

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Інженерний навчально-науковий інститут**

А.О. Ніконова, З.А. Ніконова

СПЕЦІАЛЬНІ РОЗДІЛИ ЕЛЕКТРОДИНАМІКИ

Методичні рекомендації до лабораторних робіт
для здобувачів вищої освіти бакалавра
спеціальності «Мікро- та наносистемна техніка»
освітньо-професійної програми
«Мікро- та наносистемна техніка»

Затверджено
Вченою радою ЗНУ
Протокол № 11 від 23.06 2021

Запоріжжя
2021

УДК 621.38
НЗ77

Ніконова А.О., Ніконова З.А. Спеціальні розділи електродинаміки: методичні рекомендації до лабораторних робіт для здобувачів вищої освіти бакалавра спеціальності «Мікро- та наносистемна техніка» освітньо-професійної програми «Мікро- та наносистемна техніка»: ЗНУ, 2021. 50с.

Методичні рекомендації до лабораторних робіт розроблено для студентів денної та заочної форми навчання спеціальності «Мікро- та наносистемна техніка». Запропоновані завдання призначені для освоєння теоретичного матеріалу, опанування навичок розв'язання практичних задач, виконання самостійних завдань.

Рецензент

В.Л. Коваленко, доктор технічних наук, професор кафедри електротехніки та енергоефективності

Відповідальний за випуск

Т.В. Критська, доктор технічних наук, професор кафедри мікроелектронних та електронних інформаційних систем

Зміст

Вступ.....	4
1 Лабораторна робота №1	
Гradient скалярної функції, дивергенція та ротор векторного поля.....	5
2 Лабораторна робота №2	
Символьні розрахунки в пакеті MATLAB.....	8
3 Лабораторна робота №3	
Випадкова функція. Розрахунок градієнта скалярної функції, дивер- генції та ротора векторного поля в пакеті MATLAB.....	18
4 Лабораторна робота № 4	
Побудова графіків в пакеті MATLAB.....	29
5 Лабораторна робота № 5	
Випадкова функція.....	32
6 Лабораторна робота № 6	
Основи програмування в MATLAB.....	40
ЛІТЕРАТУРА.....	50

ВСТУП

Методичні рекомендації «Спеціальні розділи електродинаміки» розроблені для здобувачів вищої освіти бакалавра спеціальності «Мікро- та наносистемна техніка» освітньо-професійної програми «Мікро- та наносистемна техніка»

Метою викладання навчальної дисципліни «Спеціальні розділи електродинаміки» є набуття знань та навичок, пов'язаних з розумінням математичного апарату класичної теорії поля, поглиблення знань з електродинаміки, які одержано в загальному курсі фізики.

Основними завданнями викладання дисципліни «Спеціальні розділи електродинаміки» є формування у студентів базової підготовки в області класичної теорії поля, класичної електродинаміки, та спеціальної теорії відносності.

Вивчення навчальної дисципліни забезпечує набуття студентами наступних загальних та спеціальних компетентностей:

- здатність розв'язувати складні спеціалізовані задачі та практичні проблеми, що характеризуються комплексністю та невизначеністю умов, під час професійної діяльності у галузі мікро- та наносистемної техніки, або у процесі навчання, що передбачає застосування теорій та методів автоматизації та електроніки;
 - здатність застосовувати знання у практичних ситуаціях
 - здатність вчитися і оволодівати сучасними знаннями.
 - здатність використовувати знання і розуміння наукових фактів, концепцій, теорій, принципів і методів для проектування та застосування мікро- та наносистемної техніки.
 - здатність ідентифікувати, класифікувати, оцінювати і описувати процеси у мікро- та наносистемній техніці за допомогою побудови і аналізу їх фізичних і математичних моделей.
 - здатність визначати та оцінювати характеристики та параметри матеріалів мікро- та наносистемної техніки, аналогових та цифрових електронних пристроїв, мікропроцесорних систем.
 - здатність використовувати знання з оптичної аналогової та цифрової схемотехніки, оптоелектроніки, фотовольтаїки та геліоелектроніки.
- здатність застосовувати знання з моделювання функціональних вузлів мікроелектроніки та конструювання приладів на їх основі

ЛАБОРАТОРНА РОБОТА №1

Гradient скалярної функції, дивергенція та ротор векторного поля

Мета роботи: *визначити та розрахувати основні величини векторної алгебри та векторного аналізу*

Завдання:

1. Розрахувати gradient скалярної функції в точці.
2. Побудувати векторне поле яке задане функцією
3. Визначити дивергенцію і ротор векторного поля

Gradient функції- векторна величина, знаходження якої пов'язане з визначенням власних похідних функції. Напрямок gradientа вказує шлях якнайшвидшого зростання функції від однієї точки скалярного поля до іншої.

Для вирішення завдання на gradient функції застосовуються способи диференціального обчислення, а саме знаходження власних похідних першого порядку по трьом змінним. При цьому передбачається, що сама функція і всі її приватні похідні володіють властивістю безперервності в області визначення функції.

Gradient скалярного поля має напрямок, що характеризується максимальною швидкістю спаду скалярної функції. Такий толк даної векторної величини обгрунтовується вираженням для визначення її компонент.

$$\text{grad } \varphi = \nabla \varphi = \mathbf{i} \frac{\partial \varphi}{\partial x} + \mathbf{j} \frac{\partial \varphi}{\partial y} + \mathbf{k} \frac{\partial \varphi}{\partial z}.$$

Оператор набла:

$$\nabla = \mathbf{i} \frac{\partial}{\partial x} + \mathbf{j} \frac{\partial}{\partial y} + \mathbf{k} \frac{\partial}{\partial z},$$

Правила розрахунку gradientа:

1. $\nabla c = 0$, где $c = const$
2. $\nabla(\varphi_1 + \varphi_2) = \nabla\varphi_1 + \nabla\varphi_2$
3. $\nabla(c\varphi) = c\nabla\varphi$
4. $\nabla(\varphi_1\varphi_2) = \varphi_1\nabla\varphi_2 + \varphi_2\nabla\varphi_1$

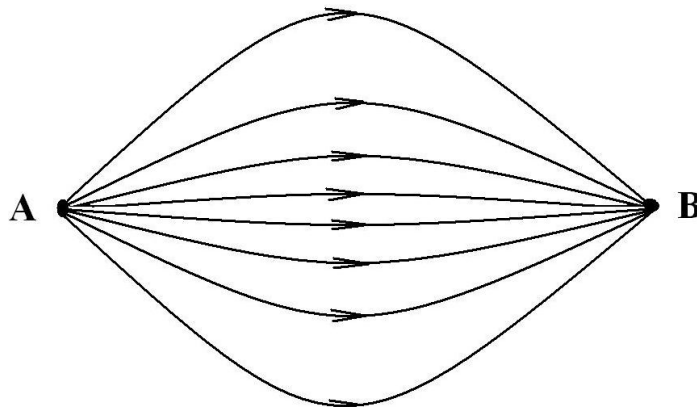
Скалярне поле. Якщо кожній точці деякої області простору поставлено у відповідність певне число (частіше дійсне), то говорять, що в цій області задано скалярний поле.

Векторне поле — векторнозначна функція, відображення, яке кожній точці даного простору ставить у відповідність вектор. У сучасній диференціальній геометрії розглядається також узагальнення на довільні многовиди (див. векторне розшарування).

Якщо вибрати декартову систему координат, то поле може бути подане як:

$$\vec{a} = P(x, y, z)\vec{i} + Q(x, y, z)\vec{j} + R(x, y, z)\vec{k}$$

Дивергенція. Векторне поле може мати точки в яких починаються та закінчуються силові лінії. На рисунку 1.1 силові лінії починаються в т.А та закінчуються в т.В. В цьому випадку т.А називається джерелом, а т.В – стоком. «Потужність» джерела або стока називають дивергенцією (від лат. *divergo*- розхожість)



Дивергенція векторного поля в т. А розраховується як:

$$\operatorname{div} \mathbf{a} = \lim_{\Delta V \rightarrow 0} \frac{\oint \mathbf{a} d\mathbf{S}}{\Delta V}$$

Правила розрахунку дивергенції:

1. $\operatorname{div} c = 0$, где $c = \operatorname{const}$
2. $\operatorname{div}(\mathbf{a}_1 + \mathbf{a}_2) = \operatorname{div} \mathbf{a}_1 + \operatorname{div} \mathbf{a}_2$
3. $\operatorname{div}(c\mathbf{a}) = c \operatorname{div} \mathbf{a}$
4. $\operatorname{div}(\varphi\mathbf{a}) = (\mathbf{a} \nabla \varphi) + \varphi \operatorname{div} \mathbf{a}$

Ротор, або вихор - векторний диференційний оператор над векторним полем. Показує, наскільки і в якому напрямку закручено поле в кожній точці. Ротор поля F позначається символом $\operatorname{rot} F$

$$\operatorname{rot} (F_x \mathbf{i} + F_y \mathbf{j} + F_z \mathbf{k}) = \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z} \right) \mathbf{i} + \left(\frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x} \right) \mathbf{j} + \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right) \mathbf{k}.$$

Завдання

1. Знайти градієнт скалярної функції

- 1в.) $Y = x^3 - z^2$
- 2в.) $Y = x^2 - 2xy + y^2 + z^2$
- 3в.) $Y = x^3 - 2xz + yz$
- 4в.) $Y = x + y^2 + z^3$
- 5в.) $Y = xyz$

2. Побудувати скалярне та векторне поля:

- 1в.) $F(x, y) = x^2 + y^2$
- 2в.) $F(x, y) = x^2 + 2xy$
- 3в.) $F(x, y) = -3x + y^2$
- 4в.) $F(x, y) = x^2 - xy^2$
- 5в.) $F(x, y) = 2y + 3y^2$

Векторне поле задане векторною функцією:

- 1в.) $F(x, y) = (x-y, x+y)$
- 2в.) $F(x, y) = (x, -y)$
- 3в.) $F(x, y) = (-x, 2y)$
- 4в.) $F(x, y) = (x-y, -y)$
- 5в.) $F(x, y) = (-x, -y)$

$$\operatorname{div} \bar{a} = \frac{\partial P}{\partial x} + \frac{\partial Q}{\partial y} + \frac{\partial R}{\partial z}$$

3. Знайти дивергенцію та ротор векторного поля

Приклад:

$$\mathbf{a} = (2x+y)\mathbf{i} + (2z+5x)\mathbf{j}$$

$$P(x, y, z) = (2x+y)$$

$$Q(x, y, z) = (2z+5x)$$

$$R(x, y, z) = 0$$

$$\frac{dP}{dx} = 2$$

$$\frac{dQ}{dx} = 0$$

$$\frac{dR}{dx} = 0$$

$$\text{Div } a = 2 + 0 + 0 = 2$$

1. $a = (2x + y)\mathbf{i} + (-2z + 5x)\mathbf{k}$
2. $a = (x - 8y)\mathbf{i} + (-2x + 6z)\mathbf{j} + (-2z + 5y)\mathbf{k}$
3. $a = (x - 4y)\mathbf{i} + (9y - 3z)\mathbf{j} + (-5z + y)\mathbf{k}$
4. $a = (x + 3y)\mathbf{i} + (x - 6y)\mathbf{j} + (-2z + 5x)\mathbf{k}$
5. $a = (-x + 2y)\mathbf{i} + (-2x - 3y)\mathbf{j} + (-2z - 3y)\mathbf{k}$

ЛАБОРАТОРНА РОБОТА № 2

Символьні розрахунки в пакеті MATLAB

Інтерактивне середовище matlab для
Перетворення сигналу

MATLAB - це високорівнева мова і інтерактивне середовище для програмування, чисельних розрахунків і візуалізації результатів. За допомогою MATLAB можна аналізувати дані, розробляти алгоритми, створювати моделі і додатки.

Мова, інструментарій та вбудовані математичні функції дозволяють досліджувати різні підходи і отримувати рішення швидше, ніж з використанням електронних таблиць або традиційних мов програмування, таких як C / C++ або Java.

MATLAB широко використовується в таких областях, як:

1. обробка сигналів і зв'язок,
2. обробка зображень і відео,
3. системи управління,

4. автоматизація тестування і вимірювань,
5. фінансовий інжиніринг,
6. обчислювальна біологія і т.п.

MATLAB являє собою основу всього сімейства продуктів MathWorks і є головним інструментом для вирішення широкого спектра наукових і прикладних задач, в таких областях як: моделювання об'єктів і розробка систем управління, проектування комунікаційних систем, обробка сигналів і зображень, вимірювання сигналів і тестування, фінансове моделювання, обчислювальна біологія та ін.

Ядро MATLAB дозволяє максимально просто працювати з матрицями реальних, комплексних і аналітичних типів даних і зі структурами даних і таблицями пошуку.

Всі вбудовані функції ядра MATLAB розроблені й оптимізовані фахівцями і працюють швидше або так само, як їх еквівалент на C / C ++

Створення символьних змінних, виразів, матриць.

Для створення символьних змінних використовується команда `sym`, у якій наступний синтаксис: ім'я змінної = `sym('ім'я змінної')`

Наприклад, створимо дві символьні змінні `x` та `alpha`:

```
>> x = sym(' x ')
x =
x
>> a = sym(' alpha ')
a =
alpha
```

Для створення одночасно трьох символьних змінних `a`, `b`, `c` треба виконати команду:

```
>> syms a b c
```

Створення символьного виразу виконується командою:

```
>> sym (' символний вираз')
```

Наприклад, для створення символної змінної, що містить вираз $a x^2 + b x + c$, треба виконати команду:

```
>> f = sym (' a * x ^ 2 + b * x + c')
```

В цьому випадку вираз розглядається як єдина змінна. Для того, щоб мати можливість змінювати значення коефіцієнта та невідомої, що входить в вираз $a x^2 + b x + c$ треба виконати команди:

```
>> syms a b c x
```

```
>> f = sym (' a * x ^ 2 + b * x + c')
```

```
f =
```

```
 a * x ^ 2 + b * x + c
```

Створення символної матриці

Для створення символної матриці необхідно створити символні змінні, які є елементами матриці та потім створити матрицю, задав її строки та стовбці.

```
>> syms a b c
```

```
>> A = [a b c ; b c a ; c a b]
```

```
A = [a, b, c] [b, c, a] [c, a, b]
```

Далі зі створеною символною матрицею можна виконувати різні арифметичні операції

Розв'язок алгебраїчних рівнянь

Для розв'язку алгебраїчних рівнянь використовується команда solve

Приклад 1.

Розв'язати рівняння:

$$\frac{2x}{x-1} - \frac{7}{2} = \frac{x+1}{x-1} + \frac{5}{2-2x}$$

```
>> [x] = solve (' 2 * x / (x - 1) - 7 / 2 = (x + 1) / (x - 1) + 5 / (2 - 2 * x)')
```

$$x = 2$$

Рішення системи алгебраїчних рівнянь

Для рішення системи алгебраїчних рівнянь використовується команда solve

Приклад 1. Розв'язати систему алгебраїчних рівнянь:

$$\begin{cases} 3x + 4y = 18 \\ 2x + 5y = 19 \end{cases}$$

```
>> [ x, y ] = solve (' 3 * x + 4 * y = 18', ' 2 * x + 5 * y = 19 ')
```

$$x = 2$$

$$y = 3$$

Спрощення алгебраїчних рівнянь

Для спрощення виразів використовують команду simplify.

Приклад 2:

Спростити вираз

$$\left(a - \frac{4a - 9}{a - 2} \right) : \left(2a - \frac{2a}{a - 2} \right).$$

```
>> syms x
```

% описуємо символну змінну

```
>> p = (a - (4 * a - 9)) / (a - 2) / (2 * a - 2 * a / (a - 2)) % задаємо символний
```

вираз

```
>> simplify (p)
```

ans =

$$-3/2 / a$$

Символьне диференціювання

Для розрахунку похідної функції $f(x)$ необхідно:

- задати вираз, що описує функцію;

– звернутися до функції diff.

Приклад 3:

Розрахувати похідну функції $\sin(ax)$ по змінній x .

```
>> sym a x                               % описуємо символъну змінну
>> y = sin (a * x)                       % задаємо диференціювану функцію
>> diff ( y )                            % розраховуємо похідну в символъному виді

ans =
      cos (a * x) * a
```

Приклад 4:

Розрахувати похідну функцію $\sin(ax)$ по параметру a .

```
>> sym a x                               % описуємо символъну змінну
>> y = sin (a * x)                       % задаємо диференціювану функцію
>> diff (y, a)                            % розраховуємо похідну в символъному виді

ans =      cos(a * x) * x
```

Приклад 5:

Розрахувати похідну функцію x^n

```
>> sym x y n                             % описуємо символъну змінну
>> y = x ^ n                              % задаємо функцію  $x^n$ 
>> diff (y, x)                            % розраховуємо похідну функцію  $x^n$  в сим-
вильному виді

ans =
      x ^ n * n / x
```

Приклад 6:

Розрахувати інтеграл:

$$\int \frac{dx}{a^2 + (bx)^2} ;$$

```
>> syms a b c % задаємо символні змінні
>> int (1 / a ^ 2 + ( b * x ) ^ 2) % розраховуємо інтеграл в символному виді
ans =
1/ a / b * atan (b * x / a)
```

Приклад 7:

Розрахувати інтеграл:

$$\int_a^{a/b} \frac{dx}{a^2 + (bx)^2};$$

```
>> syms a b c % задаємо символні змінні
>> int (1 / a ^ 2 + (b * x) ^ 2, 0, a / b) % розраховуємо інтеграл в символному
виді
ans =
1 / 4 * pi / a / b
```

Порядок виконання лабораторної роботи

1. Завдання № 1

Розв'язати алгебраїчне рівняння, використовуючи команду solve:

$$a) \frac{1}{x-3} + \frac{x+8}{2x^2-18} = \frac{1}{3-x} - 1;$$

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

$$c) 1 + \frac{6}{x-1} = \frac{5-2x}{x-7};$$

2. Завдання № 2

Розв'язати систему алгебраїчних рівнянь, використовуючи команду solve.

Обрати рівняння згідно варіанту з табл.2.1

Таблиця 2.1- Завдання для самостійної роботи

1	$\begin{cases} y^2 - 3xy + x^2 - x + y + 9 = 0, \\ y - x = 2. \end{cases}$
2	$\begin{cases} (2x-6)/3 - (x-2)/2 = 2y, \\ (3x-6)/2 + y/2 = x. \end{cases}$
3	$\begin{cases} 2x + y = 1, \\ y^2 - 2xy - x^2 = -28 \end{cases}$
4	$\begin{cases} x^2 - xy = 12 - y^2, \\ x - 2y = 6. \end{cases}$
5	$\begin{cases} (x-6)/2 - (y+1)/3 = 1, \\ (2-y)4 + (x-1)/2 = 2. \end{cases}$
6	$\begin{cases} x - y = 7, \\ x^2 + y^2 = 9 - 2xy. \end{cases}$
7	$\begin{cases} 1/x - 1/y = -4/5, \\ x - y = 4. \end{cases}$
8	$\begin{cases} 4/(x-y) + 12(x+y) = 3, \\ 8/(x-y) + 18(x+y) = -1. \end{cases}$
9	$\begin{cases} 3/(x-y) - 4/(x+y) = -1, \\ 9/(x-y) - 10/(x+y) = 8. \end{cases}$
10	$\begin{cases} 6/(x-y) - 8/(x+y) = -2, \\ 9/(x-y) - 10/(x+y) = 8. \end{cases}$
11	$\begin{cases} 1/x + 4/y = 4, \\ 1/e - 2/x = 10. \end{cases}$
12	$\begin{cases} 2/x + 1/y = 4, \\ 1/x - 3/y = 9. \end{cases}$
13	$\begin{cases} 1/x + 1/y = 7/12, \\ 1/x - 1/y = 1/12. \end{cases}$
14	$\begin{cases} 1/x - 1/y = 3/8, \\ x + y = 12. \end{cases}$
15	$\begin{cases} x + y = 3, \\ x^2 + 3xy + y^3 - x - y = 2. \end{cases}$
16	$\begin{cases} 3x - y = 10, \\ x^2 - y^2 = 20 - xy. \end{cases}$

3. Завдання № 3

Спростити вираз, використовуючи команду simplify

$$\text{a) } p = \left(\frac{c}{c-2} - \frac{c}{c+2} - \frac{c^2+4}{4-c^2} \right) \cdot \frac{(2-c)^2}{2c+c};$$

$$\text{b) } p = \frac{1}{(x-1)(x-3)} + \frac{1}{(x-3)(x-5)} + \frac{1}{(x-5)(x-7)};$$

$$\text{c) } p = \left(3x - \frac{3x}{x-4} \right) : \left(x - \frac{6x-25}{x-4} \right).$$

4. Завдання № 4

Розрахувати похідну функцію по x , використовуючи команду diff.

Обрати рівняння згідно свого варіанту з табл. 2.2.

Таблиця 2.2- Завдання для самостійної роботи

1	$\frac{\ln x}{x\sqrt{1+\ln x}}$
2	$\frac{\ln^2 x}{x}$
3	$x \cdot e^x \cdot \sin x$
4	$\frac{1}{\sqrt{9+x^2}}$
5	$\frac{1}{x^2} \cdot \sin\left(\frac{1}{x}\right)$
6	$\text{ArcSin} \sqrt{\frac{x}{1+x}}$
7	$x^x \cdot (1 + \ln x)$
8	$\frac{\sqrt{x^2 - 0,16}}{x}$
9	$\frac{e^{3x} + 1}{e^x + 1}$
10	$\frac{e^x \cdot (1 + \sin x)}{1 + \cos x}$

11	$\sin x \cdot \ln(\operatorname{Tg} x)$
12	$\frac{1}{(x+1) \cdot \sqrt{x^2+1}}$
13	$\frac{1}{(3\sin x + 2\cos x)^2}$
14	$\left(\frac{\ln x}{x}\right)^3$
15	$\frac{x}{(x^4 + 3x^2 + 2)}$
16	$\frac{1}{x \lg x} dx$

5. Завдання №5

Використовуючи команду `int`, підготувати та організувати розрахунок даного інтегралу:

$$y = \int_a^b f(x) dx.$$

Інтегральна функція $f(x)$ визначена та неперервна на інтервалі $a \leq x \leq b$. Вигляд функції $f(x)$, а також інтервал інтегрування $[a, b]$ визначається номером варіанту з табл. 2.3.

Таблиця 2.3- Завдання для самостійної роботи

		a	b
1	$\frac{\ln x}{x\sqrt{1+\ln x}}$	1	3,5
2	$\frac{\ln^2 x}{x}$	1	4
3	$x \cdot e^x \cdot \sin x$	0	1
4	$\frac{1}{\sqrt{9+x^2}}$	0	2
5	$\frac{1}{x^2} \cdot \sin\left(\frac{1}{x}\right)$	1	2,5
6	$\text{ArcSin} \sqrt{\frac{x}{1+x}}$	0	3
7	$x^x \cdot (1 + \ln x)$	1	3
8	$\frac{\sqrt{x^2 - 0,16}}{x}$	1	2
9	$\frac{e^{3x} + 1}{e^x + 1}$	0	2
10	$\frac{e^x \cdot (1 + \sin x)}{1 + \cos x}$	0	1,5
11	$\sin x \cdot \ln(\text{Tg} x)$	1	1,5
12	$\frac{1}{(x+1) \cdot \sqrt{x^2+1}}$	0	0,75
13	$\frac{1}{(3 \sin x + 2 \cos x)^2}$	0	1
14	$\left(\frac{\ln x}{x}\right)^2$	1	2
15	$\frac{x}{(x^4 + 3x^2 + 2)}$		2
16	$\frac{1}{x \lg x} dx$	2	3

ЛАБОРАТОРНА РОБОТА № 3

Побудова графіків в пакеті MATLAB

Мета роботи: *Побудувати графіки функцій та отримати практичні навички в роботі в діалоговому режимі*

Одною з переваг системи MATLAB є велика кількість засобів графіки, починаючи від побудовання простих графіків функцій однієї змінної в декартовій системі координат та закінчуючи комбінованими та презентаційними графіками з елементами презентації.

Побудування графіків відрізками прямих.

Для відображення функції однієї змінної $y(x)$ використовують графіки в декартовій (прямокутній) системі координат. При цьому зазвичай будують дві осі – горизонтальна X та вертикальна Y , та задаються координати x і y , які визначають $y(x)$. Ці точки з'єднуються одна з іншою відрізками прямих, тобто при побудові графіка відбувається лінійна інтерполяція для проміжних точок. Оскільки MATLAB – матрична система, сукупність точок $y(x)$ задається векторами X та Y однакового розміру. Команда `plot` здійснює побудову графіків функцій в декартовій системі координат. Ця команда має ряд параметрів, що розглядаються нижче.

`plot(X, Y)` будує графік функції $y(x)$, координати точок (x, y) яких беруть з векторів однакового розміру Y и X . Якщо X або Y – матриця, то будується сімейство графіків по даним які містяться в матриці.

Приклад 1:

Приклад (рис. 3.1) ілюструє побудову графіків двох функцій – $\sin(x)$ та $\cos(x)$, дані яких містяться в матриці Y , а значення x зберігаються в векторі X :

```
>> x = [0 1 2 3 4 5 6]
>> y = [sin(x); cos(x)];
>> plot(x, y)
```

В даному випадку графік складається з відрізків, і якщо потрібно щоб відображена функція мала вид більш пологої кривої, необхідно збільшити кількість точок. Положення їм може бути довільним.

Побудова графіків функцій $\sin(x)$, $\cos(x)$ відрізками прямих `plot(Y)` будує графік $y(i)$, де значення Y беруться з вектора Y , а i представляє собою індекс відповідного елемента. Якщо Y містить комплексні елементи, то виконується команда `plot(real(Y), imag(Y))`. У всіх інших випадках уявна частина даних ігнорується.

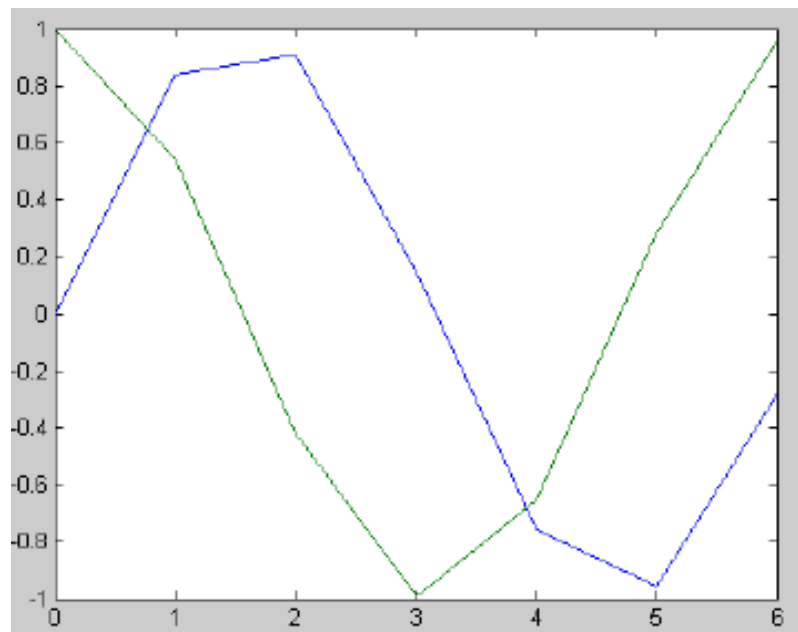


Рисунок 3.1 – Графіки двох функцій – $\sin(x)$ та $\cos(x)$

Приклад 2:

Побудувати графік функції $y = \sin(x) + i * \cos(3 * x)$, де аргумент $x \in [-2\pi; 2\pi]$ та змінюється з шагом 0,02 (рис 3.2).

```
>>x = - 2 * pi : 0.02 * pi : 2 * pi ;
>> y = sin (x) + i * cos (3 * x) ;
>> plot (y)
>> plot (y , ' b : * ' )
```

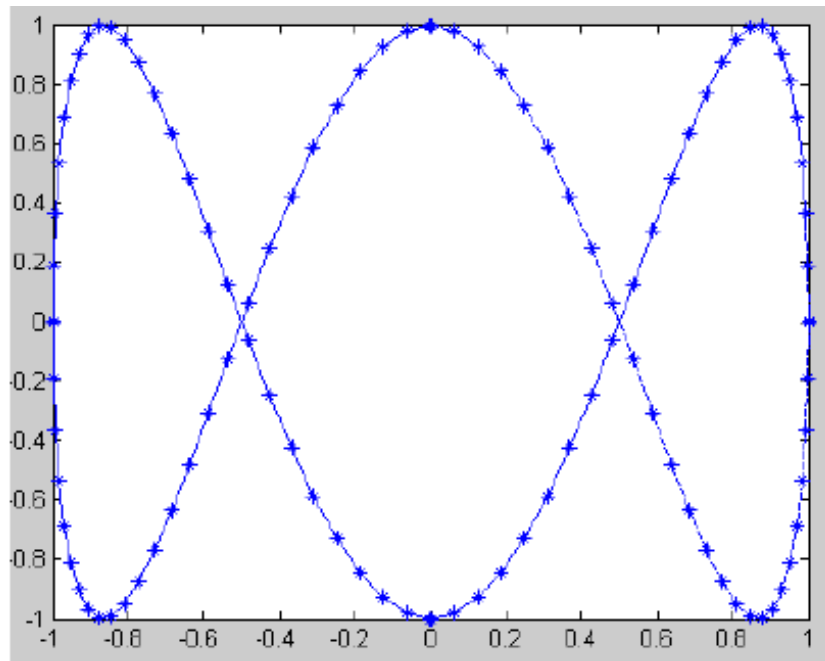


Рисунок 3.2 - Графік функції $y = \sin(x) + i * \cos(3 * x)$

Приклад 3:

Побудувати графік функції параболу в інтервалі $[1; 10]$ з кроком 1.

Розглянемо, як можуть виглядати завдання.

- `>> x = 1 : 1 : 10;`
`>> y = x .* x;`
`>> plot(x, y)`
- `>> x = 1 : 1 : 10;`
`>> y = x .^ 2;`
`>> plot(x, y)`
- `>> for x = 1 : 10;`
`>> y(x) = x * x; end`
`>> plot(y)`

Приклад 4:

Побудувати графік параболу в інтервалі $[1; 20]$ з кроком 0,1. Розглянемо, як можуть виглядати завдання.

- `>> x = 1 : 0.1 : 20;`

```

>> y = x . ^ 2 ; plot (x, y)
– >> for i = 1 : 20;
>> x = 0.1 * i ;
>> y (i) = x * x ; end
>> plot (Y)

```

plot (X, Y, S) аналогічна команді plot (X, Y), але тип лінії графіка можна задавати за допомогою строкової константи S, яка задає колір лінії (табл.3.1) або вид символу (табл.3.2).

Таблиця 3.1 – Колір лінії

Колір лінії	
Y- Жовтий	G- Зелений
M- Фіолетовий	B - Синій
C- Блакитний	W- Білий
R- Червоний	K - Чорний

Таблиця 3.2 – Вид символу

Тип точки	Тип лінії
. - Точка	
+ - Плюс	
O - Коло	_ - Суцільна
X - Хрест	: - Двійний пунктир
* - Зірочка	-. - Штрих-пунктир
S – Квадрат	-- - Штрихова
D - Ромб	
V - Трикутник	

Таким чином, за допомогою строкової константи S можливо:

- змінювати колір лінії;
- задавати вузлові точки різними точками (точка, коло, трикутник);

– змінювати тип лінії.

`plot (X1, Y1, S1, X2, Y2, S2, X3, Y3, S3,...)` – ця команда будує на одному графіку ряд ліній, які представлені даними виду (X_i, Y_i, S_i) , де X_i та Y_i – вектори або строки матриці а S_i – задає колір та тип точок або ліній, з яких буде побудовано графік. При відсутності вказівки на колір лінії та точок він обирається автоматично з таблиці кольорів (білий не враховується).

Розглянемо приклад побудови графіків трьох функцій з різним стилем представлення кожного з них:

Приклад 5:

```
>> x = - 2 * pi : 0.1 * pi : 2 * pi ;
>> y1 = sin (x);
>> y2 = sin (x). ^ 2;
>> y3 = sin (x). ^ 3;
>> plot (x, y1, ' - b ', x, y2, ' - . + r ', x, y3, ' - o k ')
```

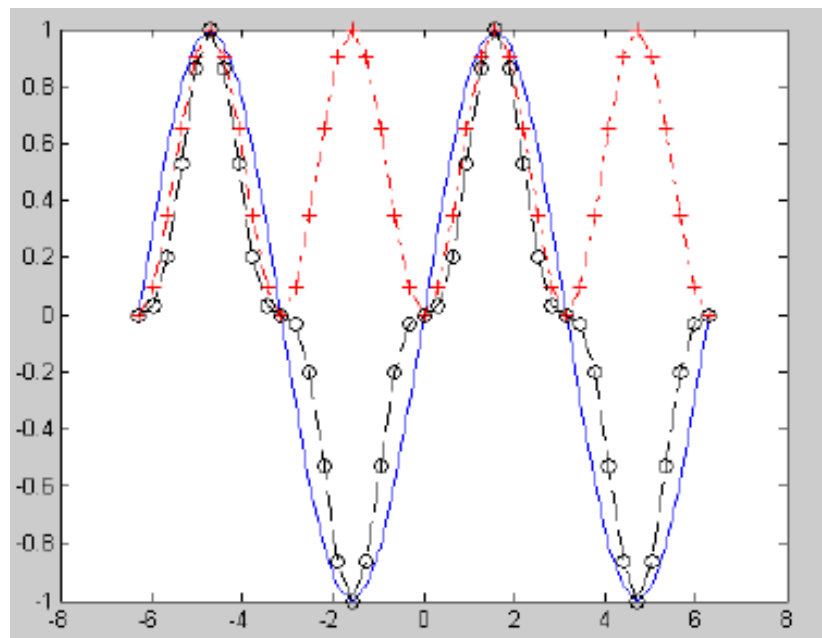


Рисунок 3.3- Графіки трьох функцій : графік функції y_1 будується суцільною синій лінією ; – графік y_2 будується штрихпунктирною лінією з точками у вигляді знака «плюс» червоного кольору; – графік y_3 будується штриховою лінією чорного кольору.

Будова графіків в полярній системі координат

В полярній системі координат будь-яка точка виглядає як кінець радіус-вектора, який виходить з початку системи координат, що має довжину r та кут Q . Для будови функції $r(Q)$ використовуються приведені нижче команди. Кут Q зазвичай змінюється від 0 до 2π . Для будови графіків функцій в полярній системі координат використовуються команди типу `polar(...)`:

- `polar (ТНЕТА, RНО)` будує графік в полярній системі координат, що представляє собою положення кінця радіус-вектора довжиною $RНО$ та с кутом $ТНЕТА$;
- `polar (ТНЕТА, RНО, S)` – аналогічна попередній команді, але дозволяє задавати стиль побудови за допомогою строкової константи S по аналогії з командою `plot`.

Приклад 6:

Графік `Polar(y)` в полярній системі координат (рис. 3.4).

```
>> phi = 0 : 0.1 : 2 * pi; r = sin (3 * phi) ;
```

```
>> polar (phi, r)
```

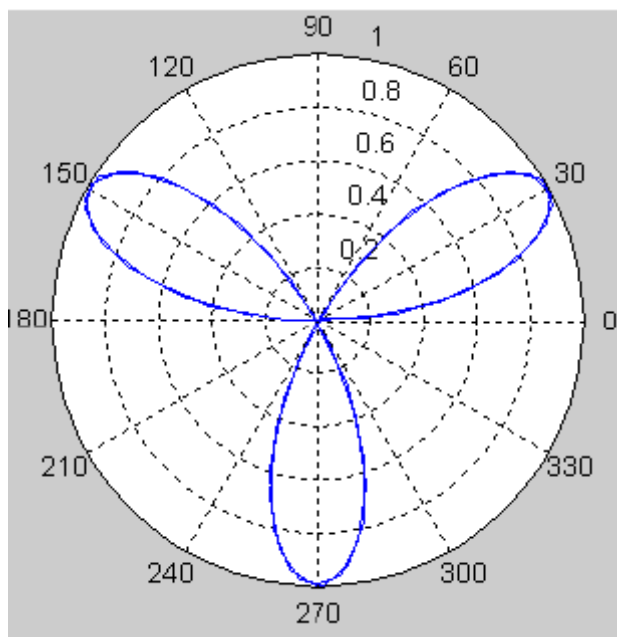


Рисунок 3.4- Графік функції $\sin(3\alpha)$ в полярній системі координат

Будова графіків векторів

Радіус-вектор в звичайному вигляді, тобто у вигляді стрілки, що виходить з початку координат та має кут і довжину, що визначаються дійсною та уявною частиною комплексних чисел, що представляють ці вектори. Для цього існує група команд `compass`:

- `Compass (U, V)` – будує графіки радіус-векторів за компонентами (U, V) , що представляють дійсну та уявну частини кожного із радіус-векторів.
- `Compass (Z)` – еквівалентно `Compass (real(Z), imag(Z))` ;
- `Compass (U, V, LINESPEC)` и `Compass (Z, LINESPEC)`- аналогічні представленим вище командам, але дозволяють задавати специфікацію ліній побудови `LINESPEC` подібно як для `plot`.

Приклад 7:

Побудова радіус-векторів для вектора z , заданого комплексними числами

```
>> z = [- 1 + 2i, -2 - 3i, 2 + 3i, 5 + 2i]
```

```
>> compass (z)
```

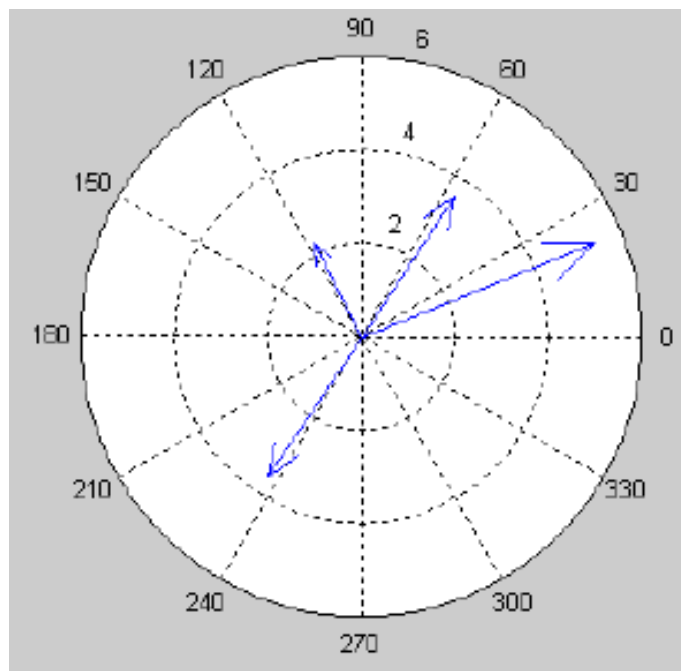


Рисунок 3.5 – Побудова радіус-векторів

Тривимірні графіки

Функція `meshgrid` для створення масивів тривимірної графіки.

Тривимірні поверхні зазвичай описуються функцією двох змінних $z(x, y)$. Побудова тривимірних графіків потребує не просто рядка значень x та y , тобто векторів x та y – вона потребує визначення двовимірних масивів X та Y — матриць. Для створення таких масивів служить функція `meshgrid`. Функція `meshgrid` записується в наступних формах:

– $[X, Y] = \text{meshgrid}(x, y)$ перетворює область, що задана векторами x та y , в масиви X та Y , які можуть бути використані для розрахунку функції двох змінних та побудови тривимірних графіків.

Рядки вихідного масиву X є копіями вектору x , а стовпці Y - копіями вектору y ;

- $[X, Y] = \text{meshgrid}(x)$ - аналогічна $[X, Y] = \text{meshgrid}(x, x)$;

- $[X, Y, Z] = \text{meshgrid}(x, y, z)$ повертає тривимірні масиви, використовувані для обчислення функцій трьох змінних і побудови тривимірних 3d-графіків.

Побудова графіків тривимірних поверхонь

Команда `plot3(...)` є аналогом команди `plot(...)`. Вона будує аксонометричне зображення тривимірних поверхонь і представлена наступними формами:

`plot3(x, y, z)` будує масив точок, представлених векторами x , y і z , сполучаючи їх відрізками прямих. Ця команда має обмежене використання;

`plot3(X, Y, Z)` - будує точки з координатами $X(i,:)$, $Y(i,:)$ і $Z(i,:)$ і сполучає їх відрізками прямих, де X , Y і Z - три матриці однакового розміру;

`plot3(X, Y, Z, S)` - забезпечує побудови аналогічні попередній команді, але із специфікацією стилю ліній і точок, що відповідає

Приклад 8:

Розглянемо приклад побудови тривимірної поверхні, що описується функцією

$$z(x, y) = x^2 + y^2$$

```
>> [x, y] = meshgrid ([ - 3 : 0. 15 : 3 ]);
>> z = x . ^ 2 + y . ^ 2;
>> plot3 (x, y, z)
```

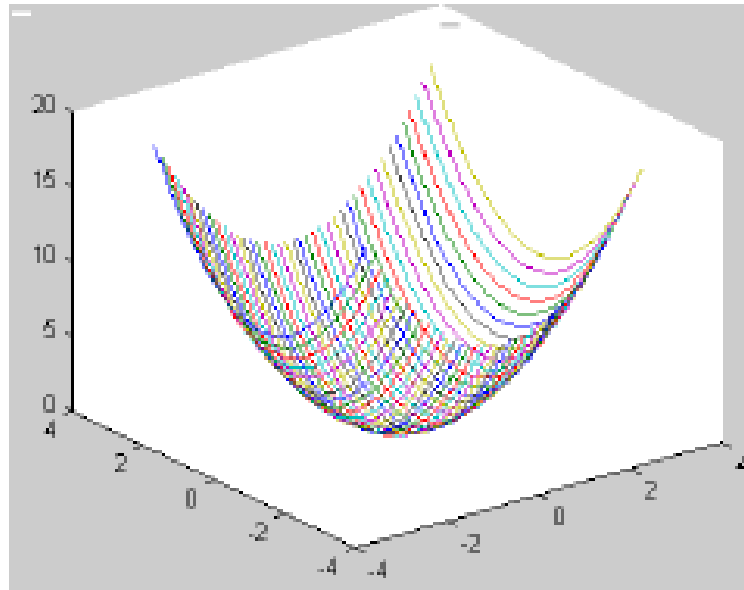


Рисунок 3.6 - Графік тривимірної поверхні

Приклад 9:

Побудувати графік 3d поверхонь (рис3.7).

```
>> % Побудова графіку 3d - поверхні кольоровими кухлями
>> [x, y] = meshgrid ([ - 3 : 0. 15 : 3 ]);
>> z = x . ^ 2 + y . ^ 2;
>> plot 3 (x, y, z, 'o')
```

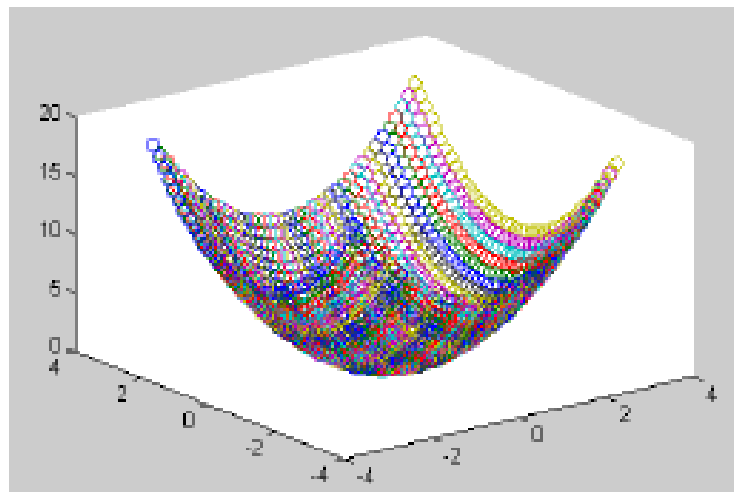


Рисунок 3.7 - Графік тривимірної поверхні, побудований різноколірними кухлями

Управління осями Функція *axis* має декілька можливостей для налаштування масштабу, орієнтації і коефіцієнта стискування.

Зазвичай MATLAB знаходить максимальне і мінімальне значення і вибирає відповідний масштаб і маркетування вісей. Функція *axis* замінює значення за умовчанням граничними значення, такими, що вводяться користувачем. *axis* ([*xmin*, *xmax*, *ymin*, *ymax*]) Наприклад, для того, щоб детальніше проаналізувати поведінку функції *sin* в околиці точки максимуму, можна встановити межі по осях координат : *axis* ([1.5, 2.5, 0.5, 2]). У функції *axis* можна також використати ключові слова для управління зовнішнім виглядом осей.

Назва функції	Дія
<i>Axis square</i>	Створює вісі <i>x</i> та <i>y</i> вісі однакової довжини
<i>Axis equal</i>	Створює окремі відмітки приросту для <i>x</i> та <i>y</i> однакової довжини
<i>Plot</i> (<i>exp</i> (1*t))	Наступна або за <i>Axis square</i> або <i>Axis equal</i> перетворює овал в правильне коло
<i>Axis auto</i>	Повертає значення за замовчуванням та переходить в автоматичний режим
<i>Axis on</i>	Вмикає позначення вісей та мітки проміжних ділень
<i>Axis off</i>	Вимикає позначення вісей та мітки проміжних ділень
<i>Grid off</i>	Вимикає сітку координат
<i>Grid on</i>	Вмикає сітку координат

Загальний заголовок для графіку виводиться функцією *title*, наприклад: *title* ('Function *sin* graph').

Завдання № 1 Побудувати графіки функцій.

Використати функцію plot. Треба самостійно задати крок і діапазон зміни аргументу

- $y = x^2 - 4$;

- $y = x^4 - 4x^2$;

- $y = \sin(x) + \log(x)$.

Завдання № 2 Побудувати криві, задані параметрично. Використати функцію plot. Рівняння кривої: $x(t) = t - \sin(t)$, $y(t) = 1 - \cos(t)$

- побудуйте цю криву при t , що змінюється в діапазоні $[-\pi; \pi]$.

- побудуйте цю криву при t , що змінюється в діапазоні $[0; 2\pi]$.

Завдання № 3 Побудувати криві, задані в полярній системі координат в діапазоні $[0; 2\pi]$. Використати функцію polar.

- крива задана рівнянням $r = 1 + \sin(t)$.

- крива задана рівнянням $r = 1 + \cos(2t)$.

Завдання № 4 Побудувати графіки тривимірних поверхонь. Використати функції meshgrid, plot3 .

- графік тривимірної поверхні, що описується функцією

$$z = \sqrt{|x|} - \sqrt{|y|}$$

- графік тривимірної поверхні, що описується функцією $z = y^2 - x^2$.

Побудова графіків векторів с комплексними елементами

Завдання № 5 Побудувати в декартовій системі координат графік вектору з комплексними елементами

$y = \sin x + i * \cos x$ в діапазоні $[0; 2\pi]$.

Використати функцію plot.

Завдання № 6 Побудувати графік радіус-векторів для вектору z , елементами якого є комплексні числа із завдання 10, тобто z_1, z_2, z_3 . Використати функцію compass.

ЛАБОРАТОРНА РОБОТА № 4

Випадкова функція

Випадкова функція, функція довільного аргументу t (задана на безлічі T його значень і що сама приймає або числові значення або, більш у загальних рисах, значення з якогось векторного простору) така, що її значення визначаються за допомогою деякого випробування і залежно від його результату можуть бути різними, причому для них існує певний розподіл вірогідності. Якщо безліч T звичайно, то функція є кінцевим набором випадкових величин, які можна розглядати як одну векторну випадкову величину. З числа випадкова функція. з безконечним T найбільш вивчений найважливіший окремий випадок, коли t набуває числових значень і є часом; відповідна випадкова функція $X(t)$ тоді називається випадковим процесом (а якщо час t пробігає лише цілочисельні значення, то також і випадковою послідовністю, або тимчасовим рядом). Якщо ж значеннями аргументу t є крапки з деякої області багатовимірного простору, то випадкова функція. називається випадковим полем.

Побудувати масив вручну

```
clear all;
clc;
A=[1 5
    2 7
    3 9];           % Матриця з 2 стовбців по 10 елементів
s=size(A,1);
for i=1:s
    X(i) = A(i,1);
    Y(i) = A(i,2);
end

plot(X,Y,'r - x')
grid on
axis square
xlabel('Вхід')
```

```
ylabel('Вихід')
title('Стохастичний сигнал')
legend('Випадкова функція')
```

Побудувати випадкову функцію

```
clear all;
clc;
% Побудова двувимірного графіка по точкам
% 1. Задання масиву у вигляді матриці. При цьому треба пам'ятати, що X
- будемо брати з першого стовбчика, Y - із другого.
c=[1:100]; % створювання строки з шагом 1
a=c'; % транспонування - строка перетворюється в стовбчик.
Перевірити!!!!
% a=a(1:100,1); % Можна одразу задати матрицю з 100 елементів в
першому стовбці
s=size(a,1); % розмір стовбця. Звичайно, буде 100
b=rand(s,1)*100;% Створюємо матрицю з одного стовбця з s елементами
A=[a b]; % Матриця з 2 стовбців по 100 елементів

for i=1:s
    X(i) = A(i,1);
    Y(i) = A(i,2);
end

plot(X,Y,'r - x')
grid on
axis square
xlabel('Вхід')
ylabel('Вихід')
```

```
title('Стохастичний сигнал')
```

```
legend('Випадкова функція')
```

Побудова тривимірна по точкам

```
clear all;
```

```
clc;
```

```
A =[84 110 0.11  
    93 130 0.14  
    104 150 0.17  
    115 170 0.25  
    129 190 0.22  
    143 210 0.19];
```

```
B =[87 110 0.4  
    98 130 0.7  
    110 150 0.2  
    124 170 0.24  
    139 190 0.45  
    153 210 0.34];
```

```
C =[92 110 0.44  
    106 130 0.33  
    120 150 0.45  
    136 170 0.67  
    150 190 0.1  
    165 210 0.18];
```

```
for i=1:size(A,1)
```

```
    X(i) = A(i,1);
```

```
    Y(i) = A(i,2);
```

```
    Z(i) = A(i,3);
```

```
    X1(i) = B(i,1);
```

```
    Y1(i) = B(i,2);
```

```
    Z1(i) = B(i,3);
```

```
    X2(i) = C(i,1);
```

```
    Y2(i) = C(i,2);
```

```
    Z2(i) = C(i,3);
```

```
end
```

```

plot3(X,Y,Z,'k - +',X1,Y1,Z1,'r - x',X2,Y2,Z2,'g - *')
grid on
% axis([20 200 100 220 0.1 0.35])
axis square
xlabel('Струм, мА')
ylabel('Напруга, мВ')
zlabel('Третій параметр')
title('Залежність')
legend('Перша крива', 'Друга кривая', 'Головна кривая')

```

ЛАБОРАТОРНА РОБОТА № 5

Обробка даних

Мета роботи: ознайомитись з командами обробки масивів даних в системі MATLAB.

Знайомства зі способами диференціювання та інтегрування, а також апроксимації, інтерполяції та спектрального аналізу даних.

Програма роботи

1. Погодити з викладачем варіант завдання.
2. За допомогою системи MATLAB та згідно з інструкціями, які викладені в розділі «Загальні відомості про систему MATLAB», а також в пункті 2 «Методичні вказівки щодо виконання роботи» (див. нижче), виконати обчислення прикладів наведених в підпунктах 1–5 пункту 2.

Методичні вказівки щодо виконання роботи в системі MATLAB є велика кількість команд та засобів для обробки масивів числових даних. Перелічимо ці засоби та наведемо ряд прикладів. Нехай маємо масив даних Y , що складається з двох рядків та п'яти стовпців:

```
>> Y=[1 2 3 4 5; 3 4 2 1 7]
```



```

Y =
    1 2 3 4 5
    3 4 2 1 7

```

Якщо необхідно з'ясувати розмірність певного масиву даних та (можливо) використати ці дані для якогось обчислення, то можна скористатися наступною функцією: `size(Y)` – функція, призначена для визначення числа рядків і стовпців матриці `Y`. Вона формує вектор `[n, p]`, що містить ці величини:

```
>> [n, p] = size(Y) n = 2 p = 5
```

Наведемо ряд інших функцій:

`max(V)` – функція, де `V` – деякий вектор, видає значення максимального елемента цього вектору. Аналогічно, функція `min(V)` визначає мінімальний елемент вектору `V`. `mean(V)` і `std(V)` – функції, які визначають, відповідно середнє значення і середньоквадратичне відхилення від нього значень елементів вектору `V`. `sort(V)` – функція сортування, формує вектор, елементи якого розташовані в порядку зростання їх значень. `sum(V)` – обчислює суму елементів вектору `V`. `prod(V)` – видає твір усіх елементів вектору `V`. `sumsum(V)` – формує вектор того ж типу і розміру, будь-який елемент якого є сумою усіх попередніх елементів вектору `V` (вектор кумулятивної суми). `sumprod(V)` – створює вектор елементи якого є твором усіх попередніх елементів вектору `V`. Також, в системі MATLAB передбачені інші математичні функції: `diff(V)` – функція чисельного диференціювання, створює вектор, який має розмір на одиницю менший від розміру вектору `V`, елементи якого є різницею між сусідніми елементами вектору `V`.

Приклад:

```

>>v=[1 0.1 0.5 0.1 0.1 0.4];
>>diff(v)
- 0.9000 0.4000 -0.4000 0 0.3000

```

Зауважимо, що знайдене значення не є похідною в канонічному розумінні. Це лише різниця між сусідніми елементами вектору. Для знаход-

ження реальної похідної $y'(x)$ необхідно знайдене вище значення поділити на крок Δt . `trapz(x, y)` – чисельна інтеграція методом трапецій – обчислення площі під графіком функції $y(x)$, в якому сусідні точки, задані векторами x і y , сполучені відрізками прямих. Якщо перший вектор x не вказаний в зверненні, за умовчанням допускається, що крок інтеграції дорівнює одиниці. Приклад. Вичислимо інтеграл від функції $y = \sin(x)$ в діапазоні від 0 до π (його точне значення дорівнює 2).

```
>>x = 0: pi/100: pi;
>>y = sin(x); >>trapz(x, y)
1.9998
```

Апроксимація та інтерполяція даних. `polyfit(X, Y, n)` – поліноміальна апроксимація даних, які сформовані як деякий вектор Y , при деяких значеннях аргументу, які утворюють вектор X тієї ж довжини, що і вектор Y . Тут n – порядок апроксимуючого полінома. Результатом дії цієї процедури є вектор завдовжки $(n + 1)$ з коефіцієнтів апроксимуючого полінома.

Приклад:

```
>>x = [1 2 3 4 5 6 7 8];
>>y = [-1.1 0.2 0.5 0.8 0.7 0.6 0.4 0.1];
>>polyfit(x, y, 2)
ans = -0.1024 1.0357 -1.7750
```

Це означає, що задана залежність була апроксимована рівнянням 2-го порядку (параболою).

$$y(x) = -0,1024x^2 + 1,0357x - 1,775.$$

Одновимірну табличну інтерполяцію здійснює процедура `interp1`. Звернення до неї дозволяє додатково вказати метод інтерполяції в четвертому вхідному аргументі:

- 'nearest' – ступінчаста інтерполяція;
- 'linear' – лінійна;
- 'cubic' – кубічна;
- 'spline' – кубічними сплайнами.

Якщо метод не вказаний, здійснюється за умовчанням лінійна інтерполяція. Приклад.

Задамо значення x та y такі ж як і в попередньому прикладі. Знайдемо значення функції u при значенні $x = 4,5$.

```
>> interp1(x,y,4.5)
```

```
ans =
```

```
0.7500
```

Спектральний аналіз даних. Теоретичні відомості щодо прямих та зворотних перетворень Фур'є для розкладання певної функції в ряд гармонічних складових докладно викладається в літературі та лекціях. Тому для виконання завдання практикуму лише приведемо приклад програми на мові MATLAB для здійснення подібного обчислення.

```
T=2; % Проміжок часу виміру
h=0.001; % Крок за часом виміри
t=0:h:T; % Часовий діапазон
x=sin(2*pi*5*t)+cos(2*pi*12*t); % Вхідний сигнал
y=fft(x); % Фур'є-зображення вхідного сигналу
Df=1/T; % Крок по частоті
F=1/h; % Максимальна частота
f=0:Df:F; % Формування вектору частот
f1=-F/2:Df:F/2; % Перебудова вектору частот
v=fftshift(y); % Перебудова вектору Фур'є-зображення
a=abs(v); % Модуль перебудованого Фур'є-зображення
N=length(y); % Визначення числа елементів Фур'є-зображення
a=2*abs(v)/N; % Визначення реальних амплітуд гармонік
stem(f1, a); grid % Побудова графіку Результатом є дві гармоніки з амплітудою 1 і частотами 5 та 12Гц.
```

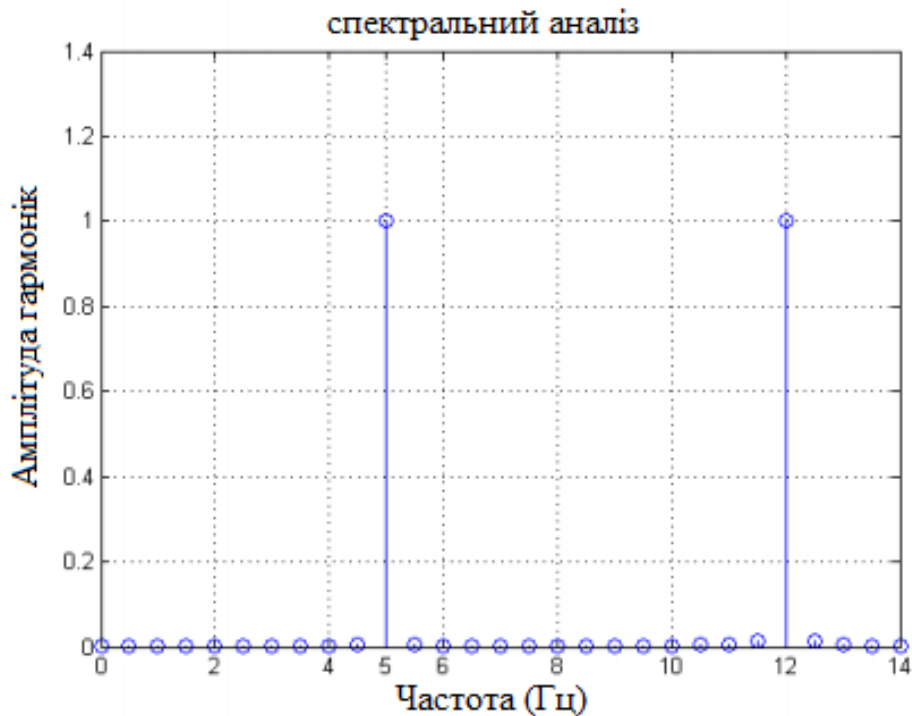


Рисунок 5.1 - Спектр гармонік

Завдання:

1. Знайти похідну від заданої функції. За допомогою функції $\text{plot}(x, y)$ зобразити на одному графіку початкову функцію і її похідну для аргументу який змінюється в заданому діапазоні і з заданим кроком. В таблиці 4.1 наведені варіанти завдань.

Таблиця 4.1

№ варіанту	Функція	Діапазон	Крок
1, 6, 11, 16	$\sin(x)$	$[0, 3\pi]$	$\pi/50$
2, 7, 12, 17	$\cos(x)$	$[0, 5\pi]$	$\pi/30$
3, 8, 13, 18	$\lg(x)$	$[1, 20]$	0,01
4, 9, 14, 19	e^{x-5}	$[0, 8]$	0,01
5, 10, 15, 20	$\ln(x)$	$[5, 23]$	0,02

2. Проінтегрувати задану функцію для аргументу який змінюється в заданому діапазоні і з заданим кроком. В таблиці 4.2 наведені варіанти завдань.

Таблиця 5.1

<i>№ варіанту</i>	<i>Функція</i>	<i>Діапазон</i>	<i>Крок</i>
1, 6, 11, 16	x^2	$[0, 5]$	0,025
2, 7, 12, 17	$x + x^3$	$[1, 8]$	0,05
3, 8, 13, 18	x^3	$[2, 5]$	0,01
4, 9, 14, 19	$2x^2 + 5$	$[1, 15]$	0,1
5, 10, 15, 20	$\sin(2x)$	$[0, 5\pi]$	$\pi/100$

3. Для заданої табличної функції $y(x)$ знайти аналітичний вираз використовуючи поліноміальну апроксимацію. Порядок апроксимаційного поліному заданий для кожного варіанту в табл. 4.3.

Функція $y(x)$:

x	1	5	8	11	15	18	22	27	34
y	-12	-20	-6	3	10	12	14	7	1

Таблиця 4.3

<i>№ варіанту</i>	<i>Порядок поліному</i>
1, 6, 11, 16	2
2, 7, 12, 17	3
3, 8, 13, 18	4
4, 9, 14, 19	5
5, 10, 15, 20	6

4. Для заданої табличної функції $y(x)$ знайти значення цієї функції в заданих точках використовуючи певний метод одномірної інтерполяції.

Точки в яких потрібно знайти значення функції, а також тип інтерполяції для кожного варіанту наведено в табл. 4.4.

Функція $y(x)$:

x	-8	-5	0	1	6	10	14	20	25
y	5	3	7	11	20	25	12	22	24

Таблиця 5.4

<i>№ варіанту</i>	<i>Точки x_i</i>	<i>Тип інтерполяції</i>
1, 6, 11, 16	-7; -2; 3; 12; 22	linear
2, 7, 12, 17	-6; -1; 0,5; 11; 19	nearest
3, 8, 13, 18	-4; -1.5; 2; 13; 21	cubic
4, 9, 14, 19	-7,5; -3; 5; 11; 23	slpine
5, 10, 15, 20	-6; -4; 6; 13,4; 24	linear

5. Розкласти наведену функцію в заданому діапазоні аргументу на спектр гармонік. Визначити амплітуди і частоти цих гармонік. Варіанти завдань наведені в табл. 5.5.

Таблиця 5.5

<i>№ варіанту</i>	<i>Функція</i>	<i>Діапазон</i>	<i>Крок</i>
1	$2 \sin(x) + 5 \cos(3x)$	$[0, 6,28]$	0,001
2	$3 \sin(2x) + 5 \cos(7x)$	$[0, 12,57]$	0,002
3	$7 \sin(2x) + 5 \cos(4x)$	$[0, 18,85]$	0,001
4	$12 \sin(4x) + 15 \cos(3x)$	$[0, 12,57]$	0,002
5	$11 \sin(2x) + 15 \cos(3x)$	$[0, 18,85]$	0,005
6	$22 \sin(4x) + 15 \cos(3x)$	$[0, 25,13]$	0,005
7	$9 \sin(2x) + 7 \cos(4x)$	$[0, 12,57]$	0,002
8	$6 \sin(6x) + 3 \cos(2x)$	$[0, 18,85]$	0,001
9	$10 \sin(5x) + 12 \cos(3x)$	$[0, 12,57]$	0,002
10	$4 \sin(x) + 8 \cos(4x)$	$[0, 6,28]$	0,001
11	$5 \sin(3x) + 10 \cos(4x)$	$[0, 18,85]$	0,005
12	$4 \sin(5x) + 5 \cos(17x)$	$[0, 25,13]$	0,002
13	$7 \sin(2x) + 4 \cos(8x)$	$[0, 12,57]$	0,001
14	$13 \sin(2x) + 15 \cos(5x)$	$[0, 18,85]$	0,005
15	$11 \sin(x) + 9 \cos(8x)$	$[0, 6,28]$	0,001
16	$18 \sin(5x) + 5 \cos(x)$	$[0, 6,28]$	0,001
17	$9 \sin(2x) + 15 \cos(7x)$	$[0, 12,57]$	0,002
18	$3 \sin(4x) + 7 \cos(12x)$	$[0, 18,85]$	0,005
19	$10 \sin(3x) + 15 \cos(4x)$	$[0, 18,85]$	0,005
20	$17 \sin(2x) + 14 \cos(7x)$	$[0, 25,13]$	0,001

ЛАБОРАТОРНАЯ РАБОТА № 6

Основи програмування

Мета роботи: *ознайомитись з елементами мови програмування системи MATLAB. Навчитись працювати з М-файлами, створювати файли-функції та Script-файли.*

Програма роботи

1. Погодити з викладачем варіант завдання.
2. За допомогою системи MATLAB та згідно з інструкціями виконати завдання.

Методичні вказівки щодо виконання роботи

Створення М-файлів. Програми на мові MATLAB мають два різновиди – Script-файли (файли-сценарії, або керуючі програми) і файли-функції (процедури). Обидва різновиди повинні мати розширення імені файлу .m. Для створення М-файлу необхідно з меню File командного вікна системи MATLAB вибрати New > Blank M-file. Після чого з'являється вікно редактора M-file Editor (рис. 5.1).



Рисунок 6.1 - Вікно редактора М-файлів

За допомогою Script -файлів оформляють основні програми, які керують організацією усього обчислювального процесу від початку до кінця, а також оформлюють окремі частини основних програм. Файли-функції служать для оформлення окремих процедур і функцій (тобто частин програми розрахованих на неодноразове використання). Ці функції не можуть бути виконані без попереднього завдання значень вхідних змінних). Головною зовнішньою відмінністю текстів цих двох видів файлів є те, що файл-функції має перший рядок виду

```
function [<ПКВ>] = <ім'я процедури >(<ПВВ>)
```

де ПКВ – Перелік Кінцевих Величин,

ПВВ – Перелік Вхідних Величин.

Script-файли такого рядка не мають. Якщо перелік кінцевих (вихідних) величин (ПКВ) містить тільки один об'єкт, то файл-функція є звичайною функцією (однієї або декількох змінних). Перший рядок в цьому випадку має вигляд:

```
function <ім'я змінної> = <ім'я процедури >(<ПВВ>)
```

Якщо ж в результаті виконання файл-функції має бути визначені декілька об'єктів (матриць), то така файл-функція є процедурою (підпрограмою). Загальний вигляд першого рядка в цьому випадку стає таким:

```
function [y1, y2, ..., yn] = <ім'я процедури >(<ПВВ>)
```

тобто ПКВ y_1, y_2, \dots, y_n має бути представлений як вектор-рядок з елементами y_1, y_2, \dots, y_n (усі вони можуть бути матрицями).

Приклад. В якості прикладу розглянемо складання М-файлу для функції

$$y = f_1(x) = d^3 \cdot ctg(x) \cdot \sqrt{\sin^4(x) - \cos^4(x)}$$

У вікні текстового редактора треба набрати такий текст зображений на рис. 5.2. Після цього необхідно зберегти цей текст у файлі під ім'ям F1.m. Тепер в командному вікні системи MATLAB введемо команду

```
>>y = F1(5, 0.9)
```

Отримаємо відповідь:

```
y =
-0.1975
```

Аналогічно можна отримати вектор відразу усіх значень вказаної функції при різних значеннях аргументу:

```
>>z=1:0.1:1.8;
>>m=F1(z, 1)
```

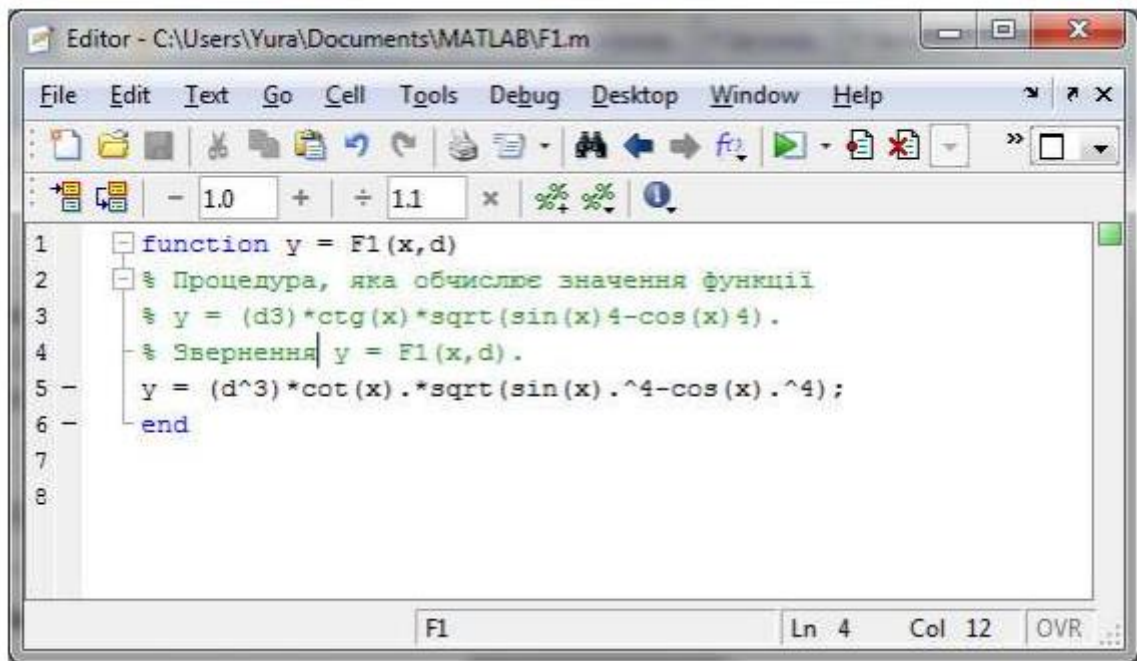


Рисунок 6.2 - Файл-функція (приклад)

Щоб отримати інформацію про створену процедуру, досить набрати в командному вікні команду:

```
>>help F1
```

Після цього в командному вікні буде виведений текст, який написаний у файлі F1.m після знаку % (див. рис. 6.2).

Побудова Script-файлів.

Розглянемо способи побудови власних програм на прикладі організації змінювання даних в діалоговому режимі. Для організації подібної програми знадобляться оператори введення та виведення даних, оператори управління

обчислювальним процесом, оператори порівняння та алгоритм організації повторень в Script – файлах.

Розглянемо ці оператори окремо.

Оператори виведення та введення даних.

Якщо усі змінні, що виводяться, мають однаковий формат (числовий або текстовий), то щоб вивести значення декількох змінних в один рядок необхідно об'єднати відповідні змінні у вектор та застосувати оператор `disp`. `disp` (`[x1 x2 ... x]`)

Приклад:

```
>>x1=1.24; x2=-3.45; x3=5.76; x4=-8.07;
>>disp([x1 x2 x3 x4]) 1.2400 -3.4500 5.7600 -8.0700
```

Або для текстових змінних, наприклад:

```
>>x1='psi'; x2='fi'; x3='teta'; x4='w1';
>>disp([x1 x2 x3 x4]) psi fi teta w1
```

Якщо усі змінні, що виводяться, мають різний формат, то щоб вивести значення декількох змінних в один рядок необхідно використовувати функцію `sprintf`. Звернення до неї має вигляд:

`Y = sprintf ('<текст1> %g <текст2>' X).`

В результаті виходить текстовий рядок `Y`, що складається з тексту, вказаного в , і значення числової змінної `X` у відповідності з форматом `%g`, причому текст з фрагмента розташовується після значення змінної `X`.

Приклад:

```
>>x = 5;
>>sprint ('Параметр 1 = %g Volt ', x)
```

Параметр 1 = 5 Vol

`t input` – оператор введення інформації з клавіатури в діалоговому режимі.

`x = input('запрошення')`

При цьому, виконання операторів програми припиняється. ПК переходить в режим очікування закінчення введення інформації з клавіатури. Після

закінчення введення з клавіатури введена інформація запам'ятовується в програмі під ім'ям "x", і виконання програми триває.

`menu` – оператор, який створює поточне вікно меню користувача.

Синтаксис:

`k=menu ('Заголовок меню', 'Альтернатива1', 'Альтернатива2', ..., 'Альтернатива n')`

Таке звернення призводить до появи на екрані меню, що зображене на рис.5.3. Виконання програми тимчасово призупиняється, і система чекає вибору однієї з кнопок меню з альтернативами. Після відповіді початковому параметру "k" привласнюється значення номера обраної альтернативи (1, 2, ..., n)

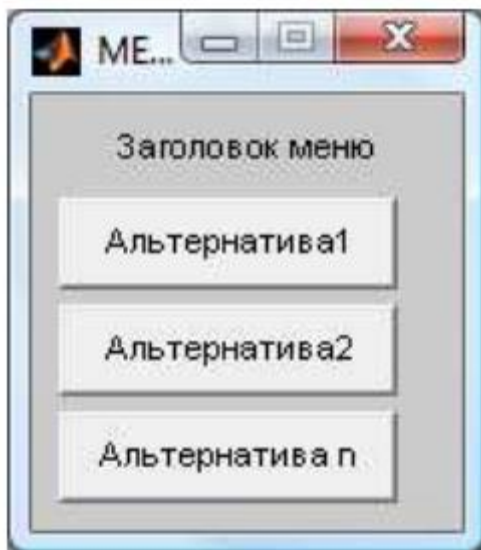


Рисунок 6.3 - Меню користувача, створене оператором `menu`

Оператори управління обчислювальним процесом.

До операторів управління обчислювальним процесом відносять: оператори умовного переходу і оператори організації циклічних процесів.

Усі оператори циклу і умовного переходу побудовані в MATLAB у вигляді складеного оператора, який починається одним із службових слів `if`, `while`, `switch` або `for` і закінчується службовим словом `end`. Оператор умовного переходу Конструкція:

`if <умова>`

```

    <оператори1>
else
    <оператори2>
end

```

Допустима ще одна конструкція оператора умовного переходу:

```

if <умова>
    <оператори1>
else if <умова2>
    <оператори2>
else if <умова3>
    <оператори3>
...
else
    <оператори>
end

```

Оператор `elseif` виконується тоді, коли `<умова1>` не виконана.

В якості умови використовуються вирази типу:

```
<ім'я змінної1> <операція порівняння> <ім'я змінної2>
```

Операції порівняння в мові MATLAB:

< - менше;

> - більше;

<= - менше або дорівнює;

>= - більше або дорівнює;

= - дорівнює;

~ = - не дорівнює.

Оператори циклу

У мові MATLAB є два різновиди операторів циклу - умовний і арифметичний. Оператор циклу з передумовою має вигляд:

```
while <умова>
```

<оператори>

end

Оператори усередині циклу виконуються лише у разі, якщо виконана умова записане після слова while. При цьому серед операторів усередині циклу обов'язково мають бути такі, які змінюють значення однієї зі змінних, вказаних в умові циклу.

Приклад обчислення значень синуса від 0.2 до 4 з кроком 0.2:

```
i = 1;
```

```
while i <= 20
```

```
x = i/5;
```

```
si = sin(x);
```

```
disp([x, si]) i = i+1;
```

```
end
```

Арифметичний оператор циклу має вигляд:

```
for <ім'я > = <НПЗ > : <К > : <КЗ >
```

```
<оператори>
```

```
end
```

де <ім'я > – ім'я змінної циклу ("лічильника" циклу); <НПЗ > – задане початкове значення цієї змінної; <К > – значення кроку; <КЗ > – кінцеве значення змінної циклу. Якщо параметр не вказаний, за умовчанням його значення набуває рівним одиниці.

Для прикладу використовуємо попереднє завдання:

```
a = ['i', 'x', 'sin(x)'];
```

```
for i = 1:20
```

```
x = i/5;
```

```
si = sin(x);
```

```
if i==1
```

```
disp(a)
```

```
end
```

```
disp([i,x,si])
```

end

Організація повторення дій в Script-файлах.

Одним з головних завдань створення самостійної програми є забезпечення повернення до початку програми з метою продовження її виконання при нових значеннях початкових даних.

Нехай основні оператори створеної програми розташовані в Script -файлі з ім'ям "Yadro.m". Тоді схема забезпечення повернення до початку виконання цього Script -файла може бути, наприклад, такою:

```
k=1;
while k==1
Yadro k = menu('Що робити?','Продовжити роботу','Закінчити роботу');
end
```

Тоді, після першого виконання Script-файла "Yadro.m" на екрані з'явиться вікно меню (рис.5.4).

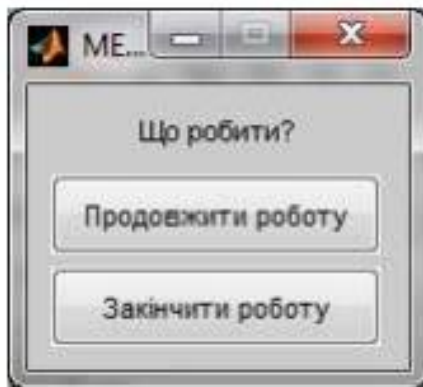


Рисунок 5.4 - Вікно меню

При натисненні кнопки першої альтернативи значення k залишиться рівним 1, цикл повториться, а при натисненні другої кнопки k стане рівним 2, цикл закінчиться і програма перейде до закінчення роботи.

Завдання:

1. Створити функцію від функції присвоївши їй унікальне ім'я. Обчислити значення цієї функції в заданому діапазоні і з заданим кроком. Побудувати

графік цієї функції за допомогою процедури fplot. Варіанти завдань вказані в табл.5.1.

Таблиця 6.1

<i>№ варіанту</i>	<i>Функція</i>	<i>Діапазон</i>	<i>Крок</i>
1	$\frac{x^2}{1+0,25\sqrt{x}}$	[1,1; 3,1]	0,2
2	$\frac{x^3-0,3x}{\sqrt{1+2x}}$	[2,05; 3,05]	0,1
3	$\frac{2e^{-x}}{2\pi+x^3}$	[0; 1,6]	0,16
4	$\frac{\cos \pi x^2}{\sqrt{1-3x}}$	[-1; 0]	0,1
5	$\sqrt{1+4x} \cdot \sin \pi x$	[0,1; 0,8]	0,07
6	$\frac{e^{x/3}}{1+x^2}$	[1,4; 2,4]	0,1
7	$e^{-2x} + x^2 - 1$	[0,25; 2,25]	0,2
8	$(e+x) \cdot \sin(\pi\sqrt{x-1})$	[1,8; 2,8]	0,1
9	$\sqrt{3+2x} \cdot \operatorname{tg}\left(\frac{\pi x^3}{2}\right)$	[0,1; 0,9]	0,08
10	$\sqrt{2+3x} \cdot \ln(1+3x^2)$	[-0,1; 0,9]	0,1
11	$\sqrt[3]{x^2+3} \cdot \cos\left(\frac{\pi x}{2}\right)$	[1; 2,5]	0,15
12	$(4+7x) \cdot \sin(\pi\sqrt[3]{1+x})$	[0; 7]	0,7
13	$e^{-x^2} \cdot (1+3x-x^2)$	[0; 2]	0,2
14	$x^3-3x+\frac{8}{\sqrt{1+x^2}}$	[0; 1,7]	0,17

<i>№ варіанту</i>	<i>Функція</i>	<i>Діапазон</i>	<i>Крок</i>
15	$\sqrt{\frac{e^{\sqrt{2\pi x}} - e^{-\sqrt{2\pi x}}}{2}}$	[0; 1,2]	0,12
16	$\sqrt{\frac{e^{\frac{x}{\sqrt{2\pi}}} - e^{-\frac{x}{\sqrt{2\pi}}}}{2}}$	[0,5; 1,5]	0,1
17	$\frac{x^3 + 2x}{\sqrt{1 + e^x}}$	[-0,2; 0,8]	0,1
18	$\sqrt{1 + 2x^2} \cdot \sin\left(\frac{3x}{2}\right)$	[2; 4]	0,2
19	$\sqrt{3x^2 + 5} \cdot \cos\left(\frac{\pi x}{2}\right)$	[0,5; 1,5]	0,1
20	$\cos\left(e^{-\sqrt[3]{3x}}\right)$	[0,2; 1,5]	0,1

2. Скласти програму яка виконує:

- а) діалогове введення-змiна-вивiд на екран значень дiапазону змiни аргументу функцiї створеної в завданнi 1;
- б) виведення графiку цiєї функцiї при змiнених параметрах.

ЛІТЕРАТУРА

1. Мороз І.О. Основи електродинаміки. Магнітостатика: навчальний посібник (гриф МОН України лист №1/11-6715 від 21 липня 2010 р.) / І.О. Мороз. – Суми: Видавництво «МакДен», 2011. – 162 с.
2. Мороз І.О. Електромагнітні хвилі. Текст лекцій (електронний посібник), СумДПУ імені А.С.Макаренка, 2013. Колонтаєвський Ю. П., Сосков А. Г. Електроніка і мікросхемотехніка. — К.: Каравела 2006.
3. Мороз І.А., Іваний В.С., Холодов Р.И. Основы специальной теории относительности. – Сумы: из-во СумГПУ ім.А.С.Макаренка, 2007. Прохоров Е.Д. Твердотіла електроніка : навч. посіб. Харків : ХНУ ім. В.Н.Каразіна, 2007, 1 прим.
4. Мороз І.О. Основи електродинаміки. Електростатика: навчальний посібник (гриф МОН України лист №1/11-6714 від 21 липня 2010 р.) / Мороз І.О. – Суми: Видавництво «МакДен», 2011. – 162 с.
5. Панащевний Б. І., Свергун Ю. Ф. Загальна електротехніка, теорія і практикум. — К.: Каравела, 2004. 2. Колонтаєвський Ю. П., Сосков А. Г. Електроніка і мікросхемотехніка. — К.: Каравела 2006
6. Gilat A. MATLAB: An introduction with Applications. – John Wiley & Sons, 2004.
7. Панасенко М.В., Сенько Є.В., Юрченко М.М. Електроніка і мікросхемотехніка : Аналогові та імпульсні пристрої : Підручник для вnz. Харків : ФОЛІО, 2002, 7 прим.
8. Матвійків М. Д. Елементи та компоненти електронних пристроїв: Підручник / М. Д. Матвійків , Б. С. Вус, О. М. Матвійків. – Львів: Видавництво Львівської політехніки, 2015. – 496 с.