

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

**Ю. В. Триус**  
**І. В. Герасименко**

## **ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ АНАЛІЗУ СИСТЕМ**

**ЛАБОРАТОРНИЙ ПРАКТИКУМ**  
для здобувачів освітнього ступеня бакалавра  
зі спеціальностей  
**122 «Комп'ютерні науки» та 124 «Системний аналіз»**

Черкаси  
2018

УДК 519.8:330.45(075.8)  
ББК 22.18я73  
Т 68

*Затверджено вченою радою ФІТІС,  
протокол № 12 від 26.06.2018 р.,  
згідно з рішенням кафедри комп'ютерних наук  
та інформаційних технологій управління,  
протокол № 11 від 18.06.2018 р.*

Упорядники Триус Ю.В., д.пед.н., к.ф.-м.н., професор  
Герасименко І.В., к.пед.н

Рецензент Тимченко А. А., д.т.н, професор

Т 68 **Інформаційні технології аналізу систем:** лабораторний практикум для здобувачів освітнього ступеня бакалавр зі спеціальності 122 «Комп'ютерні науки» та 124 «Системний аналіз» [Електронний ресурс] / [упоряд. Ю.В. Триус, І. В. Герасименко] ; М-во освіти і науки України, Черкас. держ. технол. ун-т. – Черкаси : ЧДТУ, 2018. – 191 с. – Назва з титульного екрана.

Видання містить теоретичний матеріал і методичні рекомендації до виконання лабораторних робіт із використанням сучасного апаратного і програмного забезпечення. Наведено рисунки, схеми та необхідні пояснення в авторській редакції. Подано порядок виконання та оформлення робіт, зразки та джерела інформації.

Для здобувачів освітнього ступеня бакалавра зі спеціальностей галузі знань «Інформаційні технології» та «Системний аналіз».

Навчальне електронне видання  
комбінованого використання

## ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ АНАЛІЗУ СИСТЕМ ЛАБОРАТОРНИЙ ПРАКТИКУМ

для здобувачів освітнього ступеня бакалавр зі спеціальностей  
122 «Комп'ютерні науки» та 124 «Системний аналіз»

Упорядники Триус Юрій Васильович  
Герасименко Інна Володимирівна

*В авторській редакції.*

## ЗМІСТ

ВСТУП	4
1. ЛАБОРАТОРНА РОБОТА 1 «ІНТЕРФЕЙС, ТИПИ ДАНИХ І МАТЕМАТИЧНІ ФУНКЦІЇ В СКМ МАТНСАД»	5
2. ЛАБОРАТОРНА РОБОТА 2 «ЛІНІЙНА АЛГЕБРА В СКМ МАТНСАД»	23
3. ЛАБОРАТОРНА РОБОТА 3 «ОСНОВИ МАТЕМАТИЧНОГО АНАЛІЗУ В СКМ МАТНСАД»	35
4. ЛАБОРАТОРНА РОБОТА 4 «ГРАФІЧНА ВІЗУАЛІЗАЦІЯ ОБЧИСЛЕНЬ В СКМ МАТНСАД»	43
5. ЛАБОРАТОРНА РОБОТА 5 «РОЗВ'ЯЗУВАННЯ ЗАДАЧ ОПТИМІЗАЦІЇ В СКМ МАТНСАД»	52
6. ЛАБОРАТОРНА РОБОТА 6 «ГЕОМЕТРИЧНИЙ МЕТОД РОЗВ'ЯЗУВАННЯ ЗАДАЧІ ЛІНІЙНОГО ПРОГРАМУВАННЯ»	73
7. ЛАБОРАТОРНА РОБОТА 7 «ПАРАБОЛІЧНЕ ІНТЕРПОЛЮВАННЯ. ІНТЕРПОЛЯЦІЙНА ФОРМУЛА ЛАНГРАНЖА»	79
8. ЛАБОРАТОРНА РОБОТА 8 «НАБЛИЖЕННЯ ФУНКЦІЙ МЕТОДОМ НАЙМЕНШИХ КВАДРАТІВ ЗА ДОПОМОГОЮ СКМ»	88
9. ЛАБОРАТОРНА РОБОТА 9 «ОСНОВИ ПРОГРАМУВАННЯ В СКМ МАТНСАД»	96
10. ЛАБОРАТОРНА РОБОТА 10 «ІНТЕРФЕЙС, ТИПИ ДАНИХ І МАТЕМАТИЧНІ ФУНКЦІЇ В SMATH STUDIO»	105
11. ЛАБОРАТОРНА РОБОТА 11 «ОСНОВИ РОБОТИ З СКМ МАХІМА»	121
12. ЛАБОРАТОРНА РОБОТА 12 «ЗНАЙОМСТВО З МАТЛАВ»	138
13. ЛАБОРАТОРНА РОБОТА 13 «РІШЕННЯ РІВНЯНЬ В МАТЛАВ»	153
14. ЛАБОРАТОРНА РОБОТА 14 «ЧИСЕЛЬНЕ ІНТЕГРУВАННЯ В МАТЛАВ»	174
ОСНОВНІ ТЕРМІНИ І ВИЗНАЧЕННЯ	185
ЛІТЕРАТУРА	189
ДОДАТКИ	190

## ВСТУП

### **Мета навчання дисципліни «Інформаційні технології аналізу систем»**

– ознайомлення студентів з теорією і практикою аналізу систем, що є необхідним для розробки та проектування інформаційних систем різного рівня складності, та оволодіння інформаційними технологіями щодо їх дослідження, зокрема універсальними математичними пакетами, котрі являють собою сукупність методів і засобів, що забезпечують максимально комфортну і швидку підготовку математичних моделей досить складних реальних задач, процесів і явищ, а також алгоритмів і програм, які реалізують аналітичні, графічні і чисельні методи їх розв'язання.

### **Завдання дисципліни:**

- подати у систематизованій формі теоретичні відомості про системний підхід до дослідження систем, математичні методи та інформаційні технології їх аналізу;
- підвищити рівень математичної і професійної підготовки майбутніх фахівців у галузі інформатики та ІКТ;
- поглибити знання з питань, які стосуються математичного моделювання і комп'ютерного експерименту, дослідження ефективності математичних методів розв'язування складних задач за допомогою комп'ютерної техніки, аналізу та інтерпретації отриманих результатів;
- підвищити математичну та інформаційну культуру студентів, шляхом використання інформаційних технологій при вивченні і розв'язуванні задач дослідження систем;
- сформувати у студентів систематизовані знання про класифікацію і структуру систем комп'ютерної математики, можливості і сфери їх використання, а також на прикладі одного або кількох математичних пакетів, сформувати вміння і навички щодо застосування систем комп'ютерної математики для розв'язування різних класів математичних і прикладних задач.

## Лабораторна робота № 1

**Тема:** «Інтерфейс, типи даних і математичні функції в СКМ Mathcad»

Питання:

1. Основні елементи інтерфейсу СКМ Mathcad: головне вікно, головне меню, робота з файлами, панелі інструментів.
2. Довідкова система в СКМ.
3. Засоби набору і редагування математичних документів.
4. Числа і числові константи.
5. Рядкові дані.
6. Складні типи даних: вектори, матриці, масиви, аранжовані змінні.
7. Оператори, функції і вирази. Алгебраїчні й арифметичні функції, тригонометричні і гіперболічні функції. Функції користувача.
8. Початкові відомості про побудову графіків функції в пакеті MathCad.
9. Системи комп'ютерної математики в Internet.

### I. Теоретичний матеріал

#### 1. Основні елементи інтерфейсу СКМ Mathcad

На рис.1.1 показано головне вікно пакету Mathcad Professional 2000, його головне меню і основні режими роботи з файлами.

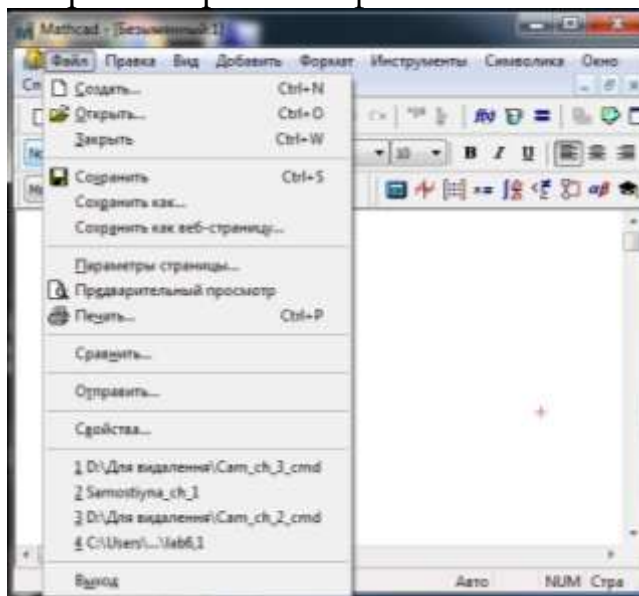


Рисунок 1.1 – Робота з файлами

На рис. 1.2 показано меню панелі інструментів Math пакету Mathcad для введення математичних виразів, математичних моделей, їх розв'язування в чисельному та аналітичному вигляді, а також засоби створення математичних документів.

На рис. 3 подано меню, з допомогою якого в Mathcad-документ можна вставляти різноманітні об'єкти:

- графіки (Graph) (рис. 1.3);
- матриці (Matrix) (рис. 1.4);



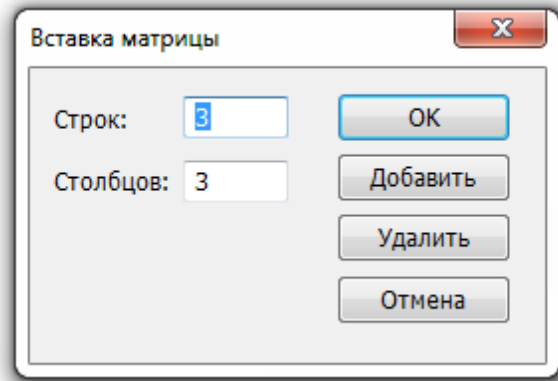


Рисунок 1.4 – Режим створення матриці (Matrix)

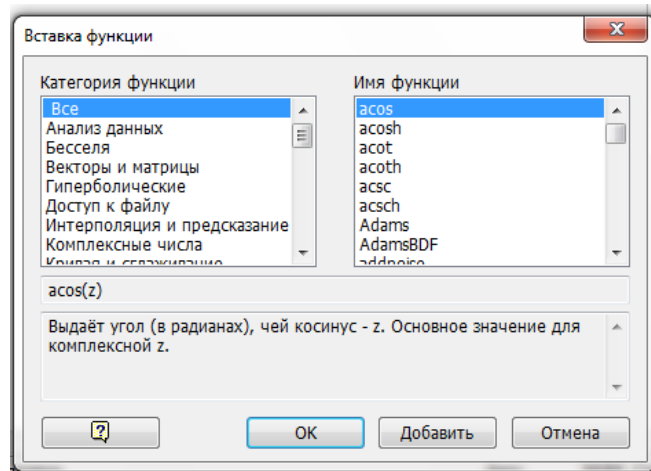


Рисунок 1.5 – Режим створення функцій (Function)



Рисунок 1.6 – Режим выбора единиц измерения различных величин (Unit)

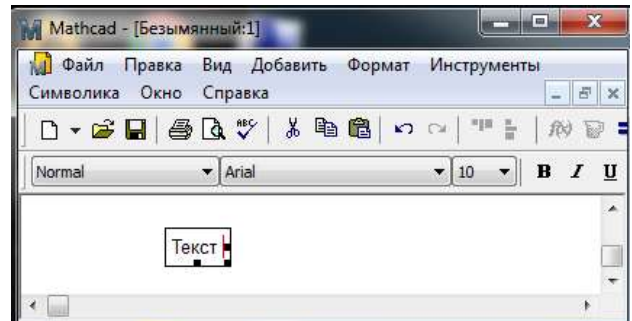


Рисунок 1.7 – Режим введения текста (Text Region)

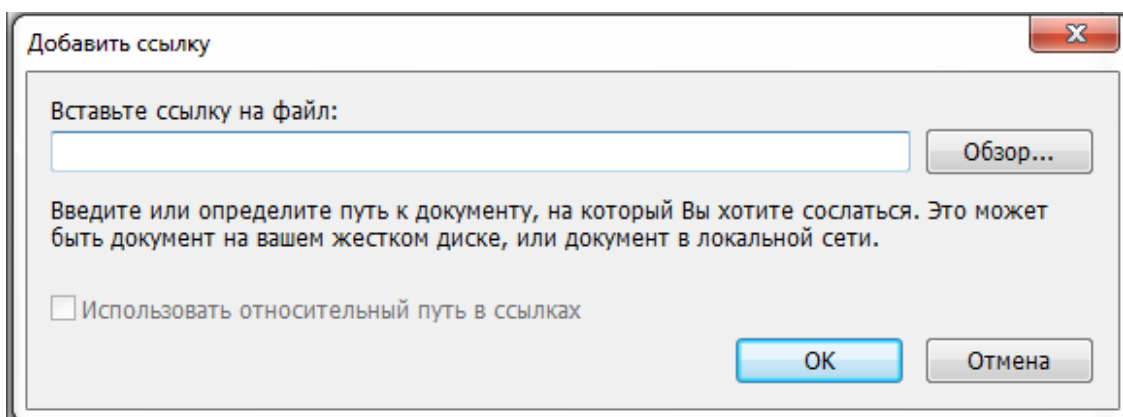


Рисунок 1.8 – Режим введения послания на документ чи файл

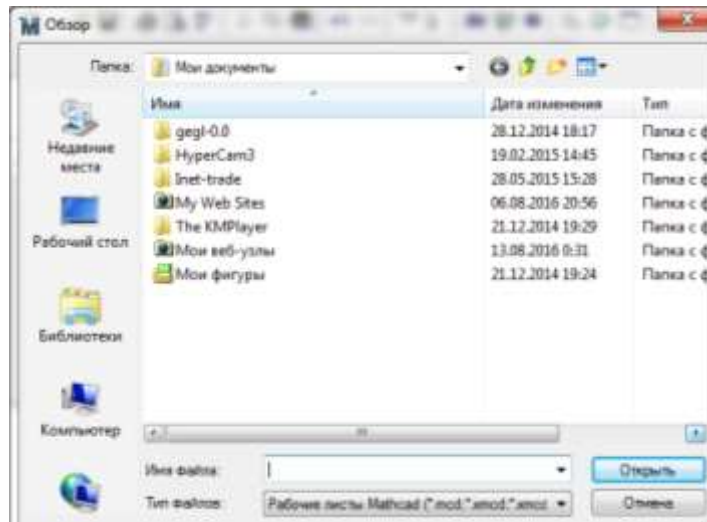


Рисунок 1.9 – Вікно перегляду файлів Mathcad

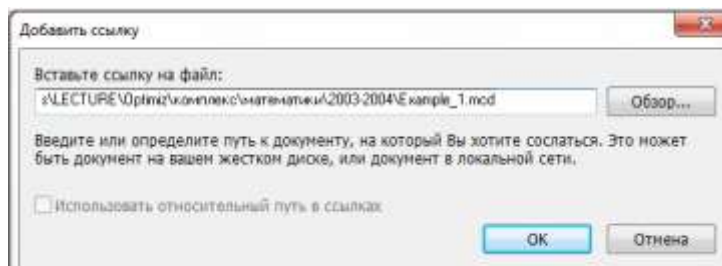


Рисунок 1.10 – Введення шляфу до файла

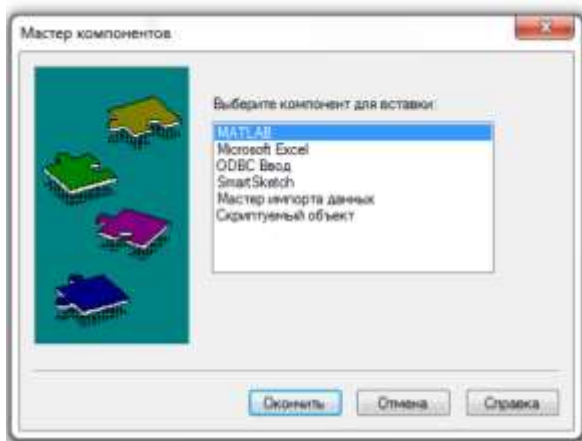


Рисунок 1.11 – Інтегрування в Mathcad інших компонентів (Component)

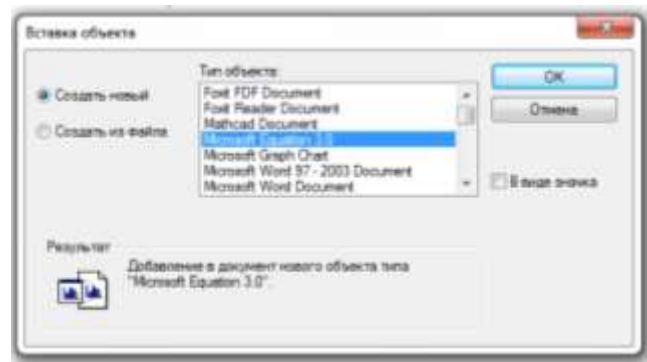


Рисунок 1.12 – Додавання формул у Mathcad-документ

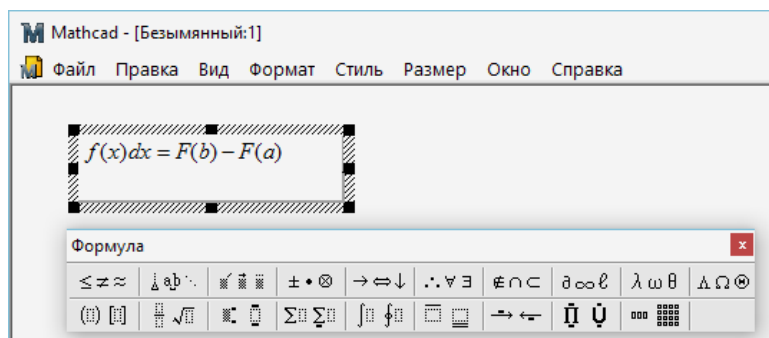


Рисунок 1.13 – Робота з редактором формул



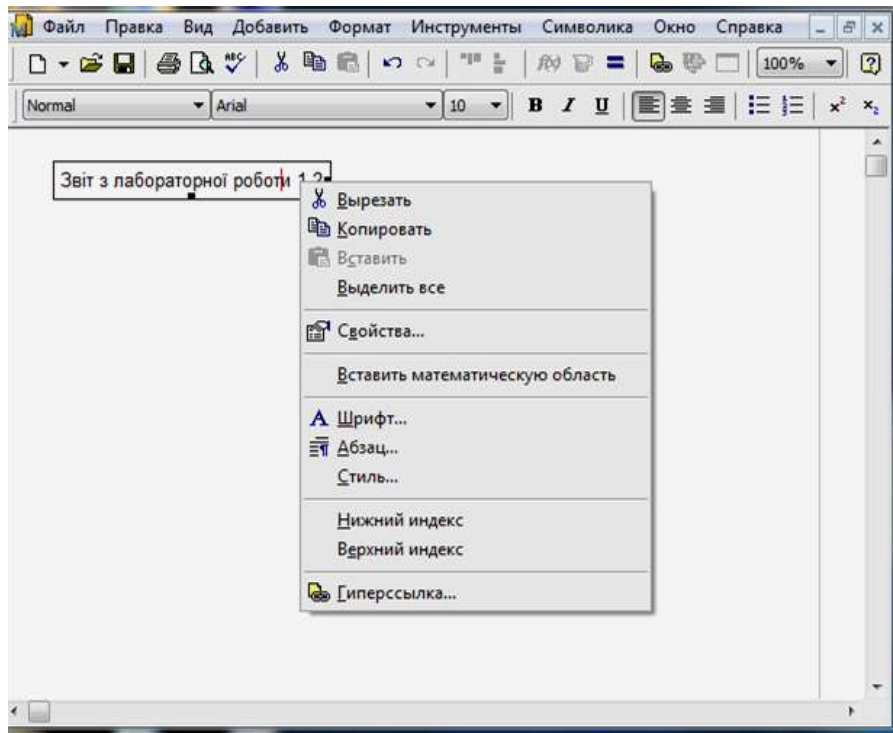


Рисунок 1.14 – Приклад введення текстової області

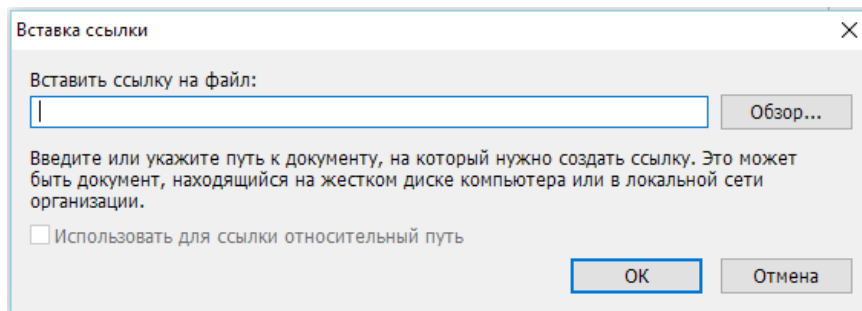


Рисунок 1.15 – Вікно введенню URL-посилання

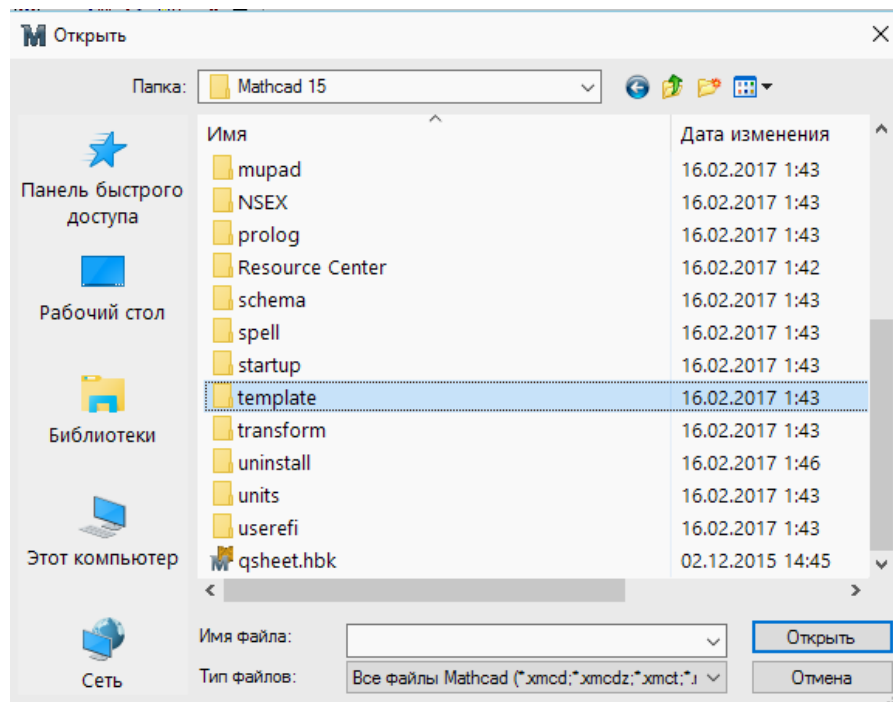


Рисунок 1.16 – Вікно перегляду файлів Mathcad

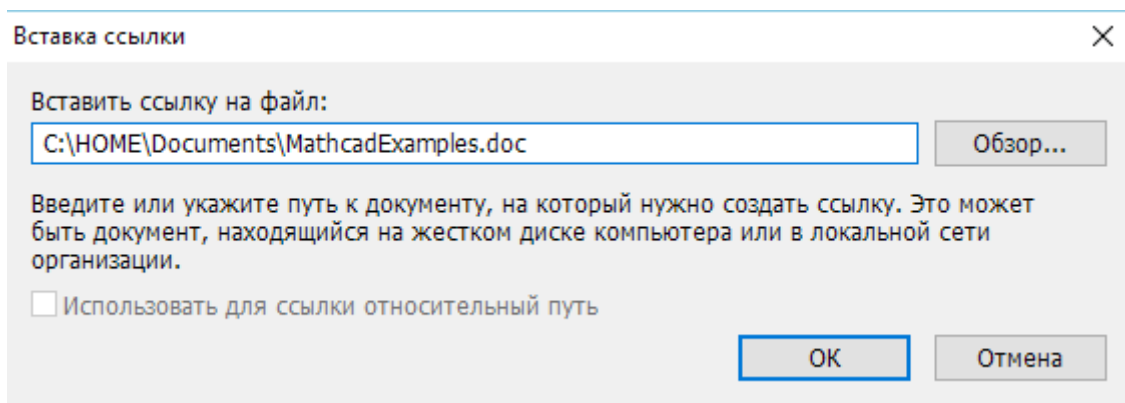


Рисунок 1.17 – Приклад введення гіперпосилання

Головне вікно довідки пакету Mathcad 2000 має вигляд, поданий на рис. 1.18, де також подана інформація про розв'язування рівнянь від однієї змінної за допомогою команди root.

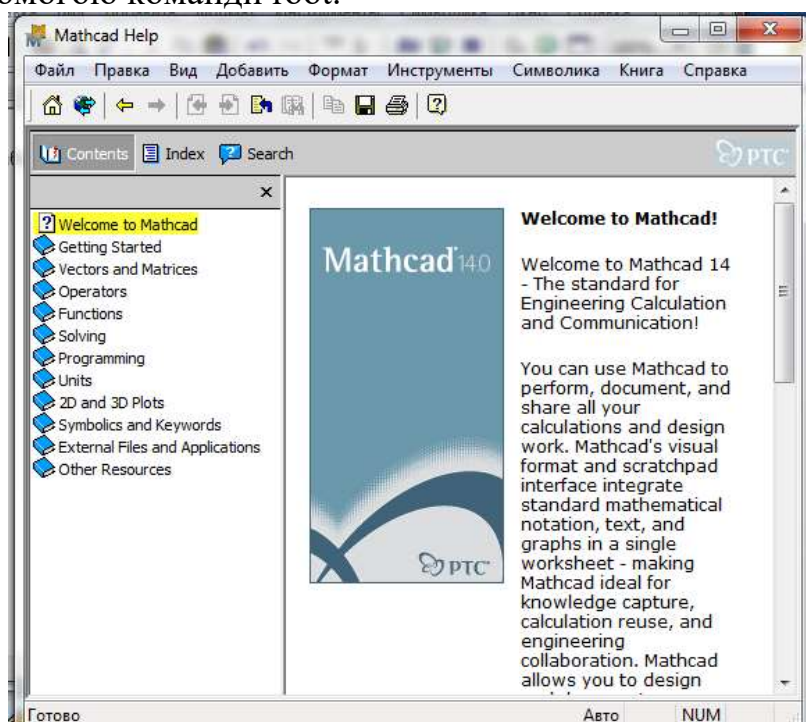


Рисунок 1.18 – Головне вікно довідки Mathcad

## 4. Числа і числові константи

Числа в ролі вхідних даних, констант і результатів обчислень є, мабуть, найбільш розповсюдженими об'єктами математичних систем. Математичні системи можуть працювати з різними типами чисел: натуральними, цілими, раціональними, дійсними, комплексними. Також математичні системи можуть працювати з числами в різних системах числення.

### 4.1. Системи числення

За замовченням використовуються десяткові числа (decimal (d)). Також найбільш часто використовуються числа з основою 2 (двійкові або бінарні числа – binary (b)), з основою 8 (вісімкові числа – octal (o)) і з основою 16 (шістнадцяткові – hexadecimal (h)):

$$100101b = 37$$

$$475o = 317$$

$$0A3CFh = 4.194 \times 10^4.$$

В пакеті Mathcad можна встановлювати поточний формат виведення чисел. Наприклад, щоб встановити потрібну основу системи числення необхідно виконати таку послідовність дій в режимі меню Format: Result -> Display Options (рис. 1.19).

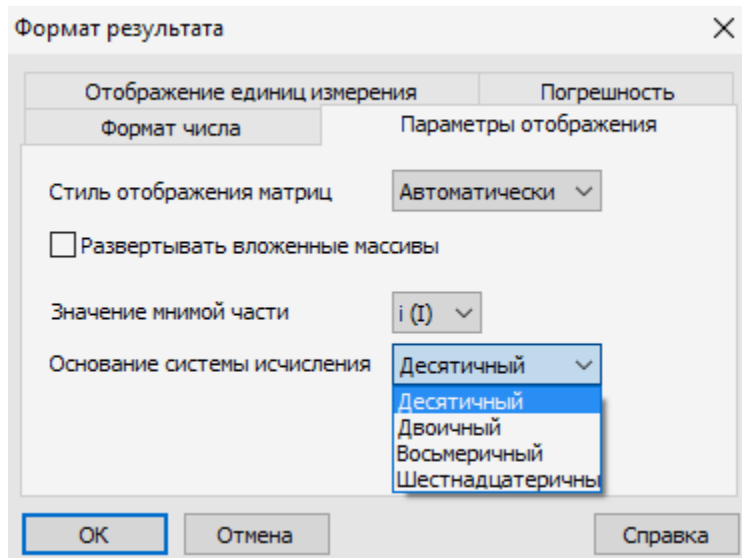


Рисунок 1.19 – Вікно встановлення поточного формату виведення чисел

Крім того, можна встановити виведення числа у формі: Scientific – науковий (експоненціальна форма) та Engineering – інженерний (експоненціальна з порядком кратним 3).

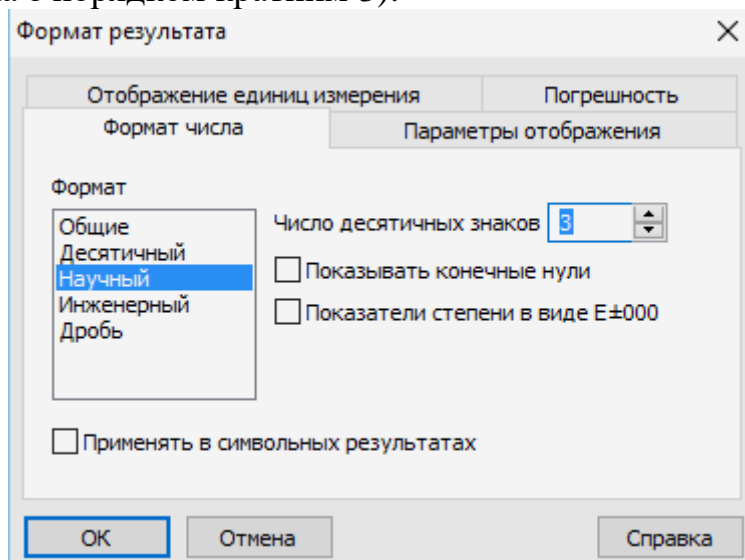


Рисунок 1.20 – Вікно встановлення експоненціальної форми числа

В пакеті Mathcad є можливість працювати з комплексними числами як в алгебраїчній формі:  $Z = \text{Re}(Z) + i\text{Im}(Z)$ , так і експоненціальній:  $Z = Me^{i\varphi}$ . Можна, зокрема, встановити границю, до якої може зменшуватися неявна частина числа при його поданні у комплексній формі, за допомогою опції Tolerance:

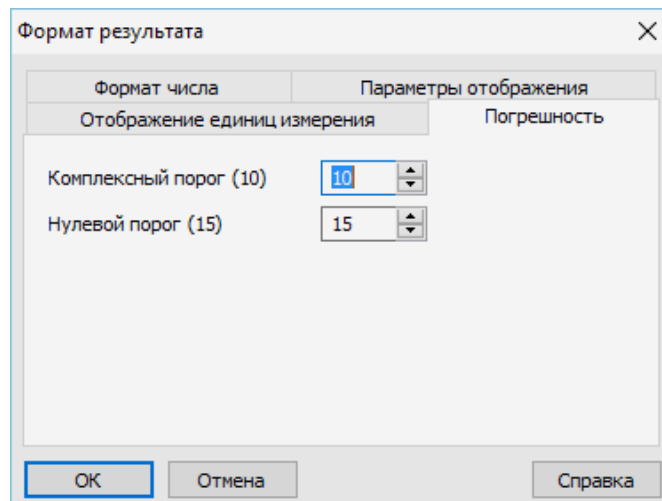


Рисунок 1.21 – Налаштування вікна для роботи з комплексними числами

В математичних пакетах використовуються деякі відомі константи, які вбудовані в ядро системи у ролі глобальних або системних змінних. В пакеті Mathcad системні змінні мають статус об'єктів і вводяться спеціальними комбінаціями клавіш, наприклад,  $\pi \approx 3.1415\dots$  – Ctrl+Shift+P,  $e \approx 2.71828\dots$  – «e», або за допомогою відповідної палітри: Math ->Options:

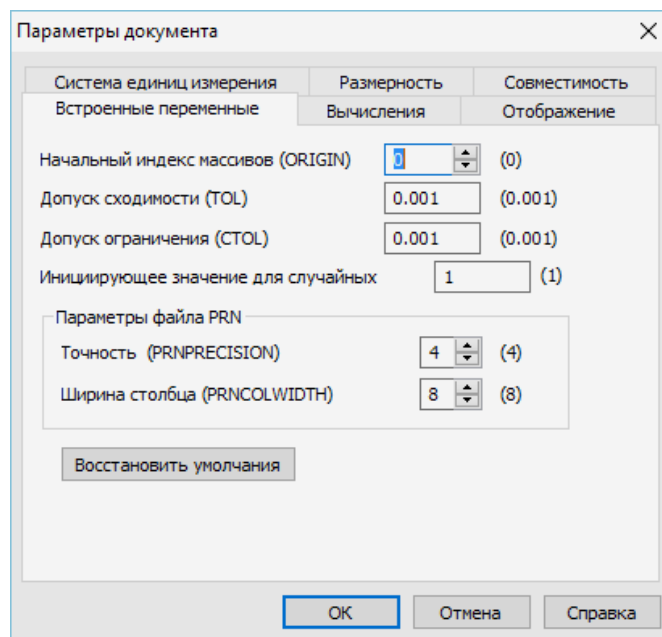


Рисунок 1.22 – Вікно налаштування констант

## 5. Рядкові дані

Рядкові дані (string) в пакеті Mathcad задаються в текстовому полі: Insert -> Text Region.

## 6. Складні типи даних: вектори, матриці, масиви, таблиці, списки

Складними типами даних є такі, що містять у собі інші об'єкти, зокрема числові типи даних, які були розглянуті вище. До них відносяться: вектори, матриці, масиви, таблиці, списки.

Розглянемо як вони описуються і формуються в пакеті Mathcad.

Вектори бувають двох видів: *вектор-рядки* і *вектор-стовпчики*. Вони є частинним випадком матриць, які формуються за допомогою меню Matrix (рис. 1.23).

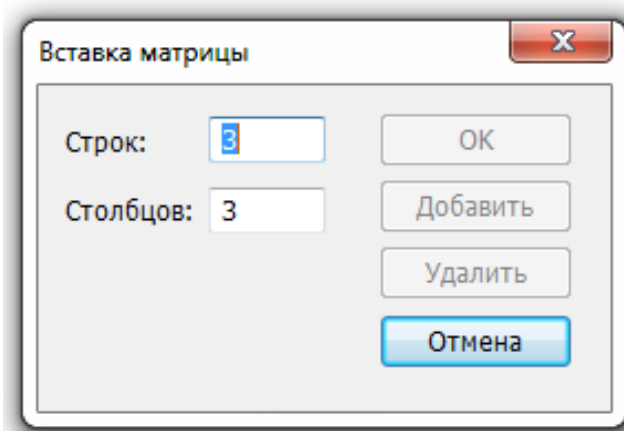


Рисунок 1.23 – Вікно задання матриці

Вектори і матриці можуть використовуватися як самостійні об'єкти або як змінні величини:

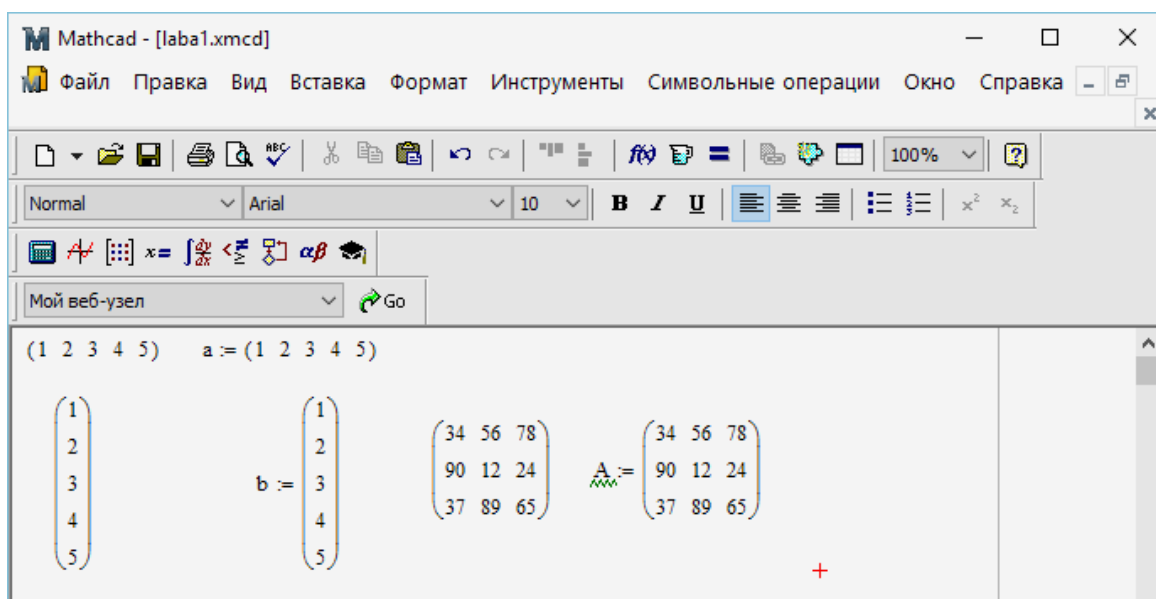


Рисунок 1.24 – Приклади введення матриць

**Масиви** в пакеті Mathcad можна подавати або у вигляді матриць або у вигляді електронних таблиць. Для подання масиву у вигляді електронної таблиці треба скористатися меню Insert -> Component -> Input Table:

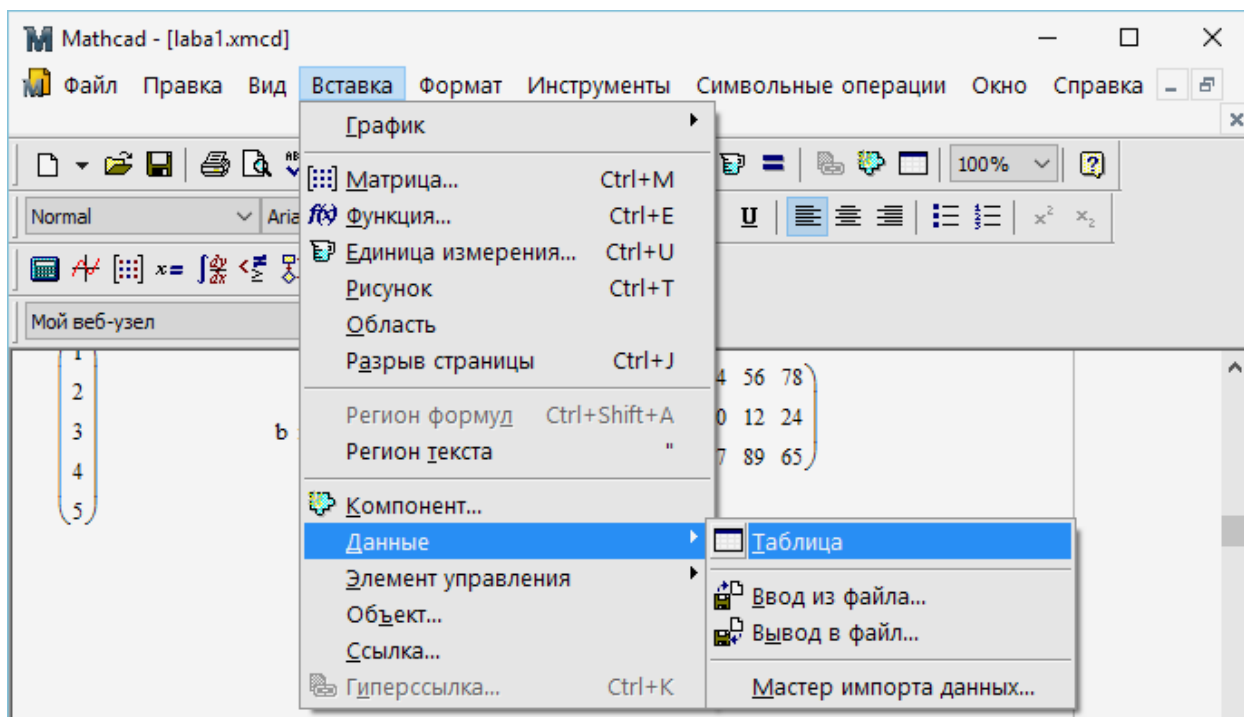


Рисунок 1.25 – Вікно налаштування масиву у вигляді електронної таблиці

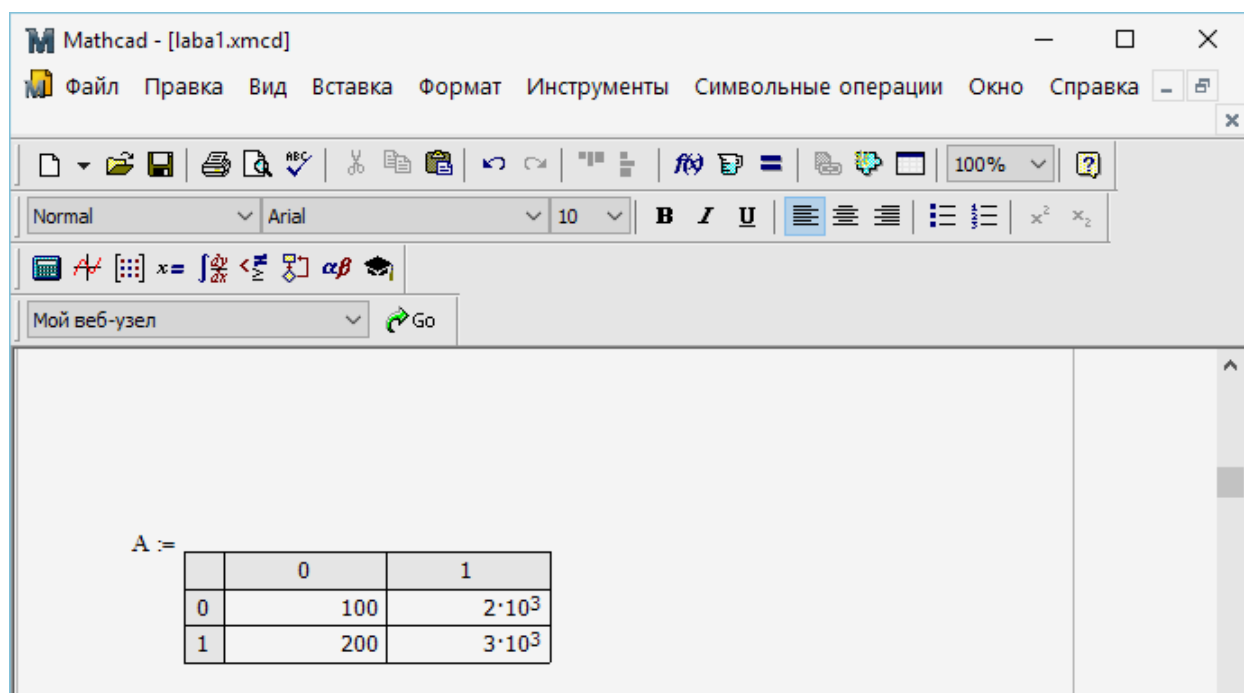


Рисунок 1.26 – Приклад введеного масиву у вигляді таблиці

Досить корисними при різноманітних обчисленнях є **ранжовані змінні** – особливий клас змінних, які мають ряд фіксованих значень, які змінюються з деяким кроком  $h$  від початкового значення  $N_{begin}$  до кінцевого значення  $N_{end}$ . Для введення таких змінних використовується палітра Matrix і шаблон  $m..n$  (рис. 1.27).

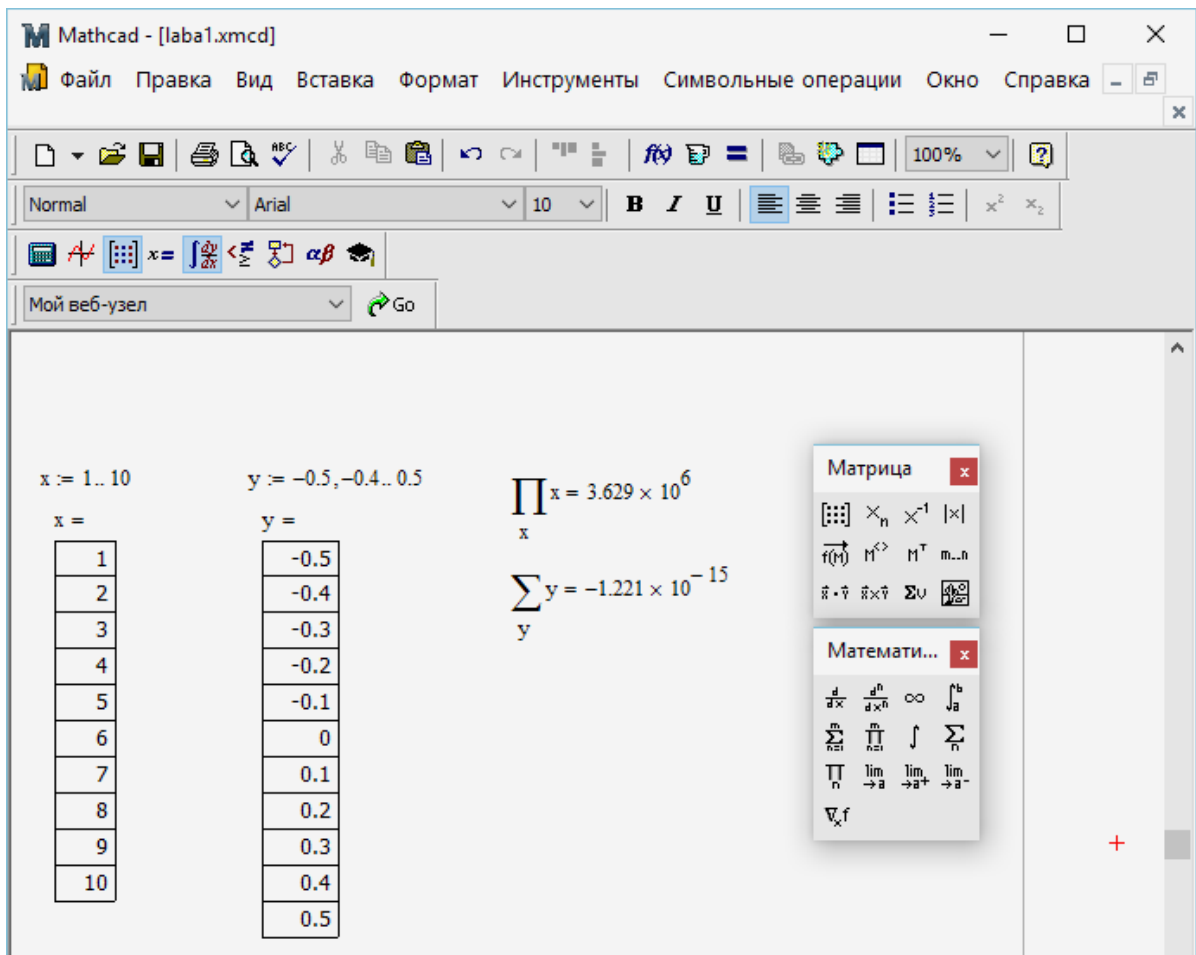


Рисунок 1.27 – Приклад введення ранжованих змінних

## 7. Оператори, функції і вирази

Оператори, функції і вирази з ними складають основу обчислювальних засобів будь-якої СКМ.

### 7.1. Оператори та їх пріоритет

Оператори вводяться спеціальними знаками, які використовуються в математичних виразах для зазначення того, які види дій повинні виконуватися. Найбільш розповсюдженими і однаковими для всіх СКМ є арифметичні оператори: «+» – додавання, «-» – віднімання, «\*» – множення, «/» – ділення, «^» – піднесення до степеня. До операторів відносять також різні дужки: «(, )», «[, ]», «{ }», розділююча крапка, кома, «:», «;». Оператори здійснюють деякі дії над об'єктами, які називають *операндами*.

Нажаль позначення інших операторів в СКМ сильно відрізняються. Ближче за всіх до природної математичної мови є системи класу Mathcad. В них оператори представлені звичними символами, які можна ввести або за допомогою клавіатури, використовуючи фіксовані комбінації клавіш, або за допомогою математичних палітр, активізація яких призводить до появи на екрані відповідних математичних символів і шаблонів для введення операндів у вигляді чорних прямокутників. Це забезпечує візуалізацію обчислень вже на етапі введення вхідної інформації (рис. 1.28).

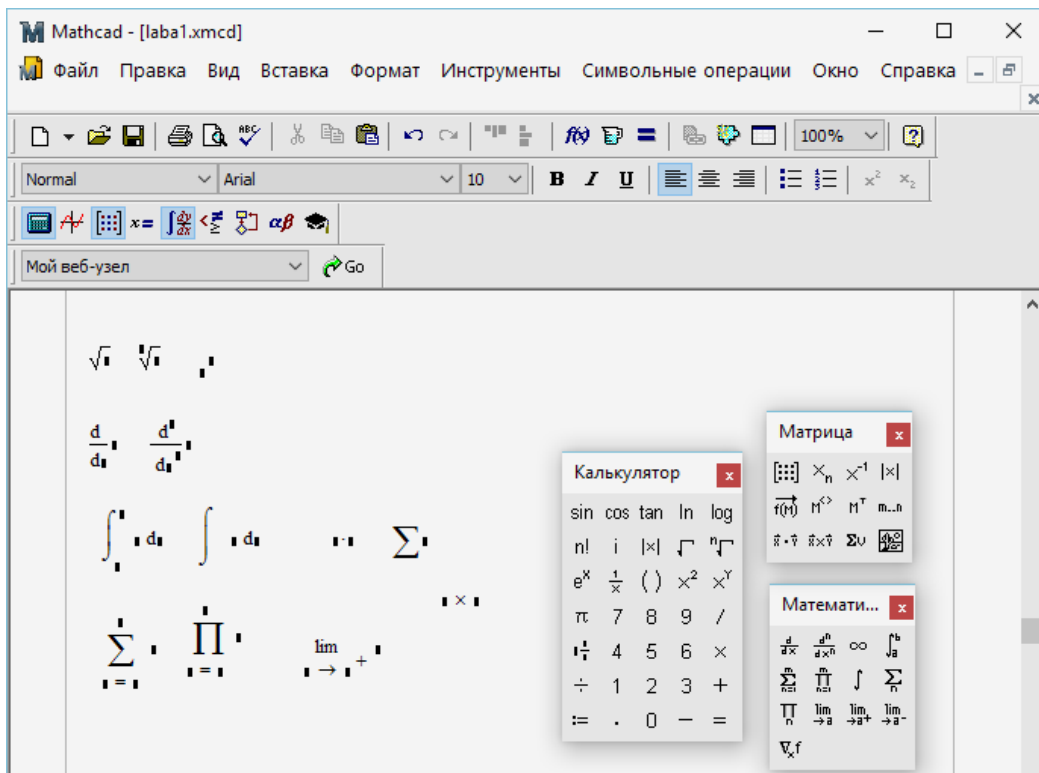


Рисунок 1.28 – Приклад введення символів з математичної палітри

В математичних виразах оператори мають загальноприйнятий пріоритет, тобто порядок виконання операторів у складних виразах.

## 7.2. Функції та їх класифікація

**Функція** – математичний об’єкт, який має унікальне ім’я (ідентифікатор) і виконує певні перетворення своїх вхідних даних, що являють собою список вхідних параметрів.

Функції в СКМ, як правило, розділяються на такі типи:

- *внутрішні функції*, які входять до ядра системи;
- *функції користувача*, тобто функції, які створюються самим користувачем для власних потреб;
- *бібліотечні функції*, які викликаються з пакетів або бібліотек розширення;
- *функції*, що задаються у вигляді *програмного модуля*.

На рис. 1.29 наведено приклади внутрішніх функцій, функції користувача і складної функції, яка задана за допомогою програмного модуля.

До внутрішніх функцій відносяться, наприклад, елементарні математичні функції, серед яких виділяють:

- степеневі;
- показникові і логарифмічні;
- многочлени (поліноми);
- раціональні функції;
- тригонометричні і обернені тригонометричні функції;
- гіперболічні і обернені гіперболічні функції.



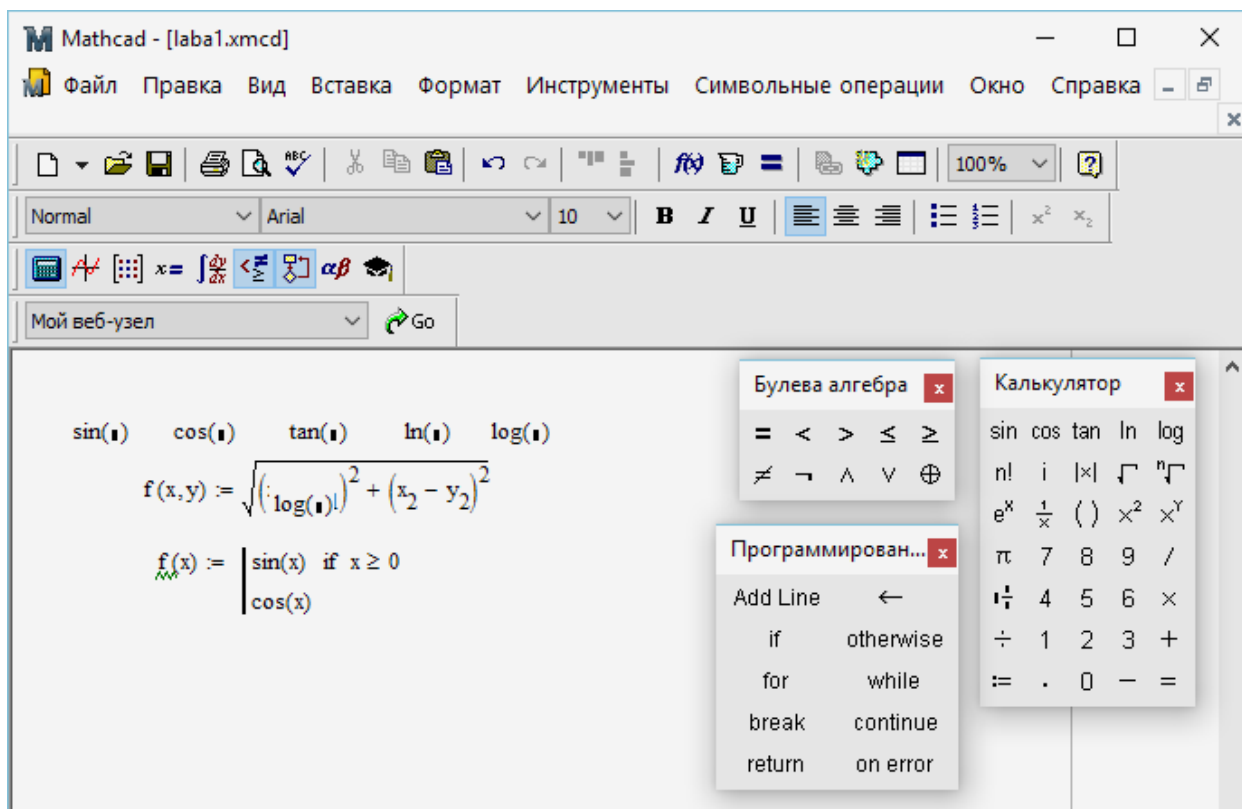


Рисунок 1.29. наведено приклади внутрішніх функцій, функції користувача і складної функції

**7.3. Математичні вирази** – це складні (комбіновані) об’єкти, які складаються з операторів, операндів і функцій з їх параметрами. Правила запису математичних виразів у середовищі Mathcad див. у таблиці 1.1. Також для запису математичних виразів можна використовувати панелі інструментів **Calculator** – для створення основних математичних виразів, **Matrix** – для запису матричних виразів, **Calculus** – для записів складних сум та інтегралів, **Boolean** – для запису логічних виразів (використовується при заданні обмежень), **Evaluation** – деякі додаткові можливості, **Greek** – для запису літер грецького алфавіту, які відкриваються за допомогою панелі **Math** середовища.

Таблиця 1.1 – Правила запису математичних виразів у середовищі Mathcad

Вираз	Комбінація клавіш
Додавання (+)	+
Віднімання (-)	-
Множення (·)	*
Ділення або риска дроби (/)	/
Присвоєння значення змінній (:=)	:
Отримання значення виразу (=)	=
Піднесення до степеня ( $x^y$ )	^
Корінь квадратний ( $\sqrt{x}$ )	\

Корінь n-го степеня ( $\sqrt[n]{x}$ )	Ctrl+\
Діапазон змінної ( $i = 1..10$ )	;
Факторіал (n!)	!
Нижній індекс ( $x_n$ )	[
Задання матриці або вектора	Ctrl+m
Детермінант ( M )	
Стовпчик матриці ( $M^\diamond$ )	Ctrl+6
Транспонування матриці ( $M^T$ )	Ctrl+1
Добуток векторів або матриць	*
Векторний добуток векторів	Ctrl+8
Рівність двох виразів (=)	Ctrl+=
Знак (<)	<
Знак (>)	>
Знак ( $\leq$ )	Ctrl+9
Знак ( $\geq$ )	Ctrl+0
Знак ( $\neq$ )	Ctrl+3
Знак суми ( $\sum_{i=1}^n x_i$ )	Ctrl+Shift+4
$\pi$	Ctrl+Shift+p
Перехід по формулі вище на один рівень	Space (пропуск)

## 8. Системи комп'ютерної математики в Internet

Нижче наведено дані про найбільш відомі системи комп'ютерної математики:

- Matlab;
- Maple;
- MuPad;
- Macsyma;
- SAGE;
- Octave.
- Mathematica;
- Derive;
- Mathcad;
- GAUSS;
- Scilab;

## II. Завдання до лабораторної роботи

1. Розглянути теоретичний матеріал до роботи.
2. Ознайомитися з основними елементами інтерфейсу СКМ Mathcad: головне вікно, головне меню, робота з файлами, панелі інструментів.
3. За допомогою панелі інструментів пакету Mathcad набрати математичні вирази, подані на рис. 1.30.
4. За допомогою довідки пакету Mathcad знайти розділи, які присвячені розв'язуванню різних видів рівнянь та систем рівнянь. З'ясувати, які функції використовуються для розв'язування цих класів задач.
5. Підготувати коротку довідку про засоби розв'язуванню різних видів рівнянь та систем рівнянь в пакетів Mathcad і включити її до звіту.

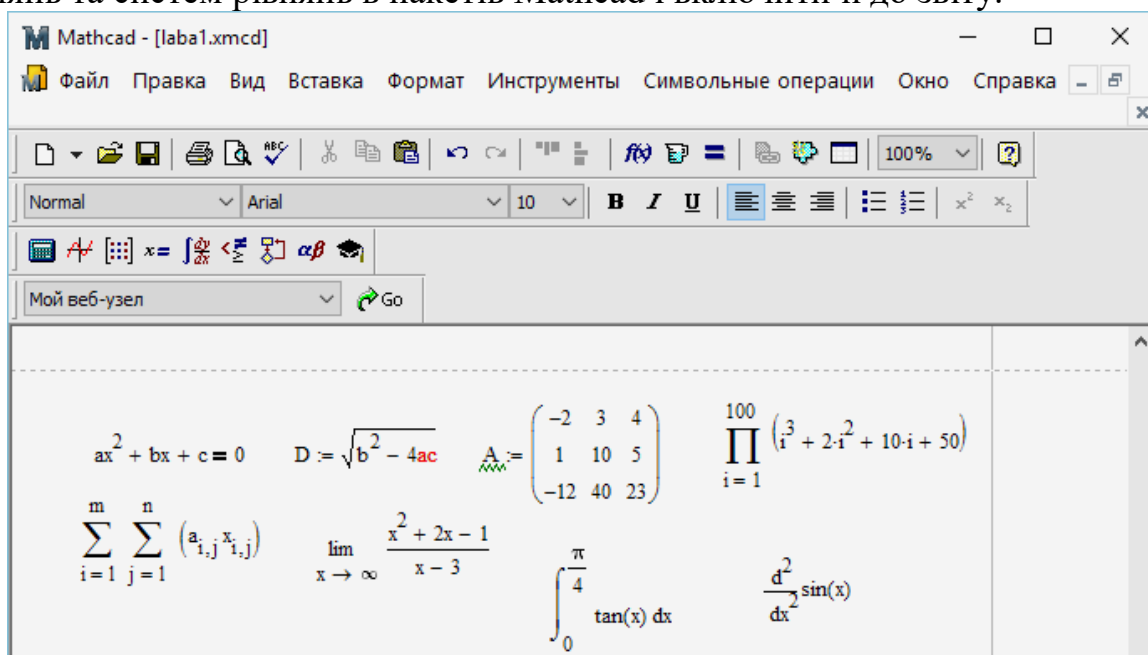


Рисунок 1.30 – Приклад застосування панелі інструментів пакету Mathcad

5. Зайти на сайт математичного пакету Mathcad, з'ясувати яка остання версія цього продукту зараз розповсюджується на ринку і скопіювати цю інформацію до звіту.
6. Проаналізувати розв'язування задачі з параметром і реалізувати її за допомогою пакету Mathcad:

Знайти, при яких значеннях параметра  $a$  система

$$x - 5 \cdot y = 7$$

$$a \cdot x - y = -3$$

- 1) має один корінь;
- 2) немає коренів.

### Розв'язування.

Для розв'язування заданої системи в символному вигляді скористаємось командою *find*. Для цього спочатку, використовуючи розв'язковий блок, який починається з службового слова *given* введемо задану систему:

given

$$x - 5 \cdot y = 7$$

$$a \cdot x - y = -3$$

Після цього наберемо команду  $\text{find}(x,y) \rightarrow$ , де « $\rightarrow$ » – знак розв’язування задачі у символічному вигляді (меню «Symbolic»), і одержимо:

$$\text{find}(x,y) \rightarrow \left[ \begin{array}{c} \frac{-22}{(5 \cdot a - 1)} \\ \frac{-(7 \cdot a + 3)}{(5 \cdot a - 1)} \end{array} \right]$$

Один розв’язок система буде мати тоді, коли обидва корені рівні, тому розв’яжемо наступне рівняння за допомогою команди  $\text{solve}$  меню «Symbolic»:

$$\frac{-22}{(5 \cdot a - 1)} - \frac{-(7 \cdot a + 3)}{(5 \cdot a - 1)} = 0 \text{ solve, } a \rightarrow \frac{19}{7}$$

Система не буде мати розв’язків, коли знаменник у її коренях буде рівним 0:

$$5 \cdot a - 1 = 0 \text{ solve, } a \rightarrow \frac{1}{5}$$

Відповідь:

1) система має один розв’язок тоді, коли  $a = \frac{19}{7}$ ;

2) система немає розв’язків тоді, коли  $a = \frac{1}{5}$ .

7. Виконати наступні завдання з різними типами даних:

1) знайти чому дорівнюють задані числа в десятковій системі числення:

$100001111_2$ ;  $567_8$ ;  $0AD8F5_{16}$ ;

2) придумати квадратні рівняння, у яких є один комплексний корінь і два комплексних корені;

3) з’ясувати, які дії можна робити з комплексними числами у пакеті Mathcad;

4) ввести дані, які подані на рис. 24, за допомогою довідки пакету Mathcad з’ясувати, які операції можна виконувати над матрицями; виконати

знайдені операції над матрицею  $M := \begin{pmatrix} 1 & 4 & 5 \\ -3 & -1 & 8 \\ 3 & -2 & 9 \end{pmatrix}$ ;

5) ввести дані, які подані на рис. 26, 27;

б) створити ранжовані змінні з такими характеристиками:  $i := 5..15$ ;  $k := -5, -4.9..-4$ ; і знайти значення виразів:  $\sum_i i$ ;  $\prod_k k$ .

### III. Індивідуальні завдання

1. Знайти розв'язки наступних рівнянь аналітичним способом і за допомогою пакету Mathcad:

$$1) \frac{6x - x^2 - 6}{x - 1} - \frac{2x - 3}{x - 1} = 1;$$

$$2) \frac{2x + 1}{x} + \frac{4x}{2x + 1} = 5;$$

$$3) \frac{2}{x^2 + 5x} + \frac{3}{2x - 10} = \frac{15}{x^2 - 25};$$

$$4) \frac{14}{x^2 - 4} + \frac{3}{(2 - x)^2} = \frac{5}{(x + 2)^2};$$

$$5) \frac{4}{x^2 + 4} + \frac{5}{x^2 + 5} = 2;$$

$$6) \frac{x^2 + 1}{x} + \frac{x}{x^2 + 1} = \frac{5}{2};$$

$$7) x^4 - 5x^2 + 2 = 0;$$

$$8) \left(\frac{x - 1}{x}\right)^2 + 3\left(\frac{x - 1}{x}\right) + 2 = 0;$$

$$9) \frac{x^2 + 2x - 3}{x^2 - 2x + 8} = 0;$$

$$10) \frac{x^2 + 5x + 4}{x^2 - 5x - 6} = 0.$$

2. Знайти розв'язки наступних систем рівнянь аналітичним способом і за допомогою пакету Mathcad:

$$1) \begin{cases} \frac{y}{x} + \frac{x}{y} = \frac{13}{6}, \\ x + y = 5; \end{cases}$$

$$2) \begin{cases} \frac{y}{x} = 2, \\ (x - 1)^2 + y^2 = 1; \end{cases}$$

$$3) \begin{cases} \frac{1}{x} + \frac{1}{y} = 5, \\ \frac{1}{x^2} + \frac{1}{y^2} = 13; \end{cases}$$

$$4) \begin{cases} x^2 y^3 + x^3 y^2 = 12, \\ x^2 y^3 - x^3 y^2 = 4; \end{cases}$$

$$5) \begin{cases} (x - y)(x^2 - y^2) = 45, \\ x + y = 5; \end{cases}$$

$$6) \begin{cases} x^3 + y^3 = 9, \\ xy = 2; \end{cases}$$

$$7) \begin{cases} x^3 + y^3 = 7, \\ x^3 y^3 = -8; \end{cases}$$

$$8) \begin{cases} x^2 - xy = 28, \\ y^2 - xy = -12; \end{cases}$$

$$9) \begin{cases} x^2 + y^4 = 5, \\ xy^2 = 2; \end{cases}$$

$$10) \begin{cases} x^2 y^3 = 16, \\ x^3 y^2 = 2. \end{cases}$$

3. Підготувати коротку довідку про логічні оператори і функції, основні алгебраїчні, тригонометричні і гіперболічні функції, цілочислові і комбінаторні функції та перевірити їх роботу.

4. Створити функції користувача і побудувати їх графіки:

$$1) f(x) = 2\cos x \operatorname{tg} x;$$

$$2) f(x) = \operatorname{tg} x + \operatorname{ctg} x;$$

$$3) f(x) = \frac{\sin x \cos^2 x}{x};$$

$$4) f(x) = (\sin x - \cos x)^2;$$

$$5) f(x) = \sqrt{x^2 3^x - 3^{x+1}};$$

$$6) f(x) = 5^{\log_3(x-1)};$$

$$7) f(x) = \begin{cases} \frac{1}{x+1}, & x \geq 0 \\ x^5 + 3x^4 - 2x^2 + 7x - 6, & x < 0 \end{cases};$$

$$8) y = (\log_2 3x)^{\sin x};$$

$$9) f(x) = |x-3| + |x-1| + |x+2| + |x+6|;$$

$$10) f(x) = \frac{1}{\sqrt{4x+1}} - \sqrt{1-x^2}.$$

Власноруч знайти області визначення заданих функцій і дослідити, чи правильно побудовані графіки цих функцій.

5. Зайти на сайт системи комп'ютерної математики (у відповідності з номером варіанту) (табл.2) і дати загальну характеристику цієї системи.

6. Результати розв'язування індивідуальних завдань зберегти у файлі пакету Mathcad з іменем: Task\_lab\_1.mcd.

7. Оформити звіт про виконану роботу (див. додаток) в електронному вигляді і надіслати його через систему електронного навчання.

## Лабораторна робота № 2

### Тема: «Алгебра в СКМ Mathcad»

Питання:

1. Введення матриць.
2. Матричні оператори і функції.
3. Розв'язування алгебраїчних рівнянь та їх систем.
4. Символьні матричні перетворення.

### I. Теоретичний матеріал

Пакет Mathcad, починаючи з восьмої версії, має значний набір матричних операцій і функцій, тому в області чисельних застосувань лінійної алгебри він є цілком функціонально завершеним, а, враховуючи чудову візуалізацію матричних і векторних обчислень, що дуже важливо для навчального процесу, Mathcad істотно краще за інші, навіть більш складні і дорогі математичні пакети.

#### 1. Введення матриць

У математичному пакеті Mathcad, зокрема Mathcad 2000 Professional або Mathcad 2001 Professional, введення елементів матриць можна виконувати наступним чином.

1. За допомогою шаблону задання матриць, який викликається з палітри матричних операцій «Matrix» або комбінації клавіш «Ctrl»+ «M» (рис. 2.1). У цьому випадку кожен елемент вводиться «вручну» на своє місце в запропонований «макет» матриці, розмір якої попередньо задається користувачем. При подальшій роботі для того щоб дізнатися, яке число знаходиться в матриці в  $i$ -му рядку та  $j$ -му стовпці, досить вказати ім'я матриці та індекси цього елемента через кому (рис. 2.1). Індекси елементів матриці вводяться за допомогою символу  $\times_n$  палітри «Matrix» або за допомогою натискання клавіші «[», причому нумерація рядків і стовпців матриці за замовченням починається з нуля, а не з одиниці, як прийнято традиційно.

Зауваження 1. Для зміни початкового значення індексу матриці використовується змінна ORIGIN. Наприклад за допомогою виразу  $ORIGIN:=1$ , який задається перед описом матриці, всі її індекси будуть нумеруватися з 1 (рис. 2.1). Значення цієї змінної можна також змінити за допомогою режиму встановлення опцій системи:  $Math\Options\Array\Origin$ .

Зауваження 2. Якщо спробувати визначити елемент матриці з номером рядка або стовпця, яких не існує, то на екран виводиться повідомлення про помилку і про причину її виникнення, зокрема, «Величина верхнього або нижнього індексу занадто велика (або занадто мала) для цього масиву» (рис. 2.1).

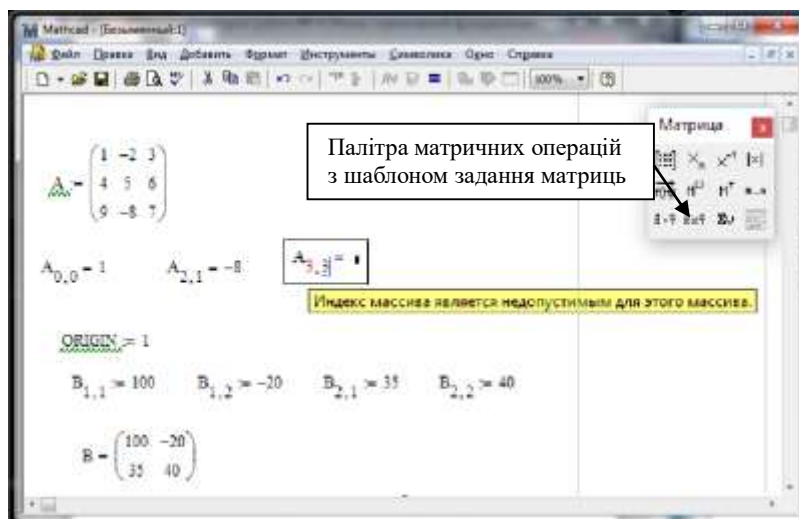


Рисунок 2.1 – Приклад введення матриць

2. Безпосереднє задання елементів матриці за допомогою операції присвоювання (рис. 2.1).

3. За допомогою застосування ранжованих змінних матриці без ручного заповнення шаблонів (рис. 2.2).

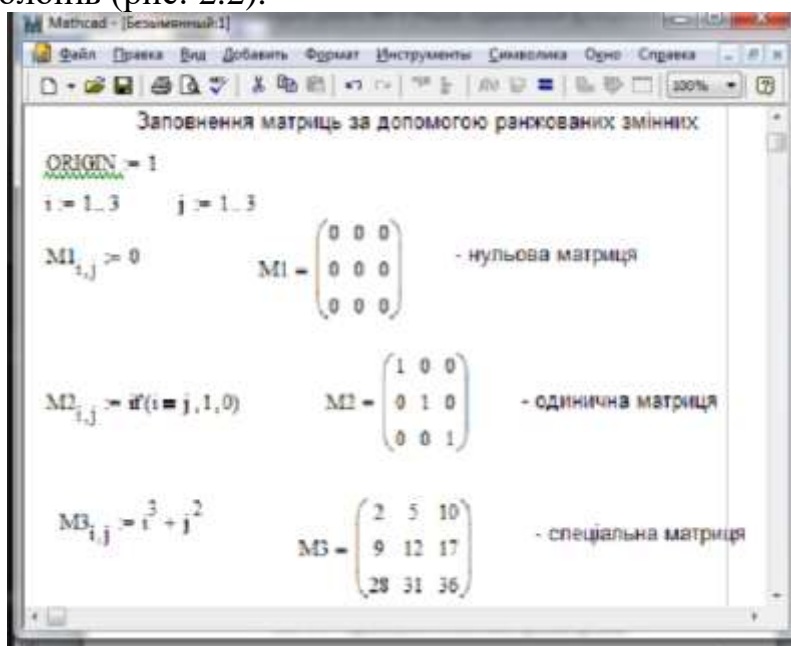


Рисунок 2.2 – Приклад застосування ранжованих змінних

## 2. Матричні оператори

Для роботи з матрицями пакет Mathcad пропонує такі основні оператори:

Оператор	Введення	Призначення
$-M$	$-M$	Зміна знаків елементів матриці на протилежні
$Z * M, M * Z$	$Z * M, M * Z$	Множення матриці $M$ на скаляр $Z$
$M * V$	$M * V$	Множення матриці $M$ на вектор $V$
$M_1 * M_2$	$M_1 * M_2$	Множення матриць $M_1$ і $M_2$
$\frac{M}{Z}$	$M/Z$	Ділення матриці $M$ на скаляр $Z$



$M^{-1}$	$M^{-1}$	Обчислення оберненої матриці до матриці $M$
$M^n$	$M^n$	Піднесення матриці $M$ до $n$ -го ступеня
$ M $	$ M $	Обчислення визначника матриці $M$
$M^T$	$M^T$	Транспонування матриці $M$
$M^{<n>}$	$M^{<n>}$	Виділення $n$ -го стовпчика матриці $M$
$M_{m,n}$	$M[(m,n)]$	Виділення елемента $(m, n)$ матриці $M$
$\overline{M}$	$M^*$	Отримання комплексно-спряженої матриці до матриці $M$

Всі вказані вище оператори, крім останнього, можна викликати з палітри матричних операцій. На рис. 3 наведено приклади застосування операторів для роботи з матрицями.

### 3. Матричні функції

Для роботи з матрицями в пакеті Mathcad існує ряд вбудованих функцій:

- $\text{augment}(M_1, M_2)$  – об'єднує в одну дві матриці  $M_1$  та  $M_2$ , які мають однакову кількість рядків (об'єднання йде „по горизонталі»);
- $\text{stack}(M_1, M_2)$  – об'єднує „по вертикалі» дві матриці  $M_1$  та  $M_2$ , які мають однакову кількість стовпців;
- $\text{identity}(n)$  – створює одиничну квадратну матрицю порядку  $n$ ;
- $\text{submatrix}(A, ir, jr, ic, jc)$  – створює підматрицю, яка складається з усіх елементів, які містяться в рядках з  $ir$  по  $jr$  та стовпців з  $ic$  по  $jc$ ;
- $\text{matrix}(m, n, f)$  – створює матрицю, в якій  $(i, j)$ -й елемент дорівнює  $f(i, j)$ , де  $i=0,1,\dots,m, j=0,1,\dots,n, f(i, j)$  – деяка функція;
- $\text{diag}(V)$  – створює діагональну матрицю, головною діагоналлю якої є вектор  $V$ ;
- $\text{Re}(M)$  – створює матрицю дійсних частин матриці  $M$  з комплексними елементами;
- $\text{Im}(M)$  – створює матрицю уявних частин матриці  $M$  з комплексними елементами;
- $\text{cols}(M)$  – визначає кількість стовпців матриці  $M$ ;
- $\text{rows}(M)$  – визначає кількість рядків матриці  $M$ ;
- $\text{rank}(M)$  – обчислює ранг матриці  $M$ ;
- $\text{tr}(M)$  – обчислює слід (суму діагональних елементів) квадратної матриці  $M$ ;
- $\text{mean}(M)$  – обчислює середнє значення елементів масиву;
- $\text{csort}(M, n)$  – переставляє рядки матриці  $M$  таким чином, що відсортованим стає  $n$ -ий стовпець;
- $\text{rsort}(M, n)$  – переставляє стовпці матриці  $M$  таким чином, що відсортованим стає  $n$ -ий рядок;
- $\text{rref}(M)$  – повертає спеціальну форму матриці  $M$  (у вигляді східців), коли перший ненульовий елемент в кожному рядку дорівнює 1, яка знаходиться справа від першого ненульового елемента попереднього рядка, тобто всі елементи вище першого ненульового в рядку – нулі (використовується для розв'язування систем лінійних рівнянь методом Гаусса);

- $\text{eigenvals}(M)$  – знаходження власних значень матриці  $M$ ;
- $\text{eigenvecs}(M)$  – знаходження власних векторів матриці  $M$ ;
- $\text{eigenvec}(M,z)$  – знаходження власного вектора матриці  $M$ , що відповідає власному значенню.

Зауваження 3. Пакет Mathcad має ще цілий ряд додаткових матричних функцій, які не застосовуються при вивченні даного курсу, тому розгляд додаткових матричних функцій залишається на самостійне опрацювання студентів.

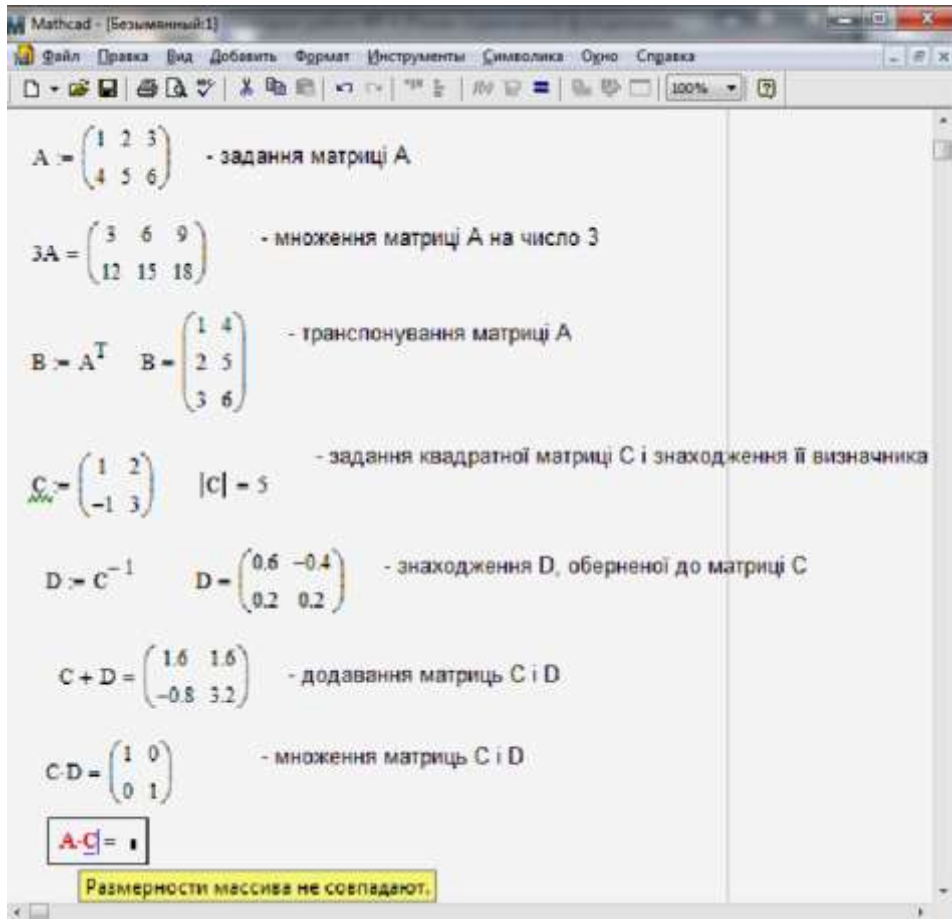


Рисунок 2.3 – Приклад використання матричних операторів

Зауваження 4. Якщо спробувати виконати некоректне множення двох матриць, то на екран виводиться повідомлення про помилку і про причину її виникнення, зокрема, «Кількість рядків і/або стовпців у цих масивах не співпадає» (рис. 2.3).

На рис. 2.4 зображено дії над матрицями, елементами яких є комплексні числа.

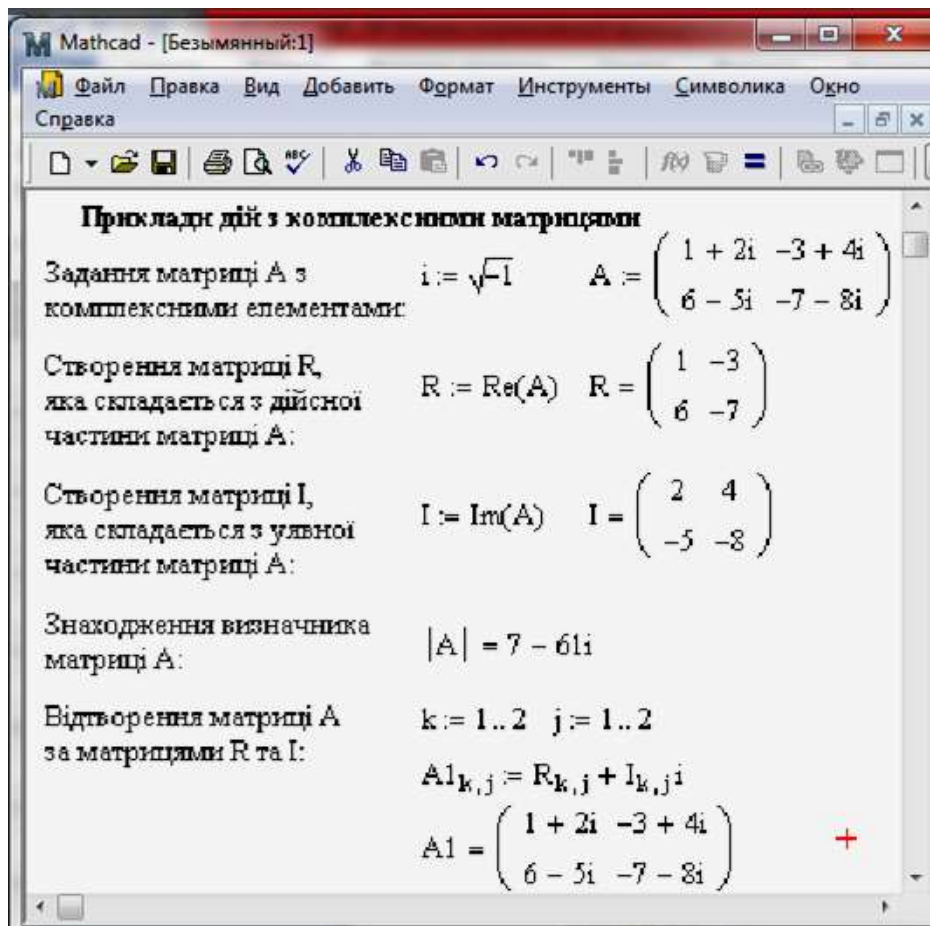


Рисунок 2.4 – Дії над матрицями, елементами яких є комплексні числа

#### 4. Розв'язування матричних рівнянь

Матричні оператори і функції пакету Mathcad дозволяють розв'язувати широке коло задач лінійної алгебри, зокрема матричні рівняння (рис. 2.5).

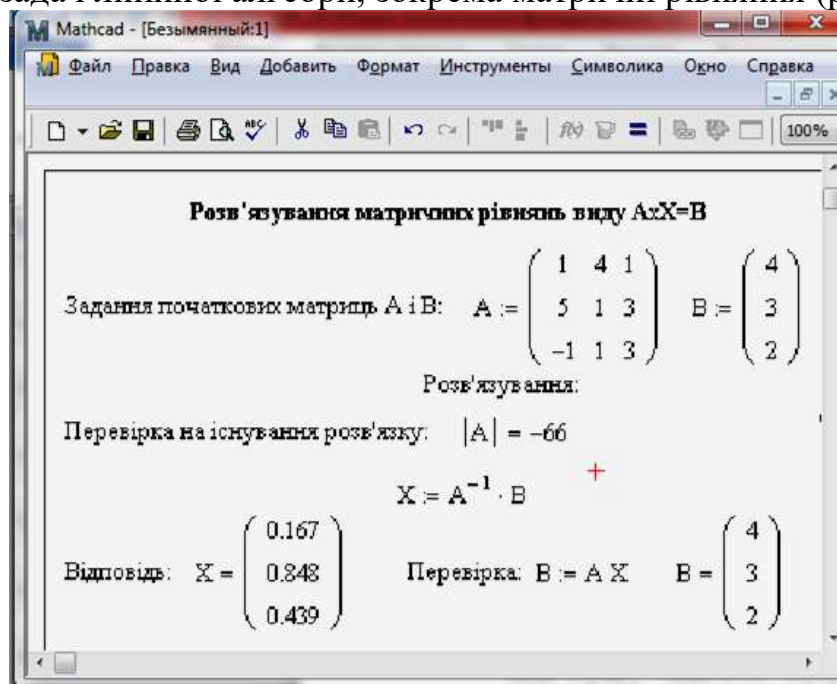


Рисунок 2.5 – Приклад розв'язування матричного рівняння

**Зауваження 5.** При спробі розв'язати матричне рівняння, яке не має розв'язку, тобто матриця  $A$  сингулярна (її визначник дорівнює 0), Mathcad виводить відповідне повідомлення. Але, як показує досвід, іноді цього не трапляється і рівняння розв'язується неправильно. Тому студентам варто порадити організувати самим перевірку на рівність визначника відповідної матриці 0 (рис. 2.5).

Оскільки задача розв'язування систем лінійних рівнянь є досить поширеною, то для цього в пакеті Mathcad, починаючи з шостої версії, присутня вбудована функція  $\text{lsolve}(A,B)$ , яка, при заданні матриці коефіцієнтів  $A$  і вектору вільних членів  $B$ , повертає вектор  $X$ , котрий є розв'язком системи  $AX=B$  (рис. 2.6).

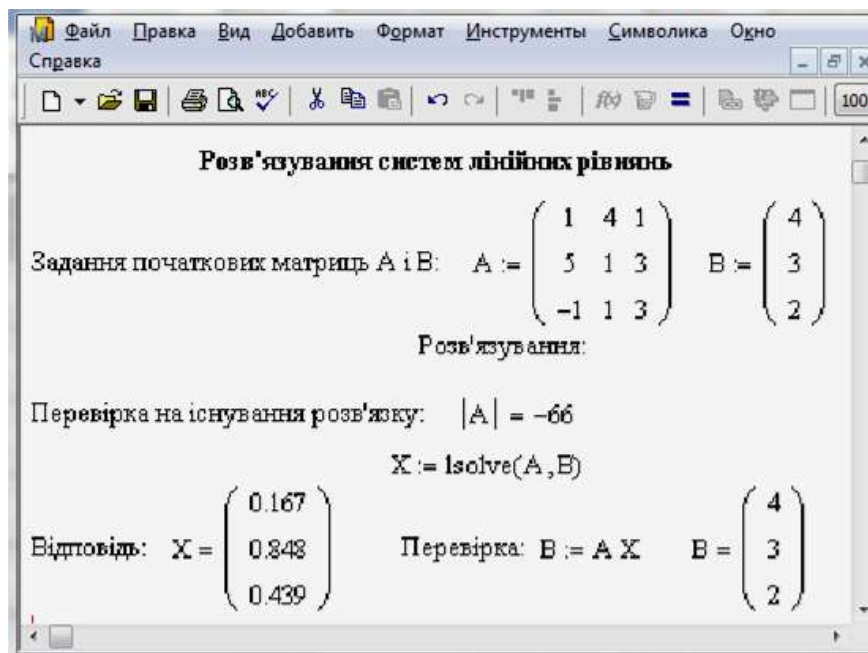


Рисунок 2.6 – Приклад розв'язування систем лінійних рівнянь

## 5. Символьні матричні перетворення

Останні версії пакету Mathcad мають досить широкі можливості щодо розв'язування задач лінійної алгебри в символічному вигляді. Символьні перетворення матриць здійснюються за допомогою операції символічних обчислень (Symbolic Evaluation) (символ « $\rightarrow$ »), яку можна активізувати, наприклад, за допомогою палітри символічних перетворень (Symbolic) або комбінації клавіш «Ctrl»+«+» (рис. 2.7). Крім того, символічні перетворення матриць можна виконувати за допомогою наступних дій:  $\text{Symbolics} \backslash \text{Evaluate} \backslash \text{Symbolically}$  (комбінація клавіш «Shift»+«F9»).

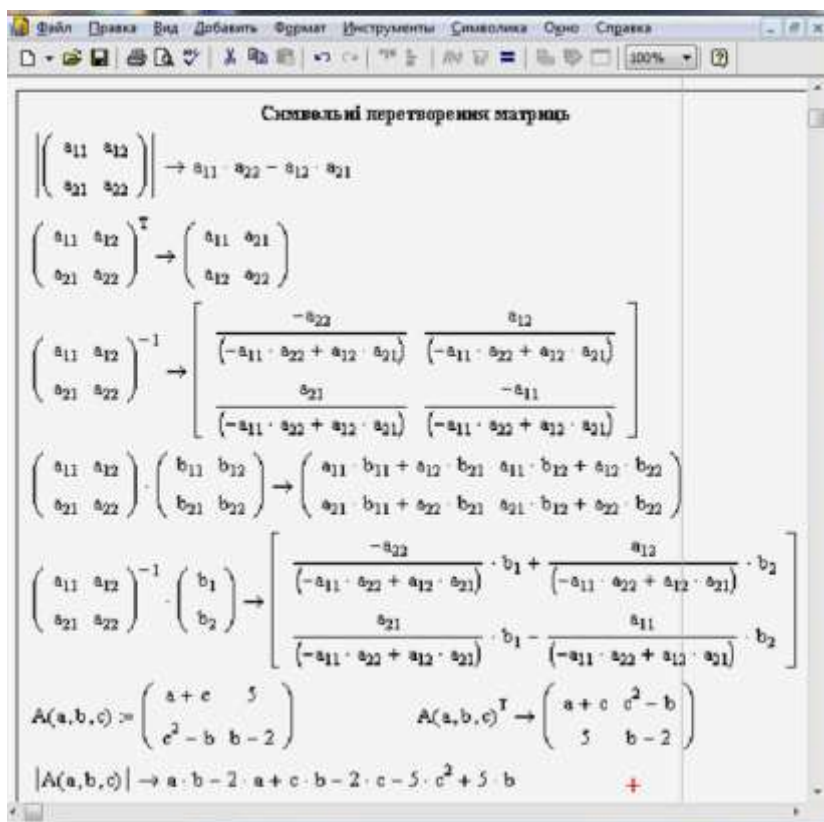


Рисунок 2.7 – Застосування символьних перетворень матриці

## II. Завдання до лабораторної роботи

1. Ознайомитися з теоретичним матеріалом. Виконати перетворення подані на рис. 1-7.

2. Обчислити визначники 2-го та 3-го порядків:

$$1) \begin{vmatrix} 13547 & 13647 \\ 28423 & 28423 \end{vmatrix}; \quad 2) \begin{vmatrix} \sin \alpha & \cos \alpha \\ -\cos \alpha & \sin \alpha \end{vmatrix}; \quad 3) \begin{vmatrix} 246 & 427 & 327 \\ 1014 & 543 & 443 \\ -342 & 721 & 621 \end{vmatrix}; \quad 4) \begin{vmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{vmatrix}$$

3. Розв'язати рівняння та нерівності:

$$1) \begin{vmatrix} x^2 & 4 & 9 \\ x & 3 & 3 \\ 1 & 1 & 1 \end{vmatrix} = 0; \quad 2) \begin{vmatrix} x & 0 & 1 \\ 2 & 1 & 3 \\ 4 & x & 1 \end{vmatrix} \leq 0;$$

4. Виконати дії:

$$1) \begin{pmatrix} 2 & 1 & 1 \\ 3 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 3 & 1 \\ 2 & 1 \\ 1 & 0 \end{pmatrix}; \quad 2) \begin{pmatrix} 3 & 1 & 1 \\ 2 & 1 & 2 \\ 1 & 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & -1 \\ 2 & -1 & 1 \\ 1 & 0 & 1 \end{pmatrix}; \quad 3) \begin{pmatrix} 2 & 1 & 1 \\ 3 & 1 & 0 \\ 0 & 1 & 2 \end{pmatrix}^2; \quad 4) \begin{pmatrix} 3 & 2 \\ -4 & -2 \end{pmatrix}^5.$$

5. Представити матрицю  $A$  у вигляді суми симетричної та кососиметричної матриць:

$$1) A = \begin{pmatrix} 5 & -2 & -3 \\ 6 & 1 & 2 \\ 4 & -1 & 1 \end{pmatrix}; 2) A = \begin{pmatrix} 2 & 2 & -3 \\ 3 & -1 & 7 \\ -5 & 2 & 8 \end{pmatrix}.$$

**П р и м і т к а.** Квадратна матриця  $A$  називається *симетричною*, якщо вона дорівнює транспонованій до неї матриці, тобто  $A = A^T$ , і *кососиметричною*, якщо вона дорівнює транспонованій до неї матриці, яка взята з протилежним знаком, тобто  $A = -A^T$ . Причому, довільну квадратну матрицю можна записати у вигляді суми симетричної і кососиметричної матриць  $A = S + K$ , де симетрична матриця знаходиться за формулою  $S = \frac{A + A^T}{2}$ , а кососиметрична  $K = \frac{A - A^T}{2}$ .

6. Дослідити системи на сумісність, використовуючи критерій Кронекера-Капеллі. Якщо система сумісна, то знайти її розв'язок:

$$1) \begin{cases} x_1 - 2x_2 + x_3 + x_4 - x_5 = 0, \\ 2x_1 + x_2 - x_3 - x_4 + x_5 = 0, \\ x_1 + 7x_2 - 5x_3 - 5x_4 + 5x_5 = 0, \\ 3x_1 - x_2 - 2x_3 + x_4 - x_5 = 0; \end{cases} \quad 2) \begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 11, \\ 2x_1 + 3x_2 + 4x_3 + x_4 = 12, \\ 3x_1 + 4x_2 + x_3 + 2x_4 = 13, \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 14; \end{cases}$$

7. Розв'язати системи лінійних рівнянь:

а) методом Гауса;

б) методом Крамера;

с) матричним методом.

$$1) \begin{cases} x_2 - 3x_3 + 4x_4 = -5, \\ x_1 - 2x_3 + 3x_4 = -4, \\ 3x_1 + 2x_2 - 5x_4 = 12, \\ 4x_1 + 3x_2 - 5x_3 = 5; \end{cases} \quad 2) \begin{cases} x_1 - 2x_2 + 3x_3 - 4x_4 = 4, \\ x_2 - x_3 + x_4 = -3, \\ x_1 + 3x_2 - 3x_4 = 1, \\ -7x_2 + 3x_3 + x_4 = -8; \end{cases}$$

З'ясувати чи задані системи рівнянь погано обумовлені чи ні.

**П р и м і т к а.** Якщо числа  $d$  і  $D$  відповідно найменше і найбільше власні значення матриці системи, то при  $\left| \frac{d}{D} \right| \approx 1$  (тобто коли вони мало відрізняються по модулю одне від одного) вважається, що матриця і відповідно система рівнянь добре обумовлена. Якщо ж  $\left| \frac{d}{D} \right| \ll 1$ , то вважається, що матриця і відповідно система рівнянь погано обумовлена.

8. Підготувати звіт про виконання завдань 1-7 та індивідуального завдання.

9. Підготуватися до самостійної роботи з теми «Розв'язування задач лінійної алгебри за допомогою пакету Mathcad».

### III. Індивідуальні завдання

1. Обчислити визначник, використовуючи символні перетворення:

$$1. \begin{vmatrix} \sin^2 \alpha & 1 & \cos^2 \alpha \\ \sin^2 \beta & 1 & \cos^2 \beta \\ \sin^2 \gamma & 1 & \cos^2 \gamma \end{vmatrix}. 2. \begin{vmatrix} \sin \alpha & \cos \alpha & \sin(\alpha + \delta) \\ \sin \beta & \cos \beta & \sin(\beta + \delta) \\ \sin \gamma & \cos \gamma & \sin(\gamma + \delta) \end{vmatrix}. 3. \begin{vmatrix} \sin \alpha & \cos \alpha & 1 \\ \sin \beta & \cos \beta & 1 \\ \sin \gamma & \cos \gamma & 1 \end{vmatrix}.$$

$$4. \begin{vmatrix} \alpha^2 + 1 & \alpha\beta & \alpha\gamma \\ \alpha\beta & \beta^2 + 1 & \beta\gamma \\ \alpha\gamma & \beta\gamma & \gamma^2 + 1 \end{vmatrix}. 5. \begin{vmatrix} \cos \alpha & \sin \alpha \cos \beta & \sin \alpha \sin \beta \\ -\sin \alpha & \cos \alpha \cos \beta & \cos \alpha \sin \beta \\ 0 & -\sin \beta & \cos \beta \end{vmatrix}. 6. \begin{vmatrix} a+x & x & x \\ x & b+x & x \\ x & x & c+x \end{vmatrix}.$$

$$7. \begin{vmatrix} a_1 & b_1 & a_1x + b_1y + c_1 \\ a_2 & b_2 & a_2x + b_2y + c_2 \\ a_3 & b_3 & a_3x + b_3y + c_3 \end{vmatrix}. 8. \begin{vmatrix} 1 & a & bc \\ 1 & b & ca \\ 1 & c & ab \end{vmatrix}. 9. \begin{vmatrix} a & -2 & 3 & 4 \\ 2 & b & -4 & 3 \\ 3 & -4 & c & -3 \\ 4 & -3 & 2 & d \end{vmatrix}. 10. \begin{vmatrix} 0 & 1 & 1 & a \\ 1 & 0 & 1 & b \\ 1 & 1 & 0 & c \\ a & b & c & d \end{vmatrix}.$$

2. Знайти невідому матрицю з рівняння:

$$1. X \begin{pmatrix} 1 & 1 & -1 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 3 \\ 4 & 3 & 2 \\ 1 & -2 & 5 \end{pmatrix}. 2. X \begin{pmatrix} 1 & 3 & 5 \\ 2 & 7 & -8 \\ 1 & -3 & 4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 9 & -3 & 6 \\ 3 & 0 & 3 \end{pmatrix}.$$

$$3. X \begin{pmatrix} 1 & 1 & -1 \\ 2 & 2 & 0 \\ 3 & 4 & 5 \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 \\ 2 & 1 & 1 \end{pmatrix}. 4. \begin{pmatrix} 1 & 2 & -3 \\ 3 & 2 & -4 \\ 2 & -1 & 0 \end{pmatrix} X = \begin{pmatrix} 1 & -3 & 0 \\ 10 & 2 & 7 \\ 10 & 7 & 8 \end{pmatrix}. 5. X \begin{pmatrix} 2 & 5 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 4 & -6 \\ 2 & 1 \end{pmatrix}.$$

$$6. X \begin{pmatrix} 1 & -1 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 2 \end{pmatrix}. 7. X \begin{pmatrix} 2 & 2 & -1 \\ 2 & -1 & 2 \\ -1 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 5 & 1 & 1 \\ 2 & 3 & 4 \end{pmatrix}.$$

$$8. \begin{pmatrix} 1 & 1 & 0 \\ -1 & 2 & 1 \\ 0 & 0 & 3 \end{pmatrix} X \begin{pmatrix} 1 & 2 & -1 \\ 0 & 1 & 3 \\ 4 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 1 \\ -1 & 0 & 3 \\ 1 & 1 & 1 \end{pmatrix}. 9. \begin{pmatrix} 1 & 1 & 2 \\ -1 & 3 & 1 \\ 4 & 1 & -1 \end{pmatrix} X = \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}.$$

$$10. \begin{pmatrix} 1 & 1 \\ 2 & -1 \end{pmatrix} X \begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}.$$

3. Знайти ранг матриці:

$$1. \begin{pmatrix} 3 & 2 & 5 & -1 & 3 \\ 2 & 2 & 7 & 2 & 2 \\ 6 & 5 & 12 & 4 & 7 \\ 4 & 3 & 5 & 2 & 2 \\ -2 & -1 & -5 & 4 & -4 \end{pmatrix}. 2. \begin{pmatrix} 1 & -2 & 3 & -1 & -2 \\ 2 & -1 & 1 & 0 & -2 \\ -2 & -5 & 8 & -4 & -1 \\ 6 & 0 & -1 & 2 & -5 \\ -1 & -1 & 1 & -1 & 1 \end{pmatrix}. 3. \begin{pmatrix} 3 & -1 & 3 & 2 & 5 \\ 5 & -3 & 2 & 3 & 4 \\ 1 & -3 & -5 & 0 & -7 \\ 7 & -5 & 1 & 4 & 1 \end{pmatrix}.$$

$$4. \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 \\ 1 & 1 & 4 & 1 \\ 1 & 1 & 1 & 5 \\ 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} \cdot 5. \begin{pmatrix} 1 & 2 & -1 & 3 & 4 \\ 1 & 3 & 2 & 5 & 7 \\ 2 & 5 & 1 & 8 & 11 \\ 3 & 8 & 3 & 13 & 18 \end{pmatrix} \cdot 6. \begin{pmatrix} 5 & -2 & 1 & 3 & 4 \\ 2 & 5 & 5 & -3 & 2 \\ 2 & -5 & -1 & 4 & 1 \\ 6 & 4 & -2 & 5 & 5 \end{pmatrix} \cdot 7. \begin{pmatrix} 14 & 12 & 6 & 8 & 2 \\ 6 & 104 & 21 & 9 & 17 \\ 7 & 6 & 3 & 4 & 1 \\ 35 & 30 & 15 & 20 & 8 \end{pmatrix}$$

$$8. \begin{pmatrix} 1 & 0 & 0 & 1 & 4 \\ 0 & 1 & 0 & 2 & 5 \\ 0 & 0 & 1 & 3 & 6 \\ 1 & 2 & 3 & 14 & 32 \\ 4 & 5 & 6 & 32 & 77 \end{pmatrix} \cdot 9. \begin{pmatrix} 1 & -1 & 2 & 3 & 4 \\ 2 & 1 & -1 & 2 & 0 \\ -1 & 2 & 1 & 1 & 3 \\ 1 & 5 & -8 & -5 & -12 \\ 3 & -7 & 8 & 9 & 13 \end{pmatrix} \cdot 10. \begin{pmatrix} 3 & 2 & -1 & 2 & 0 & 1 \\ 4 & 1 & 0 & -3 & 0 & 2 \\ 2 & -1 & -2 & 1 & 1 & -3 \\ 3 & 1 & 3 & -9 & -1 & 6 \\ 3 & -1 & -5 & 7 & 2 & -7 \end{pmatrix}.$$

4. Дослідити систему лінійних рівнянь на сумісність і розв'язати її:

$$1. \begin{cases} x_1 + 2x_2 - x_3 = -3 \\ 2x_1 + 3x_2 + x_3 = -1 \\ x_1 + x_2 + 2x_3 = 2 \\ -x_1 - x_2 - 2x_3 = -2 \end{cases} \quad 2. \begin{cases} 3x_1 + 4x_2 - 2x_3 = 8 \\ 2x_1 - x_2 - 3x_3 = -4 \\ -13x_1 - 10x_2 + 12x_3 = -16 \\ 5x_1 + 3x_2 - 5x_3 = 4 \end{cases}$$

$$3. \begin{cases} 2x_1 - x_2 - x_3 = 4 \\ 3x_1 + 4x_2 - 2x_3 = 11 \\ -x_1 + 6x_2 = 3 \\ 5x_1 + 14x_2 - 4x_3 = 25 \end{cases} \quad 4. \begin{cases} x_1 - 4x_2 - 2x_3 = -3 \\ 3x_1 + x_2 + x_3 = -5 \\ 4x_1 - 3x_2 - x_3 = -8 \\ 6x_1 - 11x_2 - 5x_3 = -14 \end{cases}$$

$$5. \begin{cases} 3x_1 - 2x_2 + 4x_3 = -4 \\ 2x_1 + 3x_2 + 4x_3 = -20 \\ -4x_1 + 7x_2 - 4x_3 = -12 \\ 5x_1 + x_2 + 8x_3 = -24 \end{cases} \quad 6. \begin{cases} x_1 + 2x_2 + 4x_3 = 31 \\ 5x_1 + x_2 + 2x_3 = 20 \\ -4x_1 + x_2 + 2x_3 = 11 \\ 6x_1 + 3x_2 + 6x_3 = 51 \end{cases}$$

$$7. \begin{cases} 2x_1 - x_2 + x_3 = -4 \\ 3x_1 + x_2 - x_3 = -1 \\ -x_1 + 3x_2 - 3x_3 = 7 \\ x_1 + x_2 - 2x_3 = 3 \end{cases} \quad 8. \begin{cases} x_1 + 3x_2 - 3x_3 = 12 \\ 2x_1 - 3x_2 + 3x_3 = -10 \\ x_1 - 6x_2 + 6x_3 = -23 \\ 5x_1 - 3x_2 + 3x_3 = -7 \end{cases}$$

$$9. \begin{cases} 4x_1 - 2x_2 + 3x_3 = -7 \\ 3x_1 - x_2 + x_3 = 9 \\ 15x_1 - 7x_2 + 10x_3 = -12 \\ x_1 - x_2 + 2x_3 = -16 \end{cases} \quad 10. \begin{cases} 2x_1 - x_2 - x_3 = 0 \\ x_1 - x_2 - 3x_3 = 13 \\ 3x_1 - 2x_2 - 4x_3 = 13 \\ 9x_1 - 4x_2 - 2x_3 = -13 \end{cases}$$

5. Розв'язати систему лінійних рівнянь:

а) за методом Крамера;



б) матричним способом;

в) методом Гаусса.

$$1. \begin{cases} 3x_1 + 2x_2 + x_3 = 5 \\ 2x_1 + 3x_2 + x_3 = 1 \\ 2x_1 + x_2 + 3x_3 = 11 \end{cases} \quad 2. \begin{cases} 2x_1 + 3x_2 + 4x_3 = 3 \\ x_1 + 2x_2 + 3x_3 = 3 \\ 5x_1 + x_2 + 2x_3 = -2 \end{cases}$$

$$3. \begin{cases} 7x_1 + 2x_2 + 4x_3 = 1 \\ -x_1 + 3x_2 + 2x_3 = -2 \\ x_1 - 4x_2 - x_3 = 8 \end{cases} \quad 4. \begin{cases} x_1 + x_2 + 2x_3 = -1 \\ 2x_1 - x_2 + 2x_3 = -4 \\ 4x_1 + x_2 + 4x_3 = -2 \end{cases}$$

$$5. \begin{cases} x_1 - 2x_2 + 3x_3 = 6 \\ 2x_1 + 3x_2 - 4x_3 = 20 \\ 3x_1 - 2x_2 - 5x_3 = 6 \end{cases} \quad 6. \begin{cases} x_1 - 4x_2 - 2x_3 = -3 \\ 3x_1 + x_2 + x_3 = 5 \\ 3x_1 - 5x_2 - 6x_3 = -9 \end{cases}$$

$$7. \begin{cases} 4x_1 - 3x_2 + 2x_3 = 9 \\ 2x_1 + 5x_2 - 3x_3 = 4 \\ 5x_1 + 6x_2 - 2x_3 = 18 \end{cases} \quad 8. \begin{cases} 2x_1 + x_2 - x_3 = 2 \\ x_1 - x_2 + x_3 = 1 \\ 5x_1 + 2x_2 + 2x_3 = -1 \end{cases}$$

$$9. \begin{cases} x_1 + x_2 - x_3 = 1 \\ 8x_1 + 3x_2 - 6x_3 = 2 \\ 4x_1 + x_2 - 3x_3 = 3 \end{cases} \quad 10. \begin{cases} 3x_1 + 5x_2 + 7x_3 = 0 \\ 5x_1 + 7x_2 + x_3 = 8 \\ 7x_1 + x_2 + 3x_3 = -8 \end{cases}$$

З'ясувати чи задані системи рівнянь погано обумовлені чи ні.

#### IV. Завдання для самостійного виконання

1. Обчислити визначники 2-го та 3-го порядків:

$$1) \begin{vmatrix} 1+\sqrt{2} & 2-\sqrt{3} \\ 2+\sqrt{3} & 1-\sqrt{2} \end{vmatrix}; \quad 2) \begin{vmatrix} \cos(a-b) & \cos(b-c) & \cos(c-a) \\ \cos(a+b) & \cos(b+c) & \cos(c+a) \\ \sin(a+b) & \sin(b+c) & \sin(c+a) \end{vmatrix}; \quad 3) \begin{vmatrix} x & y & x+y \\ y & x+y & x \\ x+y & x & y \end{vmatrix}.$$

2. Розв'язати рівняння та нерівності:

$$1) \begin{vmatrix} x^2 & 3 & 2 \\ x & -1 & 1 \\ 0 & 1 & 4 \end{vmatrix} = 0; \quad 2) \begin{vmatrix} x & x^2 & 1 \\ 1 & x & x^2 \\ x^2 & 1 & x \end{vmatrix} < 1.$$

3. Використовуючи властивості визначників, довести рівності:

$$1) \begin{vmatrix} 1+2a & 1 & a & x \\ 1+2b & 2 & b & x \\ 1+2c & 3 & c & x \\ 1+2d & 4 & d & x \end{vmatrix} = 0; \quad 2) \begin{vmatrix} a & b & c & 1 \\ b & c & a & 1 \\ c & a & b & 1 \\ \frac{b+c}{2} & \frac{a+c}{2} & \frac{a+b}{2} & 1 \end{vmatrix} = 0; \quad 3) \begin{vmatrix} 0 & x & y & z \\ x & 0 & z & y \\ y & z & 0 & x \\ z & y & x & 0 \end{vmatrix} = \begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & z^2 & y^2 \\ 1 & z^2 & 1 & x^2 \\ 1 & y^2 & x^2 & 0 \end{vmatrix}.$$

4. Обчислити  $AB-BA$ , якщо:

$$1) A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \\ 1 & 2 & 3 \end{pmatrix} \text{ і } B = \begin{pmatrix} 4 & 1 & 1 \\ -4 & 2 & 0 \\ 1 & 2 & 1 \end{pmatrix}; \quad 2) A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 2 \\ -1 & 2 & 1 \end{pmatrix} \text{ і } B = \begin{pmatrix} 3 & 1 & -2 \\ 3 & -2 & 4 \\ -3 & 5 & -1 \end{pmatrix}.$$

5. Знайти  $f(A)$ , якщо:

$$1) f(x) = x^2 - x - 1, \quad A = \begin{pmatrix} 2 & 1 & 1 \\ 3 & 1 & 2 \\ 1 & -1 & 0 \end{pmatrix}; \quad 2) f(x) = x^2 - 5x + 3, \quad A = \begin{pmatrix} 2 & -1 \\ -3 & 3 \end{pmatrix};$$

$$3) f(x) = \frac{1+x}{1-x}, \quad A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}.$$

6. Розв'язати матричні рівняння:

$$1) \begin{pmatrix} 2 & 1 \\ 5 & 3 \end{pmatrix} \cdot X = \begin{pmatrix} 4 & -6 \\ 2 & 1 \end{pmatrix}; \quad 2) X \cdot \begin{pmatrix} 2 & 1 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; \quad 3) \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix} \cdot X \cdot \begin{pmatrix} -3 & 2 \\ 5 & -3 \end{pmatrix} = \begin{pmatrix} -2 & 4 \\ 3 & -1 \end{pmatrix}.$$

7. Дослідити системи на сумісність, використовуючи критерій Кронекера-Капеллі. Якщо система сумісна, то знайти її розв'язок:

$$1) \begin{cases} x_1 - 2x_2 + 3x_3 - 4x_4 + 2x_5 = -2, \\ x_1 + 2x_2 - x_3 - x_5 = -3, \\ x_1 - x_2 + 2x_3 - 3x_4 = 10, \\ x_2 - x_3 + x_4 - 2x_5 = -5, \\ 2x_1 + 3x_2 - x_3 + x_4 + 4x_5 = 1; \end{cases} \quad 2) \begin{cases} x_1 + 2x_2 - 3x_4 + 2x_5 = 1, \\ x_1 - x_2 - 3x_3 + x_4 - 3x_5 = 2, \\ 2x_1 - 3x_2 + 4x_3 - 5x_4 + 2x_5 = 7, \\ 9x_1 - 9x_2 + 6x_3 - 16x_4 + 2x_5 = 25; \end{cases}$$

$$3) \begin{cases} 3x_1 + x_2 - 2x_3 + x_4 - x_5 = 1, \\ x_1 + 2x_2 - x_3 + x_4 - 2x_5 = 1, \\ 4x_1 - 10x_2 + 5x_3 - 5x_4 + 7x_5 = 1, \\ 2x_1 - 14x_2 + 7x_3 - 7x_4 + 11x_5 = -1; \end{cases} \quad 4) \begin{cases} x_1 + 2x_2 + 3x_3 - x_4 = 1, \\ 3x_1 + 2x_2 + x_3 - x_4 = 1, \\ 2x_1 + 3x_2 + x_3 + x_4 = 1, \\ 2x_1 + 2x_2 + 2x_3 - x_4 = 1, \\ 5x_1 + 5x_2 + 2x_3 = 2. \end{cases}$$

8. Розв'язати системи лінійних рівнянь:

а) методом Гауса;

б) методом Крамера;

в) матричним методом.

$$1) \begin{cases} 3x_1 + 4x_2 - 5x_3 + 7x_4 = 0, \\ 2x_1 - 3x_2 + 3x_3 + 2x_4 = 0, \\ 4x_1 + 11x_2 - 13x_3 + 16x_4 = 0, \\ 7x_1 - 2x_2 + x_3 + 3x_4 = 0; \end{cases} \quad 2) \begin{cases} x_1 + 2x_2 + 3x_3 - 2x_4 = 6, \\ 2x_1 - x_2 - 2x_3 - 3x_4 = 8, \\ 3x_1 + 2x_2 - x_3 + 2x_4 = 4, \\ 2x_1 - 3x_2 + 2x_3 + x_4 = -8; \end{cases}$$

$$3) \begin{cases} 2x_1 + x_2 + 3x_3 + 4x_4 = 11, \\ 7x_1 + 3x_2 + 6x_3 + 8x_4 = 24, \\ 3x_1 + 2x_2 + 4x_3 + 5x_4 = 14, \\ x_1 + x_2 + 3x_3 + 4x_4 = 10; \end{cases} \quad 4) \begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 5, \\ 2x_1 + x_2 + 2x_3 + 3x_4 = 1, \\ 3x_1 + 2x_2 + x_3 + 2x_4 = 1, \\ 4x_1 + 3x_2 + 2x_3 + x_4 = -5. \end{cases}$$

**Лабораторна робота № 3**  
**Тема: «Основи математичного аналізу в СКМ»**

Питання:

1. Знаходження границь за допомогою системи Mathcad.
2. Аналітичне і чисельне диференціювання та їх реалізація в системі Mathcad.
3. Дослідження функцій і побудова їх графіків за допомогою системи Mathcad.
4. Знаходження невизначених і визначених інтегралів за допомогою системи Mathcad.

**I. Теоретичний матеріал**

**1. Знаходження границь функцій**

В системі Mathcad обчислення границі функції можливе лише у режимі символічних обчислень за допомогою оператора «lim» і оператора символічного виводу «→». Приклади обчислення границь різних функцій та їх графічна інтерпретація подані на рис. 3.1.

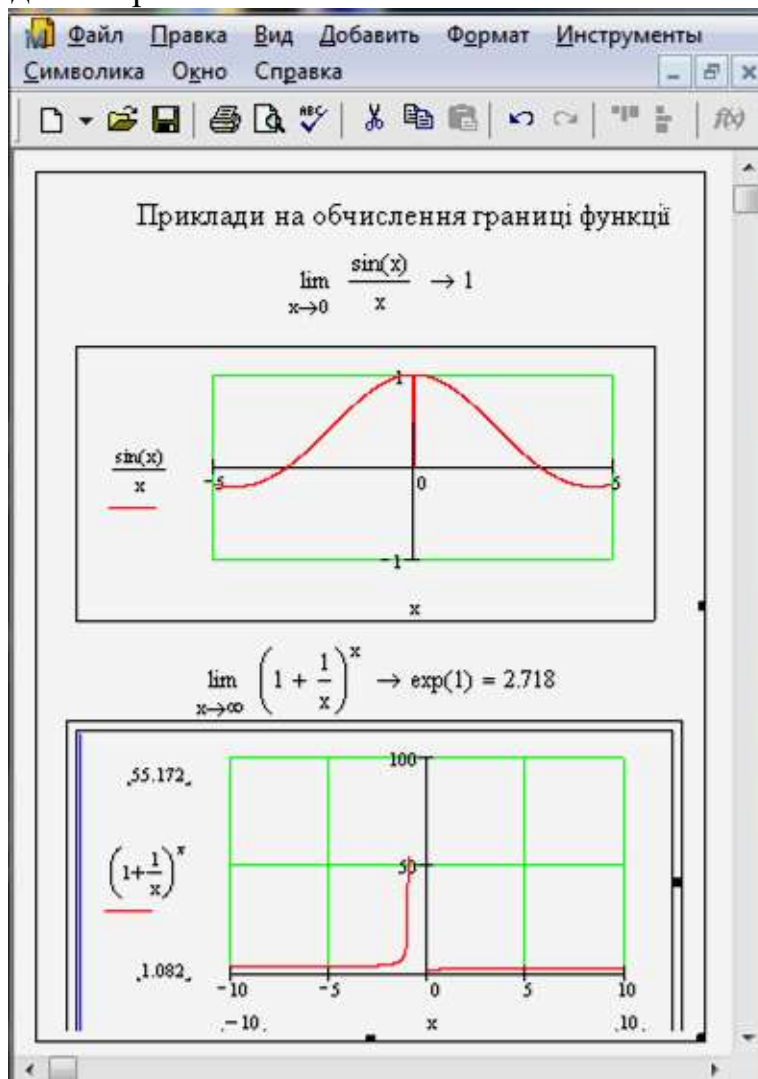


Рисунок 3.1 – Приклад на обчислення границі функції

## 2. Аналітичне і чисельне диференціювання та їх реалізація в системі Mathcad

В системі Mathcad аналітичне і чисельне диференціювання здійснюється в режимі символьних обчислень за допомогою операторі « $\frac{d}{dx}$ » і  $\frac{d^n}{dx^n}$ , які активізується через меню Calculus (рис. 3.2) і оператора символічного виведення « $\rightarrow$ », який активізується через меню Symbolic.

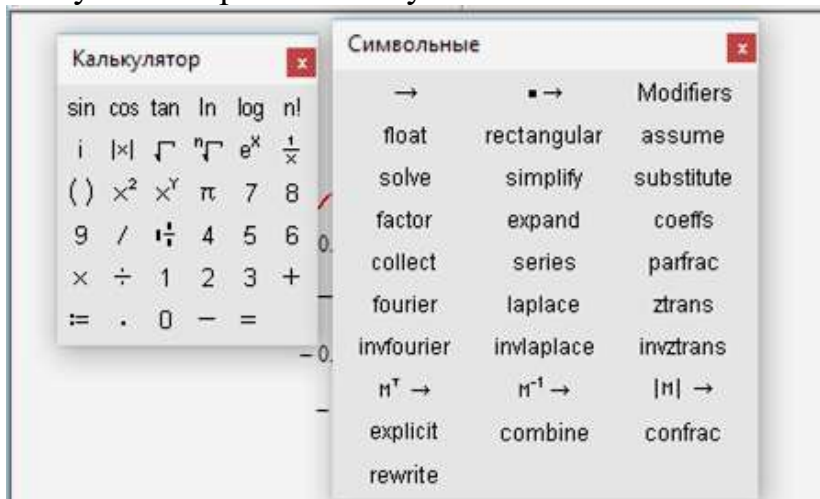


Рисунок 3.2 – Активація меню Calculus та Symbolic

Приклади обчислення похідних за означенням і з використанням вбудованих операторів подано на рис. 3.3.

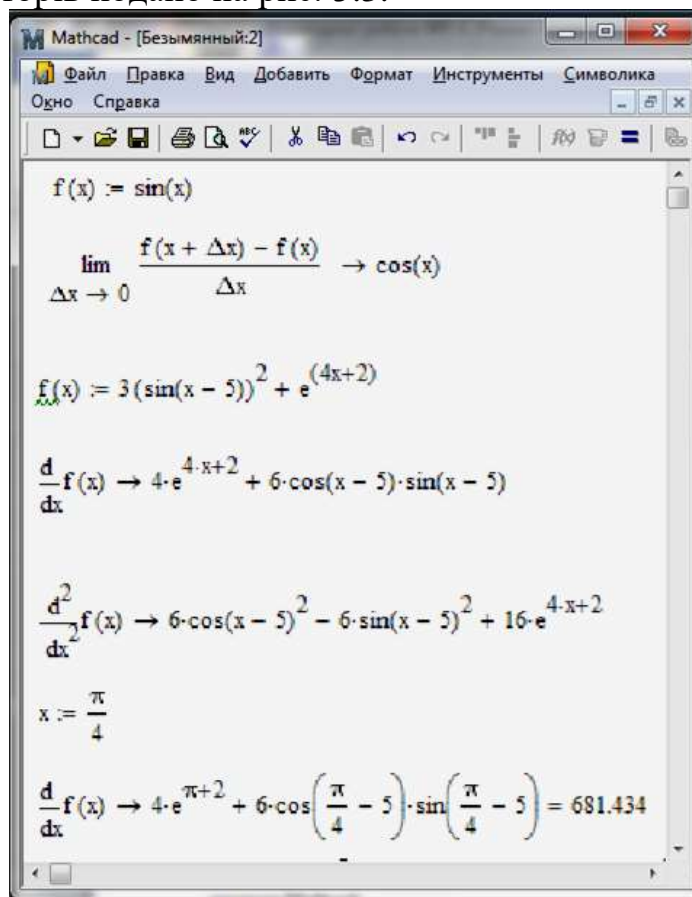


Рисунок 3.3 – Приклади обчислення похідних

### 3. Дослідження функцій і побудова їх графіків за допомогою системи Mathcad

Схема повного дослідження функції однієї змінної та побудова її графіка:

1. Знайти область визначення функції.
2. Дослідити функцію на парність, непарність.
3. Дослідити функцію на періодичність.
4. Знайти проміжки монотонності функції.
5. Знайти точки екстремуму функції.
6. Знайти інтервали опуклості (вгнутості) й точки перегину.
7. Знайти асимптоти функції:

- вертикальні виду  $x = a$ , де  $\lim_{x \rightarrow a} f(x) = \infty$ , або  $\lim_{x \rightarrow a-0} f(x) = \infty$ , або

$$\lim_{x \rightarrow a+0} f(x) = \infty;$$

- горизонтальні виду  $y = b$ , де  $\lim_{x \rightarrow +\infty} f(x) = b$ , або  $\lim_{x \rightarrow -\infty} f(x) = b$ ;

- похилі виду  $y = kx + b$ , де  $k = \lim_{x \rightarrow \pm\infty} \frac{f(x)}{x}$ ,  $b = \lim_{x \rightarrow \pm\infty} (f(x) - kx)$ .

8. Знайти точки перетину графіка функції з осями координат та проміжки знакосталості функції.

9. Використовуючи результати дослідження та обчисливши значення функції в кількох контрольних точках, побудувати схему графіка.

**Приклад 1.** Провести повне дослідження функції і побудувати її графік:

$$f(x) = \frac{1}{x^2 - 5x + 6}.$$

**Розв'язування:**

$$1) \quad x^2 - 5x + 6 = 0 \text{ solve, } x \rightarrow \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

$$D(f(x)) : \begin{cases} x \neq 2, \\ x \neq 3. \end{cases}$$

$$2) \quad f(-x) \rightarrow \frac{1}{(x^2 + 5x + 6)}$$

Функція ні парна, ні непарна.

3) Функція неперіодична.

$$4) \quad \frac{d}{dx} f(x) \rightarrow \frac{-1}{(x^2 - 5x + 6)^2} \cdot (2x - 5)$$

$$\frac{-1}{(x^2 - 5x + 6)^2} \cdot (2x - 5) = 0 \text{ solve, } x \rightarrow \frac{5}{2}$$

$$f'(x) := \frac{-1}{(x^2 - 5x + 6)^2} \cdot (2x - 5)$$

$f_1(-1) = 0.049$  На проміжку  $(-\infty; 2)$  функція є зростаючою.

$f_1(2.3) = 9.07$  На проміжку  $\left(2; \frac{5}{2}\right)$  функція є зростаючою.

$f_1(2.7) = -9.07$  На проміжку  $\left(\frac{5}{2}; 3\right)$  функція є спадною.

$f_1(5) = -0.139$  На проміжку  $(3; +\infty)$  функція є спадною.

$$f(2.5) = -4$$

$$5) \frac{d^2}{dx^2} f(x) \rightarrow \frac{2}{(x^2 - 5x + 6)^3} (2x - 5)^2 - \frac{2}{(x^2 - 5x + 6)^2}$$

$$\left[ \frac{2}{(x^2 - 5x + 6)^3} (2x - 5)^2 - \frac{2}{(x^2 - 5x + 6)^2} \right] = 0 \text{ solve, } x \rightarrow \left( \frac{5}{2} + \frac{1}{6}i\sqrt{3} \right)$$
$$\left( \frac{5}{2} - \frac{1}{6}i\sqrt{3} \right)$$

$$f_2(x) := \frac{2}{(x^2 - 5x + 6)^3} (2x - 5)^2 - \frac{2}{(x^2 - 5x + 6)^2}$$

$f_2(-1) = 0.043$  На проміжку  $(-\infty; 2)$  графік функції опуклий вниз.

$f_2(2.3) = -79.905$   
На проміжку  $(2; 3)$  графік функції опуклий вверх.

$$f_2(2.7) = -79.905$$

$f_2(5) = 0.176$  На проміжку  $(3; +\infty)$  графік функції опуклий вниз.

6)  $\lim_{x \rightarrow 2} f(x) \rightarrow \text{undefined}$   $x = 2$  - вертикальна асимптота

$\lim_{x \rightarrow 3} f(x) \rightarrow \text{undefined}$   $x = 3$  - вертикальна асимптота

$\lim_{x \rightarrow \infty} f(x) \rightarrow 0$   $y = 0$  - горизонтальна асимптота

$\lim_{x \rightarrow \infty} \frac{f(x)}{x} \rightarrow 0$  похилих асимптот немає

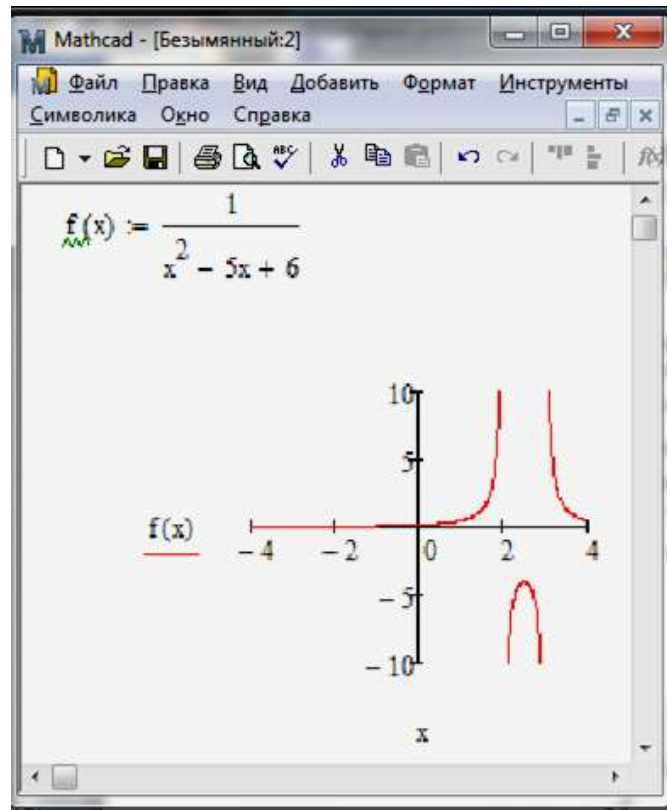


Рисунок 3.4. – Побудова досліджуваного графіка функції

### II. Завдання до лабораторної роботи

1. Самостійна робота з теми «Лінійна алгебра в СКМ».
2. Знайти границю заданої функції і дати її геометричну інтерпретацію:

$$1) \lim_{x \rightarrow 2} \frac{1}{x-2}; \quad 2) \lim_{x \rightarrow 0} \cos \frac{\pi}{x}.$$

3. Обчислити першу похідну функції за означенням і за допомогою табличних похідних, а потім знайти похідні 2-5 порядків та їх значення при  $x = \frac{\pi}{4}$ :  $f(x) = \cos^2 x$  (рис.3.5).

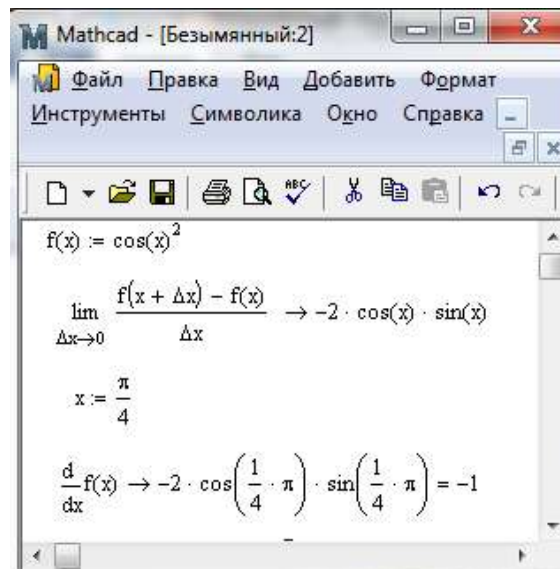


Рисунок 3.5 – Приклад обчислення похідної

4. Знайти визначений інтеграл:  $\int_0^2 \frac{\sqrt{x}}{1+\sqrt{x}} dx$ .

**Примітка. 1.** В системі Mathcad для обчислення визначених інтегралів можна використовувати різні методи. Для цього необхідно викликати контекстно-залежне меню, навівши курсор мишки на знак інтеграла і натиснувши праву клавішу мишки (рис. 3.6).

В системі Mathcad можна демонструвати геометричну сутність визначеного інтеграла, як площу криволінійної трапеції (рис. 3.7).

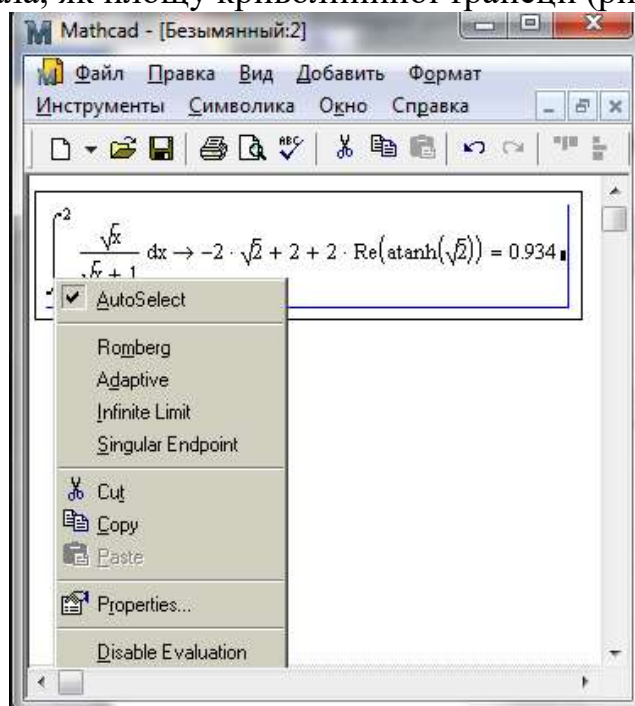


Рисунок 3.6 – Приклад обчислення інтегралу

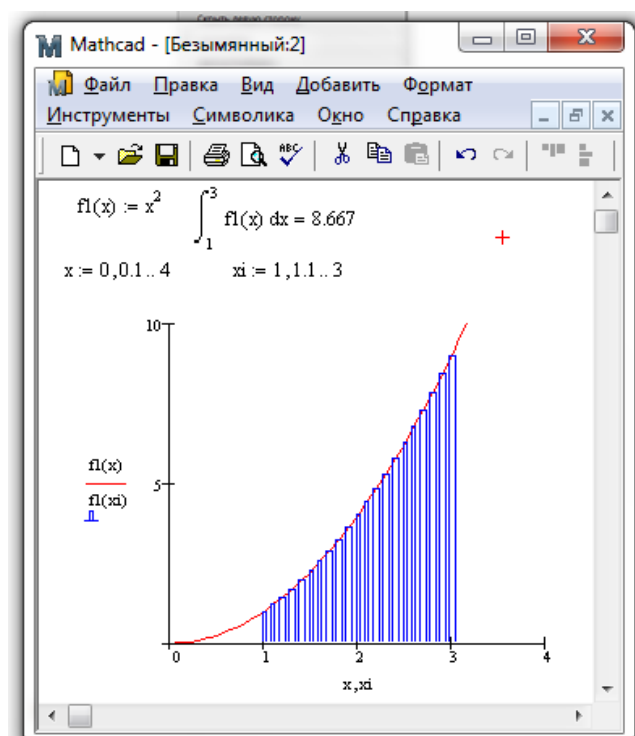


Рисунок 3.7 – Геометрична сутність визначеного інтеграла



6. З'ясувати, які задачі математичного аналізу можна розв'язувати за допомогою пакету Mathcad, і підготувати деякі методичні матеріали щодо цих застосувань.

7. Виконати індивідуальні завдання.

8. Оформити звіт про виконану роботу.

### III. Індивідуальні завдання

Розв'язати завдання спочатку математично, а потім перевірити одержані результати за допомогою СКМ Mathcad:

1. Знайти границі заданих функцій і дати їх геометричну інтерпретацію:

$$1) \lim_{x \rightarrow -2} \frac{2x^2 + 5x - 7}{3x^2 - x - 2};$$

$$2) \lim_{x \rightarrow -1} \frac{2x^3 + 2x^2 + 3x + 3}{x^3 + x^2 + x + 1};$$

$$3) \lim_{x \rightarrow 0} \frac{x^2 - \sqrt{x}}{\sqrt{x} - 1};$$

$$4) \lim_{x \rightarrow 0} \frac{\sqrt{x+1} - (x+1)}{\sqrt{x+1} - 1};$$

$$5) \lim_{x \rightarrow 1} \frac{\sqrt{5-x} - 2}{\sqrt{2-x} - 1};$$

$$6) \lim_{x \rightarrow 0} \frac{\cos 3x - \cos x}{\cos x - 1};$$

$$7) \lim_{x \rightarrow 0} \frac{\cos 3x - 1}{x \operatorname{tg} 2x};$$

$$8) \lim_{x \rightarrow 0} \frac{\operatorname{tg} x - \sin x}{x^3};$$

$$9) \lim_{x \rightarrow \frac{\pi}{2}} \frac{\cos \frac{x}{2} - \sin \frac{x}{2}}{\cos x};$$

$$10) \lim_{x \rightarrow \frac{\pi}{4}} \frac{\sqrt{2} \cos x - 1}{1 - \operatorname{tg}^2 x};$$

$$11) \lim_{x \rightarrow \infty} \left( \frac{2x+3}{2x+1} \right)^{x+1};$$

$$12) \lim_{x \rightarrow 0} (\cos x)^{\frac{1}{\sin^2 x}}.$$

2. Провести повне дослідження функції і побудувати її графік:

$$1) y = \log_{\frac{1}{3}}(|9 - x^2|); \quad 2) y = \frac{1}{x+2} - \frac{1}{x-2};$$

$$3) y = \left( \frac{x+1}{x} \right)^{\frac{4}{5}}; \quad 4) y = \left( \frac{x}{x+1} \right)^5;$$

$$5) y = (1 - 2^x)^2; \quad 6) y = \left( \frac{x-2}{x+2} \right)^2;$$

$$7) y = \frac{x^4}{x^3 - 8}; \quad 8) y = \frac{x^2 - 3}{x};$$

$$9) y = \log_{x^2 - 3x + 2} 2; \quad 10) y = \sqrt{(x^2 - 4)^2} + \sqrt{(x^2 - 2)^2};$$

$$11) y = \left| \frac{x-3}{x-4} \right|; \quad 12) y = \cos x - \ln \cos x.$$

3. Знайти задані невизначені інтеграли, а потім обчислити їх як визначені інтеграли при заданих межах  $a$  і  $b$ :

- 1)  $\int \left( \frac{1}{x\sqrt{x}} - \frac{1}{x^2} \right) dx; a=1; b=4;$
- 2)  $\int \frac{x^2 - 3x + 4}{\sqrt{x}} dx; a=2; b=5;$
- 3)  $\int (\cos x - 2\sqrt[5]{x^3}) dx; a=-3; b=3;$
- 4)  $\int \left( 4\cos x - \frac{5}{\sqrt{9-9x^2}} \right) dx; a=-0.5; b=0.5;$
- 5)  $\int \left( 4^{-x} + \frac{1}{\sqrt{2-2x^2}} \right) dx; a=-0.9; b=0.8;$
- 6)  $\int \cos^5 x \sin x dx; a=-2\pi; b=\pi$
- 7)  $\int \frac{\sin \sqrt{x}}{\sqrt{x}} dx; a=1; b=5;$
- 8)  $\int \frac{\sqrt[3]{x} + 1}{\sqrt{x}} dx; a=10; b=100$
- 9)  $\int \frac{dx}{(1-x^2)\sqrt{1-x^2}}; a=-0.3; b=0.4$
- 10)  $\int x \ln^2 x dx; a=1; b=10;$
- 11)  $\int e^{-x} \sin x dx; a=-5; b=5$
- 12)  $\int \frac{x^5 - 2x^2 + 3}{(x-2)^2} dx; a=-10; b=-3.$


## Лабораторна робота № 4


Тема: «Графічна візуалізація обчислень в пакеті Mathcad».

Питання:

1. Загальна характеристика графічних можливостей систем комп'ютерної математики.
2. Двовимірні графіки.
3. Тривимірні графіки.
4. Робота з анімаційною графікою.

### I. Теоретичний матеріал

В пакеті Mathcad побудова графіків функцій здійснюється за допомогою режиму побудови графіків (кнопка ) математичної панелі.

Для побудови графіка функції однієї змінної потрібно вибрати кнопку  — «X – Y Plot», а потім у середині шаблону задати ім'я функції та аргументу, а також діапазон, в якому відбуватиметься побудова (рис. 4.1).

В пакеті Mathcad можлива побудова декількох графіків в одній системі координат (до 16 функцій). Для цього потрібно вказати відповідні імена функцій через кому (рис. 4.1).

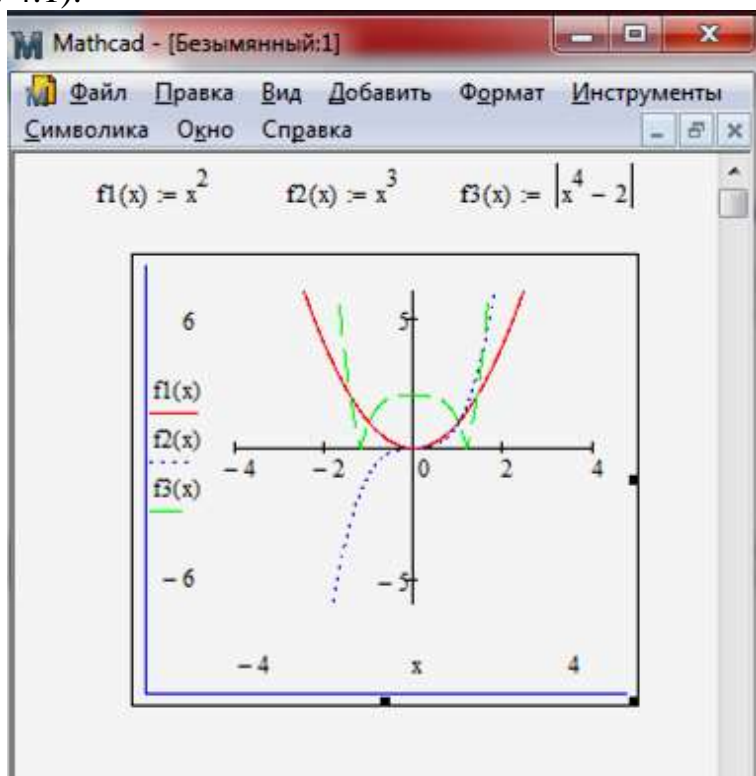


Рисунок 4.1. – Приклад побудови декількох графіків функцій

Для нормального зображення графіків функцій бажано використовувати настройки, які подані на рис. 4.2.

Для задання властивостей графіків функцій використовують режим Формат / Графіки (рис. 4.3).

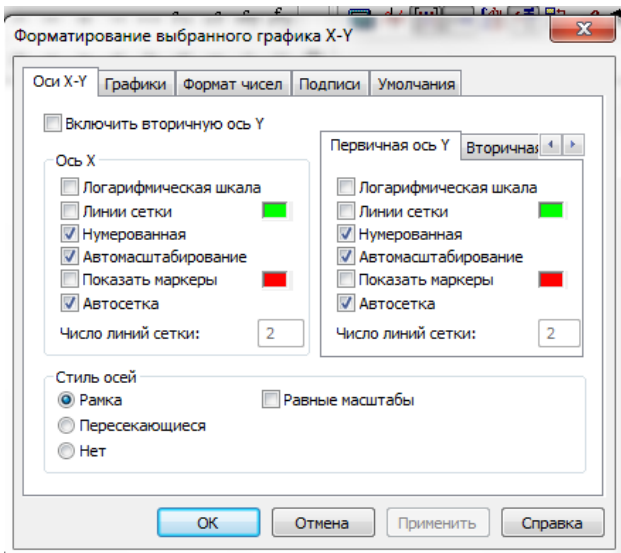


Рисунок 4.2 – Налаштування осей координат

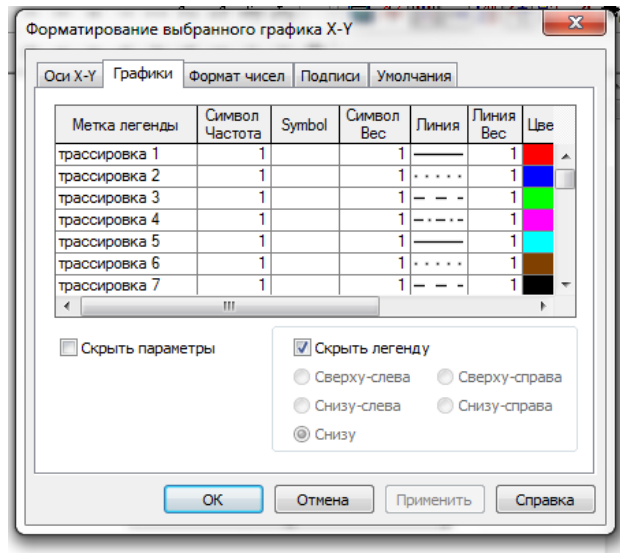



Рисунок 4.3 – Налаштування трассировки

Для задания сложных функций возможно использование средств программирования системы MathCad. Для этого в режиме программирования (Programming) (кнопка ) математической панели выбрать команду Add line, а потом ввести необходимые выражения и операторы. Зокрема, якщо треба задати програму знаходження значень складної функції виду  $f(x) = \begin{cases} 2x, & \text{якщо } x > 0 \\ x^2, & \text{інакше} \end{cases}$ , необхідно

скористатися оператором *if* і у представлений шаблон записати відповідні вирази (рис. 4).

Для задания функции максимума  $f(x) = \max\{2x, x^2\}$  можна скористатися вбудованою функцією  $\max(f_1(x), f_2(x), f_3(x), \dots)$ , де  $f_1(x)$ ,  $f_2(x)$ ,  $f_3(x)$  – деякі функції (рис. 5), або шаблоном, поданим на рис. 5.

У пакеті Mathcad є також вбудована функція  $\min(f_1(x), f_2(x), f_3(x), \dots)$ , де  $f_1(x)$ ,  $f_2(x)$ ,  $f_3(x)$  – деякі функції.

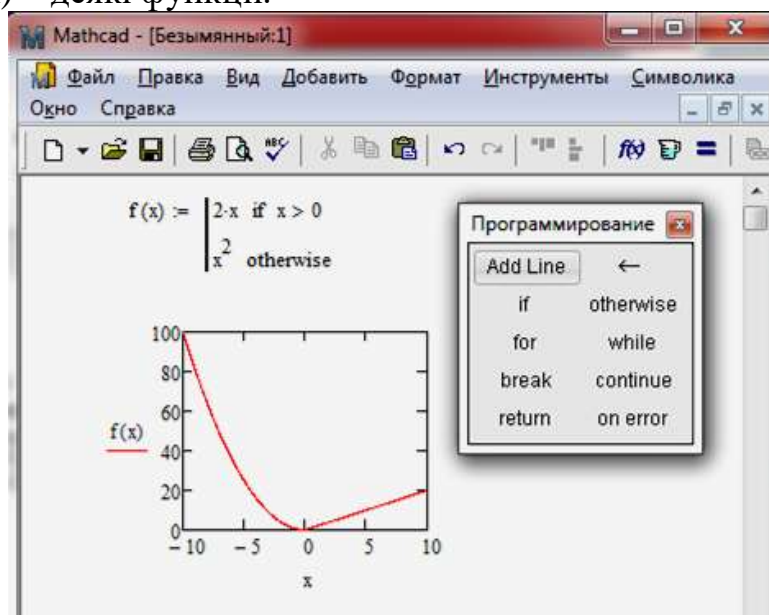


Рисунок 4.4 – Знаходження значень складної функції

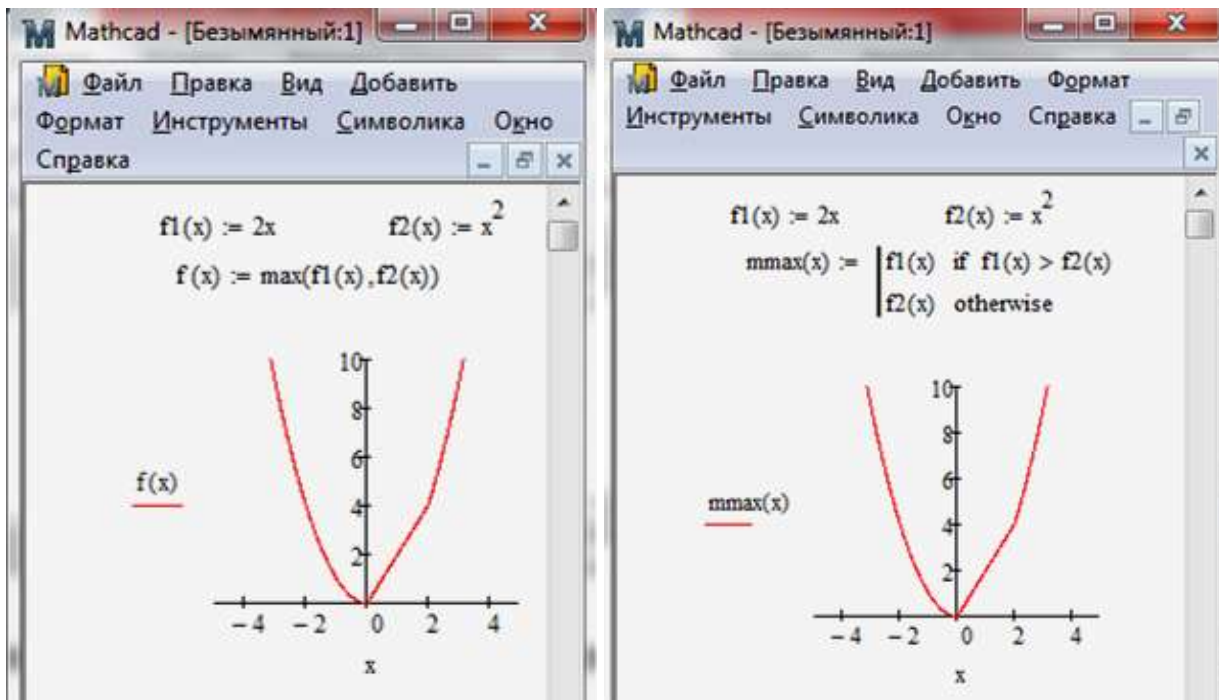


Рисунок 4.5 – Приклад знаходження максимуму функції

В пакеті Mathcad можна будувати графіки функцій, заданих у параметричній формі (рис. 4.6).

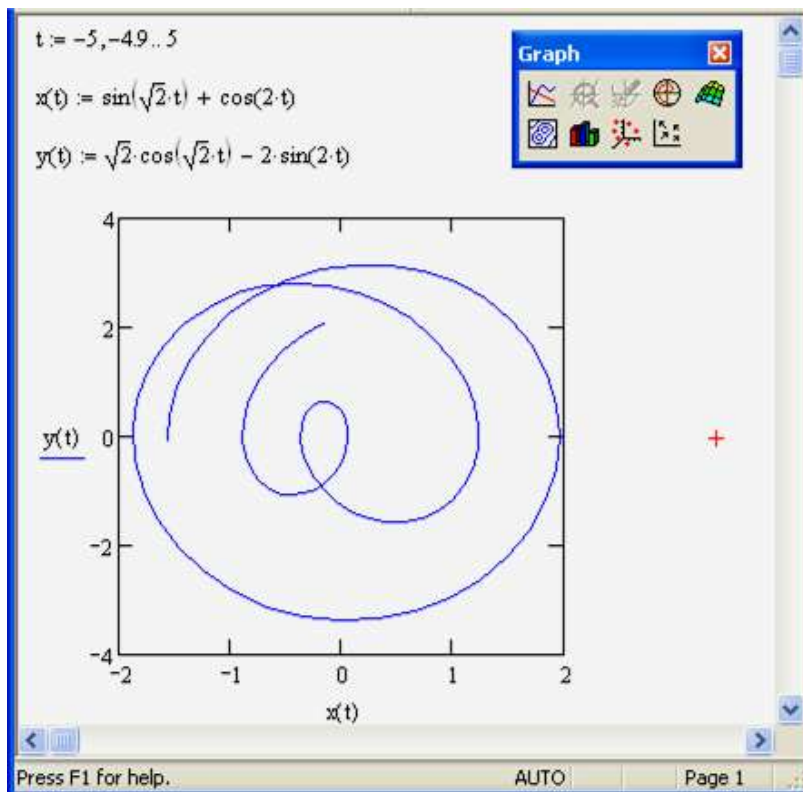



Рисунок 4.6. – Побудова графіка функцій у параметричній формі

Для побудови графіка функції у полярних координатах потрібно вибрати кнопку  — «Polar Plot», а потім у середині шаблону задати ім'я функції та аргументу, а також діапазон, в якому відбуватиметься побудова (рис. 4.7).

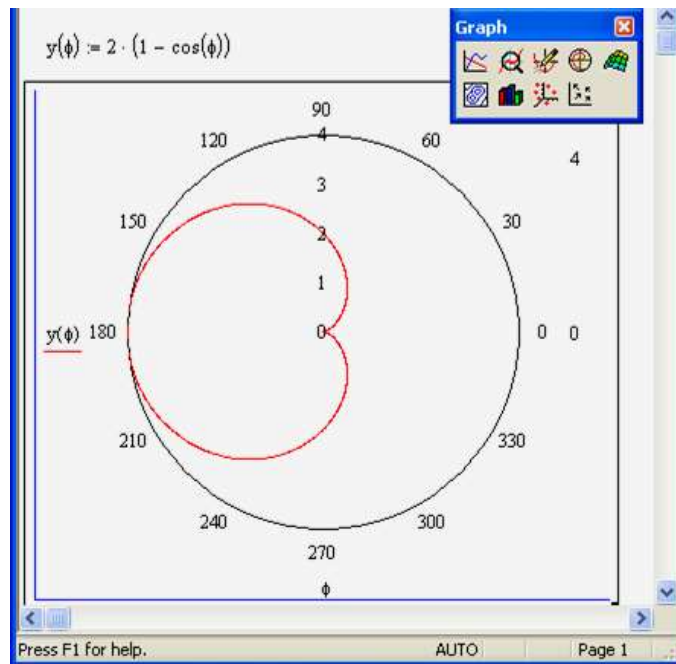



Рисунок 4.7 – Побудова графіка функції у полярних координатах

### 3. Тривимірна графіка

Для побудови графіка функції двох змінних, яка введена у робочому вікні пакету, потрібно вибрати кнопку  — «Surface Plot», а потім у середині шаблону задати ім'я функції без зазначення аргументів (рис. 4.8).

**Зауваження.** В пакеті Mathcad при побудові графіка функції двох змінних не можна, щоб імена цих змінних не були елементами вектора.

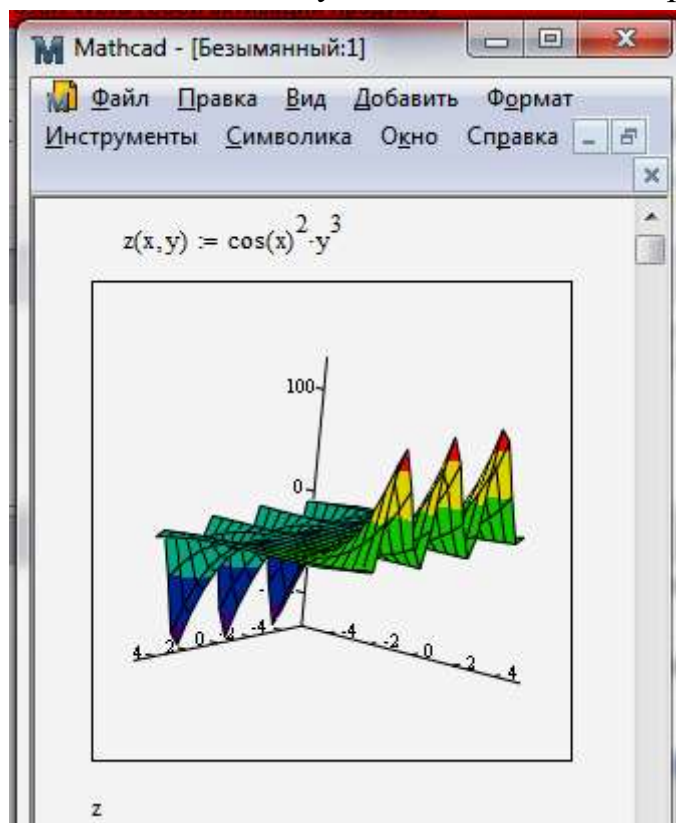


Рисунок 4.8 – Побудова графіка функції двох змінних

В пакеті Mathcad можлива побудова декількох графіків функцій від двох змінних в одній системі координат. Для цього потрібно вказати відповідні імена функцій через кому.

Для налаштування зображення графіків функцій бажано використовувати настройки, які подані на рис. 4.9. Вони викликаються при виділеній області графіка функції після натискання правої клавіші мишки і вибору пункту меню Properties. Зокрема можна зобразити лінії рівня функції (параметр Contour Plot).

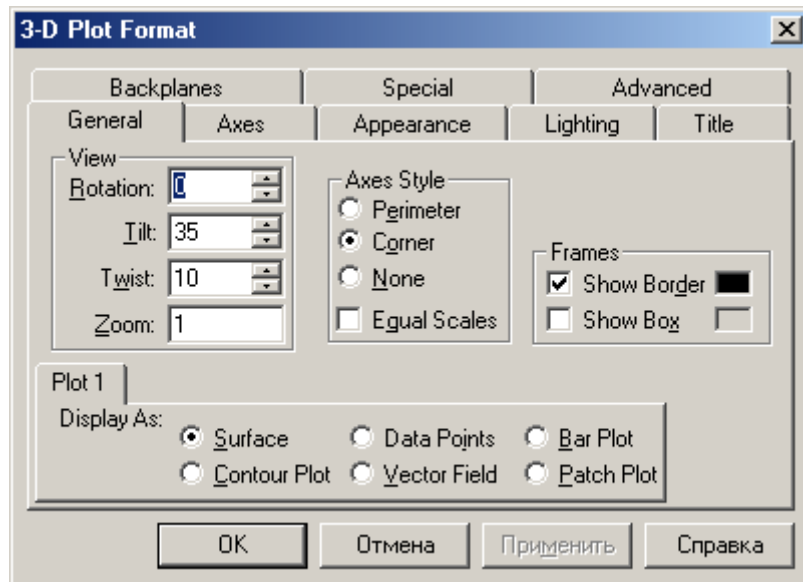


Рисунок 4.9 – Вікно налаштування графіка функцій

#### 4. Робота з анімаційною графікою

Основні можливості для створення анімаційної графіки пакету Mathcad подано на рис. 4.10-4.11.

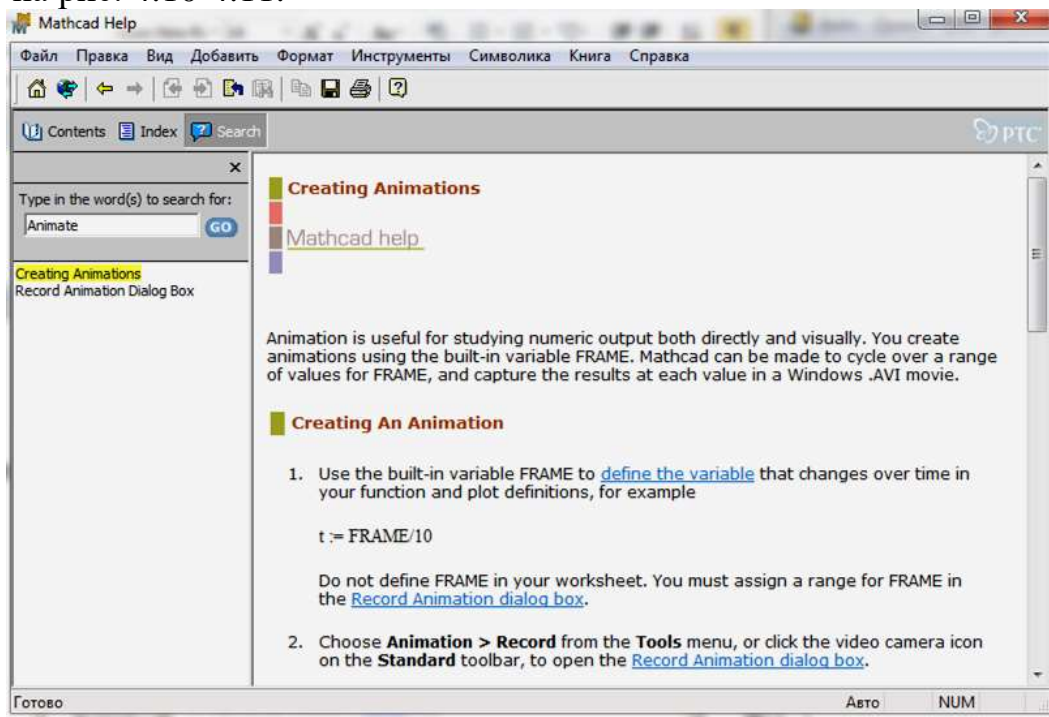


Рисунок 4.10 – Довідка про анімацію в Mathcad

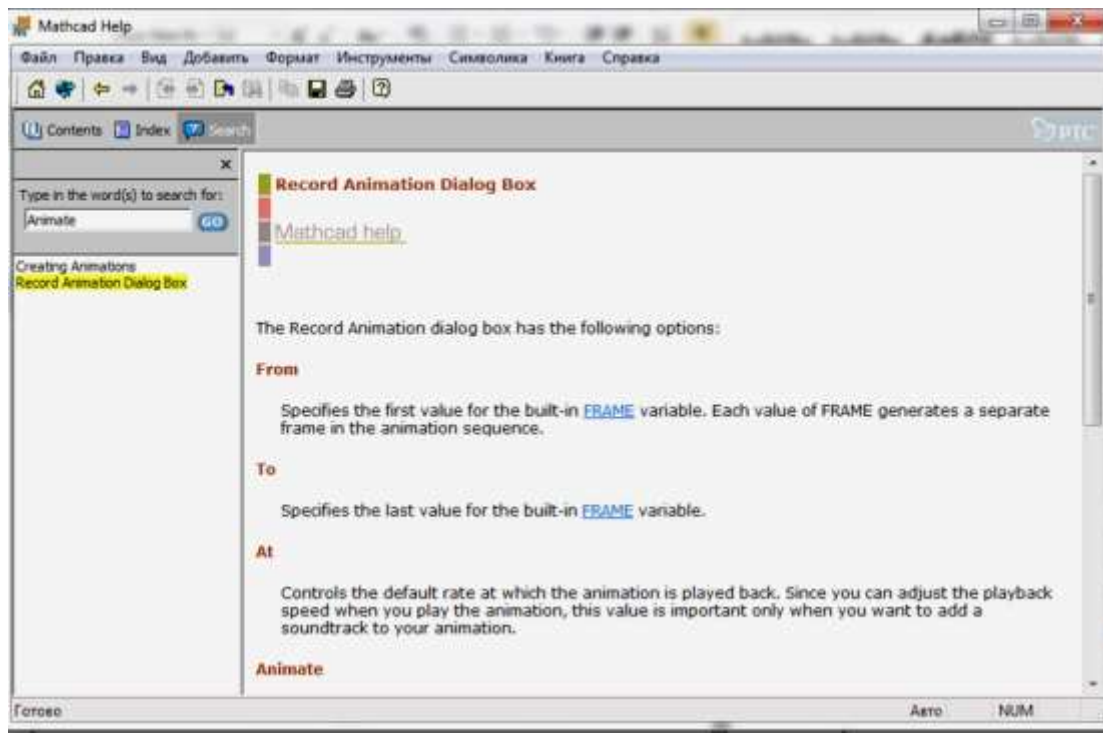


Рисунок 4.11 – Довідка про анімацію в Mathcad

На рис. 4.12-4.14 подано основні етапи створення анімаційного зображення для побудови «квітки», яка задана функцією у полярних координатах, де анімаційний параметр FRAME впливає на зміну радіуса функції і кількості пелюсток «квітки»:

1. Створення функції з параметром анімації FRAME (рис. 4.12).

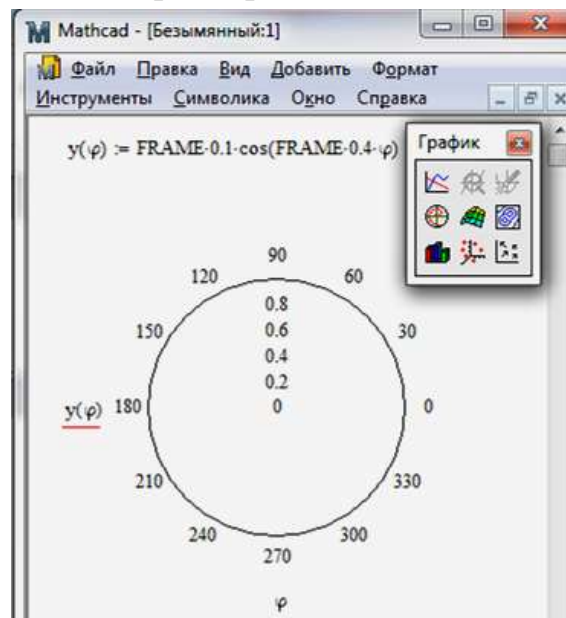


Рисунок 4.12 – Створення функції з параметром анімації FRAME

2. Виклик інструменту для створення анімаційного зображення та його запису за допомогою пункту Інструменти / Анімація (рис. 4.13). При цьому після виклику інструменту Record Animation необхідно мишкою вказати у вікні Worksheet область, де будуть здійснюватися графічні побудови. Потім вказати



початкове (From) і кінцеве (End) значення параметра FRAME, а також кількість кадрів (фреймів), яка буде демонструватися за одну секунду (At).

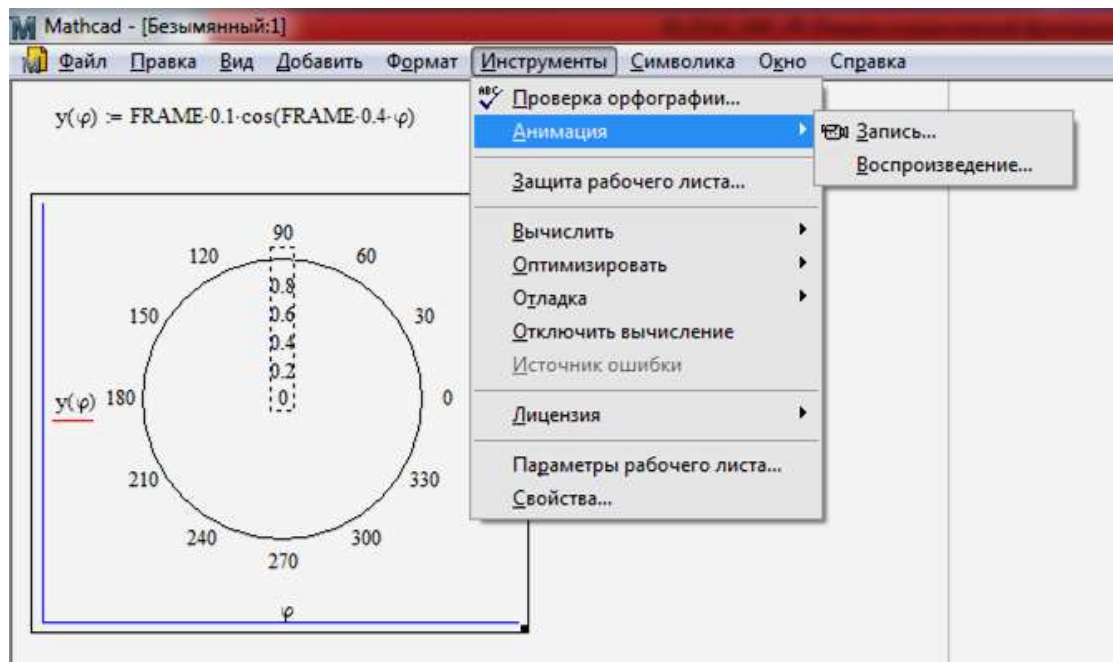


Рисунок 4.13 – Запуск анімації



Рисунок 4.14 – Вікно налаштування анімації

3. Натиснути клавішу Анімувати (рис. 4.14) для генерації анімаційних кадрів. У результаті чого в маленькому віконці буде видно процес побудови зображення. Після цього автоматично викликається програвач Play Animation (рис. 4.15), за допомогою якого, натиснувши кнопку ►, можна відтворити анімаційне зображення. Якщо результат анімації влаштовує, то його можна зберегти на електронному носії з розширенням avi, натиснувши клавішу Зберегти як (рис. 4.14). Збережений файл можна «прокрутити», використовуючи будь-який мультимедійний програвач, що підтримує формат avi.

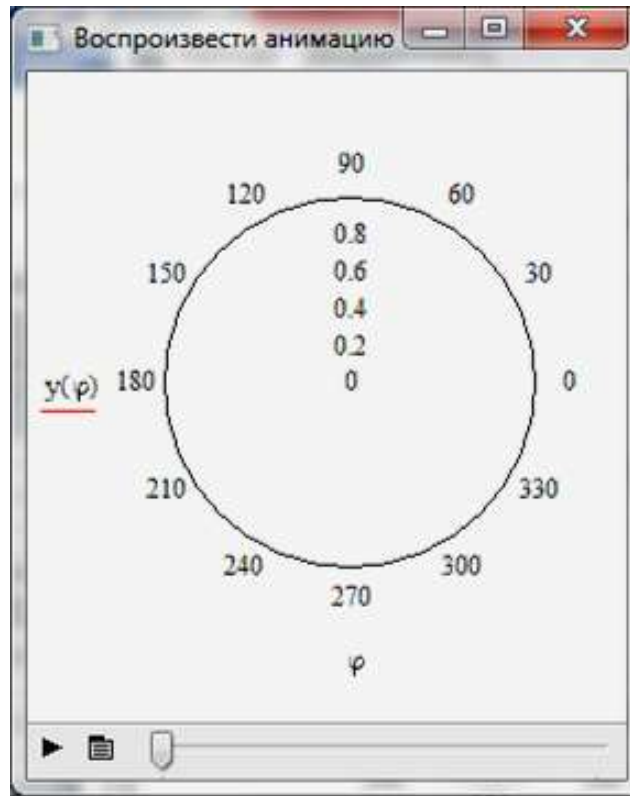


Рисунок 4.15 – Вікно побудови зображення

## II. Завдання до лабораторної роботи

1. Загальна характеристика графічних можливостей СКМ (самостійно).
2. Двовимірна графіка.
3. Підготувати звіт про виконану роботу та надіслати його до СПДН ФІТІС.

## III. Індивідуальні завдання

1. Систему лінійних рівнянь розв'язати аналітичним і графічним способами:

$$1. \begin{cases} 3x_1 + 2x_2 = 5 \\ 2x_1 + 3x_2 = 11 \end{cases}$$

$$2. \begin{cases} 2x_1 + 3x_2 = 3 \\ x_1 + 2x_2 = 3 \end{cases}$$

$$3. \begin{cases} 7x_1 + 2x_2 = 1 \\ -x_1 + 3x_2 = -2 \end{cases}$$

$$4. \begin{cases} x_1 + x_2 = -1 \\ 2x_1 - x_2 = -4 \end{cases}$$

$$5. \begin{cases} x_1 - 2x_2 = 6 \\ 2x_1 - 4x_2 = 20 \end{cases}$$

$$6. \begin{cases} x_1 - 4x_2 = -3 \\ 3x_1 + x_2 = 5 \end{cases}$$

$$7. \begin{cases} 4x_1 - 3x_2 = 9 \\ 2x_1 + 5x_2 = 4 \end{cases}$$

$$8. \begin{cases} 2x_1 + x_2 = 2 \\ x_1 - x_2 = 1 \end{cases}$$

$$9. \begin{cases} x_1 + x_2 = 1 \\ 8x_1 + 3x_2 = 2 \end{cases}$$

$$10. \begin{cases} 3x_1 + 5x_2 = 8 \\ 5x_1 + 7x_2 = 18 \end{cases}$$

2. Графічним і аналітичним способом знайти екстремуми функції:

1)  $y = (1 - 2^x)^2$ ;

2)  $y = \log_{x^2 - 3x + 2} 2$ ;

3)  $y = \left| \frac{x-3}{x-4} \right|$ ;

4)  $y = \left( \frac{x}{x+1} \right)^5$ ;

5)  $y = \sqrt{(x^2 - 4)^2} + \sqrt{(x^2 - 2)^2}$ ;

6)  $y = \left( \frac{x-2}{x+2} \right)^2$ ;

7)  $y = \frac{1}{x+2} - \frac{1}{x-2}$ ;

8)  $y = \cos x - \ln \cos x$ ;

9)  $y = \frac{x^4}{x^3 - 8}$ ;

10)  $y = \log_{\frac{1}{3}} (|9 - x^2|)$ .

4. До функції із завдання 2 ввести параметр FRAME і створити анімаційну демонстрацію зміни графіка в певному діапазоні.

5. Для заданих чудових кривих знайти їх аналітичний вигляд, побудувати графіки і створити анімаційну демонстрацію зміни графіка в певному діапазоні:

- 1) лемніската Бернуллі;
- 2) конхоїда Нікомеда;
- 3) астроїда;
- 4) листок Декарта ;
- 5) кардіоїда;
- 6) спіраль Архімеда;
- 7) логарифмічна спіраль;
- 8) строфоїда;
- 9) гіперболічна спіраль;
- 10) подовжена циклоїда.

## Лабораторна робота № 5

Тема: «Розв'язування задач оптимізації в СКМ Mathcad»

Питання:

1. Умови існування розв'язку екстремальної задачі.
2. Класичні методи знаходження екстремумів функцій однієї змінної.
3. Класичні методи знаходження екстремумів функцій багатьох змінних.
4. Основні засоби пакету Mathcad розв'язування задач оптимізації.
5. Розв'язування задач математичного програмування в пакеті Mathcad.

### I. Теоретичний матеріал

Нехай функція  $f(x)$  визначена в деякому околі точки  $x_0$ . Кажуть, що функція  $f(x)$  **зростає** в точці  $x_0$ , якщо існує таке число  $\delta > 0$ , що  $f(x) < f(x_0)$  для всіх  $x \in ]x_0 - \delta, x_0[$  і  $f(x_0) < f(x)$  для всіх  $x \in ]x_0, x_0 + \delta[$ .

Має місце:

**Теорема 1.** Якщо  $f'(x_0) > 0$ , то функція  $f(x)$  зростає в точці  $x_0$ .

Кажуть, що функція  $f(x)$  **спадає** в точці  $x_0$ , якщо існує таке число  $\delta > 0$ , що  $f(x) > f(x_0)$  для всіх  $x \in ]x_0 - \delta, x_0[$  і  $f(x_0) > f(x)$  для всіх  $x \in ]x_0, x_0 + \delta[$ .

Має місце:

**Теорема 2.** Якщо  $f'(x_0) < 0$ , то функція  $f(x)$  спадає в точці  $x_0$ .

**Означення 1.** Точка  $x_0$  називається **точкою локального мінімуму (строго локального мінімуму)** функції  $f(x)$ , якщо існує таке число  $\delta > 0$ , що  $f(x_0) \leq f(x)$  ( $f(x_0) < f(x)$ ) для всіх  $x \in ]x_0 - \delta; x_0 + \delta[$ .

**Означення 2.** Точка  $x_0$  називається **точкою локального максимуму (строго локального максимуму)** функції  $f(x)$ , якщо існує таке число  $\delta > 0$ , що  $f(x_0) \geq f(x)$  ( $f(x_0) > f(x)$ ) для всіх  $x \in ]x_0 - \delta; x_0 + \delta[$ .

З розглянутих означень видно, що наявність екстремуму функції  $f(x)$  в точці  $x_0$  залежить від значень цієї функції в цій точці і поблизу неї. Такі властивості функції називають *локальними* на відміну від глобальних властивостей, що визначаються значеннями функції на всій множині  $X$  (наприклад, властивість функції спадати на проміжку).

Слід зауважити, що кінцеві точки замкненого проміжка, який є областю визначення деякої функції не можуть бути точками екстремуму згідно наведених означень, оскільки не існує числа  $\delta > 0$  такого, що всі точки  $x \in ]x_0 - \delta; x_0 + \delta[$  належать області визначення розглядуваної функції, де  $x_0$  - кінцева точка проміжка.

#### 1. Умови існування розв'язку екстремальної задачі

Перш ніж приступати до розв'язування вже формалізованої екстремальної задачі, необхідно визначити *умови існування розв'язку* цієї задачі при заданих обмеженнях.

Розглянемо питання про існування розв'язку екстремальної задачі для задачі мінімізації функції однієї змінної  $f(x)$  на деякій множині  $X \subseteq R^1$ :

$$f(x) \rightarrow \min, x \in X. \quad (1)$$

Припустимо, що  $X$  непорожня множина і містить більше, ніж одну точку, оскільки тільки в цьому випадку задача (1) має сенс.

Якщо множина  $X$  скінченна, то розв'язок задачі існує і може бути знайдений методом послідовного перебору всіх точок множини  $X$  і визначення серед них точки  $x^*$ , яка задовольняє умову:  $f(x^*) \leq f(x)$  для будь-яких  $x \in X$ .

Якщо множина  $X$  нескінченна, то задача (1) може не мати розв'язку, тобто  $\operatorname{Arg} \min_{x \in X} f(x) = \emptyset$ .

Нехай  $f(x)$  – монотонно спадна функція на необмеженій множині  $X = \{x \in R^1 \mid x \geq a\}$ , тобто  $f(x_1) > f(x_2)$  для будь-яких  $x_1, x_2$  з  $X$  таких, що  $x_1 < x_2$  (рис. 5.1 а), тоді точки мінімуму цієї функції на множині  $X$  не існує.

Нехай  $f(x)$  – монотонно спадна функція на незамкненій множині  $X = \{x \in R^1 \mid x \in [a; b)\}$  (рис. 5.1 б). Оскільки монотонно спадна функція не досягає своєї нижньої межі на незамкненій множині  $[a; b)$ , то не існує точки  $x^* \in X$ , яка задовольняє умову:  $f(x^*) \leq f(x)$  для будь-яких  $x \in X$ .

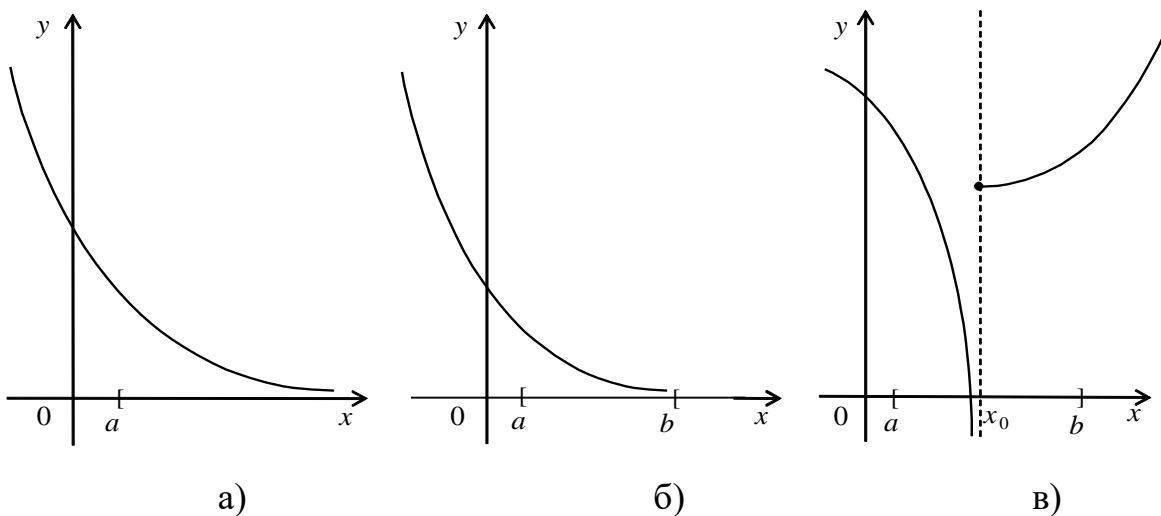


Рисунок 5.1 – Умови існування розв'язку екстремальної задачі

З наведених прикладів видно, що якщо множина  $X$  не є замкненою і обмеженою, то задача (1) може не мати розв'язку.

Нехай функція  $f(x)$  на замкненій і обмеженій множині  $X = \{x \in R^1 \mid x \in [a; b]\}$ , має точку розриву другого роду  $x_0 \in X$ , причому  $f(x) \rightarrow -\infty$  при  $x \rightarrow x_0 - 0$  (рис. 5.1 в) чи  $x \rightarrow x_0 + 0$ . У даному випадку точки мінімуму функції  $f(x)$  на множині  $X$  також не існує.

Отже, задача мінімізації виду (1), як і задача максимізації, не завжди має розв'язок, але відома теорема Вейерштрасса в багатьох випадках гарантує існування розв'язку цієї задачі.

**Теорема 3 (Вейерштрасса).** Якщо функція  $f(x)$  визначена і неперервна на замкненому проміжку  $[a; b]$ , то вона досягає на цьому проміжку своїх найменшого і найбільшого значень.

## 2. Класичний метод знаходження екстремумів функції однієї змінної

Класичними називають такі методи знаходження точок екстремуму функції, які ґрунтуються на диференціальному численні. Нагадаємо суть класичного методу відшукування екстремумів функції однієї змінної.

При вивченні екстремальних задач будь-якого класу важливе місце займає питання про **умови оптимальності** або **умови екстремуму**. Вони складають основу якісних методів теорії оптимізації, які вивчають властивості екстремальних задач, використовуються при побудові і обґрунтуванні чисельних методів розв'язування цих задач, а також надають можливість у деяких випадках явно розв'язати екстремальну задачу.

Розрізняють необхідні і достатні умови екстремуму. **Необхідними** умовами екстремуму є умови, яким повинна задовольняти точка, яка є розв'язком екстремальної задачі. **Достатніми** умовами екстремуму є умови, з яких випливає, що знайдена точка є точкою екстремуму певного типу (точкою мінімуму чи максимуму).

Розглянемо необхідні і достатні умови екстремуму для функції однієї змінної. Має місце:

**Теорема 4 (Ферма).** Нехай функція  $f(x)$  визначена на деякому проміжку  $[a, b]$ , який містить точку  $x_0$  ( $a < x_0 < b$ ), і диференційована в цій точці. Тоді, якщо точка  $x_0$  є точкою локального екстремуму (мінімуму або максимуму) цієї функції, то  $f'(x_0) = 0$ .

Точки  $x \in ]a, b[$ , для яких  $f'(x) = 0$ , називаються **стаціонарними**.

Стаціонарні точки разом з кінцевими називаються **критичними** («підозрілими» на екстремум).

Співвідношення  $f'(x_0) = 0$  є лише **необхідною умовою** для того, щоб точка  $x_0$  була точкою екстремуму функції  $f(x)$ . Воно дає можливість відібрати стаціонарні точки, але не дає підстав стверджувати, що в цих точках функція справді має екстремум. Для цього потрібно робити додаткові дослідження.

**Теорема 5 (достатні умови строгого екстремуму).**

1. Якщо функція  $f(x)$  неперервна в точці  $x_0$ , а  $f'(x) < 0$  в інтервалі  $]x_0 - \delta; x_0[$  і  $f'(x) > 0$  в інтервалі  $]x_0; x_0 + \delta[$  для деякого  $\delta > 0$ , то точка  $x_0$  є точкою мінімуму функції  $f(x)$ .

2. Якщо функція  $f(x)$  неперервна в точці  $x_0$ , а  $f'(x) > 0$  в інтервалі  $]x_0 - \delta; x_0[$  і  $f'(x) < 0$  в інтервалі  $]x_0; x_0 + \delta[$  для деякого  $\delta > 0$ , то точка  $x_0$  є точкою максимуму функції  $f(x)$ .

Сформулюємо також достатні умови строгого екстремуму функції в термінах значень похідних вищих порядків в точці  $x_0$ .

**Теорема 6.** Нехай функція  $f(x)$  визначена в деякій точці  $x_0$  і нехай в цій точці функція  $f(x)$  має похідні до порядку  $n$  включно, при цьому  $f^{(i)}(x_0) = 0$ ,  $i = 1, 2, \dots, n-1$ ,  $f^{(n)}(x_0) \neq 0$ .

Тоді, якщо  $n$  - парне число, то функція  $f(x)$  має в точці  $x_0$  строгий екстремум, а саме: **мінімум** при  $f^{(n)}(x_0) > 0$  і **максимум** при  $f^{(n)}(x_0) < 0$ .

Якщо  $n$  - непарне число, то функція не має в точці  $x_0$  екстремуму, ця точка є точкою зростання при  $f^{(n)}(x_0) > 0$  і спадання при  $f^{(n)}(x_0) < 0$ .

**Наслідок.** Якщо  $f'(x_0) = 0$ , а  $f''(x_0) \neq 0$ , то при  $f''(x_0) > 0$  точка  $x_0$  є точкою **строого мінімуму**, а при  $f''(x_0) < 0$  - точкою **строого максимуму** функції  $f(x)$ .

Враховуючи теореми Вейєрштраса і Ферма, а також достатні умови екстремуму, можна дати загальне правило відшукування розв'язків одновимірних задач оптимізації.

### **Правило I.**

Щоб знайти найменше (найбільше) значення функції  $f(x)$ , яка визначена і неперервна на відрізку  $[a, b]$ , диференційована в інтервалі  $]a, b[$ , за винятком, можливо, скінченного числа точок, і має скінченне число точок, в яких  $f'(x) = 0$  (якщо такі точки існують), потрібно:

- 1) знайти всі стаціонарні точки функції  $f(x)$ , тобто розв'язати рівняння  $f'(x) = 0$ ;
- 2) за допомогою достатніх умов екстремуму визначити, які з стаціонарних точок (якщо вони існують) є точками локального мінімуму (максимуму);
- 3) знайти значення функції  $f(x)$  у точках локального мінімуму (максимуму) (якщо такі точки існують);
- 4) визначити точки, в яких функція  $f(x)$  не має похідної, і знайти в цих точках (якщо вони існують) значення функції;
- 5) знайти значення функції  $f(x)$  в кінцевих точках проміжка  $[a, b]$ :  $f(a), f(b)$ ;
- 6) серед усіх так знайдених значень функції  $f(x)$  вибрати найменше (найбільше).

### **3. Класичний метод знаходження екстремумів функції багатьох змінних**

Сформулюємо спочатку умови існування екстремумів для функції багатьох змінних в  $n$ -вимірному евклідовому просторі. Для цього нагадаємо декілька важливих понять і означень.

**Означення 3.** Множина впорядкованих наборів з  $n$  дійсних чисел, тобто множина  $n$ -вимірних векторів (точок)  $x = (x_1, x_2, \dots, x_n)$  з дійсними координатами  $x_i (i = \overline{1, n})$ , в якій відстань між двома точками  $x = (x_1, x_2, \dots, x_n)$  і  $y = (y_1, y_2, \dots, y_n)$  визначається за формулою

$$\rho(x, y) = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 + \dots + (y_n - x_n)^2},$$

називається  $n$ -вимірним *евклідовим простором* і позначається  $R^n$ .

**Означення 4.** Нехай точка  $O \in R^n$  і  $r$  – деяке додатне число. Множина точок  $\bar{S}(O, r) = \{x \in R^n \mid \rho(x, O) \leq r\}$  називається  $n$ -вимірною *замкненою кулею* радіуса  $r$  з центром в точці  $O$ , а множина точок  $S(O, r) = \{x \in R^n \mid \rho(x, O) < r\}$ , називається  $n$ -вимірною *відкритою кулею* радіуса  $r$  з центром в точці  $O$ .

Відкрита куля радіуса  $\varepsilon > 0$  з центром в точці  $O$  називається  $\varepsilon$ -*околом* точки  $O$ .

Нехай  $X$  – деяка множина точок простору  $R^n$ .

**Означення 5.** Точка  $x$  називається *внутрішньою точкою* множини  $X$ , якщо існує  $\varepsilon$ -окіл точки  $x$ , який повністю лежить у множині  $X$ .

Множина всіх внутрішніх точок множини  $X$  утворює *внутрішність* цієї множини і позначається  $\text{int } X$ :  $\text{int } X = \{x \in R^n \mid S(x, \varepsilon) \subset X \quad \forall \varepsilon > 0\}$ .

**Означення 6.** Множина  $X$  називається *відкритою*, якщо всі її точки внутрішні, тобто якщо  $X = \text{int } X$ .

**Означення 7.** Точка  $x$  називається *граничною точкою* множини  $X$ , якщо будь-який  $\varepsilon$ -окіл точки  $x$  містить нескінченну множину точок з множини  $X$ .

Множина всіх граничних точок множини  $X$  утворює *замикання* цієї множини і позначається  $\bar{X}$ .

**Означення 8.** Множина  $X$  називається *замкненою*, якщо вона містить всі свої граничні точки, тобто якщо  $X = \bar{X}$ .

**Означення 9.** Множина  $X$  називається *обмеженою*, якщо існує замкнена куля скінченного радіуса, що містить всі точки множини  $X$ .

Функцію від  $n$  змінних будемо записувати так  $y = f(x) = f(x_1, x_2, \dots, x_n)$ .

**Теорема 7 (Вейєрштрасса).** Якщо функція  $f(x)$  визначена і неперервна на обмеженій і замкненій множині  $X$  з простору  $R^n$ , то вона досягає на цій множині своїх найменшого і найбільшого значень.

Нехай функція  $y = f(x) = f(x_1, x_2, \dots, x_n)$  визначена в деякому околі точки  $x^{(0)} = (x_1^0, x_2^0, \dots, x_n^0) \in R^n$ .

**Теорема 8 (необхідна умова екстремуму).** Якщо функція  $y = f(x_1, x_2, \dots, x_n)$  в точці  $x^{(0)} = (x_1^0, x_2^0, \dots, x_n^0)$  має екстремум і скінченні частинні похідні першого порядку за аргументами  $x_i (i = \overline{1, n})$ , то

$$\frac{\partial f(x^{(0)})}{\partial x_i} = 0, i = \overline{1, n}.$$

**Наслідок.** Якщо функція  $y = f(x)$  має в точці  $x^{(0)}$  екстремум і диференційована в цій точці, то

$$dy|_{x=x^{(0)}} = \frac{\partial f(x^{(0)})}{\partial x_1} dx_1 + \dots + \frac{\partial f(x^{(0)})}{\partial x_n} dx_n = 0$$

(при будь-яких значеннях диференціалів незалежних змінних  $dx_1, dx_2, \dots, dx_n$ ).



Точки, в яких перший диференціал функції дорівнює нулеві, називають **стаціонарними** або **точками можливого екстремуму** цієї функції.

Щоб знайти точки можливого екстремуму функції  $y = f(x_1, x_2, \dots, x_n)$ , треба розв'язати систему рівнянь

$$\frac{\partial f(x_1, \dots, x_n)}{\partial x_i} = 0, i = \overline{1, n}.$$

Для формулювання достатніх умов екстремуму нагадаємо деякі факти з лінійної алгебри.

Функція виду

$$Q(x_1, \dots, x_n) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j,$$

де  $a_{ij}$  - числа з  $R^1$ ,  $a_{ij} = a_{ji}$  ( $i, j = \overline{1, n}$ ), називається **квадратичною формою** від змінних  $x_1, x_2, \dots, x_n$ . Числа  $a_{ij} \in R^1$  називаються **коефіцієнтами квадратичної форми**, а складена з цих коефіцієнтів симетрична матриця

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = (a_{ij})_{i,j=1}^n$$

називається **матрицею квадратичної форми**  $Q(x_1, \dots, x_n)$ .

Визначники

$$\delta_1 = a_{11}, \delta_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \dots, \delta_n = \begin{vmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{vmatrix}$$

називаються **кутовими мінорами** матриці  $A$ .

В матричному поданні квадратична форма має вигляд  $Q(x_1, \dots, x_n) = \langle Ax, x \rangle$ .

Квадратична форма  $Q(x_1, \dots, x_n)$  називається **додатно визначеною**, якщо вона набуває додатних значень, тобто

$$\langle Ax, x \rangle > 0,$$

для будь-яких значень змінних  $x_1, x_2, \dots, x_n$ , які одночасно не дорівнюють нулеві. Зазначимо, що  $Q(0, \dots, 0) = 0$ .

Квадратична форма  $Q(x_1, \dots, x_n)$  називається **від'ємно визначеною**, якщо вона набуває від'ємних значень для будь-яких значень змінних  $x_1, x_2, \dots, x_n$ , які одночасно не дорівнюють нулеві.

Наприклад,  $Q(x_1, x_2) = x_1^2 + 4x_2^2$  - додатно визначена квадратична форма, оскільки  $Q(x_1, x_2) > 0$  в усіх точках  $(x_1, x_2)$ , крім точки  $(0, 0)$ .

Квадратична форма називається **знаковизначеною**, якщо вона є або додатно визначеною, або від'ємно визначеною.

Квадратична форма  $Q(x_1, \dots, x_n)$  називається **невід'ємно (недодатно) визначеною**, якщо вона набуває невід'ємних (недодатних) значень для будь-яких

значень змінних  $x_1, x_2, \dots, x_n$ , але при цьому значення нуль набуває не тільки при  $x_1 = x_2 = \dots = x_n = 0$ .

Квадратична форма називається **квазізнаковизначеною**, якщо вона є або невід'ємно визначеною, або недодатно визначеною.

Наприклад,  $Q(x_1, x_2) = x_1^2 + 2x_1x_2 + x_2^2$  - квазізнако-визначена (невід'ємно визначена) квадратична форма, оскільки  $Q(x_1, x_2) = (x_1 + x_2)^2 \geq 0$  в усіх точках  $(x_1, x_2)$ , але  $Q(x_1, x_2) = 0$  не тільки в точці  $(0, 0)$ , а й у всіх точках прямої  $x_1 = -x_2$ .

Квадратична форма  $Q(x_1, \dots, x_n)$  називається **знакозмінною**, якщо вона набуває як додатних, так і від'ємних значень.

Наприклад,  $Q(x_1, x_2) = x_1^2 - 2x_1x_2 - x_2^2$  - знакозмінна квадратична форма, оскільки вона набуває як додатних, так і від'ємних значень:  $Q(1, 0) = 1 > 0, Q(0, 1) = -1 < 0$ .

Має місце **критерій Сільвестра**:

1) для того щоб квадратична форма  $Q(x_1, \dots, x_n)$  була додатно визначеною, необхідно і достатньо, щоб всі кутові мінори її матриці були додатні:

$$\delta_1 > 0, \delta_2 > 0, \dots, \delta_n > 0;$$

2) для того щоб квадратична форма  $Q(x_1, \dots, x_n)$  була від'ємно визначеною, необхідно і достатньо, щоб знаки кутових мінорів її матриці чергувались таким чином:

$$\delta_1 < 0, \delta_2 > 0, \delta_3 < 0, \delta_4 > 0, \dots$$

В математичному програмуванні часто використовуються відповідні поняття знаковизначеності для симетричних матриць. Симетрична матриця

$$A = (a_{ij})_{i,j=1}^n, a_{ij} = a_{ji} (i, j = \overline{1, n})$$

називається **додатно визначеною**, якщо квадратична форма

$$\langle Ax, x \rangle = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j > 0$$

для будь-яких значень змінних  $x_1, x_2, \dots, x_n$ , які одночасно не дорівнюють нулеві. Відповідно формулюється і критерій Сільвестра.

Другий диференціал функції  $y = f(x_1, x_2, \dots, x_n)$ , де  $x_1, x_2, \dots, x_n$  - незалежні змінні, в точці  $x^{(0)}$  можна подати у вигляді

$$d^2 y|_{x=x^{(0)}} = \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 f(x^{(0)})}{\partial x_i \partial x_j} dx_i dx_j.$$

Цей вираз показує, що другий диференціал функції  $y = f(x_1, x_2, \dots, x_n)$  в точці  $x^{(0)}$  є квадратичною формою від змінних  $dx_1, \dots, dx_n$ , де частинні похідні другого порядку  $\frac{\partial^2 f(x^{(0)})}{\partial x_i \partial x_j}$  (коефіцієнти квадратичної форми) задовольняють умову

$$\frac{\partial^2 f(x^{(0)})}{\partial x_i \partial x_j} = \frac{\partial^2 f(x^{(0)})}{\partial x_j \partial x_i}, i, j = \overline{1, n}, i \neq j. \quad (2)$$

Матриця других частинних похідних функції  $y = f(x_1, x_2, \dots, x_n)$  в точці  $x^{(0)} \in R^n$

$$f''(x^{(0)}) = \left( \frac{\partial^2 f(x^{(0)})}{\partial x_i \partial x_j} \right)_{i,j=1}^n$$

називається **матрицею Гессе** або **гессіаном**.

Зауважимо, що умова (2) виконується коли мішані частинні похідні другого порядку, як функції від  $x_i (i = \overline{1, n})$ , є неперервними в точці  $x^{(0)}$ .

**Теорема 9 (достатні умови екстремуму).** Нехай функція  $y = f(x_1, x_2, \dots, x_n)$  визначена, неперервна і має неперервні частинні похідні першого і другого порядку в околі деякої стаціонарної точки  $x^{(0)}$  даної функції. Тоді якщо другий диференціал  $d^2 y|_{x=x^{(0)}}$  є додатно (від'ємно) визначеною квадратичною формою від змінних  $dx_1, \dots, dx_n$ , то функція  $y = f(x)$  має в точці  $x^{(0)}$  локальний мінімум (максимум). Якщо ж  $d^2 y|_{x=x^{(0)}}$  є знакозмінною квадратичною формою, то в точці  $x^{(0)}$  функція  $y = f(x)$  не має локального екстремуму.

**Зауваження.** Якщо  $dy|_{x=x^{(0)}} = 0$ , а  $d^2 y|_{x=x^{(0)}}$  є квазізнаковизначеною квадратичною формою, то функція  $y = f(x)$  може мати в точці  $x^{(0)}$  локальний екстремум, і може не мати. В цьому випадку треба додатково досліджувати поведінку функції в околі точки  $x^{(0)}$ , користуючись означенням точок екстремуму або властивостями похідних більш високих порядків. Наприклад, легко перевірити, що для кожної з функцій  $y = x_1^4 + x_2^4$  і  $y = x_1^3 x_2^3$  в точці  $x^{(0)} = (0, 0)$  виконуються умови  $dy|_{x=x^{(0)}} = 0$ ,  $d^2 y|_{x=x^{(0)}} = 0$ , тобто точка  $x^{(0)}$  - стаціонарна для цих функцій і їх другий диференціал є квазізнаковизначеною квадратичною формою. Але при цьому перша функція має в точці  $x^{(0)} = (0, 0)$  локальний мінімум (оскільки  $y = x_1^4 + x_2^4 \geq 0 \forall x \in R^2$ ), а друга функція в цій точці не має екстремуму, оскільки  $y = x_1^3 x_2^3$  для деяких точок з околу точки  $x^{(0)}$  має значення більші від нуля, в деяких - менші.

Розглянемо окремо достатні умови екстремуму для функції двох змінних.

Нехай функція  $y = f(x_1, x_2)$  визначена, неперервна і має неперервні частинні похідні першого і другого порядку в околі деякої стаціонарної точки  $x^{(0)} = (x_1^0, x_2^0)$  даної функції.

Введемо позначення:

$$a_{11} = \frac{\partial^2 f(x^{(0)})}{\partial x_1^2}, \quad a_{12} = \frac{\partial^2 f(x^{(0)})}{\partial x_1 \partial x_2},$$

$$a_{21} = \frac{\partial^2 f(x^{(0)})}{\partial x_2 \partial x_1}, \quad a_{22} = \frac{\partial^2 f(x^{(0)})}{\partial x_2^2},$$

при цьому, згідно припущення,  $a_{12} = a_{21}$ .

З теореми 6 і критерія Сільвестра випливає таке твердження:

1) якщо  $\delta_2 = a_{11} a_{22} - a_{12}^2 > 0$ , то в точці  $x^{(0)}$  функція  $y = f(x_1, x_2)$  має локальний екстремум: мінімум при  $\delta_1 = a_{11} > 0$  і максимум при  $\delta_1 = a_{11} < 0$ ;

2) якщо  $\delta_2 < 0$ , то в точці  $x^{(0)}$  функція  $y = f(x_1, x_2)$  не має екстремуму;

3) якщо  $\delta_2 = 0$ , то в точці  $x^{(0)}$  функція  $y = f(x_1, x_2)$  може мати локальний екстремум, а може й не мати його.

Сформулюємо загальне правило відшукування розв'язків багатовимірних задач безумовної оптимізації.

### **Правило II.**

Нехай функція  $y = f(x_1, x_2, \dots, x_n)$  визначена, неперервна і диференційовна в усіх точках простору  $R^n$ .

Щоб знайти глобальний мінімум (максимум) функції  $y = f(x_1, x_2, \dots, x_n)$  на всьому просторі  $R^n$ , потрібно:

1) знайти всі стаціонарні точки функції  $y = f(x_1, x_2, \dots, x_n)$ , тобто розв'язати систему рівнянь

$$\frac{\partial f(x_1, \dots, x_n)}{\partial x_i} = 0, i = \overline{1, n}; \quad (3)$$

2) якщо система (3) не має розв'язків, то функція  $y = f(x_1, x_2, \dots, x_n)$  не має екстремумів; якщо система (3) має розв'язки, то провести додаткове дослідження і визначити, які з стаціонарних точок є точками локального мінімуму (максимуму); при цьому, якщо цільова функція має неперевні частинні похідні першого і другого порядку в околі стаціонарної точки, то для такого дослідження можна використати достатні умови екстремуму;

3) знайти значення функції  $y = f(x_1, x_2, \dots, x_n)$  у точках локального мінімуму (максимуму) (якщо такі точки існують);

4) серед усіх знайдених значень функції  $y = f(x_1, x_2, \dots, x_n)$  вибрати найменше (найбільше).

При розв'язуванні задач умовної оптимізації, коли допустима множина  $X$  є підмножиною простору  $R^n$ , знайти точки екстремуму значно складніше навіть коли множина  $X$  обмежена і замкнена.

Формально загальне правило розв'язування таких задач можна одержати з правила II, замінивши пункт 4 на такий:

- знайти найменше (найбільше) значення цільової функції на межі множини  $X$ , далі знайти значення, яких функція набуває в точках локальних мінімумів (максимумів), і вибрати найменше (найбільше) серед усіх так знайдених значень.

Однак класичними методами це вдається зробити лише у випадках, коли множина  $X$  має досить просту структуру або задається як множина розв'язків деякої системи рівнянь (принцип Лагранжа).

## **4. Основні засоби пакету Mathcad щодо розв'язування задач оптимізації**

В системі Mathcad екстремальні задачі можна розв'язувати за допомогою класичних методів, при цьому розв'язок можна одержати не лише у чисельному,

а й в аналітичному вигляді.

**Приклад 1.** Розв'язати задачу Тартальї: розділити число 8 на дві такі частини, щоб добуток їх добутку на їх різницю був максимальним, за допомогою класичного методу (див. правило I) в аналітичному і чисельному вигляді

**Розв'язування:**

На рис. 5.2 показано знаходження розв'язку задачі Тартальї класичним методом за допомогою системи Mathcad, при цьому  $x_1$  – точка максимуму функції  $f(x)$ , оскільки друга похідна в точці  $x_1$  менше нуля, а  $x_2$  – точка мінімуму, оскільки друга похідна в точці  $x_2$  більше нуля.

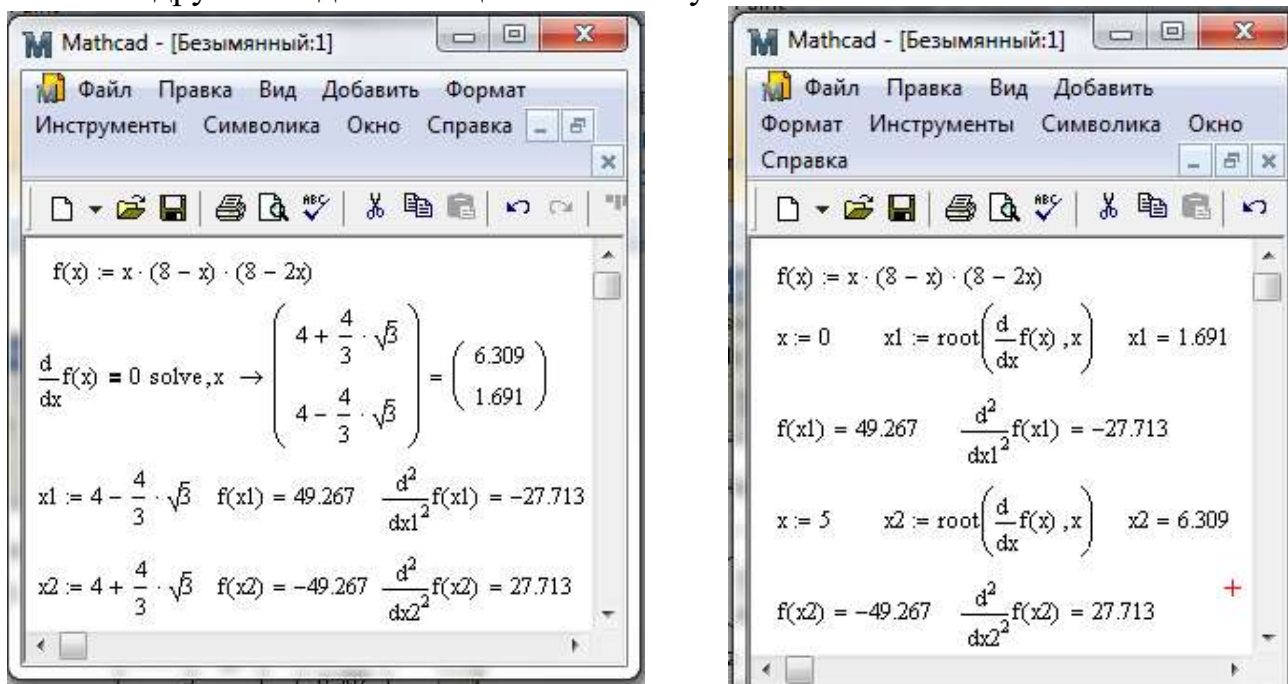


Рисунок 5.2 – Знаходження розв'язку задачі

При розв'язуванні задачі було використано операції диференціювання  $\frac{d}{dx} f(x)$ ,  $\frac{d^2}{dx^2} f(x)$  – для знаходження першої і другої похідних функції  $f(x)$ , команду solve для розв'язування рівняння  $\frac{d}{dx} f(x) = 0$ , яка активізується або через систему меню Symbolics\Variable\Solver, або за допомогою панелі Symbolic, а також функцію  $root(f(var), var, [a, b])$ , за допомогою якої знаходяться наближені корені рівняння виду  $f(var)=0$  на проміжку  $[a, b]$  відносно змінної  $var$ .

**З а у в а ж е н н я.** Для знаходження всіх коренів многочлена  $n$ -го степеня виду  $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$  в системі Mathcad можна використовувати функцію  $polyroots(a)$ , де  $a = (a_0, a_1, \dots, a_{n-1}, a_n)^T$  – вектор-стовпчик коефіцієнтів заданого многочлена.

**П р и м і т к и.**

1. В системі Mathcad для наближеного розв'язування систем нелінійних рівнянь можна використовувати також функцію  $Minerr(var1, var2, \dots, varN)$ .

2. Для розв'язування систем лінійних рівнянь виду  $Ax = b$  можна використовувати функцію  $lsolve(A,b)$ , де  $A$  – невироджена квадратна матриця коефіцієнтів системи,  $b$  – вектор-стовпчик вільних членів системи рівнянь.

Також для розв'язування задач оптимізації за допомогою системи Mathcad використовуються вбудовані функції  $Maximize$  і  $Minimize$ , які мають такий синтаксис:

де  $f$  – ім'я цільової функції задачі,  $var1, var2, \dots, varN$  – змінні, від яких залежить цільова функція.

Результатом виконання функцій  $Minimize$  і  $Maximize$  є вектор-стовпчик, який є точним або наближеним розв'язком задачі відшукування відповідного екстремуму функції  $f$  із заданою точністю.

При розв'язуванні задачі оптимізації у вікні редактора Mathcad треба спочатку задати вхідні дані задачі (значення констант, векторів, матриць), цільову функцію та початкові значення змінних, від яких вона залежить. Потім, якщо розв'язується задача умовної оптимізації, записати службове слова  $Given$ , яке визначає початок розв'язкового блоку (solve block), і після нього записати обмеження задачі (обмеження-рівняння, обмеження-нерівності, прями обмеження). При цьому запис можна здійснити у векторному або матричному вигляді.

Коли цільова функція і обмеження задачі (якщо вони є) описані, записується одна з функцій оптимізації ( $Maximize()$  або  $Minimize()$ ) за правилами відповідного синтаксису. Причому можна відразу отримати результат, натиснувши на клавішу «=», або надати одержане значення змінній, яку можна використати при подальших обчисленнях, наприклад, для знаходження значення цільової функції в оптимальній точці.

При розв'язуванні задач нелінійного програмування, вказавши за допомогою мишки на функції  $Maximize$  або  $Minimize$ , в контекстному меню можна обрати один з ітераційних методів пошуку екстремуму, зокрема метод лінійних наближень, метод квазіньютонівських наближень, метод спряжених градієнтів, метод квадратичних наближень.

## 5. Розв'язування задач математичного програмування в середовищі

**Приклад 2.** Знайти розв'язок екстремальної задачі в середовищі Mathcad:

$$f(x_1, x_2) = 3x_1 + 2x_2 \rightarrow \max ,$$
$$\begin{cases} x_1 + x_2 \leq 2000, \\ 3x_1 + x_2 \leq 3000, \\ -10x_1 + x_2 \leq 1000, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$$

### Розв'язування.

Маємо задачу лінійного програмування від двох змінних. Розглянемо два можливих способи розв'язування даної задачі. У першому варіанті (рис. 3) обмеження задаються у такому ж вигляді, як і в умові задачі, а у другому (рис. 4) – обмеження задаються у матричному вигляді. Щоб задати обмеження в матричному вигляді треба записати матрицю коефіцієнтів і вектор-стовпчик вільних членів для системи обмежень:

$$A = \begin{pmatrix} 1 & 1 \\ 3 & 1 \\ -10 & 1 \end{pmatrix} \quad \text{Тоді обмеження задачі будуть мати більш компактний вигляд:}$$

$$A \cdot x \leq b,$$

$b = \begin{pmatrix} 2000 \\ 3000 \\ 1000 \end{pmatrix}$  Як видно з рис. 5.3 в результаті обчислень одержуються однакові оптимальні точки. Але матричний спосіб запису обмежень надає можливість значно зекономити час на запис умови задачі, особливо, при великій кількості змінних.

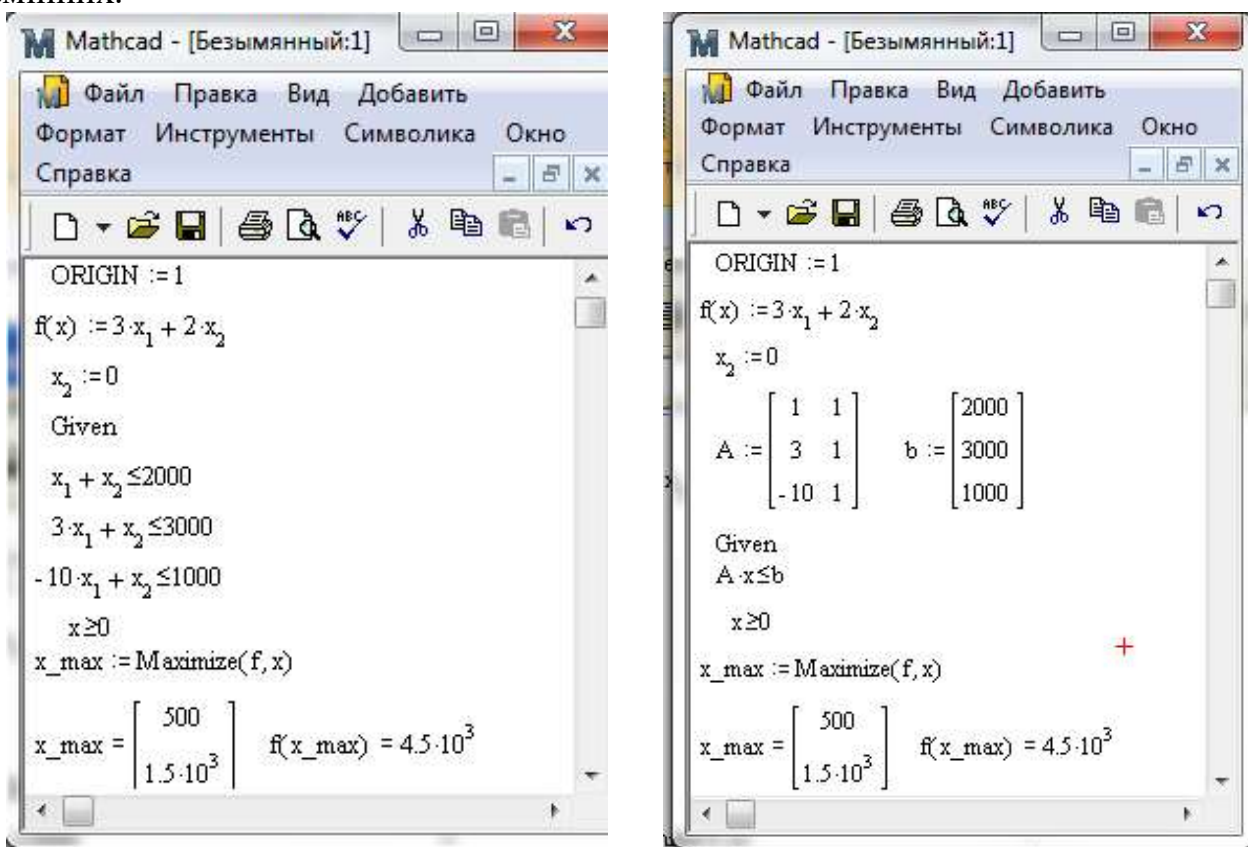


Рисунок 5.3 – Розв'язок задачі лінійного програмування від двох змінних

### Приклад 3. Знайти розв'язок транспортної задачі в середовищі Mathcad.

У трьох пунктах постачання зосереджений однорідний вантаж в обсязі 420 т, 380 т і 400 т. Цей вантаж необхідно перевезти в три пункти призначення відповідно в обсязі 260 т, 520 т і 420 т. Вартості перевезення (в грн) однієї тони вантажу з кожного пункту постачання до кожного пункту призначення є відомими величинами і задаються матрицею

$$\begin{pmatrix} 2 & 4 & 3 \\ 7 & 5 & 8 \\ 6 & 9 & 7 \end{pmatrix}.$$

Знайти план перевезень, який забезпечує вивезення наявного в пунктах постачання вантажу і завезення його до пунктів призначення в необхідній кількості при мінімальній загальній вартості перевезень.

Для транспортної задачі її умову зручно подавати у табличній формі:

Таблиця 5.1.

Пункти постачання	Пункти призначення			Запаси (т)
	B1	B2	B3	
A1	2	4	3	420
A2	7	5	8	380
A3	6	9	7	400
Потреби (т)	260	520	420	

Побудуємо математичну модель цієї задачі. Позначимо через  $x_{ij}$  – кількість одиниць вантажу, яку заплановано перевезти з  $A_i$ -го пункту постачання до  $B_j$ -го пункту призначення, де  $i = \overline{1,3}$ ,  $j = \overline{1,3}$ . Тоді план перевезень можна записати у вигляді матриці

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix},$$

при цьому загальна вартість перевезення вантажу буде дорівнювати

$$f(X) = 2x_{11} + 4x_{12} + 3x_{13} + 7x_{21} + 5x_{22} + 8x_{23} + 6x_{31} + 9x_{32} + 7x_{33}$$

і її потрібно мінімізувати.

За умовою задачі необхідно вивезти весь вантаж з пунктів постачання, тому план перевезень повинен задовольняти такі умови

$$\begin{cases} x_{11} + x_{12} + x_{13} = 420, \\ x_{21} + x_{22} + x_{23} = 380, \\ x_{31} + x_{32} + x_{33} = 400, \end{cases}$$

а оскільки, крім того, необхідно повністю задовольнити потреби пунктів призначення, то план перевезень повинен також задовольняти умови

$$\begin{cases} x_{11} + x_{21} + x_{31} = 260, \\ x_{12} + x_{22} + x_{32} = 520, \\ x_{13} + x_{23} + x_{33} = 420. \end{cases}$$



Враховуючи те, що за планом вантаж може бути перевезено з  $A_i$ -го пункту постачання до  $B_j$ -го пункту призначення або ні, то на змінні  $x_{ij}$  накладається умова:

$$x_{ij} \geq 0, \quad i = \overline{1,3}, \quad j = \overline{1,3}.$$

Відомо, що для того щоб транспортна задача мала розв'язок, необхідно і достатньо щоб були рівними запаси вантажу і його потреби (умова замкненості задачі). Перевіряючи умову замкненості транспортної задачі, тобто обраховуючи сумарні запаси і потреби, маємо:  $\sum_{i=1}^3 a_i = 1200$  т,  $\sum_{j=1}^3 b_j = 1200$  т. Отже задача є замкненою і тому має розв'язок, який знайдемо за допомогою пакету Mathcad.

Щоб розв'язати транспортну задачу в середовищі *Mathcad* потрібно скласти матрицю вартостей перевезень одиниці товару для замкненої задачі, а також записати вектори-стовпчики запасів і потреб.

Матриця вартостей  $c$  буде мати вигляд:  $c = \begin{pmatrix} 2 & 4 & 3 \\ 7 & 5 & 8 \\ 6 & 9 & 7 \end{pmatrix}$ . Вектори-стовпчики

запасів і потреб дорівнюють відповідно:  $a = \begin{pmatrix} 420 \\ 380 \\ 400 \end{pmatrix}$ ,  $b = \begin{pmatrix} 260 \\ 520 \\ 420 \end{pmatrix}$ . Далі складаємо

математичну модель задачі, тобто цільову функцію та систему обмежень, задаємо початкове наближення та отримуємо розв'язок (рис. 5.4).

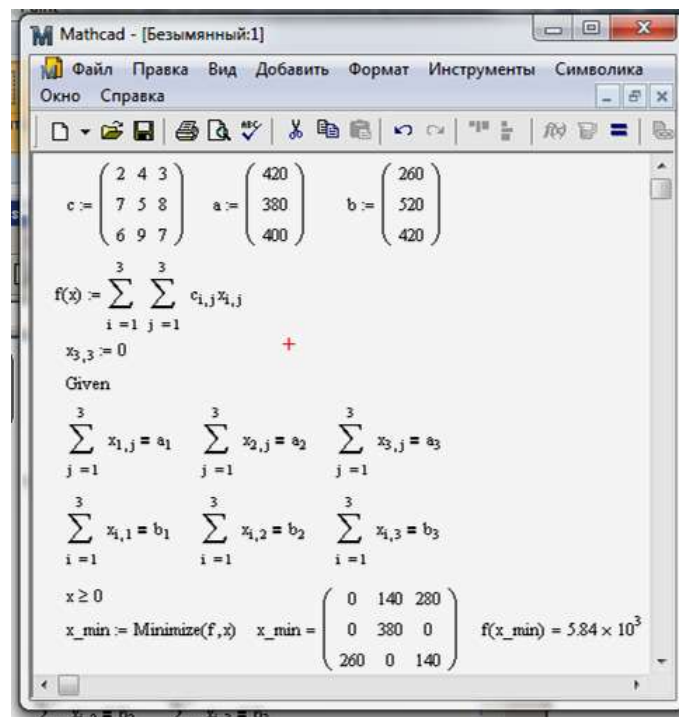


Рисунок 5.4 – Приклад розв'язку транспортну задачу в середовищі *Mathcad*

**Зауваження.** При розв'язуванні даної задачі потрібно в контекстному

М  
е  
н  
ю  
,  
В  
Л  
Л  
я  
а  
Н  
О  
М  
у  
Д  
я  
К  
о  
М  
а  
Н  
Д  
и

## II. Завдання до лабораторної роботи

1. Розв'язати за допомогою пакету Mathcad такі екстремальні задачі:

**Задача 1.** Селянин хоче розчистити поле у вигляді прямокутної ділянки, до однієї з сторін якої приєднується ділянка у формі півкруга. Прямокутна ділянка буде відведена під сіно, яке дає прибуток у 5 одиниць вартості на  $1 \text{ м}^2$ , а ділянка у формі півкруга буде засіяна гречкою, яка дає прибуток у 6 одиниць вартості на  $1 \text{ м}^2$ . Периметр поля повинен бути рівним 800 м. Як селянин повинен спланувати поле, щоб одержати найбільший прибуток?

Розв'язати задачу за допомогою пакету Mathcad:

- 1) безпосередньо, використовуючи оператор Maximize;
- 2) використовуючи метод виключення змінних.

**Розв'язування:**

Побудуємо математичну модель задачі. Нехай  $a$  – ширина прямокутного поля,  $b$  – довжина прямокутного поля,  $r$  – радіус поля у формі півкруга, який дорівнює  $a/2$ .

Тоді площа відповідних частин поля дорівнює:

$$S_{\text{прямокут.}} = ab, \quad S_{\text{півкруга}} = \frac{\pi a^2}{8}, \quad \text{а периметр поля } P = a + 2b + \pi a = 800.$$

Звідси цільова функція задачі має такий вигляд:

$$f(a, b) = 5ab + 6 \frac{\pi a^2}{8}.$$

Таким чином, математична модель задачі буде такою:

, вказати метод лінійної ітерації, тобто вибрати пункт **Linear**.

$$f(a, b) = 5ab + 6 \frac{\pi a^2}{8} \rightarrow \max, \quad a + 2b + \pi a = 800, \quad a \geq 0, b \geq 0.$$

1. Безпосереднє розв'язування з використанням оператора Maximize.

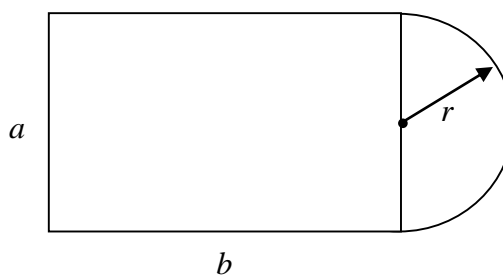
$$f(a, b) := 5a \cdot b + \frac{3\pi a^2}{4}$$

$$a := 1$$

$$\text{Given } b := 1$$

$$a + 2 \cdot b + \pi a = 800$$

$$x := \text{Maximize}(f, a, b)$$



$$x = \begin{pmatrix} 125.035 \\ 141.079 \end{pmatrix}$$

$$f(x_1, x_2) = 1.25 \times 10^5$$

2. Метод виключення змінних.

$$f(a) := 5a \cdot \left[ 400 - \frac{a \cdot (1 + \pi)}{2} \right] + \frac{3\pi a^2}{4}$$

$$\frac{d}{da} f(a) \rightarrow$$

$$a := 1$$

$$a\_max := \text{root} \left[ 2000 - \frac{5}{2} \cdot a \cdot (1 + \pi) + 5 \cdot a \cdot \left( \frac{-1}{2} - \frac{1}{2} \cdot \pi \right) + \frac{3}{2} \cdot \pi \cdot a, a \right]$$

$$a\_max = \blacksquare$$

$$b\_max := 400 - \frac{a\_max \cdot (1 + \pi)}{2}$$

$$b\_max = \blacksquare$$

$$f(a\_max) = \blacksquare$$

**Задача 2.** Відрізок заданої довжини  $c$  переміщується так, що його кінці ковзають вздовж сторін прямого кута. При якому положенні цього відрізка площа трикутника, який відтинається, буде найбільшою?

Скласти математичну модель задачі та розв'язати її класичними методами та за допомогою програми Mathcad при  $c=10$ .

**Задача 3.** Розкласти додатне число  $a$  на три додатних доданки так, щоб добуток їх був найбільшим.

Скласти математичну модель задачі та розв'язати її класичними методами і за допомогою програми Mathcad при  $a=100$ .

**Задача 4.** Людина, що гуляє у лісі, знаходиться за 5 км від прямолінійної дороги, яка вважається нескінченно довгою, і за 13 км від дому, який стоїть біля дороги. Людина може рухатися із швидкістю 3 км за годину лісом і 5 км за годину дорогою. Спочатку вона іде по прямій лінії до дороги, а потім дорогою до дому. Знайти мінімальний час, за який людина може дістатися до дому.

Скласти математичну модель задачі та розв'язати її класичними методами і за допомогою програми Mathcad.

2.2. Розв'язати задачі з індивідуального завдання класичними методами і за допомогою програми Mathcad.

2.3. Оформити звіт про виконання завдань.

### III. Індивідуальні завдання

1. Використовуючи необхідні і достатні умови локального мінімуму, розв'язати задачі безумовної оптимізації за правилом II:

1)  $f(x) = x_1^2 + x_2^2 - x_1x_2 \rightarrow \min, x \in R^2;$

2)  $f(x) = x_1^3 + x_2^3 - 5x_1x_2 \rightarrow \min, x \in R^2;$

3)  $f(x) = x_1^3 + x_2^3 - x_1^2 - 2x_1x_2 - x_2^2 \rightarrow \min, x \in R^2;$

4)  $f(x) = x_1x_2 + \frac{1}{2(x_1 + x_2)} \rightarrow \min, x \in R^2;$

5)  $f(x) = \sin(x_1 + x_2) - \sin x_1 - \sin x_2 \rightarrow \min, x \in R^2;$

6)  $f(x) = 3x_1^4 + 2x_2^4 - 5x_1x_2 \rightarrow \min(\max), x \in R^2;$

7)  $f(x) = 10 \ln x_1 + x_1x_2^2 - x_2^3 \rightarrow \min(\max), x \in R^2;$

8)  $f(x) = (x_1 - x_1x_2 + x_2)(x_1 + x_2) \rightarrow \min(\max), x \in R^2;$

9)  $f(x) = 2x_1^2 + x_2^2 + 10x_1x_2 \rightarrow \min, x \in R^2;$

10)  $f(x) = x_1x_2^2(1 - x_1 - 2x_2) \rightarrow \min(\max), x \in R^2.$

2. Розв'язати реальні оптимізаційні задачі за допомогою пакету Mathcad:

1. Для збереження здоров'я і працездатності людина повинна споживати за добу певну кількість корисних речовин, наприклад, білків, жирів, вуглеводів, води і вітамінів. Запаси цих інгредієнтів в різних видах їжі  $P_i$  ( $i=1,2,\dots$ ) різні. Обмежимося для простоти двома видами їжі. Запаси вказаних речовин, вартість 1 кг їжі певного виду наведено в таблиці.

Корисні речовини	Види їжі та вміст корисних речовин (г)		Мінімальна норма (г)
	$P_1$	$P_2$	
Жири	10	<b>50</b>	100
Білки	30	20	120
Вуглеводи	20	40	160
Вода	200	300	1000
Вітаміни	10	5	10
Вартість	20	30	

Яким чином треба організувати харчування, щоб вартість його була найменшою, але організм одержував би не менше мінімальної добової норми корисних речовин всіх видів.

2. Для виготовлення двох видів виробів А і В використовується токарне, фрезерне та шліфувальне обладнання. Норми затрат часу для кожного типу обладнання на один виріб кожного виду наведені в таблиці. В тій же таблиці

вказаний загальний фонд робочого часу кожного з типів обладнання, а також прибуток від реалізації одного виробу.

Тип обладнання	Норми витрат часу на виготовлення однієї деталі (год)		Загальний фонд робочого часу (год)
	Виріб А	Виріб В	
Фрезерне	10	8	168
Токарне	5	10	180
Шліфувальне	6	12	144
Прибуток від реалізації одного виробу (грн.)	1400	1800	

Знайти план виготовлення деталей, який забезпечує максимальний прибуток від її реалізації.

3. Для перевезення пасажирів між двома населеними пунктами автопарком використовуються автобуси типу «ЛАЗ» і «Ікарус». Кількість пасажирів, яку можуть за один рейс перевезти ці автобуси, відповідно становить 36 і 42. За один рейс «ЛАЗ» витрачає 2,5 кг мастильних матеріалів і 45 л палива, «Ікарус» – відповідно 2 кг і 55 л. В автопарку є 250 кг мастильних матеріалів і 1400 л палива. Витрати на експлуатацію одного «ЛАЗа» за один рейс становлять 12 грн, а на одного «Ікаруса» – 17 грн. Треба перевезти не менше 500 пасажирів, при цьому кількість працюючих «ЛАЗів» у автопарку не більше 8, а «Ікарусів» не більше 10. Скільки «ЛАЗів» та «Ікарусів» необхідно використати автопарку, щоб перевезти всіх пасажирів і експлуатаційні витрати за наявних ресурсів були мінімальними?

4. Кондитерська фабрика для виготовлення двох видів цукерок «Батончик» і «Каракум» використовує три види сировини: цукор, шоколад і фруктове пюре. Норми витрат сировини кожного виду на виробництво 1 т цукерок даного виду наведено в таблиці. В ній також вказана загальна кількість сировини кожного виду, що може бути використана фабрикою, та наведено прибуток від реалізації 1 т цукерок даного виду.

Вид сировини	Норми витрат сировини (т) на одну тону цукерок		Загальна кількість сировини (т)
	«Батончик»	«Каракум»	
Цукор	1.6	0.6	400
Шоколад	3.1	4.3	600
Фруктове пюре	0.1	0.1	320
Прибуток від реалізації 1 т карамелі (грн.)	132	156	

Знайти план виготовлення цукерок, який забезпечує максимальний прибуток від їх реалізації.

5. Підприємство виготовляє три види продукції PVC: глухі вікна, комбіновані вікна (глухі поворотно відкідні), балконні двері. Відпускні ціни на цю продукцію становлять: глухе вікно – 385 грн, комбіноване вікно – 920 грн, балконні двері – 985 грн.

Для виготовлення вікон і дверей використовують три види пластикового профілю: раму, створку та імпост. Норми витрат цих матеріалів на одиницю продукції подано в таблиці.

Вид профілю	Норми витрат (м) на один виріб		
	Глухе вікно	Комбіноване вікно	Балконні двері
Рама	5.16	5.8	6.23
Створка	0	4.23	5.87
Імпост	0	1.39	0.53

Кожен місяць на підприємство завозять 780 м рами, 390 м створки, 120 м імпосту. В середньому попит на місяць становить: на глухі вікна – від 45 шт – до 60 шт; комбіновані вікна – 55шт – 70 шт, балконні двері – 20шт – 30 шт.

Скільки вікон та дверей необхідно виготовляти кожного місяця, щоб максимізувати дохід від продажу і не мати дефіциту матеріалів та задовольнити попит?

6. Машинобудівний завод виготовляє 3 види продукції: візок вантажний гідравлічний; штабелер вантажний гідравлічний, кран гідравлічний. Ціни на цю продукцію становлять: візок вантажний гідравлічний – 1040 грн; штабелер вантажний гідравлічний – 2750 грн, кран гідравлічний – 5200 грн. На виготовлення цієї продукції використовуються метал і шестерні. Норми витрат цих матеріалів на одиницю продукції та їх записи подано в таблиці.

Матеріали	Норми витрат на один виріб			
	Візок вантажний гідравлічний	Штабелер вантажний гідравлічний	Кран гідравлічний	Запаси (на місяць)
Метал (кг)	30	100	170	2000
Шестерні (шт)	4	12	8	500

Визначити кількість продукції, яку необхідно виготовити, щоб прибуток від її реалізації був найбільшим.

7. Фірма займається працевлаштуванням сільгосп працівників в Англії. На сьогодні робочі місця існують у трьох містах Норіджі, Брайтоні та Редінзі. У них працюють відповідно 8, 5 та 3 працівники. Термін контракту закінчується, тому робітників необхідно перевезти до Дувру, Лутону, Суїндону та Бердфорду за

рахунок фірми у кількості відповідно 4, 6, 3 та 3 працівники. Витрати на організацію переїзду (у фунтах стерлінгах) подано у таблиці.

Пункти відправлення	Пункти призначення			
	Дувр	Лутон	Суіндон	Бердфорд
Норидж	98	45	87	39
Брайтон	29	40	43	51
Редінг	52	28	21	33

Яким чином організувати перевезення працівників, щоб витрати були найменшими.

8. У трьох пунктах постачання  $A_1, A_2, A_3$  зосереджений однорідний вантаж. Цей вантаж необхідно перевезти в три пункти призначення  $B_1, B_2, B_3$ . Вартість перевезення однієї тони вантажу з кожного пункту постачання до кожного пункту призначення є відомими величинами і задані у таблиці. Там же вказано обсяги наявних запасів вантажу на кожному з трьох пунктах постачання, а також потреби пунктів призначення у вантажі. Знайти план перевезень, який забезпечує вивезення наявного в пунктах постачання вантажу і завезення його в необхідному обсязі до кожного пункту призначення при мінімальній загальній вартості перевезень.

Пункти постачання	Пункти споживання			Запаси (т)
	$B_1$	$B_2$	$B_3$	
$A_1$	7	6	3	205
$A_2$	2	5	4	195
$A_3$	9	10	11	240
Потреби (т)	175	225	230	

9. Фірма має філіали в містах  $A_i$  країни ( $i=1,2,3$ ). В кожному місці працює  $a_i$  рекламних агентів. Фірма збирається провести рекламну кампанію, для чого необхідно направити рекламних агентів до міст  $B_j$  ( $j=1,2,3,4$ ) країни у кількості  $b_j$  співробітників, відповідно. Вартість проїзду  $c_{ij}$  для однієї людини з міста  $A_i$  до міста  $B_j$  вказана в таблиці, де також наведені дані для  $a_i$  і  $b_j$ . Необхідно скласти план поїздок, який задовольняє потреби рекламної кампанії і має найменшу вартість.

Пункти Відправлення	Пункти призначення				Штат рекламних агентів ( $a_i$ )
	$B_1$	$B_2$	$B_3$	$B_4$	
$A_1$	2	4	1	5	6
$A_2$	3	4	2	1	3
$A_3$	3	2	1	5	7
Потреби ( $b_j$ )	5	7	1	3	

10. На трьох елеваторах є зерно в обсязі 4200 т, 3800 т і 4000 т. Це зерно необхідно перевезти до трьох заводів з виготовлення борошна відповідно в обсязі 2600 т, 5200 т і 4200 т. Вартості перевезення однієї тони зерна з кожного елеватора до кожного заводу (в грн.) задані в таблиці.

Пункти постачання	Пункти призначення		
	Завод 1	Завод 2	Завод 3
Елеватор 1	2	4	3
Елеватор 2	7	5	8
Елеватор 3	6	9	7

Знайти план перевезень, який забезпечить вивезення наявного на елеваторах зерна і завезення його в необхідній кількості на заводи з виготовлення борошна при мінімальній загальній вартості перевезень.



## Лабораторна робота № 6

**Тема:** «Геометричний метод розв'язування задачі лінійного програмування»

Питання:

1. Геометрична інтерпретація стандартної задачі лінійного програмування.
2. Геометричний метод розв'язування задачі лінійного програмування.
3. Розв'язування практичних задач.

Програмне забезпечення: програма **EXTREMUM**.

### I. Теоретичний матеріал

1. Розглянемо геометричну інтерпретацію стандартної задачі лінійного програмування.

Задача оптимізації виду

$$f(x) = \sum_{j=1}^n c_j x_j \rightarrow \max(\min), \quad (1)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, i = \overline{1, k}, \sum_{j=1}^n a_{ij} x_j = b_i, i = \overline{k+1, m}, \quad (2)$$

$$x_j \geq 0, j = \overline{1, n},$$

де  $c_j, a_{ij}, b_i \in R^1$  для  $i = \overline{1, m}, j = \overline{1, n}$  називається **загальною** задачею лінійного програмування.

Якщо допустима множина визначається обмеженнями виду

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, i = \overline{1, m}, x_j \geq 0, j = \overline{1, n}, \quad (3)$$

то задача (1), (3) називається **стандартною**.

Особливість геометричної інтерпретації двовимірної задачі лінійного програмування полягає в тому, що, по-перше, лінія рівня цільової функції  $f(x_1, x_2) = c_1 x_1 + c_2 x_2$  є пряма, при цьому **градієнт** функції  $f(x_1, x_2)$  (вектор  $\bar{c} = (c_1, c_2)$ ) є напрямом її найшвидшого зростання, по-друге, якщо задача має розв'язок, то він досягається обов'язково на межі допустимої множини  $X$ , яка в цьому випадку являє собою обмежену многокутну область, а сам розв'язок задачі є або деяка вершина многокутника  $x^*$  (рис. 6.1 а) або множина точок деякої його сторони  $X^*$  (рис. 6.1 б).

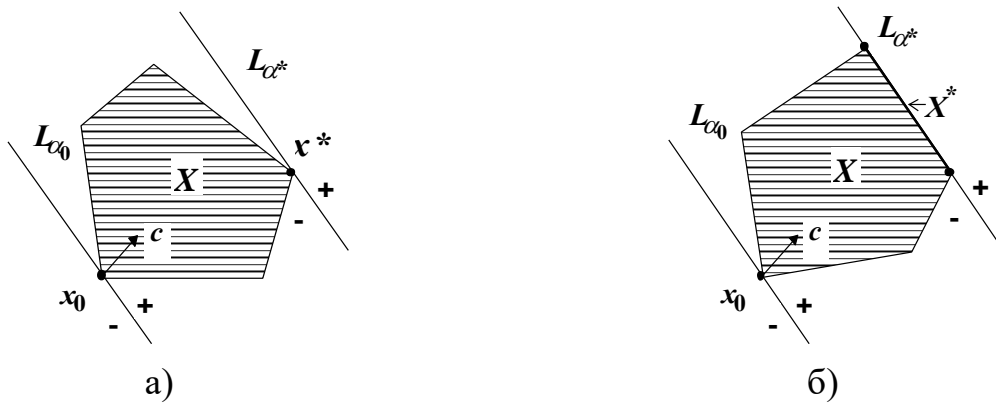


Рисунок 6.1 – Геометричну інтерпретація задачі лінійного програмування

Наведемо алгоритм розв'язування стандартної двовимірної задачі лінійного програмування геометричним методом.

### АЛГОРИТМ розв'язування стандартної двовимірної задачі лінійного програмування геометричним методом

1. Побудувати прямі, рівняння яких одержуються внаслідок заміни в обмеженнях (3) знаків нерівностей на знаки рівностей.

2. Знайти півплощини, які визначаються кожним з обмежень-нерівностей задачі.

3. Знайти багатокутник допустимих розв'язків.

4. Побудувати пряму  $c_1x_1 + c_2x_2 = h$  (лінію рівня цільової функції), при цьому величина  $h$  підбирається так, щоб лінія рівня проходила через багатокутник допустимих розв'язків.

5. Побудувати вектор  $\bar{c} = (c_1, c_2)$ .

6. Рухаючи пряму  $c_1x_1 + c_2x_2 = h$  в напрямку вектора  $\bar{c}$  при розв'язанні задачі максимізації (або в зворотньому напрямку при розв'язанні задачі мінімізації), знайти точку (множину точок), де цільова функція приймає максимальне (мінімальне) значення, або встановити необмеженість зверху (знизу) функції на допустимій множині (див. рис.б.2).

7. Якщо існує єдиний розв'язок задачі, визначити координати знайденої точки і обчислити значення цільової функції в цій точці. Якщо існує безліч розв'язків, то визначити координати принаймні однієї екстремальної точки і обчислити значення цільової функції в цій точці.

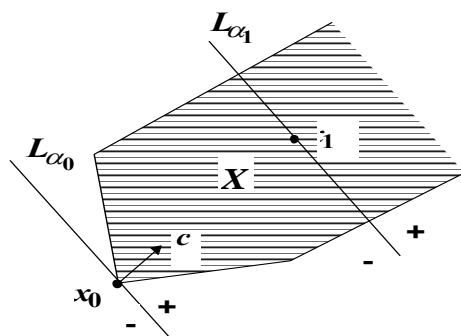


Рисунок 6.2 – Геометричну інтерпретація задачі лінійного програмування

## II. Завдання до лабораторної роботи

1. Опрацювати теоретичний матеріал.
2. Розв'язати задачу лінійного програмування за допомогою програми EXTREMUM.

### Задача

Для виробництва столів та шаф фабрика використовує певні ресурси. Норми витрат ресурсів на один вироб даного виду, прибуток від реалізації одного виробу та загальна кількість наявних ресурсів наведено в таблиці:

Ресурси	Норми витрат ресурсів на один вироб		Загальна кількість ресурсів
	Стіл	Шафа	
Дерево (м <sup>2</sup> ):			
І вида	2	1	450
II вида	1	4	700
Трудомісткість (людино-год.)	15	12	3732
Прибуток від реалізації одного виробу (ум.од.)	18	24	

Визначити скільки столів та шаф необхідно виготовити фабриці, щоб прибуток від їх реалізації був максимальним.

**Розв'язування.** Математична модель даної задачі має такий вигляд:

$$F(x_1, x_2) = 18x_1 + 24x_2 \rightarrow \max ,$$

$$\begin{cases} 2x_1 + x_2 \leq 450, \\ x_1 + 4x_2 \leq 700, \\ 15x_1 + 12x_2 \leq 3732, \end{cases}$$

$$x_1 \geq 0, x_2 \geq 0, x_1 \in Z, x_2 \in Z.$$

Отже поставлена задача є стандартною задачею лінійного програмування, Процес розв'язування задачі за допомогою програми EXTREMUM відбувається за наведеним вище алгоритмом.

1. За допомогою меню програми «**Модель**» (в режимі «**Задача лінійного програмування**») і команди «**Редагування моделі**» ввести математичну модель задачі (рис. 6.3-6.4).

Після введення даних модель задачі матиме вигляд, який подано на мал. 4.

2. В меню «**Побудова**» вибрати команду «**Прямих**», за допомогою якої автоматично будуються прямі, які відповідають обмеженням-нерівностям задачі (рис. 6.5).



Рисунок 6.3

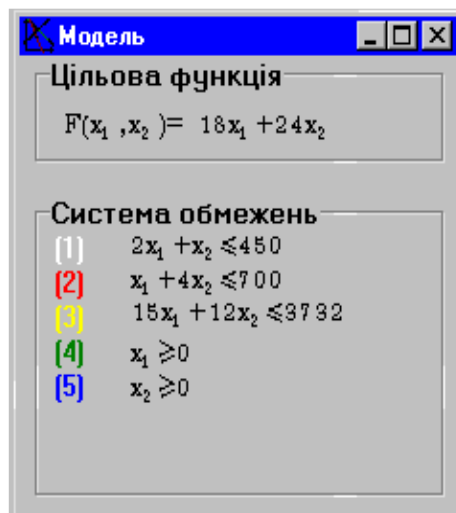


Рисунок 6.4

3. За допомогою команди «Півплощин» в меню «Побудова», використовуючи «мишку», визначити півплощини, які відповідають обмеженням-нерівностям задачі (рис. 6.6).

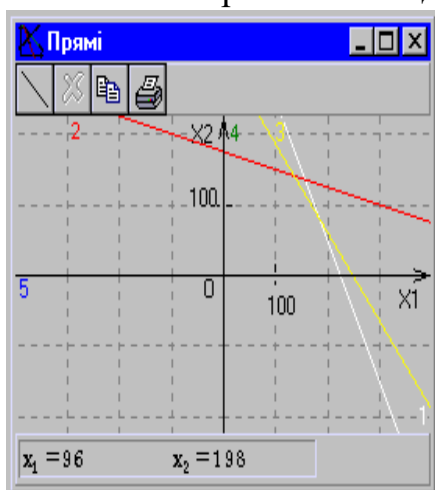


Рисунок 6.5

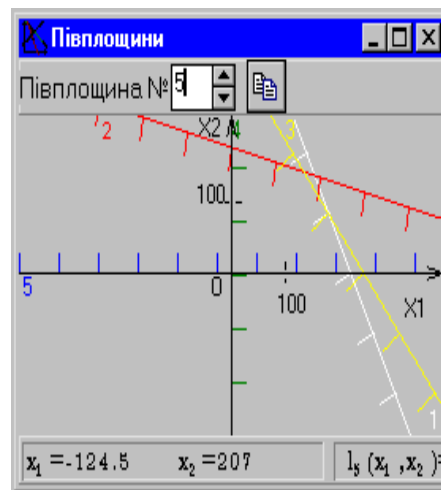


Рисунок 6.6

4. За допомогою команди «Многокутника» в меню «Побудова», побудувати многокутник допустимих розв'язків, який є перетином знайдених півплощин, вказуючи «мишкою» його вершини (рис. 6.7). Як видно з рис. 6.7, многокутник допустимих розв'язків є обмеженим і опуклим, тому розв'язок задачі існує і міститься на його межі.

5. За допомогою команди «Лінії рівня» в меню «Побудова», визначити координати точки максимуму цільової функції на допустимій множині та обчислити значення функції в цій точці.

Для цього за допомогою «мишки» треба переміщувати лінію рівня в напрямку вектора  $\bar{c}$ . Як видно з рис. 6.8, точкою максимуму цільової функції на допустимій множині є точка перетину прямих з номерами 2 і 3. Згідно графічного розв'язку вона має координати (136;141.5), а цільова функція в цій точці набуває значення 5844. Його сума наблизений розв'язок поставленої задачі.

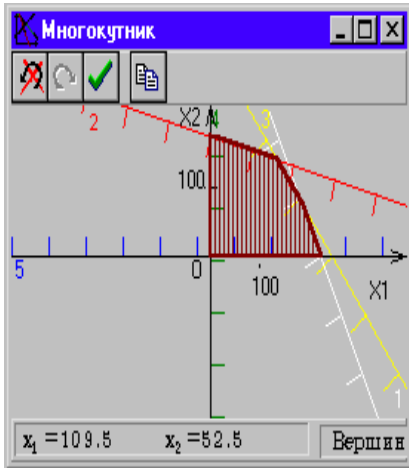


Рисунок 6.7

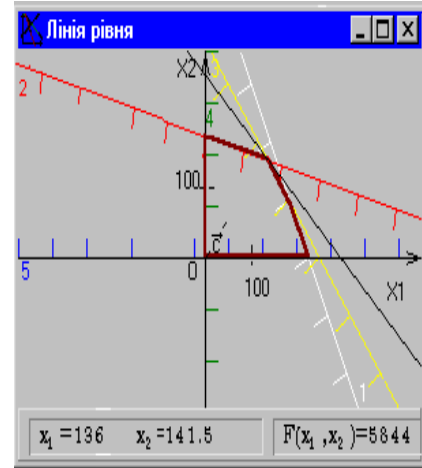


Рисунок 6.8

Для уточнення результатів треба скористатися командою «Числове уточнення». Для цього необхідно вказати номери прямих, на перетині яких міститься шукана точка (рис. 6.9). Розв'язком задачі є  $F_{\max} = 5832$  при  $x_1^* = 136$ ,  $x_2^* = 141$ .

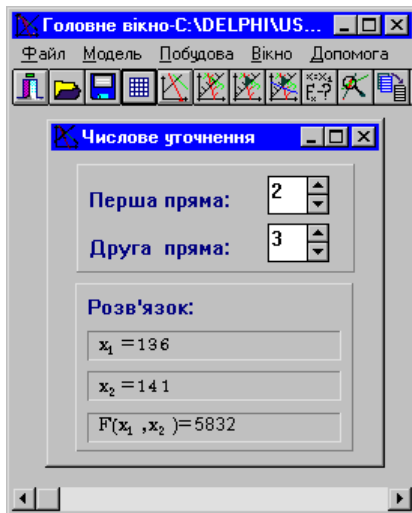


Рисунок 6.9

3. За допомогою програми Mathcad розв'язати попередню задачу і порівняти результати з результатами, одержаними геометричним методом.

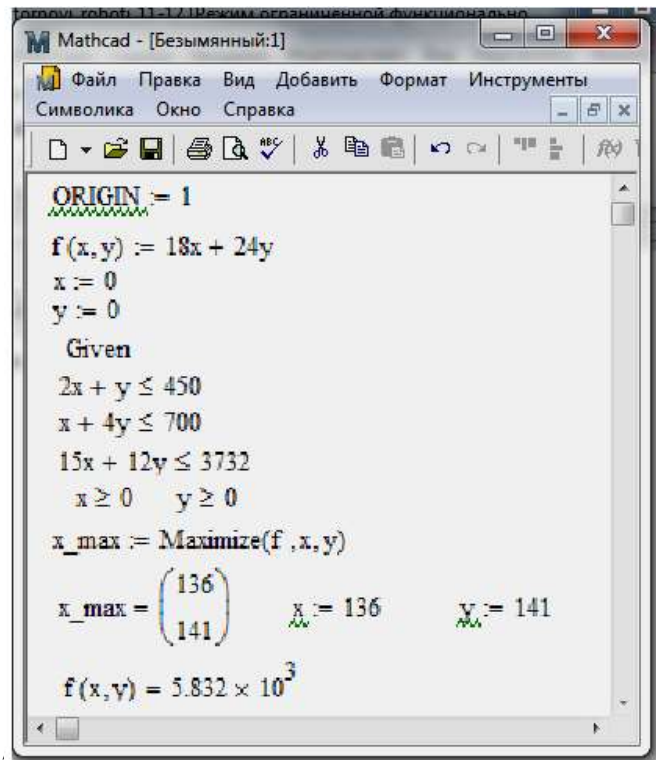


Рисунок 6.10 – Розв'язок за допомогою Mathcad

4. Виконати індивідуальне завдання.

5. Оформити звіт про виконання лабораторної роботи.

### III. Індивідуальні завдання

Для задач лінійного програмування геометричним методом за допомогою програми EXTREMUM знайти всі опорні плани і оптимальний план:

1.  $f(x_1, x_2) = 3x_1 + 7x_2 \rightarrow \min (\max)$ ,      2.  $f(x_1, x_2) = 5x_1 - 2x_2 \rightarrow \min (\max)$ ,

$$\begin{cases} x_1 + 2x_2 \leq 14 \\ -5x_1 + 3x_2 \leq 15 \\ 2x_1 + 3x_2 \geq 12 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}; \quad \begin{cases} 2x_1 - x_2 \leq 6 \\ -x_1 + 3x_2 \leq 3 \\ x_1 + 2x_2 \leq 8 \\ x_1 \geq 0, x_2 \geq 0 \end{cases};$$

3.  $f(x_1, x_2) = 4x_1 + 3x_2 \rightarrow \min (\max)$

4.  $f(x_1, x_2) = 10x_1 + 6x_2 \rightarrow \min (\max)$

$$\begin{cases} 2x_1 - x_2 \leq 2 \\ x_1 + 2x_2 \leq 3 \\ 4x_1 - x_2 \geq 1 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}; \quad \begin{cases} 2x_1 + 3x_2 \leq 21 \\ 2x_1 + x_2 \leq 15 \\ 3x_1 + 2x_2 \geq 10 \\ x_1 \geq 0, x_2 \geq 0 \end{cases};$$

5.  $f(x_1, x_2) = 2x_1 + 8x_2 \rightarrow \min (\max)$

6.  $f(x_1, x_2) = 11x_1 + 3x_2 \rightarrow \min (\max)$

$$\begin{cases} 6x_1 - 5x_2 \leq 10 \\ x_1 + 4x_2 \geq 5 \\ -x_1 + 3x_2 \geq 5 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}; \quad \begin{cases} 2x_1 - 7x_2 \leq 12 \\ -x_1 + x_2 \leq 4 \\ 3x_1 + 4x_2 \leq 30 \\ x_1 \geq 0, x_2 \geq 0 \end{cases};$$

7.  $f(x_1, x_2) = 3x_1 + 12x_2 \rightarrow \min (\max)$

8.  $f(x_1, x_2) = 4x_1 + 9x_2 \rightarrow \min (\max)$

$$\begin{cases} 2x_1 - x_2 \leq 14 \\ x_1 + 2x_2 \leq 8 \\ 7x_1 + 4x_2 \geq 25 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}; \quad \begin{cases} 3x_1 + 4x_2 \geq 3 \\ 2x_1 + 3x_2 \leq 11 \\ 2x_1 + x_2 \leq 7 \\ x_1 \geq 0, x_2 \geq 0 \end{cases};$$

9.  $f(x_1, x_2) = 25x_1 + 18x_2 \rightarrow \min (\max)$

10.  $f(x_1, x_2) = 100x_1 + 54x_2 \rightarrow \min (\max)$

$$\begin{cases} x_1 + 2x_2 \leq 14 \\ -5x_1 + 3x_2 \leq 15 \\ 2x_1 + 3x_2 \geq 12 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}; \quad \begin{cases} 5x_1 + 4x_2 \geq 31 \\ x_1 + 3x_2 \leq 12 \\ 2x_1 + 5x_2 \geq 16 \\ x_1 \geq 0, x_2 \geq 0 \end{cases};$$

2. За допомогою програми Mathcad задачі з індивідуального завдання розв'язати геометричним методом і за допомогою операторів minimize і maximize. Одержані результати порівняти між собою.

## Лабораторна робота № 7

Тема: «Параболічне інтерполювання. Інтерполяційна формула Лагранжа»

Питання:

1. Постановка задачі інтерполювання функцій.
2. Геометричний смисл задачі інтерполювання функцій.
3. Лінійна і квадратична інтерполяція.
4. Параболічне інтерполювання. Інтерполяційна формула Лагранжа.
5. Приклади інтерполювання функцій.
6. Екстраполювання функцій.
7. Засоби інтерполювання функцій в системах комп'ютерної математики.

### I. Теоретичний матеріал

#### 1. Постановка задачі інтерполювання функцій

З усіх способів задання функції найбільш зручним у багатьох випадках є аналітичний спосіб у вигляді формули. Цей спосіб дає можливість обчислити значення функції для будь-якого фіксованого значення аргументу, а отже, і скласти таблицю її значень у деяких точках. Складання таблиці значень функції називають **табулюванням функції**.

У практичних задачах значення функції, що представляють деяку фізичну величину, часто одержують у результаті експерименту у вигляді таблиці або графіка. Досить часто виникає необхідність знайти значення функції при значеннях аргументів, що відсутні в таблиці. Така задача, яку образно можна назвати задачею «читання таблиці між рядками», й одержала назву **задачі інтерполювання** (*inter* – між, *polio* – прикладати).

Задача інтерполювання функції розв'язується шляхом побудови деякого аналітичного виразу, який співпадає зі значеннями таблично заданої функції в скінченній кількості табличних значень аргументу.

Тому, задача інтерполювання функції в деякому розумінні обернена до задачі табулювання функції: при табулюванні від аналітичного способу задання функції переходять до табличного, а при інтерполюванні – за табличними значеннями функції будується деякий аналітичний вираз, тобто формула, що задає шукану функцію наближено.

**Постановка задачі.** Нехай деяка функція задана таблично, тобто для заданих значень аргументу  $x_i$  задані значення функції  $y_i = f(x_i)$  ( $i = 0, 1, 2, \dots, n$ ).

Треба знайти аналітичний вираз деякої функції  $F_n(x)$ , котра наближала би дану функцію  $f(x)$ , тобто у точках  $x_i$  приймала би значення  $y_i$  ( $i = 0, 1, 2, \dots, n$ ):

$$F_n(x_i) = f(x_i), i = 0, 1, \dots, n.$$

#### 2. Геометричний смисл задачі інтерполювання функцій

**Геометричний смисл** задачі інтерполювання функцій полягає у тому, щоб знайти деяку криву  $F_n(x)$ , котра проходила би через задану систему точок  $(x_i, y_i)$  ( $i = 0, 1, 2, \dots, n$ ) (рис. 7.1).

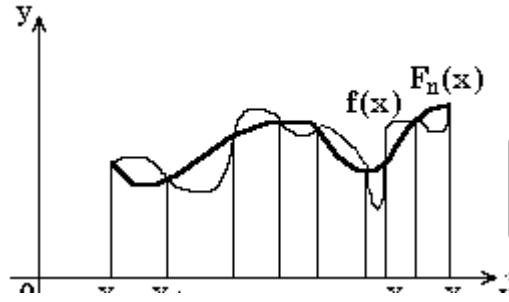


Рисунок 7.1 – Геометричний смисл інтерполювання

Інтуїтивно зрозуміло, що через задані точки можна провести нескінченну кількість різних кривих. Отже, задача відшукування функції  $f(x)$  за скінченною кількістю її значень є невизначеною.

Але ця задача стає однозначною, якщо в якості наближеної функції  $F_n(x)$  для функції  $y = f(x)$ , заданої  $n + 1$ -им своїм значенням, обрати многочлен

$$y = F_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0$$

степені  $n$ , що задовольняє таким умовам:

1) в точках  $x_i$  ( $i = 0, 1, 2, \dots, n$ ) значення многочлена  $F_n(x_i)$  співпадає зі значеннями функції  $f(x_i)$ , тобто  $F_n(x_i) = f(x_i) = y_i$  ( $i = 0, 1, 2, \dots, n$ );

2) в будь-якій іншій точці  $\tilde{x} \in (x_0; x_n)$  виконується наближена рівність  $F(\tilde{x}) \approx f(\tilde{x})$ .

Інтерполювання за допомогою алгебраїчних многочленів називається **поліноміальним** або **параболічним інтерполюванням**, многочлен  $F_n(x)$  називається **інтерполяційним многочленом**, а точки  $x_0, x_1, \dots, x_n$  називаються **вузлами інтерполювання**.

Отже, основна ідея інтерполювання полягає в заміні функції, заданої таблицею її значень, інтерполяційним многочленом, що являє собою наближений аналітичний вираз даної функції  $f(x)$ . Одержаний інтерполяційний многочлен використовують не лише для аналітичного наближення таблично заданої функції, але й у тому випадку, якщо функція задана досить складним аналітичним виразом, яким незручно користуватися в процесі розрахунків, диференціювання або інтегрування.

Розглянемо математичні основи інтерполювання функцій.

### 3. Лінійне і квадратичне інтерполювання

Найпростішими випадками інтерполювання многочленом  $F_n(x)$  є випадки, коли  $n = 1$  і  $n = 2$ .

При  $n = 1$  маємо многочлен першого степеня  $F_1(x) = a_1 x + a_0$ , тобто лінійну функцію, тому інтерполювання многочленом  $F_1(x)$  називається **лінійним інтерполюванням**.

Нехай для невідомої функції  $f(x)$  маємо її значення у двох точках  $x_0, x_1$ :



$$f(x_0) = y_0, f(x_1) = y_1.$$

Графік лінійної функції  $F_1(x) = a_1x + a_0$  повинен проходити через ці точки, тому невідомі коефіцієнти  $a_1$  і  $a_0$  можна знайти з системи рівнянь:

$$\begin{cases} a_1x_0 + a_0 = y_0 \\ a_1x_1 + a_0 = y_1 \end{cases}.$$

Розв'язавши цю систему одержимо:

$$a_1 = \frac{y_0 - y_1}{x_0 - x_1}, a_0 = \frac{x_0y_1 - x_1y_0}{x_0 - x_1},$$

тоді

$$F_1(x) = \frac{y_0 - y_1}{x_0 - x_1} \cdot x + \frac{x_0y_1 - x_1y_0}{x_0 - x_1},$$

або

$$F_1(x) = \frac{x - x_1}{x_0 - x_1} \cdot y_0 + \frac{x - x_0}{x_1 - x_0} \cdot y_1, (*)$$

відповідно функцію  $f(x)$  можна представити наближеною рівністю:

$$f(x) \approx \frac{x - x_1}{x_0 - x_1} \cdot y_0 + \frac{x - x_0}{x_1 - x_0} \cdot y_1,$$

яка називається **формулою лінійного інтерполювання**.

**Геометричний смисл:** заміна дуги кривої  $y = f(x)$  на  $[x_0, x_1]$  відрізком прямої лінії  $y = F_1(x)$ , що проходить через точки  $(x_0, y_0)$ ,  $(x_1, y_1)$  (рис. 7.2).

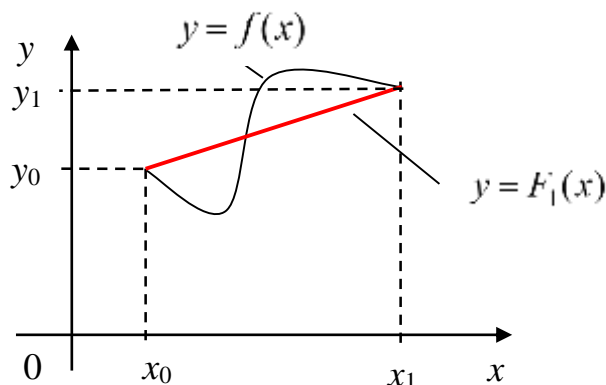


Рисунок 7.2

Більш точні результати можна одержати, якщо замість лінійної інтерполяції використовувати квадратичну інтерполяцію ( $n = 2$ ), тобто якщо дугу кривої  $y = f(x)$  замінити не відрізком прямої лінії, а параболою

$$F_2(x) = a_2x^2 + a_1x + a_0.$$

Нехай для невідомої функції  $f(x)$  маємо її значення у трьох точках  $x_0, x_1, x_2$ :

$$f(x_0) = y_0, f(x_1) = y_1, f(x_2) = y_2.$$

Графік інтерполяційного многочлена повинен проходити через ці точки, тому невідомі коефіцієнти  $a_2$ ,  $a_1$  і  $a_0$  можна знайти з системи рівнянь:

$$\begin{cases} a_2x_0^2 + a_1x_0 + a_0 = y_0 \\ a_2x_1^2 + a_1x_1 + a_0 = y_1 \\ a_2x_2^2 + a_1x_2 + a_0 = y_2 \end{cases} .$$

Наближене значення функції  $f(x)$  в будь-якій точці  $x \in (x_0, x_2)$  обчислюється за формулою:

$$f(x) \approx \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}y_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}y_2 \quad (**)$$

Цю формулу називають **формулою квадратичного інтерполювання**.

**Геометричний смисл:** заміна дуги кривої  $y = f(x)$  на  $[x_0, x_2]$  дугою параболи  $y = F_2(x)$ , що проходить через точки  $(x_0, y_0)$ ,  $(x_1, y_1)$ ,  $(x_2, y_2)$  (рис. 7.3).

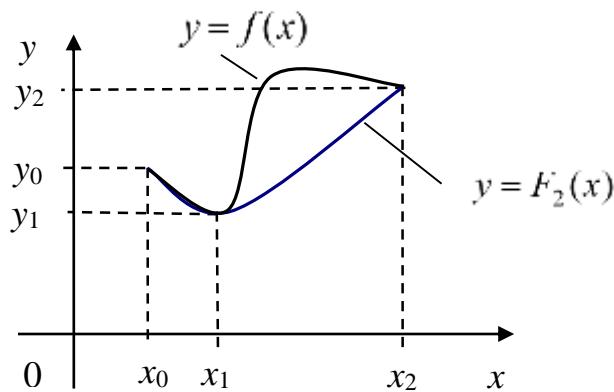


Рисунок 7.3

#### 4. Параболічне інтерполювання. Інтерполяційна формула Лагранжа

Нехай деяка функція  $f(x)$  задана своїми значеннями в  $(n+1)$ -му вузлі інтерполювання  $x_0, x_1, \dots, x_n$ :

$$f(x_i) = y_i, \quad i = 0, 1, \dots, n. \quad (1)$$

В якості інтерполяційного многочлена візьмемо многочлен  $n$ -го степеня виду:

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0, \quad (2)$$

причому значення  $P_n(x)$  у вузлах інтерполювання повинні співпадати зі значеннями заданої функції, тобто:

$$P_n(x_i) = f(x_i) = y_i, \quad i = 0, 1, \dots, n. \quad (3)$$

Ця умова визначає систему з  $(n+1)$ -го лінійного рівняння виду:



Дійсно, добутки  $F_i(x)y_i$  ( $i = 0,1,2,\dots,n$ ) перетворюються в нуль у всіх вузлах інтерполювання, крім вузла  $x_i$ , де вони рівні  $y_i$ , оскільки  $F_i(x_i) = 1$ , тобто  $P_n(x_i) = y_i$  ( $i = 0,1,2,\dots,n$ ).

Крім того, степінь многочлена  $P_n(x)$  рівна  $n$ , оскільки кожний доданок суми  $F_i(x)y_i$  є многочленом степені  $n$ .

Побудований многочлен (5) називають **інтерполяційним многочленом Лагранжа** і записують так:

$$P_n(x) = \sum_{i=0}^n \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)} y_i. \quad (6)$$

Інтерполяційна формула Лагранжа для наближеного обчислення значень функції  $f(x)$  у проміжних точках  $\tilde{x} \in (x_0; x_n)$ ,  $\tilde{x} \neq x_i$  ( $i = 0,1,2,\dots,n$ ) має вигляд:

$$f(\tilde{x}) \approx \sum_{i=0}^n \frac{(\tilde{x}-x_0)(\tilde{x}-x_1)\dots(\tilde{x}-x_{i-1})(\tilde{x}-x_{i+1})\dots(\tilde{x}-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)} y_i. \quad (7)$$

У частинному випадку, якщо є два вузли інтерполювання  $x_0$  і  $x_1$ , то з формули (6) одержуємо формулу лінійного інтерполювання (\*), а якщо три вузли  $x_0, x_1, x_2$ , то формулу квадратичного інтерполювання (\*\*).

Оскільки крива, яка є графіком многочлена  $P_n(x)$ , називається параболою  $n$ -го порядку, то інтерполювання многочленом Лагранжа називають **параболічним інтерполюванням**.

Зрозуміло, що, чим більше вузлів інтерполювання на відрізку  $[x_0; x_n]$ , тим точніше інтерполяційний многочлен наближає задану функцію.

**Приклад 1.** Для функції, значення якої в чотирьох точках задані таблицею побудувати інтерполяційний многочлен Лагранжа:

$x$	0	1	2	4
$f(x)$	1	3	2	4

Відповідь:  $P_3(x) = \frac{13}{24}x^3 - \frac{75}{24}x^2 + \frac{55}{12}x + 1$ , де  $a_3=0.541(6)$   $a_2=-3.125$   $a_1=4.58(3)$ ,  $a_0=1$ .

## 6. Екстраполювання функцій

Інтерполяційні формули застосовуються для знаходження значень функції для проміжних значень аргументів, відсутніх у таблиці. Проте за цими формулами можна знаходити і значення функцій для значень аргументу, що розташовані за межами таблиці.

Знаходження значень функції  $y=f(x)$  для значень аргументу  $x$ , що розташовані за межами таблиці, називається **екстраполюванням** або **екстраполяцією**.

Операція екстраполювання, взагалі кажучи, менш точна, ніж операція інтерполювання, і її слід застосовувати тоді, коли:

- функція біля кінців таблиці змінюється плавно;
- відстань від кінців таблиці, на якій екстраполюють, невелика (менша ніж відстань між сусідніми вузлами).

Приклади екстраполювання вперед і назад подано на рис. 7.4 і 7.5.

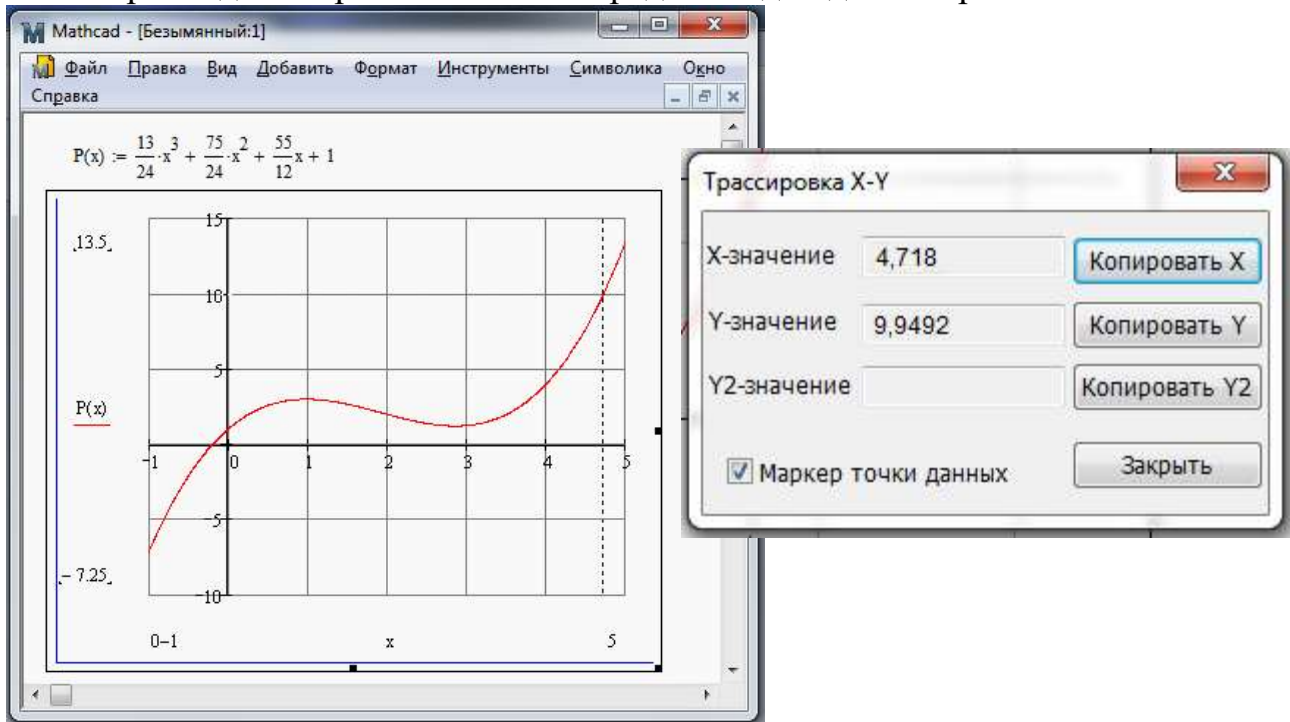


Рисунок 7.4 – Екстраполювання вперед

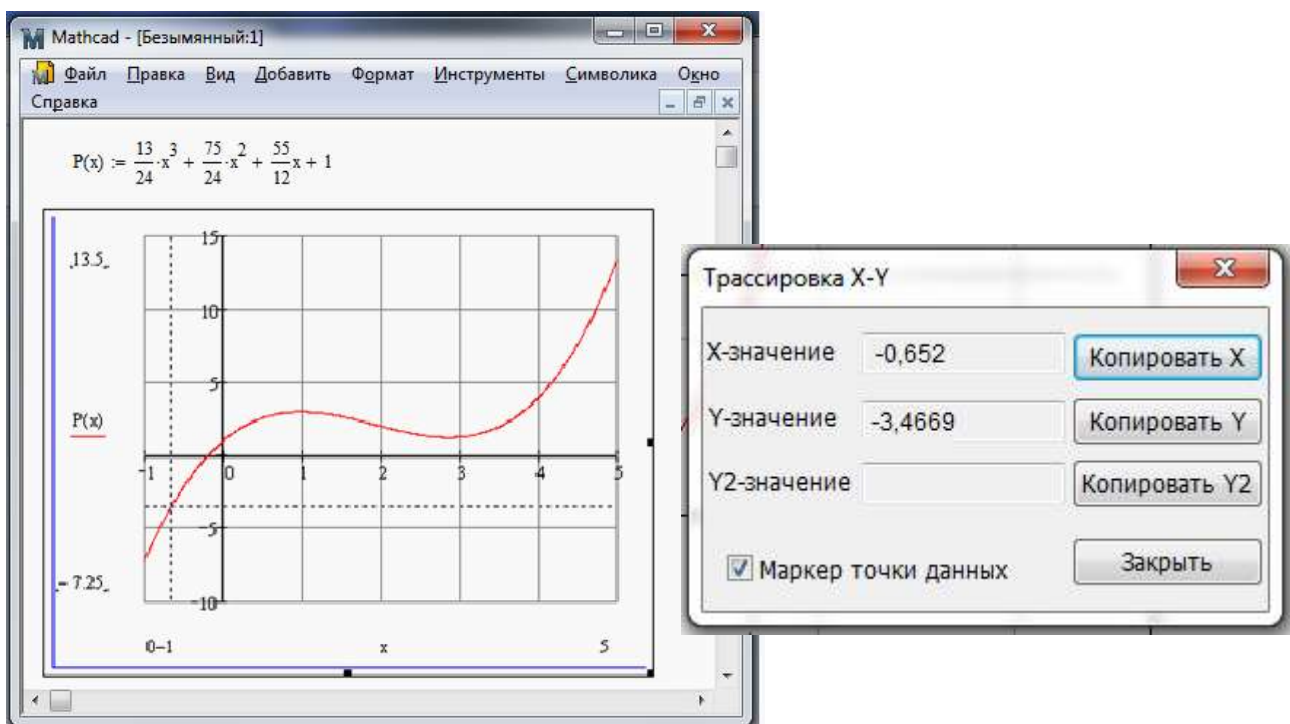


Рисунок 7.5 – Екстраполювання назад

## 7. Засоби інтерполювання функцій в системах комп'ютерної математики

Системи комп'ютерної математики мають відповідні функції для побудови інтерполяційних многочленів. Зокрема на рис. 6. показано як здійснюється інтерполяція функції однієї змінної в СКМ Mathcad.

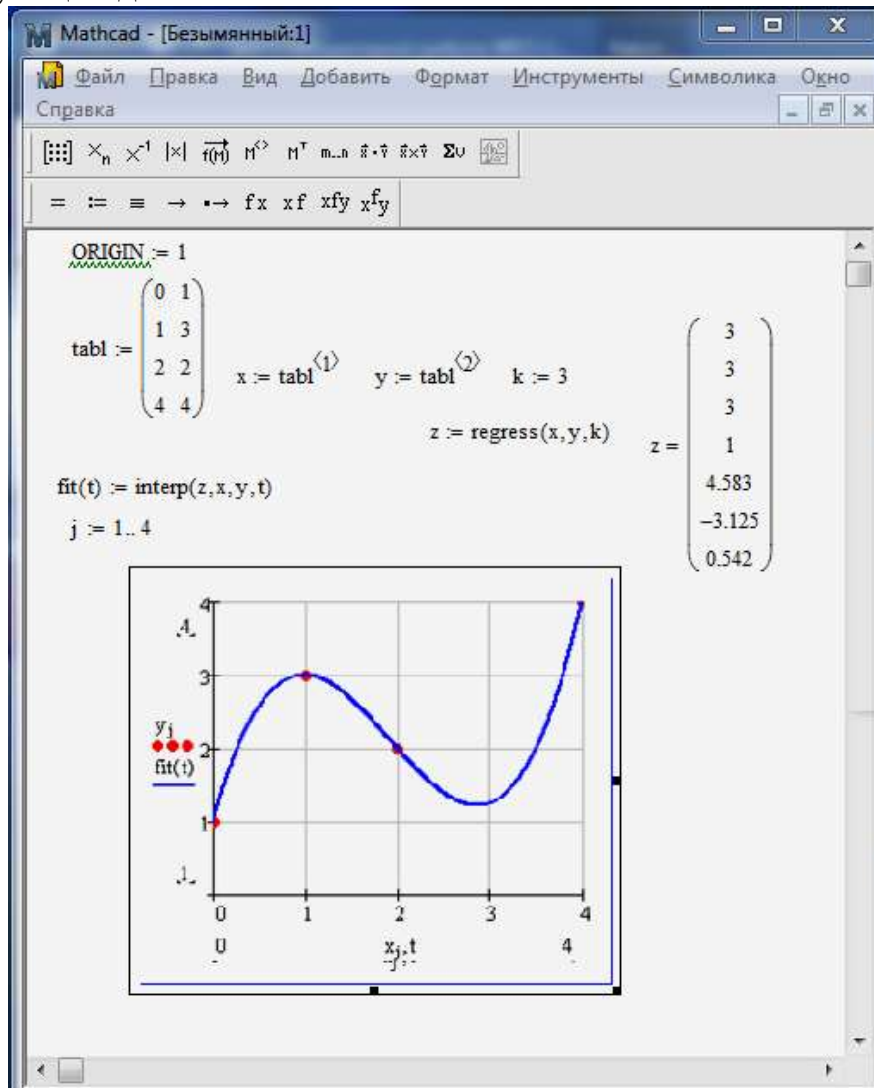


Рисунок 7.6 – Інтерполювання засобами пакету Mathcad

### II. Завдання до лабораторної роботи

1. Розв'язати задачу інтерполювання функції з прикладу 1, використовуючи співвідношення (6). Знайти інтерполяційний многочлен Лагранжа за допомогою системи Mathcad. За допомогою знайденого інтерполяційного многочлена провести екстраполявання вперед і назад (див. рис. 4 і 5) для  $x=5$  і  $x=-1.5$ .
2. Виконати індивідуальне завдання.
3. Підготувати звіт про виконання лабораторної роботи №13-14.

### III. Індивідуальні завдання

1. Розв'язати задачу інтерполювання функції, що задана таблично, використовуючи співвідношення (6). Знайти інтерполяційний многочлен

Лагранжа за допомогою системи Mathcad. За допомогою знайденого інтерполяційного многочлена провести екстраполявання вперед і назад для заданих точок .

Варіант		Вузли інтерполювання $x_i$				Точки екстраполявання для	
<b>1</b>	$x_i$	0	1	2	3	-1	3.5
	$f(x_i)$	3	2	1	5		
<b>2</b>	$x_i$	0	1	3	4	-1.5	4.5
	$f(x_i)$	2	1	3	0		
<b>3</b>	$x_i$	-1	0	1	2	-2	3
	$f(x_i)$	-1	2	1	4		
<b>4</b>	$x_i$	-1	0	2	3	-1.5	4
	$f(x_i)$	2	-1	0	5		
<b>5</b>	$x_i$	-2	-1	0	1	-2.5	1.5
	$f(x_i)$	3	0	1	6		
<b>6</b>	$x_i$	-2	0	1	2	-3	2.5
	$f(x_i)$	-3	2	-1	4		
<b>7</b>	$x_i$	-2	0	2	3	-2.5	3.5
	$f(x_i)$	3	2	1	5		
<b>8</b>	$x_i$	-3	-2	-1	1	-3.5	1.5
	$f(x_i)$	1	-2	0	3		
<b>9</b>	$x_i$	-3	-1	0	1	-4	2
	$f(x_i)$	3	-2	1	0		
<b>10</b>	$x_i$	-3	-2	-1	0	-4.5	-0.5
	$f(x_i)$	-1	2	0	5		
<b>11</b>	$x_i$	0	2	3	4	-1	4.5
	$f(x_i)$	2	0	-1	1		
<b>12</b>	$x_i$	-1	0	2	3	-1.5	4
	$f(x_i)$	3	2	1	5		

## Лабораторна робота № 8

**Тема:** «Наближення функцій методом найменших квадратів за допомогою СКМ»

Питання:

1. Постановка задачі наближення функцій.
2. Метод найменших квадратів наближення функцій.
3. Засоби наближення функцій в системах комп'ютерної математики.

### I. Теоретичний матеріал

#### 1. Засоби наближення функцій в системі комп'ютерної математики

##### Mathcad

За допомогою довідки пакету Mathcad (розділ Statistics) можна знайти основні функції, які надають можливість розв'язувати задачу наближення функцій однієї змінної.

#### 1.1. Лінійна регресія (Linear regression)

[slope\(vx,vy\) and intercept\(vx,vy\)](#)

These functions find the slope and the intercept of a set of data points.

##### Slope and intercept

**slope(vx,vy)** Slope of line that best fits data in **vx** and **vy**.

**intercept(vx,vy)** Intercept of line that best fits data in **vx** and **vy**.

##### Arguments:

**vx** is a vector of real data values. **vx** is a vector of real data values for the slope function. The values in **vx** correspond to the  $x$  values.

**vy** is a vector of real data values. These correspond to the  $y$  values. The number of elements is the same as **vx**.

##### Notes:

- The *slope* and *intercept* functions are useful not only when your data is inherently linear but [when it is exponential](#) as well.
- The equation of the line that best fits your data in **vx** and **vy** is  $y=mx+b$  where  $m$  is the result from *slope* and  $b$  is the result from *intercept*.

The example below illustrates how to do linear regression using the *slope* and *intercept* functions. For a short-cut function, use [line or medfit](#) instead.

На рис. 1. показано приклад використання лінійної регресії.

#### 1.2. Поліноміальна регресія (Polynomial regression)

[regress\(vx,vy,k\)](#)

Finds the polynomial that best fits a set of data points.

##### Polynomial regression

**regress(vx,vy,k)** Returns a vector which *interp* uses to find the  $k$ th order polynomial that best fits the  $x$  and  $y$  data values in **vx** and **vy**.



**interp(vs,vx,vy,x)** Returns interpolated  $y$  value corresponding to  $x$ . You get **vs** from *regress*.

**Arguments:**

**vx** is a vector of real data values in ascending order. These correspond to the  $x$  values.  
**vy** is a vector of real data values. These correspond to the  $y$  values. The number of elements is the same as **vx**.

**vs** is a vector generated by *regress*.

$k$  is a positive integer specifying the order of the polynomial you want to use to fit the data. Usually you'll want  $k < 5$ .

$x$  is the value of the independent variable at which you want to evaluate the regression curve.

**Notes:**

- *Regress* is useful when you have a set of measured  $y$  values corresponding to  $x$  values and you want to fit a single polynomial of any order to those  $y$  values.
- You should always use *interp* after using the *regress* function.
- Since *regress* tries to accomodate all your data points using a single polynomial, it will not work well when your data does not behave like a single polynomial. Use the loess function to perform a more localized regression.
- If you want to fit a surface to  $x$ ,  $y$ , and  $z$  data, use *regress* for multivariate polynomial regression.

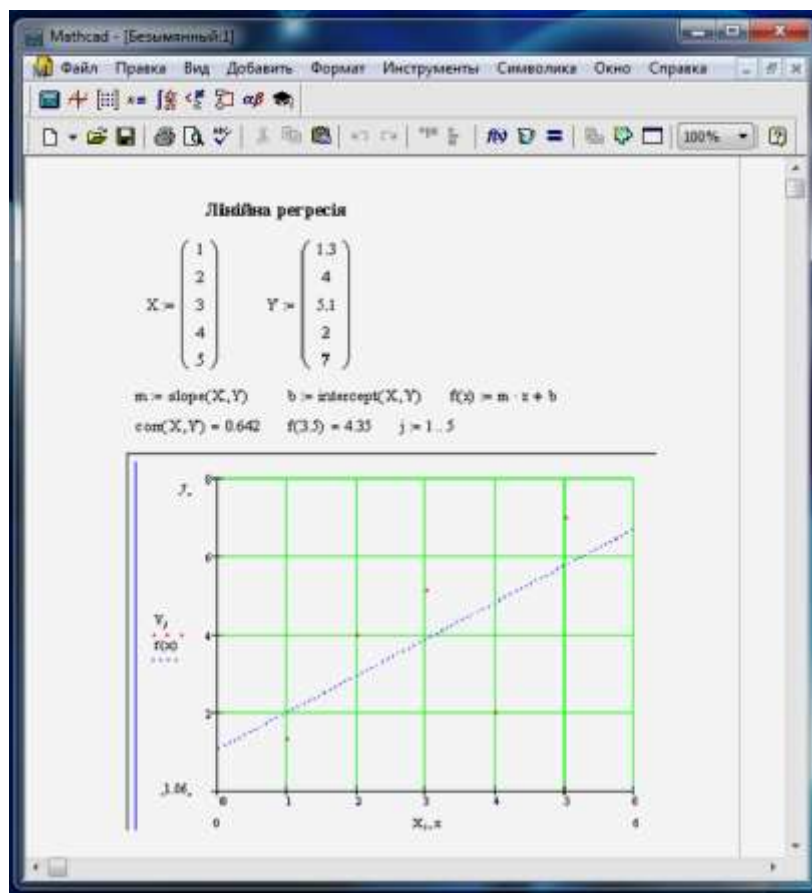


Рисунок 8.1 – Приклад використання лінійної регресії

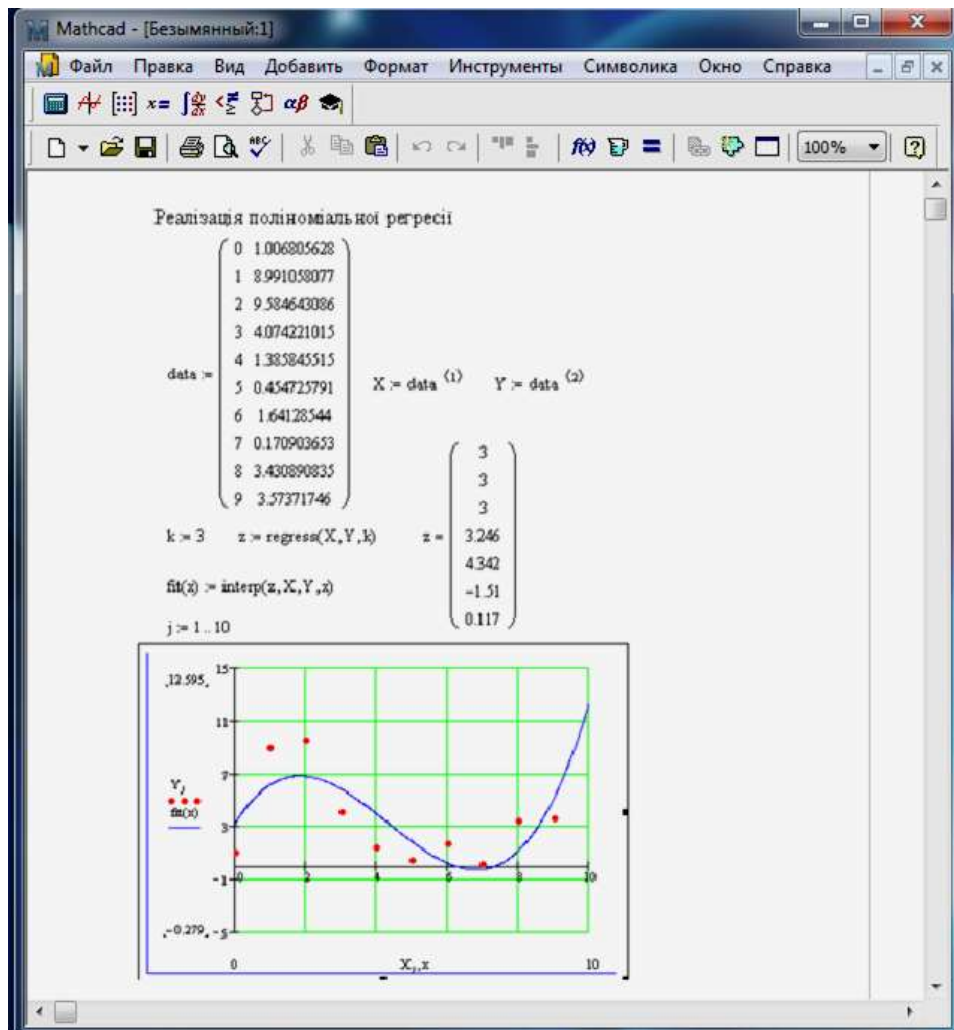


Рисунок 8.2 – Приклад використання поліноміальної регресії

### [loess\(vx,vy,span\)](#)

Finds the set of second order polynomials that best fit particular neighborhoods of data points.

**loess(vx,vy,span)** Returns a vector which *interp* uses to find a set of second order polynomials that best fit a neighborhood of the *x* and *y* data values in **vx** and **vy**. *span* controls the size of the neighborhood.

**interp(vs,vx,vy,x)** Returns interpolated *y* value corresponding to *x*. You get **vs** from *loess* or *regress*.

### **Arguments:**

**vx** is a vector of real data values in ascending order. These correspond to the *x* values.

**vy** is a vector of real data values. These correspond to the *y* values. The number of elements is the same as **vx**.

**vs** is a vector generated by *loess*.

*span* is a positive real number for specifying how big a neighborhood you want to use. Use larger values of *span* when the data behaves very differently over different ranges of *x*. A good default value is *span*=0.75.

*x* is the value of the independent variable at which you want to evaluate the regression curve.

## Multivariate polynomial regression

[regress\(Mxy,vz,k\)](#)

Finds the polynomial surface that best fits a set of points.

[loess\(Mxy,vz,span\)](#)

Finds the set of second order polynomials that best fits a particular neighborhoods of data points.

### 1.3. Спеціальні регресії (Specialized regression)

#### 1.3.1. Експоненціальна регресія

[expfit\(vx,vy,vg\)](#)

Finds the coefficients for an exponential curve that best approximates the data in **vx** and **vy**. **vg** contains guess values for the three coefficients.

**expfit(vx,vy,vg)** Returns a vector containing the coefficients for an exponential curve of the form  $ae^{(bx)}+c$  that best approximates the data in **vx** and **vy**. **vg** contains guess values for the three coefficients.

#### Arguments:

**vx** is a vector of real data values. The values in **vx** correspond to the  $x$  values.

**vy** is a vector of real data values. The values in **vy** correspond to the  $y$  values.

**vg** is a three-element vector of real guess values for the parameters  $a$ ,  $b$ , and  $c$  in the exponential equation:

**Note:** **vx** and **vy** must have the same number of elements.

#### 1.3.2. Тригонометрична регресія

[sinfit\(vx,vy,vg\)](#)

Finds the coefficients for a sine curve that best approximates the data in **vx** and **vy**. **vg** contains guess values for the three coefficients.

#### Sinusoidal regression

**sinfit(vx,vy,vg)** Returns a vector containing the coefficients for a sine curve of the form  $a*\sin(x+b)+c$  that best approximates the data in **vx** and **vy**. **vg** contains guess values for the three coefficients.

#### Arguments:

**vx** is a vector of real data values. The values in **vx** correspond to the  $x$  values.

**vy** is a vector of real data values. The values in **vy** correspond to the  $y$  values.

**vg** is a three-element vector of real guess values for the parameters  $a$ ,  $b$ , and  $c$  in the sinusoidal equation:

#### 1.3.3. Логарифмічна регресія

[logfit\(vx,vy,vg\)](#)

Finds the coefficients for a logarithmic curve that best approximates the data in **vx** and **vy**. **vg** contains guess values for the three coefficients.

**logfit(vx,vy,vg)** Returns a vector containing the coefficients for a logarithmic curve of the form  $a*\ln(x+b)+c$  that best approximates the data in **vx** and **vy**. **vg** contains guess values for the three coefficients.

#### Arguments:

**vx** is a vector of real data values. The values in **vx** correspond to the  $x$  values.

**vy** is a vector of real data values. The values in **vy** correspond to the *y* values.

**vg** is a three-element vector of real guess values for the parameters *a*, *b*, and *c* in the logarithmic equation:

**Note:** **vx** and **vy** must have the same number of elements.

### 1.3.4. Степенева регресія

[pwrfit\(vx,vy,vg\)](#)

Finds the coefficients for a power curve that best approximates the data in **vx** and **vy**. **vg** contains guess values for the three coefficients.

**pwrfit(vx,vy,vg)** Returns a vector containing the coefficients for a power curve of the form  $a*x^b+c$  that best approximates the data in **vx** and **vy**. **vg** contains guess values for the three coefficients.

#### Arguments:

**vx** is a vector of real data values. The values in **vx** correspond to the *x* values.

**vy** is a vector of real data values. The values in **vy** correspond to the *y* values.

**vg** is a three-element vector of real guess values for the parameters *a*, *b*, and *c* in the power equation:

## II. Завдання до лабораторної роботи

1. Розглянути теоретичний матеріал (презентація лекції в системі електронного навчання: [ias.cdtu.edu.ua/moodle19-test](http://ias.cdtu.edu.ua/moodle19-test) ).

2. Розв'язати задачу наближення функцій за методом найменших квадратів з прикладу 1 лекції, використовуючи лінійну і квадратичну функції регресії.

#### Приклад 1.

Нехай дослідним шляхом знайдено таблицю значень певної залежності. За допомогою МНК знайти лінійну функцію і квадратичну функцію, що якнайкраще наближають шукану залежність.

$x_i$	1	2	3	4	5	6
$y_i$	5,2	6,3	7,1	8,5	9,2	10,0

#### Розв'язування

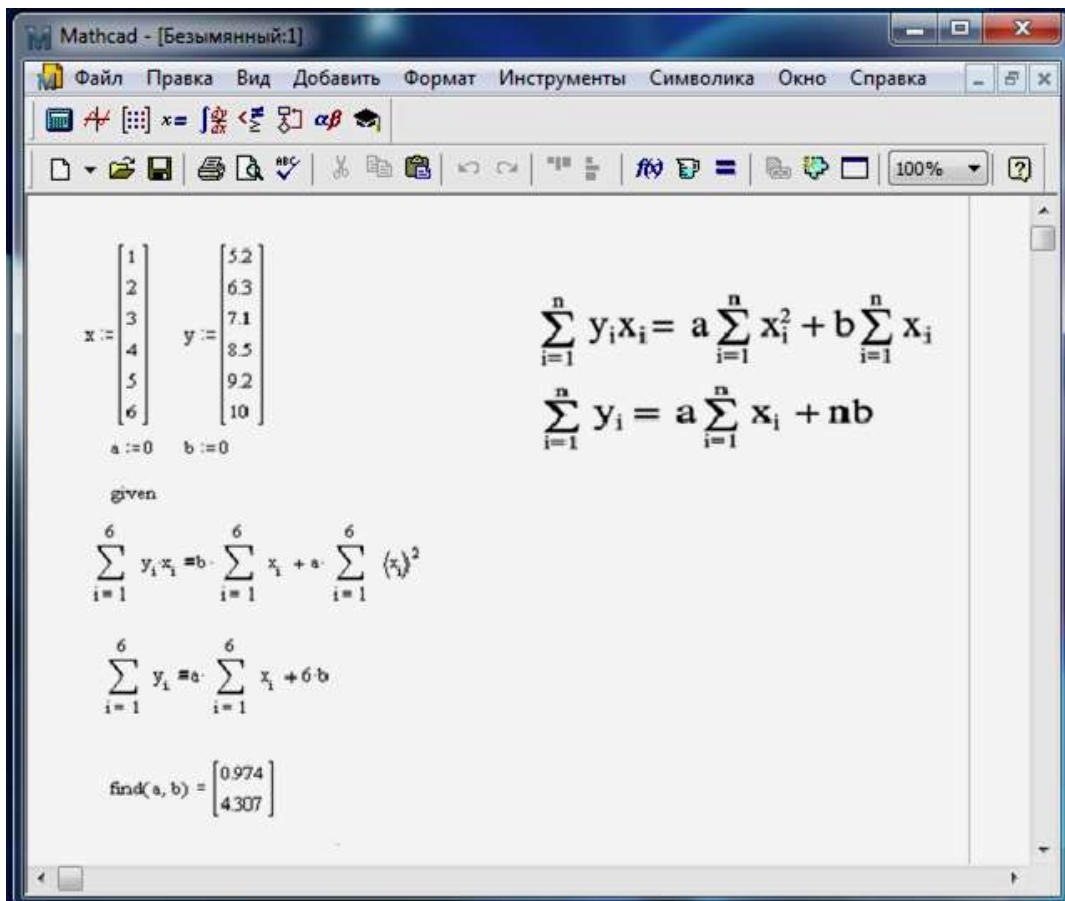


Рисунок 8.3

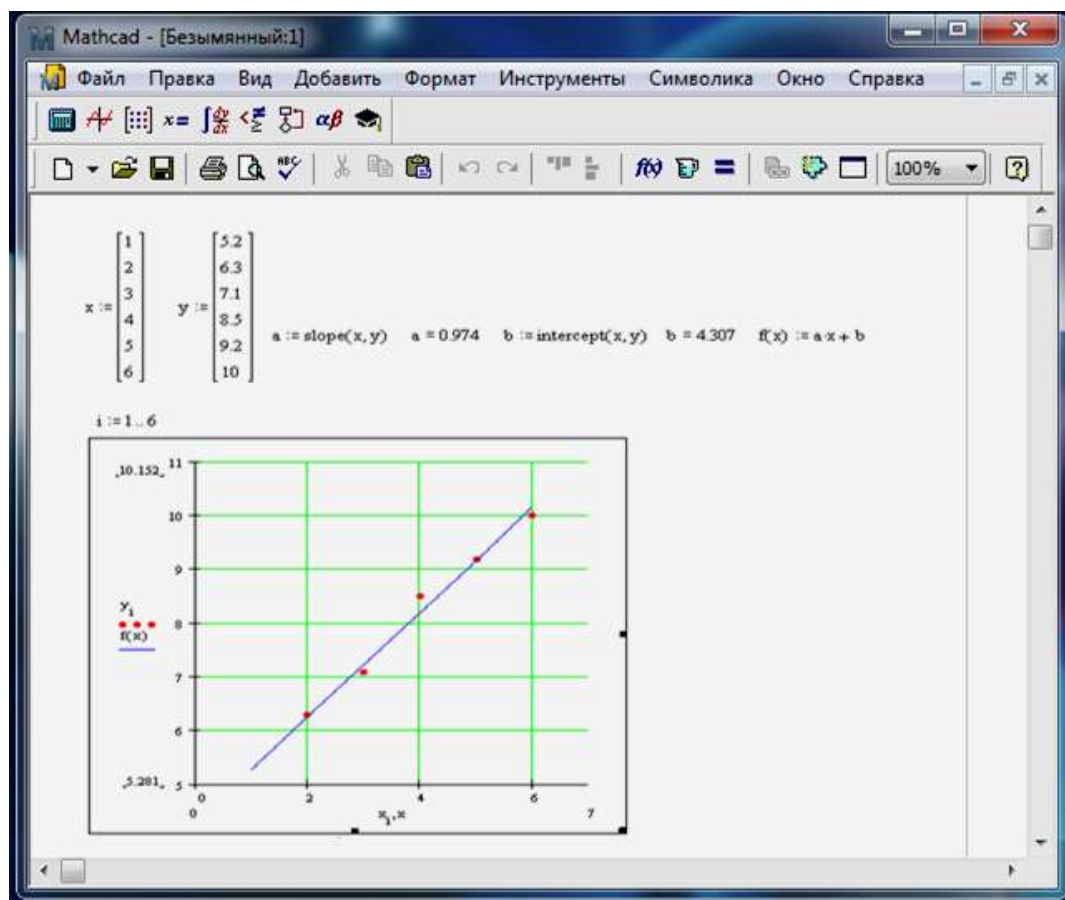


Рисунок 8.4

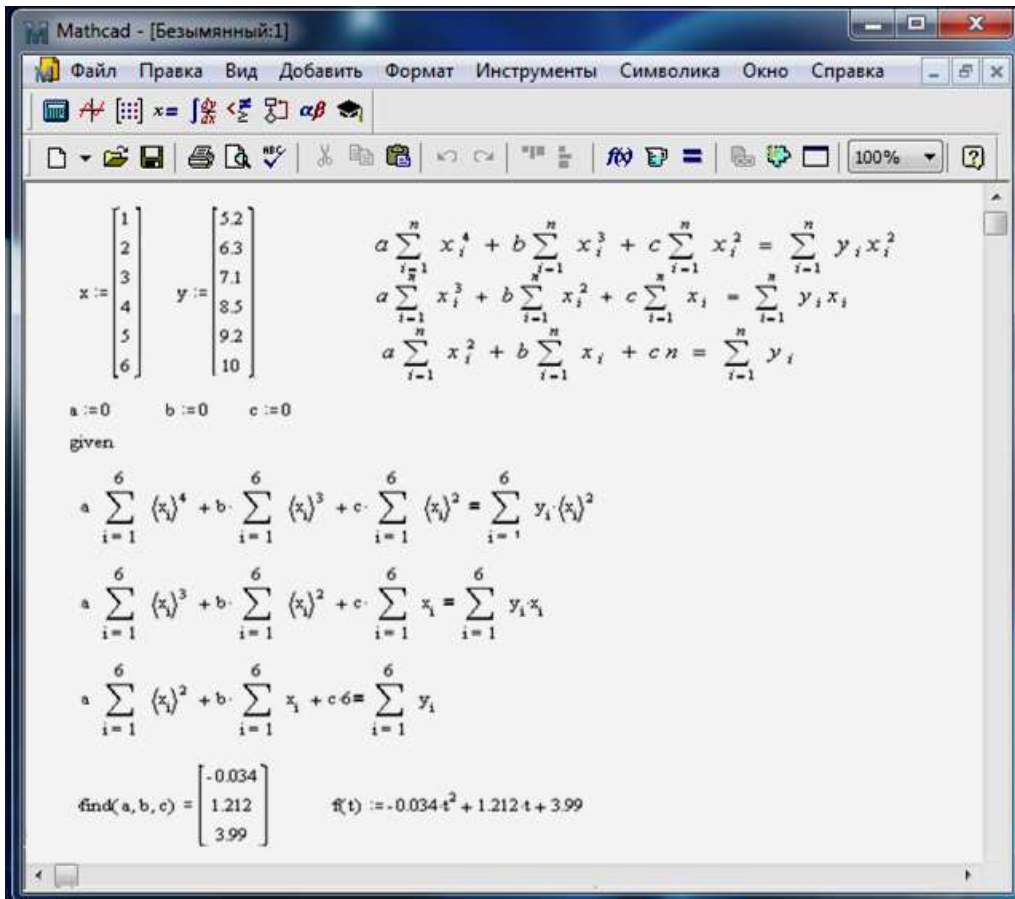


Рисунок 8.5

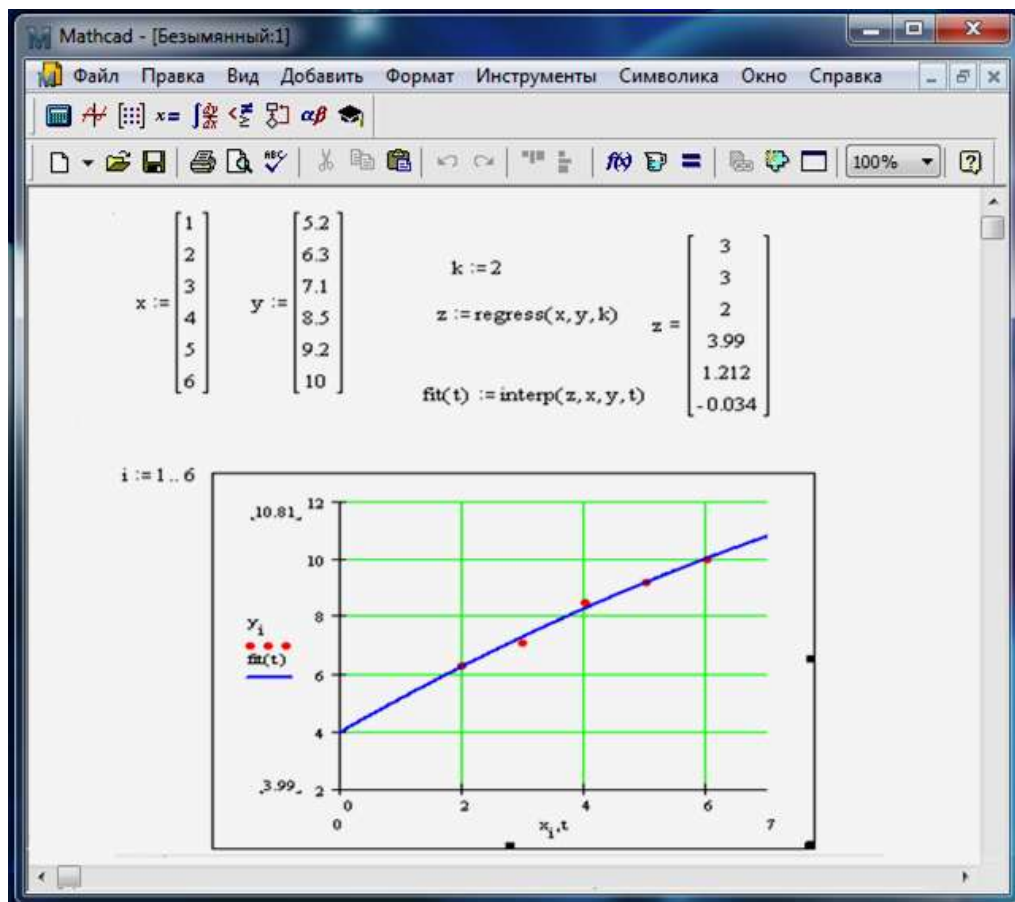


Рисунок 8.6

3. Для таблично заданої функції з прикладу 1 побудувати функцію лінійної регресії, поліноміальної регресії (не вище 5-го степеня), експоненціальної, логарифмічної, степеневі, тригонометричної регресій.

4. Виконати індивідуальне завдання.

5. Оформити звіт про виконання завдань до лабораторної роботи.

### III. Індивідуальні завдання

1. Для таблично заданої функції знайти і побудувати наближені функції за методом найменших квадратів: функцію лінійної регресії, поліноміальної регресії (не вище 5-го степеня), експоненціальної, логарифмічної, степеневі, тригонометричної регресій:

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
0	13	-7	17	-16	4	-9	3	20	-8	11
0.5	1	1	4	7	-1	4	-8	3	4	-24
1	-3	2	2	-3	1	1	10	-9	7	27
1.5	4	13	5	6	-11	-15	7	7	-5	-25
2	3	1	-9	-4	3	4	5	4	9	9
2.5	-14	15	1	2	-9	5	-4	-9	2	-21
3	11	-6	-1	-2	15	14	3	6	5	10
3.5	2	1	3	5	-4	15	8	10	-2	-12
4	3	8	7	-2	1	-1	17	14	9	10
4.5	2	-7	1	1	-6	1	-12	-30	1	-20

2. Для отриманих результатів зробити висновок про характер залежності, яка задана таблично.

## Лабораторна робота № 9

### Тема: «Основи програмування в СКМ Mathcad»

Питання:

1. Загальні положення щодо програмування у СКМ.
2. Процедурне програмування в СКМ Mathcad.
3. Приклади програм в СКМ Mathcad.
4. Розв'язування задач.

### I. Теоретичний матеріал

#### 1. Загальні положення щодо програмування у СКМ

Більшість СКМ створювались для того, щоб позбавити масового користувача від проблем програмування при розв'язуванні математичних задач. Набір вбудованих операторів і функцій у таких системах настільки великий, що значну кількість математичних задач можна розв'язати без їх програмування. Наприклад, так обраховуються похідні, інтеграли, границі функцій, їх екстремуми, розв'язуються системи різноманітних рівнянь.


Але досить багато задач, де відмова від звичайного процедурного програмування призводить до ускладнення їх розв'язання. Особливо це стосується таких задач, де необхідно створювати матриці спеціального виду, створювати інтерактивну взаємодію з програмою, створювати власні елементи інтерфейсу, реалізувати нові методи розв'язування задач певного класу, здійснювати графічні побудови тощо. У решті решт користувач сам повинен вирішити, потрібно йому використовувати елементи програмування у своїх задачах чи ні. Тому практично всі СКМ мають власні засоби програмування і мови програмування.

**Програма** – просто вираз, складений з більше чим однієї інструкції. Незважаючи на основну схожість між програмами і простими виразами, програми пропонують дві відмінних переваги:

- якщо використовувати керуючі структури подібні до циклів або умовному переходу, програма може стати набагато гнучкою, ніж простий вираз;
- програма, складена з декількох простих кроків, - часто набагато простіше.

#### 2. Процедурне програмування в СКМ Mathcad

Починаючи з версії Mathcad Plus 6.0, до систем Mathcad були введені традиційні засоби програмування. Вони задаються у вигляді оригінальних програмних блоків, які виділяються відрізками ліній.

Для активізації засобів програмування у пакеті Mathcad необхідно скористатися математичною інструментальною панеллю (рис. 9.1) і обрати розділ «Programming» (кнопка ) (рис. 9.2).



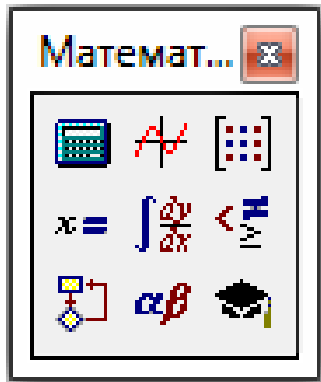


Рисунок 9.1 – Панель інструментів Mathcad

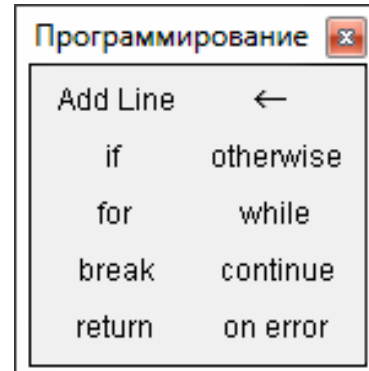


Рисунок 9.2 – Панель програмування

Кнопки панелі «Programming» надають швидкий доступ до операторів, що використовуються у програмах в середовищі Mathcad. Для того щоб визначити, що робить відповідна кнопка, треба перемістити курсор в основне вікно робочого листа, і не чіпати курсор. Тоді будуть доступні такі допоміжні функції:

- підказка, яка показує заголовок кнопки;
- опис кнопки з'являється у рядку повідомлення.

Основні оператори панелі «Programming» («Програмування»):

[Add Line](#) - створює і при необхідності розширює вертикальну лінію, справа від якої описується у шаблонах тіло програмного блоку;



- символ локального присвоювання (у тілі модуля);

[if](#) – умовний оператор;

[for](#) – оператор задання циклу з параметром (циклу з фіксованою кількістю повторів);

[while](#) – оператор задання циклу типу «поки» або «циклу з передмовою», який виконується доти, поки виконується деяка умова;

[otherwise](#) – оператор альтернативного вибору, який зазвичай використовується з оператором if;

[continue](#) – оператор продовження, який використовується для продовження роботи після переривання програми; найчастіше використовується сумісно з операторами циклів для забезпечення повернення на початок циклу після його переривання;

[return](#) – оператор повернення значення операнда, який йде за ним, з циклу до його завершення;

[break](#) – оператор переривання роботи програми, частіше за все використовується для переривання циклів;

[on error](#) – оператор обробки помилки, який дозволяє створювати конструкції обробників помилок виду: S1 on error S2, в якому якщо при виконанні виразу S1 виникає помилка, то виконується вираз S2.

#### **Зауваження.**

1. Не треба вводити службові слова операторів з клавіатури. Їх необхідно вибрати на панелі «Programming».

2. Якщо Ви використовуєте більше ніж один оператор «if» , то команда «otherwise» виконується тільки тоді, коли всі умови помилкові.

3. **Цикл** - інструкція програми, яка змушує одну або більше інструкцій (тіло циклу) виконуватися неодноразово, поки певна умова (стан) не порушиться.

4. Для того щоб аварійно зупинити виконання циклу, треба натиснути клавішу **Esc**.

## II. Завдання до лабораторної роботи

### 3. Приклади програм в СКМ Mathcad

Приклади використання основних операторів програмування в пакеті Mathcad.



## PROGRAMMING

### Using For Loops

This QuickSheet provides some examples of using **For Loops**.

#### Simple for loop, increasing or decreasing the range variable by 1

**Example 1:** Finding the sum of the first n integers (equivalent to using Mathcad's summation operator).

$$\text{sum}(n) := \left| \begin{array}{l} s \leftarrow 0 \\ \text{for } i \in 1..n \\ s \leftarrow s + i \end{array} \right. \quad \text{sum}(5) = 15$$

**Example 2:** Flattening a matrix into a long column vector, starting with the last column.

$$M := \text{identity}(3)$$

$$\text{flatten}(M) := \left| \begin{array}{l} v \leftarrow M \langle \text{cols}(M)-1 \rangle \\ \text{for } j \in \text{cols}(M) - 2..0 \quad \text{if } \text{cols}(M) > 1 \\ v \leftarrow \text{stack}(v, M \langle j \rangle) \\ v \end{array} \right.$$

$$\text{flatten}(M)^T = (0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0)$$

**Example 3:** Finding the iterated product of the elements of a vector (equivalent to using Mathcad's iterated product operator).

$$v := \pi e \frac{1 + \sqrt{5}}{2}^T$$

$$\text{prod}(v) := \left| \begin{array}{l} p \leftarrow 1 \\ \text{for } i \in v \\ p \leftarrow p \cdot i \end{array} \right.$$

**Example 4:** Filling a vector with values in a nonintegral range.

$$\text{vec}(\text{low}, \text{upp}, \text{incr}) := \left| \begin{array}{l} \text{ind} \leftarrow 0 \\ \text{for } i \in \text{low}, \text{low} + \text{incr}.. \text{upp} \\ \left| \begin{array}{l} v_{\text{ind}} \leftarrow i \\ \text{ind} \leftarrow \text{ind} + 1 \end{array} \right. \\ v \end{array} \right.$$

$$\text{vec}(1, 2, 0.2)^T = ( 1 \quad 1.2 \quad 1.4 \quad 1.6 \quad 1.8 \quad 2 )$$

### Nested for loop

**Example 5:** Filling a matrix over two equal, nonintegral ranges.

$$f(x, y) := x - y$$

$$\text{Mat}(\text{low}, \text{upp}, \text{incr}) := \left| \begin{array}{l} r \leftarrow 0 \\ \text{for } i \in \text{low}, \text{low} + \text{incr}.. \text{upp} \\ \left| \begin{array}{l} c \leftarrow 0 \\ \text{for } j \in \text{low}, \text{low} + \text{incr}.. \text{upp} \\ \left| \begin{array}{l} M_{r,c} \leftarrow f(i, j) \\ c \leftarrow c + 1 \end{array} \right. \\ r \leftarrow r + 1 \end{array} \right. \\ M \end{array} \right.$$

$$\text{Mat}(0,0.4,0.1) = \begin{pmatrix} 0 & -0.1 & -0.2 & -0.3 & -0.4 \\ 0.1 & 0 & -0.1 & -0.2 & -0.3 \\ 0.2 & 0.1 & 0 & -0.1 & -0.2 \\ 0.3 & 0.2 & 0.1 & 0 & -0.1 \\ 0.4 & 0.3 & 0.2 & 0.1 & 0 \end{pmatrix}$$

### Defining an iterative function

**Example 6:** Factorial function (equivalent to Mathcad's factorial operator).

$$F(n) := \left| \begin{array}{l} f \leftarrow 1 \\ \text{for } i \in 1..n \\ \quad f \leftarrow f \cdot i \\ f \end{array} \right. \quad F(6) = 720$$

**Example 7:** Fibonacci function/sequence, also using conditional.

$$G(n) := \left| \begin{array}{l} g \leftarrow 1 \text{ if } n \leq 1 \\ \text{otherwise} \\ \quad \left| \begin{array}{l} h \leftarrow 1 \\ g \leftarrow 1 \\ \text{for } k \in 2..n \\ \quad \left| \begin{array}{l} \text{tmp} \leftarrow h + g \\ h \leftarrow g \\ g \leftarrow \text{tmp} \end{array} \right. \end{array} \right. \\ g \end{array} \right.$$

$$G(7) = 21$$

## Nested While loops

**Example 5:** Finding the first element of a square matrix that is suitably close to zero.

$$i := 0..15 \quad j := 0..15 \quad m_{i,j} := \sin(i) + \cos(j)$$

	0	1	2	3	4	5
0	1	0.54	-0.416	-0.99	-0.654	0.284
1	1.841	1.382	0.425	-0.149	0.188	1.125
2	1.909	1.45	0.493	-0.081	0.256	1.193
3	1.141	0.681	-0.275	-0.849	-0.513	0.425
4	0.243	-0.217	-1.173	-1.747	-1.41	-0.473
5	0.041	-0.419	-1.375	-1.949	-1.613	-0.675
6	0.721	0.261	-0.696	-1.269	-0.933	$4.247 \cdot 10^{-3}$
m = 7	1.657	1.197	0.241	-0.333	$3.343 \cdot 10^{-3}$	0.941
8	1.989	1.53	0.573	$-6.342 \cdot 10^{-4}$	0.336	1.273
9	1.412	0.952	$-4.028 \cdot 10^{-3}$	-0.578	-0.242	0.696
10	0.456	$-3.719 \cdot 10^{-3}$	-0.96	-1.534	-1.198	-0.26
11	$9.793 \cdot 10^{-6}$	-0.46	-1.416	-1.99	-1.654	-0.716
12	0.463	$3.729 \cdot 10^{-3}$	-0.953	-1.527	-1.19	-0.253
13	1.42	0.96	$4.02 \cdot 10^{-3}$	-0.57	-0.233	0.704
14	1.991	1.531	0.574	$6.149 \cdot 10^{-4}$	0.337	1.274
15	1.65	1.191	0.234	-0.34	$-3.356 \cdot 10^{-3}$	0.934

```

z(m, δ) :=
  k ← 0
  while k < rows(m)
    i ← k
    j ← 0
    while (i ≥ 0) · (|mi,j| ≥ δ)
      i ← i - 1
      j ← j + 1
    k ← k + 1 if i < 0
    break if i ≥ 0
  (
    i
    j
    mi,j
  )

```

**Note:** The matrix is searched here in bands, starting in the upper left-hand corner and sweeping downward.

$$z(m, 0.0001) = \begin{pmatrix} 11 \\ 0 \\ 9.793 \times 10^{-6} \end{pmatrix}$$

## Defining an Iterative Function

**Example 6:** Factorial function; again, equivalent to Mathcad's built-in factorial operator.

$$F(n) := \begin{array}{l} f \leftarrow 1 \\ \text{while } n \leftarrow n - 1 \\ \quad f \leftarrow f \cdot (n + 1) \\ f \end{array} \quad F(6) = 720$$

Alternative formulation using break statement:

$$F(n) := \begin{array}{l} f \leftarrow n \\ \text{while } 1 \\ \quad \begin{array}{l} f \leftarrow f \cdot (n - 1) \\ n \leftarrow n - 1 \\ \text{break if } n = 1 \end{array} \\ f \end{array} \quad F(6) = 720$$

### III. Індивідуальні завдання

1. Для заданого ряду знайти його суму із заданою точністю  $\varepsilon > 0$ , використовуючи засоби програмування:

$$\begin{array}{llll}
1) \sum_{n=1}^{\infty} \frac{n}{2^n}; & 2) \sum_{n=1}^{\infty} \frac{n^2}{3^n}; & 3) \sum_{n=1}^{\infty} \frac{1}{(2n-1)!}; & 4) \sum_{n=1}^{\infty} \frac{(n+1)!}{10^n}; \\
5) \sum_{n=1}^{\infty} \frac{1+n}{1+n^2}; & 6) \sum_{n=1}^{\infty} \frac{1}{\sqrt{n(n+1)}}; & 7) \sum_{n=1}^{\infty} \frac{1}{(\ln n)^n}; & 8) \sum_{n=1}^{\infty} \left(\frac{2}{n}\right)^n; \\
9) \sum_{n=1}^{\infty} 2^n \sin \frac{\pi}{3^n}; & 10) \sum_{n=1}^{\infty} \sin \frac{1}{n}; & 11) \sum_{n=1}^{\infty} \frac{(n-1)^n}{n^{n+1}}; & 12) \sum_{n=1}^{\infty} \frac{1}{\sqrt{4n^4 - n^3 + n^2 + 1}}.
\end{array}$$

2. Для заданих функцій, за допомогою засобів програмування, побудувати таблицю їх значень на проміжку  $[a; b]$  з кроком  $h > 0$  і за цими значеннями побудувати схему графіка:

1) $y = \log_{\frac{1}{3}} ( 9 - x^2 )$ ;	2) $y = \frac{1}{x+2} - \frac{1}{x-2}$ ;
3) $y = \left(\frac{x+1}{x}\right)^{\frac{4}{5}}$ ;	4) $y = \left(\frac{x}{x+1}\right)^5$ ;
5) $y = (1 - 2^x)^2$ ;	6) $y = \left(\frac{x-2}{x+2}\right)^2$ ;
7) $y = \frac{x^4}{x^3 - 8}$ ;	8) $y = \frac{x^2 - 3}{x}$ ;
9) $y = \log_{x^2 - 3x + 2} 2$ ;	10) $y = \sqrt{(x^2 - 4)^2} + \sqrt{(x^2 - 2)^2}$ ;
11) $y = \left  \frac{x-3}{x-4} \right $ ;	12) $y = \cos x - \ln \cos x$ .

**Примітка.** У програмі зробити перевірку на виконання некоректних дій (наприклад ділення на 0) і якщо вони мають місце, то виводити відповідне повідомлення і продовжити роботу програми з наступними даними.

#### IV. Завдання для самостійного виконання

1. Для наступних задач побудувати програми в системі Mathcad і розв'язати їх при заданих вхідних даних:

1. Для заданих трьох дійсних чисел  $a, b, c$  знайти найбільше.
2. Для заданих натуральних чисел  $m$  і  $n$  знайти найбільший спільний дільник за алгоритмом Евкліда.
3. Реалізувати схему Горнера для знаходження значень многочлена  $P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$ .
4. Знайти всі досконалі числа, які менше заданого числа  $n$ . Досконалим називається число, яке дорівнює сумі своїх дільників крім самого себе.

5. Дане натуральне число  $n$ . З'ясувати скільки додатних елементів містить квадратна матриця  $A = (a_{ij})_{i=1, \dots, n, j=1, \dots, n}$ , якщо  $a_{ij} = \sin(i^2 + n)$ .

6. Реалізувати метод дихотомії для наближеного знаходження коренів рівняння  $f(x) = 0$  на заданому відрізку  $[a; b]$  з точністю  $\varepsilon > 0$ .



## Лабораторна робота № 10

Тема: «Інтерфейс, типи даних і математичні функції в SMath Studio»

Питання:

1. Можливості SMath Studio
2. Можливості SMath Studio для КПК
3. Основні елементи інтерфейсу СКМ SMath Studio.
4. Встановленн програми SMath Studio
5. Основи роботи в програмі
6. Довідкова інформація

### I. Теоретический материал

#### 1. Можливості SMath Studio

SMath Studio – безкоштовна програма для обчислення математичних виразів і побудови графіків функцій. Робота з інтерфейсом програми нагадує роботу із звичайним аркушем паперу, так як всі математичні вирази в ній записуються не в рядок текстом, а в графічному, зручному для людини, вигляді (рис. 10.1).

Перша публічна бета-версія програми була створена в 2005 році для кишенькових комп'ютерів на мові C # під платформу Microsoft .NET Compact Framework 1.0. На поточний момент SMath Studio має версії для декількох платформ: КПК, комунікаторів, смартфонів, персональних комп'ютерів на базі операційних систем Windows та Linux і логічно розділена на дві програми: Handheld (кишенькова) і Desktop (настільна), які відповідають типу підтримуваних платформ.

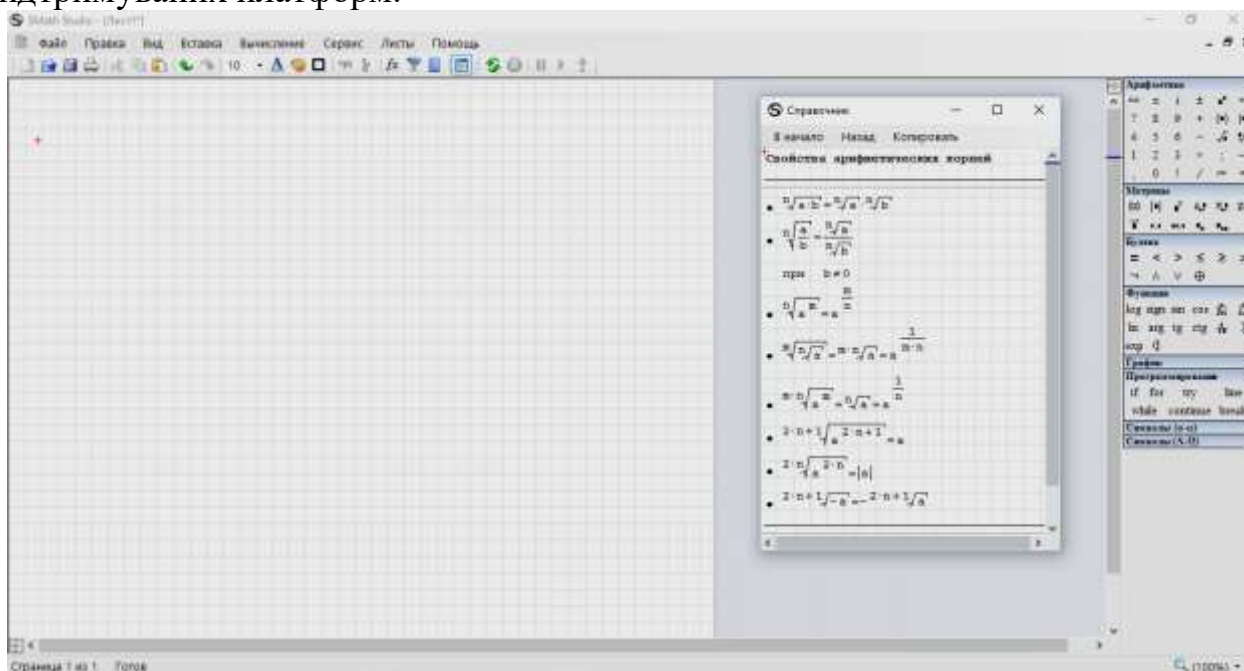


Рисунок 10.1 – Робоче вікно SMath Studio

SMath Studio має наступні можливості:

1. Підтримка наступних типів даних: системи, матриці, вектори, комплексні числа, дроби.

2. Підтримка наступних операцій і функцій: додавання, віднімання, множення (скаляре і векторне), ділення, знаходження факторіала, зведення в ступінь, робота з коренями, визначення модуля, тригонометричні функції, зворотні тригонометричні функції, гіперболічні функції, зворотні гіперболічні функції, логарифми, визначення сигнатури, обчислення аргументу числа, визначення детермінанта (визначника) матриці, транспонування матриць і векторів, обчислення мінорів матриці, обчислення алгебраїчних доповнень матриці, обчислення сліду матриці, визначення рангу, переклад комплексних чисел з алгебраїчного в тригонометричний вид, диференціювання, інтегрування, рішення рівнянь і багато іншого.

3. Відображення графіків функцій відносно змінної "x" (ікс).

4. Доступні символічні обчислення.

5. Можлива вставка в лист програми текстових областей. Причому при введенні виразу в лист, не користуючись меню програми ("Меню" >> "Вставка" >> "Текстова область"), програма сама визначає що вводиться текст або формула.

6. Експорт створених листів у форматі html – створення веб-сторінок.

7. Підтримка роботи з параметрами (константи і параметри користувача).

8. Підтримка роботи з функціями (вбудовані і призначені для користувача функції).

9. Необмежений розмір листів і їх кількість.

10. Копіювання, вирізання, вставка і видалення виразів або їх частин.

11. Робота з історією змін (скасувати, повернути).

12. Можливість зберігання і відкриття створених листів, працюючи з власним форматом файлів.

13. Автоматичне зберігання налаштувань при виході з програми.

14. Можливість зміни розміру шрифту виразу, а також його кольору і кольору його фону.

15. Підтримка роботи на пристроях з будь-яким розширенням і орієнтацією екрану (починаючи з 240x240 пікселів і вище).

16. Програма підтримує декілька мов інтерфейсу (англійська, німецька, іспанська, португальська (бразилія), хорватська, сербський, російська та українська).

## **2. Можливості SMath Studio для КПК**

Більшість користувачів кишенькових комп'ютерів використовують їх як записні й телефонні книжки, а також для читання книг. У комплекті майже всіх КПК як на основі Windows Mobile так і на базі Palm OS є набір офісних додатків, що включає електронні таблиці, які з успіхом можна використовувати для досить серйозних розрахунків.

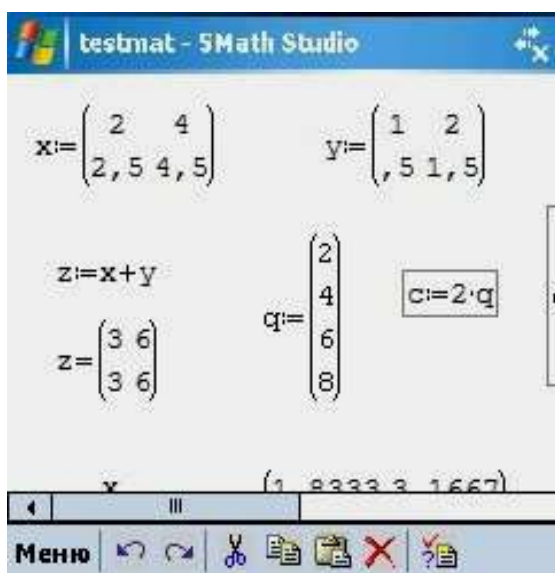


Рисунок 10.2 – Приклад введення виразів

Незважаючи на це для КПК можна знайти і більш спеціальні математичні програми, і, що найголовніше, серед них є і безкоштовні. Що ж стосується популярної платформи Windows Mobile, то тут майже всі математичні пакети представляли собою або взагалі прості наукові калькулятори, або калькулятори з графічними можливостями. Винятком є програма SMath Studio.

Вона надає можливість здійснювати досить багато математичних операцій, як над числами, так і з рівняннями.

У програму включені можливості роботи з матрицями, побудови графіків функцій, а також звичайні покрокові обчислення з використанням послідовного набору формул.

### 3. Основні елементи інтерфейсу СКМ SMath Studio

Інтерфейс SMath Studio дуже схожий на інтерфейс популярного математичного пакета Mathcad. Курсор має форму хрестика, шрифт і графічне представлення чисел і формул також перейнято у Mathcad (рис. 10.3).



Рисунок 10.3 – Робоче вікно SMath Studio на КПК

Для полегшення введення на КПК, програма має панельки з шаблонами, які надають можливість швидко вводити необхідні вирази. Зважаючи на обмеженість місця на екрані КПК схожі елементи об'єднані в одну клітинку, а прокрутка здійснюється рухом стіла.

Для присвоєння змінній значення або виразу необхідно ввести двокрапку, після чого програма сама зрозуміє наміри користувача і додасть знак "=". Для того, щоб отримати відповідь, достатньо записати на робочому листі змінну, якій присвоєно вираз, і написати після неї знак "=".

Так як курсор під час сеансу роботи можна розмістити в будь-якому місці екрану, то варто врахувати особливості, які перекочували в цю програму з Mathcad: всі визначення змінних справедливі тільки для виразів що знаходяться нижче них.

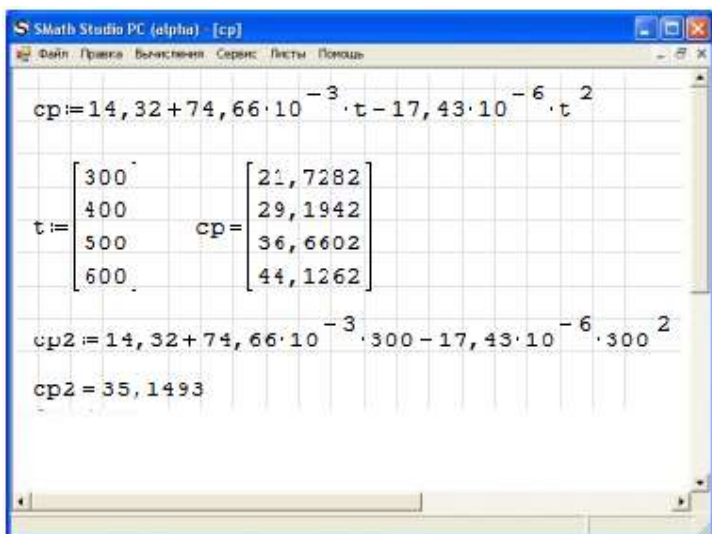


Рисунок 10.4 – Робота з SMath Studio

Прості обчислення – не єдине, що може SMath Studio. Завдяки можливості обробляти матриці, процес обчислення одного і того ж виразу при зміні змінних відчутно прискорюється. Наприклад, для виразу  $x = 12 * t$ , при  $t = 100, 200, 300, 400$ , набирати чотири різних вирази не потрібно, точно такж як і чотири рази перевизначати змінну "t". Досить визначити "t" як матрицю з чотирьох елементів і повчити відповідь відразу для чотирьох значень (рис. 4).

На жаль, поки програма позбавлена функції поелементних операцій над матрицями, тому якщо необхідно скористатися більш складною формулою начебто  $Ср = 14,32 + 74,66 * 10^{-3} * t - 17,43 * 10^{-6} * t^2$ , діяти доведеться як на калькуляторі, або перебиваючи вираз або перевизначаючи змінну "t". В іншому випадку результати розрахунку будуть нерівними.

Крім цього, програма обраховує символіні обчислення, операції з тригонометричними функціями, диференціювати функції (рис. 10.5). З більш детальним списком можливостей можна ознайомитися на офіційному сайті програми.

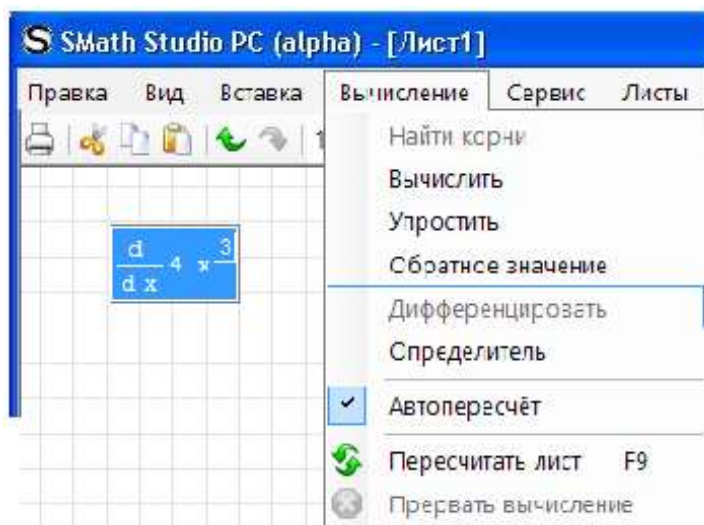


Рисунок 10.5 – Вікно налаштування диференціальних обчислень

Однією з головних переваг SMath Studio є можливість збереження файлів у форматі HTML, а також наявність версії для КПК Windows. Завдяки чому обчислення, розпочаті в дорозі, можна продовжити вдома чи на роботі, що особливо актуально для студентів, які розраховують курсові (рис. 10.6).

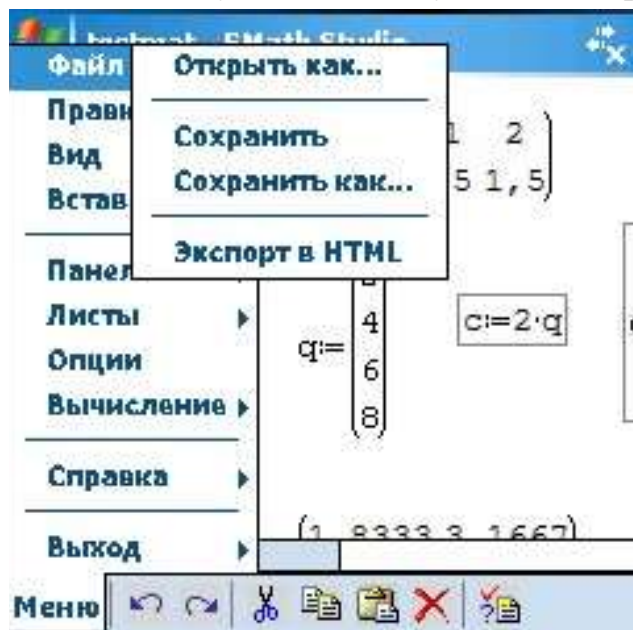


Рисунок 10.6 – Робота з випадаючим меню

Досі в програмі не реалізована робота з масивами і можливість побудови графіків функції по заданих точках, але розробник програми Андрій Івашов має намір включити такі можливості в майбутні релізи. Крім цього, доробка торкнеться обчислювального ядра програми, відображення графіків, будуть внесені зміни і в користувацький інтерфейс. Автор розглядає можливість створення версії і для смартфонів.

В цілому програма виглядає дуже привабливо, особливо своєю можливістю працювати не тільки на КПК, а й на стаціонарному комп'ютері.

Враховуючи те, що SMath Studio активно розвивається, в перспективі вона може стати дуже потужним і універсальним математичним пакетом. Радую також те, що зараз програма розповсюджується безкоштовно.

#### 4. Встановлення програми SMath Studio

Встановлення програми на комп'ютер під керівництвом Microsoft Windows досить просте: скачати або переписати інсталятор і запустити його. Якщо всі необхідні компоненти присутні, то програма буде встановлена. Для роботи програми потрібно Microsoft .NET Framework.

Ще один непоганий спосіб – завантажити десктопну версію і просто скопіювати її туди, звідки планується запускати програму, наприклад, на Робочий стіл. Тут вже не треба заздалегідь ставити Microsoft .NET Framework, всі необхідні компоненти вже є у складі виконуваного файлу. Даний спосіб зручний для перенесення програми з собою на флешці.

Встановлення програми під GNU Linux зводиться до встановлення середовища Mono, скачування архіву SMath Studio для Linux і розархівування в потрібну папку. Детально щодо дій по встановленню потрібних бібліотек зі складу Mono і на питаннях визначення прав доступу до папки з розархівованою програмою SMath Studio зупинятися не будемо. Сучасні дистрибутиви Linux надають можливість видати необхідні права на папку і вкладені файли з файлового менеджера (наприклад, з Nautilus'a), щоб не запускати виконуваний

скрипт через термінал з правами суперкористувача. Потрібні бібліотеки ставляться через менеджер пакетів.

## 5. Основи роботи в програмі

SMath Studio, в загальних рисах, повторює інтерфейс Mathcad. У програмі є основне робоче поле, в якому і проводяться всі обчислення у формі, максимально наближеній до нормальних математичних позначень. Поле при стандартних налаштуваннях виглядає як аркуш із зошита в клітинку, і в ньому користувач і вводить математичні вирази. Для цього треба встановити курсор (червоний хрестик) у вподобане місце і почати введення з клавіатури. Після того, як вираз введено, можна його порахувати, натиснувши «=». Схожим способом оголошується змінна: пишеться ім'я, ставиться знак присвоєння «:=» за допомогою кнопки на панелі «Арифметика» або двокрапкою на клавіатурі, вводиться значення. Оголошену змінну можна використовувати у виразах, її значення буде підставлено автоматично при обчисленнях. Якщо навести курсор миші на невиділений вираз, то з'явиться його результат в символній формі.

Реакція програми на натискання деяких клавіш залежить від контексту. У таблиці 10.1 наведені найбільш часті варіанти.

Таблиця 10.1 – Реакція програми на натискання деяких клавіш

Клавіші	Реакція
Пробіл	Вставка «#» в робочому полі аркушу, виділення більшої частини виразу при вже виділеному кутовим курсором частині виразу, пробіл для текстового поля.
Введення	Вставка порожнього проміжку на аркуші, вихід з поля виразу.
Курсорні клавіші	Перехід до наступного або попереднього вислову, рух по елементах виразу зі зміною кутового курсора.
	Корінь квадратний в робочому полі аркуша або зворотний слеш в тексті.
	Дріб в робочому полі аркушу або прямий слеш в тексті.
	У текстовому полі це так і залишиться двокрапкою, а в робочому полі аркуша перетвориться на знак присвоєння.
	Знак дорівнює в тексті і команда чисельного обчислення в полі формули. Для спроби обчислення неописаної змінної буде проведена заміна на «:=».
	Shift+2. «Собака» в тексті на англійській розкладці, (на віндосий українській – типографські лапки) або двовимірний графік у робочому полі аркуша.
	У тексті це буде виглядати так само, а при введенні формули стане місцем для нижнього індексу.

Зовнішній вигляд запущеної програми показаний на рис. 10.7.

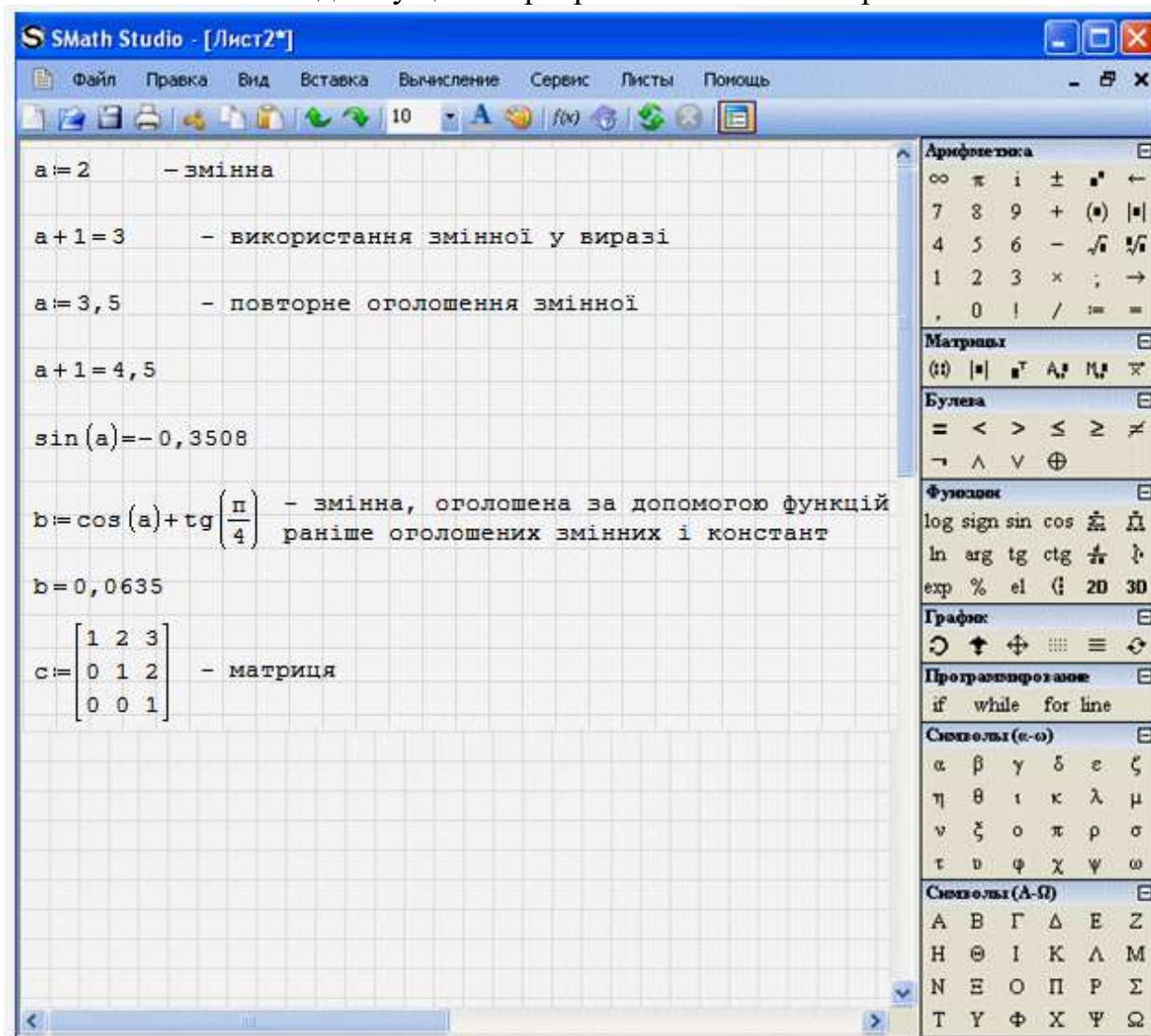


Рисунок 10.7 – Зовнішній вигляд запущеної програми

Як видно, вікно програми стандартно містить заголовок, головне меню, панель інструментів (рис. 10.8) і саме робоче поле.



Рисунок 10.8. – Рисунок панелі інструментів

Оскільки при наведенні курсору на ту чи іншу кнопку панелі інструментів з'являється підказка, то пояснювати призначення кнопок не потрібно.

У SMATH Studio є деякі правила запису виразів:

- використана змінна або функція повинні бути описані заздалегідь. Заздалегідь – значить лівіше або вище того виразу, де вона використовується в обчисленні;

- якщо змінна переописувалась, то буде використано те значення, яке зустрілося самим останнім перед використанням в обчисленнях. При описі змінної у виразі можна використовувати вбудовані і раніше описані функції,

раніше задані змінні та їх поєднання. Якщо використовуються у виразі змінні, які раніше не описувалися, то результат можна буде отримати тільки в символічному вигляді (або описати відсутні змінні і розмістити вираз нижче або правіше описаних змінних для чисельного результату);

– змінна не обов'язково повинна обчислюватися як числове значення, допускається присвоєння імен виразам, що дають при обчисленні матрицю;

– для символічних обчислень описувати змінні заздалегідь не потрібно, якщо не треба, щоб при перетворенні виразів були підставлені їх значення.

Бічна панель інструментів розташована праворуч у вікні програми і може бути прибрана, за непотрібністю, натисканням на крайню праву кнопку головної панелі інструментів. Бічна панель складається з окремих панелей, що містять набори команд у вигляді кнопок. Кожна така панель може бути згорнута за допомогою кнопки  $\Xi$ , розташованої в правому кутку заголовка панелі. Розглянемо призначення панелей.

**Панель «Арифметика»** містить цифри 0 ... 9, роздільник десяткового дробу (в залежності від налаштувань операційної системи, це може бути і крапка і кома), букву  $\pi$  для однойменного числа, знак факторіала «!», операції зведення в ступінь «^», взяття квадратного кореня «\» або кореня n-го ступеня Ctrl + «\». Ще на панелі є графічний варіант клавіші Backspace і знаки присвоєння «:», символічного обчислення Ctrl + «.» І знак рівності для обчислення в чисельній формі.

**Панель «Матриці»** надає можливість задати матрицю (Ctrl + «M»), обчислити її визначник, транспонувати матрицю, отримати алгебраїчне доповнення, мінор, обчислити векторний твір. Останнє виглядає трохи дивно, але справа в тому, що вектор задається як вектор-стовпець з розміром в 3 елементи, по одному на кожному рядку.

**Панель «Булева»** мається на увазі алгебра. Операції відношення, заперечення, «і», «або», «виключає або». Булеве «дорівнює» використовується ще й в завданні умов для функції if і в циклі for.

**Панель «Функції»** тут зібрана добірка основних тригонометричних функцій, визначений інтеграл, взяття похідної в точці, сума елементів . Ще є кнопки «2D» і «3D» для вставки двовимірних або тривимірних графіків, натуральний логарифм, відсотки, вибір елемента за індексом («e1» або квадратні дужки на клавіатурі), знак системи.

Слід детальніше зупинитися на функції диференціювання. Результат роботи цієї функції залежить від того, вибрано чи обчислення значення похідної в точці за допомогою «=» (змінна повинна бути визначена) або використано символічне обчислення «→». Якщо використано символічне обчислення, то результат залежить від того, чи була визначена змінна заздалегідь. Якщо ні, то буде видано вираз для похідної за заданими змінними. Якщо ж змінна була визначена, то буде проведена підстановка в результат її значення. Приклад показаний на рис 10.9.



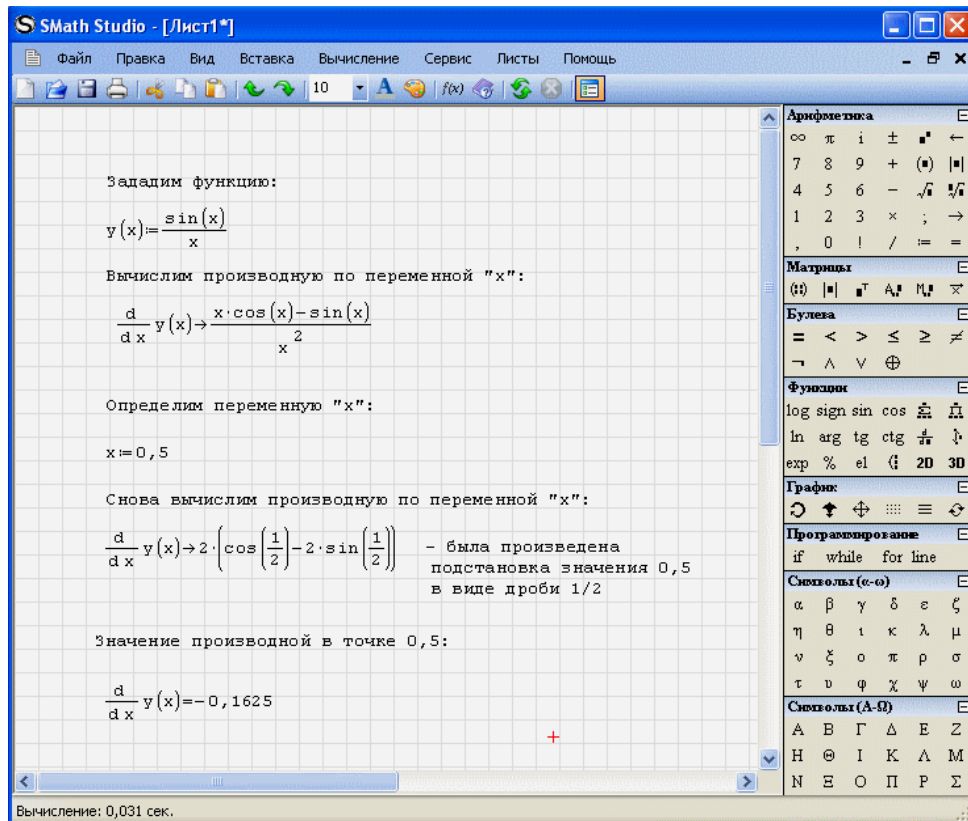


Рисунок 10.9 – Обчислення похідної

**Панель «График»** тут можна обертати, масштабувати і зрушувати графіки, задати відображення точками або лініями, перемальовувати графіки заново у разі потреби. Операція обертання доступна тільки для тривимірних графіків (зрозуміло, що для двовимірних її користь була б сумнівною). Графіки малюються в декартовій системі координат. Тако ж є можливість автомасштабування. Це може доставити певні незручності, наприклад, коли Ви намагаєтеся отримати графік частотної характеристики і змушені вибрати певну розмірність частотної сітки або перевіряти можливості програми в масштабуванні по осях  $x$  і  $y$  окремо (при натиснутих клавішах  $Ctrl$  або  $Shift$ ).

Графіки будуються для функцій від змінної « $x$ » (2D) або « $x$ » і « $y$ » (3D). Взагалі, функції можуть бути від будь-яких аргументів, але при побудові графіка повинні бути вказані в якості аргументів саме ці змінні. Ситуація пояснюється на рис. 3, коли замість « $t$ » підставляється « $x$ ».

Крім аналітичного завдання графіка, можна задати його у вигляді матриці, де кожен стовпець міститиме координати точок для осей  $X$  і  $Y$  відповідно (або ще для осі  $Z$ ). Крім того, замість функції або матриці можна поставити знак системи, а вже в саму систему вписати ім'я функцій або матриць чи виразів для побудови графіків. Це надасть можливість відобразити декілька графіків на одному рисунку.

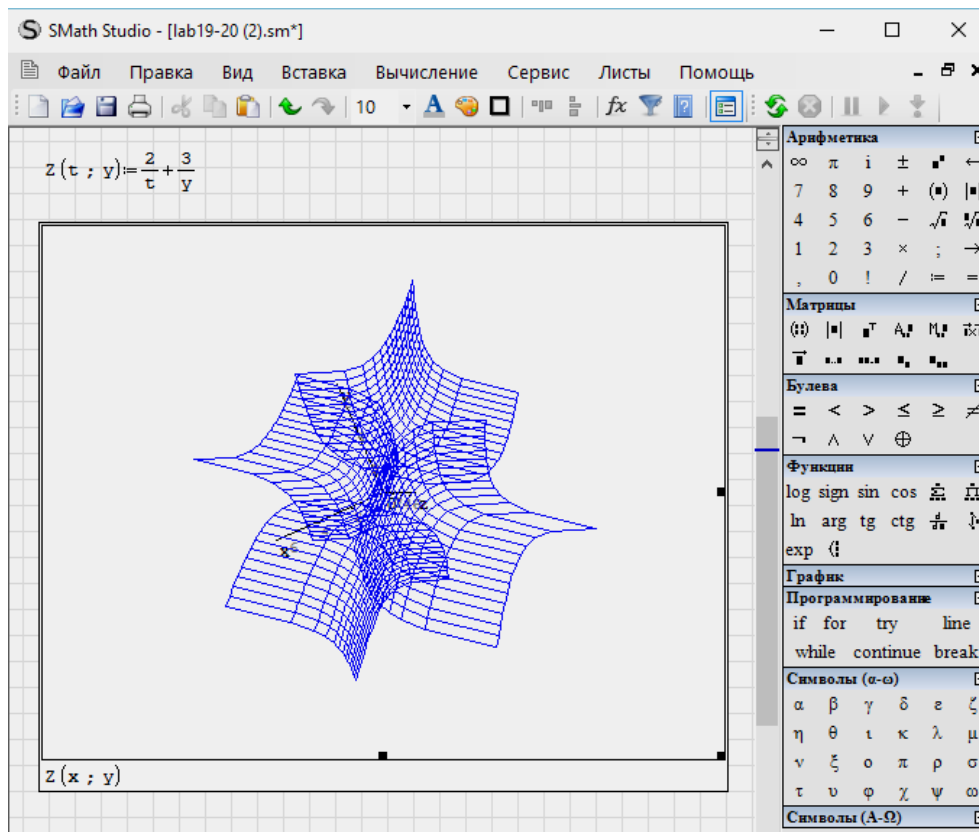


Рисунок 10.10 – Графік функції 3D

**Панель «Програмування»** містить засоби, що надають можливість організувати циклічні обчислення (цикли «for» і «while») або задати умовну функцію «if». Замість операторних дужок використовується лінія (line). Лінія за замовчуванням містить два місця для послідовних обчислень, але її можна розтягнути. Для цього потрібно виділити лінію кутовим курсором (треба клікнути по місцю, зазначеному квадратиком і натиснути пробіл, щоб виділилися обидва місця). Повинна з'явитися спеціальна квадратна мітка, зачепивши яку мишкою, можна розтягнути лінію до необхідної кількості місць. Аналогічний спосіб розтягування працює і для знака системи.

**Панель «Символи»** потрібна для вставки букв грецького алфавіту. Історична традиція використання для імен функцій і параметрів грецьких букв вимагає, щоб така можливість була. Інакше довелось б писати alpha, beta, gamma ... Тут слід зауважити, що латиниці і грецьких букв математикам давно вже не вистачає, багатобуквенні імена у них якось не приживаються, на відміну від тих же програмістів, і скоро вже будемо використовувати характерні літери з кирилиці, іврити і китайські ієрогліфи :-). Грецьку букву можна отримати і по іншому: ввести латинську і натиснути Ctrl + «G».

Ви вже помітили, що набір функцій на однойменній панелі досить обмежений. Меню «Вставка» надає можливість використовувати й інші вбудовані функції, причому їх досить багато (дивись рис. 10.11). Функції згруповані за розділами. Клацання мишею в списку «Ім'я функції» і натискання буквенної клавіші покажуть функцію, що починається з цієї букви.

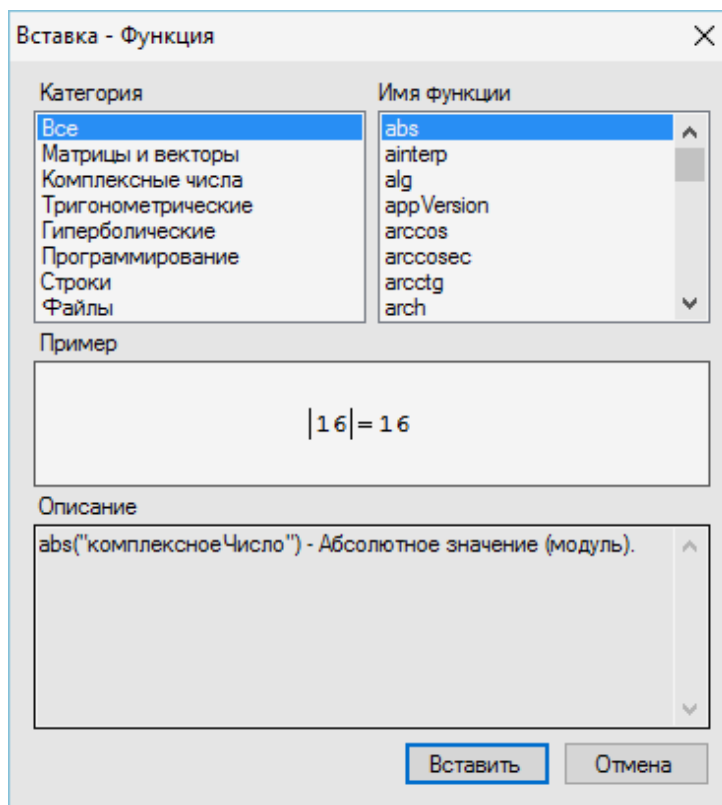


Рисунок 10.11 – Вставка функції

В полі «Опис» дається синтаксис функції та її опис, що є своєрідною заміною меню «Допомога». Знання синтаксису надає можливість не використовувати вставку функцій, а набирати безпосередньо ім'я функції. Деякі з функцій у процесі набору змінюють свій вигляд, як наприклад функція модуля  $\text{abs}(x)$  перетворюється в  $|x|$ .

Поле «Приклад» показує приклад використання з урахуванням того, як функція буде виглядати. Це суттєво допомагає розібратися з тим, які аргументи яких функцій вимагають, і як це буде виглядати після їх опису.

Ну і, природно, можна описувати свої функції з використанням вбудованих і раніше описаних.

Головне меню SMATH Studio схоже на головне меню будь-якої іншої прикладної програми, що й не дивно. Тим не менш, деякі особливості є і у такій знайомій речі, як головне меню. Перерахуємо пункти меню і дамо деякі пояснення.

**Меню «Файл»** Варто зазначити, що SMATH Studio може читати Mathcad'овські файли (\*.xmcd) і зберігати листи в такому ж форматі, крім власного. Ще одна корисна опція – генерація звіту в HTML. Для цього достатньо при збереженні файлу вибрати відповідне розширення. Звіт буде являти собою html-сторінку і папку з малюнками у форматі PNG. Такий звіт можна переглянути в браузері або відкрити в текстовому редакторі, який підтримує HTML. Увага! Малюнки та файл звіту мають імена, сформовані з імені листа.

**Меню «Правка»** теж не містить нічого незрозумілого. Єдине, що варто відзначити, – не завжди коректна робота з використанням буфера обміну між SMATH Studio версій до 0.85 та іншими додатками. Текст з OpenOffice.org.3.0

вставляється в лист коректно. Скопіювати текст і формули назад можна тільки при виділенні тексту всередині рамки текстового поля або формули. При цьому текст веде себе нормально, а формула перетворюється на напис у внутрішньому форматі SMath Studio. Рисунок 10.12 показує оголошення функції з наступною побудовою її графіка.

А ось як виглядає скопійована в буфер обміну та ж функція:  $y(x) \leftarrow \{17-$

**Меню «Вид»** містить всього один пункт з прапорцем, який слід встановити або зняти.

**Меню «Вставка»** призначене для впровадження в лист матриць, операторів, графіків, малюнків, текстових областей і розділювачів для оформлення документа. Вставка операторів і функцій виступає в якості своєрідного довідника.

Малюнки можна вставити з файлу в форматі \*.bmp або намалювати самостійно (рис. 10.13). Останнє вимагає певної вправності та вміння малювати.

Рисунок 10.12 – Функція та її графік

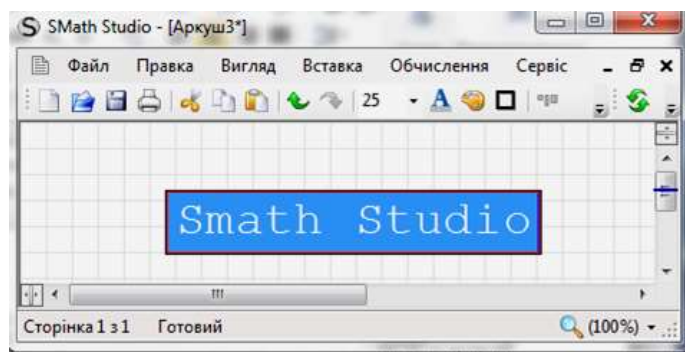


Рисунок 10.13 – Рисование в Smart Studio

Меню «Обчислення» містить цілий набір можливих дій (рис. 10.14).

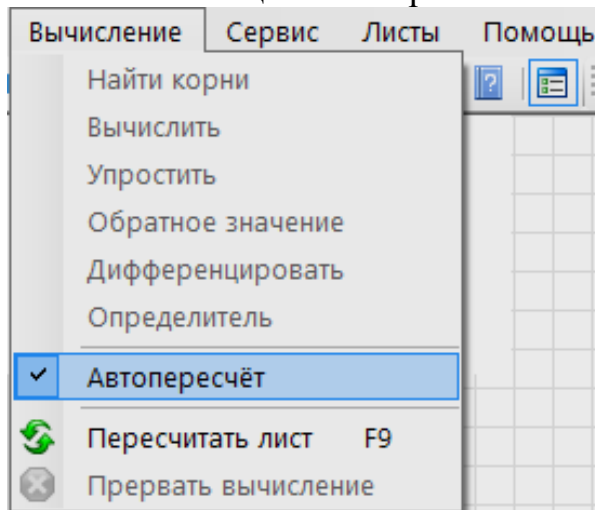


Рисунок 10.14 – Вміст меню «Обчислення»

Виділивши вираз або його частину, можна знайти корені рівняння, зробити обчислення з урахуванням підстановок і скорочень, спростити вираз, отримати

зворотне значення в символному або числовому вигляді, отримати похідну по виділеній заздалегідь змінній і визначнику матриці (рис. 10.15).

**Меню «Сервіс»** містить теж тільки один пункт «Опції», який служить для задання точності представлення результату, діапазону пошуку коренів рівнянь, а також для налаштування зовнішнього вигляду програми, включаючи кольори і мову (рис. 10.16).

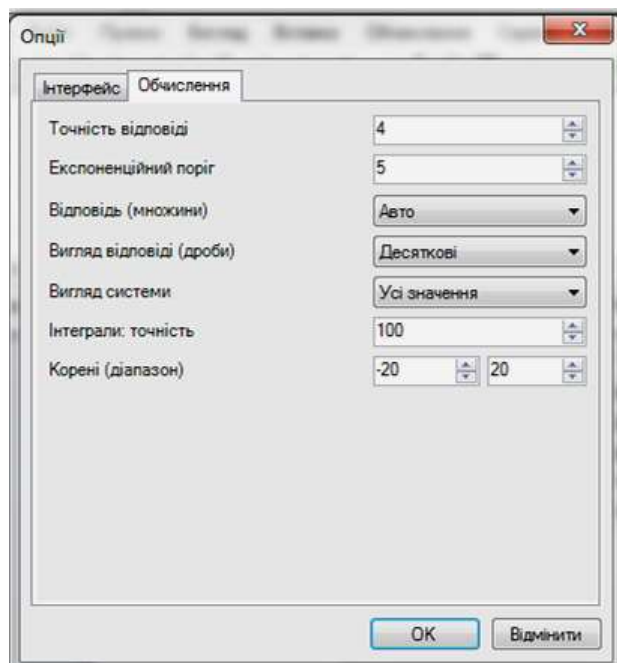


Рисунок 10.15 – Опції обчислення

Рисунок 10.16 – Опції інтерфейсу

**Меню «Листи»** надає можливість створення, видалення листа документа або перемикається між листами.

**Меню «Допомога»** відкриває доступ до довідника, повідомляє відомості про програму (збірка, автори, контактна інформація, використовувані бібліотеки), перевіряє наявність оновлень програми через Internet (ну звичайно, якщо є підключення) і повідомляє про способи фінансової підтримки проекту.

Окремо слід згадати пункт **«Приклади»** (дивись рис. 10.17). Показує, як можна, використовуючи програмування, робити деякі речі, для яких немає вбудованих функцій. Це стосується і нелінійних алгебраїчних рівнянь, їх систем, розкладання функції в ряди та інше. Приклади досить зрозумілі і містять необхідні коментарі.

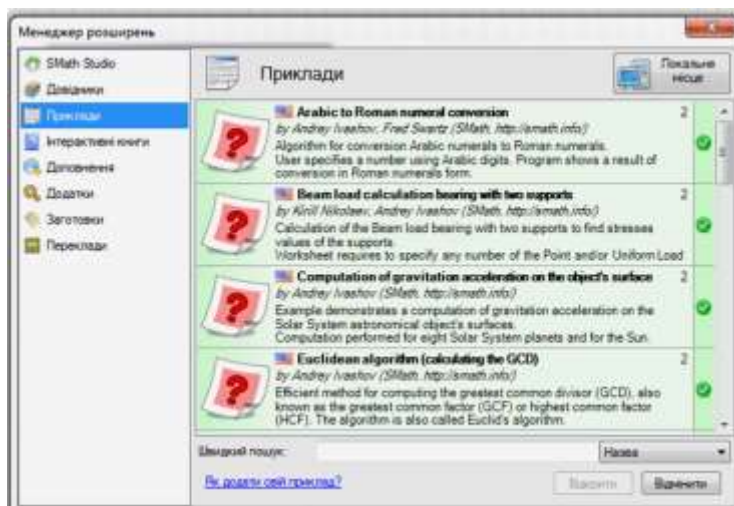


Рисунок 10.17 – Перелік прикладів

Як і у всякої складної програми, SMath Studio має особливості, які можуть спочатку здатися чимось незрозумілим або взагалі помилками.

- часткове копіювання інтерфейсу Matcad'а, особливо помітне при редагуванні виразів;
- виділення кліком мишки – служить для редагування формул, тексту або вигляду графіка, а виділення рамкою – для перетягування, копіювання та видалення об'єктів на аркуші.
- знак рівності служить для обчислення;
- для пошуку коренів рівнянь задається певний діапазон;
- інтеграл береться по дійсних межах інтегрування.

## 6. Довідкова інформація

На форумі можна уточнити наявність тієї чи іншої можливості в певній збірці програми, задати питання по використанню, повідомити про помилку. Останнє бажано робити з додатком файлів, що містять помилку (формат \*.sm), або хоча б скріншотів. Своєчасне і докладне інформування розробників про знайдені помилки серйозно допомагає поліпшити програму. Досить подивитися історію версій, щоб зрозуміти, як з імітатора mathcad'івського інтерфейсу програма перетворюється на повноцінний математичний пакет, що вміє рахувати досить складні речі. Допомога щодо використання програми надасть можливість зорієнтуватися в можливостях SMath Studio і мати уявлення про функціональні обмеження.

## II. Завдання до лабораторної роботи

1. Ознайомитись з теоретичними відомостями лабораторної роботи.
2. Познайдомитись з основними елементами інтерфейсу SMath Studio: головне вікно, робота з файлами, панелі інструментів.
3. З допомогою панелі інструментів набрати вирази з рис. 10.7, 10.9-10.17.
4. Виконати перетворення арабських цифр в римські (див. рис. 10.18).

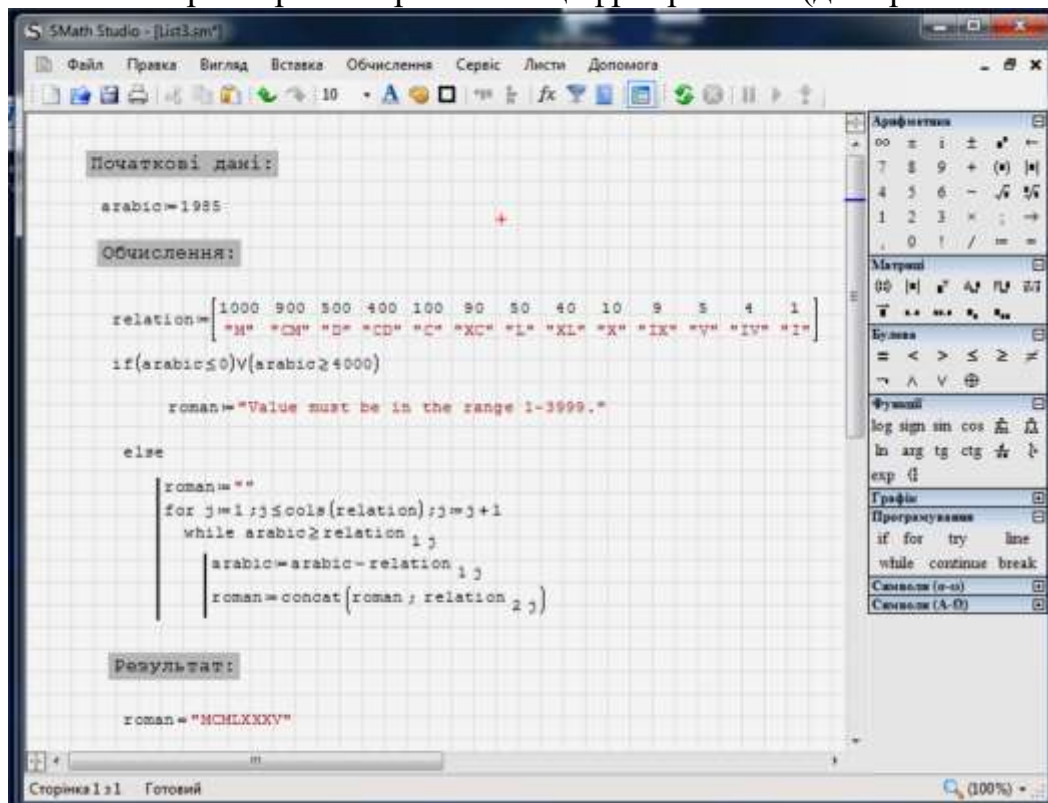


Рисунок 10.18. Приклад перетворення римських цифр в арабські

### Римські цифри

- 1 I лат. *unus, unum*
- 5 V лат. *quinque*
- 10 X лат. *decem*
- 50 L лат. *quingenta*
- 100 C лат. *centum*
- 500 D лат. *quingenti*
- 1000 M лат. *mille*

- IV = 4
- IX = 9
- XL = 40
- XC = 90
- CD = 400
- CM = 900

В українській мові для закріплення в пам'яті буквених позначень цифр в порядку убування існують мнемонічні правила:

**Ми Даруємо Соковиті Лимони, Хватить Всім Іх.**

Відповідно **M, D, C, L, X, V, I**

Наприклад, число 94 буде  $XCIV = 100 - 10 + 5 - 1 = 94$  – так зване «правило віднімання» (з'явилося в епоху пізньої античності, а до цього римляни писали число 4 як IIII, а число 40 – як XXXX).

Необхідно відзначити, що інші способи «віднімання» неприпустимі; так, число 99 має бути записане як XCIX, але не як IC. Однак, в наші дні в деяких випадках використовується і спрощений запис римських чисел: наприклад, у програмі Microsoft Excel при перетворенні арабських цифр в римські за допомогою функції «РИМСЬКЕ» можна використовувати кілька видів представлення чисел, від класичного до сильно спрощеного (так, число 499 може бути записане як CDXCIX, LDVLIIV, XDIX, VDIV або ID). Спрощення полягає в тому, що для зменшення будь-якої цифри зліва від неї може писатися будь-яка інша цифра: пізньої античності, а до цього римляни писали число 4 як IIII.

Тисяча (M), віднімемо 1 (I), отримаємо 999 (IM) замість CMXCIX.

**Приклад:** 1999 – MIM замість MCMXCIX

1950: Тисяча (M), віднімемо 50 (L), отримаємо 950 (LM).

**Приклад:** 1950 – MLM замість MCML

З допомогою римських цифр можна записувати і великі числа. Для цього над тими цифрами, які позначають тисячі, ставиться риска, а над цифрами, які позначають мільйони, – подвійна риска. **Наприклад,** число 123123 буде виглядати так:  $\overline{\overline{CXIII}}\overline{CXIII}$

А мільйон як  $\overline{\overline{I}}$ , але не з одною, а з двома рисочками вгорі:  $\overline{\overline{I}}$

5. З допомогою вкладки «Допомога» / «SMart Studio Live» перейти за посиланням та описати особливості хмарного сервісу Smart, використовуючи приклади (скріншоти).

6. Підготувати теоретичний матеріал відносно останньої версії SMart Studio та її можливостей.

### **III. Індивідуальні завдання**

1. Записати свій порядковий номер в академічному журналі групи римськими цифрами.

2. Записати свою повну дату народження арабськими (наприклад 12.05.1991 – буде виглядати як 12051991) та римськими цифрами. Результати рішень зберегти в файл.

3. Підготувати звіт в електронному вигляді та надіслати до системи електронного навчання.



## Лабораторна робота № 11

### Тема: «Основи роботи з СКМ Maxima»

Питання:

1. Історія виникнення Maxima.
2. Графічний інтерфейс в Maxima.
3. Основи роботи з СКМ Maxima.

### I. Теоретичний матеріал

**Maxima** – вільна комп'ютерна алгебраїчна система, заснована на Macsyma версії 1982 року. Написана на мові програмування Common Lisp, та розповсюджується за умовами ліцензії GNU General Public License. Працює на всіх POSIX-сумісних платформах та Microsoft Windows.

Maxima походить від системи Macsyma, що розроблялася в MIT з 1968 по 1982 роки в рамках проекту Project MAC, фінансованого Міністерством енергетики США (DOE) та іншими державними організаціями. Професор Вільям Шелтер (англ. William F. Schelter) з Техаського університету в Остіні підтримував одну із версій системи, відому як DOE Macsyma, з 1982 року до самої своєї смерті в 2001 році.

У 1998 році Шелтер отримав від Міністерства енергетики дозвіл опублікувати вихідний код DOE Macsyma під ліцензією GPL, і в 2000 році він створив проект на SourceForge.net для підтримки та подальшого розвитку DOE Macsyma під ім'ям Maxima.

Maxima має ширкий набір засобів для проведення аналітичних обчислень, чисельних обчислень і побудови графіків. По набору можливостей система близька до таких комерційних системам, як Maple і Mathematica. У той же час вона володіє високим ступенем переносимості: може працювати на всіх основних сучасних операційних системах на комп'ютерах, починаючи від КПК. На рис. 11.1 зображено сайт завантаження програми.



Рисунок 11.1 – Завантаження СКМ Maxima

На даний момент Maxima випускається під дві платформи: Unix-сумісні системи, Linux і \* BSD, і MS Windows.

Сама по собі Maxima – консольна програма, і всі математичні формули виводить звичайними текстовими символами. У цьому є як мінімум два плюси. З одного боку, саму Maxima можна використовувати як ядро, надбудовуючи поверх неї графічні інтерфейси на будь-який смак. Їх на сьогоднішній день існує чимало.

З іншого боку, сама по собі, без будь-яких інтерфейсних надбудов, Maxima невимоглива до заліза і може працювати на таких комп'ютерах, які зараз і за комп'ютери вже ніхто не вважає.

Імена функцій і змінних в Maxima чутливі до регістру, тобто великі та малі літери в них різняться. Зручно це і з точки зору математика, для якої теж звично, що великими та малими буквами можуть позначатися різні об'єкти (наприклад, множини і їхні елементи, відповідно).

Для того, щоб почати працювати з програмою, знадобиться пакет Maxima, взяти його можна на сайті проекту. Принципи роботи з програмою не залежать від того, який інтерфейс до неї обрати.

З точки зору ознайомлення з самою Maxima найбільший інтерес представляють два інтерфейси.

Перший – це окрема самостійна графічна програма на ім'я *wxMaxima* (рис. 11.2). Вона, як і сама Maxima, крім Linux / \* BSD існує ще й у версії для MS Windows. У wxMaxima користувач вводить формули в текстовому вигляді, а відповідь Maxima відображає графічно, звичними математичними символами. Крім того, великий наголос тут зроблений на зручності введення: командний рядок відділений від вікна введення-виведення, а додаткові кнопки і система меню надають можливість вводити команди не тільки в текстовому, але й у діалоговому режимі. Так зване «автодоповнення» викликається клавішею «Tab». Веде ж воно себе, на жаль, всього лише як розумна історія команд. Викликає потрібну команду з уже введених в цій сесії, яка починається з заданих в командному рядку символів, але не доповнює до імен команд і їх параметрів.

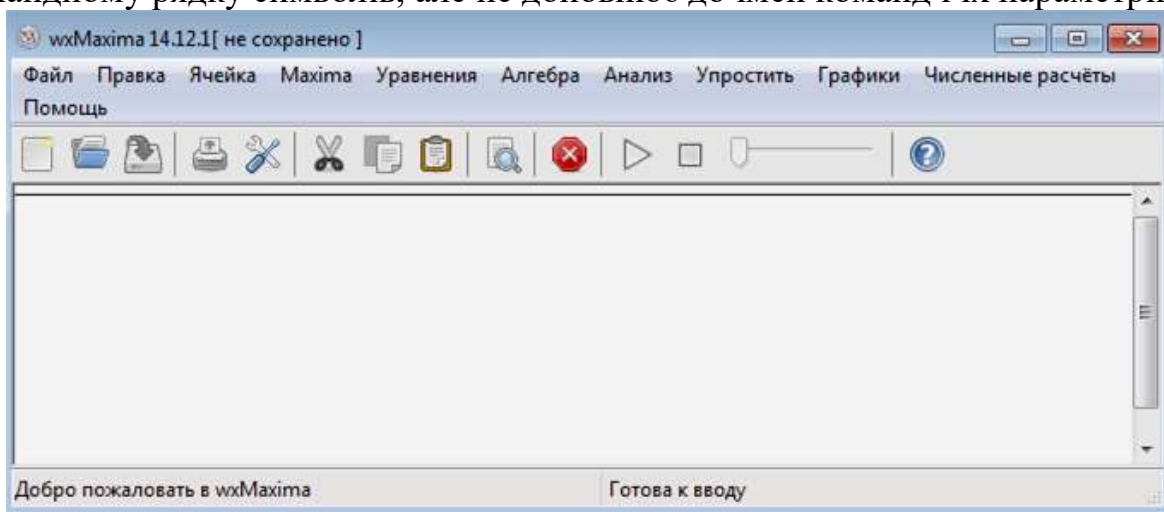


Рисунок 11.2 – Головна сторінка wxMaxima

Таким чином, цей інтерфейс найбільш зручний у тому випадку, коли вам потрібно багато обчислювати і бачити результати на екрані; і ще, можливо, в тому випадку, якщо користувач не дуже любить вводити всі команди з клавіатури. Крім того, wxMaxima надає зручний інтерфейс до документації по системі; так як документація поставляється в форматі html, замість цього можна використовувати звичайний браузер.

Другий досить цікавий інтерфейс Maxima – це додатковий режим в редакторі *TeXmacs* (рис. 3). Хоча цей редактор має спільне історичне минуле з широко відомим Emacs, що виявляється з назви, але практичної подібності між ними мало.

TeXmacs розробляється для візуального редагування текстів наукової тематики, при якому користувач бачить на екрані редагований текст практично в тому ж вигляді, в якому він буде роздрукований. Зокрема, він має так званий математичний режим введення, дуже зручний для роботи з найрізноманітнішими формулами, і вміє імпортувати / експортувати текст у LaTeX і XML / HTML.

Саме можливостями по роботі з формулами користується Maxima, викликана з TeXmacs'а. Фактично, формули відображаються у звичній математичній нотації, але при цьому їх можна редагувати і копіювати в інші документи на зразок звичайного тексту. Maxima-сесія викликається з меню: «вставити → Сесія → Maxima», при цьому з'являється додаткове меню з командами Maxima. Після запуску сесії можна вже всередині неї перейти в математичний режим введення (меню режимів введення викликається першою кнопкою на панелі введення) і при введенні також використовувати елементи математичної нотації. Цей інтерфейс буде найбільш зручний тим, хто хоче використовувати результати обчислень у своїх текстах і любить редагувати їх у візуальному режимі.

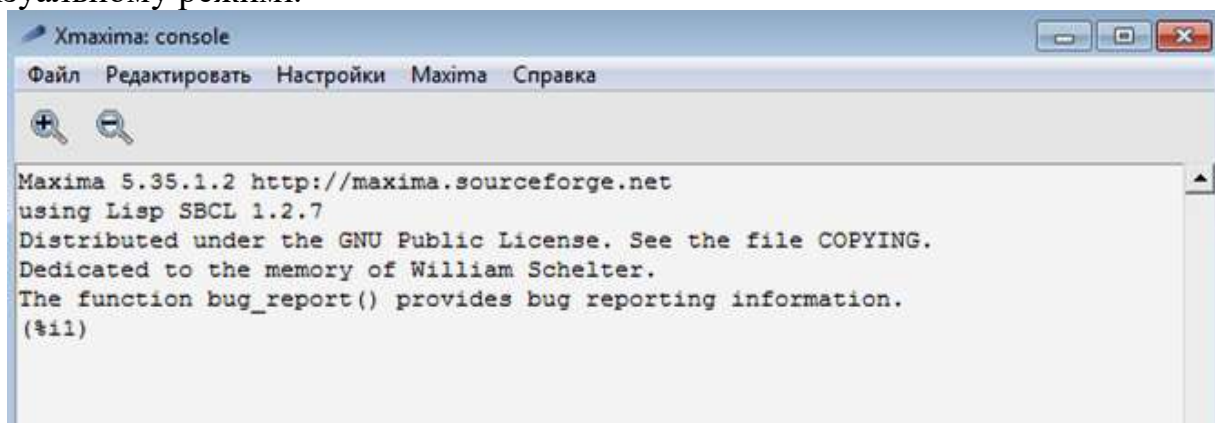


Рисунок 11.3 - Головне вікно TeXmacs

Після запуску Maxima-сесії ми бачимо перед собою такі рядки:

```
Maxima restarted.  
(%i1)
```

Перший – це повідомлення про те, що ядро Maxima тільки що запустилося (замість неї, залежно від версії і конкретної збірки, може виводитися коротка інформація про програму); друга – запрошення до введення першої команди. Команда в Maxima – це будь-яка комбінація математичних виразів і вбудованих функцій, завершена, у найпростішому випадку, крапкою з комою. Після введення команди і натиснення «Enter» Maxima виведе результат і буде чекати наступної команди:

Maxima restarted.

(%i1) (1/2 + 1/3 + 1/4)/(1/5 + 1/6 + 1/8);

(%o1)  $\frac{130}{59}$

(%i2) |

Рисунок 11.4 – Приклад введення команд

Для арифметичних дій використовуються традиційні позначення: -, +, \*, /; \*\* або ^ для зведення в ступінь, sqrt () для квадратного кореня.

Надалі будемо користуватися для наочності згадуваними раніше в математичному режимі введення редактора TeXmacs. Наприклад, наведений вище код виглядав би так:

(%i1)  $\frac{\frac{1}{2} + \frac{1}{3} + \frac{1}{4}}{\frac{1}{5} + \frac{1}{6} + \frac{1}{8}};$   
 (%o1)  $\frac{130}{59}$

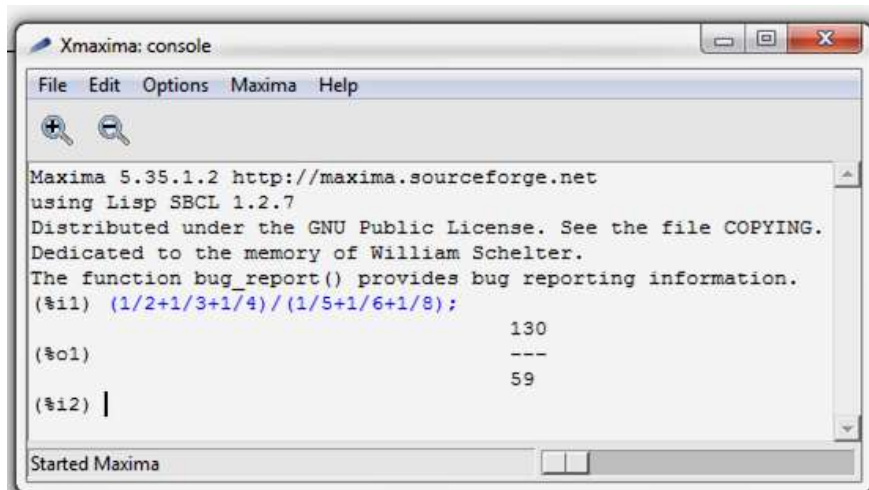


Рисунок 11.5 – Введення команд в режимі TeXmacs

Як видно, кожна клітинка має свою мітку; ця мітка – укладене в дужки ім'я комірки. Осередки введення іменуються як % і з номером (і від input – введення), осередки виводу – як % о з відповідним номером (о від output – виведення). Зі знака % починаються всі вбудовані службові імена: щоб, з одного боку зробити їх досить короткими і зручними у використанні, а з іншого – уникнути можливих накладок з одними іменами, які теж часто зручно робити короткими.

Приміром, внутрішніми іменами % e і% рі позначені загальновідомі математичні постійні; а через % с з номером позначаються константи, що використовуються при інтегруванні, для яких використання літери «с» традиційно в математиці.

При введенні можна звертатися до будь-якої з попередніх команд по її імені, підставляючи її в будь-який вирази. Крім того останній осередок виведення позначається через %, а остання комірка вводу – через \_. Це надає можливість звертатися до останнього результату, не відволікаючись на те, який його номер.

$$(\%i2) \% + \frac{47}{59};$$

$$(\%o2) 3$$

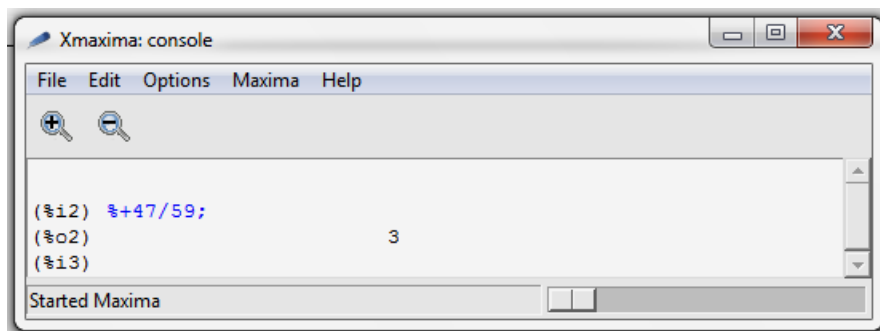


Рисунок 11.6 – Введення команд в режимі TeXmacs

Тут  $\% + 47/59$  – те ж саме, що  $\% o1 + 47/59$ .

Висновок результату обчислення не завжди потрібен на екрані; його можна заглушити, завершивши команду символом \$ замість ; . Заглушений результат при цьому все одно обчислюється; як показано в даному прикладі осередку % o1 і % o2 доступні, хоча і не показані (до осередку % o2 звернення йде через символ %, сенс якого розшифрований вище):

$$(\%i1) \sqrt{2} + 3\$$$

$$(\%i2) 2\sqrt{2} + 1\$$$

$$(\%i3) \% - \%o1;$$

$$(\%o3) \sqrt{2} - 2$$

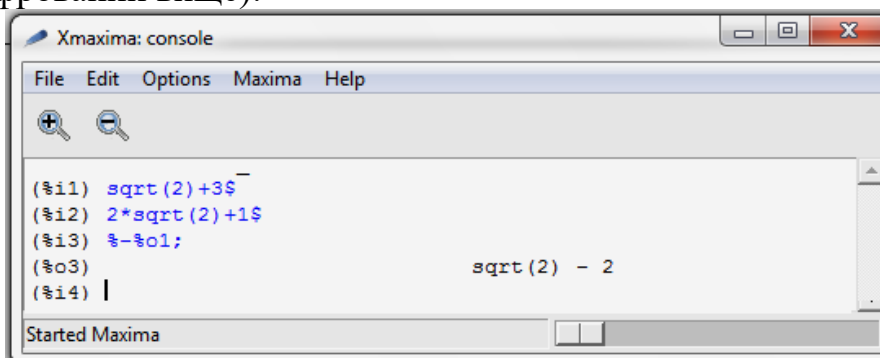


Рисунок 11.7 – Введення команд в режимі TeXmacs

Кожну наступну команду не обов'язково писати з нового рядка; якщо ввести декілька команд в одну стрічку, кожній з них все одно буде відповідати своє ім'я комірки. Наприклад, тут в рядку після мітки % i1 введені осередки від % i1 до % i4; в комірці % i3 використовуються % i1 і % i2 (позначена як \_ - попередній введений):

$$(\%i1) \operatorname{asin}\left(\frac{1}{2}\right) \$ \operatorname{acos}\left(\frac{1}{2}\right); \%i1 + \_ ; \%o1 + \_ ;$$

$$(\%o2) \frac{\pi}{3}$$

$$(\%o3) \frac{\pi}{2}$$

$$(\%o4) \frac{2\pi}{3}$$

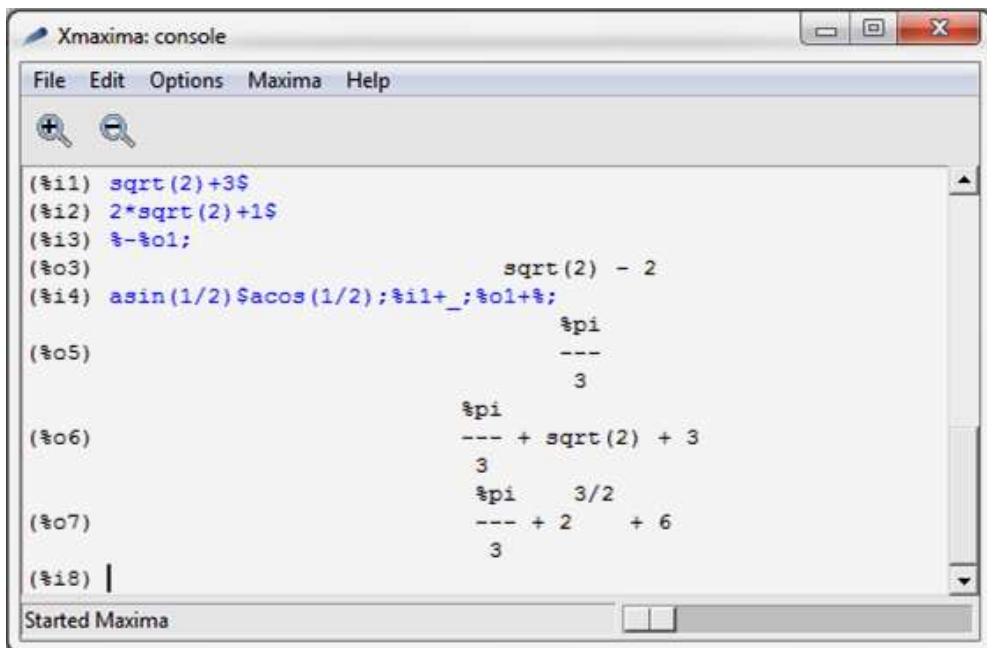


Рисунок 11.8 – Введення команд в режимі TeXmacs

У wxMaxima і TeXmacs останню або єдину команду в рядку можна не доповнювати завершальним символом – це спрацює так само, якщо б вона була завершена ; . Висновок заглушений не буде. У подальших прикладах будемо опускати ; .

Крім використання імен осередків, ми, природно, можемо й самі давати імена будь-яким виразами. По-іншому можна сказати, що ми присвоюємо значення змінним, з тією різницею, що у вигляді значення такої змінної може виступати будь-який математичний вираз. Робиться це за допомогою двокрапки – знак рівності залишений рівнянням, які, враховуючи загальний математичний контекст записи, простіше і звичніше так читаються. Приклад використання символічного запису та аналітичних обчислень:

(%i1) equation:  $x^3 - x = 0$ \$

(%i2) solve(equation)

(%o2)  $[x = -1, x = 1, x = 0]$

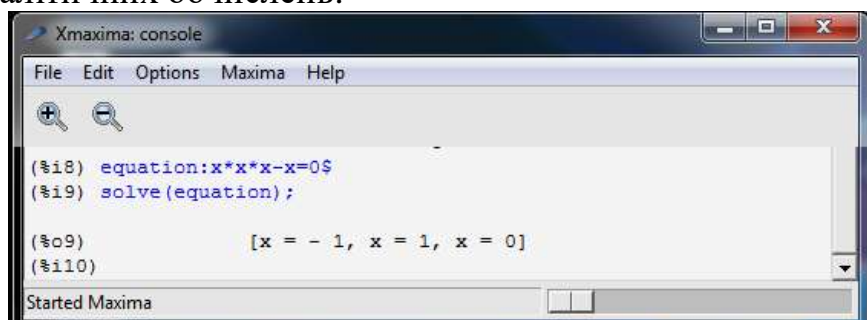


Рисунок 11.9 – Введення команд в режимі TeXmacs

У якомусь сенсі двокрапка навіть наочніше в такому контексті, ніж знак = : це можна розуміти як завдання, а потім через двокрапку розшифруємо, що саме воно означає. Після того, як вираз названо, ми в будь-який момент можемо викликати його по імені:

(%i3) `diff(equation,x)`

(%o3)  $3x^2 - 1 = 0$

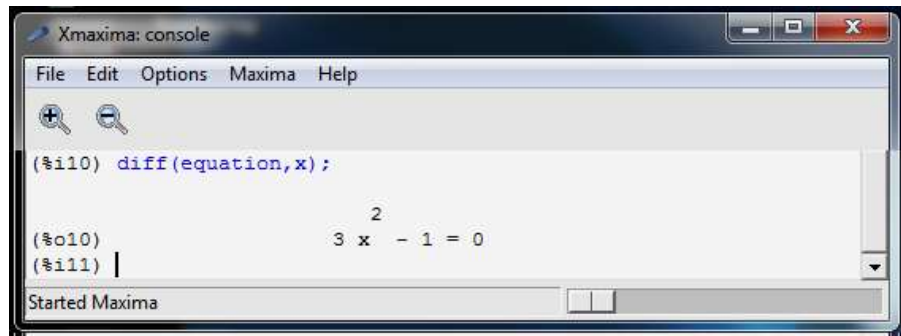


Рисунок 11.10 – Введення команд в режимі TeXmacs

Будь-яке ім'я можна очистити від присвоєного йому виразу функцією `kill` (), і звільнити займану цим виразом пам'ять. Для цього потрібно набрати **kill** (**name**), де name – ім'я знищуваного виразу; причому це може бути як ім'я, призначене користувачем, так і будь-яка комірка введення або виведення. Точно так само можна очистити всю пам'ять і звільнити всі імена, ввівши **kill** (**all**). У цьому випадку очистяться в тому числі і всі осередки введення-виведення, і їх нумерація знову почнеться з одиниці.

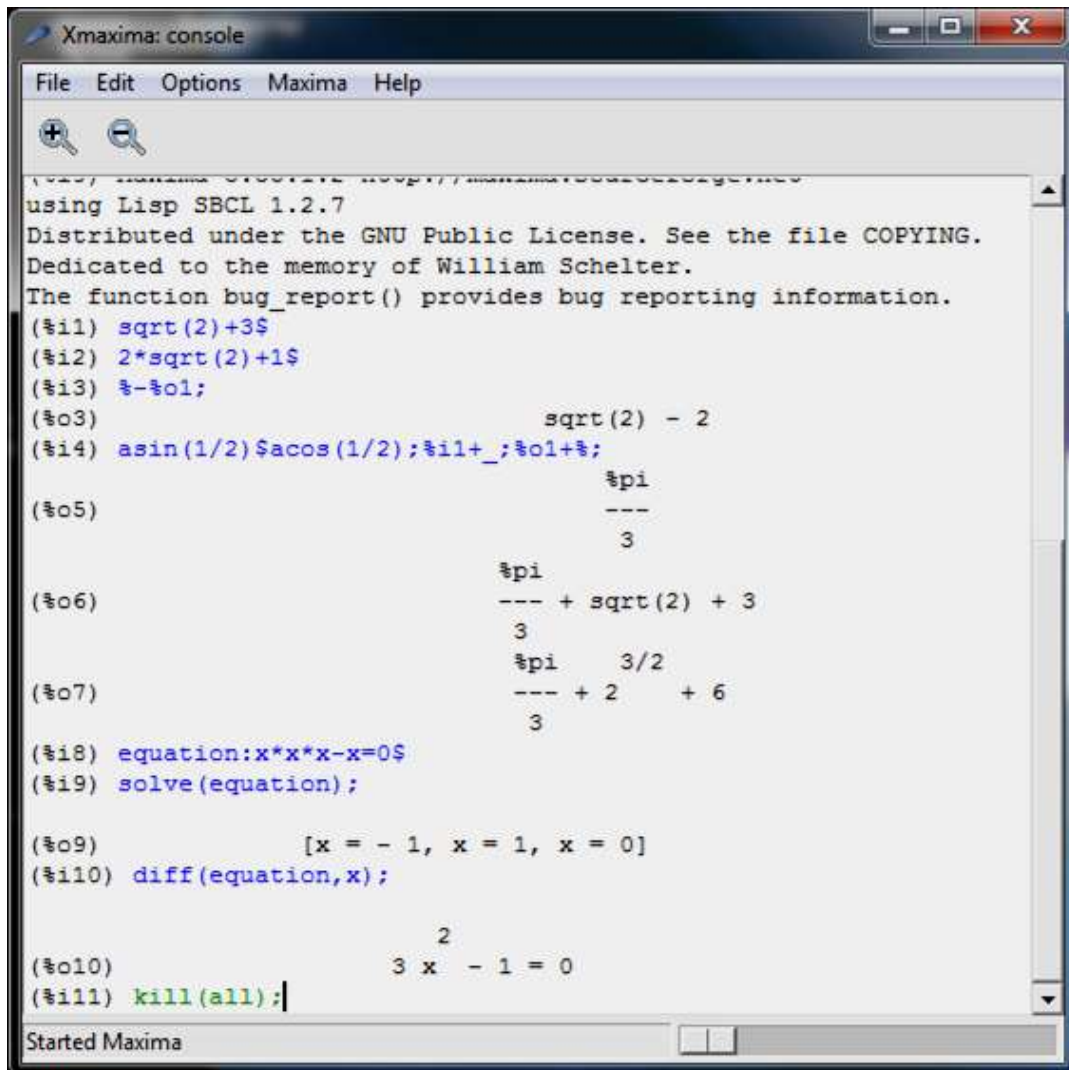


Рисунок 11.11 – Введення команд в режимі TeXmacs

```

Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1) sqrt(2)+3$
(%i2) 2*sqrt(2)+1$
(%i3) %i1-%o1;
(%o3)          sqrt(2) - 2
(%i4) asin(1/2)$acos(1/2);%i1+%;%o1+%;
(%o4)          pi
(%o5)          ---
              3
(%o6)          pi
              --- + sqrt(2) + 3
              3
(%o7)          pi    3/2
              --- + 2  + 6
              3
(%i8) equation:x*x*x-x=0$
(%i9) solve(equation);
(%o9)          [x = - 1, x = 1, x = 0]
(%i10) diff(equation,x);
(%o10)          2
              3 x  - 1 = 0
(%i11) kill(all);
(%o0)          done
(%i1)
Started Maxima

```

Рисунок 11.12 – Введення команд в режимі TeXmacs

Надалі, якщо по контексту буде матися на увазі логічне продовження попередніх рядків вводу-виводу, будемо продовжувати нумерацію (цей прийом було використано вище). Коли ж новий «сеанс» буде ніяк не пов'язаний з попереднім, розпочнемо нумерацію заново; це буде непрямою вказівкою зробити «kill (all)», якщо користувач буде набирати приклади в Maxima, так імена змінних як і осередків у таких «сеансах» можуть повторюватися.

У прикладах вище ми скористалися двома вбудованими функціями. Як неважко здогадатися з контексту, solve – це функція рішення рівняння, а diff – функція диференціювання. Практично весь функціонал Maxima реалізований через такі вбудовані функції. Функція в Maxima може мати змінне число аргументів. Наприклад, функція solve, яку використовували з одним аргументом, частіше викликається з двома аргументами. Перший задає рівняння або функцію, корінь якої потрібно знайти; другий – змінну, щодо якої потрібно вирішувати рівняння:



(%i1)  $\frac{a}{x} + ax - a^2$

(%i2) solve(%i1, a)

(%o2)  $\left[ a = \frac{x^2 + 1}{x}, a = 0 \right]$

(%i3) solve(%i1, x)

(%o3)  $\left[ x = -\frac{\sqrt{a^2 - 4} - a}{2}, x = \frac{\sqrt{a^2 - 4} + a}{2} \right]$

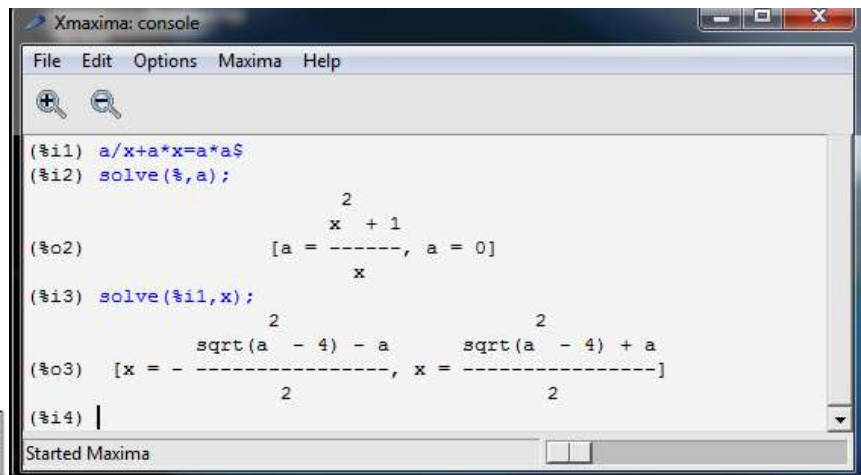


Рисунок 11.13 – Введення команд в режимі TeXmacs

Якщо формула, що задає вирішуване рівняння, містить тільки один символ, як у попередньому прикладі, то другий аргумент можна опустити, так як вибір, щодо чого потрібно вирішувати рівняння, все одно однозначний.

Друга функція з наших нових знайомих **diff** – також може приймати один аргумент; в цьому випадку вона знаходить диференціал заданого виразу:

(%i1) diff( $x y + \frac{y}{x}$ )

(%o1)  $\left( x + \frac{1}{x} \right) \text{del}(y) + \left( y - \frac{y}{x^2} \right) \text{del}(x)$

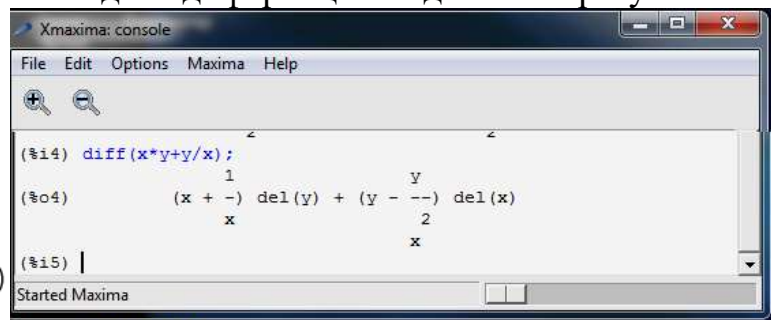


Рисунок 11.14 – Введення команд в режимі TeXmacs

Через del (x) і del (y) позначені диференціали відповідних символів.

Для кожної вбудованої функції є опис в документації по Maxima. Вона містить відомості про те, які аргументи і в яких випадках приймає функція, а також опис її дії в різних випадках і конкретні приклади застосування. Але, звичайно, шукати опис кожної потрібної функції в html-документації або info-сторінках не завжди зручно, тим більше, що потрібна ця інформація, як правило, прямо в процесі роботи. Тому в Maxima є спеціальна функція – **describe ()**, яка видає інформацію з документації по конкретним словам. Більше того, спеціально для зручності отримання довідкової інформації існує скорочена версія виклику цієї функції: **name** замість **describe (name)**. Це ім'я оператора, і аргумент потрібно відокремлювати від нього прогалиною (вираз? Name використовується для виклику функції Lisp з ім'ям name). Функція **describe і оператор?** видають список тих розділів допомоги та імен функцій, які містять заданий текст, після чого пропонують ввести номер того розділу або опису тієї функції, які користувач хоче подивитися:

```

Xmaxima: console
File Edit Options Maxima Help

(%i5) describe(solve);
-- Function: solve
    solve (<expr>, <x>)
    solve (<expr>)
    solve ([<eqn_1>, ..., <eqn_n>], [<x_1>, ..., <x_n>])
Solves the algebraic equation <expr> for the variable <x> and
returns a list of solution equations in <x>. If <expr> is not an
equation, the equation '<expr> = 0' is assumed in its place. <x>
may be a function (e.g. 'f(x)'), or other non-atomic expression
except a sum or product. <x> may be omitted if <expr> contains
only one variable. <expr> may be a rational expression, and may
contain trigonometric functions, exponentials, etc.

The following method is used:

Let <E> be the expression and <X> be the variable. If <E> is
linear in <X> then it is trivially solved for <X>. Otherwise if
<E> is of the form 'A*X^N + B' then the result is '(-B/A)^1/N'
times the 'N'th roots of unity.

If <E> is not linear in <X> then the gcd of the exponents of <X> in
<E> (say <N>) is divided into the exponents and the multiplicity
of the roots is multiplied by <N>. Then 'solve' is called again
on the result. If <E> factors then 'solve' is called on each of
the factors. Finally 'solve' will use the quadratic, cubic, or
quartic formulas where necessary.

```

Рисунок 11.15 – Введення команд в режимі TeXmacs

```

Maxima в командній стрічці
Maxima 5.35.1.2 http://maxima.sourceforge.net
using Lisp SBCL 1.2.7
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1) ? diff

@: diff <Functions and Variables for itensor>
1: diff <1> <Functions and Variables for Differentiation>
Enter space-separated numbers, 'all' or 'none': all

-- Function: diff <<expr>, <v_1>, [<n_1>, [<v_2>, <n_2>] ...1]
is the usual Maxima differentiation function which has been
expanded in its abilities for 'itensor'. It takes the derivative
of <expr> with respect to <v_1> <n_1> times, with respect to <v_2>
<n_2> times, etc. For the tensor package, the function has been
modified so that the <v_i> may be integers from 1 up to the value
of the variable 'dim'. This will cause the differentiation to be
carried out with respect to the <v_i>th member of the list
'vect_coords'. If 'vect_coords' is bound to an atomic variable,
then that variable subscripted by <v_i> will be used for the
variable of differentiation. This permits an array of coordinate
names or subscripted names like 'x[1]', 'x[2]', ... to be used.

A further extension adds the ability to 'diff' to compute
derivatives with respect to an indexed variable. In particular,
the tensor package knows how to differentiate expressions
containing combinations of the metric tensor and its derivatives
with respect to the metric tensor and its first and second
derivatives. This capability is particularly useful when
considering Lagrangian formulations of a gravitational theory,
allowing one to derive the Einstein tensor and field equations
from the action principle.

-- Function: diff
    diff <<expr>, <x_1>, <n_1>, ..., <x_m>, <n_m>
    diff <<expr>, <x>, <n>
    diff <<expr>, <x>
    diff <<expr>
Returns the derivative or differential of <expr> with respect to
some or all variables in <expr>.

'diff <<expr>, <x>, <n>' returns the <n>'th derivative of <expr>
with respect to <x>.

```

Рисунок 11.16 – Введення команд в командній стрічці

```
(%i5) ?diff

0: (maxima.info)Differentiation.
1: Definitions for Differentiation.
2: Differential Equations.
3: Definitions for Differential Equations.
4: antidiff :Definitions for Differentiation.
5: covdiff :Definitions for itensor.
6: diff <1> :Definitions for itensor.
7: diff :Definitions for Differentiation.
8: evundiff :Definitions for itensor.
9: extdiff :Definitions for itensor.
10: idiff :Definitions for itensor.
11: liediff :Definitions for itensor.
12: poisdiff :Definitions for Special Functions.
13: ratdiff :Definitions for Polynomials.
14: rediff :Definitions for itensor.
15: setdifference :Definitions for Sets.
16: symmdifference :Definitions for Sets.
17: undiff :Definitions for itensor.

Enter space-separated numbers, 'all' or 'none': |
```

Рисунок 11.17 – Інформація з командної стрічки  
Коли користувач вибирає розділ, буде видано його вміст:

```
Enter space-separated numbers, 'all' or 'none': 7

Info from file /usr/share/info/maxima.info:
-- Function: diff (<expr>, <x_1>, <n_1>, ..., <x_m>, <n_m>)
-- Function: diff (<expr>, <x>, <n>)
-- Function: diff (<expr>, <x>)
-- Function: diff (<expr>)
Returns the derivative or differential of <expr> with respect to
some or all variables in <expr>.

'diff (<expr>, <x>, <n>)' returns the <n>'th derivative of <expr>
with respect to <x>.

'diff (<expr>, <x_1>, <n_1>, ..., <x_m>, <n_m>)' returns the mixed
partial derivative of <expr> with respect to <x_1>, ..., <x_m>.
It is equivalent to 'diff (... (diff (<expr>, <x_m>, <n_m>) ...),
<x_1>, <n_1>)'.

'diff (<expr>, <x>)' returns the first derivative of <expr> with
respect to the variable <x>.
```

Рисунок 11.18 – Інформація з командної стрічки

Якщо для слова, яке ввів користувач після ? або **describe**, знайдено єдиний збіг, його опис буде показано відразу.

Крім довідки, з багатьох функцій Maxima є приклади їх використання. Приклад можна завантажити функцією `example ()`. Виклик цієї функції без аргументу відобразить список всіх імен доступних прикладів; виклик виду `example (name)` завантажить в поточну сесію і виконає вказаний файл прикладу:

```

Xmaxima: console
File Edit Options Maxima Help

^
(%i12) example (solve);
(%i13) solve(asin(cos(3*x))*(f(x)-1),x)

solve: using arc-trig functions to get a solution.
Some solutions will be lost.
          %pi
(%o13)      [x = ----, f(x) = 1]
           6

(%i14) ev(solve(5^f(x) = 125, f(x)), solveradcan)
          log(125)
(%o14)      [f(x) = -----]
           log(5)

(%i15) [4*x^2-y^2 = 12, x*y-x = 2]
          2      2
(%o15)      [4 x  - y  = 12, x y - x = 2]
(%i16) solve(%,[x,y])
(%o16) [[x = 2, y = 2], [x = 0.5202594388652008 %i
- 0.1331240357358706, y = 0.07678378523787788
- 3.608003221870287 %i], [x =
- 0.5202594388652008 %i - 0.1331240357358706,
y = 3.608003221870287 %i + 0.07678378523787788],
[x = - 1.733751846381093, y = - 0.1535675710019696]]
(%i17) solve(1+a*x+x^3,x)
          3
          sqrt(3) %i      1      sqrt(4 a  + 27)      1 1/3
(%o17) [x = (- ---- - -) (----- - -)]
           2          2          3/2          2
  
```

Рисунок 11.19 – Введення команд в режимі TeXmacs

Те, що Maxima написана на Lisp, людині, знайомій з цією мовою, стає зрозуміло вже на початку роботи з програмою. Дійсно, в Maxima чітко простежується «лісповській» принцип роботи з даними, який виявляється дуже до речі в контексті символічної математики та аналітичних обчислень. Справа в тому, що в Lisp, за великим рахунком, немає поділу на об'єкти і дані: імена змінних і вирази можуть використовуватися практично в одному і тому ж контексті. У Maxima ж це властивість розвинене ще сильніше: фактично, ми можемо використовувати будь-який символ незалежно від того, присвоєно йому якийсь вираз. За замовчуванням символ, пов'язаний з будь-яким виразом, представлятиме цей вираз; символ, не пов'язаний ні з чим, представлятиме самого себе, трактуемого знову-таки як вираз. Пояснимо на прикладі:

(%i1)  $ab$   
 (%o1)  $ab$   
 (%i2)  $a:x+yb:x-y$   
 (%i4)  $ab$   
 (%o4)  $(x-y)(y+x)$

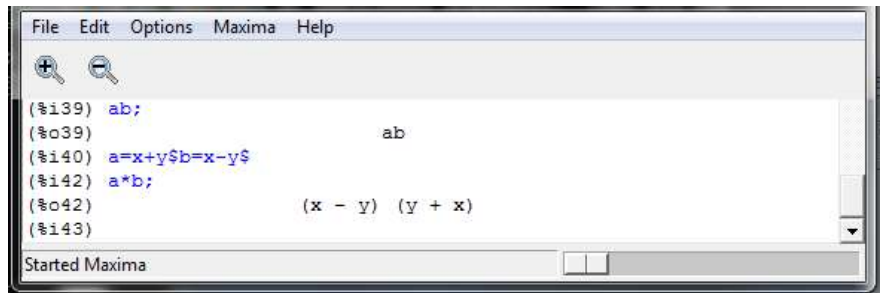


Рисунок 11.20 – Введення команд в режимі TeXmacs

З цього випливає, зокрема, що у виразі автоматично підставляється значення вхідного в нього символу тільки в тому випадку, якщо це значення було приписано символу до визначення виразу:

(%i5)  $x:\frac{1}{2}y:\frac{1}{3}$   
 (%i7)  $c:x-y$   
 (%i8)  $ab;ac$   
 (%o8)  $(x-y)(y+x)$

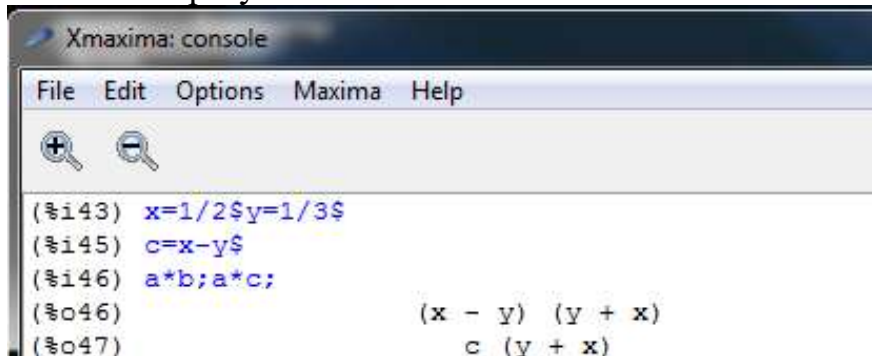


Рисунок 11.21 – Введення команд в режимі TeXmacs

Якщо деякий символ вже має якесь значення, чи можемо ми використовувати у виразі сам цей символ, а не його значення? Звичайно. Зробити це можна за допомогою знака апострофа – введений перед будь-яким символом або виразом, він запобігає його обчисленню:

(%i10)  $b+y$   
 (%o10)  $-y+x+\frac{1}{3}$   
 (%i11)  $b+'y$   
 (%o11)  $x$   
 (%i12)  $'(b+y)$   
 (%o12)  $y+b$

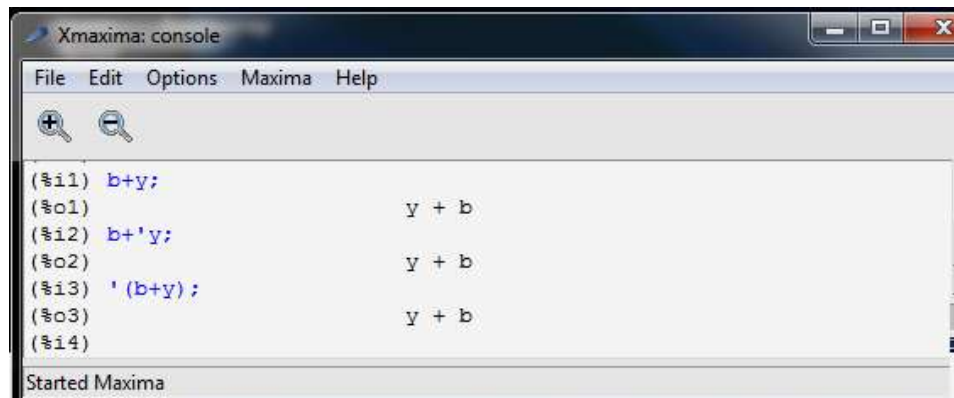


Рисунок 11.22 – Введення команд в режимі TeXmacs

Результат виразу % i12 був би аналогічний і в тому випадку, якби  $b$  і  $y$  не мали на той момент ніяких значень; таким чином, ми можемо сміливо блокувати обчислення символу, навіть не запам'ятовуючи (або не знаючи), чи присвоєні їм взагалі якісь вирази.

Точно так само можна вчинити з будь-якою вбудованою функцією, якщо ми хочемо не виконати її, а використати її у своєму математичному контексті. Наприклад, вже згадана функція диференціювання може стати в нагоді нам для

позначення похідної в диференціальному рівнянні; в цьому випадку, звичайно, обчислювати її не треба:

(%i1) 'diff(y, x) = y

(%o1)  $\frac{d}{dx} y = y$

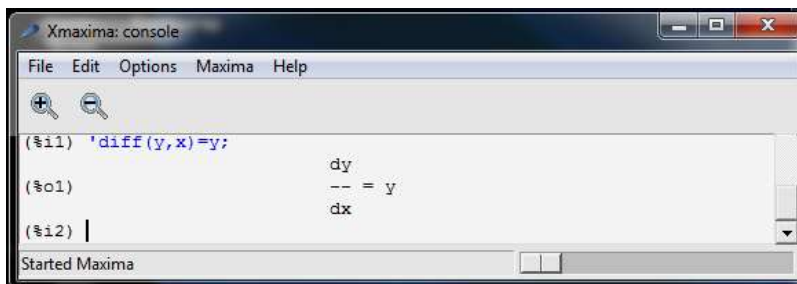


Рисунок 11.23 – Введення команд в режимі TeXmacs

Завдяки описаним особливостям робота в Maxima, з одного боку, стає багато в чому схожою на традиційну «ручну» роботу з математичними формулами, що практично зводить нанівець психологічний бар'єр на початку роботи з програмою. З іншого боку, навіть на цьому початковому етапі ви фактично позбавлені від найбільш рутинної ручної роботи, на зразок відстеження поточних значень символів, і можете повністю зосередитися на самій задачі. Звичайно, блокування обчислень – це не єдиний спосіб впливати на те, як Maxima буде обчислювати той чи інший вираз; цим процесом можна керувати досить гнучко.

Розглянемо розв'язок екстремальних задач в Maxima. В даній програмі є процедура `maximize_lp` (`minimize_lp`), за допомогою якої можна знайти екстремум лінійної функції з обмеженнями у вигляді лінійних нерівностей. Але в системі Maxima екстремальні задачі можна розв'язати і за допомогою класичних методів (рис. 11.24).

$$f(x,y)=2.5x^2-3.1xy+4.6y^2-x+5.2y$$

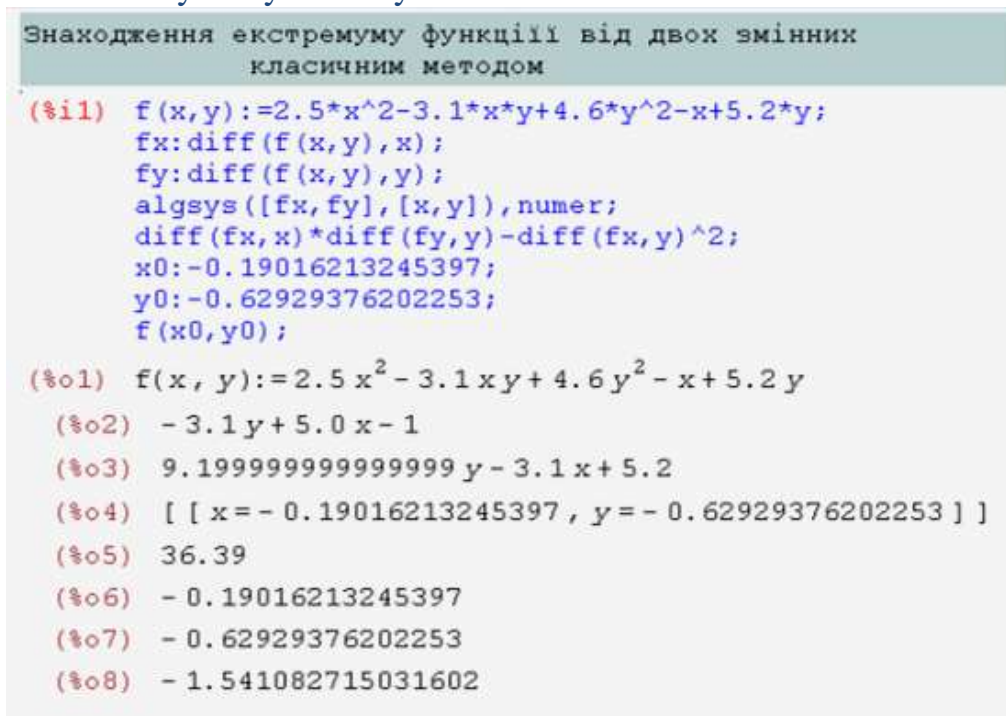


Рисунок 11.24 – Розв'язування екстремальної задачі

## II. Задання до лабораторної роботи

1. Прочитати теоретичний матеріал лабораторної роботи.
2. Познайомитися з основними елементами СКМ Махіма.
3. З допомогою панелі інструментів набрати приклади з рис. 11.1-11.24
4. Познайомитися з довідковою системою СКМ Махіма.
5. Підготувати звіт про виконання завдань за допомогою Махіма у форматі Tex та надіслати до СПДН ФІТІС.

## III. Індивідуальні завдання

### 1. Виконати обчислення

- |                       |                        |
|-----------------------|------------------------|
| 1) $7/8+42/16+23/9=$  | 7) $3/8+56/1+12/9=$    |
| 2) $9/2+78/6+3/15=$   | 8) $25/3+51/3+4/9=$    |
| 3) $2/9+78/2+19/7=$   | 9) $41/7+45/7+59/2=$   |
| 4) $15/6+47/6+23/9=$  | 10) $41/3+63/9+22/3=$  |
| 5) $23/9+65/9+25/7=$  | 11) $3/8+9/8+5/16=$    |
| 6) $17/12+45/13+2/9=$ | 12) $1/15+8/19+56/21=$ |

### 2. Розв'язати рівняння та подумувати графік функції

- |                |                     |
|----------------|---------------------|
| 1) $x^4-9x=0$  | 7) $2x-4x=8$        |
| 2) $x^5-x=10$  | 8) $7x^5-2x=15$     |
| 3) $x^5-4x=10$ | 9) $5x^3-2x^2-x=4$  |
| 4) $x-2x=7$    | 10) $x^4+x^5+x^6=2$ |
| 5) $x^3-x^2=2$ | 11) $x^2-3x=8$      |
| 6) $x-6x=3$    | 12) $x^4=7$         |

*Примітка:* приклад розв'язування рівняння та побудова графіка функції

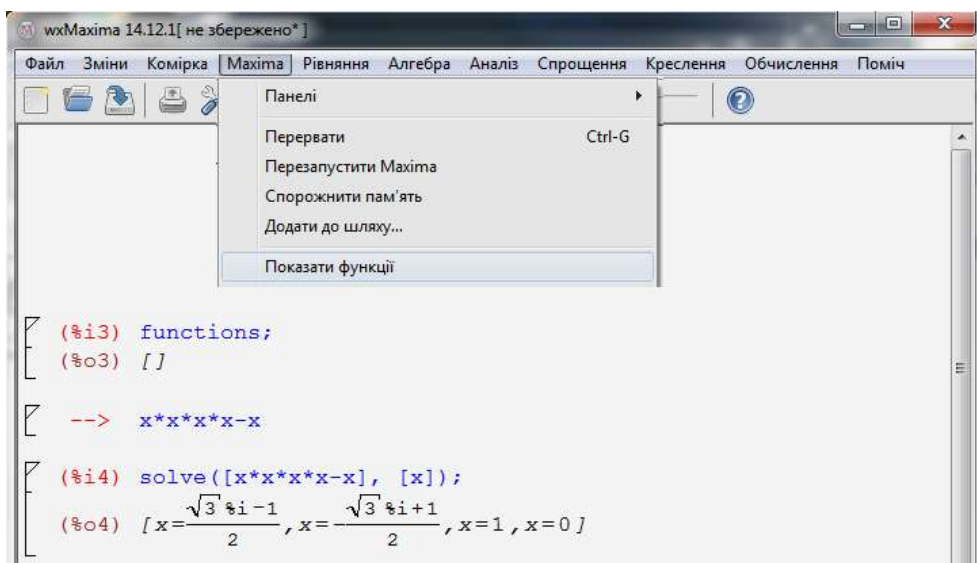


Рисунок 11.25 – Запис функції

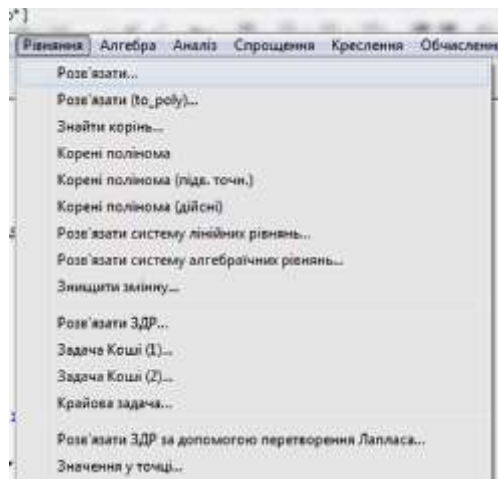


Рисунок 11.26 – Робота в меню

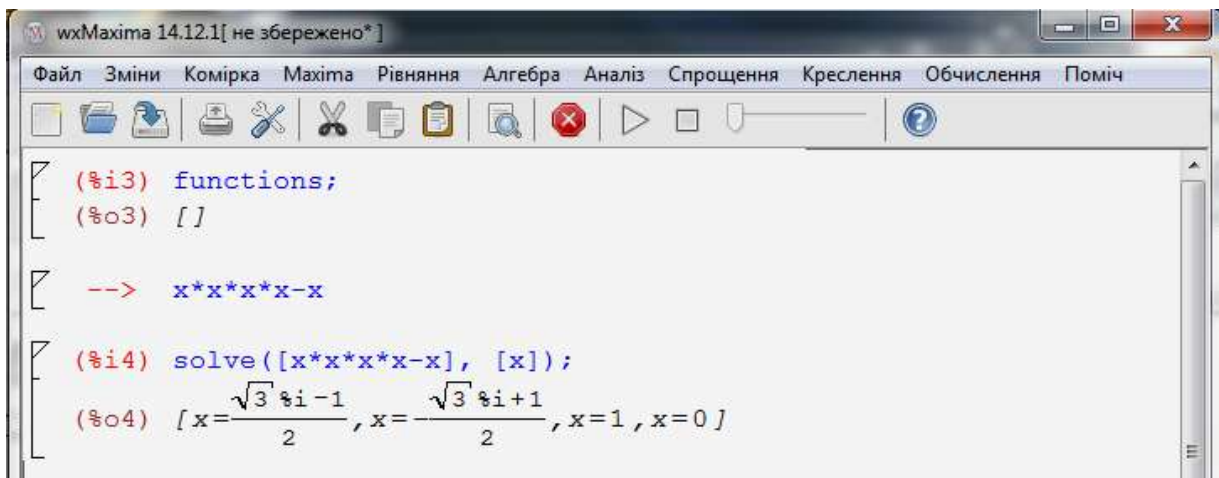


Рисунок 11.27 – Введення команд в режимі TeXmacs

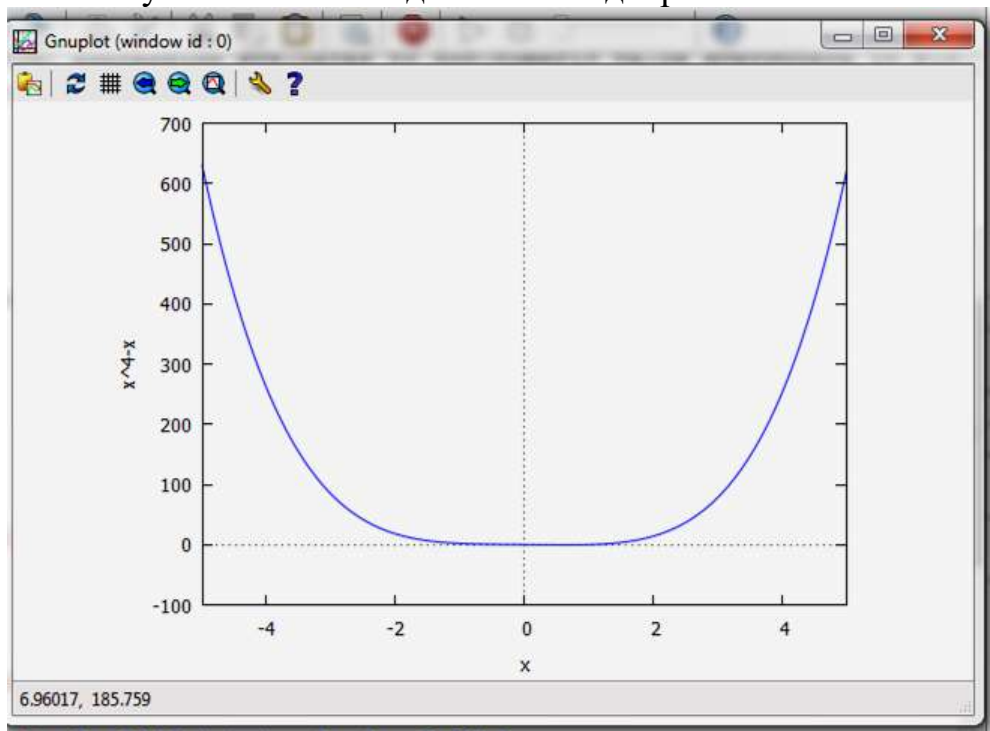


Рисунок 11.28 – Виведення графіка функції



3. Знайти екстремум функції за допомогою класичного методу та з використанням вбудованих функцій `maximize_lp` (`minimize_lp`). Побудувати графік функції та порівняти його з графіком функції в Mathcad:

1)  $f(x) = x_1^2 + x_2^2 - x_1 x_2^2$ ;

2)  $f(x) = x_1^3 + x_2^3 - 5x_1 x_2$ ;

3)  $f(x) = x_1^3 + x_2^3 - x_1^2 - 2x_1 x_2 - x_2^2$ ;

4)  $f(x) = x_1 x_2 + \frac{1}{2(x_1 + x_2)}$ ;

5)  $f(x) = \sin(x_1 + x_2) - \sin x_1 - \sin x_2$ ;

6)  $f(x) = 3x_1^4 + 2x_2^4 - 5x_1 x_2$ ;

7)  $f(x) = 10 \ln x_1 + x_1 x_2^2 - x_2^3$

8)  $f(x) = (x_1 - x_1 x_2 + x_2)(x_1 + x_2)$ ;

9)  $f(x) = 2x_1^2 + x_2^2 + 10x_1 x_2$ ;

10)  $f(x) = x_1 x_2^2 (1 - x_1 - 2x_2)$ .

## Лабораторна робота № 12

Тема: «Знайомство з MatLab»

Питання:

1. Отримання перших навичок роботи з системою комп'ютерної математики MatLab.
2. Знайомство з основними операціями над векторами і матрицями в MatLab.
3. Знайомство з найпростішими графічними засобами MatLab.
4. Рішення завдання по обчисленню суми числового ряду та аналізу похибок отриманих результатів.

### I. Теоретичний матеріал

Перехід від програмування в кодах ЕОМ до універсальних мов програмування поклав початок створенню бібліотек алгоритмів, які за багато років склали базу для побудови систем для вирішення математичних завдань, що виникають в різноманітних наукових дослідженнях і технічних розробках – Maple, Mathematica, MathCad, MatLab та ін. Для цих систем характерні простота підготовки даних, зручні форми виводу результатів обчислень, вбудовані засоби допомоги та діагностики помилок – так званий дружній інтерфейс.

Система MatLab (Matrix Laboratory) є інтерактивною системою для виконання інженерних і наукових розрахунків, орієнтована на роботу з масивами даних, в першу чергу з матрицями. Вона допускає написання на спеціальній мові програм, оформлених у вигляді М-файлів, підтримує роботу в програмному і інтерактивному режимі з векторами і матрицями, надає можливість вирішувати системи рівнянь, виконувати чисельне інтегрування, будувати графіки та ін. В MatLab вбудовано безліч пакетів прикладних програм (Toolbox), таких як:

- символічної математики;
- статистики;
- оптимізації;
- аналізу і синтезу систем управління;
- обробки сигналів та зображень;
- фінансів;
- картографії та ін.

Детальне оволодіння будь-якою великою програмною системою – це досить тривалий процес, основу якого складають індивідуальна робота, і заняття покликані дати лише початковий імпульс цьому процесу щодо MATLAB.

### II. Задання до лабораторної роботи

#### *Інтерфейс MatLab*

Запустити систему MatLab, запуску системи визиває появу головного вікна (рис. 12.1).

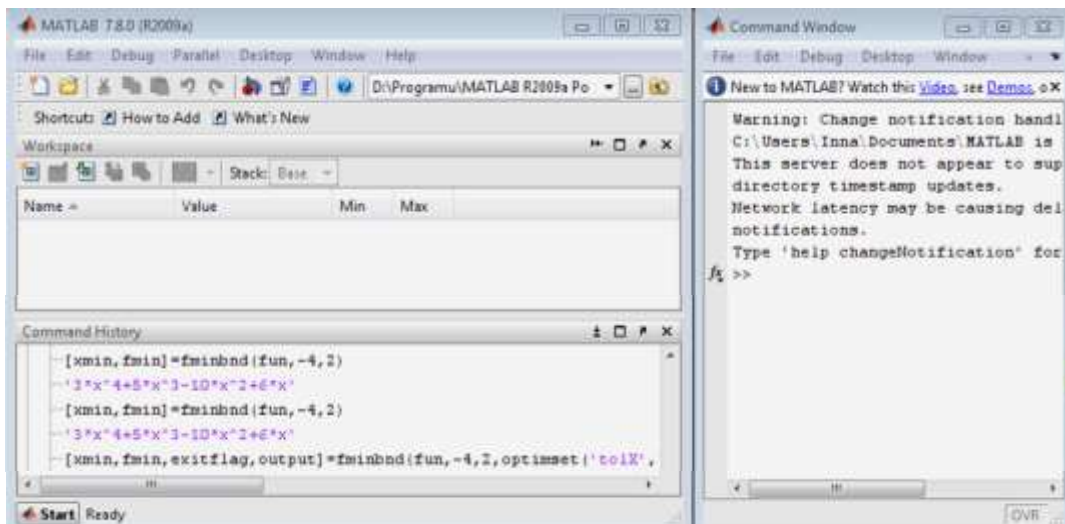


Рисунок 12.1 – Головне вікно MatLab

З допомогою команди меню **Desktop / Desktop Layout / Command Window Only** головне вікно MatLab перетвориться у скорочене (лише командне вікно) (рис. 12.2).

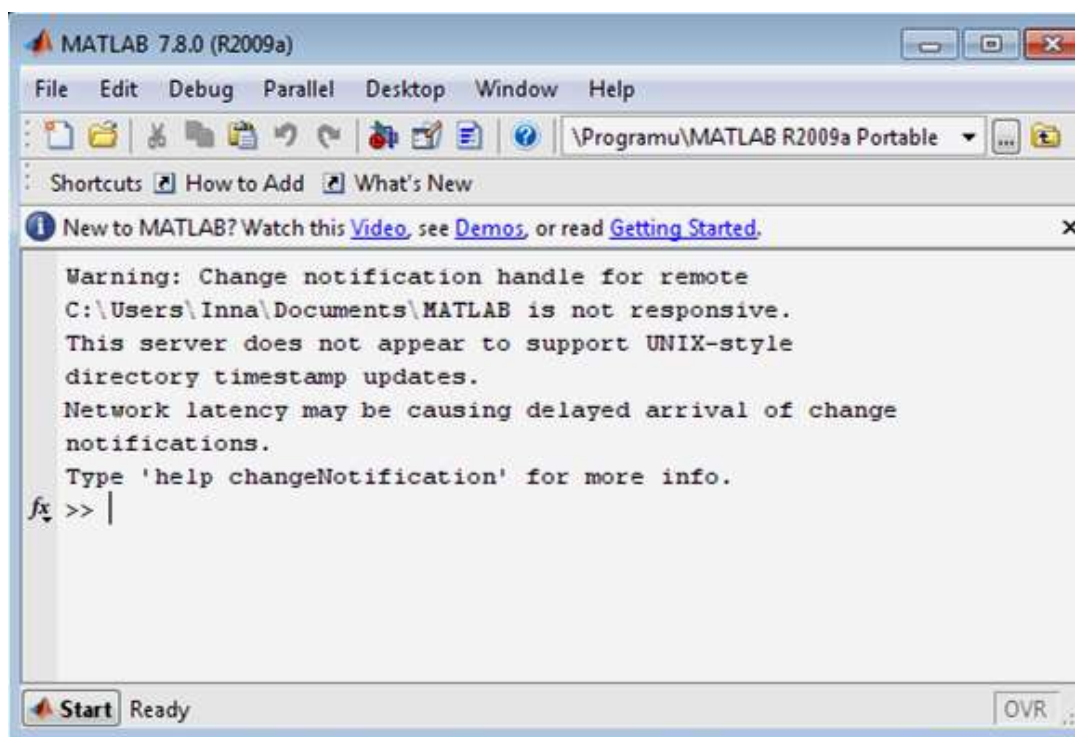


Рисунок 12.2 – Спрощений вигляд головного вікна MatLab

Зворотнє перемикання можна здійснити за допомогою команди меню **Desktop / Desktop Layout / Default**.

### **Командне вікно (Вікно Command Window)**

Вікно Command Window є для користувача найбільш важливим. За допомогою цього вікна (рис. 12.2) вводяться математичні вирази, виводяться результати обчислень, а також відображаються повідомлення, що надсилаються системою.

Математичні вирази пишуться в командному рядку після знака запрошення ». Кожна команда в командному вікні обробляється негайно після натискання клавіші Enter, при цьому значення вихідних параметрів виводяться в тому ж командному вікні, а для рисунків відкриваються графічні вікна.

Набрати в рядку введення вираз:

```
>> x=2+3
```

Для виконання дії потрібно натиснути клавішу <Enter>. Результат наступний:

```
>> x=2+3
```

```
x =
```

```
5
```

```
>>
```

Якщо потрібно виправити одну з цифр, то нічого не вийде. Нemoжливiсть редагування раніше введеної команди простою установкою курсору в потрібний рядок є однією з особливостей системи MatLab.

Система «пам'ятає» введені команди, вони завантажуються в стек, тому іноді немає необхідності набирати їх заново, а досить знайти в передісторії сеансу (у вікні історії команд) схожу команду, мишкою перетягнути її в командне вікно, відредагувати і виконати натисканням Enter.

Крім того, клавіша "↑" викликає появу останньої команди, повторне натискання переводить до передостанньої, і т.д. Стек команд закріплений і в зворотній послідовності команди викликаються клавішею "↓" (перша команда сеансу, друга і т.д.).

Очистити командне вікно можна з допомогою меню **EDIT / Clear Command Window**.

### ***Робоча область (Вікно Workspace)***

У процесі роботи використовуються змінні різних типів. Створені змінні зберігаються в спеціально відведеній області пам'яті комп'ютера. Вони не зникають самі по собі, а тільки при виході з програми або за допомогою спеціальних команд. При цьому змінні (точніше їх значення) можна використовувати в будь-якому математичному виразі, що вводиться. Вікно Workspace (Робоча область) надає користувачу список всіх змінних, що зберігаються в робочому просторі (див. рис. 12.1). Вибрати можна будь-яку змінну, переглянути її вміст або виконати будь-які інші дії. Згадані дії виконуються за допомогою контекстного меню (потрібно натиснути правою кнопкою миші по імені змінної у списку).

Очистити робочу область можна за допомогою меню **EDIT / Clear Workspce**.

### ***Вікно Current Directory (Поточний каталог)***

Є аналогом відомої програми Провідник, але має для MatLab своє особливе призначення (рис. 12.3).

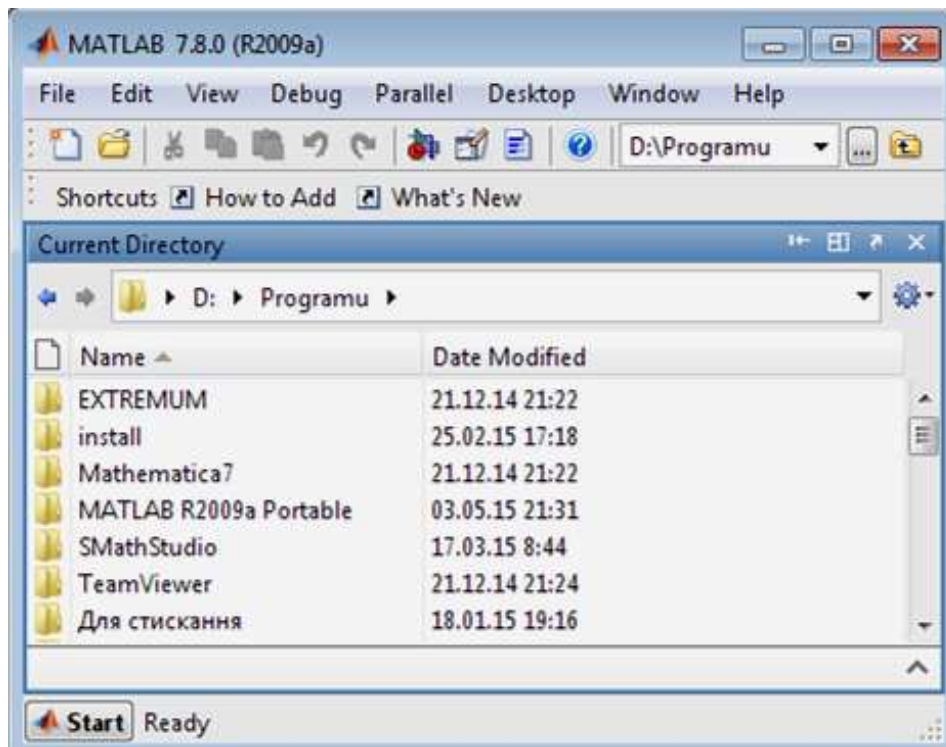


Рисунок 12.3 – Вікно Current Directory (Поточний каталог)

Річ в тім, що, крім роботи з математичними виразами у командному вікні, користувач також може працювати з файлами. До того ж математичні функції, які ми використовуємо, фізично являють собою файли, названі по іменах функцій. У цих файлах записані програми, що реалізують функції. Таким чином, користувач постійно використовує файли. Наприклад, вказуючи вбудовану функцію, ми фактично пишемо ім'я файлу (без розширення), в якому зберігається текст програми. Система виконує пошук таких файлів в першу чергу в поточному каталозі.

**Завдання:** у провіднику Windows створити папку зі своїм прізвищем (латинськими символами, це особливість MatLab) в місці, вказаному викладачем. Усередині цієї папки створити папку для лабораторної роботи, наприклад LAB 23-24. За допомогою правої кнопки вибору поточного каталогу перейдіть в даний каталог. Спочатку він повинен бути порожнім. Надалі кожен сеанс роботи з MatLab повинен починатися з вибору вашого поточного каталогу.

### ***Вікно історії команд (Command History)***

Усі команди, які набираються в командному рядку Command Window (Вікно команд), автоматично утворюють список, який і виводиться у вікні Command History (Історія команд) (рис. 12.4). Чим корисний цей список? Якщо з'явилася необхідність повторити раніше виконану команду, її можна відшукати в списку Command History (Історія команд) і, двічі клацнувши по ній лівою кнопкою миші, виконати. Вміст даного вікна не втрачається після виходу з системи та вимкнення комп'ютера.

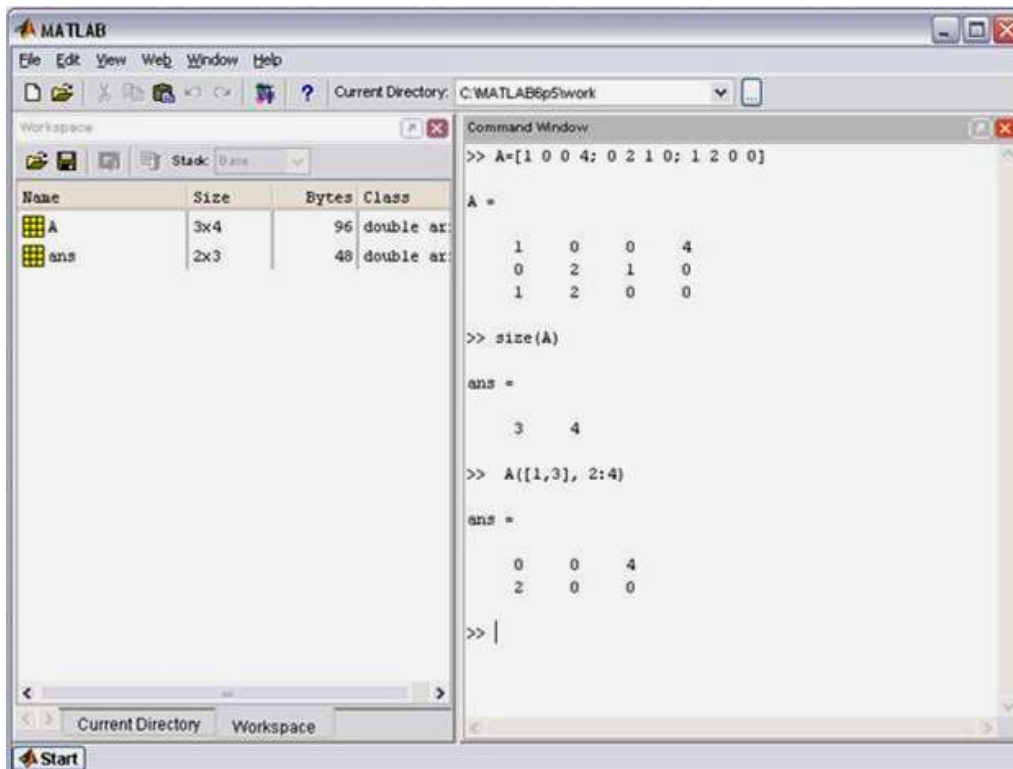


Рисунок 12.4 – Приклад зображення вікна «Історія команд»

Очистити список команд можна тільки за допомогою меню **EDIT / Clear Command History**.

### *Елементи мови програмування MatLab. Робота з матрицями*

Сформуємо кілька змінних, знайомлячись попутно з елементами мови. Виконаємо команду (елементи рядка матриці при введенні розділяються комою або пробілом; рядки відокремлюються один від одного крапкою з комою) формування матриці (рис. 4).

```
>>A=[1 0 0 4; 0 2 1 0; 1 2 0 0]
```

Завершення командного рядка символом ";" в системі MatLab означало б придушення виведення результату в командне вікно. Наша команда не закінчується крапкою з комою, тому в командному вікні з'являється відповідь у вигляді

```
A =
 1  0  0  4
 0  2  1  0
 1  2  0  0
```

Розмір змінної у форматі вектора з двох компонент (<число рядків> <число стовпців>) повертає функція **size**. В даному випадку

```
>>size(A)
```

Команда

```
>>A([1,3], 2:4)
```

формує нову матрицю з елементів матриці A, що знаходяться на перетині рядків з номерами 1 та 3 і стовпців з номерами 2, 3, 4, привласнюючи результат змінної **ans** (answer), яка завжди містить результат обчислення останнього виразу, якщо він не був привласнений будь-якою змінною. В даному випадку

```
ans =
    0    0    4
    2    0    0
```

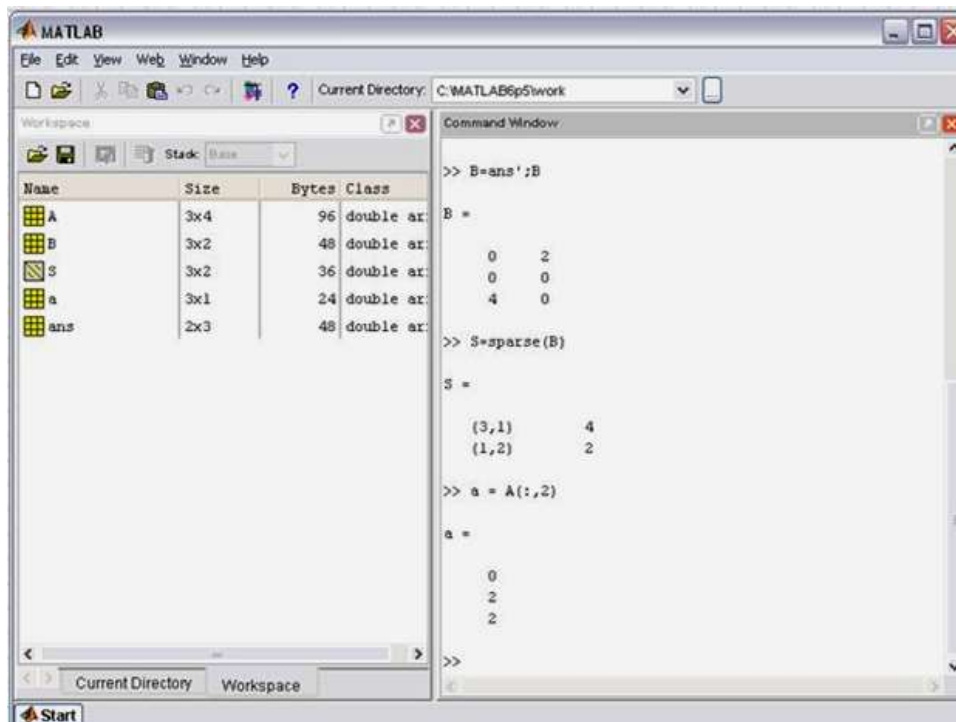


Рисунок 12.5 – Командне вікно і браузер робочої області

Змінною ans можна оперувати як і будь-якою іншою змінною. На рис. 12.5 матриця ans транспонується (символ «апостроф»), і результат отримує ім'я B (с придушенням виводу). Потім запитується вміст B командою, що складається виключно з ідентифікатора цієї змінної. В одному рядку допускається кілька команд, розділених символами "," або ";". При цьому роздільник "," означає виведення результату тієї операції, після якої він поставлений. Сформуємо вектор-стовпець, що співпадає з другим стовпцем матриці A

```
>>a = A(:,2)
```

```
a=
0
2
2
```

Роздільник ":" використовується для формування списків або векторів. Вираз  $n: m$  означає список  $n, n + 1, \dots, m$ ; а вираз  $n: p: m$  – список від  $n$  до  $m$  кроком  $p$ ; двокрапка без меж означає весь проміжок зміни індексу, в тому числі подвійного. Так команда  $A (:)$  розгорнула б матрицю A за стовпцями в один «довгий» вектор-стовпець.

Виконаємо тепер множення, використовуючи команду **disp** для виведення результатів. При такому висновку MatLab не повторює ім'я змінної, в іншому ж команди  $A$  і  $\text{disp}(A)$  еквівалентні.

```
>>C=a*a'; disp(C)
```

```
0 0 0
0 4 4
0 4 4
```

```
>>d=(a')*a; disp(d)
```

Твір матриці розміру  $[nr]$  на матрицю розміру  $[rm]$  є матриця розміру  $[nm]$ , тому тут  $C = a * a'$  – матриця, а  $d$  – скалярний квадрат вектора  $a$ . Точка в поєднанні зі знаком операції означає, що операція виконується поелементно. Так  $.*$  означає поелементне множення.

```
>>g=a.*a; disp(g')
```

```
0 4 4
```

Запитаємо тепер систему про вміст робочої області. Запит **whos** змушує систему видати докладну відповідь.

Таблиця на рис. 12.6 ілюструє дві важливі особливості MatLab:

- MatLab розрізняє імена, набрані в різних регістрах;
- в MatLab вектори і скалярні змінні трактуються як матриці розміру  $[1\ m]$ ,  $[n\ 1]$  і  $[1\ 1]$ .

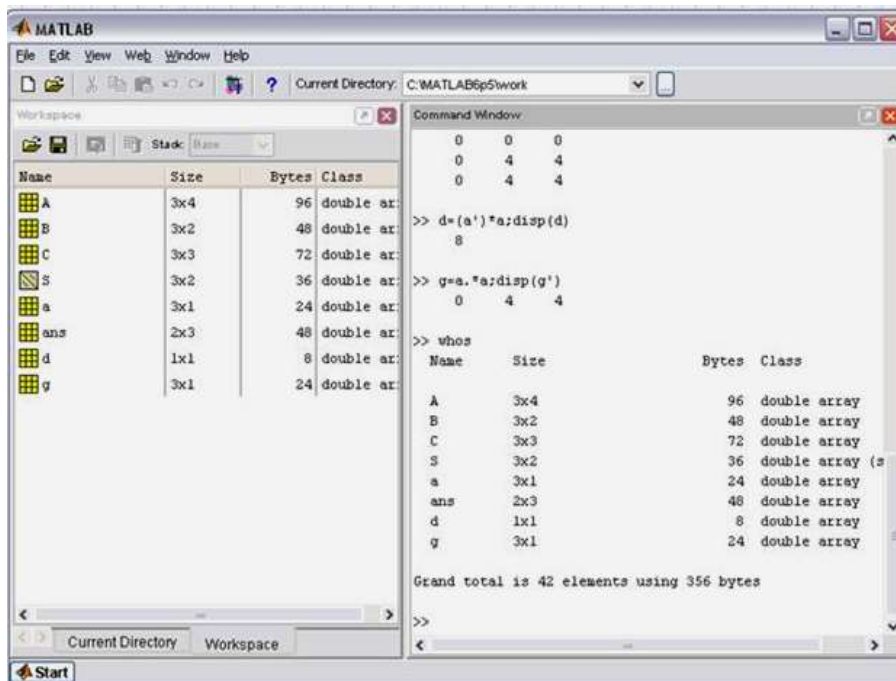


Рисунок 12.6 – Таблиця з вмістом робочої області в командному вікні

## Графіка в MatLab

Докладний опис всіх можливостей MatLab виходить далеко за рамки лабораторної роботи. Тому обмежимося лише розглядом найпростіших графічних засобів, які використовуються в MatLab.

### *Пробудова графіка функцій однієї змінної*

Графіки будуються в окремих масштабованих і переміщуваних вікнах.



Для початку візьмемо найпростіший приклад – побудова графіка синусоїди. Слід пам'ятати, що MatLab будує графіки функцій по ряду точок, з'єднуючи їх відрізками прямих, тобто застосовує лінійну інтерполяцію функції на проміжку між суміжними точками. Задамо інтервал змінюючи аргумент  $x$  від 0 до 10 з кроком 0.1. Для побудови графіка потрібно спочатку задати вектор  $x=0:0.1:10$ , а потім використати команду побудови графіків **plot(sin(x))**. Результат наведено нище.

```
>> x=0:pi/10:10;  
>> plot(x,sin(x))
```

**Завдання:** побудувати даний графік.

Вектор  $x$  задає інтервал незалежної змінної від 0 до 10 з кроком 0.1. Чому взято такий крок, а не припустим, 1? Річ в тім, що **plot** будує не істиний графік функції а лише  $\sin(x)$ , так як лише задане числом елементів вектора  $x$  число точок. Ці точки потім з'єднуються відрізками прямих, тобто здійснюється кусково-лінійна інтерполяція даних графіка. При 100 точках отримана крива оком сприймається як досить плавна, але при 10-20 точках вона буде виглядати ніби складеною з відрізків.

Графіки MatLab будує в окремих вікнах, що називаються *графічними вікнами*. Меню графічного вікна показано на рис. 12.7, воно відрізняється від головного вікна MatLab. Наприклад, в меню вікна з'явилася позиція **Tools** (Інструменти) та **Insert** (Вставка). Засоби цієї панелі надають можливість легко керувати параметрами та наносити на них текстові коментарі в будь-якому місці.

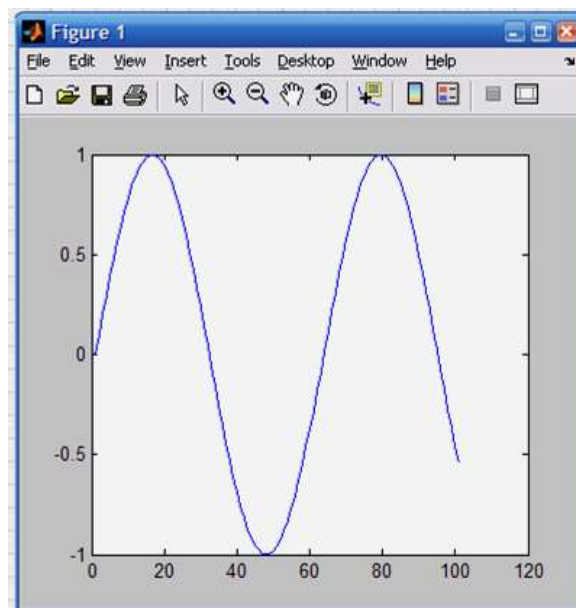


Рисунок 12.7 - Приклад побудови графіка синусоїди

Меню **File** (Файл) вікна графічних побудов містить типові файлові операції, зокрема збереження графічних вікон в файли, та їх подальше відкриття. Однак вони відносяться не до файлів документа, а до файлів графіків. Зокрема, можна надавати ім'я рисункам, що записуються на диск.

**Завдання:** зберегти графік (рис. 12.7) в файл під ім'ям **rus07**.

**Побудова кількох графіків функцій в одному вікні**

Спробуємо побудувати графіки відразу трьох функцій:  $\sin(x)$ ,  $\cos(x)$  і  $\log(x)$ .

```
>> x=0:0.1:10;
```

Значимо, що змінні  $y_1$ ,  $y_2$ ,  $y_3$  в результаті виконання даних операцій являються не функціями, а **векторами**, які мають ту ж розмірність що і вектор  $x$ . Тому ці змінні не мають явно вказаного аргументу у вигляді  $y(x)$ . Потрібно звернути увагу, що **крапка з комою** надає можливість писати кілька команд в одній стрічці.

У нашому випадку для побудови графіків вказаних функцій ми повинні написати наступне:

```
>> plot(x,y1,x,y2,x,y3)
```

де  $x_1$ ,  $x_2$ ,  $x_3$  ... - вектори аргументів (у нашому випадку всі вони дорівнюють  $x$ ), а  $y_1$ ,  $y_2$ ,  $y_3$  ... - вектори значень функції, графіки яких будуються в одному вікні.

Тепер можна скористатися однією з форм команди **plot**:

```
>> plot(x,sin(x),x,cos(x),x,log(x))
```

**Завдання:** побудувати зазначений вище графік.

Можна очікувати, що MatLab в цьому випадку побудує, як зазвичай, точки графіків цих функцій та з'єднає їх відрізками ліній. Але якщо ми виконаємо ці команди, то графіка третьої функції не отримаємо. Не виключено навіть збій в роботі програми. Причиною цього, є обрахунок функції  $y_3 = \sin(x)/x$ , якщо  $x$  являє собою масив (вектор), то неможна використовувати оператор матричного ділення « / ».

Цей приклад ще раз наочно демонструє на те, що суто поверхневе застосування навіть такої потужної системи, як MatLab, іноді призводить до збоїв. Щоб все таки отримати графік, потрібно підрахувати відношення  $\sin(x)$  до  $x$  з допомогою оператора поелементного ділення масивів ./ (точка+слеш).

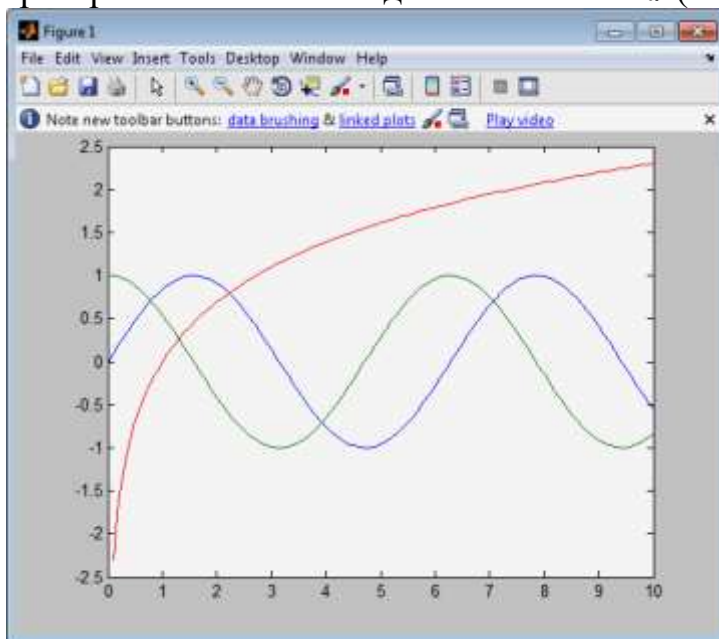


Рисунок 12.8 – Побудова графіків трьох функцій

Зверніть увагу, на те що MatLab побудував графіки усіх трьох функцій, у вікні командного режиму може з'явитися попередження про ділення на 0 – в мить, коли  $x=0$  (при відповідних налаштуваннях MatLab). Це каже про те, що MatLab в деяких випадках «може не знати» про те, що невизначеність  $\sin(x)/x$  виду  $0/0$  усунена і дає 1. Це недолік майже всіх систем для чисельних обрахунків.

**Завдання:** Зберегти даний графік під імям **rus08**.

### Графічна функція *fplot*

Звісно MatLab має засоби для побудови графіків і таких функцій, як  $\sin(x)/x$ , які мають усунені невизначеності. Не обговорюючи ці засоби докладно, просто зобразимо як це робиться з допомогою іншої графічної команди –

**fplot('f(x)', [xmin xmax])**

Вона надає можливість будувати функцію, що задана в символьному вигляді в інтервалі змін аргументу  $x$  від  $xmin$  до  $xmax$  без фіксованого шагу змін  $x$ . Один з варіантів її застосування демонструє рис. 12.9

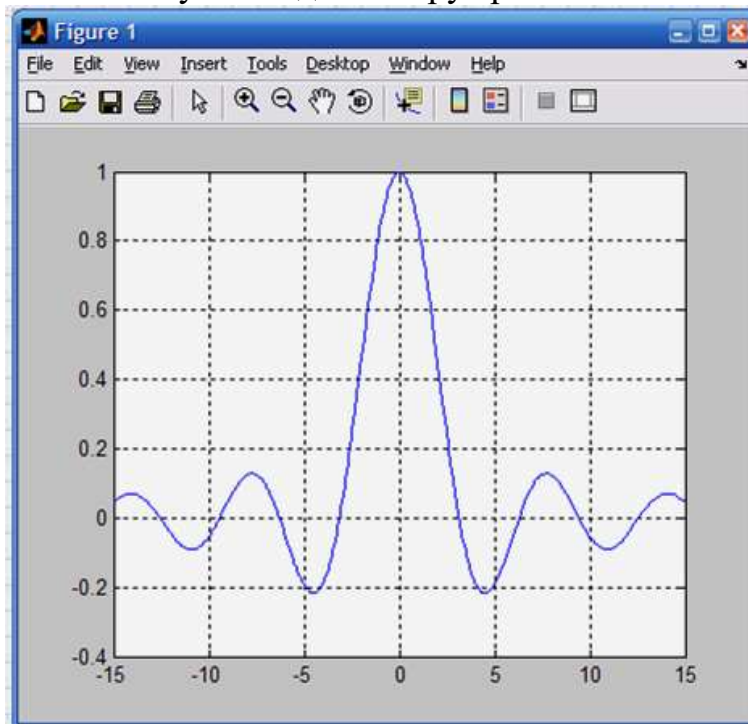


Рисунок 12.9 – Побудова графіка  $\sin(x)/x$  функцією *fplot*

```
>> fplot ('sin(x)/x', [-15 15])  
>> grid on
```

Хоча в процесі підрахунку попередження про помилку (ділення на 0) виводиться, але графік будується правильно. Тобто при  $x=0$   $\sin(x)/x=1$ . Зверніть увагу на дві команди що використовуються: **clear** (очистити) – очистка графічного вікна та **grid on** (сітка) – включення відображення сітки, яка будується пунктиром.

**Завдання:** побудувати зазначений графік. Зберегти його в файл під імям **rus 09**.

### ***Стовпчикові діаграми***

В прикладних обрахунках часто зустрічаються графіки, що називаються стовпчиковими, які відображають зміст деякого вектора  $V$ . При цьому кожен елемент вектора представлений стовпчиком, висота якого пропорційна значенню елементів. Стовпці нумеруються та масштабуються по відношенню до максимального значення найбільш високого стовпця. Виконується побудова такого графіка командою `bar(V)` (рис. 12.10).

```
>> V=1:8  
V =  
1 2 3 4 5 6 7 8  
>> bar(V)
```

**Завдання:** побудувати зазначений графік, зберегти його в файл під ім'ям `rus10`.

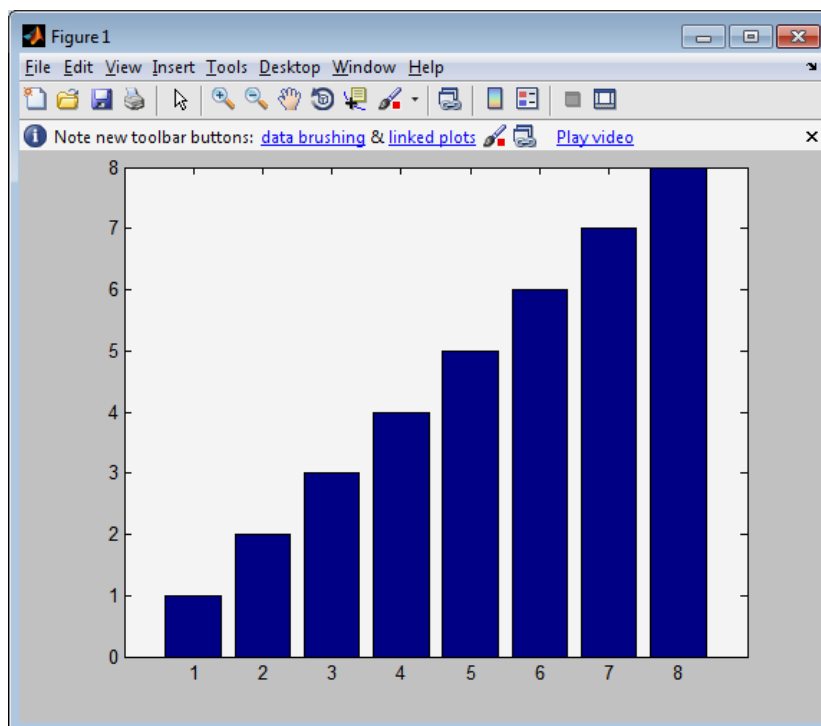


Рисунок 12.10 – Побудова стовпчикової діаграми значень елементів вектора

Стовпчикові діаграми – лише один з багатьох типів графіків, які може будувати система `MatLab`. Особливо часто стовпчикові діаграми використовуються при зображенні даних фінансово-економічних розрахунків.

### ***Побудова тривимірних графіків***

Досить легко можна побудувати і гафіки складних поверхностей. Для цього потрібно знати якою командою реалізується той чи інший графік. Наприклад для побудови графіка поверхні та її проекції у вигляді контурного графіка на площину під поверхнею достатньо використати наступні команди:

```
>> [X,Y]=meshgrid(-5:0.1:5);  
>> Z=X.*sin(X+Y);
```

```
>> meshc(X,Y,Z)
```

**Завдання:** побудувати заданий графік. Зберегти графік під ім'ям rus11.

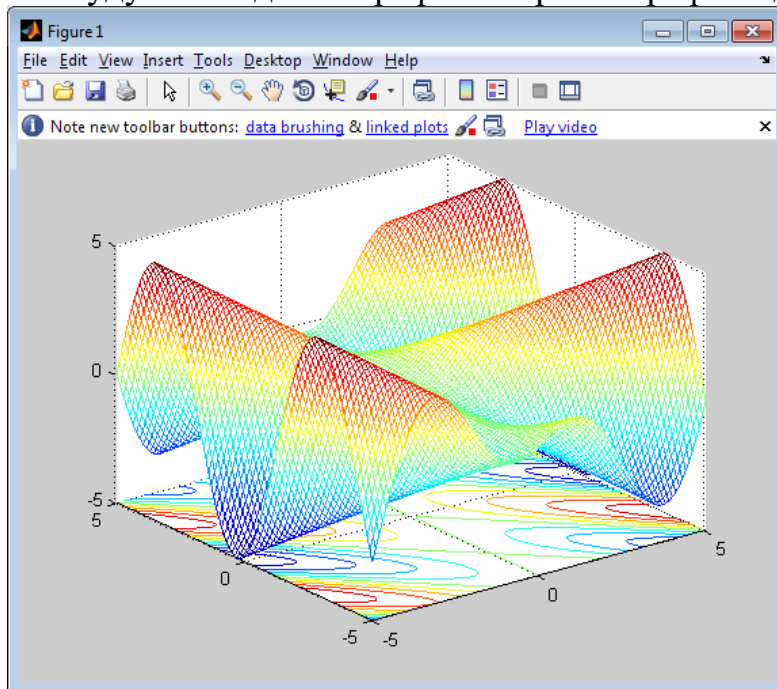


Рисунок 12.11 – Вікно з графіком поверхності та їх проекцією на площину під фігурою

```
>> [X,Y]=meshgrid(-5:0.1:5);
```

```
>> Z=X.*sin(X+Y);
```

```
>> meshc(X,Y,Z)
```

```
>> surf(X,Y,Z)
```

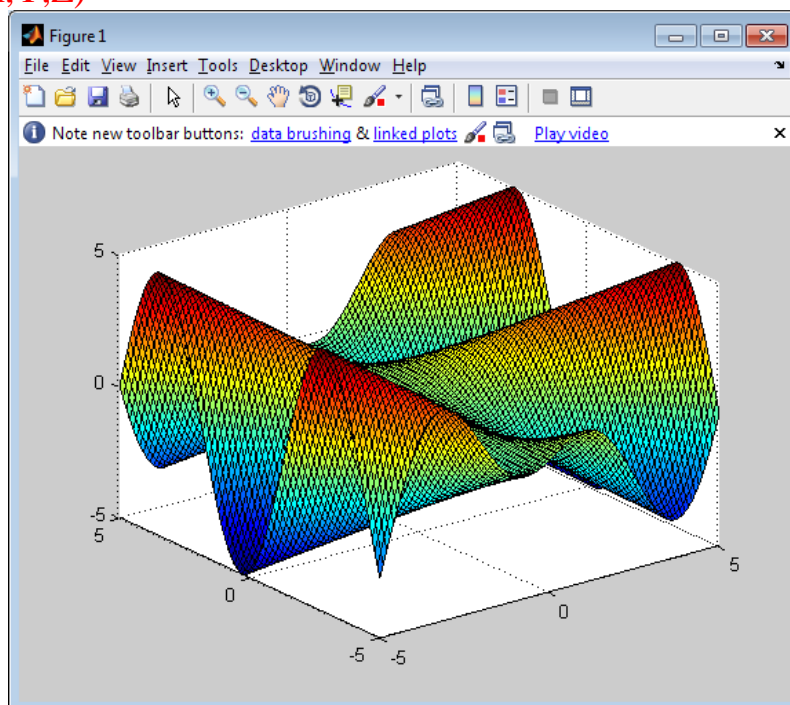


Рисунок 12.12 – Графік поверхності

В MatLab можна досить швидко змінити задану поверхню функції  $Z(X, Y)$  і тут же отримати новий графік функції поверхні, в даному випадку заданий вектором  $Z$  і його проекція на площину  $XY$ .

### **Обертання графіків курсором**

Можна обертати побудовану фігуру мишкою і спостерігати її під різними кутами. Розглянемо цю можливість на прикладі побудови логотипу системи MatLab – мембрани. Для цього потрібно ввести команду **membrane**, отримаємо графік що представлено на рис. 13.

```
>> [X,Y]=meshgrid(-5:0.1:5);  
>> Z=X.*sin(X+Y);  
>> meshc(X,Y,Z)  
>> surf(X,Y,Z)  
>> membrane
```

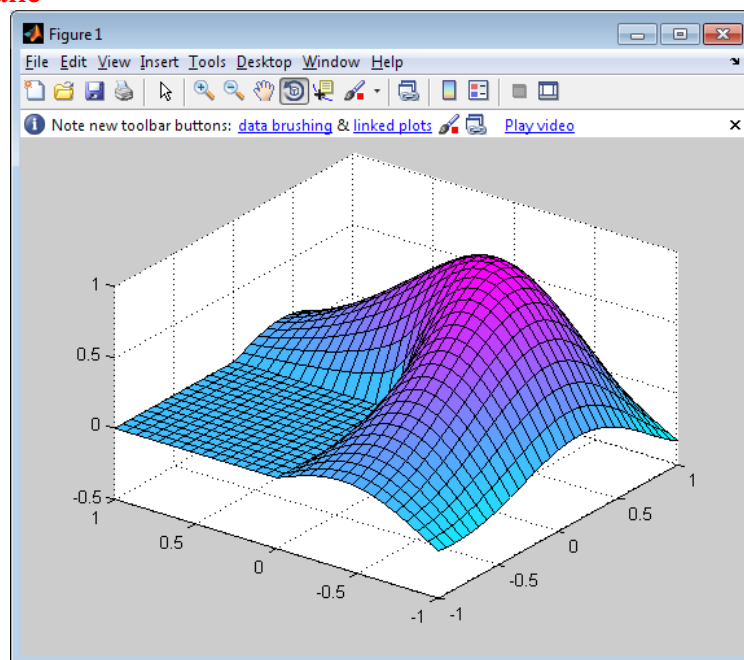


Рисунок 12.13 – Побудова мембрани – логотипу системи MatLab

Для обертання графіка досить активувати кнопку панелі інструментів з зображенням пунктирної стрілки (рис. 12.13). Тепер ввівши курсор миші в область графіка і натиснувши ліву кнопку миші, можна круговими рухами змусити обертатися разом з паралелепіпедом, що його обрамляє.

### **Контекстне меню графіків**

Для перемикання в режим редагування графіків можна виконати команду меню **Tools / Edit Plot** або клацнути на кнопці **Edit Plot** (Редагувати графік) з зображенням курсора-стрілки. В цьому режимі графіком можна керувати з допомогою контекстного меню, що викликається мишкою. Вигляд цього меню при курсорі, що розміщений в графіку в не зони побудови графічного зображення показано на рис. 12.14. З допомогою мишки можна також виділити графік. Натискання на ліву кнопку виведе рамку навколо рисунку (див. рис. 12.14).

**Завдання:** відкрити раніше збережений графік і в меню обрати **Grid**.

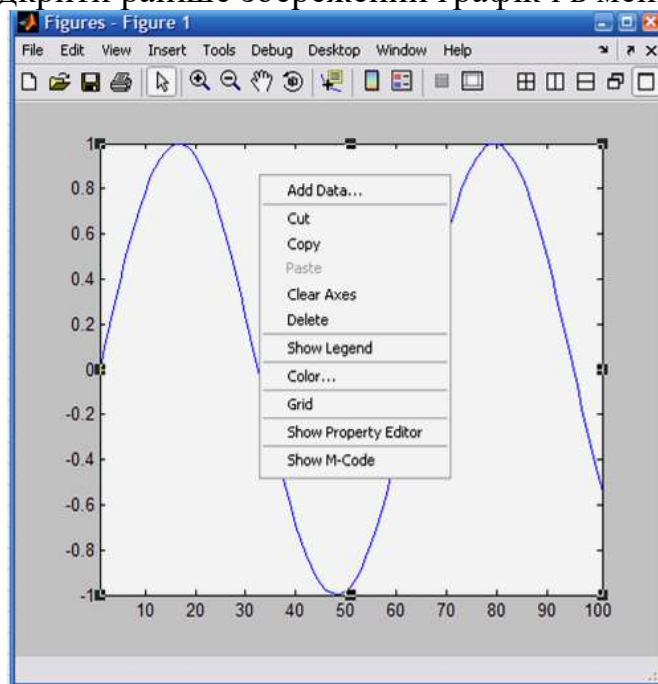


Рисунок 12.14 – Графік в стані редагування

### Форматування властивостей графіків

MatLab має змогу досить легко налаштувати та корегувати властивості графіків з допомогою спеціальних засобів. Одним з них є редактор властивостей (**Property Editor**). Редактор властивостей можна викликати через команду головного меню **View / Property Editor** або через команду контекстного меню **Show Property Editor**.

Вид редактора властивостей залежить від того, який елемент графіка виділено. Наприклад на рис. 12.15 виділена рамка, а на рис. 12.16 виділена лінія графіка.

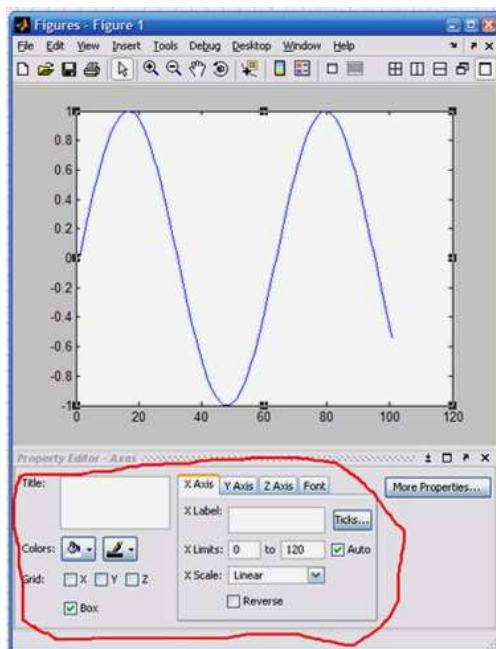


Рисунок 12.15

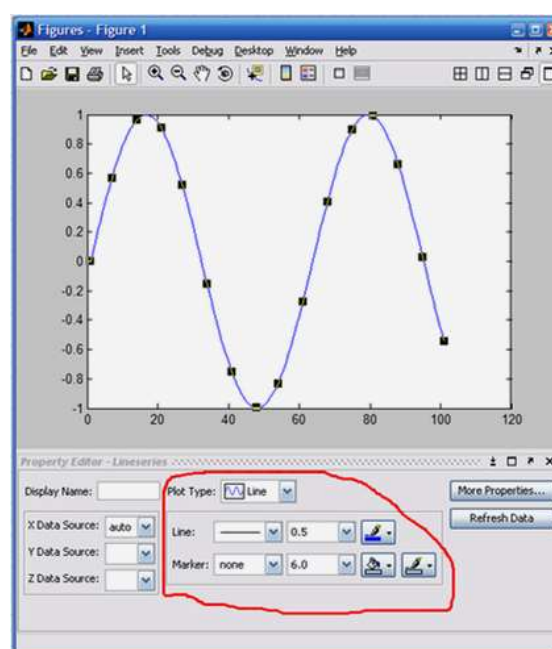


Рисунок 12.16

**Завдання:** знайти і вивести на екран вікно редактора властивостей, що зображене на рис. 12.15 та рис. 12.16. Визначити призначення всіх елементів редактора властивостей виділених червоним.

### *Нанесення графіків та стрілок на графік*

Надписи на графік можна наносити з допомогою панелі інструментів **Plot Edit Toolbar** (рис. 12.17). Місце надпису фіксується клацанням миші.

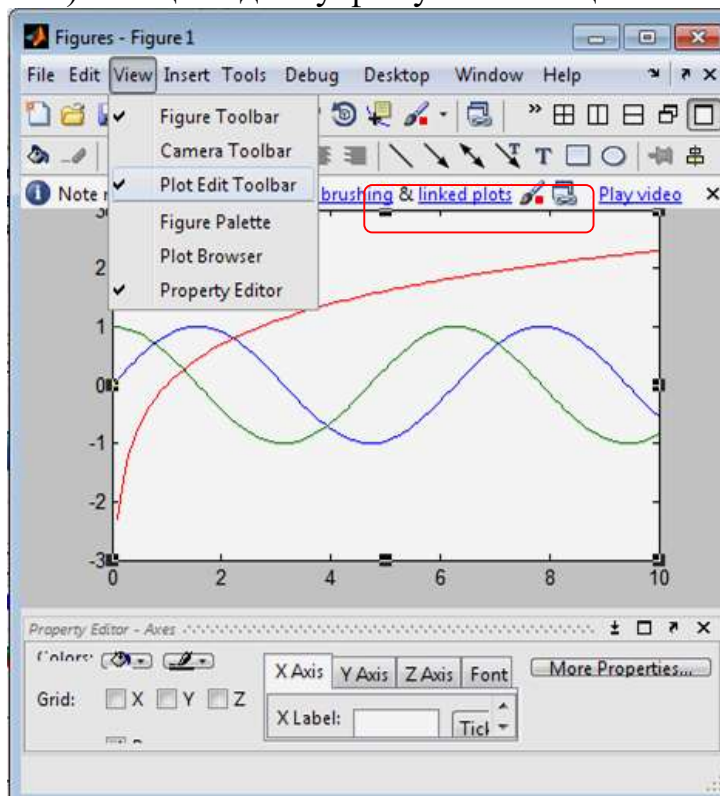


Рисунок 12.17 – Інструменти для нанесення надпису

Як видно з рис. 12.17, параметри надпису можна змінювати як мінімум трьома способами:

- з допомогою редактора властивостей;
- з допомогою панелі інструментів **Plot Edit Toolbar**;
- з допомогою контекстного меню.

Отриманий таким способом надпис можна виділити і перенести мишкою в будь-яке інше місце.

**Завдання:** нанести на графік з рис. 12.8 підписи для кожної з функцій та зберегти результат з ім'ям rus17.

### **III. Індивідуальні завдання**

1. Ознайомитись з теоретичним матеріалом лабораторної роботи.
2. Виконати всі зазначені завдання лабораторної роботи (включаючи виділення червоним та рис 12.1-12.15).
3. Зробити теоретичне пояснення до завдань, які цього потребують.
4. Підготувати відповіді на контрольні питання.
5. Підготувати звіт про виконання завдань та надіслати до СПДН ФІТІС.



## Лабораторна робота № 13

### Тема: «Рішення рівнянь в MatLab»

Питання:

1. Інтерфейс Matlab.
2. Засоби контролю робочої області та файлової системи.
3. Меню MatLab.
4. Програмування в MatLab.
5. Чисельне рішення нелінійних рівнянь.

## I. Теоретичний матеріал

### 1. Основи інтерфейсу MatLab

Користувач може налаштувати вигляд інтерфейсу та по різному розташовувати його вікна. Для редагування та налагодження m-файлів (файлів з програмами) MatLab має вбудований редактор. Також є вікно перегляду ресурсів пам'яті, вікно перегляду шляхів файлової системи, довідник щодо можливостей системи та демонстраційні програми. Редактор графіки і вікно графіки із засобами редагування рисунків були розглянуті раніше.

Панель інструментів (рис. 13.1) надає простий і зручний (особливо для користувачів-початківців) спосіб роботи з системою MatLab. При цьому основні команди вводяться натисканням курсору миші на потрібну кнопку з натисканням лівої клавіші миші. Кнопки мають зображення, яке явно підказує їх призначення.

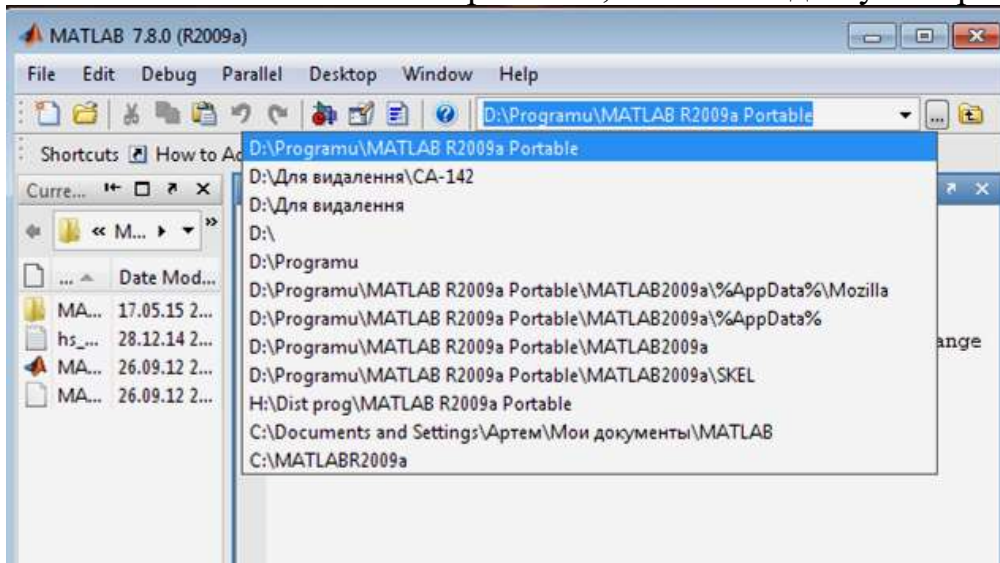


Рисунок 13.1 – Панель інструментів

Призначення кнопок панелі інструментів:

- **New M-file** (Новий m-файл) – виводить порожнє вікно редактора m-файлів;
- **Open file** (Відкрити файл) – відкриває вікно для завантаження m-файлу;
- **Cut** (Вирізати) – вирізає виділений фрагмент і поміщає його в буфер;
- **Copy** (Копіювати) – копіює виділений фрагмент в буфер;
- **Paste** (Вставити) – переносить фрагмент з буфера в поточний рядок введення;

- **Undo** (Скасувати) – скасовує попередню операцію;
- **Redo** (Повторити) – відновлює останню скасовану операцію;
- **Simulink** – відкриває вікно браузера бібліотек Simulink;
- **Guide** – відкриття вікна розробки елементів графічного інтерфейсу користувача;
- **Help** (Допомога) – відкриває вікно довідки.

Набір кнопок панелі інструментів забезпечує виконання команд, що зустрічаються частіш за все і цілком достатній для повсякденної роботи з системою.

Про призначення кнопок говорять і спливаючі підказки, що з'являються, коли курсор миші встановлюється на відповідну кнопку. Вони мають вигляд жовтого прямокутника з текстом короткої довідки.

### ***Виклик вікна відкриття нового файлу***

Кнопка New m-file відкриває вікно редактора / відладчика m-файлів. Це вікно показано на рис. 13.2.

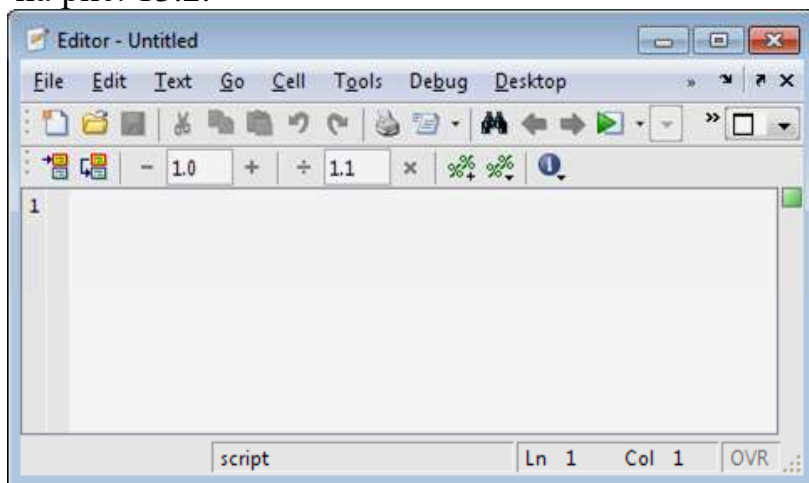


Рисунок 13.2 – Вікно редактора / відладчика m-файлів

За замовчуванням файлу надається ім'я Editor, яке згодом (при записі файлу) можна змінити на інше, відображає тему завдання. Це ім'я відображається в титульному рядку вікна редагування m-файла, яке розміщується у вікні редактора / відладчика і зображене на рис. 2. У редакторі / відладчику можна редагувати кілька m-файлів, і кожен з них буде перебувати у своєму вікні редагування, хоча активним може бути тільки одне вікно, розташоване поверх інших вікон.

**Примітка.** Зверніть увагу, що панель інструментів є контекстно-залежною. Для редактора / відладчика m-файлів вона має дещо інший набір інструментів, ніж для вікна командного режиму роботи.

### ***Виклик вікна завантаження наявного файлу***

Кнопка Open file (Відкрити файл) служить для завантаження в редактор / відладчик раніше створених m-файлів, наприклад які входять в пакет розширення (Toolbox) системи або розроблені користувачем. Вона відкриває

вікно, яке є типовим елементом інтерфейсу Windows-додатків, яке показано на рис. 3 всередині вікна редактора / відладчика m-файлів.

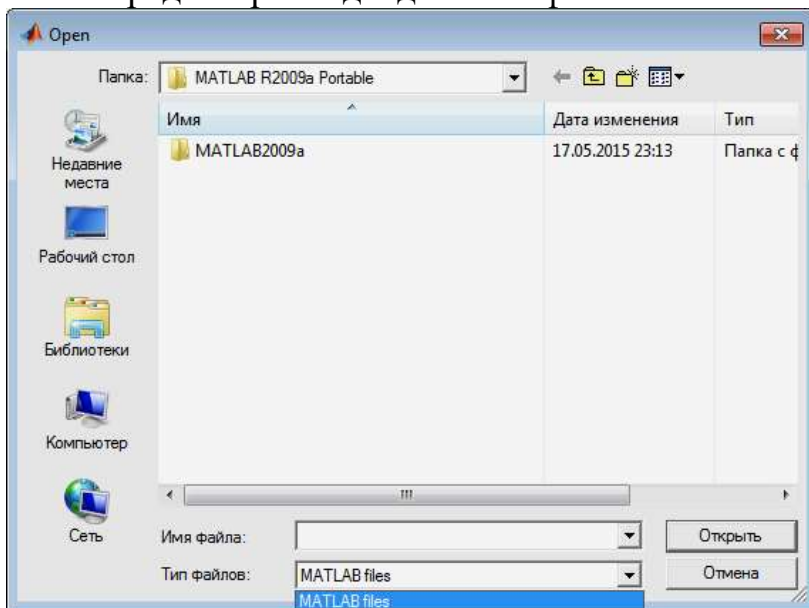


Рисунок 13.3 – Вікно завантаження файлу у вікні редактора / відладчика

У вікні завантаження файлів за допомогою списку Папка і вертикальної смуги прокрутки можна «пройтися» по всіх дисках, папках і файлах. Щоб вибрати потрібний файл для завантаження, його потрібно виділити мишею. Вибір завершується подвійним клацанням по імені файлу, натисканням клавіші Enter або клацанням по кнопці Відкрити. Будь-яка з цих дій призводить до завантаження документу в поточне вікно системи. Кнопка Скасування або клавіша Esc надає можливість відмовитися від завантаження.

### ***Виклик довідки MatLab***

Остання кнопка панелі інструментів Help (Допомога) відкриває вікно з переліком розділів довідкової системи (рис. 13.4).

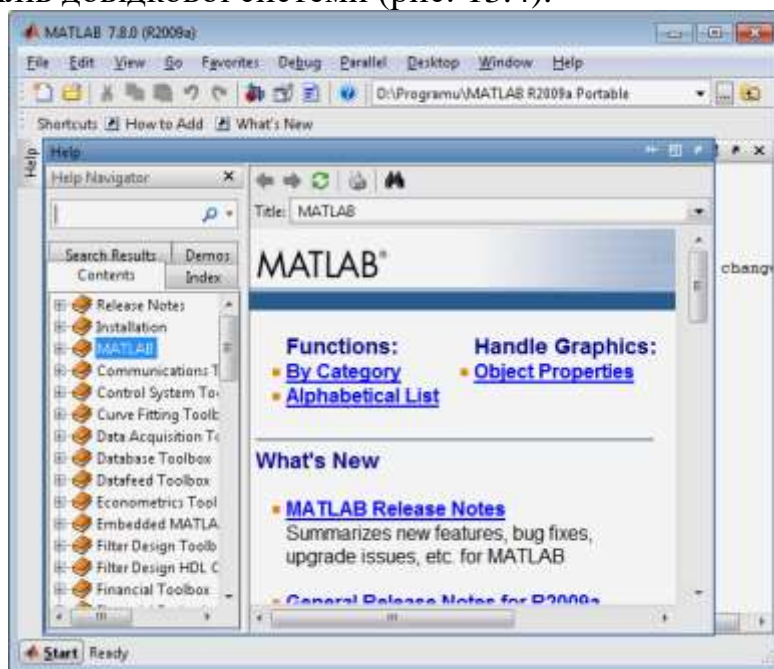


Рисунок 13.4 – Вікно довідкової системи MatLab

## 2.Засоби контролю робочої області та файлової системи

### *Браузер робочої області*

Браузер робочої області призначений для стеження за векторами, матрицями, змінними і т.д., створюваними в процесі сеансу роботи з MatLab.

У лівій частині вікна системи MatLab є вікно спеціального браузера робочої області – Workspace Browser.

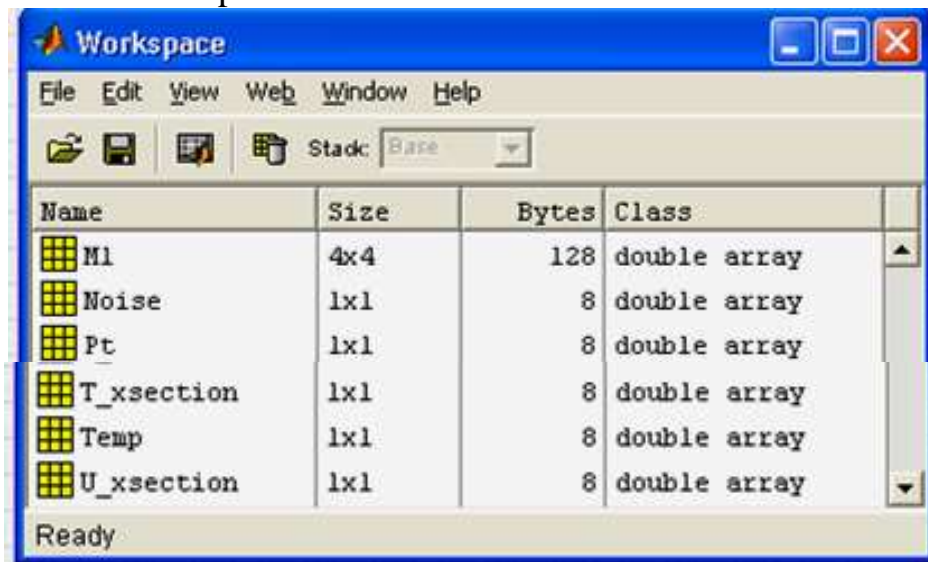


Рисунок 13.5 – Вікно спеціального браузера робочої області – Workspace Browser

Workspace Browser слугує для перегляду ресурсів робочої області пам'яті. Браузер дає наочну візуалізацію вмісту робочої області. Вікно браузера робочої області виконує й інші важливі функції – надає можливість переглядати існуючі в пам'яті об'єкти, редагувати їх вміст і видаляти об'єкти з пам'яті. Для виведення вмісту об'єкта досить виділити його ім'я за допомогою миші і клацнути на кнопці Open (Відкрити) (рис. 13.6).

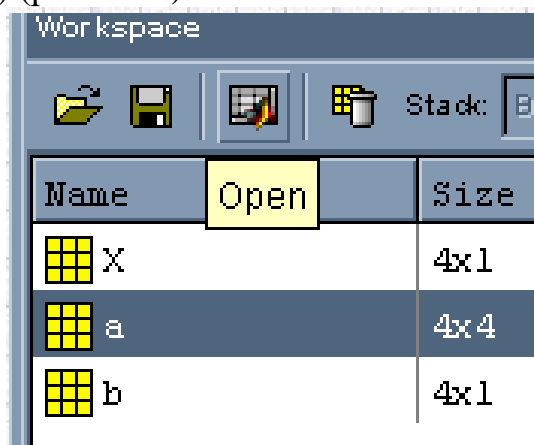


Рисунок 13.6 – Операція відкриття об'єкта

Об'єкт можна відкрити і подвійним клацанням на його імені в списку. Відкриття вікна редагування масиву Array Editor показано на рис. 13.7-13.8.

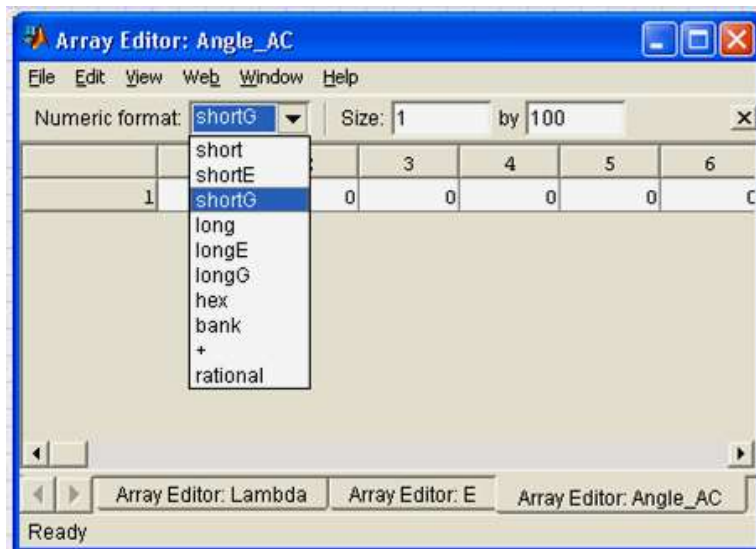


Рисунок 13.7 – Приклад перегляду вмісту матриці

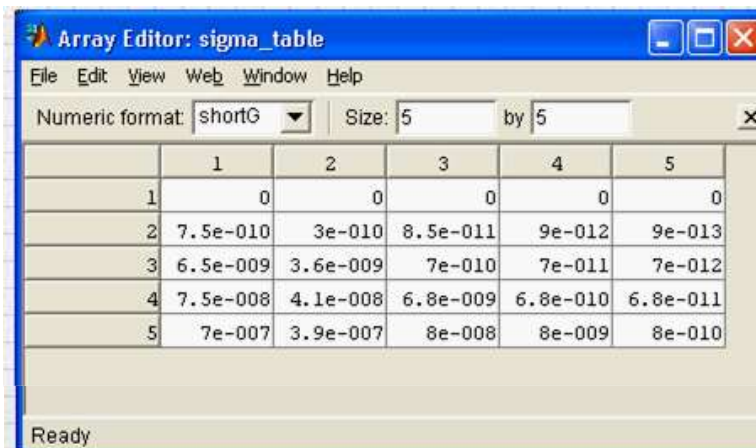


Рисунок 13.8 – Приклад перегляду вмісту матриці

Вікно редагування матриці надає зручний доступ для редагування будь-якого елемента матриці за правилами, прийнятим при роботі з електронними таблицями. Основне з них – швидкий доступ до будь-якого елемента матриці. Можна також змінювати тип значень елементів, вибираючи його зі списку, наданого меню **Numeric format** (Формат чисел). У вікні також виводяться дані про число рядків і стовпців матриці.

### ***Команди перегляду робочої області `who` і `whos`***

Слід зазначити, що перегляд робочої області можливий і в командному режимі, без звернення до браузера **Workspace Browser**. Команда **who** виводить список певних змінних, а команда **whos** – список змінних із зазначенням їх розміру і об'єму займаної пам'яті. Наступний приклад ілюструють дію цих команд:

```
>> X=1.234;
>> V=[1 2 3 4];
>> M=magic(4);
>> who
```

Your variables are:

M V X x

>> whos

Name	Size	Bytes	Class	Attributes
M	4x4	128	double	
V	1x4	32	double	
X	1x1	8	double	
x	1x1	8	double	

Якщо потрібно переглянути дані однієї змінної, наприклад M, слід використовувати команду **whos M**. Природно, перегляд робочої області за допомогою браузера робочої області (Workspace Browser) більш зручний і наочний.

>> whos M

Name	Size	Bytes	Class	Attributes
M	4x4	128	double	

**Завдання.** Створити в командному рядку матрицю розміром 3 \* 3 і вектор з 4-х елементів. Далі: вивчити можливості браузера робочої області на прикладі даних змінних; виконати перегляд робочої області за допомогою команд who і whos.

### *Браузер файлової структури*

Для перегляду файлової структури в MatLab використовується спеціальний браузер файлової системи Current Directory (Поточна папка), який запускається при завантаженні системи. На рис. 13.9 в лівій частині показано вікно цього браузера.

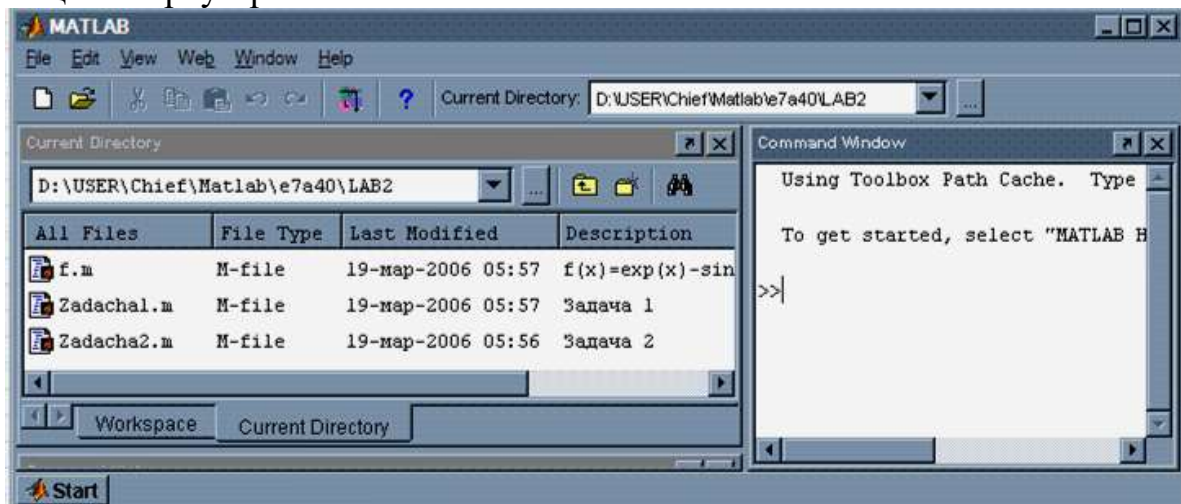


Рисунок 13.9 – Браузер файлової структури

Неважко помітити, що браузер файлової системи побудований за типом загальновідомого Провідника операційних систем Windows Me / 2000 / XP. Вікно дає можливість перегляду файлової системи ПК і вибору будь-якого файлу. Для прикладу на рис. 10 показаний вибір m-файла Zadacha1.m. Виконавши команду Open (Відкрити) з контекстного меню правої клавіші миші (рис. 13.10) або двічі

клацнувши по виділеному рядку з ім'ям файлу, можна завантажити цей файл у вікно редактора / відладчика m-файлів. При цьому редактор запуститься автоматично і його вікно з готовим для редагування обраним файлом з'явиться на екрані.

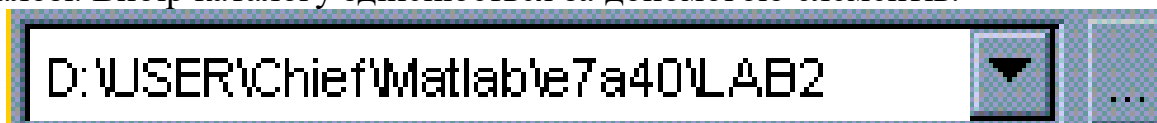


Рисунок 13.10 – Відкриття файлу

**Примітка.** Зверніть увагу на те, що в полі Description відображається перший зустрічний в m-файлі коментар, в якому користувач описує файл.

Таким чином, браузер перегляду файлової структури надає можливість детально ознайомитися з файловою системою MatLab і вивести будь-який з m-файлів або файлів демонстраційних прикладів для перегляду, редагування та модифікації.

Важливо, що в MatLab перед початком вирішення завдання користувач зобов'язаний вибрати робочий каталог для вирішення даного завдання. Всі файли, необхідні для вирішення завдання, повинні зберігатися в певному каталозі. Вибір каталогу здійснюється за допомогою елементів.



**Завдання.** Створити каталог для лабораторної роботи і вибрати його в якості робочого каталогу в MatLab.

**Завдання.** Створити m-файл для вирішення першого завдання і зберегти його в ваш робочий каталог з ім'ям, наприклад *Zadacha1.m*.

### 3. Меню системи MatLab

Перейдемо до опису основного меню системи MatLab. Це меню (див. рис. 1) виглядає досить скромно і містить всього шість пунктів:

**File** – робота з файлами;

**Edit** – редагування сесії;

**View** – висновок і приховування панелі інструментів;

**Web** – доступ до інтернет-ресурсів;

**Windows** – установка Windows-властивостей вікна;

**Help** – доступ до довідкових підсистем.

#### *Меню File*

Меню File містить ряд операцій і команд для роботи з файлами. Воно показано на рис. 13.11:

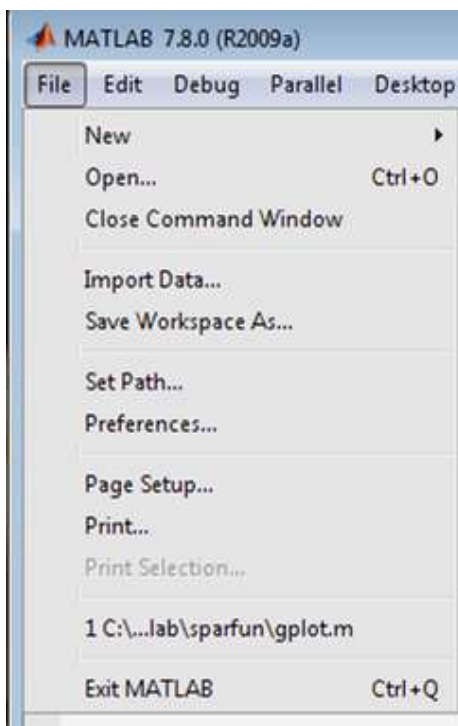


Рисунок 13.11 – Меню файлових операцій File

**New** – відкриває підменю з позиціями:

**M-file** – відкриття вікна редактора / відладчика m-файлів;

**Figure** – відкриття порожнього вікна графіки;

**Model** – відкриття порожнього вікна для створення Simulink-моделі;

**Open** – відкриває вікно завантаження файлу;

**Close Command Windows** – закриває вікно командного режиму роботи (воно при цьому зникає з екрану);

**Import data** – відкриває вікно імпорту файлів даних;

**Save Workspace As ...** – відкриває вікно запису робочої області у вигляді файлу з заданим ім'ям;



**Set Path** – відкриває вікно установки шляхів доступу файлової системи;  
**Preferences ...** – відкриває вікно налаштування елементів інтерфейсу;  
**Print ...** – відкриває вікно друку всього поточного документу;  
**Print Selection ...** – відкриває вікно друку виділеної частини документу;  
**Exit** – завершує роботу з системою;

### **Меню Edit:**

Меню Edit (рис.13.12) містить операції і команди редагування, типові для більшості додатків Windows:

**Undo** (Скасувати) – скасування результату попередньої операції;

**Redo** (Повторити) – скасування дії останньої операції Undo;

**Cut** (Вирізати) – вирізання виділеного фрагменту і перенесення його в буфер;

**Copy** (Копіювати) – копіювання виділеного фрагменту в буфер;

**Paste** (Вставити) – вставка фрагменту з буфера в поточну позицію курсору;

**Select All** (Виділити) – виділення всієї сесії;

**Delete** (Стерти) – знищення виділеного об'єкта;

**Clear Command Windows** (Очистити командне вікно) – очищення тексту сесії (зі збереженням створених об'єктів);

**Clear Command History** (Очистити вікно історії команд) – очищення вікна історії;

**Clear Workspace** – очищення вікна браузера робочої області.

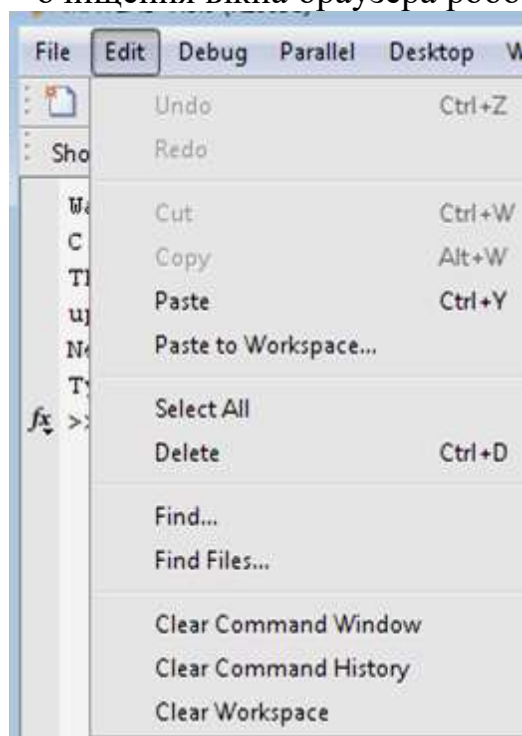


Рисунок 13.12 – Меню Edit

Відзначимо, що команда Clear Command Window очищає вікно командного режиму роботи і поміщає курсор у верхній лівий кут вікна. Однак всі визначення, зроблені протягом стертих таким чином сесій, зберігаються в пам'яті комп'ютера. Для

очищення екрану використовується також команда `clc`, що вводиться в командному режимі.

**Завдання.** Очистити робочу область. Очистити командне вікно. Очистити історію команд.

#### 4. Програмування в MATLAB

До цих пір ми в основному використовували систему MatLab в режимі обрахунку – в командному режимі. Однак при вирішенні серйозних завдань виникає необхідність збереження використовуваних послідовностей обчислень, а також їх подальшої модифікації. Іншими словами, існує необхідність програмування вирішення завдань.

Це може здатися відступом від важливої мети, яка переслідується розробниками більшості математичних систем, – виконання математичних обчислень без використання традиційного програмування. Однак це не так. Безліч математичних завдань вирішуються в системі MatLab без програмування. З використанням мов високого рівня для їх вирішення потрібно було б написати і відтестити сотні програм.

Практично неможливо передбачити в одній, навіть найбільшій і потужній математичній системі можливість вирішення всіх завдань, які можуть цікавити користувача. Програмування в системі MatLab є ефективним засобом її розширення та адаптації до вирішення специфічних проблем. Воно реалізується за допомогою мови програмування системи.

Більшість об'єктів цієї мови, зокрема всі команди, оператори та функції, одночасно є об'єктами вхідної мови спілкування з системою в командному режимі роботи. Так що фактично ми приступили до вивчення мови програмування системи MatLab, починаючи з перших лабораторних робіт.

Так в чому ж відмінність вхідної мови від мови програмування? В основному – в способі фіксації створюваних нею кодів. Сесії в командному режимі роботи не зберігаються в пам'яті комп'ютера. Зберігаються тільки визначення створених в ході їх виконання змінні та функції. А ось програми на мові програмування MatLab зберігаються у вигляді текстових *m*-файлів. При цьому можуть зберігатися як цілі програми у вигляді файлів-сценаріїв, так і окремі програмні модулі – функції. Крім того, важливо, що програма може змінювати структуру алгоритмів обчислень в залежності від вхідних даних і даних, що створюються в ході обчислень.

З позицій програміста мова програмування системи є типовою проблемно-орієнтованою мовою програмування високого рівня. Точніше кажучи, це навіть мова надвисокого рівня, що містить складні оператори та функції, реалізація яких на звичайних мовах (наприклад Бейсик, Паскаль або Сі) зажадала б багато зусиль і часу. До таких функцій відносяться матричні функції, функції швидкого перетворення Фур'є та ін. А до операторів – оператори побудови різноманітних графіків, генерації матриць певного виду і т.д.

## Основи редагування та налагодження m-файлів

### Інтерфейс редактора / відладчика m-файлів

Для підготовки, редагування та налагодження m-файлів служить спеціальний багатовіконний редактор (рис. 13.13). Він виконаний як типовий додаток Windows. Редактор можна викликати командою `edit` з командного рядка або командою `New> Blank M-file` з меню `File`. Після цього у вікні редактора можна створювати свій файл, користуватися засобами його налагодження і запуску. Перед запуском файлу його необхідно записати на диск, використовуючи команду `File> Save as` в меню редактора.

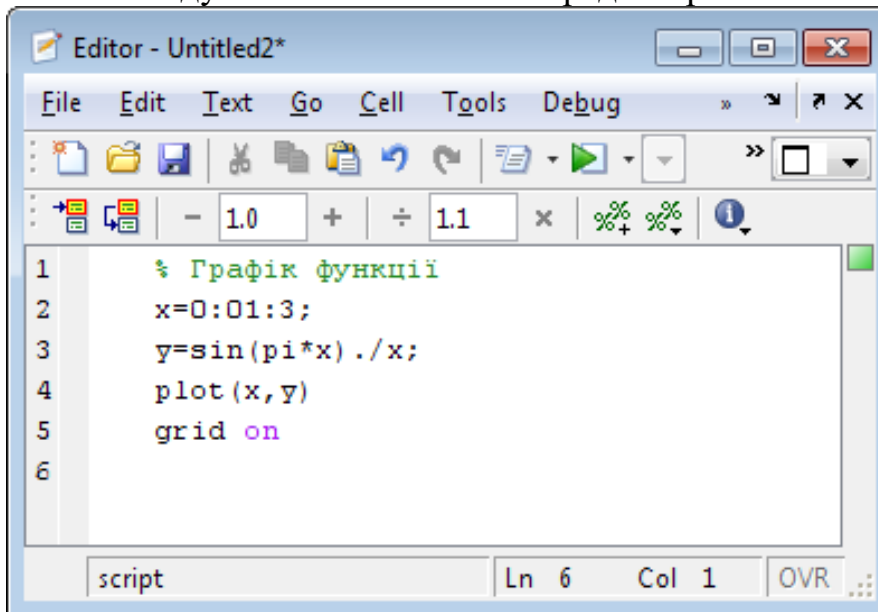


Рисунок 13.13 – Редактор / відладчик файлів при записі файлу на диск

На рис. 13.13 показано вікно редактора / відладчика з текстом простого файлу у вікні редагування та налагодження.

Після запису файлу на диск, можна помітити, що команда `Run` (панелі Інструментів) редактора стає активною (до запису файлу на диск вона пасивна) і дозволяє провести запуск файлу. Запустивши команду `Run`, можна спостерігати виконання m-файлу; в нашому випадку – це побудова рисунка в графічному вікні і висновок написи про поділ на нуль в ході обчислення функції  $\sin(x) / x$  в командному вікні системи.

**Завдання.** Виконати приклад, зображений на рис. 13.13.

На перший погляд може здатися, що редактор / відладчик – просто зайва ланка в ланцюжку «користувач – MatLab». І справді, текст файлу можна було б ввести у вікно системи і отримати той же результат. Однак на ділі редактор / відладчик виконує важливу роль. Він надає можливість створити m-файл (програму) без тих численних кодів, які супроводжують роботу в командному режимі. Далі ми переконаємося, що текст такого файлу піддається ретельній синтаксичній перевірці, в ході якої виявляються і відсіваються більшість помилок користувача. Таким чином, редактор забезпечує синтаксичний контроль файлу.

Редактор має й інші важливі налагоджувальні засоби – він надає можливість встановлювати в тексті файлу спеціальні мітки, іменовані точками переривання (breakpoints). При їх досягненні обчислення припиняються, і користувач може оцінити проміжні результати обчислень (наприклад, значення змінних), перевірити правильність виконання циклів і т. д. Нарешті, редактор надає можливість записати файл у текстовому форматі файлової системи MatLab.

Для зручності роботи з редактором / відладчиком рядки програми в ньому нумеруються в послідовному порядку. Редактор є багато віконним. Вікно кожної програми оформляється як вкладка.

### ***Колірні виділення і синтаксичний контроль***

Редактор / відладчик m-файлів виконує синтаксичний контроль програмного коду у міру введення тексту. При цьому використовуються наступні колірні виділення:

- ключові слова мови програмування – синій колір;
- оператори, константи і змінні – чорний колір;
- коментарі після знака% - зелений колір;
- символічні змінні (в апострофах) – зелений колір;
- синтаксичні помилки – червоний колір.

Завдяки колірним виділенням ймовірність синтаксичних помилок знижується.

Однак далеко не всі помилки діагностуються. Помилки, пов'язані з неправильним застосуванням операторів або функцій (наприклад, застосування оператора – замість + або функції cos (x) замість sin (x) і т. д.), не спроможна виявити жодна система програмування. Усунення такого роду помилок (їх називають семантичними) – справа користувача, що налаштовує свої алгоритми і програми.

### ***Панель інструментів редактора і відладчика***

Редактор має своє меню і свою інструментальну панель. Зовнішній вигляд інструментальної панелі показаний на рис. 13.14. За стилем дана панель схожа на панель Інструментів вікна командного режиму роботи, але має дещо інший набір кнопок.



Рисунок 13.14. Панель інструментів редактора / відладчика m-файлів

Призначення кнопок панелі інструментів редактора / відладчика наступне:

**New** – створення нового m-файлу;

**Open** – висновок вікна завантаження файлу;

**Save** – запис файлу на диск;

**Cut** – вирізання виділеного фрагменту і перенесення його до буферу;

**Copy** – копіювання виділеного об'єкту в буфер;

**Paste** – розміщення фрагменту з буфера в позиції текстового курсору;

**Undo** – скасування попередньої операції;

**Redo** – повтор скасованої операції;  
**Print** – друк вмісту поточного вікна редактора;  
**Save and publish** – збереження та друк;  
**Find text** – знаходження зазначеного тексту;  
**Back** – повертає курсор на крок назад;  
**Forward** – повертає курсор на крок вперед;  
**Show function** – показ функції;  
**Save and Run** – збереження та запуск;  
**Set / Clear Breakpoint** – встановлення / скидання точки переривання;  
**Clear All Breakpoints** – скидання всіх точок переривання;  
**Step** – виконання кроку трасування;  
**Step In** – покрокове трасування із заходом що викликається m-файлом;  
**Step Out** – покрокова трасування без заходу, що викликається m-файлом;  
**Continue** – продовження виконання;  
**Exit Debug Mode** – вихід з режиму відладки.

### *Алфавіт мови програмування*

В MatLab, як і в інших системах комп'ютерної математики, використовуються всі букви латинського алфавіту від А до Z і арабські цифри від 0 до 9. Як і в C ++, MatLab чутлива до регістру. Крім букв латинського алфавіту використовуються всі спеціальні символи клавіатури комп'ютера.

### *Арифметичні і логічні оператори*

Таблиця 13.1 – Список арифметичних операторів

Функція	Позначення (синтаксис)
Додавання	+ (M1+M2)
Віднімання	- (M1-M2)
Множення чисел або матриць	* (M1*M2)
Поелементне множення матриць	.* (M1.*M2)
Піднесення чисел матриці або матриці в степінь	^ (M1 ^ x)
Поелементне піднесення матриці до степеня	.^ (M1.^ x)
Ділення чисел або ділення матриці зліва на право	/ (M1 / M2)
Поелементне ділення матриці зліва на право	./ (M1 ./ M2)
Ділення матриці з права на ліво	\ (M1 \ M2)
Поелементне ділення матриці з права на ліво	.\ (M1 .\ M2)

У математичних виразах оператори мають певний пріоритет виконання. В MatLab пріоритет логічних операторів вище, ніж арифметичних, пріоритет зведення в ступінь вище пріоритетів множення і ділення, пріоритет множення і ділення вище додавання і віднімання. Для підвищення пріоритету операцій

потрібно використовувати круглі дужки. Ступінь вкладення дужок не обмежується.

Оператори відношень служать для порівняння двох величин, векторів або матриць, всі оператори відношень мають дві порівнювані величини і записуються, як показано в таблиці 13.2.

Таблиця 13.2 – Оператори відношень

Функція	Оператор (синтаксис)
Дорівнює	$== (x == y)$
Не дорівнює	$\sim = (x \sim = y)$
Меньше	$< (x < y)$
Більше	$> (x > y)$
Меньше або дорівнює	$< = (x < = y)$
Більше або дорівнює	$> = (x > = y)$

Дані оператори виконують поелементне порівняння векторів або матриць однакового розміру і логічний вираз приймає значення 1 (True), якщо елементи ідентичні, в іншому випадку значення 0 (False).

Логічні оператори служать для реалізації поелементних логічних операцій над елементами однакових за розміром масивів згідно таблиці 13.3.

Таблиця 13.3 – Логічні оператори

Функція	Оператор (синтаксис)
Логічне «І»	$\&; \text{and} (\text{and} (a, b))$
Логічне «АБО»	$ ; \text{or} (\text{or} (a, b))$
Логічне «НІ»	$\sim ; \text{not} (\text{not} (a, b))$
Виключаюче АБО	$\text{xor} (\text{xor} (a, b))$
Вірно, якщо всі елементи вектора дорівнюють нулю	$\text{any} (\text{any} (a))$
Вірно, якщо всі елементи вектора не дорівнюють нулю	$\text{all} (\text{all} (a))$

### Елементарні функції

В тригонометричних функціях кути вимірюються в радіанах.

Таблиця 13.4 – Запис тригонометричних функцій

Функція	Синтаксис
$ x $ – модуль	$\text{abs}(x)$
$e^x$ – експонента	$\text{exp}(x)$
$\ln x$ – натуральний логарифм	$\text{log}(x)$
логарифм з основою 2	$\text{log2}(x)$
$\lg x$ – десятковий логарифм	$\text{log10}(x)$
$2^x - 2$ в степені x	$\text{pow}(x)$
квадратний корінь	$\text{sqrt}(x)$
$\sqrt[3]{x}$ – корінь третього степіня	$\text{nthroot}(x,3)$

$\arccos x$ – арккосинус	$\text{acos}(x)$
$\text{arcctg } x$ – арккотангенс	$\text{acot}(x)$
$\text{arccosec } x$ – арккосеканс	$\text{acsc}(x)$
$\text{arcces } x$ – арксеканс	$\text{asec}(x)$
$\arcsin x$ – арксинус	$\text{asin}(x)$
$\text{arctg } x$ – арктангенс	$\text{atan}(x)$
$\cos x$ – косинус	$\text{cos}(x)$
$\text{ctg } x$ – котангенс	$\text{cot}(x)$
$\sec x$ – секанс	$\text{sec}(x)$
$\text{cosec } x$ – косеканс	$\text{csc}(x)$
$\sin x$ – синус	$\text{sin}(x)$
$\text{tg } x$ – тангенс	$\text{tan}(x)$
$\text{arcch } x$ – арккосинус гіперболічний	$\text{acosh}(x)$
$\text{arccth } x$ – арккотангенс гіперболічний	$\text{acoth}(x)$
$\text{arccosech } x$ – арккосеканс гіперболічний	$\text{acsch}(x)$
$\text{arcsech } x$ – арксеканс гіперболічний	$\text{asech}(x)$
$\text{arcsh } x$ – арккосинус гіперболічний	$\text{asinh}(x)$
$\text{arctgh } x$ – арктангенс гіперболічний	$\text{atanh}(x)$
$\text{ch } x$ – косинус гіперболічний	$\text{cosh}(x)$
$\text{ctgh } x$ – котангенс гіперболічний	$\text{coth}(x)$
$\text{cosech } x$ – косеканс гіперболічний	$\text{csch}(x)$
$\text{sech } x$ – секанс гіперболічний	$\text{sech}(x)$
$\text{sh } x$ – синус гіперболічний	$\text{sinh}(x)$
$\text{tgh } x$ – тангенс гіперболічний	$\text{tanh}(x)$

Слід пам'ятати, що всі елементарні функції повинні записуватися малими літерами. Існують також спеціальні математичні функції, на яких ми не будемо зупинятися.

### ***Поняття про файли-сценарій і файл-функцій***

Тут корисно зазначити, що m-файли, створювані редактором / відладчиком, діляться на два класи:

- файли-сценарії, що не мають вхідних параметрів;
- файли-функції, що мають вхідні параметри.

### ***Файл-сценарій***

Файл сценарій називається також Script-файлом або просто скриптом.

Файл-сценарій має вельми просту структуру:

- % основний коментар, якщо необхідно;
- % додатковий коментар, якщо необхідно;
- тіло програми з будь-якими допустимими виразами.

Важливими є такі властивості файлів-сценаріїв:

- вони не мають вхідних і вихідних аргументів;
- працюють з даними з робочої області;
- у процесі виконання не компілюються;
- являють собою послідовність операцій, аналогічну тій, що використовується в сеансі роботи з командним рядком.

Наведений у вікні редакторі файл (рис. 13.13) є файлом-сценарієм, або Script-файлом.

### **Файл-функція**

Відмінність m-файлу функції від сценарію полягає в тому, що він є аналогом підпрограми типу `function` в мові Pascal.

Файл-функція має вхідні параметри, список яких вказується в круглих дужках. Використовувані в файлі-функції змінні є локальними змінними, зміна значень яких в тілі функції жодним чином не впливає на значення, які ті ж самі змінні можуть мати за межами функції.

Іншими словами, локальні змінні можуть мати ті ж імена (ідентифікатори), що і глобальні змінні (хоча правила не рекомендують змішувати імена локальних і глобальних змінних). Надалі ми розглянемо це питання більш детально, а поки повернемося до опису інтерфейсу компонентів системи MatLab.

Структура m-файлу функції з одним вихідним параметром має вигляд:

- `function var = f_name` (список параметрів);
- `%` основний коментар, якщо необхідно;
- `%` додатковий коментар, якщо необхідно;
- тіло програми з будь-якими виразами;
- `var = вираз`.

M-файл функції має наступні властивості:

Він починається з ключового слова `function`, після якого вказується ім'я змінної `var` – вихідного параметра, ім'я самої функції **f\_name** і список її вхідних параметрів, відокремлених комою.

**Увага:** Ім'я m-файлу функції має збігатися з ім'ям самої функції (`f_name`). MatLab автоматично присвоює дане ім'я при виконанні команди `Save as`.

Результатом виконання m-файлу функції присвоюється ім'я функції, яке може використовуватися в математичних виразах подібно до функцій `sin(x)`, `log(x)` і т. п.

Всі змінні, використовувані у файлі-функції, є локальними, тобто діють тільки в межах тіла функції.

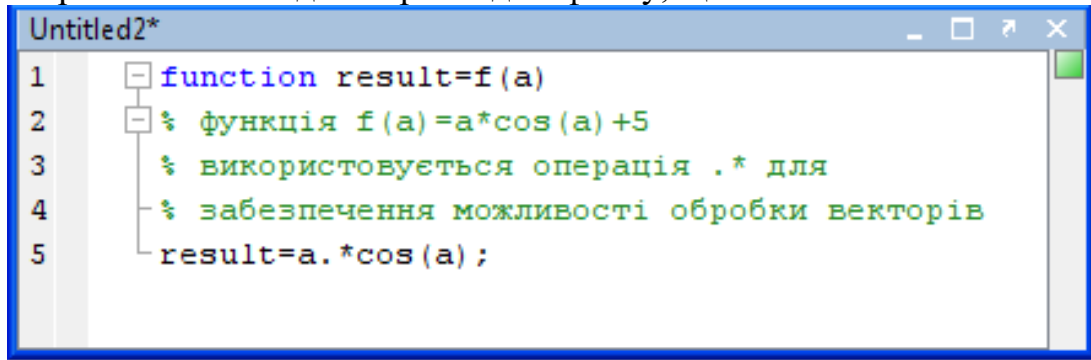
Остання конструкція `var = вираз` вводиться, якщо потрібно, щоб функція повертала результат обчислень. Замість імені `var` можна використати будь-яке інше ім'я.

Файл-функція є самостійним програмним модулем, який пов'язаний з іншими модулями і головною програмою через вхідні і вихідні параметри.

При виклику файлу-функції він компілюється і потім виконується. M-файл функції повинен зберігатися в робочому каталозі.



На рис. 13.15 наведено приклад m-файлу, що містить текст.



```
1 function result=f(a)
2 % функція f(a)=a*cos(a)+5
3 % використовується операція .* для
4 % забезпечення можливості обробки векторів
5 result=a.*cos(a);
```

Рисунок 13.15 – Приклад m-файлу, що містить текст

**Завдання.** Виконати даний приклад, придумавши свою власну функцію.

### 3. Основи програмування

#### *Оператор присвоювання*

Програмування в системі MatLab є засобом її розширення і використання при вирішенні специфічних проблем. Деякі питання програмування викладені вище, тут розглянемо правила, що доповнюють синтаксис мови MatLab.

Програми оперують зі змінними і константами. Змінні – імена об’єктів, здатні зберігати різні за значенням дані. Залежно від цих даних змінні можуть бути числовими або символічними, векторними або матричними.

Для задання змінним певних значень використовується оператор присвоювання, що вводиться знаком рівності =

**Ім’я \_ змінної = Вираз;**

Типи змінних заздалегідь не оголошуються. Вони визначаються виразом, значення якого присвоюється змінній.

Ім’я змінної може містити скільки завгодно символів, але ідентифікується тільки 31 початковий символ. Ім’я будь-якої змінної повинно бути унікальним. Ім’я повинно починатися з літери, може містити букви, цифри та символ підкреслення `_`. Неприпустимо включати в імена прогалини і спеціальні знаки.

#### *Перенесення рядка*

Якщо математичний вираз виходить за розмір екрану монітора, то доцільно перенести його частину на наступний рядок. Для цього використовується символ крапки ... – три і більше крапок. У командному режимі число можливих символів в одному рядку – 4096, в m-файлі – не обмежене, але з такими довгими рядками працювати незручно. Тому застосування в файлах-сценаріях символу переносу рядка покращує наочність програм.

#### *Введення і виведення даних*

У мові MatLab немає явних операторів введення / виведення даних в режимі діалогу. Ця проблема вирішується для введення даних оператором присвоювання і використанням системних констант. Обчислення даних здійснюється ще простіше. Для цього необхідно після математичного виразу не ставити символ крапку з комою (;).

До системних констант відносяться:

- $\pi$  = 3,1415 ... число "Пі";
- $i$  чи  $j$  – уявні одиниці;
- NaN – невизначеність у вигляді 0/0;
- Inf – нескінченність;
- ans – результат останньої операції.

### **Оператор двокрапка**

Вельми часто необхідно виконати формування впорядкованих числових послідовностей. Такі послідовності потрібні для створення векторів або значень аргументів  $x$  при побудові графіків. В MatLab для цього використовується оператор двокрапка (:), який представляється наступним чином:

**$x =$  Початкове \_ значення: Крок: Кінцеве \_ значення;**

Ця конструкція створює зростаючу послідовність чисел, яка починається з початкового значення, змінюється на заданий крок і завершується кінцевим значенням. Якщо крок не заданий, то він приймає значення 1. Якщо кінцеве значення вказано меншим, ніж початкове значення, то видається повідомлення про помилку.

Наприклад, оператор  
 $s = 2: 0.25: 3;$

формує вектор

$s = [2 \ 2.25 \ 2.5 \ 2.75 \ 3]$

### **Повідомлення про помилки і виправлення помилок**

Система MatLab контролює правильність написання програм і, за наявності помилок, видає відповідне повідомлення у вікні команд. При цьому вказується номер рядка, де допущена помилка, і характер помилки. Після з'ясування суті помилки її необхідно виправити в тексті програми, зберегти  $m$ -файл командою Save і знову виконати програму. Перед цим бажано очистити вікно команд від повідомлення про помилки (щоб не захарашувати отриману картинку) за допомогою команди Clear Command Windows (Очистити вікно команд) в меню Edit.

## **5. Чисельне рішення нелінійних рівнянь**

Задача знаходження коренів нелінійних рівнянь зустрічається в різних областях науково-технічних досліджень. Проблема формулюється таким чином. Нехай задана безперервна функція  $f(x)$  і потрібно знайти корінь рівняння

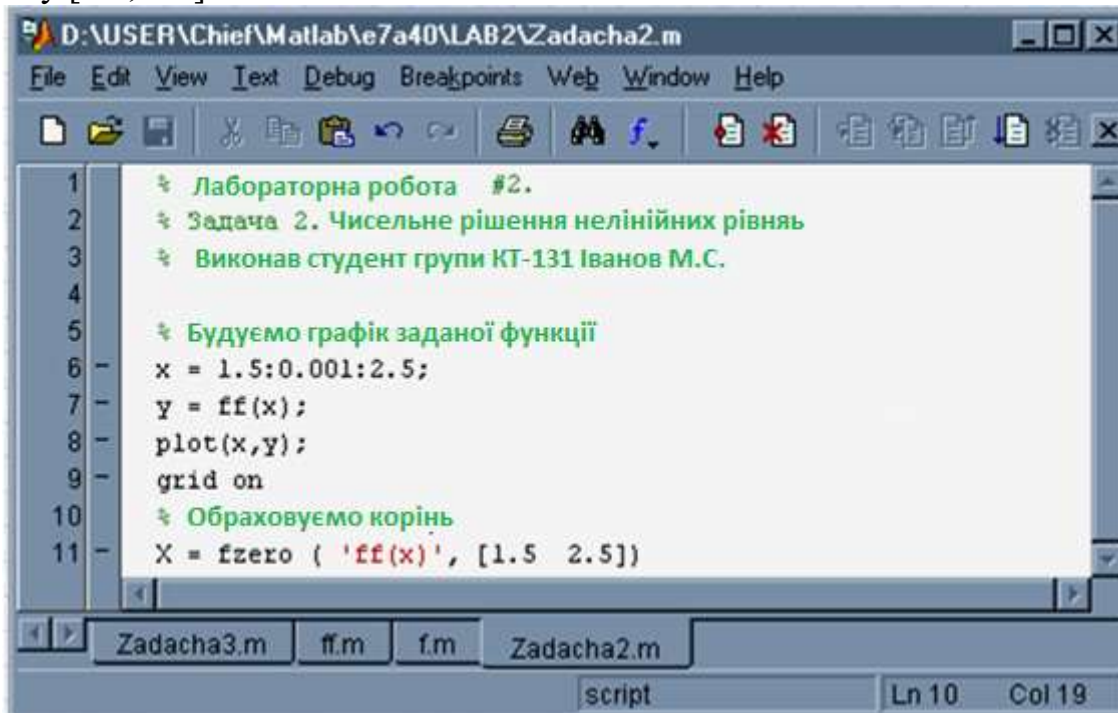
$$f(x) = 0.$$

Припустимо, що мається інтервал зміни  $x$   $[a; b]$ , на якому необхідно досліджувати функцію  $f(x)$  і знайти значення  $x_0$ , при якому  $f(x_0)$  дорівнює або вельми мало відрізняється від нуля.

Дане завдання в системі MatLab може бути вирішена наступним чином. Спочатку необхідно побудувати графік функції  $f(x)$  на заданому інтервалі і переконатися в існуванні кореня або декількох коренів. Потім застосувати програми пошуку коренів. Якщо існує один корінь і графік  $f(x)$  перетинає вісь  $ox$ , то можна застосувати програму fzero. Програма fzero використовує відомі

чисельні методи: поділ відрізка навпіл, січну і зворотнє квадратичне інтерполювання.

**Завдання.** Знайти корінь нелінійного рівняння  $10x + 2x - 100 = 0$  на відріжку  $[1.5; 2.5]$ .



```
1 % Лабораторна робота #2.
2 % Задача 2. Чисельне рішення нелінійних рівнянь
3 % Виконав студент групи КТ-131 Іванов М.С.
4
5 % Будуємо графік заданої функції
6 x = 1.5:0.001:2.5;
7 y = ff(x);
8 plot(x,y);
9 grid on
10 % Обраховуємо корінь
11 X = fzero ('ff(x)', [1.5 2.5])
```

Рисунок 13.16 – Знаходження кореня нелінійного рівняння

З'являється вікно з графіком функції  $10x + 2x - 100$ , з якого випливає, що корінь функції на заданому інтервалі існує. Для точного визначення кореня застосовуємо функцію `fzero` (рис. 13.17).

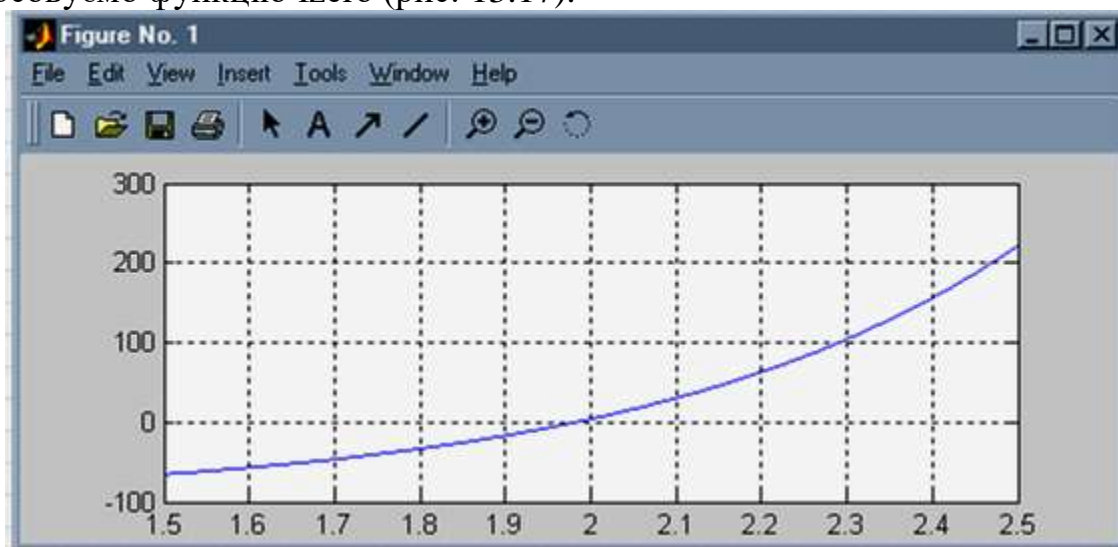


Рисунок 13.17 – Приклад використання функції `fzero`

Природно, що для роботи даної програми нам знадобиться `m`-функція:



Рисунок 13.18 – Виклик m-функції

Результат:  $X = 1.9824$

У більш загальному випадку функція `fzero()` має такі реалізації:

`fzero('f(x)', x)`  
`fzero('f(x)' [x1, x2])`

Тут прийняті наступні позначення:

- 'f(x)' – вирішуване рівняння, взяте в одиночні лапки;
- x - початкове наближення (значення) шуканого кореня;
- [x1, x2] – область ізоляції кореня.

Наприклад, для вирішення даного завдання ми могли б написати

$X = \text{fzero}('10.^X + 2.*x - 100.0', 2)$

Очевидно, що в даному випадку файл-функція не потрібна.

## II. Задання до лабораторної роботи

1. Прочитати теоретичний матеріал лабораторної роботи.
2. Виконати рис. 1-17.
3. Виконати зазначені завдання в теоретичних відомостях.
4. Виконати індивідуальні завдання.
5. Підготувати звіт про виконання завдань та надіслати до СПДН ФІТІС.

## III. Індивідуальні завдання

Побудувати графік з метою знаходження відрізка локалізації, на якому знаходиться тільки один корінь, і знайти даний корінь нелінійного рівняння на прикладі матеріалу, викладеного в пункті 4.

- 1)  $x \operatorname{arctg} x - 1 = 0$
- 2)  $e^{x-2} - \ln(x+2) = 0$
- 3)  $x^3 - 9x^2 + 5x - 6 = 0$
- 4)  $e^x - \frac{1}{x} - 1 = 0$
- 5)  $\operatorname{arctg} 2x - \frac{1}{1+x} = 0$

- 6)  $e^x - \ln x - 20 = 0$   
7)  $\sqrt{x} - \operatorname{tg}x(1-x) = 0$   
8)  $\sin \sqrt{x} - \cos \sqrt{x} + 2\sqrt{x} = 0$   
9)  $x^4 + 2x^3 - x - 1 = 0$   
10)  $x^3 - e^{4x} - 5.5 = 0$   
11)  $x^6 - 3x^2 + x - 1 = 0$   
12)  $\sqrt[3]{5-x} - x = 0$   
13)  $x^2 - \ln x = 0$   
14)  $x^2 - \cos x = 0$   
15)  $\ln x - \operatorname{arctg}x = 0$

## Лабораторна робота № 14

Тема: «Чисельне інтегрування в MatLab»

Питання:

1. Аналітичне інтегрування засобами MatLab.
2. Аналітичне диференціювання засобами MatLab.
3. Теоретична оцінка похибки чисельного інтегрування.
4. Чисельне інтегрування.
5. Чисельне інтегрування засобами MatLab.
6. Правило Рунге-Кута оцінки похибки інтегрування.

### I. Теоретичний матеріал

До обчислення певних інтегралів зводяться багато практичних завдань фізики, хімії, екології, механіки та інших природничих наук. На практиці взяти інтеграл аналітично не завжди вдається. У цьому випадку використовуються методи чисельного інтегрування. У даній лабораторній роботі розглядаються методи Ньютона-Котеса, зокрема методи прямокутників, трапецій і Сімпсона (парабол). Крім того, в лабораторній роботі розглядаються способи аналітичного і чисельного відшукування інтегралів засобами MatLab.

Обчислювальні формули для всіх розглянутих методів наведені в додатку, теоретичний матеріал слід вивчати за матеріалами лекцій та рекомендованої літератури. Тут же будуть розглянуті практичні аспекти реалізації методів чисельного інтегрування в MatLab.

Вам слід уважно вивчити і виконати наведені тут приклади, може бути для функцій з ваших варіантів завдань.

#### 1. Аналітичне інтегрування засобами MatLab.

У лабораторній роботі, основним завданням якої є дослідження методів чисельного інтегрування, в деяких випадках для відшукування похибки результатів потрібно точні значення інтегралів. Тобто ці величини необхідно визначити аналітично.

Для обчислення визначених і невизначених інтегралів в MatLab використовується функція `int`.

Таблиця 14.1 – Синтаксис

Виклик	Опис
<code>int(S)</code>	Обчислюється невизначений інтеграл від функції $S$ по її символічній змінній, визначеної в <code>syms</code> .
<code>int(S,v)</code>	Обчислюється невизначений інтеграл від функції $S$ по її символічній змінній $v$ , визначеної в <code>syms</code> .
<code>int(S,a,b)</code>	Обчислюється визначений інтеграл від $a$ до $b$ функції $S$ по її символічній змінній, визначеної в <code>syms</code> . $A$ і $b$ можуть бути змінними символічного або дійсного типу.
<code>int(S,v,a,b)</code>	Обчислюється визначений інтеграл від $a$ до $b$ функції $S$ по її символічній змінній $v$ , визначеної в <code>syms</code> . $A$ і $b$ можуть бути змінними символічного або дійсного типу.



```

1  % Обчислення точного значення інтеграла. Другий спосіб
2  a=1; b=1.44; % Межі інтегрування
3
4  sym x; % Задання символічної змінної
5  f=2.5*x^5-6.1*x^4+3.4*x^2-9.1*x+18; % Задання підінтегральної функції
6
7  In=int(f) % - невизначений інтеграл,
8  Io=int(f,a,b) % - визначений інтеграл.
9
10 format long e % Необхідне для переводу з
11 I=double(Io) % символічного формату даних в речовий.
12
13 % Графік підінтегральної функції:
14 fplot('f',[a b]) % Побудова графіка. Використовуються m-файл f.m
15 title('y=f(x)'), grid on % Невеликі приклади
16 xlabel('x'), ylabel('f(x)') % багаточисельних графічних операцій Matlab

```

```

>>
In =

5/12*x^6-61/50*x^5+17/15*x^3-91/20*x^2+18*x

Io =

1098807919/488281250

I =

2.250358618112000e+000

```

Рисунок 14.3 – Форматування графіка в тексті програми

Зверніть увагу, що в даному прикладі для побудови графіка використовується **m - файл f.m** (рис. 14.4):

```

function res=f(x)
% Лабораторна робота 4. Задача 1
% Функція, що інтегрується
res=2.5*x^5-6.1*x^4+3.4*x^2-9.1*x+18;

```

Рисунок 14.4 – Використання **m - файл** для побудови графіка

Це пов'язано з тим, що аргументом функції **fplot** не може виступати символічна функція.

Графік функції, що інтегрується.



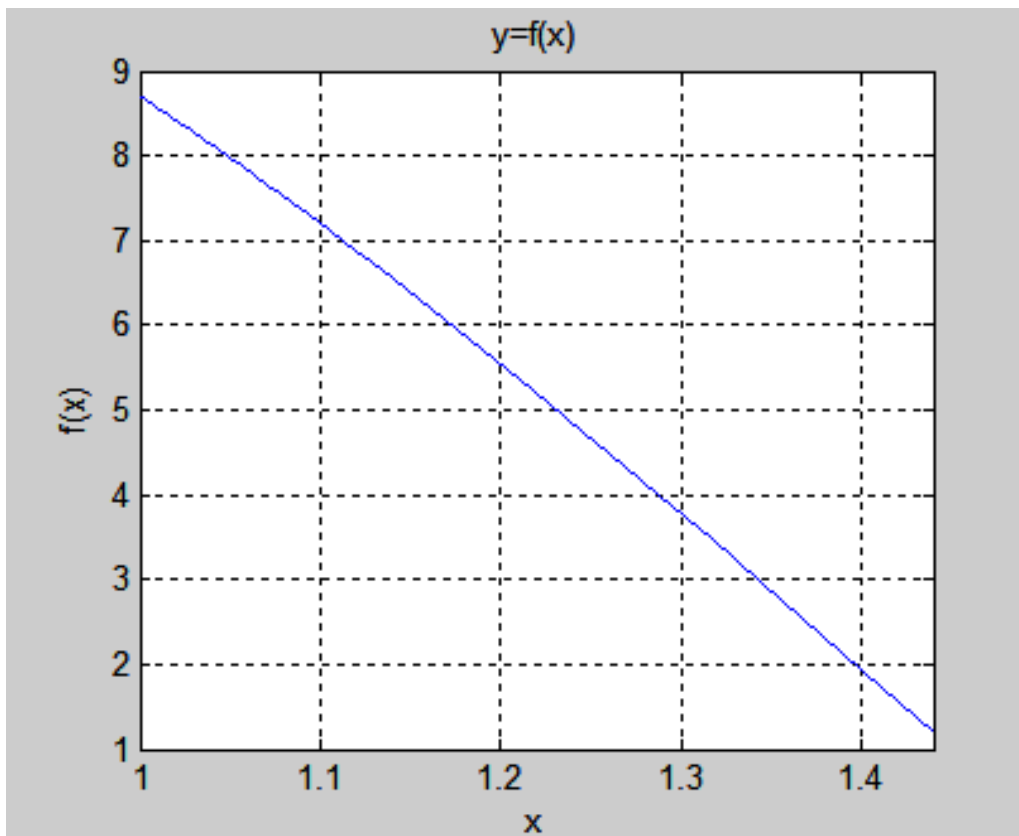


Рисунок 14.4 – Побудова графіка функції

## 2. Аналітичне диференціювання засобами MatLab

У лабораторній роботі, основним завданням якої є дослідження методів чисельного інтегрування, для відшукування теоретичної похибки результатів потрібне знання похідною від інтегрованої функції до четвертого порядку включно.

Для обчислення похідних аналітично в MatLab використовується функція **diff**.

Синтаксис:

Викликов	Опис
diff(S)	Обчислюється аналітичний вираз для похідної від функції S, заданої символьно.
diff(S,n)	Обчислюється аналітичний вираз для похідної від функції S порядку n.

Приклад використання функції diff при обчисленні максимуму другої похідної на відріжку [a, b] зображено на рис. 14.5

```

1  % Обчислення максимуму другої категорії
2
3  % Перша частина
4  syms x; % Задання символічної змінної
5  f=2.5*x^5-6.1*x^4+3.4*x^2-9.1*x+18; % Задання символічної функції
6
7  f2=diff(f,2) % Обчислення другої похідної
8
9  % Друга частина
10 a=1; b=1.44;
11 x=a:0.001:b;
12 % Наступний вираз береться з Command Windows
13 % після обчислення першої частини
14 m2=50.*x.^3-366/5.*x.^2+34/5;
15
16 format short
17 M2=max(abs(m2)) % Обчислення максимуму
18
19 % Графік другої похідної
20 plot(x,m2) % Побудова графіка
21 title('y=d^2f/dx^2'), grid on
22 xlabel('x'), ylabel('y')

```

Command Window

```

>>
f2 =
50*x^3-366/5*x^2+34/5

M2 =
16.4000

```

Рисунок 14.5 – Приклад використання функції diff

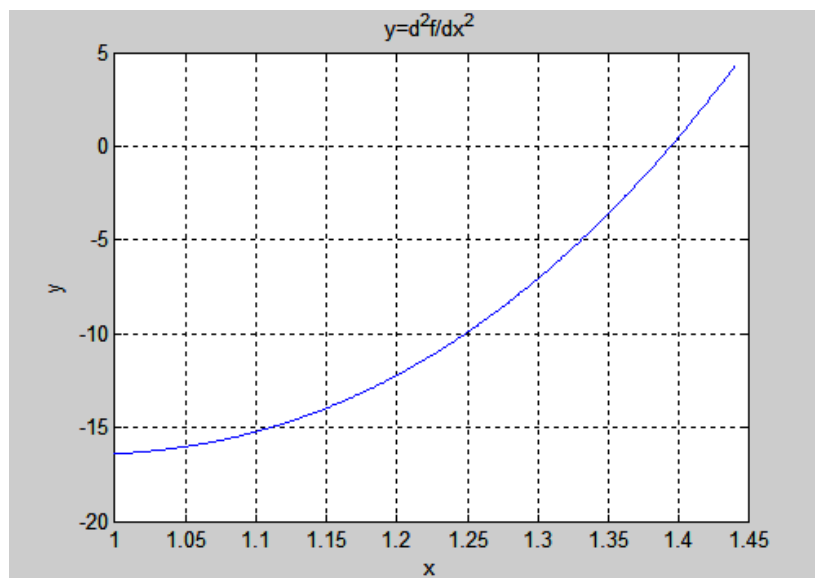


Рисунок 14.6 – Графік похідної

### 3. Теоретична оцінка похибки чисельного інтегрування

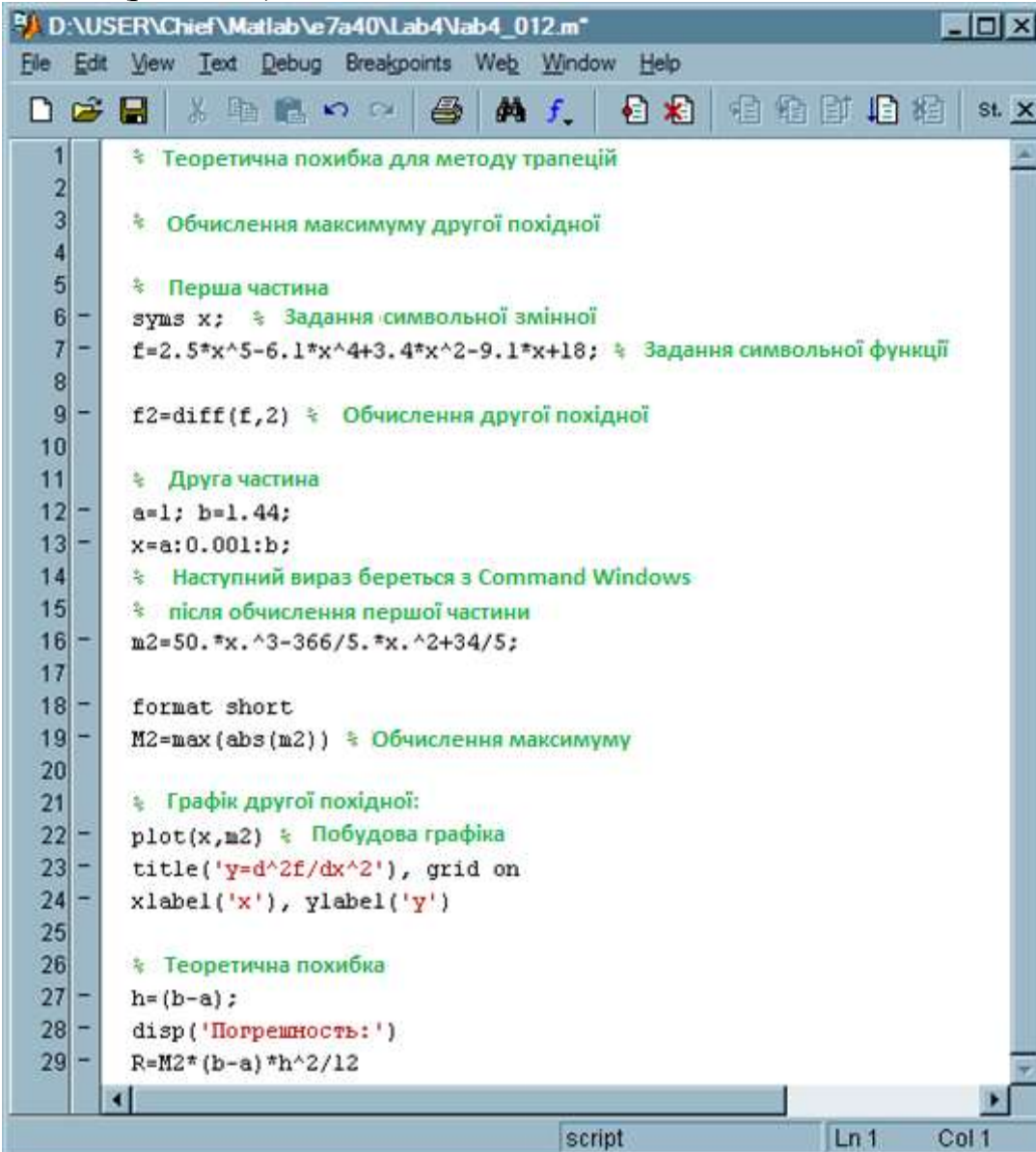
Розглянемо приклади використання теоретичної оцінки похибки інтегрування на прикладі двох завдань.

**Завдання 1.** Визначити теоретичну похибку чисельного інтегрування методом трапецій у випадку одного елементарного відрізка інтегрування ..

Теоретична похибка для методу трапецій становить

$$R = \frac{M_2 (b-a)h^2}{12}$$

У разі елементарного відрізка інтегрування, якщо задані межі інтегрування і підінтегральна функція, завдання можна вирішити наприклад таким чином (рис. 14.7):



```
1 % Теоретична похибка для методу трапецій
2
3 % Обчислення максимуму другої похідної
4
5 % Перша частина
6 - sum(x); % Задання символічної змінної
7 - f=2.5*x^5-6.1*x^4+3.4*x^2-9.1*x+18; % Задання символічної функції
8
9 - f2=diff(f,2) % Обчислення другої похідної
10
11 % Друга частина
12 - a=1; b=1.44;
13 - x=a:0.001:b;
14 % Наступний вираз береться з Command Windows
15 % після обчислення першої частини
16 - m2=50.*x.^3-366/5.*x.^2+34/5;
17
18 - format short
19 - M2=max(abs(m2)) % Обчислення максимуму
20
21 % Графік другої похідної:
22 - plot(x,m2) % Побудова графіка
23 - title('y=d^2f/dx^2'), grid on
24 - xlabel('x'), ylabel('y')
25
26 % Теоретична похибка
27 - h=(b-a);
28 - disp('Погрешность: ')
29 - R=M2*(b-a)*h^2/12
```

Рисунок 14.7 – Визначити теоретичну похибку чисельного інтегрування

В результаті отримаємо

f2 =

$$50*x^3-366/5*x^2+34/5$$

M2 =

16.4000

Погрешность

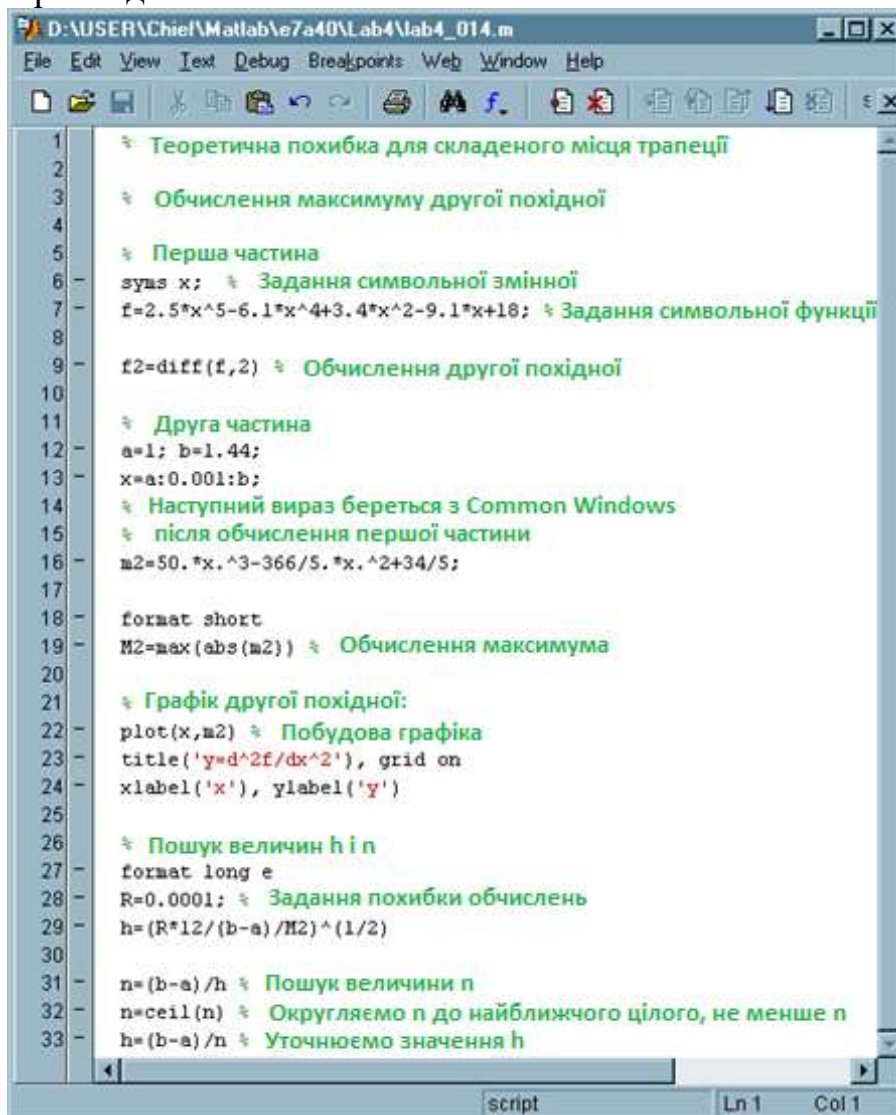
R =

0.1164

Т.ч. теоретична оцінка абсолютної похибки становить 0.1. У задачі 1 лабораторної роботи вам пропонується перекоонатися, що дана теоретична оцінка дійсно справедлива.

**Завдання 2.** Визначити крок дискретизації і кількість вузлів інтегрування, необхідні для того, щоб похибка інтегрування не перевищувала величини 0.0001.

Якщо задані межі інтегрування і інтегрована функція, завдання можна вирішити наприклад таким чином:



```
D:\USER\Chief\Matlab\7a40\Lab4\lab4_014.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons]
1  % Теоретична похибка для складеного місця трапеції
2
3  % Обчислення максимуму другої похідної
4
5  % Перша частина
6  sym x; % Задання символічної змінної
7  f=2.5*x^5-6.1*x^4+3.4*x^2-9.1*x+18; % Задання символічної функції
8
9  f2=diff(f,2) % Обчислення другої похідної
10
11 % Друга частина
12 a=1; b=1.44;
13 x=a:0.001:b;
14 % Наступний вираз береться з Common Windows
15 % після обчислення першої частини
16 m2=50.*x.^3-366/5.*x.^2+34/5;
17
18 format short
19 M2=max(abs(m2)) % Обчислення максимуму
20
21 % Графік другої похідної:
22 plot(x,m2) % Побудова графіка
23 title('y=d^2f/dx^2'), grid on
24 xlabel('x'), ylabel('y')
25
26 % Пошук величин h і n
27 format long e
28 R=0.0001; % Задання похибки обчислень
29 h=(R*12/(b-a)/M2)^(1/2)
30
31 n=(b-a)/h % Пошук величини n
32 n=ceil(n) % Округляємо n до найближчого цілого, не менше n
33 h=(b-a)/n % Уточнюємо значення h
script Ln 1 Col 1
```

Рисунок 14.8 – Визначення кроку дискретизації і кількості вузлів інтегрування  
В результаті отримаємо

```

f2 =
50*x^3-366/5*x^2+34/5
M2 =
16.4000
h =
1.289562396771206e-002
n =
3.412010160203707e+001
n =
35
h =
1.257142857142857e-002

```

Таким чином, щоб обчислити інтеграл від функції  
 $f(x)=2.5*x^5-6.1*x^4+3.4*x^2-9.1*x+18$   
за методом трапецій з точністю 0.0001, необхідно задати 35 вузлів інтегрування.  
При цьому крок дискретизації становить  $h = 1.257142857142857 * 10^{-2}$ .

Вам залишається переконатися, що похибка інтегрування з даними параметрами дійсно не перевищує 0.0001.

#### 4. Чисельне інтегрування

Обчислювальні формули наведені в додатку. Їх аналіз показує, що формули для одного елементарного відрізка інтегрування не вимагають будь-яких нових знань. Необхідно тільки приготувати m-функцію, в якій слід визначити інтегруються функцію. І звичайно для оцінки похибки інтегрування слід знати точне значення інтеграла (див. Вище).

Випадок формул для складеного відрізка інтегрування більш складний. Тут потрібне обчислювати значення сум. Наприклад в методі Сімпсона

$$I = \frac{h}{6} \times \left[ f(a) + 4 \sum_{i=0}^{n-1} f(x_{i+1/2}) + 2 \sum_{i=1}^{n-1} f(x_i) + f(b) \right]$$

де

$$x_i = a + i \times h \quad x_{i+1/2} = a + (i + 1/2) \times h = a + i \times h + h/2 \quad h = (b - a) / n$$

У мові программовані MATLAB, як і в інших мовах програмування, існує оператор циклу for, можливостей якого досить для вирішення поставлених завдань. Синтаксис оператора циклу описаний в розділі Програмування. Далі ми просто розглянемо приклад вирішення задачі чисельного інтегрування.

**Приклад.** Чисельне інтегрування методом Сімпсона.

```

1  % Чисельне інтегрування методом Сімпсона
2  a=1; b=1.44; % Межі інтегрування
3  format long e
4  disp('Результат')
5
6  % А. Обчислення точного значення інтеграла
7  syms x; % Задання символічної змінної
8  f=4-exp(x); % Задання підінтегральної функції
9
10 In=int(f); % - невизначений інтеграл,
11 Io=int(f,a,b); % - визначений інтеграл.
12 I=double(Io); % Необхідне для перетворення результату з
13 % символічного формату даних в речовий.
14 disp('Точне значення інтеграла '), disp(I)
15
16 % Б. Графік підінтегральної функції
17 fplot('f1',[a b]) % Побудова графіка. Використовується m-файл f1.m
18 title('y=f1(x)'), grid on % Невеликі приклади
19 xlabel('x'), ylabel('f(x)') % багаточисельних граничних опцій Matlab
20
21 % В. Обчислення методом Сімпсона для елементарного відрізка
22
23 САМОСТІЙНО
24
25 disp('Результат і похибка для елементарного відрізка ')
26 disp([s d])

```

Рисунок 14.9 – Чисельне інтегрування методом Сімпсона

Замальовану область слід обрахувати самостійно.

```

27
28 % Г. Обчислення за загальним методом Сімпсона для 4-ох вузлів
29 n=4;
30 h=(b-a)/n;
31
32 sum=f1(a)+f1(b);
33
34 for i=0:n-1
35     sum=sum+4*f1(a+i*h+h/2);
36 end
37
38 for i=1:n-1
39     sum=sum+2*f1(a+i*h);
40 end
41
42 sum=sum*h/6;
43
44 % Д. Обчислення абсолютної похибки
45 d=abs(I-sum);
46
47 % Е. Виведення результатів
48 disp('Кількість вузлів '), disp(n)
49 disp('Результат і похибка по методу Сімпсона '), disp([sum d])

```

Рисунок 14.9 – Чисельне інтегрування методом Сімпсона

## Результаты

>> Результаты

Точное значение интеграла

2.575860114624924e-001

Результат и погрешность для элементарного отрезка

2.575665708043314e-001 1.944065816106289e-005

Количество узлов

4

Результат и погрешность по методу Симпсона

2.575859351120611e-001 7.635043136611230e-008

Звичайно, розглянутий тут спосіб вирішення завдання є не єдиним і не найкращим. Зокрема, можна було б створити m-функцію, в якій і вироблялося б обчислення інтеграла.

## 5. Чисельне інтегрування засобами MatLab

В MatLab реалізовані безліч сучасних методів чисельного інтегрування. Ми розглянемо найпростіші з них.

**А. Метод трапецій.** Метод трапеції реалізований в MatLab декількома функціями:

- `cumtrapz(y)`
- `cumtrapz(x,y)`
- `trapz(y)`
- `trapz(x,y)`

Найбільш цікава остання з них. Ця функція обчислює інтеграл від функції  $y(x)$  по  $x$  методом трапецій. Аргумент і функція задаються у вигляді векторів або  $x - y$  у вигляді вектора, а  $y - u$  у вигляді матриці. Якщо  $y(x)$  - матриця, то функція повертає вектор значень інтеграла для кожного шпальти матриці.

**Приклад.** Нехай подинтегральна функція має вигляд

$$y(x) = x * e^x + \ln(x) + 1$$

Необхідно обчислити визначений інтеграл в діапазоні від 1 до 10 з кроком 0.5.

Рішення:

```
x=1:0.5:10;
```

```
y=x.*exp(x)+log(x)+1;
```

```
trapz(x,y)
```

Рішення:

```
ans =
```

```
2.032841320958599e+005
```

**Б. Метод Сімпсона.** Метод Сімпсона реалізований в MatLab також декількома функціями. Ми розглянемо найпростішу з них:

```
quad ('fun', a, b)
```

```
quad ('fun', a, b, tol)
```

Позначення:

'fun'- подинтегральна функція, взята в одинарні лапки;

a, b - межі інтегрування;

tol - відносна похибка, що задається користувачем; за замовчуванням tol =  $10^{-3}$ .

**Приклад.** Обчислити інтеграл від функції  $x+e^x$  на відрізку [1, 2] з точністю  $10^{-5}$ .

```
quad('x+exp(x)', 1, 2, 1e-5)
```

Результат:

```
ans = 6.170774280645000e+000
```

## 6. Правило Рунге оцінки похибки інтегрування

У формулах для оцінки похибки квадратурних формул R використовуються значення похідних підінтегральної функції, що вимагає додаткового аналізу і обчислень. У зв'язку з цим набуло поширення практичне правило Рунге оцінки похибки.

Нехай

I - точне значення інтеграла,

I (n) - значення інтеграла обчислене при n вузлах інтегрування  $h = (b-a) / n$ ,

I (2n) - значення інтеграла обчислене при  $2 * n$  вузлах інтегрування,  $h = (b-a) / 2n$ .

Необхідно визначити, з якою точністю обчислений інтеграл I (2n), тобто знайти абсолютну похибку

$$\Delta = |I - I(2n)|$$

Для безпосередньо визначення даної похибки необхідно знайти максимум модуля відповідної похідної від інтегрованої функції на відрізку [a, b]. Часто це досить трудомісткий або взагалі неможливий процес. Наприклад якщо інтегрована функція задана таблично. У таких випадках оцінку похибки величини I (2n) можна провести наступним чином:

$$\Delta = \frac{|I(2n) - I(n)|}{m}$$

Тут  $m = 3$  для методів середніх прямокутників і трапецій,  $m = 15$  для методу Сімпсона.

Примітка. Якщо вирішується завдання чисельного обчислення інтеграла із заданою точністю, процес подвоєння числа вузлів інтегрування продовжується до тих пір, поки величина не стане менше заданої похибки.

## II. Задання до лабораторної роботи

1. Прочитати теоретичний матеріал лабораторної роботи.
2. Обрахувати звідки та виконати побудову графіків.
3. Підготувати звіт про виконання завдань та надіслати до системи електронного навчання.



## ОСНОВНІ ТЕРМІНИ І ВИЗНАЧЕННЯ

**Алгоритмічна інформаційна система** – елементи- алгоритми, а зв'язки реалізуються за допомогою інформаційних масивів.

**Аналогові обчислювальні машини (АОМ)** – використовуються для обробки неперервних повідомлень.

**Апаратура комп'ютера.** Введення програми в комп'ютер, її виконання і одержання результатів роботи забезпечується завдяки різноманітним електронним і електромеханічним пристроям.

**Високі технології** – найбільш високорозвинуті (найсучасніші) технології, що є «наукоємними», тобто які інтенсивно використовують найновіші наукові досягнення.

**Гіперпосилання** – кольоровий підкреслений текст або графічне зображення, вибір якого дозволяє виконати перехід до іншого файлу, Web-сторінки та ін.

**Гіпертекст** – це інформація, яка подана таким чином, що певні слова в тексті можуть бути пов'язані з додатковою інформацією різного виду (іншим текстом, малюнком, звуковим файлом, відеокліпом та ін.), тобто ці слова є посиланням на інші документи. Такі слова називають гіперпосиланнями.

**Документальна інформаційна система** – елементи–документи ІС, а зв'язки між елементами – взаємодії, вихідності і супідрядності.

**Екстремальними (оптимізаційними) задачами** називаються задачі на відшукання максимуму чи мінімуму певних величин за наявності або відсутності обмежень на параметри, від яких вони залежать.

**Застаріла технологія** – технологія, що застаріває й замінюється більш досконалою, на неї малий попит, або повна відмова від цієї технології на користь нової.

**Ідеальні системи** – це такі системи, елементами яких є ідеальні об'єкти (поняття, ідеї), які зв'язані певними відношеннями.

**Інформаційна система** – це сукупність технічного, програмного і організаційного забезпечення, а також персоналу, що призначена для своєчасного забезпечення осіб, що приймають рішення, потрібною інформацією.

**Інформаційна технологія (information technology)** – система методів і засобів збирання, накопичення, збереження, пошуку, опрацювання і передавання інформації.

**Інформаційна технологія** – це цілеспрямована організована сукупність інформаційних процесів з використанням засобів комп'ютерної техніки і методів, що забезпечують високу швидкість обробки даних, оперативний пошук

інформації, розподілення даних, доступ до інформаційних ресурсів незалежно від місця їх розташування.

**Інформаційний ресурс** – це особливий вид ресурсу, що ґрунтується на ідеях і знаннях, накопичених у результаті науково-технічної діяльності людей і поданий у формі, придатній для збирання, реалізації та відтворення.

**Інформаційно-комунікаційні технології або ІКТ** – технології, пов'язані зі створенням, збереженням, передачею, обробкою інформації та управлінням нею, а також технології, що використовуються для спілкування користувачів та їх роботи з інформацією.

**Інформація** – будь-які відомості про ті чи інші явища природи, процеси в технічних пристроях, події в суспільному житті, які деяка система сприймає з навколишнього середовища, зберігає в собі, переробляє і видає в навколишнє середовище.

**Комп'ютерна математика** – це сукупність методів і засобів, які забезпечують зручну і швидко підготовку математичних моделей досить складних реальних задач, процесів і явищ, а також алгоритмів і програм, які реалізують аналітичні, графічні і чисельні методи їх розв'язування.

**Критерій ефективності операції** – показник необхідної, бажаної, досягнутої відповідності між результатом дій, що було здійснено, і метою операції.

**Масив** - впорядкований набір фіксованої кількості однотипних елементів, що зберігаються в послідовно розташованих комірках оперативної пам'яті, мають порядковий номер і спільне ім'я, що надає користувач.

**Математичні вирази** – це складні (комбіновані) об'єкти, які складаються з операторів, операндів і функцій з їх параметрами.

**Матеріальні системи** – це системи, які складаються з матеріальних елементів, що пов'язані між собою певними відношеннями. Вони бувають відносно простими і відносно складними.

**Модель (modulus - міра, зразок)** – це матеріальна, знакова або уявна система, що відтворює, імітує або відображає принципи внутрішньої організації, функціонування, певні властивості, ознаки чи характеристики об'єкта дослідження (оригіналу), безпосереднє вивчення якого неможливе, ускладнене або недоцільне, і може замінити об'єкт дослідження в пізнавальному процесі з метою одержання нових знань про нього.

**Мультимедіа (multimedia) система** – інтерактивна система, яка забезпечує роботу з нерухомими зображеннями і «живим» відео, анімаційною комп'ютерною графікою і текстовою інформацією, природною мовою і високоякісним звуком.

**Не нова технологія** – ще корисна технологія, але вже існує більш нова технологія, тому попит на неї починає знижуватися.

**Новітня технологія** – будь-яка нова технологія, котра має великий потенціал для певної сфери людської діяльності.

**Операція** – сукупність взаємоузгоджених дій, спрямованих на досягнення цілком визначеної мети.

**Організаційна інформаційна система** – елементи - колективи людей і окремі виконавці, а зв'язки між елементами – інформаційні, супідрядності та взаємодії.

**Передова технологія** – технологія, котра зарекомендувала себя, але ще досить нова і має не досить широке поширення на ринку.

**Перетворення системи** – це зміна вихідних параметрів під впливом вхідних.

**Перспективні технології** – це технології, характер котрих у теперішній час визначено як «передовий» та існує можливість для їх реалізації.

Під **моделлю** потрібно розуміти матеріальну, знакову або уявну (мислену) систему, що відтворює, імітує чи відображає принципи внутрішньої організації або функціонування, певні властивості, ознаки чи характеристики об'єкта дослідження (оригіналу), безпосереднє вивчення якого неможливе, ускладнене або недоцільне, і може в пізнавальному процесі замінити об'єкт дослідження з метою одержання нових знань про нього.

**Повідомлення** – інформація, втілена та зафіксована в деякій матеріальній формі.

**Програмне забезпечення (ПЗ) або Software** – набір програм, що забезпечують ефективну взаємодію комп'ютера із зовнішнім середовищем, а також взаємодію людини з комп'ютером.

**Протокол** – сукупність правил, згідно яких комп'ютери взаємодіють між собою.

**Ранжовані змінні** – особливий клас змінних, які мають ряд фіксованих значень, які змінюються з деяким кроком  $h$  від початкового значення  $N_{begin}$  до кінцевого значення  $N_{end}$ .

**Система** – являє собою сукупність скінченої кількості елементів (об'єктів), відношень і зв'язків між цими елементами, які утворюють єдине ціле і мають загальну мету функціонування.

**Структура системи** – спосіб її існування, який фіксує певні пріоритети і взаємозв'язки елементів цієї системи.

**Сучасна інформаційна технологія** – це ІТ, що базується на сучасних мультимедіа і комп'ютерних телекомунікаційних системах.

**Сучасна технологія** – технологія, що є стандартом і на неї є підвищений попит.

**Термін «модель»** – походить від латинського *modulus* (міра, мірило, зразок).

**Технічна інформаційна система** – це структура, елементами якої є обладнання комплексу технічних засобів ІС, а зв'язки відтворюють інформаційний обмін.

**Технологія (technology)** – система взаємопов'язаних засобів обробки матеріалів, прийомів і методів виготовлення продукції під час виробничого процесу.

**Функціональна інформаційна система** – це структура, елементами якої є підсистеми (компоненти), а зв'язки між елементами - це потоки інформації, що циркулюють між ними під час функціонування ІС.

**Цифрові електронні обчислювальні машини (ЦЕОМ, або ЕОМ)** використовуються для обробки дискретних цифрових повідомлень.

## ЛІТЕРАТУРА

### Базова

1. Говорухин В., Цибулин В. Компьютер в математических исследованиях. – СПб.: Питер, 2001. – 624 с.
2. Дьяконов В. П. Mathcad 2001: учебный курс. – СПб: Питер, 2001. – 624с.
3. Дьяконов В. П. Компьютерная математика. Теория и практика. – М.: Нолидж, 2001. – 1296 с.
4. Кирьянов Д. MathCAD 11. Самоучитель. – С.-Пб.: БХВ-Петербург, 2003.
5. Плис А., Славина Н. Mathcad. Математический практикум для инженеров и экономистов. – М.: Финансы и статистика, 2003.

### Допоміжна

1. Ануфриев И.Е., Смирнов А.Б., Смирнова Е.Н. МАТЛАВ7. - СПб.: БХВ-Петербург, 2005. – 1104 с.
2. Аладьев В. З. Системы компьютерной алгебры: Maple: Искусство программирования. — М.: Лаборатория Базовых Знаний, 2006. — 792 с.
3. Васильев А. Н. Maple 8: Самоучитель / Алексей Николаевич Васильев. — М. ; СПб. ; К. : Издательский дом „Вильямс», 2003. — 351 с.
4. Гаев Є. О. Універсальний математичний пакет MatLab і типові задачі обчислювальної математики: Навч. посіб. [для студ. техн. спец. вищих навч. закл.]/Є. О. Гаєв, Б. М. Нестеренко. — К. : НАУ, 2004. — 175 с.
5. Дьяконов В. П. МАТНЕМАТИСА 5.1/5.2/6.0. Программирование и математические вычисления / Владимир Павлович Дьяконов. — М. : ДМК Пресс, 2006. — 576 с.
6. Steinhaus S. Comparison of Mathematical Programs for Data Analysis (Edition 5.03) [Электронный ресурс] – Munchen/Germany. – 64 p. – Режим доступа : <http://www.scientificweb.de/ncrunch/>.
7. Васильев А. Н. Mathcad 13 на примерах . – СПб.: БХВ-Петербург, 2006. – 528 с.: ил

### Інформаційні ресурси

1. Система підтримки дистанційного навчання ЧДТУ. [Електронний ресурс] – Режим доступу: <http://ias.cdtu.edu.ua/>.
2. Перетворення чисел у різні системи числення [Електронний ресурс] – Режим доступу: [http://office.microsoft.com/uk-ua/excel-help/HA010070511.aspx#BMconverts\\_a\\_decimal\\_number\\_to\\_binary](http://office.microsoft.com/uk-ua/excel-help/HA010070511.aspx#BMconverts_a_decimal_number_to_binary).
3. Відеокурс «Спеціальні засоби для розв'язування систем лінійних рівнянь в Mathcad 14» числення [Електронний ресурс] – Режим доступу: <http://www.teachvideo.ru/v/2019>

### Програмне забезпечення

1. Офіційний сайт Mathcad. [Електронний ресурс] – Режим доступу: <http://mathcad.com.ua/>
2. Офіційний сайт Matlab. [Електронний ресурс] – Режим доступу: <http://www.mathworks.com/>
3. Офіційний сайт Smath Studio [Електронний ресурс] – Режим доступу: <http://ru.smath.info/>
4. Офіційний сайт Maxima. [Електронний ресурс] – Режим доступу: <http://maxima.sourceforge.net/>

Додаток А

Міністерство освіти і науки України  
Черкаський державний технологічний університет

Кафедра комп'ютерних наук та інформаційних технологій управління

Дисципліна: «Інформаційні технології аналізу систем»

## **З В І Т**

**про виконання індивідуального завдання № 1**  
на тему: *«Інтерфейс, типи даних і математичні функції в СКМ Mathcad»*

***Коваленка Віктора Івановича***

Подано на перевірку \_\_\_\_\_  
(Дата) (Підпис студента)

Оцінка \_\_\_\_\_

Перевірено \_\_\_\_\_  
(Дата)

Викладач \_\_\_\_\_ / \_\_\_\_\_ /  
(Підпис) (Прізвище та ініціали)

Черкаси – 2018 р

## **З В І Т**

про виконання індивідуального завдання № 1

Тема: *«Інтерфейс, типи даних і математичні функції в СКМ Mathcad»*

1. Мета роботи:
2. Завдання для виконання:
3. Протокол розв'язування завдань.
4. Висновки: