.p],{x,0.5,2Pi},{y,0,2Pi}],
          ContourPlot[Evaluate[u[x, y]/.p],{x,0.1,2Pi},{y,0,2Pi},ColorFunction→Hue]}]]
```




```
Out[11]= -GraphicsArray-
```

7.2.2. Диференціальні рівняння в частинних похідних другого порядку

Далі представимо символьні розв’язки ряду задач, які відносяться до класичних задач математичної фізики.

Символьні розв’язки, якщо такі буде можливо знайти, будуть визначатися різними методами комп'ютерної математики.

7.2.2.1. Метод перетворення Лапласа. Розглянемо розв’язки лінійних диференціальних рівнянь у частинних похідних другого порядку. Для знаходження розв’язків зазначених рівнянь скористаємося методом перетворення Лапласа.

Як перший приклад розглянемо розв’язок одновимірного параболічного рівняння при відповідних початкових і граничних умовах вигляду:

$$\frac{\partial u}{\partial t} = c\frac{\partial^2 u}{\partial x^2}; \qquad (7.7)$$

$$0 \le x \le l;$$

$$t \ge 0;$$

$$u(x,0) = dSin(2px/l);$$

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

Подібні початково-крайові задачі, які зводяться до розв’язування диференціального рівняння в частинних похідних параболічного типу, виникають у різних додатках математичної фізики при розв’язуванні задач:

- тепло- і масопереносу енергії і речовини в однорідних суцільних середовищах;
- дифузії;
- фільтрації однорідного флюїду в однорідному та ізотропному насиченому середовищі.

Вводимо рівняння (7.7) в *Mathematica:*

```
In[12]:= Quit[];
In[1]:= eg = D[u[x,t],t] - cD[u[x,t],x,x] = 0
```

$$Out[1] = u^{(0,1)}(x,t) - cu^{(2,0)}(x,t) = 0$$

Для символьного розв'язування початково-крайової задачі (7.7) застосуємо метод перетворення Лапласа за незалежного змінного ***t.***

```
In[2]:= IpTransform = LaplaceTransform[eg,t,s]
```

$$Out[2] = s(L_t[u(x,t)](s)) - c(L_t[u^{(2,0)}(x,t)](s)) - u(x,0) = 0$$

Початкові умови, також представимо в розв'язку перетворення Лапласа

```
In[3]:= functX = IpTransform /.{L_t[u(x,t)](s) → U[x], L_t[u^(2,0)(x,t)](s) - U''[x], u[x,0] → dSin[2px/l]}
```

$$Out[3] = -d\sin\left(\frac{2px}{l}\right) + sU(x) - cU''(x) = 0$$

Відповідне розв'язання крайової задачі має вигляд

```
In[4]:= solX = DSolve[{functX, U[0] = 0, U[l] = 0}, U[x], x] // Flatten // Simplify
```

$$Out[4] = \{U(x) \to \frac{dl^2 \sin(\frac{2px}{l})}{sl^2 + 4cp^2}\}$$

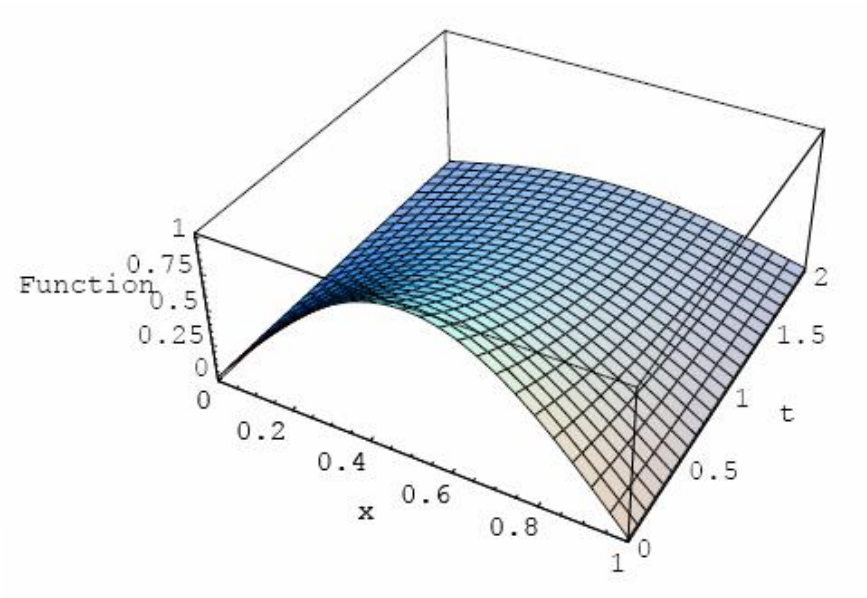
Далі застосовуємо зворотне перетворення Лапласа для одержання розв'язку вихідної початково-крайової задачі (7.7)

```
In[5]:= solTime = InverseLaplaceTransform[U[x]/.solX, s, t]
```

$$Out[5] = de^{-\frac{4cp^2t}{l^2}}\sin\left(\frac{2px}{l}\right)$$

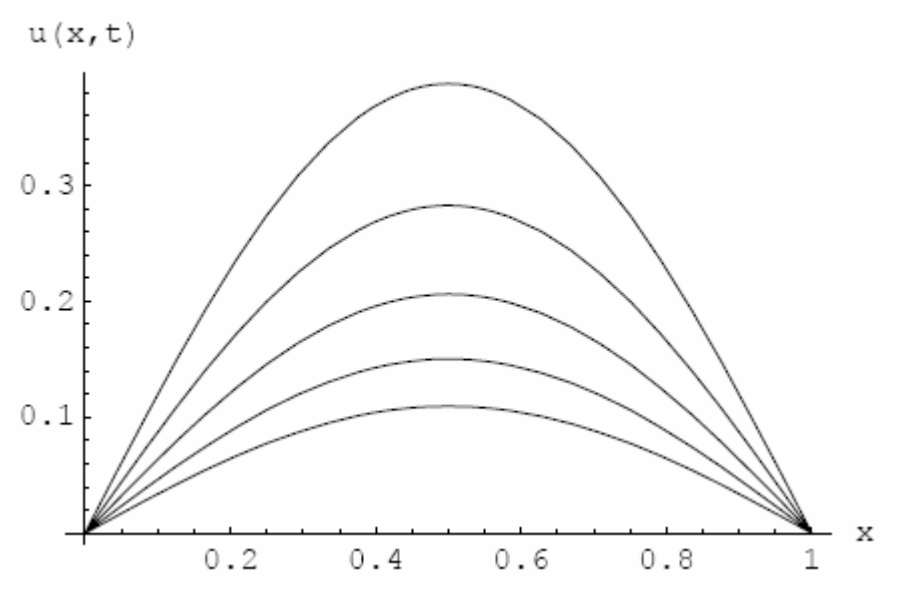
Потім представимо графічну реалізацію отриманого розв'язку

```
In[6]:= gr = Plot3D[solTime /.{d → 1, c → 0.08, l → 2},
            {x,0,1},{t,0,2}, AxesLabel → {"x","t","Function"}];
```



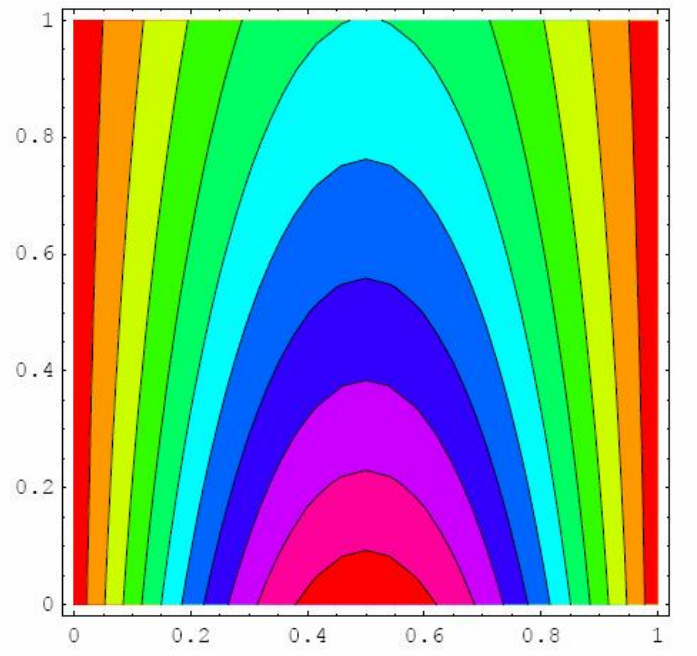
Графік проекцій розв'язку на площину *(u,x)* при дискретній варіації часу *t=1+i0,2* при *(i = 1,2,...)* представлений нижче у вигляді сімейства графіків.

```
In[8]:= Plot[Evaluate[Table[solTime /. {t → 1 + i.2} /. {d → 1, c → 0.08, l → 2}, {i,1,10,2}]],
          {x,0,1}, AxesLabel → {"x","u(x,t)"}];
```



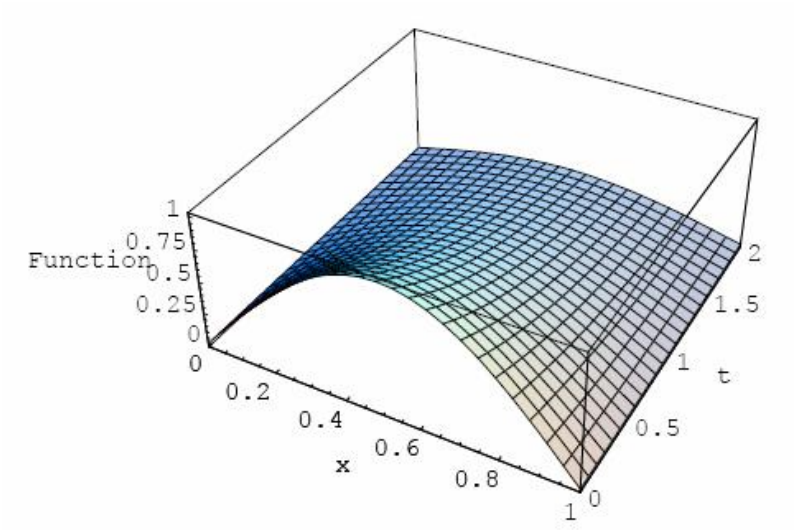
Візуалізація розв'язку початково-крайової задачі (7.7) при заданих початкових і крайових умовах на контурному графіку представлена нижче.

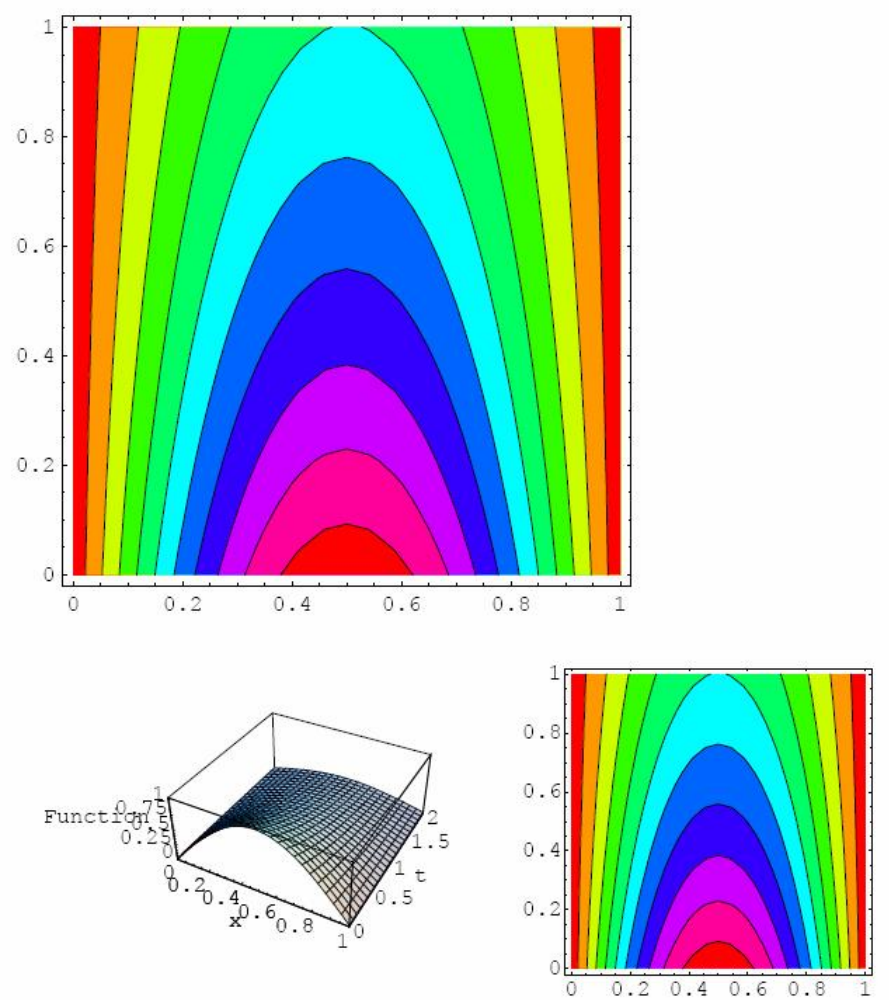
```
In[22]:= ContourPlot[solTime /. {d → 1, c → 0.08, l → 2}, {x,0,1}, {t,0,1}, ColorFunction → Hue];
```



Інші графічні подання розв'язків початково-крайової задачі (7.7), які зручно представити на спільному плоті, зображено нижче.

```
In[24]:= Show[GraphicsArray[{Plot3D[solTime /. {d → 1, c → 0.08, l → 2},
                {x,0,1}, {t,0,2}, AxesLabel → {"x","t","Function"}],
        ContourPlot[solTime /. {d → 1, c → 0.08, l → 2}, {x,0,1}, {t,0,1}, ColorFunction → Hue]}]];
```





Далі розглянемо гіперболічне рівняння (хвильове рівняння) вигляду:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}; \qquad (7.8)$$

$$0 \le x \le l;$$

$$t \ge 0;$$

$$\frac{\partial u(x,0)}{\partial t} = 0;$$

$$u(x,0) = dSin(2px/l);$$

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

Так само як і раніше, вводимо рівняння в *Mathematica:*

$$In[9] := Quit[];$$

$$In[1] := eg1 = D[u[x,t],t,t] - c\text{^}2D[u[x,t],x,x] = 0$$

$$Out[1] = u^{(0,2)}(x,t) - c^2 u^{(2,0)}(x,t) = 0$$

Вихідне диференціальне рівняння, до якого застосували перетворення Лапласа, має вигляд

$In[2] := IpTransform1 = LaplaceTransform[eg1, t, s]$

$Out[2] = -(L_t[u^{(2,0)}(x,t)](s))c^2 + s^2(L_t[u(x,t)](s)) - su(x,0) - u^{(0,1)}(x,0) = 0$

Початкові умови також представимо в розв'язку перетворення Лапласа.

$In[3] := functXx = IpTransform1/.\{L_t[u(x,t)](s) \to U[x],$

$L_t[u^{(2,0)}(x,t)](s) \to U''[x], u[x,0] \to dSin[2Pix/l], Derivative[0,1][u][x,0] \to 0\}$

$$Out[3] = -U''(x)c^2 - ds\sin\left(\frac{2px}{l}\right) + s^2U(x) = 0$$

Розв'язки крайової задачі представлено нижче:

$In[4] := solXx = DSolve[\{functXx, U[0] = 0, U[l] = 0\}, U[x], x] // Simplify // Flatten$

$$Out[4] = \{U(x) \to \frac{dl^2 s\sin(\frac{2px}{l})}{4p^2c^2 + l^2s^2}\}$$

Далі застосовуємо зворотне перетворення Лапласа для знаходження вихідної початково-крайової задачі (7.8)

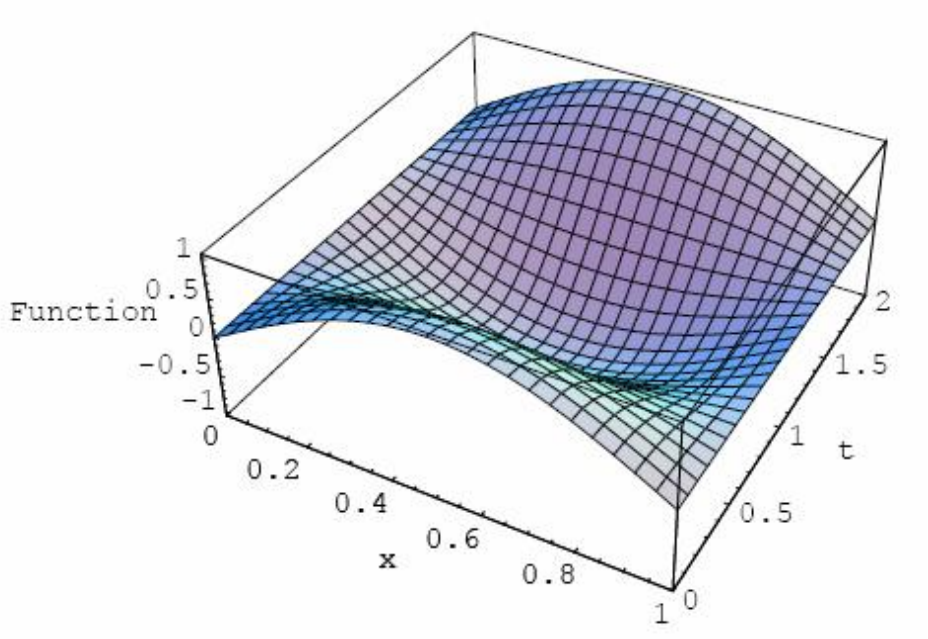
$In[5] := solHypTime = InverseLaplaseTransform[U[x]/.solXx, s, t]$

$$Out[5] = d\cos\left(\frac{2cpt}{l}\right)\sin\left(\frac{2px}{l}\right)$$

Потім представимо графічну реалізацію отриманого розв'язку.

$In[6] := gr = Plot3D[solHypTime/.\{d \to 1, c \to 1, l \to 2\},$

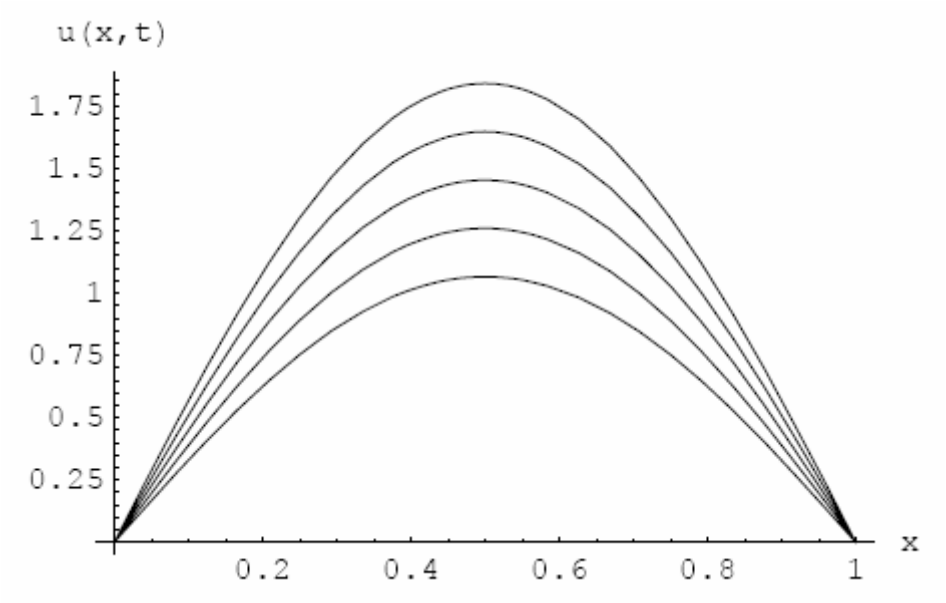
$\{x,0,1\}, \{t,0,2\}, AxesLabel \to \{"x","t","Function"\}]$



$Out[6] = -SurfaceGraphics-$

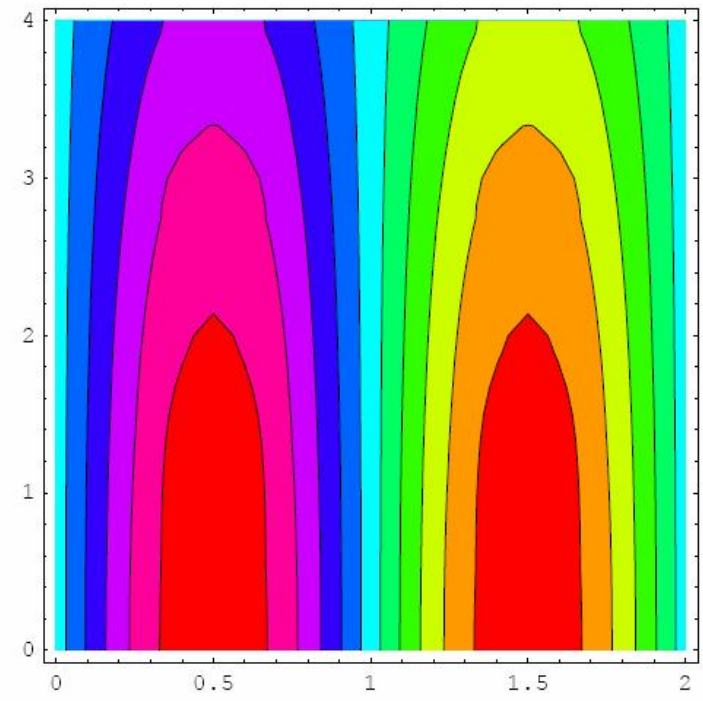
Графік проекції розв'язку на площину *(u,x)* при *t=1*, але у випадку рівномірного зростання амплітуди початкових збурювань представлений нижче у вигляді сімейства графіків.

$In[7] := Plot[Evaluate[Table[solHypTime /. t \to 1 /. \{d \to 1 + i.1, c \to 0.08, l \to 2\}, \{i,1,10,2\}]], \{x,0,1\}, AxesL$



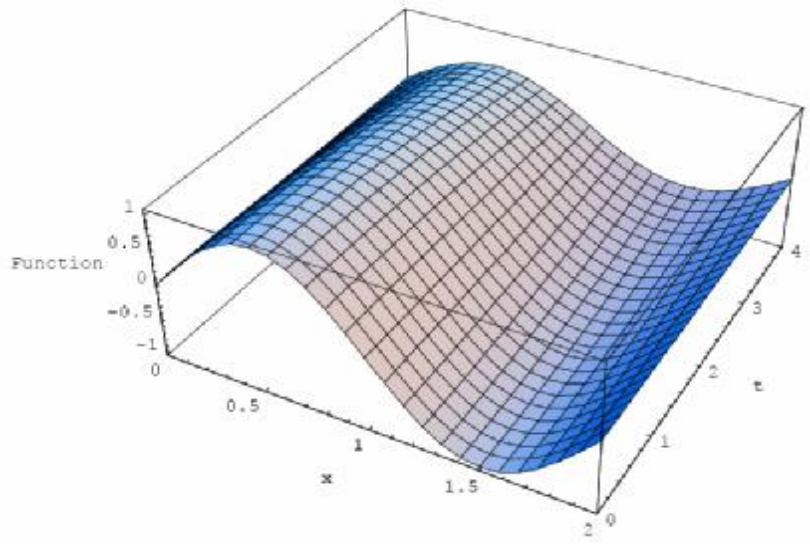
Візуалізація розв'язку початково-крайової задачі (7.8) при заданих початкових і крайових умовах на контурному графіку представлена нижче.

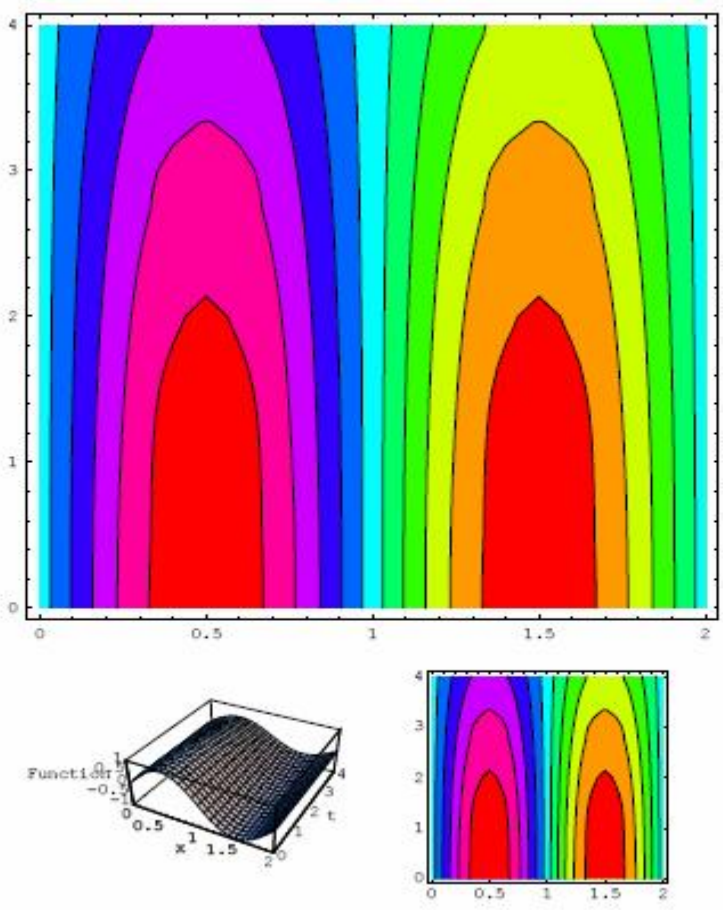
$In[12] := ContourPlot[solHypTime /. \{d \to 1, c \to 0.08, l \to 2\}, \{x,0,2\}, \{t,0,4\}, ColorFunction \to Hue];$



Інші графічні зображення розв'язку задачі (7.8), які зручно представити на спільному плоті, представлені нижче.

```
In[14]:= Show[GraphicsArray[{Plot3D[solHypTime /. {d → 1, c → 0.08, l → 2},
          {x,0,2}, {t,0,4}, AxesLabel → {"x","t","Function"}], ContourPlot[
          solHypTime /. {d → 1, c → 0.08, l → 2}, {x,0,2}, {t,0,4}, ColorFunction → Hue]}]];
```





Як інший приклад розв'язування хвильового рівняння розглянемо приклад розв'язування задачі про поширення хвилі збурювань в однорідному середовищі при імпульсному збурюванні засобів у початковий момент часу. Як відомо, початково-крайова задача у цьому випадку буде формуватися із залученням спеціальної функції δ-функції Дірака.

Сформована початково-крайова задача в цьому випадку має вигляд

$$\begin{aligned}
&\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2};\\
&0 \le x \le l;\\
&t \ge 0;\\
&\frac{\partial u(x,0)}{\partial t} = \mathrm{dDiracDelta}(x);\\
&u(x,0) = 0;\\
&u(0,t) = u(l,t).
\end{aligned} \qquad (7.9)$$

Як і у попередньому випадку для її розв'язування застосуємо перетворення Лапласа. Коди, які розв'язують задачу (7.9), представлені нижче.

In[11] := *Quit*[].

Вихідне диференціальне рівняння, до якого застосовуємо перетворення Лапласа в кодах, має вигляд

In[1] := *eg*1 = *D*[*u*[*x*, *t*], *t*, *t*] − *c*^2*D*[*u*[*x*, *t*], *x*, *x*] = 0

$$Out[1] = u^{(0,2)}(x,t) - c^2 u^{(2,0)}(x,t) = 0,$$

а його зображення представлене нижче:

In[3] := *IpTransform*1 = *LaplaceTransform*[*eg*1, *t*, *s*]

$$Out[3] = -(L_t[u^{(2,0)}(x,t)](s))c^2 + s^2(L_t[u(x,t)](s)) - su(x,0) - u^{(0,1)}(x,0) = 0.$$

Початкові умови також представимо в розв'язку перетворення Лапласа:

In[4] := *functXx* = *IpTransform*1 /. {$L_t[u(x,t)](s) \to U[x]$,

$$L_t[u^{(2,0)}(x,t)](s) \to U''[x], u[x,0] \to 0, Derivative[0,1][u][x,0] \to DiracDelta[x]\}$$

$$Out[4] = -U''(x)c^2 - d(x) + s^2 U(x) = 0$$

Розв'язок крайової задачі представлено нижче:

In[6] := *solXx* = *DSolve*[{*functXx*, *U*[0] = 0, *U*[*l*] = 0}, *U*[*x*], *x*] // *Simplify* // *Flatten*

$$Out[6] = \{U(x) \to \frac{e^{-\frac{sx}{c}}(-1 + e^{\frac{2sx}{c}})(\Theta(l) - \Theta(x))}{2cs}\}$$

Далі застосовуємо зворотне перетворення Лапласа для знаходження вихідної початково-крайової хвильової задачі.

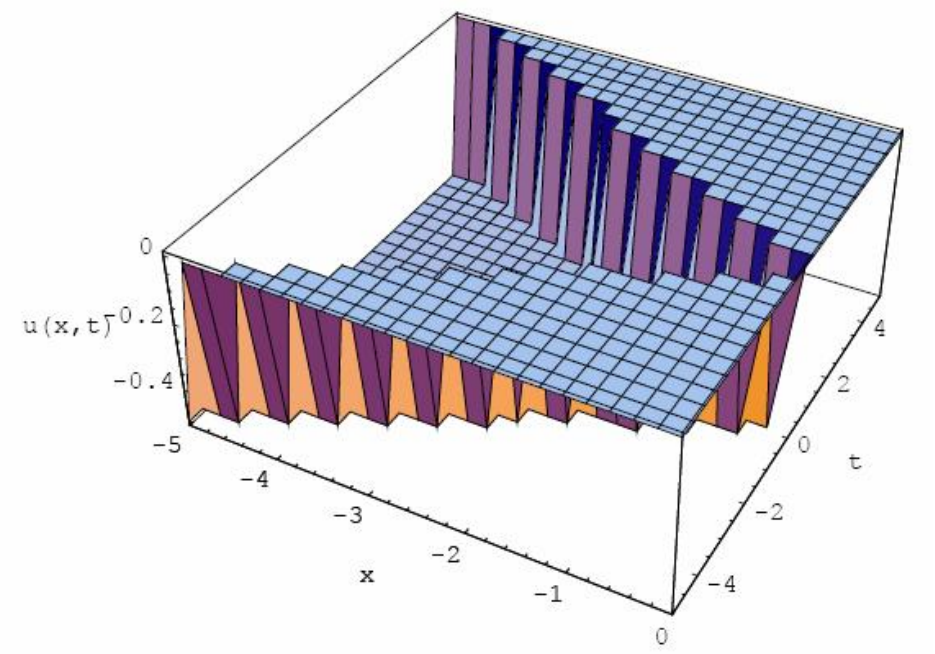
In[7] := *solHypTime* = *InverseLaplaceTransform*[*U*[*x*] /. *solXx*, *s*, *t*]

$$Out[7] = \frac{(\Theta(l) - \Theta(x))(\Theta(t + \frac{x}{c}) - \Theta(t - \frac{x}{c}))}{2c}$$

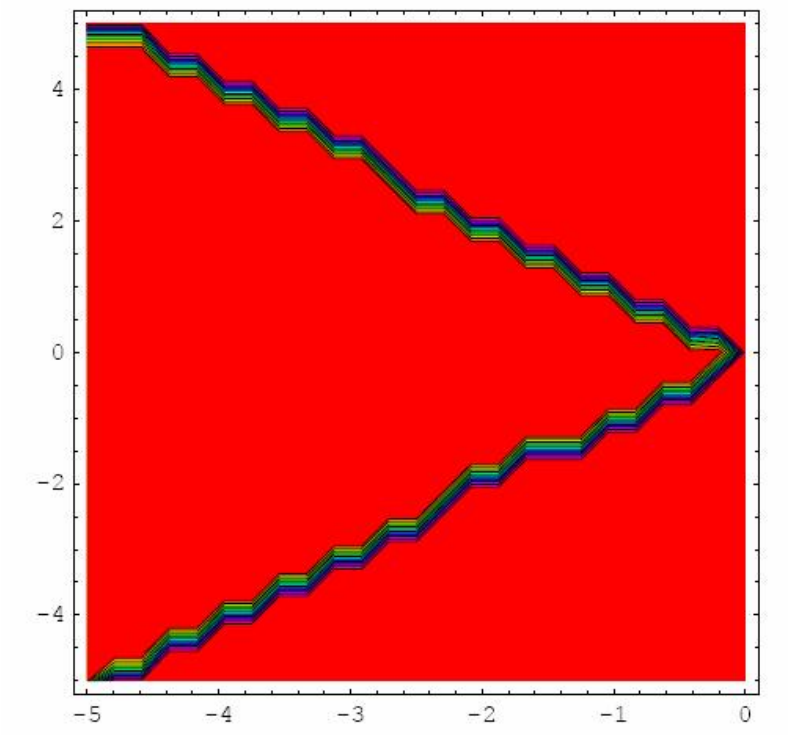
Очевидно, що символьне розв'язання описує поширення дивергентної і конвергентної хвилі в просторі параметрів зі швидкістю c.

Потім представимо графічну реалізацію отриманого розв'язку хвильової задачі.

```
In[12]:= gr = Plot3D[Evaluate[solHypTime /. {d → 10, c → 1, l → 2}],
          {x, -5, 0}, {t, -5, 5}, AxesLabel → {"x", "t", "\nu(x, t)"}];
```




```
In[11]:= gr = ContourPlot[Evaluate[solHypTime /. {d → 10, c → 1, l → 2}],
          {x, -5, 0}, {t, -5, 5}, AxesLabel → {"x", "t"}, ColorFunction → Hue];
```




```
Out[11]= -ContourGraphics-
```

7.2.2.2. Розв'язання неоднорідного хвильового рівняння. Як інший приклад розв'язування хвильового рівняння розглянемо приклад розв'язку задачі про поширення хвилі збурювань в однорідному середовищі, які поширюються внаслідок зовнішніх збурювань. У цьому випадку збурювання в однорідному середовищі описуються неоднорідною початково-крайовою задачею, формулювання якої подано нижче.

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} + dSin(2px/l)Sin(2pt); \quad (7.10)$$
$$0 \le x \le l;$$
$$t \ge 0;$$
$$\frac{\partial u(x,0)}{\partial t} = 0;$$
$$u(x,0) = 0;$$
$$u(0,t) = u(l,t).$$

Коди для символьного розв'язування мають вигляд

$In[13] := Quit[];$

$In[82] := eg1 = D[u[x,t],t,t] - c^2 D[u[x,t],x,x] = dSin[2px/l]Sin[2pt]$

$$Out[82] = u^{(0,2)}(x,t) - c^2 u^{(2,0)}(x,t) = d\sin(2pt)\sin\left(\frac{2px}{l}\right)$$

Вихідне диференціальне рівняння, до якого застосували перетворення Лапласа, має вигляд

$In[83] := IpTransfom1 = LaplaceTransform[eg1, t, s]$

$$Out[83] = -(L_t[u^{(2,0)}(x,t)](s))c^2 + s^2(L_t[u(x,t)](s)) - su(x,0) - u^{(0,1)}(x,0) = \frac{2dp\ \sin(\frac{2px}{l})}{s^2 + 4p^2}$$

Початкові умови також представимо в розв'язку перетворення Лапласа.

$In[84] := functXx =$

$$IpTransform1/.\{L_t[u(x,t)](s) \to U[x], L_t[u^{(2,0)}(x,t)](s) \to U''[x], u(x,0) \to 0, u^{(0,1)}(x,0) \to 0\}$$

$$Out[84] = s^2 U(x) - c^2 U''(x) = \frac{2dp\ \sin(\frac{2px}{l})}{s^2 + 4p^2}$$

Розв'язок крайової задачі представлено нижче

$In[85] := solXx = DSolve[\{functXx, U[0] = 0, U[l] = 0\}, U[x], x] // Simplify // PowerExpand // Flatten$

$$Out[85] = \{U(x) \to \frac{2dl^2 p\ \sin(\frac{2px}{l})}{(s^2 + 4p^2)(4p^2c^2 + l^2 s^2)}\}.$$

Далі застосовуємо зворотне перетворення Лапласа для знаходження вихідної початково-крайової хвильової задачі.

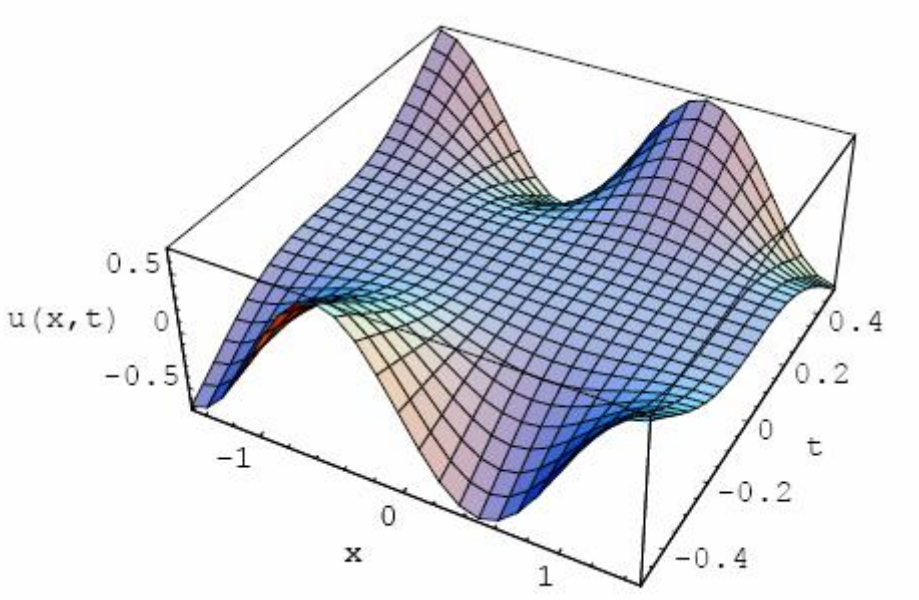
$$In[91] := solHypTime = InverseLaplaceTransform[U[x]/.solXx, s, t]// Simplify$$

$$Out[91] = \frac{dl^2(c\sin(2pt) - l\sin(\frac{2cpt}{l}))\sin(\frac{2px}{l})}{4(c^3 - cl^2)p^2}.$$

Очевидно, що символьний розв'язок описує стоячі хвилі в обмеженому суцільному і однорідному середовищі, а їхній розподіл в обмеженому середовищі розглянемо на відповідних графіках.

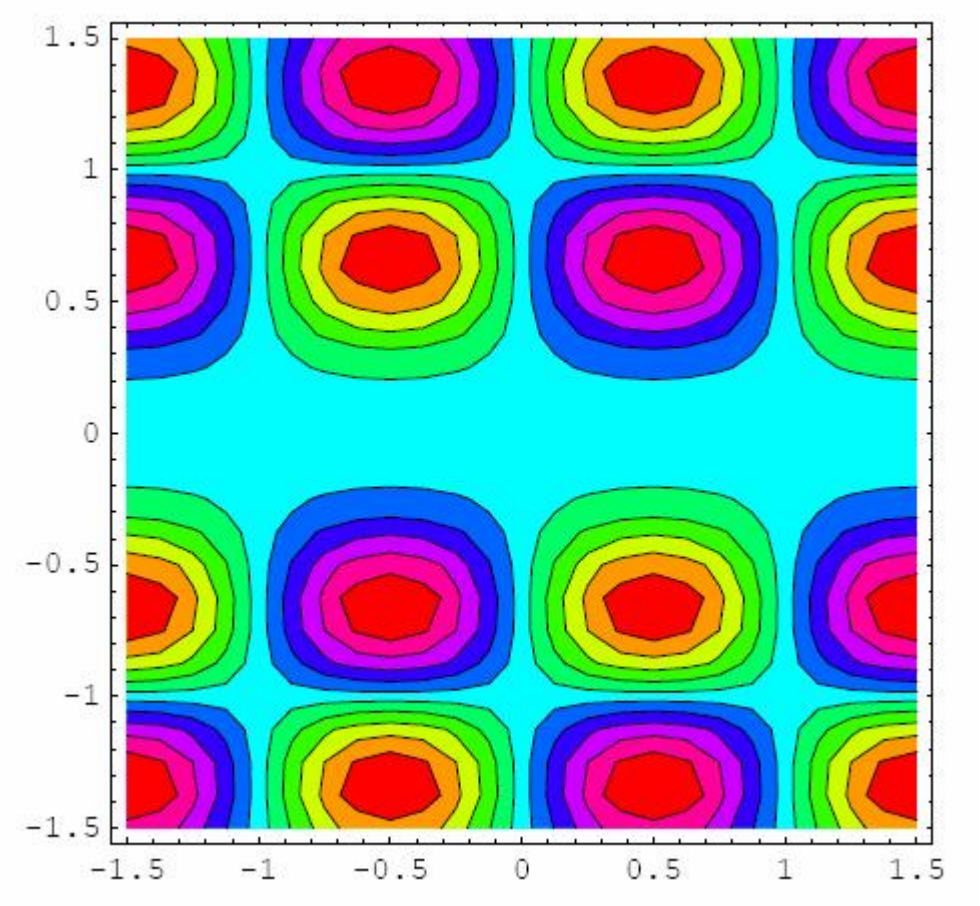
Спочатку представимо графічну реалізацію отриманого розв'язку хвильової задачі на плоті 3D.

```
In[95]:= gr = Plot3D[Evaluate[solHypTime /. {d → 10, c → 1, l → 2}],
          {x, -1.5, 1.5}, {t, -.5, .5}, AxesLabel → {"x", "t", "\nu(x,t)"}];
```



Розподіл хвильових збурювань у вигляді стоячих хвиль на контурному графіку представлено нижче.

```
In[93]:= gr = ContourPlot[Evaluate[solHypTime /. {d → 10, c → 1, l → 2}],
          {x, -1.5, 1.5}, {t, -1.5, 1.5}, AxesLabel → {"x", "t"}, ColorFunction → Hue];
```



Різні кольори ілюструють розподіл пучностей і западин хвильових рухів в однорідному середовищі, що є зручною візуалізацією хвильових збурювань, наприклад, в оптиці.

Тема 8

ЧИСЛОВІ РОЗВ'ЯЗКИ РІВНЯНЬ У ЧАСТИННИХ ПОХІДНИХ ДРУГОГО ПОРЯДКУ

8.1. Алгоритм розв'язку

Як алгоритм розв'язку початково-крайової задачі обраний метод кінцевих елементів. Суть якого викладена докладно в [1,2].

Метод скінчених різниць при розв'язанні задач, які зводяться до диференціальних рівнянь у частинних похідних.

8.1.1. Параболічне рівняння

Розглянемо таку початково-крайову задачу, яка відноситься до класу параболічних.

$$\begin{aligned}
&\frac{\partial u}{\partial t} = c\frac{\partial^2 u}{\partial x^2} + f(x,t);\\
&x_0 \le x \le x_1;\\
&t_0 \le t \le t_1;\\
&u(x,t_0) = I(x);\\
&u(x_0,t) = B_0(t);\\
&u(x_1,t) = B_1(t).
\end{aligned} \tag{8.1}$$

Розділимо інтервал (x_1, x_2) на n_x субінтервалів довжини h_x, інтервал (t_0, t_1)- на n_t субінтервалів довжини h_t. У такий спосіб ми одержимо
$(n_x+1)*(n_t+1)$ точок на площині (x, t).

Позначимо величину функції в "точці" через $u[i,j]$ і апроксимуємо частинні похідні кінцевими різностями. В результаті одержимо такі формули для кінцевих різностей.

$$\begin{aligned}
&\frac{1}{h_t}(u_{i,j+1} - u_{i,j}) - \frac{c}{h_x{}^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) = f_{i,j};\\
&\frac{1}{h_t}(u_{i,j+1} - u_{i,j}) - \frac{c}{h_x{}^2}(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}) = f_{i,j+1.}
\end{aligned} \tag{8.2}$$

Перші співвідношення розв'язують поставлену задачу спочатку на стадії i, а потім на $j+1$. Наступна формула розв'язує початково-крайову задачу по рекурсивному співвідношенню для $u[i,j+1]$.

Якщо об'єднати обидва методи, то доходимо до такого співвідношення:

$$au_{i-1,j} + 2(1-a)u_{i,j+1} - au_{i+1,j+1} = au_{i-1,j} + 2(1-a)u_{i,j} + au_{i+1,j} + h_t(f_{i,j} + f_{i,j+1}); \quad (8.3)$$

$a = c$.

$\frac{h_t}{h_x^2}$.

Математична задача є задачею розв'язання трикутної системи лінійних алгебраїчних рівнянь. Кожна система має *(n_x-1)* змінну і в остаточному підсумку відповідно необхідно розв'язати n_t таких систем.

У результаті розв'язання одержуємо модуль зі списком у вигляді *{...} (List)* у такій формі:

$$\{\{x_0, t_0, u(0,0)\}, \{x_0, t_1, u(0,1)\}, ...\}. \quad (8.4)$$

*Я*к приклад розглянемо числове розв'язання початково-крайової задачі для параболічного рівняння (8.1). Модуль, що дозволяє числове розв'язання початково-крайової задачі, представлений нижче.

```
In[1]:=<< LinearAlgebra 'Tridiagonal'
In[2]:= parabolicfd[{c_, F_, i0_, b0_, b1_}, {x_, t_},
        {x0_, xl_, t0_, tl_, nx_, nt_}]:= Module[
        {hx, ht, a, al, xx, tt, u, rhs, subdiag, maindiag},
          hx = N[(xl - x0) / nx];
          ht = N[(tl - t0) / nt];

          a = c ht / hx^2,   al = 2    (1 - a);

          Do[xx[i] = x0 + i  hx, {i,0, nx}];
          Do[tt[j] = t0 + j  ht, {j,0, nt}];

          Do[FF[i, j] = N[F/.{x - f xx[i], t - f tt[j]}], {i,0, nx}, {j,0, nt}];
          Do[u[i,0]= N[i 0 / .x - f xx[i]], {i,0, nx}];

          Do[u[0, j + l] = N[b0 / .t - f tt[j + l]];

          u[nx, j + l] = N[bl / .t - f tt[j + 1]];

          rhs = Table[a  (u[i - l, j] + u[i + l, j]) + al  u[i, j] +
          ht  (FF[i, j] + FF[i, j + l]), {i, nx - l}];
```

```
rhs[[1      ]]  + = a u[0, j + 1];

rhs[[nx - 1]] + = a u[nx, j + l];

subdiag  = Table [-a,{nx - 2}];

maindiag  = Table [2 + 2  a,{nx - l}];

Evaluate [Table [u [i, j + 1], {i, nx - l}]] =

TridiagonalSolve [subdiag, maindiag, subdiag,
rhs , {j, 0, nt - l}];

Flatten [Table [{xx [i] , tt [j] , u[i, j]},
{i, 0, nx} , {j, 0, nt}] , 1]]
```

Розв’язання задачі при вихідних початкових і граничних умовах запишемо в такий спосіб. Тут же оператором *Timing[...]* вказується час розв’язання задачі.

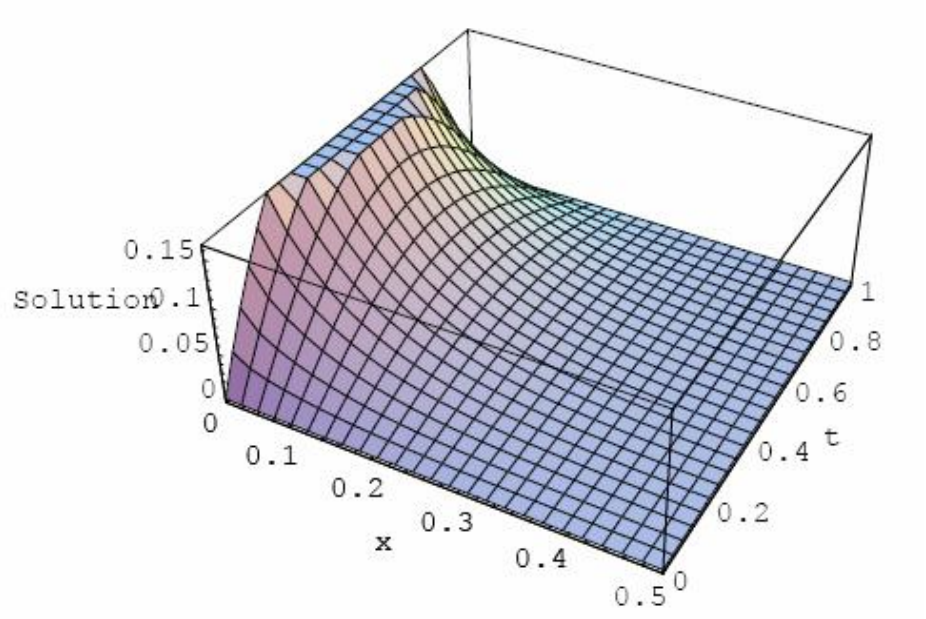
```
In[3] := uappr = parabolicfd[{1,0, x(1 – x),0,0}, {x, t},
        {0,1,0,0.5,60,60}]; // Timing
Out[3] = {0.172Second, Null}
```

Після числового розв’язку проводимо інтерполяцію і здійснюємо візуалізацію здобутого розв’язку.

```
In[4] := int = Interpolation[uappr].
```

Графічне зображення розв’язку на плоті 3D має вигляд

```
In[5] := Plot3D[int[x, t], {t,0,0.5}, {x,0,1},
         AxesLabel– f {"x","t","Solution"}]
```

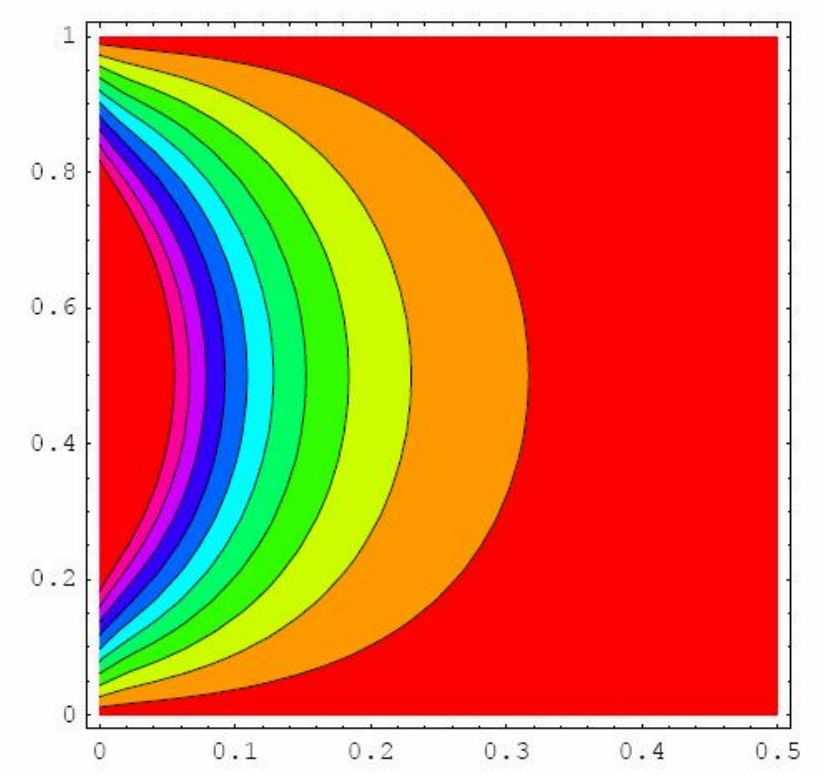


Out[5] = –*SurfaceGraphics* –

Те ж розв’язання на контурному графіку представлено нижче.

In[6] := *ContourPlot*[int[*x*, *t*],{*t*,0,0.5},{*x*,0,1},

AxesLabel– **f** {"*x*","*t*"},*ColorFunction* → *Hue*];



Таким чином, представлений вище модуль можна використати у вигляді ядра користувальницького пакета для розв’язування одновимірних початково-крайових задач параболічного типу.

8.1.2. Еліптичне рівняння

Розглянемо розв'язок еліптичного рівняння в частинних похідних такого вигляду:

$$\begin{aligned}
&\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=f(x,y);\\
&x_0 \le x \le x_1;\\
&y_0 \le y \le y_1;\\
&u(x_0,y)=B_0(y);\\
&u(x_1,y)=B_1(t).
\end{aligned} \tag{8.5}$$

Застосовуючи як і у попередньому випадку розбивку області інтегрування на "точки" розміру $(n_x+1)*(n_y+1)$, переходимо від диференціального рівняння в частинних похідних до алгебраїчних рівнянь у скінчених різницях. Вигляд останніх такий

$$\frac{1}{h_x^{\,2}}(u_{i+1,j}-2u_{i,j}+u_{i-1,j})+\frac{1}{h_y^{\,2}}(u_{i,j+1}-2u_{i,j}+u_{i,j-1})=F_{i,j}. \tag{8.6}$$

У підсумку одержуємо систему лінійних рівнянь із матрицею розміру $(n_x-1)*(n_y-1)$, розв'язок якої апроксимує розв'язку диференціального рівняння в частинних похідних.

Модуль у середовищі *Mathematica,* який дозволяє виконати представлену вище процедуру, записуємо так:

```
In[7]:= ellipticfd[{F_, b0_, bl_, b2_, b3_}, {x_, y_},
          {x0_, xl_, y0_, yl_, nx_, ny_, eps_, maxit_}]:= Module[
          {hx, hy, a, it, xx, yy, FF, u, unew, omega, omegal, const},

          hx = N[(xl - x0) / nx];
          hy = N[(yl - y0) / ny];
          a = (hx / hy) ^2; hx2 = hx ^2; it = 0;

          Do[xx[i] = x0 + i  hx, {i,0, nx}];
          Do[yy[j] = y0 + j  hy, {j,0, ny}];

          Do[FF[i, j] = hx2  N[F /. {x - f  xx[i], y - f  yy[j]}];
          u[i, j] = unew[i, j] = 0., {i, 0, nx}, {j, 0, ny}];
```

```
Do[u[0, j] = N[b0 /. y - f yy[j]];
u[nx, j] = N[b1 /. y - f yy[j]], {j, 0, ny}];

Do[u[i, 0] = N[b2 /. x - f xx[i]];
u[i, ny] = N[b3 /. x - f xx[i]], {i, 0, nx - l}];

omega = N[4 / (2 + Sqrt[4 - (Cos[Pi / nx] + Cos[Pi / ny])^2])];
omegal = l - omega; const = 0.5 omega / (a + 1);

Do[Do[unew[i, j] = const (unew[i - 1, j] + u[i + l, j] +
a (unew[i, j - 1] + u[i, j + 1]) - FF[i, j]) + omega1 u[i, j],
{i, nx - l}, {j, ny - l}];
If[Max[Table[Abs[u[i, j] - unew[i, j]],
{j, ny - 1}]] >= eps && it < maxit,

Do[u[i, j] = unew[i, j], {i, nx - 1}, {j, ny - 1}]; it++,
Break],
{maxit}];

Print[" = ", maxit];
Flatten[Table[{xx[i], yy[j], unew[i, j]}, {i, 0, nx},
{j, 0, ny}
], 1]]
```

Як приклад розглянемо розв'язок такої крайової задачі:

$$\begin{aligned} &u_{x,x} + u_{y,y} = 10x^2 y; \\ &u(0, y) = u(1, y) = 0; \\ &u(x,0) = u(x,1) = 0. \end{aligned} \qquad (8.7)$$

Для розв'язання задачі використаємо розбивку на 10 субінтервалів по кожній з координат. Відповідне розв'язання запишеться у такий спосіб:

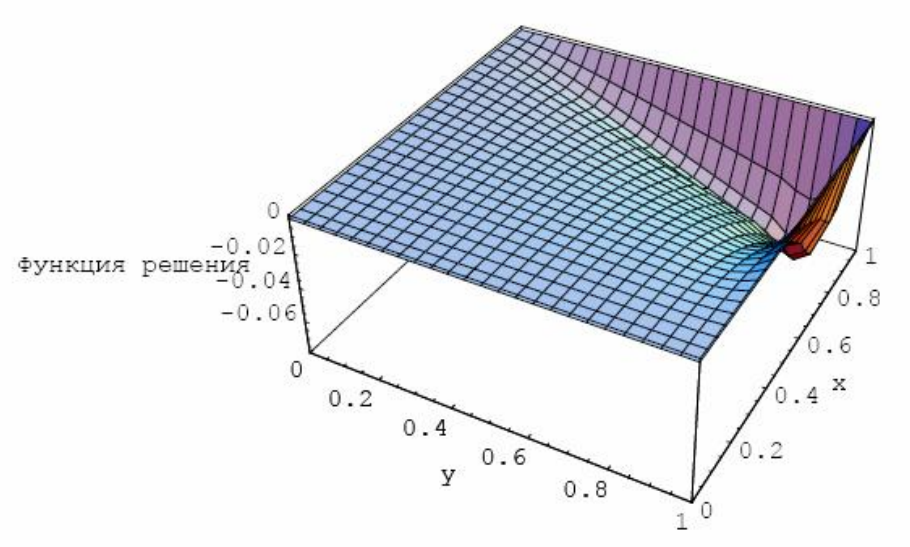
```
In[8] := uappr = ellipticfd[{10x^2y, 0, 0, 0, 0}, {x, y},
        {0,1,0,1,10,10,0.0000001,100}]; // Timing

        кількість ітерацій = 100
Out[8] = {0.171 Second, Null}
```

У здобутому розв'язку зазначена кількість ітерацій і час розв'язування задачі. Після того як розв'язок знайдений, робимо відповідну інтерполяцію і будуємо графік функції розв'язку.

In[9] := int = *Interpolation*[*uappr*];

In[10] := *Plot*3*D*[int[*x*, *y*], {*y*,0,1}, {*x*,0,1},

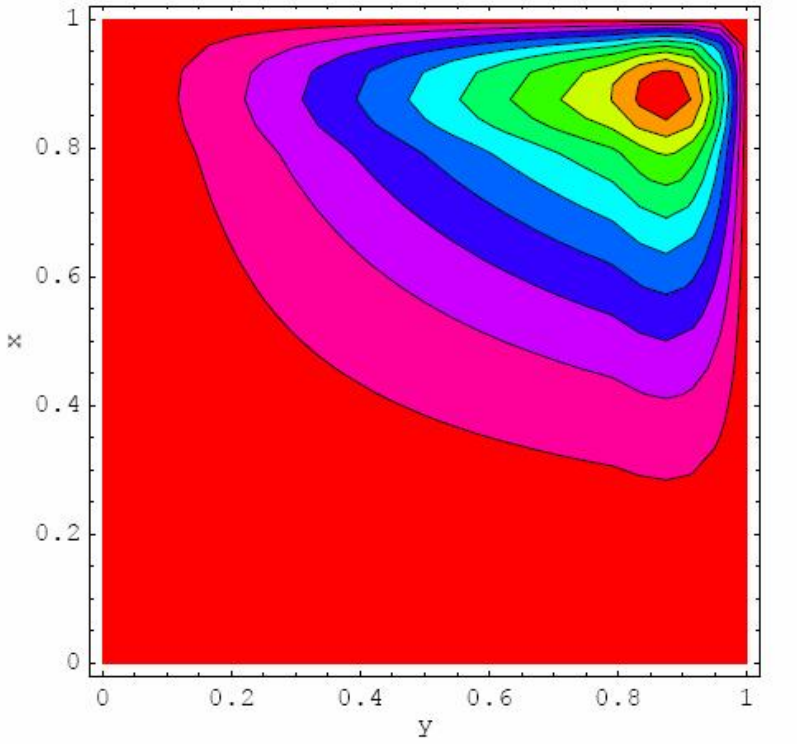
AxesLabel– **f** {"*y*","*x*","функція розв'зку"}]



Out[10] = –*SurfaceGraphics* –

In[12] := *ContourPlot*[int[*x*, *y*], {*y*,0,1}, {*x*,0,1},

*FrameLabe*1– **f** {"*y*","*x*"}, *ColorFunction* → *Hue*];



За здобутим розв'язком можна знайти і побудувати графіки функцій, які є частинними похідними розв'язку. Нижче наведений графік другої частинної похідної розв'язку по *x*.

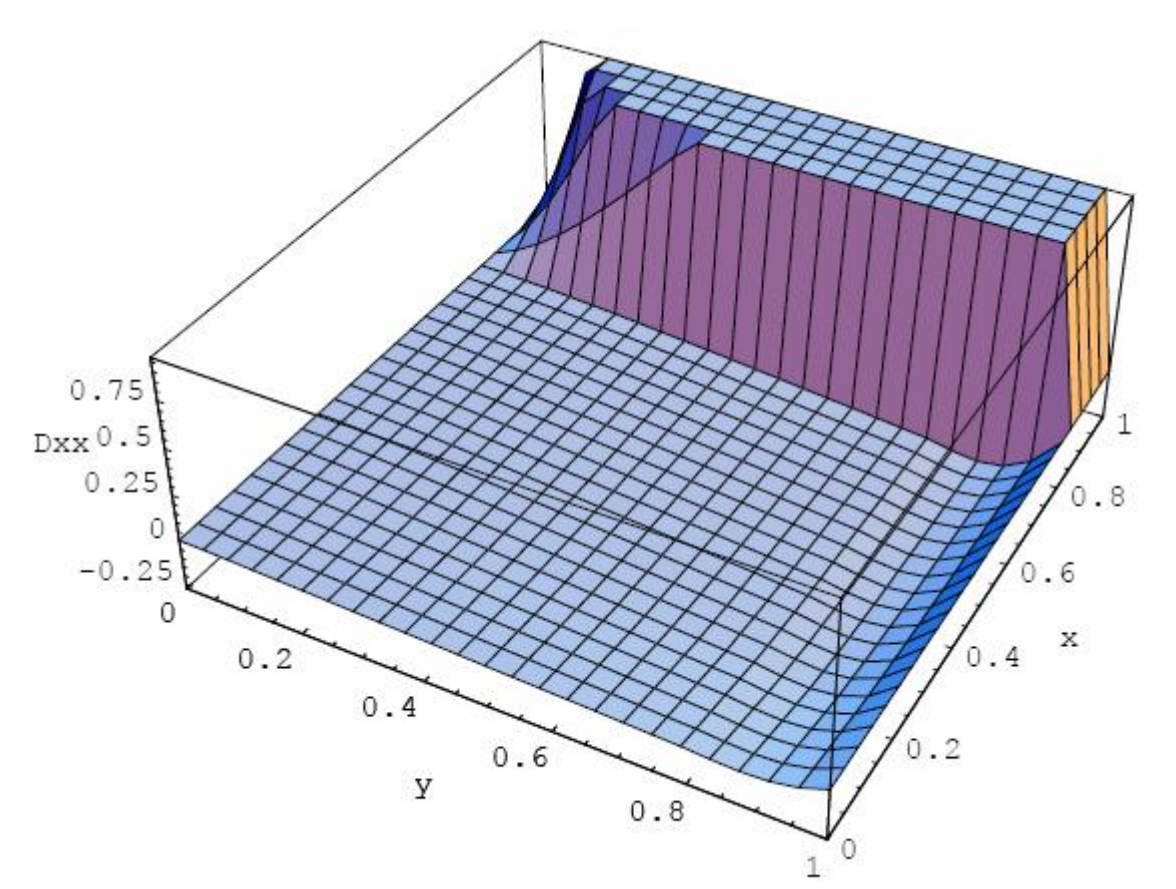
In[13] := *sigma* = *D*[int[*x*, *y*], *x*, *x*]

$$Out[13] = InterpolatingFunction\left[\begin{pmatrix} 0.1. \\ 0.1. \end{pmatrix}, <>\right][x, y]$$

Графік другої частинної похідної по координаті *x* представлений нижче.

In[14] := *Plot*3*D*[*sigma*, {*y*, 0, 1}, {*x*, 0, 1},

AxesLabel – **f** {"*y*", "*x*", "*Dxx*"}]



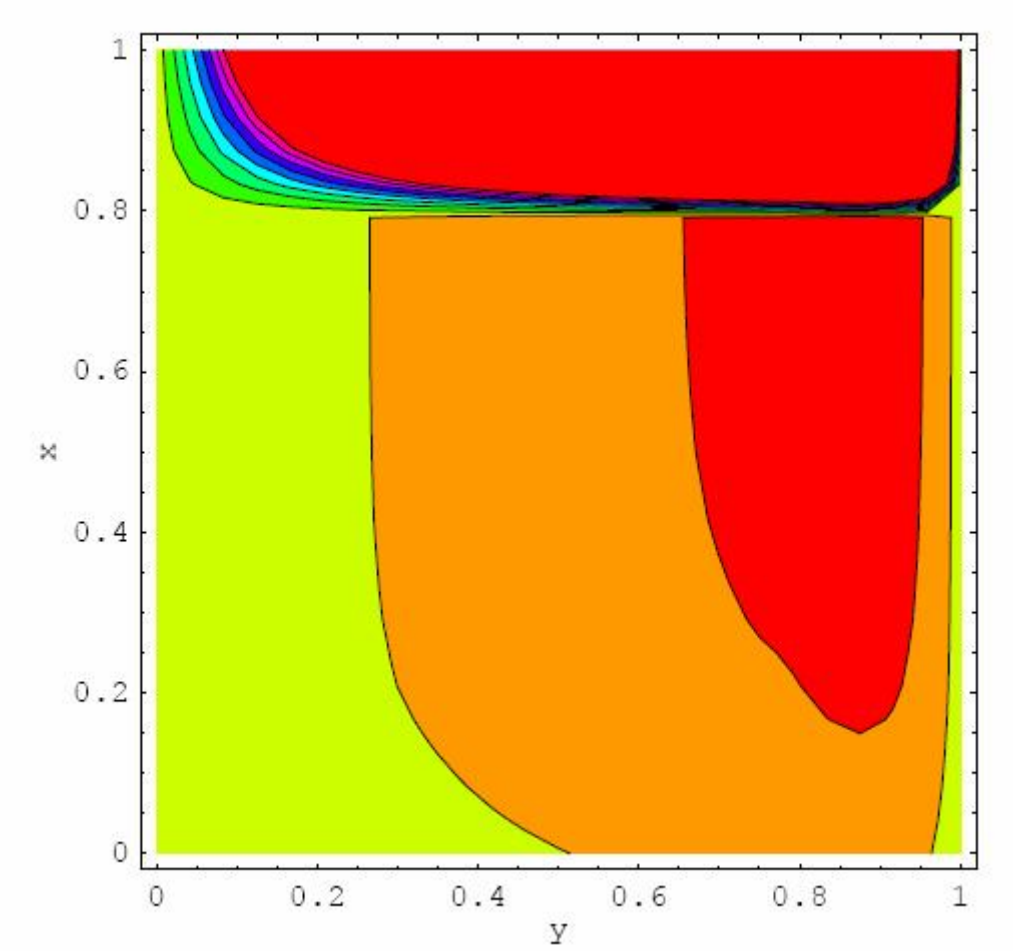
Out[14] = – *SurfaceGraphics* –

Як відомо, частинні похідні від гармонійної функції, що задовольняє рівнянню Лапласа, як у задачі (8.5), у прикладних задачах, які описуються еліптичними диференціальними рівняннями в частинних похідних, найчастіше несуть корисну інформацію про розподіл, наприклад, напруги у задачах механіки суцільних середовищ, розподілі швидкостей руху флюїду в пористих середовищах і т.ін.

Тому візуалізація таких полів має важливий практичний зміст, реалізація якого представлена нижче на контурному графіку.

In[15] := *ContourPlot*[*sigma*, {*y*, 0, 1}, {*x*, 0, 1},

*FrameLabe*1– **f** {"*y*","*x*"},*ColorFunction* → *Hue*];



Таким чином, програмний модуль, розроблений у цій темі, вирішує числову задачу про визначення гармонійної функції на площині, що задовольняє рівнянню Лапласа.

Тема 9

ЕКОНОМІКО-МАТЕМАТИЧНІ МОДЕЛІ: ФІНАНСОВА МАТЕМАТИКА. ОПТИМІЗАЦІЯ І ТЕОРІЯ ІГОР. ТЕОРІЯ МАСОВОГО ОБСЛУГОВУВАННЯ

9.1. Вступна частина

У цій темі представлені результати розв’язку задач лінійного програмування і задач теорії ігор методами комп'ютерного моделювання в програмному середовищі вищого рівня.

Підходи до розв’язання зазначених завдань базуються на загальних постановках завдань математичного моделювання, що виникають у фінансовій математиці, які нині вирішуються в загальному вигляді в сучасних комп'ютерних середовищах.

У піцдручнику представлені як загальні розв’язки завдань лінійного програмування, так і деякі конкретні завдання оптимізації, причому як конкретні завдання вирішені:

- оптимізація плану випуску;
- знаходження оптимальних стратегій завоювання ринку в умовах конкуренції для двох конкуруючих фірм.

Важливим наслідком, що випливає із загальної теорії лінійного програмування (оптимального керування) є висновок про можливості розв’язку завдань оптимізації теорії ігор методами лінійного програмування.

Суть підходу полягає в знаходженні оптимальних стратегій розвитку для конкуруючих структур на тому самому "полі гри", в іграх з нульовою сумою, тобто в грі, коли виграш однієї зі сторін автоматично приводить до збитку на відповідну суму для конкуруючої сторони.

Як відомо подібні завдання вирішуються в теорії ігор наближеними, методами причому з невизначеним результатом для однієї із сторін.

Як дано в підручнику така невизначеність може бути дозволена в рамках розв’язку завдання лінійного програмування, причому для обох конкуруючих сторін.

Задача вирішується в комп'ютерному середовищі *Mathematica* і може бути розширена на мережній технології Internet.

9.2. Загальна задача лінійного програмування

У даному розділі наведений розв'язок задачі лінійного програмування в кодах середовища *Mathematica*. Крім викладу теорії питання, наведені кілька прикладів фінансової математики і теорії ігор.

9.2.1. Матрично-векторний спосіб розв'язання загальної задачі лінійного програмування

Спочатку наведемо приклад розв'язання задачі лінійного програмування в матрично-векторному вигляді. Крім того покажемо розв'язок цієї ж задачі у вихідних змінних задачах з відповідною візуалізацією.

Сформулюємо задачу у такому вигляді:

Максимізувати величину (або мінімізувати величину) функцію цілі

$$f = \vec{C} \cdot \vec{x} \Rightarrow \min. \tag{9.1}$$

• де $\vec{C}$ -відомий вектор мети (матриця мети);

• де $\vec{x}$ -невідомий вектор розв'язку.

Зв'язки, накладені на стан, є відомими функціональними величинами, які в загальному вигляді становлять систему лінійних нерівностей.

$$\begin{aligned} &\sum_{i=1}^{N} M_{i,j} x_i \geq b_j \\ &x_j \geq 0, \\ &(j = 1,2,...K). \end{aligned} \tag{9.2}$$

Матричний вигляд системи нерівностей (9.2) такий:

$$\hat{M} . \vec{x} \geq \vec{b}. \tag{9.3}$$

де в (9.3) введені позначення

$$\begin{aligned} \hat{M} &= \begin{pmatrix} M_{11} M_{12} ... M_{1n} \\ M_{21} M_{22} ... M_{2n} \\ ... \quad ... \quad ... \quad ... \\ M_{k1} M_{k2} ... M_{kn} \end{pmatrix}; \\ \vec{x} &= \begin{pmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{pmatrix}; \\ \vec{b} &= \begin{pmatrix} b_1 \\ b_2 \\ ... \\ b_n \end{pmatrix}; \end{aligned} \tag{9.4}$$

Загальна математична постановка задачі лінійного програмування в матричному вигляді записується в такий спосіб:

$$\begin{aligned} &f = \vec{C} \cdot \vec{x} \Rightarrow \min; \\ &\hat{M} \cdot \vec{x} \geq \vec{b}; \\ &x_j \geq 0. \end{aligned} \qquad (9.5)$$

9.2.2. Приклад: Система із трьома зв'язками

Як приклад розглянемо задачу знаходження мінімуму величини

$$f = \vec{C} \cdot \vec{x} \to \min \qquad (9.6)$$

де в (9.6) введені позначення:

$\vec{C} = \{3, \quad 1\}$ - вектор відомих величин;

$\vec{x} = \{x, \quad y\}$ - вектор невідомих.

Явний вид функції мети має вигляд:

$$(x + y) \Rightarrow \min \qquad (9.7)$$

Зв'язки, накладені на систему, мають вигляд

$$\begin{aligned} &-2x - y \geq -2; \\ &x - y \geq \frac{1}{2}; \\ &-x - 2y \geq -2; \\ &x \geq 0; y \geq 0. \end{aligned} \qquad (9.8)$$

■ Коди розв'язку в середовищі *Mathematica*

Задамо вихідні дані для векторів і матриць:

$In[1] := c = \{-1, \quad -1\};$

$M = \{\{-2, \quad -1\}, \{1, \quad -1\}, \{-1, \quad -2\}\};$

$\vec{b} = \{-2, -\frac{1}{2}, -2\};$

$In[4] := M$ // *MatrixForm*

$Out[4]$// *MatrixForm* =

$$\begin{pmatrix} -2 & -1 \\ 1 & -1 \\ -1 & -2 \end{pmatrix}$$

Невідомий вектор оптимального розв'язку знаходимо у вигляді:

$In[5] := \vec{x} = \{x, y\} = Linear\,\mathrm{Pr}\,ogram\,\min\, g[c, M, \vec{b}]$

$Out[5] = \left\{\frac{2}{3}, \frac{2}{3}\right\}$

Оптимальна величина функції мети в точці оптимального розв’язку запишеться у вигляді

$In[6] := prize = -c.\vec{x}$

$Out[6] = \frac{4}{3}$

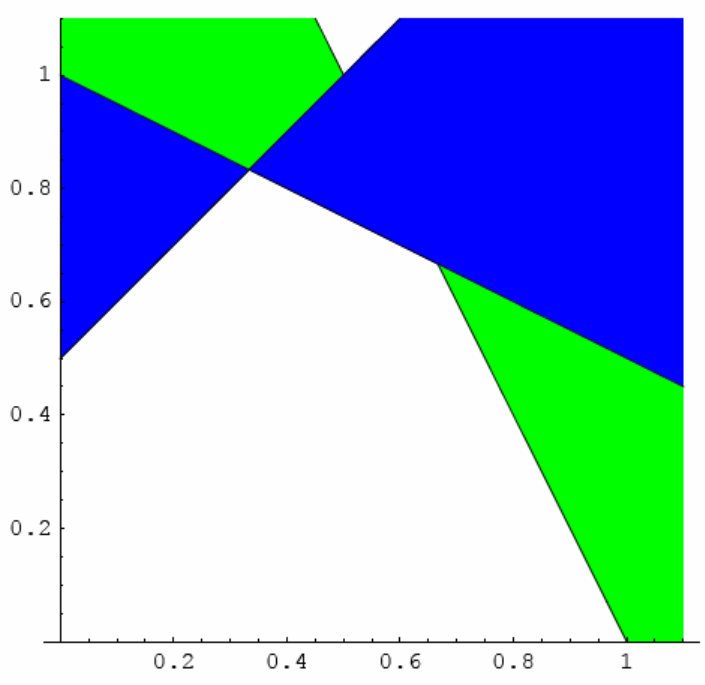
Візуалізація розв’язку і зв'язків для функції двох змінних може бути виконана на площині *(x,y):*

$In[7] := Quit[];$

$In[1] :=<< Graphics'FilledPlot'$

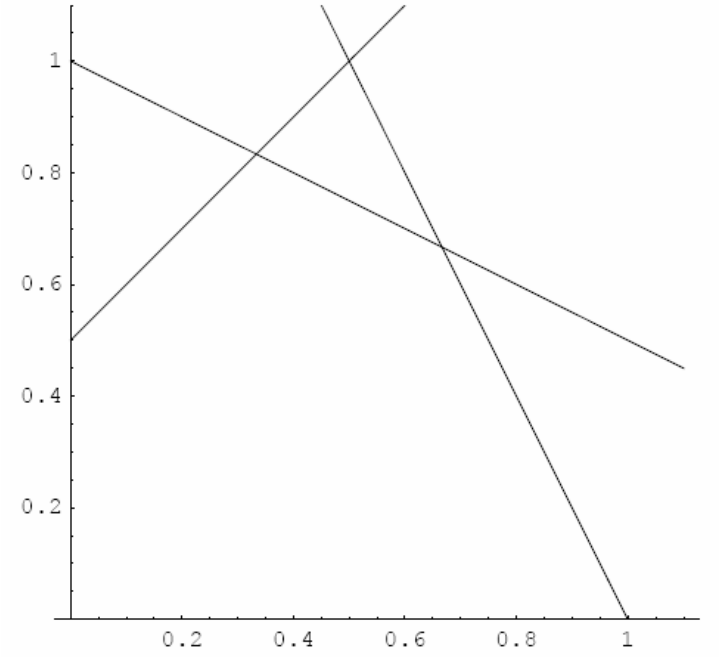
$In[2] := con1 = FilledPlot[\{2-2x, x+1/2, 1-x/2\},$

$\{x,0,1.1\}, AspectRatio \to Automatic, PlotRange \to \{0,1.1\}];$



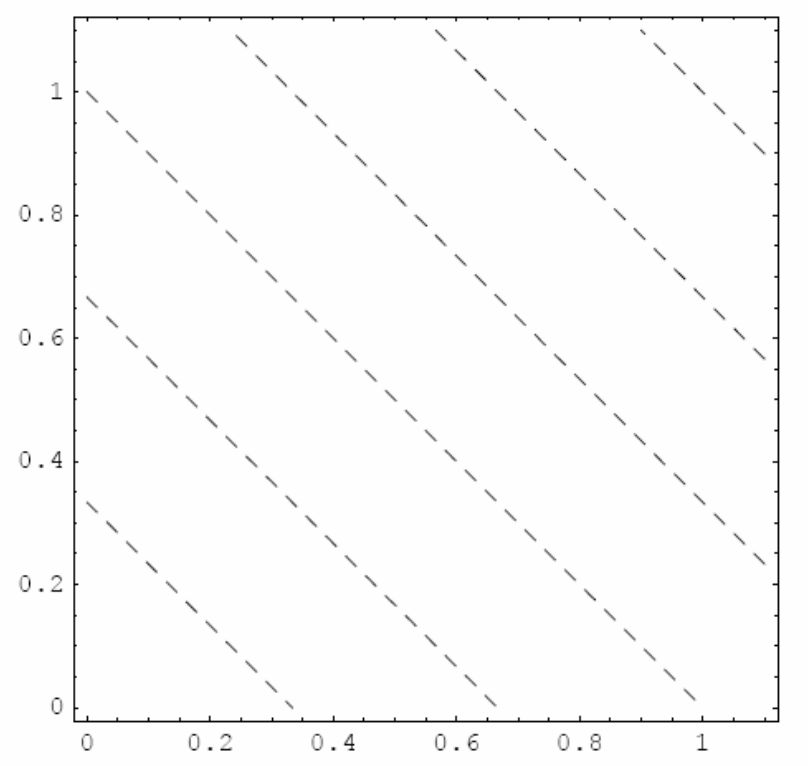
Графіки зв'язків представлені нижче.

$In[3] := con2 = Plot[\{2-2x, x+1/2, 1-x/2\}, \{x,0,1.1\}, AspectRatio \to Automatic, PlotRange \to \{0,1.1\}];$



Далі зображуємо графік функції мети пунктирними лініями на площині.

In[4] := *obj* = *ContourPlot*[*x* + *y*, {*x*,0,1.1}, {*y*,0,1.1}, *Contours* → *Range*[1/3,2,1/3],
ContourShading → *False*, *ContourStyle* → {*Dashing*[{0.02}]}];



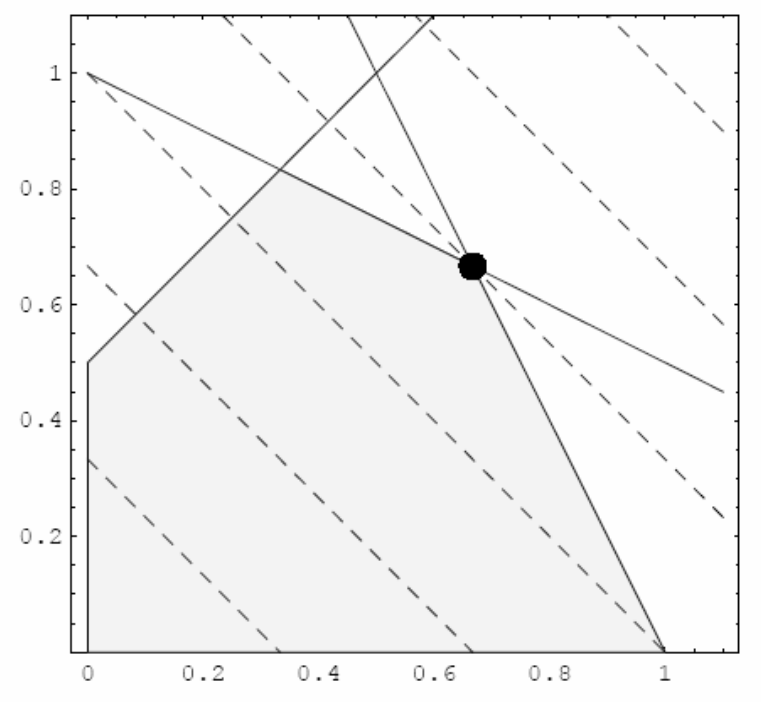
In[8] := *point s* = {{0,0}, {1,0}, {2/3,2/3}, {1/3,5/6}, {0,1/2}, {0,0}};

In[9] := *feasible* = *Graphics*[{*GrayLevel*[.95], *Polygon*[*point s*],

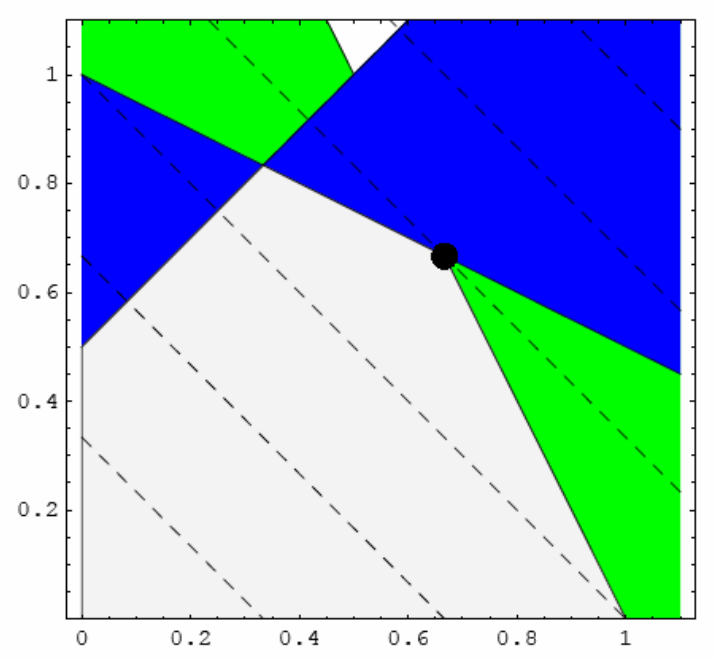
GrayLevel[0], *Line*[*po* int *s*], *Po* int *Size*[.04], *Po* int[{2 / 3,2 / 3}]}];

Нарешті оптимальний розв'язок представлений у вигляді точки, що є вершиною багатокутника відтинаючими прямими зв'язками, які є для функції мети. Очевидно, що лінії, які є зв'язками, і функція мети перетинаються в одній і тій же точці, що є оптимальним розв'язком.

In[10] := *Show*[*con*2, *feasible*, *obj*, *Frame* → *True*];



In[11] := *Show*[*con*1, *feasible*, *obj*, *Frame* → *True*];



9.2.3. Оптимізація плану виробництва

9.2.3.1. Постановка задачі в термінах економічної теорії. Вихідні дані для розв’язку задачі про оптимізацію виробництва.

Умова задачі взята з навчального посібника [3].

Виробництво: Завод випускає дві моделі автомобілів: "1-1" і "2-2".

Робоча сила зайнята на виробництві:

800-кваліфікованих робітників;

1000-некваліфікованих робітників.

*Оплата праці***:** завод оплачує всім робітникам 40 годин на тиждень.

Витрати праці на виробництво одного авто:

~ модель "1-1" потребує 50 годин кваліфікованої праці і 30 годин некваліфікованої праці;

~ модель "2-2" потребує 20 годин кваліфікованої праці і 40 годин некваліфікованої праці.

Фінансові витрати на виробництво одного авто:

■ модель "1-1" потребує на своє виробництво 1500 у.о.;

■ модель "2-2" потребує на своє виробництво 500 у.о;

Сумарні фінансові запаси (кредит банку) на заводі становлять у тиждень: 900 000 у.о.

Збут:

■ робота 5 днів у тиждень;

■ вивіз не більше 210 машин у день.

Прибуток:

■ від реалізації "1-1" завод одержує 1000 у.о.

■ від реалізації "2-2" завод одержує 500 у.о.

Задача полягає в тому, щоб:

■ знайти оптимальні обсяги випуску продукції;

■ знайти оптимальні обсяги випуску автомобілів моделі " 1-1";

■ знайти оптимальні обсяги випуску автомобілів моделі "2-2";

■ знайти оптимальні витрати на випуск цієї продукції, які б не перевищували відомі фінансові "запаси";

■ знайти оптимальні шляхи підвищення ефективності випуску продукції.

9.2.3.2. Формулювання функції мети і складання зв'язків у системі. Як функція мети вибираємо максимум прибутку заводу від випуску продукції, тобто від реалізації авто. Очевидно, що прибуток дорівнює

$$f = (1000x_1 + 500x_2) \Rightarrow Max \qquad (9.9)$$

Природними обмеженнями при виробництві автомобілів вважаються:

- сумарні витрати на виробництво випуску авто не можуть перевищувати фінансові ресурси заводу;

$$500x_1 + 1500x_2 \le 900000; \qquad (9.10)$$

- кількість годин на виробництво випуску авто, які витрачають некваліфіковані робітники, при виробництві обох марок авто становить 30 годин на тиждень, а обмеженнями є сумарна кількість годин затрачуваних всіма некваліфікованими робітниками (1000 чол.) в 40-годинний тиждень, тобто 1000*40;

$$30x_1 + 40x_2 \le 100040 = 40000; \qquad (9.11)$$

- витрати на кваліфікованих робітників при виробництві обох марок авто становлять:

$$50x_1 + 20x_2 \le 80040 = 32000; \qquad (9.12)$$

- кількість автомобілів, які вивозять із заводу не може бути більша, чим 5*210 авто:

$$x_1 + x_2 \le 5 \times 210 = 1050. \qquad (9.13)$$

Таким чином математичне формулювання задачі набуває вигляд:

• знайти оптимальний випуск авто двох марок (марки "1-1" і "2-2") при природних обмеженнях у системі:

$$\begin{aligned} &f = (1000x_1 + 500x_2) \Rightarrow Max; \\ &\text{при обмеженнях} \\ &1500x_1 + 500x_2 \le 900000; \\ &30x_1 + 40x_2 \le 100040 = 40000; \\ &50x_1 + 20x_2 \le 80040 = 32000; \\ &x_1 + x_2 \le 5210 = 1050. \end{aligned} \qquad (9.14)$$

Нижче представлені коди середовища *Mathematica*, які реалізують задачу (9.14).

9.2.3.3. Коди розв'язків у середовищі *Mathematica*: Оптимізація плану випуску. Розв'язання задачі в матрично-векторному вигляді.

Для розв'язання задачі використовується оператор:

$$Linear\,\mathrm{Pr}\,ogram\,\min\, g[\vec{c}, \hat{M}, \vec{b}]$$

Задамо вихідні дані для векторів і матриць:

$In[15] := Quit[];$

$In[1] := \vec{c} = -\{1000,500\};$

$M = -\{\{1500,500\},\{30,40\},\{50,20\},\{1,1\}\};$

$\vec{b} = -\{900000,40000,32000,5210\};$

$In[4] := M \,//\, MatrixForm$

$Out[4] // MatrixForm =$

$$\begin{pmatrix} -1500 & -500 \\ -30 & -40 \\ -50 & -20 \\ -1 & -1 \end{pmatrix}$$

Невідомий вектор $\vec{x}$ оптимального випуску автомобілів обох марок знаходимо у вигляді:

$In[5] := \vec{x} = \{x_1, x_2\} = Linear\,\mathrm{Pr}\,ogram\,\min\, g[\vec{c}, M, \vec{b}]$

$Out[5] = \{\frac{1100}{3}, \frac{2050}{3}\}$

Оскільки автомобілі можуть випускатись тільки у зібраному вигляді, а їхня кількість визначатись тільки цілими, то

$In[6] := \vec{x} = \{x_1, x_2\} = Linear\,\mathrm{Pr}\,ogram\,\min\, g[\vec{c}, M, \vec{b}] // N // Round$

$Out[6] = \{367,683\}.$

Визначаємо оптимальний план випуску авто марки "1-1":

$In[7] := x_1$

$Out[7] = 367.$

Сумарний оптимальний план випуску авто обох марок:

$In[8] := \sum_{j=1}^{2} x_j$

$Out[8] = 1050.$

Далі визначаємо оптимальні витрати виробництва. Обчисливши функцію мети, як оптимальні витрати на випуск 356 авто "1-1" і 733 авто "2-2" запишемо у вигляді

$In[9] := \exp enseOutPut = -\vec{c}.\vec{x}$

$Out[9] = 708500.$

Залишок від одного робочого тижня запишемо у вигляді

$In[10] := debit = 900000 + \vec{c}.\vec{x}$

$Out[10] = 191500.$

Розв’язок задачі у вихідних змінних *x1 і x2.*

Зручність розв’язання вихідної задачі в змінних і полягає в тому, що використовуючи стандартну процедуру:

$$\begin{matrix}\hat{M}.\vec{x} \le \vec{b};\\ \{x_j\} \ge 0\end{matrix} \qquad (9.15)$$

використовується оператор вигляду:

$Maximize[Object, Constrains, Variables]$,

у якому функціональні змінні можна записати в їхньому оригінальному вигляді, тобто у вигляді співвідношень (9.5).

$Clear[x]$

Далі введемо вектор вихідних змінних $\vec{x} = \{x_1, x_2\}$

$In[1] := Quit[];$

$In[1] := \vec{x} = \{x_1, x_2\};$

потім вводимо відомі вектори мети $\vec{c}$ і $\vec{b}$ і відому матрицю $\hat{M}$:

$In[2] := \vec{c} = \{1000,500\};$

$\hat{M} = \{\{1500,500\},\{30,40\},\{50,20\},\{1,1\}\};$

$\vec{b} = \{900000,40000,32000,5210\}.$

Відповідний розв’язок, що дає як значення *Max функції мети*, так і кількість випуску авто, які з погляду розрахунку є "координатами точки", у якій цей максимум досягається, представлено нижче.

Важливо відзначити, що зв'язки в системі записані в загальному, тобто матрично-векторному вигляді, що є безсумнівною зручністю при формулюванні відповідної задачі.

$In[5] := planExpense = Maximize[\vec{c}.\vec{x}, \{Thread[\hat{M}.\vec{x} \le \vec{b}]\} // Flatten, \vec{x}]$

$Out[5] = \{\frac{2125000}{3}, \{x_1 \to \frac{1100}{3}, x_2 \to \frac{2050}{3}\}\}.$

Звичайно, отримаємо те ж саме розв’язання, якщо є обмеження в системі, тобто нерівності математичної задачі.

Відповідний розв’язок наведений нижче:

$In[7] := planExpenseHand = Maximize[1000x_1 + 500x_2,$

$\{1500x_1 + 500x_2 \le 900000, 30x_1 + 40x_2 \le 40000, 50x_1 + 20x_2 \le 32000, x_1 + x_2 \le 1050\}, \vec{x}]$

$Out[7] = \{\frac{2125000}{3}, \{x_1 \to \frac{1100}{3}, x_2 \to \frac{2050}{3}\}\}$

З другого боку, функція мети може мати вигляд $(f = S - \vec{C}.\vec{x} = Min)$, інакше кажучи, ми хочемо мінімізувати:

$$In[8] := planExpense = Minimize\left[(S = 900000) - \vec{c}.\vec{x}, \{Thread\left[\hat{M}.\vec{x} \le \vec{b}\right]\} // Flatten, \vec{x}\right]$$

$$Out[8] = \{\frac{575000}{3}, \{x_1 \to \frac{1100}{3}, x_2 \to \frac{2050}{3}\}\}$$

Далі знайдемо витрати виробництва і залишок від виділених засобів

$In[9] := planExpense[[1]] // N$

$Out[9] = 191667.$

$In[9] := 900000 - planExpense[[1]] // N$

$Out[9] = 708333.$

і оптимальний план випуску продукції.

$In[10] := \{x_1, x_2\} /. planExpense[[2]] // Round$

$Out[10] = \{367,683\}$

Візуалізація розв'язків: Економічні діаграми

Далі приведемо відповідні діаграми, які ілюструють отримані розв'язки.

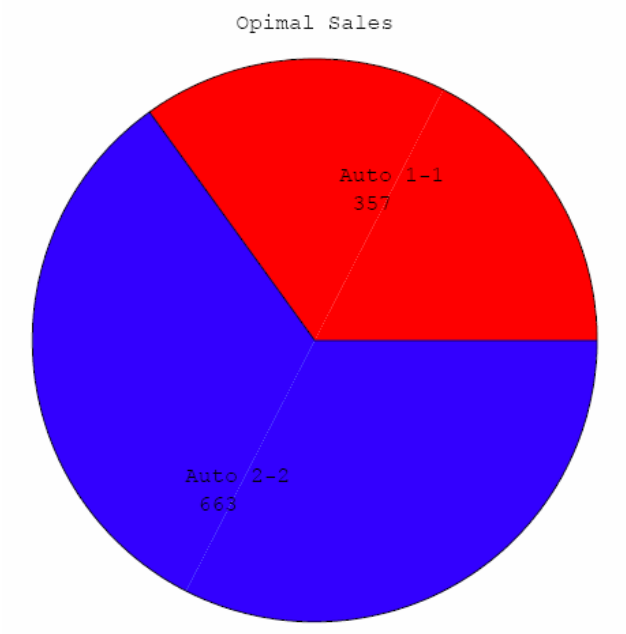
$In[11] :=<< Graphics'Graphics'$

Спочатку приведемо кругову діаграму оптимального випуску авто першої і другої марок,

$In[12] := plan1 = PieChart[\{x_1, x_2\} /. planExpense[[2]] // N // Round,$

$PieLabels \to \{"Auto1 - 1 \backslash n357", "Auto2 - 2 \backslash n663"\},$

$PlotLabel \to "OpimalSales"];$

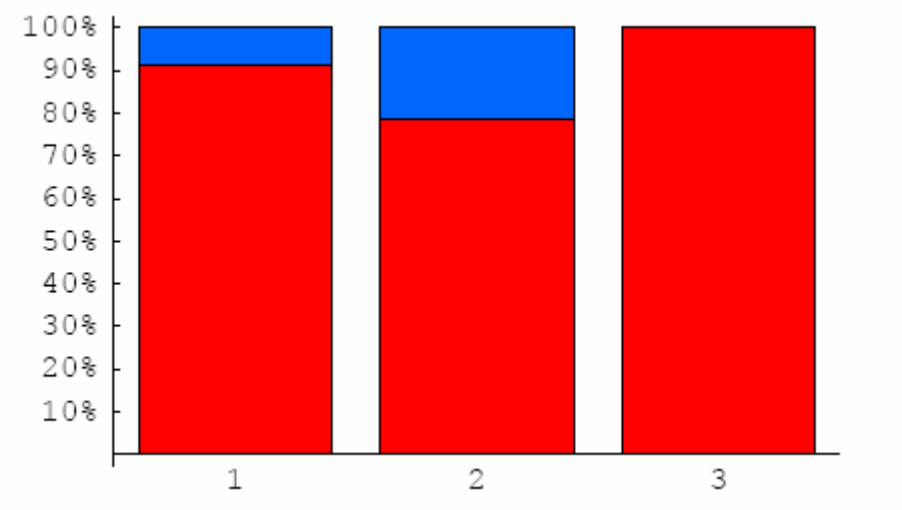


яка наочно показує величину секторів, які займають зазначені марки авто в загальному випуску автовиробника.

Інший вигляд діаграм, а саме стовбцеві діаграми, показує процентний розподіл оптимального випуску стосовно загального (стовбець № 3 на діаграмі) випуску, зроблених авто.

```
In[15]:= pl2 = PercentileBarChart[{x1 + x2, x1, x2} /. planExpense[[2]] // N // Round,
            {100,100}, PlotLabel– f"Оптимальний план \ n    випуску"];
```

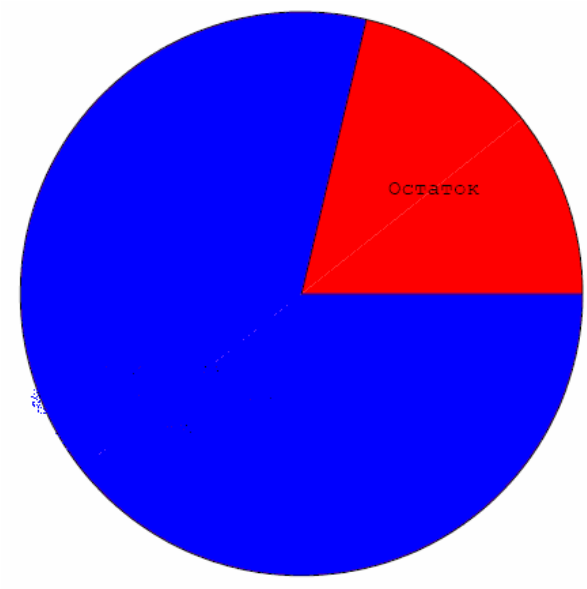
Оптимальний план випуску



Далі приведемо діаграму оптимальних щомісячних витрат виробника.

```
In[20]:= exp1 = PieChart[{planExpense[[1]] // N, 900000 – planExpense[[1]] // N},
            PieLabels– f {"Залишок","Витрати на \ n оптимальний план \ n  випуску"},
            PlotLabel - f"Діаграма розподілу витрат"];
```

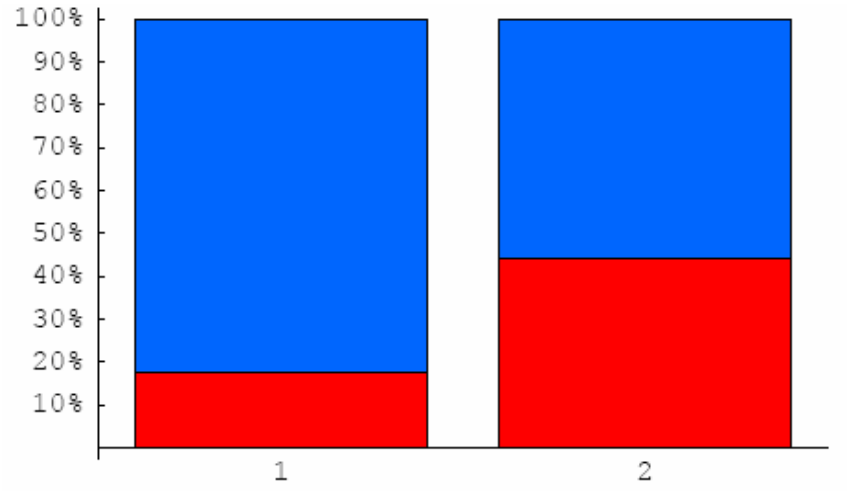
Діаграма розподілу витрат



Інший вигляд діаграми оптимальних витрат виробника і щомісячний залишок банківського кредиту представлений нижче.

```
In[22]:= exp2 = PercentileBarChart[{planExpense[[1]]// N, 900000 - planExpense[[1]]// N},
          {900000,900000}, PlotLabel →"Діаграма розподілу  витрат"];
```

Діаграма розподілу витрат



Розподіл даних стосовно величини банківського кредиту, який становить приблизно 90000 у.о.

Зазначимо, оскільки авто може збуватись із заводу тільки "цілим" об'єктом, то відповідне розв’язання наведене в цілих числах.

9.2.3.4. Висновки по оптимізації плану виробництва. *Відповідь на питання про оптимальний план виробництва авто:*

Оптимальний план випуску:

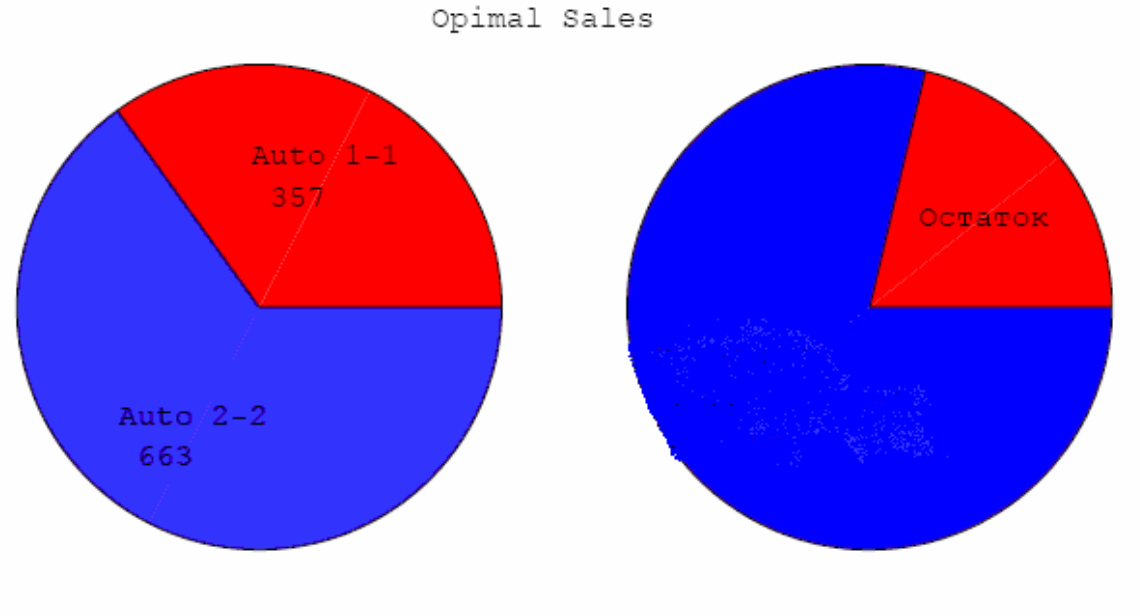
- модель "1-1" - - 367 штук;
- модель "2-2" - - 683 штуки;

Оптимальні витрати: 708333.3 у.о.

Остача оптимального плану: 191667. у.о.

Сполучені діаграми оптимального плану виробництва і оптимальних витрат реалізуючий оптимальний план виробництва.

```
In[23]:= Show[GraphicsArray[{plan1, exp1}]];
```

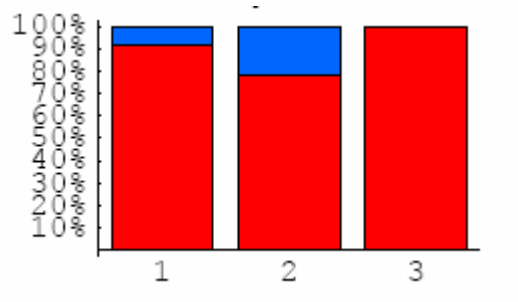


Діаграма розподілу витрат

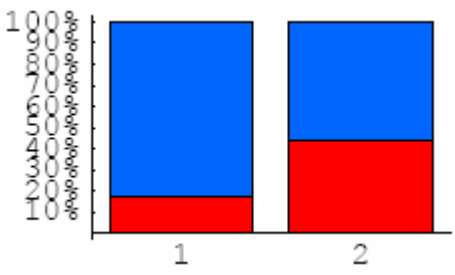
Той же самий результат, але представлений у вигляді діаграм зі стовпців.

In[24] := *Show*[*GraphicsArray*[{pl2, exp2}]];

Оптимальний план випуску



Діаграма розподілу витрат



9.2.4. Постановка і розв'язок двоїстої задачі оптимізації виробництва

9.2.4.1. Розв'язок двоїстої задачі. Задача розв'язується у вихідних змінних y_1, y_2, y_3, y_4, економічний зміст яких наведений далі.

Приведемо розв'язок двоїстої задачі у вихідних змінних. Відповідний оператор, що дозволяє розв'язати двоїсту задачу, має вигляд

ConstrainedMin[*Object*, *Constrains*, *Variables*].

Задаємо вектор у вигляді:

In[25] := *Quit*[];

In[2] :=<< *Graphics*'*Graphics*'

In[1] := $\vec{x} = \{y_1, y_2, y_3, y_4\}$;

де y_1-передбачувані витрати виробництва в тиждень;

y_2-кількість некваліфікованої робочої сили;

y_3-кількість кваліфікованої робочої сили;

y_4-кількість машин, що відправляють;

і матриці $\vec{c}, \hat{M}, \vec{b}$ таким чином

$$In[3] := \vec{c} = \{900000,40000,32000,1050\};$$

$$\hat{M} = \{\{1500,30,50,1\},\{500,40,20,1\}\};$$

$$\vec{b} = \{1000,500\}.$$

Далі розв’язуємо двоїсту задачу у векторно-матричному виді

$$In[6] := planExpenseDual = ConstrainedMin\left[\vec{c}.\vec{x}, \left\{Thread\left[\hat{M}.\vec{x} \geq \vec{b}\right]\right\} // Flatten, \vec{x}\right]$$

$$Out[6] = \left\{\frac{2125000}{3}, \left\{y_1 \to 0, y_2 \to 0, y_3 \to \frac{50}{3}, y_4 \to \frac{500}{3}\right\}\right\}.$$

Як видно з комп'ютерного розв’язку, який тут сформульований у матричному вигляді, комп'ютерний код і вид формулювання задачі практично не відрізняється від "книжкового" формулювання задачі лінійного програмування [3].

Цей вид формулювання і розв’язання задачі оптимізації є явною перевагою, тому що не потребує значних зусиль щодо конвертації вихідних даних у машинному розв’язку.

Нарешті приведемо розв’язок двоїстої задачі в природному вигляді. Відповідний комп'ютерний код наведений нижче.

$$In[7] := planExpenseHandDual = ConstrainedMin[900000y_1 + 40000y_2 + 32000y_3 + 1050y_4,$$

$$\{1500y_1 + 30y_2 + 50y_3 + y_4 \geq 1000, 500y_1 + 40y_2 + 20y_3 + y_4 \geq 500\}, \vec{x}]$$

$$Out[7] = \{\frac{2125000}{3}, \{y_1 \to 0, y_2 \to 0, y_3 \to \frac{50}{3}, y_4 \to \frac{500}{3}\}\}.$$

Окремі результату вищенаведеного розв’язку подані нижче без коментарів.

$$In[8] := planExpenseDual[[1]] // N$$

$$Out[8] = 708333.$$

$$In[9] := \{y_3, y_4\} /. planExpenseDual[[2]] // N$$

$$Out[9] = \{16.6667,166.667\}$$

$$In[10] := y_4 /. planExpenseDual[[2]] // N // Round$$

$$Out[10] = 167$$

$$In[11] := y_3 /. planExpenseDual[[2]] // N$$

$$Out[11] = 16.6667.$$

Відповідь на розв'язок двоїстої задачі оптимізації:

Оптимальний план випуску

- модель "1-1" - 367 штук;
- модель "2-2" - 683 штуки.

Оптимальні витрати: 708333.3 у.о.

Залишок засобів внаслідок оптимального плану: 191667. у.о.

Аналіз: Для збільшення випуску необхідно збільшити кількість кваліфікованої робочої сили на 17 чоловік і кількість авто, які вивозяться із заводу, - на 167 штук.

9.3. Оптимальна стратегія розвитку в умовах конкуренції

9.3.1. Розв'язок задачі в матрично-векторному вигляді

Дві конкуруючі фірми (галузі), які надалі будемо називати *"гравцями"*, можуть здійснювати капіталовкладення (інвестиції) в чотири об'єкти, що перебувають у полі конкурентного середовища і які підлягають інвестуванню. Стратегія галузей така:

i-та стратегія галузі полягає у фінансуванні *i-го* об'єкта інвестування.

Припустимо, що зважаючи на місцеві умови і особливості інвестицій, прибуток першої галузі відповідно до *i-ї* стратегії на *j-му* об'єкті виражається наступною матрицею

$$M = \begin{pmatrix} 0 & 1 & -1 & 2 \\ -1 & 0 & 3 & 2 \\ 0 & 1 & 2 & -1 \\ 2 & 0 & 0 & 0 \end{pmatrix}; \quad (\$) \qquad (9.16)$$

У цій ситуації величина прибутку першої галузі вважається такою ж як величина збитку для другої галузі. Тому представлена гра може розглядатися як матрична гра для конкуруючих гравців з нульовою сумою.

Для розв'язку ігрової задачі додамо до кожного елемента матриці *M + 2:*

$In[4] := Quit[];$

$$In[1] := \hat{M} = \begin{pmatrix} 0 & 1 & -1 & 2 \\ -1 & -1 & 3 & 2 \\ 0 & 1 & 2 & -1 \\ 2 & 0 & -1 & 0 \end{pmatrix};$$

$In[2] := \hat{M} + 2$

$$Out[2] = \begin{pmatrix} 2 & 3 & 1 & 4 \\ 1 & 1 & 5 & 4 \\ 2 & 3 & 4 & 1 \\ 4 & 2 & 1 & 2 \end{pmatrix}$$

$In[3] := Det[\hat{M}]$

$Out[3] = 33$

Тоді задача набуває вигляду:

мінімізувати функцію мети, тобто величину ціни гри (ціна гри - величина обернена до суми інвестицій в об'єкти), при якій прибуток першого гравця максимальний

$$\sum_{i=1}^{4} p_i \Rightarrow \min \tag{9.17}$$

при наступних зв'язках

$$\begin{pmatrix} 2 & 3 & 1 & 4 \\ 1 & 2 & 5 & 4 \\ 2 & 3 & 4 & 1 \\ 4 & 2 & 2 & 2 \end{pmatrix}^T \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}; \tag{9.18}$$

$p_i \geq 0.$

де вектор $\vec{p} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix}$ - становить невідомий вектор стратегії.

Знайдемо оптимальні стратегії інвестицій для першого гравця. Коди в середовищі *Mathematica* мають вигляд:

$In[14] := Quit[];$

$In[4] := \vec{c} = \{1,1,1,1\};$

$\hat{M} = \hat{M} + 2;$

$\vec{b} = \{1,1,1,1\};$

Розв’язок задачі виконаємо оператором *LinearProgramming[...]*

$In[7] := \vec{x} = \{p_1, p_2, p_3, p_4\} = Linear\ \mathrm{Pr}ogram\ \min\ g[\vec{c}, Transpose[\hat{M}], \vec{b}]$

$Out[7] = \{\frac{19}{177}, \frac{3}{59}, \frac{23}{177}, \frac{7}{59}\}$.

Наприклад, величина являє собою числове значення ціни вкладень у перший об'єкт при оптимальній стратегії гри першим гравцем.

$In[8] := p_1$

$Out[8] = \frac{19}{177}$

Оптимальна ціна всієї гри дорівнює:

$In[9] := prise = \left(\sum_{i=1}^{4} p_i\right)$.

$Out[9] = \frac{24}{59}$

Обсяг інвестицій дорівнює:

$In[10] := v = \left(\sum_{i=1}^{4} p_i\right)^{-1}$.

$Out[10] = \frac{59}{24}$

Оскільки задача формулюється як ігрова, то змішані стратегії першого гравця мають вигляд:

$In[11] := strategy = v\vec{x}$

$Out[11] = \{\frac{19}{72}, \frac{1}{8}, \frac{23}{72}, \frac{7}{24}\}$.

Повертаючись до початкової умови *(-2)*, загальна ціна початкової гри дорівнює:

$In[12] := V = v - 2$

$Out[12] = \frac{11}{24}$.

Отримане значення інвестицій в чотири об'єкти є оптимальним для першого гравця. Нарешті, знаходимо оптимальні стратегії гри першого "гравця":

$In[13] := \vec{X} = \{X_1, X_2, X_3, X_4\} = v\vec{x}$

$Out[13] = \{\frac{19}{72}, \frac{1}{8}, \frac{23}{72}, \frac{7}{24}\}$.

$In[14] := \sum_{i=1}^{4} X_i$

$Out[14] = 1$

9.3.2. Візуалізація стратегій першого гравця

Візуалізація отриманих стратегій може бути показана у вигляді відповідних діаграм, як це прийнято в економічних задачах.

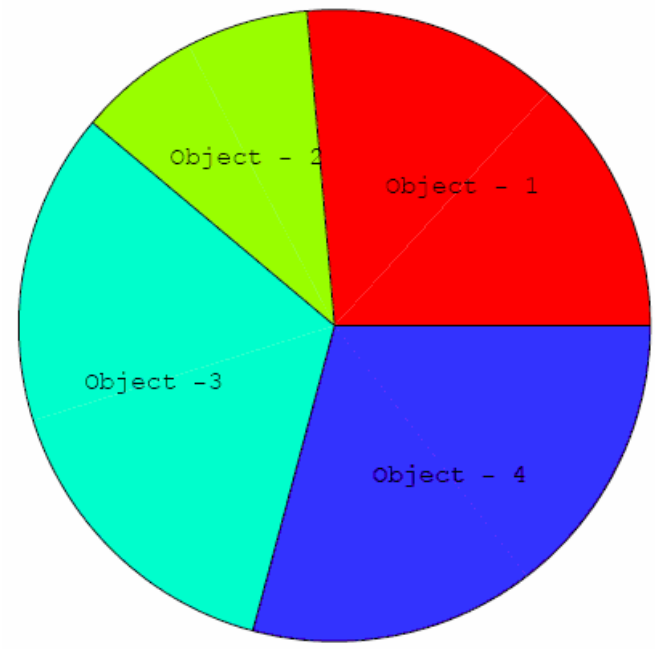
Із цією метою спочатку завантажуємо відповідний пакет *Mathematica*.

```
In[15]:=<< Graphics'Graphics'
```

Спочатку представимо розв'язок у вигляді сегментів ділення загальних інвестицій по об'єктах 1-4.

```
In[16]:= diagr1 = PieChart[X→,
          PieLabels– f {"Object – 1","Object – 2","Object – 3","Object – 4"},
          PlotLabel– f"Оптимальні сратегії завоювання \ n  ринку першим гравцем"
          ];
```

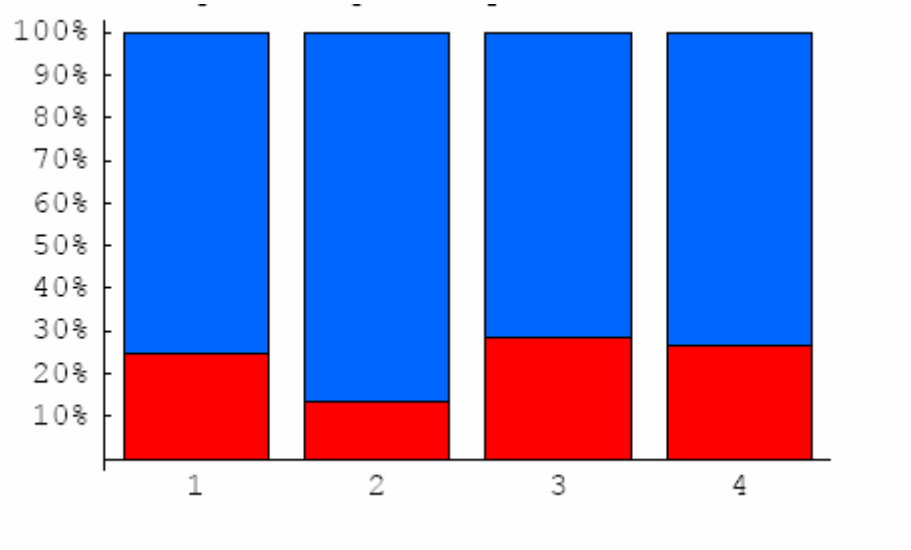
Оптимальні стратегії завоювання ринку першим гравцем



Далі інвестиції представимо у вигляді наповнень відповідних "стовпців"

```
In[17]:= col1 = PercentileBarChart[X→,
          {0.8,0.8,0.8,0.8},
          PlotLabel– f"Оптимальні стратегії завоювання \ n  ринку першим гравцем"];
```

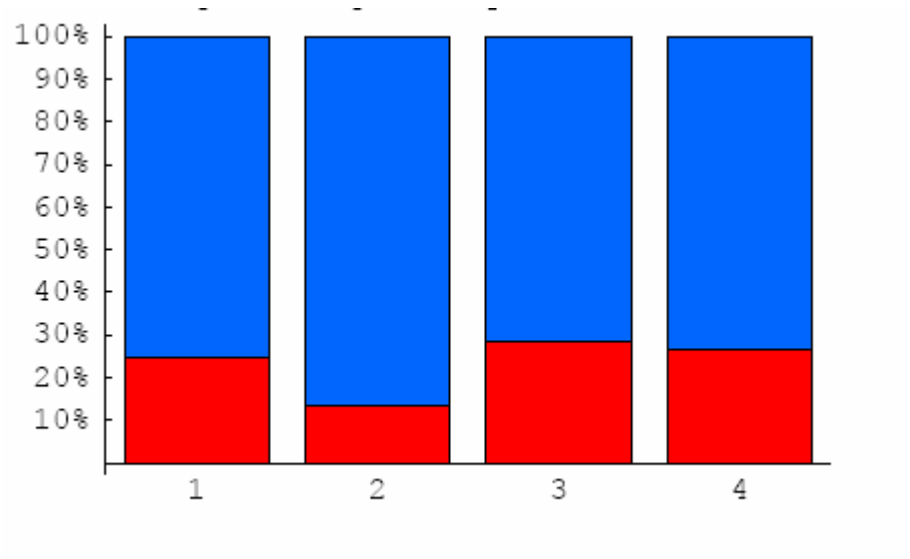
Оптимальні стратегії завоювання ринку першим гравцем



При інфляції в 25% відповідні вкладення змінюються згідно з діаграмою представленою нижче:

```
In[18]:= col2 = PercentileBarChart[X→,
            {0.75,0.75,0.75,0.75},
            PlotLabel– f"Оптимальні стратегії завоювання \ n   ринку першим гравцем"];
```

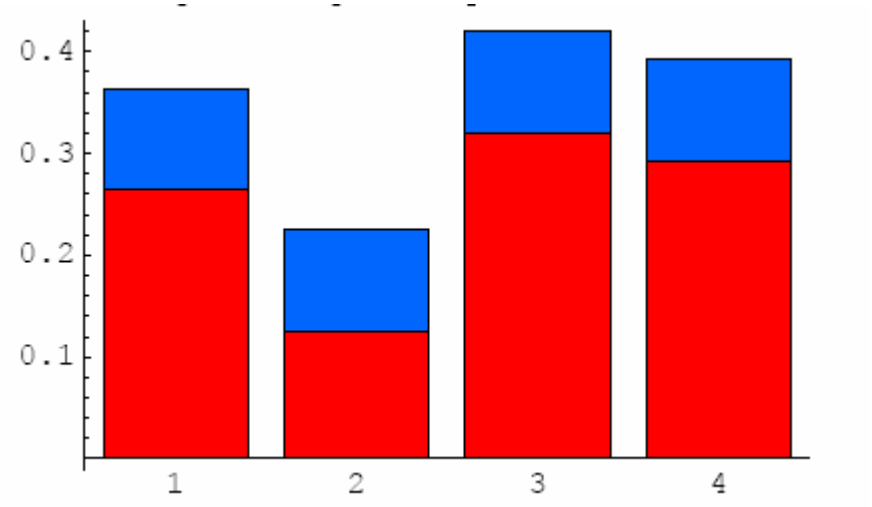
Оптимальні стратегії завоювання ринку першим гравцем



Якщо інфляція змінюється по районах, то відповідна діаграма відображена нижче.

```
In[19]:= col3 = StackedBarChart[X→,
            {0.1,0.1,0.1,0.1},
            PlotLabel– f"Оптимальні стратегії завоювання \ n   ринку першим гравцем"];
```

Оптимальні стратегії завоювання ринку першим гравцем



Числові значення стратегії першого гравця на першому об'єкті дорівнють:

$In[20] := X_1 // N$

$Out[20] = 0.263889$

або те ж саме в відсотках дорівнює:

$In[19] := (X_1 // N)100$

$Out[19] = 26.3889$

9.4. Оптимальна стратегія розвитку в умовах конкуренції: Розв'язок задачі в *Min/Max* термінах

9.4.1. Розв'язок задачі завоювання ринку другим гравцем

Як було зазначено раніше, для першого гравця задача оптимальної стратегії завоювання ринку складається в знаходженні мінімум інвестицій по інвестованих об'єктах,

$$\sum_{i=1}^{4} r_i \Rightarrow \min \tag{9.19}$$

при таких зв'язках:

$$\begin{pmatrix} 2 & 3 & 1 & 4 \\ 1 & 2 & 5 & 4 \\ 2 & 3 & 4 & 1 \\ 4 & 2 & 2 & 2 \end{pmatrix}^T \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}; r_i \geq 0. \tag{9.20}$$

де вектор $\vec{r} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}$ - являє собою невідомий вектор стратегії другого гравця.

Для другого гравця знаходимо максимум інвестицій:

$$\sum_{i=1}^{4} g_i \Rightarrow \max \tag{9.21}$$

при таких зв'язках

$$\begin{pmatrix} 2 & 3 & 1 & 4 \\ 1 & 2 & 5 & 4 \\ 2 & 3 & 4 & 1 \\ 4 & 2 & 2 & 2 \end{pmatrix} \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{pmatrix} \le \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}; g_i \ge 0. \tag{9.22}$$

де вектор $\vec{g} = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{pmatrix}$ - являє собою невідомий вектор стратегії другого гравця.

Комп'ютерні коди для розв’язку в термінах *MinMax* операторів наведені нижче.

$In[21] := Quit[];$

$In[1] := Clear[x];$

$In[20] := \vec{x} = \{r_1, r_2, r_3, r_4\};$

$\vec{c} = \{1,1,1,1\};$

$\hat{M} = \{\{2,3,1,4\},\{1,2,5,4\},\{2,3,4,1\},\{4,2,2,2\}\}; \vec{b} = \{1,1,1,1\}$.

Оптимальні стратегії для другого гравця наведені нижче.

$In[23] := strategySecond = Minimize[\vec{c} . \vec{x}, \{Thread[\hat{M} . \vec{x} \ge \vec{b}]\} // Flatten, \vec{x}]$.

$Out[23] = \{\frac{9}{23}, \{r_1 \to \frac{5}{46}, r_2 \to \frac{7}{46}, r_3 \to \frac{3}{46}, r_4 \to \frac{3}{46}\}\}$

Відповідний обсяг інвестицій дорівнює:

$In[24] := v = strategySecond[[1]]^{-1}$

$Out[24] = \frac{23}{9}$.

Для другого гравця знаходимо максимум інвестицій

$In[25] := \vec{y} = \{g_1, g_2, g_3, g_4\};$

$In[26] := strategySecond = Maximize[\vec{c} . \vec{y}, \{Thread[\hat{M} . \vec{y} \le \vec{b}]\} // Flatten, \vec{y}].$

$Out[26] = \{\frac{9}{23}, \{g_1 \to \frac{5}{46}, g_2 \to \frac{7}{46}, g_3 \to \frac{3}{46}, g_4 \to \frac{3}{46}\}\}$

Оптимальні стратегії другого гравця розв'язуються таким кодом:

$In[27] := \vec{Y} = \{Y_1, Y_2, Y_3, Y_4\} = vTable[strategySecond[[2, i, 2]], \{i, 4\}]$

$Out[27] = \{\frac{5}{18}, \frac{7}{18}, \frac{1}{6}, \frac{1}{6}\}$

9.4.2. Візуалізація оптимальних стратегій другого гравця

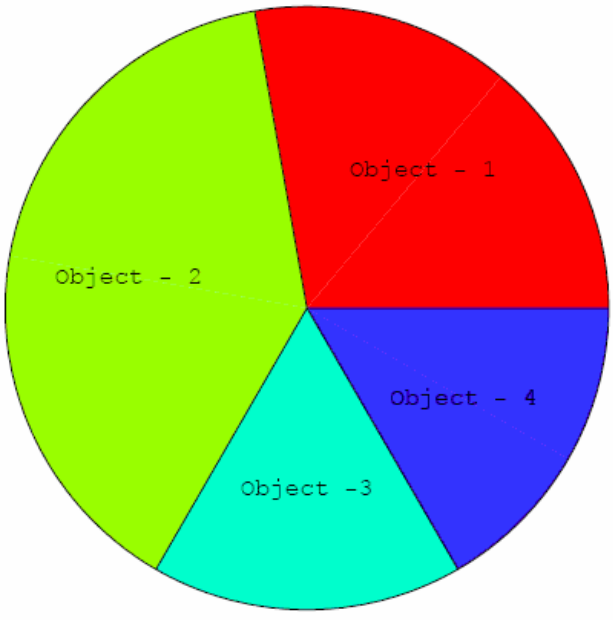
Для візуалізації розв'язків, які за змістом розв'язуваної задачі відповідають оптимальним стратегіям, реалізованим другим гравцем, спочатку завантажуємо пакет

$In[11] :=<< Graphics'Graphics'$

Далі представимо розв'язок у вигляді сегментів ділення загальних інвестицій по об'єктах 1-4 для другого гравця.

```
In[28] := diagr2 = PieChart[Y,
          PieLabels– f {"Object – 1","Object – 2","Object – 3","Object – 4"},
          PlotLabel– f"Оптимальні сратегії завоювання \ n   ринку другим гравцем"
          ];
```

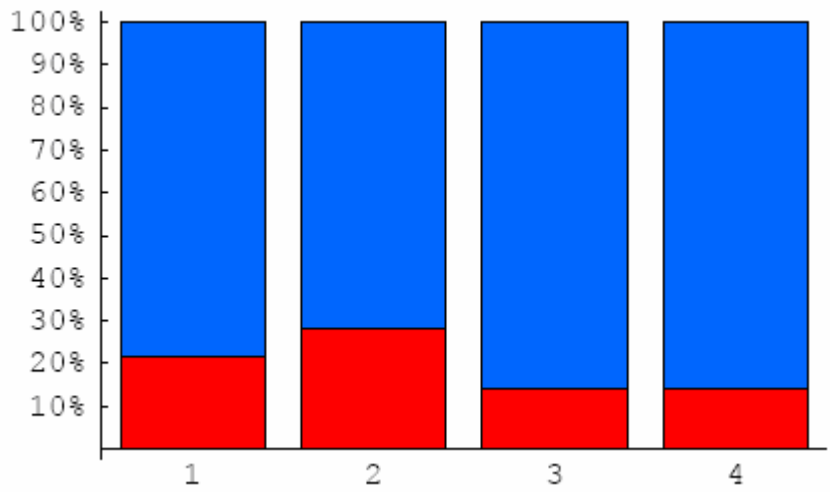
Оптимальні стратегії завоювання ринку другим гравцем



Реалізація стратегій гравця у вигляді іншої діаграми представлена нижче:

```
In[29] := col2 = PercentileBarChart[Y→,
          {1,1,1,1},
          PlotLabel – f"Оптимальні стратегії завоювання \ n  ринку другим гравцем"];
```

Оптимальні стратегії завоювання ринку другим гравцем



Перевірка оптимального розв’язку для другого гравця представлена нижче

```
In[13] := Σ_{i=1}^{4} Y_i
Out[13] = 1
```

Зіставлення стратегій завоювання ринку

Спочатку розглянемо зіставлення стратегій, які будуть виконуватись гравцями, розглянуті і зіставлені на кругових діаграмах.

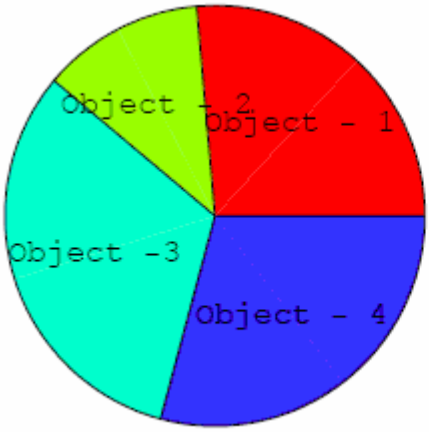
Як видно із зіставлень інвестиції гравців на об'єктах 2-4 будуть істотно відрізнятися один від одного.

У той же час інвестиції в перший об'єкт вкладені гравцями, практично однакові, які можна використати керівникові при розв’язанні питання про залучення гравців у свій регіон.

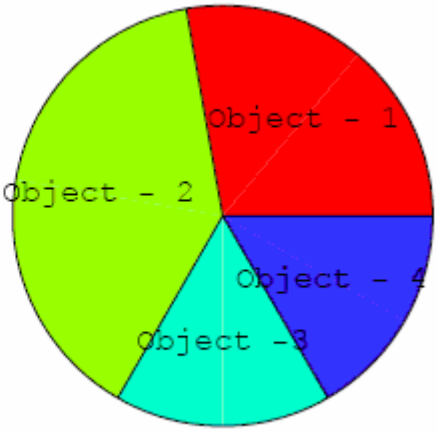
```
In[30] := gr = Show[GraphicsArray[{diagr1, diagr2}], PlotLabel – f
          "Оптимальні стратегії завоювання \ n  ринку першим і другим гравцем"];
```

Оптимальні стратегії завоювання ринку обома гравцями

Оптимальні стратегії завоювання ринку першим гравцем



Оптимальні стратегії завоювання ринку другим гравцем



Те ж саме бачимо на діаграмах зі стовпцями

```
In[31]:= gr2 = Show[GraphicsArray[{col1, col2}], PlotLabel– f
                "Оптимальні стратегії завоювання \ n   ринку першим і другим гравцем"];
```

Оптимальні стратегії завоювання ринку обома гравцями

Оптимальні стратегії завоювання ринку першим гравцем

Оптимальні стратегії завоювання ринку другим гравцем

Таким чином, відбувається вибір "кращої" стратегії розвитку регіону, від якої залежить план інвестицій в об'єкти і від вибору якого залежить вибір однієї зі стратегій, запропонованих двома конкуруючими "гравцями". Так, наприклад, розвиток території або продуктивних сил остаточно будуть залежати від управлінського рішення для гравців. Інші умови, від яких залежить управлінське рішення, в цій моделі не враховуються.

До таких умов можуть відноситись, наприклад, розвиток соціальної сфери чи взагалі певного регіону, на якому знаходиться об'єкт інвестування.

9.5. СИСТЕМИ МАСОВОГО ОБСЛУГОВУВАННЯ

У цьому розділі розглядається задача оцінки ефективності роботи морського порту, як системи масового обслуговування.

9.5.1. Морський порт для обробки (розвантаження/завантаження) суден: Система масового обслуговування з очікуванням і необмеженим потоком обмежень

Ці системи відрізняються такими особливостями функціонування:

- система обслуговування складається з обмеженої кількості n апаратів;
- кожний апарат здатний одночасно обслуговувати тільки одну вимогу;
- кожна нова вимога, яка надходить заставши всі апарати уже зайнятими, стає в чергу і перебуває в ній доти, поки один з апаратів не звільниться.

Якщо вимога надходить у систему, коли є вільний апарат, вона відразу ж приймається на обслуговування.

Функціонування системи розглядається за умови надходження в неї пуассонівського потоку вимог. Джерело потоку вимог необмежене за своїми можливостями (наприклад, пасажири в метро, покупці в магазинах та ін.), хоча щільність потоку λ має кінцеве значення. Час обслуговування кожної вимоги є випадковою величиною, яка підпорядковується закону розподілу з параметром μ. Всі прилади системи мають однакову продуктивність.

Як основні показники роботи системи пропонується ймовірність того, що всі апарати вільні або зайняті, математичне очікування довжини черги, коефіцієнти зайнятості і простою приладів обслуговування.

Можливі стани такої системи масового обслуговування в процесі її функціонування описуються системою диференціальних рівнянь:

$$
\begin{aligned}
&p_0'(t) = -l\,p_0(t) + m p_1(t);\\
&\ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\\
&p_k'(t) = -(l + km)\,p_k(t) + (1+k)m p_{k+1}(t) + l\,p_{k-1}(t);\ \text{при}\ 1 \le k \le n;\ ,\\
&\ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\\
&p_k'(t) = -(l + nm)\,p_k(t) + nm p_{k+1}(t) + l\,p_{k-1}(t);\ \text{при}\ k \ge n
\end{aligned}
\qquad (9.23)
$$

де p_0 і p_k — імовірності станів, коли в системі відповідно жодної або k вимог.

Розглянемо стаціонарний стан системи, при якому $t \to \infty, \text{і}\, p_k'(t) \to 0, p_k(t) \to p_{k0}$. У цьому випадку рівняння станів запишемо в такому вигляді:

$$0 = -l p_0(t) + m p_1(t);$$
$$\ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots$$
$$0 = -(l + km) p_k(t) + (1+k) m p_{k+1}(t) + l p_{k-1}(t); \text{ при } 1 \le k \le n; \quad . \tag{9.24}$$
$$\ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots$$
$$0 = -(l + nm) p_k(t) + nm p_{k+1}(t) + l p_{k-1}(t); \text{ при } k \ge n$$

Нормована умова

$$\sum_{k=0}^{\infty} p_k = 1. \tag{9.25}$$

Не зупиняючись на виведенні залежностей для визначення всіх перерахованих показників, отриманих для стаціонарного стану системи, приводимо лише розрахункові формули.

1. Параметр

$$a = \frac{l}{m}, \tag{9.26}$$

де λ - щільність вхідних потоків вимог;

μ - параметр показового закону часу обслуговування вимог у системі.

2. Імовірність того, що всі обслуговуючі прилади вільні:

$$P_0 = \frac{1}{\left(\sum_{k=0}^{n-1} \frac{a^k}{k!}\right) + \frac{a^n}{(n-1)!(n-a)}}, \text{ при } \frac{a}{n} < 1, \tag{9.27}$$

де n - кількість обслуговуючих приладів у системі.

3. Імовірність того, що зайнято обслуговуванням k приладів (k вимог у системі):

$$P_k = \frac{a^k}{k!} P_0, \text{ при } 1 \le k \le n \quad . \tag{9.28}$$

4. Імовірність того, що всі прилади системи зайняті ($k \le n$):

$$p_z = \frac{a^n}{(n-1)!(n-a)} P_0, \text{ при } \frac{a}{n} < 1 \ . \tag{9.29}$$

5. Імовірність того, що всі апарати зайняті обслуговуванням і y вимог знаходяться в черзі:

$$P_{n+s} = \frac{a^{n+s}}{n! n^s} P_0, \text{ при } s > 0; \tag{9.30}$$

6. Імовірність того, що час перебування вимоги в черзі більше деякої величини t.

$$p(t > t) = p_z e^{-m(n-a)t} \ . \tag{9.31}$$

7. Середній час очікування вимогою початку обслуговування в системі:

$$t_{wait} = \frac{p_z t_{serv}}{(n-a)}, \text{ при } \frac{a}{\text{n}} < 1, \tag{9.32}$$

де $t_{serv} = \frac{1}{m}$ - середнє значення обслуговування вимог у системі.

8. Середня довжина черги:

$$M_{wait} = \frac{aP_n}{n(1-\frac{a}{n})^2} \tag{9.33}$$

9. Середня кількість вимог, що перебувають у системі:

$$M = M_{wait} + \frac{nP_n}{(1-\frac{a}{n})} + P\sum_{k=1}^{n-1}\frac{a^k}{(k-1)!} \quad . \tag{9.34}$$

10. Середнє число вільних від обслуговування приладів:

$$N_0 = \sum_{k=0}^{n-1}\frac{(n-k)a^k}{k!}P_0 \, . \tag{9.35}$$

11. Коефіцієнт простою приладів:

$$K_{st} = \frac{N_0}{n} . \tag{9.36}$$

12. Середня кількість зайнятих обслуговуванням приладів:

$$N_z = n - N_0 . \tag{9.37}$$

13. Коефіцієнт завантаження приладів:

$$K_{load} = \frac{N_z}{n} . \tag{9.38}$$

14. Економічний показник для вибору кращого варіанта системи обслуговування при її проектуванні:

$$G_{lost} = (l g_{wiat} + N_0 g_{k,unit} + N_z g_k) t_{wait} , \tag{9.39}$$

де G_{lost} - величина втрат у системі за час t_{wait};

g_{wait} - вартість втрат, пов'язаних із простоюванням вимог у черзі протягом одиниці часу;

$g_{k,unit}$ - вартість одиниці часу простою обслуговуючого приладу системи;

g_k - вартість експлуатації приладу при обслуговуванні за одиницю часу.

9.5.2. Розрахунок ефективності роботи морського порту: Варіант розрахунку з фіксованою кількістю причальних комплексів

Як приклад розрахунку ефективності роботи морського порту розглянемо морський порт із фіксованою та обмеженою кількістю причалів і відомих кількісних характеристик його роботи потоку суден.

9.5.2.1. Формулювання задачі масового обслуговування для морського порту

Вихідні дані для постановки задачі розрахунку ефективності морського порту:

Робота морського порту характеризується наступними об'єктивними показниками, які можна сформулювати так.

Зовнішні, не залежні від порту, обставини:

Надходження суден у порт має випадковий характер, тому що вони виходять із різних портів і долають різні відстані до пункту розвантаження. Крім того, на швидкість руху суден впливає погода. Дослідження статистики частоти приходу суден у порт свідчить, що надійшовші на розвантаження судна утворюють пуассонівський потік. Час розвантаження кожного судна є також випадковою величиною, яка залежить від тоннажу суден, особливості вантажу і багатьох інших причин.

Об'єктивні (внутрішні) показники морського порту:

- Морський порт має $n = 5$ причалів для розвантаження сухо вантажних суден.
- В середньому протягом місяця в порт прибуває з вантажами близько 20 суден великого тоннажу.
- В середньому на розвантаження судна витрачається на причалі 6 робочих днів.

Питання на які необхідно дати відповідь:

- Потрібно оцінити роботу порту, тобто оцінити його економічну ефективність.
- Необхідно також розглянути можливість збільшення пропускної здатності порту за рахунок збільшення кількості причалів.

Умови, що накладаються на розв’язання цього питання:

■ При розв’язанні цієї задачі потрібно досліджувати економічну доцільність розширення можливості порту по розвантаженню і навантаженню суден.

9.5.2.2. Приклад одиничного розрахунку з наперед заданою кількістю причалів. *Розв'язання задачі:*

Вихідні дані для розрахунку:

n = 5; (кількість причалів *)*

t_s = 6; (середній час розвантаження одного судна *)*

T = 30; (період роботи *)*

λ = 20; (інтенсивність надходження суден у порт *)*

Розв'язок питання по визначенню ефективності роботи порту:

1. Варто визначити параметр *α:*

$$t_{load} = \frac{t_s}{T};$$
$$a = l\,t_{load}$$

2. Імовірність того, що всі причали вільні і очікують судна під розвантаження:

$$P_0 = \frac{1}{\left(\sum_{k=0}^{n-1} \frac{a^k}{k!}\right) + \frac{a^n}{(n-1)!(n-a)}};$$

P_0 // N

0.012987

3. Імовірність того, що всі причали зайняті суднами під розвантаження:

$$p_t = \frac{a^n}{(n-1)!(n-a)} P_0;$$

p_t // N

0.554113

Це означає, що приблизно 56% часу всі причали повністю зайняті вантажно-розвантажувальними роботами з наявними на причалах суднами.

4. В середньому час очікування кожним судном початку розвантаження дорівнює

$$t_{wait} = \frac{p_t t_s}{(n-a)};$$

t_{wait} // N

3.32468

5. Визначимо середню кількість суден, які будуть перебувати в порту, очікуючи своєї черги для розвантаження:

$$M_{wait} = \frac{ap_t}{n(1-\frac{a}{n})^2};$$

M_{wait} // N

11.0823

тобто 11.1 судна.

6. Імовірність того, що в порту на обслуговуванні перебувають шість суден ($n = 6$ суден):

$$p_b = Table[P_k = \frac{a^k}{k!}P_0, \{k,0,7\}];$$

p_b // N

{0.012987,0.0519481,0.103896,0.138528,0.138528,0.110823,0.0738817,0.0422181}

p_b[[6]]// N

0.110823

7. Середня кількість суден, що перебувають у порту:

$$M = M_{wait} + \frac{np_b[[n]]}{(1-\frac{a}{n})} + P_0\sum_{k=1}^{n-1}\frac{a^k}{(k-1)!};$$

M // N

15.7749

8. Визначимо середню кількість причалів, що простоюють:

$$N_0 = \sum_{k=0}^{n-1}\frac{(n-k)a^k}{k!}P_0;$$

N_0 // N

1.

Таким чином, в середньому за місяць простоює один причал.

9. Коефіцієнт простою причалів дорівнює:

$$K_{st} = \frac{N_0}{n};$$

K_{st} // N

0.2

Висновок щодо роботи порту: Це означає, що кожний причал буде простоювати 20% часу.

Управлінське розв’язання:

Для зменшення часу простою суден вирішено порт розширити.

Можливі варіанти розв’язання питання:

Варіант 1: Зовнішні фактори економічного зростання.

При цьому необхідно розглянути збільшення щільності потоку суден λ у порт, виходячи з тенденції розвитку судноплавства в цьому районі, а також збільшення швидкості руху суден і т. д.

Варіант 2: Внутрішні фактори економічного зростання. Інвестиції в розширення порту.

Нехай для прикладу щільність приходу суден у порт зберігається колишньою.

Скільки потрібно мати причалів у порту, щоб істотно зменшити кількість суден, які очікують розвантаження, і час їхнього простою?

Для розв'язання цього питання виконаємо весь комплекс розрахунків для кількості причалів n = 5, 6, 7, 8.

9.5.3. Розрахунок оптимізації кількості причалів морського порту

Розв'язання задачі полягає у визначенні оптимальної кількості причалів морського порту при заданій інтенсивності потоку суден.

Для побудови потрібних графіків завантажуємо пакет:

$In[1] := << Graphics'Graphics'.$

Вихідні дані для розрахунку по вантажопотоку суден у морському порту такі:

$In[2] := t_s = 6;$ (* середній час розвантаження одного судна*)

$T = 30;$ (* період роботи*)

$l = 20;$ (* інтенсивність надходжень суден у порт*)

Розв'язання питання щодо визначення економічної ефективності роботи морського порту здійснюємо за такою методикою.

1. Варто визначити параметр α:

$$In[5] := t_{load} = \frac{t_s}{T};$$

$$In[6] := a = l\, t_{load}$$

Далі всі параметри ефективності роботи морського порту будемо знаходити як функцію кількості причалів цього ж порту.

У попередньому розділі кількість причалів була фіксованою величиною і дорівнювала $n=5$. Далі кількість причалів у моделі розрахунку буде змінною, що дозволить вибрати потім оптимальне її число.

2. Імовірність того, що всі причали вільні і очікують судна під розвантаження:

$$In[7] := P_0 = Table[\frac{1}{\left(\sum_{k=0}^{n-1} \frac{a^k}{k!}\right) + \frac{a^n}{(n-1)!(n-a)}}, \{n,5,8\}];$$

Порівнюючи з попереднім розрахунком, одержимо ту ж імовірність, що і раніше при $n = 5, \ldots$.

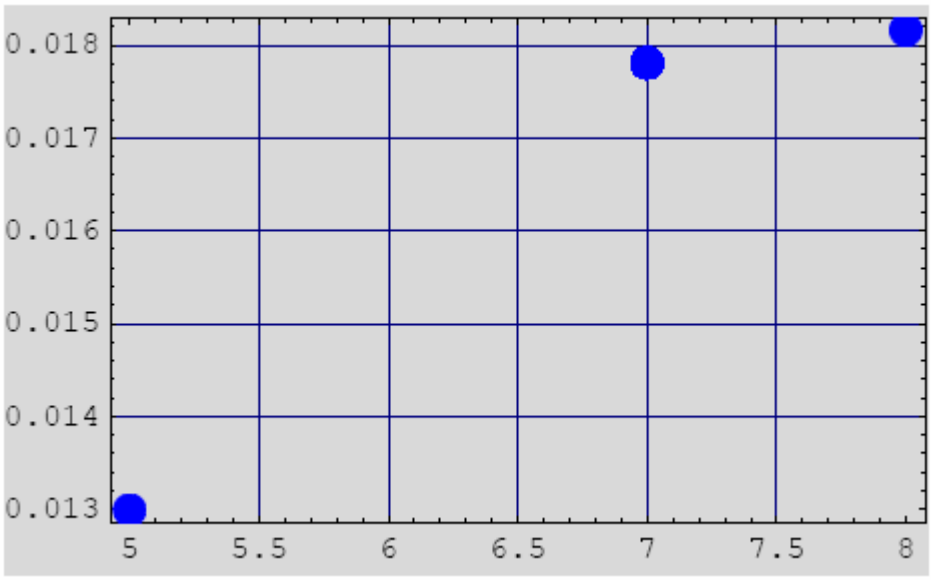
$P_0[[1]] // N$

0.012987

Тут кількість причалів змінювалася від 5 до 8, тому ймовірність P_0 є змінною величиною.

Графік залежності ймовірності того, що всі причали вільні залежно від кількості причалів, представлений нижче.

$$In[8] := ListPlot[Table[\{s+4, P_0[[s]]\}, \{s,1,4\}], PlotStyle- \mathbf{f} \{Point Size[.04], RGBColor[0,0,1]\},$$
$$Background- \mathbf{f} GrayLevel[.85], Frame- \mathbf{f} True, GridLines- \mathbf{f} Automatic];$$



Як видно з наведеного графіка, із збільшенням кількості причалів імовірність того, що всі причали вільні, *збільшується.* Що закономірно, тому що за умовою задачі інтенсивність приходу суден у порт не збільшується, а ймовірність вільного причалу збільшується.

3. Імовірність того, що всі причали зайняті суднами під розвантаження:

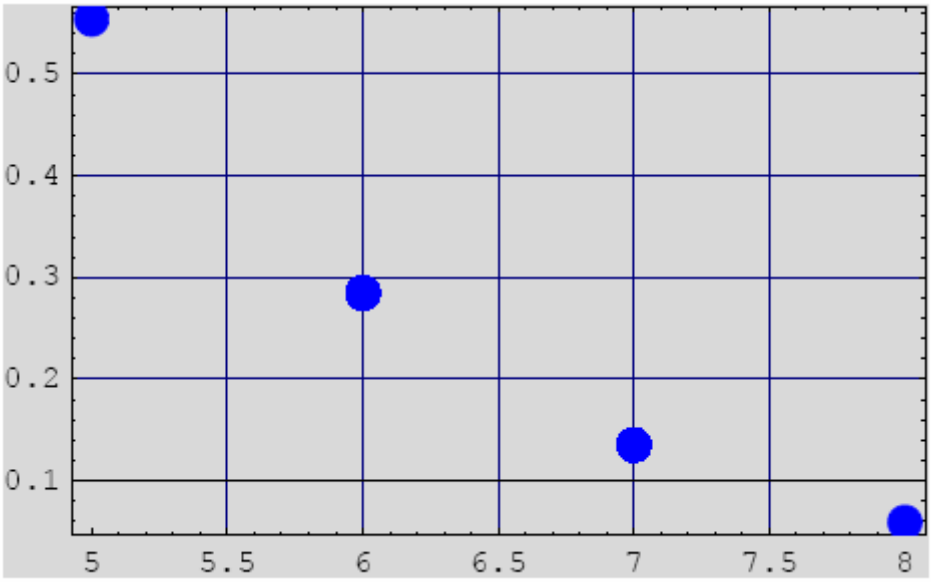
$$In[9] := p_t = Table[\frac{a^n}{(n-1)!(n-a)} P_0[[n-4]], \{n,5,8\}];$$

$$In[10] := p_t[[4]] // N$$

$$Out[10] = 0.059044$$

Графік залежності ймовірності зайнятості причалів від кількості причалів у порту.

$$In[11] := ListPlot[Table[\{s+4, p_z[[s]]\}, \{s,1,4\}], PlotStyle- \mathbf{f} \{Point Size[.04], RGBColor[0,0,1]\},$$
$$Background- \mathbf{f} GrayLevel[.85], Frame- \mathbf{f} True, GridLines- \mathbf{f} Automatic];$$



Графік залежності $\pi_z = \pi_z(n)$ показує, що ймовірність зайнятості причалів від наявності причалів для вантажно-розвантажувальних робіт у порту падає при їхньому надлишку приблизно від 56% до 6 %, коли всі причали повністю зайняті звичайними для них роботами.

4. В середньому час очікування кожним судном початку його обробки дорівнює:

$$In[12] := t_{wait} = Table[\frac{p_z[[n-4]] t_s}{(n-a)}, \{n,5,8\}];$$

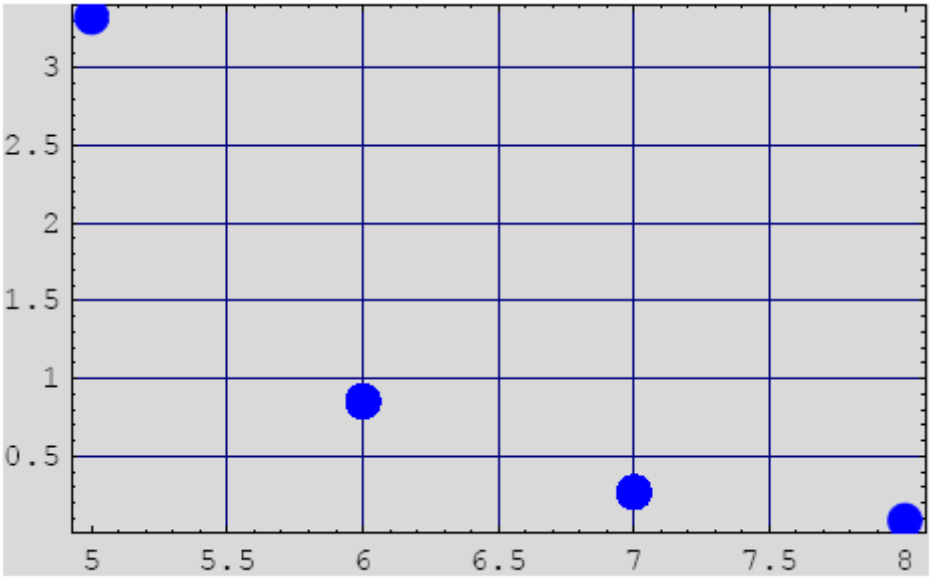
Так, при наявності восьми причалів середній час очікування дорівнює приблизно 2.1 години.

$$t_{wait}[[4]] 24 // N$$

2.12558

Графік часу очікування з даними початку їхньої обробки, як функція кількості причалів порту представлена нижче.

$$In[13] := ListPlot[Table[\{s+4, t_{wait}[[s]]\}, \{s,1,4\}], PlotStyle- \mathbf{f} \{Point Size[.04], RGBColor[0,0,1]\},$$
$$Background- \mathbf{f} GrayLevel[.85], Frame- \mathbf{f} True, GridLines- \mathbf{f} Automatic];$$



Як видно із графіка, час очікування розвантаження зменшується від 3.5 доби (при наявності п’яти причалів у порту) до 2.1 години (при наявності восьми причалів).

5. Далі визначимо середню кількість суден, що буде перебувати в порту, очікуючи своєї черги для розвантаження:

$$In[14] := M_{wait} = Table[\frac{ap_z[[n-4]]}{n(1-\frac{a}{n})^2}, \{n,5,8\}];$$

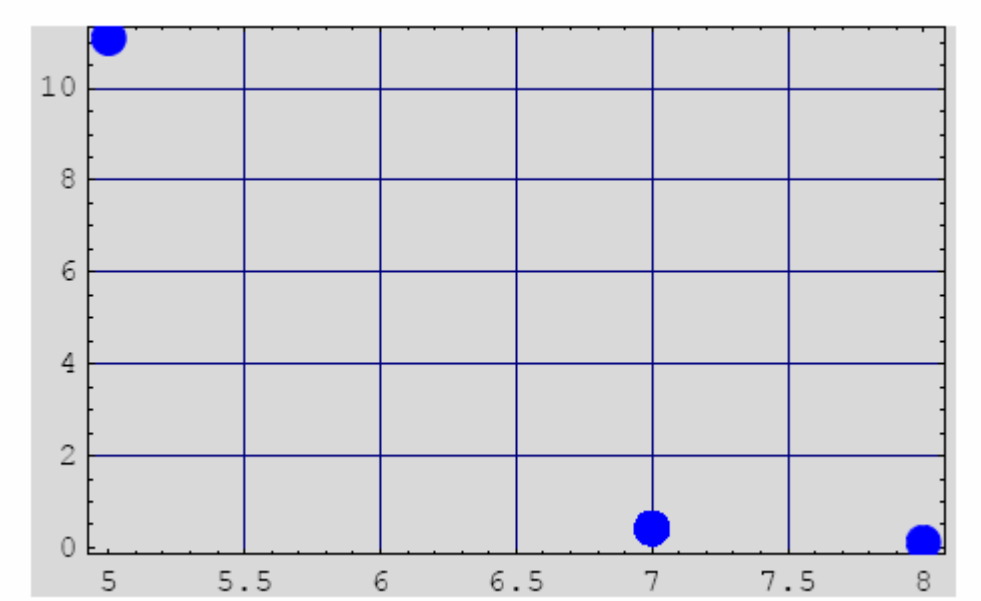
Так, наприклад, середня кількість суден, які ждуть своєї обробки при наявності восьми причалів, становить менше одного судна на місяць.

$$M_{wait}[[4]] // N$$

0.118088

Графік залежності кількості суден, які перебувають у порту, очікуючи своєї черги на розвантаження, як функція кількості причалів представлений нижче.

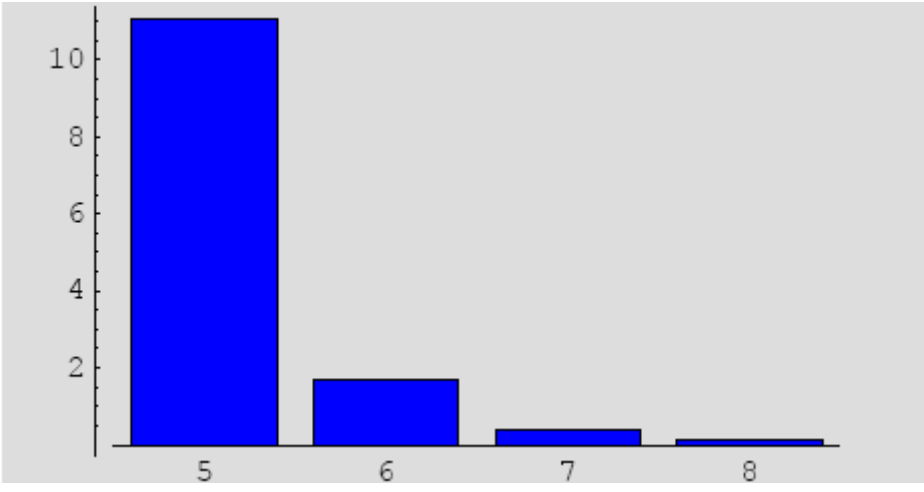
$$In[15] := ListPlot[Table[\{s+4, M_{wait}[[s]]\}, \{s,1,4\}], PlotStyle- \mathbf{f} \{Point Size[.04], RGBColor[0,0,1]\},$$
$$Background- \mathbf{f} GrayLevel[.85], Frame- \mathbf{f} True, GridLines- \mathbf{f} Automatic];$$



Останній результат зручно представити у вигляді діаграми, як це показано нижче.

```
In[16]:= grNumberShip =
  BarChart[Table[{M_wait[[s]], s + 4}, {s,1,4}], BarStyle→ {RGBColor[0,0,1]}, PlotRange→ All,
  PlotLabel→ "Кількість суден у порту, очікуючих розвантаження \ n при
  різній кількості причалів", Background→ GrayLevel[.85]];
```

Число суден у порту, очікуючих розвантаження при різній кількості причалів



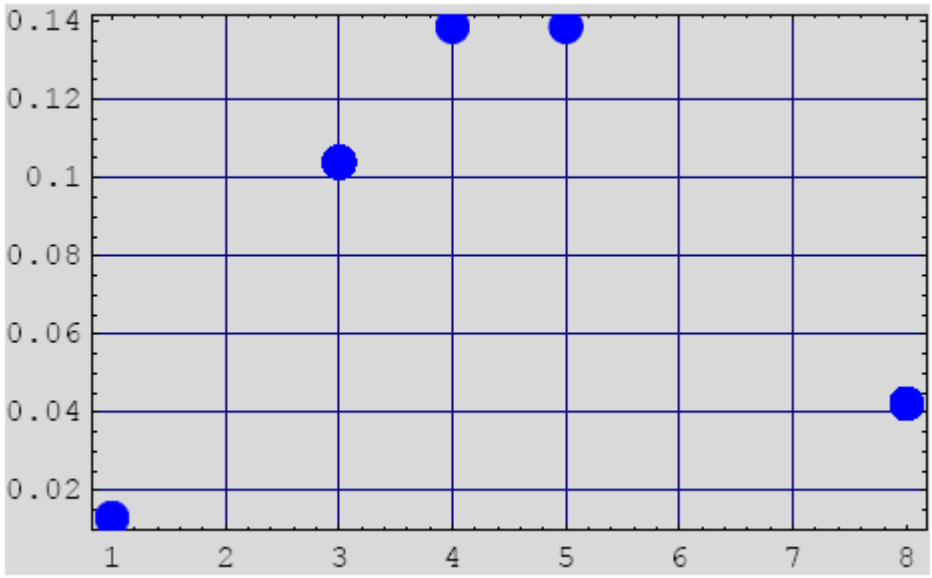
Як видно із графіка, кількість суден очікуючих розвантаження, зменшується від 11 суден (є п’ять причалів) до 0.1 судна при наявності восьми причалів.

6. Імовірність того, що в порту на обслуговуванні перебувають шість суден (n = 6 суден):

$$In[17] := p_b = Table[\frac{a^k}{k!} P_0[[1]], \{k,0,7\}];$$

Графік імовірності того, що в порту на обслуговуванні перебувають від 0 до 7 суден:

$$In[18] := ListPlot[p_b, PlotStyle- \mathbf{f} \{Point Size[.04], RGBColor[0,0,1]\},$$
$$Background- \mathbf{f} GrayLevel[.85], Frame- \mathbf{f} True, GridLines- \mathbf{f} Automatic];$$



7. Середня кількість суден, які знаходяться в порту:

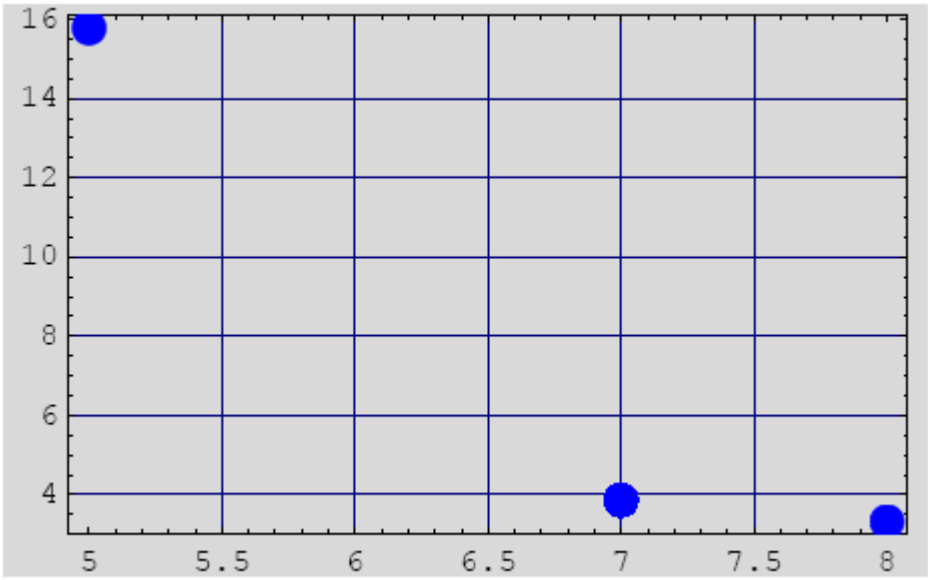
$$In[19] := M = Table[M_{wait}[[n-4]] + \frac{np_b[[n]]}{(1-\frac{a}{n})} + P_0[[1]] \sum_{k=1}^{n-1} \frac{a^k}{(k-1)!}, \{n,5,8\}];$$

Середнє кількість суден у порту при наявності в останньому восьми причальних комплексів дорівнює:

$$In[20] := M[[4]] // N$$
$$Out[20] = 3.31594$$

Графік середньої кількості суден, які знаходяться в порту, представлений нижче.

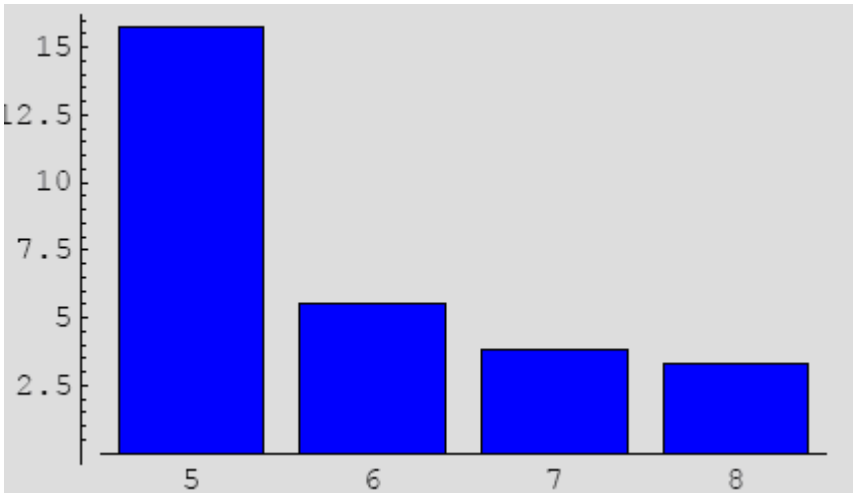
$$In[21] := ListPlot[Table[\{s+4, M[[s]]\}, \{s,1,4\}], PlotStyle- \mathbf{f} \{Point Size[.04], RGBColor[0,0,1]\},$$
$$Background- \mathbf{f} GrayLevel[.85], Frame- \mathbf{f} True, GridLines- \mathbf{f} Automatic];$$



Діаграма кількості суден у порту при зміні в порту кількості причалів представлена нижче.

```
In[23]:= grNumberShip =
  BarChart[Table[{M[[s]], s + 4}, {s,1,4}], BarStyle– f {RGBColor[0,0,1]}, PlotRange– f All,
  PlotLabel – f"Кількість суден, які знаходятьс я в порту \ n, при різному числі причалів",
  Background – f GrayLevel[.85]];
```

Кількість суден, які знаходяться в порту, при різній кількості причалів



Як видно із графіка, середня кількість суден, які знаходяться в порту, зменшується від майже 16 (коли в порту п’ять причалів) до трьох суден при наявності восьми причалів.

8. Визначимо середню кількість причалів, що простоюють:

$$In[24] := N_0 = Table\sum_{k=0}^{n-1}\frac{(n-k)a^{k}}{k!}P_0[[1]],\{n,5,8\}];$$

Кількість причалів, що простоюють, за умови, що в порту є вісім причальних комплексів дорівнює:

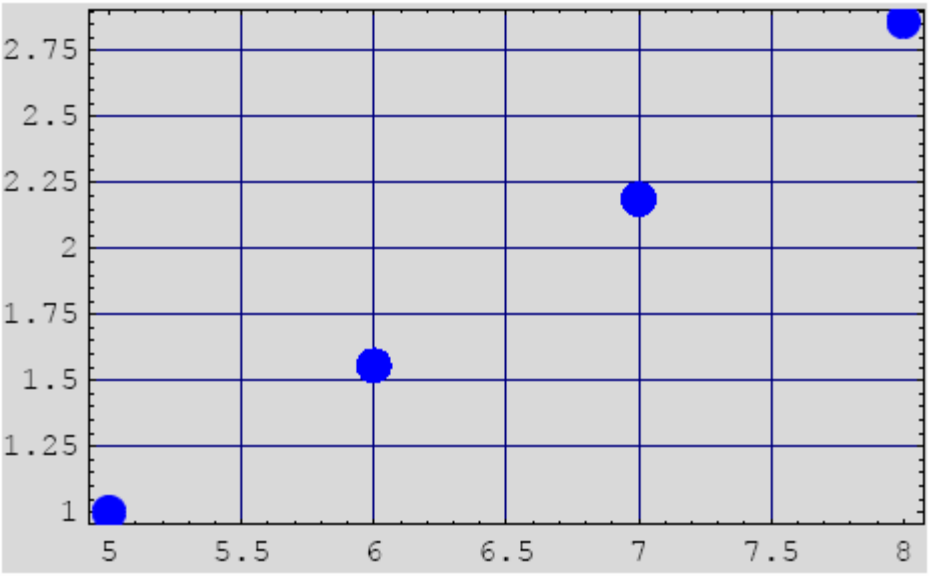
$In[25] := N_0[[4]] // N$

$Out[25] = 2.86011$

Отже, приблизно три причали простоюють протягом одного місяця роботи порту. Графік, який визначає кількість причалів, що простоюють, представлений нижче.

$In[26] := ListPlot[Table[\{s+4, N_0[[s]]\},\{s,1,4\}], PlotStyle- \mathbf{f} \{Po\,\text{int}\, Size[.04], RGBColor[0,0,1]\},$

$Background- \mathbf{f}\ GrayLevel[.85], Frame- \mathbf{f}\ True, GridLines- \mathbf{f}\ Automatic];$



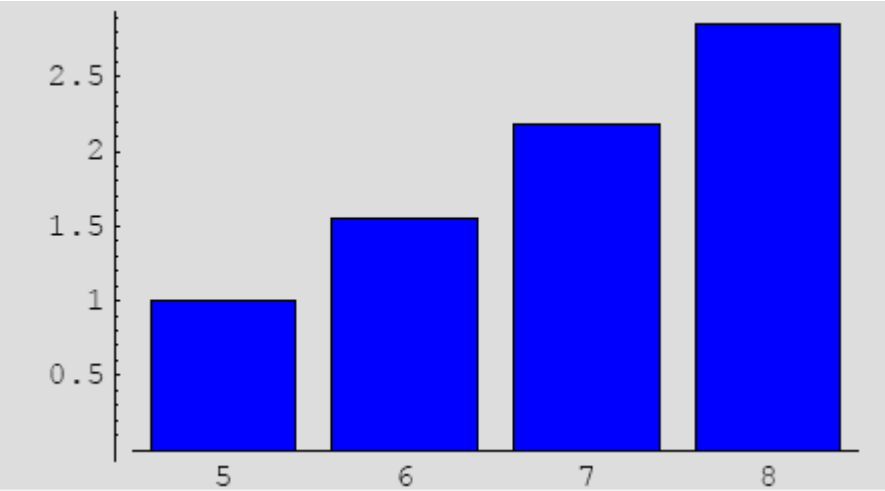
і на відповідній діаграмі.

$In[27] := grNumberPort =$

$BarChart[Table[\{N_0[[s]], s+4\},\{s,1,4\}], BarStyle- \mathbf{f} \{RGBColor[0,0,1]\}, PlotRange- \mathbf{f}\ All,$

$PlotLabel- \mathbf{f}$"Кількість причалів, що простоюють в порту \ n при різній к ількості причалів",
$Background- \mathbf{f}\ GrayLevel[.85]];$

Кількість причалів, що простоюють в порту, при різній кількості причалів



Як видно із представленого графіка, кількість причалів, що простоюють, збільшується від одного (при $n = 5$) до майже трьох причалів на місяць при наявності $n = 8$ причалів.

Таким чином в середньому за місяць збільшується кількість причалів, що простоюють, від одного до трьох.

9. Коефіцієнт простою причалів:

$In[28] := K_{st} = Table[\frac{N_0[[n-4]]}{n}, \{n,5,8\}];$

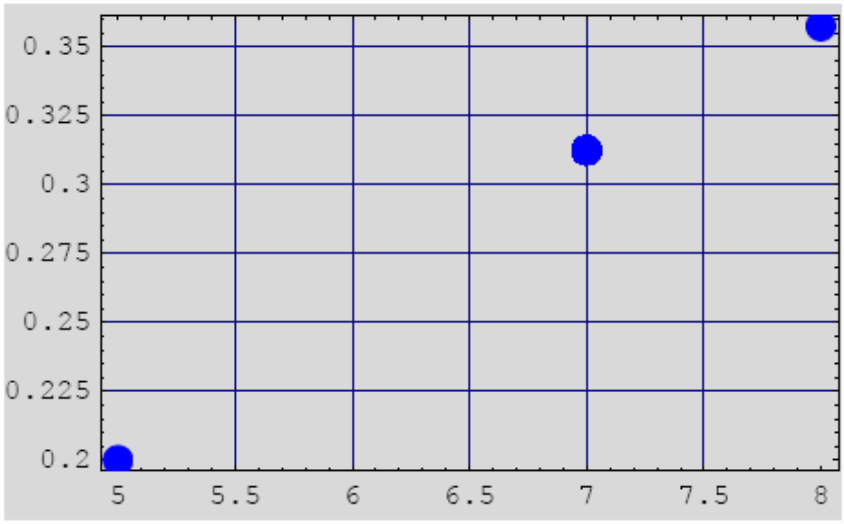
$K_{st}[[3]] // N$

0.312472

Графік коефіцієнта простою причалів представлений нижче.

$In[29] := ListPlot[Table[\{s+4, K_{st}[[s]]\}, \{s,1,4\}], PlotStyle- \mathbf{f} \{Point Size[.04], RGBColor[0,0,1]\},$

$Background- \mathbf{f} GrayLevel[.85], Frame- \mathbf{f} True, GridLines- \mathbf{f} Automatic];$



Як видно із графіка, коефіцієнт простою причалів збільшується пропорційно кількості. Звичайно це справедливо при заданому і незмінному вантажопотоці суден.

Висновок щодо роботи порту: Аналізуючи результати розрахунків, які наведені в графіках вище, можна зробити висновок про те, що із збільшенням кількості причалів від $n = 5$ до $n = 6$ вдається істотно знизити час очікування суден (майже в 4 рази), а кількість суден, що очікують розвантаження — в 6,4 разу. Подальше збільшення кількості причалів приводить до зменшення значення t_{wait} і M, але самі ці величини вже досить малі. Щоб прийняти остаточне рішення, доцільно виконати класичний економічний аналіз.

9.5.4. Розрахунок економічної ефективності при зміні кількості причалів морського порту

Представимо класичний економічний розрахунок ефективності

- Нехай простій кожного судна протягом доби обходиться *судновласникові/державі* g_{wait} = 1000 у.о. вартості.
- В той же час місячний простій причалу порту через несвоєчасний прихід суден обходиться *власникові порту* (державі) $g_{0,wait}$ = 10000 у.о. вартості.
- Вартість місячної експлуатації одного причалу g_p = 100000 у.о. вартості.

Наведені вихідні дані за вартістю аж ніяк не вважаються відповідними реально існуючим витратам у роботі конкретного порту, а взяті лише для ілюстрації порядку проведення розрахунків.

Для проведення економічного аналізу скористаємося формулою, за якою визначається сума витрат за $T = 1$ місяць (30 днів).

$$G_{lost} = lg_{0,wiat} t_{wait} + K_{st} ng_p + ng_p. \qquad (9.40)$$

При виборі оптимального варіанта необхідно вибрати той варіант, для якого ці витрати мінімальні. Вводимо вихідні дані для економічного розрахунку.

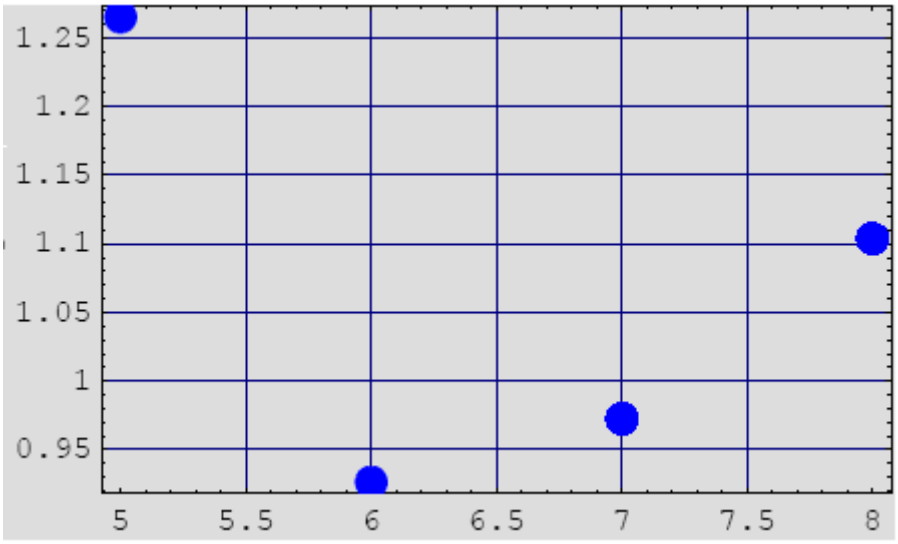
$In[30] := g_{0,wiat} = 10000; g_p = 100000.$

Формула для розрахунку втрат при різних значеннях кількості причалів має вигляд

$In[31] := G_{lost} = Table[(lg_{0,wiat} t_{wait}[[n-4]] + K_{st}[[n-4]] ng_p + ng_p), \{n,5,8\}].$

Графік залежності витрат порту (втрат), як функція кількості причалів представлена нижче

```
In[33] := ListPlot[Table[{s + 4, G_lost[[s]]10^-6}, {s,1,4}], PlotStyle– f {Po int Size[.04], RGBColor[0,0,1]},
          Background– f GrayLevel[.85], Frame– f True, GridLines– f Automatic];
          FrameLabel → {" число причалів ", "витрати млн. ];
```

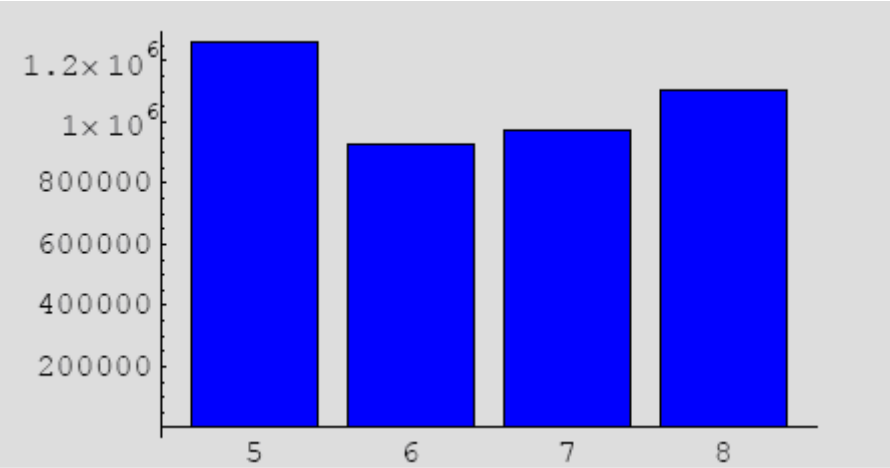


витрати млн. грн.

Як видно із представленого графіка величина втрат роботи порту має локальний мінімум. Цей мінімум відповідає *шести причалам* порту.

```
In[34] := grLost = BarChart[Table[{G_lost[[s]], s + 4}, {s,1,4}], BarStyle → {RGBColor[0,0,1]},
          PlotLabel– f "Величина втрат в у.о власника порту \ n
                    при різній кількості причалів \ n", Background - f GrayLevel[.85]];
```

Величина втрат в у. о. власника порту при різні кількості причалів



Таким чином наведені розрахунки показують, що найбільш ефективним варіантом є порт із шістьома *(n = 6)* причалами.

Числові значення втрат наведені нижче:

$In[35] := G_{lost} \,//\, N$

$Out[35] = \{1.26494 \times 10^6, 926528., 972774., 1.10372 \times 10^6\}$

Значення втрат із шістьома причалами, що відповідає оптимальному значенню втрат порту, наведено нижче.

$In[36] := G_{lost}[[2]] \,//\, N$

$Out[36] = 926528.$

Наведений приклад показує можливість вибору оптимального варіанта проектування порту або інших транспортних пунктів обслуговування.

9.5.5. Висновок по управлінському розв'язку

• Кількість причалів заданого порту (при відомих і незмінних параметрах продуктивності порту) необхідно збільшити до шести.

• Мінімальна сума витрат у цьому випадку дорівнює 92658 у.о.

Отже у цій темі представлені розв'язки наступних задач:

• загальної задачі лінійного програмування;

• задачі оптимізації випуску продукції;

• задачі поліпшення плану випуску продукції та оптимізації ігрових стратегій;

• задачі ефективності роботи морського порту.

Тема 10

СПЕЦІАЛЬНІ ТЕМИ МОДЕЛЮВАННЯ ХАОТИЧНИХ СИСТЕМ

У цій темі як приклад хаотичного поводження динамічної системи наведена методика застосування методів символьної алгебри до вивчення відомої динамічної системи Лоренца.

10.1. Модель Лоренца

Модель Лоренца є першою нелінійною моделлю, у якій був виявлений хаотичний режим поводження динамічної системи, виявлений при числовому розв'язуванні задачі. Хаос, виявлений у моделі, характеризується непередбачуваною, неперіодичною зміною тимчасових змінних динамічної системи, але відтвореною у моделюванні системи. Незважаючи на значну кількість публікацій, присвячених дослідженню хаотичного атрактора Лоренца, висновку самих динамічних рівнянь Лоренца присвячено незначна кількість праць. Тому, у цій темі приводиться висновок методами символьної алгебри динамічних рівнянь Лоренца та їхнє числове розв'язання.

10.1.1. Механічна схема

Виведемо динамічні рівняння Лоренца, які описують конвективні потоки флюїду в задачі Релея про конвекцію у підігрітому знизу (температура підошви $T_0+\Delta T_0$ і T_0 - температура покришки шару) горизонтальному шарі товщини h, коли перебуває і протікає в шарі грузлої, нестислива рідина утворить у русі конвективні комірки. Схема комірок і потоку в них представлена нижче.

```
In[1]:=<< Graphics'Arrow'
In[2]:= Show[Graphics[{Circle[{0, 0}, {2, 3}], Circle[{4 + .1, 0}, {2, 3}],
```

```
Circle[{8 + .2, 0}, {2, 3}], Line[{{-2, - 3}, {10 + .2, - 3}}], Line[{{-2, 3}, {10 + .2, 3}}],
Line[{{-2, - 3}, {-2, 3}}], Line[{{10 + .2, - 3}, {10 + .2, 3}}],
Line[{{2 + .05, - 3}, {2 + .05, 3}}], Line[{{6 + .15, - 3}, {6 + .15, 3}}], Arrow[{2, 0}, {2, 1}],
Arrow[{0, - 3}, {1, - 3}], Arrow[{0, 3}, {-1, 3}], Arrow[{-2, 0}, {-2, - l}],
Text[FontForm["T0 + ΔT0", {"Times - Italic", 14}], {4, - 4 + .7}],
Text[FontForm["T0", {"Times - Italic", 14}], {4, 4 - .5}],
Text[FontForm["v", {"Times - Italic", 16}], {2 - .3, 0}],
Text[FontForm]"u", {"Times - Italic", 16}], {0, - 4 + .5}],
Text[FontForm["h", {"Times - Italic", 16}], {-2 - .3, 3 - .3}]}],
AspectRatio → Automatic, Background → GrayLevel[.85]];
```

From In[2]:=

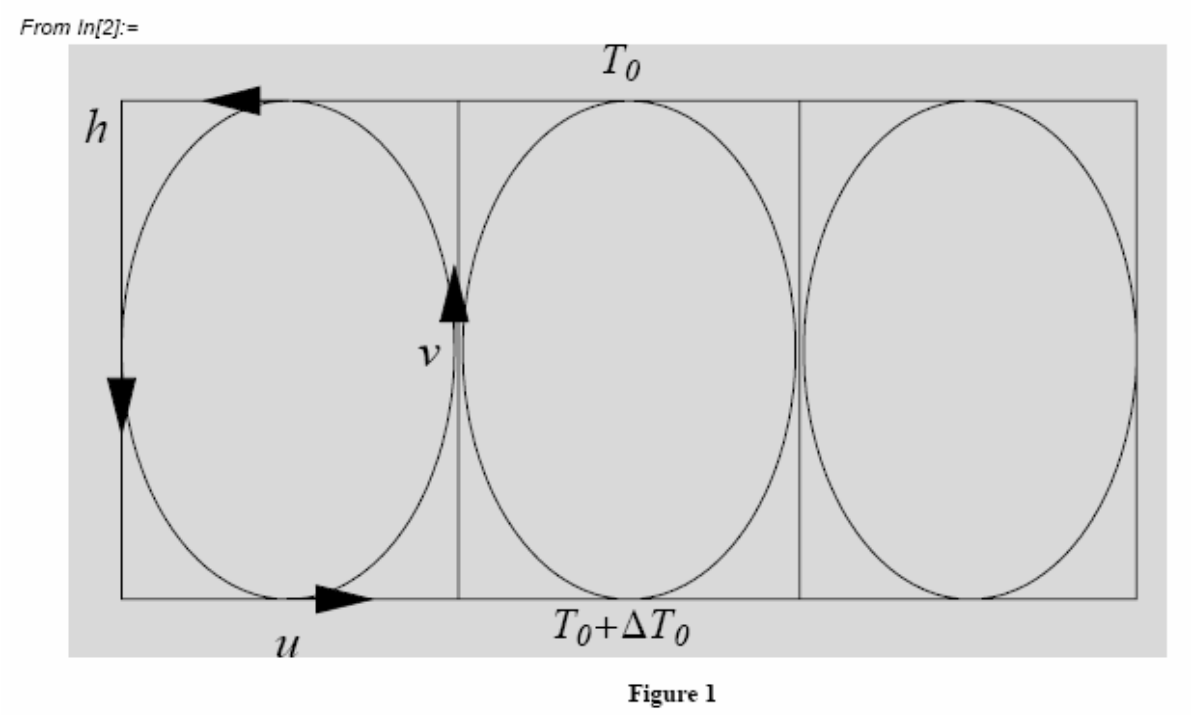


Figure 1

Тут через ***u*** і ***v*** позначені горизонтальна і вертикальна складові швидкості течії.

Як відомо, ця система стала однією з найбільш популярних і відомих динамічних систем, що показує перехід до хаосу і утворення дивних атракторів.

Тут основна увага буде приділена висновку динамічних рівнянь Лоренца, методами комп'ютерної алгебри системи *Mathematica*.

Цей висновок базується на відомій праці Лоренцо - *Lorenz E. Deterministic Non-Periodic Flow// Jounal of Atmospheric Sciences.- 1963. v.20.-p. 130 – 141.*

10.1.2. Динамічні рівняння в’язкої рідини

Спочатку зупинимося на одержанні динамічних рівнянь в’язкої нестисливої рідини в наближенні Буссінеска.

Вихідні дані постановці рівняння руху в’язкої рідини:

- рівняння нерозривності потоку:

$$\frac{\partial p}{\partial t} + p(\nabla . \vec{v}) = 0; \tag{10.1}$$

- рівняння збереження кількості руху:

$$\frac{\partial \vec{v}}{\partial t} + \vec{v}(\nabla . \vec{v}) = \vec{b} - \frac{1}{p}\nabla p + \nu \nabla^2 \vec{v}; \tag{10.2}$$

- рівняння теплопровідності в рідині, що протікає:

$$\frac{\partial T}{\partial t} + T(\nabla . \vec{v}) = k\nabla^2 T; \tag{10.3}$$

- рівняння стану:

$$p = p_0(1 - b_f(T - T_0)). \tag{10.4}$$

Де $\vec{b} = \vec{j}\, g$ - вектор масової сили;

$p = p(x, y, t)$ - поле тиску у потоці;

ν-коефіцієнт кінематичної в'язкості;

k-коефіцієнт температуропровідності;

b_f -коефіцієнт теплового розширення в’язкої рідини.

У наближенні Буссінеска вважаємо рідину малостисненою, тобто залежність щільності від температури враховується в рівняннях руху тільки раз, зокрема в правій частині рівняння збереження кількості руху.

Крім того, вважаємо, що виконуються наступні співвідношення для тиску і температури:

$$\begin{aligned} p &= p_0 - p_0 g(1 - b_f(T - T_0))y + p(x, y, t); \\ T &= T_0 + T(x, y, t). \end{aligned} \tag{10.5}$$

У рівняннях (10.5) $p(x, y, t)$ і $T(x, y, t)$ - збурені тиск і температура. Далі для доданка $\frac{1}{p}\nabla p$ маємо таке розкладання:

$$\begin{aligned} &\frac{1}{p}\nabla p = \\ &\frac{-p_0 g\,\vec{j} + \nabla p(x, y, t)}{p_0(1 - b_f(T - T_0))} \approx -g\,\vec{j} - b_f g(T(x, y, t) - T_0)\,\vec{j} + \frac{1}{p_0}\nabla p(x, y, t) + \ldots \end{aligned} \tag{10.6}$$

10.1.3. Наближення Буссінеска

Якщо вектор швидкості потоку має дві складові і рух відбувається у вертикальній площині, то тоді динамічні рівняння Буссінеска, які виводяться із системи (10.1) - (10.4) мають вигляд:

$$\begin{aligned}
&\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{p_0}\frac{\partial p}{\partial x}+\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right);\\
&\frac{\partial v}{\partial t}+v\frac{\partial v}{\partial x}+u\frac{\partial v}{\partial y}=-\frac{1}{p_0}\frac{\partial p}{\partial y}+\nu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)+gb_f T;\\
&\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial y}=k\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right);\\
&\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0,
\end{aligned} \tag{10.7}$$

де $p=p(x,y,t), T=T(x,y,t)$. Для складових швидкостей вводиться поняття функції струму:

$$\vec{v}=\{u,v,0\}=\nabla y\,, \tag{10.8}$$

тобто маємо співвідношення

$$u=\frac{\partial y}{\partial y}; v=-\frac{\partial y}{\partial x}. \tag{10.9}$$

У системі (10.7) за умови (10.9) останнє рівняння задовольняється тотожно, а рівняння, що залишилися, набувають вигляду:

$$\begin{aligned}
&\frac{\partial}{\partial t}\frac{\partial y}{\partial y}+\frac{\partial y}{\partial y}\frac{\partial^2 y}{\partial x\partial y}-\frac{\partial y}{\partial x}\frac{\partial^2 y}{\partial y^2}=-\frac{1}{p_0}\frac{\partial p}{\partial x}+\nu\nabla^2\left(\frac{\partial y}{\partial y}\right);\\
&-\frac{\partial}{\partial t}\frac{\partial y}{\partial x}-\frac{\partial y}{\partial y}\frac{\partial^2 y}{\partial x^2}-\frac{\partial y}{\partial y}\frac{\partial^2 y}{\partial x\partial y}=-\frac{1}{p}\frac{\partial p}{\partial y}-\nu\nabla^2\left(\frac{\partial y}{\partial x}\right)+gb_f T;\\
&\frac{\partial T}{\partial t}+\frac{\partial y}{\partial y}\frac{\partial T}{\partial x}-\frac{\partial y}{\partial x}\frac{\partial T}{\partial y}=k\nabla^2 T,
\end{aligned} \tag{10.10}$$

$$\text{де } \nabla^2\equiv\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right) \tag{10.11}$$

Після процедури перехресного диференціювання першого рівняння по y, а другого - по x і наступного віднімання першого із другого одержимо:

$$\begin{aligned}
&\frac{\partial}{\partial t}(\nabla^2 y)+\{y,\nabla^2 y\}=\nu\nabla^2(\nabla^2 y)+gb_f\frac{\partial T}{\partial x};\\
&\frac{\partial T}{\partial t}+\{y,T\}=k\nabla^2 T,
\end{aligned} \tag{10.12}$$

де $\{\Phi\_,y\ \_\} \equiv \left(\frac{\partial \Phi}{\partial x}\frac{\partial y}{\partial y} - \frac{\partial \Phi}{\partial y}\frac{\partial y}{\partial x} \right)$ - позначення дужок Пуассона. (10.13)

Якщо прийняти лінійну залежність від y для рівноважної температури за висотою смуги течії, то збурене значення температури дорівнює:

$$T(x,y,t) = \frac{\Delta T_0 y}{h} + t\,(x,y,t), \tag{10.14}$$

де $t\,(x,y)$-збурений розподіл температур у потоці, а $\Delta T_0 = T_1 - T_2$ - не збурені температури на поверхні потоку.

У підсумку збурені рівняння руху потоку у полі температур (10.14) набувають вигляду:

$$\begin{aligned} &\frac{\partial}{\partial t}(\nabla^2 y\ ) + \{y\ ,\nabla^2 y\ \} = \nu\nabla^2(\nabla^2 y\ ) + gb_f\ \frac{\partial t\,(x,y,t)}{\partial x}; \\ &\frac{\partial t\,(x,y,t)}{\partial t} + \{y\ ,t\,(x,y,t)\} = \frac{\Delta T_0}{h}\frac{\partial y}{\partial x} + k\nabla^2 t\,(x,y,t), \end{aligned} \tag{10.15}$$

при таких граничних умовах:

$$\begin{aligned} &t\,(x,0,t) = t\,(x,h,t) = 0; \\ &\nabla^2 y\ (x,0,t) = \nabla^2 y\ (x,h,t) = 0; \\ &y\ (x,0,t) = y\ (x,h,t) = 0. \end{aligned} \tag{10.16}$$

10.1.4. Маломодове наближення

Отже, відповідно до стандартного методу розв’язання подібних нелінійних початково-крайових задач, представимо функцію струму і збурену температуру у вигляді подвійних рядів Фур'є з коефіцієнтами залежними від часу:

$$\begin{aligned} &y\ (x,y,t) = \sum_{n,m=1}^{\infty} y\ _{n,m}(t) Sin\frac{pmx}{h} Sin\frac{pny}{h}; \\ &t\,(x,y,t) = \sum_{n,m=1}^{\infty} t\ _{n,m}(t) Cos\frac{pmx}{h} Sin\frac{pny}{h}. \end{aligned} \tag{10.17}$$

Підставляючи ці розкладання (10.17) у рівняння (10.15) і прирівнюючи коефіцієнти при однакових тригонометричних функціях, одержимо нескінченну систему ЗДР для шуканих коефіцієнтів $y\ _{n,m}(t)$ і $t\ _{n,m}(t)$.

Але Лоренц запропонував залишити в розкладаннях мінімальну кількість членів розкладання (10.17), які б зберегли істотну нелінійну систему.

Цей мінімальний вибір був обумовлений чисельними експериментами, проведеними Б. Зальцманом на початку 60-х років (див. B. Saltzman Finite amplitude free convection as an initial value problem// Journal of Atmospheric Sciences. – 1962.-vob.19.-p.329 - 341). Було встановлено, що при деяких значеннях параметрів системи

дійсно виникають режими течій, при яких істотними є тільки нижчі моди рухів, а амплітуди інших-прагнуть до нуля. Причому перші характеризуються неперіодичними коливаннями.

Тому згідно з Лоренцом залишимо в (10.17) тільки три істотні моди, амплітуди яких позначимо *X, Y, Z.*

Приведення системи (10.15) до системи ЗДР для амплітуд представимо нижче після процедури відповідного безрозмірювання системи (10.15).

Приймемо згідно з Лоренцом такі параметри для введення безрозмірних величин:

h-висота смуги;

$t = \dfrac{h^2}{p^2(1-a^2)x}$ - час;

h^2/t - потенціал функції струму;

ΔT_0 -температура.

Далі вводимо позначення:

$$b = \frac{4}{1+a^2};$$
$$r = \frac{R}{R_c} = \frac{gb_f h^2 a^2 \Delta T_0}{p^4(1-a^2)^3 kv}. \qquad (10.18)$$

Система (10.15) у безрозмірних параметрах має вигляд

$$\frac{\partial}{\partial t}(\nabla^2 y) + \{y, \nabla^2 y\} = \frac{sb}{4p^2}\nabla^2(\nabla^2 y) + \frac{4sr}{ba^2}\frac{\partial t(x,y,t)}{\partial x};$$
$$\frac{\partial t(x,y,t)}{\partial t} + \{y, t(x,y,t)\} = \frac{\partial y}{\partial x} + \frac{sb}{4p^2}\nabla^2 t(x,y,t). \qquad (10.19)$$

Нарешті розв’язок системи (10.19) представимо у вигляді

$$y(x,y,t) = X(t)\frac{\sqrt{2}}{p^2 a}Sin(pax)Sin(py);$$
$$t(x,y,t) = \frac{1}{pr}(Y(t)\sqrt{2}Cos(pax)Sin(py) - Z(t)Sin(2py)). \qquad (10.20)$$

Надалі система (10.19) і (10.20) її кінцево мірне наближення (10.20) буде основою застосування комп'ютерних методів для системи Лоренца.

Представимо вид векторних полів течій в’язкої рідини в маломодовому наближенні. Для того, щоб візуалізувати стаціонарне поле швидкостей течії, завантажимо вбудований пакет

In[1]:=<< *Graphics*'*PlotField*',

а потім вводимо відповідні коди для запису функції струму і збуреної температури течії

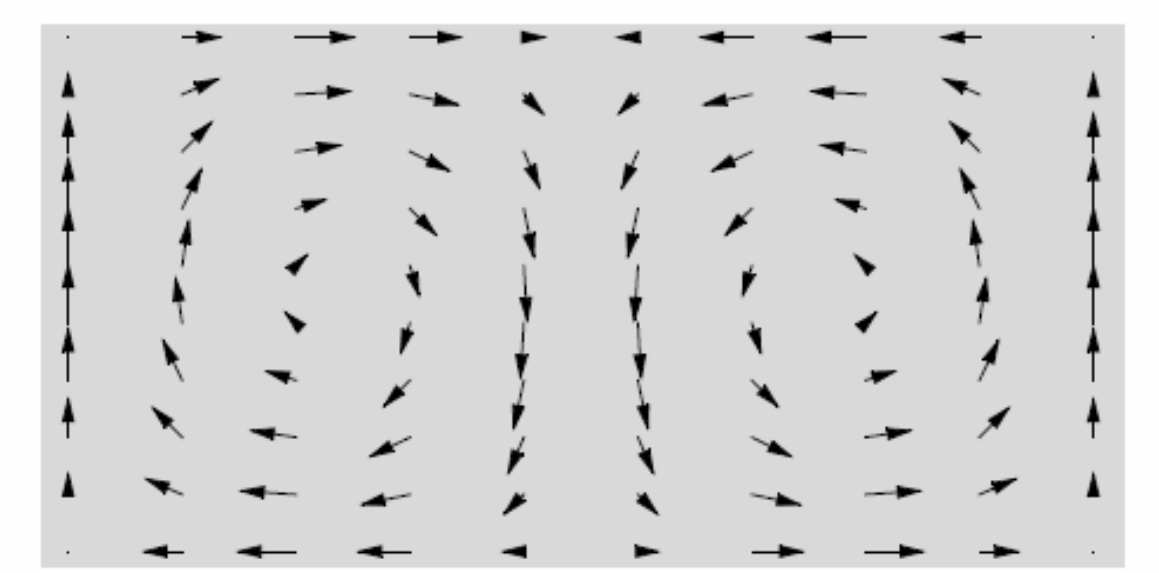
$$In[2] := y\,(x,y,t) = X(t)\frac{\sqrt{2}}{p^2 a}Sin(pax)Sin(py);$$

$$t\,(x,y,t) = \frac{1}{pr}(Y(t)\sqrt{2}Cos(pax)Sin(py) - Z(t)Sin(2py));$$

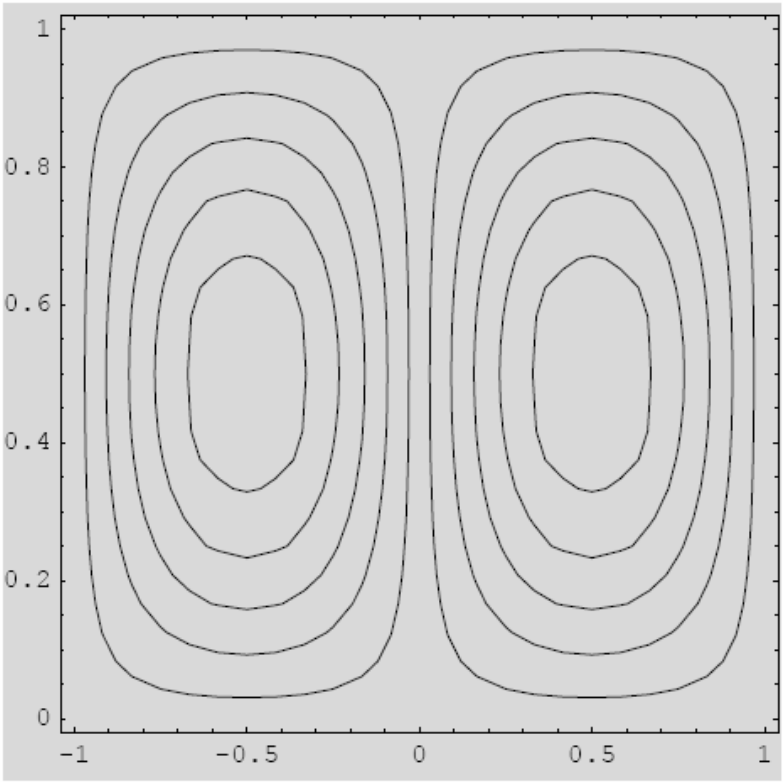
$$In[4] := u = \frac{\partial y\,(x,y,t)}{\partial y}; v = -\frac{\partial y\,(x,y,t)}{\partial x}.$$

Тоді вигляд векторного поля швидкостей течії представлений нижче на двох векторних діаграмах у різних масштабах течії.

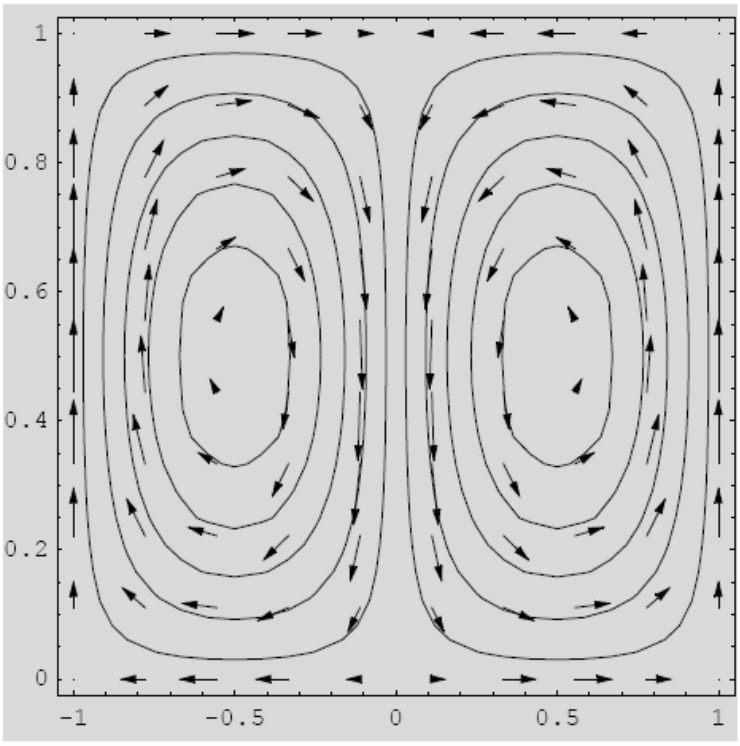
```
In[5]:= PlotVectorField[{u,v}/.{X(t)→.1, a→1},
        {x,-1,1},{y,0,1}, PlotPoint s→10, Background→GrayLevel[.85]];
```



```
In[6]:= ContourPlot[Evaluate[y (x,y,t)/.{X(t)→1, a→1}],{x,-1,1},
        {y,0,1}, ContourShading→False, Background→GrayLevel[.85]];
```



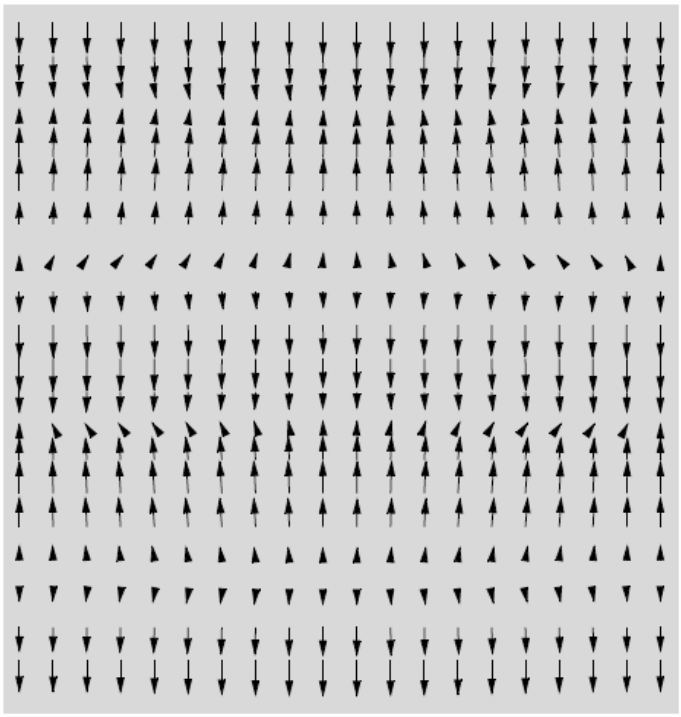
$$In[7] := Show[ContourPlot[Evaluate[y\,(x, y, t) /. \{X(t) \to 1, a \to 1\}], \{x,-1,1\}, \{y,0,1\}, ContourShading \to False], PlotVectorField[\{u, v\} /. \{X(t) \to .1, a \to 1\}, \{x,-1,1\}, \{y,0,1\}, PlotPo\,\text{int}\, s \to 10], Background \to GrayLevel[.85]];$$



Очевидно, що течія структурована відповідно до попередніх припущень про характер руху рідини. Вигляд течії на векторних полях відповідає передбачуваній схемі течії.

Зобразимо вигляд поля градієнтів температури, у вигляді векторного поля, як представлено нижче.

```
In[8] := PlotGradientField[t (x, y, t) /. {Y(t) → .1, a → 1, Z(t) → 1, r → 1},
        {x,–1,1}, {y,–1,1}, PlotPo int s → 20, Background → GrayLevel[.85]];
```



Таким чином, попередні припущення про характер конвективної течії відповідають маломодовому наближенню течії в'язкої рідини. Залишилося тільки визначити тимчасові амплітуди течії, звичайні диференціальні рівняння для яких будуть визначені нижче методами символьної алгебри, а їхня еволюція в часі - шляхом числових розв'язків.

10.1.5. Застосування символьної алгебри для перетворень динамічних рівнянь

Розв'язання кінцевомірної моделі Лоренца виконаємо з використанням диференціальних операторів, які вводимо комп'ютерними кодами.

Спочатку складемо оператори обчислення операторів *дужок Пуассона:*

```
Quit[];
```

$$In[1] := p[\Phi\_, y\_] := \left(\frac{\partial \Phi}{\partial x}\frac{\partial y}{\partial y} - \frac{\partial \Phi}{\partial y}\frac{\partial y}{\partial x} \right)$$

і оператора Лапласа

$$In[2] := l[y\_] := \left(\frac{\partial^2 y}{\partial x^2} + \frac{\partial^2 y}{\partial y^2} \right)$$

Якщо ввести функцію струму і збурене значення температури в загальному вигляді

$$In[3] := y = y(x, y, t);$$

$$\Phi = t(x, y, t) + \frac{\Delta T_0 y}{h},$$

то із попереднього оператора легко одержуємо, наприклад, один із доданків для збуреного значення температури у рівнянні (1.19).

$$In[5] := (p[y, \Phi] - p[y, t(x, y, t)]) // Simplify$$

$$Out[5] = \frac{\Delta T_0 y^{(1,0,0)}(x, y, t)}{h}.$$

Як відомо, диференціальне рівняння в частинних похідних виду

$$In[6] := l[y] = 0$$

$$Out[6] = y^{(0,2,0)}(x, y, t) + y^{(2,0,0)}(x, y, t) = 0$$

називається *рівнянням Лапласа*.

Для наступного висновку рівнянь Лоренца скористаємося введеними раніше операторами.

Як приклад виконання і приклад перетворень функціональних залежностей методами символьної алгебри, покажемо, що останнє рівняння системи (1.7), тобто з механічної точки зору це рівняння нерозривності потоку, тотожно виконується, якщо функція струму течії введена системою (1.9).

Для цього складаємо оператор введення *рівняння нерозривності для нестисливої рідини* у вигляді.

$$In[7] := c[u\_, v\_] := \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0.$$

Його застосування до функції струму $\psi(x,y,t)$

$$In[8] := c[D[y(x, y, t), y], -D[y(x, y, t), x]]$$

$$Out[8] = True$$

приводить до очевидної тотожності.

Крім того, для функції струму справедлива наступна тотожність, яка буде використана нижче:

$$In[9] := ((p[y, l[y]] /. \{D[l[y], x] \to 0, D[l[y], y] \to 0\}) // Simplify) = 0$$

$$Out[9] = True.$$

Очевидно, що символьний оператор Лапласа дозволяє одержати також і символьний вигляд бігармонічного рівняння вигляду

$$In[10] := l[l[y\ ]] = 0$$

$$Out[10] = y^{(0,4,0)}(x,y,t) + 2y^{(2,2,0)}(x,y,t) + y^{(4,0,0)}(x,y,t) = 0,$$

який у цій праці використовується вперше у рівнянні (10.13).

Після того, як складені основні диференціальні оператори системи (10.13), перейдемо до безпосереднього складання моделі Лоренца.

10.1.6. Висновок системи рівнянь Лоренца

Висновок системи нелінійних диференціальних рівнянь Лоренца проведемо методами символьної алгебри з використанням кінцевовимірних наближень розв'язків системи (10.20), запропонованих Лоренцом.

Далі складаємо перше диференціальне рівняння системи використовуючи такі коди:

$$In[11] := egFirst[y\ \_, T\ \_] := \frac{\partial(l[y\ ])}{\partial t} + ((p[y\ , l[y\ ]]/.\{D[l[y\ ], x] \to 0, D[l[y\ ], y] \to 0\})\ //\ Simplify) =$$

$$\frac{sb}{4p^2} l[l[y\ ]] + \frac{4sr}{ba^2}\frac{\partial T}{\partial x};$$

$$y\ = X(t)\frac{\sqrt{2}}{p^2 a} Sin(pax) Sin(py);$$

$$T = \frac{1}{pr}(Y(t)\sqrt{2} Cos(pax) Sin(py) - Z(t) Sin(2py));$$

$$eg1 = \left(egFirst[y\ , T]/.b \to \frac{4}{1+a^2} \right)\ //\ Simplify$$

$$Out[14] = \frac{(a^2+1)\sin(apx)\sin(py)(sX(t) - sY(t) + X'(t))}{a} = 0.$$

Із останнього оператора одержуємо перше рівняння системи Лоренца, що набуває вигляду

$$In[15] := Part[eg1,1,5] = 0$$

$$Out[15] = sX(t) - sY(t) + X'(t) = 0.$$

Далі складаємо розкладання другого рівняння за запропонованою вище схемою.

$$In[16] := egSecond[y\_, T\_] := \frac{\partial T}{\partial t} + (p[y, T]) = \frac{\partial y}{\partial x} + \frac{b}{4p^2} l[T];$$

$$y = X[t]\frac{\sqrt{2}}{p^2 a} Sin(pax) Sin(py);$$

$$T = \frac{1}{pr}(Y(t)\sqrt{2} Cos(pax) Sin(py) - Z(t) Sin(2py));$$

$$(egSecond[y, T]) // Simplify$$

$$Out[19] = \frac{1}{r}(\sin(py)(-\sqrt{2}(a^2+1) b \cos(apx) Y(t) +$$

$$8b\cos(py) Z(t) + 4X(t)(\sqrt{2}\cos(apx)(r + 2\cos(2py) Z(t)) - 2\cos(py) Y(t)) -$$

$$4\sqrt{2}\cos(apx) Y'(t) + 8\cos(py) Z'(t))) = 0$$

Очевидно, що ліва частина отриманої рівності являє собою розкладання амплітудних складових рівнянь Лоренца по функціями *Sin(π y)* і *Sin(2π y)*, який представлений нижче:

$$In[20] := (Part[(egSecond[y, T]) // Simplify, 1])$$

$$Out[20] = \frac{1}{r}(\sin(py)(-\sqrt{2}(a^2+1) b \cos(apx) Y(t) + 8b\cos(py) Z(t) + 4X(t)$$

$$(\sqrt{2}\cos(apx)(r + 2\cos(2py) Z(t)) - 2\cos(py) Y(t)) - 4\sqrt{2}\cos(apx) Y'(t) + 8\cos(py) Z'(t))).$$

Тому послідовно перемножуючи праву частину на *Sin(π y)* і *Sin(2π y)*, виконуючи інтегрування за *y* знаходимо:

- друге рівняння системи Лоренца у вигляді:

$$In[21] := eg2 =$$

$$\left(\left(Integrate[(Part[(egSecond[y, T]) // Simplify, 1]) Sin(py), \{y, 0, 1\}] /. b \to \frac{4}{1+a^2}\right) // Simplify\right) = 0$$

$$Out[21] = \frac{2\sqrt{2}\cos(apx)(rX(t) - Z(t)X(t) - Y(t) - Y'(t))}{r} = 0$$

$$In[22] := Part[eg2, 1, 5] = 0$$

$$Out[22] = rX(t) - Z(t)X(t) - Y(t) - Y'(t) = 0;$$

- третє рівняння шуканої системи:

$$In[23] := eg3 = (Integrate[Part[(egSecond[y, T]) // Simplify, 1] Sin(2py), \{y, 0, 1\}] // Simplify) = 0$$

$$Out[23] = \frac{2(-X(t)Y(t) + bZ(t) + Z'(t))}{r} = 0$$

$$In[24] := Part[eg3, 1] = 0$$

$$Out[24] = \frac{2(-X(t)Y(t) + bZ(t) + Z'(t))}{r} = 0.$$

Таким чином, у результаті символьних обчислень одержуємо систему ЗДР Лоренца для амплітуд. Фізичний зміст змінних $\{X(t), Y(t), Z(t)\}$ можна сформулювати в такий спосіб із (10.20).

Так, змінна $\{X(t)\}$ визначає *швидкість обертання конвекційних валів* течії в'язкої рідини.

Інші фазові змінні $\{Y(t), Z(t)\}$ визначають *розподіл температури* по горизонталі і вертикалі в конвекційній течії в довільно обраній комірці течії в'язкої рідини.

Далі зупинимося на фізичній інтерпретації $\{b, s, r\}$- постійних параметрів у системі Лоренца. Так, параметр $b = \frac{4}{1+a^2}, (\mathrm{a} = a\ /\ b)$ визначається геометрією конвекційної комірки, зокрема відношенням її вертикального і горизонтального розмірів. Параметр σ є просто відношенням коефіцієнта кінематичної в'язкості і коефіцієнта температуропровідності. У динаміці флюїдів це відношення v/k називається *числом Прандля*.

Відношення вигляду $R = \frac{g b_f h^3 \Delta T_0}{vk}$ - є *число Релея*, причому, як було показано раніше Релеєм, умовою виникнення конвекційної течії є критичне число Релея вигляду $R_c = \frac{p^4 (1+a^2)^3}{a^2}$. Очевидно, що параметр r у системі Лоренца являє собою відношення (R / R_c), тим самим визначаючи величину кінцевого градієнта температур, при яких виникає конвекційна течія рідини.

10.1.7. Розв'язок системи Лоренца. Дивний аттрактор

Очевидно, що система Лоренца є трипараметричною. Тому задамо спочатку наступну систему параметрів:

$In[25] := parameters = \{s \to 3, r \to 25, b \to 1\}.$

Тоді вигляд системи Лоренца при введених вище параметрах представлений нижче:

$In[26] := \{Part[eg1,1,5] = 0, Part[eg2,1,5] = 0,$

$Part[eg3,1] = 0, X(0) = 0, Y(0) = 1, Z(0) = 0\} /. parameters$

$Out[26] = \{3X(t) - 3Y(t) + X'(t) = 0, -Z(t)X(t) + 25X(t) - Y(t) - Y'(t) = 0,$

$\frac{2}{25}(-X(t)Y(t) + Z(t) + Z'(t)) = 0, X(0) = 0, Y(0) = 1, Z(0) = 0\}.$

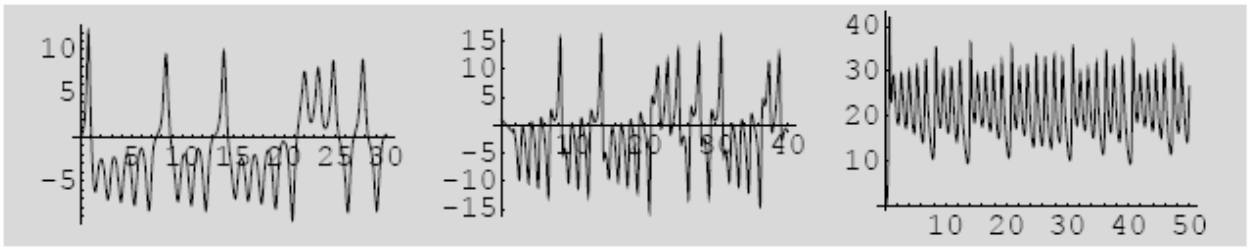
Миттєвий стан системи визначається набором трьох фазових координат $\{X(t), Y(t), Z(t)\}$, які відповідно до теореми існування і одиничності системи ОДУ однозначно визначається початковим станом системи.

Чисельне розв'язання системи ЗДР представлено нижче:

```
In[27]:= Lorentz =
    {X(t), Y(t), Z(t)}/.(NDSolve[{Part[eg1,1,5] = 0, Part[eg2,1,5] = 0, Part[eg3,1] = 0, X(0) = 0,
        Y(0) = 1, Z(0) = 0}/.parameters, {X(t), Y(t), Z(t)}, {t,0,130}] // Flatten.
```

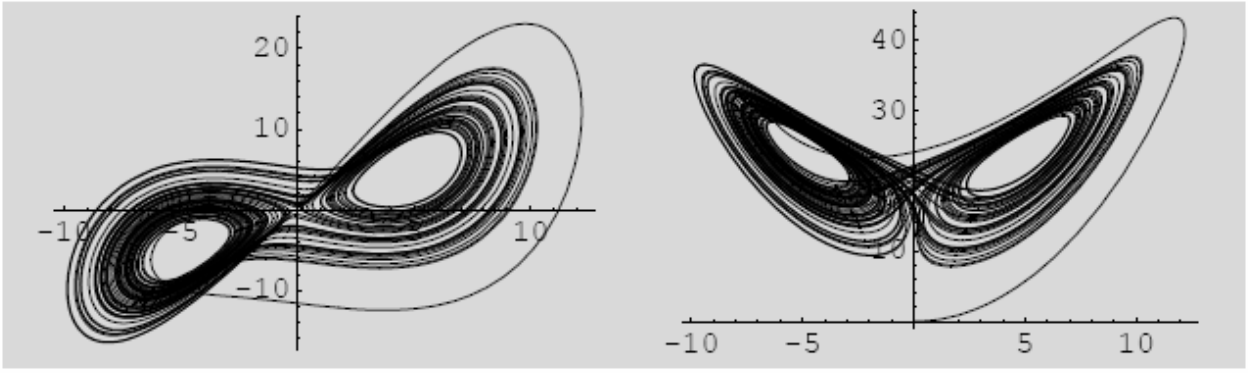
Далі представимо тимчасові реалізації фазових координат.

```
In[28]:= gr1 = Plot[Lorentz[[1]], {t,0,30}, DisplayFunction → Identity];
    gr2 = Plot[Lorentz[[2]], {t,0,40}, DisplayFunction → Identity];
    gr3 = Plot[Lorentz[[3]], {t,0,50}, DisplayFunction → Identity];

In[29]:= Show[GraphicsArray[{gr1, gr2, gr3}],
        DisplayFunction → SDisplayFunction, Background → GrayLevel[.85]];
```



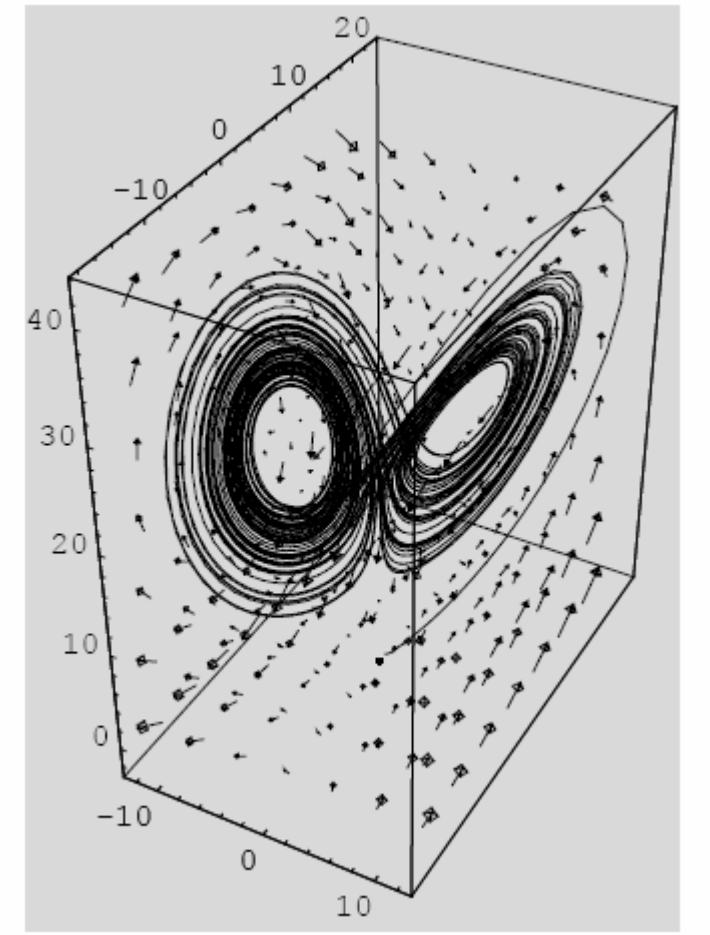
Проекції фазових кривих на площині представлені нижче:

```
In[30]:= gr4 = ParametricPlot[Evaluate[{Lorentz[[1]], Lorentz[[2]]}],
        {t,0,130}, PlotPoints → 5000, DisplayFunction → Identity];

In[31]:= gr5 = ParametricPlot[Evaluate[{Lorentz[[1]], Lorentz[[3]]}],
        {t,0,130}, PlotPoints → 5000, DisplayFunction → Identity];
In[32]:= Show[GraphicsArray[{gr4, gr5}],
        DisplayFunction → SDisplayFunction, Background → GrayLevel[.85]].
```



а сполучена картина фазової кривої і векторного поля на 3D – графіку має вигляд

```
In[33]:=<< Graphics'PlotField3D'

In[34]:= grGeneral = ParametricPlot3D[Evaluate[{Lorentz[[1]], Lorentz[[2]], Lorentz[[3]]}],
          {t,0,130}, PlotPoints → 5000, DisplayFunction → Identity];

In[35]:= grVectors = PlotVectorField3D[{-3(X1 − Y1), −Z1X1 + 25X1 − Y1, X1Y1 − Z1}, {X1, −12, 12},
          {Y1, −15, 15}, {Z1, 0, 40}, VectorHeads → True, DisplayFunction → Identity];
In[36]:= Show[{grGeneral, grVectors}, DisplayFunction → SDisplayFunction,
          Background → GrayLevel[.85]];
```

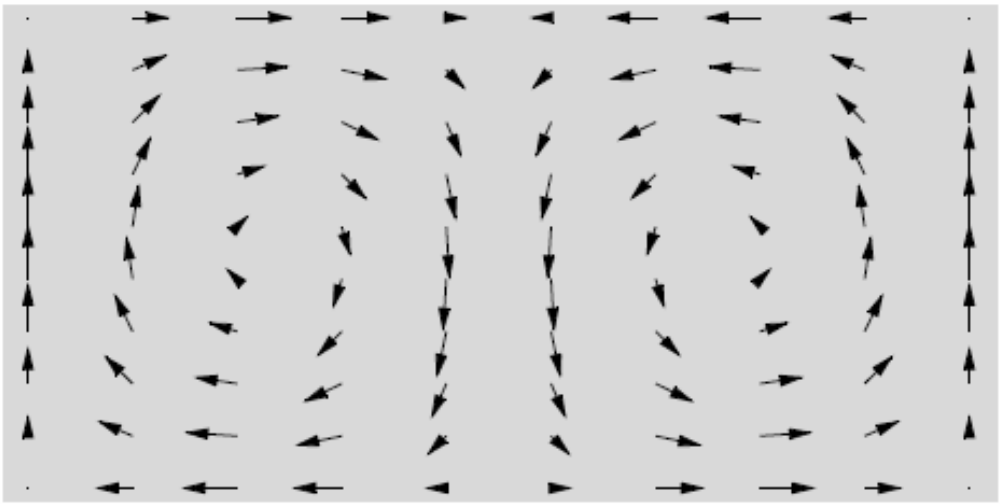


Нестаціонарне поле течії моделюється такими кодами:

```
In[37]:=<< Graphics'PlotField'

In[38]:= y (x, y, t) = X(t) √2/p²a Sin(pax)Sin(py) /. X(t) → Lorentz[[1]];

In[39]:= u = ∂y (x, y, t)/∂y ; v = − ∂y (x, y, t)/∂x ;
```

Виконавши наступний оператор, візуалізація течії має вигляд анімаційного ролика.

```
In[41]:= Table[PlotVectorField[{u, v}/.{t → i, a → 1}, {x, -1, 1},
            {y, 0, 1}, PlotPoints → 10, Background → GrayLevel[.85]], {i, 0, 130, 1}];
```



Отже маємо всі характеристики хаотичного руху в системі Лоренца.

ОСНОВНІ ТЕРМІНИ І ВИЗНАЧЕННЯ

Апроксима́ція (лат. approximare – наближати) – наближене вираження одних математичних об'єктів іншими, простішими, напр. кривих ліній – ламаними, ірраціональних чисел – раціональними, неперервних функцій – многочленами і т. д.

Ди́вний атра́ктор Ло́ренца – атрактор що демонструє хаотичну поведінку і є розв'язком системи трьох нелінійних диференціальних рівнянь, вперше записаних в 1963 році Едвардом Лоренцом при розгляді конвекційного руху в однорідному шарі рідини, що підігрівається знизу. Рівняння Лоренца також описують конвекцію в кільцевій трубці та поведінку одномодового лазера.

Диференціальні рівняння – розділ математики, який вивчає теорію та способи розв'язування рівнянь, що містять шукану функцію та її похідні різних порядків одного аргументу (звичайні диференціальні) чи кількох аргументів (диференціальні рівняння в частинних похідних). Диференціальні рівняння широко використовуються на практиці, зокрема для опису перехідних процесів.

Зада́ча оптиміза́ції – задача знаходження точки (точок) мінімуму, або декількох мінімумів заданої функції.

Крайова задача – задача теорії диференціальних рівнянь, в якій граничні умови задаються в різних точках. Наприклад, при коливаннях струни із закріпленими кінцями зміщення на кожному з кінців дорівнює нулю.

Лінійне рівняння – рівняння, обидві частини якого визначаються лінійними функціями.

Метод найменших квадратів – метод знаходження наближеного розв'язку надлишково-визначеної системи. Часто застосовується в регресійному аналізі.

Метод найшвидшого спуску (градієнтний метод) – ітераційний метод пошуку локального мінімум функції багатьох змінних f(x). Метод полягає в крововому переміщенні в напрямі оберненому до градієнту функції в актуальній точці на відстань пропорційну величині цього градієнта.

Ортогоналізація – алгоритм побудови для даної лінійно незалежної системи векторів евклідового або ермітової простору V ортогональної системи ненульових векторів, що породжують те ж саме підпростір в V.

Перетворення Лапласа – інтегральне перетворення, що зв'язує функцію комплексної змінної (зображення) з функцією дійсної змінної (оригінал). З його допомогою досліджуються властивості динамічних систем і розв'язуються диференціальні і інтегральні рівняння.

Систе́ма (від дав.-гр. σύστημα – «сполучення») – множина взаємопов'язаних елементів, відокремлена від середовища й взаємодіюча з ним, як ціле

Сплайн (англ. spline – планка, рейка) – функція, область визначення якої розбита на куски, на кожному з кусків функція є деяким поліномом (многочленом).

Тео́рія і́гор – теорія математичних моделей прийняття оптимальних рішень в умовах конфлікту. Оскільки сторони, що беруть участь в більшості конфліктів, зацікавлені в тому, щоб приховати від супротивника власні наміри, прийняття рішень в умовах конфлікту, зазвичай, відбувається в умовах невизначеності.

Тео́рія хао́су – підрозділ математики та фізики, який займається дослідженням систем, динаміка яких, за певних умов, значною мірою залежить від початкових умов, що робить довгострокове прогнозування неможливим.

Трансценде́нтна фу́нкція – аналітична функція, що не є алгебраїчною. Простими прикладами трансцендентних функцій є показникова функція, тригонометричні функції, логарифмічна функція.

Хвильове́ рівня́ння – рівняння, яке описує розповсюдження хвиль у просторі.

Mathematica – система комп'ютерної алгебри компанії Wolfram Research . Містить безліч функцій як для аналітичних перетворень, так і для чисельних розрахунків. Крім того, програма підтримує роботу з графікою і звуком, включаючи побудову двох-і тривимірних графіків функцій, малювання довільних геометричних фігур, імпорт і експорт зображень і звуку.

ЛІТЕРАТУРА

1. Stephen Wolfram. The Mathematica. Third Edition Mathematica Vertion 3 – Cambridge University Press, 1996. – 1402 p.
2. Heikki Ruskeepaa. Mathematica for Applying Mathematicians.-Partial manuscript, 1995. – 235 p.
3. Грэхэм Р., Кнут Д., Поташник О. Конкретная математика. Основания информатики/ пер. с англ.; под ред. А.В. Ходулева. – М.: Мир, 1998.
4. Лега Ю.Г., Яценко В.М., Мельник В.В. Моделі і методи прийняття рішень в аналізі та аудиті. – Черкаси: ЧДТУ, 2008. – 147 с.
5. Бережна Л.В., Снитюк О.І. Економіко математичні методи та моделі в фінансах. – К: Кондор, 2009. – 301с.
6. Бережная Е. В., Бережной В. И. Математические методы моделирования экономических систем: Учеб. пособие. – М.: Финансы и статистика, 2001.– 368 с.
7. Вітлінський В.В. Моделювання економіки: Навч. посібник. – К.: КНЕУ, 2003. – 408 с.
8. Горчаков А.А., Орлова И.В. Компьютерные экономико-математические модели: Учеб. пособие для вузов. – М.: Компьютер, ЮНИТИ, 1995. – 136 с.
9. Лук'янова В.В. Комп'ютерний аналіз даних: Посібник. – К.: Видавничий центр «Академія», 2003. – 364 с.
10. Машина Н.І. Математичні методи в економіці: Навч. посібник. – К.: Центр навчальної літератури, 2003. – 148 с.
11. Прокопенко І.Ф., Ганін В.І., Москаленко В.В. Комп'ютеризація економічного аналізу (теорія, практика): Навч. посіб. – К.: Центр навчальної літератури, 2005. – 340 с.
12. Цисарь И.Ф., Нейман В.Г. Компьютерное моделирование экономики. – М.: Диалог - МИФИ, 2002. – 304 с.
13. Экономико-математическое моделирование: Учебник для студентов вузов / Под общ. ред. И.Н. Дрогобыцкого. – М.: Изд-во «Экзамен», 2004. – 800 с.
14. Перевозчикова О.Л. Основи системного аналізу об'єктів і процесів комп'ютеризації. – К.: Видавничий дім "КМ Академія", 2003. – 432 с.
15. Дьяков В. Mathematica 4: учебный курс – СПб: Питер, 2001. – 656 с.
16. Замков О.О., Толстопятенко А.В., Черемных Ю.Н. Математические методы в экономике: Учебник / Под общ. ред. д.э.н., проф. А.В. Сидоровича; МГУ им. М.В. Ломоносова. – 3-е изд., перераб. – М.: Изд-во "Дело и Сервис", 2001. - 368 с.

17. Трояновский В.М. Математическое моделирование в менеджменте: Учебное пособие. – М.: Русская деловая литература, 1999. – 240 с.
18. Шикин Е.В., Чхартишвили А.Т. Математические методы и модели в управлении: Учебное пособие. – М.: Дело, 2000. – 440 с.
19. Варфоломеев В. И. Алгоритмическое моделирование элементов экономических систем. – М.: Финансы и статистика, 2000.
20. Петраков Н.Я., Ротарь В.И. Фактор неопределенности и управление экономическими системами. – М.: Наука, 1995.
21. Сіднев С.П., Шарапов О.Д. Математичні методи підвищення якості управлінських рішень. – К., 1997.
22. Соколов В.Г., Смирнов В.А. Исследование гибкости и надежности экономических систем. – М.: Наука, 1990.
23. Федосеев В.В. Экономико-математические методы и прикладные модели. – М., 1999.
24. Экономико-математические методы и прикладные модели: Учеб. пособие для вузов / Под ред. В.В. Федосеева. – М.: ЮНИТИ, 2000.
25. Кугаенко А.А. Основы теории и практики динамического моделирования социально-экономических объектов и прогнозирования их развития. – М.: Вузовская книга, 1998.
26. Айвазян С.А., Мхитарян В.С. Прикладная статистика и основы эконометрики: Учебник для вузов. – М.: ЮНИТИ, 1998.
27. Секторальні моделі прогнозування економіки України / За ред. В.М. Гейця. – К.: Фенікс, 1999.
28. Боровиков В.П., Ивченко Г.И. Прогнозирование в системе STATISTICA в среде Windows. Основы теории и интенсивная практика на компьютере: Учеб. пособие. – М.: Финансы и статистика, 1999.
29. Дубров А.М. Многомерные статистические методы: Учебник. – М.: Финансы и статистика, 2000.
30. Шелобаев С.И. Математические методы и модели в экономике, финансах, бизнесе: Учебное пособие для вузов. – М.: ЮНИТИ-ДАНА, 2000.
31. Кобелев Н.Б. Практика применения экономико-математических методов и моделей: Учебно-практическое пособие. – М.: ЗАО Финстатинформ, 2000.
32. Петерс Е. Хаос и порядок на рынках капитала. Новый аналитический взгляд на циклы, цены и изменчивость рынка: Пер. с англ. – М.: Мир, 2000.

33. Лук’яненко І. Г., Городніченко Ю. О. Сучасні економетричні методи у фінансах. – К.: Літера ЛТД, 2002.
34. Черняк О.І., Ставицький А.В. Динамічна економетрика: Навч. посібник. – К., 2000.
35. Бокс Дж., Дженкинс Г. Анализ временных рядов. Прогноз и управление. – М.: Мир, 1974.
36. Горчаков А.А., Орлова И.В. Компьютерные экономико-математические модели. – М.: ЮНИТИ, 1995.
37. Парсаданов Г.А. Планирование и прогнозирование социально-экономической системы: Учебное пособие для вузов. – М.: ЮНИТИ-ДАНА, 2001.
38. Колемаев В.А. Математическая экономика. – М.: ЮТИТИ, 2002.
39. Моделирование народнохозяйственных процессов. Под ред. В.С. Дадаяна. – М.: Экономика, 1973.
40. Прогнозування і розробка програм. (Методичні рекомендації) / За ред. В.Ф. Бесєдіна. – К.: Науковий світ, 2000.
41. Канторович Г.Г. Анализ временных рядов. Лекционные и методические материалы // Экономический журнал ВШЭ. – 2002. – № 3.
42. Бушуев С.Д., Лега Ю.Г., Златкин Ал.А., Златкин Ан.А. Анализ методов и моделей формализованного принятия оптимальних и удовлетворительных решений в системах и проектах // Вісник Черкаського державного технологічного університету. – 2010. – № 2. – С. 50–59.
43. Лега Ю.Г., Прокопенко Т.О., Данченко О.Б. Експертні процедури та методи прийняття рішень в інвестиційних проектах // Вісник Черкаського державного технологічного університету. – 2010. – № 2. – С. 69–73.
44. Бушуев С.Д., Лега Ю.Г., Златкин Ал.А. Златкин Ан.А. Формализация и метод решения задачи управления рисками // Вісник Черкаського державного технологічного університету. – 2010. – № 3. – С. 10–19.
45. Лега Ю.Г., Палагин В.В., Лелеко С.А. Обнаружители радиосигналов на фоне ассимметричных негауссовских помех, оптимальные по моментному критерию типа Неймана–Пирсона // Вісник Черкаського державного технологічного університету. – 2009. – № 3. – С. 76–81.