ЛАБОРАТОРНА РОБОТА №1

Знайомство з програмою MathCAD

1. Виконати такі обчислення:

1.1.

$$2+5 \qquad 3\cdot 7 \qquad \frac{128}{4} \qquad 7+5^2-2^3 \qquad (2\cdot\pi+10)\cdot 2-\frac{4}{5}$$

1.2.

$x := 3$ $\qquad z := 7 \qquad w := 2x$

$y(x) := 5x^3 + 8\cdot x^2 - 25x$ $\qquad s(z,w) := 11z - w\cdot z + 7\frac{w}{z}$

$y(x) =$ $\qquad s(z,w) =$

1.3.

$k := 1..100$

$$\sum_{n=1}^{10} \frac{1}{n^2} = \qquad \sum_{k} \frac{1}{(2\cdot k-1)\cdot(2\cdot k+1)} =$$

1.4.

$t := -10..10$

$f(t) := \sin(t)$ Побудувати графік залежності f(t)

$f(t) =$

1.5.

$f1(t) := \cos(t)$

$f2(t) := \sin(t) + \cos(t)$ Побудувати графіки залежностей f, f1, f2 від t

$f3(t) := \frac{\sin(t)}{t}$ Побудувати графік залежності f3(t)

2. Виконати обчислення у символьній формі

2.1.

$$\int \sin(x)\,dx \rightarrow \qquad \int \sin(x)\cdot\cos(k\cdot x)\,dx \rightarrow \qquad \int_{-\pi}^{\pi} \frac{\sin(x)}{x}\,dx \rightarrow$$

$$\frac{d}{dx}\sin(x) \rightarrow \qquad \frac{d}{dx}x^{n} \rightarrow \qquad \frac{d}{dx}\cos(x) \rightarrow$$

2.2.

$$(a+b)^3 \text{ expand},a$$

$$ax^2 + bx + c \text{ solve},x \rightarrow$$

$$3\cdot a\cdot x^2 + 6\cdot(a\cdot x)^2 - x^5 + 4\cdot c x^3 - 25\cdot x^5 \text{ collect},x \rightarrow$$

2.3.

$\frac{\sin(x)}{x}$ has Fourier transform

Побудувати графіки залежностей вихідної функції та її Фур'є перетворення від x

3. Розв'язати рівняння

3.1. Рівняння виду f(x) = 0

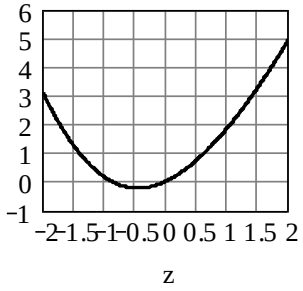
$$f(z) := z^2 + \sin(z)$$

$$z := 3$$

$$y := \text{root}(f(z),z)$$

$$y = 5.072 \times 10^{-4}$$

f(z)



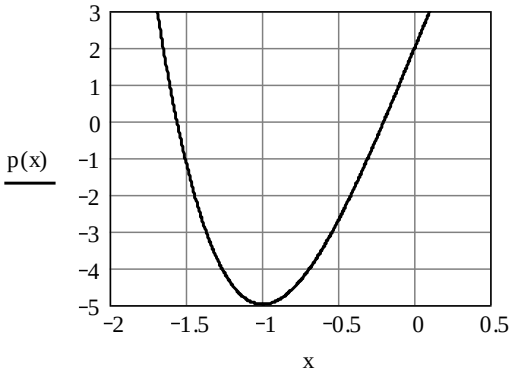
3.2.Пошук коренів многочлена

$$p(x) := \left(x^4 - 2\cdot x^3 + 10\cdot x + 2\right)$$

$x^4 - 2\cdot x^3 + 10\cdot x + 2$ has coefficients $\begin{pmatrix} 2 \\ 10 \\ 0 \\ -2 \\ 1 \end{pmatrix}$

$$v := \begin{pmatrix} 2 \\ 10 \\ 0 \\ -2 \\ 1 \end{pmatrix} \qquad r := \text{polyroots}(v)$$

$$r = \begin{pmatrix} -1.564 \\ -0.202 \\ 1.883 - 1.67i \\ 1.883 + 1.67i \end{pmatrix}$$



3.3. Розв’язування СЛАУ

$$x := 1 \qquad y := 1 \qquad z := 0$$

Given

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

$$y^3 + 4\cdot z = 2 - x$$

$$x + z = e^z$$

$$s := \text{find}(x, y, z) \qquad s = \begin{pmatrix} 1.168 \\ 1.504 \\ -0.642 \end{pmatrix}$$

4. Ранжировані змінні

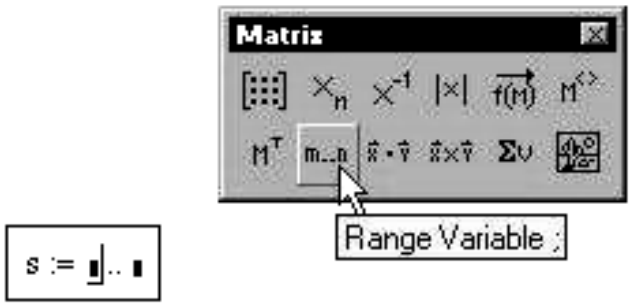
Ранжировані змінні MathCAD є різновидом векторів і призначені, головним чином, для створення циклів або ітераційних обчислень. Найпростіший приклад ранжированной змінної - це масив з числами, які у деякому діапазоні з деяким кроком. Наприклад, для створення ранжированої змінної s з елементами 0,1,2,3,4,5:

1. Помістіть курсор введення в потрібне місце документа.

2. Введіть назву змінної (s) та оператор присвоювання ":".

3. На панелі **Matrix** (Матриця) натисніть кнопку **Range Variable** (Ранжирована змінна) або введіть символ точки з комою з клавіатури.

4. У місцезаповнювачі, що з'явилися, введіть ліву і праву межі діапазону зміни ранжированої змінної про і 5. Результат створення ранжированої змінної показаний на малюнку.



Створення ранжированої змінної

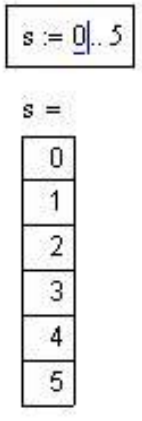
Щоб створити ранжовану змінну з кроком, не рівним 1, наприклад, 0,2, 4, 6, 8:

1. Створіть ранжировану змінну в діапазоні від 0 до 8.

2. Поставте лінії введення на початок діапазону (0).

3. Введіть кому.

4. У місцезаповнювач, що з'явився, введіть значення кроку зміни ранжированої змінної (2). Створена ранжирована змінна матиме значення від 0 до 8 включно, з кроком, рівним 2.



Висновок ранжированої змінної з кроком, не рівним 1

Створення ранжированої змінної

Найчастіше ранжировані змінні використовуються:

- при паралельних обчисленнях;

- для присвоєння значень елементам інших масивів

Більшість математичних дій, реалізованих у MathCAD, здійснюються над ранжированими змінними так само, як над звичайними числами. У цьому випадку одна й та сама дія здійснюється паралельно над усіма елементами ранжированої змінної.

$i := 0, 2 .. 8$

$s(i) := i^2 + 1$

i =	s(i) =	sin(s(i)) =
0	1	0.841
2	5	-0.959
4	17	-0.961
6	37	-0.644
8	65	0.827

Ранжирована змінна при паралельних обчисленнях

$x := 0 .. 10$

$x =$

0
1
2
3
4
5
6
7
8
9
10

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

$f(x) =$

-52
-31.5
-15
-2.5
6
10.5
11
7.5
0
-11.5
-27

$x := 1 , 1.1 .. 1.8$

$x =$

1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8

$data := -10 , -8 .. 0$

$data =$

-10
-8
-6
-4
-2
0

$n := 202 , 192 .. 102$

$n =$

202
192
182
172
162
152
142
132
122
112
102

5. Комплексні числа

Комплексні числа вводяться у звичайному запису алгебри, в якості уявної одиниці використовується символ i або j.

$x := 1 \qquad y := 1$

Примітка: ***не можна просто запровадити i, потрібно написати 1i.***

$a := 2 + 3i \qquad b := -1 + 4j \qquad c := a + b \qquad c = 1 + 7i$

$c := a - b \qquad c = 3 - i$

$a \cdot b = -14 + 5i \qquad \frac{a}{b} = 0.588 - 0.647i$

$\bar{a} = 2 - 3i \qquad \bar{b} = -1 - 4i$ Комплексне сполучення, що виводиться символом подвійних лапок після набору імені змінної ".

$e^i = 0.54 + 0.841i \qquad \sqrt{-1} = i$

$\sin(i) = 1.175i \qquad \sqrt[3]{-1} = -1$

У разі багатозначності коренів система поверне корінь із найменшою уявною частиною.

$\cos(i) = 1.543 \qquad \sqrt[6]{-1} = 0.866 + 0.5i$ або так $\sqrt[6]{-1} \to (-1)^{\frac{1}{6}}$

Функції до роботи з комплексними числами:

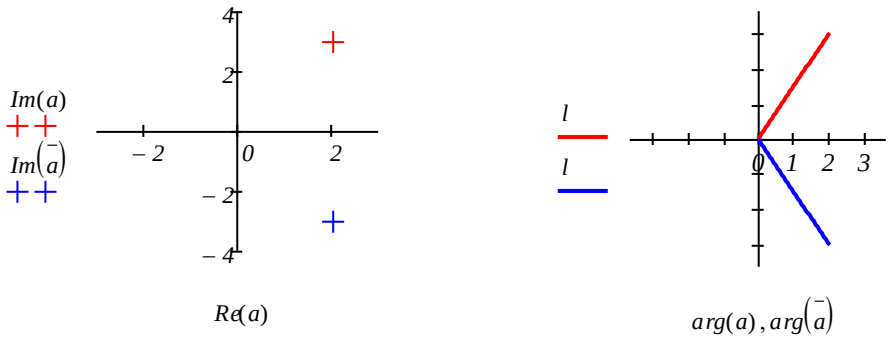
$\mathrm{Re}(z)$ –дійсна частина числа,

$\mathrm{Im}(z)$ –уявна частина числа.

$\arg(z)$ –аргумент (кут у комплексній площині між речовинною віссю та z)

$|z|$ –модуль $|a| = 3.606$

Розглянемо графічне уявлення комплексного числа на декартовому та полярному графіках.



У нашому випадку:

$\mathrm{Re}(a) = 2\ \mathrm{Im}(a) = 3 \qquad |a| = 31.606 \qquad arg(a) = 0.983$

$\mathrm{Re}(\bar{a}) = 2 \qquad \mathrm{Im}(\bar{a}) = -3 \qquad |\bar{a}| = 3.606 \qquad \mathrm{arg}(\bar{a}) = -0.983$

Дуже часто рівняння повертають в якості кореня комплексні числа. Найпростіший випадок – це корені полінома. Наприклад, позначимо через M матрицю коефіцієнтів полінома 4-го порядку. Корені полінома знайдемо за допомогою вбудованої функції. Як видно, всі корені комплексні. Змінюючи значення коефіцієнтів полінома, можна спостерігати, як змінюються корені.

$$M := 3m^4 - m^3 + 2m^2 + 1\,coeffs, m \rightarrow \begin{pmatrix} 1 \\ 0 \\ 2 \\ -1 \\ 3 \end{pmatrix} \qquad polyroots(M) = \begin{pmatrix} -0.286 - 0.614i \\ -0.286 + 0.614i \\ 0.453 + 0.722i \\ 0.453 - 0.722i \end{pmatrix}$$

Тут для виділення коефіцієнтів полінома ми скористалися операцією coeffs з меню символічної палітри.

ЛАБОРАТОРНА РОБОТА №2

ПРОГРАМУВАННЯ В СИСТЕМІ MATHCAD.

1 Мета роботи: Отримання практичних навичок роботи з пакетом MATHCAD та використання його для спектрального аналізу періодичних сигналів.

2 Опис лабораторного макета: У ході лабораторної роботи використовується інструментарій пакету MATHCAD 12.

3 Короткі довідкові дані

Для вставки програмного коду в документи в MathCAD є спеціальна панель інструментів **Programming** (Програмування), яку можна викликати на екран натисканням кнопки **Programming Toolbar** на панелі Math (Математика), як показано на рис.1. більшість кнопок цієї панелі виконано у вигляді текстового представлення операторів програмування, тому їх зміст легко зрозумілий.

Викладемо послідовно основні складові мови програмування MathCAD і розглянемо приклади його використання.

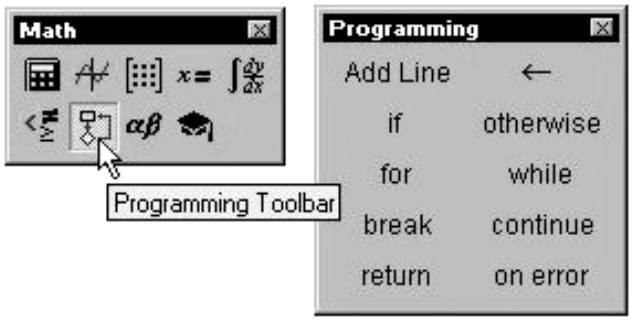


Рис. 1. Панель інструментів **Programming**

Основними інструментами роботи в MathCAD є математичні вирази, змінні та функції. Нерідко записати формулу, яка використовує ту чи 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 та функцій у кілька рядків, часто із застосуванням специфічних програмних операторів.

Програмний модуль позначається в MathCAD вертикальною рисою, справа від якої послідовно записуються оператори мови програмування (рис.2).

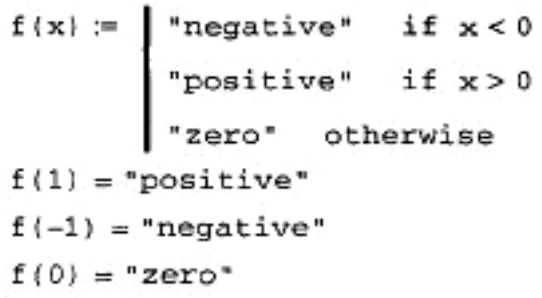


Рис. 2. Функція умови, визначена за допомогою програми

3.1. Оператори панелі інструментів Programming

3.1.1. Створення програми (Add Line) - створює і при необхідності розширює жирну вертикальну лінію, праворуч від якої в шаблонах задається запис програмного блоку

Щоб створити програмний модуль, наприклад, представлений на рис.2:

1. Введіть частину виразу, яка перебуватиме ліворуч від знака присвоювання і сам знак присвоювання. У прикладі це ім'я функції f(x).

2. За потреби викличте панель інструментів **Programming** на екрані (див. рис. 1).

3. Натисніть кнопку **Add Line** (Додати лінію) на цій панелі.

4. Якщо приблизно відомо, скільки рядків коду буде містити програма, можна створити потрібну кількість ліній повторним натисканням кнопки **Add Line** (Додати лінію) відповідне число разів (на рис. 3 показаний результат триразового натискання).

5. У місцезаповнювачі введіть бажаний програмний код, використовуючи програмні оператори. У аналізованому прикладі в кожен місцезаповнювач вводиться рядок, наприклад, "positive" (рис. 4), потім натискається кнопка **If** (Якщо) на панелі **Programming** і в місцезаповнювач, що виник, вводиться вираз x> 0 (рис. 5).

Після того як програмний модуль повністю визначений, і жоден місцезаповнювач не залишився порожнім, функція може використовуватися звичайним чином, як у чисельних, так і в символьних розрахунках.

Не вводьте імена програмних операторів з клавіатури. Для їх вставки можна застосовувати лише сполучення клавіш, які наведені в тексті підказки (рис. 3 і 4)

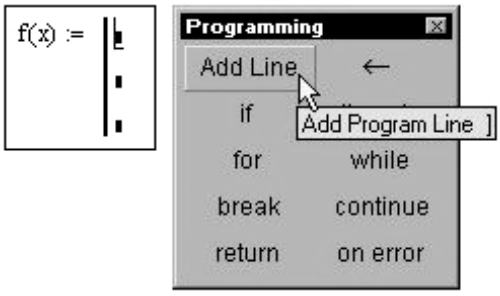


Рис. 3. Початок створення програмного модуля

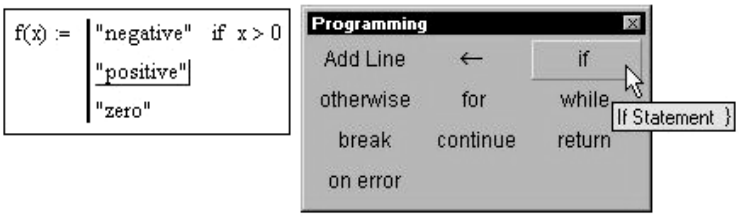


Рис. 4. Вставка програмного оператора

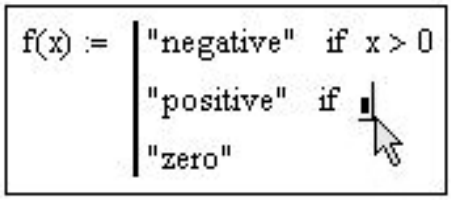


Рис. 5. Вставлення умови в програму

Вставити рядок програмного коду у вже створену програму можна в будь-який момент за допомогою тієї ж кнопки **Add Line** (Додати лінію). Для цього слід попередньо помістити на потрібне місце усередині програмного модуля лінії введення. Наприклад, розташування лінії введення на рядку, показаної на рис. 6, призведе до появи нової лінії з місцем-заповнювачем перед цим рядком. Якщо пересунути вертикальну лінію введення з початку рядка (як на рис. 6) на її кінець, то нова лінія з'явиться після рядка. Якщо виділити рядок не цілком, а лише деяку її частину (рис. 7), то це вплине на положення в програмі нового рядка коду (результат натискання кнопки **Add Line** показано на рис. 8).

Нарада *Не забувайте, що для бажаного розміщення ліній введення всередині формули можна використовувати не тільки мишу та клавіші зі стрілками, а й пробіл. За допомогою послідовних натискань пробілу лінії введення "захоплюють" різні частини формули.*

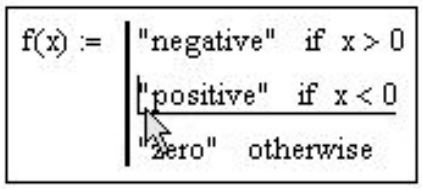


Рис. 6. Вставка нового рядка в існуючу програму

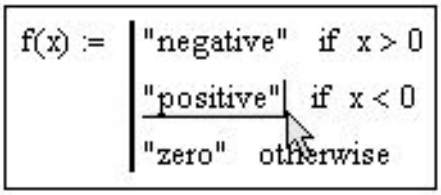


Рис. 7. Положення ліній введення впливає на становище нової лінії

Для чого може знадобитися вставка нової лінії в положення, показане на рис. 8. Нова вертикальна риса з двома лініями виділяє фрагмент програми, який відноситься до умови x> 0, що знаходиться в його заголовку. Приклад можливого подальшого програмування показано на рис.9

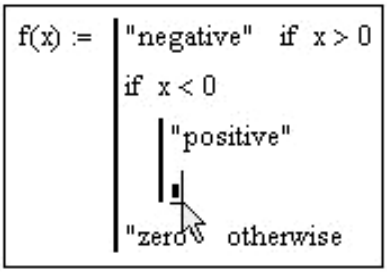


Рис. 8. Результат вставки нової лінії в програму (з положення рис. 7)

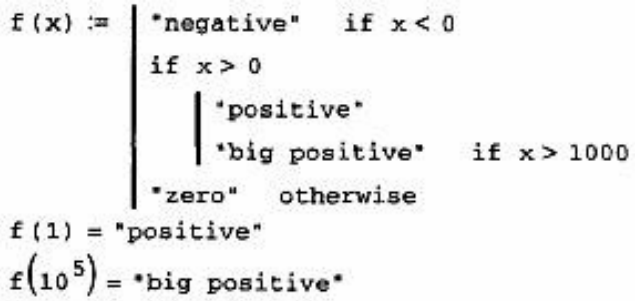


Рис. 9. Приклад удосконалення програми

У режимі виконання програми, а це відбувається при будь-якій спробі обчислити f (x), виконується послідовно кожен рядок коду. Наприклад, у передостанньому рядку програми обчислюється f(1). Розглянемо роботу кожного рядка коду цього лістингу (рис.9):

1. Оскільки x=1, то умова x<0 не виконано, й у першому рядку нічого не відбувається.

2. Умова другого рядка x>0 виконано, тому виконуються обидва наступні рядки, об'єднані короткою вертикальною межею в загальний фрагмент.

3. Функції f(x) надається значення f(x) = "positive".

4. Умова x>1000 не виконано, тому значення "big positive" не присвоюється f (x), вона так і залишається рівною рядку "positive".

5. Останній рядок не виконується, тому що одна з умов (x>0) виявилася істинною, і оператор otherwise (тобто, "інакше") не знадобився.

Таким чином, основний принцип створення програмних модулів полягає в правильному розташуванні рядків коду. Орієнтуватися в їхній дії досить легко,

тому що фрагменти коду одного рівня згруповані в програмі за допомогою вертикальних характеристик.

3.1.2. Локальне присвоєння

Мова програмування MathCAD не була б ефективною, якби не дозволяв створювати всередині програмних модулів локальні змінні, які "не видно" ззовні, з інших частин документа. Присвоєння в межах програм, на відміну від документів MathCAD, проводиться за допомогою оператора **Local Definition** (Локальне присвоєння), який вставляється натисканням кнопки із зображенням стрілки на панелі **Programming**.

Ні оператор присвоювання :=, ні оператор виведення = у межах програм не застосовуються.

Локальне присвоєння ілюструється наступним прикладом (рис.10). Змінна z існує лише всередині програми, виділеної вертикальною межею. З інших місць документа набути її значення неможливо.

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

$$f(1) = 5$$

Рис. 10. Локальне присвоєння у програмі

3.1.3. Умовні оператори (if, otherwise)

Дія умовного оператора **if** складається із двох частин. Спочатку перевіряється логічне вираз (умова) праворуч від нього. Якщо воно істинно, виконується вираз зліва від оператора **if.** Якщо хибно - нічого не відбувається, а виконання програми продовжується переходом до її наступного рядка. Вставити умовний оператор у програму можна так:

1. Якщо необхідно, введіть ліву частину виразу та оператор присвоювання.
2. Створіть новий рядок програмного коду, натиснувши кнопку **Add Line** на панелі **Programming**.
3. Натисніть кнопку умовного оператора **if.**
4. Праворуч від оператора **if** введіть умову. Користуйтеся логічними операторами, вводячи їх з панелі **Boolean** (Булеві оператори).
5. Вираз, який має виконуватися, якщо умова виявляється виконаною, введіть ліворуч від оператора **if**.
6. Якщо в програмі передбачаються додаткові умови, додайте в програму ще один рядок натисканням кнопки **Add Line** і введіть їх таким же чином, використовуючи оператор **if** або **otherwise** (рис.11).

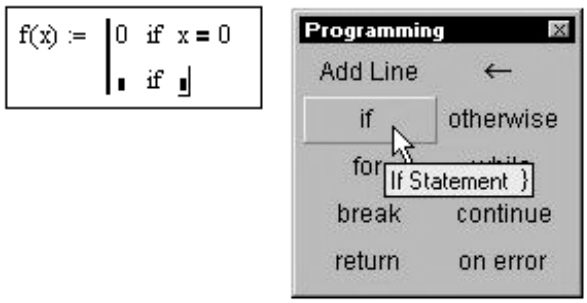


Рис. 11. Вставка умовного оператора

Оператор **otherwise** використовується спільно з одним або декількома умовними операторами if і вказує на вираз, який буде виконуватися, якщо жодна з умов не виявилося істинним. Приклади використання операторів **if** та **otherwise** наведені раніше.

3.1.4. Оператори циклу (for, while, break, continue)

У мові програмування MathCAD є два оператори циклу: **for** та **while.** Перший з них дає можливість організувати цикл за деякою змінною, змушуючи її пробігати деякий діапазон значень. Другий створює цикл із виходом із нього за деякою логічною умовою. Щоб вставити у програмний модуль оператор циклу:

1. Створіть нову лінію в програмному модулі.

2. Вставте один із операторів циклу for або while натисканням однойменної кнопки на панелі **Programming**.

3. Якщо обраний оператор for (рис. 12), то вставте у відповідні місце-заповнювачі ім'я змінної та діапазон її значень (рис.13 а і б), а якщо **while** – то логічний вираз, при порушенні якого має здійснюватися вихід із циклу (рис.13 в).

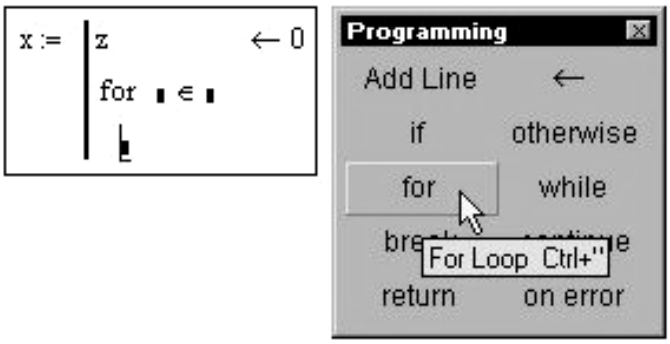
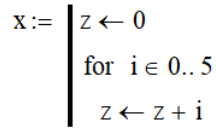


Рис.12 Вставка оператора циклу

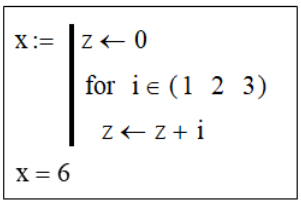
4. У нижнє місцезаповнювач введіть тіло циклу, тобто вирази, які мають виконуватися циклічно.

5. За необхідності доповніть програму іншими рядками та введіть у них потрібний код.

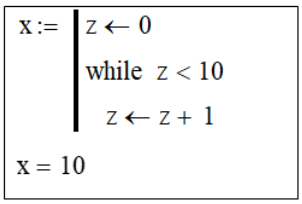
Примітка*Діапазон значень змінної за умови циклу for можна задати як з допомогою діапазону ранжувальної змінної (рис.13 а), і з допомогою вектора (рис.13 б).*



а) оператор циклуfor з ранжувальною змінною



б) оператор циклуfor з вектором



в) оператор циклу while

Рис13. Оператори циклу

3.1.5. Оператор break

Іноді необхідно достроково завершити цикл, тобто не за умовою у його заголовку, а деякому рядку в тілі циклу. Для цього використовують оператор **break**.

3.1.6. 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. Відповідно, по мірі виконання програми розрахований результат зазнає змін. Як остаточний результат видається останнє присвоєне значення. Щоб підкреслити повернення програмним модулем певного значення, можна взяти за правило робити це в останньому рядку програмного модуля (рис.14).

Разом з тим, можна перервати виконання програми в будь якій її точці (наприклад, за допомогою умовного оператора) і видати деяке значення, застосувавши оператор **return**. У цьому випадку при виконанні зазначеної умови (рис.15) значення, введене в місцезаповнювач після **return**, повертається як результат, і код більше не виконується. Вставляється в програму оператор **return** за допомогою однойменної кнопки на панелі **Programming**.

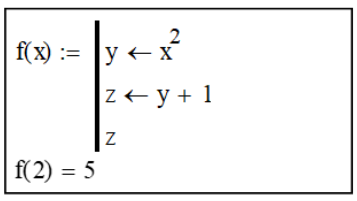


Рис.14. Повернення значення показано явно в останньому рядку програми

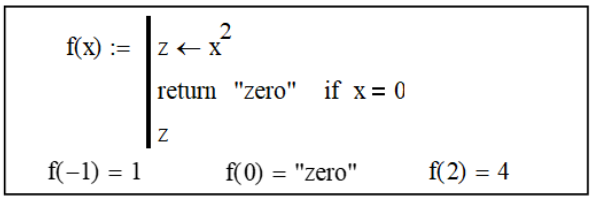


Рис.15 Застосування оператора return

3.1.7. Перехоплення помилок (on error)

Програмування MathCAD дозволяє здійснювати додаткову обробку помилок. Якщо користувач припускає, що виконання коду в якомусь місці програмного модуля здатне викликати помилку (наприклад, поділ на нуль), то цю помилку можна перехопити за допомогою оператора **on error**. Щоб вставити його в програму, потрібно помістити лінії введення в ній у потрібне положення та натиснути кнопку з ім'ям оператора **on error** на панелі **Programming**. В результаті з'явиться рядок із двома місцезаповнювачами та оператором on error посередині (рис. 16).

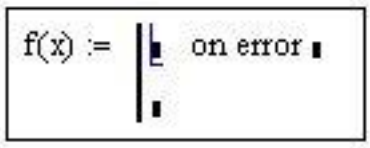
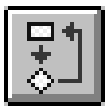


Рис. 16.Вставка оператора переходу за помилкою

У правому місцезаповнювачі слід ввести вираз, який повинен виконуватися в даному рядку програми. У лівому - вираз, яке буде виконано замість правого виразу, якщо при виконанні останнього виникне помилка.

4. Виконати з використанням блоку програмних операторів такі приклади:

Для написання програм використовується програмна палітра, що викликається Кнопкою на панелі керування. Як видно, всього є 10 операторів, з яких і будується програма. Для прикладу наведемо просту програму, що повертає 1, якщо число парне та 0, якщо непарне.



$$even(n) := \begin{cases} 1 & \text{if } mod(n,2) = 0 \\ 0 & \text{otherwise} \end{cases}$$

$even(55) = 0 \quad even(78) = 1$

Починаємо створення програми з кнопки **Add Line**.

Вертикальна лінія відіграє роль операторних дужок.

Після того, як функцію визначено, вона може використовуватись нарівні із вбудованими функціями**.**

Більш складний приклад визначення максимальної координати вектора та її позиції.

$$maximum(v) := \begin{cases} k \leftarrow 0 \\ max \leftarrow v_0 \\ \text{for } i \in 1..length(v) - 1 \\ \quad \text{if } v_i > max \\ \quad\quad \begin{cases} max \leftarrow v_i \\ k \leftarrow i \end{cases} \\ \begin{pmatrix} k \\ max \end{pmatrix} \end{cases}$$

Привласнення початкових значень змінним.

Цикл елементів вектора (не слід забувати, що елементи вектора відраховуються від 0).

Привласнення більшого значення та збереження його координати. Операторну дужку створюємо кнопкою Add Line.

Значення, що повертаються.

Визначимо тепер вектор:

$$v := \begin{pmatrix} 1 \\ 5 \\ 8 \end{pmatrix}$$

$$maximum(v) = \begin{pmatrix} 2 \\ 8 \end{pmatrix}$$

Справді, максимальне значення 8 має номер 2.

Розглянемо як ще один приклад, як програмним способом побудувати скалярне твір.

Обчислимо квадрат модуля вектора. v

$$sc(x, y) := \begin{| s \leftarrow 0 \\ n \leftarrow length(x) - 1 \\ for\ \ i \in 0..n \\ \quad s \leftarrow s + x_i \cdot y_i \\ return\ \ s \end|$$

$sc(v, v) = 90$

$(|v|)^2 = 90$

Оператор return тут необов'язковий.

Програмні рядки створюються кнопкою AddLine, оператори запроваджуються відповідною кнопкою.

Зверніть увагу, що в програмах ми не користуємося оператором присвоєння :=,

а замість нього пишемо оператор локального присвоєння ←

Усі змінні, визначені у програмі, втрачають значення при виході з неї.

Розглянемо, як такий приклад, проблему обчислення невласного інтеграла. Нескладно переконатися, що спроба підставити нескінченну межу інтегрування не матиме успіху. Наприклад:

$\int_0^\infty e^{-x^2}\,dx$

Цей інтеграл система обчислити не змогла, але у символьному вигляді він вважається.

$\int_0^\infty e^{-x^2}\,dx \to \frac{1}{2}\cdot\pi^{\frac{1}{2}}$

Складемо програму обчислення таких інтегралів.

$$Integral(f, a) := \begin{| b \leftarrow if(a = 0, 1, a) \\ I \leftarrow 0 \\ W \leftarrow 1 \\ while\ \ |I - W| > TOL \\ \quad \begin{| W \leftarrow I \\ b \leftarrow b \cdot 2 \\ I \leftarrow \int_a^b f(x)\,dx \end| \\ I \end|$$

$f(x) := e^{-x^2}$

$Integral(f, 0) = 0.886$

$\frac{1}{2}\cdot\sqrt{\pi} = 0.886$

Порівняємо з точним рішенням:

Алгоритм працює, але обраний не найвдаліший, оскільки доводиться багаторазово перераховувати інтегральні суми на тому самому діапазоні.

Дуже цікава особливість програмування Mathcad полягає в тому, що в програмі ми можемо використовувати оператори типу обчислення інтеграла, суми, похідної тощо.

Вдалий алгоритм вийде, якщо ми продовжуватимемо інтегрування від тієїточки, де закінчили попереднє інтегрування.

$$Integral(f,a) := \begin{array}{l} N \leftarrow 10 \\ b \leftarrow a + N \\ S \leftarrow 0 \\ R \leftarrow 1 \\ \text{while } |R| > TOL \\ \quad \left| \begin{array}{l} R \leftarrow \int_a^b f(x)\,dx \\ S \leftarrow S + R \\ N \leftarrow N \cdot 2 \\ a \leftarrow b \\ b \leftarrow b + N \end{array} \right. \\ S \end{array}$$

$Integral(f,0) = 0.886$

Отримали той самий результат, що й не дивно,

але алгоритм працює швидше.

За наочністю тексту, навряд чи якась мова

програмування зрівняється з Mathcad-програмою.

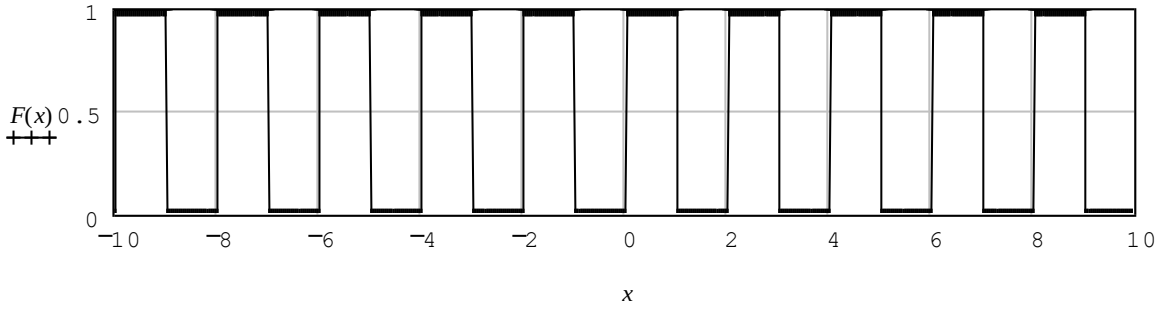
Тут для прискорення роботи програми ми на кожному кроці

подвоює інтервал інтегрування.

Можна використовувати програмні можливості Mathcad просто для завдання більш складного вигляду. Наприклад, визначимо функцію, яка дорівнює 1, якщо аргумент розміщений між парним і непарним числом, і 0, інакше.

$$F(x) := \begin{array}{ll} 0 & \text{if } mod(ceil(x),2) = 0 \\ 1 & \text{otherwise} \end{array}$$

Функція $ceil(x)$ повертає найближче ціле xбільше.
Наприклад: $ceil(2.0001) = 3$



Наведемо ще програму перекладу десяткового числа в двійкове представлення. Тут функція $floor(x)$ - найближче ціле менше x.

Функція $mod(x,2)$ - залишок від розподілу за модулем.

Наприклад:

$Binary(255) = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$

$Binary(373) = (1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1)$

$$Binary(N) := \left| \begin{array}{l} p \leftarrow floor\left(\frac{ln(N)}{ln(2)}\right) \\ for\ \ i \in 0..p \\ \quad b_i \leftarrow mod\left(floor\left(\frac{N}{2^i}\right), 2\right) \\ b^T \end{array} \right.$$

Примітка: b^T - транспонована матриця, а не степінь T.

5. Отримати індивідуальні завдання у викладача.

6. Варіанти індивідуальних завдань

Варіант	Завдання	
1	Скласти програму для обчислення	$z = \begin{cases} x^2 - y^2, & \text{если } x < y; \\ x^2 + y^2, & \text{если } x = y; \\ x - y, & \text{если } x > y. \end{cases}$
2	Скласти програму для обчислення	$f(x) = \begin{cases} \frac{\sin x}{x}, & \text{если } x \neq 0; \\ 1, & \text{если } x = 0. \end{cases}$
3	Скласти програму для обчислення	$z = \begin{cases} x + 3, & \text{если } x \leq 0; \\ x \cdot \sin(3x), & \text{если } 0 < x < 1{,}5; \\ x^2 - 1, & \text{если } 1{,}5 < x < 2{,}5; \\ 5, & \text{если } 2{,}5 = x; \\ x + \cos(x + 0{,}5), & \text{если } 2{,}5 < x < 4{,}5. \end{cases}$
4	Двовимірний масив А містить і рядків і j стовпців. Одновимірний	

	масив X містить j елементів. Скласти фрагмент програми для обчислення n елементів одновимірного масиву B за формулою множення матриці на вектор: $B_i = \sum_{j=1}^{n} A_{ij} \cdot X_j$, i = 1, 2, 3, … , n	
5	Скласти програму для обчислення: $y = \sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + ...,$ з точністю до $\varepsilon=10^{-6}$.	
6	Скласти програму для обчислення: $y = e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + ... + \frac{x^n}{n!},$ с точностью до $\varepsilon=10^{-4}$.	
7	Знайти суму цілих додатніх чисел кратних 4 та менших 100.	
8	Знайти суму цілих додатніх чисел менших за 200.	
9	Знайти суму цілих додатніх чисел кратних 3, більших 20 та менших 100.	
10	Знайти суму цілих додатніх непарних чисел менших за 200.	
11	Знайти суму 10 членів ряду $a_n = \frac{(n!)^2}{(2^{n^2})}$.	
12	Знайти суму 15 членів ряду $a_n = \frac{(n^{\ln n})}{(\ln n)^n}$.	
13	Знайти суму 7 членів ряду $a_n = e^{-\sqrt[3]{n}}$.	
14	Знайти суму 9 членів ряду $a_n = n^2 e^{-\sqrt[2]{n}}$.	
15	Знайти суму цілих додатніхпарних чисел меньших за 300.	
16	Знайти суму цілих додатніх чисел кратних5, більших 50 та менших 500.	