

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут»

## Основи роботи з MathCad.

Методичні вказівки до виконання комп'ютерного практикуму  
з дисципліни «Комп'ютерні технології та програмування -2.  
Програмні засоби для числового аналізу»,  
для студентів напряму підготовки «Автоматизація та комп'ютерно-інтегровані  
технології»

*Рекомендовано Вченою радою інженерно-хімічного факультету*

Київ  
НТУУ «КПІ»  
2015

Основи роботи з MathCad. Методичні вказівки до виконання комп'ютерного практикуму з дисципліни «Комп'ютерні технології та програмування -2. Програмні засоби для числового аналізу», для студентів напряму підготовки «Автоматизація та комп'ютерно-інтегровані технології» / Уклад.: О.В. Ситніков, 2015. – 101с.

*Гриф надано Вченою радою ІХФ  
(Протокол № від 2015р.)*

Навчальне видання

## ОСНОВИ РОБОТИ З MATHCAD.

Методичні вказівки до виконання комп'ютерного практикуму з дисципліни «Комп'ютерні технології та програмування -2. Програмні засоби для числового аналізу», для студентів напряму підготовки «Автоматизація та комп'ютерно-інтегровані технології»

Укладачі: Ситніков Олексій Володимирович

Відповідальний редактор А.І.Жученко, д-р техн.наук, проф.

Рецензенти : А.Р. Степанюк, к.т.н., доц.

Авторська редакція

## Вступ

На сучасному ринку програмного забезпечення для виконання розрахунків і зручного подання їх результатів існує велика кількість математичних пакетів. До них належать *Eureka*, *MathCad*, *Matlab*, *Maple*, *Mathematica* та ін. Їх умовно можна розділити на дві групи.

До першої групи належать системи з переважно числовим поданням інформації (*Eureka*, *MathCad*, *Matlab* та ін.). Ці пакети пропонують широкі можливості для застосування числових методів аналізу та графічної інтерпретації результатів. Найпотужнішим серед останніх є *Matlab*, пристосований для інженерних та наукових розрахунків, моделювання фізичних систем та керування технічними об'єктами. *MathCad*, поступаючись йому у спектрі можливостей, є максимально наближеним до звичайної мови описання математичних задач.

Необхідність застосування пакетів з числовим поданням інформації обумовлена неможливістю для ряду інженерних задач отримати аналітичний розв'язок.

До другої групи математичних пакетів належать системи з символьним поданням інформації (*Maple*, *Mathematika*). Вони призначені для розв'язання здебільшого суто математичних задач – аналітичних перетворень і розрахунків.

Даний курс лабораторних робіт призначений для вивчення можливостей *MathCad*, що застосовуються для розв'язання типових інженерних і математичних задач студентами напряму “Автоматизація та комп'ютерно-інтегровані технології” спеціальності “Автоматизоване управління технологічними процесами”, але може бути корисним і для студентів інших технічних спеціальностей.

## ОСНОВИ РОБОТИ З MATHCAD. ПОБУДОВА ВИРАЗІВ. ОБЧИСЛЕННЯ

**Мета роботи** – дослідити можливості редактора формул.

### Теоретичні відомості

Створення математичних виразів виконується за підтримки формульного редактора. До складу математичного виразу входять оператори та операнди. Прості операнди можуть бути числом, ідентифікатором змінної, матриці, функції. Складні оператори, в свою чергу, є математичними виразами. Оператор може складатися з символів або декількох символів, і вказує *MathCad*'у, яку дію виконати над операндами (операндом), що належать до цього оператора.

Оператори є арифметичні, розширені, порівняння, логічні, надання значення та виведення в документі *MathCad* результату обчислень.

Звертатися до них зручно, відкривши кнопкою з палітри математичних знаків відповідне вікно. На рис.1.1 панель палітр математичних знаків розташовано поряд з лінійкою вертикальної прокрутки. Кнопки цієї палітри відкривають вікна з відповідними номерами, наведеними на рисунку.

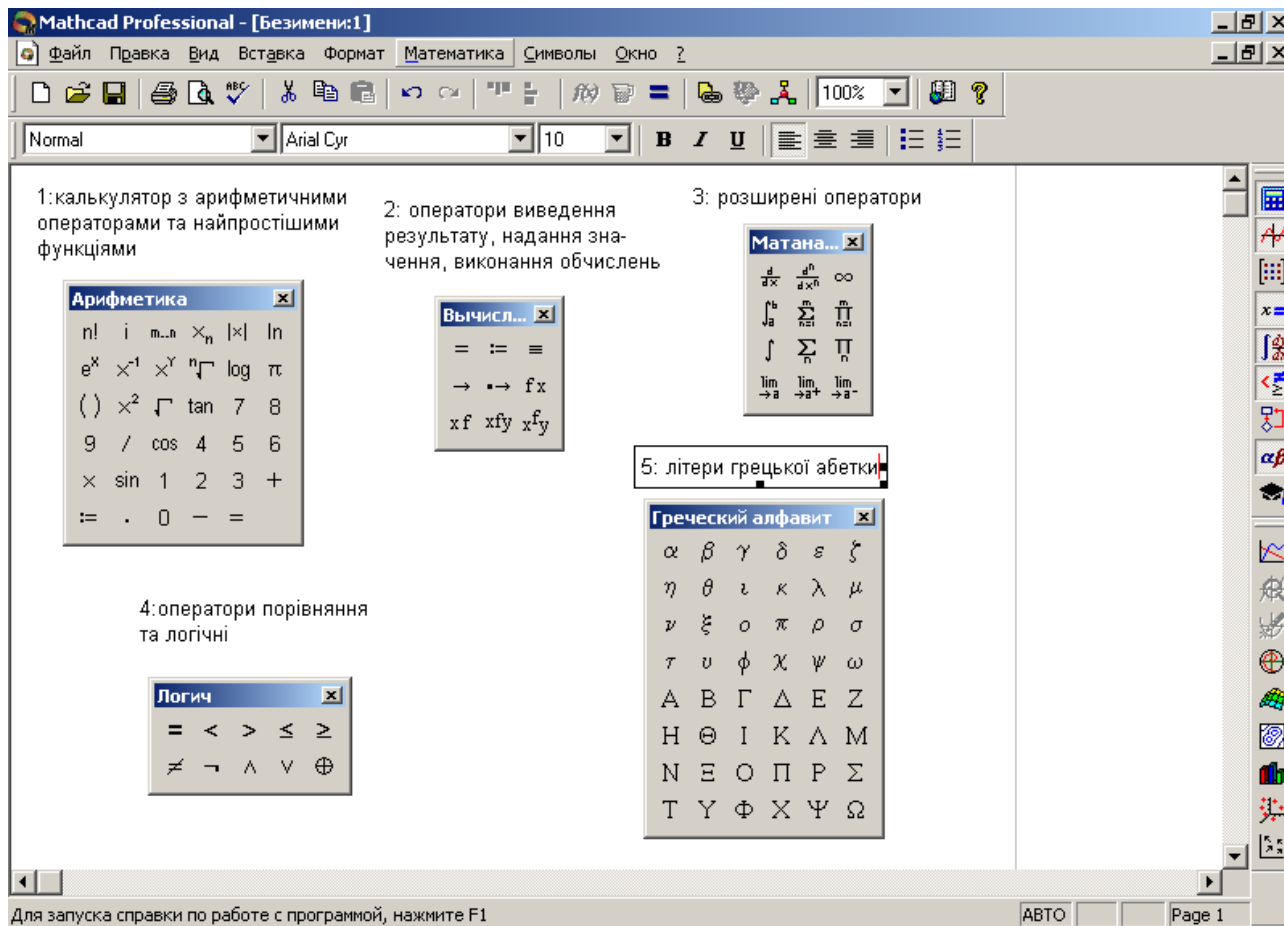


Рис.1.1. Палітри з панелі математичних символів:

- 1 – палітра обчислень;
- 2 – палітра операцій надання значення та виведення;
- 3 – палітра математичних операцій;
- 4 – палітра булевих операцій;
- 5 – палітра грецьких символів

Найчастіше використовуваний оператор надання значення в документі *MathCad* має вигляд  $:=$  та вводиться кнопкою з таким зображенням 1 або 2 (рис.2.1). При першому наданні значення змінній можна використати кнопку з вікон 1 чи 2 (рис.1.1) або клавішу зі знаком рівності “=”. Якщо змінна в документі вже набула значення, після введення знаку рівності праворуч від нього буде виведено це значення.

Якщо змінній в певному місці документу *MathCad* надано значення оператором  $:=$ , то воно буде відомо *MathCad* правіше та нижче цього місця.

Якщо ж змінна набула значення за допомогою оператора глобального надання значення  $\equiv$  (кнопка з вікна 2 на рис.1.1) в будь-якому місці документу, то вона має це значення також у будь-якому місці документу.

*MathCad* розрізняє три типи змінних: скаляр, матриця (або вектор) та рядок. Скаляр (змінна або константа ) може бути заданий такими типами даних:

- цілим числом ;
- дійсним числом з мантиєю і порядком (наприклад,  $2.564 \times 10^{-3}$  );
- двійковим числом, що закінчується літерою *b* (від слова binary);
- вісімковим числом, що закінчується літерою *o* ( від слова octal);
- шістнадцятковим числом, що закінчується літерою *h* (від слова hexadecimal); якщо це число починається з літери, то перед нею треба поставити нуль;
- числом з розмірністю;
- комплексним числом.

Комплексне число подається сумою його дійсної та уявної частини, поряд з якою праворуч вказується позначення уявної одиниці *i* або *j*.

Встановити бажане позначення уявної одиниці можна, звернувшись до команди **Format** (Формат)=> **Result** (Результат). У вікні, що відкривається, вкладка **Result Format** (Настройка показа) дає таку можливість командою **Imaginary Value** (Мнимые значения) (рис.1.2).

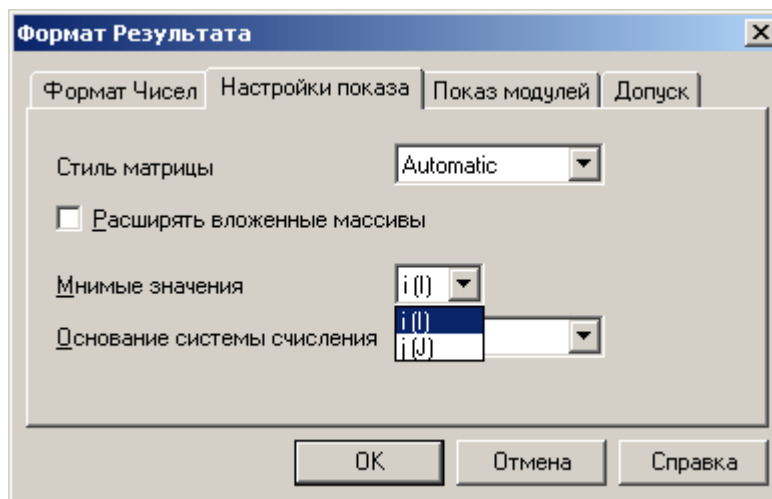


Рис.1.2. Возможность зміни подання уявної одиниці

Якщо уявна частина комплексного числа дорівнює одиниці, її треба обов'язково вказати перед позначенням уявної одиниці. При цьому в документі *MathCad* таке комплексне число (наприклад,  $0,3+1i$ ) відобразиться як  $0.3+i$ .

Тип “рядок” задається послідовністю символів, обмежених лапками. Для дій над змінними цього типу призначені такі функції:

- ***concat(S1,S2)*** – об'єднує рядкові змінні *S1* та *S2*;
- ***num2str(N)*** – перетворює скалярну змінну *N* (число) у змінну типу “рядок”;
- ***str2num(S)*** – перетворює значення рядкової змінної *S* у число;
- ***str2vek(S)*** – повертає вектор, елементами якого є *ASCII*-коди символів, з яких складається змінна *S*;
- ***strlen(S)*** – визначає кількість символів, що входять до складу рядкової змінної *S* та ін.

Формульний редактор синхронно з введенням математичного виразу перевіряє його коректність та наявність значень змінних, ідентифікатори яких входять до складу виразу, обчислює значення виразу та виводить його значення після введення оператора обчислення результату (знаку рівності).

Якщо функції редактора гальмують формування документа, можна вимкнути їх, знявши прапорець з функції ***Automatic Calculation*** команди ***Math*** (Математика) головного меню.

Якщо в документі вказано ідентифікатор, після якого в круглих дужках записані один або декілька (через кому) ідентифікаторів або виразів, то цей запис *MathCad* сприймає як ідентифікатор функції з одним або декількома аргументами, наприклад:

$$F1(x,a):=a\sin(ax)$$

або

$$F1(\pi,-0.8)=$$

Крім звичайної змінної у *MathCad* існує дискретний аргумент.

На відміну від звичайної змінної, що в будь-який момент зберігає лише одне значення, дискретний аргумент (ранжована змінна) зберігає сукупність значень та задається так:

$N := Nstart, Nnext..Nend$

або

$N := Nstart..Nend.$

Тут  $Nstart$  і  $Nend$  – початкове і кінцеве значення дискретного аргумента,  $Nnext$  – наступне після  $Nstart$  значення. Отже, різниця  $(Nnext - Nstart)$  — це крок, з яким змінюється значення дискретного аргумента в межах між  $Nstart$  і  $Nend$ ; він може бути цілим або дійсним, додатнім або від’ємним.

Якщо використано скорочений формат дискретного аргумента (без заданого значення  $Nnext$ ), то кроком є 1 або -1 залежно від того, чи  $Nstart < Nend$  або  $Nstart > Nend$ . Наприклад:

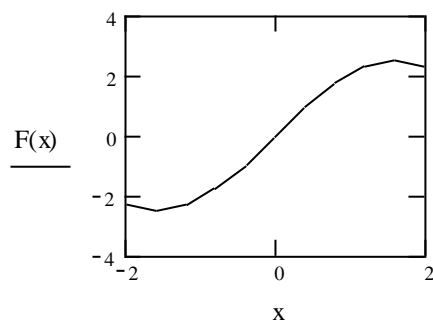
$X := 2.5, 3.0..10,$

$S1 := -2.2, -2.19..-1.$

Дискретний аргумент зручно застосовувати як аргумент функції для побудови її графіка або визначення ряду її значень, наприклад так, як показано на рис.1.3.

$a := 2.5$        $x := -2, -1.6..2$

$F(x) := a \cdot \sin(x)$



$x =$	$F(x) =$
-2	-2.273
-1.6	-2.499
-1.2	-2.33
-0.8	-1.793
-0.4	-0.974
0	$-1.11 \cdot 10^{-15}$
0.4	0.974
0.8	1.793
1.2	2.33
1.6	2.499
2	2.273

Рис.1.3. Приклади застосування дискретного аргумента

Як видно з рис.1.3, якщо задані дискретний аргумент та функція, аргументом якої є цей дискретний аргумент, можна отримати у вигляді таблиці їх значення, поставивши після їх ідентифікаторів знаки рівняння.



Іноді виникає задача обчислення суми (добутку) послідовності чисел або значень функції, тобто, наприклад, на проміжку аргумента  $x$  від 1 до 5 обчислити суму  $x^1+x^2+x^3+x^4+x^5$ . Для цього треба з палітри *Матаналіз* вибрати

кнопку виду  $\sum_{n=1}^m$  (див. рис.1.1) і задати початкове (у нижньому полі) та кінцеве (у верхньому полі) значення та формулу для обчислення суми у полі після символу підсумовування. Аналогічно задається добуток.

### Послідовність виконання роботи

1. Оформіть головик протоколу лабораторної роботи .
2. Надайте значення таким змінним :  $m=4$ ;  $n=2$ ;  $\beta=0.25$ .
3. Обчисліть числове значення виразу, номер якого збігається з номером бригади :

$$1. \frac{\sin^{m+1}(\beta)}{(n-1)\cos^{n-1}(\beta)} - \frac{m-n+2}{n-1}; 2. \left(m-2n-\frac{\beta}{10,5}\right)^2 - \sin^2 \beta;$$

$$3. \frac{1}{(n-1)\cos^{n-1}(\beta)} - \frac{1}{(n-3)\cos^{n-3}(\beta-0.1)}; 4. \frac{\beta}{n} \left[ \sin(\beta)+1 \right]^{5/3} - m^{-1/4};$$

$$5. \frac{1}{\sqrt{\beta}} \left( m^2 - \frac{1}{n} \right) + \sin \beta - \cos^3 \beta / \beta; 6. \frac{x^3}{3} - \frac{nx}{6} \sqrt{x^2 - n^2} - \frac{n^3}{6} (x + \sqrt{x^{2n} - n^2});$$

$$7. \frac{1}{(m-1)\sin^{m-1}(\beta)} + \frac{2(x-4)}{\sin^{m-2}(\beta^2)\cos(\beta)}; 8. \frac{\sqrt{e^m+1} - \cos^2(\beta)}{\sin(\beta)+1/n};$$

$$9. \frac{1}{(n-1)\sin^{m-1}(\beta)\cos^{n-1}(\beta/n)} + \frac{m+n-2}{n^3-1};$$

$$10. \frac{3\sqrt{\beta+n^2} - (m+4)\frac{3}{\beta} (n+4)^{1/2} 2n}{(n+2n)^3};$$

При формуванні заданого виразу зверніть увагу на розбіжність місця розташування показника степеня у рукописній та Mathcad-формі запису.

4. Зробіть змінну  $\beta$  ранжованою.

5. Задайте функцію з аргументом  $\beta$ , надайте їй вираз, заданий у п.3, та обчисліть її значення .
6. Надайте трьом різним змінним значення цілого, комплексного числа, десяткового дробу з порядком та перетворіть їх на рядкові змінні.
7. Об'єднайте всі отримані рядкові змінні в одну та визначте кількість символів, з яких вона складається.
8. Задайте функцію одного аргумента у вигляді полінома  $p$ 'ятого порядку з дійсними значеннями коефіцієнтів.
9. Задайте комплексне значення аргумента.
10. Обчисліть дійсну та уявну частини значення полінома та модуль його значення.
11. Визначте значення поліному 5-го степеня за схемою Хорнера за умови, що аргумент дійсний.
12. Визначте значення полінома 5-го степіня за схемою Хорнера за умови, що аргумент комплексний.
13. Обрахуйте суму результатів виразу з п.3 за умови, що змінюється тільки одна змінна(на власний розсуд), а інші є константами.
14. Обрахуйте добуток результатів виразу з п.3 за умови що змінюється тільки одна змінна(на власний розсуд), а інші є константами.

### **Контрольні запитання**

1. Який вигляд матиме змінна, значенням якої є двійкове число?
2. Як при формуванні математичного виразу вийти з режиму введення показника степеня, підкореневого виразу?
3. Як задати дискретний аргумент, значення якого матимуть від'ємний крок?
4. Що відбуватиметься з документом Mathcad з вимкненим режимом автоматичних обчислень?
5. Як, задавши функцію, отримати таблицю її значень, якщо вона має :
  - один аргумент;
  - два аргумента?

## ЗАСТОСУВАННЯ ПРОГРАМУВАННЯ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНИХ ЗАДАЧ

**Мета роботи** – навчитися використовувати можливості програмування *MathCad* для розв'язання поставлених задач.

### Теоретичні відомості

Для вставлення програмного коду в документи в *Mathcad* є спеціальна панель інструментів ***Programming*** (Програмування), яку можна викликати на екран натисненням кнопки *Programming Toolbar* на панелі ***Math*** (Математика), як показано на рис. 2.1. Більшість кнопок цієї панелі має назву операторів програмування.

Викладемо послідовно основні складові елементи мови програмування *Mathcad* і розглянемо приклади її використання.

Програма в *Mathcad* має блокову структуру з підпорядкованими блоками,

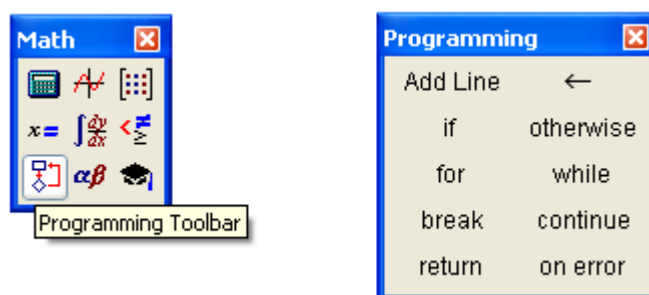


Рис.2.1 . Панель інструментів *Programming*

що виділені вертикальними лініями. Щоб створити програмний модуль, наприклад, наведений на рис. 2.3, треба виконати наступні дії.

1) Введіть частину виразу, що знаходиться зліва від оператора надання значення і сам оператор надання значення. У нашому прикладі це функція  $f(x)$ .

2) При необхідності викличте на екран панель інструментів ***Programming*** (Програмування) (див. рис. 2.1).

3) Натисніть на цій панелі кнопку ***Add Line*** (Добавить линию).

4) Якщо наперед відомо, скільки рядків коду міститиме програма, можна створити потрібну їх кількість повторним натисненням кнопки *Add Line* (Добавить линию) відповідну кількість разів (на рис. 2.2 показаний результат триразового натиснення).

5) У полях введення, що з'явилися, введіть бажаний програмний код, використовуючи програмні оператори (див рис. 2.3, 2.4).

Після того, як програмний модуль повністю визначений і жодне поле введення не залишиться порожнім, функція  $f(x)$  може використовуватися звичним способом.

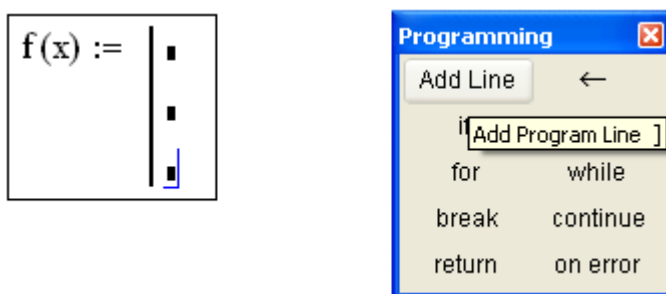


Рис. 2.2. Початок створення програмного модуля

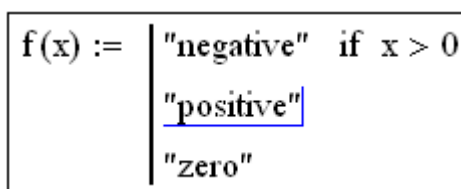


Рис. 2.3. Вставлення програмного оператора

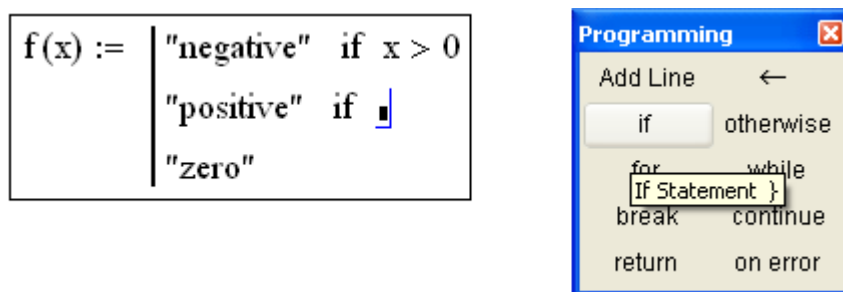


Рис. 2.4. Встлення умови в програму

Вставити рядок програмного коду у вже створену програму можна у будь-який момент за допомогою тієї ж кнопки *Add Line* (Додати лінію). Для цього

слід заздалегідь помістити на потрібне місце всередині програмного модуля курсор введення. Наприклад, розташування курсора введення на рядку, показаному на рис. 2.5, приведе до появи нової лінії з полем введення перед цим рядком. Якщо пересунути курсор введення з початку рядка (як на рис. 2.5) в його кінець, то нова лінія з'явиться після рядка.

```
f(x) := | "negative" if x < 0
        | "positive" if x > 0
        | "zero" otherwise
```

Рис. 2.5. Вставка нового рядка в існуючу програму

Якщо виділити рядок не цілком, а лише деяку його частину (рис. 2.6), то це вплине на місце розташування в програмі нового рядка (результат натиснення кнопки *Add Line* показаний на рис. 2.7).

```
f(x) := | "negative" if x < 0
        | "positive" if x > 0
        | "zero" otherwise
```

Рис. 2.6. Положення курсора введення впливає на розташування нового рядка

```
f(x) := | "negative" if x < 0
        | if x > 0
          | "positive"
        | "zero" otherwise
```

Рис. 2.7. Результат вставлення нової лінії в програму (з положення рис. 2.6)

Результати утворення нових рядків наведені на рис. 2.8.

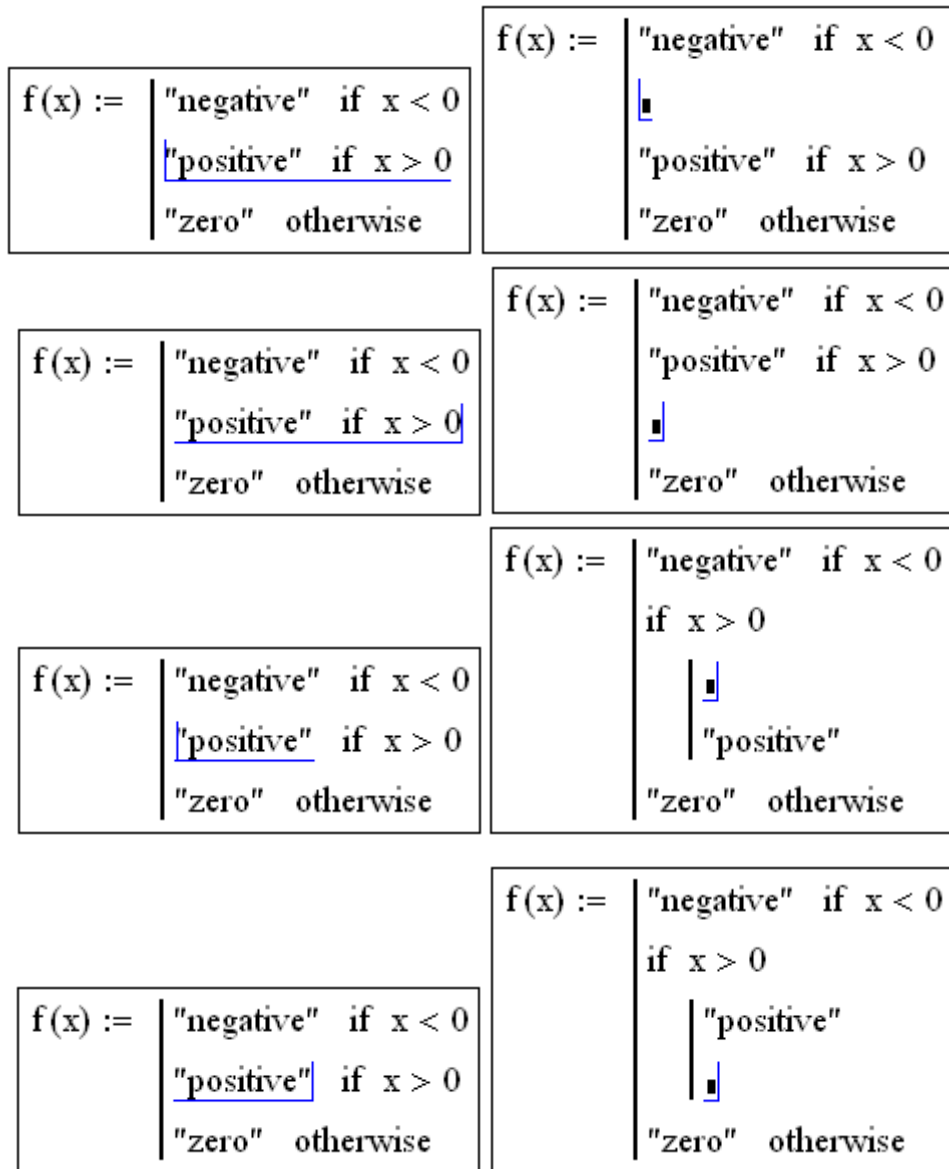


Рис. 2.8. Результат вставлення нового рядка в програму з різних положень курсора (праві фрагменти відтворюють результат операції)

Нова вертикальна межа з двома лініями виділяє фрагмент програми, який належить до умови  $x > 0$ , що знаходиться в його заголовку. У режимі виконання програми, а це відбувається при будь-якій спробі обчислити  $f(x)$ , виконується послідовно кожен рядок коду.

Надання значення в межах програм проводиться за допомогою оператора **Local Definition** (Локальное присвоение значения), який вставляється натисненням кнопки із зображенням стрілки  $\leftarrow$  на панелі **Programming** (Програмирование) (див. рис. 2.9).

$$f(x) := \begin{cases} z \leftarrow 4 \\ z + x \end{cases}$$

$$f(1) = 5$$

Рис. 2.9. Локальне надання значення в програмі

Дія умовного оператора *if* складається з двох частин. Спочатку перевіряється логічний вираз (умова) праворуч від нього. Якщо він є істиною, то виконується вираз зліва від оператора *if*. Якщо він помилковий – нічого не відбувається, а виконання програми продовжується переходом до її наступного рядка.

У мові програмування Mathcad є два оператори циклу: *for* і *while*. Перший з них дає можливість організувати цикл за деякою змінною, що приймає ряд значень із заданого діапазону зі сталим кроком. Другий створює цикл, вихід з якого здійснюється за заданою логічною умовою. Щоб вставити в програмний модуль оператор циклу, потрібно виконати дії показані на рис. 2.10 та 2.11.

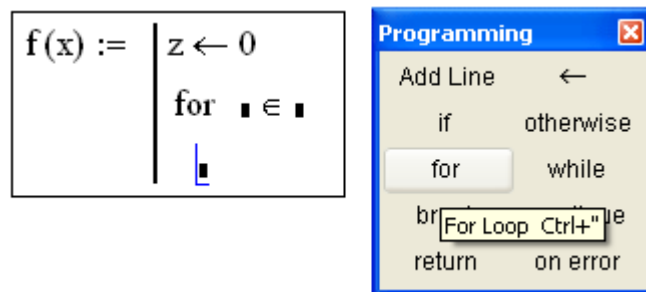


Рис. 12.10. Вставка оператора циклу

$x := \begin{cases} z \leftarrow 0 \\ \text{for } i \in 0..5 \\ z \leftarrow z + i \end{cases}$ <p><math>x = 15</math></p>	$x := \begin{cases} z \leftarrow 0 \\ \text{while } z < 10 \\ z \leftarrow z + 1 \end{cases}$ <p><math>x = 10</math></p>
----------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------

Рис.2.11. Робота оператор циклу *for* з ранжованою змінною та *while*

## Послідовність виконання роботи

1. Побудуйте графік функції:

1)  $y=x-\sin(x)$ ;

2)  $y=x+1/x$  (врахуйте наявність розриву функції);

3)  $y=x\sin(x)$ ;

4)  $y=x\operatorname{tg}(x)$  (врахуйте наявність розриву функції);

5)  $y=x e^{-x}$ ;

6)  $y=e^x/x$  (врахуйте наявність розриву функції);

7)  $y=e^{-x}\cos(x)$ ;

8)  $y=e^{-x}/x$  (врахуйте наявність розриву функції);

9)  $y=e^x\sin(x)$ ;

10)  $y=\cos(x)/x$  (врахуйте наявність розриву функції).

2. Створіть програмний блок, результатом виконання якого є вектор (матриця) ординат графіка функції з п.1 за умови, що:

1) частина графіка з III-го квадранту дзеркально відобразиться у II-ий квадрант;

2) частина графіка з III-го квадранту дзеркально відобразиться у II-ий квадрант;

3) від'ємна частина графіка дзеркально відобразиться відносно осі абсцис;

4) від'ємна частина графіка дзеркально відобразиться відносно осі абсцис;

5) частина графіка з III-го квадранту відобразиться у II-ий, а з I-го у IV-ий;

6) частина графіка з III-го квадранту відобразиться у II-ий, а з I-го у четвертий;

7) від'ємна частина графіка дзеркально відобразиться відносно осі абсцис;

8) частина графіка з I-го квадранту відобразиться у II-ий, а з



III-го у IV-ий;

9) від'ємна частина графіка дзеркально відобразиться відносно осі абсцис;

11) частина графіка з I-го квадранту відобразиться у IV-ий.

3. Побудуйте графік цієї функції.

### Контрольні запитання

1. Поясніть призначення кнопки *Add Line*.
2. Які цикли є у мові програмування Mathcad, чим вони відрізняються від вам відомих?
3. Який формат має умовний оператор у програмному блоці Mathcad ?  
Наведіть приклад.
4. Що означає локальне надання значення?

## СИМВОЛЬНІ ОБЧИСЛЕННЯ

**Мета роботи** – навчитися виконувати символльні операції над математичними виразами та дослідити особливості їх застосування.

### Теоретичні відомості

Розв'язання великої кількості математичних та інженерних задач потребує виконання різного виду аналітичних перетворень, які у ряді випадків значно спрощують подальші викладки.

Перші версії Mathcad будувалися для виконання лише числових розрахунків. Лише, починаючи з версії *MathCad 3.0 for Windows*, до її складу було введено ядро символних (аналітичних) операцій пакету Maple, яке значно розширило можливості *MathCad*.

Виконання символних (аналітичних) операцій реалізується через команду ***Symbolics*** (Символи) головного меню в командному режимі або за допомогою спеціального оператора символного виведення результату у вигляді подовженої горизонтальної стрілки ( $\rightarrow$ ), яку можна ввести комбінацією клавіш „*Ctrl*” + „*.*” або через панель математичних знаків (кнопка із зображенням магістерського капелюха), обравши у палітрі символних операцій, що відкриється, кнопку із зображенням вказаної стрілки. Обидва способи використання символних операцій ілюструє рис.3.1.

При застосуванні палітри символних операцій результат перетворення виводиться праворуч від стрілки (приклад 3), а результат аналітичних перетворень, що виконуються у командному режимі, може розташовуватися праворуч, нижче (приклади 1, 2 на рис.3.1) та замість початкового виразу. Змінити місце розташування результату можна командою ***Symbolics*** (Символи)  $\Rightarrow$  ***Evaluate Style*** (Стиль вичислений).

Існує суттєве обмеження при виконанні аналітичних перетворень в командному режимі: у виразах, що підлягають перетворенню, не повинні міститися функції користувача. При застосуванні оператора символного

виведення результату (через палітру символічних перетворень) такого обмеження не існує.

Команда *Symbolics* головного меню містить список команд перетворень, які можна поділити на дві групи за правилами їх застосування.

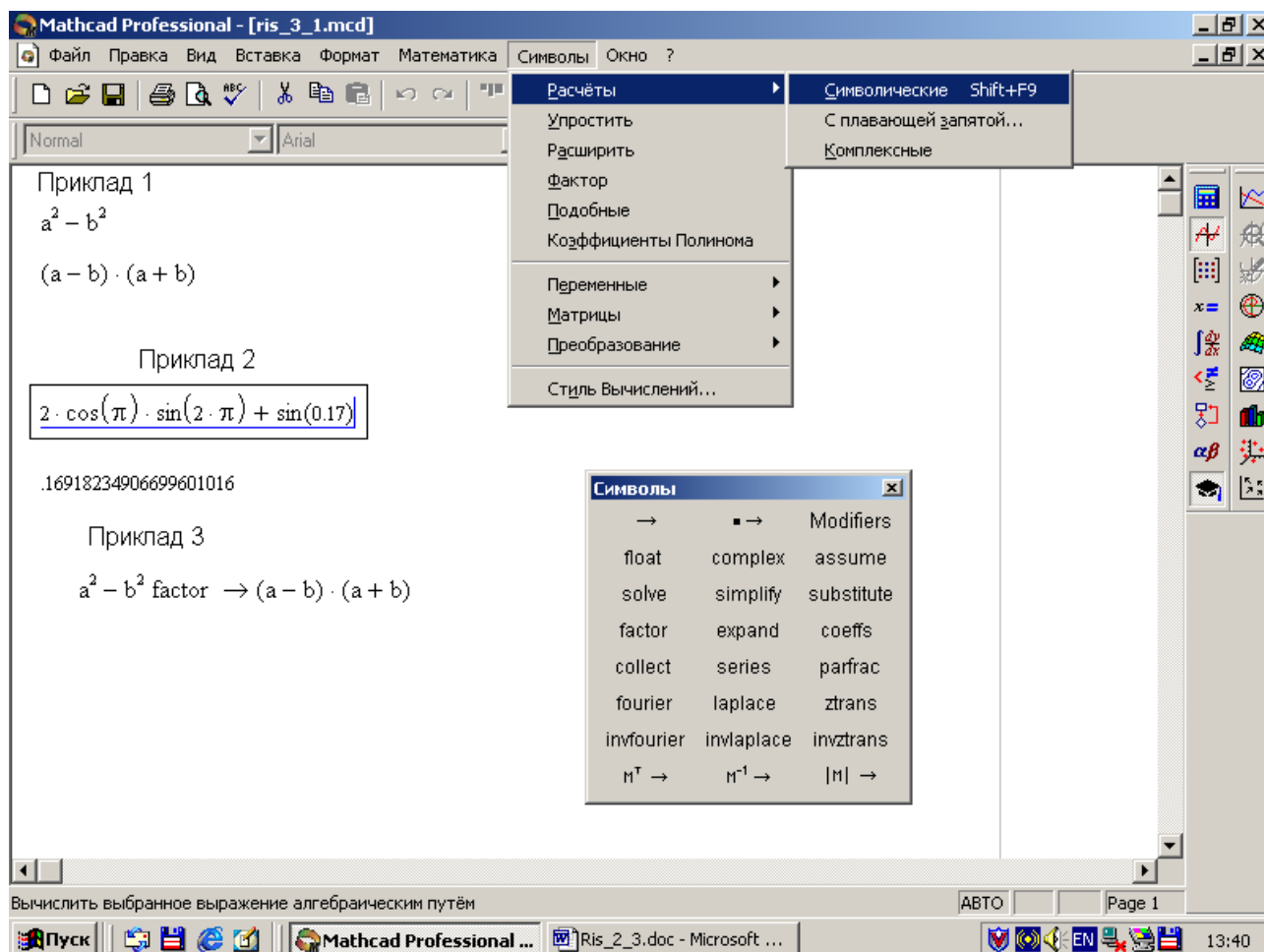


Рис.3.1. Приклади виконання символічних перетворень різними способами

До першої групи належать такі команди:

- **Evaluate** (Расчеты) – перетворення виразу з вибором виду перетворення (приклад 2 на рис.3.1);
- **Simplify** (Упростить) – спрощення виразу зі зведенням подібних, зведенням до спільного знаменника, із застосуванням основних тригонометричних тотожностей і т.і.(рис.3.2, приклад 1);
- **Expand** (Расширить) – розкладання за степенями (приклад 2 на рис.3.2);

- **Factor** (Фактор) – розкладання числа або виразу на множники; протилежна Expand команда (приклад 1 на рис.3.1);
- **Collect** (Подобные) – розкладання виразу за підвиразом (приклади 3,4 на рис.3.2);
- **Polynomial Coefficients** (Коэффициенты полинома) – визначення коефіцієнтів поліноміального виразу (приклади 5,6 на рис.3.2).

Ці команди можна виконати, попередньо виділивши курсором вираз або його частину, що підлягає перетворенню. Винятком є команда пошуку подібних та визначення коефіцієнтів полінома. Перед їх виконанням треба виділити не весь вираз, а ту його частину, що є аргументом.

У прикладах 3 та 4 (рис.3.2) при зведенні подібних отримані різні результати, оскільки виділеним підвиразом у прикладі 3 був  $x$ , а у прикладі 4 –  $d$ .

Приклад 1:спростити

$$\frac{a^2 - 2 \cdot a \cdot b + b^2}{a - b}$$

$$a - b$$

Приклад 4:подібні

$$a + b \cdot x^2 + 8 \cdot b \cdot x + 16 \cdot b - d \cdot x^2 - d \cdot x + c \cdot x^2 + c \cdot x$$

$$\left( -x^2 - x \right) \cdot d + a + b \cdot x^2 + 8 \cdot b \cdot x + 16 \cdot b + c \cdot x^2 + c \cdot x$$

Приклад 2:розширити

$$a + b \cdot (x + 4)^2 - d \cdot \left( x^2 + x \right) + c \cdot x \cdot (x + 1)$$

Приклад 5:коефіцієнти полінома

$$a \cdot x^2 + b \cdot x$$

$$a + b \cdot x^2 + 8 \cdot b \cdot x + 16 \cdot b - d \cdot x^2 - d \cdot x + c \cdot x^2 + c \cdot x$$

$$\begin{pmatrix} 0 \\ b \\ a \end{pmatrix}$$

Приклад 3:подібні

$$a + b \cdot x^2 + 8 \cdot b \cdot x + 16 \cdot b - d \cdot x^2 - d \cdot x + c \cdot x^2 + c \cdot x$$

Приклад 6:коефіцієнти полінома

$$a \cdot x^2 + (b + a) \cdot x$$

$$(-d + b + c) \cdot x^2 + (-d + 8 \cdot b + c) \cdot x + a + 16 \cdot b$$

$$\begin{pmatrix} a \cdot x^2 \\ x \end{pmatrix}$$

Рис.3.2. Приклади виконання операцій спрощення, зведення подібних та визначення коефіцієнтів полінома

У прикладах 5, 6 (рис.3.2) були виділені відповідно аргументи  $x$  та  $a+b$ . Як видно з рисунка, результатом команди визначення коефіцієнтів полінома є

вектор-стовпець коефіцієнтів, першим елементом якого є вільний коефіцієнт, останнім – старший.

До другої групи команд належать команди, що вимагають виділення змінної у виразі. Це команди *Variable* (Переменные), *Matrix* (Матрицы) та *Transform* (Преобразование). Кожна з них має своє меню.

Меню команди *Variable* містить такі пункти:

- *Solve* (Вычислить) – пошук розв'язку рівняння чи нерівності (приклади 1,2 на рис.3.3);
- *Substitute* (Замена) – виконання заміни змінної у виразі заданим підвиразом (приклад 3 на рис.3.3);
- *Differentiate* (Дифференциал) – диференціювання виразу за виділеною змінною;
- *Integrate* (Интегралы) – інтегрування виразу за виділеною змінною;
- *Expand to Series* (Разложить на составляющие) – визначення членів розкладання виразу в ряд Тейлора відносно виділеної змінної;
- *Convert to Partial Fraction* (Преобразование в Частичные Доли) – розкладання дробово-раціонального виразу на елементарні дроби (приклад 4 на рис.3.3).

Результатом виконання команди *Solve* є вектор-стовпець з коренями виразу з виділеною змінною (приклад 1 на рис.3.3). Для розв'язання можна задати рівняння у загальноприйнятому вигляді (приклад 5 на рис.3.3), вказавши знак рівняння з палітри логічних операторів, а не у вигляді виразу.

При підвищенні порядку рівняння вирази з його коренями у символному вигляді можуть мати громіздкий вигляд. У такому випадку результат у текстовому вигляді розташовується в буфері обміну, звідки користувач може вивести його в робоче вікно для аналізу.

Приклад 3 на рис.3.3 ілюструє виконання команди заміни *Substitute*, яка вимагає задання виразу зі змінною, що замінюватиметься, виразу для заміни (у прикладі  $(\omega+1)/(\omega-1)$ ) з його подальшим вирізанням чи копіюванням до

буферу обміну та виділення у першому виразі змінної (у прикладі  $x$ ), що зазнає змін.

Приклад 1

$$a \cdot x^2 + b \cdot x$$

$$\begin{pmatrix} 0 \\ -b \\ a \end{pmatrix}$$

Приклад 2

$$2 \cdot x \geq 3$$

$$\frac{3}{2} \leq x$$

Приклад 3

$$b \cdot x^2 + c \cdot x + d$$

$$\frac{\omega + 1}{\omega - 1}$$

$$b \cdot \frac{(\omega + 1)^2}{(\omega - 1)^2} + c \cdot \frac{(\omega + 1)}{(\omega - 1)} + d$$

Приклад 4

$$\frac{8 \cdot x + 1}{6 \cdot x^2 + 5 \cdot x + 1}$$

$$\frac{6}{(2 \cdot x + 1)} - \frac{5}{(3 \cdot x + 1)}$$

Приклад 5

$$a \cdot x^2 + b \cdot x = 0$$

$$\begin{pmatrix} 0 \\ -b \\ a \end{pmatrix}$$

Рис.3.3. Приклади розв'язання рівнянь і нерівностей, виконання заміни змінної, розкладання на елементарні дроби і визначення коефіцієнтів полінома

Меню команди **Matrix** містить такі пункти:

- **Transpose** (Транспонировать) – транспонування виділеної матриці або її виділеної частини;
- **Invert** (Инвертирование) – обернення квадратної матриці або її виділеної частини;
- **Determinant** (Определитель) – визначення детермінанту матриці.

Меню команди **Transform** містить команди для прямого та зворотного перетворень Фур'є, Лапласа та  $z$ -перетворення.

**MathCad** надає можливість визначення похідних заданого порядку та інтегралу від функції.

Похідна береться за змінною, яка записується у нижньому полі, а у верхнє заноситься функція з аргументом, за яким відбувається



$$7) 5 \cos(3\pi/2) + \frac{\sin(2\pi)}{e^{0.6\pi}} + \cos^2(0.2);$$

$$8) 2 \sin\left(\frac{a+b}{2}\right) \sin\left(\frac{b-a}{2}\right);$$

$$9) e^{-0.25\pi} \sin \pi + \cos 1.2;$$

$$10) \cos(a) \cos(b) + \sin(a) \sin(b).$$

2. У командному режимі обчисліть вираз:

$$1) \sqrt{-4}/8; \quad 2) (4-i)^2/3; \quad 3) 6/\sqrt{-2}; \quad 4) 1/\sqrt{-7}; \quad 5) 85/\sqrt{-1};$$

$$6) i^3; \quad 7) (2+1i)^2; \quad 8) 8/(2+3)^2; \quad 9) 0.2/\sqrt{-5}; \quad 10) (5+2i)^4$$

в режимі комплексних обчислень і без нього.

3. У командному режимі спростіть вираз:

$$1) \frac{\sin^2(x) + \cos^2(x)}{2e^{-x}};$$

$$6) \left(1 + \frac{2}{3x-1}\right) \left(1 - \frac{9x-9x^2}{3x+1}\right) + 1;$$

$$2) \frac{\sqrt{\sin(x)}}{\sin^2(x) + 2\sin(x) + 1};$$

$$7) \frac{9x^2 - 4}{2(x+1) + x \cos^2(x) + \sin^2(x)};$$

$$3) \frac{25}{x+1} + \frac{30}{x-1} - \frac{0.15}{(x+2)^2};$$

$$8) \frac{(x+1)(x-1) - x^2}{\sin\left(\frac{\pi}{2}\right)};$$

$$4) \frac{a^2 - x^2}{a+x} - a + x;$$

$$9) \frac{x^2 + 2x + 1}{(x+1)^2} - \frac{\sin(\pi)}{x};$$

$$5) \frac{\sin^2(x) + \cos^2(x)}{1-x^2} - \frac{x}{x-x^3};$$

$$10) \frac{2e^{-x}}{\sin^2(x) + \cos^2(x)}.$$

4. У командному режимі розкладіть за степенями аргумента  $x$  вираз:

$$(-a^3 + 2a^2x - x^3)(4a^2 + 8ax) + (a^3x^2 + 2a^2x^3 - 4ax^4) - (a^5 + 4a^3x^4 - 3a^2x^3 - 4ax^4).$$

5. У командному режимі визначте коефіцієнти полінома, отриманого у п.4.

6. За допомогою палітри символічних перетворень виконайте п.п. 1,3,4,5.

7. Задайте поліном з числовими значеннями коефіцієнтів та через палітру символічних перетворень знайдіть його корені, задавши кількість позицій дробової частини числа результату.



8. Розв'яжіть рівняння 3, 4, 5-го порядку з символічними та числовими коефіцієнтами.
9. Розкладіть довільний дробово-раціональний вираз з числовими коефіцієнтами на елементарні дроби.
10. Виконайте заміну змінної на підвираз у заданому викладачем виразі.
11. Обрахуйте диференціал функції з п.3 .
12. Обрахуйте визначений та невизначаний інтеграл функції з п.3 (для визначеного інтегралу - у межах від 3 до 8).

### Контрольні запитання

1. Яким може бути результат символічних перетворень?
2. Чи може бути результатом символічних перетворень текстовий вираз?
3. Розкладіть заданий викладачем дробово-раціональний вираз на елементарні дроби.
4. Які з наведених виразів є коректними для визначення коефіцієнтів полінома:

$$A(p) := p^2 + (p + 4)^3 + 3;$$

$$P^2 + (p + 4)^3 + 3;$$

$$P^2 + (p + 4)^3 + 3 = 0;$$

$$2p^6 + 3p^3 ?$$

5. У якому вигляді треба задати рівняння, щоб отримати його розв'язок? Наведіть приклади.

## Лабораторна робота №4

### ДВОВИМІРНА ГРАФІКА

**Мета роботи** – навчитися будувати графіки функцій однієї та декількох змінних, дослідити можливості їх форматування.

#### Теоретичні відомості

Графічний процесор *MathCad* забезпечує можливість користування дво- та тривимірною графікою за допомогою відповідних шаблонів. Відкрити список шаблонів можна командою **Insert => Graph** (Вставка =>График) або кнопкою із зображенням двовимірної графіки на панелі палітр математичних знаків (рис.4.1).

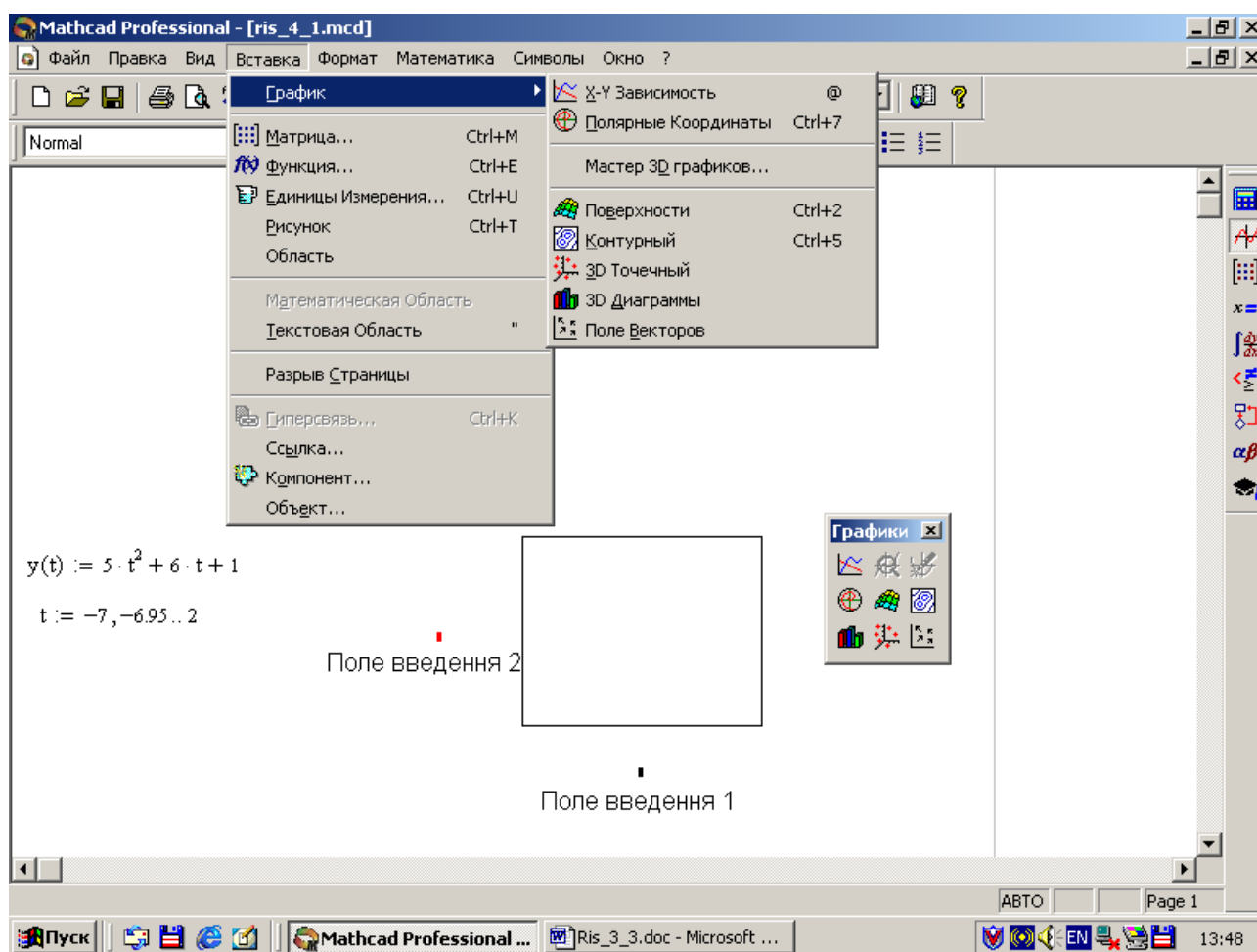


Рис.4.1. Побудова графіка функції однієї змінної у декартових координатах

Шаблон для побудови графіка в декартових координатах містить поле для побудови графіка та два поля введення. У полі введення 1 (рис.4.1) треба вказати аргумент функції, графік якої будуватиметься, у полі введення 2— ідентифікатор функції зі своїм аргументом.

Якщо шаблон повинен містити декілька графіків, ідентифікатори усіх функцій зі своїми аргументами вказують списком у полі введення 2 через кому. Якщо ці функції мають різні аргументи, то їх список вказують у полі введення 1 (рис.4.2). Поля введення 1 та 2 можуть містити ідентифікатори змінної чи функції, вираз (наприклад,  $nT$ ) або індексовану змінну (наприклад,  $ya_n$  на рис. 4.2.).

```

T := 0.4          m := 100  n := 0..m

p(t) := 166.67 * exp(-2 * t) - .4 * exp(-1 * t)
Y(t) := 8 + 42.67 * exp(-.5 * t) - 111.1 * exp(-.4 * t) + 227.56 * exp(-.25 * t) - p(t)
Y1(t) := Y(t) + (rnd(1.4) - 0.7)
ya_n := 10.172 - 11.818 * exp[-( 5.032 * 10^-2 ) * n * T]
t := 0..40

```

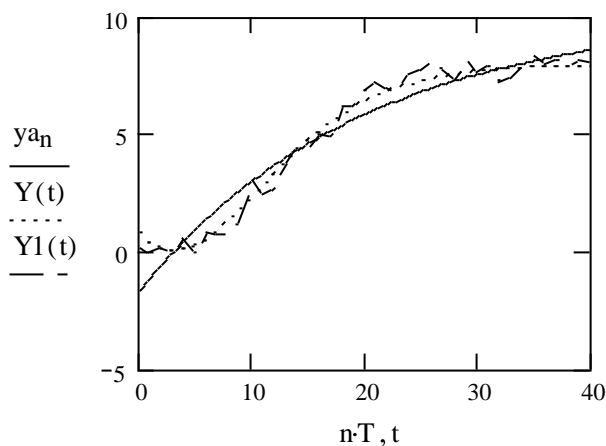


Рис.4.2. Способи побудови серій графіків

Якщо користувач не вказав діапазон і крок зміни аргумента, за мовчазною угодою він вважається заданим дискретним аргументом виду :  $-10..10$ .

Графік утворюється сукупністю точок, кількість яких дорівнює кількості заданих значень аргументу, що з'єднуються між собою лініями різного типу

(рис.4.2). Зменшення кроку зміни аргумента робить графік плавнішим. До того ж, зменшення кроку аргумента може виявити на графіку особливості функції.

Якщо *MathCad* працює в автоматичному режимі обчислень, обидва поля введення коректно заповнені, то в шаблоні побудується графік після виведення курсора миші за межі графічної області та натискання на її ліву кнопку.

Для побудови годографів у декартових координатах застосовують функції, задані параметрично, а саме рівняннями виду

$$x = \varphi(\omega); \quad y = \psi(\omega),$$

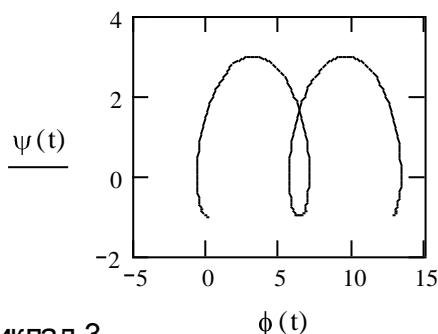
де  $\varphi(\omega)$ ,  $\psi(\omega)$  є неперервними при  $\omega \in (\alpha, \beta)$ . Ці рівняння, що описують залежність декартових координат  $(x, y)$  точки площини від значення параметра  $\omega \in (\alpha, \beta)$ , визначають на площині криву, задану в параметричній формі (задану параметрично).

Приклади зображення таких кривих наведені на рис.4.3.

#### Приклад 1

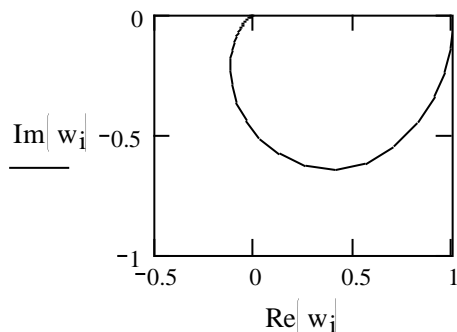
$$t := 0, 0.1 .. 4 \cdot \pi \quad \lambda := 2$$

$$\phi(t) := t - \lambda \cdot \sin(t) \quad \psi(t) := 1 - \lambda \cdot \cos(t)$$



#### Приклад 3

$$i := 1 .. 100 \quad w_0 := 0 \quad w_i := w_{i-1} + 0.005 \cdot i$$



#### Приклад 2

$$\omega := 0, 0.1 .. 2 \cdot \pi$$

$$\operatorname{Re}(\omega) := \frac{1 - 6 \cdot \omega^2}{1 + 13 \cdot \omega^2 + 36 \cdot \omega^4}$$

$$\operatorname{Im}(\omega) := \frac{-5 \cdot \omega}{1 + 13 \cdot \omega^2 + 36 \cdot \omega^4}$$

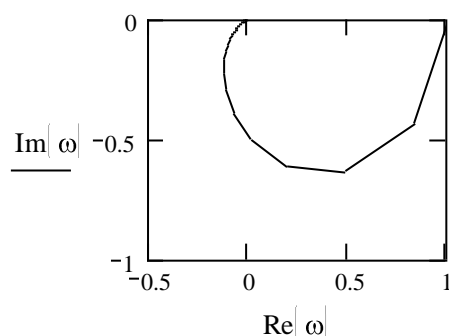


Рис.4.3. Приклади зображення кривих, заданих в параметричній формі

Якщо зображення годографа будується у вигляді ламаної лінії (рис.4.3, приклад 2), можна зменшити крок зміни параметра (у прикладі -  $\omega$ ), заданого дискретним аргументом, або задати параметр як індексовану змінну (рис.4.3, приклад 3), що змінюється з кроком, який, наприклад, зростає. Так зручно задавати параметр при побудові годографа, якщо цей параметр ( $\omega$  у прикладі 3 на рис.4.3) змінюється нерівномірно.

При побудові годографів для подальшого їх аналізу важливо, щоб масштаби вздовж обох осей були однаковими.

*MathCad* пропонує широкі можливості форматування графіків. До команд форматування можна звернутися, виділивши попередньо графічну область і натиснувши праву клавішу миші або через команди **Format** => **Graph** (Формат => Графік). Розгорнуте після цього вікно форматування міститиме вкладки форматування осей, ліній графіка, відображення написів поблизу осей та вкладку для роботи з параметрами, що встановлюються автоматично (рис.4.4).

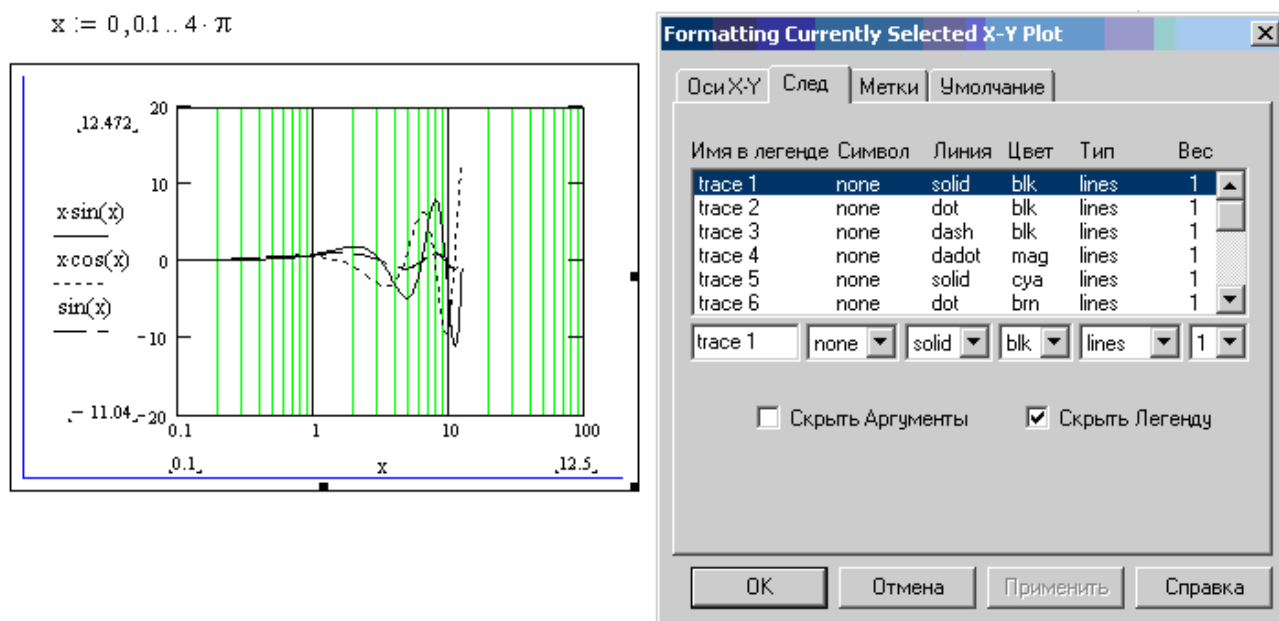


Рис.4.4. Ілюстрація можливостей форматування декартового графіка

### Послідовність виконання роботи

1. Задайте поліноми першого, другого, третього та п'ятого степеня з однаковими (кратними) коренями (відомими наперед).

2. Побудуйте в одних координатних осях графіки цих поліномів для діапазону їх аргумента, що містив би їх корені.
3. Виконайте форматування цих графіків.
4. Введіть параметрично задану функцію та побудуйте її годограф:

1	Завиток Паскаля	$x = a \cdot \cos^2(\omega) + I \cdot \cos(\omega);$ $y = a \cdot \sin(\omega) \cdot \cos(\omega) + I \cdot \sin(\omega);$ $I = 10; \quad a = 50; \quad \omega = 0..2\pi$
2	Епіциклоїду	$x = (A + a) \cdot \cos(\omega) - a \cdot \cos\left(\frac{A + a}{a} \omega\right);$ $y = (A + a) \cdot \sin(\omega) - a \cdot \sin\left(\frac{A + a}{a} \omega\right);$ $A = 50; \quad a = 10; \quad \omega = 0..2\pi$
3	Гіпоциклоїду	$x = (A + a) \cdot \cos(\omega) - \lambda \cdot a \cdot \cos\left(\frac{A + a}{a} \omega\right);$ $y = (A + a) \cdot \sin(\omega) - \lambda \cdot a \cdot \sin\left(\frac{A + a}{a} \omega\right);$ $A = 50; \quad a = 10; \quad \lambda = 15; \quad \omega = 0..2\pi$
4	Кардіоїду	$x = a \cdot \cos(\varphi) (1 + \cos(\varphi)) ;$ $y = a \cdot \sin(\varphi) (1 + \cos(\varphi)) ;$ $a = 10; \quad \varphi = 0..2\pi$
5	Криву Штейнера	$x = 2 \cdot a \cdot \cos\left(\frac{\omega}{3}\right) + a \cdot \cos\left(\frac{2\omega}{3}\right);$ $y = 2 \cdot a \cdot \sin\left(\frac{\omega}{3}\right) - a \cdot \sin\left(\frac{2\omega}{3}\right);$ $a = 10; \quad \omega = 0..2\pi$
6	Декартів лист	$x = 3a\omega / (1 + \omega^3); \quad y = 3a\omega^2 / (1 + \omega^3);$ $\omega = -1..100; \quad a = 2$
7	Циссоїду	$x = \frac{a}{\omega^2 + 1}; \quad y = \frac{a}{\omega \omega^2 + 1};$ $a = 5; \quad \omega = -100..100;$

8	Строфоїду	$x = a(\omega^2 - 1)/(\omega^2 + 1);$ $y = a\omega(\omega^2 - 1)/(\omega^2 + 1);$ $a = 0.5; \quad \omega = -100..100$
9	Звичайну циклоїду	$x = a(\omega - \sin(\omega)); \quad y = a(1 - \cos(\omega));$ $a = 10; \quad \omega = 0..30$
10	Астроїда	$x = a \cos^3(t); \quad y = a \sin^3(\omega);$ $a = 10; \quad t = -20..20$

- Оформіть графік написами і коментарями, створеними як частина графічної області і як текстовий блок.
- Перегляньте у збільшеному масштабі фрагмент графіка, зафіксуйте його в документі. Виконайте трасування графіка.
- Побудуйте серію годографів параметрично заданих функцій з різними співвідношеннями коефіцієнтів  $a, A, I, \lambda$ . Застосуйте для них команди зміни діапазонів на осях, зміни накреслення кривих.

### Контрольні запитання

- Як звужити на побудованому графіку діапазон аргументу?
- Як встановити однакові масштаби на обох осях декартового графіка?
- Як накласти текстову область на (під) графічну?
- Як зробити видимими (невидимими) коментарі на осях?
- Як утворити графік сукупністю значень функції у вузлах у вигляді символів обраної форми?
- Як перейти від рівномірної до логарифмічної сітки на осях координат?
- Як нанести сітку на графік?

## ТРИВИМІРНА ГРАФІКА

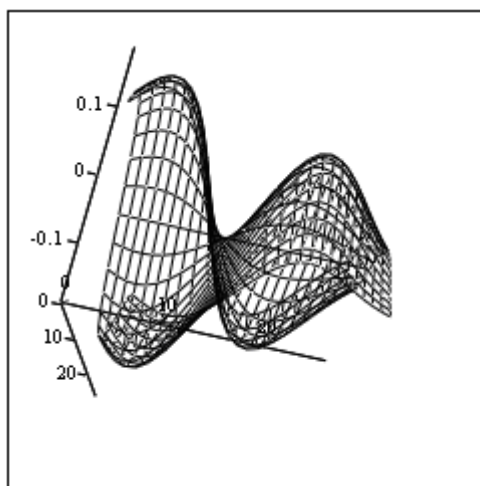
**Мета роботи** - дослідити побудову і можливості форматування поверхні функції двох змінних.

### Теоретичні відомості

*MathCad* пропонує два способи побудови поверхні функції двох змінних  $F(x, y)$ .

Перший спосіб (рис.5.1) передбачає попереднє формування матриці, елементами якої є значення функції  $F$ .

$$\begin{aligned} F(x, y) &:= \sin(x \cdot y) \cdot \exp[-|x^2 + y^2|] \\ N &:= 25 \quad s := 0..N \quad z := 0..N \\ x_s &:= -1 + 0.1 \cdot s \quad y_z := -1 + 0.1 \cdot z \\ M_{s, z} &:= F|x_s, y_z| \end{aligned}$$



М

Рис.5.1. Ілюстрація першого способу побудови поверхні функції двох змінних

Перший спосіб реалізує алгоритм, застосований на рис.5.1, а саме:

- задається функція двох змінних ( $F(x, y)$ );
- задається кількість значень кожної змінної ( $N$ );
- задаються обидві змінні як ранжовані ( $s, z$ );
- формується набір значень аргументів функцій, заданих індексованими змінними ( $x_s, y_z$ );

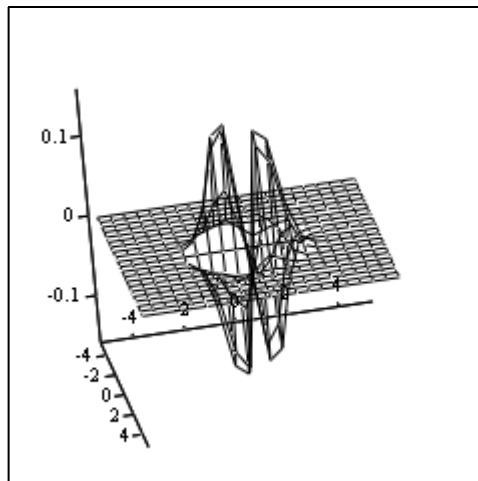


- формуються елементи матриці  $M_{s,z}$  які є значеннями функції;
- з палітри графіки обирається шаблон тривимірного графіка;
- у полі введення шаблону вводиться ідентифікатор  $M$  сформованої матриці.

Другий спосіб потребує для побудови задання лише самої функції, став можливим, починаючи з версії *MathCad 2000*, реалізує такий алгоритм (див. рис.5.2):

- задається функція двох змінних ( $F(x, y)$ );
- обирається шаблон тривимірного графіка;
- у полі введення шаблону вводиться ідентифікатор змінної ( $F$ ).

$$F(x, y) := \sin(x \cdot y) \cdot \exp\left[-\sqrt{x^2 + y^2}\right]$$



F

Рис.5.2. Ілюстрація другого способу побудови поверхні функції двох змінних

Результатом застосування обох алгоритмів буде графік поверхні (рельєфу) функції двох змінних у вигляді прозорого дротяного каркасу.

При необхідності зображення поверхні можна повертати навколо осей, розташувавши курсор миші в межах графічної області, натиснувши її ліву кнопку і, не відпускаючи, рухаючи її в бажаному напрямку.

Якщо під час руху миші буде натиснута клавіша **Ctrl**, зображення масштабуватиметься.

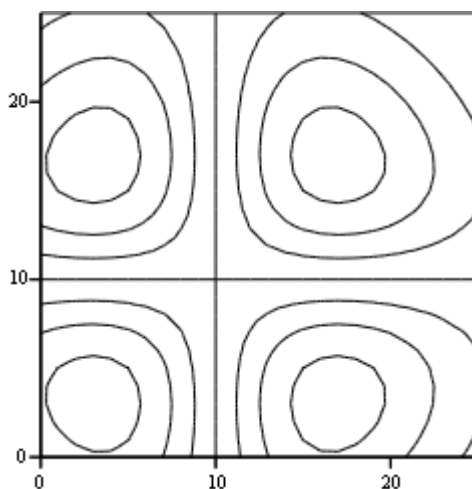
Палітра графіки пропонує можливість подання поверхні функції двох змінних у вигляді ліній однакового рівня (ізоліній). Її, як і поверхню, можна будувати одним із вищенаведених способів, а в полі ведення шаблону вказати відповідний ідентифікатор (рис.5.3).

```

F(x,y) := sin(x.y) . exp[-| x^2 + y^2 | ]
N := 25  s := 0..N    z := 0..N
x_s := -1 + 0.1 . s    y_z := -1 + 0.1 . z

M_{s,z} := F | x_s , y_z |

```



M

Рис.5.3. Поверхня функції двох змінних, побудована ізолініями

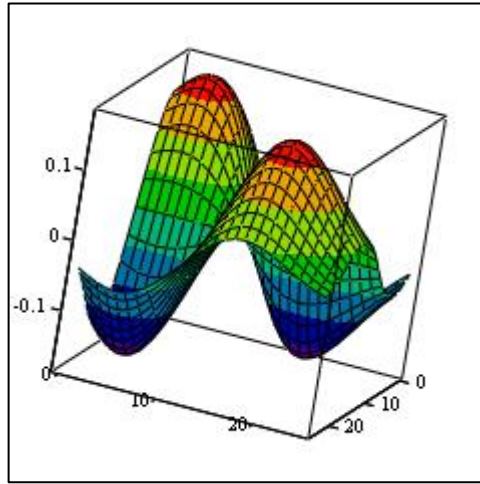
Широкі можливості форматування поверхні, побудованої першим способом, стають доступними після того, як, розташували курсор в межах графічної області, два рази натиснути на ліву кнопку миші (рис.5.4). Застосування кольорів, підсвічування робить рельєф інформативнішим (порівняйте з рис.5.2).

```

F(x,y) := sin(x.y) . exp[-| x^2 + y^2 | ]
N := 25  s := 0..N    z := 0..N
x_s := -1 + 0.1 . s    y_z := -1 + 0.1 . z

M_{s,z} := F | x_s , y_z |

```

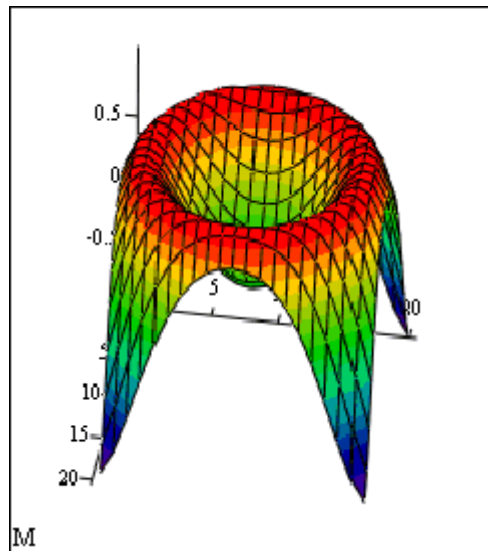


M

```

N := 20
i := 0..N      j := 0..N
xi := -1.5 + 0.15 · i    yj := -1.5 + 0.15 · j
f(x,y) := sin(x2 + y2)    Mi,j := f(xi,yj)

```



M

$$G(a,b) := \begin{bmatrix} (5 + 2 \cdot \cos(a)) \cdot \cos(b) \\ (5 + 2 \cdot \cos(a)) \cdot \sin(b) \\ 2 \cdot \sin(a) \end{bmatrix}$$

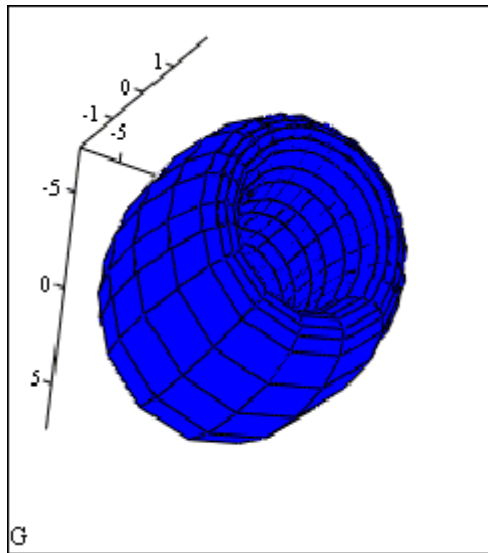


Рис.5.5. Ілюстрація можливостей форматування тривимірних зображень

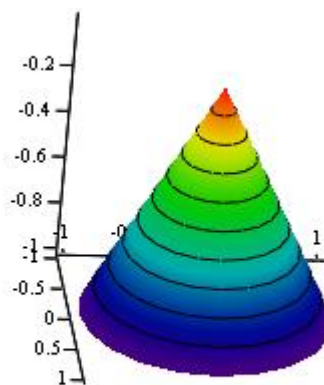
Також засобами *MathCad* існує можливість будувати фігури стереометрії, приклади побудови наведені на рис. 5.5.

$$s := 0..20 \quad n := 0..20$$

$$X_{s,n} := \frac{s \cdot \cos\left(\frac{\pi \cdot n}{10}\right)}{20}$$

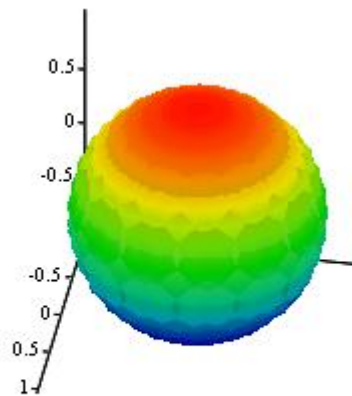
$$Y_{s,n} := \frac{s \cdot \sin\left(\frac{\pi \cdot n}{10}\right)}{20}$$

$$Z_{s,n} := \frac{-s}{20}$$



(X,Y,Z)

$$F(x,y) := \begin{pmatrix} \cos(x) \cdot \cos(y) \\ \cos(x) \cdot \sin(y) \\ \sin(x) \end{pmatrix}$$



F

Рис.5.5. Ілюстрація можливостей побудови фігур стереометрії

### Послідовність виконання роботи

1. Побудувати приклади поверхонь приведених на рис. 5.1. - 5.5
2. Побудуйте графік поверхні функції.

1	$\frac{\sin^2(x) + \cos^2(y)}{2e^{-x}}$
2	$\frac{\sqrt{\sin(x)}}{\sin^2(y) + 2\sin(x) + 1}$
3	$\frac{25}{x+1} + \frac{30}{x-1} - \frac{0.15}{(y+2)^2}$
4	$\frac{a^2 - x^2}{a+x} - a + x$
5	$\frac{\sin^2(x) + \cos^2(y)}{1-x^2} - \frac{x}{y-x^3}$
6	$\left(1 + \frac{2}{3y-1}\right) \left(1 - \frac{9x-9y^2}{3x+1}\right) + 1$

7	$\frac{9x^2 - 4}{2(y+1) + x \sqrt{\cos^2(x) + \sin^2(y)}}$
8	$\frac{(x+1)(y-1) - x^2}{\sin\left(\frac{y}{2}\right)}$
9	$\frac{x^2 + 2x + 1}{(y+1)^2} - \frac{\sin(y)}{x}$
10	$\frac{2e^{-y}}{\sin^2(y) + \cos^2(x)}$

3. Змініть масштаб та орієнтацію в просторі отриманої поверхні.
4. Дослідіть вплив зміни параметрів форматування на вигляд поверхні функції двох змінних.

### Контрольні запитання

1. Які існують особливості форматування поверхонь, побудованих першим та другим способами?
2. Як, використовуючи команди форматування, перебудувати поверхню функції двох змінних у її зображення лініями однакового рівня?
3. Як нанести значення рівнів на ізолінії?
4. Як за графічним зображенням функції двох змінних визначити її нулі?
5. Як з матриці, яку Mathcad використовує для побудови поверхні, отримати сукупність значень функції, що відображувала б вплив на неї лише однієї змінної?

## ФОРМУВАННЯ РУХОМИХ ГРАФІЧНИХ ЗОБРАЖЕНЬ

**Мета роботи** - дослідження можливостей *MathCad* зі створення рухомих зображень.

### Теоретичні відомості

При розв'язанні задач, пов'язаних з аналізом впливу параметрів на поведінку функції, зручно подавати її рухоме (анімаційне) зображення, а не серію графіків, що розкривають вплив на її вигляд кожного параметра. Особливістю анімації у *MathCad* є те, що цей процес відбувається не в самому документі, а в окремому, самостійному, відеофайлі, який можна проглянути будь-яким відеопрогравачем (наприклад, вбудованим в *Windows* програвачем *Windows Media Player*).

Розглянемо етапи створення анімації в математичному пакеті на прикладі обчислення простого виразу, наприклад,  $87+X$ , де  $X$  змінюється в межах від 0 до 9 :

- введіть в документ *MathCad* задану формулу; ідентифікатор змінюваної у виразі величини стає *FRAME* (за мовчазною угодою початкове значення змінної  $FRAME = 0$ );
- після виразу введіть оператор обчислення результату; отже, в робочому документі *MathCad* побачимо:  $87+FRAME=87$ ;
- виберіть в меню пункт **View-Animate**, після чого відкриється вікно редагування анімації (рис. 6.1);

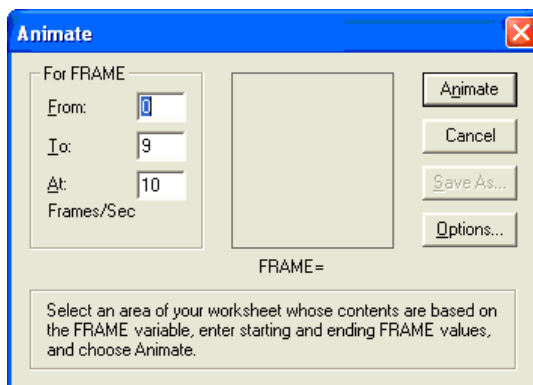



Рис.6.1. Вікно редагування анімації

- задайте початкове, кінцеве значення змінної *FRAME* та швидкість її зміни у вікнах, відповідно, *From*, *To*, *At*;
- виділіть у документі область, що містить майбутню анімацію (у прикладі – це область, в якій знаходиться формула), і натисніть у вікні кнопку *Animate*, кінцевий результат анімації можна бачити на рис. 6.2.; одночасно з вікном *Animate* у верхньому лівому кутку екрана з'явиться додаткове вікно *Playback* перегляду результатів анімації (див. рис. 6.3.); для перегляду анімації у цьому вікні призначена кнопка  .

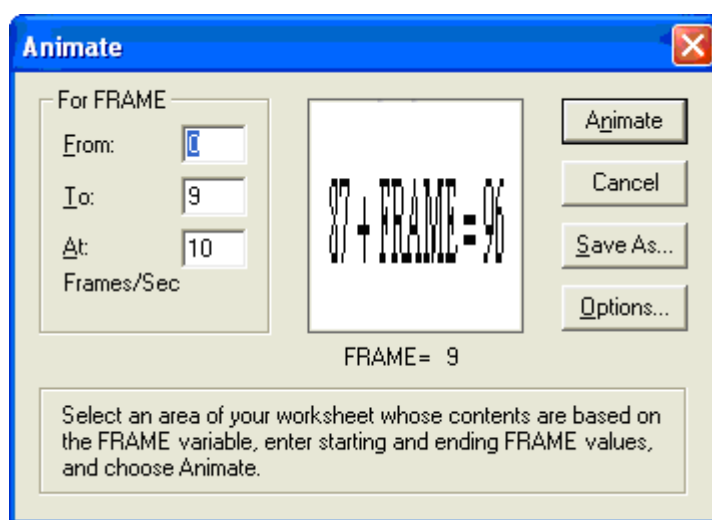


Рис.6.2. Задання параметрів анімації

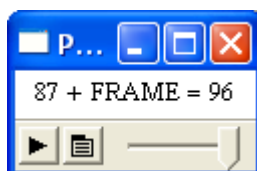


Рис.6.3. Перегляд результатів

Після перегляду анімації можна зберегти її у відеофайлі, натиснувши кнопку *Save As* у вікні *Animate*. Збережений відеофайл можна запускати без запуску математичного пакету.

Розглянемо етапи створення анімації в математичному пакеті при побудові анімованого графіка. Зобразимо рух крапки синусоїдною траєкторією. Використаємо вищеописаний алгоритм і у графічній області побудови декартового графіка у явному вигляді задамо нерухому ( $\sin(x)$ ) та рухому ( $\sin(\text{FRAME})$ ) функції ( див. рис.6.4.).



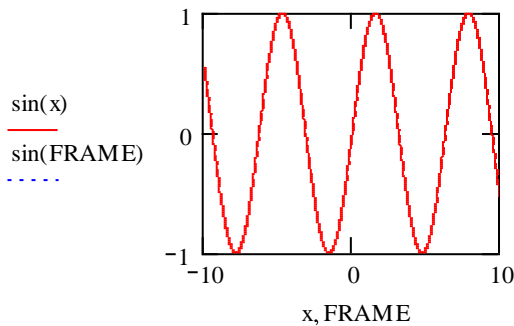


Рис.6.4. Графік функції  $\sin(x)$

Видно, що перший графік буде фіксованим, а другий мінятиметься відповідно до значень змінної  $FRAME$ . У кожному кадрі змінна  $FRAME$  приймає дискретне значення, отже, графіком другої функції буде сукупність крапок, кількість яких дорівнює кількості кадрів. Щоб крапку було добре видно, необхідно скористатися можливостями форматування графіка (див. рис. 6.5, 6.6).

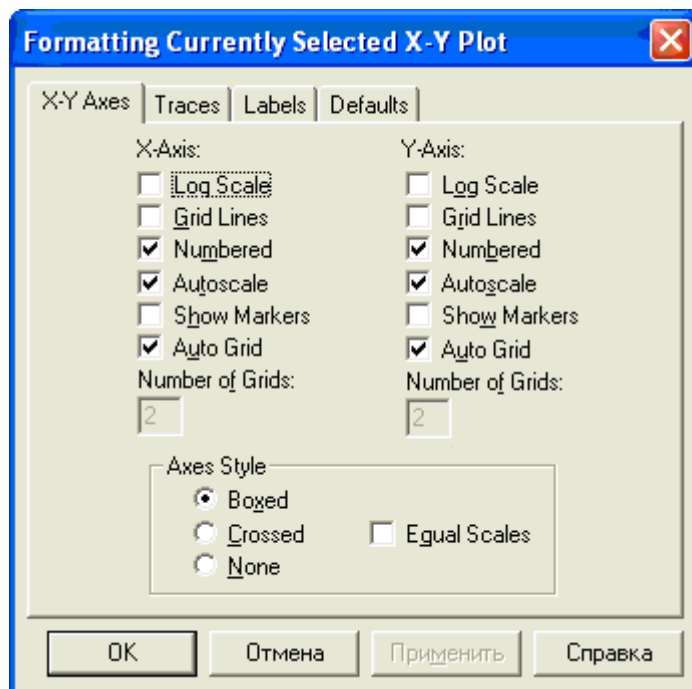


Рис.6.5. Форматування графіка функції  $\sin(x)$

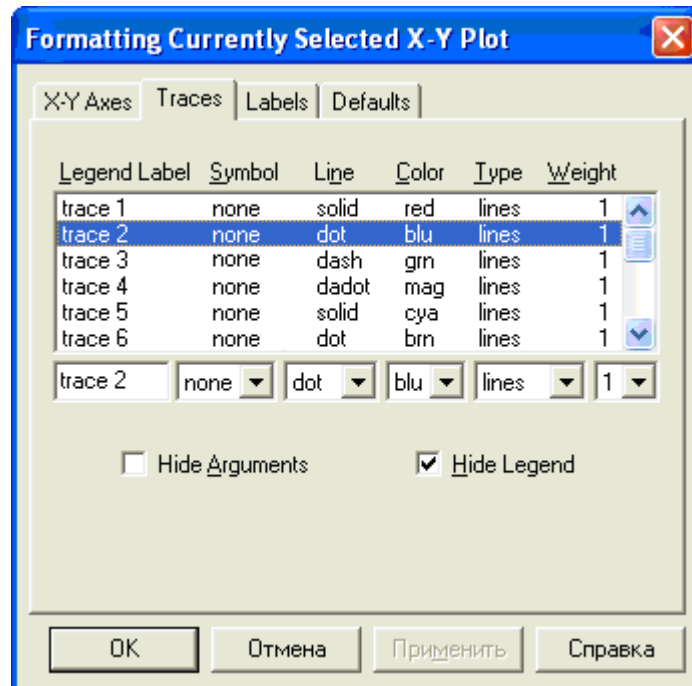


Рис.6.6. Форматування графіка функції  $\sin(x)$

Результат форматування може мати, наприклад, вигляд, наведений на рис. 6.7.

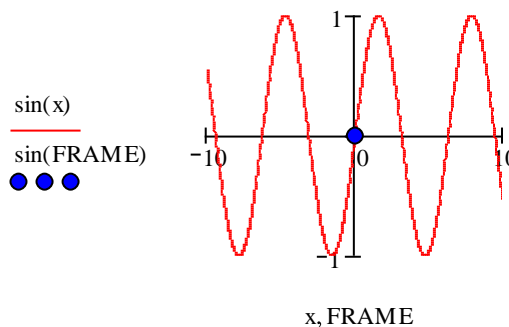


Рис.6.7.Результат форматування

Для створення анімаційного графічного зображення застосовуємо розглянутий вище алгоритм. Результат роботи можна переглянути у вікні *Playback* (див. рис. 6.8.).

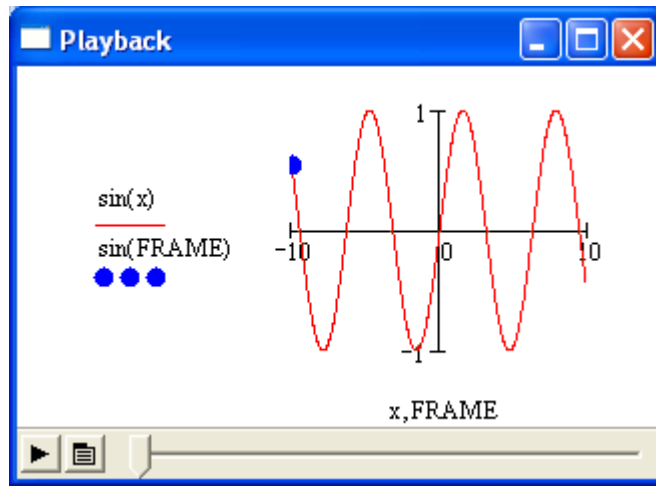


Рис.6.8. Перегляд результатів анімації

### Послідовність виконання роботи

1. Побудуйте анімаційне зображення функції з п.4 лабораторної роботи №4.
2. Створіть анімацію гармонійної функції зі змінною амплітудою.
3. Побудуйте анімаційне зображення значення  $f(x)=x+FRAME$  в полярних координатах, діапазон зміни  $x$  від 0 до 30, з кроком 0,1

### Контрольні запитання

1. Що таке FRAME ?
2. Чи можливо під час показу анімації зупинити зображення в якійсь певний момент?
3. Як переглянути збережену анімацію?
4. Створіть анімацію графіка тривимірної функції з лабораторної роботи 5.

## ОПЕРАЦІЇ З МАТРИЦЯМИ І ВЕКТОРАМИ

**Мета роботи** – навчитися задавати та виконувати операції над матрицями та векторами.

### Теоретичні відомості

Матриці у вигляді двовимірних масивів та вектори у вигляді одновимірних широко застосовуються при розв'язанні технічних задач.

Матриця чи вектор, як і ранжована змінна, є сукупністю елементів. Елементи ранжованої змінної розташовуються у порядку зростання або зменшення. До того ж різниця між двома сусідніми елементами ранжованої змінної є однаковою у межах заданого її діапазону. Значення елементів матриці чи вектора можуть утворюватись довільно.

Звернення до ранжованої змінної означає послідовне звернення до кожного з її елементів. На відміну від ранжованої змінної до будь-якого елемента матриці або вектора можна звернутися окремо, вказавши номери рядка і стовпця, на перетині яких він знаходиться, наприклад,  $M_{z,s}:=2zx+s$ ,  $B_s:=a+2s^n$ , де  $M$ ,  $B$  – ідентифікатори матриці та вектора відповідно;  $i$  – номер рядка,  $s$  – номер стовпця елемента матриці  $M$ , та номер елемента вектора-стовпця  $B$ .

Якщо в *MathCad* відбувається звернення до елементів матриці або вектора, які до цього моменту не набули значень, вони вважаються рівними нулю.

Нумерація елементів у межах рядка та стовпця починається з нуля. Початковий номер елемента зберігається у вбудованій константі *ORIGIN*. Його можна змінити, надавши константі *ORIGIN* бажане значення, використавши оператори локального або глобального надання значення ( $:=$ ,  $\equiv$ ), а також визначити, застосувавши оператор обчислення результату  $=$ .

Вектором *MathCad* вважає лише вектор-стовпець, а тому його елементи мають лише один номер – рядка, наприклад,  $a_s$ ,  $B_k$ . Вектор-рядок сприймається *MathCad* як матриця. Тому його елементи, як і елементи матриці мають два

номери. Отже, елементом матриці є індексована змінна, індексом якої є номери рядка і стовпця, розділені комою.

Для того, щоб увійти в режим введення індексу, треба натиснути клавішу із зображенням квадратної дужки, що відкривається. Щоб вийти з режиму введення індексу, треба потрібну кількість разів натиснути клавішу пропуску.

Палітра операцій з матрицями відкривається кнопкою із зображенням порожнього шаблону матриці на панелі математичних знаків (рис.7.1).

Задавати матрицю або вектор можна різними способами. Якщо значення елемента матриці залежить від номерів його рядка і стовпця, зручно використовувати спосіб 1 (див. рис.7.1); якщо існує залежність між сусідніми елементами – спосіб 2 (рис.7.1). Спосіб 3 передбачає заповнення шаблону матриці вручну. Сформувані шаблон можна в командному режимі командами ***Insert*** => ***Matrix*** (Вставка => Матриця) або, звернувшись до палітри математичних операцій.

#### Спосіб 1

$s := 0..5 \quad z := 0..6$   
 $M_{s,z} := s + z^2$

$$M = \begin{pmatrix} 0 & 1 & 4 & 9 & 16 & 25 & 36 \\ 1 & 2 & 5 & 10 & 17 & 26 & 37 \\ 2 & 3 & 6 & 11 & 18 & 27 & 38 \\ 3 & 4 & 7 & 12 & 19 & 28 & 39 \\ 4 & 5 & 8 & 13 & 20 & 29 & 40 \\ 5 & 6 & 9 & 14 & 21 & 30 & 41 \end{pmatrix}$$

#### Спосіб 3

$$M3 := \begin{pmatrix} 14 & -3 & 5 \\ 0 & 23 & -1 \\ \color{red}{\cdot} & \color{red}{\cdot} & \color{red}{\cdot} \end{pmatrix}$$

missing operand

ORIGIN = 0

#### Спосіб 2

$k := 1..5 \quad Vec_0 := 0.02$   
 $Vec_k := Vec_{k-1} + k \cdot 0.5$

$$Vec = \begin{pmatrix} 0.02 \\ 0.52 \\ 1.52 \\ 3.02 \\ 5.02 \\ 7.52 \end{pmatrix}$$

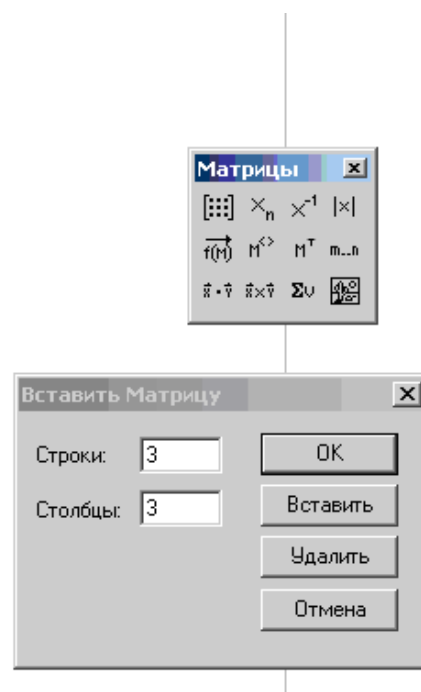


Рис.7.1. Способи введення матриць

У вікні, що відкриється (рис.7.1), треба задати кількість рядків і стовпців шаблону.

Можна надавати окремим елементам значення вручну (вектор  $V$  на рис.7.2), решту елементів *MathCad* заповнить нулями.

До матриці можна звернутися в цілому, вказавши її ідентифікатор. До елемента матриці можна звернутись, вказавши ідентифікатор індексованої змінної, тобто ідентифікатор матриці з нижнім числовим індексом, який міститиме номери рядка і стовпця елемента матриці, розділені комою. До стовпця матриці можна звернутися так: вказати ідентифікатор матриці, натиснути клавіші „*Ctrl*” та „*^*” та в полі верхнього індексу вказати номер стовпця.

Формат виведення матриці можна змінювати командами ***Format*** => ***Result*** (Формат => Результат) (рис.2.2). Вкладка „*Display Options*” (Настройка показа) вікна, що відкриється, пропонує різні стилі виведення матриці: ***Matrix*** – матричний, ***Table*** – табличний (рис.7.2).

$$s := 0..40 \quad z := 0..4$$

$$M2_{s,z} := \sin(s) + z$$

$$V_0 := 5.5 \quad V_4 := 12$$

$$V_8 := 6$$

Стиль "Матриця"

$$V = \begin{pmatrix} 5.5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 12 \\ 0 \\ 0 \\ 0 \\ 0 \\ 6 \end{pmatrix}$$

Стиль "Таблица"

M2 =

	0	1	2	3	4
0	0	1	2	3	4
1	0.84	1.84	2.84	3.84	4.84
2	0.91	1.91	2.91	3.91	4.91
3	0.14	1.14	2.14	3.14	4.14
4	-0.76	0.24	1.24	2.24	3.24
5	-0.96	0.04	1.04	2.04	3.04
6	-0.28	0.72	1.72	2.72	3.72
7	0.66	1.66	2.66	3.66	4.66
8	0.99	1.99	2.99	3.99	4.99
9	0.41	1.41	2.41	3.41	4.41
10	-0.54	0.46	1.46	2.46	3.46
11	-1	$9.79 \cdot 10^{-6}$	1	2	3
12	-0.54	0.46	1.46	2.46	3.46
13	0.42	1.42	2.42	3.42	4.42
14	0.99	1.99	2.99	3.99	4.99
15	0.65	1.65	2.65	3.65	4.65

Рис.7.2. Способи виведення матриці

Крім операцій з матрицями та векторами, представленими на панелі математичних операцій, *MathCad* підтримує ряд векторних та матричних функцій та функцій, що визначають характеристики матриць.

Серед матричних функцій є:

**augment** ( $M1, M2$ ) - утворює матрицю, лівою частиною якої є матриця  $M1$ , правою -  $M2$ ; кількість рядків  $M1$  і  $M2$  має бути однакою (об'єднання по горизонталі);

**diag** ( $V$ ) – утворює діагональну матрицю, елементами головної діагоналі якої є елементи вектора  $V$ ;

**identity** ( $n$ ) – утворює одиничну квадратну матрицю розміру  $n$ ;

**stack**( $M1, M2$ ) – утворює матрицю, верхньою частиною якої є матриця  $M1$ , нижньою  $M2$ ; кількість стовпців  $M1$  і  $M2$  має бути однакою (об'єднання по вертикалі);

**submatrix** ( $M, nr, mr, nl, ml$ ) – утворює з матриці  $M$  підматрицю з елементів, що знаходяться з  $nr$  по  $mr$  рядок та з  $nl$  по  $ml$  стовпець (рис.7.3).

$$M := \begin{pmatrix} 2 & 5 & 6 \\ 23 & 5 & 8 \\ 11 & 0 & 5 \end{pmatrix} \quad V := \begin{pmatrix} -2 \\ 1 \\ 9 \end{pmatrix} \quad M2 := \begin{pmatrix} 2 & 3 + 2i & 5 \\ 3i & 6 & 8 \end{pmatrix}$$

$$M1 := \text{augment}(M, V)$$

$$M1 = \begin{pmatrix} 2 & 5 & 6 & -2 \\ 23 & 5 & 8 & 1 \\ 11 & 0 & 5 & 9 \end{pmatrix}$$

$$\text{Re}(M2) = \begin{pmatrix} 2 & 3 & 5 \\ 0 & 6 & 8 \end{pmatrix}$$

$$\text{Im}(M2) = \begin{pmatrix} 0 & 2 & 0 \\ 3 & 0 & 0 \end{pmatrix}$$

$$|M| = -415$$

$$\text{identity}(4) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{length}(V) = 3$$

Рис.7.3. Приклади застосування матричних функцій

Серед функцій, що визначають характеристики матриць, є:

*cols* ( $M$ ) - повертає кількість стовпців матриці  $M$ ;

*rows* ( $M$ ) – повертає кількість рядків матриці  $M$ ;

*rank* ( $M$ ) - повертає ранг матриці  $M$ .

### Послідовність виконання роботи

1. Сформууйте вектор-стовпець, елементами якого є квадрати номерів цих елементів (кількість елементів вектора  $n=5$ ).
2. Задайте матрицю розміру  $n \times n$ , кожний елемент якої є сумою номерів його рядка і стовпця. Обчисліть визначник цієї матриці.
3. Визначте, який номер має перший елемент головної діагоналі матриці.
4. Задайте довільну матрицю та транспонуйте її.
5. Сформууйте діагональну матрицю із сформованого в п.1 вектора.
6. Змініть нумерацію елементів матриць і векторів документу.
7. Задайте матрицю розміру  $3 \times 3$  в символному вигляді, та обчисліть її визначник.

### Контрольні запитання

1. Як видалити рядок (рядки) з матриці або її шаблону?
2. Як додати в матриці або її шаблоні рядки (стовпці)?
3. Як звернутися до певного стовпця матриці?
4. Як змінити нумерацію елементів матриці?
5. Як об'єднати матриці по горизонталі, вертикалі?
6. Як обчислити ранг матриці вручну та за допомогою Mathcad?
7. Як помножити кожний елемент матриці на скаляр?
8. Як у символному вигляді знайти обернену матрицю?
9. Як змінити стиль виведення матриці?



## ОПЕРАЦІЇ НАД ПОЛІНОМАМИ

**Мета роботи** – навчитися за допомогою Mathcad виконувати дії над поліномами.

### Теоретичні відомості

Поліноми – дуже зручний інструмент в математиці. З одного боку – що може бути простіше від полінома? З іншого – це досить гнучкий інструмент. Нарощуючи степінь полінома, можна розширювати його можливості чи не до нескінченності. Поліноми можна використовувати для інтерполяції, екстраполяції, згладжування. Характеристичні поліноми, що відповідають диференціальним рівнянням та їх системам, несуть масу цінної інформації про досліджувані системи. Поліном має вид  $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0$ , де  $a_i$  – коефіцієнти,  $x$  – аргумент полінома,  $i=0, 1, 2, \dots, n$ .

Розглянемо найпростіші операції над поліномами, такі як додавання та віднімання. Задамо два поліноми  $A(x)$  та  $B(x)$  третього степеня з рівними коефіцієнтами. Далі поліном  $C(x)$  буде служити поліномом-результатом, наприклад,  $C(x) := A(x) + B(x)$ . В наступному рядку напишемо  $C(x)$  і виберемо команду **Evaluate Symbolically** (Символический знак равенства) з панелі **Evaluation** (рис 8.1). Віднімання та множення поліномів виконується аналогічно.

$$+ \quad A(x) := -8 \cdot x^3 + 5 \cdot x^2 - 12 \cdot x + 1 \quad B(x) := 2 \cdot x^3 - 2 \cdot x^2 + 7 \cdot x - 5$$

$$1. \quad C(x) := A(x) + B(x)$$

$$C(x) \rightarrow -6 \cdot x^3 + 3 \cdot x^2 - 5 \cdot x - 4$$

$$2. \quad C(x) := A(x) - B(x)$$

$$C(x) \rightarrow -10 \cdot x^3 + 7 \cdot x^2 - 19 \cdot x + 6$$

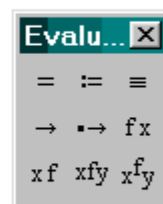


Рис 8.1. Додавання та віднімання поліномів

При множенні поліномів необхідно ввести поліноми  $A(x)$  та  $B(x)$ , далі поліному-добутку  $C(x)$ , задавши вираз  $C(x):=A(x)*B(x)$ . Для виведення результату на екран використовується команда **Expand** (Символическая оценка) з панелі **Evaluation** (рис 8.2).

$$A(x) := x^6 + 5x^5 - 6x^4 + 21x^3 - 7x^2 - 8 \cdot x + 1$$

$$B(x) := 5x^2 + 4x + 1$$

$$C(x) := A(x) \cdot B(x)$$

$$C(x) \text{ expand} \rightarrow 5 \cdot x^8 + 29 \cdot x^7 - 9 \cdot x^6 + 86 \cdot x^5 + 43 \cdot x^4 - 47 \cdot x^3 - 34 \cdot x^2 - 4 \cdot x + 1$$

+

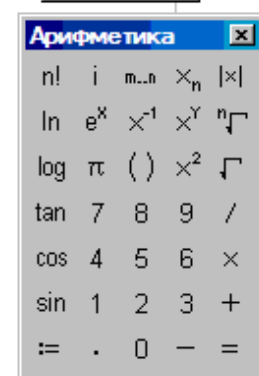
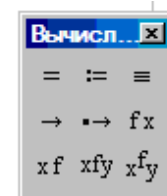


Рис. 8.2. Множення поліномів

### Послідовність виконання роботи

1. Задайте два полінома 5-го степеня з довільними коефіцієнтами.
2. Знайдіть третій поліном, що буде їх сумою, різницею, добутком.
3. Намалюйте вручну на аркуші протоколу у декартових координатах довільну криву. Оцифруйте осі.
4. Визначте набори коефіцієнтів інтерполяційних поліномів, обравши для кожного з них, з намальованої кривої, координати 5-ти, 10-ти, 15-ти точок.

Примітка: абсциси та ординати точок збережіть як елементи одновимірних масивів (векторів-стовпців).

5. В одних координатних осях намалюйте графіки отриманих інтерполяційних поліномів.

## Контрольні запитання

1. Що називають поліномом?
2. Опишіть алгоритм додавання поліномів, якщо один з них 3-го степеня, а другий 6-го?
3. Чи можна помножити поліноми, коли в одного ненульові коефіцієнти тільки при парних степенях аргумента, а в іншого – при непарних?
4. Що таке інтерполяційний поліном?
5. У чому особливість інтерполяційного полінома?



Для пошуку розв'язку системи лінійних рівнянь призначена також вбудована функція *lsolve*(A,b), що повертає вектор розв'язків (приклад 2 на рис.9.1).

Якщо розв'язок системи використовуватиметься в подальших розрахунках, його треба запам'ятати, використавши ідентифікатор вектора як, наприклад, у способі 1 на рис.9.1.

Для символного розв'язання систем лінійних рівнянь можна застосовувати метод, що використовує обернену матрицю коефіцієнтів системи (аналогічно способу 2 на рис.9.1), але замість оператора обчислення результату треба використати оператор  $\rightarrow$  з палітри символних перетворень, що відкривається кнопкою із зображенням магістерського капелюха на палітрі математичних знаків (див. рис.9.2).

$$M := \begin{pmatrix} a & b \\ -b & c \end{pmatrix} \quad V := \begin{pmatrix} -d \\ c \end{pmatrix}$$

$$M^{-1} \cdot \begin{pmatrix} -d \\ c \end{pmatrix} \rightarrow \begin{bmatrix} \frac{-c}{(a \cdot c + b^2)} \cdot d - \frac{b}{(a \cdot c + b^2)} \cdot c \\ \frac{-b}{(a \cdot c + b^2)} \cdot d + \frac{a}{(a \cdot c + b^2)} \cdot c \end{bmatrix}$$

Рис.9.2. Символьне розв'язання систем лінійних рівнянь

### Послідовність виконання роботи

1. За наперед відомим розв'язком сформулюйте систему чотирьох лінійних рівнянь.
2. Розв'яжіть систему рівнянь різними способами в числовому вигляді
3. Перевірте вірність отриманого розв'язку.
4. Розв'яжіть систему рівнянь у символному вигляді.
5. Розв'яжіть задану систему рівнянь, застосувавши функцію Find :

$$1) \begin{cases} x + y = 5, \\ x - 2y = 14; \end{cases}$$

$$2) \begin{cases} -2x + y = 7, \\ x - 3y = -1; \end{cases}$$

$$3) \begin{cases} -2x - 2y = -10, \\ 3x - 6y = 42; \end{cases}$$

$$7) \begin{cases} -8x + y = 12, \\ 9x - 5y = 2; \end{cases}$$

$$4) \begin{cases} -4x + 2y = 14, \\ x - 3y = -1; \end{cases}$$

$$8) \begin{cases} x - 7y = 7, \\ 5x - y = 1; \end{cases}$$

$$5) \begin{cases} 4x + 5y = 17, \\ x - 2y = 1; \end{cases}$$

$$9) \begin{cases} 8x + 10y = -34, \\ -x + 2y = 1; \end{cases}$$

$$6) \begin{cases} -2x - 5y = 4, \\ 3x + y = 7; \end{cases}$$

$$10) \begin{cases} 2x - 14y = 14; \\ -5x + y = -1. \end{cases}$$

6. Перевірте графічно отриманий у п.5 розв'язок.
7. Знайдіть коефіцієнти функції  $F(x) = a_2 \cdot x^2 + a_1 \cdot x + a_0$ , якщо відомі такі її значення :  $F(3) = 5$ ,  $F(-1) = 0$ ,  $F(-2) = 3.5$ .
8. Перевірте графічно отриманий розв'язок.

### Контрольні запитання

1. Які способи розв'язання систем рівнянь передбачають задання матриць їх коефіцієнтів?
2. В якому вигляді може подавати Mathcad символний розв'язок системи рівнянь?
3. Чи завжди система рівнянь має розв'язок?
4. Чи може система рівнянь мати декілька розв'язків?

## РОЗВ'ЯЗАННЯ СИСТЕМ НЕЛІНІЙНИХ АЛГЕБРИЧНИХ РІВНЯНЬ

**Мета роботи** – дослідити способи розв'язання систем нелінійних алгебричних рівнянь.

### Теоретичні відомості

Для пошуку точного розв'язку системи нелінійних рівнянь в *MathCad* існує вбудована функція *Find* ( $x_1, x_2, \dots, x_n$ ), результатом виконання якої є  $n$  невідомих системи (для системи порядку  $n$ ) – її розв'язок.

Застосування цієї функції вимагає створення в документі спеціального обчислювального блоку, що відкривається службовим словом *Given* та має таку структуру:

початкові умови

*Given*

рівняння

обмежувальні умови

вираз з функцією *Find*.

Блок такої структури дозволяє шукати розв'язок одного нелінійного рівняння або системи.

Початкові умови задають початкові значення шуканих змінних. В рівняннях системи, що задаються, замість звичайного знаку рівності використовується “жирний” знак рівності з палітри булевих операторів (див.рис.2.1). Обмежувальні умови зазвичай задаються рівностями або нерівностями, яким повинен задовольняти розв'язок.

Система рівнянь, задана на рис.10.1, має два набори розв'язку, оскільки пряма, описана другим рівнянням системи, перетинає параболу, що описана першим рівнянням, у двох точках. Кожний набір розв'язків шукається окремо зі своїми початковими і обмежувальними умовами.

$$x := 1 \quad y := 0$$

Given

$$3 \cdot x \cdot y + 2 \cdot x^3 = 0 \quad 5 \cdot x - y = 11$$

$$x > 0$$

$$\begin{pmatrix} x0 \\ y0 \end{pmatrix} := \text{Find}(x, y) \quad \begin{pmatrix} x0 \\ y0 \end{pmatrix} = \begin{pmatrix} 1.778 \\ -2.108 \end{pmatrix}$$

Given

$$3 \cdot x \cdot y + 2 \cdot x^3 = 0 \quad 5 \cdot x - y = 11$$

$$x < 0$$

$$\begin{pmatrix} x1 \\ y1 \end{pmatrix} := \text{Find}(x, y) \quad \begin{pmatrix} x1 \\ y1 \end{pmatrix} = \begin{pmatrix} 0 \\ -11 \end{pmatrix}$$

Рис.10.1. Приклади розв'язання системи нелінійних рівнянь

Для символного розв'язання систем нелінійних рівнянь використовується обчислювальний блок зі словом *Given*, але без початкових і обмежувальних умов, та зі знаком  $\rightarrow$  замість  $=$  після ідентифікатора функції пошуку розв'язку *Find*.

Приклад символного розв'язання рівняння наведений на рис.10.2.

Given

$$x \cdot y = a \quad b \cdot x + c \cdot y = b$$

$$\text{Find}(x, y) \rightarrow \begin{bmatrix} \left[ \frac{\frac{1}{2} \cdot b - \frac{1}{2} \cdot (b^2 - 4 \cdot c \cdot a \cdot b)^{\left(\frac{1}{2}\right)}}{b} \right] & \left[ \frac{\frac{1}{2} \cdot b + \frac{1}{2} \cdot (b^2 - 4 \cdot c \cdot a \cdot b)^{\left(\frac{1}{2}\right)}}{b} \right] \\ \frac{1}{(2 \cdot c)} \cdot \left[ b + (b^2 - 4 \cdot c \cdot a \cdot b)^{\left(\frac{1}{2}\right)} \right] & \frac{1}{(2 \cdot c)} \cdot \left[ b - (b^2 - 4 \cdot c \cdot a \cdot b)^{\left(\frac{1}{2}\right)} \right] \end{bmatrix}$$

Рис.10.2. Символьне розв'язання системи нелінійних рівнянь



### **Послідовність виконання роботи**

1. За наперед відомим розв'язком сформууйте систему трьох нелінійних рівнянь з двома невідомими.
2. Розв'яжіть систему рівнянь в числовому вигляді за допомогою блоку *Given*.
3. Перевірте вірність отриманного розв'язку.
4. Розв'яжіть систему рівнянь у символьному вигляді.
5. Відтворіть розв'язок системи рівнянь з пп.1 графічно.

### **Контрольні запитання**

1. Яке рівняння називають нелінійним? Наведіть приклади.
2. Наведіть структуру обчислювального блоку *Given*. Поясніть призначення його складових.
3. Чи можливо отримати графічний розв'язок системи нелінійних рівнянь? У чому його суть?
4. Чи може система нелінійних рівнянь не мати жодного розв'язку? Наведіть приклади.

## РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ ТА ЇХ СИСТЕМ

**Мета роботи** – навчитися шукати за допомогою *MathCad* числовий розв'язок диференціальних рівнянь та їх систем.

### Теоретичні відомості

Можливості *MathCad* забезпечують лише числове розв'язання звичайних диференціальних рівнянь та їх систем.

Усі функції *MathCad* призначені для розв'язання задачі Коші. Функція

$$rkfixed(y, t0, tn, n, D)$$

повертає матрицю розв'язків, яку формує із застосуванням методу Рунге-Кутта. У векторі  $y$  мають бути задані початкові умови, у змінних  $t0$  та  $tn$  – початкове та кінцеве значення незалежної змінної, у змінній  $n$  – кількість кроків на інтервалі  $(t0, tn)$ , у символічному векторі-функції  $D$  – праві частини рівнянь системи у формі Коші.

Нехай потрібно знайти розв'язок диференціального рівняння першого порядку виду

$$5y'(t) + y(t) = 2x(t)$$

з початковою умовою  $y(0) = 0$  за умови, що  $x(t) = 1(t)$ .

Подамо рівняння у формі Коші, підставивши замість загального позначення  $x(t)$  його конкретне значення:

$$y'(t) = \frac{1}{5}(2 \cdot 1(t) - y(t)).$$

Для заданого рівняння вектор початкових умов матиме лише один елемент і його можна задати напряму (див.рис.11.1). Функція  $D$  набуде значення правої частини рівняння у формі Коші.

Матриця розв'язків ( $RY$  у прикладі на рис.11.1), яку повертає функція *rkfixed*, має таку структуру: перший стовпець містить  $n$  значень незалежної змінної, починаючи з  $t0$ , другий стовпець – відповідні значення розв'язку  $y$ ; для

диференційного рівняння  $k$ -го порядку наступні стовпці містять значення  $y'$ ,  $y''$ , ...,  $y^{(k)}$ .

При розв'язанні рівняння першого порядку будуть заповнені перші три стовпці матриці-результату.

Побудувати графік розв'язку можна, обравши будь-який варіант, наведений на рис.11.1.

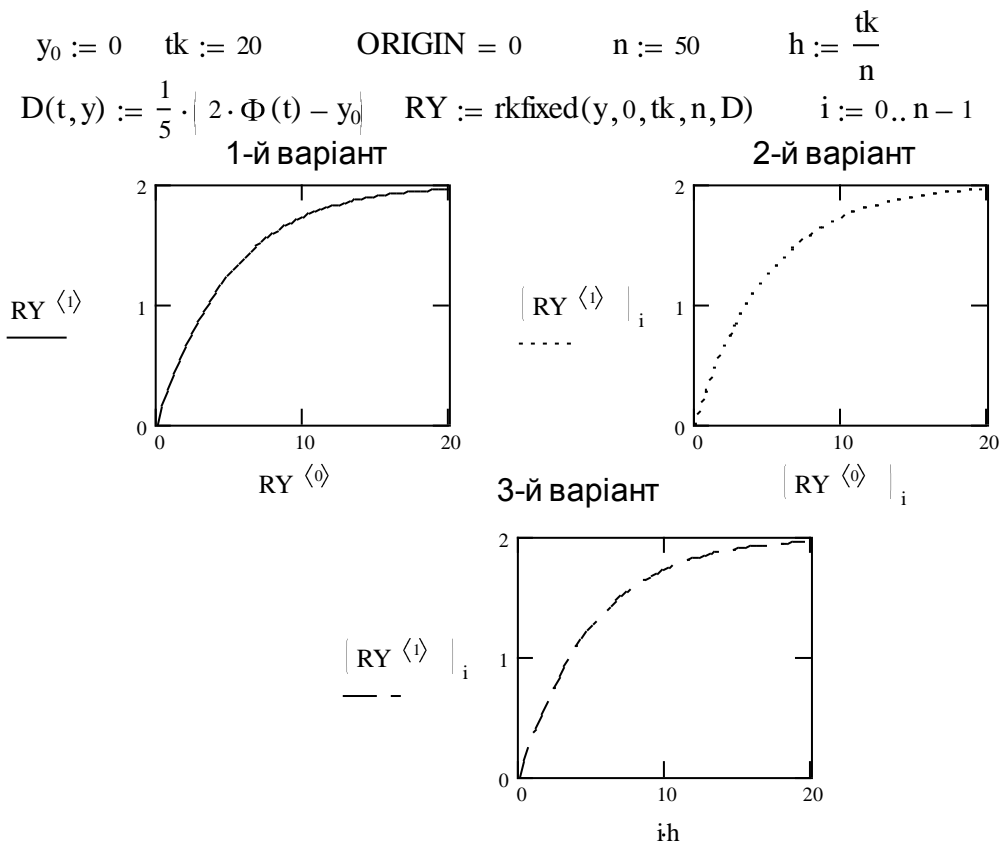


Рис.11.1. Варіанти побудови графіків розв'язку диференційного рівняння

У прикладі  $\Phi(t)=1(t)$  – функція Хевісайда.

Нехай треба розв'язати диференційне рівняння третього порядку виду

$$ay'''(t) + by''(t) + cy'(t) + dy(t) = r$$

з початковими умовами

$$y(0) = 1; \quad y'(0) = 0; \quad y''(0) = 0.$$

Введемо заміну:

$$y(t) = x_0; \quad y'(t) = x_1; \quad y''(t) = x_2.$$

Тоді початкові умови набудуть виду

$$x_0=1; x_1=0; x_2=0,$$

а початкове рівняння замінить система рівнянь у формі Коші:

$$\begin{cases} \frac{dx_0}{dt} = x_1, \\ \frac{dx_1}{dt} = x_2, \\ \frac{dx_2}{dt} = \frac{1}{a_3}(-a_0 - a_1x_1 - a_2x_2 - c). \end{cases}$$

Приклад розв'язання цього рівняння наведений на рис.11.2.

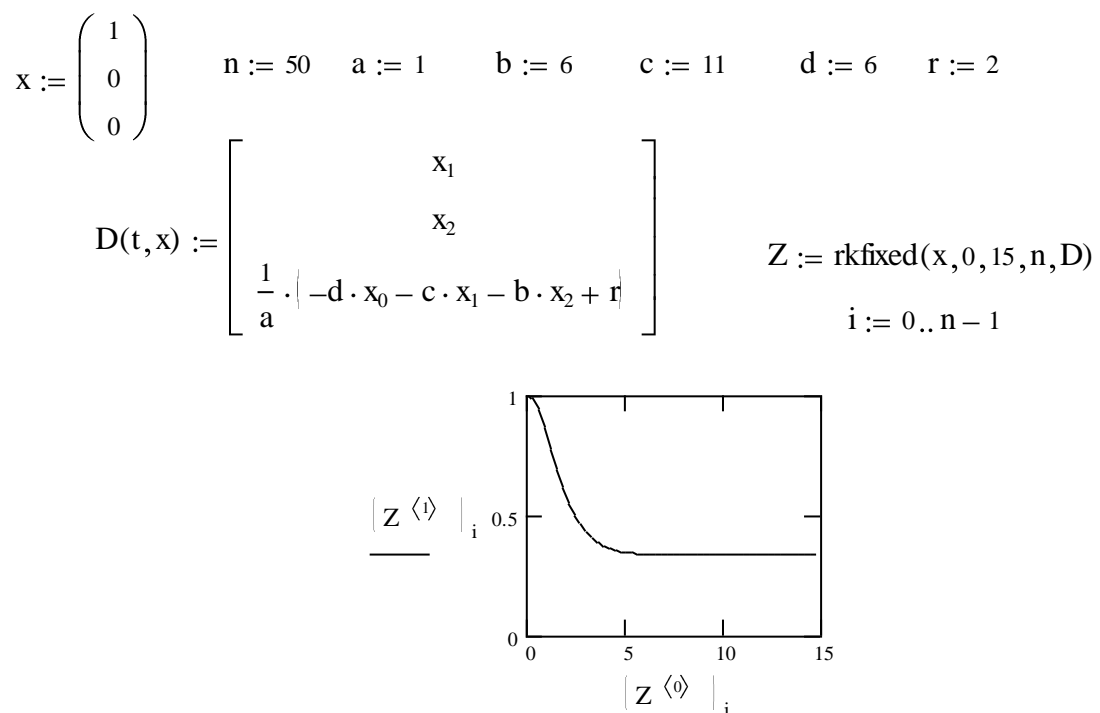


Рис.11.2. Приклад розв'язання диференційного рівняння третього порядку

Для розв'язання диференціальних рівнянь, записаних у *MathCad* у звичному вигляді, призначена функція *odesolve(t,tn,n)*, що використовується в обчислювальному блоці *Given*. Тут *t* – незалежна змінна рівняння, *tn* – кінцеве значення незалежної змінної, *n* – кількість кроків пошуку розв'язку на інтервалі  $(0, tn)$ . Третій параметр функції *odesolve* може бути відсутнім.

Приклад розв'язання вже розглянутого рівняння третього порядку за допомогою функції *odesolve* наведений на рис.11.3.

Given

$$\frac{d^3}{dt^3}y(t) + 6 \cdot \frac{d^2}{dt^2}y(t) + 11 \cdot \frac{d}{dt}y(t) + 6 \cdot y(t) = 2 \cdot \Phi(t)$$

$$y(0) = 1 \quad y'(0) = 0 \quad y''(0) = 0$$

$$y := \text{odesolve}(t, 15, 100) \quad t := 0, 0.15 \dots 15$$

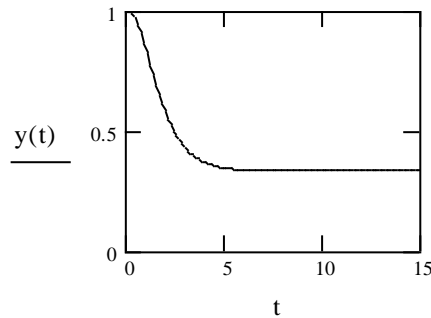


Рис.11.3. Приклад застосування блоку **Given** для розв’язання диференційного рівняння

Символи похідних при заданні початкових умов вводяться комбінацією клавіш “Ctrl”+”F7”.

### Послідовність виконання роботи

1. Задайте диференційне рівняння першого порядку  $Ty'(t) + y(t) = k$  з нульовими початковими умовами та довільними числовими значеннями коефіцієнтів  $T$  та  $k$ .
2. Знайдіть розв’язок рівняння і побудуйте графік розв’язку.
3. Знайдіть розв’язки диференційного рівняння з п.1 з ненульовими початковими умовами. Побудуйте ці розв’язки графічно в єдиних координатних осях.
4. Задайте диференційне рівняння  $y''(t) + w^2y(t) = 0$  з початковими умовами  $y(0)=1, y'(0)=0$ ;  $w$  – номер студента у списку групи.
5. Побудуйте в єдиних координатних осях графіки розв’язків, отриманих для різної кількості кроків на однаковому інтервалі незалежної змінної.
6. Проаналізуйте вплив кількості кроків (величини кроку) на точність

розв'язку.

7. Розв'яжіть диференціальне рівняння  $6y''(t) + 5y'(t) + y(t) = 2 \cdot 1(t)$  з нульовими початковими умовами різними способами.
8. Побудуйте графік функції Хевісайда.

### Контрольні запитання

1. Що таке задача Коші?
2. Який вигляд має диференціальне рівняння  $n$  – го порядку, записане у формі Коші?
3. Які функції Mathcad призначені для розв'язання диференціальних рівнянь?
4. Які методи (аналітичні, числові) використовує Mathcad при розв'язанні диференціальних рівнянь?

## 1. ПРОГРАМУВАННЯ

Розробники *MATHCAD* спочатку поставили перед собою завдання, метою якого було дати можливість професіоналам-математикам, фізикам і інженерам самостійно виконувати складні розрахунки, не звертаючись по допомогу до програмістів. Не дивлячись на блискуче втілення цих задумів, з'ясувалося, що зовсім без програмування *MATHCAD* серйозно втрачає в своїй силі, в основному, внаслідок незадоволення користувачів, знайомих з технікою створення програм і охочих здійснити свої розрахунки у звичному для себе стилі програміста. Замість знайомих принципів програмування користувачам старих версій *MATHCAD* пропонувалося комбінувати декілька специфічних вбудованих функцій і ранжовані змінні.

Останні версії *MATHCAD* мають не досить могутню, але вельми елегантну власну мову. З одного боку, вона дає можливість програмістові ефективно застосовувати програмний код в документах *MATHCAD*. З іншого, простота і інтуїтивність мови програмування дозволяє швидко її опанувати. До того ж, програмні модулі усередині документа *MATHCAD* поєднують в собі і відособленість (тому їх легко відрізнити від решти формул), і простоту смислового сприйняття.

Не дивлячись на невелику кількість операторів, мова програмування *MATHCAD* дозволяє розв'язувати різні, у тому числі і досить складні, завдання і є серйозною підмогою для розрахунків

### 1.1 .Програмування без програмування

У ранніх версіях *MATHCAD* вбудованої мови програмування не було. Щоб застосовувати звичні операції перевірки умов і організувати цикли, доводилося винаходити химерну суміш з вбудованих функцій умови *if* (лістинг 1.1) і *until* і комбінацій ранжованих змінних (лістинг 1.2).

#### Примітка

У зв'язку із сталими традиціями застосування мови програмування функцію *until* настійно не рекомендується використовувати в подальшій роботі (проте, вона діє в *MATHCAD* 11, але віднесена до застарілих функцій).

#### Лістинг 1.1 Функція умови

$$f(x) := \text{if}(x < 0, \text{"negative"}, \text{"positive"})$$

$$f(1) = \text{"positive"}$$

$$f(-1) = \text{"negative"}$$

#### Лістинг 1.2 Організація циклу за допомогою ранжованої змінної

$i := 0..10$

$x_i := i^2$

Використання ранжованих змінних – могутній апарат *MATHCAD*, схожий на застосування циклів в програмуванні. У переважній більшості випадків набагато зручніше організувати цикли (зокрема вкладені) за допомогою ранжованих змінних, ніж займатися для цього програмуванням. Корисно освоїти техніку, пов'язану з ранжованими змінними, векторами і матрицями, оскільки на ній засновані головні принципи розрахунків в *MATHCAD*, зокрема підготовка графіків.

## 1.2. Мова програмування MATHCAD

Для вставлення програмного коду в документи в *MATHCAD* є спеціальна панель інструментів *Programming* (Програмування), яку можна викликати на екран натисненням кнопки *Programming Toolbar* на панелі *Math* (Математика), як показано на рис. 1.1. Більшість кнопок цієї панелі виконана у вигляді текстового подання операторів програмування, тому їх зміст легко зрозуміти. Викладемо послідовно основні складові елементи мови програмування *MATHCAD* і розглянемо приклади її використання.

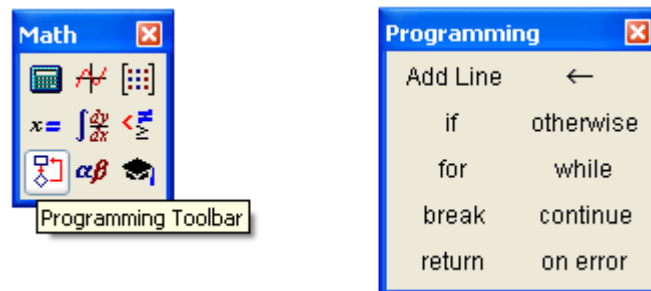


Рис.1.1 . Панель інструментів *Programming*

### 1.2.1. Що таке програма?

Основними інструментами роботи в *MATHCAD* є математичні вирази, змінні і функції. Нерідко записати формулу, що використовує ту або іншу внутрішню логіку (наприклад повернення різних значень залежно від умов), в один рядок не вдається. Призначення програмних модулів якраз і полягає у визначенні виразів, змінних і функцій в декілька рядків, часто із застосуванням специфічних програмних операторів.

Порівняйте визначення функції  $f(x)$  з лістингу 1.1 з визначенням  $f(x)$  за допомогою програмного модуля (лістинг 1.3).

Лістинг 1.3. Функція умови, визначена за допомогою програми



$$f(x) := \begin{cases} \text{"negative"} & \text{if } x < 0 \\ \text{"positive"} & \text{if } x > 0 \\ \text{"zero"} & \text{otherwise} \end{cases}$$

$$f(1) = \text{"positive"}$$

$$f(-1) = \text{"negative"}$$

$$f(0) = \text{"zero"}$$

Не дивлячись на принципову еквівалентність визначення функцій і змінних через вбудовані функції *MATHCAD* або програмні модулі, програмування має ряд істотних переваг, які у ряді випадків роблять документ простішим, а саме:

- можливість застосування циклів і умовних операторів;
- простота створення функцій і змінних, що вимагають декількох простих кроків (як в прикладі лістингу 1.3);
- можливість створення функцій, що містять закритий для решти документа код, включаючи переваги використання локальних змінних і обробку виняткових ситуацій (помилки).

Як видно з лістингу 1.3, програмний модуль позначається в *MATHCAD* вертикальною межею, праворуч від якої послідовно записуються оператори мови програмування.

### 1.2.2. Створення програми (Add Line)

Програма в *MATHCAD* має блокову структуру з підпорядкованими блоками, що виділені вертикальними лініями.

Щоб створити програмний модуль, наприклад, наведений в попередньому розділі (див. лістинг 1.3) треба виконати наступні дії:

- Введіть частину виразу, що знаходиться зліва від оператора надання значення і сам оператор присвоювання. У нашому прикладі це ім'я функції  $f(x)$ .
- При необхідності викличте на екран панель інструментів *Programming* (Програмування) (див. рис. 1.1).
- Натисніть на цій панелі кнопку *Add Line* (Додати лінію).
- Якщо наперед відомо, скільки рядків коду міститиме програма, можна створити потрібну кількість ліній повторним натисненням кнопки *Add Line* (Додати лінію) відповідну кількість разів (на рис. 1.2 показаний результат триразового натиснення).

У полях введення, що з'явилися, введіть бажаний програмний код, використовуючи програмні оператори. У даному прикладі в кожному полі введення вводиться рядок, наприклад, *"positive"* (рис. 1.3), потім натискається кнопка *If* (Якщо) на панелі *Programming* (Програмування) і у полі введення, що з'явиться, вводиться вираз  $x > 0$  (рис. 1.4).

Після того, як програмний модуль повністю визначений і жодне поле введення

не залишиться порожнім, функція може використовуватися звичайним способом, як в чисельних, так і в символічних розрахунках.

Не вводьте з клавіатури імена програмних операторів. Для їх вставки можна застосовувати лише поєднання клавіш, які наведені в тексті підказки, що з'являється (рис. 1.2 і 1.3).

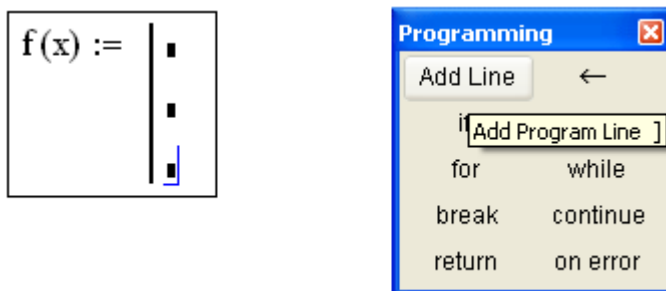


Рис. 1.2. Початок створення програмного модуля

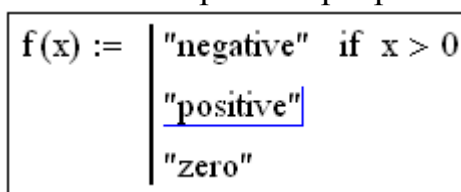


Рис. 1.3. Вставлення програмного оператора

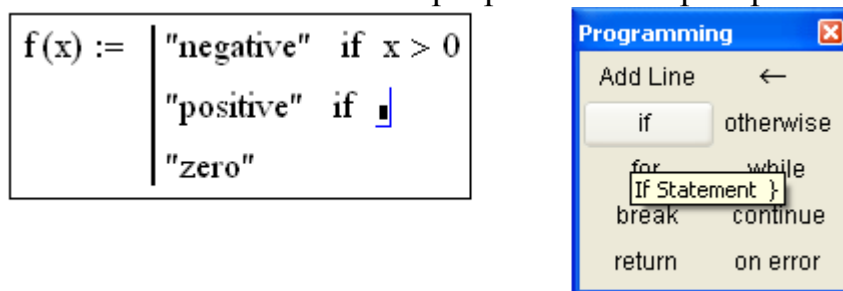


Рис. 1.4. Встлення умови в програму

### 1.2.3. Розробка програми

Вставити рядок програмного коду у вже створену програму можна у будь-який момент за допомогою тієї ж кнопки *Add Line* (Додати лінію). Для цього слід заздалегідь помістити на потрібне місце всередині програмного модуля курсор введення. Наприклад, розташування курсора введення на рядку, показаному на рис. 1.5, приведе до появи нової лінії з полем введення перед цим рядком. Якщо пересунути курсор введення з початку рядка (як на рис. 1.5) в її кінець, то нова лінія з'явиться після рядка.

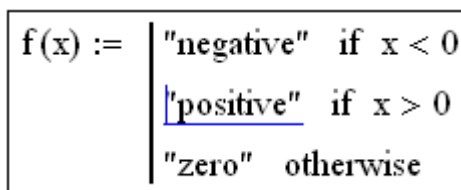


Рис. 1.5. Вставка нового рядка в існуючу програму

Якщо виділити рядок не цілком, а лише деяку його частину (рис. 1.6), то це вплине на положення в програмі нового рядка коду (результат натиснення кнопки *Add Line* показаний на рис. 1.7).

```
f(x) := | "negative" if x < 0
        | "positive" if x > 0
        | "zero" otherwise
```

Рис. 1.6. Положення ліній введення впливає на положення нової лінії

```
f(x) := | "negative" if x < 0
        | if x > 0
          | "positive"
        | "zero" otherwise
```

Рис. 1.7. Результат вставлення нової лінії в програму (з положення рис. 1.6)

Результати утворення нових рядків наведені на рис. 1.8.

Не забувайте, що для бажаного розміщення курсора введення всередині формули можна використовувати не тільки мишу і клавіші із стрілками, але і клавішу пропуску. За допомогою послідовних натиснень клавіші пропуску курсор введення "охоплює" різні частини формули.

<pre>f(x) :=   "negative" if x &lt; 0           "positive" if x &gt; 0           "zero" otherwise</pre>	<pre>f(x) :=   "negative" if x &lt; 0                     "positive" if x &gt; 0           "zero" otherwise</pre>
<pre>f(x) :=   "negative" if x &lt; 0           "positive" if x &gt; 0           "zero" otherwise</pre>	<pre>f(x) :=   "negative" if x &lt; 0           "positive" if x &gt; 0                     "zero" otherwise</pre>
<pre>f(x) :=   "negative" if x &lt; 0           "positive" if x &gt; 0           "zero" otherwise</pre>	<pre>f(x) :=   "negative" if x &lt; 0           if x &gt; 0                         "positive"           "zero" otherwise</pre>

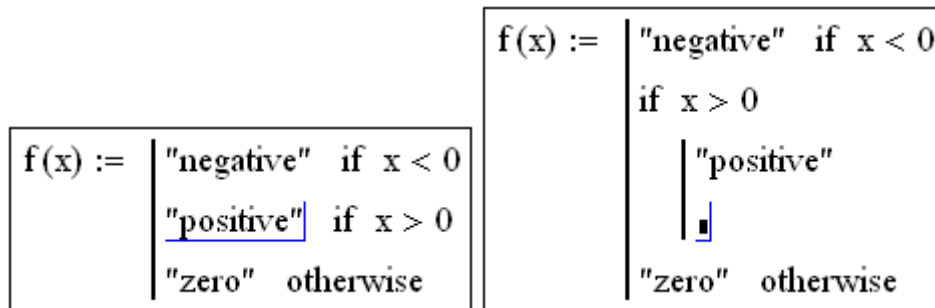


Рис. 1.8. Результат вставлення нової лінії в програму з різних положень курсора (праві фрагменти відтворюють результат операції)

Навіщо може бути потрібно вставити нову лінію в положення, показане на рис. 1.7. Нова вертикальна межа з двома лініями виділяє фрагмент програми, який належить до умови  $x > 0$ , що знаходиться в його заголовку. Приклад можливого подальшого програмування показаний в лістингу 1.4.

Лістинг 1.4. Приклад удосконалення програми

```
f(x) := | "negative" if x < 0
        | if x > 0
        | | "positive"
        | | "big positive" if x > 1000
        | "zero" otherwise
```

$f(1) = \text{"positive"}$

$f(10^5) = \text{"big positive"}$

У режимі виконання програми, а це відбувається при будь-якій спробі обчислити  $f(x)$ , виконується послідовно кожен рядок коду. Наприклад, в передостанньому рядку лістингу 1.4 обчислюється  $f(1)$ . Розглянемо алгоритм, реалізований у лістингу, на прикладі визначення значення функції  $f$  для аргументу  $x = 1$ .

1. Оскільки  $x = 1$ , та умову  $x < 0$  не виконано, і в першому рядку функція  $f$  не може бути визначена.

2. Умову другого рядка  $x > 0$  виконано, тому виконуються обидва наступні рядки, що належать до цієї умови і об'єднані короткою вертикальною межею в підблок, а саме функції  $f(x)$  надається значення "positive".

3. Умову  $x > 1000$  не виконано, тому значення "Big positive" не надається  $f(x)$ , вона так і залишається зі значенням, набутим у попередньому рядку.

4. Останній рядок не виконується, оскільки одна з умов ( $x > 0$ ) виявилась істинною, і оператор *otherwise* (тобто "інакше") не знадобився.

Отже, основний принцип створення програмних модулів полягає в правильному розташуванні рядків коду. Орієнтуватися в їх дії досить легко, оскільки фрагменти коду одного рівня згруповані в програмі за допомогою вертикальних ліній.

#### 1.2.4. Локальне надання значення ( $\leftarrow$ )

Мова програмування *MATHCAD* не була б ефективною, якби не дозволяла створювати всередині програмних модулів локальні змінні, які є недоступними ззовні, з інших частин документа. Надання значення в межах програм проводиться за допомогою оператора *Local Definition* (Локальне надання значення), який вставляється натисненням кнопки із зображенням стрілки  $\leftarrow$  на панелі *Programming* (Програмування).

Ні оператор надання значення  $:=$ , ні оператор визначення результату  $=$  в межах програм не застосовуються.

Локальне надання значення ілюструється лістингом 1.5. Змінна  $z$  існує тільки всередині програми, виділеної вертикальною лінією. З інших місць документа отримати її значення неможливо.

Лістинг 1.5. Локальне надання значення в програмі

$$\begin{array}{l} f(x) := \left| \begin{array}{l} z \leftarrow 4 \\ z + x \end{array} \right. \\ f(1) = 5 \end{array}$$

#### 1.2.5. Умовний оператор (if)

Дія умовного оператора *if* складається з двох частин. Спочатку перевіряється логічний вираз (умова) праворуч від нього. Якщо він істинний, то виконується вираз зліва від оператора *if*. Якщо він помилковий – нічого не відбувається, а виконання програми продовжується переходом до її наступного рядка.

Вставити умовний оператор у програму можна так (див. рис. 1.8).

- Якщо необхідно, введіть ліву частину виразу і оператора надання значення.
- Створіть перший рядок програмного коду, натиснувши на панелі *Programming* (Програмування) кнопку *Add Line* (Додати рядок).
- Натисніть кнопку умовного оператора *if*.
- Праворуч від оператора *if* введіть умову. Користуйтеся логічними операторами, вводячи їх з панелі *Boolean* (Булеві оператори).
- Вираз, який повинен виконуватись, якщо умова істинна, введіть зліва від оператора *if*.
- Якщо в програмі передбачаються додаткові умови, додайте в програму ще один рядок натисненням кнопки *Add Line* і введіть їх, використовуючи оператори *if* або *otherwise*.

Оператор *otherwise* використовується спільно з одним або декількома умовними операторами *if* і указує на вираз, який виконуватиметься, якщо жодна з умов не виявилась істинною. Приклади використання операторів *if* і *otherwise* приведені в попередніх розділах (див. лістинги 1.3 і 1.4).

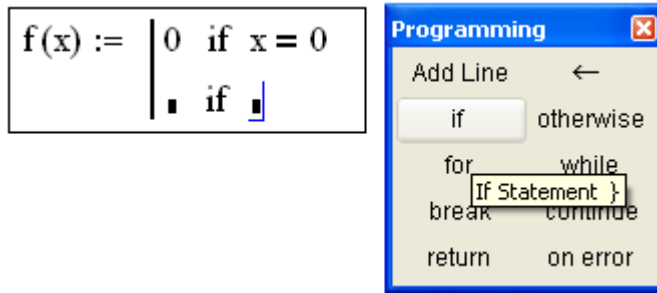


Рис. 1.8. Вставка умовного оператора

### 1.2.6. Оператори циклу (for, while)

У мові програмування *MATCAD* є два оператори циклу: *for* і *while*. Перший з них дає можливість організувати цикл за деякою змінною, що приймає ряд значень із заданого діапазону зі сталим кроком. Другий створює цикл, вихід з якого здійснюється за деякою логічною умовою. Щоб вставити в програмний модуль оператор циклу, потрібно виконати наступні дії.

- Створіть в програмному модулі нову лінію.
- Вставте один з операторів циклу *for* або *while* натисненням однойменної кнопки на панелі *Programming* (Програмування).
- Якщо вибраний оператор *for* (рис. 1.9), то вставте у відповідні поля введення ім'я змінної і діапазон її значень (лістинги 1.6 і 1.7), а якщо *while* – то логічний вираз, при порушенні якого повинен здійснюватись вихід з циклу (лістинг 1.8).

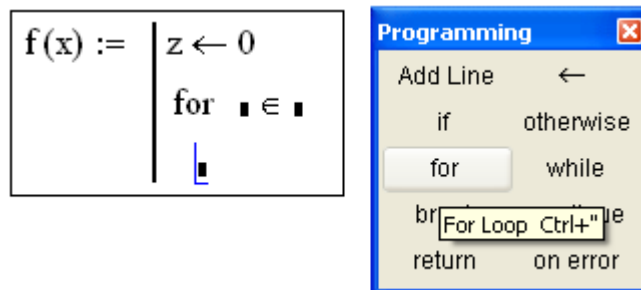


Рис. 1.9. Вставка оператора циклу

- У нижнє поле введення введіть тіло циклу, тобто вирази, які повинні виконуватись циклічно.

При необхідності доповніть програму іншими рядками і введіть в них потрібний код.

Діапазон значень змінної в умові циклу *for* можна задати як за допомогою ранжованої змінної (лістинг 1.6), так і за допомогою вектора (лістинг 1.7).

Лістинг 1.6. Оператор циклу *for* з ранжованою змінною

```
x := | z ← 0
      | for i ∈ 0.. 5
      |   z ← z + i
x = 15
```

Лістинг 1.7. Оператор циклу *for* з вектором

```
x := | z ← 0
      | for i ∈ 1, 2, 3
      |   z ← z + i
x = 6
```

Лістинг 1.8. Оператор циклу *while*

```
x := | z ← 0
      | while z < 10
      |   z ← z + 1
x = 10
```

Іноді необхідно достроково завершити цикл, тобто не за умовою в його заголовку, а в деякому рядку в тілі циклу. Для цього призначений оператор *break*. Модифікації лістингів 1.6 і 1.8 з перериванням циклу оператором *break* наведені в лістингах 1.9 і 1.10, відповідно. Наприклад в лістингу 1.9, як тільки значення змінної циклу *i* досягає 2, цикл завдяки оператору *break* в останньому рядку програмного модуля переривається. Відповідно, значення змінної *x* залишається рівним  $0 + 1 + 2 = 3$ .

Лістинг 1.9. Оператор *break* усередині циклу *for*

```
x := | z ← 0
      | for i ∈ 0.. 5
      |   | z ← z + i
      |   | break if i = 2
x = 3
```

Лістинг 1.10. Оператор *break* усередині циклу *while*

```

x := | z ← 0
      | while z < 10
      |   | z ← z + 1
      |   | break if z > 5
x = 6

```

Щоб чітко позначити межі завершення тіла циклу, в його кінці може використовуватись додатковий рядок з оператором *continue*, що вводиться однойменною кнопкою панелі *Programming*. Приклади, що модернізують лістинги 1.7 і 1.8, ілюструються лістингами 1.11 і 1.12, відповідно. Як видно, на результат програми наявність оператора *continue* не впливає.

Лістинг 1.11. Оператор *continue* в кінці циклу *while*

```

x := | z ← 0
      | while z < 10
      |   | z ← z + 1
      |   | continue
x = 10

```

Лістинг 1.12. Оператор *continue* в кінці циклу *for*

```

x := | z ← 0
      | for i ∈ 1,2,3
      |   | z ← z + i
      |   | continue
x = 6

```

### 1.2.7. Повернення значення (return)

Якщо для визначення змінної або функції застосовується програмний модуль, то його рядки виконуються послідовно при обчисленні в документі цієї змінної або функції. У ході виконання програми результат, що розраховується, зазнає зміни. Як остаточний результат видається останнє привласнене значення (приклади можна знайти в лістингах 1.3-1.12). Щоб підкреслити повернення програмним модулем певного значення, можна узяти за правило робити це в останньому рядку програмного модуля (лістинг 1.13). В результаті виконання програмного модуля функція  $f(x)$  набуде останнє значення локальної змінної  $z$ .



Лістинг 1.13. Повернення значення, заданого явно в останньому рядку програми

$$f(x) := \begin{cases} y \leftarrow x^2 \\ z \leftarrow y + 1 \\ z \end{cases}$$

$$f(2) = 5$$

Перервати виконання програми можна в будь-якому її рядку (наприклад за допомогою умовного оператора) і повернути деяке значення, застосувавши оператор *return*. У цьому випадку при виконанні вказаної умови (лістинг 1.14) значення, введене у поле введення після *return*, повертається як результат, а решта програмного модуля не виконується. Вставляється в програму оператор *return* за допомогою однойменної кнопки панелі *Programming* (Програмування).

Лістинг 1.14. Застосування оператора *return*

$$f(x) := \begin{cases} z \leftarrow x^2 \\ \text{return "zero" if } x = 0 \\ \text{return "i" if } x = 1 \\ z \end{cases}$$

$$f(-1) = 1$$

$$f(2) = 4$$

$$f(0) = \text{"zero"}$$

$$f(1) = \text{"i"}$$

### 1.2.8. перехоплення помилок (on error)

Програмування в *MATHCAD* дозволяє здійснювати додаткову обробку помилок. Якщо користувач припускає, що виконання коду в певному місці програмного модуля здатне викликати помилку (наприклад ділення на нуль), то цю помилку можна перехопити за допомогою оператора *on error*. Щоб вставити його в програму, треба помістити курсор введення в ній у потрібне положення і натиснути кнопку з ім'ям оператора *on error* на панелі *Programming* (Програмування). В результаті з'явиться рядок з двома полями введення і оператором *on error* посередині (рис. 1.10).

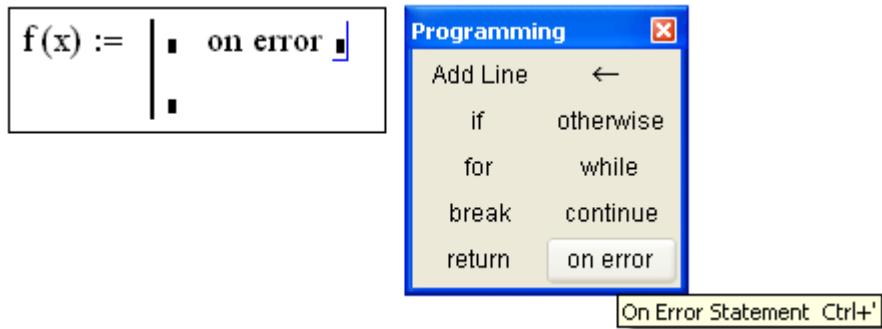


Рис. 1.10. Вставлення оператора переходу *no error*

У правому полі введення слід ввести вираз, що повинен виконуватись у даному рядку програми. У лівому – вираз, який буде виконано замість правого виразу, якщо при виконанні останнього виникне помилка. Приведемо приклад застосування оператора *on error* (лістинг 1.15) в програмному модулі, який розраховує обернене до *n* число. Якщо *n = 0*, то *i* значення  $z = 0$ , тому в останньому рядку програми виконується правий вираз розрахунку  $1/z$ . Так відбувається при розрахунку  $f(-2)$ . Якщо спробувати обчислити  $f(0)$ , як у кінці лістингу, то виконання програми, закладеної в  $f(n)$ , викличе помилку ділення на нуль в останньому рядку програми. Відповідно, замість виразу праворуч від оператора *on error* буде виконано лівий вираз, що надає функції  $f(n)$  значення типу рядок "Помилка користувача: ділення на нуль".

Лістинг 1.15. Перехоплення помилки ділення на нуль

$$f(n) := \left| \begin{array}{l} z \leftarrow n \\ \text{"Помилка користувача: ділення на нуль"} \quad \text{on error} \quad \frac{1}{z} \end{array} \right.$$

$f(-2) = -0.5$

$f(0) = \text{"Помилка користувача: ділення на нуль"}$

Оператор перехоплення помилок зручно застосовувати в комбінації з вбудованою функцією  $error(S)$ . Вона приводить до генерації помилки в звичній для MATHCAD формі з повідомленням *S*.

Зверніть увагу, що зроблені зміни звелися до переміщення тексту повідомлення про помилку в аргумент функції *error*.

### 1.3. Модульне програмування в MATHCAD

Загальна ідея модульного програмування полягає в наступному:

- реалізації обчислювальних процесів у вигляді окремих програмних одиниць-модулів;
- зверненні до цих модулів в інших програмах з передачею даних, необхідних для обчислювального процесу.

Модульне програмування дозволяє зменшити об'єм початкових текстів програм, прискорити написання і тестування програм, зменшити витрати на супровід (експлуатацію) програм.

Модульне програмування в *MATHCAD* можна реалізувати двома методами:  
модульне програмування в межах одного документа;  
модульне програмування в декількох документах *MATHCAD*.

### 1.3.1. Модульне програмування в одному документі

Цей метод характеризується тим, що:

- для реалізації простих обчислень використовуються локальні функції, а складніших – програми-функції;
- опис локальних функцій, програм-функцій і звернення до них знаходяться в межах одного документа і зберігаються в одному файлі. При цьому часто усередині однієї програми-функції знаходяться виклики локальних функцій, вбудованих функцій *MATHCAD* і іншої програми-функції.

Приклад 1.1. Реалізуємо у вигляді програми-функції обчислення визначеного інтеграла

$$\int_a^b f(x) dx,$$

використовуючи формулу Сімпсона з автоматичним вибором кількості вузлів. При цьому програма-функція *Simpson(f, a, b, N)* обчислює визначений інтеграл за формулою Сімпсона для фіксованої кількості інтервалів *N*, а програма-функція *Adapt(f, a, b)* вибирає за заданою точністю обчислення інтеграла (рівною  $10^{-8}$ ) кількість інтервалів (див. лістинг 1.16).

Використовуючи ці програми-функції, обчислимо інтеграл від функції  $f(x) = x^2$  на відрізку  $[0, 1]$ . Точне значення інтеграла дорівнює  $1/3 = 0.3333333333333333...$  Звернення до програми-функції *Adapt* дає результат *Adapt(f, 0, 1) = 0.3333333333333333*.

Перед зверненням до програми-функції *Adapt* необхідно описати функцію користувача  $f(x)$  у вигляді

$$f(x) := x^2,$$

оскільки ідентифікатор функції  $f(x)$  використовується як фактичний параметр.

Лістинг 1.16. Програма-функція обчислення визначеного інтеграла

$$\text{Simpson}(f, a, b, N) := \left| \begin{array}{l} h \leftarrow \frac{b - a}{N} \\ S \leftarrow f(a) - f(b) \\ \text{for } i \in 0.. N - 1 \\ \quad S \leftarrow S + 4f\left(a + i \cdot h + \frac{h}{2}\right) \\ \text{for } i \in 1.. N - 1 \\ \quad S \leftarrow S + 2f(a + i \cdot h) \\ \frac{h}{6} \cdot S \end{array} \right.$$
  

$$\text{Adapt}(f, a, b) := \left| \begin{array}{l} \text{eps} \leftarrow 10^{-8} \\ l1 \leftarrow \text{Simpson}(f, a, b, 5) \\ l2 \leftarrow \text{Simpson}(f, a, b, 10) \\ l2 \text{ if } |l1 - l2| < \text{eps} \\ \left( \text{Adapt}\left(f, a, \frac{a + b}{2}\right) + \text{Adapt}\left(f, \frac{a + b}{2}, b\right) \right) \text{ otherwise} \end{array} \right.$$

### 1.3.2. Модульне програмування в декількох документах MATHCAD

У попередньому способі реалізації модульного програмування опис модулів (функцій користувача і програм-функцій) і їх виклик знаходяться в одному документі. Такий спосіб має ряд недоліків:

- неможливість паралельної розробки програм декількома розробниками;
- неможливість "автономного" відлагодження програм-функцій і їх модифікації в процесі експлуатації програмного забезпечення;
- неможливість використання розробленої програми-функції в декількох документах без дублювання опису програми-функції.

Для подолання цих недоліків опис програми-функції виконують в одному документі *MATHCAD*, а її виклик розміщують в іншому документі (цей прийом широко використовується у сучасних алгоритмічних мовах високого рівня).

Проте при цьому виникає питання: як при виклику програми-функції в одному документі приєднати файл з іншим документом *MATHCAD*, в якому знаходиться опис програми-функції, що викликається? Для такого приєднання призначений спеціальний оператор *Reference*, який записується у вигляді, показаному на рис. 1.11.

☞ Reference:D:\Adapt\_Integration.mcd

Рис. 1.11 Структура оператора *Reference*

Оператор *Reference* вставляється у текст документа, в якому викликається програма-функція перед її викликом. Для вставки цього оператора необхідно виконати наступні кроки.

Крок 1. Клацнути лівою кнопкою миші в тому місці, куди буде вставлений оператор *Reference*.

Крок 2. Звернутись до пункту меню *Insert* і виконати команду *Reference*.

Крок 3. У полі введення діалогового вікна, що з'явилося, ввести повне ім'я файлу, що містить документ з описом функції, яка викликається. Для задання імені можна клацнути кнопкою "*Browse*" і в діалоговому вікні, що з'явилося, вказати диск, папку та ім'я файлу (у полі введення відіб'ється повне ім'я файлу).

Крок 4. Після виконаних установок клацнути кнопку "*OK*".

Після виконання цих кроків у документі з'явиться оператор *Reference*, показаний на рис. 1.11.

Таким чином, реалізація модульного програмування в декількох документах *MATHCAD* включає наступні етапи (які ілюструватимемо на прикладі обчислення визначеного інтеграла з використанням програм-функцій прикладу 1.1):

– опис у документі *MATHCAD* необхідних програм-функцій і збереження цього документа у файлі в потрібній папці під потрібним ім'ям (у нашому прикладі документ включатиме опис двох програм-функцій *Simpson* і *Adapt* і документ буде збережений на диску *D* у файлі під ім'ям *Adapt Integration.mcd*);

– вставлення оператора *Reference* у документі, в якому викликаються описані програми-функції шляхом виконання кроків 1-4; у нашому прикладі вставлений оператор *Reference* матиме вигляд, наведений у лістингу 1.17;

Описана реалізація модульного програмування дозволяє створювати бібліотеки програм-функцій, що реалізують обчислювальні алгоритми різної складності і використовувати бібліотеки програм-функцій, розроблені іншими користувачами.

Лістинг 1.17. Використання оператора *Reference* у документі

☞ Reference:D:\Adapt\_Integration.mcd

$$f(x) := x^2$$

$$\text{Adapt}(f, 0, 1) = 0.333$$

$$g(x) := \sin(x) + \cos(2x)$$

$$\text{Adapt}\left(g, \pi, 3 \frac{\pi}{2}\right) = -1$$

## 2. СИМВОЛЬНІ ОБЧИСЛЕННЯ

Розглядаються можливості символного процесора *MATHCAD*, оскільки він дозволяє розв'язувати багато задач математики аналітично, без застосування числових методів і, відповідно, без похибок обчислень. У методичних вказівках коротко розглядаються способи виконання символних обчислень у системі *MATHCAD*, а основний зміст розділу присвячений організації символних обчислень для розв'язання конкретних завдань математики. *MATHCAD* дозволяє виконувати широкий спектр аналітичних перетворень, таких, як алгебричні, матричні дії, основні операції математичного аналізу та інтегральні перетворення функцій.

### 2.1. Способи символних обчислень

Символьні обчислення в *MATHCAD* можна здійснювати в двох різних варіантах:

- 1) за допомогою команд меню;
- 2) за допомогою оператора пошуку символного результату  $\rightarrow$ , ключових слів символного процесора і звичайних формул (у довідковій системі *MATHCAD* цей спосіб називається символними обчисленнями у режимі реального часу).

Перший спосіб зручніший, якщо потрібно швидко отримати аналітичний розв'язок для одноразового використання, не зберігаючи сам хід обчислень. Другий спосіб наочніший, оскільки дозволяє записувати вирази в традиційній математичній формі і зберігати символні обчислення в документах *MATHCAD*. Крім того, аналітичні перетворення, що виконуються через меню, стосуються тільки одного, виділеного в даний момент, виразу. На перетворення не впливають вирази, що знаходяться в документі *MATHCAD* вище виділеного (наприклад, оператори надання значення змінним). Оператор пошуку символного результату, навпаки, враховує весь попередній вміст документа і визначає результат з його урахуванням.

У символних обчисленнях допускається використання більшості вбудованих функцій *MATHCAD*.

Для символних обчислень за допомогою команд призначено головне меню *Symbolics* (Символіка), що об'єднує математичні операції, які *MATHCAD* виконує аналітично (рис. 2.1). Для реалізації другого способу застосовуються всі засоби *MATHCAD*, пристосовані для числових обчислень (наприклад, панелі *Calculator*, *Evaluation* і т. д.), і спеціальна математична панель інструментів, яку можна викликати на екран натисненням кнопки *Symbolic Keyword Toolbar* (Панель символіки) на панелі *Math*. На панелі *Symbolic* (Символічні) знаходяться кнопки, що відповідають специфічним командам символних перетворень (рис.2.2). Наприклад, таким, як розкладання виразу на множники,

визначення перетворення Лапласа і іншим операціям, які в *MATHCAD* не можна виконувати в числовому вигляді, і для яких не передбачені вбудовані функції.

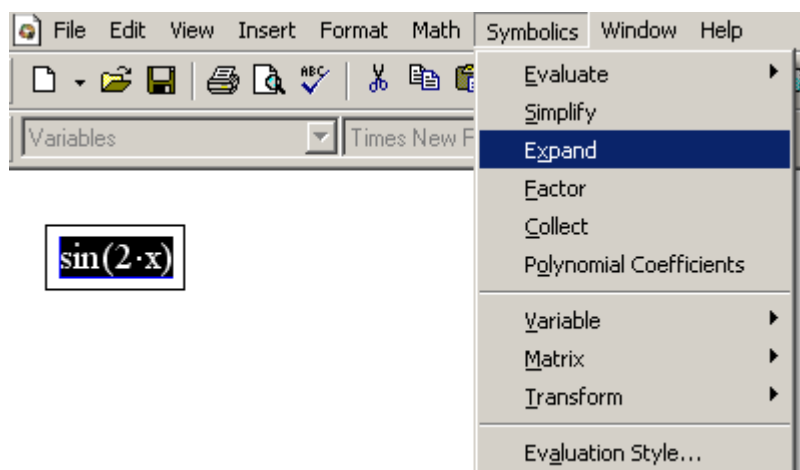


Рис. 2.1. Меню Symbolics

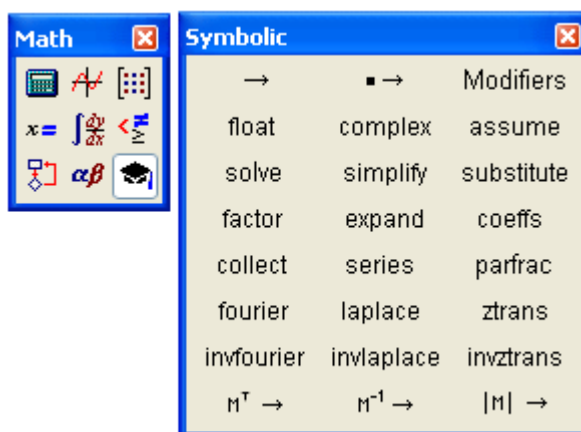


Рис. 2.2. Панель Symbolic

Розглянемо обидва способи символічних обчислень на простому прикладі розкладання на множники виразу  $\sin(2x)$ .

Спосіб 1 (за допомогою меню).

1. Введіть вираз  $\sin(2x)$ .
2. Виділіть його весь (див. рис. 2.1).
3. Виберіть в головному меню пункти *Symbolics/Expand* (Символіка/Розкласти). Після цього результат розкладання виразу виведеться у наступному рядку (рис. 2.3).

$$\sin(2x)$$

$$2 \cdot \sin(x) \cdot \cos(x)$$

Рис. 2.3. Результат застосування команди меню *Symbolics/Expand*

Примітка.

Символьні операції за допомогою меню можливі лише над виразом, його частиною або окремою змінною і є недоступними для функцій користувача. Для того, щоб правильно здійснити бажане аналітичне перетворення, заздалегідь необхідно виділити той об'єкт, над яким воно виконуватиметься. У даному випадку перетворення було застосоване до всього виразу  $\sin(2x)$ . Якщо ж виділити частину формули, як показано на рис.2.4, то відповідне перетворення буде виконане над виділеною частиною (нижній рядок на цьому рисунку).

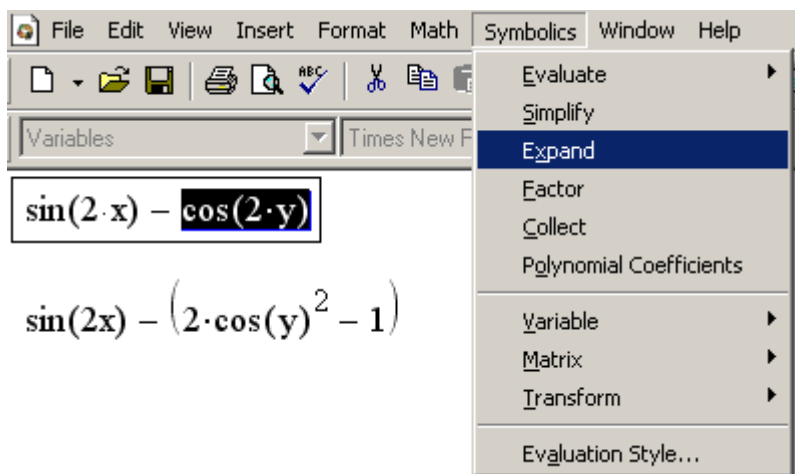


Рис. 2.4. Символьне розкладання частини виразу і його результат

Спосіб 2 символічних перетворень (за допомогою оператора  $\rightarrow$ ).

1. Введіть вираз  $\sin(2x)$ .
2. Натисніть кнопку *Expand* (Розкласти) на панелі *Symbolic*.
3. Введіть у поле введення після оператора *expand* (рис. 2.5, верхній рядок) ім'я змінної  $x$  або натисніть клавішу  $\langle Del \rangle$ , щоб просто видалити поле введення.

$\sin(2 \cdot x) \text{ expand, } \rightarrow$

$\sin(2x) \text{ expand, } x \rightarrow 2 \cdot \sin(x) \cdot \cos(x)$

Рис. 2.5. Символьне розкладання виразу

4. Введіть оператор визначення символічного результату  $\rightarrow$ .
5. Натисніть клавішу  $\langle Enter \rangle$  або просто клацніть мишею за межами виразу.

Оператор визначення символічного результату можна ввести в редакторі *MATHCAD* декількома способами: натисненням кнопки  $\rightarrow$  на будь-якій з панелей *Evaluation* (Вирази) або *Symbolic* (Символіка) або поєднанням клавіш  $\langle Ctrl \rangle + \langle . \rangle$ . Результат символічного розкладання виразу наведений на рис 2.5.

Якщо символічні обчислення здійснюються способом 2, символічний процесор враховує всі вирази, заздалегідь введені в документі (рис. 2.6). Але якщо ті ж перетворення виконуються за допомогою меню, символічний



процесор "не бачить" нічого, крім одного, і сприймає усі його змінні аналітично, навіть якщо їм заздалегідь були надані значення (рис. 2.6). З цієї причини, наприклад, символьним перетворенням через меню недоступні попередні визначення функцій користувача.

Більш доцільно при використанні символьних обчислень є оператор  $\rightarrow$ , оскільки при цьому в документі зберігаються дії користувача. Наявність спеціального меню символьних обчислень – певною мірою дань колишнім версіям *MATHCAD*. У них аналітичні перетворення були вбудовані не так гармонійно і були доступні, головним чином, через меню.

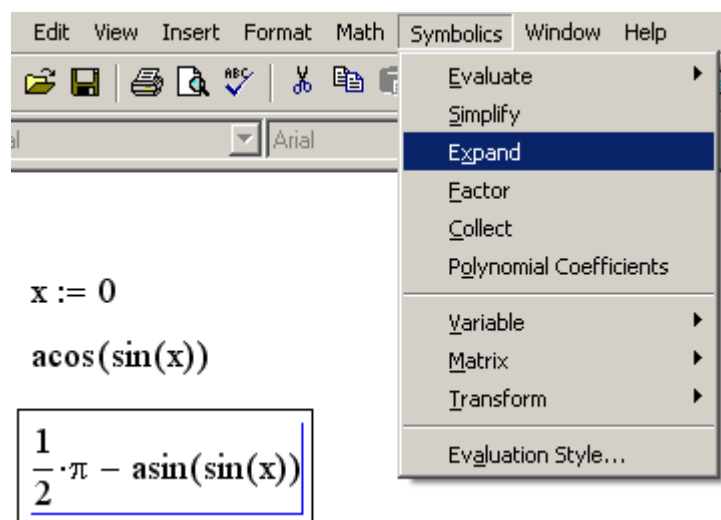


Рис. 2.6. Символьне обчислення, виконане за допомогою меню

Не кожний вираз піддається аналітичним перетворенням. Якщо це так (або внаслідок того, що завдання зовсім не має аналітичного розв'язку, або він виявляється дуже складним для символьного процесора *MATHCAD*), то як результат виводиться сам вираз (лістинг 2.1).

Лістинг 2.1. Символьні перетворення

$$\cos(2x) \text{ expand, } x \rightarrow 2 \cdot \cos(x)^2 - 1$$

$$\cos(x) \text{ expand, } x \rightarrow \cos(x)$$

Далі у цьому розділі, розглядаючи символьні обчислення за допомогою меню, ілюструватимемо результати на рисунках, а символьні обчислення із застосуванням оператора  $\rightarrow$  наводитимемо у вигляді лістингів.

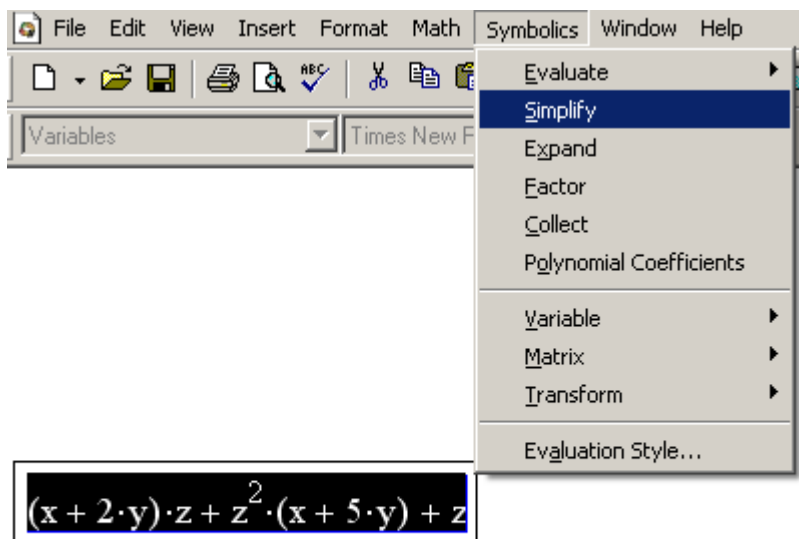
## 2.2. Символьна алгебра

Символьний процесор *MATHCAD* виконує основні перетворення алгебри, такі, як спрощення виразів, розкладання їх на множники, символьне підсумовування і перемножування.

### 2.2.1. Спрощення виразів (Simplify)

Спрощення виразів – найчастіше вживана операція. Символьний процесор *MATHCAD* намагається так перетворити вираз, щоб він прийняв простішу форму. При цьому використовуються різні арифметичні формули, зведення подібних, тригонометрична тотожність, перерахунок зворотних функцій і ін. Щоб спростити вираз за допомогою меню (рис. 2.7) треба виконати такі дії.

1. Введіть вираз.
2. Виділіть вираз разом або його частину, яку потрібно спростити.
3. Виберіть команду *Symbolics/Simplify* (Символіка/Спростити).



$$z \cdot x + 2 \cdot z \cdot y + z^2 \cdot x + 5 \cdot z^2 \cdot y + z$$

Рис. 2.7. Спрощення виразу

Для спрощення виразу за допомогою оператора визначення символьного результату використовуйте ключове слово `simplify` (лістинг 2.2). Не забувайте, якщо деяким змінним, що входять до виразу, раніше були надані значення, вони будуть враховані при виконанні символьних обчислень (лістинг 2.3).

Лістинг 2.2. Спрощення виразу

$$(x + 2 \cdot y) \cdot z + z^2 \cdot (x + 5 \cdot y) + z \text{ simplify} \rightarrow z \cdot x + 2 \cdot z \cdot y + z^2 \cdot x + 5 \cdot z^2 \cdot y + z$$

Лістинг 2.3. Спрощення виразу з підстановкою значення змінних

$$x := 10 \quad y := 1$$

$$(x + 2 \cdot y) \cdot z + z^2 \cdot (x + 5 \cdot y) + z \text{ simplify} \rightarrow 13 \cdot z + 15 \cdot z^2$$

Спрощення виразів, що містять числа, проводиться по-різному, залежно від наявності в числах десяткової крапки. Якщо вона є, то виконується безпосереднє обчислення виразу (лістинг 2.4).

Лістинг 2.4. Спрощення числового виразу

$$\sqrt{3} \text{ simplify} \rightarrow 3^{\frac{1}{2}}$$

$$\sqrt{3.01} \text{ simplify} \rightarrow 1.7349351572897472412$$

### 2.2.2. Розкладання виразів (Expand)

Операція символічного розкладання, (розширення), виразів протилежна за змістом операції спрощення. У ході розкладання розкриваються всі суми і добутки, а складні тригонометричні залежності розкладаються за допомогою тригонометричних тотожностей. Розкладання виразів виконується командою *Symbolics/Expand* (Символіка/Розкласти) або використанням разом з оператором введення символічного результату ключового слова *expand* (див. рис. 2.3 – 2.6 і лістинг 2.1).

### 2.2.3. Розкладання на множники (Factor)

Розкладання виразів на прості множники виконується командою *Symbolics/Factor* (Символіка/Розкласти на множники) (рис. 2.8) або використанням разом з оператором визначення символічного результату ключового слова *factor* (лістинг 2.5). Ця операція дозволяє розкласти поліноми на добутки простіших поліномів, а цілі числа – на прості множники. Застосовуючи команду меню, не забувайте перед її викликом виділити весь вираз або його частину, яку плануєте розкласти на множники.

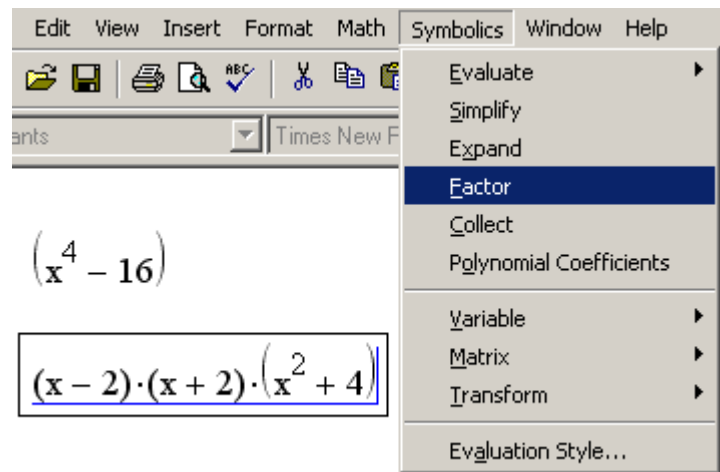


Рис. 2.8. Розкладання виразу на множники

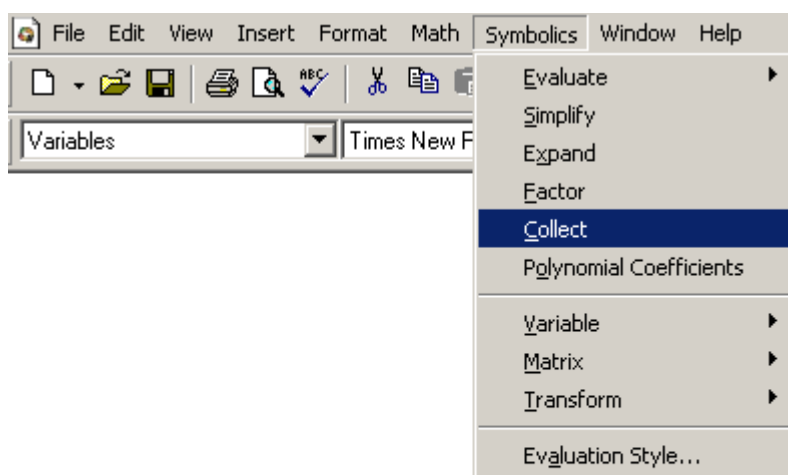
#### Лістинг 2.5 Розкладання на множники

$$(x^4 - 16) \text{ factor} \rightarrow (x - 2) \cdot (x + 2) \cdot (x^2 + 4)$$

### 2.2.4. Зведення подібних (Collect)

Для зведення подібних у поліномі за допомогою меню (рис. 2.9) потрібно виконати наступні кроки.

1. Ввести вираз.
2. Виділити у виразі ідентифікатор змінної, відносно якої треба звести подібні (у прикладі на рис. 2.9 – це змінна  $y$ ).
3. Вибрати команду *Symbolics/Collect* (Символика/Звести подібні). Нижче з'явиться рядок з результатом зведення подібних (нижній рядок на рис. 2.9).



$$(2 \cdot z - 5 \cdot z^2) \cdot y + x \cdot z - z^2 \cdot x + z$$

Рис.2.9. Зведення подібних у виразі

Щоб звести подібні доданки за допомогою оператора визначення символічного результату (лістинг 2.6), треба виконати наступні дії.

1. Ввести вираз.
2. Натиснути кнопку *Collect* на панелі *Symbolic* (Символіка).
3. Ввести у полі введення після вставленого ключового слова *collect* ідентифікатор змінної, відносно якої потрібно звести подібні (у прикладі з лістингу 2.6 це змінна  $y$ ).
4. Ввести оператор визначення символічного результату  $\rightarrow$ .
5. Натиснути клавішу  $\langle \text{Enter} \rangle$ .

Лістинг 2.6 Зведення подібних у виразі

$$(x + 2y) \cdot z - z^2 \cdot (x + 5y) + z \text{ collect } y \rightarrow (2 \cdot z - 5 \cdot z^2) \cdot y + z - x \cdot z^2 + x \cdot z$$

Примітка.

Після ключового слова *collect* допускається задання декілька змінних через кому. У цьому випадку, зведення подібних доданків виконується послідовно для кожної змінної.

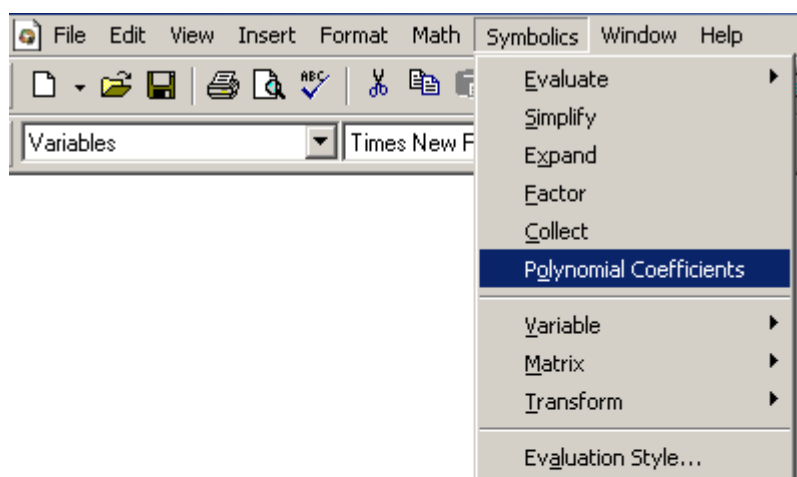
## 2.2.5. Коефіцієнти полінома (Polynomial Coefficients)

Якщо вираз є поліномом відносно аргумента  $x$ , заданим у вигляді добутку простіших поліномів, то коефіцієнти  $a_0, a_1, \dots, a_n$  поліноміального виразу  $a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ , легко визначаються символьним процесором *MATHCAD*. Коефіцієнти можуть бути функціями (іноді досить складними) інших змінних. Щоб обчислити поліноміальні коефіцієнти у виразі за допомогою меню (рис. 2.10) треба виконати наступні дії.

1. Ввести вираз.

2. Виділити в ньому ідентифікатор змінної або вираз, відносно якого потрібно визначити поліноміальні коефіцієнти (у прикладі на рис. 2.10 – це змінна  $z$ ).

3. Виконати команду *Symbolic/Polynomial Coefficients* (Символіка/ Коефіцієнти полінома).



$$(x - 2 - y) \cdot z - z^2 \cdot (x + y) + z$$

$$\begin{pmatrix} 0 \\ x - 1 - y \\ -x - y \end{pmatrix}$$

Рис.2.10. Обчислення коефіцієнтів полінома

Як результат під виразом з'явиться вектор, що містить поліноміальні коефіцієнти. Першим елементом вектора є вільний коефіцієнт  $a_0$ , другим –  $a_1$ , і т.д.

Щоб обчислити поліноміальні коефіцієнти за допомогою оператора визначення символьного результату, треба виконати такі дії.

1. Введіть вираз.

2. Натисніть кнопку *Coeffs* на панелі *Symbolic* (Символіка).

3. Введіть в полі введення після вставленого ключового слова *coeffs* аргумент полінома.

4. Введіть оператор символного результату  $\rightarrow$ .

5. Натисніть клавішу  $\langle \text{Enter} \rangle$ .

Приклади обчислення коефіцієнтів полінома наведені в лістингах 2.7 і 2.8.

Лістинг 2.7 ілюструє розрахунок коефіцієнтів для різних аргументів. Лістинг 2.8 демонструє можливість визначення коефіцієнтів не тільки відносно змінної, але і відносно підвиразу, що входить до складу початкового виразу.

Лістинг 2.7. Обчислення коефіцієнтів полінома

$$(x + 2 \cdot y) \cdot z - z^2 \cdot y \cdot (x + 5 \cdot y) + z \text{ coeffs}, z \rightarrow \begin{pmatrix} 0 \\ x + 2 \cdot y + 1 \\ -y \cdot x - 5 \cdot y^2 \end{pmatrix}$$

$$(x + 2y) \cdot z - z^2 y(x + 5y) + z \text{ coeffs}, x \rightarrow \begin{pmatrix} 2 \cdot z \cdot y - 5 \cdot z^2 \cdot y^2 + z \\ z - z^2 \cdot y \end{pmatrix}$$

Лістинг 2.8. Обчислення поліноміальних коефіцієнтів відносно змінної і виразу

$$(x - 4) \cdot (x - 7) \cdot x + 99 \text{ coeffs}, x \rightarrow \begin{pmatrix} 99 \\ 28 \\ -11 \\ 1 \end{pmatrix}$$

$$(x - 4)^3 + (x - 4)(x - 7) \cdot x + 99 \text{ coeffs}, x - 4 \rightarrow \begin{pmatrix} 99 \\ x^2 - 7 \cdot x \\ 0 \\ 1 \end{pmatrix}$$

### 2.2.6. Суми та добутки

Обчислення у символному вигляді скінченної або нескінченної суми або добутку, потребує наступних кроків.

1. Ввести вираз, використовуючи панель *Calculus* (Обчислення) для вставлення відповідних символів підсумовування або визначення добутку. При необхідності введіть як межу ряду символ нескінченності (клавіші  $\langle \text{Ctrl} \rangle + \langle \text{Shift} \rangle + \langle z \rangle$ ).

2. Залежно від бажаного стилю символічних обчислень виберіть команду *Symbolics/Simplify* (Символіка/Упростити) або введіть оператор визначення символічного результату.

Приклади числового і символічного обчислення сум та добутоків наведені в лістингах 2.9 і 2.10.

Лістинг 2.9. Символьні і числові розрахунки сум

$$\sum_{i=0}^{10} 2^i \rightarrow 2047 \qquad \sum_{i=0}^{\infty} a^i \rightarrow \frac{-1}{(a-1)}$$

$$\sum_{n=0}^{\infty} \frac{x^n}{2n!} \rightarrow \frac{1}{2} \cdot \exp(x) \qquad \sum_{n=0}^{100} \frac{n}{2n} \rightarrow \frac{101}{2}$$

Лістинг 2.10. Символьний розрахунок добутку

$$\prod_{n=1}^{\infty} \frac{1}{n^3 + 1} \rightarrow 0 \qquad \prod_{n=1}^{\infty} \sqrt{n} \rightarrow \infty$$

### 2.2.7. Розкладання на елементарні дроби (Convert to Partial Fractions)

Щоб розкласти складний дріб на елементарні дроби, слід або виконати команду *Symbolics/Variable/Convert to Partial Fractions* (Символіка/Змінна/ Розкласти на елементарні дроби) (рис. 2.11), або вказати ключове слово *parfrac* (лістинг 2.11). Застосовуючи перший спосіб (меню), не забувайте перед вибором його команди виділити змінну, відносно якої проводитиметься розкладання, а якщо використовується другий спосіб (з оператором визначення символічного результату), то ідентифікатор змінної слід вказати після ключового слова *parfrac*.

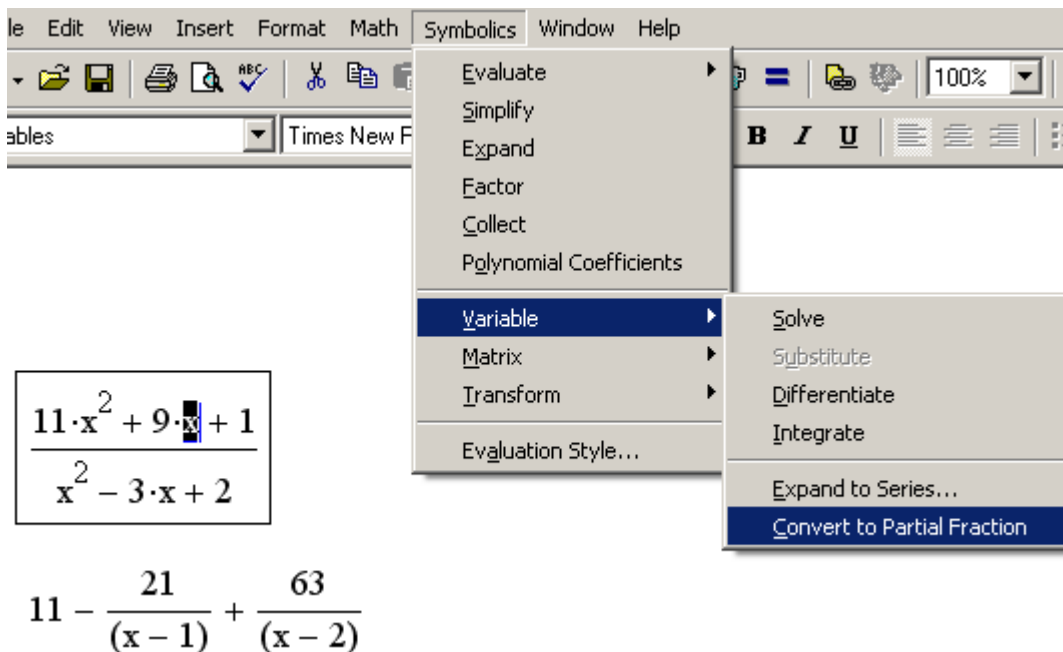


Рис.2.11. Розкладання складного дробу на елементарні дроби

Лістинг 2.11. Розкладання на елементарні дроби

$$\frac{11 \cdot x^2 + 9 \cdot x + 1}{x^2 - 3 \cdot x + 2} \text{ convert, parfrac, x} \rightarrow 11 - \frac{21}{(x - 1)} + \frac{63}{(x - 2)}$$

### 2.2.8. Заміна змінної (Substitute)

Дуже зручна можливість символічних обчислень – це операція заміни змінної у виразі. За допомогою меню заміна виконується так (рис. 2.12).

1. Введіть змінну (вираз), яку необхідно підставити в деякий вираз замість іншої (на рис. 2.12 це перший рядок документа).

2. Скопіюйте цю змінну (вираз) до буферу обміну, наприклад, натисненням клавіш *<Ctrl>* + *<C>* або кнопки *Copy* (Копіювати) на панелі інструментів *Standard* (Стандартна).

3. Виділіть у виразі, що підлягає зміні, змінну, яка замінюватиметься (у другому рядку на рис. 2.12 – виділена змінна *x*).

4. Виконайте команду *Symbolics/Variable/Substitute* (Символіка/Змінна/Підставити).

Результат цих дій ілюструється нижнім рядком документа на рис. 2.12.



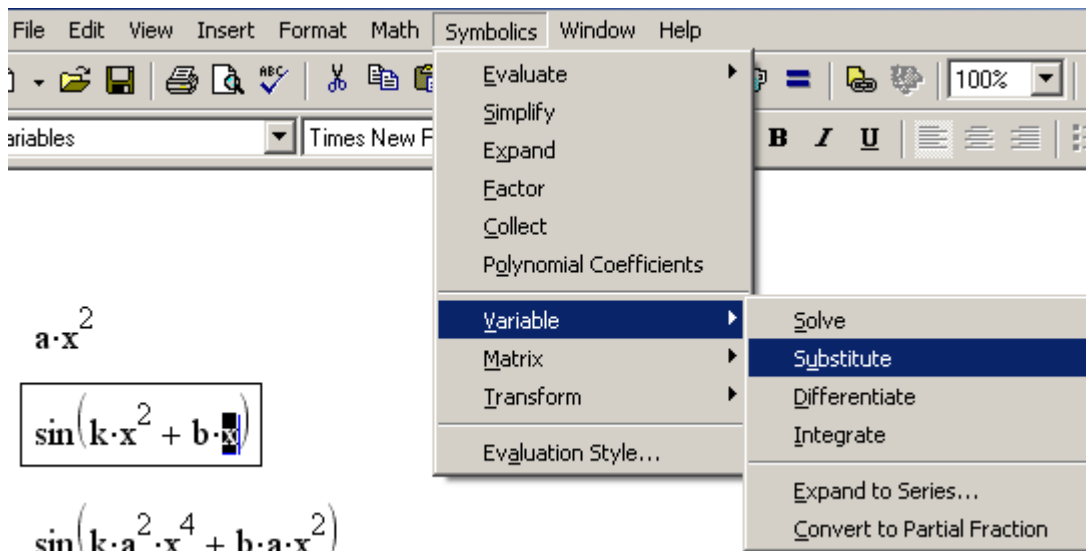


Рис. 2.12. Заміна змінної

Для здійснення тієї ж операції разом з оператором символьного результату використовуйте ключове слово *substitute*, яке вставляється в документ однойменною кнопкою на панелі *Symbolic* (Символіка). Після ключового слова *substitute* необхідно ввести в полі введення логічний вираз, що показує, яку саме змінну яким виразом слід замінити (лістинг 2.12).

Лістинг 2.12. Заміна змінної

$$\sin(k \cdot x^2 + b \cdot x) \text{ substitute } ,k = a \cdot x^2 \rightarrow \sin(a \cdot x^4 + b \cdot x)$$

### 2.2.9. Матрична алгебра

Символьний процесор *MATHCAD* дозволяє аналітично виконувати матричні обчислення. Пам'ятаючи про те, що більшість операцій і вбудованих функцій здійснюються над матрицями точно так, як і над звичайними числами, до матричних обчислень можна застосовувати розглянуту вище команду спрощення (*Simplify*) з меню символьних обчислень.

Крім того, є ряд специфічних матричних операцій, які можна організувати за допомогою пункту меню *Symbolics/Matrix* (Символіка/Матриця) або за допомогою декількох кнопок на панелі *Symbolic* (Символіка), призначених для операцій над матрицями (див. рис. 2.2). Це такі матричні операції:

- *Transpose* (Транспонування);
- *Invert* (Обернення матриці);
- *Determinant* (Визначення визначника).

Виконуються дії з матрицями в тій же послідовності, що і розглянуті символьні операції з скалярними змінними. Перед їх застосуванням не забувайте виділити у виразі матрицю, над якою виконуватиметься операція.

## 2.3. Математичний аналіз

Найбільш яскравим проявом можливостей символічного процесора в *MATHCAD* є аналітичні обчислення меж, похідних, інтегралів, розкладання в ряд, а також розв'язання рівнянь алгебри. Усі ці операції при виконанні їх за допомогою меню *Symbolics* (Символіка) знаходяться в його підменю *Variable* (Змінна). Для виконання цих операцій потрібно попередньо виділити у виразі змінну, відносно якої здійснюватиметься операція.

Усі перераховані операції можна здійснювати і за допомогою оператора визначення символічного результату.

### 2.3.1. Диференціювання (Differentiate)

Щоб аналітично продиференціювати вираз за певною змінною, виділіть в ньому цю змінну і виберіть команду *Symbolics/Variable/Differentiate* (Символіка/Змінна/Диференціювати) (рис. 2.13).

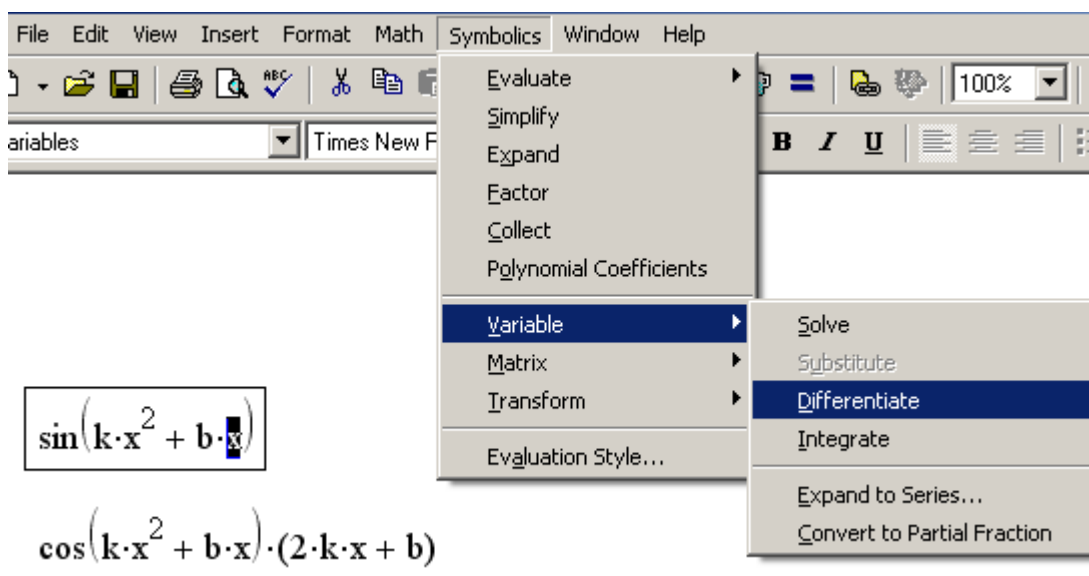


Рис. 2.13. Диференціювання за змінною

В результаті в наступному рядку після виразу з'явиться вираз його похідної. Для того, щоб знайти другу похідну, повторно застосуйте цю послідовність дій, але вже до отриманого результату диференціювання. Так шукають і похідні вищих порядків.

### 2.3.2. Інтегрування (Integrate)

Для обчислення невизначеного інтегралу від деякого виразу за певною змінною виділіть у виразі змінну і виконайте команду *Symbolics/Variable/Integrate* (Символіка/Змінна/Інтегрувати) (рис. 2.14). Обчислений аналітичний вираз невизначеного інтегралу з'явиться нижче. При цьому результат може містити як вбудовані в *MATHCAD* функції, так і інші спеціальні функції, які не можна

безпосередньо розрахувати в *MATHCAD*, але символічний процесор видає їх як результат деяких символічних операцій.

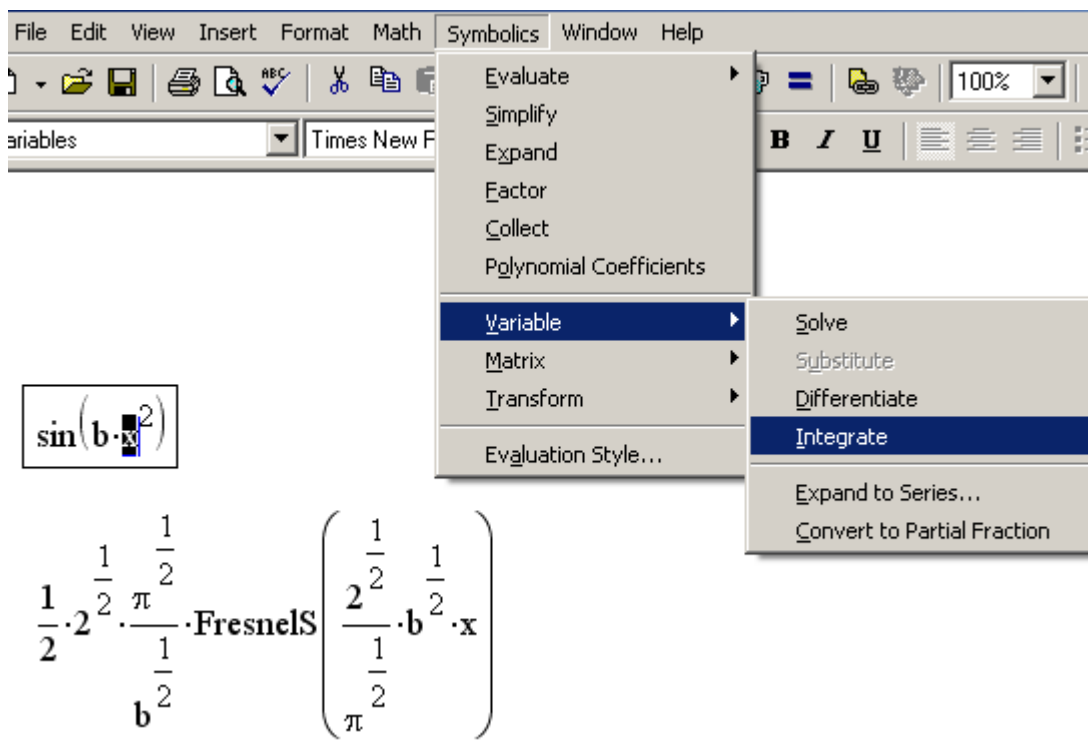


Рис. 2.14. Інтегрування за змінною

### 2.3.3. Розкладання в ряд (Expand to Series)

За допомогою символічного процесора *MATHCAD* можна розкласти вираз у ряд Тейлора за будь-якою змінною  $x$  в точці  $x = 0$ , тобто представити вираз в околі точки  $x$  сумою виду  $a_0 + a_1x + a_2x^2 + a_3x^3 + \dots$ . Тут  $a_i$  – деякі коефіцієнти, незалежні від  $x$ , але, можливо, залежні від інших змінних, що входять до початкового виразу. Якщо вираз має в точці  $x = 0$  особливість, то відповідне розкладання називають рядом Лорана.

Щоб розкласти вираз в ряд треба виконати наступні кроки.

1. Ввести вираз.
2. Виділити значення змінної, відносно якої потрібно отримати ряд.
3. Виконайте команду *Symbolics/Variable/Expand to Series* (Символіка/Змінна/Розкласти в ряд) (рис. 2.15).

4. У діалоговому вікні (рис. 2.16), що з'явиться, введіть бажану точність апроксимації (*Order of Approximation*) і натисніть кнопку *OK*.

Результат розкладання з'явиться під виразом (рис. 2.17).

Примітка.

Не забувайте, що розкладання виконується тільки в точці  $x = 0$ . Щоб отримати розкладання в іншій точці  $x = a$ , можна, наприклад, підставити замість змінної  $x$  вираз  $(x - a)$ .

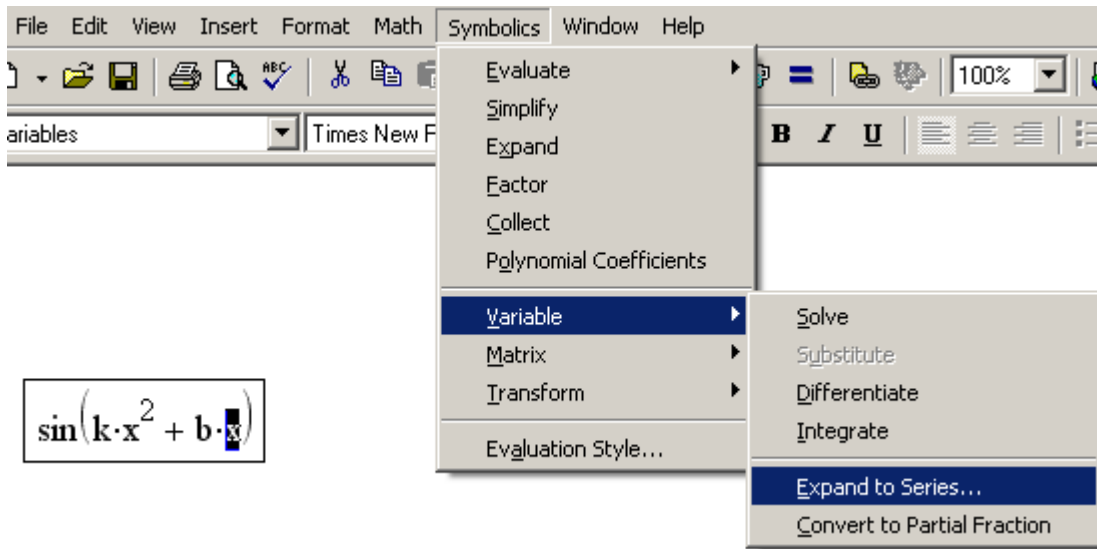


Рис. 2.15. Підготовка виразу для розкладання в ряд відносно змінної  $x$

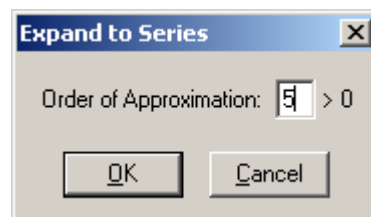


Рис. 2.16. Розкладання в ряд Тейлора

$$\sin(k \cdot x^2 + b \cdot x)$$

$$b \cdot x + k \cdot x^2 + \frac{-1}{6} \cdot b^3 \cdot x^3 + \frac{-1}{2} \cdot k \cdot b^2 \cdot x^4 + O(x^5)$$

Рис. 2.17. Результат розкладання в ряд Тейлора

Для розкладання в ряд альтернативним способом за допомогою оператора визначення символьного результату використовуйте ключове слово *series*, вставляючи його однойменною кнопкою панелі *Symbolic* (Символіка). Після ключового слова *series* через кому вказується ідентифікатор змінної, відносно якої проводиться розкладання, і точність апроксимації (лістинги 2.13 і 2.14). На рис. 2.18 задана функція  $y_0(x)$  та функції  $y_1(x)$  та  $y_2(x)$ , що утворені розкладанням функції  $y_0(x)$  у ряд Тейлора з різною кількістю членів ряду, та побудовані графіки цих функцій.

Лістинг 2.13. Розкладання виразу в ряд Тейлора

$$\sin(k \cdot x^2 + b \cdot x) \text{ series ,x,2} \rightarrow b \cdot x$$

$$\sin(k \cdot x^2 + b \cdot x) \text{ series ,x,3} \rightarrow b \cdot x + k \cdot x^2$$

$$\sin(k \cdot x^2 + b \cdot x) \text{ series ,x,4} \rightarrow b \cdot x + k \cdot x^2 + \frac{-1}{6} \cdot b^3 \cdot x^3$$

$$\sin(k \cdot x^2 + b \cdot x) \text{ series ,x,5} \rightarrow b \cdot x + k \cdot x^2 + \frac{-1}{6} \cdot b^3 \cdot x^3 + \frac{-1}{2} \cdot k \cdot b^2 \cdot x^4$$

Лістинг 2.14. Розкладання виразу в ряд за різними змінними

$$\sin(k \cdot x^2 + b \cdot x) \text{ series ,k,3} \rightarrow \sin(b \cdot x) + \cos(b \cdot x) \cdot x^2 \cdot k + \frac{-1}{2} \cdot \sin(b \cdot x) \cdot x^4 \cdot k^2$$

$$\sin(k \cdot x^2 + b \cdot x) \text{ series ,b,3} \rightarrow \sin(k \cdot x^2) + \cos(k \cdot x^2) \cdot x \cdot b + \frac{-1}{2} \cdot \sin(k \cdot x^2) \cdot x^2 \cdot b^2$$

$$k := 0.1 \quad b := 0.1$$

$$y0(x) := \sin(k \cdot x^2 + b \cdot x) \quad y1(x) := b \cdot x + k \cdot x^2 + \frac{-1}{6} \cdot b^3 \cdot x^3$$

$$y2(x) := b \cdot x + k \cdot x^2 + \frac{-1}{6} \cdot b^3 \cdot x^3 + \frac{-1}{2} \cdot k \cdot b^2 \cdot x^4$$

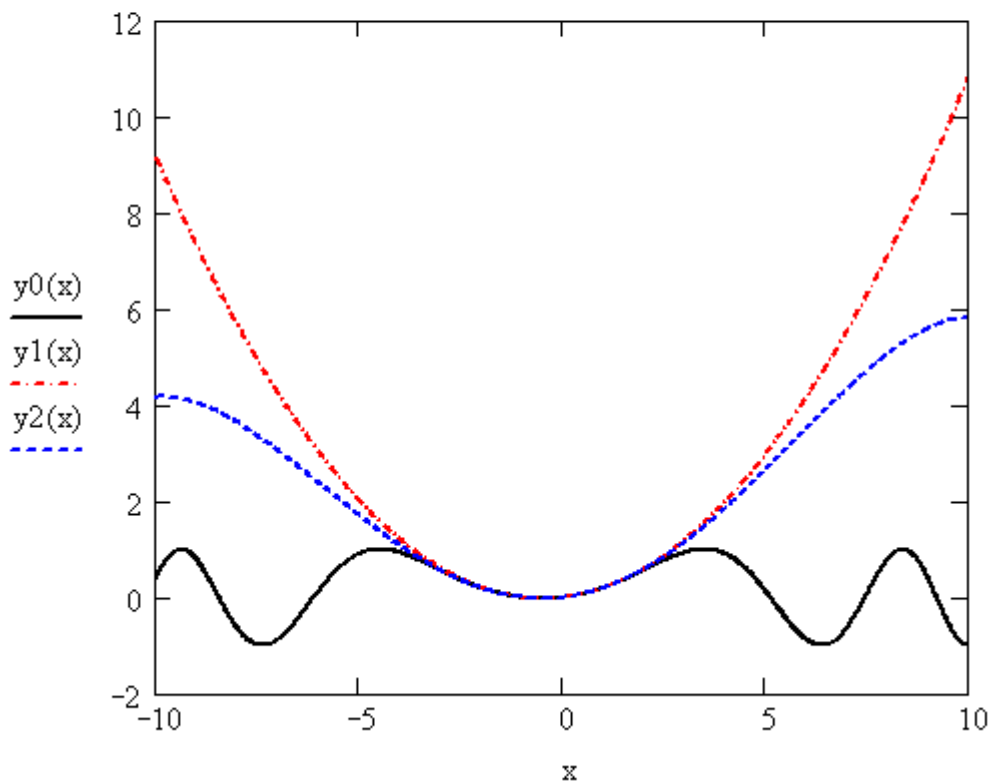


Рис. 2.18. Графіки функцій та рядів на які вона розкладена

### 2.3.4. Розв'язання рівнянь (Solve)

За допомогою символічного процесора можна обчислити аналітично значення змінної, при якому вираз перетворюється на нуль. Для цього треба виконати наступні дії.

1. Введіть вираз.
2. Виділіть змінну, відносно якої розв'язуватиметься рівняння.
3. Виберіть в меню *Symbolics* (Символіка) пункт *Variable/Solve* (Змінна/Розв'язати) (рис. 2.19).

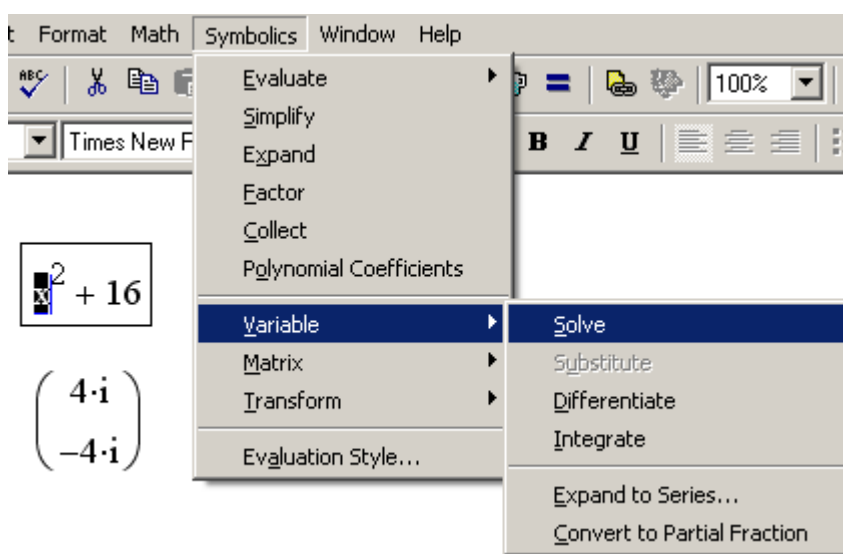


Рис. 2.19. Символьне розв'язання рівняння

### 2.4. Інтегральні перетворення

Згідно з визначенням інтегральні перетворення ставлять у відповідність деякій функції  $f(x)$  іншу функцію від іншого аргументу  $F(\omega)$ . Причому ця відповідність  $f(x) \rightarrow F(\omega)$  задається інтегральною залежністю. Символьний процесор *MATHCAD* дозволяє здійснювати три види інтегральних перетворень функцій – перетворення Фур'є, Лапласа і Z-перетворення. Разом з прямими перетвореннями є можливість здійснювати будь-яке з цих трьох зворотних перетворень, тобто  $F(\omega) \rightarrow f(x)$ .

Усі символічні інтегральні перетворення аналогічно вже розглянутим операціям. Для обчислення перетворення виразу виділяється змінна, відносно якої здійснюватиметься перетворення, і потім вибирається відповідний пункт меню. Перетворення із застосуванням оператора визначення символічного результату використовується з одним із відповідних ключових слів, після якого потрібно вказати ідентифікатор змінної.

Приведемо приклади символічного визначення кожного з трьох інтегральних перетворень.

### 2.4.1. Перетворення Фур'є (Fourier)

Перетворення Фур'є представляє функцію  $f(x)$  у вигляді інтеграла Фур'є. Аналітичний розрахунок перетворення Фур'є за допомогою меню показаний на рис. 2.20. У лістингу 2.15 приведено два приклади обчислення прямого перетворення Фур'є із застосуванням ключового слова `fourier` і оператора визначення символічного результату  $\rightarrow$ . Лістингом 2.16 ілюструється зворотне перетворення Фур'є однієї з функцій попереднього лістингу.

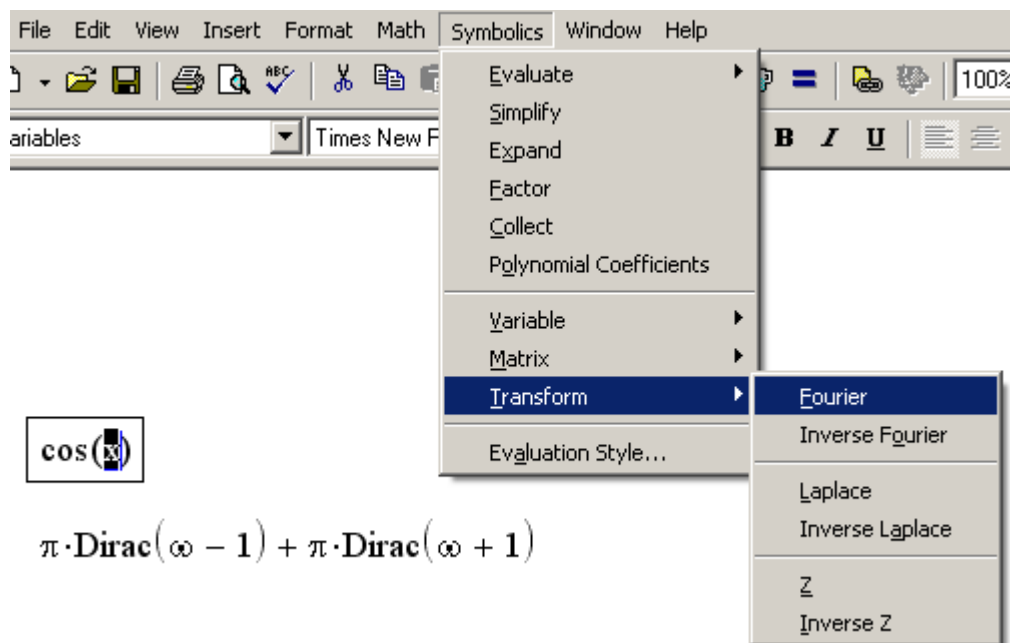


Рис. 2.20. Розрахунок Фур'є-перетворення за допомогою меню

#### Примітка

У *MATHCAD* перетворення Фур'є можна обчислити і за допомогою числового процесора, що використовує популярний алгоритм "Швидке Перетворення Фур'є".

Вставити функцію дірака

Лістинг 2.15. Пряме перетворення Фур'є

$$\begin{aligned} \cos(x) \text{ fourier, } x &\rightarrow \pi \cdot \text{Dirac}(\omega - 1) + \pi \cdot \text{Dirac}(\omega + 1) \\ x^2 + 4 \text{ fourier, } x &\rightarrow -2 \cdot \pi \cdot \text{Dirac}(2, \omega) + 8 \cdot \pi \cdot \text{Dirac}(\omega) \end{aligned}$$

Лістинг 2.16. Зворотне перетворення Фур'є

$$-2 \cdot \pi \cdot \text{Dirac}(2, \omega) + 8 \cdot \pi \cdot \text{Dirac}(\omega) \text{ invfourier, } \omega \rightarrow t^2 + 4$$

### 2.4.2. Перетворення Лапласа (Laplace)

Перетворенням Лапласа називають інтеграл від  $f(t)$  виду:

$$\int_0^{\infty} f(t) e^{-st} dt.$$

Перетворення Лапласа виконується аналогічно перетворенню Фур'є (див. попередній розділ). Приклади перетворення Лапласа наведені в лістингу 2.17.

Лістинг 2.17. Пряме і зворотне перетворення Лапласа

$$t^2 + 4 \text{ laplace } ,x \rightarrow \frac{2}{s^3} + \frac{4}{s}$$

$$\frac{2}{s^3} + \frac{4}{s} \text{ invlaplace } ,s \rightarrow t^2 + 4$$

### 2.4.3. Z-перетворення (Z)

Z-перетворення  $F(z)$  функції  $f(x) \Big|_{x=nT}$  визначається так:

$$F(z) = \sum_{k=-\infty}^{\infty} f(nT) z^{-n}.$$

Приклад Z-перетворення приведений в лістингу 2.18.

Лістинг 2.18. Пряме і зворотне Z-перетворення

$$t^2 + 4 \text{ ztrans } ,x \rightarrow z \cdot \frac{(-7 \cdot z + 5 + 4 \cdot z^2)}{(z - 1)^3}$$

$$z \cdot \frac{(-7 \cdot z + 5 + 4 \cdot z^2)}{(z - 1)^3} \text{ invztrans } ,z \rightarrow n^2 + 4$$

## 2.5. Додаткові можливості символьного процесора

Вище були розглянуті основні прийоми символьних обчислень в *MATHCAD*. Вони, як правило, були показані на простих прикладах, що ілюстрували ту або іншу символьну операцію. Проте, при проведенні різноманітних (і числових також) розрахунків в *MATHCAD* можливості символьного процесора можна використовувати ефективніше. Розглянемо деякі з них.

### 2.5.1. Застосування функцій користувача

При проведенні символьних обчислень з оператором визначення символьного результату функції користувача і змінні, визначені раніше в документі *MATHCAD*, сприймаються символьним процесором коректно. Отже, є могутній апарат застосування символьних розрахунків у програмах користувача. Приклади застосування функції користувача наведені в лістингах 2.19 і 2.20.



Порівняйте останні рядки цих лістингів. Не зважаючи на їх ідентичність зліва від оператора визначення символьного результату, результати праворуч від нього відрізняються. Це пов'язано з тим, що в лістингу 2.20 змінній  $x$  заздалегідь надано значення 4. Оскільки значення змінних впливають на символьні обчислення, то результат враховує числове значення  $x$ .

Лістинг 2.19. Застосування функції користувача в символьних обчисленнях

$$f(k, x) := \cos(k \cdot x) + 4x^{2-k}$$

$$f(k, x) \text{ substitute } ,k = \sqrt{x} \rightarrow \cos\left(\frac{3}{x^2}\right) + 4 \cdot x^{\left(2 - \frac{1}{2}\right)}$$

$$f(k, x) \text{ series } ,k, 2 \rightarrow 1 + 4 \cdot x^2 + -4 \cdot x^2 \cdot \ln(x) \cdot k$$

Лістинг 2.20. Значення змінних впливають на результат символьних обчислень

$$f(k, b, x) := \sin(k \cdot x^2 + b \cdot x)$$

$$x := 5$$

$$f(k, b, x) \text{ series } ,k, 2 \rightarrow \sin(5 \cdot b) + 25 \cdot \cos(5 \cdot b) \cdot k$$

Однак, при здійсненні символьних операцій через меню *Symbolics* (Символіка) символьний процесор "не бачить" нічого, крім виразу, над яким має виконуватись символьна операція. Тому ні функції користувача, ні заздалегідь надані значення змінним не впливають на обчислення.

### Примітка

Використовуйте меню *Symbolics* (Символіка), якщо потрібно виконати аналітичні дії з виразом і отримати відповідь у загальному вигляді, що не враховує поточні значення змінних, що входять у вираз.

### 2.5.2. Набуття виразом числового значення

За допомогою символьного процесора можна розрахувати числове значення виразу (дійсне або комплексне). Іноді такий шлях є зручнішим, ніж застосування числового процесора (тобто з використанням знаку звичайної рівності). Щоб розрахувати значення деякого виразу (рис. 2.21), виберіть команду *Symbolics/Evaluate/Symbolically* (Символіка/Обчислити/Символьно) або пункт *Symbolics/Evaluate/Floating Point* (Символіка/Обчислити/ З крапкою, що плаває). У останньому випадку Вам буде запропоновано за допомогою діалогу *Floating Point Evaluation* (Обчислення з крапкою, що плаває) задати точність результату. У результаті застосування даних команд *MATHCAD* замінює символьні результати, де це можливо, значеннями у вигляді чисел з крапкою, що плаває.

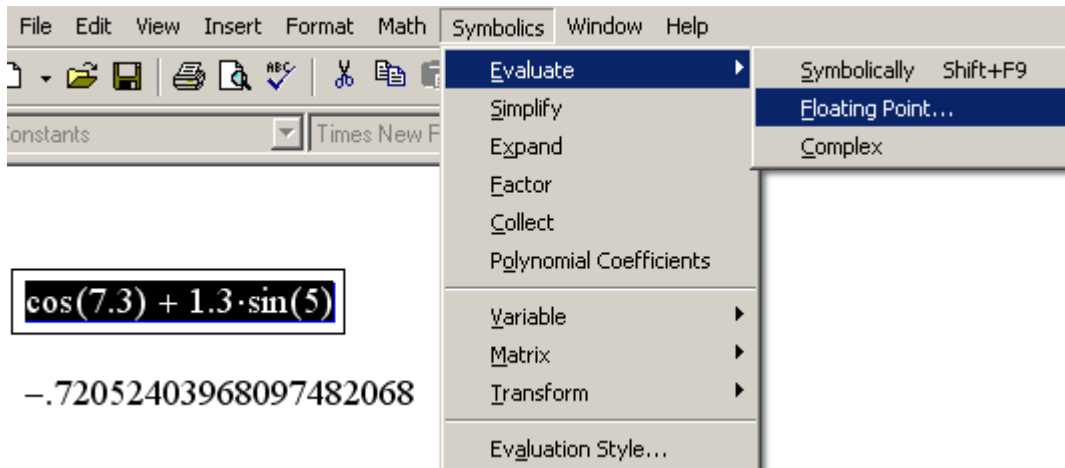


Рис. 2.21 . Обчислення виразу з крапкою, що плаває

Ще один пункт меню *Symbolics/Evaluate/Complex*

(Символіка/Обчислити/Комплексно) дозволяє подати вираз у форматі комплексного числа, тобто у вигляді  $a + bi$ .

Аналогічні за дією ключові слова *float* і *complex* можна використовувати в документах, вводячи їх з панелі *Symbolic* (Символіка). Ключове слово *float* застосовується разом із значенням точності виведення результату з крапкою, що плаває (лістинг 2.21). За допомогою оператора *complex* можна перетворювати вирази як в символічному вигляді, так і з урахуванням числових значень, якщо вони були раніше надані змінним (декілька прикладів приведено в лістингу 2.22).

Лістинг 2.21. Обчислення виразу з крапкою, що плаває

$$\begin{aligned}
 x &:= 5 \quad k := 2.5 \\
 \cos(k \cdot x) + 4x^{2-k} \text{ float}, 3 &\rightarrow 2.79 \\
 \cos(k \cdot x) + 4x^{2-k} \text{ float}, 10 &\rightarrow 2.786652661 \\
 \cos(k \cdot x) + 4x^{2-k} \text{ float}, 25 &\rightarrow 2.7866526611784124209
 \end{aligned}$$

Лістинг 2.22. Комплексні перетворення виразів

$$\begin{aligned}
 e^{z+2i} \text{ complex} &\rightarrow \exp(z) \cdot \cos(2) + i \cdot \exp(z) \cdot \sin(2) \\
 \ln(e^{2-5i}) \text{ complex} &\rightarrow \frac{1}{2} \cdot \ln(\exp(2)^2 \cdot \cos(5)^2 + \exp(2)^2 \cdot \sin(5)^2) + i \cdot (-5 + 2 \cdot \pi)
 \end{aligned}$$

### 2.5.3. Послідовності символічних команд

Символьні обчислення допускається виконувати із застосуванням ланцюга ключових слів. Для цього ключові слова, що відповідають послідовності символічних операцій, повинні бути введені по черзі з панелі *Symbolic* (Символіка). Принцип організації ланцюгів символічних обчислень дуже

схожий на застосування вбудованої мови програмування *MATHCAD*. Декілька прикладів використання послідовності символьних операторів наводиться в лістингах 2.23 і 2.24.

#### Примітка

Послідовності символьних команд допускають введення додаткових умов в розрахунки, наприклад таких, як обмеження на дійсну або комплексну форму результату. Це робиться за допомогою ключового слова *assume*. Докладнішу інформацію читач знайде в довідковій системі *MATHCAD*.

Лістинг 23. Перетворення Фур'є, розкладання в ряд із заданою точністю

$$\begin{array}{l}
 e^{x^2} \text{ fourier, } x \rightarrow 2 \cdot \pi \cdot \text{Dirac}(\infty) \\
 e^{x^2} \left| \begin{array}{l} \text{fourier, } x \\ \text{series, } \infty, 2 \end{array} \right. \rightarrow 2 \cdot \pi \cdot \text{Dirac}(0) + 2 \cdot \pi \cdot \text{Dirac}(1, 0) \cdot \infty \\
 e^{x^2} \left| \begin{array}{l} \text{fourier, } x \\ \text{series, } \infty, 2 \\ \text{float, } 3 \end{array} \right. \rightarrow 6.28 \cdot \text{Dirac}(0) + 6.28 \cdot \text{Dirac}(1, 0) \cdot \infty
 \end{array}$$

Лістинг 24. *Z*-перетворення і розкладання на прості дроби

$$\begin{array}{l}
 x^2 + 4 \text{ ztrans, } x \rightarrow z \cdot \frac{(-7 \cdot z + 5 + 4 \cdot z^2)}{(z - 1)^3} \\
 x^2 + 4 \left| \begin{array}{l} \text{ztrans, } x \\ \text{convert, parfrac, } z \end{array} \right. \rightarrow 4 + \frac{2}{(z - 1)^3} + \frac{3}{(z - 1)^2} + \frac{5}{(z - 1)}
 \end{array}$$

## Список рекомендованой літератури

1. Гурский Д.А. Вычисления в Mathcad.- Минск: Новое Знание,2003.-814с.
2. Дьяконов В. Mathcad 2000р.:учебный курс.-СПб.: Питер, 2001.-592с.
3. Мудров А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран и Паскаль.-Томск: МП «РАСКО», 1991.-272с.
4. Плис А.И., Сливина Н.А. Mathcad: математический практикум для экономистов и инженеров: Учеб. пособие.- М.: Финансы и статистика, 1999.-656с.

## Зміст

Вступ.....	3
<i>Лабораторна робота №1</i>	
ОСНОВИ РОБОТИ З МАТНСАД. ПОБУДОВА ВИРАЗІВ ОБЧИСЛЕННЯ .....	4
<i>Лабораторна робота №2</i>	
ЗАСТОСУВАННЯ ПРОГРАМУВАННЯ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНИХ ЗАДАЧ .....	11
<i>Лабораторна робота №3</i>	
СИМВОЛЬНІ ОБЧИСЛЕННЯ.....	18
<i>Лабораторна робота №4</i>	
ДВОВИМІРНА ГРАФІКА.....	26
<i>Лабораторна робота №5</i>	
ТРИВИМІРНА ГРАФІКА.....	32
<i>Лабораторна робота №6</i>	
ФОРМУВАННЯ РУХОМИХ ГРАФІЧНИХ ЗОБРАЖЕНЬ.....	39
<i>Лабораторна робота №7</i>	
ОПЕРАЦІЇ З МАТРИЦЯМИ І ВЕКТОРАМИ.....	42
<i>Лабораторна робота №8</i>	
ОПЕРАЦІЇ НАД ПОЛІНОМАМИ.....	49
<i>Лабораторна робота №9</i>	
РОЗВ'ЯЗАННЯ СИСТЕМ ЛІНІЙНИХ АЛГЕБРИЧНИХ РІВНЯНЬ.....	52
<i>Лабораторна робота №10</i>	
РОЗВ'ЯЗАННЯ СИСТЕМ НЕЛІНІЙНИХ АЛГЕБРИЧНИХ РІВНЯНЬ....	55
<i>Лабораторна робота №11</i>	
РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ ТА ЇХ СИСТЕМ.....	58
<i>Додатки.....</i>	63
Список рекомендованої літератури.....	100