МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Київський національний університет будівництва і архітектури

**МЕТОДИ ЕКСПЕРТНИХ ОЦІНОК**

**У СИСТЕМАХ ПРИЙНЯТТЯ РІШЕНЬ**

Лабораторний практикум

*Рекомендовано вченою радою Київського національного університету*

*будівництва і архітектури як навчальний посібник*

*для здобувачів першого (бакалаврського) рівня вищої освіти*

*галузі знань 12 «Інформаційні технології»*

*спеціальностей 122 «Комп’ютерні науки» та*

*126 «Інформаційні системи і технології»*

Київ 2025 р.

УДК 005.53(075)

М 54

Автори: Терентьєв О. О., д-р техн. наук, професор;

Серпінська О. І., старший викладач;

Баліна О. І., канд. техн. наук, доцент;

Безклубенко І. С., канд. техн. наук, доцент

Рецензенти: *Ю. П. Буценко*, канд. фіз.-мат. наук, доцент,

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»;

*І. М. Доманецька*, канд. техн. наук, доцент,

Київський національний університет імені Тараса Шевченка

Ю.В. Рябчун, PhD, доктор філософії в галузі інформаційних

систем і технлогій, доцент,

Київський національний університет будівництва і

архітектури

*Затверджено на засіданні вченої ради Київського національного*

*університету будівництва і архітектури, протокол № 6 від 27 березня*

*2025 року.*

М54 Методи експертних оцінок у системах прийняття рішень.

Лабораторний практикум: навчальний посібник /О. І. Серпінська та ін. – Київ : КНУБА, 2025. – 92с.

ISBN 978-966-279-2

Містить необхідні теоретичні відомості, рекомендації для виконання лабораторних робіт, варіанти завдань для самостійної роботи студентів і контрольні запитання до кожної теми.

Призначений для здобувачів галузі знань 12 “Інформаційні технології” спеціальностей 122 “Комп’ютерні науки” та 126 “Інформаційні системи і технології” освітньо-кваліфікаційного рівня “бакалавр”.

УДК 005.53 (075)

© О. І. Серпінська та ін., 2025

© КНУБА, 2025

ЗМІСТ

Вступ …………………………………………………………………………4

Лабораторні роботи № 1,2 ……………………….…………..……………. 6

Основні теоретичні відомості…………………………………………... 6

Порядок виконання роботи……..…………………………………….. 15

Вимоги до звіту…………………………..…………………………..... 15

Контрольні запитання…………….…………………………………… 16

Варіанти завдань до виконання лабораторних робіт 1 і 2……………19

Основні теоретичні відомості………………………………………….19

Порядок виконання роботи…………………………………………….21

Вимоги до звіту ………………………………………………………...53

Контрольні запитання…………………………………………………..53

Варіанти завдань до лабораторної роботи 3…….…………………….54

Варіанти завдань до лабораторної роботи 4 ………………………….20

Лабораторна робота № 5………………………………………………….. 59

Основні теоретичні відомості………………..…………………………59

Порядок виконання роботи……………………………………………..72

Вимоги до звіту………………………………………………………….73

Контрольні запитання …………………………………………………..73

Варіанти завдань……...…………………………………………………73

Лабораторна робота № 6………………………………………………….. 76

Основні теоретичні відомості……………………………………….....76

Порядок виконання роботи………………..………………………….. 84

Вимоги до звіту………………………………………………………... 87

Контрольні запитання…………………………………………………. 87

Варіанти завдань………………………………………………………. 88

**ВСТУП**

Метод експертних оцінок використовується в умовах часткової або повної невизначеності, яка може виникнути: за відсутності достовірної інформації за досить тривалий період; при наявності інформації, що відбиває тільки якісну сторону явищ, і неможливості кількісної характеристики всіх факторів, що роблять істотний вплив; в умовах нестійкого розвитку і порушення інерції в динаміці процесів і явищ; в процесах, напрямки розвитку яких залежать від прийнятих рішень, і, отже, далеких від об'єктивності; при аналізі якісно нових процесів і явищ. Запропонований курс робить акцент на застосуванні методів експертних оцінок в системах підтримки і прийняття рішень в умовах невизначеності та кризових станів.

**Мета дисципліни** - формування у майбутніх фахівців комплексу компетенцій, які дозволять їм у майбутній професійній діяльності застосовувати знання щодо створення і використання систем підтримки прийняття рішень для накопичення та математичної обробки даних у процесі прийняття управлінських рішень, володіння єдиної державної інформаційної системи діагностики технічного стану будівель; освоєння методичних підходів до створення інформаційної системи діагностики технічного стану; володіння єдиної інформаційної бази даних, що дала б можливість порівнювати результати обстежень. Забезпечення довготривалої експлуатації будівель за рахунок використання моделей та методів системи діагностики технічного стану є актуальною проблемою, що потребує застосування ефективних рішень на всіх етапах його життєвого циклу.

**Основні завдання дисципліни** “Методи експертних оцінок в системах прийняття рішень” – вивчення теоретичних основ створення систем підтримки прийняття рішень; набуття практичних навичок оволодіння методами пошуку найкращого або прийнятного способу дій для досягнення цілей, методами підтримки прийняття рішень в умовах слабо структурованих або неструктурованих ситуацій; формулювання вимог до систем підтримки прийняття; формування навичок використання систем підтримки прийняття рішень для вирішення прикладних завдань; проведення порівняльного аналізу й оцінки ринку СППР; вивчення методів і засобів побудови експертних систем.

Лабораторна робота (ЛР) є важливою складовою навчального процесу, що дозволяє здобувачам закріпити та поглибити теоретичні знання, отримані під час лекційних занять, шляхом практичного застосування їх для вирішення конкретних задач. Виконання ЛР сприяє розвитку аналітичного мислення, поглибленого аналізу та вдосконалення методів, моделей та інструментів ІСППР із використанням новітніх інформаційних систем та сучасних програмних продуктів, таких як Ms Excel та Mathlab, самостійного пошуку та обробки інформації, а також навичок ефективного представлення результатів своєї діяльності.

Як правило, рекомендації для виконання кожної лабораторної роботи розділені на дві складові: першу частину призначено для демонстрації виконання пропонованих завдань. У більшості випадків наведено необхідні теоретичні відомості. Друга частина містить завдання для виконання студентами самостійно згідно варіантів, що призначаються викладачем відповідно до списку групи.

Під час виконання передбачених даними рекомендаціями завдань студент формує звіт, в якому повинні бути такі елементи:

* номер та назва лабораторної роботи;
* зміст завдання, що виконується;
* результати розрахунків та (або) скріншоти отриманих зображень згідно варіанту.

**Лабораторні роботи № 1,2**

**ВИКОРИСТАННЯ ТА ДОСЛІДЖЕННЯ СИСТЕМ ЕКСПЕРТНОГО ОЦІНЮВАННЯ. МЕТОДИ РАНЖУВАННЯ І ПОПАРНИХ ПОРІВНЯНЬ**

**Мета роботи** – навчитися отримувати, обробляти та аналізувати знання експертів при автоматизації технологічних процесів, дослідити особливості систем експертного оцінювання.

**Основні теоретичні відомості**

При автоматизації складних технологічних систем виникають наступні задачі:

- оцінювання тісноти зв’язків між технологічними змінними, як вимірюваними, так і не вимірюваними;

- визначення факторів для включення у математичну модель;

- визначення пріоритетності встановлення контрольно – вимірювальної апаратури;

- визначення керувальних змінних;

- визначення причин аварійних ситуацій тощо.

Ці задачі можуть бути успішно розв’язані лише з урахуванням досвіду фахівців. Отже, класифікацію ситуацій й усунення тих з них, які можуть призвести до руйнування самої системи, доцільно здійснювати на основі систем експертних знань. Алгоритми і програмне забезпечення, які дають змогу отримати і обробити таку експертну інформацію, називаються ***системами експертного оцінювання*** (СЕО). При їх використанні експерти надають факторам, які впливають на певну систему, бали (так звані ***ранги***).

Правила виставляння рангів залежать від вибраного способу ранжування. В цій лабораторній роботі будуть досліджуватись два алгоритми: ***одночасного та попарного ранжування.***

На рис. 1.1 наведено загальну для обох способів схему алгоритму експертного оцінювання.

При ***одночасному ранжуванні*** експерт отримує для порівняння одразу весь перелік факторів і проставляє кожному з них відповідний ранг.

Алгоритм ***попарного ранжування (метод парних порівнянь)*** базується на тому, що експерт порівнює фактори попарно.

Алгоритм, наведений на рис.1.1, має в цілому лінійну структуру, але його використання передбачає декілька циклів. Вони викликані, по-перше, опитуванням декількох фахівців а, по-друге, неодноразовим їх опитуванням. Опишемо вищенаведений алгоритм спочатку на прикладі одночасного ранжування.

Крок 1. *Визначення критеріїв компетентності експертів*. При виборі експертів керуються предметом експертизи, її метою і вимогами до однорідності експертів.

У прикладі, який наведено у лабораторній роботі, критеріями компетентності стали стаж роботи на даному технологічному обладнанні та виробничий розряд (кількість критеріїв *KR*=2).

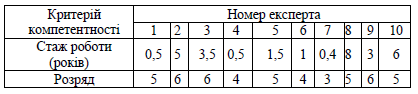
Крок 2. *Збирання даних про експертів*. У цій частині алгоритму до ЕОМ заводять дані про кожного експерта відповідно до вибраних критеріїв компетентності (кількість експертів *NE*=10). Приклад до блоку 2 наведено у табл. 1.1.



Рис.1.1. Схема алгоритму роботи системи експертного оцінювання

Таблиця 1.1.

Дані про компетентність експертів

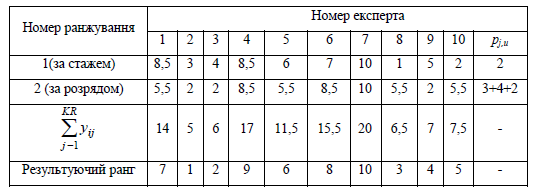


Крок 3. *Ранжування експертів*. У цьому блоці експертам надають ранги у відповідності до числових значень критеріїв їхньої компетентності (чим більші стаж та розряд, тим менші відповідні ранги). Ранжування виконують за кожним з критеріїв окремо. Якщо є однакові ранги, то переходять до нормальної матриці ранжування, а потім розраховують суму рангів за всіма ранжуваннями.

Результат такої обробки даних поданий у табл. 1.2.

Таблиця 1.2.

Нормальна матриця ранжування експертів



Крок 4. *Оцінка узгодженості ранжування експертів*. У табл. 1.2 величина – це кількість повторень *u*-о рангу у j-у рядку; - сума рангів, отриманих *і*-м експертом у *KR* ранжуваннях.

Для з’ясування узгодженості ранжування експертів визначають коефіцієнт конкордації і оцінюють його статистичну значущість.

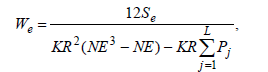
Приклад розрахунку коефіцієнта конкордації ранжування експертів:

1. розрахунок середнього рангу нормальної матриці рангів експертів:

б) обчислення суми квадратів різниць між членами сумарного ранжування і членами ряду, який складено із середніх значень:



в) визначення коефіцієнта конкордації ранжування експертів:



Де *L* – кількість рядків у табл. 1.2, які містять однакові («зв’язані») ранги, *L*=2.

Величину розраховують за формулою :



де *U* – кількість типів «зв’язаних» рангів у j- му рядку табл. 1.2 (для j=1, *U*=1; для j=2*U*=3).

Статистичну значущість коефіцієнта конкордації перевіряють за - критерієм (критерієм Пірсона). Розрахункове значення визначають за виразом:



При рівні значущості і ступені вільності *NU=NE*-1, табличне значення критерію дорівнює 14.68.

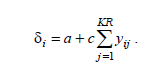
Отже, виконується умова



Таким чином, можна вважати узгодженим ранжування експертів при .

Крок 5. *Розрахунок «ваги» думок експертів*. Найдосвідченішим експертом вважається той, у якого сума рангів буде найменшою (у розглянутому прикладі це другий експерт). Його думці надається «вага» . Думка найменш досвідченого експерта (7-го) має «вагу» .

Для визначення «ваги» думок інших експертів використовують рівняння



Для визначення коефіцієнтів *a* та *с* запишемо систему



звідки a; с=-.

Для прикладу визначимо «вагу» думки 1-о експерта:



Крок 6. *Складання переліку факторів ранжування.* У список для ранжування включають фактори, які впливають на хід технологічного процесу.

Крок 7. *Ранжування факторів*. У цій частині алгоритму виконують власне ранжування як таке. Згідно з методом одночасного ранжування експерти ставлять у відповідність кожному фактору ранг – число з натурального ряду , де *K* – кількість факторів у списку. Правило ранжування таке – чим менше фактор впливає на технологію, тим більший ранг йому треба надати.

Значущість кількох факторів може бути однаковою, і їм надають однакові ранги. Приклад ранжування наведений у табл. 1.3.

Таблиця 1.3.

Результати ранжування факторів



У цьому ж блоці виконується перехід до нормального виду матриці рангів (див. табл. 1.4), тобто враховують той факт, що декілька факторів набули одного і того ж рангу.

Таблиця 1.4.

Нормальна матриця рангів



Крок 8. *Корекція результатів ранжування факторів*. Елементи нормальної матриці коригують з урахуванням «ваги» думок фахівців, які розраховують у блоці 3.

Основними показниками сили впливу фактору на процес є «зважені» суми рангів:

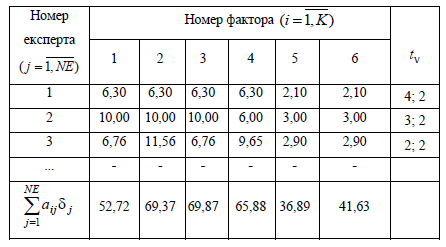


де - ранг, що був наданий *i* -у фактору *j* -м експертом без урахування його компетентності.

Результати експертного опитування, в яких врахована «вага» думок експертів, наведені у табл. 1.5.

Таблиця 1.5.

Таблиця «зважених» рангів

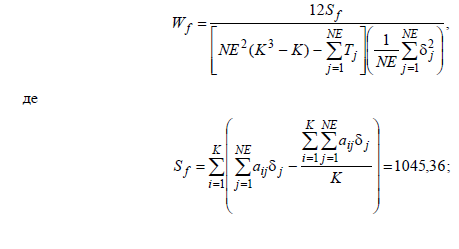


У табл. 1.5 параметр - кількість однакових рангів різних типів у рядку. Так, у першому рядку два різних типи: 6,30 та 2,10. Перших значень – 4, інших – 2; тому  Якщо немає однакових значень у рядку, то 

Крок 9. *Визначення сумарного ранжування факторів*. Виконують підсумовування «зважених» рангів по всіх експертах (див.  табл. 1.5).

Крок 10. *Вибір найбільш суттєвого фактора*. Умова вибору – найменше значення суми рангів (у прикладі це 5-й фактор).

Крок. 11. *Оцінка узгодженості ранжування факторів*. Обчислюють коефіцієнт конкордації за формулою, аналогічною (1.1):





*P* – кількість типів «зв’язаних» рангів в *j* -у рядку.

Після цього розраховують значення критерію Пірсона за формулою



Табличне значення критерію Пірсона визначають за рівнем значущості α і ступенем вільності *NF=K-1*=5. Так, для α=0,05 і *NF*=5 отримаємо  У цьому разі виконується умова

 (1.3)

і можна говорити про наявність узгодженості у думках експертів при ранжуванні факторів.

Крок 12. *Планування подальших дій.* Якщо умова (1.3) виконується, то задачу можна вважати розв’язаною. У тому випадку, коли умова (1.3) не виконується, то потрібно звернути увагу на правильність формулювання задачі дослідження, уточнити склад групи експертів і повернутися до початку розв’язання задачі (від блоку 1).

Розглянемо застосування алгоритму для метода попарного ранжування. Відмінності є для блоків 7-11, далі позначатимемо ці кроки індексом «п».

Крок 7п. Експерт згідно з цим методом оцінює більш значущий фактор одиницею (1), а інший нулем (0). Результат такого ранжування може бути подано таким, наприклад, як у табл. 1.6.

Таблиця 1.6.

Результати ранжування при парному порівнянні



Одиницю записуємо у комірку *ij* ( *i* – номер рядка, *j* – номер стовпчика), якщо фактору *і* віддаємо перевагу перед фактором *j* . Таку

таблицю заповнює кожний з *NЕ* експертів.

Крок 8п. Якщо треба врахувати компетентність експерта, то кожний

елемент таблиці перемножують на «вагу» думки відповідного експерта.

Крок 9п. Для визначення результату ранжування створюють нову

таблицю, у кожній її комірці вказують суму чисел, які знаходяться у

відповідних комірках всіх *NE* таблиць.

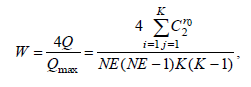
Крок 10п. Підраховують суму чисел у всіх комірках для кожного рядка  Чим більша ця сума, тим більше значення має *i* -й фактор.

Крок 11п. Для визначення узгодженості ранжування використовують

лише частину таблиці, розташовану або над головною діагоналлю, або під

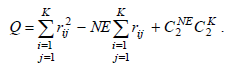
нею.

Коефіцієнт конкордації обчислюють за формулою



де  - біноміальний коефіцієнт; - число, що розміщено у комірці з координатами ij(i≠j).

Значення Q може бути отримане таким чином:



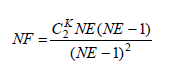
Для оцінювання значущості коефіцієнта узгодженості *W* використовують критерій 

Розраховують значення цього критерію так:



Далі порівнюють  з табличним значенням  для рівня значущості α і *NF* ступенів вільності.

Параметр *NF* обчислюють за формулою:



і округлюють до найближчого цілого.

**Порядок виконання роботи**

1. Сформулювати задачу дослідження. Визначити список факторів, які треба ранжувати за ступенем впливу на певний критерій.

2. Провести експертні опитування з врахуванням компетентності експертів та без нього. Роботу з програмами виконувати за зразками, наведеними в лабораторній роботі (див. рис.1.2 – 1.8 для одночасного ранжування та рис.1.9 – 1.11 для попарного).

3. Розрахувати показники експертизи за обома способами ранжування. Визначити узгодженість ранжувань факторів та експертів за заданими критеріями компетентності. Розрахунки виконати у середовищі *MS Excel*. 16

**Вимоги до звіту**

Опис предметної області, задача дослідження, початкові та нормальні матриці ранжування факторів та експертів, результати ранжування у числовому та графічному видах. Розрахунки, виконані у *MS Excel*.

За результатами лабораторної роботи дослідити суперечливість думок експертів. Зробити висновки з наступних питань:

а) про можливість роз’язання поставленої задачі аналітичним чи експериментально-статистичним методами;

б) про узгодженість думок експертів;

в) про узгодженість вибраних критеріїв компетентності;

г) про вплив критеріїв компетентності експертів на результати ранжування факторів;

д) про наявність суперечностей між результатами двох методів ранжування.

**Контрольні запитання**

1. Для чого використовують системи експертного оцінювання?

2. Які є СЕО? Як виконується ранжування в кожній з них?

3. Як визначити узгодженість думок експертів у кожній із систем?

4. Як визначити «вагу» думки експерта?

5. Як використати дані про компетентність експертів?

5. Які можуть бути подальші дії при неузгодженості роботи експертів?

6. Як подати результати експертизи графічно?

7. Як можна оцінити результативність експертизи за видом графічного зображення її результатів?

**Варіанти завдань до виконання лабораторних робіт 1 і 2.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № варіанту завдання |  | Метод експертних оцінок | Змістовна характеристика об’єктів оцінювання | Число об’єктів | Число експертів |
| 1 |  | Ранжування,  парних порівнянь | Функції ІТП житлового будинку | 8 | 7 |
| 2 |  | Ранжування, | Критерії оцінки архітектурно-планувальних рішень | 9 | 10 |
| 3 |  | парних порівнянь | Варіанти проектних рішень | 7 | 8 |
| 4 |  | Ранжування, | Критерії оцінки естетичного вигляду споруди | 8 | 8 |
| 5 |  | парних порівнянь | Методи сучасних ІТП | 8 | 4 |
| 6 |  | Ранжування,  парних порівнянь | Критерії оцінки конструктивних рішень споруди | 10 | 3 |
| 7 |  | Ранжування, парних порівнянь | Професійні дисципліни спеціальності ІСТ | 9 | 4 |
| 8 |  | Ранжування, парних порівнянь | Варіанти конструктивно-технологічних рішень | 7 | 5 |
| 9 |  | Ранжування, парних порівнянь | Функції ІТП промислового об’єкту | 7 | 6 |
| 10 |  | Ранжування, парних порівнянь | Варіанти спорядження | 8 | 14 |
| 11 |  | Ранжування, парних порівнянь | Напрями інвестицій | 8 | 8 |
| 12 |  | Ранжування, парних порівнянь | Цілі створення ІТП | 7 | 4 |
| 13 |  | Ранжування, парних порівнянь | Функції ІТП житлового будинку | 8 | 5 |
| 14 |  | Ранжування, парних порівнянь | Критерії оцінки архітектурно-планувальних рішень | 9 | 3 |
| 15 |  | Ранжування, парних порівнянь | Варіанти проектних рішень | 7 | 5 |
| 16 |  | Ранжування, парних порівнянь | Цілі створення ІТП | 7 | 13 |
| 17 |  | Ранжування, парних порівнянь | Критерії оцінки естетичного вигляду споруди | 8 | 5 |
| 18 |  | Ранжування, парних порівнянь | Методи сучасних ІТП | 8 | 4 |
| 19 |  | Ранжування, парних порівнянь | Критерії оцінки конструктивних рішень споруди | 10 | 3 |
| 20 |  | Ранжування, парних порівнянь | Професійні дисципліни спеціальності ІСТ | 9 | 4 |
| 21 |  | Ранжування, парних порівнянь | Варіанти конструктивно-технологічних рішень | 7 | 5 |
| 22 |  | Ранжування, парних порівнянь | Функції ІТП житлового будинку | 7 | 6 |
| 23 |  | Ранжування, парних порівнянь | Критерії оцінки архітектурно-планувальних рішень | 3 | 10 |
| 24 |  | Ранжування, парних порівнянь | Варіанти проектних рішень | 7 | 6 |
| 25 |  | Ранжування, парних порівнянь | Цілі створення ІТП | 8 | 4 |

**Лабораторні роботи № 3,4**

**ВСТУП ДО СИСТЕМИ ІНЖЕНЕРНИХ І НАУКОВИХ РОЗРАХУНКІВ MATLAB**

Мета роботи: Ознайомлення з основними операціями у середовищі MatLAB на прикладі використання стандартних функцій, побудови файлів-сценаріїв та створення функцій користувача.

**Основні теоретичні відомості**

Система MatLAB (скорочено від MATrix LABoratory – матрична лабораторія) є мовою технічних обчислень для виконання наукових та інженерних розрахунків. До основних переваг системи можна віднести таке:

– математичний апарат MatLAB наближений до звичного математичного апарату інженера, а правила виконання обчислень орієнтовані на роботу з матрицями та комплексними числами;

– мова програмування системи MatLAB є простою та зручною;

– MatLAB є відкритою системою, що передбачає можливість її доповнення створюваними користувачем власними функціями.

Робота у середовищі MatLAB може відбуватися у двох режимах:

1) в режимі калькулятора, коли формули вводяться в командному вікні, туди ж виводяться результати обчислення, користувач може присвоювати результати змінним, керувати пам’яттю, викликати та використовувати вбудовані функції, будувати графіки;

2) шляхом виконання програми, написаної користувачем на мові MatLAB.

MatLAB надає не тільки мову програмування високого рівня, а також має функції, властиві операційним системам. Він має багато можливостей, які забезпечують діалог з користувачем у режимі командного рядка або з графічним інтерфейсом, перегляд робочої області та шляхів доступу до файлів MatLAB, редактор m-файлів, можливість експорту та імпорту даних іншим апаратним та програмним системам. ,

До системи MatLAB разом з вбудованими функціями входять пакети прикладних програм (toolbox), в яких зібрані функції, орієнтовані на вирішення спеціалізованих інженерних задач, наприклад:

– Signal Processing Toolbox (Оброблення сигналів) – набір програм для детермінованого, статистичного та спектрального аналізу, апроксимації передавальних функцій, побудови фільтрів зі скінченною та нескінечною

імпульсними характеристиками, моделювання сигналів, оброблення звуку, модуляції, генерування стандартних сигналів тощо;

– Image Processing Toolbox (Оброблення зображень) – програми для візуалізації зображень різних форматів, обробки та аналізу зображень, фільтрації, роботи з кольорами;

– Statistics Toolbox (Статистика) – набір програм для оцінювання параметрів законів розподілу, функцій щільності ймовірностей, генерації випадкових чисел, побудови лінійних та нелінійних моделей випадкових сигналів, кластерного аналізу, перевірки гіпотез тощо;

– Mapping Toolbox (Робота з картами) – програми для оброблення картографічних даних;

– Symbolic Math Toolbox (Символьна математика) – програми для символьних обчислень та спрощень символьних виразів;

– Fuzzy Logic Toolbox (Нечітка логіка) – програми для реалізації апарату нечіткої логіки;

– Neural Network Toolbox (Нейронні мережі) – програми для створення нейронних мереж, їх навчання за різними алгоритмами і оцінювання їх роботи;

– Wavelet Toolbox (Вейвлет) – програми для проведення вейвлет-розкладання сигналів у базисах вейвлетів, створення вейвлет-фільтрів, стиснення і знешумлення сигналів за допомогою вейвлетів;

– Control System Toolbox (Системи управління) – набір програм для моделювання лінійних інваріантних в часі систем, аналізу моделей у часовій та частотній областях;

– Optimization Toolbox (Оптимізація) – одномірна та багатомірна мінімізація функцій, пошук нулів та екстремумів, вирішення задач лінійного та квадратичного програмування;

– Matrix functions - numerical linear algebra (Матричні функції – чисельна лінійна алгебра)

– набір програм для дослідження матриць, вирішення алгебраїчних рівнянь, обернення матриць, факторизації, обчислення власних векторів та власних значень тощо;

– Data Acquisition Toolbox (реєстрація даних) – програми для отримання даних з зовнішніх пристроїв.

Робоча область MatLAB – це область пам‘яті, у якій розташовані змінні, з якими працює система MatLAB.

Система MatLAB використовує поняття поточного каталогу під час сеансу роботи, та список шляхів доступу до файлів.

**Порядок виконання роботи**

1. Ознайомитися з командним вікном MatLAB, поняттями робочої області та шляхів доступу.

Після виклику Matlab із середовища Windows на екрані виникає вікно, зображене на рис. 1.

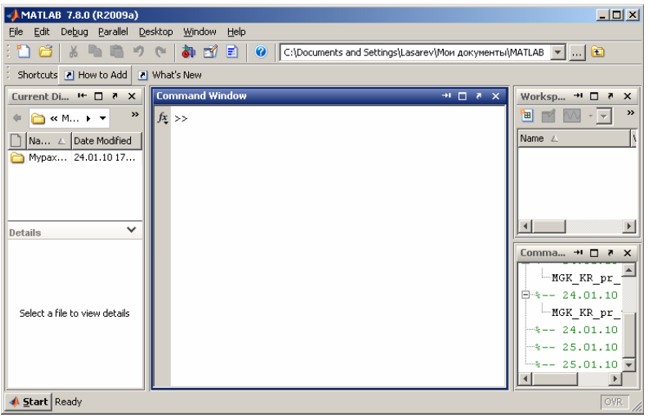


Рисунок 1. Вигляд повного вікна MatLab

Якщо закрити усі бокові допоміжні підвікна, залишиться одне вікно, яке називають "командним" (Command Window) середовища Matlab (рис. 2).

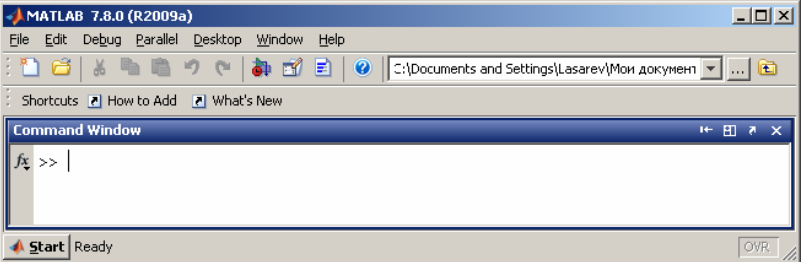


Рисунок 2. Командне вікно MatLab

Це вікно є головним у Matlab. У ньому виникають символи команд, що

набираються користувачем на клавіатурі дисплея, відображуються результати виконання цих команд, текст програми, яка виконується, і інформація про помилки виконання програми, розпізнані системою .

Ознакою того, що Matlab готова до сприйняття і виконання чергової команди, є поява в останньому рядку текстового поля командного вікна знака запрошення (>>), праворуч якого миготить вертикальна риса.

У верхній частині вікна (під заголовком) розміщений рядок меню, в якому містяться меню File, Edit, Debug, Parallel, Desktop, Windows, Help. Щоб відчинити якесь меню, потрібно встановити на ньому курсор миші і клацнути її лівою кнопкою.

Тут відзначимо лише, що для виходу із середовища Matlab достатньо відчинити меню File і обрати у ньому команду Exit MATLAB, або просто зачинити командне вікно, клацнувши лівою клавішою миші, коли курсор миші встановлений на зображенні верхньої крайньої правої кнопки цього вікна (з позначенням хрестика).

2. Ознайомитися з правилами введення змінних та констант, представлення даних у MatLAB.

**Введення дійсних чисел**

Введення чисел із клавіатури здійснюється по загальних правилах, прийнятих для мов програмування високого рівня:

для відділення дробової частини мантиси числа застосовується десяткова крапка (замість коми при звичайному записі);

десятковий показник числа записується у вигляді цілого числа після попереднього запису символу "е";

між записом мантиси числа й символом "е" (який відокремлює мантису

від показника) не повинно бути ніяких символів, включаючи і символ пропуску.

Якщо, наприклад, ввести в командному вікні Matlab рядок

120357.9245e-78, то після натискання клавіші <Enter> у цьому вікні виникне запис (рис. 3):

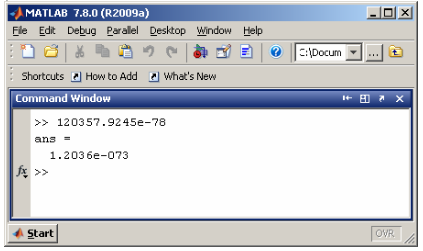


Рисунок 3. Введення і відображення чисел

Слід зазначити, що результат виводиться у виді (форматі), що визначається попередньо встановленим форматом подання чисел. Цей формат може бути встановлений за допомогою команди Preferences меню File (рис. 4).

Після її виклику на екрані з'явиться однойменне вікно (рис. 5). У поділі

Command Window одна з ділянок цього вікна має назву Numeric Format. Її призначено для встановлення і змінювання формату подання чисел, які виводяться в командне вікно в процесі розрахунків.

Передбачені такі формати:

Short (default) - стислий запис (застосовується за умовчанням);

Long - довгий запис;

Hex - запис у виді шістнадцяткового числа;

Bank - запис до сотих часток;

Plus - записується тільки знак числа;

Short E - стислий запис у форматі із плаваючою комою;

Long E - довгий запис у форматі із плаваючою комою;

Short G - друга форма стислого запису у форматі з плаваючою комою;

Long G - друга форма довгого запису у форматі з плаваючою комою;

Rational - запис у вигляді раціонального дробу.

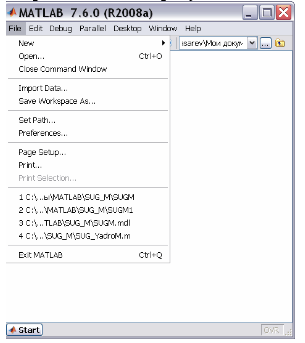


Рисунок 4. Меню «*File*»

Обираючи за допомогою мишки потрібний вид подання чисел, можна забезпечити надалі виведення чисел у командне вікно саме в цій формі.

Як видно з рис. 3, число, яке виведено на екран, не збігається з введеним. Це обумовлено тим, що встановлений за замовчуванням формат подання чисел (Short G) не дозволяє вивести більше 5 значущих цифр числа. Насправді введене число усередині Matlab зберігається з усіма введеними його цифрами. Наприклад, якщо обрати мишкою селекторну кнопку Long E (тобто установити цей формат подання чисел), то, повторюючи ті ж дії, отримаємо результат, відображений на рис. 6, де вже всі цифри відображені вірно.

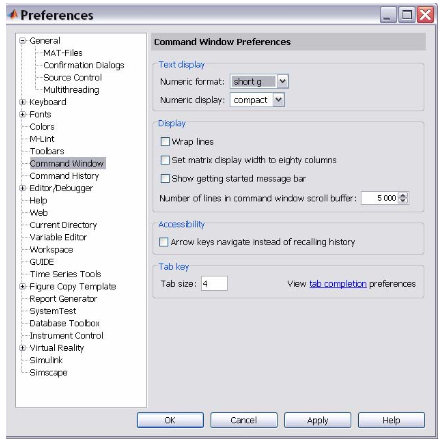


Рисунок 5. Вікно *Preference* меню «*File*»

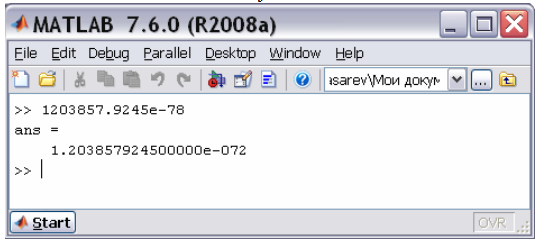


Рисунок 6. Виведення числа у форматі *Long E*

Слід пам’ятати:

- уведене число й результати всіх обчислень у системі Matlab зберігаються в пам'яті ПК із відносною похибкою біля (тобто з точними значеннями в 15 десяткових розрядах);

- діапазон подання модуля дійсних чисел лежить у проміжку між  і .

**Введення комплексних чисел**

Мова системи Matlab, на відміну від багатьох мов програмування високого рівня, містить у собі дуже просту в користуванні вбудовану арифметику комплексних чисел. Більшість елементарних математичних функцій побудовано у такий спосіб, що аргументи припускаються комплексними числами, а результати також формуються як комплексні числа. Ця особливість мови робить її дуже привабливою й корисною для інженерів і науковців.

Для позначення уявної одиниці в мові Matlab зарезервовано два імені i і j. Уведення із клавіатури значення комплексного числа здійснюється шляхом запису в командне вікно рядка виду:

*<ім’я комплексної змінної> = <значення ДЧ> + i [j] \*<значення УЧ>,*

де ДЧ – дійсна частина комплексного числа, УЧ – уявна частина. Приклад наведено на рис. 7:

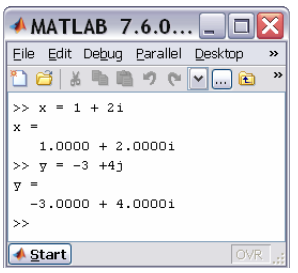


Рисунок 7. Введення комплексних чисел і виведення результату

З нього видно, у якому виді система виводить комплексні числа на екран (і "до друку").

**Введення векторів і матриць**

Початкові значення векторів можна задавати із клавіатури шляхом поелементного введення. Для цього в рядку треба спочатку вказати ім’я вектора, потім поставити знак присвоювання '=', а далі, – відкривальну квадратну дужку, а за нею ввести задані значення елементів вектора, відділяючи їх пропусками або комами. Закінчується рядок записом квадратної дужки, що закриває.

Наприклад, запис рядка V = [ 1.2 -0.3 1.2e-5] задає вектор V, що містить

три елементи зі значеннями 1.2, -0. 3 і 1.2е-5 (рис. 8):

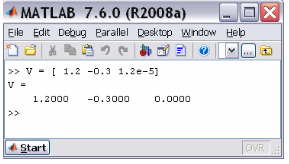


Рисунок 8. Введення вектора

Після введення вектора система виводить його на екран. Те, що в наведеному прикладі останній елемент виведений як 0, обумовлено встановленим форматом short, відповідно до якого виводяться дані на екран.

Довгий вектор можна вводити частинами, які потім об'єднувати за допомогою операції об’єднання векторів у рядок : v = [ v1 v2 ]. Приклад наведений на рис. 9.

Мова Matlab дає користувачеві можливість скороченого введення вектора, значення елементів якого є арифметичною прогресією. Якщо позначити: nz – початкове значення цієї прогресії (значення першого елемента вектора); kz – кінцеве значення прогресії (значення останнього елемента вектора); h – різницю прогресії (крок), то вектор можна ввести за допомогою короткого запису

V = nz : h : kz .

Наприклад, введення рядка V = - 0.1 : 0.3 : 1.4 приведе до результату,

показаному на рис. 10.

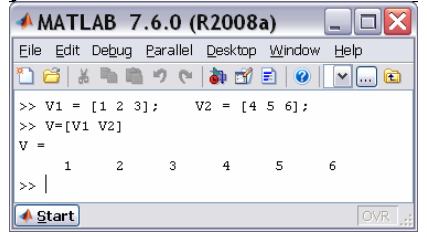


Рисунок 9. Конкатенція (об’єднання) векторів

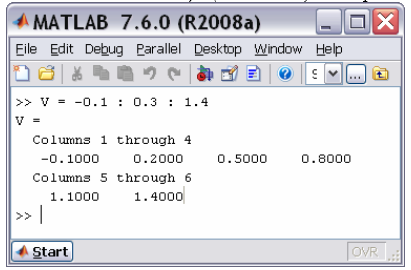


Рисунок 10. Введення вектора арифметичної прогресії

Якщо середній параметр (різниця прогресії) не зазначений, то він за замовчуванням приймається рівним одиниці. Наприклад, команда >> -2. 1 : 5 приводить до формування такого вектора ans = -2.1000 -1.1000 -0.1000 0.9000 1.9000 2.9000 3.9000 4. 9000

У такий спосіб уводяться вектори-рядки. Вектор-стовпець вводиться

аналогічно, але значення елементів відокремлюються знаком ";".

Введення значень елементів матриці здійснюється в Matlab у квадратних дужках, по рядках. При цьому елементи рядка матриці один від одного відокремлюються пропуском або комою, а рядки один від одного відокремлюються знаком ";" (рис. 11).

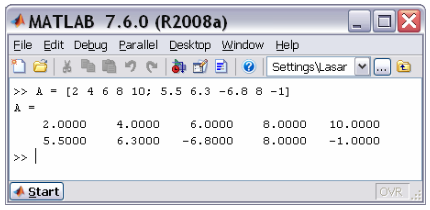


Рисунок 11. Приклад введення матриці

3. Ознайомитися з операціями над числами, матрицями.

**Найпростіші арифметичні дії**

В арифметичних виразах мови Matlab використовуються такі знаки арифметичних операцій:

+ – додавання;

– – віднімання;

\* – множення;

/ – ділення зліва праворуч;

\ – ділення справа ліворуч;

^ - –піднесення до степеня.

Використання Matlab у режимі калькулятора може відбуватися шляхом простого запису в командний рядок послідовності арифметичних дій з числами, тобто звичайного арифметичного виразу, наприклад:

(4.5 )^2\*7.23 – 3.14\*10.4.

Якщо після введення із клавіатури цієї послідовності натиснути клавішу <Enter>, у командному вікні виникне результат виконання у виді, поданому на рис. 12, тобто на екран під ім’ям системної змінної **ans** виводиться результат дії останнього виконаного оператора.

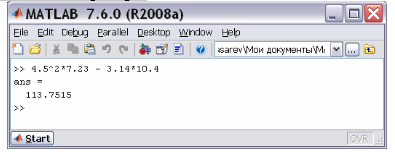


Рисунок 12.Введення арифметичного виразу і відображення результату

Взагалі виведення проміжної інформації у командне вікно підпорядковується таким правилам:

- якщо запис оператора не закінчується символом ';', результат дії цього оператора одразу ж виводиться в командне вікно;

- якщо оператор закінчується символом ';', результат його дії не відображується в командному вікні;

- якщо оператор не містить знака присвоювання (=), тобто є просто записом деякої послідовності дій над числами і змінними, значення результату присвоюється спеціальній системній змінній за ім'ям **ans**;

- отримане значення змінної **ans** можна використовувати в наступних операторах обчислень, використовуючи це ім’я **ans**; при цьому варто пам’ятати, що значення системної змінної **ans** змінюється після дії чергового оператора без знака присвоювання;

- у загальному випадку форма подання результату в командне вікно має вид:

<Ім'я змінної> = <результат>.

***Приклад.***

Нехай потрібно обчислити вираз (25+17)\*7. Це можна зробити таким чином. Спочатку набираємо послідовність 25+17 і натискаємо <Enter>. Одержуємо на екрані результат у вигляді **ans** = 42. Тепер записуємо послідовність **ans**\*7 і натискаємо <Enter>. Одержуємо **ans** = 294 (рис. 13 а). Щоб запобігти виведення проміжного результату дії 25+17, достатньо після запису цієї послідовності додати символ ' ; '. Тоді будемо мати результати у вигляді, поданому на рис.13 б.

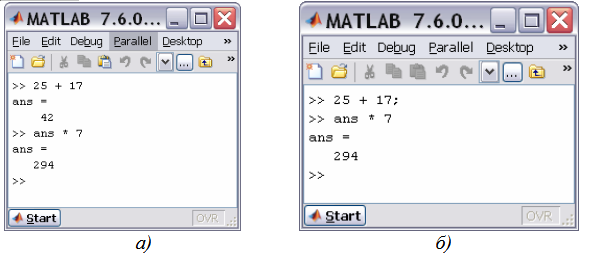


Рисунок 13. Використання системної змінної***ans***

Особливістю Matlab як калькулятора є можливість використання імен змінних для запису проміжних результатів у пам’ять ПК. Для цього застосовується операція присвоювання, що вводиться знаком рівності ' = ' у відповідності зі схемою :

<Ім’я змінної> = <вираз>[;]

Ім’я змінної може містити до 30 символів і повинно не збігатися з іменами функцій, процедур системи і системних змінних. При цьому система розрізнює великі й малі букви в змінних. Так, імена 'amenu' , 'Аmenu', 'aMenu' у Matlab позначають різні змінні.

Вираз праворуч від знака присвоювання може бути просто числом, арифметичним виразом, рядком символів (тоді ці символи потрібно укласти в апострофи) або символьним виразом. Якщо вираз не закінчується символом ' ; ', після натискання клавіші <Enter> у командному вікні виникне результат виконання у вигляді :

<Ім'я змінної> = <результат>.

Наприклад, якщо ввести в командне вікно рядок 'x = 25 + 17', то на екрані виникне запис (рис. 14) :

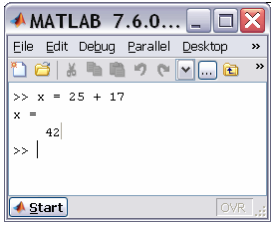


Рисунок 14. Присвоювання значення змінній

Система Matlab має кілька імен змінних, що використовуються самою системою і входять до складу зарезервованих:

***i, j*** – уявна одиниця (корінь квадратний з -1);

***pi*** – число π (зберігається у виді 3.141592653589793);

***inf*** – позначення машинної нескінченності;

***NaN*** – позначення невизначеного результату (наприклад, типу 0/0 або inf/inf);

***eps*** – похибка операцій над числами із плаваючою комою

***ans*** – результат останньої операції без знака присвоювання;

***realmax*** – максимальна величина числа, що може бути використана;

***realmin*** – мінімальна величина числа, що може бути використана.

Ці змінні можна використовувати в математичних виразах.

**Елементарні дії з комплексними числами**

Найпростіші дії з комплексними числами – додавання, віднімання, множення, ділення й піднесення до степеня – здійснюються за допомогою звичайних арифметичних знаків +, –, \*, /, \ і ^ відповідно.

Приклади використання наведені на рис. 15

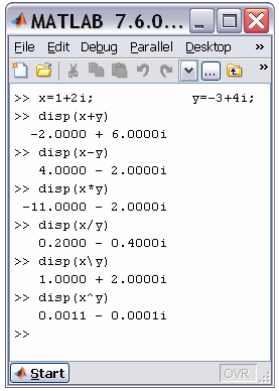


Рисунок 15.Приклади використання функції **disp**

Примітка. У наведеному фрагменті використана функція **disp** (від слова 'дисплей'), яка також дозволяє виводити в командне вікно результати обчислень або деякий текст. При цьому чисельний результат, як очевидно, виводиться вже без указування імені змінної або **ans**.

**Дії над векторами**

Розрізнюватимемо дві групи дій над векторами:

а) векторні дії – тобто такі, що передбачені векторним зчисленням у математиці;

б) дії по перетворенню елементів – це дії, що перетворюють елементи вектора, але не є операціями, дозволеними математикою з векторами.

**Векторні дії над векторами**

Додавання векторів. Як відомо, складатися (підсумовуватися) можуть тільки вектори однакового типу (тобто такі, які обидва є або векторами-рядками, або векторами-стовпцями), що мають однакову довжину (тобто однакову кількість елементів). Якщо X і Y є саме такими векторами, то їхню суму Z можна одержати, увівши команду Z = X + Y, наприклад:

» x = [1 2 3] ; y = [ 4 5 6];

» v = x + y

v = 5 7 9

Аналогічно за допомогою арифметичного знака "– " здійснюється віднімання векторів, що мають однакову структуру ( Z = X – Y ).

Наприклад:

» v = x – y

v = -3 -3 -3

Транспонування вектора здійснюється застосуванням знака апострофу, що записується одразу за записом імені вектора, який транспонується.

Наприклад:

» x'

ans =

1

2

3

Множення вектора на число здійснюється в Matlab за допомогою знака арифметичного множення ( \* ) у такий спосіб: Z = X\*r або Z = r\*X, де r –деяке дійсне число.

Приклад:

» v = 2\*x

v = 2 4 6

Множення двох векторів визначено у математиці тільки для векторів однакового розміру (довжини) і лише тоді, коли один із векторів-множників є рядком, а другий – стовпчиком. Тобто, якщо вектори X і Y є рядками, то математичний зміст мають лише дві форми множення цих векторів: U = X' \* Y і

V = X \* Y'. Причому в першому випадку результатом буде квадратна матриця, а

в другому – число.

У МatLAB множення векторів здійснюється застосуванням звичайного знака множення (\*), який записується між множниками-векторами.

Приклад:

» x = [1 2 3] ; y = [ 4 5 6];

» v = x' \* y

v =

4 5 6

8 10 12

12 15 18

» v = x \* y'

v = 32

Для трикомпонентних векторів у Matlab існує функція **cross**, яка дозволяє знайти векторний добуток двох векторів. Для цього, якщо задані два трикомпонентних вектори v1 i v2, достатньо ввести оператор **cross**(v1, v2).

Приклад:

» v1 = [1 2 3]; v2 = [4 5 6];

» cross(v1,v2)

ans = -3 6 -3

На цьому перелік припустимих математичних операцій з векторами вичерпується.

**Поелементне перетворення векторів**

У мові Matlab є ряд операцій, які перетворюють заданий вектор в інший того ж розміру й типу, але в той же час не є математичними операціями з вектором як математичним об'єктом. Усі ці операції перетворюють елементи вектора як елементи звичайного одновимірного масиву чисел. До таких операцій належать, наприклад, усі елементарні математичні функції, наведені раніше і які залежать від одного аргументу. У мові Matlab запис, наприклад, виду Y = sin(X), де X – деякий відомий вектор, приводить до формування нового вектора Y, що має той самий тип і розмір, але елементи якого дорівнюють синусам відповідних елементів вектора-аргументу X.

Наприклад:

» x = [ -2,-1,0,1,2];

» y = sin(x)

y = -0. 9093 -0. 8415 0 0. 8415 0. 9093

» z = tan(x)

z = 2. 1850 -1. 5574 0 1. 5574 -2. 1850

» v = exp(x)

v = 0. 3679 1. 0000 2. 7183 7. 389

Крім цих операцій у МаtLAB передбачено декілька операцій поелементного перетворення, що здійснюються за допомогою знаків звичайних арифметичних дій. Ці операції застосовуються до векторів однакового типу й розміру. Результатом їх є вектор того ж типу й розміру.

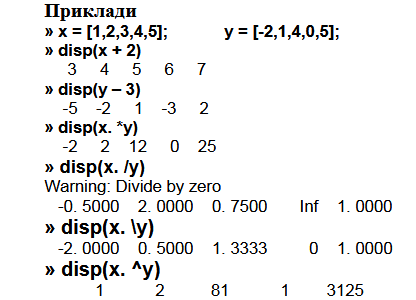
Додавання (віднімання) числа до (від) кожного елемента вектора. Здійснюється за допомогою знаку "+ " ("– ").

Поелементне множення векторів. Проводиться за допомогою сукупності знаків " .\* ", що записується між іменами векторів, які перемножуються. У результаті утворюється вектор, кожний елемент якого є добутком відповідних елементів векторів –"співмножників".

Поелементне ділення векторів. Здійснюється за допомогою сукупності знаків "./". Результат – вектор, кожний елемент якого є часткою від ділення відповідного елемента першого вектора на відповідний елемент другого вектора.

Поелементне ділення векторів в оберненому напрямку. Здійснюється за допомогою сукупності знаків ".\". В результаті отримують вектор, кожний елемент якого є часткою від ділення відповідного елемента другого вектора на відповідний елемент першого вектора.

Поелементне піднесення до степеня. Здійснюється за допомогою сукупності знаків ".^". Результат – вектор, кожний елемент якого є відповідним елементом першого вектора, піднесеним до степеня, розмір якого дорівнює значенню відповідного елемента другого вектора.



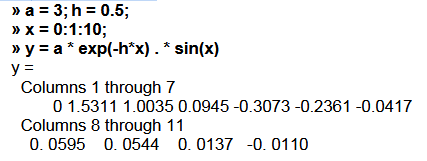
Вищевказані операції дозволяють дуже просто обчисляти (а потім – будувати графіки) складних математичних функцій, не використовуючи при цьому оператори циклу, тобто робити побудову графіків у режимі калькулятора.

Для цього достатньо задати значення аргументу як арифметичну прогресію так, як це було показано раніше, а потім записати потрібну функцію, використовуючи знаки поелементного перетворення векторів.

Наприклад, нехай потрібно обчислити значення функції:



при значеннях аргументу х від 0 до 10 із кроком 1. Обчислення масиву значень цієї функції у зазначених умовах можна здійснити за допомогою лише двох простих операторів :



**Операції з поліномами**

В системі Matlab передбачені деякі додаткові можливості математичного

оперування з поліномами.

Поліном (багаточлен) як функція визначається виразом :



В системі Matlab поліном задається й зберігається у виді вектора, елементами якого є коефіцієнти полінома від до :

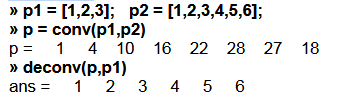


Введення полінома у Matlab здійснюється у такий самий спосіб, як і введення вектора довжиною n+1, де n – порядок полінома.

Множення поліномів. Добутком двох поліномів степенів n і m відповідно, як відомо, називають поліном степеня n+m, коефіцієнти якого визначають простим перемножуванням цих двох поліномів. Фактично операція множення двох поліномів зводиться до побудови розширеного вектора коефіцієнтів по заданих векторах коефіцієнтів поліномів-співмножників. Цю операцію в математиці називають згорткою векторів (а самий вектор, одержуваний у результаті такої процедури – вектором-згорткою двох векторів). У Matlab її здійснює функція conv(P1, P2).

Аналогічно, функція **deconv**(P1, P2) здійснює ділення полінома P1 на поліном P2, тобто обернену згортку векторів P1 і P2. Вона визначає коефіцієнти полінома, що є часткою від ділення P1 на P2.

Приклад :



В загальному випадку ділення двох поліномів призводить до одержання двох поліномів – полінома-результату (частки) і полінома-остачі. Щоб одержати обидва ці поліноми, треба оформити звернення до функції у такий спосіб:

[Q,R] = **deconv**(B,A) .

Тоді результат буде виданий у виді вектора Q з остачею у виді вектора R таким чином, що буде виконане співвідношення

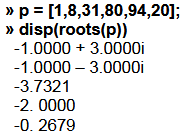
B = **conv**(A,Q) + R.

Система Matlab має функцію **roots**(P), що обчислює вектор, елементи якого є коренями заданого полінома Р.

Нехай потрібно знайти корені полінома:



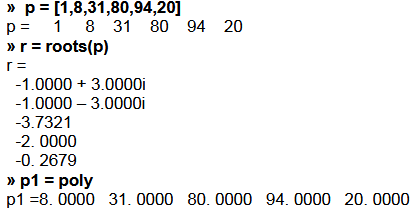
Нижче показано, як просто це зробити :



Обернена операція – побудова вектора р коефіцієнтів полінома за заданим вектором його коренів – здійснюється функцією **poly**:

p = **poly**

Тут r – заданий вектор значень коренів, p – обчислений вектор коефіцієнтів полінома. Наведемо приклад:



Зауважимо, що одержуваний вектор не показує старшого коефіцієнта, який за замовчуванням покладається рівним одиниці.

Ця ж функція у випадку, коли аргументом її є деяка квадратна матриця А розміром (n\*n), будує вектор характеристичного полінома цієї матриці. Звернення

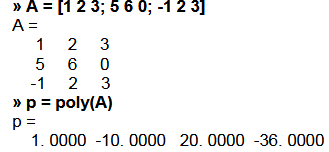
**p =** **poly**(A)

формує вектор p коефіцієнтів полінома



де Е – позначення одиничної матриці розміром (n\*n).

Розглянемо приклад:



Для обчислення значення полінома за заданим значенням його аргументу в Matlab передбачена функція polyval. Звернення до неї здійснюється за схемою:

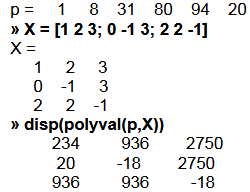
**y = polyval(p,x)**,

де p – заданий вектор коефіцієнтів полінома, а x – задане значення аргументу.

Приклад:



Якщо як аргумент поліному зазначена матриця Х, то функція **polyval(p,X)** обчислює матрицю Y, кожний елемент якої є значенням зазначеного полінома при значенні аргументу, рівному відповідному елементу матриці Х, наприклад:

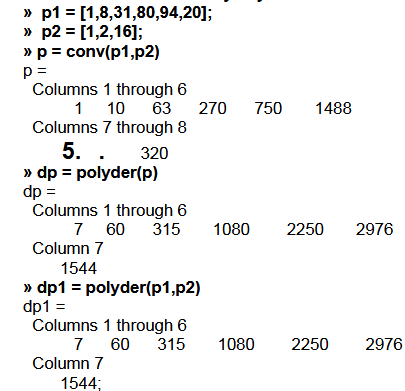


У цьому випадку функція обчислює значення полінома для кожного елемента матриці Х, і тому розміри вхідної й вихідної матриць однакові **size(Y) = size(X).**

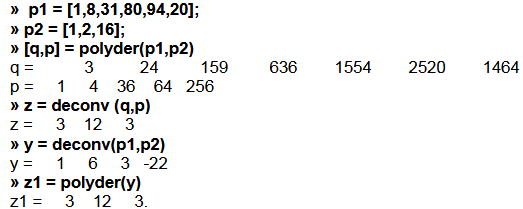
Обчислення похідної від полінома здійснюється функцією **polyder.** Ця функція створює вектор коефіцієнтів полінома, який є похідною від заданого полінома. Вона має три види звернень: **dp = polyder(p)** за заданим поліномом р обчислює вектор **dp**, елементи якого є коефіцієнтами полінома-похідної від заданого:



**dp = polyder(p1,p2)** обчислює вектор **dp**, елементи якого є коефіцієнтами полінома-похідної від добутку двох поліномів **р1** і **р2**:



**[q,p] = polyder(p1,p2)** обчислює похідну від відношення **(p1/p2)** двох поліномів **р1** і **р2** і видає результат у виді відношення **(q/p)** поліномів **q** і **p**:



4. Ознайомитися з правилами виклику функцій MatLAB.

**Елементарні математичні функції**

Загальна форма використання функції у Matlab така:

<ім’я результату> = <ім’я функції>(<перелік аргументів або їх значень>).

У мові Matlab передбачені наступні елементарні арифметичні функції.

*Тригонометричні й гіперболічні функції*

**sin**(Z) – синус числа Z;

**sinh**(Z) – гіперболічний синус;

**asin**(Z) – арксинус (у радіанах, у діапазоні від - π /2 до + π /2);

**asinh**(Z) – обернений гіперболічний синус;

**cos**(Z) – косинус;

**cosh**(Z) – гіперболічний косинус;

**acos**(Z) – арккосинус (у діапазоні від 0 до π );

**acosh**(Z) – обернений гіперболічний косинус;

**tan**(Z) – тангенс;

**tanh**(Z) – гіперболічний тангенс;

atan(Z) – арктангенс (у діапазоні від - π /2 до + π /2);

**atan2**(X,Y) – чотириквадрантний арктангенс (кут у діапазоні від – π до + π між горизонтальним правим променем і променем, що проходить через точку з координатами X і Y);

**atanh**(Z) – обернений гіперболічний тангенс;

**sec**(Z) – секанс;

**sech**(Z) – гіперболічний секанс;

**asec**(Z) – арксеканс;

**asech**(Z) – обернений гіперболічний секанс;

**csc**(Z) – косеканс;

**csch**(Z) – гіперболічний косеканс;

**acsc**(Z) – арккосеканс;

**acsch**(Z) – обернений гіперболічний косеканс;

**co**t(Z) – котангенс;

**coth**(Z) – гіперболічний котангенс;

**acot**(Z) – арккотангенс;

**acoth**(Z) – обернений гіперболічний котангенс.

*Експоненціальні функції*

**exp**(Z) – експонента числа Z;

**log**(Z) – натуральний логарифм;

**log10**(Z) – десятковий логарифм;

**sqrt**(Z) – квадратний корінь із числа Z;

**abs**(Z) – модуль числа Z.

*Цілочислові функції*

**fix**(Z) – округлення до найближчого цілого убік нуля;

**floor**(Z) – округлення до найближчого цілого убік від’ємної нескінченності;

**ceil**(Z) – округлення до найближчого цілого убік додатної нескінченності;

**round**(Z) – звичайне округлення числа Z до найближчого цілого;

**mod**(X,Y) – цілочислове ділення ділення X на Y;

**rem**(X,Y) – обчислення остачі від ділення X на Y;

**sign**(Z) – обчислення сигнум-функції числа Z (0 при Z=0, -1 при Z<0, 1 при Z>0).

**Спеціальні математичні функції**

Крім елементарних у мові Matlab передбачено цілу низку спеціальних математичних функцій. Нижче наведено перелік і стислий зміст цих функцій. Правила звернення до них і використання користувач може відшукати в описах цих функцій, що виводяться на екран, якщо набрати команду help і вказати в тому ж рядку ім’я функції.

*Функції перетворення координат*

**cart2sph** – перетворення декартових координат у сферичні;

**cart2pol** – перетворення декартових координат у полярні;

**pol2cart** – перетворення полярних координат у декартові;

**sph2cart** – перетворення сферичних координат у декартові.

*Функції Бесселя*

**besselj** – функція Бесселя першого роду;

**bessely** – функція Бесселя другого роду;

**besseli** – модифікована функція Бесселя першого роду;

**besselk** – модифікована функція Бесселя другого роду.

*Бета-функції*

**beta** – бета-функція;

**betainc** – неповна бета-функція;

**betaln** – логарифм бета-функції.

*Гамма-функції*

**gamma** – гамма-функція;

**gammainc** – неповна гамма-функція;

**gammaln** – логарифм гамма-функції.

*Еліптичні функції й інтеграли*

**ellipj** – еліптичні функції Якобі;

**ellipke** – повний еліптичний інтеграл;

**expint** – функція експоненціального інтегралу.

*Функції похибок*

**erf** – функція похибок;

**erfc** – додаткова функція похибок;

**erfcx** – масштабована додаткова функція похибок;

**erfinv** – обернена функція похибок.

*Інші функції*

**gcd** – найбільший загальний дільник;

**lcm** – найменше загальне кратне;

**legendre** – узагальнена функція Лежандра;

**log2** – логарифм за основою 2;

**pow2** – піднесення 2 до зазначеного степеня;

**rat** – подання числа у вигляді раціонального дробу;

**rats** – подання чисел у вигляді раціонального дробу.

**Функції комплексного аргументу**

Практично всі елементарні математичні функції, наведені раніше, обчислюються за комплексних значень аргументу й одержують у результаті цього

комплексні значення результату.

Завдяки цьому, наприклад, функція sqrt обчислює, на відміну від інших мов програмування, квадратний корінь із від’ємного аргументу, а функція **abs** при комплексному значенні аргументу обчислює модуль комплексного числа.

Приклади наведені на рис. 16.

У Matlab є кілька додаткових функцій, розрахованих тільки на комплексний аргумент:

**real**(Z) – виділяє дійсну частину комплексного аргументу Z;

**imag**(Z) – виділяє уявну частину комплексного аргументу;

**angle**(Z) – обчислює значення аргументу комплексного числа Z (у радіанах від - π до + π );

**conj**(Z) – видає число, комплексно спряжене щодо Z.

Приклади наведені на рис. 17.

Крім того, у Matlab є спеціальна функція **cplxpair**(V), що здійснює сортування заданого вектора V із комплексними елементами у такий спосіб, що комплексно-спряжені пари цих елементів розташовуються у вихідному векторі в порядку зростання їхніх дійсних частин, при цьому елемент із від'ємною уявною частиною завжди розташовується першим. Дійсні елементи завершують комплексно-спряжені пари.

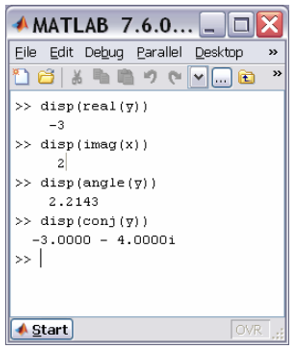
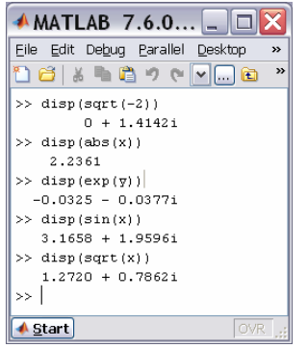
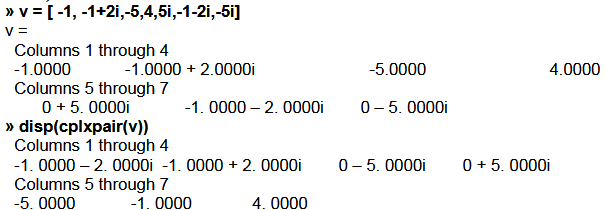


Рисунок 16.Застосування функцій Рисунок 17. Застосування функцій

з комплексним аргументом комплексного аргументу

Наприклад (надалі в прикладах команди, що набираються із клавіатури, будуть написані масним шрифтом, а результат їхнього виконання – звичайним шрифтом):



Пристосованість більшості функцій Matlab до оперування з комплексними числами дозволяє значно простіше будувати обчислення з дійсними числами, результат яких є комплексним, наприклад, знаходити комплексні корені квадратних рівнянь.

5. Ознайомитися з роботою команди HELP.

6. Ознайомитися з роботою функції plot для побудови графіків у MatLAB. Побудувати графік заданої функції, позначити вісі та заголовок графіку (функції xlabel, ylabel, title), нанести координатну сітку (функція grid).

**Виведення графіків. Процедура plot**

Виведення графіків у системі Matlab є настільки простою і зручною процедурою, що нею можна користуватися навіть при обчисленнях у режимі калькулятора. Основною функцією, яка забезпечує побудову графіків на екрані дисплея,

є функція ***plot***. Загальна форма звернення до цієї процедури така: ***plot***(x1,y1,s1,x2,y2,s2,...).

Тут x1, y1 – задані вектори, елементами яких є масиви значень аргументу (х1) і функції (у1), що відповідають першої кривої графіка; x2, y2 – масиви значень аргументу й функції другої кривої і т.д. При цьому передбачається, що значення аргументу відкладаються уздовж горизонтальної осі графіка, а значення функції – уздовж вертикальної осі. Змінні s1, s2,... є символьними (їхня вказівка не є бов'язковою). Кожна з них може містити до трьох спеціальних символів, що визначають відповідно: а) тип лінії, що з'єднує окремі точки графіка; б) тип точки графіка; в) колір лінії. Якщо змінні s не зазначені, то тип лінії за умовчанням - відрізок прямої, тип точки – пиксел, а колір установлюється (у версії 5) за такою черговістю: - *синій, зелений, червоний, блакитний, фіолетовий, жовтий, чорний і білий* - у залежності від того, яка по черзі лінія виводиться на графік. Наприклад, звернення виду ***plot***(x1,y1,x2,y2,...) призведе до побудови графіка, у якому перша крива буде лінією з відрізків прямих синього кольору, друга крива – такого ж типу зеленою лінією і т.д.

*Графіки* в Matlab завжди *виводяться в окреме (графічне) вікно*, яку називають *фігурою*.

Наведемо приклад. Нехай потрібно вивести графік функції

y = 3sin(x + π /3) на проміжку від -3 π до +3 π із кроком π/100.

Спочатку треба сформувати масив значень аргументу х:

x = -3\*pi : pi/100 : 3\*pi, потім обчислити масив відповідних значень функції:

y = 3\*sin(x+pi/3) і, нарешті, побудувати графік залежності у(х).

У цілому в командному вікні ця послідовність операцій буде виглядати

так:

**» x = -3\*pi:pi/100:3\*pi;**

**» y = 3\*sin(x+pi/3);**

**» plot(x,y)**

В результаті на екрані з'явиться додаткове вікно із графіком (див. рис.

18 а).

*Якщо вектор аргументу при зверненні до функції* ***plot*** *не зазначено явно,то система обирає за умовчанням як аргумент номер елемента вектора функції.*

Наприклад, якщо ввести команду

**» plot(y),** то результатом буде поява графіка у виді, наведеному на рис. 18 б.

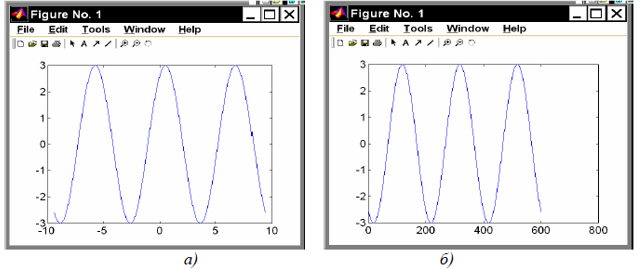


Рисунок 18. Виведення графіка синусоїди

Графіки, наведені на рис. 18, мають декілька недоліків:

- на них не нанесено сітку з координатних ліній, що утруднює "зчитування" графіків;

- немає загальної інформації про криві графіка (заголовка);

- невідомо, які величини відкладені по осях графіка.

Перший недолік усувається за допомогою функції ***grid***. Якщо цю функцію записати одразу після звернення до функції ***plot***:

**» x = -3\*pi:pi/100:3\*pi;**

**» y = 3\*sin(x+pi/3);**

**» plot(x,y), grid,**

то графік буде споряджений координатною сіткою (рис. 19).

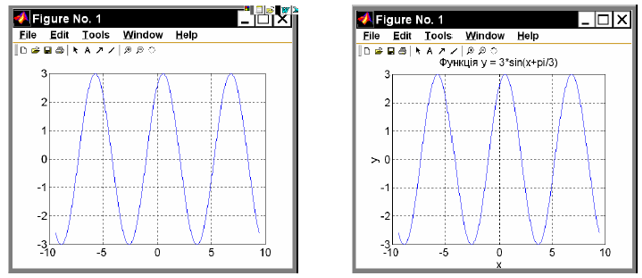


Рисунок 19. Застосування***grid*** Рисунок 20. Застосування ***title,***

***xlabel, Ylabel***

Цінною особливістю графіків, побудованих у системі Matlab, є те, що *сітка координат завжди відповідає "цілим" крокам змінювання*, що робить графіки "читабельними", тобто за графіком можна робити "відлік" значення функції при будь-якому заданому значенні аргументу і навпаки. Такої властивості не має жодний із графічних пакетів-додатків до мов програмування високого рівня.

Заголовок графіка виводиться за допомогою процедури ***title.*** Якщо після звернення до процедури ***plot*** викликати ***title*** у такий спосіб:

***title***('<текст>'), то понад графіком з'явиться текст, записаний між апострофами у дужках. При цьому варто пам'ятати, що *текст завжди має міститися в апострофах.*

Аналогічно можна вивести пояснення до графіка, що розміщаються уздовж горизонтальної осі (функція ***xlabel***) і уздовж вертикальної осі (функція ***ylabel***).

Наприклад, сукупність операторів

**» x = -3\*pi : pi/100 : 3\*pi;**

**» y = 3\*sin(x+pi/3);**

**» plot(x,y), grid**

**» title('Функція y = 3\*sin(x+pi/3)');**

**» xlabel('x'); ylabel('y');**

призведе до оформлення поля фігури у виді, поданому на рис. 6.3. Очевидно, така форма вже цілком задовольняє вимоги, що висуваються до інженерних графіків.

Не більш складним є виведення у середовищі Matlab графіків функцій, заданих *параметрично*.

Нехай, наприклад, необхідно побудувати графік функції у(х), що задана параметричними формулами:

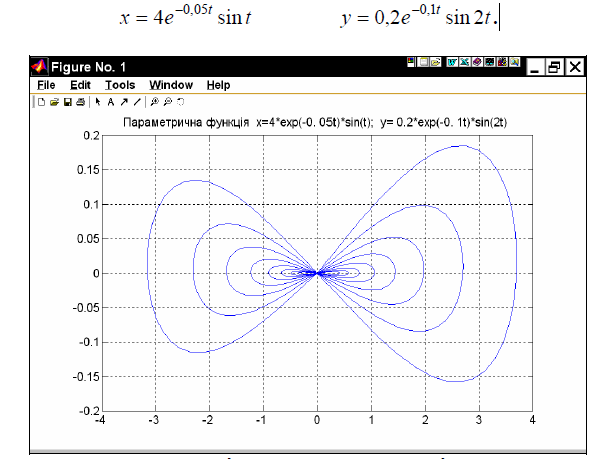


Рисунок 21. Графік параметрично заданої функції

Виберемо діапазон змінювання параметра *t* від 0 до 50 із кроком 0.1. Тоді, набираючи сукупність операторів

**» t = 0:0.1:50;**

**» x = 4\*exp(-0.05\*t).\*sin(t);**

**» y = 0.2\*exp(-0. 1\*t). \*sin(2\*t);**

**» plot(x,y)**

**» title**('Параметрична функція x=4\*exp(-0. 05t)\*sin(t); y= 0.2\*exp(-0. 1t)\*sin(2t) ')

**» grid,** одержимо графік рис. 21.

**Спеціальні графіки**

Великою зручністю, наданою системою Matlab, є зазначена раніше можливість не вказувати аргумент функції при побудові її графіка. У цьому випадку як аргумент система приймає номер елемента вектора, графік якого будується.

Користуючись цим, наприклад, можна побудувати "графік вектора":

**» x = [ 1 3 2 9 6 8 4 6];**

**» plot (x)**

**» grid**

**» title('Графік вектора Х')**

**» ylabel('Значення елементів')**

**» xlabel(' Номер елемента').**

Результат поданий на рис. 22.

Ще більш наочним є подання вектора у виді *стовпцевої діаграми* за допомогою функції ***bar*** (див. рис. 23):

**» bar(x)**

**» title(**'Графік вектора Х'**)**

**» xlabel(**' Номер елемента'**)**

**» ylabel**('Значення елементів'**)**

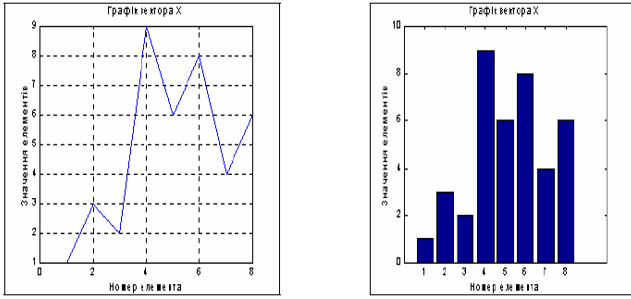


Рисунок 22. Застосування ***plot*** Рисунок 23. Застосування ***bar***

Якщо функція задана своїми значеннями при дискретних значеннях аргументу, і невідомо, як вона може змінюватися в проміжках між значеннями аргументу, зручніше подавати графік такої функції у виді окремих вертикальних ліній для кожного із заданих значень аргументу. Це можна зробити, застосовуючи процедуру ***stem***, звернення до якої цілком аналогічно зверненню до процедури ***plot***:

**x = [ 1 3 2 9 6 8 4 6];**

**stem(x,'k')**

**grid**

**set(gca,'FontName','Arial','FontSize',14),**

**title('Графік вектори Х')**

**ylabel('Значення елементів')**

**xlabel(' Номер елемента')**

На рис. 24 зображено одержаний при цьому графік.

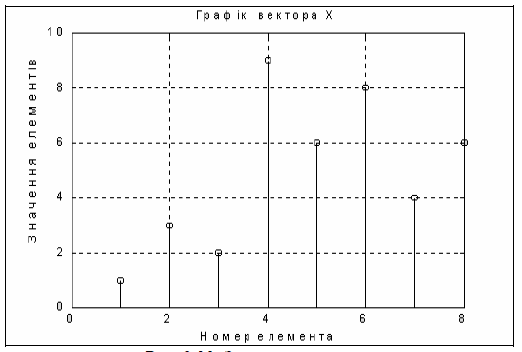


Рисунок 24. Застосування ***stem***

Інший приклад – побудова графіка функції у виді стовпцевої діаграми (рис. 6.8):

**» x = - 2.9 : 0.2 : 2.9;**

**» bar(x, exp(-x . \* x))**

**» title(**'Стовпцева діаграма функції y = exp(-x^2)'**)**

**» xlabel (**' Аргумент х'**)**

**» ylabel** (' Значення функції у'**)**

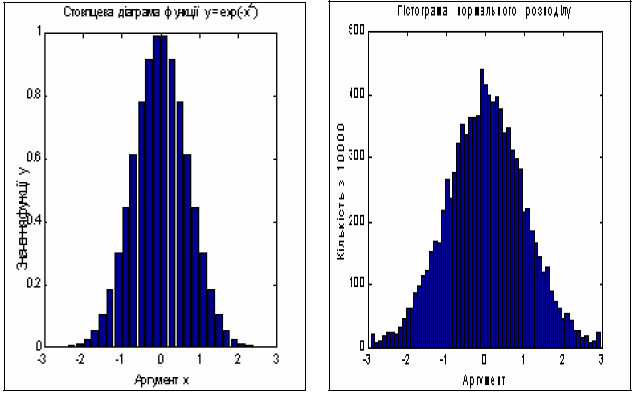


Рисунок 25. Процедура ***bar*** Рисунок 26. Процедура ***hist***

Ще одна корисна інженеру функція – ***hist*** (побудова графіка *гістограми* заданого вектора). Стандартне звернення до неї має вид:

***hist***(y,x), де у – вектор, гістограму якого потрібно побудувати; х – вектор, що визначає інтервали змінювання першого вектора, усередині яких підраховується кількість елементів вектора “у”. Ця функція робить дві операції

- підраховує кількість елементів вектора “у”, значення яких потрапляють усередину відповідного діапазону, зазначеного вектором “х”;

- будує стовпцеву діаграму підрахованих чисел елементів вектора “у” як функцію діапазонів, зазначених вектором “х” .

Як приклад роздивимося побудову гістограми випадкових величин, що формуються вмонтованою функцією ***randn***. Візьмемо загальну кількість елементів вектора цих випадкових величин 10 000. Побудуємо гістограму для діапазону змінювання цих величин від -2,9 до +2,9. Інтервали змінювання нехай будуть рівні 0,1. Тоді графік гістограми можна побудувати за допомогою сукупності таких операторів:

**» x = -2.9:0.1:2.9;**

**» y = randn(10000,1);**

**» hist(y,x)**

**» ylabel(**'Кількість з 10000'**)**

**» xlabel(** 'Аргумент'**)**

**» title**('Гістограма нормального розподілу'**)**

Результат поданий на рис. 26. З нього, зокрема, випливає, що вмонтована функція ***randn*** достатньо вірно відображає нормальний гауссовий закон розподілу випадкової величини.

**Вставлення графіків до документу**

Щоб вставити графік із графічного вікна (фігури) до документу Word, слід скористатися командами меню, розташованого у верхній частині вікна фігури. У меню ***Edit*** оберіть команду ***Copy Figure***. Перейдіть у вікно документу і натисніть клавіші <Ctrl>+<C>. Вміст графічного вікна виникне у тому місці документу, де містився курсор.

7. Ознайомитися із роботою функцій генерації випадкових чисел (rand, randn, randi) із заданими густинами розподілу імовірності.

Функцsя randn генерує масив з випадковими елементами, розподіленими за нормальним законом з нульовим математичним очікуванням і середньоквадратичним відхиленням, що дорівнює 1:

* randn(n) — повертає матрицю розміром n х n. Якщо n — не скаляр, то з’явиться повідомлення про помилку;
* randn(m.n) або randn([m n]) — повертає матрицю розміром m x n;
* randn(m,n,p,...) або randn([m n р...]) — повертає масив з елементами, значення яких розподілені за нормальним законом;
* randn(size(A)) — повертає масив того ж розміру, що і А, з елементами, розподіленими за нормальним законом;
* randn (без аргументів) — повертає одне випадкове число, яке змінюється при кажному наступному виклику і має нормальний розподіл;
* randn( 'state') — повертає двоелементний вектор, що вміщує поточний стан нормального генератора. Для зміни стану генератора можна застосувати наступні форми цієї функції:
  + randn('state',s) — встановлює стан в *s;*
  + randn('state' ,0) — скидає генератор в початковий стан;
  + randn('state', j) — для цілих j встановлює генератор в *J-ий* стан;
  + randn('state', sum( 100\*clock)) — кожного разу скидає генератор в стан, який залежить від часу.

Функція rand генерує масиви випадкових чисел, значення елементів яких *рівномірно* розподілені на проміжку (0, 1):

* rand(n) — повертає матрицю розміром n х n. Якщо n — не скаляр, то з’явиться повідомлення про помилку;
* rand(m.n) або rand([m п]) — повертає матрицю розміром m x n;
* rand(m.n,p....) або rand([m n р...]) — повертає багатовимірний масив;
* rand(size(A)) — повертає масив того ж розміру, що і А, з елементами, розподіленими за рівномірним законом;
* rand (без аргументів) — повертає одне випадкове число, яке змінюється при кожному наступному виклику і має рівномірний закон розподілу;
* rand(' state') — повертає вектор з 35 елементами, що вміщує поточний стан генератора випадкових чисел з рівномерним розподілом. Для зміни стану генератора можна застосувати наступні форми цієї функції:
* rand('state' .s) — встановлює стан в *s;*
* rand( 'state' ,0) — скидає генератор в початковий стан;
* rand( 'state'. j) — для цілих j встановлює генератор в *J-ий* стан;
* rand( 'state' ,sum(100\*clock)) — кожного разу скидає генератор в стан, який залежить від часу.

8. Ознайомитися з роботою функції hold, побудувати два графіки в одному вікні.

Для того, щоб декілька послідовно обчислюваних графіків були зображені в одному графічному вікні в одному стилі і одному графічному полі, можна використати команду ***hold on***, тоді кожний такий графік буде будуватися в тому ж попередньо відкритому графічному вікні, тобто кожна нова лінія буде додаватися до раніше побудованих. Команда ***hold off*** виключає режим зберігання графічного вікна, встановленого попередньою командою.

9. Ознайомитися з роботою функцій save та load, зберегти дані розрахунку в файл. Прочитати їх із файлу в іншому сценарії, побудувати графік.

**Вимоги до звіту**

Звіт до лабораторних робіт повинен містити:

* 1. Скріни екранів вводу вхідних операндів і операцій над ними.
  2. Скріни екранів побудови графіків функцій, згідно варіантів завдань.

**Контрольні запитання**

1. В чому полягають переваги системи?

2. Назвіть два режими роботи в системі MatLAB.

3. Наведіть приклади пакетів програм, що входять до системи MatLAB та проаналізуйте їх вміст.

4. Опишіть можливості функції plot для візуалізації результатів моделювання.

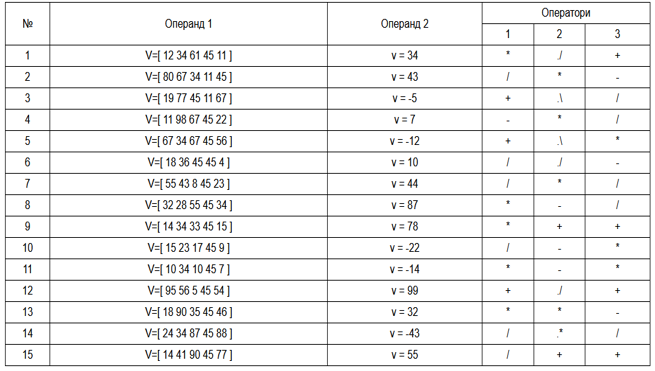
5. Яка функція використовується для розбиття графічного вікна на декілька частин?

**Варіанти завдань до лабораторної роботи 3:**

Виконати в режимі калькулятору наступні дії:

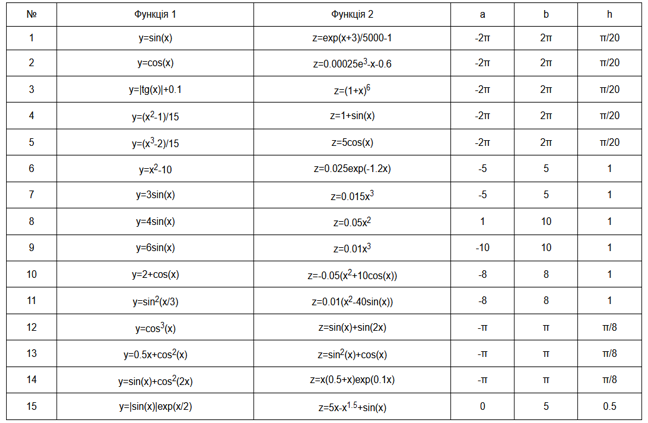
1. ввід вихідних операндів;
2. виконати над операндами 1 та 2 операцію 1;
3. виконати над результатом та операндом 1 операцію 2;
4. виконати над результатом та операндом 2 операцію 3;
5. піднести поелементно операнд 1 до степеня 3.

Таблиця 1.



*Побудова графіків .* **(Варіанти завдань див. табл. 2)**

Таблиця 2.



**Варіанти завдань до лабораторної роботи 4:**

*Поверхневий і контурний графіки.* **(Варіанти завдань див. табл. 3)**

Для формування поверхневого або контурного графіка необхідно:

- задати число точок по координатах X та Y,

- створити вкладені цикли по X та Y, обчислити функцію Z=f(X,Y),

- ввести номер графічного вікна, вивести туди графік обраного типу.

Необхідно використовувати графіки:

- тривимірний з аксонометрією, функція plot3(X,Y,Z),

- тривимірний з функціональним забарвленням, функція mesh(X,Y,Z),

- тривимірний з функціональним забарвленням і проекцією, функція meshc(X,Y,Z),

- тривимірний з функціональним забарвленням і проекцією, функція surf(X,Y,Z),

- контурний, функція contour(X,Y,Z),

- об'ємний контурний, функція contour3(X,Y,Z),

- тривимірний з висвітленням, функція surfl(X,Y,Z).

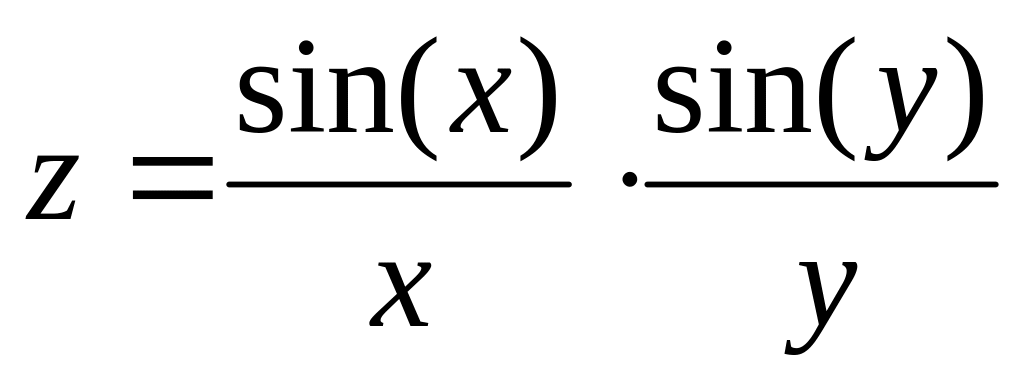
У кожному вікні можна малювати декілька графіків з накладенням один на одного. У списку параметрів для кожного графіка параметри перераховуються групами послідовно (у роботі графік для вікна один). У кожну групу входять:

- X - перша координата площадки основи,

- Y - друга координата площадки основи,

- Z - значення функції.

**Приклад виконання**

Функція 

Межі зміни аргументів -2π...2π

Варіант 1

*% Число точок і крок*

N=40;

h=pi/20;

*% Розрахунок матриці*

for n=1:2\*N+1

if n==N+1 A(n)=1; else A(n)=sin(h\*(n-N-1))/(h\*(n-N-1)); end;

end;

for n=1:2\*N+1

for m=1:2\*N+1 Z(n,m)=A(n)\*A(m);

end; end;

*% Завдання площадки*

[X,Y]=meshgrid([-N:1:N]);

*% Вивід графіка в аксонометрії у вікно 1*

figure(1); plot3(X,Y,Z);

*% вивід тривимірного графіка з функціональним забарвленням у вікно 2*

figure(2); mesh(X,Y,Z);

*% вивід тривимірного графіка з функціональним забарвленням і проекцією у вікно 3*

figure(3); meshc(X,Y,Z);

*% вивід тривимірного графіка із проекцією у вікно 4*

figure(4); surf(X,Y,Z);

*% Вивід контурного графіка у вікно*

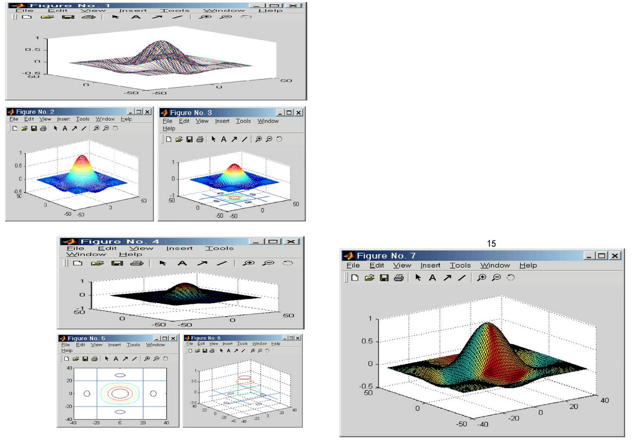
5 figure(5); contour(X,Y,Z)

*% Вивід об'ємного контурного графіка у вікно 6*

figure(6); contour3(X,Y,Z)

*% Вивід об'ємного графіка з висвітленням у вікно 7*

figure(7);surfl(X,Y,Z)



Варіант 2

*% Число точок і крок*

N=40;

h=pi/20;

*% Розрахунок матриці*

for n=1:2\*N+1

if n==N+1 A(n)=1; else A(n)=sin(h\*(n-N-1))/(h\*(n-N-1)); end;

end;

for n=1:2\*N+1

for m=1:2\*N+1 Z(n,m)=A(n)\*A(m);

end; end;

*% Завдання площадки*

[X,Y]=meshgrid([-N:1:N]);

*% Вивід графіка в аксонометрії в підвікно 1*

subplot(3,3,1),plot3(X,Y,Z);

*% вивід тривимірного графіка з функціональним забарвленням у підвікно 2*

subplot(3,3,2),mesh(X,Y,Z);

*% вивід тривимірного графіка з функціональним фарбуванням і проекцією в підвікно 3*

subplot(3,3,3),meshc(X,Y,Z);

*% вивід тривимірного графіка із проекцією в підвікно 4*

subplot(3,3,4),surf(X,Y,Z);

*% Вивід контурного графіка в підвікно 5*

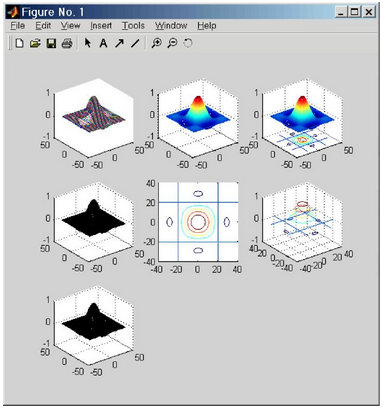
subplot(3,3,5),contour(X,Y,Z)

% *Вивід об'ємного контурного графіка в підвікно 6*

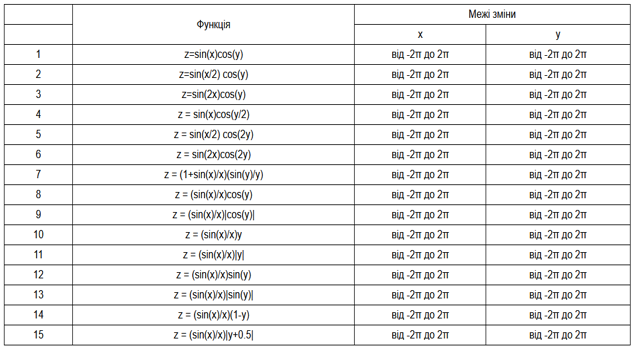
subplot(3,3,6),contour3(X,Y,Z)

*% Вивід об'ємного графіка з висвітленням у підвікно 7*

subplot(3,3,7),surfl(X,Y,Z)



Таблиця 3.



**Лабораторна робота №5.**

**РОЗРОБКА СИСТЕМИ НЕЧІТКОГО ВИВОДУ В ІНТЕРАКТИВНОМУ РЕЖИМІ.**

**Основні теоретичні відомості**

Розглянемо розробку системи нечіткого виводу на прикладі «Чайові в ресторані».

Як приклад розробки системи нечіткого виведення в інтерактивному режимі за допомогою графічних засобів пакету Fuzzy Logic Toolbox розглянемо модель, що входить до демонстраційних прикладів системи MatLab.

Приклад «*Чайові в ресторані*». Розглянемо ситуацію в ресторані, при якій, згідно прийнятим в США традиціям, після закінчення обслуговування відвідувача прийнято залишати офіціантові чайові. Грунтуючись на сталих в цій країні звичаях і інтуітивних представленнях відвідувачів ресторанів величина суми чайових не є постійною, а залежить, наприклад, від якості обслуговування і якості приготування замовлених блюд.

Завдання полягає в тому, щоб розробити деяку експертну систему, яка б була реалізована у вигляді системи нечіткого виводу і дозволяла б визначити величину чайових на основі суб’єктивних оцінок відвідувачів якості обслуговування і якості приготування замовлених блюд.

Емпіричні знання про дану предметну область можуть бути представлені в формі наступних евристичних правил продукції:

* 1. Якщо обслуговування погане або вечеря підгоріла, то чайові – малі.
  2. Якщо обслуговування добре, то чайові – середні.
  3. Якщо обслуговування відмінне або вечеря чудова, чайові – щедрі.

**Примітка**. Приведені вище правила дійсно суб’єктивні і не звільнені від критики. Зокрема, для багатьох відвідувачів наших ресторанів може показатися дивним правило 1, згідно якому слід залишати чайові у разі поганого обслуговування або вечері, що підгоріла, і правило 2, згідно якому слід залишати середні чайові навіть у разі вечері, що підгоріла. Можливо, деякі визнають можливим взагалі відмовитись від залишення чайових в подібних ситуаціях і будуть по своєму мати рацію. Проте, оскільки даний приклад широко використовується в літературі для демонстрації можливостей системи MatLab він приводиться тут без зміни в своєму оригінальному вигляді.

Як вхідні параметри системи нечіткого виводу розглядатимемо 2 нечіткі лінгвістичні змінні «*якість обслуговування*» і «*якість приготування замовлених страв*» ( або скорочено – «*якість вечері*»). Як вихідні параметри – нечітку лінгвістичну змінну «*величина чайових*».

В якості терм-множини першою лінгвістичною змінною «*якість обслуговування*» використовуватимемо множину ={«*погане*», «*добре*», «*відмінне*»}, а в якості терм-множини другої вхідної лінгвістичної змінної «якість вечері» використовуватимемо множину = {«*підгорілий*», «*чудовий*»}. В якості терм-множини вихідної лінгвістичної змінної «*величина чайових*» використовуватимемо множину = {«*малі*», «*середні*», «*щедрі*»}. При цьому кожен з термів першої і другої вхідної змінної (якість обслуговування і приготування замовлених страв) оцінюватимемо за 10-бальною порядковою шкалою, при які1 цифрі 0 відповідає найгірша оцінка, а цифрі 10 – найкраща оцінка. Що стосується термів вихідної змінної, то припускатимемо, що малі чайові складатимуть 5% від вартості замовлених страв, середні чайові – близько 15%, а щедрі чайові – близько 25 %.

З урахуванням зроблених уточнень розглянута суб’єктивна інформація про величину чайових може бути представлена у формі 3-х правил нечітких продукцій наступного вигляду (Система нечіткого виводу типу Мамдані):

Правило 1: ЯКЩО «*якість обслуговування*» погана АБО «*вечеря підгоріла*»

ТО «*величина чайових мала*».

Правило 2: ЯКЩО «*якість обслуговування хороша*»,

ТО «*величина чайових середня*».

Правило 3: ЯКЩО «*якість обслуговування відмінна*» АБО «*вечеря смачна*»,

ТО «*величина чайових щедра*»

Процес розробки системи нечіткого виводу в інтерактивному режимі для розглянутого вище прикладу «Чайові в ресторані» полягає у наступній послідовності дій:

* + 1. Викликати редактор системи нечіткого виведення **FIS*,*** для чого у вікні команд набрати ім’я відповідної функції **fuzzy.** Після виконання цієї команди на екрані з’явиться графічний інтерфейс редактора **FIS** з ім’ямсистеми нечіткого виведення Untitled і типом системи нечіткого виведення (Мамдані) запропонованими за умовчанням:
    2. Оскільки в прикладі розглядається система нечіткого виведення з двома входами, необхідно додати в систему **FIS**, що розробляється, ще одну вхідну змінну. Для цього слід виконувти команду меню **Edit**>**Add**>**Variable**>**Input.** В результаті виконання цієї команди на діаграмі системи нечіткого виведення з’явиться новий жовтий прямокутник з ім’ям другої вхідної змінної: input2 (рис. 27)

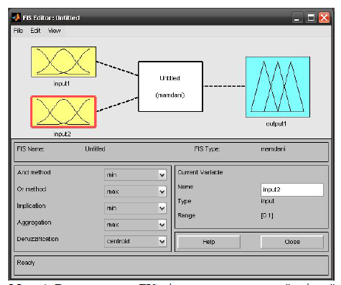


Рисунок 27. Вид редактора **FIS** після додавання другої змінної

* + 1. Змінемо імена вхідних і вихідних змінних, запропонованих системою **MatLab** за умовчанням. Для цього необхідно виділити прямокутник з ім’ям відповідної змінної, виконавши клацання на його зображенні на діаграмі (сторони виділеного прямокутника мають червоний колір). Після чого слід набрати нове ім’я змінної в полі введення ***Name*** в правій частині редактора **FIS**. Результат зміни імен змінних системи нечіткого виведення зображений на рисунку 28.

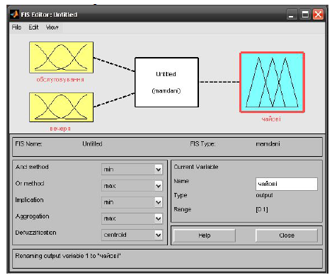


Рисунок 28. Вид редактора **FIS** після зміни імен змінних, запропонованих системою **MatLab** за умовчанням

**Примітка**. Щоб уникнути проблем з коректним відображенням символів кирилиці, слід давати такі імена змінним, якіскладаються з одного слова без додаткових службових символів.

* + 1. Змінемо ім’я системи нечіткого виведення Untitled, запропоноване за умовчанням. Для цього збережемо створювану структуру **FIS** в зовнішньому файлі з ім’ям **mytip.fis,** виконавши команду меню **File**>**Export**>**To Disk** …. При цьому буде викликано стандартне діалогове вікно збереження файлу, в якому коритстувачеві пропонується ввести ім’я відповідного файлу (розширення файлу приписується автоматично). Залишимо без зміни запропоновані системою **MatLab** за умовчанням: метод нечіткого логічного І (**And method**) – значення «min», метод нечіткого логічного АБО (**Or method**) значення «max», метод імплікації (**Implication**) – значення «min», метод агрегації (**Aggrgation**) значення «max» і метод дефазифікації (**Defuzzification**) значення «centroid». Очевидно, ці значення можуть бути змінені користувачем.
    2. Тепер необхідно визначити терми і їх функції приналежності для вхідних і вихідних змінних нашої системи нечіткого виведення. Для цього слід скоритсуватися редактором функцій приналежності, який може бути викликаний одним з наступних способів:
* подвійним клацанням на значку прямокутника з ім’ям відповідної змінної
* командою меню **Edit**>**Membership Functions**…(заздалегідь повинен бути виділений прямокутник з ім’ям відповідної змінної)
* натисненням клавіши <**Ctrl**>+<2> (заздалегідь також повинен бути виділений прямокутник з ім’ям відповідної змінної)

Після виклику редактора функцій приналежності кожній із змінних за умовчанням пропонується 3 терми з трикутними функціями приналежности (рис. 29).

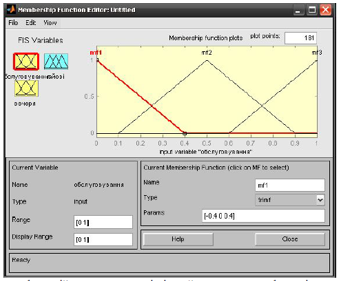


Рисунок 29. Вид редактора функцій приналежності після його виклику з функціями приналежності для термів змінної «обслуговування», запропонованих системою MatLab за замовчуванням

Спочатку змінемо діапазон визначення значень вхідних змінних, для чого в полях введення **Range** і **Display Range** змінемо верхнє значення з 1 на 10 (балів). Аналогічно виконуються зміни відповідних діапазонів для взідної змінної «чайові», при цьому верхнє значення 1 слід замінити на 30(%). Зміни підтверджуються натисненням на клавішу <**Enter**> на клавіатурі.

Далі змінемо назви термів першої вхідної змінної «*обслуговування*», запропоновані системою MatLab за умовчанням (mf1, mf2, mf3) на «*погане*», «*добре*», «*відмінне*» відповідно. Після чого змінимо тип функцій приналежності першої змінної, запропонований за умовчанням на функції типу Гауса (gaussmr), вибравши відповідний пункт в полі **Type**. Параметри знов заданих функцій приналежності залишимо без зміни. Вид редактора функцій приналежності після внесених змін для першої з вхідних змінних зображений на рисунку 30. Аналогічним чином змінемо назви термів другої вхідної змінної «вечеря» і видалимо один з термів з відповідною функцією приналежності. Для виделенння терма слід виділити функцію приналежності, що видаляється, і натиснути клавішу <**Delete**> на клавіатурі. Перехід до редагування змінної здійснюється клацанням на зобраєенні прямокутника з ім’ям необхідної змінної. Для змінної «*вечеря*» змінимо тип функцій приналежності і її термів на трапецієвидні функції (trapmf) і їх параметри таким чином: для терма «підгоріла» задамо переметри [0 0 1 3}, а для терма «*смачна*» - {7 9 10 10].

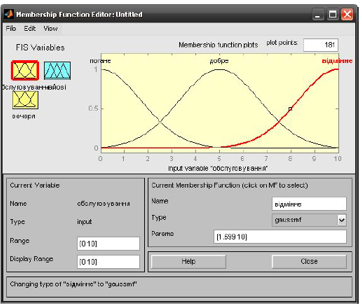


Рисунок 30. Вид редактора функцій приналежності після зміни назви термів і типу їх функцій приналежності для першої вхідної змінної «*обслуговування*»

Вид редактора функцій приналежності після внесених змін для другої вхідної змінної зображений на рисунку 31.

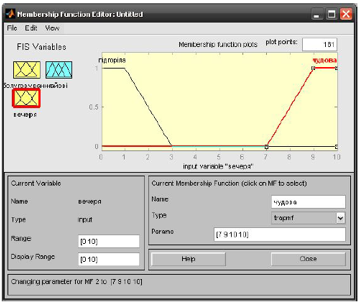


Рисунок 31. Вид редактора функцій приналежності після зміни назви термів і типу їх функцій приналежності для другої вхідної змінної «*вечеря*»

Нарешті, змінимо назви термів і параметри функцій приналежності для вихідної змінної «*чайові*», залишивши без зміни трикутний тип функції приналежності, запропонований сисемою **MatLab**. Для терма «*малі*» задамо параметри [0 5 10], а для терма «*середні*» - [10 15 20], для терма «*щедрі*» - [20 25 30]. Вид редактора функцій приналежності після зроблених змін для вихідної змінної «*чайові*» зображений на рисунку 32.

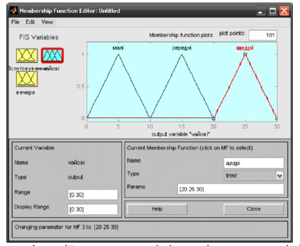


Рисунок 32. Вид редактора функцій приналежності після зміни назви термів і типу їх функцій приналежності для вихідної змінної «обслуговування»

* + 1. Тепер настала черга визначити правила нечіткого виведення для створюваної експертної системи. Для цього слід скористатися редактором правил, який може бути викликаний одним з наступних способів:
* Подвійним клацанням на значку квадрата в центрі з ім’ям створюваної системи нечіткого виведення (myfis);
* Командою меню **Edit**>**Rules**…;
* Натисненням клавіш <**Ctrl**>+<**3**>.

Оскільки спочатку база правил нечіткого виведення порожня, то після виклику редактора правил центральне багаторядкове поле введення не містить ніяких правил. Для їх визначення слід використовувати поля меню і перемикачі в нижній частині графічного інтерфейсу редактора правил. Для задання першого правила слід залишити виділені за умовчання поле з ім’ям терма «*погане*» для першої вхідної змінної, поле з ім’ям терма «*підгоріла*» для другої вхідної змінної і поле з ім’ям терма «*малі*» для вихідної змінної. Далі слідує перемикач **Connection** поставити в положення or (логічне АБО) і натиснути на кнопку **Add rule**. Після цього перше правило з символами кирилиці відобразиться у верхньому вікні.

Аналогічним чином задається друге правило, для якого слід виділити імена термів «*добре*», «*none*» і «*середні*», і третє правило з іменами термів «*відмінне*», «*чудовий*» і «*щедрі*» для відповідних змінних. Вид редактора правил після їх визначення для створюваної експертної системи зображений на рис.33.

Відмітимо, що в полі введення Weight відображається вага кожного правила, яку можна змінювати в інтервалі [0 1] (залишимо без зміни його значення за умовчанням, рівне 1 для всіх правил). Ця ж вага правил записується в круглих дужках у вікні правил після кожного з правил нечіткого виведення.

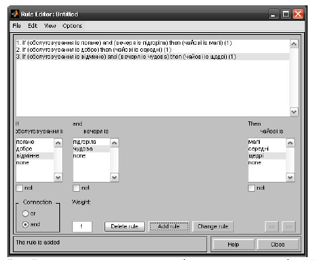


Рисунок 33. Вид редактора правил нечіткого виведення після їх визначення

* + 1. Після задання правил нечіткого виведення виявляється можливим отримати результати нечіткого виведення (значення вихідної змінної) для конкретних значень вхідних змінних. З цією метою необхідно відкрити програму перегляду правил одним з наступних способів:
* командою меню **View>Rules** редактора **FIS**;
* командою меню **View>Rules** редактора функцій приналежності;
* командою меню **View>Rules** редактора правил;
* натисненням клавіш <**Ctrl**>+<**5**>.

Після виклику програми перегляду правил для нашої системи нечіткого виведення за умовчанням для вхідних змінних запропоновані середні значення з інтервалу їх допустимих значень (значення [5 5] в полі введенні Input). Це означає, що відвідувач ресторану оцінює якість обслуговування в 5 балів і якість вечері також в 5 балів. Цим значенням вхідних змінних відповідає значення «чайових» в 15%, яке відображається вище за прямокутники правил в правій частині вікна програми перегляду. Змінимо значення вхідних змінних для іншого випадку, якому відповідає якість обслуговування в 0 балів («гірше нікуди») і якість вечері в 10 балів («краще не буває»). Для цього курсор миші перемістимо в поле введення Input і введемо відповідні значення вхідних змінних [0 10]. Система **MatLab** залишить значення чайових без зміни (15%), проте на діаграмі правил можна відмітити результати виконаних змін. (рис. 34).

Оскільки процес нечіткого моделювання припускає аналіз результатів нечіткого виведення при різних значеннях вхідних змінних з метою встановлення адекватності розробленої нечіткої моделі ( в даному випадку – експертної системи), розглянемо і інші випадки. Припустимо, що якість обслуговування оцінюєься в 10 балів («краще не буває»), а якість вечері в 2 бали («буває і гірше, але рідше»). Введемо відповідні значення змінних аналогічним чином. В цьому випадку розроблена система нечіткого виведення рекомендує нам залишити чайові у розмірі 16,4%.

Якщо ж припустити, що якість обслуговування все ще відмінна (10 балів), а якість вечері дещо покращалась і оцінюється в 3 бали, то величина чайових істотно зміниться і стане рівною 24,7%. Більш того, подальше збільшення якості вечері не надає зміни величини чайових. Зокрема, для значень вхідних змінних [10 10] величина чайових складе як і раніше 24,7%. Якщо деяким з відвідувачів така експертна система здається неадекватною (зокрема, для випадку значень вхідних змінних [10 10] можна залишити максимальні чайові 30%), то розроблена система нечіткого виведення потребує модифікації. Дана модифікація може потребувати зміни існуючих правил або додавання нових, а також зміни параметрів функцій приналежності вхідних і вихідних змінних. Точніше настройка моделі може бути пов’язана зі збільшенням кількості термів для кожної з вхідних і вихідних змінних, що у свою чергу, приведе до збільшення кількості правил в системі нечіткого виведення і загальному ускладненню моделі.

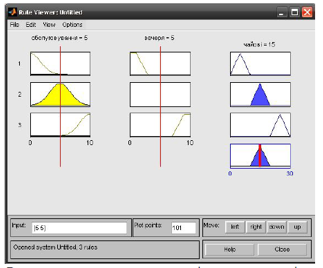


Рисунок 34. Вид програми перегляду правил нечіткого виведення після зміни значень вхідних змінних на [0 10]

* + 1. Для остаточного аналізу розробленої нечіткої моделі може виявитися корисною програма перегляду поверхні нечіткого виводу, яка може бути викликана одним з наступних способів:
* командою меню **View**>**Surface** редактора **FIS**;
* командою меню **View**>**Surface** редактора функцій приналежності;
* командою меню **View**>**Surface** редактора правил;
* командою меню **View**>**Surface** програми перегляду правил;
* натисненням клавіш <**Ctrl**>+<**6**>.

Графічний інтерфейс програми перегляду поверхні нечіткого виведення для розробленої нечіткої моделі зображений на рис. 35.

Ця програма служить для загального аналізу адекватності нечіткої моделі, дозволяючи оцінити вплив зміни значень вхідних нечітких змінних на значення однієї з вихідних нечітких змінних. Для цього необхідно вибрати потрібну змінну в списку, що розкривається, **X (Input**), а в списку **Y (input)**, що розкривається, вибрати значення «*none*». Отриманий графік залежності зображений на рис. 36. Отриманий графік залежності відповідає середньому значенню другої вхідної змінної («*якість вечері*») в 5 балів. Це значення може бути змінене користувачем, для чого слід ввести потрібне значення в поле введення Ref.Input. Зазначимо, що значення NAN для першої вхідної змінної відповідає її зміні у всьому інтервалі визначення [0 10].

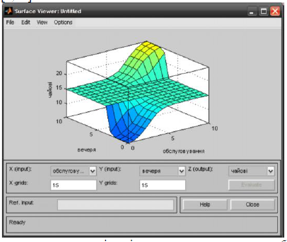


Рисунок 35. Вид програми перегляду поверхні нечіткого виведення для розробленої нечіткої моделі

Закінчуючи розгляд процесу розробки простої системи нечіткого виведення в інтерактивному режимі, слід зазначити, що найбільш ефективним цей спосіб виявляється для складних нечітких моделей з великим числом змінних і правил нечіткого виведення. В цьому випадку задання змінних і функцій приналежності їх термів в графічному режимі, а також візуалізація правил дозволяють істотно зменшити трудомісткість розробки нечіткої моделі, кількість можливих помилок і скоротити загальний час нечіткого моделювання.

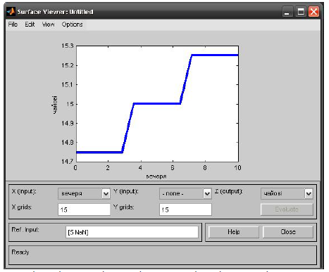


Рисунок 36. Графік залежності вихідної змінної від першої з вхідних змінних для розробленої нечіткої моделі

В той же час, слід пам’ятати, що кількість змінних і правил в нечіткій моделі, які можуть бути візуалізовані, обмежена. Зокрема, якщо число вхідних змінних перевищує 10, то їх відображення у відповідних графічних редакторах відбувається з відхиленнями.

Процес розробки нечіткого виведення в режимі команд може доповнити, а в окремих випадках, і замінити процес розробки в інтерактивному режимі, надаючи користувачеві повний контроль над всіма змінними в робочій області системи **MatLab**.

Для виконання завдання необхідно:

1. Запустити **Fuzzy Logic Toolbox** за допомогою команди «**fuzzy***».*

2. Додати другу вхідну лінгвістичну змінну за допомогою меню «**Edit → Add Variable → Input**» (рис. 37):

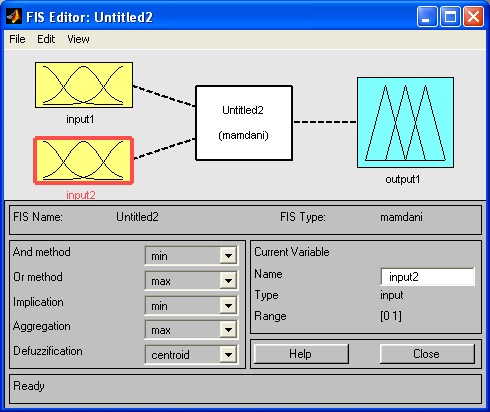
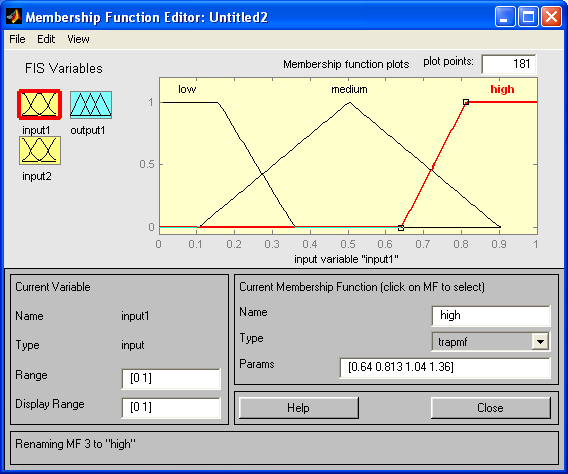
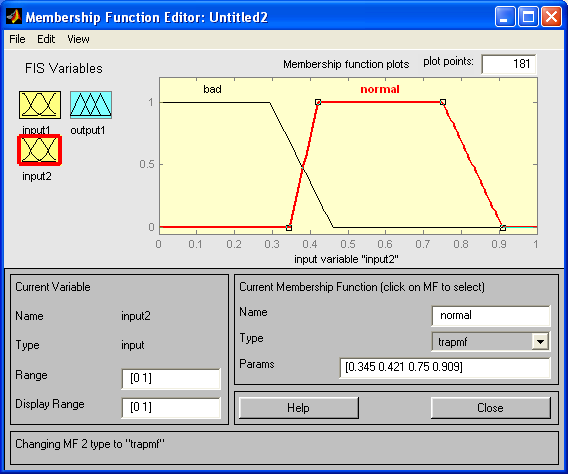


Рис. 37. Додавання лінгвістичної змінної

3.Відредагувати параметри функцій приналежності відповідно до умов завдання (рис. 38).

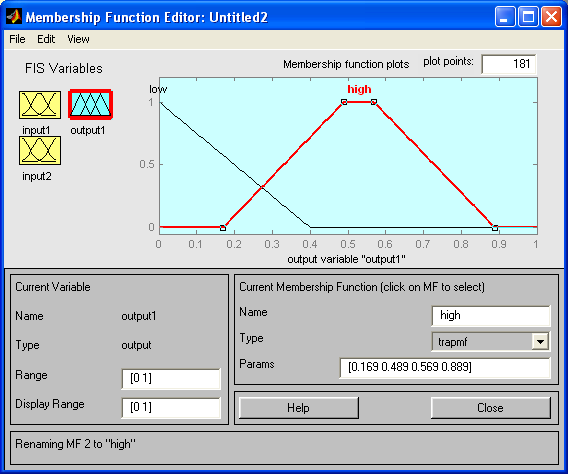


Рисунок 38. Налаштування параметрів функції приналежності

* 1. Сформулювати набір правил (рис. 39).

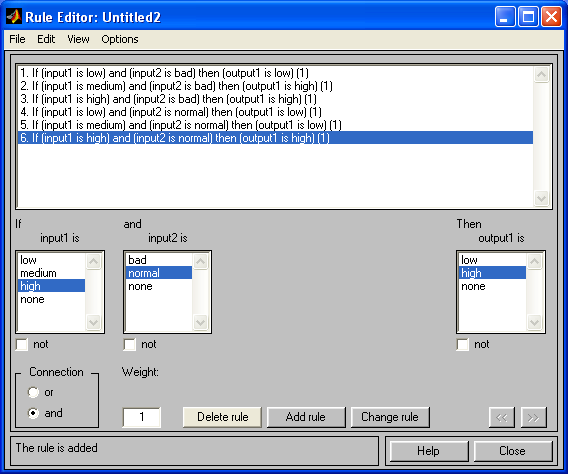


Рисунок. 39. Вікно редактора правил

* 1. Розглянути результати моделювання механізму прийняття рішень за допомогою меню «**View → Rules**» (рис. 40).

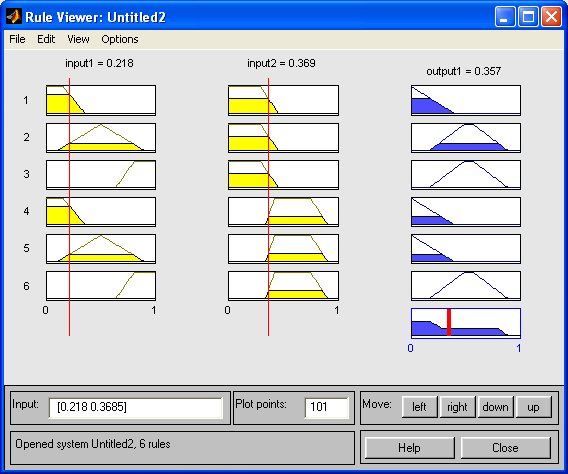


Рисунок. 40. Вікно результатів моделювання

Наприкінці лабораторної роботи необхідно проаналізувати результати і зробити висновок.

*Приклад висновку:*

Досліджено програмний продукт **MatLab** та пакет математичного моделювання **Fuzzy Toolbox** у ньому. Набуто більш глибоких знань щодо нечіткої логіки та механізмів прийняття рішень на її основі. Для виконання всіх завдань з нечіткої логіки використовувалися 3 функції приналежності – трапецієподібна, Гауса та трикутна. Ці функції були використані для передачі можливостей методу експертних думок.

Наприклад, для рівня води було обрано три функції Гауса для трьох параметрів нечіткої логіки у другому завданні для очікуваного рівня води, а для всіх інших комбінування трапецієподібної та трикутної. Таким чином можна легше передати різні параметри нечіткої логіки для всіх завдань.

Порядок виконання роботи

1) Здійсніть вибір функцій приналежності, використовуючи метод експертних оцінок (варіанти завдання наведено у Додатку А).

2) Побудуйте базу правил за допомогою вибраних функцій приналежності. Реалізовану базу правил відобразіть у звіті в табличному вигляді за зразком таблиці 4.

Визначаємо всі правила, із зазначенням відповідних значень нечітких змінних *x*, *y* і *z,* та заповнюємо таблицю на прикладі таблиці 4:

Таблиця 4

База правил

|  |  |  |  |
| --- | --- | --- | --- |
| *y / х* | *S* | *M* | *L* |
| *S* |  |  |  |
| *L* |  |  |  |

3) Промоделюйте механізм нечіткого логічного виведення за допомогою пакету прикладних програм *Matlab*.

4) Відобразіть результати роботи в звіті.

5) Підготуйте відповіді на контрольні запитання.

Вимоги до звіту

Звіт подається в електронному вигляді.

Звіт повинен містити такі результати:

1) Опис сформульованих правил логічного висновку у табличній формі.

2) Скріншоти (знімки екранів) графіків функцій приналежності, бази правил та результатів моделювання.

3) Короткі висновки з виконаної роботи (з оцінкою ефективності розглянутих методів).

Контрольні запитання

1) Вкажіть відмінності нечіткої змінної від лінгвістичної змінної.

2) Наведіть основні типи функцій приналежності.

3) Опишіть основні етапи нечіткого логічного виведення з використанням методу max/min.

4) У чому полягає суть етапу дефаззифікації?

5) Назвіть основні переваги нечітких алгоритмів прийняття рішень.

Варіанти завдання

**Варіант 1**

1.1

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Середня заробітна плата клієнта  *y* – Тривалість роботи на останньому місці | *z* – ступінь фінансової довіри |

1.2

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Вік  *y* – Соціальний статус | *z* – Ступінь соціальної довіри |

1.3

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Ступінь фінансової довіри  *y* – Ступінь соціальної довіри | *z* – Максимальна сума споживчого кредиту |

**Вариант 2**

2.1

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Рівень опадів  *y* – Середньодобова температура | *z* – Швидкість танення снігового покриву |

2.2

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Максимальний очікуваний рівень води у водосховищі  *y* – Площа басейну водосховища | *z* – Обсяг спуску води |

2.3

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Швидкість танення снігового покриву  *y* – Обсяг спуску води | *z* – Очікуваний рівень повені |

**Вариант 3.**

3.1

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Якість сценарію  *y* – Якість гри акторів | *z* – Якість режисерської роботи |

3.2

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Бюджет фільму  *y* – Рівень спецефектів | *z* – Видовищність фільму |

3.3

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Якість режисерської роботи  *y* – Видовищність фільму | *z* – Розмір касових зборів фільму |

**Вариант 4.**

4.1

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *х* – Якість викладання  *y* – Якість лабораторно-методичної бази | *z* – Якість освітнього процесу |

4.2

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Рівень мотивації студента  *y* – Ступінь винахідливості студента | *z* – Середня успішність студента |

4.3

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Якість освітнього процесу  *y* – Середня успішність студента | *z* – Рівень засвоєння знань |

**Вариант 5**

5.1

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Рівень фінансових вкладень  *y* – Термін окупності проекту | *z* – Ступінь ефективності інвестицій |

5.2

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Кваліфікація персоналу  *y* – Якість матеріальної бази | *z* – Якість виконаних робіт |

5.3

|  |  |
| --- | --- |
| **Вхідні лінгвістичні змінні** | **Вихідна лінгвістична змінна** |
| *x* – Ступінь ефективності інвестицій  *y* – Якість виконаних робіт | *z* – Ефективність проекту |

**Лабораторна робота №6**

**ПРОЕКТУВАННЯ СППР НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ. ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ РОБОТИ АЛГОРИТМІВ МАМДАНІ ТА СУГЕНО В ОДНОТИПНИХ ДОДАТКАХ**

**Основні теоретичні відомості**

Створення та настройка експертної системи з використанням **Fuzzy Logic Toolbox Fuzzy Logic Toolbox**– це пакет прикладних програм, що входять до складу середовища **MatLab**. Він дозволяє створювати системи нечіткого логічного виведення і нечіткої класифікації в рамках середовища MatLab з можливістю їх інтеграції в **Simulink**.

Основні властивості:

* визначення змінних, нечітких правил і функцій належності;
* інтерактивний перегляд нечіткого логічного виведення;
* сучасні методи: адаптивне нечітке виведення з використанням нейронних

мереж, нечітка кластеризація;

* інтерактивне динамічне моделювання в Simulink;
* генерація переносного *С* коду за допомогою **Real-Time Workshop**.

Пакет **Fuzzy Logic**містить п’ять графічних редакторів для представлення необхідної інформації в процесі проектування, створення і тестування нечітких моделей.

Пакет **Fuzzy Logic**містить сучасні методи нечіткого моделювання, включаючи:

* адаптивне нечітке виведення з використанням нейронних мереж для автоматичного формування функції належності в процесі навчання їх на вхідних даних;
* нечітку логіку і кластеризацію для задач розпізнавання образів;
* можливість вибору широко відомого метода Мамдані або метода Сугено для створення гібридних нечітких систем.

Пакет дозволяє роботу:

* у режимі графічного інтерфейсу;
* у режимі командного рядка;
* з використанням блоків та прикладів пакета Simulink.

1. Базовим поняттям **Fuzzy Logic Toolbox**є **FIS***-структура* – система нечіткого виведення (**Fuzzy Inference System**). **FIS***-структура* містить усі необхідні дані для реалізації функціонального відображення «входи-виходи» на основі нечіткого логічного виведення згідно зі схемою, наведеною на рис. 41.



Рисунок 41. Нечітке логічне виведення

Позначення: *X* – вхідний чіткий вектор; - вектор нечітких множин, що відповідає вхідному вектору *X*; - результат логічного виведення у вигляді вектора нечітких множин; *Y* – вихідний чіткий вектор.

*Склад графічного інтерфейсу*

**Fuzzy Logic Toolbox**містить наступні редактори:

- редактор нечіткої системи виведення **Fuzzy Inference System Editor(FIS Editor**або **FIS***- редактор*) разом з додатковими програмами – редактором функцій належності (**Memberhip Function Editor**), редактором правил (***Rule Editor***), вікно перегляду правил (**Rule Viewer**) і вікном перегляду поверхні відгуку (**SurfaceViewer**);

1. - редактор гібридних систем (**ANFIS Editor, ANFIS***-редактор*);
2. - програма знаходження кластерів (програма **Clustering**– кластеризація).

*Редактор нечіткої системи виведення*

Командою (функцією) **fuzzy**з режиму командного рядка запускається основна інтерфейсна програма пакета **Fuzzy Logic**– редактор нечіткої системи виведення. Головне вікно наведено на рисунку 42.

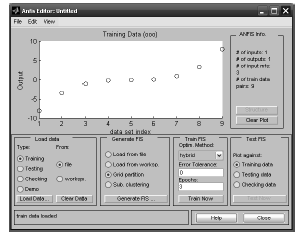


Рисунок 42. Вікно редактора гібридних систем

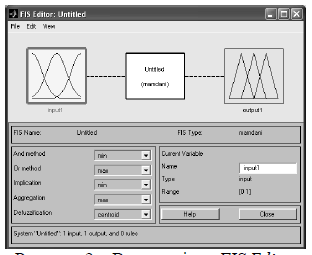


Рисунок 43. Вигляд вікна **FIS Editor**

*Графічний інтерфейс гібридних мереж*

Головне вікно редактора **ANFIS Editor**викликається командою **anfisedit**з командного рядка, вигляд якого наведено на рисунку 44.

За допомогою даного редактора виконується створення або завантаження гібридної системи, перегляд структури, настроювання її параметрів, перевірка якості функціонування такої системи.

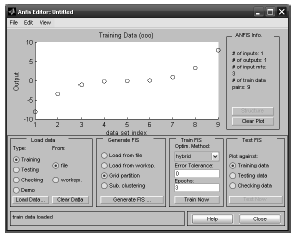


Рисунок 44. Вікно редактора гібридних систем

*Графічний інтерфейс програми кластеризації*

Програма **Clustering**(кластеризація) дозволяє виявляти центри кластерів, тобто точки в багатовимірному просторі даних, біля яких групуються (скупчуються) експериментальні дані.

Запуск програми **Clustering**виконується командою **findcluster**. На рисунку 45 наведено приклад використання програми.

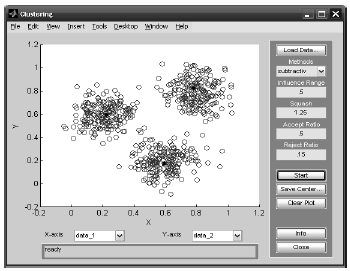


Рисунок 45. Результат роботи програми **Clastering**

Робота в редакторі нечіткої системи виведення **Fuzzy Inference System Editor.**

Для завантаження основного **FIS***-редактора* надрукуємо слово **fuzzy**в командному рядку. Після цього відкриється нове графічне вікно, зображене на рисунку 42. Для того щоб додати нову вхідну змінну, необхідно в меню **Edit**вибрати команду **Add Variable***…\***Input**. Для зміни імені змінної необхідно ввести нове ім’я в полі **Name**і натиснути клавішу **Enter**. Для того щоб задати ім’я системі, необхідно в меню **File**вибрати в підменю **Export**команду **To File**і ввести ім'я файла.

Щоб перейти в редактор функцій приналежності, необхідно двічі натиснути на будь-якій з функцій, де можна вибирати властивості конкретної, вікно відображено на рисунку 46:

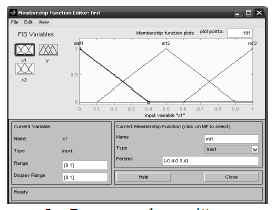


Рисунок 46. Редактор функцій належності

Внизу вікна вказуються наступні властивості функції:

*Current Variable:*

* Name – ім’я функції;
* Type – тип (вхідна чи вихідна);
* Range – діапазон змінної;
* Display Range – відображуваний діапазон.

*Current Membership Function:*

* Name – ім’я поточної функції належності;
* Type – тип терму функції належності – вибирається з переліку (трикутна, трапецеїдальна, ґаусові 1 та 2-го порядку та інші);
* Params – числові значення терму функції належності.

Для задання нових функцій належності для змінної необхідно в меню **Edit**вибрати команду **Add MFs**... У результаті з'явитися діалогове вікно (рис. 47) вибору типу і кількості функцій належності.

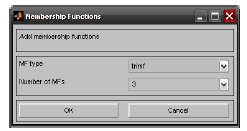


Рисунок 47. Задання функції належності

Вибравши необхідну кількість термів та їх тип, натиснути OK.

Ім’я та числове значення термів можна змінити, виділивши необхідний, і задати у відповідних полях області **Current Membership Function**нові значення.

*Редактор бази знань* **RuleEditor**

Для виклику редактора необхідно вибрати в меню **Edit**команду **Rules...**, відобразиться головне вікно, зображене на рисунку 48.

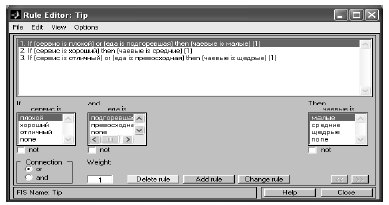


Рисунок 48. Редактор правил

Для створення нових правил необхідно вибрати відповідну комбінацію термів і залежностей, вибрати тип зв’язку: **or**або **and**, вагу правила **Weight,** значення вихідної змінної та натиснути кнопку **Add rule**.

Для перегляду вікна візуалізації нечіткого логічного виведення викликаємо його командою **View rules**... меню **View**.

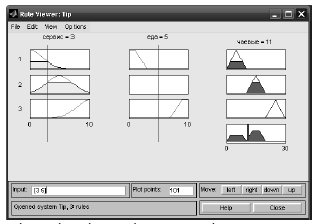


Рисунок 49. Візуалізація нечіткого логічного виведення в **Rule Viewer**

Можна переглянути поверхню “входи-виход”, відповідну синтезованій нечіткій системі. Для виведення цього вікна необхідно використовувати команду **View surface...** меню **View**.

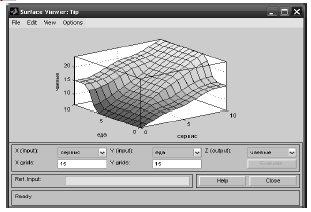


Рисунок 50. Поверхня відгуку нечіткої системи

Алгоритми нечіткого виведення різняться, головним чином, видом використовуваних правил, логічних операцій і різновидом методу дефазифікації. Розроблені моделі нечіткого виведення Мамдані, Сугено, Ларсена, Цукамото. При розгляді алгоритмів для спрощення припустимо, що базу знань організують два нечітких правила вигляду:



де *х* і *у* – імена вхідних змінних, *z* – ім’я змінної виведення, - деякі задані функції належності, при цьому чітке значення *z0* необхідно визначити на основі наведеної інформації та чітких значень *x0* і *y0*.

**Алгоритм Мамдані (Mamdani)**

Алгоритм Мамдані є одним з перших, який знайшов застосування в системах нечіткого виведення. Він був запропонований 1975 р. англійським математиком Е. Мамдані (Ebrahim Mamdani) як метод для керування паровим двигуном. Формально *алгоритм Мамдані* може бути визначений таким чином.

1. Процедура фазифікації: визначаються ступені істинності, тобто значення функцій належності для лівих частин кожного правила (передумов): 
2. Нечітке виведення: знаходяться рівні відтинання для передумов кожного з правил з використанням операції мінімум:



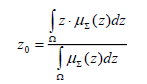
де через  позначена операція логічного мінімуму (min), потім знаходяться «зрізані» функції належності



3. Композиція: з використанням операції максимуму (max, позначення: ) виконується об'єднання знайдених зрізаних функцій, що приводить до отримання підсумкової нечіткої підмножини для змінної виходу з функцією належності.



4. Приведення до чіткості (для знаходження z0) проводиться, наприклад, центроїдним методом (як х – координата центра ваги функції належності підсумкової нечіткої підмножини для змінної виходу):



Алгоритм ілюструється на рис. 51:

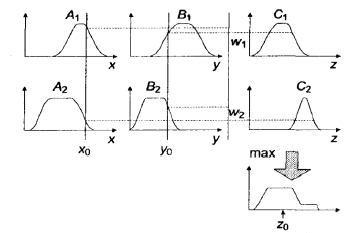


Рисунок 51. Графічна реалізація

**Алгоритм Сугено (Sugeno)**

Формально алгоритм Сугено, запропонований Сугено і Такагі, може бути визначений таким чином.

1. Перший етап – як в алгоритмі Мамдані.
2. На другому етапі знаходяться  та індивідуальні виходи правил:



1. На третьому етапі визначається чітке значення змінної виведення:



Алгоритм ілюструється на рис. 52:

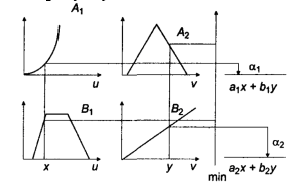


Рисунок 52. Графічна реалізація алгоритму Сугено

**Порядок виконання роботи**

1. Порівняти алгоритми Мамдані і Сугено на прикладі створення системи нечіткого логічного виведення, що моделює залежність .Проектування системи нечіткого логічного виведення необхідно провести на основі графічного зображення вказаної залежності. (Кожен вибирає свою залежність і проходить пункти 2-12).

2. Для побудови тривимірного зображення функції необхідно скласти наступну програму, прописавши її в m-файлі:

%Побудова графіка функції y = x1^2\*sin(x2-1) в межах x1є[-7, 3] і x2є[-4.4, 1.7].

n = 15; % кількість точок

x1 = -7:10/(n-1):3; % задання параметрів змінної *х1*

x2 = -4.4:6.1/(n-1):1.7; % задання параметрів змінної *х1*

y = zeros (n, n); % формування нульового масиву

% розміром n×n для вихідної змінної

for j = 1:n

y (j,:) = x1.^2\*sin(x2(j)-1);

end

surf (x1, x2, y) % зображення поверхні функції

xlabel ('x1')

ylabel ('x2')

zlabel ('y')

title ('Target');

У результаті виконання програми отримаємо графічне зображення, наведене на рис. 53.

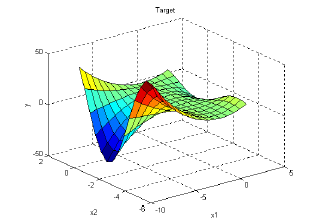


Рисунок 53. Еталонна поверхня

3. Реалізуйте дві нечіткі системи для заданої функції, вибравши для першої тип системи Сугено, для другої – Мамдані.

4. Для створення першої системи завантажте **FIS**-редактор. Виберіть тип системи – Sugeno. Додайте другу вхідну змінну та назвіть усі змінні відповідними іменами, а саме першу вхідну змінну перейменуйте на х1, другу – на х2, а вихідну – на у.

5. Перейдіть у редактор функцій належності. Задайте діапазон змінення змінної x1 та створіть для неї функції належності, вказавши ім’я, тип та числові значення термів:

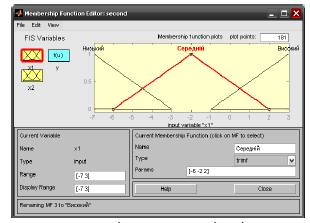


Рисунок 54. Функції належності змінної х1

6. Аналогічно задайте діапазон змінення змінної x2 та створіть для неї функції належності, вказавши ім’я, тип та числові значення термів:

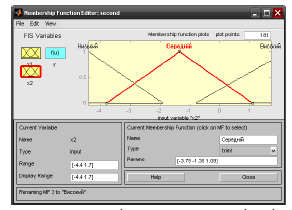


Рисунок 55. Функції належності змінної х2

7. Для алгоритму Сугено для вихідної змінної задаються лінійні залежності між входами і виходом, що мають міститися в базі знань. У базі знань вказано 5 різних залежностей: y=50; y=4x1-x2; y=2x1+2x2+1; y=8x1+2x2+8; y=0. Тому додайте ще дві залежності шляхом вибору команди **Add Mfs**меню **Edit.** У діалоговому вікні в полі **Number of MFs**виберіть 2 і натисніть кнопку OK. Задайте найменування і параметри лінійних залежностей. Для цього виберіть першу залежність mf1. Надрукуйте найменування залежності, наприклад 50, у полі **Name**і встановіть тип залежності - константа шляхом вибору опції ***Сonstant*** в меню **Type**. Після цього введіть значення параметра 50 у полі **Params**. Аналогічно для другої залежності mf2 введіть найменування залежності, наприклад 8+8x1+2x2. Потім вкажіть лінійний тип залежності шляхом вибору опції Linear у меню **Type**і введіть параметри залежності 8 2 8 в полі **Params**. Для лінійної залежності порядок параметрів наступний: перший параметр – коефіцієнт при першій змінній, другий, – при другій і т. д., останній параметр – вільний член залежності. У результаті маєте отримати графічне вікно, зображене на рис.56.



Рисунок 56. Вікно лінійних залежностей «входи-вихід»

8. Аналізуючи еталонну поверхню, можна скласти наступні залежності та правила:

якщо x1=середній, то y=0;

якщо x1=високий і x2=високий, то y=2x1+2x2+1;

якщо x1=високий і x2=низький, то y=4x1-x2;

якщо x1=низький і x2=середний, то y=8x1+2x2+8;

якщо x1=низький і x2=низький, то y=50;

якщо x1=низький і x2=високий, то y=50.

9. Перейдіть у редактор бази знань **RuleEditor**і введіть правила бази знань, що наведені вище.

10. Перегляньте вікно візуалізації нечіткого логічного виведення, а також поверхню “входи–вихід” для синтезованої нечіткої системи.

11. Реалізуйте нечітку логічну систему для заданої функції, використовуючи алгоритм Мамдані. Самостійно складіть правила для відповідних функцій.

12. Порівняйте отримані різними методами поверхні з еталонною поверхнею. Зробіть висновки щодо ефективності кожної з них.

**Вимоги до звіту звіту**

1. Указати номер, тему й мету лабораторної роботи.

2. Зобразити **FIS**-структури для розроблених систем за різними алгоритмами.

3. Навести перелік правил.

4. Відобразити отримані результати – перехідні процеси, поверхні відгуку.

5. Зробити порівняльні висновки стосовно роботи системи з різним настройками.

**Контрольні запитання**

1. Яка множина називається «нечіткою»?

2. Чим нечітка логіка відрізняється від звичайної?

3. Які є методи побудови функції належності?

4. Призначення функції належності?

5. Які існують функції приналежності?

6. Чим відрізняється алгоритм Mamdani від алгоритма Sugeno?

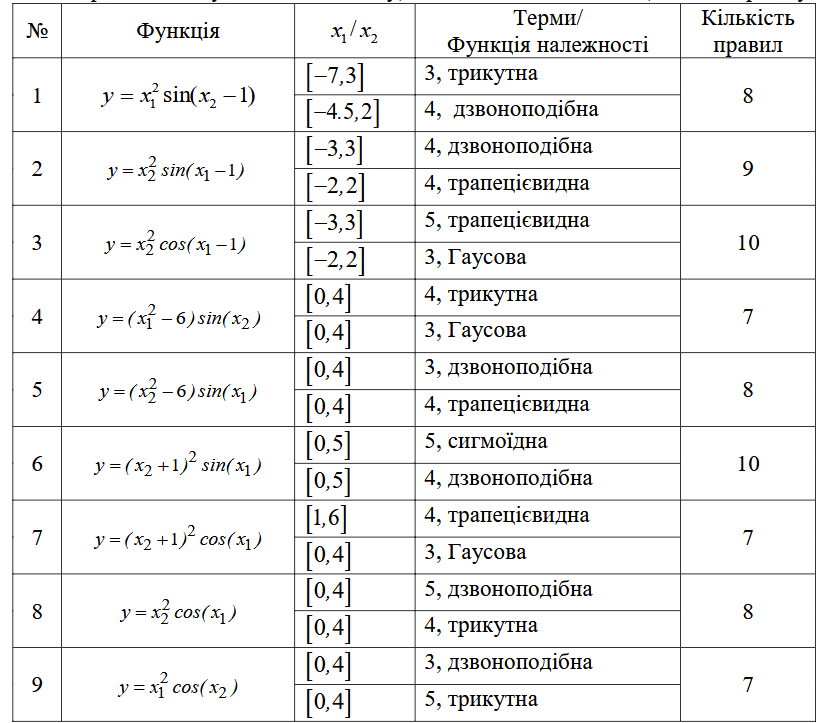
7. Що таке лінгвістична змінна?

8. Що таке терм-множина?

9. Що таке фазифікація та дефазифікація змінних?

**Варіанти завдання**

Таблиця 5.

****



**Список літератури**

1**.** Методи експертних оцінок в системах прийняття рішень. Навчальний посібник/Терентьєв О.О., Київська К.І., Делембовський М.М., Серпінська О.І. -Київ: ФОП Ямчинський О.В.,2020. – 116 с. іл.

2. Аналіз сучасних інформаційних технологій системи діагностики технічного стану будівель і споруд /Терентьєв О.О., Київська К.І., Доля О.В., Горбатюк Є.В.// – Київ: Управління розвитком складних систем, збірник наукових праць, випуск 36/2018, КНУБА, 2018. С. 100-107.

3. Архітектура інформаційної системи діагностики технічного стану безпечної експлуатації будівель / Терентьєв О.О., Горбатюк Є.В., Доля О.В., Лященко Т.А., Серпінська О.І. // – Київ: Управління розвитком складних систем, збірник наукових праць, випуск 44/2020, КНУБА, 2020.

4. Горбатюк Є.В., Терентьєв О.О., Доля О.В., Бородиня В.В. Оцінка недосконалостей будівельних конструкцій на основі нечітких множин. The 2nd International scientific and practical conference “Scientific achievements of modern society” (October 9-11, 2019) Cognum Publishing House, Liverpool, United Kingdom. 2019. P.163-169.

5. Інтегровані моделі і методи автоматизованої системи діагностики технічного стану об’єктів будівництва [Текст] : монографія /В.М. Міхайленко, П.Є. Григоровський, І.В. Русан, О.О. Терентьєв // – Київ: ЦП «Компринт», 2017. –230 с.

6. Інтегровані моделі та методи автоматизованої системи діагностики технічного стану конструкцій будівель та споруд. Підручник /О.О. Терентьєв, І.В. Русан, Є.В. Горбатюк, І.С. Івахненко, О.В. Петроченко, О.П. Куліков. – Київ: Компрінт, 2019. – 239 с.:іл.

7. Інтегровані моделі, які забезпечують прогнозування надійності прийняття рішень для задачі системи діагностики технічного стану будівель /Терентьєв О.О., Шабала Є.Є., Саченко І.А.// – Київ: Управління розвитком складних систем, збірник наукових праць, випуск 32/2017, КНУБА, 2017. С. 76-80.

8. Інтелектуальні інформаційні системи і технології діагностики технічного стану будівель. Навчальний посібник / О.О. Терентьєв, І.В. Русан, Є.В. Бородавка, Є.В., Горбатюк, К.І. Київська. – Київ: Компрінт, 2019. – 121 с.:іл.

9. Міхайленко В.М. Експериментальні дослідження та реалізація інформаційної системи тестування нейронної мережі для задачі діагностики технічного стану будівель / Терентьєв О.О., Шабала Є.Є., Турушев О.С.// – Київ: Управління розвитком складних систем, збірник наукових праць, випуск 27/2016, КНУБА, 2016. – С. 139-144.

10. Міхайленко В.М. Аналіз сучасних інформаційних методів системи діагностики технічного стану будівель /Терентьєв О.О., Шабала Є.Є.// – Київ: Управління розвитком складних систем, збірник наукових праць, випуск 29/2017, КНУБА, 2017. – С. 136-143.

11. Моделі та методи інформаційної системи діагностики технічного стану об’єктів будівництва. Підручник /В.М. Міхайленко, І.В. Русан, П.Є. Григоровський, О.О. Терентьєв, А.Т. Свідерський, Є.В. Горбатюк. – Київ: Компрінт, 2018. – 325 с.:іл.

12. Методи та моделі пошкодження автоматизованої системи діагностики технічного стану об`єктів будівництва / Терентьєв О.О., Горбатюк Є.В., Доля О.В., Київська К.І., Азенко В.В., Бородиня В.В.// – Київ: Управління розвитком складних систем, збірник наукових праць, випуск 38/2019, КНУБА, 2019. – С. 82–91.

13. Методологія створення експертної оцінки автоматизованої системи діагностики технічного стану об’єктів будівництва /Терентьєв О.О., Шабала Є.Є., Доля О.В., Чередніченко Д.О.// – Київ: Управління розвитком складних систем, збірник наукових праць, випуск 33/2018, КНУБА, 2018. С. 157-162.

14. Побудова системи діагностики технічного стану будівель на прикладі балки перекриття з використанням методів теорії нечітких множин /Терентьєв О.О., Шабала Є.Є., Саченко І.А.// – Київ: Управління розвитком складних систем, збірник наукових праць, випуск 31/2017, КНУБА, 2017. С. 145-153.

15. Розробка програмного забезпечення підсистеми інтелектуальної інформаційної технології діагностики технічного стану екологічних будівель / Терентьєв О.О., Горбатюк Є.В., Доля О.В., Лященко Т.А., Серпінська О.І. // – Київ: Управління розвитком складних систем, збірник наукових праць, випуск 43/2020, КНУБА, 2020.

16. Терентьєв О.О. Моделі визначення фізичного зношення конструктивних елементів будівлі для задач діагностики технічного стану / Баліна О.І., Шабала Є.Є.// – Київ: Управління розвитком складних систем, збірник наукових праць, випуск 26/2016, КНУБА, 2016. – С. 153-157.

17. Терентьєв О.О. Побудова діагностичних моделей основних конструкцій будівель /Шабала Є.Є., Баліна О.І., Доля О.В.// – Київ: Управління розвитком складних систем, збірник наукових праць, випуск 28/2016, КНУБА, 2016. – С. 155-159.

18. Терентьєв О.О., Григоровський П.Є., Русан І.В., Горбатюк Є.В. VII Міжнародна науково-технічна конференція “Нові технології в будівництві. BIM. Досвід та перспективи впровадження будівельних інформаційних технологій”. Програмно-технічний комплекс реалізації автоматизованої смстеми діагностики технічного стану будівельних конструкцій. Державне підприємство «Науково-дослідний інститут будівельного виробництва», 9-10 грудня 2019 р., м. Київ, ДП «НДІБВ».

19. Терентьєв О.О. Підвищення ефективності інформаційної системи комплексної безпеки захисту будівель на етапі проектування, будівництва та експлуатації / Київська К.І., Петроченко О.В.// - Київ: Нові технології в будівництві, міжвідомчий науково-технічний журнал, випуск 35/2019, НДІБВ, 2019. – С. 53–59.

20. Терентьєв О.О., Баліна О.І., Безклубенко І.С.. Розробка автоматизованої системи управління, моніторингу експлуатаційних витрат будівельного об’єкту. VIІ міжнародна науково-практична конференція «Управління розвитком технологій». Тема: Інформаційні технології розвитку змісту освіти, 25-26 березня 2020 р., м. Київ, КНУБА. – С. 95-96.

21. Терентьєв О.О., Горбатюк Є.В., Серпінська О.І.. Інтелектуальна інформаційна технологія системи діагностики технічного стану об`єктів будівництва. VIІ міжнародна науково-практична конференція «Управління розвитком технологій». Тема: Інформаційні технології розвитку змісту освіти, 25-26 березня 2020 р., м. Київ, КНУБА. – С. 97-98.

22. Gorbatyuk I.V., Terentyev O.O. Methodology of estimation of imperfection of building constructions on basis of fuzzy sets. The 14th International youth conference “Perspectives of science and education” (January 17, 2020). Slovo\word, New York, USA. 2020. Р. 122-128**.**

23. Kyivska K., Tsiutsiura M., Tsiutsiura S., Terentyev A. Methodology for building project portfolio. Abstracts of II International Scientific and Practical Conference Barcelona, Spain 24-25 February 2020. Р. 147-151.

24. Kyivska К., Tsiutsiura М., Tsiutsiura S., Terentyev А. Сomponents of information modeling of building objects. Abstracts of I International Scientific and Practical Conference Athens, Greece 29-31 March 2020. Р. 138-142.

Навчальне видання

СЕРПІНСЬКА Ольга Ігорівна,

ТЕРЕНТЬЄВ Олександр Олександрович,

БАЛІНА Олена Іванівна,

БЕЗКЛУБЕНКО Ірина Сергіївна

**МЕТОДИ ЕКСПЕРТНИХ ОЦІНОК У СИСТЕМАХ ПРИЙНЯТТЯ РІШЕНЬ**

Лабораторний практикум

Редагування та коректура

Комп’ютерне верстання

Ум. друк. арк. . Обл.-вид. акр. .

Електронний документ. Вид. №

Видавець і виготовлювач

Київський національний університет будівництва і архітектури

Проспект Повітряних Сил, 31, Київ, 03037

Свідоцтво про внесення до Державного реєстру суб’єктів

видавничої справи ДК № 808 від 13.02.2002