

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ СИСТЕМИ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів,
які навчаються за спеціальністю 151 «Автоматизація та комп'ютерно-
інтегровані технології»*

Київ
КПІ ім. Ігоря Сікорського
2018

Рецензенти: *Гордієнко Валентин Іванович*, д-р. техн. наук
Черняк Сергій Іванович, д-р. техн. наук

Відповідальний редактор *Колобродов Валентин Георгійович*, д-р. техн. наук, проф.

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол №7 від 29.03.2018 р.) за поданням Вченої ради приладобудівного факультету (протокол № 2/18 від 26.02.2018 р.)

Електронне мережне навчальне видання

Кравченко Ігор Володимирович

Микитенко Володимир Іванович, канд. техн. наук, доц.

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ СИСТЕМИ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ

Інформаційні технології: Системи комп'ютерної математики [Електронний ресурс] : навч. посіб. для студ. спеціальності «Автоматизація та комп'ютерно-інтегровані технології» / І. В. Кравченко, В. І. Микитенко; КПІ ім. Ігоря Сікорського . – Електронні текстові дані (1 файл: 5,57 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 243с.

Розглянуто історію створення, класифікацію, сучасний стан систем комп'ютерної математики (СКМ). Викладено відомості щодо технології розв'язання математичних та технічних задач в СКМ «MATHCAD», «MATLAB». Містить теоретичний матеріал, приклади та комплекс завдань для комп'ютерного практикуму та самостійної роботи. Розглянуто питання одиночних чисельних та символічних розрахунків, програмування алгоритмів користувача, роботи з матрицями, графічного виведення результатів. Може бути корисний студентам та фахівцям технічних спеціальностей для набуття навичок застосування інформаційних технологій у вигляді сучасних СКМ в учбовій та науково-технічній практиці.

© І. В. Кравченко, В. І. Микитенко, 2018

© КПІ ім. Ігоря Сікорського, 2018

ЗМІСТ

ПЕРЕДМОВА	7
1. СИСТЕМИ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ	9
1.1. Класифікація СКМ.....	11
1.2. Типова структура СКМ	12
1.3. Історія розвитку СКМ.....	13
1.4. Функції СКМ	16
1.5. Сучасні СКМ	19
1.6. Контрольні питання	24
2. СКМ MathCAD	26
2.1. Загальні характеристики	26
2.1.2. Контрольні питання	37
2.2. Робота з документами MathCAD.....	37
2.2.1. Створення та редагування документа.....	37
2.2.2. Робота з текстовою областю документа	39
2.2.3. Збереження документа MathCAD	40
2.2.4. Контрольні питання	40
2.3. Організація обчислень	41
2.3.1. Введення змінних, виразів. Виведення результатів	42
2.3.2. Режими обчислень	47
2.3.3. Вбудовані функції.....	49
2.3.4. Функції користувача.....	56
2.3.5. Розрахунки в діапазоні значень.....	58
2.3.6. Контрольні питання	59
2.4. Перевірка умов	60
2.4.1. Контрольні питання	64
2.5. Робота з векторами та матрицями	64
2.5.1. Контрольні питання	71

2.6. Програмування в MathCAD	71
2.6.1. Етапи застосування підпрограм-функцій	72
2.6.2. Різновиди алгоритмів в підпрограмах-функціях	75
2.6.3. Модульне програмування MATHCAD	83
2.6.4. Контрольні питання	85
2.7. Графічне оформлення результатів	85
2.7.1. Двовірні графіки	87
2.7.2. Тримірні графіки	94
2.7.3. Контрольні питання	107
2.8. Розрахунки з розмірностями	107
2.8.1. Контрольні питання	112
2.9. Символьні обчислення	113
2.9.1. Символьне обчислення виразів	115
2.9.2. Символьні тотожні перетворення виразів	119
2.9.3. Додаткові символьні обчислення	123
2.9.4. Контрольні питання	128
3. ПРАКТИКУМ ДО РОЗДІЛУ 2.	130
3.1. Робота з документами	130
3.2. Організація обчислень	133
3.3. Перевірка умов	138
3.4. Робота з масивами	140
3.5. Програмування	143
3.6. Двовірні графіки	144
3.7. Тримірні графіки	147
3.8. Розрахунки з розмірностями	152
3.9. Символьні обчислення	156
4. СКМ MATLAB	158
4.1. Загальні характеристики	158

4.1.1. Контрольні питання	164
4.2. Робота в режимі калькулятора.....	164
4.2.1. Змінні, вирази, вбудовані функції	164
4.2.2. Функції користувача	171
4.2.3. Формат виведення результатів	173
4.2.4. Масиви	176
4.2.4. Контрольні питання	182
4.3. Програмування в Matlab.....	183
4.3.1. Редактор М-файлів.....	183
4.3.2. Файли сценарію.....	186
4.3.3. Файл – функції.....	186
4.3.4. Керування обчислювальним процесом	189
4.3.5 Введення – виведення інформації	193
4.3.6. Контрольні питання	195
4.4. Двовірні графіки	196
4.4.1. Базові команди керування графікою.....	197
4.4.2. Оформлення графіків.....	200
4.4.3. Спеціальні графіки.....	204
4.4.4. Контрольні питання	211
4.5. Тримірні графіки	212
4.5.1. Графіки просторових кривих	212
4.5.2. Графіки просторових поверхонь	213
4.5.3. Оформлення тримірних графіків.....	214
4.5.4. Контрольні питання	217
4.6. Символьні обчислення	217
4.6.1. Символьні змінні та функції	217
4.6.2. Обчислення з символьними змінними.....	218
4.6.3. Контрольні питання	225

5. ПРАКТИКУМ ДО РОЗДІЛУ 4	227
5.1. Розрахунки в режимі калькулятора.....	227
5.2. Програмування	228
5.3. Двовірні графіки	231
5.4. Тримірні графіки	233
5.5. Символьні перетворення.....	234
Література	236
Додаток А. Основні функції Matlab	238

ПЕРЕДМОВА

Навчальний посібник призначено для забезпечення інформаційними та методичними матеріалами кредитного модуля "Інформаційні технології 2".

В посібнику викладено необхідні відомості щодо термінології, класифікації, розвитку систем комп'ютерної математики. Детально розглянуті можливості, особливості застосування, технології розв'язання математичних та технічних задач в системах комп'ютерної математики (СКМ) *MathCAD*, *Matlab*. Матеріал забезпечений практичними прикладами та комплексом завдань для самостійної роботи.

Системам комп'ютерної математики присвячено значна кількість джерел. Більшість є довідниками з описом реалізації чисельних методів математики в конкретних системах [2, 7, 8, 10, 16, 17]. Деякі з джерел мають технічну спрямованість [11, 17, 19]. Кількість навчальної літератури значно менше [1, 6, 13, 14,]. Серед вітчизняних видань [12, 13, 14, 15, 18] україномовними навчальними є джерела, які присвячені питанням моделювання механічних [13], динамічних систем [14, 15] та орієнтовані на студентів старших курсів.

Дисципліна "Інформаційні технології" викладається на першому курсі навчання та має обсяг в 18 годин лекцій та 36 годин комп'ютерного практикуму. Тому видання не торкається питань чисельних методів, методів оптимізації, обробки експериментальних даних. Увага приділена базовим можливостям для початку роботи з комп'ютерними системами математики студентами початкового курсу. Розглянуті питання одиночних чисельних та символічних розрахунків, програмування алгоритмів користувача, роботи з матрицями, графічного виведення результатів.

В розділі 1 навчального посібника розглядаються питання історії виникнення, класифікації, типової структури, можливостей систем комп'ютерної математики. Наводиться інформація про стан ринку систем комп'ютерної математики.

Розділ 2 навчального посібника охоплює питання, що стосуються засобів керування, можливостей, особливостей застосування, методики розв'язання математичних та технічних задач в СКМ *MathCAD* чисельно та аналітично, графічного відображення інформації.

Розділ 4 навчального посібника присвячено роботі в СКМ *Matlab* в режимі калькулятора та в програмному режимі.

Розділи 3, 5 містять практикуми з прикладами та завданнями для самостійної роботи в *MathCAD* та *Matlab*, відповідно.

Приклади та завдання в посібнику базуються на версії *MathCAD* 14, *Matlab* 7X. Види вікон діалогу та синтаксис команд в інших версіях пакетів можуть дещо відрізнятись від наведених у даному посібнику.

Мета навчального посібника - допомога студентам в самостійному вивченні відповідних розділів навчальної дисципліни, в набутті навичок застосування інформаційних технологій у вигляді сучасної СКМ в учбовій та науково-технічній практиці. Посібник може використовуватись для самостійної роботи та дистанційного навчання.

1. СИСТЕМИ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ

Комп'ютерна математика - це напрямок в математиці, що з'явився на перетині класичної математики та інформатики. Вона виникла наприкінці двадцятого століття, що пов'язано з успіхами впровадження персональних комп'ютерів (ПК) в практику вирішення математичних задач. Головним засобом комп'ютерної математики є *системи комп'ютерної математики* (СКМ). Вони дають можливість використовувати математичні методи без процедури програмування і тим самим створюють користувачу зручне середовище для роботи.

Комп'ютерна математика - сукупність теоретичних, методичних, алгоритмічних, апаратних і програмних засобів, які призначені для ефективного розв'язування за допомогою комп'ютерів широкого кола математичних задач з високим ступенем точності та продуктивності, а також виконувати складні ланцюги обчислювальних алгоритмів з широкими можливостями візуалізації всіх етапів обчислень [1, 2].

СКМ є програмними засобами, за допомогою яких можна автоматизувати виконання як чисельних, так і аналітичних (символьних) обчислень і розрахунків. У зарубіжній літературі застосовується термін - *Computer Mathematics Systems (CMS)*.

СКМ - поліфункціональні, універсальні програмні засоби, які призначені для ефективного виконання математичних операцій з даними як у символьній, так і в числовій формі, візуалізації математичних закономірностей, проведення навчальних і наукових досліджень, а також моделювання процесів і явищ у різних предметних галузях [1].

СКМ значно полегшує розв'язування типових математичних задач, таких як обчислення значень функцій і побудова їхніх графіків, розв'язування рівнянь, нерівностей і їх систем, обчислення інтегралів, знаходження похідних функцій, тощо.

Наразі, у користувача зменшується потреба в різноманітних довідниках і математичних таблицях, і разом з тим, з'являються можливості за короткий час розв'язувати значну кількість задач, готувати електронні книги.

Кожна із СКМ має певні особливості, які потрібно враховувати під час розв'язування конкретних математичних задач. Завдяки реалізації СКМ на ПК вони стають все більш доступними для використання.

Зараз системи комп'ютерної математики професійного призначення представлені в основному великими західними фірмами (*MathSoft*, *MathWorks*, *Waterloo Maple* тощо). СКМ стали потужними засобами діяльності як професійних математиків, так і тих, хто використовує математику для побудови й дослідження математичних моделей в різних предметних галузях. Їх використовують для розв'язування наукових, інженерних, навчальних задач, наочної візуалізації даних і результатів обчислень, як зручні та повні довідники з математичних обчислень. Завдяки потужній графіці, засобам візуального програмування й використання техніки мультимедіа роль СКМ виходить далеко за межі тільки математичних розрахунків.

Завдяки створенню СКМ професійні математики, а також ті, хто використовує математичні методи, одержали потужні засоби інтенсифікації діяльності. Їх використання дає змогу значною мірою підсилити інтелектуальну діяльність, можливість автоматизувати виконання не тільки чисельних, а й аналітичних (символьних) обчислень та графічних побудов. За допомогою СКМ можна виконувати такі види аналітичних обчислень, як знаходження границь функцій, похідних, обчислення невизначених та визначених інтегралів, розкладання функцій в ряди, розв'язування багатьох класів диференціальних рівнянь в аналітичному поданні, виконання різноманітних спрощень, перетворень, підстановок, тощо.

Однак, слід зауважити, що застосування комп'ютерної математики тими, хто не має достатніх знань у вирішенні математичних завдань, у край небажано, оскільки може привести до барвисто представлених, але невірних і, навіть, абсурдних результатів. З одного боку, комп'ютери полегшують процес розрахунків, з іншого - надмірне захоплення комп'ютерами, невміле їх застосування може стати джерелом втрати пізнавальних інтересів, лінощів мислення та інших небажаних наслідків [3].

1.1. Класифікація СКМ

На сьогодні не існує сталої класифікації СКМ. Різні автори пропонують свою класифікацію [1, 3, 4, 5].

В залежності від обраного критерію системи комп'ютерної математики умовно можна поділити:

По відношенню до форми отриманих результатів та застосованих математичних методів: системи для чисельних розрахунків, системи для аналітичних розрахунків – *системи комп'ютерної алгебри* (computer algebra system, CAS), комбіновані системи.

По відношенню до оплати за користування: безкоштовні системи, умовно-коштовні системи, комерційні системи. Більшість СКМ є досить дорогавартностними комерційними проектами.

По відношенню до типу задач, які розв'язує СКМ: математичні та технічні.

По відношенню до кількості виконуваних математичних дій: спеціалізовані та універсальні. Серед спеціалізованих розрізняють системи статистичних розрахунків, математичні системи для розв'язання нелінійних рівнянь, розв'язання систем диференціальних рівнянь, обчислення поліномів, аналітичної геометрії, задач теорії груп, програмні засоби візуалізації математичних даних.

По відношенню до організації даних: табличні процесори, матричні системи, універсальні системи.

Кожен з типів має власні специфічні властивості, які необхідно враховувати при розв'язанні конкретних математичних задач.

Ресурс Вікіпедія (https://en.wikipedia.org/wiki/Mathematical_software) дає наступну класифікацію математичного програмного забезпечення:

- Калькулятори – застосунки, що реалізують прості арифметичні дії, як додавання, множення, піднесення до ступеню, тригонометричні обчислення.
- Системи чисельної комп'ютерної математики.
- CAS – системи символної комп'ютерної математики.

- Засоби статистичного аналізу.
- Засоби оптимізації.
- Засоби геометрії
- Інтернет засоби
- Математичні бібліотеки.

1.2. Типова структура СКМ

Сучасні СКМ представлені розробками різних виробників, у кожній СКМ можуть бути нюанси в архітектурі. Утім, всі сучасні універсальні СКМ мають типову структуру [3] (рис. 1.1).

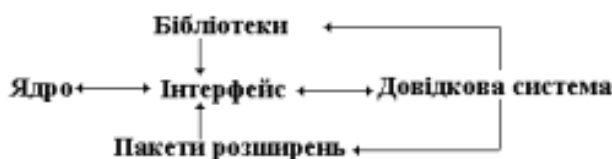


Рис. 1.1 Типова структура універсальних СКМ

Інтерфейс СКМ зазвичай базується на операційній системі Windows та є типовим для Windows застосунків. Інтерфейс забезпечує зручність звернень до ядра із запитом дії та отримання результатів на екрані в звичній для користувача формі.

Основу СКМ складає набір базових функцій та алгоритмів, так званих вбудованих функцій, які створюють *ядро системи*. Ядро СКМ реалізує алгоритм функціонування СКМ, забезпечує сумісне функціонування всіх її частин, організує приймання даних з інтерфейсу, обробку запиту користувача, виклик потрібних процедур.

Ядро (kernel) системи містить коди функцій, які виконуються максимально швидко. Кількість та швидкість виконання вбудованих функцій забезпечує потужність СКМ. Наприклад, одна з найбільш потужних СКМ *Mathematica* містить інформацію про більше, ніж 5000 інтегралів, а для обчислення використовує тільки кілька десятків функцій ядра.

Вбудовані в ядро функції виконуються швидко, якщо їхня кількість обмежена. Більш складні та не так часто виконувані функції реалізуються засобами зовні ядра - *бібліотеками*. Загальна кількість функцій СКМ може досягати кількох тисяч (рис.1.2).

Кардинальне розширення можливостей СКМ та їхня адаптація до конкретних задач досягається за рахунок *пакетів розширення*. Ці пакети можуть бути написані на мовах програмування або на власній мові СКМ.

Ядро, бібліотеки, пакети розширення СКМ акумулюють знання в царинах математики, інформатики, прикладних наук. Тому СКМ відносять до *інтелектуальних програмних продуктів*, одним з призначень яких є надання користувачу знань з чисельних методів, аналітичної математики та комп'ютерної графіки [3].

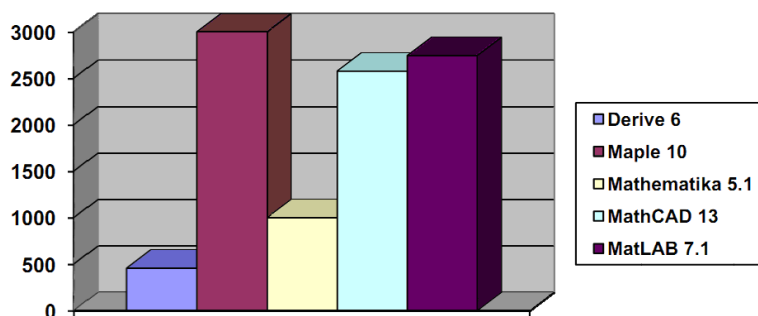


Рис. 1.2. Обсяг функцій СКМ [2]

1.3. Історія розвитку СКМ

Практичного вжитку математичне програмне забезпечення набуло в середині ХХ століття. Саме тоді з'явилися перші розробки СКМ, які використовуються сьогодні. Реалізовувалися вони лише на великих комп'ютерах, являли собою системи колективного використання, застосування яких, навіть, для простих задач потребувало участі багатьох висококваліфікованих фахівців, обізнаних в математиці. Доступні такі системи були тільки вузькому колу користувачів.

На першому етапі комп'ютер виправдовував свою назву (англ. "обчислювач") та працював як потужний програмований калькулятор, здатний швидко та автоматично за введеним алгоритмом виконувати складні та громіздкі арифметичні та логічні операції над числами.

Такий результат часто не вдовольняв ані математиків, ані інженерів. Більшість результатів математичних обчислень в класичній математиці традиційно записується у вигляді виразів з застосуванням спеціальних сталих: π , e , іраціональних значень – за допомогою радикала.

Знадобився універсальний математичний інструмент, розрахований на широке коло користувачів, які не є професіоналами з математики та програмування.

З'явилась потреба заставити комп'ютер виконувати перетворення традиційними для математики способами (дрібно-раціональні перетворення, підстановки, спрощення, розв'язання рівнянь, диференціювання тощо), а результат отримувати не у вигляді числа, а в вигляді формул. Такі перетворення називають *перетвореннями в символному вигляді або аналітичними перетвореннями*.

Це призвело до буму досліджень в області комп'ютерної математики. Саме в провідних університетах світу були розроблені прообрази сьгоднішніх СКМ та СКА.

Вікіпедія вважає першими успішним проектом СКМ програму *Schoonschip* для символних обчислень в галузі фізики, яка розроблена Мартіусом Велтманом в 1963 в ЦЕРНі, та програму *MATHLAB*, яка розроблена в 1964 році Карлом Енгельманом в рамках проекту MITRE.

Під керівництвом академіка В.М. Глушкова на початку 60-х років ХХ століття у Києві були створені передвісники майбутніх персональних комп'ютерів – інженерні малі комп'ютери серії «Мир» з апаратною реалізацією мов програмування високого рівня та унікальними можливостями виконання чисельних та аналітичних розрахунків. Розроблені мови високого рівня «МИР» і «АНАЛІТИК» давали змогу одержувати ефективну реалізацію роботи з дійсними числами довільної розрядності, цілими числами необмеженої розрядності, точних операцій над дробовими раціональними числами тощо. Система «АНАЛІТИК» була однією з перших систем комп'ютерної алгебри, а в мові «АНАЛІТИК» вперше була використана техніка запису алгебраїчних виразів (застосування співвідношень), яка в наш час є основою технології декларативного програмування. На жаль, ці роботи В.М. Глушкова не були підтримані й згодом ініціатива в цій галузі перейшла до західних країн [3].

До першого покоління СКА відносять:

MACSYMA. Розробник Джоел Мозес, Масачусетський технологічний інститут на замовлення Департаменту енергетики США. Сьогодні не

виробляється. Потомок – безкоштовна СКА MAXIMA,
<http://maxima.sourceforge.net/>

MATHLAB. Сьогодні не виробляється.

SCRATCHPAD. Розробник Ричард Димик з Thomas J. Watson Research Center по замовленню ІВМ. Сьогодні не виробляється Потомок – безкоштовна система *Axiom*.

REDUCE. Розробник Тоні Хірн Безкоштовна система,
<http://www.reduce-algebra.com/>

Комп'ютерні математичні системи як клас спеціалізованих програмних засобів індивідуального використання виникли лише в 70 - 80- ті роки ХХ століття завдяки розвитку індустрії персональних комп'ютерів (ПК). Вони утворили друге покоління СКМ. Ознакою систем другого покоління є графічний інтерфейс, орієнтований на користувача.

До систем другого покоління відносяться:

Maple. Розробники Кейт Геддес, Гастон Гоннет з University of Waterloo, фірма Maplesoft, <http://www.maplesoft.com/products/maple/>

Mathematica. Розробник Стівен Вольфрам, фірма Wolfram Research, <http://www.wolfram.com/mathematica>

SageMath – потомок СКА *Sage* "System for Algebra and Geometry Experimentation", безкоштовна, <http://www.sagemath.org/>

DERIVE – потомок системи *muMATH* фірми Soft Warehouse, підрозділу Texas Instruments. Сьогодні не виробляється.

MATLAB «matrix laboratory». СКМ для чисельних розрахунків. Розробник Клів Моулер з університету Нью-Мехіко, фірма Mathworks. Зараз є універсальною СКА-СКМ, <http://www.mathworks.com/>.

MATHCAD. Розробник Аллен Раздов з Масачусетського технологічного інституту, фірма Mathsoft, <http://www.ptc.com/products/mathcad/>

До третього покоління відносять:

MUPAD. Розробник Бенно Фукштейнер з університету Падерборн, фірма SciFace Software GmbH & Co. Використовувалася самостійно та в

складі СКМ *SciLAB*, *MathCAD*. Сьогодні не виробляється, є частиною СКМ *MATLAB*.

Scilab - випускається INRIA та École nationale des ponts et chaussées (ENPC), Scilab Enterprises, безкоштовна, <http://www.scilab.org/>

Сьогодні СКМ активно використовуються у навчальному процесі у всьому світі. Більше тисячі університетів з 61 країни є офіційними користувачами системи *Mathematica*. Серед них такі освітні заклади: Пекінський, Кембриджський, Колумбійський, Гарвардський, Стенфордський, Московський державний, Австралійський національний, Каліфорнійський, Оксфордський університети, Лондонська школа економіки та політичних наук і багато інших [3].

На вітчизняному та світовому ринку СКМ представлені лише закордонними виробами. Це пов'язано з необхідністю великих інтелектуальних, трудових та фінансових витрат. Пік розробок СКМ припав на 90-ті роки ХХ століття, коли країни бувшого СРСР переживали глибу кризи.

1.4. Функції СКМ

До можливостей чисельних розрахунків відносяться:

- арифметичні та алгебраїчні оператори та функції;
- функції для роботи з комплексними числами;
- тригонометричні та гіперболічні функції;
- зворотні тригонометричні гіперболічні функції;
- логічні оператори та функції;
- векторні та матричні оператори та функції;
- розв'язання систем лінійних алгебраїчних рівнянь;
- спеціальні математичні функції;
- арифметика ступеневих поліномів;
- знаходження комплексних коренів поліномів;
- розв'язання систем нелінійних алгебраїчних рівнянь;
- розв'язання систем диференціальних рівнянь;

- оптимізація функцій;
- лінійне програмування;
- одномірна та багатомірна інтерполяція;
- операції над текстом;
- обчислення трикутників;
- визначення площин в тримірному просторі;
- визначення кривих другого порядку.

Можливості СКМ в частині *аналітичних перетворень* суттєво відрізняються. Серед типових дій можна виділити

- поєднання символічних та чисельних обчислень;
- операції точної арифметики;
- обчислення з довільною точністю;
- спрощення математичних виразів;
- проведення підстановки одних виразів в інші;
- розкриття добутків в математичних виразів;
- часткова та повна факторизація (розкладання на множники);
- пошук спільних множників та дільників;
- операції з ступеневими багаточленами;
- обчислення сум та добутків рядів;
- диференціювання виразів в часткових та повних похідних;
- знаходження визначених та невизначених інтегралів;
- обчислення границі функцій;
- розкладання функцій в ряди Тейлора, Маклорена;
- пошук екстремумів функцій;
- інтегральні перетворення Лапласа, Фур'є, Z перетворення;
- розв'язання задач теорії графов, нейронних сіток;
- обчислення елементарних та спеціальних математичних функцій:

- розв'язок рівнянь та систем рівнянь;
- розв'язання диференціальних рівнянь та систем диференціальних рівнянь;
- розв'язок задач лінійного та нелінійного програмування;
- операції з векторами та матрицями;

Графічна візуалізація:

- графіки функцій в декартовій, полярній системі координат;
- графіки функцій, які визначені в параметричній формі;
- графіки тримірних поверхонь;
- графіки контурні, векторні, діаграми;
- анімовані графіки;
- редагування графіки.

Програмування:

- вбудована мова програмування;
- простий синтаксис мови програмування;
- великий набір типів даних та вбудованих функцій;
- типи даних користувача;
- засоби роботи з файлами та потоками інформації;
- засоби налагодження програм.

Мова програмування СКМ є інтерпретуючою та інтерактивною (діалоговою). Це означає, що конструкція мови розпізнається СКМ та, при відсутності в ній помилок, негайно виконується. За наявності помилок генерується діагностичне повідомлення. В мовах програмування СКМ наявні засоби структурного, іноді об'єктного та візуально-орієнтованого програмування.

Також СКМ розв'язують основні задачі аналітичної геометрії, теорії імовірності та математичної статистики тощо.

Деякі системи мають додаткові можливості прикладної математики для таких царин, як фізика, біоінформатика, хімія, інженерія, обробка сигналів та зображень.

Можна визначити такі напрямки розвитку сучасних СКМ [4]:

- перетворення СКМ в інтелектуальні системи подання знань і їх експертної оцінки;
- інтеграція систем одна з одною і деякими офісними і графічними програмами;
- розширення можливості обчислень, що охоплюють практично всі галузі застосувань математики;
- розширення засобів візуалізації обчислень;
- перетворення СКМ в універсальні системи;
- упровадження нових функцій, наприклад, для реалізації нечіткої логіки, нейронних мереж тощо;
- упровадження в СКМ засобів для створення електронних підручників у різних форматах;
- створення документів з текстами, формульними виразами, рисунками і графіками найвищої поліграфічної якості.

1.5. Сучасні СКМ

Комерційні СКА

Спеціалізовані комерційні СКА орієнтовані на розв'язання проблем однієї галузі математики, наприклад:

- *Fermat* - вирішує задачі лінійної алгебри з поліномами. Розроблена Robert H. Lewis, <http://home.bway.net/lewis/>. Вартість \$70.
- *Magma* - вирішує задачі теорії груп. Розроблена в University of Sydney, <http://magma.maths.usyd.edu.au/>. Вартість \$2275.

Універсальні комерційні СКА забезпечують виконання великої кількості математичних дій з різних галузей математики:

- *Mathematica*. Розроблена Stephen Wolfram в 1988 році. Випускається фірмою Wolfram Research of Champaign, Illinois, <http://www.wolfram.com/mathematica>. Вартість \$2,495 - Professional, \$140 - Student. Вважається найбільш потужною СКА.

- *Maple*. Розроблена Symbolic Computation Group з University of Waterloo в 1984 році, випускається фірмою Waterloo Maplesoft, <http://www.maplesoft.com>. Вартість \$2,275- Commercial, \$99 –Student.
- *Matlab (Matrix Laboratory)*. Найбільш розповсюджена в інженерних колах СКМ завдяки спеціалізованим пакетам розширення. Відноситься до універсальних СКА завдяки придбанню прав на СКА *MuPad* та включення її в склад *Matlab*. Виробляється фірмою MathWorks Inc, <http://mathworks.com>. Вартість \$2350- Standard, \$55- Student.
- *Mathcad* – Одна з небагатьох, що орієнтована на реалізацію принципу WYSIWYG (What You See Is What You Get) текстових процесорів на кшталт Microsoft Word. Випускалась фірмою Mathsoft. З 2013 року придбана Parametric Technology Corporation , <http://www.ptc.com>. Вартість \$650 - Commercial, \$105 - Student, безкоштовно - Express Edition. До універсальних СКА відноситься завдяки придбанню ліцензій на використання символічних процесорів СКА *Maple* - в минулих версіях та - *MuPad* – сьогодні. Має значно менші можливості символічних перетворень.
- *LiveMath* – найпростіша з комерційних СКА. Розроблена Carl Spitznagel з John Carroll University. Випускалася фірмою розробником під назвами Theorist, MathView, and MathPlus. Відрізняється зручним графічним інтерфейсом. Забезпечує можливість покрокових дій. В оформленні орієнтована на СКА MathCAD. <https://www.livemath.com/>. Вартість 400\$. Має засоби інтернет розробки інтернет за стосунків LiveMath Maker - \$249.

Безкоштовні СКА

Спеціалізовані безкоштовні СКА:

- *Cadabra*. Розроблена Kasper Peeters в 2007 році для задач тензорної теорії поля, <http://cadabra.science/index.html>.
- *FORM*. Розроблена в 1989 році J.A.M. Vermaseren з Dutch institute for subatomic physics для задач квантової фізики, <http://www.nikhef.nl/~form/>.
- *GAP (Groups, Algorithms and Programming)*. Розроблена Lehrstuhl D für Mathematik, RWTH Aachen, Germany в 1986 році для задач теорії груп та комбінаторики. Розповсюджується GAP Group, <http://www.gap-system.org>.

- *Macaulay2*. Розроблена Daniel Grayson в 1994 році для задач лінійної алгебри, <http://www.math.uiuc.edu/Macaulay2>.

Універсальні безкоштовні СКА:

- *Axiom*. Розроблена Richard Jenks на заміну СКМ *Scratchpad*, <http://www.axiom-developer.org>.
- *FriCAS*. Розроблена Waldek Hebisch в 2007 році, <http://fricas.sourceforge.net/>
- *Maxima*. Розроблена в MIT, <http://maxima.sourceforge.net/>
- *Reduce*. Розроблена Anthony C. Hearn в 1968 році, <http://www.reduce-algebra.com>.
- *SageMath* (Software for Algebra and Geometry Experimentation). Розроблена в 2005 році William Stein з University of Washington, <http://www.sagemath.org>. Є Інтернет орієнтованою СКА.
- *Mathics*. Розроблена в 2011 році Jan Pöschko за подобою до СКА *Mathematica* на базі бібліотеки SymPy, <https://mathics.github.io>. Є Інтернет орієнтованою СКА.

СКМ для чисельних розрахунків

СКМ для чисельних розрахунків історично з'явилися раніше СКА та не мають можливостей для символічних перетворень.

Спеціалізовані комерційні СКМ для чисельних розрахунків:

- *TK Solver*. Розроблена в 1982 році Milos Konopasek з фірми Software Arts як засіб розв'язання систем нелінійних рівнянь. Сьогодні підтримується фірмою Universal Technical Systems, Inc, <https://www.uts.com>. Вартість 599\$.

Комерційні універсальні СКМ для чисельних розрахунків:

- *DADiSP (Data Analysis and Display)*. Розроблена в 1987 році фірмою DSP Development Corporation, <http://www.dadisp.com>. Вартість \$1995 - Commercial, \$129 - Academic.
- *IGOR Pro*. Розроблена фірмою WaveMetrics Inc., <http://www.wavemetrics.com>. Орієнтовна на обробку даних та зображень. Вартість \$995 - Commercial, \$85 - Academic.

- *MLAB*. Розроблена фірмою Civilized Software, Inc. <http://www.civilized.com/mlabdesc.html>. Вартість \$2250 - Standard, \$50 – Student.
- *Origin*. Розроблена фірмою OriginLab, <http://originlab.com>. Орієнтовна на обробку даних та зображень. Вартість \$1800 - Standard, \$550 – Student.

Безкоштовні універсальні СКМ для чисельних розрахунків:

- *Euler Math Toolbox*. Розроблена René Grothmann з Catholic University of Eichstätt-Ingolstadt, Germany в 1988 році. Має інтерфейс для роботи спільно з безкоштовною СКА *Maxima*, <http://www.euler-math-toolbox.de>.
- *Scilab*. Розроблена Scilab Enterprises в 1990 як альтернатива СКМ *Matlab*. Поставляється з пакетом *Xcos*, який є подобою пакета *Simulink*, <http://www.scilab.org/>
- *SMath Studio*. Розроблена Андрієм Івашовим в 2005 році як безкоштовна альтернатива СКМ *MathCAD* за принципом WYSIWYG, <http://en.smith.info>.
- *GNU Octave*. Середовище чисельного моделювання з мовою програмування високого рівня. Розроблено на зразок СКМ *MATLAB*, <https://www.gnu.org/software/octave>.
- *RlabPlus*. Розроблено на зразок СКМ *MATLAB*, <http://rlabplus.sourceforge.net/>

Спеціалізовані безкоштовні СКМ для чисельних розрахунків:

- *ADMB (AD Model Builder)*. Розроблена D. Fournier для задач, що потребують диференційного обчислення, <http://admb-project.org>.
- *Dynamics Solver*. Розроблена Juan M. Aguirregabiria для розв'язання систем диференціальних рівнянь, <http://tp.lc.ehu.es/jma/ds/ds.html> .

Широкого вжитку набули спеціалізовані математичні програми наукової графіки для візуалізації та обробки наукових, технічних та інших даних. Асортимент застосунків цього типу дуже широкий: від найпростіших, що виконують тільки дії з візуалізації функцій, до складних систем, які дозволяють проводити обробку та аналіз даних.

Програми побудови та аналізу графіків

Безкоштовні: *MathPlot* (<http://rigaux.org>), *GeoGebra* (<http://geogebra.org>), *XGraph* (<http://www.isi.edu/nsnam/xgraph>), *Kig* (<http://edu.kde.org/kig>), *kst* (<http://kst-plot.kde.org>), *Simple Grapher* (<http://www.simplegrapher.com>), *OpenPlaG* (<http://rechneronline.de/openPlaG>).

Комерційні: *Fityk* (<http://fityk.nieto.pl>), вартість \$115, *Sysquake* (<http://www.calerga.com>), вартість \$2500, *MagicPlot* (<http://www.magicplot.com>), вартість \$149. *Advanced Grapher - Graphing Software* (<http://www.alentum.com/agrapher>).

З українських продуктів можна відмітити програми *GRAN* [5] розробки Національного педагогічного університету України (<http://www.zhaldak.npu.edu.ua/prohramnyi-zasib-gran>). За можливостями вони є аналогами найпростіших систем типу *GraphAnder* (<http://andron-st.narod.ru>) чи студентської розробки Субботіна Віталія *FlatGraph* (<http://old.exponenta.ru/soft/others/flatgraph/fg.asp>).

Системи аналізу та візуалізації даних

Безкоштовні:

- *DataMelt*. Розроблена Сергієм Чекановим, <http://jwork.org/dmelt>.
- *Dataplot*. Розроблена National Institute of Standards and Technology, <http://www.itl.nist.gov/div898/software/dataplot/index.htm>.
- *SciDAVis* (Scientific Data Analysis and Visualization). Безкоштовна альтернатива *Origin*. <http://scidavis.sourceforge.net>.

Комерційні:

- *Algebrator*. Вартість \$58.99, <http://www.softmath.com>.
- *MaxStat Pro*. Розроблена фірмою MaxStat Software для обробки наукових даних, <http://www.maxstat.de>.
- *Origin*. Розроблена фірмою OriginLab Corporation для обробки та візуалізації наукових даних, <https://www.originlab.com>. Вартість \$1800.
- *Sigma Plot*. Розроблена фірмою SYSTAT Software, <http://www.sigmaplot.co.uk>. Вартість \$525.

Слід окремо відмітити *СКМ для статистичних розрахунків*. Головна відмінність таких програм – велика кількість вбудованих спеціальних

функцій, які дозволяють без програмування виконувати багатоваріантні статистичні розрахунки, засоби для автоматичної фільтрації помилкових даних.

Універсальні комерційні статистичні СКМ:

- *SPSS Statistics (Statistical Package for the Social Sciences)*. Придбана фірмою IBM за 1,2 млрд. дол. США. Вважається провідною статистичною програмою в світі, <http://www-01.ibm.com/software/analytics/spss>. Вартість \$99 в місяць за одне робоче місце.
- *Statgraphics*. Розроблена в Princeton university, виробляється фірмою Statpoint Technologies Incorporated, <http://statgraphics.com>.
- *Statistica*. Розроблена фірмою Statsoft - Dell Software, <http://statistica.io>.
- *StatPlus*. Розроблена фірмою AnalystSoft, <http://www.analystsoft.com>

Спеціалізовані комерційні статистичні СКМ:

- *EViews*. Розроблена фірмою HIS (<http://www.eviews.com>) для економічних задач.
- *GenStat*. Розроблена фірмою VSN International (<http://www.vsnl.co.uk/software/Genstat>) для сількогосподарських задач.
- *MedCalc*. Розроблена фірмою MedCalc Software bvba (<https://www.medcalc.org>) для медичних задач.

Універсальні безкоштовні статистичні СКМ:

- *PSPP*. Розроблена Ben Pfaff як безкоштовний аналог СКМ IBM SPSS Statistics, <https://www.gnu.org/software/pspp>.
- *ADaMSoft*. Розроблена Marco Scarno для UNESCO проста статистична програма, <http://adamsoft.sourceforge.net>.

1.6. Контрольні питання

Що таке комп'ютерна математика?

Що таке СКМ?

За якими критеріями класифікують СКМ?

Що відомо про вітчизняні СКМ?

В чому різниця чисельних та символічних обчислень в СКМ?

Які СКМ вважаються найкращими?

З яких типових елементів складаються універсальні СКМ?

Які типові функції чисельних розрахунків СКМ?

Які типові функції аналітичних перетворень СКМ?

До якого типу мов програмування відносять мови СКМ?

В чому полягає особливість статистичних СКМ?

2. СКМ MathCAD

2.1. Загальні характеристики

Система *MathCAD* з'явилася наприкінці 1980-х років для чисельного розв'язання математичних задач. Розроблена Алленом Раздовом (Allen Razdow) з Масачусетського технологічного інституту. Вироблялася фірмою Mathsoft.

Визначальною відмінністю системи є її орієнтація на принцип *WYSIWYG* (*What You See Is What You Get* — «що бачиш, те й отримуєш»). *Mathcad* первісно створювався для чисельного розв'язання математичних задач, він орієнтований на вирішення завдань саме прикладної, а не теоретичної математики, коли потрібно отримати результат без поглиблення в математичну суть завдання.

Окрім проведення обчислень, *MathCAD* дозволяє підготувати якісні тексти з наочним графічним наданням результатів.

MathCAD відображає на екрані формули в їхньому звичному вигляді так, як вони надаються в друкованому вигляді в книгах. Наприклад, знаходження кореня квадратного рівняння в *Excel* записується як:

$$(-B1 + \text{КОРЕНЬ}(B1 * B1 - 4 * A1 * C1)) / (2 * A1)$$

В *MathCAD* те ж саме рівняння виглядає значно звичайніше:

$$x_1 := \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

З 1994 року в систему було інтегровано ліцензований символічний процесор СКА *Maple* на той час фірми *Waterloo Maple*, що перевело *MathCAD* в універсальну математичну систему класу СКА. Остання версія фірми - *MathCAD 15*.

В 2006 році систему та торговельну марку *Mathcad* придбала фірма *PTC* (www.pts.com), яка спеціалізується в царині CAD програм з основним продуктом *Creo*. Тоді ж змінили колектив розробників та постачальника символічного ядра. Ним стала фірма *SciFace Software* з СКА *MuPad*. Символьне ядро *Mathcad*, на відміну від оригінального *Maple* (*MuPAD*), має обмежений функціонал.

Примітка. В деяких версіях до 13 є можливість використовувати майже повний функціонал ядра Maple (так звані «недокументовані можливості»).

Новий продукт набув назву *MathCAD Prime* (\$600 – commercial, \$500 – academic, \$56 - students), при цьому сумісність з “класичним” *MathCAD* розробник не гарантує.

MathCAD є інтегрованою системою, яка орієнтована на проведення математичних та інженерно-технічних розрахунків. До достоїнств пакета відносяться також можливості збереження документів в форматі Web-сторінок та *Microsoft Word*. Документ *MathCAD* одночасно є лістингом програми, результатом виконання цієї програми та звітом, який може бути роздрукований на принтері чи опублікований в Web.

Пакет дозволяє:

- робити чисельні обчислення з точністю 15 значущих цифр у двійковій, восьмеричній, шестнадцятирічній і десятковій системах із цілими, у десятковій – з дійсними й комплексними числами, похідних і інтегралів, рядів сум і добутків, математичних функцій, включаючи спеціальні (Беселя, Фур'є, Лапласа і т.д.);
- розв'язувати рівняння, системи рівнянь і нерівностей;
- проводити операції з векторами й матрицями;
- проводити фізичні обчислення з перевіркою розмірності;
- у будь-якому місці документа розміщувати й редагувати текстові фрагменти;
- виводити результати на екран, у дисковий файл;
- відображати результати в графічному вигляді на плоских і поверхневих графіках, графіках у вигляді контурних ізоліній і карт змінної щільності;
- реалізувати символні (аналітичні) перетворення;
- зберігати результати у форматах обміну з офісними пакетами;
- програмувати алгоритми обчислень.

Ядро MathCAD складають власні обчислювальний та графічний процесори, ліцензійний символний процесор (рис. 2.1). Власний обчислювальний процесор *MathCAD* здійснює чисельні обчислення виразів,

рядів, сум, добутків, визначених інтегралів та похідних, розв'язання рівнянь та систем рівнянь. *Символьний процесор* дозволяє отримати результати в аналітичному вигляді. В версіях після 12 використовується процесор *MuPad*, в більш давніх – *Maple*. *Графічний процесор* слугує для побудови графіків функцій однієї та двох змінних.

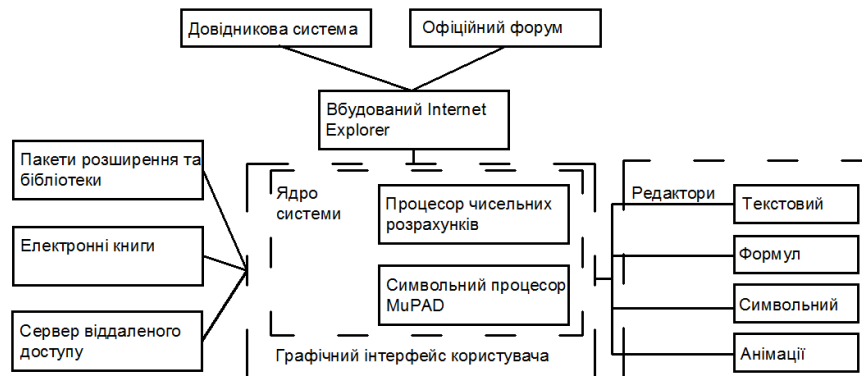


Рис. 2.1 Структура СКМ MathCAD

Текстовий редактор слугує для введення та редагування текстів. За можливостями та принципами керування текстовий редактор схожий на редактор *Microsoft WordPad*.

За допомогою *редактора формул* проводиться введення математичних виразів та підпрограм користувача.

Символьний редактор забезпечує користувача засобами підготовки даних для проведення аналітичних перетворень символьним процесором.

Редактор анімації точніше слід назвати засобом підготовки даних для анімації графіків.

В систему вбудовано браузер *Internet Explorer*, за допомогою якого реалізовано доступ до довідникової підсистеми пакета та форуму користувачів на сайті виробника.

Елементами довідникової системи *MathCAD* є: *QuickSheets* (Заготовки), *Reference Table* (Довідникові таблиці), *E-books* (Електронні книги).

QuickSheets - це *MathCAD* документи, які вміщують готові приклади розв'язання широкого кола математичних та науково-технічних задач (рис. 2.2).

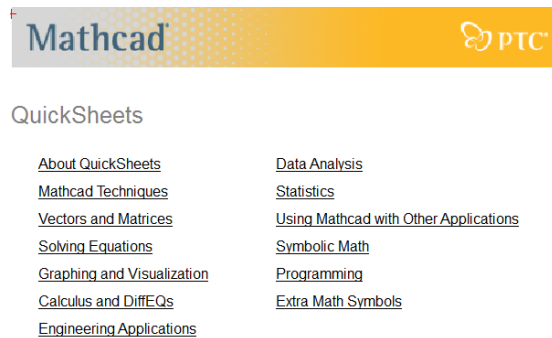


Рис. 2.2. Зміст розділу QuickSheets

Reference Table – розділ, який вміщує довідники з даними про хімічні та фізичні властивості, математичні та технічні константи, математичні таблиці інтегралів та похідних, формули з механіки, електротехніки, радіотехніки тощо (рис. 2.3).



Рис. 2.3. Вікно довідникових таблиць

Електронні книги - документи *MathCAD* з описом роботи для розв'язання спеціальних завдань за окремими темами та довідниковою інформацією з гіперпосиланнями та інтерактивними прикладами розрахунків. Сторінки електронних книг є повноцінними файлами з діючими розрахунками з можливістю змінювати вихідні дані. На додаток до звичайних документів *MathCAD*, книги мають зміст, предметний показник та систему гіперпосилань. *Електронні книги* є версіезалежні. Для кожної версії системи застосовуються свої книги.

Виробник *Mathcad* пропонує багато бібліотек, «електронних книг» (*e-books*) з описами методів розв'язання специфічних задач, діючими алгоритмами та готовими до використання документами системи:

- *Пакет аналізу даних (Data Analysis Extension Pack)* для версій до MathCAD Prime.
- *Пакет обробки сигналів (Signal Processing Extension Pack)* для версій до MathCAD Prime. Містить біля 70 функцій для аналогової та цифрової обробки сигналів, аналізу та графічного представлення результатів.
- *Пакет обробки зображень (Image Processing Extension Pack)* для версій до MathCAD Prime.
- *Пакет для роботи з вейвлетами (Wavelets Extension Pack)* для версій до MathCAD Prime. Містить біля 60-ти функцій ортогональних та біортогональних вейвлетів Хаара, Добеши, симлетів, койфлетів тощо.
- *Бібліотека будівництва (Civil Engineering Library, \$200)* для MathCAD Prime, що містить біля 300 шаблонів розрахунків для будівельного проектування та приклади теплових розрахунків.
- *Електротехнічна бібліотека (Electrical Engineering Library, \$200)* для MathCAD Prime, яка містить біля 500 обчислювальних процедур, формул, довідникові таблиці.
- *Бібліотека машинобудування (Mechanical Engineering Library, \$200)* для MathCAD Prime— містить біля 300 шаблонів розрахунків,
- *Довідник Роарка (Roark's Formulas for Stress and Strain, \$425)* для MathCAD Prime для розрахунку напружень і деформацій, моментів інерції, торсійного навантаження.
- *Бібліотека PTC том 1 (Mathcad Prime Worksheet Library - Volume 1, \$520.00, тільки для MathCAD Prime)* містить шаблони з алгебри, геометрії, статистики, аналізу даних, хімії, електротехніки, механіки, FEA.
- *Бібліотека PTC том 2 (Mathcad Worksheet Library - Volume 2, тільки для MathCAD Prime, \$520.00)* містить біля 700 шаблонів з дев'яти напрямів: механіка, статистика, фізика, оптимізація, фізика напівпровідників, обробка сигналів, аналіз даних.
- *Бібліотека прикладної математики (Mathcad Worksheet Library - Applied Math, тільки для MathCAD Prime, \$200)* містить біля 400 шаблонів з диференційних рівнянь, статистики, оптимізації, аналізу даних.

В мережі існують також електронні книги сторонніх виробників. Наприклад, *Visual Electromagnetics for Mathcad* (видавництво McGraw-Hill), *Mathcad E-Book on Wireless Channels*.

Mathcad має можливість запуску документів без встановлення самого *Mathcad* на комп'ютері користувача з використанням «хмарної» технології. Для цього документи *Mathcad* розташовують на мережному сервері віддаленого доступу з спеціальним програмним забезпеченням.

В 2003 році був запущений перший *Mathcad Application Server* для файлів «mcd»(MAS). В 2006 було додано *Mathcad Calculation Server* для файлів «xmsd» (MA/CS). Сервер дозволяє звертатися до нього для «online» розрахунків дистанційно через Інтернет браузер. При цьому користувачеві здається, що на його комп'ютері відкрито документ в *Mathcad*, в якому можна змінити дані, провести розрахунки та роздрукувати відповідь.

3 березня 2016р. PTC анонсувала *Mathcad Gateway* на базі *Mathcad Prime 3.1* (<https://www.ptc.com/en/mathcad-software-blog/announcing-ptc-mathcad-gateway>). Користувачі можуть купити підписку на цей розрахунковий сервер та відсилати на нього свої файли, щоб проводити розрахунки без встановлення самого пакета *Mathcad*.

Методика розрахунків та проміжні дані можуть бути відкриті або частково чи повністю закриті (рис. 2.4).

Технологія MA/CS надає такі переваги [9]:

- Економія ресурсів. Немає необхідності купляти математичне забезпечення для всіх комп'ютерів корпорації, ставити всім співробітникам або у всіх комп'ютерних класах *Mathcad* для ведення рутинних розрахунків. Досить поставити її тільки тим, хто створює розрахунки, решта можуть вести розрахунки через корпоративний або університетський MA / CS.
- Мобільність. Нові розрахункові методики стають миттєво доступні всім авторизованим користувачам. Досить тільки повідомити їм відповідні інтернет-адреси сторінок.

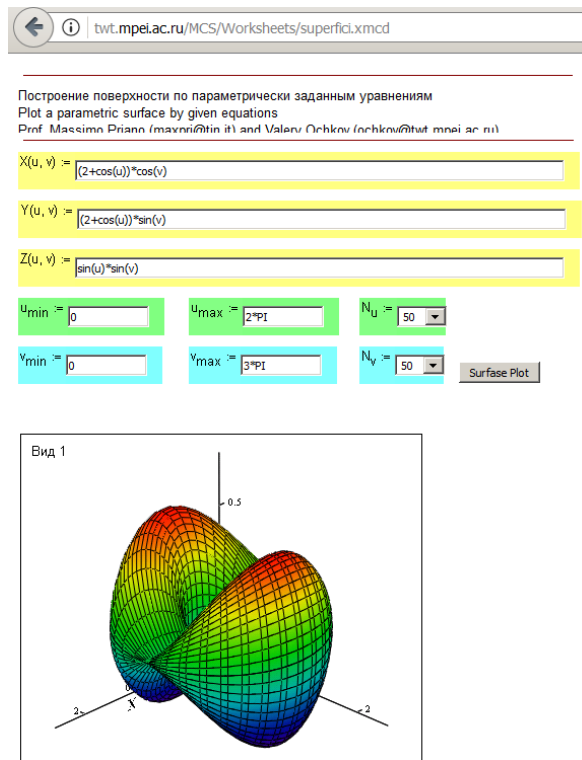


Рис. 2.4. Документ МА/СS

- Комерціалізація. Щоб розрахунки стали товаром, можна адмініструвати доступ до сайту, зробити його платним (помірно платним, частково платним), оточити посилання на них банерами і т. д.
- Будь-які помилки, помилки, недоробки і допущення в розрахунку швидко (і непомітно для користувачів) виправлені. Непомітно для користувачів можна також модернізувати і розширювати розрахунки.
- Зменшення вимог до кваліфікації користувачів. Проектувальники розробляють документи, модифікують, виправляють помилки в повноцінному середовищі. Користувачі заносять в форми потрібні вихідні дані та отримують результат в звичайних Інтернет браузерях.

Програмне вікно MathCAD

Програмне вікно *MathCAD* до 15-ї версії має стандартне оформлення *Windows* програм (рис. 2.5). Програмне вікно містить наступні елементи:

рядок заголовка, головне меню, панелі інструментів, робочу область, рядок стану. Версії *MathCAD Prime* мають інтерфейс на базі елементів типу стрічка (*ribbon*).

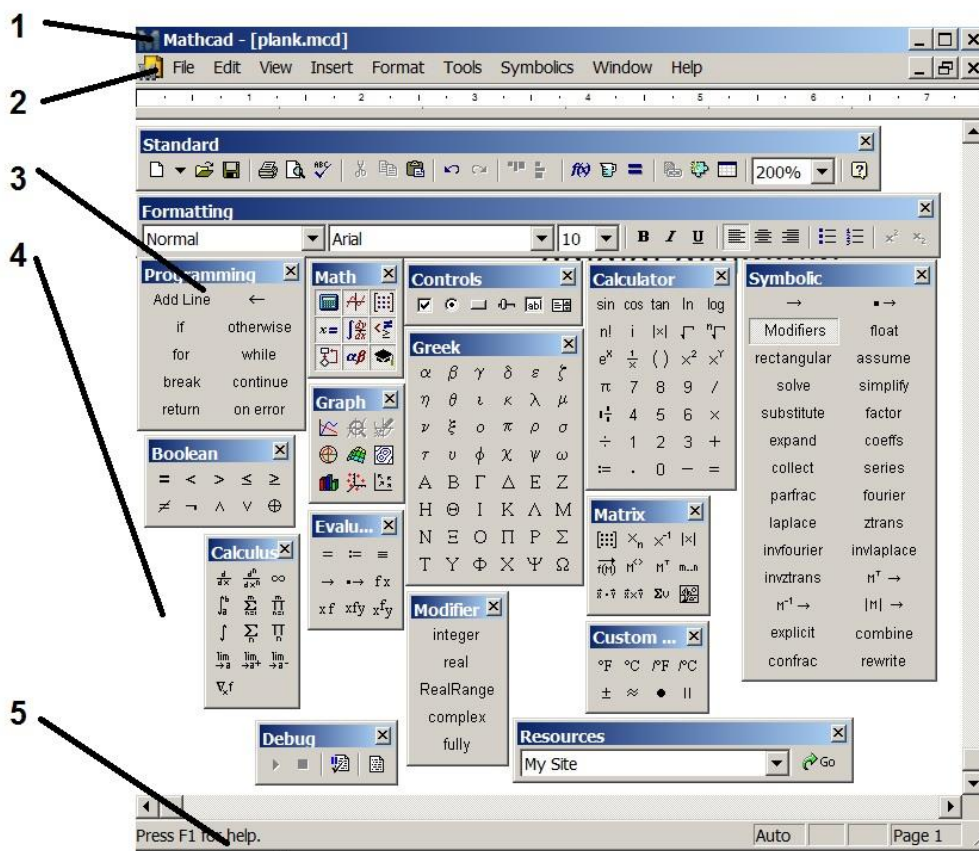


Рис.2.5. Програмне вікно MathCAD

- 1 - рядок заголовка; 2 - головне меню; 3 - панелі інструментів;
4- робоча область; 5 - рядок стану

Рядок заголовка містить ім'я відкритого файлу. Якщо відкрито новий документ, то в рядку знаходиться назва *Untitled*.

В нижній частині програмного вікна знаходиться *рядок стану*. В полях рядка відображається інформація про режими роботи пакета та документ, що редагується.

В верхній частині вікна нижче рядка заголовка знаходиться меню. Меню має наступні пункти: *File, Edit, View, Format, Tools, Symbolic, Window, Help*.

File - робота з файлом (створити, відкрити, зберегти, друкувати тощо);

Edit - стандартні операції вирізки, копіювання, вставлення, OLE операції з документом;

View – керування виглядом програмного вікна;

Format – керування виглядом елементів документа в робочому вікні;

Tools – керування режимами обчислень (**Calculate**), анімації (**Animation**), налагодження програм (**Debug**) та значеннями системних змінних точності, форматів даних, матриць;

Symbolics – визначення символічних операцій та виглядом результатів;

Window – керування розташування кількох робочих вікон;

Help – довідникова система.

Пункт *View* меню дозволяє керувати виглядом:

Toolbars – панелей, які можна вивести у вікно документа або сховати;

Ruler - мірильної лінійки згори у вікні документа;

Status bar - рядка стану внизу вікна документа;

Trace Window - вікна трасування програм;

Header and Footer - колонтитулів документа;

Regions – виділенням кольором існуючих в документі областей;

Annotations - приміток у вигляді кольорових дужок біля кожної області документа;

Refresh – оновленням зображення документа;

Zoom - масштабом зображення документа.

Елементи довідникової системи зосереджені в меню *Help* (рис. 2.6).

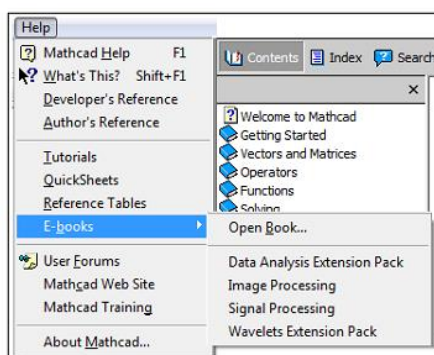


Рис. 2.6. Довідникова система MathCAD

Для виклику довідникової системи достатньо натиснути клавішу F1.

Якщо курсор встановлено на елемент документа, то натискання клавіші F1 відкриває вікно довідникової системи з інформацією саме про цей елемент.

Для введення в документ *MathCAD* шаблонів математичних знаків, конструкцій програмування, графічних областей тощо, зручно використовувати *кнопкові панелі* (рис. 2.5). Керують видимістю панелей пункти меню *View – Toolbars* пакета. Кнопкову панель можна розмістити в будь-якому місці вікна *MathCAD*.

Кнопки виведення на екран більшості панелей знаходяться на панелі *Math*. Призначення кнопкових панелей пояснено в таблиці 2.1.

Таблиця 2.1. Кнопкові панелі *MathCAD*

Значок кнопки	Призначення панелі	Значок кнопки	Призначення панелі
	“Calculator” Введення знаків арифметичних операцій, цифр		“Boolean” Введення знаків логічних дій
	“Matrix” Введення матриць, дії з матрицями		“Calculus” Введення знаків алгебраїчних дій
	“Graph” Побудова графіків		“Symbolic” Проведення обчислень в символічному вигляді
	“Programming” Опис підпрограм-функцій		“Evaluation” Введення знаків обчислення
	“Greek” Введення літер грецького алфавіту		

Панелі *Standard*, *Formatting*, *Controls*, *Modifier*, *Debug*, *Custom Characters* викликати можна тільки через меню *View - Toolbars*.

Перші дві панелі (рис. 2.7) нагадують відповідні панелі текстового процесора *Word* та табличного процесора *Excel*.

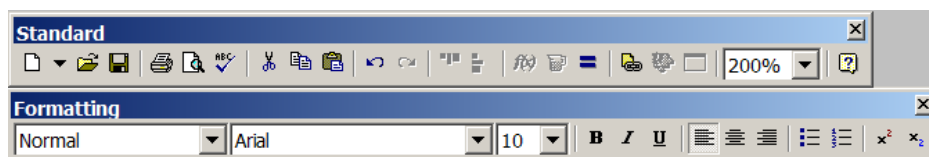


Рис. 2.7. Панелі *Standard*, *Formatting*

Панель *Controls* вміщує органи керування для створення візуального діалогу введення даних користувачем. Панель *Modifier* слугує для тонкого налаштування типів даних при символічних перетвореннях. Панель *Debug* слугує для відлагодження програм. Панель *Custom Characters* містить

символи, які часто використовуються: градуси Фаренгейта, Цельсія, плюс-мінус тощо.

Для виклику на екран потрібної панелі достатньо клацнути лівою кнопкою миші на значку, а потім прийомом "*перетягнути-і-залишити*" (*drag and drop*) розмістити панель в зручному для роботи місці вікна *MathCAD*.

Для вставки потрібної конструкції з панелі у вікно слід клацнути мишею на відповідному значку панелі.

Робоче поле MathCAD знаходиться під панелями інструментів. В ньому розташовуються одне чи декілька вікон документів *MathCAD*. Документ відображає математичну, текстову, графічну інформацію, яку вводить користувач та результати розрахунків системи. Інформація в документі групується в *математичних, текстових областях, областях графіків та оформлення (regions)*. *Області* в документі можуть розташовуватися в будь-якому порядку та кількості без обмежень

Математична область призначена для введення конструкцій пакета, необхідних для реалізації алгоритму розв'язання задачі: змінних, констант, формул, функцій, програм. Курсор в математичній області має форму червоного хреста.

Текстова область може містити будь які символи (кириличні, латинські) у вигляді коментарів, пояснень тощо. Робота в текстовій області здійснюється точно так, як в звичайному текстовому редакторі.

В *області графіків* виводяться двомірні та тримірні графіки.

Область оформлення використовується для розміщення в ній бітових графічних зображень в стандартному форматі *jpg, bmp*. Крім оформлення ці зображення можна редагувати. Для цього слугує відповідна кнопкова панель (рис. 2.8).

Примітка. Для оформлення документа слід вставляти зображення через *Clipboard*. В протилежному випадку навколо зображення буде зображуватися рамка з службовою інформацією. Для отримання можливості в подальшому отримати дані про зображення та редагувати його в документі – пунктом *Picture* меню *Insert*.

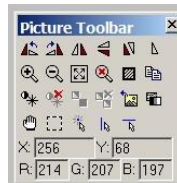


Рис. 2.8. Панель оброблення зображень

Справа на робочому полі знаходиться вертикальна зелена риска. Вона розділяє документ на стовбці. Ширина стовбцю відповідає ширині аркуша. Область документа друкується по стовбцях згори вниз. По досягненню кінця інформації одного стовбцю, система переходить до друкування сусіднього, який знаходиться праворуч.

2.1.2. Контрольні питання

До якого класу програмного забезпечення відноситься MathCAD?

Які особливості пакета MathCAD?

Перерахуйте основні можливості пакета MathCAD?

Що розміщується в програмному вікні MathCAD?

Які функції довідникової системи MathCAD?

Що таке електронні книги MathCAD?

З яких елементів формується робоче поле MathCAD?

Для чого призначені панелі, які панелі присутні в пакеті MathCAD?

Що означає вертикальна лінія на робочому полі MathCAD?

З яких елементів формується робоче поле MathCAD?

Назвіть складові СКМ MathCad та їхнє призначення.

Що таке Mathcad Application /Computation Server (MathCad Edge), для чого він використовується?

2.2. Робота з документами MathCAD

2.2.1. Створення та редагування документа

Створення документа *MathCAD* складається з введення та редагування математичних формул, графіків, визначення форматів відображення змісту

документа, введення текстових описів тощо. Введення інформації в документ *MathCAD* проводиться в визначеній області.

Виділену рамкою область можна переносити в інше місце. Для цього треба клацнути мишею на області, підвести курсор до рамки. При цьому курсор перетворюється на зображення руки. Якщо до початку зсуву натиснути та тримати клавішу *Ctrl*, область буде скопійована зі збереженням її копії на первісному місці.

Аналогічним способом можна змінити розмір текстових та графічних областей. Для цього слід потягнути курсором у вигляді стрілки маркер рамки області з натиснутою лівою кнопкою миші.

В *робочому полі* *MathCAD* зазвичай знаходяться кілька областей. При їх зсуві чи масштабуванні можливо їхнє перекриття. Для того, щоб візуально виділити межі областей призначено пункт *Regions* меню *View*. Всі проміжки між областями зафарбовуються в сірий колір так, що світлі області стають виділеними на темному фоні.

Пункт *Separate Regions* меню *Format* автоматично зсуває області таким чином, що вони розташовуються без перекриття.

Для горизонтального чи вертикального вирівнювання областей призначені пункт *Align Regions* меню *Format*. Декілька областей групуються щикликом миші з утриманням клавіші *Shift* або рамкою виділення мишею. Далі, в залежності від вибору: *Across*, *Down*, - вони вирівнюються відповідно по горизонталі чи вертикалі.

Властивості областей можна редагувати за допомогою контекстного меню. Контекстне меню викликається клацанням правої кнопки миші на області. Зміст меню визначається типом області.

При запуску системи за замовчанням встановлюється математична область введення.

Для обмеження доступу до інформації області можуть бути закриті паролем (*lock*) або „сховані” (*collapse*), тобто прибрані з екрану.

Керування (*lock, collapse*) областями можливо, коли вони поєднані в *area*. *Area* займає весь рядок документа та вміщує в себе всі області, які знаходяться всередині неї. Вводиться в документ пунктом *Area* меню *Insert*.

Для прибирання областей з екрану можна застосувати три способи.

Спосіб 1. Слід два рази клацнути на верхньому маркері *Area*. Повторне клацання виводить зміст *Area* на екран.

Спосіб 2. Пунктами *Collapse, Expand* контекстного меню *Area* .

Спосіб 3. Пунктами *Collapse, Expand* меню *Format*.

Для блокування доступу до *Area* можна застосувати пункти *Lock, Unlock* контекстного меню верхнього маркера *Area* або пунктами *Lock, Unlock* меню *Format*.

Введення команд може бути виконано декількома способами:

- шляхом натискання комбінацій "гарячих" клавіш;
- вибором пункту меню або кнопки однієї з панелей;
- написанням команди безпосередньо в тексті документа.

2.2.2. Робота з текстовою областю документа

Текстова область може бути створена трьома способами. В усіх способах спочатку треба встановити курсор в потрібне місце документа.

Перший спосіб є найпростішим: треба просто почати вводити текст. Коли після першого слова буде набрано пробіл, математична область перетвориться в текстову. Текст буде взято в прямокутну рамку, яка відображає текстову область.

Другий спосіб полягає в тому, що введення треба почати з символу " (подвійні лапки). З'явиться прямокутна рамка текстової області.

Третій спосіб - застосувати пункт *Text Region* меню *Insert*.

Текстова область розширюється при введенні тексту. Щоб завершити введення тексту, необхідно клацнути мишею поза текстовою областю. Для редагування текстової області необхідно клацнути на ній мишею. Ознакою режиму редагування є курсор у вигляді червоної вертикальної лінії.

Для текстової області можна задавати тип шрифту, його розмір, стиль, колір, параметри абзацу, колір фона. Визначення можна проводити за допомогою головного меню *Format - Text* , панелі *Formatting*, контекстного меню.

Корисною можливістю, яка забезпечує наочність документа є можливість вставляння математичної області безпосередньо всередину текстової області. Для цього слід поставити курсор редагування текстової області в потрібне місце та використати пункт *Insert Math Region* контекстного меню текстової області.

2.2.3. Збереження документа MathCAD

Для збереження документа *MathCAD* в файлі необхідно застосувати пункт *Save as* меню *File*.

Документи мають власний формат. Пакети старших версій можуть читати документи, які розроблені пакетами молодших версій. До версії 12 документи мали розширення **.mcd* . За впливом компанії *Microsoft* далі документи було переведено у стандарт *xml* та вони отримали розширення **,xmcd* .

Зберегти документ (рис. 2.9) можна не тільки в форматі *MathCAD* з розширеннями **.xmcd*, **.mcd*, а також в форматі web (**.htm*) та форматі RTF (**.rtf*).

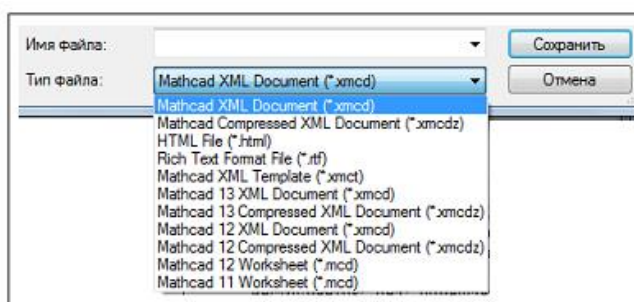


Рис. 2.9. Список форматів документа MathCAD

2.2.4. Контрольні питання

Яке розширення мають файли документів MathCad?

Які дії можна виконувати з областями робочого поля?

Як викликається контекстне меню?

Перерахуйте способи виклику команд MathCad.

Для чого області прибираються з екрану?

Як створити текстову область?

2.3. Організація обчислень

В системі передбачено засоби для роботи з наступними типами даних:

- цілі (12, -24, 0 і т.д.);
- дійсні. Можуть записуватися в формі фіксованої крапки (3.265) та в експоненційній формі ($a \cdot 10^p$, де a – ціле або дійсне число з фіксованою точкою – мантиса, p – десятковий порядок);
- комплексні ($a + bi$, де a – дійсна частина , b – уявна, i або j – уявна одиниця). Літера уявної одиниці: «i», «j» - визначається пунктом *Result-Display options – Imaginary Unit* меню *Format* (рис. 2.10). Між уявною частиною та уявною одиницею знак множення НЕ СТАВИТЬСЯ ;
- рядки – будь яка послідовність символів, що забрана в лапки ("Це рядок");
- одиниці вимірів фізичних величин.

Особливим типом даних є *NaN (NotANumber)*. Він призначений для ідентифікації елементів масивів, в які не занесені жодні дані. Наприклад, при зчитуванні з зовнішнього файлу матриці, пропущеним елементам буде надано тип *NaN*. При побудові графіків елементи з типом *NaN* ігноруються.

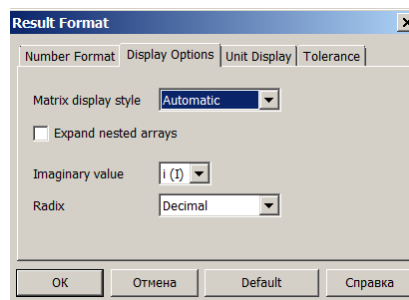


Рис. 2.10. Визначення літери уявної одиниці

2.3.1. Введення змінних, виразів. Виведення результатів

Змінні

Дані використовуються у вигляді змінних: *простих змінних, масивів у вигляді векторів та матриць, ранжованих змінних*. Змінним може бути надана розмірність для розв'язання фізичних задач.

За зоною дії в документі змінні діляться на *локальні та глобальні*. *Локальне* визначення діє від місця визначення вниз до кінця документа, *глобальне* визначення діє на весь документ, незалежно від місця визначення.

Кожна змінна повинна мати унікальне ім'я. Ім'я змінних складається з латинських чи грецьких літер та цифр. Використання в імені кирилиці чи пробілів не дозволяється.

Для керування математичними діями в пакеті призначені кілька *системних змінних*. Визначення системних змінних можна проводити безпосередньо в робочому полі або у вікні меню *Tools – Worksheet Options – Built-in-Variables* (рис. 2.11).

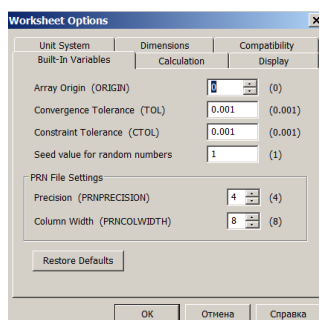


Рис. 2.11. Вікно системних змінних

ORIGIN – визначає початковий індекс масивів. Значення за замовчанням – 0.

TOL – визначає точність чисельного обчислення рівнянь, інтегралів, похідних. Значення за замовчанням - 0.001.

CTOL – визначає точність чисельного обчислення систем рівнянь блоків «*Given*». Значення за замовчанням - 0.001.

PRNPRECISION, *PRNCOLWIDTH* визначають формат запису даних в текстові файли ASCII функціями *WRITEPRN*, *APPENDPRN*. Значення за замовчанням – 4, 8, відповідно.

Примітка. При повторному застосуванні імені, яке вже використовується системою, ім'я підкреслюється зеленою хвилястою лінією. Така ситуація можлива, коли користувач вводить змінну, яка в системі визначена, як фізична константа тощо. Наприклад, c – швидкість світла $2.998 \cdot 10^8 \text{ м} \cdot \text{с}^{-1}$.

Грецькі літери вводяться з панелі *Greece* або комбінацією «латинська літера + *Ctrl* +G (табл. 2.2).

Таблиця 2.2. Клавіатурна відповідність літер грецького алфавіту

Буква	α	β	δ	ε	φ	Γ	Φ	λ	η	Ω	π	θ	ρ	σ	τ
Клавіша	a	B	d	e	F	G	H	l	n	O	p	q	r	s	t

В пакет вбудовано кілька констант (табл. 2.3):

Таблиця 2.3. Вбудовані константи

Назва	Знак	Введення	Значення
Основа натурального логарифму	e	e	2.718
Нескінченність	∞	Ctrl+Shift+z	$1 \cdot 10^{304}$
Квадратура кола	π	Ctrl_Shift+p	3.141592
Відсоток	%	%	0.01

Для візуального ефекту ім'я може мати підрядковий індекс (підрядковий індекс НЕ Є ЧИСЛОМ ІНДЕКСУ МАСИВУ). Підрядковий індекс вводиться за допомогою клавіші «.» (точка).

Наприклад, послідовність «C» «.» «i» дає змінну C_i , де «i» не є індексом, а є часткою імені.

Великі та малі літери в іменах визначають різні імена.

Дані цілого типу можуть оброблятися в десятковій, восьмеричній, двійковій та шістнадцятиричній системах числення. За замовчанням використовується десяткова система. Для явного визначення системи числення слід після чисельного значення вказати літеру «o» для восьмеричної системи, «h»- для шістнадцятиричної, «b» - двійкової. Визначення системи числення результату проводиться пунктом *Result-Display options – Radix* меню *Format*.

Для визначення змінної застосовується знак прирівнювання. В системі розрізняють *локальне та глобальне* прирівнювання. *Локальне* «:=» прирівнювання розповсюджується вправо – вниз від місця визначення. *Глобальне* прирівнювання «≡» діє на увесь документ незалежно від місця визначення. Глобальне прирівнювання доцільно застосовувати для констант, які передбачається використовувати у всьому документі.

Для введення знаку локального прирівнювання слід використати панель *Evaluate* або ОДНУ клавішу «:=». Наприклад,

Ім'я змінної := Вираз

Кількість прирівнювань однієї змінної в документі не обмежується.

Для введення знаку глобального прирівнювання слід використати панель *Evaluate* або клавішу «~». Наприклад,

Ім'я змінної ≡ Вираз



Для виведення значень змінних та обрахунків призначено знак рівняння «=». Для введення знаку рівняння слід використати панель *Evaluate* або клавішу «=».

Знаки рівняння та прирівнювання можна використовувати ланцюгом. Наприклад, запис « $c := \sqrt{5} =$ » присвоює змінній *c* значення обрахованого виразу та виводить на екран значення 2.236

Введення математичних виразів

Обчислення проводяться в *математичній області*. *Математична область* встановлюється за замовчанням. Ознакою *математичної області* є курсор у вигляді червоного хреста. З цього місця можна починати введення даних. З початком введення курсор набуває вигляд блакитного кута.

Кут може мати праву та ліву орієнтацію. Орієнтація показує напрям введення виразу. Зміна орієнтації проводиться натисканням клавіші [Insert], зсув маркера по виразу – клавішами «→», «←», зміна горизонтального розміру маркера – клавішею «*Space*».

Наприклад, якщо кут правий, то наступний символ «+» вводиться праворуч:  .

Якщо кут лівий, то символ вводиться зліва:  

Фрагмент, який горизонтально охоплює кут є операндом для операції чи функції. При введенні арифметичного знаку, дія якого відноситься до фрагменту математичного виразу, необхідно змінити розмір кута так, щоб він охоплював фрагмент.

Знаки арифметичних операцій вводяться з клавіатури або кнопками панелі *Calculator*. Деякі математичні операції наведено в таблиці 2.4.

Таблиця 2.4. Введення основних арифметичних операцій з клавіатури

Дія	Зображення	Клавіші	Дія	Зображення	Клавіші
додавання	2+3	+	модуль	3	
віднімання	2-3	-	корінь	$\sqrt{3}$	\
ділення	$\frac{2}{3}$	/	факторіал	2!	!
множення	2.3	*	n-й корінь	$n\sqrt{3}$	Ctrl+\
ступінь	X2	^	комплексне пов'язання	$\overline{2+3j}$	Shift +"
додавання в два рядки	2 ... +3	Ctrl+Enter	ділення в один рядок	2I3	Ctrl+/

Всі результати та проміжні дані зберігаються в пам'яті системи з максимальною точністю 15 знаків мантиси. Вигляд результату може визначати користувач в діалоговому вікні (рис. 2.12). Виклик вікна проводиться пунктом *Result – Number Format* меню *Format* або подвійним щигликом безпосередньо на результаті.

$$\frac{13+15}{3 \cdot (27-8)} \cdot \frac{56-34}{19} = 0.569.$$

Наприклад, обрахування значення виразу

може складатися з наступних дій:

1) Набрати чисельник першого першого множника та клавішею *Space*

охопити його: 13 + 15 :

2) до виділеного виразу застосувати арифметичну операцію ділення :

$$\frac{13 + 15}{}$$

3) набрати знаменник та охопити весь дріб правим кутом, щоб справа

ввести знак множення: $\frac{13 + 15}{3 \cdot (27 - 8)}$;

4) ввести знак множення:

$$\frac{13 + 15}{3 \cdot (27 - 8)} \cdot$$

5) аналогічно пунктам 1-3 провести набір другого множника та охопити весь вираз правим кутом:

$$\frac{13 + 15}{3 \cdot (27 - 8)} \cdot \frac{56 - 34}{19}$$

б) ввести оператор виведення =:

$$\frac{13 + 15}{3 \cdot (27 - 8)} \cdot \frac{56 - 34}{19} = 0.569$$

Формати виведення результатів

Результати можуть бути виведені в наступних форматах *Decimal*, *Scientific*, *Engineering*, *General*, *Fraction*.

Decimal - формат дійсних чисел з фіксованою крапкою. Кількість цифр після коми задається в полі *Number of decimal places*. Поле *Show trailing zeros* визначає, чи додавати в кінці числа нулі до потрібної кількості цифр.

Scientific - експоненційний формат. Поле *Show exponents as E+000* визначає як виводити основу: у вигляді 10 або E (1.22 · 10² чи 1.22E+2).

Engineering - експоненційний формат. Відмінність від формату *Scientific* полягає в тому, що десятковий порядок кратний 3.

General - універсальний формат. Якщо кількість цифр цілої частини числа не перебільшує значення з поля *Exponential threshold*, то виводиться число з фіксованою крапкою. В протилежному випадку – в експоненційній формі.

Fraction - формат у вигляді дробу. Поле *Use mixed numbers* дозволяє виділити цілу частину дробу.

Наприклад, формат *Fraction* для виразу $\frac{8}{6}$ без поля *Use mixed numbers* дає результат $\frac{4}{3}$, з полем - $1\frac{1}{3}$.

Визначення формату проводиться обраним пунктом діалогового вікна (рис. 2.12).

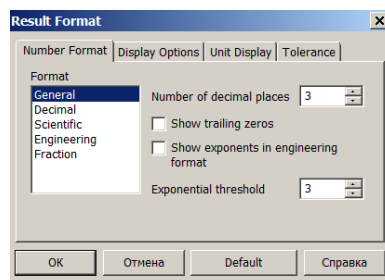


Рис. 2.12. Вікно визначення форматів результатів обчислень

Вікно викликається подвійним щигликом миші на результаті або пунктом *Result* меню *Format*.

2.3.2. Режими обчислень

В системі передбачено два режими обчислень:

- *автоматичний режим*, коли всі обчислення виконуються автоматично під час введення формул;
- *ручний режим*, коли старт обчислень окремого виразу чи всього документа задається користувачем.

Коли вираз знаходиться в процесі розрахунку чисельним або символьним процесором, він виділяється рамкою зеленого кольору, а будь-які дії користувача щодо подальшого редагування документа блокуються.

Автоматичні обчислення спрощують роботу з документом, оскільки результати розрахунків з'являються в реальному часі, і користувач має можливість аналізувати їх відразу. З іншого боку, якщо обчислення складні, вони можуть забирати багато часу. Зміна якого-небудь виразу на початку великого документа, який впливає на наступні обчислення, призводить до перерахунку всіх залежностей заново. У таких випадках часто зручніше

редагувати текст в ручному режимі, а обчислення включати в міру необхідності.

Автоматичний режим включається за замовчанням при створенні нового документа.

В *ручному режимі* користувач повинен запускати обчислення самостійно. При редагуванні тексту в ручному режимі не виконуються ані обчислення виразів, ані побудова графіків, відповідні місця в виразах відзначаються маркерами (рис. 2.13).

Для того щоб обчислити увесь документ слід обрати пункт *Calculate Worksheet* меню *Tools* – *Calculate* або клавіші *Ctrl+F9*.

Для обчислення всіх формул у видимій на екрані частині документа слід обрати пункт *Calculate Now* меню *Tools* - *Calculate* або клавішу *<F9>*.

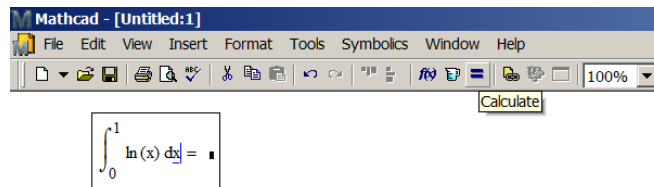


Рис. 2.13. Вигляд документа в ручному режимі обчислень

Найпростіше запуснути розрахунки можна клацанням на кнопці *Calculate* із зображенням знака рівності на панелі *Standard* (рис.2.13).

Щоб перервати тривалий процес обчислень треба натиснути клавішу *<Esc>*. У цьому разі вирази, які *Mathcad* не встиг вирахувати, будуть позначені в документі червоним кольором.

Перерване обчислення поновлюються натисканням клавіші *<F9>*.

Оптимізація обчислень

Особливістю *Mathcad* є прискорення чисельних обчислень за рахунок застосування елементів символної математики. Безпосередньо перед чисельним розрахунком *Mathcad* автоматично намагається спростити вираз, використовуючи символний процесор. Це називається *оптимізацією*. Режим оптимізації вмикається для цілого документа або для окремих формул.

Щоб увімкнути або вимкнути режим оптимізації всіх виразів в документі слід застосувати пункт *Optimize* меню *Tools*.

Щоб змінити режим оптимізації для окремих виразів слід виділити цю формулу лініями введення і вибрати у меню *Tools - Optimize* пункт *Equation* або обрати такий пункт в контекстному меню виразу.

Альтернативою є пункт *Disable Evaluation* на сторінці *Calculation* контекстного меню *Properties* виразу. Ознакою того, що вираз знаходиться в режимі оптимізації є блакитний квадрат справа від виразу.


Відключення обчислення окремих формул

Mathcad дозволяє відключити обчислення будь-якої формули. При цьому вона не буде впливати на подальші обчислення. Щоб не обчислювати окрему формулу в документі треба в контекстному меню виразу обрати пункт *Disable Evaluations*.

Еквівалентний спосіб відключення обчислення окремої формули полягає у виклику діалогового вікна *Properties* через однойменний пункт контекстного меню. У діалозі *Properties* слід перейти на сторінку *Calculation* і обрати *Disable Evaluations*. Про виключення формули з обчислень можна судити за наявністю чорного маркера відразу за цією формулою.

2.3.3. Вбудовані функції

Для обчислення значень найбільш поширених математичних функцій в *MathCAD* застосовуються так звані *вбудовані функції* (наприклад, синус, корінь квадратний і т.д.). Вбудовані функції дозволяють вирішувати складні завдання (наприклад, побудова регресії, розв'язання диференціального рівняння). Для звернення до функції необхідно ввести її ім'я або умовний символ та аргумент.

Функцію можна ввести з клавіатури, пунктом *Function* меню *Insert*. Найбільш уживані функції розташовані на панелі *Calculate*. Повний список функцій і їх синтаксис можна отримати у вікні *Insert Function* (рис. 2.14) клацнувши на кнопці  панелі інструментів *Standart*.

Весь набір вбудованих функцій у вікні *Insert Function* розбитий на групи відповідно до функціонального значення:

Bessel — спеціальні звичайні та модифіковані функції Беселя;

Complex Numbers — функції роботи з комплексними числами;

Curve Fitting and Smoothing — функції апроксимації кривих;

Differential Equation Solving — функції розв'язання диференціальних рівнянь;

Expression Type — функції для визначення типів даних;

File Access — функції для роботи з файлами;

Finance — фінансові функції;

Fourier Transform — функції перетворення Фур'є;

Graphing — функції для роботи з графічними об'єктами;

Hyperbolic — гіперболічні функції;

Image Processing — функції для обробки зображень;

Interpolation and Prediction — функції інтерполяції та екстраполяції;

Log and Exponential — логарифмічні функції;

Lookup — функції для пошуку даних;

Number Theory/Combinatorics — функції теорії чисел;

Piecewise Continuous — кусково-неперервні функції;

Probability Density, Probability Distribution, Random Numbers — функції теорії імовірності;

Signal Processing — функції для обробки сигналів;

Solving — функції розв'язання рівнянь;

Sorting — функції сортування даних;

Statistics — статистичні функції;

String — функції для роботи з текстом;

Trigonometric — тригонометричні функції;

Truncation and Round Off — функції округлення;

User Defined — функції користувача;

Vector and Matrix — функції для роботи з матрицями;

Wavelet Transform — функції для вейвлет-перетворення.

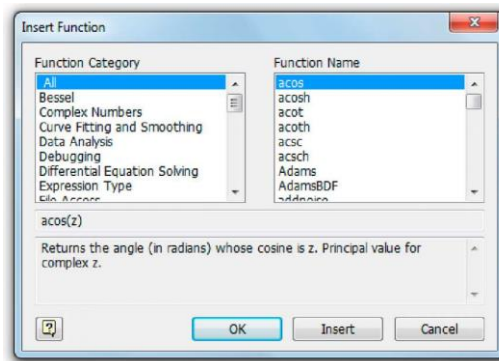


Рис. 2.14. Вікно вбудованих функцій

У вікні необхідно вибрати відповідну категорію *Function Category*. При цьому у вікні *Function Name* будуть представлені функції *MathCAD*, що відносяться до даної категорії. Якщо необхідна функція в обраній категорії не буде знайдена, можна переглянути категорію *All*, що містить всі вбудовані функції. Короткий опис виділеної функції показано в нижній частині вікна. Повний опис можна отримати з довідки, яка викликається за посиланням «?» в лівому нижньому кутку вікна.

Далі наведені найбільш уживані вбудовані математичні функції *MathCAD*:

sin(z), cos(z), tan(z), cot(z), sec(z), csc(z) - тригонометричні функції: синус, косинус, тангенс, котангенс, секанс, косеканс.

Для зворотних тригонометричних функцій перед іменем додається літера «а». Наприклад, **asin(z)**, - арксинус.

Для гіперболічних функцій і кінці імені додається літера «h». Наприклад **sinh(z)** - гіперболічний синус, **asech(z)** - зворотній гіперболічний арксеканс. Для зворотніх гіперболічних – попереду додається «а», в кінці – «h».

atan2(x, y) - $\text{atan2}(x,y) = \text{atan}(y/x)$

sinc(z) - функція $\text{sinc}(z) = \sin(z)/z$

exp(z) - експоненційна функція (e^z)

ln(z) - логарифм натуральний z

log(z,b) - логарифм від z по основі b. Якщо другий аргумент «b» не визначено, основа вважається 10.

DMS(x) – значення кута x в радіанах.

Примітка. Аргументи тригонометричних функцій визначаються в радіанній мірі. Для позначення даних в градусах можна додати після виразу службове слово «deg».

Функції роботи з комплексними числами

Більшість вбудованих функцій можуть мати в якості аргументів комплексні числа, результатами обчислень таких функцій також є комплексні числа.

Для аналізу комплексних значень в системі передбачені функції обробки комплексних чисел.

Для визначення комплексного числа достатньо знати два параметри. Це можуть бути дійсна C і уявна B частини комплексного числа або його модуль A і фаза φ . Ці параметри взаємопов'язані.

$$z = C + jB = A \cdot e^{j\varphi} \quad A = \sqrt{C^2 + B^2} \quad \varphi = \operatorname{arctg}\left(\frac{B}{C}\right)$$

Функції розрахунку дійсної та уявної частин комплексного числа z в *Mathcad* мають загальноприйняті математичні назви: $\operatorname{Re}(z)$ для розрахунку дійсної частини та $\operatorname{Im}(z)$ - для уявної. В термінах *Mathcad* фаза комплексного числа називається його аргументом. Визначити аргумент можна за допомогою функції $\operatorname{arg}(z)$.

Функція $\operatorname{Csgn}(z)$ повертається знак дійсної частини комплексного числа z . Якщо дійсна частина числа дорівнює нулю, функція повертає знак уявної частини.


Примітка. Часто вбудовані функції *Mathcad* мають деякі особливості застосування (наприклад, обмежена область визначення, обробка комплексних аргументів тощо), незнання яких може привести до похибок. Це зауваження, відноситься не тільки до *Mathcad*, а й до більшості аналогічних математичних пакетів. Якщо є хоча б найменший сумнів в адекватності обраної функції для розв'язуваної задачі, то доцільно витратити додатковий час на ознайомлення з повною довідковою інформацією по функції.


Розширені арифметичні оператори

MathCAD має ряд операторів, що дозволяють спростити запис сум, добутків, похідних, визначених інтегралів, меж. Такі оператори часто називають *розширеними арифметичними операторами* або *операторами математичного аналізу*.

Ці оператори (значок оператора) зручно ввести в документ *MathCAD* використовуючи панель *Calculus*.

При введенні оператора зображується шаблон для подальшого заповнення числовими або символічними значеннями.

Оператор обчислення суми. Для введення оператора в документ необхідно клацнути на кнопці  та заповнити поля шаблону (рис. 2.15а).

Оператор обчислення доданку. Для введення оператора в документ слід клацнути на кнопці  та заповнити поля шаблону (рис. 2.15б).

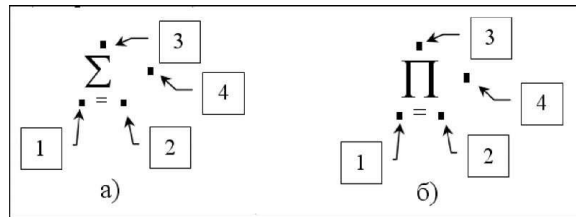


Рис. 2.15. Шаблони операторів сум та доданків

а – шаблон сум, б – шаблон доданку.

1 – ім'я параметра сумування (множення); 2 - нижня межа; 3 – верхня межа; 4 - вираз, який залежить від параметра сумування (множення)

Як більшість команд системи, розширені арифметичні оператори мають «гарячі» клавіші (рис. 2.16).

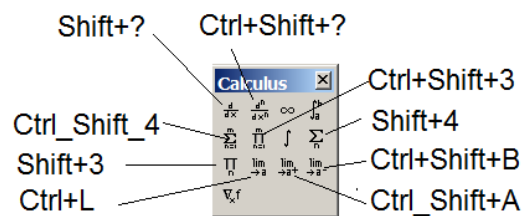





Рис. 2.16. Клавіші розширених операторів

Оператори обчислення похідних. Для введення оператора диференціювання в документ необхідно клацнути на кнопці  і заповнити поля шаблону (рис. 2.17а).

Примітка. Значення похідних обчислюються чисельним методом, тому необхідно попередньо задавати значення аргументів, при яких будуть обчислені похідні.

Для введення оператора обчислення похідної n -го ($n < 5$) порядку необхідно клацнути на кнопці  і заповнити поля шаблону (рис. 2.17б).

Оператор обчислення визначених інтегралів. Для введення оператора в документ необхідно клацнути на кнопці  і заповнити поля шаблону (рис. 2.17в).

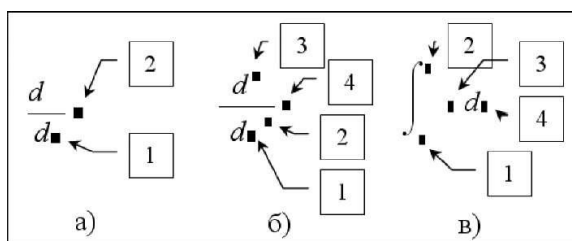


Рис. 2.17. Шаблони операторів диференціювання та інтегрування

а – диференціювання, 1 - змінна диференціювання; 2 - диференційована функція.

б – n -кратне диференціювання, 1 - змінна диференціювання; 2 - порядок похідної; 3 - порядок похідної; 4 - диференційована функція.

в – інтегрування, 1, 2 - нижня та верхня межі інтегрування; 3 - інтегрований вираз; 4 - змінна інтегрування.

Шаблони виразів та функцій

Натисканням кнопки панелі (наприклад *Evaluate, Calculate*) на екран виводиться шаблон виразу з позначеннями відповідної дії та маркерами (рис. 2.18). В поля маркерів необхідно вставити чисельні значення, імена змінних або інші математичні символи. Це дозволяє використовувати для запису виразів окрім звичного клавіатурного порядку – шаблонний [8].

Наприклад, клавіатурний порядок введення виразу

$$\left(\frac{2}{3+4}\right)^5 = 1.904 \times 10^{-3}$$

може мати наступний вигляд:

2 → / → 3 → + → 4 → Space → Space → Shift+6 → 5 → =

Такого ж результату можна отримати із застосування шаблонів.



Наприклад, застосування панелей для введення виразу

$$\frac{\ln\left(\frac{x+y}{5+y}\right) + y^2}{\sqrt[3]{x+y}} \cdot e^{-x}$$

дає наступний шаблон

Ця структура заповнюється за 10 кроків [8]: визначаються 7 змінних та 3 константи.

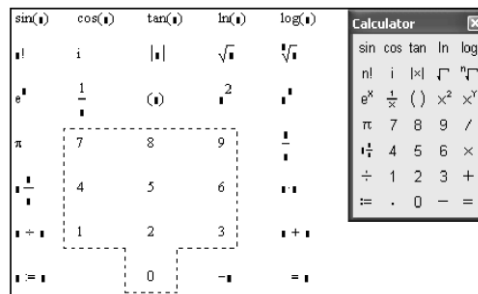


Рис. 2.18. Шаблиони з панелі Calculator [8]

Шаблиони дозволяють зробити дуже універсальним виклик функцій. Шаблиони функцій зосереджено на панелі *Evaluate* (рис. 2.19). Панель містить чотири шаблиони виклику функцій, відмінних від традиційного: префіксний, постфіксний для функцій одного аргументу, інфіксний та «дерево» – для функцій двох аргументів

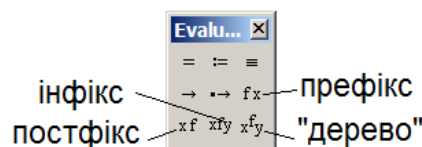


Рис. 2.19. Шаблиони виклику функцій

Наприклад, записати вираз $\arctg(\frac{3}{4})$ можна з використанням вбудованої функції **atan2** можна

традиційним клавіатурним способом **atan2(3,4)** ;

префіксним способом: **fx** → ■ ■ → **atan 0.75** ;

постфіксним способом: **xf** → ■ ■ → **0.75 atan2** ;

змішаним способом: **xfy** → ■ ■ ■ → **4 atan2 3** ;

у вигляді «дерева»: **xfy** → ■ ■ → $\begin{array}{c} \text{atan2} \\ \diagdown \quad \diagup \\ 4 \quad 3 \end{array}$.

Не можна стверджувати, що якийсь спосіб має переваги. Обидва способи є тільки інструментами. В кожному конкретному випадку користувач має можливість використати переваги одного чи іншого, комбінуючи клавіатурне та шаблонне введення.

2.3.4. Функції користувача

У *MathCAD* користувач може на додачу до вбудованих функцій створювати *функції користувача* (локальні функції). На відміну від простої змінної, значення такої функції залежить від значень аргументів, а на відміну від вбудованої функції (наприклад, функції **sin(x)**) ця функція визначається самим користувачем. Функції користувача спрощують структуру документа при необхідності багаторазових розрахунків одного виразу з різними вихідними значеннями, дозволяють модифікувати вбудовані функції в разі їхньої невідповідності вимогам користувача, створювати нові функції, які адаптовано до конкретної науково-технічної тематики.

Перед тим як використовувати функцію користувача, потрібно її задати, тобто виконати *опис*. *Опис функції* користувача розміщується в робочому документі перед викликом функції і включає в себе ім'я функції, список формальних параметрів і тіло функції.

Кожна функція користувача в *MathCAD* повинна мати оригінальне ім'я, по якому здійснюється звернення до цієї функції. Через це ж ім'я (і тільки

через це ім'я) "повертається" в робочий документ результат виконання функції користувача.

Після імені функції розміщується список *формальних* параметрів, взятий у круглі дужки. Через формальні параметри "всередину" функції "передаються" дані необхідні для виконання обчислень всередині програми. Як формальні параметри можуть використовуватися імена простих змінних, масивів і функцій. формальні параметри відокремлюються один від одного комами.

Вираз тіла функції може вміщувати тільки один рядок.

Визначення функції користувача має вигляд

<ім'я функції> (<список формальних параметрів>): = <вираз, залежний від формальних параметрів>

Як формальні параметри використовуються тільки імена (а не більш складні вирази) і ці параметри показують як значення функції залежить від аргументів, тобто ці параметри повинні бути присутніми і в правій частині опису функції користувача. Не має значення чи були раніше визначені або використані в робочому документі імена формальних параметрів. Імена формальних параметрів автономні в тому сенсі, що вони не співвідносяться до імен, які використовуються в документі.

Для виконання під функції користувача необхідно звернутися до імені функції з вказівкою списку *фактичних* параметрів.

Фактичні параметри вказують, при яких конкретних значеннях здійснюються обчислення в тілі програми. фактичні параметри відокремлюються один від одного комами. Імена фактичних параметрів при виклику функції можуть не збігатися з іменами її формальних параметрів.

Між фактичними і формальними параметрами має бути відповідність по кількості, порядку проходження і типу. Остання відповідність означає, що:

- якщо формальним параметром є проста змінна, то в якості фактичного може використовуватися константа, змінна, арифметичний вираз;
- якщо формальним параметром є вектор чи матриця, то фактичним параметром повинен бути вектор або матриця;

- якщо формальним параметром є ім'я вбудованої функції чи іншої програми, то і фактичним параметром повинний бути той самий об'єкт.

Звернення до функції має відбуватися після опису функції.

Для обчислення значення функції користувача достатньо записати ім'я функції і список фактичних параметрів, взятий у круглі дужки. Фактичні вказують, при яких конкретних чисельних значеннях аргументів буде обчислюватися функція. В якості фактичних параметрів може виступати константа, число, змінна, арифметичний вираз. Очевидно, що кількість, типи і порядок запису фактичних і формальних параметрів повинен бути однаковий.

Найбільш близька за змістом до функції користувача пакета *MathCad* є функція *DEFFN* мови *BASIC*.

Наприклад, визначення функції користувача *dist (x, y)*, яка обчислює відстань між точкою з координатами (x, y) і початком координат може мати наступний вигляд:

$$\text{dist}(x, y) := \sqrt{x^2 + y^2}$$

Можливі варіанти використання функції *dist (x, y)* описано нижче:

$$x1 := 2.1 \quad y1 := -0.1$$

$$z := \frac{\text{dist}(x1, y1)}{2} \quad z = 1.051 \quad \text{dist}(-0.2, 8.1) = 8.102$$

Примітка. Функції користувача можуть мати **ТІЛЬКИ** один рядок тіла функції.

2.3.5. Розрахунки в діапазоні значень

Для побудови графіків, розрахунків функцій в діапазоні значень (табуляція функцій) тощо. необхідно проводити обчислення одного виразу не один, а багато разів при різних значеннях аргументів.

Для таких випадків у пакеті передбачені засоби для розрахунків в діапазоні значень. Цей вид розрахунків реалізований за допомогою *ранжованих* змінних (в деяких джерелах вони називаються *дискретними змінними* [7,9]).

Дискретна змінна – змінна, яка приймає не одне, а декілька значень. Значення ранжованої змінної змінюються за законом арифметичної прогресії. Априорі відомо початкове значення змінної, крок зміни, кінцеве значення, а значить – кількість повторень циклу.


Аналогом ранжованої змінної є операція зумовленого циклу **for** в мовах програмування високого рівня.

Примітка. Відміною ранжованої змінної від циклу програмування є те, що цикл виконується по черзі для одного значення змінної циклу над рядками, а ранжований цикл виконується по черзі над рядками для всіх значень змінної.

Ранжована змінна задається в такій спосіб:

<ім'я змінної>:=початок, друге значення .. останнє значення

Друге значення дорівнює сумі першого значення та кроку зміни змінної. Якщо крок змінної дорівнює одиниці, то задавати його не обов'язково.

Для введення двох крапок між другим та останнім значеннями застосовується символ «:» (двокрапка) або кнопка  панелі *Matrix*.

Наприклад, ранжована змінна $x = 0.1, 0.3 .. 1.5$ змінюється від 0.1 до 1.5 з кроком 0.2.

Для виведення всіх значень дискретної змінної слід набрати її ім'я та поставити після нього знак прирівнювання: $x =$

Примітка. Вивести одне значення ранжованої змінної неможливо. Присвоїти значення ранжованої змінної іншій змінній **НЕМОЖЛИВО**. Для присвоєння треба використовувати масиви або функції користувача.

Наприклад, можливо розрахувати та вивести значення виразу, вбудованої функції чи функції користувача з ранжованою змінною x : $2*x$, $\sin(x)$, - але присвоїти значення цього виразу іншій змінній $y:=2*x$ – неможливо.

2.3.6. Контрольні питання

Якими типами даних оперує MathCAD?

Які змінні існують в MathCAD?

Як вводяться грецькі літери в MathCAD?

Що означає символ «крапка» в імені змінної MathCAD?

В яких системах числення працює MathCAD?

Які дії прирівнювання реалізовані в MathCAD?

Що означає маркер типу блакитного кута?

Які є формати виведення результатів в MathCAD?

Що таке режими обчислень в MathCAD?

Що означає оптимізація обчислень в MathCAD?

Які основні типи вбудованих функцій MathCAD?

Для чого використовуються розширені арифметичні оператори?

Що таке шаблони виразів?

Що таке ранжована змінні?

Які особливості застосування дискретної змінної?

2.4. Перевірка умов

При виконанні розрахунків часто доводиться оцінювати різноманітні умови. Це трапляється при описуванні кусково - неперервних функцій, імпульсних сигналів. Такі дії називаються логічними або умовними. Вони реалізують виконання алгоритму з розгалуженням.

У розгалужених алгоритмах присутні кілька гілок обчислювального процесу. Вибір конкретної гілки залежить від виконання (чи невиконання) заданих умов.

Умовні дії в системі реалізовано у вигляді унітарних логічних операторів (виразів відношень), логічних операцій, логічних виразів. Перевірка умов та розгалуження дій – логічною функцією **if**.

Введення символів логічних умов проводиться з клавіатури (табл. 2.4) або кнопками панелі *Boolean* (рис. 2.20).

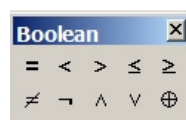


Рис. 2.20. Панель Boolean

Унітарні логічні оператори - вирази відношення

Ці вирази використовуються для порівняння двох арифметичних виразів між собою. Вираз відношення записуються в вигляді:

<вираз А> <знак відношення> <вираз В> ,

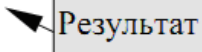
де в якості знака відношення виступають символи, наведені в таблиці 2.4. Якщо задане відношення виконується, то вираз відносин набуває значення рівне 1 ("істина"), в іншому випадку - 0 ("брехня").

Таблиця 2.4. Символи логічних відносин

Знак відношення	Введення	Знак відношення	Введення
=	Ctrl + =	≤	Ctrl + 0
<	<	≥	Ctrl + 9
>	>	≠	Ctrl + 3

Увага. Знак операції порівняння не є знаком виведення значень обчислень. Знак операції порівняння називається «жирним =».

Приклад. Обчислення виразу відношення

$x := 6 \quad x \leq 2 = 0$ 

На відміну від мов програмування можна відразу в одному виразі перевірити кілька умов шляхом додавання знаків відношення і арифметичних виразів:

$x := 6 \quad 2 \leq x \leq 8 = 1$

Логічні операції

Логічна операція «АБО» (OR)

Позначається знаком «+» або «∨» (Ctrl+Shift+6 або знаком ∨ панелі *Boolean*) і за підписується у вигляді

<Логіч.вир.1> + <Логіч.вир. 2>

<Логіч.вир.1> ∨ <Логіч.вир. 2>

Результат операції дорівнює 0, якщо обидва логічних вирази дорівнюють 0 і дорівнює 1 для всіх інших значень логічних виразів.

Логічна операція «І» (AND)

Позначається знаком «*» або «^» (Ctrl+Shift+7 або знаком \wedge панелі *Boolean*) і за підписується у вигляді

<Логіч.вир.1>. <Логіч.вир. 2>

<Логіч.вир.1> \wedge <Логіч.вир. 2>

Результат дорівнює 1, якщо обидва логічних вирази дорівнюють 1 і дорівнює 0 для всіх інших значень логічних виразів.

Логічна операція «НІ» (NOT)

Позначається знаком «¬» (Ctrl+Shift+1 або знаком \neg панелі *Boolean*) і за підписується у вигляді

¬ <Логіч.вираз>

Результат дорівнює зворотним результату логічного виразу.

Логічна операція «ВИКЛЮЧАЮЧЕ АЛЕ» (XOR)

Позначається знаком «⊕» (Ctrl+Shift+5 або знаком \oplus панелі *Boolean*) і за підписується у вигляді

<Логіч.вираз 1> \oplus <Логіч.вираз 2>

Результат дорівнює 1, якщо тільки одне з логічних вирази дорівнює 1 і дорівнює 0 для всіх інших значень логічних виразів.

Можливі значення логічних операцій наведено в таблиці 2.5

Таблиця 2.5. Значення логічних операцій

NOT \neg	AND \wedge	OR \vee	XOR \oplus
$0\neg=1$	$0\wedge 0=0$	$0\vee 0=0$	$0\oplus 0=0$
$1\neg=0$	$0\wedge 1=0$	$0\vee 1=1$	$1\oplus 0=1$
	$1\wedge 0=0$	$1\vee 0=1$	$0\oplus 1=1$
	$1\wedge 1=1$	$1\vee 1=1$	$1\oplus 1=0$

Логічні вирази

Логічним виразом називається конструкція, складена з унітарних логічних операторів, логічних операцій арифметичних операцій. Значення логічного виразу обчислюється зліва направо. Вищий пріоритет мають дії в

круглих дужках, далі логічні операції (AND, OR, XOR), найнижчий – вирази відношень (<, >, =).

Примітка. Для однозначного обчислення логічного виразу використовуйте круглі дужки:

$$x := 2 \quad y := 3 \quad 0 \leq x \leq 5 \cdot y \leq 4 = 0 \quad (0 \leq x \leq 5) \cdot (y \leq 4) = 1$$

Умовна функція if

Ця функція записується у наступному вигляді (символи *if* вводяться з клавіатури):

if (<умова>, <вираз 1>, <вираз 2>)

В якості умови можна застосовувати арифметичний вираз цілого типу або логічний вираз.

Правило обчислення умовної функції *if*: якщо вираз більше 0, то функція приймає значення рівне значенню виразу 1, якщо умова дорівнює 0, то функція набуває значення рівне значенню виразу 2. В якості виразів може використовуватися будь-яка припустима арифметична, логічна дія і сама функція *if*.

Функція реалізує структуру «якщо - то - інакше», тобто повну альтернативу. При програмуванні алгоритмів з розгалуженням по трьох та більше умовах функція *if* застосовується в якості виразів 1, 2. Така структура називається вкладеними умовними функціями.

Примітка. При перевірці умов, що описують діапазони значень, перевірку слід проводити послідовно з боку менших або більших значень.

Для підготування виразів з метою перевірки умов в системі існують такі корисні функції, як:

Ceil(x,y) - функція повертає найменше число, яке кратне *y* та більше *x*

ceil(x) - функція повертає найменше ціле число, яке більше *x*

Floor(x,y) - функція повертає найбільше число, яке кратне *y* та менше *x*

- $\text{floor}(x)$ - функція повертає найбільше ціле число, яке менше x
- $\text{Round}(x,y)$ - значення x округляється до найближчого числа за правилами математики, яке кратне y .
- $\text{round}(x,n)$ - число x округляється до n знаків після десяткової точки. Якщо другий аргумент відсутній, то округлення відбувається до найближчого цілого. Від'ємне значення n визначає кількість розрядів цілої частини числа
- $\text{Trunc}(x,y)$ - функція повертає добуток від множення y на цілу частину ділення x на y . Якщо $|x| < |y|$, то повертається нуль.
- $\text{trunc}(x)$ - функція повертає цілу частину x
- $\text{signum}(x)$ - функція повертає 1, коли $x=0$, в інших випадках - $x/|x|$
- $\text{sign}(x)$ - Повертає 0, коли $x=0$, 1 – коли $x>0$, -1 – коли $x <0$
- $\Phi(x)$ - функція Хевісайда (функція сходинки). Повертає 0, коли $x<0$, повертає 1, коли $x>0$
- $\text{max}(c)$ - функція повертає значення максимального елемента списку аргументів. В якості списку може бути масив
- $\text{min}(c)$ - функція повертає значення мінімального елемента списку аргументів. В якості списку може бути масив

2.4.1. Контрольні питання

Як ввести символи логічних умов?

Що таке унітарні логічні оператори?

Які значення приймають логічні оператори?

Які логічні логічні операції реалізовані в *MathCAD*?

Яку структуру реалізує умовна функція?

2.5. Робота з векторами та матрицями

Масиви - це об'єднання декількох однотипних об'єктів (чисел, символів, рядків, тощо), які розглядаються як єдине ціле. Доступ до кожного елемента масиву здійснюється за допомогою індексу. Індекс є даним порядкового типу, що служить своєрідним іменем елемента в масиві. При обробці елемента масиву безпосередньо за іменем елемента масиву повинен слідувати індекс. Індекс не повинен виходити за межі, визначені діапазоном. Розрізняють одновимірні масиви, що містять елементи в одному рядку -

вектори, двомірні масиви, елементи яких розташовані по рядкам і стовпцям - матриці, багатовимірні масиви складної структури.

В *Mathcad* матрицею A розміром $R \times C$ вважається сукупність даних, які розташовані у вигляді таблиці з рядків та стовпців

$$A = \begin{bmatrix} a_{1,1} & \dots & a_{1,C} \\ a_{2,1} & \dots & a_{2,C} \\ \dots & \dots & \dots \\ a_{R,1} & \dots & a_{R,C} \end{bmatrix}.$$

Накоротко матриця позначається через свій елемент. Індокси матриці розташовують після імені нижче назви. Перший індекс визначає рядок, другий – стовець.

$$A \rightarrow (a_{i,j}) \quad (i=1,2\dots R, \quad j=1,2\dots C)$$

У середовищі *MathCAD* допустимі одномірні масиви-вектори й двомірні масиви - матриці. Вектор є матрицею, в якій кількість стовпців становить 1.

Обробка масивів за вибором користувача здійснюється поелементно або за допомогою матричних (векторних) операцій. Для вибору елементу масиву використовуються *індексовані змінні*. *Індексована змінна Mathcad* аналогічна елементу масиву в мовах програмування. Індокси масивів в можуть приймати тільки цілі значення.

Введення індексованих змінних

Вбудовані засоби *Mathcad*, які призначені для роботи з матрицями, зібрані на панелі *Matrix* (рис. 2.21).

Початковий індекс матриць та векторів задається системною змінною *ORIGIN*. За замовчанням значення початкового індексу дорівнює 0. В системі індекс масивів може приймати не тільки натуральні, а й цілі значення, тобто бути від'ємним. Змінити значення початкового індексу можна безпосередньо в тексті документу або у вікні системних змінних (рис. 2.11) через пункт *Worksheet Options* меню *Tools*.

Індекс змінної виступає ключем при обчисленнях чи реалізації ітераційних алгоритмів.

Визначення матриць можливо двома способами:

1. Безпосереднє присвоювання значення індексованій змінній виконується натисканням клавіші «[» після імені елемента. При цьому на екрані з'являється маркер індексу, на місце якого потрібно ввести необхідне значення

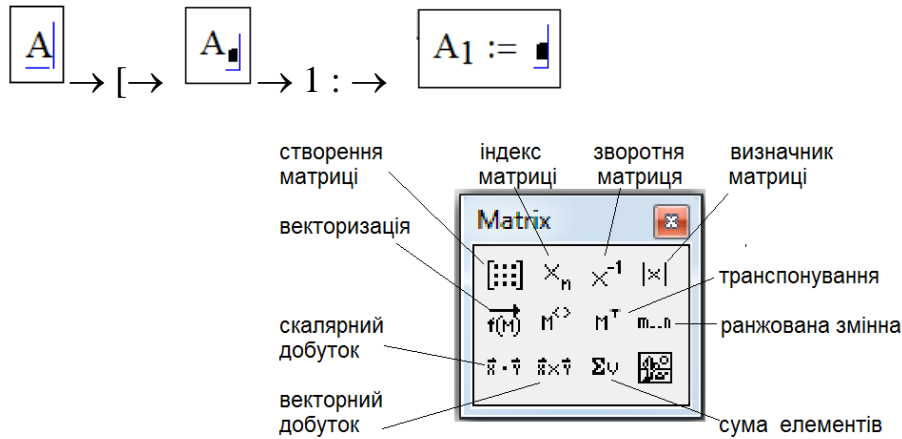



Рис. 2.21. Матрична панель

Примітка. Якщо визначається елемент $a_{i,j}$, то автоматично створюється відповідний масив розміром $i \times j$

$$x_2 := 10 \quad A_{2,3} := 30$$

$$x = \begin{pmatrix} 0 \\ 0 \\ 10 \end{pmatrix} \quad A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 30 \end{pmatrix}$$

2. Визначення вектора (матриці) в діалоговому вікні. Вікно активується клавішами «Ctrl+M» клавіатури, пунктом *Matrix* меню *Insert*, кнопкою  панелі. У вікні потрібно визначити кількість рядків (rows) і стовпців (columns) матриці (рис. 2.22).

Після введення значень у полі документа на місці знаходження курсора відображається структура матриці, яка створюється. На місці маркерів необхідно ввести потрібні значення

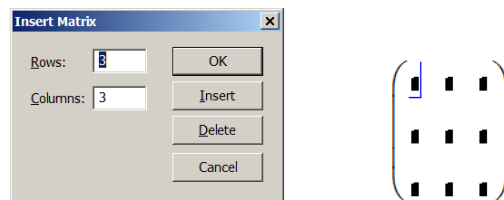


Рис.2.22. Визначення матриць

Таблиця 2.6. Основні матричні операції MathCAD

Оператор	Дія
length(v)	повертає число елементів вектора v
last(v)	повертає індекс останнього елемента вектора v
rows(M)	повертає число рядків матриці M
cols(M)	повертає число стовпців матриці M
M<i>	виділяє i-й стовпець матриці M
identity (n)	формує одиничну квадратну матрицю розміром n x n
diag (v)	формує діагональну матрицю, на головній діагоналі котрої розташовані елементи вектора v.
matrix (m,nf)	формує матрицю, в якій (i,j)-й елемент дорівнює значенню функції f(i,j)
augment (A, B)	формує матрицю приєднанням горизонтально матриць A та B
stack (A, B)	формує матрицю приєднанням після останнього рядка матриці A матрицю B вниз
submatrix (A, ir, jr, ic, jc)	вирізає з матриці A підматрицю з рядками з ir по jr та стовпцями з ic по jc
rank(A)	ранг матриці - кількість лінійно незалежних рядків та стовпців
tr(A)	слід - сума діагональних елементів
mean(A)	середнє значення елементів масиву
norme(M)	евклідова норма матриці $\sqrt{\sum_{i=1}^n \sum_{j=1}^m M_{i,j}^2}$
sort (v)	сортує елементів вектору за зростанням їх значень
reverse (v)	переставляє елементів в зворотному порядку
csort (M,n)	переставляє рядки матриці так, що n-й стовпець сортується за зростанням
rsort (M,n)	переставляє стовпці матриці так, що n-й рядок сортується за зростанням
-A	множить всі елементи A на -1

Таблиця 2.6. Продовження

A^{-1}	зворотня матриця
$ v , A $	довжина вектора v , детермінант матриці A
A^T	транспонування A (Ctrl+1)
A^*	комплексне спряжіння (“)
$\sum v$	сума елементів вектора v (Ctrl+4)

Арифметичні дії з масивами

Над векторами та матрицями можна проводити обчислення, як і над числами. Основні арифметичні матричні функції системи наведено в таблиці 2.7.

Примітка. Треба пам'ятати, що результат арифметичної дії над масивами залежить від типу операндів. Дія з скаляром-числом та масивом проводиться над кожним елементом масиву. Дії над двома масивами виконуються за правилами матричної математики!

Векторизація обчислень

Будь-яке обчислення, яке *MathCAD* може виконати з поодинокими значеннями, він може виконувати поелементно з векторами і матрицями (табл. 2.7). Це можна реалізувати двома способами: послідовно виконуючи дії над кожним елементом масиву або використанням *оператора векторизації*. Для введення цього оператора необхідно над виділеним об'єктом натиснути одночасно клавіші *Ctrl+''* (обрати символ \rightarrow з панелі *Matrix*). Векторизований об'єкт зображується з стрілкою нагорі.

Оператор векторизації змінює зміст операцій. Наприклад, якщо A деяка матриця. Тоді запис $\exp(A)$ є некоректним, оскільки аргументом функції \exp повинна бути проста змінна, а не матриця. Застосування до цієї функції оператора векторизації призводить до обчислення функції \exp від кожного елемента матриці і результатом також є матриця.

Таблиця 2.7. Оператори арифметичних дій з масивами

Позначення: матриця – A, B, M, вектор – v, u, скаляр - k.

Дія	Вигляд	Опис
Множення	$A * k$	Множить кожен елемент A на скаляр k
	$w * v$	Скалярний добуток вектора-рядка на вектор-стовпець
	$A * B$	Матричне множення: рядок – стовпець
	$A * v$	Множення матриця – вектор-стовпець
Ділення	A / k	Ділить кожен елемент масиву на k
Складання	$A + B$	Поелементне складання масивів
	$A + k$	Додавання до кожного елемента A числа k
Віднімання	$A - B$	Поелементне віднімання масивів
	$A - k$	Віднімання з кожного елемента A числа k
Векторне множення	$u \times v$	Добуток векторів [Ctrl+8]. Результат - вектор

Виведення результатів

В системі передбачено відображення масивів в формі матриці та в формі таблиці (рис. 2.23).

$$b = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} +$$

$$a = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 1 \end{array}$$

стиль «матриця»

стиль «таблиця»

Рис. 2.23. Стили виведення масивів

Визначення стилю відображення масивів проводиться пунктом *Matrix display style* на сторінці *Display Options* меню *Format – Result*. В списку знаходяться три стилі *Automatic*, *Matrix*, *Table* (рис. 2.24).

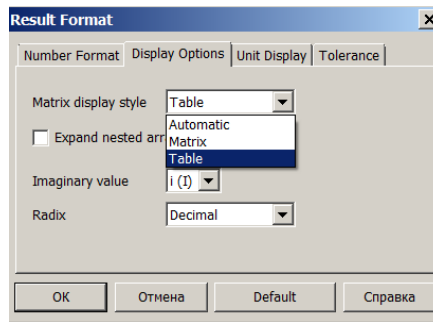


Рис. 2.24. Вікно визначення стилю масивів

Стиль Matrix (Матриця)

Матриця завжди відображується як прийнято з точки зору математики. При цьому видимі всі елементи масиву. Однак, коли масив має великі розміри, то зображення займає дуже багато місця на екрані. Крім того, важко проводити ідентифікацію конкретного елемента масиву, бо номери рядків та стовпців на екрані не відображаються.

Стиль Table (Таблиця)

Матриця завжди відображується як таблиця з виведенням у перший рядок номерів стовпців, в перший стовпець – номери рядків. На екран виводиться тільки частина масиву в діапазоні індексів 1 ... 10. Перегляд частини масиву, що залишилася, проводиться за допомогою вертикального та горизонтального повзунків.

Значення максимального індексу елемента для відображення можна відредагувати в пункті *Properties* контекстного меню масиву на сторінці *Data Range*.

Зображення у вигляді таблиці допускає вирівнювання матриці відносно знаку прирівнювання операцією *Alignment* контекстного меню масиву.

Зображення індексів в таблиці можна відмінити. Для цього в пункті *Properties* контекстному меню масиву на сторінці *Display* слід змінити поле *Show column/row labels*.

Стиль Automatic

Система сама визначає стиль виведення. Для розмірів масиву менше 10 використовується стиль матриця, для більших – таблиця.

За замовчанням встановлено стиль *Automatic*.

Оперативно змінювати стиль виведення масивів можна викликом подвійним щигликом на масиві вікна *Result Format*.

2.5.1. Контрольні питання

Які масиви реалізовано в MathCAD?

Що таке індексовані змінні?

Як створюються масиви в MathCAD?

Що таке векторизація?

В чому особливість арифметичних дій з масивами?

Чим визначається початковий індекс масиву?

2.6. Програмування в MathCAD

Реалізувати алгоритм обчислення в пакеті *MathCAD* можна двома способами:

- вставляючи відповідні оператори чи функції в текст документа *MathCAD*. Такий спосіб називається програмуванням в тексті документа. Він є прийнятним для простих алгоритмів;
- використовуючи підпрограми - функції, які містять конструкції, багато в чому подібні до конструкцій таких мов як PASCAL або C.

В обох випадках реалізуються дії лінійного обчислення, розгалуження, циклічних обчислень.

Програмування із застосуванням підпрограм-функцій реалізує концепцію модульного програмування, коли незалежні алгоритми створюються у вигляді окремих програмних модулів. Застосування його в *MathCAD* дозволяє:

- «розпаралелювати» розробку програм між кількома виконавцями;

- створювати бібліотеки програм для розв'язання науково-технічних задач;
- зменшити витрати на розробку програм для пакета MathCAD.

Застосування підпрограм - функцій в *MathCAD* дозволяє вирішити завдання, які неможливо вирішити, використовуючи тільки оператори і функції *MathCAD*.

2.6.1. Етапи застосування підпрограм-функцій

Застосування підпрограм - функцій подібне до застосування функцій користувача та також включає два етапи: *опис підпрограми-функції*; *виклик підпрограми-функції*.

Опис підпрограми - функції

Перед тим як використовувати підпрограму-функцію, потрібно її задати, тобто *виконати опис*. Опис підпрограми-функції виконується за тими ж правилами, що й опис функції користувача. Відмінністю є те, що список формальних параметрів може бути відсутнім.

Тіло підпрограми-функції включає будь-яке число локальних операторів присвоювання, умовних операторів і операторів циклу, а також викликів інших підпрограм-функцій і функцій користувача. Для введення операторів тіла підпрограми- функції використовується панель *Programming* (рис.2.25). В ній розміщено кнопки введення додаткового рядка підпрограми *Add line*, локального оператора присвоювання, умовного оператора *if* та оператора повного вибору *otherwise*, передумовленого циклу *for*, ітераційного циклу *while*, спеціальні оператори переривання обчислень *break*, *continue*, *return*, *on error*.

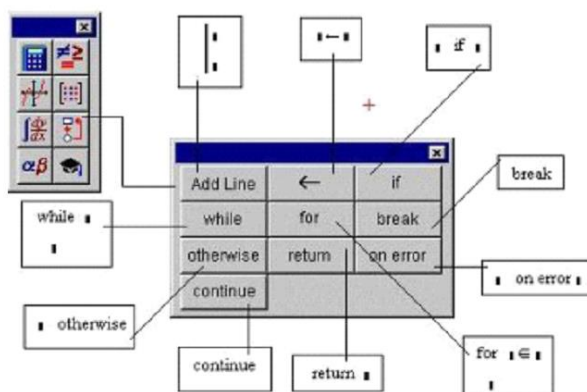


Рис. 2.25. Панель програмування

Коли підпрограма-функція не має формальних параметрів, тоді дані передаються через імена змінних сеансу, визначених вище опису програми-функції. **ВИКОРИСТАННЯ ПІДПРОГРАМ-ФУНКЦІЙ БЕЗ ФОРМАЛЬНИХ ПАРАМЕТРІВ НЕ РЕКОМЕНДУЄТЬСЯ.**

Порядок опису підпрограми-функції MathCAD

Для введення в робочий документ опису програми-функції необхідно виконати наступні дії:

1. Ввести ім'я підпрограми-функції і список формальних параметрів, взятий у круглі дужки;
2. Ввести символ присвоювання;
3. Додати вертикальну лінію зліва від знака присвоювання клавішами "]" або кнопкою "Add line" панелі *Programming*. На екрані з'явиться вертикальна риска і вертикальний стовпець з двома маркерами для введення операторів, що утворюють тіло програми-функції (рис. 2.26).

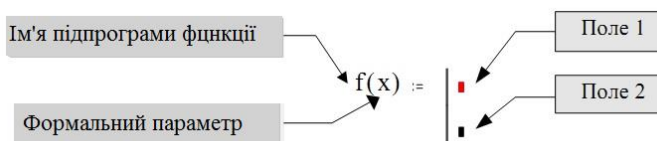


Рис. 2.26. Структура підпрограми-функції

Поля шаблону призначені для розміщення в них відповідних операторів алгоритму. Додавання нових рядків проводиться дією "Add line" клавішами клавіатури або кнопкою панелі *Programming*. При цьому нове поле введення додається внизу виділеного на цей момент оператора. Для видалення поля потрібно помістити на нього покажчик і натиснути клавішу [Delete], для видалення оператора, потрібно виділити його і натиснути клавішу [Delete].

Підпрограма - функція повертає результат останньої виконаної дії. Для усунення неоднозначностей рекомендується в самому нижньому полі розміщувати ім'я результату. Результатом може бути число, вектор, матриця, тобто будь-який тип даних пакета.

У наведеному на рис. 2.27 прикладі формальним параметром є проста змінна x . Підпрограма включає два локальних оператора присвоювання (див.

наступний пункт) і змінну z, яка повертається результатом виконання програми-функції.

$$f(x) := \begin{cases} x \leftarrow x + 2 \\ z \leftarrow x^{\frac{1}{3}} \\ z \end{cases}$$

Рис. 2.27. Приклад структури підпрограми-функції

Локальний оператор присвоювання - використовується для завдання всередині підпрограми значення будь-якої змінної:

<ім'я змінної> ← <вираз>

Увага! Використання "звичайного" оператора присвоювання (позначається: =) в тілі програми-функції призводить до синтаксичної помилки.

Звернення до програми-функції MathCAD

Для виконання підпрограми-функції необхідно звернутися до імені підпрограми-функції з вказівкою списку фактичних параметрів (якщо в описі підпрограми присутній список формальних параметрів).

Співвідношення між фактичними і формальними параметрами для підпрограми – функції таке ж, як для функцій користувача.

Звернення до підпрограми-функції має відбуватися після опису підпрограми-функції.

Наприклад, звернення до програми f(x), наведеної на рис. 2.27, може мати наступний вигляд:

$$\begin{aligned} x &:= 2 & f(x) &= 1.587 & f(-3.23) &= 0.536 + 0.928i \\ z &:= f(x + 4.5) & z &= 2.041 & & \blacksquare \end{aligned}$$

Передати дані всередину підпрограми - функції можна використанням всередині підпрограми глобальних змінних, які визначені в поточному документі *MathCAD* до опису підпрограми-функції.

Наприклад (рис. 2.28):

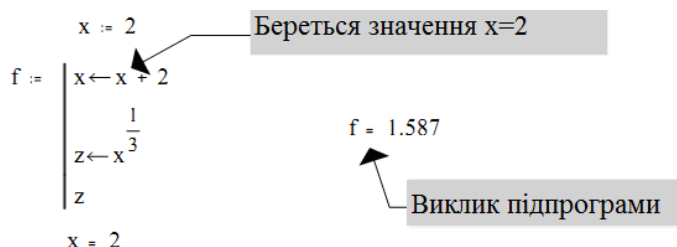


Рис. 2.28. Локальні та глобальні змінні

Хоча значення змінної «x» змінилося всередині підпрограми-функції, поза описом підпрограми-функції ця змінна зберігає своє колишнє значення, тому що використовувався локальний оператор присвоєння.

2.6.2. Різновиди алгоритмів в підпрограмах-функціях

Лінійні алгоритми

Під лінійним алгоритмом розуміється процес, в якому необхідні операції виконуються суворо послідовно. Оператори, що реалізують цей алгоритм у тілі підпрограми - функції також розміщуються послідовно і виконуються всі, починаючи з першого оператора і закінчуючи останнім.

Наприклад. Для створення підпрограми-функції обчислення кореня квадратного рівняння $ax^2 + bx + c = 0$ за формулою

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ потрібно:}$$

1. Ввести назву підпрограми $quad(a,b,c)$, поставити знак прирівнювання «:=», створити заготовку для тіла підпрограми. Кнопками панелі *Programming* або клавішами «]» додати 3 рядки (рис. 2.29).

```
quad(a, b, c) := |
                  |
                  |
                  |
                  |
                  |
```

Рис. 2.29. Заготовка підпрограми

2. Ввести рядки тіла. Обчислення детермінанту та присвоєння його значення змінній D . Обчислення знаменника формули та присвоєння його значення змінній Z . Обчислення кореню (рис. 2.30).

```
quad(a, b, c) := |
                  | D ← b2 - 4·a·c
                  | Z ← 2·a
                  |  $\frac{-b + \sqrt{D}}{Z}$ 
                  |
```

Рис. 2.30. Підпрограма **quad**

Програма **quad** має три параметри: коефіцієнти рівняння. Програма реалізує лінійний алгоритм - всі оператори виконуються завжди суворо послідовно.

Алгоритми, що розгалужуються

У розгалужених алгоритмах присутні кілька гілок обчислювального процесу. Вибір конкретної гілки залежить від виконання (чи невиконання) заданих умов.

Наприклад, функція $y(x)$ задається наступним виразом

$$y(x) = \begin{cases} x^2, & x \leq 0; \\ \sqrt{x}, & x > 0. \end{cases} \quad (2.6.1)$$

Видно, що алгоритм обчислень містить дві гілки і вибір залежить від значення змінної x .

Для програмування алгоритмів, що розгалужуються підпрограми - функції в MathCAD використовується умовний оператор.

Умовний оператор

Цей оператор використовується тільки в тілі програми-функції і для його введення необхідно клацнути на кнопці «*if*» панелі *Programming* або

клавіші "}". На екрані з'являється конструкція з двома полями введення, яка зображена на рис. 2.31.

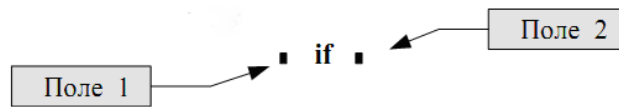


Рис. 2.31. Шаблон оператора if

В поле 2 вводиться логічний вираз (у найпростішому випадку це вираз відношення). В поле 1 вводиться вираз (як правило, арифметичне), значення якого використовується, якщо логічний вираз приймає значення 1.

Умовний оператор реалізує «неповну» альтернативу. Тобто вказує, що потрібно виконати при виконанні умови. Для «повної» альтернативи застосовується оператор *otherwise*.

Вираз, що стоїть перед словом *otherwise*, виконується тільки в тому випадку, якщо не виконано задане перед цією умовою.

В програмі можна використовувати кілька умовних операторів в сусідніх рядках з одним виразом перед словом *otherwise*.

Умовний оператор може знаходитися тільки всередині тіла програми-функції (рис. 2.32).

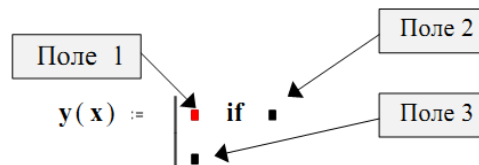


Рис. 2.32. Структура умовного оператора

В поле 3 задається вираз, значення якого використовується, якщо логічний вираз дорівнює 0, наприклад - оператор *otherwise*. В поле 1 може бути розміщено декілька рядків аналогічно блоковому оператору *begin - end* мови Паскаль. Для цього слід застосувати операцію вставки рядка "Add line".

Наприклад, обчислення функції $y(x)$ (2.6.1) в підпрограмі можна виконати з використанням умовної функції (рис. 2.32а) або з умовним оператором (рис. 2.32б).

$$y(x) := \begin{cases} z \leftarrow \text{if}(x \leq 0, x^2, \sqrt{x}) \\ z \end{cases} \quad y(x) := \begin{cases} x^2 & \text{if } x \leq 0 \\ \sqrt{x} & \text{otherwise} \end{cases}$$

а б

Рис. 2.32. Варіанти розгалуження алгоритму в підпрограмі – функції:

Наприклад, підпрограма-функція для обчислення функції $z(t)$ за формулою

$$z(t) = \begin{cases} t^3, & t < -3; \\ \ln(t), & t > 4; \\ t^2, & -3 \leq t \leq 4. \end{cases}$$

може мати наступний вигляд

$$z(t) := \begin{cases} t^3 & \text{if } t \leq -3 \\ t^2 & \text{if } -3 \leq t \leq 4 \\ \ln(t) & \text{otherwise} \end{cases}$$

Зауваження. Якщо в поле 3 ввести оператор $\ln(t)$ без слова *otherwise*, то цей оператор буде виконуватися завжди незалежно від виконання вище заданих умовних операторів. Для вірної відповіді слід ввести оператор

`ln(t) if t > 4`

Циклічні алгоритми

Циклічні алгоритми (або цикли) містять повторювані обчислення, що залежать від деякої змінної. Така змінна називається параметром циклу, а самі повторювані обчислення складають тіло циклу.

Програмування зумовленого циклу

Для програмування таких циклів використовується оператор циклу *for*.

Введення оператора проводиться клавішами *Ctrl + Shift + "*, кнопкою *for* панелі *Programming* (рис. 2.33).

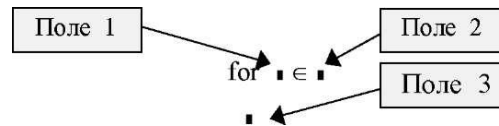


Рис. 2.33. Структура оператора циклу for

В поле 1 вводиться ім'я змінної лічильника циклу, в поле 2 - діапазон значень параметра циклу. **Діапазон значень вводиться аналогічно діапазону ранжованої змінної.** В поле 3 вводяться оператори, що складають тіло циклу. Якщо одного рядка недостатньо, то додаткові поля введення (додаткові рядки) створюються процедурою "Add line".

Наприклад. Функція form_tab1, яка для x , що змінюється від -2 до 2 з кроком 0.5, розраховує значення $f(x) = e^x \cdot \cos(2x)$ у вигляді вектора y ($y_1 = f(-2)$, $y_2 = f(-1.5)$ і т.д.), може мати наступний вигляд:

```

form_tab1(x_start, x_stop, step) :=
  i ← 1
  for x ∈ x_start, x_start + step .. x_stop
    y_i ← ex · cos(2 · x)
    i ← i + 1
  y

```

У цьому варіанті опису програми-функції формальні параметри використовуються для визначення діапазону зміни параметра циклу (змінна x). Для зміни індексу елемента масиву «у» вводиться змінна « i » цілого типу всередині циклу. Звернення до описаної підпрограми-функції може мати наступний вигляд

```
z := form_tab(-2, 2, 0.5)
```

Кількість повторень визначається за формулою

$$N = \frac{x_k - x_0}{d} + 1 = \frac{2 - (-2)}{0.5} + 1 = 9,$$

де x_k , x_0 - кінцеве і початкове значення параметра циклу, d - крок його зміни.

Опис програми-функції з циклом по кроках від 1 до N має наступний вигляд :

$$\text{form_tab}(x_{\text{start}}, x_{\text{stop}}, \text{step}) := \left\{ \begin{array}{l} N \leftarrow \frac{x_{\text{stop}} - x_{\text{start}}}{\text{step}} + 1 \\ \text{for } i \in 0..N - 1 \\ \quad \left\{ \begin{array}{l} x \leftarrow x_{\text{start}} + i \cdot \text{step} \\ y_i \leftarrow e^x \cdot \cos(2 \cdot x) \end{array} \right. \end{array} \right.$$

Зауваження. Якщо значення індексів у елементів масиву змінюється починаючи з 1 (як в цьому прикладі), то початкове значення індексу необхідно встановити рівним 1 (системною змінною *ORIGIN* або пунктом *Origin* меню *Tools – WorkSheet Option – Build in variables*).

Програмування ітераційних циклів

Для програмування таких циклів використовується оператор циклу *while*.

Оператор циклу *while* виконується в такий спосіб: по-перше MathCAD перевіряє зазначену умову. Якщо вона істинна, то виконується тіло циклу і знову перевіряється умова. Якщо вона хибна, то цикл закінчується. Для введення цього оператора необхідно клацнути на кнопці *while* панелі *Programming*. На екрані з'являються елементи, які показані на рис. 2.34.

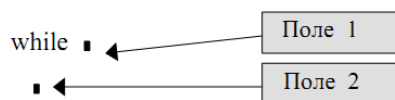


Рис. 2.34. Структура оператора циклу *while*

Поле 1 призначено для розташування умови виконання циклу, поле 2 - для операторів тіла циклу.

В тілі циклу повинні бути присутніми оператори, які роблять умову циклу хибною, інакше цикл буде продовжуватися нескінченно.

Організація ітераційного циклу за допомогою оператора *while*, без додаткових засобів контролю може привести до зациклення, коли умова не буде виконана ніколи.

В *MathCAD* є декілька операторів, які призначені розв'язання таких ситуацій: вихід з циклу *break*, продовження *continue*, переривання *return*, похибки *on error* .

Примітка! Необхідність застосування оператора break скоріш за все говорить про недолугий алгоритм розв'язання задачі.

Приклад. Підпрограма-функція, що реалізує ітераційну процедуру наближеного обчислення кореня квадратного з числа x з точністю ε за

ітераційним правилом $x_{i+1} = \frac{x_i + \frac{x}{x_i}}{2}$, $x_1 = x$, з умовою закінчення розрахунку - $|x_{i+1} - x_i| < \varepsilon$ може мати наступний вигляд

```

sqrt(x, a) :=
  first ← x
  sec ← (first + x / first) / 2
  while |sec - first| > a
    first ← sec
    sec ← (first + x / first) / 2

```

Як видно з тексту програми-функції немає необхідності зберігати в пам'яті все при наближенні значення x_0, x_1, x_2, \dots, i т.д.. Досить зберігати два: попереднє ("старе") значення *first* і наступне *sec*.

Звернення до описаної програми буде мати вигляд

sqrt(9, 0.0001)=3 sqrt(25, 0.0001)=5 .

Вихід з циклу

Оператор *break* дозволяє вийти з циклу. Застосовується разом з умовним оператором. При виконанні заданої умови цикл переривається, керування передається наступному за циклом рядку.

Введення оператора *break* проводиться клавішами *Ctrl+{* або кнопкою *break* панелі *Programming* (**не можна вводити оператор break з клавіатури по символам**). Оператор *break* розташовується в лівому полі введення умовного оператор *if*, а в правому розміщується умова, при виконанні якої відбувається припинення роботи циклу або програми.

Наступний приклад показує написання програми прикладу з оператором *break*.

```

sqrt(x, a) :=
  first ← x
  sec ← (first + x / first) / 2
  while 1
    first ← sec
    sec ← (first + x / first) / 2
    break if |first - sec| < a

```

У цій програмі застосовано нескінчений цикл. Для виходу з циклу за умови $|x_{i+1} - x_i| < \varepsilon$ або в термінах програми - $|sec - first| < a$, використано оператор *break*.

Оператор продовження

Зазвичай оператор *continue* використовується для продовження виконання циклу шляхом повернення на початок тіла циклу без виконання рядків, які розташовані після оператора *continue*. Наступний приклад пояснює роботу цього оператора.

Наприклад, потрібно сформувати з вихідного вектору x результуючий вектор y . Вектор y повинен містити тільки додатні елементи вектору x .

РОЗВ'ЯЗАННЯ

Підпрограма може мати наступний вигляд:

```

positiv(x) :=
  ix ← -1
  iy ← 0
  while ix < last(x)
    ix ← ix + 1
    continue if xix ≤ 0
    yiy ← xix
    iy ← iy + 1
  y

```

В тілі програми-функції використовується функція **last(x)**, яка визначає індекс останнього елемента вихідного масиву x . Якщо черговий елемент x_i від'ємний, то пропускаються всі оператори тіла циклу нижче умови (оператори, що обраховують елемент результуючого вектора y) і тіло циклу повторюється при новому значенні параметра циклу i .

Оператор переривання

Оператор *return* <Вираз 1> перериває виконання програми-функції і повертає значення операнда <Вираз 1>, що стоїть за ним. *Наприклад*, програма для знаходження першого додатного елемента масиву *x* може мати наступний вигляд:

```
positiv(x) := for i ∈ 1..last(x)
             return xi if xi > 0 .
```

Оператор помилки

Оператор *on error* є засобом обробки помилок, які виникають при виконанні тих чи інших обчислень, і записується у вигляді:

<Вираз 1> on error <Вираз 2> .

Оператор виконує <вираз 2>. Якщо при виконанні <виразу 2> виникає помилка, то виконується <вираз 1>.

Наприклад, обчислення функції $\frac{\sin(x)}{x}$ може мати наступний вигляд

```
s(x) := 1 on error  $\frac{\sin(x)}{x}$  .
```

2.6.3. Модульне програмування MATHCAD

Загальна ідея модульного програмування полягає в наступному:

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

Модульне програмування дозволяє зменшити обсяг вихідних текстів програм, зробити їх більш чіткими, пришвидшити написання і тестування програм, зменшити витрати на супровід (експлуатацію) програм.

Модульне програмування в пакеті *MathCAD* можна реалізувати двома методами:

- модульне програмування в межах одного документа *Mathcad*;
- модульне програмування в декількох документах *Mathcad*.

Модульне програмування в одному документа

Цей метод характеризується тим, що опис підпрограм-функцій і їхній виклик (тобто звернення до них) знаходяться в межах одного документа і всі описи зберігаються в одному файлі. При цьому всередині однієї підпрограми-функції можуть знаходитися виклики локальних функцій, вбудованих функцій *MathCAD* і інших підпрограм-функцій.

Модульне програмування в декількох документах MathCAD

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

- неможливість паралельної розробки програм декількома розробниками;
- неможливість "автономного" налагодження підпрограм-функцій і їхньої модифікації в процесі експлуатації програмного забезпечення;
- неможливість використання розробленої підпрограми-функції в декількох документах без дублювання опису підпрограми-функції.

Для подолання цих недоліків опис підпрограми-функції виконують в одному документі *MathCAD*, а її виклик розміщується в іншому документі. При цьому виникає питання: як при виклику підпрограми-функції в одному документі "приєднати" файл з іншого документа *MathCAD*, в якому знаходиться опис потрібної підпрограми-функції? Для такого приєднання служить спеціальний оператор *Reference*, який записується у наступному вигляді (рис. 2.35).

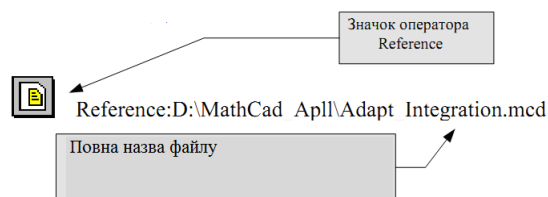


Рис. 2.35. Структура оператора *Reference*

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

1. Встановити курсор на потрібне місце, куди буде вставлений оператор *Reference*.

2. Обрати опцію *Reference* меню *Insert*. У поле введення діалогового вікна ввести повне ім'я файлу, що містить документ з описом шуканої функції. Для задання імені можна натиснути кнопку *Browse* і в діалоговому вікні вказати диск, папку та ім'я файлу.

3. Після виконання цих кроків у документі з'явиться оператор *Reference*, показаний на рис. 2. 35.

Таким чином, реалізація модульного програмування в декількох документах MathCAD включає наступні етапи:

- Опис в документі *MathCAD* необхідних підпрограм-функцій і збереження цього документа у файлі в потрібній папці і з потрібним ім'ям.
- Вставлення оператора *Reference* в документ, в якому будуть використовуватися описані підпрограми-функції.
- Виклик потрібних підпрограм-функцій.

2.6.4. Контрольні питання

Які способи програмування реалізовані в *MathCAD*?

В чому різниця між функцією користувача та підпрограмою-функцією?

Які правила опису підпрограми-функції?

Як реалізувати розгалуження в підпрограмі функції?

Які різновиди циклів є в підпрограмі-функції?

Які способи переривання дій існують в підпрограмі?

Що таке зовнішнє посилання?

Які недоліки програмування в одному файлі?

2.7. Графічне оформлення результатів

На практиці виникають задачі, які потребують візуалізації даних в графічному вигляді. *Наприклад:*

- огляд характерних точок функції для експрес – аналізу результатів,

- дослідження функції для виявлення характерних точок, точок перетину осі абсцис тощо,
- аналіз вихідних табличних даних, знаходження по табличним даним наближаючої аналітичної залежності, оцінка ступеню наближення.

В систему вбудовані засоби для виведення графічної інформації трьох видів: двомірних графіків, тримірних графіків, малюнків. Для поліпшення візуального сприйняття можливо зі статичних графіків зробити анімацію.

Для графічної інформації призначено спеціальну графічну область.

Керування графічною областю здійснюється за допомогою “гарячих клавіш”, через пункт меню *Insert - Graph* або через кнопку панель графіки (рис. 2.36).

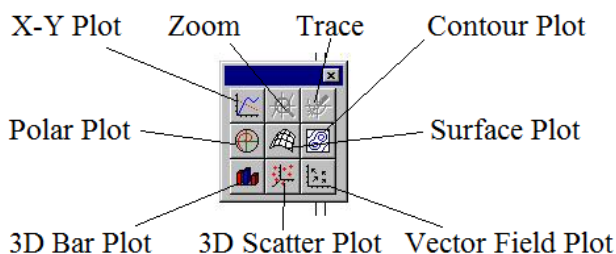


Рис. 2.36. Панель графіки

Основні види графіків

X-Y Plot (клавіша [*@*]). Побудова графіка функції $y = f(x)$ в декартових координатах.

Polar Plot (клавіші [*Ctrl+7*]). Побудова графіка функції $\rho = f(\varphi)$ в полярних координатах.

Surface Plot (клавіші [*Ctrl+2*]). Побудова графіка 3D функції $z = f(x, y)$ у вигляді поверхні.

Contour Plot (клавіші [*Ctrl+5*]). Побудова діаграми ліній рівня функції $z = f(x, y)$. Відображає точки, в яких функція приймає фіксовані значення $z = \text{const}$.

3D Scatter Plot. Будує 3D точкове зображення матриці або функції $z = f(x, y)$ у заданих точках (x_i, y_j) . Застосовується для побудови просторових кривих.

3D Bar Plot. Будує 3D зображення матриці або функції $z = f(x,y)$ у вигляді стовпкової діаграми.

Vector Field Plot. Слугує для зображення двомірних векторних полів.

2.7.1. Двомірні графіки

Плоскі графіки призначені для зображення функцій, виразів одного аргументу, векторів. Система будує графіки функцій однієї змінної в декартових $y=f(x)$ та полярних $\rho=f(\varphi)$ координатах, де $y = \rho \sin(\varphi)$, $x = \rho \cos(\varphi)$.

Залежності можуть бути записані в абсолютній або в параметричній формі $y = f(t)$, $x = \varphi(t)$ як залежності від однієї спільної змінної.

Графіки в декартових координатах

Активація графічної області плоских декартових графіків виконується натисканням клавіш Shift+@. При цьому на екрані відображується умовне поле графіка у вигляді рамки й прямокутників біля осей графіка для введення значень (рис. 2.37).

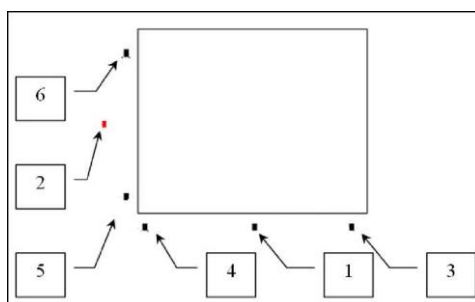


Рис. 2.37. Маркери області двомірної декартової графіки

Середні маркери призначені для введення ранжованої змінної-аргументу по осі абсцис (поле 1) та функції/й – користувача, виразів, які необхідно відобразити, по осі ординат (поле 2). Для побудови графіків декількох функцій їхні імена вводяться в поле 2 через кому.

Крайні маркери призначені для визначення границь графіків: маркери 3,4 визначають межі аргументу, маркери 5,6 - функції. Якщо межі графіка не задані, пакет пропоставляє їх автоматично. За замовчанням графік будується в діапазоні аргументу від -10 до 10 із кроком 1.

Обов'язковим є заповнення тільки поля 2. Введення тільки імені функції в поле 2 забезпечує «швидку» побудову графіка з параметрами за замовчанням.

Наприклад, для того щоб відобразити графік функції $y=e^{-x}\cdot\cos(5x)$ у діапазоні $x \in [0, 5]$ із кроком 0.1, потрібно:

1. Визначити функцію користувача $y(x):=\exp(-x)\cdot\cos(5\cdot x)$.

2. У потрібному місці вставити графічну область декартового графіка знаком “@” або іншим способом. У поле 2 ввести ім'я функції $y(x)$. Графік набуде вигляд (рис. 2.38). Діапазон аргументу -10...10 та відповідні значення в полях 5,6: -20 та 20,- система визначила автоматично.

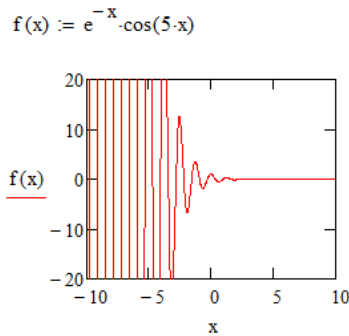


Рис. 2.38. Графік за замовчанням

3. Для зміни меж аргументу слід змінити значення маркерів 3,4 на 0 та 0.5 відповідно. Для зміни меж значень графіка по осі ординат – маркера 5 на -0.5. Графік набуде вигляд (рис. 2.39).

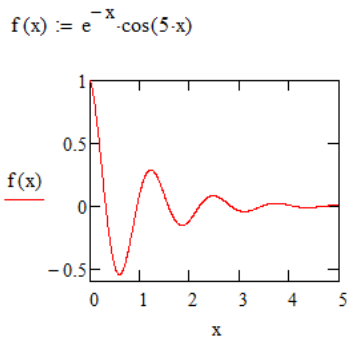


Рис. 2.39. Відкоригований графік

Примітка. Для усунення необхідності додаткових дій доцільно визначати діапазон змін аргументу через ранжовану змінну.

В області двомірних графіків можна візуалізувати табличні (векторні) дані та будувати *параметричні графіки*. При роботі з векторами аргументом графіка є змінна індексу вектора або один із векторів. Другий вектор є ординатою графіка.

На *параметричному графіку* по осях координат задаються дві функції, які залежать від одного аргументу. Використання замість функцій векторів дозволяє побудувати зображення точкової діаграми. Один вектор повинен містити координати x точок, другий – координати y .

Параметричні графіки зручно використовувати для побудови графіків функцій, які мають декілька значень функції при одному аргументі. Зазвичай такі функції описують замкнені геометричні фігури на кшталт кола.

Наприклад, система зображує в декартових координатах тільки одну гілку графіка (рис.2.40 а) функції $y = \sin\left(3\frac{\arcsin(x)}{2}\right)$. Якщо записати цю функцію в параметричному вигляді $a := \sin(2t)$ $b := \sin(3t)$, то графік стає значно інформативнішим (рис.2.40 б).

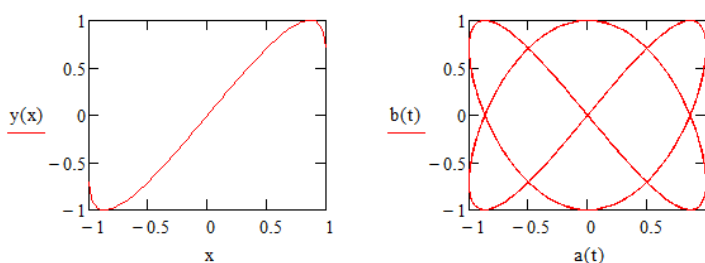


Рис. 2.40. Графік а - абсолютної функції, б - параметричної функції

Модифікацію плоского графіка можна провести за допомогою локального форматування двома способами: двократним клацанням кнопки миші при розташуванні курсору в області графіка або через меню «*Format - Graph*». В обох випадках активізується меню плоских графіків (рис. 2.41).

Вікно меню графіків має наступні вкладники:

X-Y Axes – визначає вигляд осей графіка;

Traces – визначає вигляд ліній самого графіка;

Labels - визначає вигляд пояснюючих написів по осях графіка;

Defaults - визначає значення за замовчанням;

Number Format – визначає формат виведення чисельних значень на графіку.

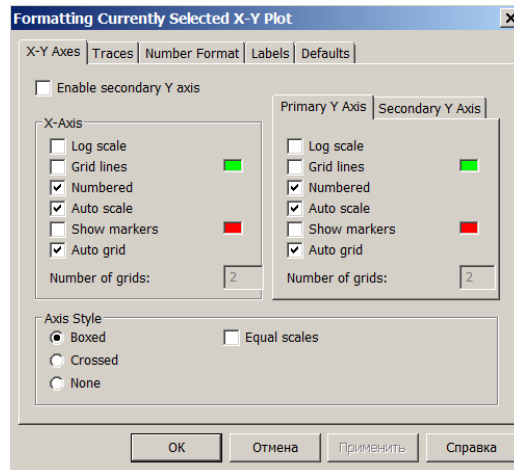


Рис. 2.41. Вікно форматування двовірних графіків

Поля закладки *X-Y Axes*:

- Enable second Y axis - додає другу вісь «Y» для відображення двох графіків з різними значеннями ординат. При малюванні кількох графіків перший малюється відносно першої осі, всі інші – відносно другої.
- Log Scale - задає логарифмічний масштаб осей. Масштаб задається по кожній осі незалежно.
- Grid Lines - керує видимістю допоміжних ліній сітки на полі графіка.
- Show markers - додає одну або дві лінії сітки з визначеним положенням.
- Numbered - керує виведенням проміжних чисел усередині діапазону по лініях сітки.
- Autoscale - задає режим автоматичного масштабування графіка.
- Autogrid - задає автоматичне нанесення сітки.
- Axis Style - керує виглядом осей: **Boxed** – прямокутник навколо графіка; **Crossed** – осі у вигляді хреста; **None** - відсутність осей; **Equal scale** – осі рівного масштабу.

Поля вкладники *Labels*: *Title* – задає титульний напис до графіка; *X-Axis* - задає напис осі X; *Y-Axis* - задає напис осі Y.

Поля вкладники *Traces*:

- Legend - керує виведенням опису ліній графіка.
- Symbol - задає символ, що буде зображено на лінії графіка: хрест, коло, квадрат.
- Symbol - задає частоту повторення символів. Значення показує

Frequency	кількість точок графіка між символами.
Symbol Weight	- задає товщину символів
Line	- задає візерунок лінії: суцільна, точкова, штрихова і т.д.
Color	- задає колір лінії.
Type	- задає тип лінії графіка: звичайна лінія (line), стовпова діаграма (bar), лінійна діаграма (stem), лінія з точок (points), лінія сходів (step). Для двох ліній можна застосувати тип “похибка”. При цьому на графіку буде заштриховано область, що окреслена лініями графіка. Тип " draw " дозволяє малювати графік без продовження до осей графіка.
Weight	- задає товщину лінії.

Графіки в полярних координатах

Активація графічної області плоских полярних графіків виконується натисканням клавіш **Ctrl+7**. При цьому на екрані відображується умовне поле графіка у вигляді рамки з маркерами біля контурного кола для введення значень (рис. 2.42). Нижній маркер 1 призначено для введення імен ранжованої змінної - аргументу, лівий 2 – для введення функцій/виразів, графіки яких необхідно відобразити. За замовчанням приймається, що діапазон зміни аргументу становить $0 - 2\pi$. Маркери 3, 4 призначені для обмежень значень ординат по радіусу: 3 – максимальне значення радіуса, 4 – мінімальне.

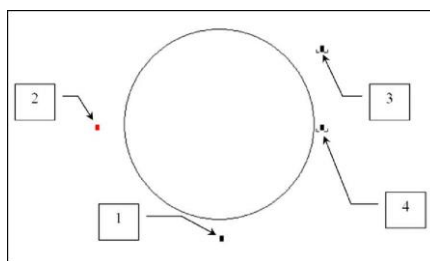


Рис. 2.42. Шаблон графіка в полярних координатах

Модифікацію графіка в полярних координатах можна провести за допомогою локального форматування аналогічно форматуванню графіка в декартових координатах. Параметри графіка редагуються у діалоговому вікні полярних графіків.

Вікно має наступні вкладники: *Polar Axes*, *Traces*, *Labels*, *Defaults*, *Number Format*. Зміст дій та поля вкладинок в цілому співпадають з полями вікна декартових графіків.

Show markers – за документацією виробника повинно виводити додаткові лінії сітки по визначених двох додаткових радіусах. На практиці не працює.

Show negative radii – дозволяє використовувати вирази з від’ємним значенням радіуса.

Наприклад, для того щоб відобразити графік функції

$$r(\varphi) = 1 + \frac{\sin(3\varphi + \pi)}{2} \quad \text{в полярних координатах, слід:}$$

- Визначити функцію користувача $r(\varphi)$;
- У потрібному місці вставити графічну область полярного графіка клавішами “Ctrl+7” або іншим способом. У поле 2 ввести ім’я функції $r(t)$. Графік набуде вигляд, зображений на рис. 2.43. Діапазон аргументу $0 \dots 2\pi$ та відповідні значення в полі 1 система визначила автоматично.

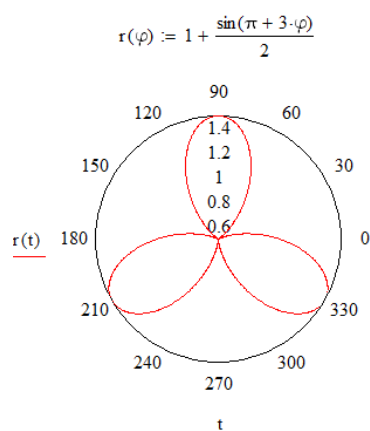


Рис. 2.43. Графік в полярних координатах

Для спрощення розрахунків можна застосовувати функції переведення декартових координат в полярні та навпаки: **xy2pol (<arg>)** та **pol2xy (<arg>)**. Аргументом можуть бути два числа – координати x, y /радіус, кут або двоелементний вектор. Перший елемент вектора містить значення x /радіуса, другий – y /кута в радіанах. Результатом є вектор

полярних координат/декартових координат. Перший елемент вектора містить значення радіуса/х, другий – кута в радіанах/у.

Наприклад, звернення до функції з аргументами «1, 1» або $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ виведе

результат $\text{xy2pol}(1,1) = \begin{pmatrix} 1.414 \\ 0.785 \end{pmatrix}$.

Додаткові засоби роботи з плоскими графіками

Логарифмічний масштаб осей

Логарифмічний масштаб слушно використовувати:

- коли необхідно мати детальну інформацію про всі ділянки графіка, діапазон змін аргументу або значень функції дуже великий;
- коли на одному полі необхідно відобразити графіки, які мають дуже різні значення.

Логарифмічний формат задається для тої осі, по якій очікується великий розбіг значень. В логарифмічному масштабі розтягується область маленьких значень та звужується для великих.

Приклад застосування логарифмічних масштабів наведено в п. 3.6.

Для спрощення форматування даних в логарифмічному масштабі призначені функції **logspace**, **logpts**.

Функція **logspace(min, max, npts)** повертає вектор, **npts** елементів якого розподілені за логарифмічному закон від **min** до **max**. Наприклад, виклик **logspace(0.1, 1, 9)** генерує вектор [0.1, 0.132, 0.173, 0.228, 0.3, 0.395, 0.52, 0.684, 0.9].

Функція **logpts(minexp, dec, dnpts)** - повертає вектор, **dec · dnpts** елементів якого лінійно розподілені від 10^{minexp} на **dec** декад з **dnpts** інтервалами на декаду. Функція не розраховує останню точку декади. Наприклад, **logpts(-1, 1, 9)** генерує вектор [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]. Початок $10^{-1}=0.1$, одна декада від 0.1 до 1 з дев'ятьма точками. Виклик **logpts(-1, 1, 10)** генерує не [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0], а [0.1, 0.19, 0.28, 0.37, 0.46, 0.55, 0.64, 0.73, 0.82, 0.91].

Приклад дії функцій наведено на рис. 2.44.

```

Y := logspace(0.1, 1000, 10)   X := logpts(1, 2, 5)
YT = (0.1 0.278 0.774 2.154 5.995 16.681 46.416 129.155 359.381 1 × 103)
XT = (10 28 46 64 82 100 280 460 640 820)

```

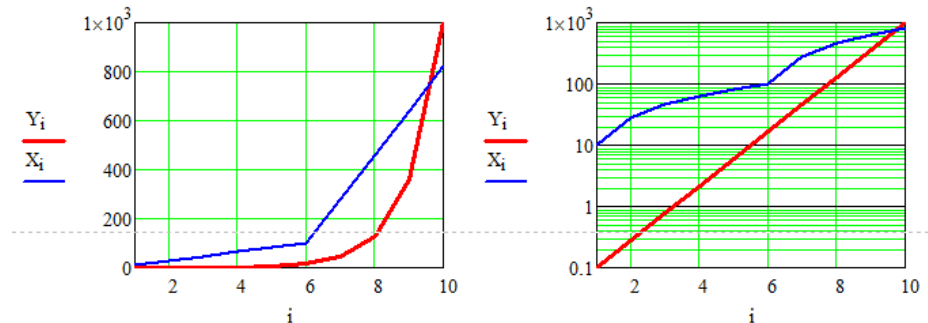



Рис. 2.44. Дія функцій `logspace`, `logpts`

Детальний кількісний аналіз графіків

Для дослідження поведінки графіків та отримання поточних кількісних даних зручно використовувати режими «збільшення» (*Zoom*) та «аналіз» (*Trace*). Доступ до цих режимів реалізовано через кнопки панелі *Graph* та поля *Zoom*, *Trace* контекстного меню графічної області. Керування режимами здійснюється через відповідні діалогові вікна (приклад 3.6.2).

Для визначення області збільшення можна задати її координати в полях «x», «y» у вікні збільшення або просто відмітити прямокутну область на графіку мишею з натиснутою лівою кнопкою. При роботі мишею координати синхронно відображаються в відповідних полях вікна збільшення. Натискання кнопки «ОК» призводить до перемалювання графіка в новому масштабі. Завжди можна повернутися до первісного вигляду графіка натисканням кнопки  у вікні збільшення.

У вікні трасування синхронно з рухами миші по полю графіка відображаються її координати. Для того, щоб миша рухалася тільки по лінії графіка слід увімкнути перемикач «*Track data points*» у вікні трасування.

2.7.2. Тримірні графіки

Активація графічної області 3D графіків здійснюється через пункт меню «*Insert - Graph*» або натисненням клавіш CTRL+2.

У місці маркера графіка для побудови 3D графіка необхідно ввести ім'я матриці/матриць, для якої передбачається відображення у вузлах, або ім'я функції/функцій двох змінних.

Зображення можна повернути та змінити його масштаб. Зсув миші з натиснутою лівою кнопкою синхронно повертає зображення. Зсув миші з натиснутими лівою кнопкою та клавішею «Ctrl» або поворот колеса миші наближає або віддаляє зображення.

Керування виглядом просторових графіків

Виклик вікна зміни параметрів 3D графіка (рис. 2.45) проводиться подвійним щикликом лівої кнопки миші на полі графіка або з меню пакета «*Format - Graph*».

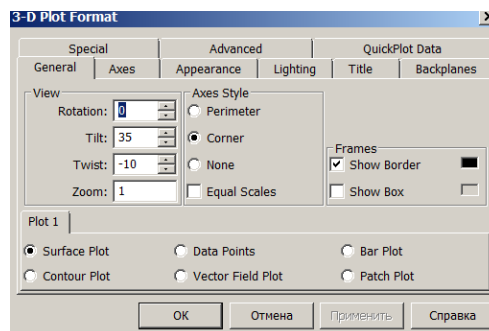


Рис. 2.45. Вікно форматування просторових графіків

Вікно має наступні вкладинки:

General – визначення точки зору, виду графіка;

Axis – визначення вигляду координатних осей;

Appearance - визначення кольорів, вигляду точок графіка;

Lighting - визначення додаткового освітлення;

Title – визначення титульних написів;

Backplanes - визначення сітки, кольору фонових площин;

Special – визначення типу ліній, контурних ліній, стовпових діаграм;

Advanced - установки прозорості, «задимлення», якості друку;

QuickPlotData - зміна діапазонів аргументів, системи координат.

Поля вікна тримірної графіки

- General** *Rозділ View* задає орієнтацію та тип зображення.
- Rotation** - поворот. Задає кут повороту осей ХУ горизонтальної площини ХОУ відносно вертикальної осі Z у діапазоні $-360^0 \dots 360^0$. Нуль градусів відповідає YOZ площині, дев'яносто градусів - ХOZ.
- Tilt** - нахил. Задає кут підвищення спостерігача відносно горизонтальної площини ХОУ у діапазоні $0^0 \dots 180^0$. Нуль градусів відповідає горизонтальній проекції площини (вид збоку), дев'яносто градусів - вертикальній проекції (вид згори).
- Twist** – кручення. Задає кут повороту осі Z у діапазоні $-360^0 \dots 360^0$ після повороту та нахилу.
- Zoom** – масштаб у діапазоні 0.0001 ... 10000.
- Rозділ Axes Style* визначає стиль осей та рамки області:
- Perimeter** - по периметру; **Corner** – осі в куті, **None** - без осей, **Equal Scales** - рівні масштаби по всім осям.
- Rозділ Frames* визначає рамку навколо графіка:
- Show Border** – прямокутна рамка навколо осей, **Show Box** – паралелепіпед навколо графіка.
- Rозділ Plot* визначає тип графіка: **Surface Plot** - графік поверхні, **Contour Plot** - графік ізоліній (контурний вид знизу), **Data Points** – точковий графік, **VectorField Plot** - графік векторного поля, **Bar Plot** – діаграма, **Patch Plot** – діаграма рівнів.
- Appearance** *Fill options* – керує фарбуванням графіка:
- Fill Surface** – заповнення кольором за схемою з вкладники **Color Options**, **Fill Contours** - заповнення кольором у напрямку з вкладники **Special**, **No Fill** – без фарбування, **Alternate Mesh** – заповнення кольоровими трикутниками, **Smooth Shading** – покращене фарбування.
- Lines options* - керує видом відображення ліній поверхні:
- Weight** – товщина ліній, **Wireframe** - у вигляді каркаса із дроту, **Contour** - з контурними лініями визначеної площини, **No Lines** - у вигляді кольорового твердого тіла, коли сховані лінії не видно, **Hide Lines** - у вигляді твердого тіла (**No fill**), коли сховані лінії не видно.
- Points Options* – визначає вигляд позначок на лініях графіка:

Draw Points - вмикає видимість позначок, **Symbol** - визначає тип позначки (точка, хрест, ромб), **Size** – визначає розмір позначки.

Color options визначає тип фарбування: **Colormap** – градієнтне фарбування з параметрами з вкладинки **Advanced**: райдуга, відтінки сірого, синього, зеленого, червоного тощо, **Solid Color** – суцільний колір.

Axis Вкладка **Axis** визначає вигляд осей X, Y, Z. Зміст є аналогічним вкладинці двовірних графіків: сітка, маркери осей, колір.

Special Розділ *Line Style* надає випадаючий список типів ліній сітки та контуру: суцільна (solid), пунктирна (dotted), штрихова (dashed), штрих пунктирна (da-dot).

Розділ *Contour* керує виглядом контурного графіка: фарбуванням **Fill**, виведенням ізоліній **Draw Lines** з автоматично **Auto Contour** чи у визначеній кількості з кількісними значеннями **Number**.

Advanced Розділ *Advanced View Options* вмикає режими туману **Enable Fog**, та перспективної проєкції **Perspective** на визначеній відстані **Viewing Distance**. Поле **Vertical Scale** у відносних координатах визначає вертикальний масштаб зображення.

Розділ *Plot* визначає оптичні якості поверхні: відбиття **Shininess**, прозорість **Transparency**, відступ зафарбованої ділянки від лінії сітки **Polygon Offset**.

Розділ *Colormap* керує напрямом градієнтного фарбування та його типом.

Розділ *Printing* керує якістю виведення зображень на принтер.

Графіки поверхні

В основі відображення 3D графіків лежать матричні перетворення. Система відображає саме елементи матриці. При введенні в графічне поле імені функції двох змінних система автоматично застосовує вбудовану функцію **CreateMesh**, яка створює матрицю з функції з аргументами за замовчанням. Визначення аргументів ранжованою змінною не впливає на діапазон зображення. *Коригування можливо тільки в ручному режимі в діалоговому вікні форматування 3D графіка (рис.2.45).*

Примітка. Введення замість імені розрахункової залежності не дозволяється.

CreateMesh(function, [s0, s1, t0, t1], [sgrid, tgrid], [fmap])

Функція повертає три матриці, які в параметричному вигляді відображають координати x , y , z функції **function** в інтервалах $[s0 - s1]$ для першої координати, $[t0 - t1]$ – для другої, з **sgrid**, **tgrid** значеннями по кожній координаті.

Первинна функція **function** може бути записана в звичайному вигляді $f(x, y) = \sin(x) + \cos(y)$ або параметричному вигляді як вектор-стовпець з трьох елементів, наприклад,

$$f(x, y) = \begin{bmatrix} x \\ y \\ \sin(x) + \cos(y) \end{bmatrix}.$$

Параметр **fmap** задається коли вихідна функція визначена не в декартових координатах. Він являє собою триелементний вектор. Елементами вектора є функції перерахування координат в декартові або ім'я функції перерахунку координат.

Примітка. Для перерахунку координат зручно використовувати вбудовані функції.

Усі параметри, окрім імені функції, не є обов'язковими. За замовчанням по обох аргументах діапазон: $-5 \dots 5$, число точок - 20. Зміною початку та кінця діапазону можна створювати графіки поверхонь із вирізом.

Наприклад, для швидкої побудови графіка функції двох змінних слід:

1. Визначити функцію користувача $f(x,y)=x^2+y^2$;
2. Активувати графічне поле клавішами «Ctrl+2» або в інший спосіб. На місце маркера вставити ім'я функції f .

Аналогічного результату можна досягти за допомогою функції **CreateMesh**. Порівняльні зображення наведені на рис. 2.47.

Наприклад, виклик функції для побудови графіка гіперболічного параболоїда $F(x, y) = x^2 - y^2$ для $-2 < x < 2$ та $-3 < y < 3$ та кількістю точок 50 по кожній змінній має наступний вигляд

$\mathbf{xy2pol}(x, y)$ - перерахунок декартових координат в полярні;
 $\mathbf{pol2xy}(r, \varphi)$ - перерахунок полярних координат в декартові;
 $\mathbf{xyz2cyl}(x, y, z)$ - перерахунок декартових координат в циліндричні;
 $\mathbf{cyl2xyz}(r, \varphi, z)$ - перерахунок циліндричних координат в декартові;
 $\mathbf{xyz2sph}(x, y, z)$ - перерахунок декартових координат в сферичні;
 $\mathbf{sph2xyz}(x, \varphi, \lambda)$ - перерахунок сферичних координат в декартові.

Результатом функції є вектор, який містить координати у відповідній системі координат. Для перерахування значень масиву слід застосовувати індексні чи векторизовані операції з векторами.

Примітка. Функції перетворення можуть мати кутові координати в радіанах чи градусах. Для введення кутів в градусах слід домножити число на *deg*. Для виведення кутів в градусах слід вписати *deg* в поле праворуч від числа.

Наприклад, для точки P з декартовими координатами $x=2$, $y=8$, $z=5$ сферичні координати можна перерахувати наступним чином:

$$\begin{array}{l}
 x := 2 \quad y := 8 \quad z := 5 \quad p := \begin{pmatrix} x \\ y \\ z \end{pmatrix} \\
 \hline
 \mathbf{xyz2sph}(x, y, z) = \begin{pmatrix} 9.644 \\ 1.326 \\ 1.026 \end{pmatrix} \quad \mathbf{xyz2sph}(p) = \begin{pmatrix} 9.644 \\ 1.326 \\ 1.026 \end{pmatrix} \quad \mathbf{xyz2sph}(p) = \begin{pmatrix} 552.54 \\ 75.964 \\ 58.77 \end{pmatrix} \text{ deg}
 \end{array}$$

Якщо чисельні значення не потрібні, то перерахунок систем координат можна задавати на вкладинці **Quick Plot Data** вікна форматування 3D графіків.

Наприклад, сферична поверхня з центром в точці початку координат та радіусом r в декартових координатах параметрично описується як

$$\begin{aligned}
 \mathbf{X}(u, v) &= r \cdot \sin(u) \cos(v) & \mathbf{Y}(u, v) &= r \cdot \sin(u) \sin(v) \\
 \mathbf{Z}(u, v) &= r \cdot \cos(u) .
 \end{aligned}$$

Зображення поверхні через параметризовані функції та вектор в декартових координатах наведено на рис. 2.48. Параметри графіка при такому визначенні даних можна змінити ТІЛЬКИ у вікні редагування.

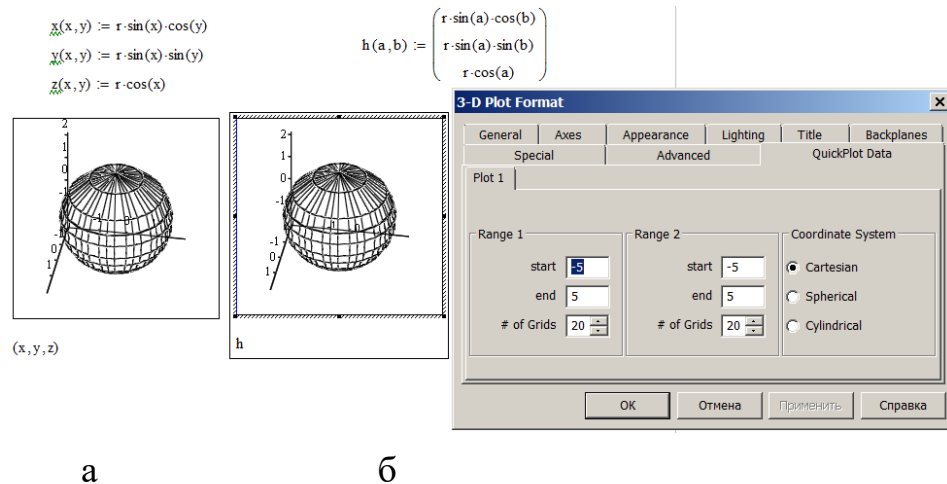


Рис.2.48. Зображення сфери

- а) параметризованими функціями в декартових координатах,
- б) вектором в декартових координатах

Недоліком цього способу можна вважати те, що точно визначити параметри графіка у вікні, наприклад, через формульні залежності, неможливо, для визначення потрібно робити зайві дії, щоб дістатися до полів редагування. Програмний контроль забезпечує саме функція **CreateMesh**.

Графік поверхні із використанням параметризації та функції **CreateMesh** показано на рис. 2.49. При такому визначенні даних вікно редагування діапазонів стає недоступним.

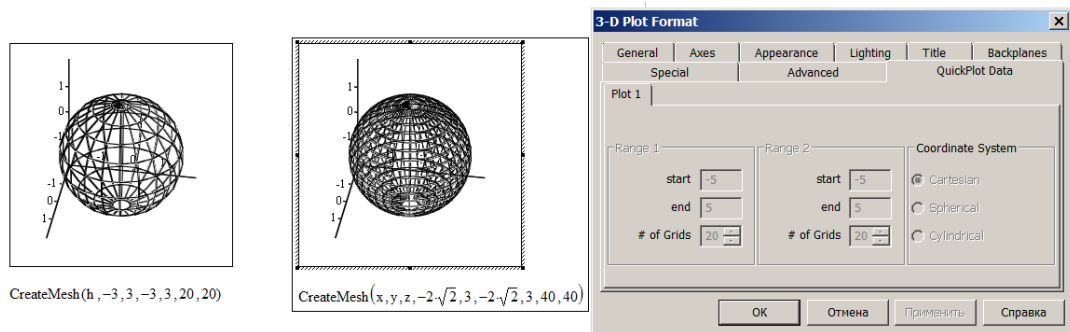


Рис.2.49. Зображення сфери із функцією **CreateMesh** в декартових координатах

Опис сфери в сферичних координатах суттєво спрощується. Поверхня визначається радіусом r , та двома кутами: місця, який вимірюється від площини $ХОУ$ до точки та має діапазон $-\pi .. \pi$, азимута, який вимірюється в площині $ХОУ$ в діапазоні $0.. 2\pi$.

Для отримання зображення із функціями в сферичних координатах слід в дужках ввести через кому імена параметричних функцій: кута місця, азимута, радіуса та на вкладниці *QuickPlot Data* обрати систему координат *Spherical* (рис. 2.50).

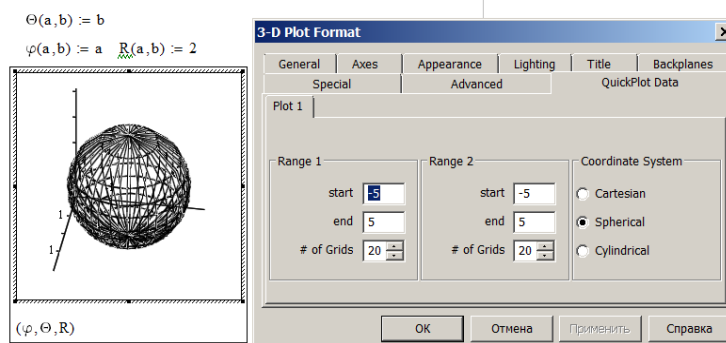


Рис. 2.50. Зображення сфери в сферичних координатах

Як і у випадку декартових координат, можна застосувати функцію **CreateMesh**. Результат застосування функції наведено на рис. 2.51. Сторінка визначення діапазонів та систем координат графіка і в цьому випадку стає недоступною. Особливістю є різний порядок розташування елементів в векторах. Для безпосереднього застосування він співпадає з порядком розташування імен параметричних функцій: кут місця, азимут, радіус. Виклик функції потребує іншого розташування імен: радіус, кут місця, азимут (рис. 2.51).

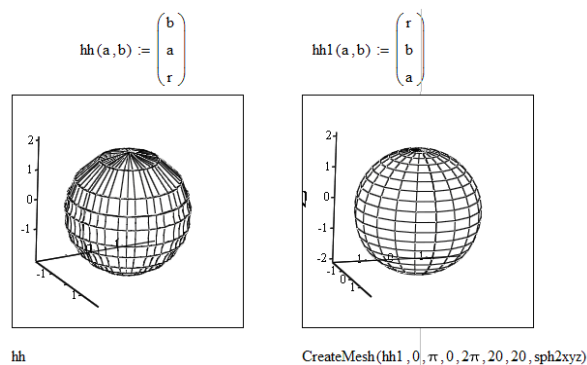


Рис. 2.51. Зображення сфери в сферичних координатах

а) через вектор, б) через функцію CreateMesh

Контурні графіки

Просторове зображення дає загальну уяву про функцію або поверхню. Для кількісного аналізу можна використати графіки ізоліній **Contour Plot** з сторінки *General* вікна редагування графіків. Цей графік зображує поверхню

ортогонально площині XOY уздовж осі Z. При цьому на зображенні виводяться лінії визначених значень функції – ізолінії та кількісні позначки (рис. 2.52).

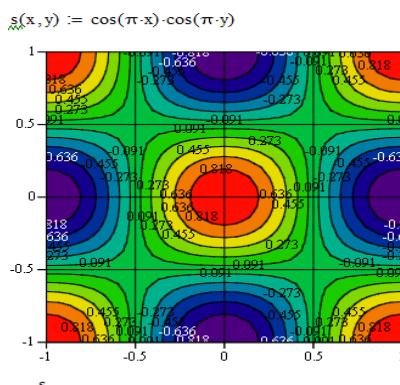


Рис. 2.52. Контурний графік

Графіки просторових кривих

Для побудови просторових кривих застосовується графіки типу **Data/Scatter Plot**. Для опису кривої використовується параметрична форма. Для побудови зображення необхідно підготувати три вектори чи функції із визначенням аргументів x, y, z . Розмір векторів визначає кількість точок графіка в самому описі. При використанні функцій кількість точок за замовчанням 20, зміна кількості проводиться на вкладниці *QuickPlot Data* вікна редагування графіка тільки вручну.

Для формування векторів призначена вбудована функція

CreateSpace(F, min, max, num, fmap),

де **F** - вектор, елементами якого є три вирази розрахунку координат x, y, z в параметричному вигляді, **min, max** - початкове та кінцеве значення аргументу, **num** - кількість точок, **fmap** – необов'язкова функція перерахунку системи координат, якщо в векторі **F** застосована не декартова система координат.

За замовчанням $\text{min}=-5, \text{max}=5, N=20$.

Наприклад, просторова спіраль описується параметричними функціями у вигляді:

$$X(t)=r \cdot \cos(t) \quad Y(t)=r \cdot \sin(t) \quad Z(t)=t$$

Для побудови графіка слід:

1. Визначити функції користувача $X(t)$, $Y(t)$, $Z(t)$. Параметр k позначає кількість витків спіралі.

2. В маркер тримірного графіка ввести в дужках через кому імена функцій. За замовчанням буде зображено графік поверхні (рис. 2.53).

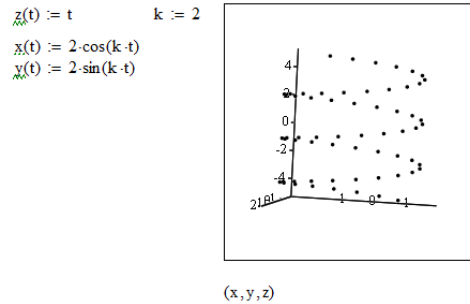


Рис. 2.53. Зображення спіралі до редагування

3. За замовчанням графік зображується у вигляді точок. На вкладинці *Appearance* обрати в групі *Line options* пункт **Line**. Графік набуває вид, який представлено на рис. 2.54.

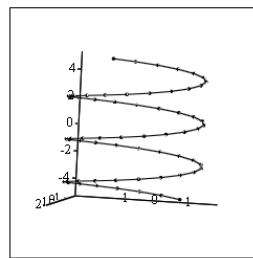
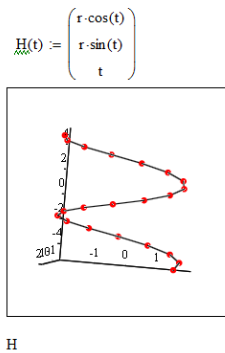


Рис. 2.54. Графік просторової спіралі

Аналогічного результату можна досягти з використанням функції - вектора. Технологія побудови зображення відрізняється тим, що в маркер ставиться тільки одне ім'я вектора. На рис. 2.55 наведено графік, для якого на сторінці *Appearance* в групі *Point options* додатково визначено опорні точки розміром 3 червоного кольору

Програмне керування кількістю точок графіка забезпечують вихідні дані у вигляді векторів або функція **CreateSpace**. В такому способі аргументи визначаються явно (рис. 2.56).



H

Рис. 2.55. Графік спіралі з використанням вектора

```
N := 50  i := 0..N  zm_i := i
xm_i := 2*cos(k * (2-pi*i)/N)
ym_i := 2*sin(k * (2-pi*i)/N)
```

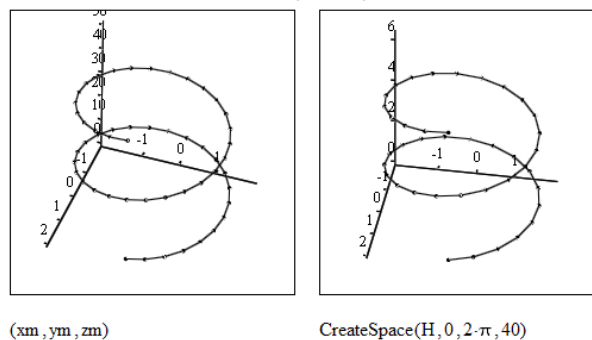


Рис. 2.56. Застосування векторів та функції Create Space

Графік векторного поля

Графік векторного поля активується пунктом *Vector Field Plot* діалогового вікна тримірних графіків. Зображення являє собою карту векторів. Кожен вектор починається у вузлі сітки та визначається локальними координатами відносно вузла сітки x, y . Даними для цього виду графіків є матриці, кожен елемент яких містить дві кординати « x, y » вектора.

Зображення виводиться для транспонованої матриці. Рядки вихідної матриці зображуються стовпцями. Для абсцис додатній напрямок праворуч, від’ємний – ліворуч, для ординат додатній напрямок вгору, від’ємний – вниз.

Вихідна матриця може готуватися кількома способами.

Спосіб 1. Кожен елемент матриці вважається комплексним числом. Дійсна частина відповідає координаті x , уявна – y вектора.

Спосіб 2. Дані вводяться у дві матриці. Перша матриця містить координати x , друга – y векторів. Матриці поєднуються в комплексну так,

щоб перша відповідала дійсній частині, друга – уявній. Можливо вивести зображення без поєднання як параметричне з двох вихідних матриць.

Наприклад, для значень координат $x=\cos(xy)$ $y=\sin(xy)$ побудова карти способом 1 може мати вигляд, зображений на рис. 2.57а, для способу 2 з однією матрицею – на рис. 2.57б, для способу 2 параметрично – на рис. 2.57в.

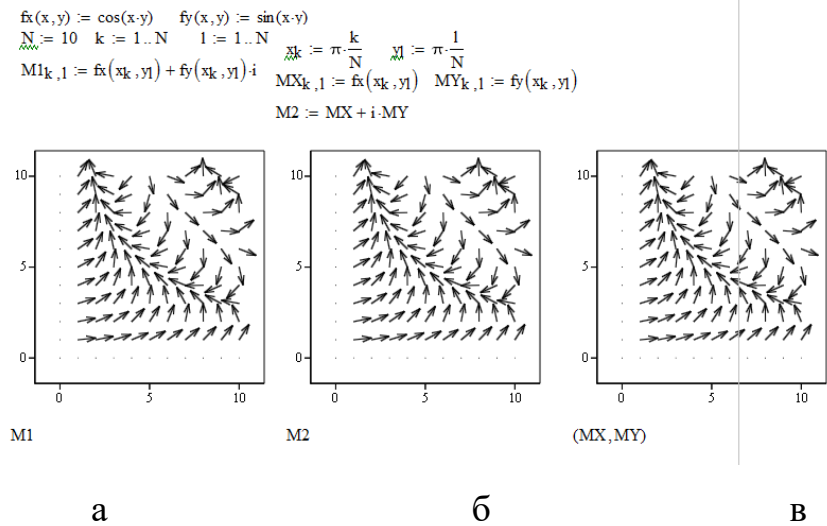


Рис. 2.57. Векторне поле а) спосіб 1 б) спосіб 2 в) спосіб 2 параметрично

Зображення з тінню

Для більшої інформативності використовуються зображення функції з «тінню». В якості тіні можна застосувати контурний графік або фон з сіткою. Для контурного графіка в якості тіні ім'я функції вводиться двічі (рис. 2.58).

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

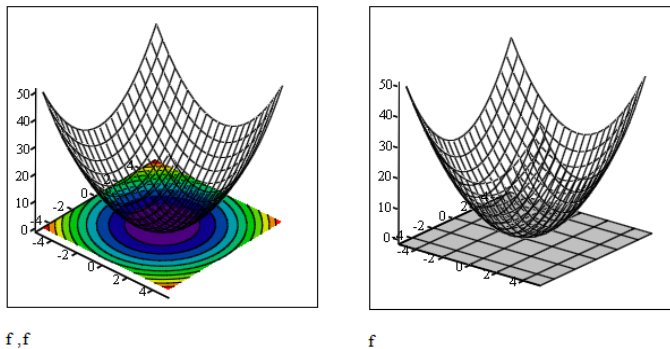


Рис. 2.58. Зображення поверхні з «тінню».

2.7.3. Контрольні питання

В якому діапазоні за замовчанням будується графік функції однієї змінної?

Як будуються графіки кількох функцій в режимі додаткової осі?

Що визначає режим «Show markers»?

Для чого використовується логарифмічний режим осей графіка?

Який режим призначено для отримання інформації про координати точок кривої?

В якому діапазоні за замовчанням будується графік функції двох змінних?

Що вводиться в поле маркера графіка двох змінних?

Як впливає визначення аргументу функції двох змінних у вигляді ранжованих змінних на вигляд графіка?

Що таке графік векторного поля?

Як враховується система координат для тримірних графіків?

Яким типом графіків зображуються просторові криві?

Яка особливість контурних графіків?

2.8. Розрахунки з розмірностями

В *Mathcad*, на відміну від більшості СКМ, передбачена можливість проведення технічних, фізичних розрахунків з розмірними величинами. Змінну, значенням якої є величина з розмірністю, називають *розмірною змінною*.

В *Mathcad* передбачена робота в таких системах розмірностей:

- система СІ (SI (International));
- система МКС (MKS, метр, кілограм, секунда);
- система СГС (CGS, сантиметр-грам-секунда);
- американська система одиниць (U.S.);
- система користувача (Custom).

Примітка. Останню можливість не можна вважати системою користувача, бо це є тільки можливість скомбінувати одиниці розмірності з вище наведених систем. Додати власні або змінити мову таким чином неможливо.

Фізичні розрахунки проводяться з застосуванням базових одиниць виміру: L - **length** (довжина), M - **mass** (маса), T - **time** (час), Q - **charge** (заряд). K - **temperature** (температура), I – **luminosity** (сила світла), S – **substance** (одиниця речовини).

Наприклад, в системі СІ основні одиниці виміру мають наступний вигляд: m (length, meter), kg (mass, kilogramm), s (time, second) q (charge, coulomb), K (temperature, kelvin), cd (luminosity, candella).

Через основні одиниці виміру визначаються інші додаткові одиниці. Наприклад, $\text{cm}:=0.01 \cdot \text{m}$, $\text{mm}:=0.001 \cdot \text{m}$, $\text{km}:=1000 \cdot \text{m}$, $\text{tonne}:=1000 \cdot \text{kg}$, $\text{hr}:=3600 \cdot \text{sec}$, $\text{min}:=60 \cdot \text{s}$, $\text{A}:=\text{coul} \cdot \text{sec}^{-1}$.

Незалежно від використаної системи розмірностей, результат операції в *Mathcad* переважно виражається в фундаментальних одиницях: довжина, маса, час, заряд, одиниця речовини, температура, сила світла. Конкретні розмірності виставляє користувач вручну.

Частково зробити приведення розмірностей система спробує, якщо на сторінці *Unit Display* (рис. 2.59) пункту *Result* меню *Format* обрано опцію *Simplify units when possible*.

Сторінка містить опції: *Format units*, *Simplify units when possible*, *Show unit exponents as a fraction*.

Якщо встановлено опцію *Format units*, вираз для розмірної частини значення відображається у вигляді дроби. Опція *Simplify units when possible* встановлюється для автоматичного приведення одиниць вимірювання. Опція *Show unit exponents as a fraction* може бути встановлена тільки для всього документа в цілому і забезпечує відображення одиниць розмірностей у вигляді раціональних дробів.

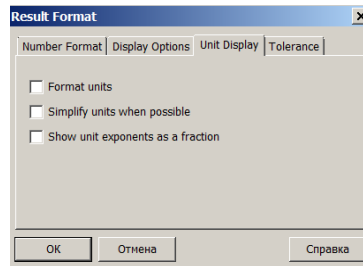


Рис. 2.59. Сторінка форматування розмірностей

Якщо діалогове вікно *Result Format* відкривалося подвійним клацанням миші на результаті, то формат застосовується тільки до цього результату. Для глобального визначення вигляду одиниці вимірювання у всьому документі діалогове вікно *Result Format* слід відкривати пунктом *Result* меню *Format*.

Визначення системи розмірностей проводиться на сторінці *Unit System* (рис. 2.60) пункту *Worksheet Options*, меню *Tools*. Для вимкнення розмірностей слід обрати пункт *None*.

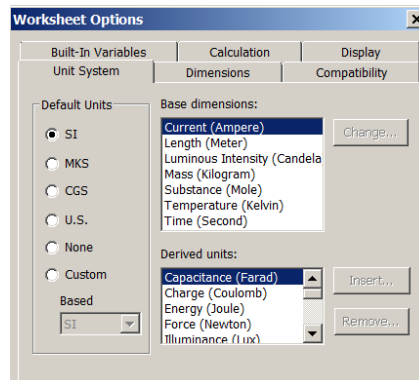



Рис. 2.60. Системи розмірностей

Для використання розмірної змінної слід ввести її ім'я, значення та знак множення. В поле маркеру після знаку множення вводиться назва розмірності. За замовчанням розмірності вводяться АНГЛІЙСЬКОЮ МОВОЮ.

Для спрощення можна використати діалогове вікно *Insert Unit* через поле *Unit* меню *Insert* або ще простіше – кнопкою  меню *Standard* (рис. 2.61).

Для скорочення часу для пошуку потрібної розмірності всі розмірності, які є в системі, розбиті по категоріях. У вікні є два списки: *Dimension* та *Unit*. Список *Dimension* представляє категорії розмірностей. Список *Unit* містить конкретні розмірності відповідної категорії (табл. 2.8).

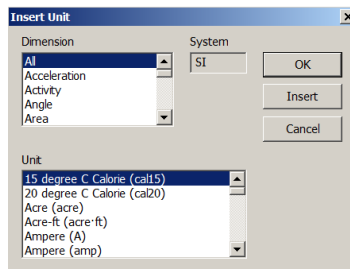


Рис. 2.61. Діалогове вікно Insert Unit

Таблиця 2.8. Типи розмірних величин *Mathcad*

Категорія	Опис
All	Всі доступні розмірності
Acceleration	Одиниці розмірності пришвидшення
Activity	Одиниці розмірності радіоактивності
Angle	Одиниці розмірності кутів
Area	Одиниці розмірності площ
Capacitance	Одиниці розмірності електричної ємності
Catalytic Activity	Одиниці розмірності каталітичної активності
Charge	Одиниці розмірності електричного заряду
Conductance	Одиниці розмірності електричної провідності
Current	Одиниці розмірності сили електричного струму
Dose	Одиниці розмірності доза випромінювання
Energy	Одиниці розмірності енергії
Flow Rate	Одиниці розмірності потоку рідини
Force	Одиниці розмірності сили
Force Density	Одиниці розмірності об'ємної щільності сили
Force per Length	Одиниці розмірності лінійна щільності сили
Frequency	Одиниці розмірності частоти
Illuminosity	Одиниці розмірності освітленості
Inductance	Одиниці розмірності індуктивності
Length	Одиниці розмірності довжини
Potential	Одиниці розмірності електричного потенціалу
Power	Одиниці розмірності потужності
Pressure	Одиниці розмірності тиску
Resistance	Одиниці розмірності електричного опору

Таблиця 2.8. Продовження

Категорія	Опис
Substance	Одиниці розмірності кількості речовини
Temperature	Одиниці розмірності температури
Time	Одиниці розмірності часу
Velocity	Одиниці розмірності швидкості
Viscosity, dynamic	Одиниці розмірності динамічної в'язкості
Volume	Одиниці розмірності об'єму
Magnetic Flux Density	Одиниці щільності магнітного потоку
Mass	Одиниці розмірності маси
Permeability	Одиниці розмірності магнітної проникненості
Permittivity	Одиниці розмірності електричної проникненості
Luminous Intensity	Одиниці розмірності яскравості світла
Magnetic Field Strength	Одиниці розмірності сили магнітного поля
Magnetic Flux	Одиниці розмірності магнітного потоку

Користувач має змогу ввести свої додаткові одиниці по аналогії з вбудованими.

Наприклад, саме так можна перевести розмірності на іншу мову або створити власні позасистемні одиниці:

**см:=0.01·m мм:=mm В:=V година:=3600·sec Вт:=watt
квартира:=60·м² дім:=100·квартира людина:=1.**

Примітка. При введенні розмірностей користувача, які не пов'язані із вбудованими розмірностями, *Mathcad* не слідкує за правильністю розмірностей. Вся відповідальність за результати покладається на користувача, який повинен сам вписувати у відповіді правильну розмірність.

Розмірність може призначатися змінним, ранжованим змінним, масивам, функціям користувача.

Розмірність для ранжованої змінної вводиться для кожного зі значень, розмірність масивів вводиться як і для чисел. Розмірність всіх елементів масиву має бути однаковою.

Наприклад, **a:=10сек, 11сек . . 20сек**

$$v = \begin{pmatrix} 20 \\ 40 \\ 60 \end{pmatrix} \text{ мм} \quad v = \begin{pmatrix} 0.02 \\ 0.04 \\ 0.06 \end{pmatrix} \text{ метр} \quad v = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} \text{ см}$$

Графіки відображаються в стандартних розмірностях. В разі використання умовного оператора результати можуть бути тільки безрозмірними.

Розмірність в підпрограмах

Для перевірки розмірностей в версіях з 13-ї слід ввести ім'я функції без параметрів. Результат має вигляд, подібний наступному

$$vp(t,v) := t \cdot v \quad vp = f(\text{any1}, \text{any2}) \rightarrow \text{any1} \cdot \text{any2}$$

Результат містить символічну позначку функції f з переліком аргументів any та стрілку. Після стрілки розташовується пояснення. Якщо параметр може мати розмірність, то виводиться його ім'я та ступінь розмірності, якщо ні - то виводиться службове слово *unitless*.

Примітка. Розплатою за введену перевірку розмірностей стало те, що в версіях системи від 13-ї деякі вбудовані функції не працюють з розмірностями, а саме: функції інтерполяції, регресії, розв'язання рівнянь, обчислення логарифмів, створення сітки для тримірних графіків.

Розмірності в підпрограмах можна застосовувати тільки коли всі результати мають однакову розмірність. Наприклад, в попередніх версіях системи існувала можливість виконання підпрограми, яка розраховувала периметр та площу геометричної фігури. В останніх версіях така підпрограма неможлива, потрібно окремо розраховувати площу та периметр.

2.8.1. Контрольні питання

Що таке розмірна змінна?

Що таке базові одиниці вимірювання?

Які системи вимірювання підтримує MathCAD ?

Як вводиться розмірна змінна?

Як змінити одиницю вимірювання?

Як ввести одиницю вимірювання, яка відсутня в системі?

2.9. Символьні обчислення

Символьними або аналітичними обчисленнями називають обчислення, результат яких отримується не у вигляді чисел, а у вигляді формульних залежностей.

Вигляд результату залежить від конкретного завдання. Одна операція може розглядатися в одному випадку як вірна, а в іншому – ні. Крім того, комп'ютерні алгоритми недосконалі, тому відповіді системи в символьному вигляді є не завжди практичні, а іноді результат є простим повторенням вихідних даних.

Символьний процесор в *Mathcad* є запозиченим. В версіях системи до 12-ї застосовувався скорочений варіант символьного процесора з системи *Maple*, далі він був замінений на символьний процесор системи *MuPad*. Відповідно, можливості аналітичних перетворень в останніх версіях системи дещо скромніші.

В документах можливо аналітично проводити диференціювання, інтегрування, розкладання функції в ряд, спектральні перетворення, спрощування виразів, знаходження границі тощо.

Символьні дії можна провести «швидким» способом з меню *Symbolics*, способом «реального часу» кнопками панелі *Symbolic* (рис. 2.62) або оператором символьного виведення результату « \rightarrow » (клавіші *Ctrl+.*).

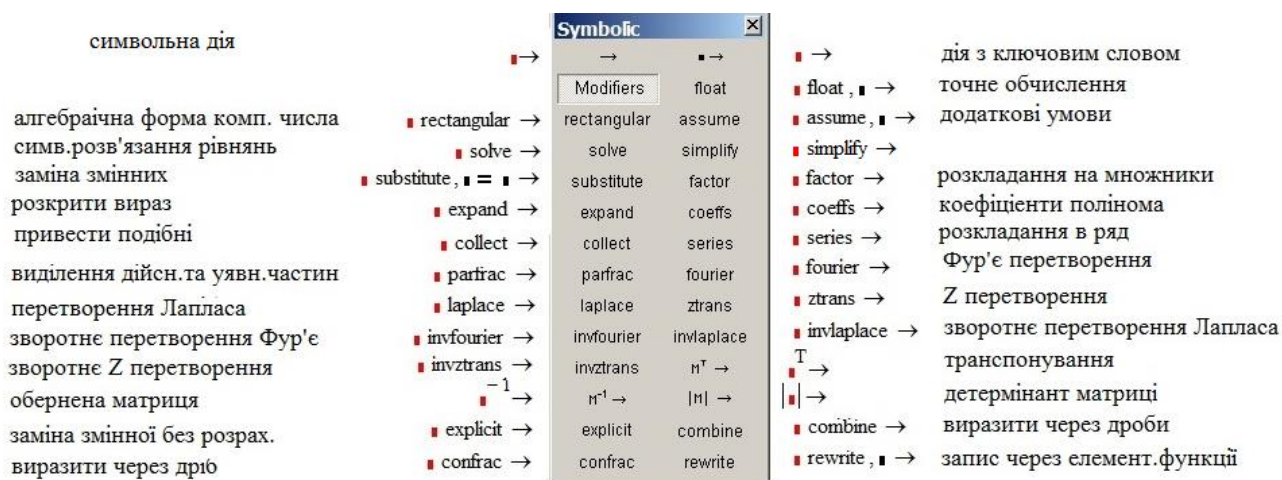


Рис. 2.62. Панель символьних перетворень

Запис перетворень «реального часу» складається з самого виразу, позначки символьного виведення та уточнюючого ключового слова. Ввести такий шаблон можна для кожної дії з панелі *Symbolic* та універсальною

позначкою (рис. 2.63), яка викликається клавішами «*Ctrl+Shift+.*». Для деяких операцій після ключового потрібно вставляти для уточнення імена змін та вирази, у відношенні до яких планується проведення перетворень.

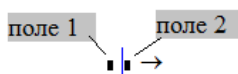


Рис. 2.63. Шаблон ключових слів

«Швидкий» спосіб є зручнішим, коли потрібно швидко отримати одноразово аналітичний результат без збереження ходу обчислень. Перетворення з меню відносяться тільки до одного виділеного виразу. Спосіб «реального часу» дозволяє записувати вираз в традиційній формі з урахуванням виразів, що вже записані в документі та зберігати хід перетворень.

Розташування відповіді при «швидкому» способі (поруч, знизу, на місці завдання, наявність коментарів, що пояснюють здійснені операції) задається форматом перетворень в пункті *Evaluation style* меню *Symbolics*.

Для більшості символічних функцій дії мають наступний порядок:

1. Введення виразу.
2. Виділення змінної або виразу.
3. Виклик відповідної команди.

Команди меню символічної математики *Symbolics*

Меню має 5 груп полів: *група одиночних дій*; *група дій, що потребують виділення аргументів*; *група спектральних перетворень*; *група операцій з матрицями та службова група завдання формату вигляду результату*.

Група одиночних дій

Пункт Evaluate

Дозволяє проводити одиночні обчислення в символічному вигляді. Перед застосуванням пункту слід позначити вираз.

Symbolically - отримання результату в символічному вигляді.

Complex - отримання результату в комплексному вигляді.

Floating point - отримання чисельного результату з великою кількістю знаків мантиси.

Simplify спрощує вираз.

Expand розкриває (розгортає) вираз.

Factor розкладає вираз, що виділено, на множники.

Collect є зворотнім до розкладання на множники.

Polynomial Coefficients обраховує коефіцієнти полінома, на який може бути розкладено вираз.

Група дій з виділеними змінними

Solve розв'язує рівняння.

Substitute проводить заміну змінних.

Differentiate диференціює.

Integrate інтегрує.

Expand to Series розкладає в ряд Тейлора.

Convert to Partial Fraction розкладає на частини.

Група матричних операцій Matrix

Пункти цієї групи дозволяють провести інвертування, транспонування матриці та обчислити її детермінант в символьному вигляді.

Група спектральних перетворень Transform

Виконує символьні перетворення Фур'є, Лапласа та дискретне z-перетворення.

2.9.1. Символьне обчислення виразів

Отримання результатів обчислень може проводитися в чисельному або символьному вигляді. Символьні обчислення можуть проводитися для знаходження похідних, невизначених інтегралів, лімітів, сум та доданків, знаходження параметрів матриць.

Наприклад, для функції користувача $f(x) := x^2 \cos(2x)$ застосування символу чисельного виведення дає чисельний результат $f(3) = 8.642$, застосування символу символного виведення дає аналітичний результат $f(3) \rightarrow 9 \cdot \cos(6)$.

Для отримання результату обчислення в символному вигляді слід:

1. Ввести вираз $A \cdot \sin(\sin(B \cdot C))$.

2. Зліва від виразу ввести символ правої стрілки з панелі *Symbolic*, пункту *Evaluate - Symbolically* меню *Symbolics* або клавішами «Ctrl+». На екрані з'явиться результат $A \cdot \sin(\sin(B \cdot C)) \rightarrow A \cdot B \cdot C$

або

1. Ввести шаблон виразу для символних обчислень символ правої стрілки з панелі або клавішами «Ctrl+» \rightarrow .

2. На місце маркеру ввести вираз $A \cdot \sin(\sin(B \cdot C)) \rightarrow A \cdot B \cdot C$. На екрані з'явиться результат

$A \cdot \sin(\sin(B \cdot C)) \rightarrow A \cdot B \cdot C$.

Символьне диференціювання, інтегрування

Операції інтегрування та диференціювання можуть проводитися через пункт *Variable* меню *Symbolics*.

Наприклад, для знаходження першої та другої похідної виразу $ax^2 + b \sin(cx)$ слід ввести вираз $a \cdot x^2 + b \cdot \sin(c \cdot x)$ та виділити змінну x , в меню *Symbolics* обрати пункт *Variable - Differentiate*. В документі з'явиться результат

$a \cdot x^2 + b \cdot \sin(c \cdot x)$ by differentiation, yields $2 \cdot a \cdot x + b \cdot c \cdot \cos(c \cdot x)$

В формулі першої похідної виділити змінну x та знов в меню *Symbolics* обрати пункт *Variable - Differentiate*. В документі з'явиться результат виразу другої похідної

$2 \cdot a \cdot x + b \cdot c \cdot \cos(c \cdot x)$ by differentiation, yields $2 \cdot a - b \cdot c^2 \cdot \sin(c \cdot x)$

Для знаходження інтеграла від $\frac{1}{x^3}$ слід ввести вираз $\frac{1}{x^3}$ та виділити змінну x , в меню *Symbolics* обрати пункт *Variable - Integrate*. В документі з'явиться результат

$$\frac{1}{x^3} \text{ by integration, yields } -\frac{1}{2 \cdot x^2}$$

Таким чином можна обробляти тільки явно визначені вирази. *Наприклад*, диференціювання функції користувача система таким способом виконує, але результат не має практичної цінності:

$$f(x) \text{ by differentiation, yields } \frac{d}{dx} f(x)$$

Операція може бути виконана із застосуванням розширених операторів.

Для наведених вище прикладів, послідовність дій має бути такою:

1. Ввести оператори диференціювання, кратного диференціювання, інтегрування

$$\frac{d}{dx}, \quad \frac{d^2}{dx^2}, \quad \int \cdot dx$$

2. В маркери ввести вирази та змінну диференціювання/інтегрування x .

3. Обрати символічний режим введенням символу « \rightarrow ». На екрані з'явиться результат

$$\frac{d}{dx} (a \cdot x^2 + b \cdot \sin(c \cdot x)) \rightarrow 2 \cdot x + \cos(x) \quad \frac{d^2}{dx^2} (a \cdot x^2 + b \cdot \sin(c \cdot x)) \rightarrow 2 - \sin(x) \quad \int \frac{1}{x^3} dx \rightarrow -\frac{1}{2 \cdot x^2}$$

Розширені оператори дозволяють отримати результати дій над функціями користувача. *Наприклад*, інтеграл від функції користувача знаходиться наступним чином :

$$f(x) := \frac{1}{x} \int f(x) dx \rightarrow -\frac{1}{2 \cdot x^2}$$

Обчислення сум, добутків, границі

Подібно до інтегрування/диференціювання проводиться обчислення нескінчених сум, добутків, лімітів.

Наприклад, для обчислення суми $\sum_{n=0}^{\infty} \frac{1}{2^n}$, добутку $\prod_{n=2}^{\infty} 1 - \frac{1}{n^2}$, ліміту $\lim_{x \rightarrow 0} \left(\frac{\sin(ax)}{x} \right)$ слід:

1. З панелі *Calculus* ввести шаблони розширених операторів

$$\sum_{n=1}^{\cdot} \quad \prod_{n=1}^{\cdot} \quad \lim_{x \rightarrow \cdot}$$

2. На місце маркерів ввести вихідні вирази та завершити введення символом «→». На екрані з'явиться результат

$$\sum_{n=0}^{\infty} \frac{1}{2^n} \rightarrow 2 \quad \prod_{n=2}^{\infty} \left(1 - \frac{1}{n^2} \right) \rightarrow \frac{1}{2} \quad \lim_{x \rightarrow 0} \frac{\sin(a \cdot x)}{x} \rightarrow a$$

Примітка. Результат символічних перетворень залежить від версії системи та якості підготування вихідних даних. Наприклад, обчислення суми в версії 11 системи дає вірний результат:

$\prod_{n=2}^{\infty} \left(\frac{n^3 - 1}{n^3 + 1} \right) \rightarrow \frac{2}{3}$, а вже в версії 14 результат важко назвати практичним:

$$\prod_{n=2}^{\infty} \left(\frac{n^3 - 1}{n^3 + 1} \right) \rightarrow \frac{2 \cdot \Gamma \left[\frac{3}{2} - \left(\frac{\sqrt{3}}{2} \right) i \right] \cdot \Gamma \left(\frac{3}{2} + \frac{1}{2} \sqrt{3} i \right)}{\Gamma \left[\frac{5}{2} - \left(\frac{\sqrt{3}}{2} \right) i \right] \cdot \Gamma \left(\frac{5}{2} + \frac{1}{2} \sqrt{3} i \right)}$$

За необхідності отримати чисельний результат з підвищеною точністю застосовується ключове слово *float*, праворуч від якого визначається

необхідна кількість знаків мантиси результату. Наприклад, число π з 11-ма знаками система обраховує як

π float, 11 \rightarrow 3.1415926536

2.9.2. Символьні тотожні перетворення виразів

При математичних обчисленнях часто доводиться проводити перетворення виразів спрощуючи, групуючи їхні частини і т.д. В системі такі операції потребують уточнення типу дії. Дію можна провести «швидким» способом з меню *Symbolics* та способом «реального часу» для збереження дії в документі.

Спрощення виразів

Для спрощення виразів шляхом приведення до спільного знаменника, приведення подібних членів, тригонометричних перетворень застосовується ключове слово *simplify*.

Наприклад, для проведення спрощення виразу «швидким» способом слід:

1. Ввести вираз $\sin(x)^2 + \cos(x)^2$.

2. Обрати в меню *Symbolics* пункт *Simplify*.

3. В документі з'явиться результат $\sin(x)^2 + \cos(x)^2$ simplifies to 1.

Для проведення спрощення виразу «в реальному часі» слід:

1. Ввести вираз $\sin(x)^2 + \cos(x)^2$.

2. Обрати на панелі *Symbolic* кнопку *Simplify*.

3. В документі з'явиться результат $\sin(x)^2 + \cos(x)^2$ simplify \rightarrow 1.

Або

1. Вивести шаблон *Simplify* з панелі *Symbolic* \blacksquare simplify \rightarrow .

2. В маркер ввести вираз $\sin(x)^2 + \cos(x)^2$.

3. В документі з'явиться результат $\sin(x)^2 + \cos(x)^2 \text{ simplify} \rightarrow 1$.

Або

1. Обрати на панелі *Symbolic* кнопку *Keyword*  \rightarrow .

2. В поле 1 ввести вираз $\sin(x)^2 + \cos(x)^2$, в поле 2 – ключове слово *Simplify*.

3. В документі з'явиться результат $\sin(x)^2 + \cos(x)^2 \text{ simplify} \rightarrow 1$.

Примітка. Система по різному виконує дії для аргументів з мантисою та без. Для аргументу без мантиси повертається сам вихідний вираз, для аргументу з мантисою проводиться чисельне обчислення:

$$\sqrt{3} \text{ simplify} \rightarrow \sqrt{3} \quad \sqrt{3.0} \rightarrow 1.7320508075688772935$$

$$\sqrt{3.0} \text{ simplify} \rightarrow 1.7320508075688772935$$

Розкриття виразу

Для проведення розкриття виразу застосовується ключове слово *expand*. Дії спрощення та розкриття є зворотними. Наприклад,

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

$$(x + y)^3 \text{ expand} \rightarrow x^3 + 3 \cdot x^2 \cdot y + 3 \cdot x \cdot y^2 + y^3 \text{ simplify} \rightarrow (x + y)^3$$

В інструкцію *expand* можна вводити для уточнення вираз, який повинен залишитися в результаті без зміни. Наприклад, розкладання добутку

$$(x - y) \cdot (x + y) \cdot (x^2 + y^2) \text{ дає правильний звичний результат}$$
$$(x - y) \cdot (x + y) \cdot (x^2 + y^2) \text{ expand} \rightarrow x^4 - y^4$$

, розкладання за умови збереження

множника $x^2 + y^2$ слід записувати наступним чином

$$(x + y) \cdot (x - y) \cdot (x^2 + y^2) \text{ expand, } x^2 + y^2 \rightarrow x^2 \cdot (x^2 + y^2) - y^2 \cdot (x^2 + y^2)$$

Розкладання на множники

Для дії розкладання на множники слугує ключове слово *factor*. Для виразів дія проводиться у вигляді множників - виразів. Таким чином можна знайти множники ЦІЛОГО числа, тобто визначити, чи є число простим.

Наприклад, щоб розкласти число 111 на множники можна ввести число, обрати дію *factor* та отримати результат $111 \text{ factor} \rightarrow 3 \cdot 37$.

Число 111 не є простим, бо складається з множників 1, 3, 37. А от число 113 є простим $113 \text{ factor} \rightarrow 113$.

Для виразів дія розкладання має наступний вигляд:

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

Функція проводить розкладання на множники, якщо корені виразу є раціональними. У випадку ірраціональних коренів функція як результат повертає сам вираз. При необхідності отримати добуток, слід явно вказати ірраціональний корінь через кому після ключового слова *factor*.

Наприклад, для виразу $x^2 - 2$ система не дасть добутку, бо коренем функції є $x = \sqrt{2}$:

$x^2 - 2 \text{ factor} \rightarrow x^2 - 2$. Додавання значення кореню введе на екран очікуваний результат:

$$x^2 - 2 \text{ factor, } \sqrt{2} \rightarrow (x - \sqrt{2}) \cdot (x + \sqrt{2})$$

Примітка. Рекомендується для проведення перетворень використовувати кілька методів. Наприклад, для відношення $\frac{1-x^5}{1-x}$ дія розгортання дає непрактичний результат без пониження ступеню

$$\frac{(1-x^5)}{1-x} \text{ expand} \rightarrow \frac{x^5}{x-1} - \frac{1}{x-1}$$

. Дія розкладання на множники не тільки розкладає різницю ступенів, а й скорочує ступінь

$$\frac{(1-x^5)}{1-x} \text{ factor} \rightarrow x^4 + x^3 + x^2 + x + 1$$

Заміна змінної в виразі

Для аналізу або для розрахунків іноді доцільно отримати результат, коли в вихідному виразі замінена змінна.

Заміна змінної може проводитися «швидким» способом та способом «реального часу».

Нашвидкуруч для зміни змінної використовується пункт *Substitute* меню *Symbolics – Variable*.

Наприклад, щоб отримати вираз $\sin(kx^2+bx)$ із заміною $k=ax^2$ слід:

1. Ввести вираз та вираз на заміну. Скопіювати вираз на заміну в кліпбоард (наприклад, клавішами Ctrl+C). В вихідному виразі виділити змінну k .

2. В меню *Symbolics – Variable* обрати пункт *Substitute*. На екрані з'явиться результат:

$$\sin(k \cdot x^2 + b \cdot x) \text{ by substitution, yields } \sin(a \cdot x^4 + b \cdot x)$$

Для отримання такого ж результату способом «реального часу» слід:

1. Ввести вихідний вираз. На панелі обрати дію *Substitute*. В маркер ввести $k=ax^2$. На екрані з'явиться результат:

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

Заміну змінних можна застосувати для перевірки отриманих відповідей. Наприклад, в якості розв'язання інтегрального рівняння $\int_0^x \sin(t) dt = a$ отримано відповідь $x = \arccos(a-1)$. Для перевірки можна:

1. Записати відповідь $x = \arccos(a-1)$ та скопіювати її в кліпбоард.

2. Записати рівняння $\int_0^x \sin(t) dt$, виділити змінну x , викликати функцію *Substitute*.

3. На екрані з'явиться відповідь

$$\int_0^x \sin(t) dt \text{ by substitution, yields } a$$

Тобто $x = \arccos(a-1)$ забезпечує значення інтеграла a .

Виведення алгебраїчної форми виразу

Для отримання виразу, в якому явно виділені дійсна та уявна частини комплексу слугує функція *rectangular*. Наприклад,

$$\ln(1+i) \text{ rectangular} \rightarrow \frac{\ln(2)}{2} + i \cdot \left(\frac{\pi}{4}\right)$$

Приведення подібних

Приведення подібних для елементарних функцій проводить дія *combine*. В якості елементарних функцій можна використовувати арктангенс, синус/косинус, логарифмічну функцію, гіперболічні синус/косинус. Їх визначають пояснюючі ключові слова: *atan*, *exp*, *ln*, *log*, *sincos*, *sinhcosh* відповідно. Пояснююче слово додається через кому наприкінці. Наприклад,

$$\text{atan}(x) + \text{atan}(y) \text{ combine, atan} \rightarrow -\text{atan}\left[\frac{1}{(x \cdot y - 1)} \cdot (x + y)\right]$$

$$\sinh(x) \sinh(y) \text{ combine, sinhcosh} \rightarrow \frac{\cosh(x+y)}{2} - \frac{1}{2} \cdot \cosh(x-y)$$

2.9.3. Додаткові символні обчислення

Розкладання виразу на алгебраїчний поліном

Функція *collect* намагається перегрупувати алгебраїчний вираз на алгебраїчний поліном з обраною змінною в якості аргументу так, щоб аргумент розташовувався з пониженням ступеню. Введення імені змінної після ключового слова для функцій кількох змінних є обов'язковим.

Наприклад, застосування дії до виразу $(x-4) \cdot (x-7) \cdot x + 99$ функції однієї змінної не потребує додаткового визначення змінної і дає наступну відповідь

$$(x-4) \cdot (x-7) \cdot x + 99 \text{ collect} \rightarrow x^3 - 11x^2 + 28x + 99$$

В результуючому поліномі аргумент x розташовується зі ступенями з «3» до «0» послідовно.

Для функції кількох змінних звернення може бути наступним:

$$x \cdot y + x^2 + y^2 + x^2 \cdot y^2 \text{ collect, } x \rightarrow (y^2 + 1) \cdot x^2 + y \cdot x + y^2$$

$$x \cdot y + x^2 + y^2 + x^2 \cdot y^2 \text{ collect, } y \rightarrow (x^2 + 1) \cdot y^2 + x \cdot y + x^2$$

Примітка. Результат дії *collect* не завжди очікувані. Деякі приклади наведені нижче. В першому прикладі функція дає результат, який не є поліномом. В другому прикладі в результаті не приведені подібні, хоча інша функція - розкладання видає результат не в порядку зменшення ступенів x , але з приведенням подібних.

$$\frac{1}{x} + \frac{1}{x-1} \text{ collect} \rightarrow -\frac{2 \cdot x - 1}{x - x^2}$$

$$\left(x + \frac{y^2}{x}\right)^3 \text{ collect, } x \rightarrow \frac{x^6 + 3 \cdot x^4 \cdot y^2 + 3 \cdot x^2 \cdot y^4 + y^6}{x^3} \quad \left(x + \frac{y^2}{x}\right)^3 \text{ expand} \rightarrow 3 \cdot x \cdot y^2 + x^3 + \frac{3 \cdot y^4}{x} + \frac{y^6}{x^3}$$

Розрахунок коефіцієнтів поліному

Для випадків, коли не потрібен весь алгебраїчний поліном, в який розкладається вираз, а потрібні тільки його коефіцієнти, слугує функція *coeffs*. Функція повертає вектор, в якому за збільшенням ступеню аргументу розташовані коефіцієнти від початкового ступеню аргументу на першому місці до найбільшого – на останньому. Викликається функція *coeffs* аналогічно інструкції *collect*.

Наприклад, добуток з прикладу, що наведено вище відбивається на поліном $x^3 - 11x^2 + 28x + 99$. Тобто, має коефіцієнти 1 -11 28 99. Виклик функції дає наступний результат

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

Всього вектор містить чотири елементи, тобто поліном має третій ступінь. Перший елемент вектора 99 є коефіцієнтом для x^0 , другий 28 – для x , третій -11 для x^2 , останній 1 для x^3 .

Розкладання в ряд Тейлора

Для отримання розкладання функції в ряд Тейлора призначена функція *series*. Викликається функція *series* аналогічно інструкції *collect*. Після функції слід визначити через кому ім'я змінної по якій проводиться розкладання в ряд. За замовчанням розкладання проводиться в околі точки 0. Після ключового слова можна визначити максимальний ступінь доданків ряду або значення аргументу, навколо якого буде проведено розкладання.

Наприклад, для розкладання в ряд Тейлора експоненційної функції за замовчанням в околі точки 0 слід наступним чином застосувати функцію *series*

$$e^x \text{ series} \rightarrow 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120},$$

для отримання 3-х доданків виклик має виглядати наступним чином

$$e^x \text{ series, 3} \rightarrow 1 + x + \frac{x^2}{2},$$

для того, щоб розкладання пройшло в околі точки $x=2$:

$$\exp(x) \text{ series, } x = 2 \rightarrow e^2 + e^2 \cdot (x - 2) + \frac{e^2 \cdot (x - 2)^2}{2} + \frac{e^2 \cdot (x - 2)^3}{6} + \frac{e^2 \cdot (x - 2)^4}{24} + \frac{e^2 \cdot (x - 2)^5}{120}.$$

Ланцюги обчислень

Система далеко не завжди видає очікуваний результат. Часто доводиться його повторно спростувати чи перетворювати. Для цього можна вручну скопіювати отриманий вираз та застосувати потрібну команду, але можна й запустити кілька символічних дій послідовно. Додавання додаткових дій проводиться вставлянням рядка дії кнопкою панелі після основної дії.

Наприклад, потрібно розкласти експоненційну функцію в ряд Тейлора з трьома доданками та обчислити її значення при $x=2$.

Можливо послідовне застосування дій розкладання в ряд Тейлора з кількістю доданків 3 функцією *series* та заміна змінної функцією *substitute*:

Такий же результат можна отримати наступним чином:

1. Ввести експоненційний вираз, ввести шаблон функції *series*, визначити кількість доданків 3.

2. Поки блакитний курсор – кут знаходиться на правому кінці рядка *series*

e^x series, 3 | →

клацнути по позначці *Substitute* на панелі *Symbolic*

e^x | series, 3
substitute, | = | →

3. Заповнити маркер рядка *Substitute*. На екрані з'явиться результат

e^x | series, 3
substitute, x = 2 → 5

Запис виразу через елементарні функції

Отримання виразу, в якому вихідний вираз представлений через елементарні функції: **exp**, **ln**, **log**, **sincos (sin, cos)**, **sinhcosh (sinh, cosh)**, - забезпечує функція *rewrite*. Функція, через яку буде проводитися опис, визначається ключовим словом через кому після слова *rewrite*. Наприклад,

$\operatorname{asinh}(x)$ rewrite, ln → $\ln[x + \sqrt{x^2 + 1}]$.

Розкладання виразу на суму правильних дробів

Для розкладання поліноміального виразу на суму правильних дробів слугує функція *parfrac*. Кількість доданків можна визначити після коми числом. Якщо в виразі присутні кілька змінних, то обов'язковим є уточнення імені змінної, для якої буде проводитися дія. За замовчанням ряд містить раціональні корені виразу. За необхідності врахування ірраціонального значення, слід додатково вписати його через кому.

Наприклад,

$$\frac{1}{(x^2 + 1)(x^2 - 1)} \operatorname{parfrac} \rightarrow \frac{1}{4 \cdot (x - 1)} - \frac{1}{2 \cdot x^2 + 2} - \frac{1}{4 \cdot x + 4}$$

$$\frac{3a}{(x^2 - 8x + 15)} \operatorname{parfrac}, x \rightarrow \frac{3 \cdot a}{2 \cdot (x - 5)} - \frac{3 \cdot a}{2 \cdot x - 6}$$

$$\frac{1}{3(x^2 - 2)} \operatorname{parfrac}, \sqrt{2} \rightarrow \frac{\sqrt{2}}{12 \cdot (x - \sqrt{2})} - \frac{\sqrt{2}}{12 \cdot x + 12 \cdot \sqrt{2}}$$

Вимкнення змінної з розрахунку

При використанні способу «реального часу» коли змінним попередньо задані якісь значення, система обраховує вирази з урахуванням цих чисельних значень. Тобто, результат може виявитись чисельним. Для того, щоб виключити кількісне значення змінних з символічних перетворень застосовується функція *explicit*. Імена змінних, які вважаються символічними, перераховуються через кому після ключового слова.

Наприклад, якщо змінні попередньо визначені як $a:= 3.7$ $b:= 12.1$ $c:= \sin(1.2\pi)$, то символічне обрахування дасть в результаті обрахунку число

$$2 \cdot a + b - c \rightarrow 20.0877852522924731292$$

Для того, щоб отримати вираз без значень змінної «с» слід ввести:

$$2 \cdot a + b - c \text{ explicit } a, b \rightarrow 2 \cdot 3.7 + 12.1 - c$$

Наприклад, якщо змінній «х» завдано попереднє значення $x:= 5$, то розкладання на множники дасть в результаті обрахунку число

$x^2 - 3x + 2$ **factor** $\rightarrow 22 \cdot 3$. Для того, щоб отримати вираз без значень змінних слід ввести:

$$x^2 - 3x + 2 \left| \begin{array}{l} \text{explicit} \\ \text{factor} \end{array} \right. \rightarrow (x - 1) \cdot (x - 2)$$

Обчислення виразу у вигляді правильного нескінченного дроби

Для запису виразу у вигляді нескінченного дроби (рис. 2.66) використовується функція *confrac*.

$$a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \dots}}$$
$$\dots + \frac{1}{a_{n-1} + \frac{1}{a_n}}$$

Рис. 2.66 Нескінчений дріб

Результатом є вектор $[a_1 \ a_2 \ a_3 \ \dots \ a_n]$. За замовчанням враховується 10 кроків ділення. Змінити кількість можна визначенням цілого числа після коми. Для того, щоб вивести не вектор, а сам дріб, слід додатково ввести ключове слово *fraction*.

Наприклад,

$$\frac{14 + \sqrt{5}}{3} \text{ confrac, 3} \rightarrow \begin{pmatrix} 5 \\ 2 \\ 2 \\ 2 \end{pmatrix} \frac{14 + \sqrt{5}}{3} \text{ confrac, 3, fraction} \rightarrow 5 + \left[2 + \frac{1}{\left(2 + \frac{1}{2} \right)} \right]$$

Обчислення при додаткових умовах

Система допускає виконання послідовності символічних дій та встановлення додаткових умов при виконанні перетворень.

Для визначення обмежень при перетвореннях застосовується ключове слово *assume*. Слово вставляється в наступний після рядка основної дії рядок кнопкою з панелі *Symbolic*.

З *assume* застосовуються наступні логічні вирази: умови більшості або меншості ($>$, $<$, \geq , \leq), рівності значенню, рівності типу (*real*, *integer*, *complex*), рівності типу в діапазоні $a < x < b$ (*RealRange(a,b)*). Типи обираються з панелі *Modifier* (рис. 2.64) або вводяться вручну. Декілька умов записуються через кому. Якщо умова стосується всіх змінних виразу, застосовується службове слово *ALL*. Наприклад,

$$\sqrt{x^2} \left| \begin{array}{l} \text{simplify} \\ \text{assume, } x = \text{real} \end{array} \right. \rightarrow |x| \quad \sqrt{x^2 + y^2} + \sqrt{z^2} \left| \begin{array}{l} \text{simplify} \\ \text{assume, ALL} > 0 \end{array} \right. \rightarrow x + y + z$$

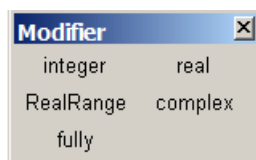


Рис. 2.64. Панель Modifier

2.9.4. Контрольні питання

Що таке символічні обчислення?

Які способи виконання символічних обчислень реалізовані в системі?

Як визначається вигляд запису при «швидких» перетвореннях?

Перерахуйте команди, які запускаються з панелі *Symbolic*.

Як згруповані дії в меню *Symbolics*?

Яким чином «в реальному часі» провести інтегрування, диференціювання?

В чому особливість спрощення дійсних чисел?

Для яких чисел застосовується розкладання на множники?

Для чого проводиться заміна змінної?

В чому різниця дій функції *collect* та *coeffs*?

Як визначається кількість доданків в розкладанні в ряд Тейлора?

Що таке ланцюги обчислень?

Для якого способу призначена функція *explicit*?

Для чого призначена функція *assume*?

3. ПРАКТИКУМ ДО РОЗДІЛУ 2

3.1. Робота з документами

Приклад 3.1.1. Створити в *MathCAD* сторінку (рис. 3.1). Малюнок вставити з папки *MathCAD template*. Результат зберегти в web форматі.

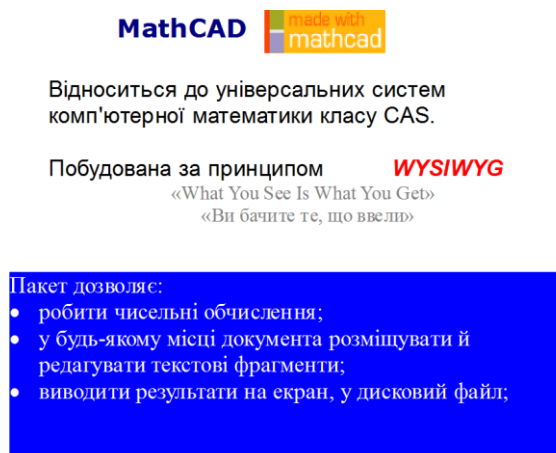


Рис. 3.1. Макет документа прикладу 3.1.1

РОЗВ'ЯЗАННЯ

Сторінку можна створити з 5-ти до 2-х текстових областей. Оберемо - 3. Перша має шрифт Verdana 16 темно-синього кольору. Друга – шрифти Arial 14 чорного кольору, Arial bold italic 16 червоного кольору, Times New Roman 14 сірого кольору. В ній розташовані два абзаци. Вирівнювання другого абзацу – по центру. Третя – шрифт Times New Roman 14 білого кольору, синій фон. В ній використано список.

Форматувати текстову область можна перед її створенням за рахунок визначення стилів документу або вже після введення тексту. Покажемо обидва способи.

1. Обрати пункт меню *Format – Style*.
2. Обрати *Text styles – Heading1 – Modify - Font*. У вікні визначити: Font – *Verdana*, Font size - 16, Font style - *полужирный*, Color – *темно синий* (рис. 3.2).
3. Обрати *Text style – List- Font*. У вікні визначити: Font – *Times New Roman*, Font size - 14, Font style - *нормальный*, Color – *белый*.
4. Обрати *Text style – Normal- Font*. У вікні визначити: Font – *Arial*, Font size - 14, Font style - *нормальный*.

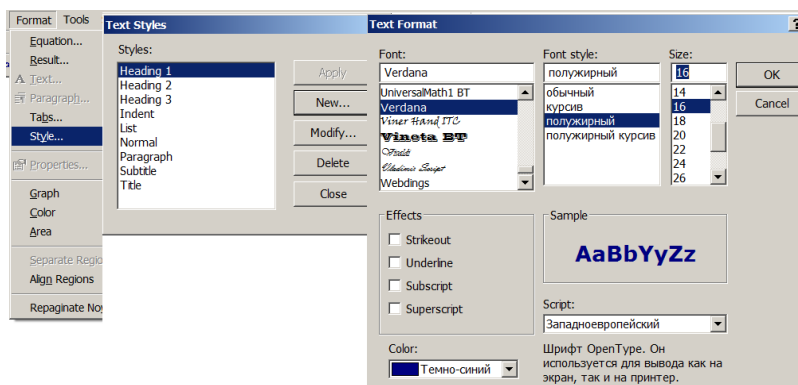


Рис. 3.2. Визначення стилю Heading1

5. Встановити курсор на місце першого рядка. На панелі *Formatting* обрати стиль *Header1*. Ввести «MathCAD».

6. Відкрити графічною програмою зображення з папки «*./MathCADxx/template/madewithmathcad.gif*», скопіювати його в *Clipboard* та вставити документ *MathCAD*. Відкрити через контекстне меню сторінку *Protect* властивостей та відмінити блокування редагування (*Protect region from editing*). Розташувати малюнок в потрібне місце та надати йому потрібного розміру (рис. 3.3).

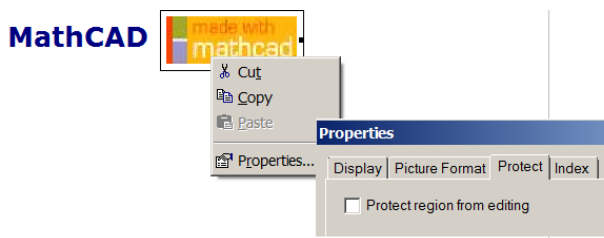


Рис. 3.3. Перший рядок документа

7. Встановити курсор на місце другого рядка. На панелі *Formatting* обрати стиль *Normal*.

8. Ввести (рис. 3.4).

«Відноситься до універсальних систем

комп'ютерної математики класу CAS.

Побудована за принципом WYSIWYG

«What You See Is What You Get» «Ви бачите те, що ввели»».

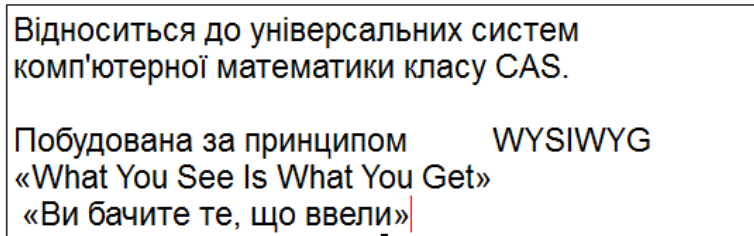


Рис. 3.4. Друга область

9. Мишею виділити *WYSIWYG*. В контекстному меню обрати пункт *Font*. Визначити стиль шрифту – *полужирный курсив*, колір – *красный*.

10. Мишею виділити останні два рядка. В контекстному меню обрати пункт *Font*. Визначити шрифт - *Times New Roman*, стиль шрифту – *полужирный курсив*, колір – *темно-серый*.

11. На панелі *Formatting* обрати вирівнювання – по центру (рис. 3.5).

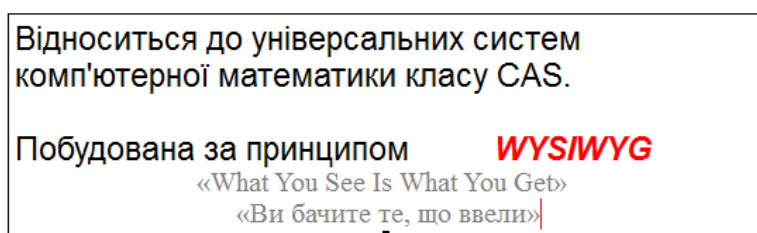


Рис. 3.5. Відформатована друга область

12. Встановити курсор на місце списку. На панелі *Formatting* обрати стиль *List*. Створити текстову область. В контекстному меню *Properties – Display – Background* позначити поле *Highlight region*, кнопкою *Choose color* обрати колір – *blue*. Ввести «**Пакет дозволяє:**» (рис. 3.6).

A screenshot of a blue highlighted text area. The text inside is "Пакет дозволяє:". The text is in a white font.

Рис. 3.6. Третя область

13. На панелі *Formatting* включити кнопку створення немаркованого списку. Ввести

«робити чисельні обчислення;

у будь-якому місці документа розміщувати й редагувати текстові фрагменти;

виводити результати на екран, у дисковий файл.»

14. Зберегти документ як *Web page*.

Завдання 3.1.1.

1. Створити документ. В документі:

- Вертикально розмістити дві текстові області.
- Перша область повинна містити умови цього завдання. Колір шрифту – чорний. Перший рядок – Arial 16, bold. Заголовок списку -Arial 14, список - Arial 14, italic.
- Друга область – 1-й рядок «Виконав студент(ка) ПІБ» . Другий рядок «Група ХХ-ХХ дата». Колір – синій. Шрифт – тексту Courier 14, ПІБ та номер групи, дата – Courier 14, bold. Вирівнюванн – по правому краю.

5. Документ зберегти в форматі *Word*.

3.2. Організація обчислень

Приклад 3.2.1. Для значення $x=2$ обрахувати вираз з функцією Беселя першого роду $J_0(x)$ та гіпергеометричною функцією $\Gamma(x)$


$$J_0(x) + \frac{\Gamma(x)}{\sin(x)} e^{2-5i}.$$

РОЗВ'ЯЗАННЯ

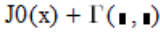
$x:=2$ - визначаємо змінну-аргумент виразу

Функції можна вводити з клавіатури, з меню «*Insert - Function*» або кнопкою $f(x)$ панелі *Standard*.

Натиснути кнопку $f(x)$ панелі *Standard* → обрати в списку з групи

Bessel функцію Беселя першого роду →  → вставити на місце маркера

змінну x , додати дію →  → в меню *Insert - Function* з групи **Special**

обрати функцію **Gamma** →  → видалити другий маркер, на місце першого маркера поставити аргумент x → завершити вираз

$$J_0(x) + \frac{\Gamma(x)}{\sin(x)} e^{2-5i} = 2.529 + 7.792i.$$

Приклад 3.2.2. Розрахувати суму. Результат вивести в десятковій, бінарній, восьмеричній, шістнадцятиричній системах. Доданки; 1110_2 , 21_8 , $3F_{16}$

РОЗВ'ЯЗАННЯ

Звичайно, можна перевести доданки у відповідну систему та скласти.

Для десяткової системи це буде

$$1110_2=14 \quad 21_8=17 \quad 3F_{16}=63 \quad 14+17+63=94.$$

Робити це немає потреби. Записати суму та отримати результат за замовчанням в десятковій системі

$$1110b+21o+3Fh= 94.$$

Обрати в меню *Format – Result – Display options - Radix – Octal* та отримати результат в восьмеричній системі

$$1110b+21o+3Fh= 136o.$$

Обрати в меню *Format – Result – Display options - Radix – Hexadecimal* та отримати результат в шістнадцятиричній системі

$$1110b+21o+3Fh= 5eh.$$

Обрати в меню *Format – Result – Display options - Radix – Binary* та отримати результат в бінарній системі

$$1110b+21o+3Fh= 1.1bx10b110b.$$

Примітка. Результат видано в двійковій системі, але вигляд його незручний. Причиною цього є те, що за замовчанням виставлено формат результату порогом переходу в експоненційну форму 3 цифри. В результаті їх більше, тому форма – експоненційна.

Обрати в меню *Format – Result – Number Format – Exponential Threshold* значення 8 та отримати результат

$$1110b+21o+3Fh= 1011110b.$$

Приклад 3.2.3. Формати відображення результату.

РОЗВ'ЯЗАННЯ

$$2^3 + \sqrt{3} = 9.732$$

$2.7 + 5/6 = 3.5333$ - результат у форматі «Decimal» із чотирма знаками.

$2.7 + 5/6 = 3.53 \times 10^0$ - результат у форматі «Scientific» із двома знаками

$$2^3 + \sqrt{3} = 9.732E + 000$$
 - результат у форматі «Engineering»

Приклад 3.2.4. Розрахувати вираз згідно наведеної формули

$$h(r) = \left[\frac{2I_1\left(\frac{2\pi D_{зр} r}{\lambda f'}\right)}{\frac{2\pi D_{зр} r}{\lambda f'}} \right]^2$$

Розрахунок провести для $\lambda = 5.5 \text{ мкм}$, $D_{зр} = 1 \text{ мм}$, $f' = 25 \text{ мм}$. Діапазон зміни аргументу r – від 0 до 1 мм з кроком 0.1 мм.

РОЗВ'ЯЗАННЯ

$D := 1$ $\lambda := 0.55 \cdot 10^{-2}$ $f := 25$ - задаємо вихідні дані. Дані масштабовано так, щоб всі вони становили значення в міліметрах ($1 \text{ мкм} = 10^{-3} \text{ мм}$)

$$x := \frac{(2 \cdot \pi \cdot D)}{(\lambda \cdot f)}$$
 - уводимо проміжну змінну

$$h(r) := \left(2 \cdot \frac{J_1(x \cdot r)}{(x \cdot r)} \right)^2$$
 - задаємо функцію користувача

$r := 0, 0.1.. 1$ - уводимо ранжовану змінну r як аргумент для розрахунку з першим значенням 0, кроком 0.1, останнім значенням 1

r =	h(r) =
0	0
0.1	0.949
0.2	0.808
0.3	0.613
0.4	0.405
0.5	0.226
0.6	0.098
0.7	0.027
0.8	$1.562 \cdot 10^{-3}$
0.9	$2.751 \cdot 10^{-3}$
1	0.012

- виводимо таблиці результату. Перший стовпчик - вектор із десятих значень аргументу r, другий стовпчик - вектор із десятих значень результату h (r).

Примітка: Зверніть увагу на НЕВІРНЕ з точки зору математики значення h(0). Поясніть можливу причину такого результату.

Завдання 3.2.1.

а) Для комплексних змінних $6+3i$ та $9 \cdot 1e^{-i0.33}$ визначити дійсну, уявну частини, модуль, фазу.

б) Обчислити значення z_4 , записати результат в алгебраїчній та експоненціальній формах.

z_1	z_2	z_3		z_4
$4+3i$	$1.82 \cdot e^{-1.2i}$	$\sqrt{3}-2i$	$z_4 = \frac{z_1^2 \cdot z_2}{z_3} + z_3$	$11.8182 + 11.9291i$

Завдання 3.2.2. Обчислити вирази. Порівняти отримані результати з наведеними, визначити формат результату.

а)
$$\frac{23.1 \cdot \sqrt{3\pi - \frac{\ln(54)}{16}}}{5 - \pi^{4i}} = -81.9085 \times 10^{-12}$$

б)
$$e^{-2.5} \ln(11.3)^{0.3} - \sqrt{\frac{\sin(2.45\pi) + \cos(3.78\pi)}{\operatorname{tg}(3.3)}} = -3.21$$

в)
$$\frac{2,625 - \frac{2}{3} \cdot 2 \frac{5}{14}}{(3 \frac{1}{12} + 4,375) \div 19 \frac{8}{9}} = 2.80952 \times 10^0$$
 г)
$$\frac{\cos(\frac{2\pi}{6}) \sin(\frac{\pi}{3})}{\operatorname{tg}(\frac{2\pi}{9}) - \lg(44)} = -0.156$$

Завдання 3.2.3. Для визначених значень змінних обчислити вирази. Порівняти отримані результати з наведеними, визначити формат результату.

$$\text{а) } \frac{a^3\sqrt{x} - \sin(x+a)}{e - |x-a|} = -0.149 \text{ для } x=0.5 \text{ а}=0.7$$

$$\text{б) } \frac{3\sqrt{a} \cdot \ln|a|}{\sin(e^{2x})\cos(xa^3)} = -2.181 \text{ для } x=0.5 \text{ а}=0.7$$

$$\text{в) } \frac{\sin\left(\frac{\lambda\pi}{D}\right)}{\frac{\lambda\pi}{D}} = 0.999 \text{ для } \lambda=0.9 \text{ D}=50$$

Примітка: функція має значення 1.0 тільки для аргументу 0.

$$\text{г) } \sqrt{\frac{4}{e^x}} - \text{cth}^3(x) \cos(|x \sin(x^2) - \ln(x)|) = -0.559 \text{ для } x=3$$

$$\text{д) } \frac{b^3 + \sin^2(a)}{\arccos(b^2) + e^{\frac{b}{2}}} = 239 / 605 \text{ для } a=83^0 \text{ b}=0.12$$

Завдання 3.2.4. Застосовуючи функції користувача для спрощення записів, обрахувати вираз для значень $x=0.1$, $y=1.2$.

$$\text{а) } \text{arctg}\left(\frac{\sqrt[3]{x - \sin(y)}}{\sqrt{1-x^2}} - \frac{|x|\sqrt{1-x^2}}{\sqrt[3]{x - \sin(y)}}\right)$$

$$\text{б) } \frac{(\sin(y) + \sin(2y) + \sin(3y))^4}{1 + \frac{\sin(y) + \sin(2y) + \sin(3y)}{1 + e^x}} + \sqrt{\frac{1 + \sin(y) + \sin(2y) + \sin(3y)}{1 + e^x}}$$

$$\text{в) } \frac{\sqrt{\cos(2y) + \sin(4y) + \sqrt{e^x + e^{-x}}}}{(e^x + e^{-x})^3 (\cos(2x) + \sin(4y) - 2)^2} \quad \text{г) } \frac{1 + \arcsin(\cos(2y))}{2^x + 3^{-x}} + \left(\frac{2^x + 3^{-x} - 1}{1 + \arcsin(\cos(2y))}\right)^2$$

Завдання 3.2.5. Обрахувати вираз з застосуванням розширених арифметичних операторів

$$\text{а) } \int_0^2 \frac{x}{x^2 - 1} dx \quad \text{б) } \int_1^4 \int_3^5 (2x^4 + 13x^5) dy dx \quad \text{в) } \sum_{i=3}^{20} e^{-\frac{i}{2}}$$

Завдання 3.2.6. Розрахувати значення функції в заданому діапазоні

$$\text{а) } \frac{\sin^2(x)}{1 + \cos(x)} + e^{-x} \ln(x) \text{ діапазон : } x= 0.2 \dots 2.4 \text{ з кроком } 0.2$$

б) $e^{-x} \sin(10x)$ діапазон: $x=0 \dots 1$ з кроком 0.1

в) $\cos(x-y)+\sin^2(x+y)$ діапазон: $x=0.1 \dots 1$, крок 0.4, $y=1 \dots 2$ крок 0.3.

Завдання 3.2.7. Модифікувати функцію користувача з *прикладу 3.2.4* так, щоб вона мала додатково за аргумент f . Розрахувати значення для $f=(25, 50, 100)$ мм, $\lambda=0.55$ мкм, $Dzr=1$ мм. Діапазон зміни аргументу $r=[0..0.1]$ мм з кроком 0.01мм. Таблиці значення аргументу r та результати для $f=(25, 50, 100)$ мм розташувати в одному рядку.

3.3. Перевірка умов

Приклад 3.3.1. Описати функцію

$$sa(x) = \frac{\sin(x)}{x} \quad \text{без помилки при } x=0.$$

РОЗВ'ЯЗАННЯ

$$sa(x) := \text{if}(x = 0, 1, \frac{\sin(x)}{x})$$

Чисельна перевірка $f(0) = 1$.

ВІРНИЙ опис функції. За допомогою умовного оператора невизначеність ліквідовано. Саме за таким алгоритмом працює вбудована функція $\text{sinc}(x)$.

Приклад 3.3.2. Описати прямокутну функцію

$$\text{rect}(x, a) = \begin{cases} 0 & |x| > \frac{a}{2} \\ 1 & |x| \leq \frac{a}{2} \end{cases}.$$

РОЗВ'ЯЗАННЯ

$$\text{rect}(x, a) := (\Phi(x + \frac{a}{2}) \cdot \Phi(\frac{a}{2} - x)) \quad \text{- з використанням функції Хевісайда}$$

$$\text{rect}(x, a) := \text{if}(-\frac{a}{2} < x < \frac{a}{2}, 1, 0) \quad \text{- з використанням функції «if»}$$

$$\text{rect}(x, a) := \text{if}(|x| > \frac{a}{2}, 0, 1) \quad \text{- функція «if» з скороченою умовою}$$

$$\text{rect}(x, a) := (|x| \leq \frac{a}{2}) \quad \text{- з використанням унітарного логічного оператора}$$

Приклад 3.3.3. Описати функцію $y(x)$

$$y(x) = \begin{cases} \frac{\sqrt{x}}{2} & x \geq 1 \\ \frac{\sqrt[3]{x}}{3} & 0 < x < 1 \\ \frac{\sqrt[4]{|x|}}{4} & x \leq 0 \end{cases} .$$

РОЗВ'ЯЗАННЯ

$$y(x) := \text{if}(x \geq 1, \frac{\sqrt{x}}{2}, \text{if}(x > 0, \frac{\sqrt[3]{x}}{3}, \frac{\sqrt[4]{|x|}}{4}))$$

Приклад 3.3.4. Описати функцію (рис. 3.7) із застосуванням логічної функції та функції Хевісайда.

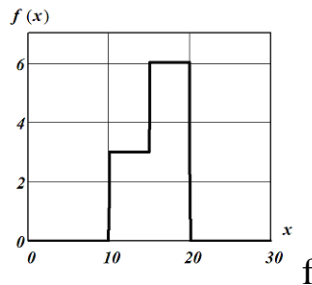


Рис. 3.7. Графік функції з прикладу 3.3.4

РОЗВ'ЯЗАННЯ

$f(x) := \text{if}(x < 10, 0, \text{if}(x < 15, 3, \text{if}(x < 20, 6, 0)))$ – логічна функція

$f(x) := 3(\Phi(x-10) + \Phi(x-15) - 2\Phi(x-20))$ - функція Хевісайда

Приклад 3.3.5. Описати функцію, яка повертає значення 1, якщо точка з координатами x, y знаходиться всередині фігури (рис. 3.8), 0 - якщо ні.

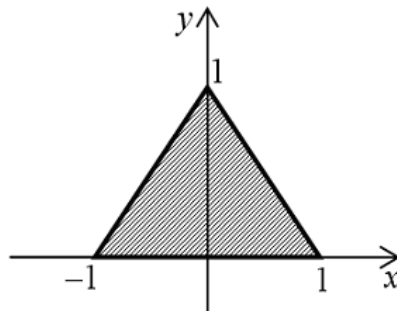


Рис. 3.8. Область з прикладу 3.3.5

РОЗВ'ЯЗАННЯ

Область обмежена прямими $y=0$, $y=x+1$, $y=1-x$ та визначається умовами $y>0$ та $y<x+1$ та $y<1-x$ одночасно.

$$z(x,y) := (y > 0) \wedge (y < x + 1) \wedge (y < 1 - x) \text{ або}$$

$$z(x,y) := \text{if}((y > 0) \wedge (y < x + 1) \wedge (y < 1 - x), 1, 0)$$

Завдання 3.3.1. Описати функцію одиничної сходинки.

Завдання 3.3.2. Описати функцію (рис. 3.9) із застосуванням логічної функції та вбудованої функції Хевісайда.

Завдання 3.3.3. Описати функцію, яка повертає значення 1, якщо точка з координатами x, y знаходиться всередині фігури (рис. 3.10), та 0 - якщо ні.

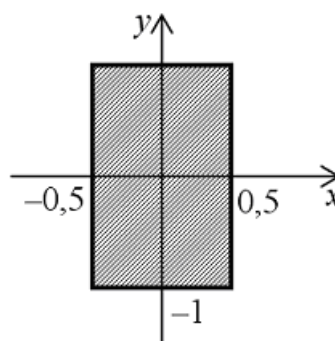
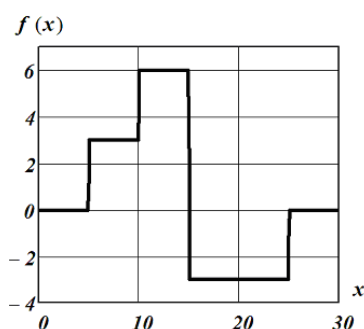


Рис. 3.9 Функція до завдання 3.3.2 Рис. 3.10. Область до завдання 3.3.3.

Завдання 3.3.4. Описати функцію

$$\text{а) } f(x,y) = \begin{cases} \sin(x+y^2) - e^{2x} & xy < 2 \\ \ln(x + \sqrt[3]{|y|}) & 2 < xy < 3 \\ \cos(x+y^3) + e & xy > 3 \end{cases} \quad \text{б) } f(x,y) = \begin{cases} \min(\frac{\cos(x^2)}{y}, \sqrt[5]{y} \sin(\frac{x}{y})) & x < y \\ \max(\sqrt{|\sin(xy)|}, \cos \frac{\sqrt{x}}{y+1}) & x \geq y \end{cases}$$

3.4. Робота з масивами

Приклад 3.4.1. Створити вектор значень функції $e^{-x} \sin(10x)$ в діапазоні $x=0 \dots 1$ з кроком 0.1.

РОЗВ'ЯЗАННЯ

На визначеному діапазоні функція має $N = \frac{X_{\max} - X_{\min}}{\text{step}} + 1 = \frac{1-0}{0.1} + 1 = 11$

відліків.

$$x_{\text{start}} := 0 \quad x_{\text{stop}} := 0 \quad \text{step} := 0.1$$

$$N := \frac{x_{\text{stop}} - x_{\text{start}}}{\text{step}} + 1$$

Визначаємо вихідні дані

Визначаємо довжину вектору

$$i := 1..11$$

Визначаємо ранжовану змінну

$$x_i := x_{\text{start}} + (i - 1) \cdot \text{step}$$

Вводимо проміжний вектор для того, щоб мати змогу присвоїти значення новій змінній - елементу вектора

$$y_i := e^{-x_i} \cdot \sin(10 \cdot x_i)$$

	1
1	0
2	0.761
3	0.744
4	0.105
5	-0.507
6	-0.582
7	-0.153
8	0.326
9	0.445
10	0.168
11	-0.2

Примітка. Ефективніше можна створити вектор-результат з використанням операції векторизації

$$z := \overrightarrow{\left(e^{-x} \cdot \sin(10 \cdot x) \right)}$$

Приклад 3.4.2. Знайти суму додатних елементів матриці A розміром 3x3.

РОЗВ'ЯЗАННЯ

1. За допомогою вікна створення матриць визначаємо матрицю A

A: →Ctrl+M→3 3→ $A := \begin{pmatrix} \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \end{pmatrix}$ → на місце маркерів вводимо значення → $A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 1 \end{pmatrix}$.

2. Для розрахування суми елементів застосуємо розширений оператор підсумування. Треба перевірити всі елементи матриці, тому використовуються два оператори. Межі в операторах підсумування

визначаємо по першому та останньому індексам матриці A . Суму будемо накопичувати в змінній S . Перевірку додатності елементів матриці A реалізує умовна функція `if`.

$$S := \sum_{i=1}^3 \sum_{j=1}^3 (\text{if}(A_{i,j} > 0, A_{i,j}, 0)) = 37$$

Приклад 3.4.3. Для матриці $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 11 \\ 7 & 8 & 9 \end{bmatrix}$ визначити основні

параметри.

РОЗВ'ЯЗАННЯ

Основними параметрами матрицями слугують

детермінант $|A| = 24$

слід $\text{tr}(A) = 7$

ранг $\text{rank}(A) = 3$

норма $\text{norme}(A) = 14.318$

Завдання 3.4.1. Визначити вектор довжиною 6 $x_i = \frac{1}{i^2 + 1}$.

а) Переставити елементи в зворотному порядку вбудованою функцією `rev` та за допомогою поелементного звертання.

б) Виділити в новий вектор елементи з парними номерами.

в) Замінити всі мінімальні елементи на максимальні.

Завдання 3.4.2. Визначити матрицю $B = \begin{bmatrix} 11 & -11 & 3 \\ 2 & -2 & 7 \\ 9 & 3 & 3 \end{bmatrix}$.

а) Обрахувати детермінант та зворотну матрицю.

б) Розрахувати параметр $r = \sqrt{\sum_{i,k=1}^N a_{i,k}^2}$

Завдання 3.4.3. Обчислити вираз $2 \cdot A \cdot B - 3 \cdot C \cdot D$, де:

$$A = \begin{bmatrix} 2 & 3 & -1 & -2 & 4 \\ 10 & 3 & -2 & -1 & 3 \\ 5 & 1 & 1 & -4 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & -1 \\ 1 & -2 \\ -1 & 0 \\ 1 & 2 \\ -2 & 3 \end{bmatrix} \quad C = \begin{bmatrix} 1 & -2 & 1 & 4 \\ -3 & 1 & 0 & 5 \\ 2 & 0 & 3 & -1 \end{bmatrix} \quad D = \begin{bmatrix} 1 & -1 \\ -2 & 0 \\ 3 & -1 \\ 2 & -1 \end{bmatrix}$$

Завдання 3.4.4. З перших двох рядків матриці D (завдання 3.4.3) сформувати матрицю R, а з двох останніх - матрицю T. Перевірити, чи є матриці перестановочними, тобто, чи виконується умова $R \cdot R^T - T \cdot T^T < 10^{-10}$.

3.5. Програмування

Завдання 3.5.1. Розробити підпрограму для знаходження всіх коренів квадратного рівняння.

Завдання 3.5.2. Розробити підпрограму з трьома аргументами a, b, c. Функція замінює числа їх квадратами якщо $a \leq b \leq c$, замінює числа на максимальне значення з чисел, якщо $a > b > c$, змінює знаки чисел в інших випадках.

Завдання 3.5.3. Розробити підпрограму з двома аргументами – координатами точки - x, y. Функція обраховує номер квадранта, в якому знаходиться точка.

Завдання 3.5.4. Розробити підпрограму з трьома аргументами – довжинами сторін трикутника - a, b, c. Функція повертає 3 якщо трикутник рівносторонній, 2 якщо – рівнобедрений, 1 – всі сторони мають різну довжину.

Завдання 3.5.5. Розробити підпрограму для обрахунку нескінченної суми послідовності $a_i = \frac{1}{\sqrt{i}}$. Сумування закінчити за умови $|a_i| < \varepsilon$.

Завдання 3.5.6. В матриці визначити індекси значення елементів, які найближчі до заданого числа.

Завдання 3.5.7. Сформувати вектор, i -й елемент якого дорівнює кількості додатних елементів в i -му рядку вихідної матриці.

Завдання 3.5.8. В векторі x порахувати добуток додатних елементів, суму елементів в діапазоні $\mathbf{a} < x_i < \mathbf{b}$.

Завдання 3.5.9. Аргумент функції $f(x)$ визначено вектором x . Визначити максимальне значення функції, номер елемента вектора x , який відповідає максимальному значенню функції.

Завдання 3.5.10. В матриці A поміняти місцями максимальний та мінімальний елементи.

Завдання 3.5.11. В матриці на місце першого елемента i -го рядка записати суму елементів цього рядка.

Завдання 3.5.12. За даними з матриці A сформувати вектор x . i -й елемент вектору x має дорівнювати номеру стовбця матриці A , в якому на i -му рядку матриці знаходиться мінімальний елемент.

Завдання 3.5.13. В матриці переставити стовпці в порядку збільшення суми елементів стовпців.

Завдання 3.5.14. В матриці визначити максимальний індекс стовпців та мінімальний рядковий індекс від'ємних елементів.

3.6. Двомірні графіки

Приклад 3.6.1. Побудувати графіки функції $Y(t) := \sin(t)$, $Y1(t) := t^2$. За допомогою функцій збільшення та трасування знайти точки перетину графіків.

РОЗВ'ЯЗАННЯ

1. Визначити функції користувача $y(t) := \sin(t)$ та $y1(t) := t^2$.
2. Визначити діапазон аргументу ранжованою змінною $x := -0.5, -0.49.. 1$.
3. Додати графічну область. В маркері аргументу вставити ім'я ранжованої змінної, в маркері функцій – через кому імена введених функцій користувача (рис. 3.11).

З графіка видно, що криві мають дві точки перетину. Перша точка очевидна – $(0,0)$.

Для уточнення значення другої застосуємо режим збільшення.

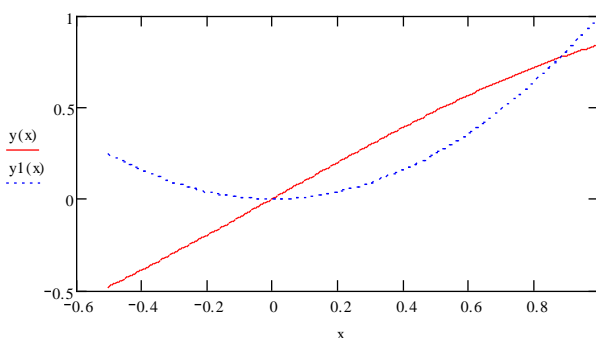


Рис. 3.11. Графіки функцій прикладу 3.6.1

4. Обрати поле *Zoom* контекстного меню графічної області. Вибрати область графіків навколо точки перетину. Можливе зображення наведено на рис. 3.12.

Для уточнення значення координат точки перетину застосуємо режим слідкування (*trace*).

5. Обрати поле *Trace* контекстного меню графічної області. Навести курсор на точку перетину. В полях «X-Value», «Y-Value» спостерігати координати точки перетину. Можливе зображення наведено на рис. 3.13.

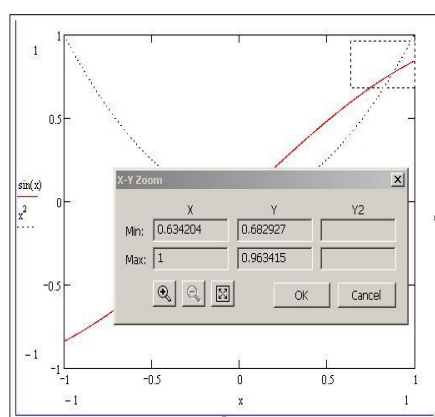


Рис. 3.12. Режим “Zoom”

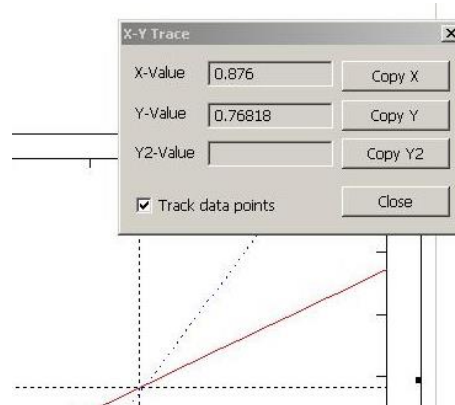


Рис. 3.13. Режим “Trace”

Приклад 3.6.2. Побудувати в одній графічній області графіки спектральних щільностей генераційно – рекомбінаційного $U_{GR}(f) = 10^{-12} \frac{RU}{f}$ та струмового $U_c = 2qR^2$, де $q = 1.6 \cdot 10^{-19}$ Кл – заряд електрону, шумів перетворювача з напругою $U = 10$ В, опором $R = 50$ кОм у смузі частот $[0.1 - 100]$ Гц.

РОЗВ'ЯЗАННЯ

$U := 10$ - значення напруги
 $R := 5 \cdot 10^4$ - значення опору перетворювача
 $R_n := R$ - значення опору навантаження
 $Uf(x) := 10^{-12} \cdot \frac{R_n^2 \cdot U}{x \cdot R}$ - спектральна щільність рекомбінаційного шуму
 $Ut(x) := 2 \cdot 1.6 \cdot 10^{-19} \cdot R^2$ - спектральна щільність струмового шуму
 $f := 0.1, 0.5 \dots 100$

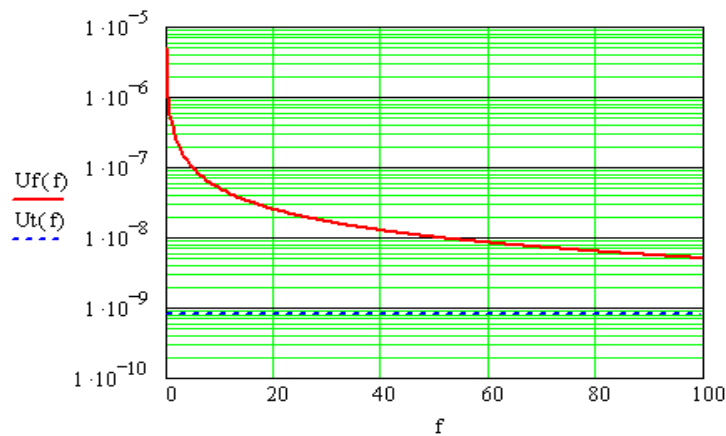


Рис. 3.14. Графік прикладу 3.6.1

Завдання 3.6.1. Для функції в одній графічній області побудувати графіки самої функції, інтегралу від 0.5 до 2, першої та другої похідної

а) $\sin(x)\cos(x)$ б) $\exp(-\frac{x}{2}) + 2\sin(\frac{x}{2})$.

Завдання 3.6.2. Побудувати графіки в двох вікнах, в одному вікні, в одному вікні на двох осях функцій в діапазоні аргументу $0..2\pi$

а) $3 - \cos(x^2)$ б) $2.5\sin(2t)$.

Завдання 3.6.3. Побудувати на одному полі графіки за формулою Планка (приклад 3.8.4) для $T=1000, 1500, 2000$ в діапазоні $[0.1 - 30]$. Використати одну функцію користувача.

Завдання 3.6.4. Побудувати на одному полі в діапазоні аргументу $[0 - 6]$ графіки функцій $y=x$ та $y=x^2$ та позначити область, яка лежить між графіками функцій та її межі.

Завдання 3.6.5. Побудувати графік функцій $r=2(1-\cos(\varphi))$, $r=3\sqrt{2\cos(2\varphi)}$ в полярних координатах.

Завдання 3.6.6. Намалювати коло з радіусом 10 та центром $p(1,3)$.

Завдання 3.6.7. Результати розрахунку координат точок зберігаються в матриці Coords. Перший стовпець уміщує координати X точок, другий – Y. Побудувати точкову діаграму (розподіл точок на площині).

	1	-8
	2	6
	3	5
	-4	4.2
Coords :=	5	2
	6	0
	-7	9.99
	8	-6
	9	2
	10	1

Завдання 3.6.8. За допомогою трасування визначити точки перетину функцій: $y=2\cos(x)$, $y=2\sin(x)^2$ та $y=x$ для $x [0..\pi]$.

3.7. Тримірні графіки

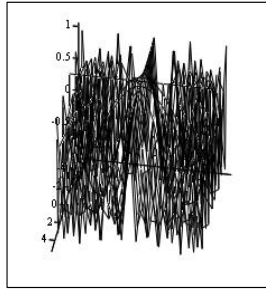
Приклад 3.7.1. Побудувати графік функції $z(x,y)=\cos(x \cdot y)$ в діапазоні аргументів $-2 .. 2$ із кроком 0.12 стандартним способом та з функцією *CreateMesh*.

РОЗВ'ЯЗАННЯ

1. Розрахувати кількість інтервалів аргументу $N = \frac{2 - (-2)}{0.12} = 33$

2. Визначити функцію користувача **$z(x, y) := \cos(x \cdot y)$** .

3. В маркер тримірного графіка вставити ім'я функції. Зображення за замовчанням (рис. 3.15) не відповідає завданню як за виглядом, так і за діапазонами аргументів та кількістю інтервалів на діапазоні.

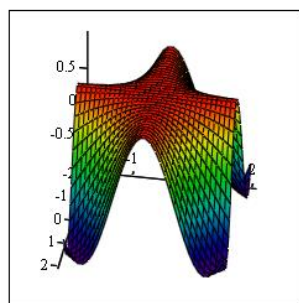


z

Рис. 3.15. Первинне зображення

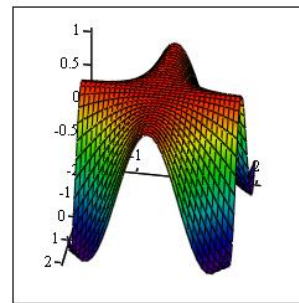
4. Змінити тип фарбування на вкладниці *Appearance* на **Fill Surface** та **Colormap**, визначити на вкладниці *Quick Plot* діапазони обох аргументів на $[-2, 2]$ та кількість ліній 33. Зображення набуває потрібного вигляду (рис.3.16а).

5. В маркер нового поля графіка ввести функцію **CreateMesh** з визначеними аргументами. Змінити тип фарбування на вкладниці *Appearance* на **Fill Surface** та **Colormap**. Зображення стає аналогічним п.4 (рис. 3.16б)



z

а



CreateMesh(z, -2, 2, -2, 2, 33, 33)

б

Рис.3.16. Зображення а) після ручного редагування
б) з функцією **CreateMesh**

Приклад 3.7.2. Накреслити поверхневий матричний, контурний та тримірний точковий графіки функції $f(x,y)=0.5-x^2-y^2$ з кроком обох аргументів 0.25 в межах $-1 < x < 1$, $-1 < y < 1$.

РОЗВ'ЯЗАННЯ

1. Визначити розмір матриці $N = \frac{1 - (-1)}{0.25} = 8$.

2. Визначити елементи матриці через індекс у вигляді ранжованої змінної та крок 0.12

$$N := 8 \quad \text{ORIGIN} := \frac{-N}{2} \quad \text{step} := 0.25$$

$$i := \frac{-N}{2} .. \frac{N}{2} \quad j := \frac{-N}{2} .. \frac{N}{2} \quad x_i := i \cdot \text{step} \quad y_j := j \cdot \text{step}$$

$$M_{i,j} := \left[0.5 - (x_i)^2 \right] - (y_j)^2$$

3. Графік поверхні. В поле маркера тримірного графіка вставити ім'я матриці. Зображення наведено на рис. 3.17а.

5. Точкове зображення. В поле маркера тримірного графіка вставити ім'я матриці. На вкладниці *General* встановити тип *Data Points*, на вкладинці *Appearance* встановити *Symbol Dot* розміром 2 та кольором *Red*. Зображення наведено на рис. 3.17б.

4. Контурний графік. В поле маркера тримірного графіка вставити ім'я матриці. На вкладниці *General* встановити тип *Contour Plot*, на вкладинці *Special* встановити фарбування *Fill*, тип *Numbered* з кількістю ліній 5. На вкладниці *Advanced* обрати *Colormap Grayscale*. Зображення наведено на рис. 3.17в.

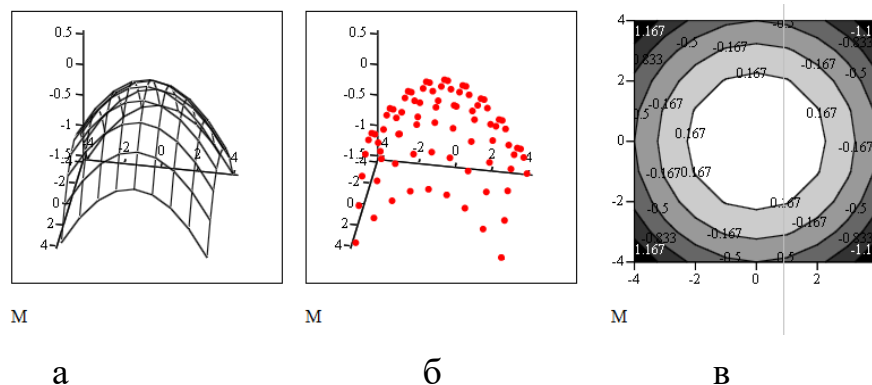


Рис. 3.17. Зображення прикладу 3.7.2

а) матричний графік поверхні; б) точковий графік; в) контурний графік

Приклад 3.7.3.

Рівняння торичної поверхні записується як:

$$y(a,b) = (R_{\max} + R_{\min} \cdot \cos(a)) \cdot \sin(b)$$

$$x(a,b)=(R_{\max}+R_{\min}\cdot\cos(a))\cdot\cos(b)$$

$$z(a,b)=R_{\min}\cdot\sin(a)$$

Діапазон зміни кута $a - [0 \dots \pi]$ $b - [0 \dots 2\pi]$.

Намалювати поверхню тора із секторним вирізом $\pi/8$ по куту b параметричним способом та функцією **CreateMesh**. Великий радіус тора 5, малий - 2. Число точок 20.

РОЗВ'ЯЗАННЯ

1. Визначити вихідні дані **Rmax:=5 Rmin:=2 N:=20**.

2. Параметричний спосіб.

- визначити матриці X, Y, Z. Крок по куту становить $\frac{2\pi}{20} = \frac{\pi}{10}$, тому точно виконати умови завдання неможливо. Для цього способу виріз становитиме один крок.

$$i := 0..N \quad a_i := i \cdot \frac{\pi}{N} \quad j := 0..N-1 \quad b_j := 2 \cdot j \cdot \frac{\pi}{N}$$

$$X_{i,j} := (R_{\max} + R_{\min} \cdot \cos(a_i)) \cdot \cos(b_j)$$

$$Y_{i,j} := (R_{\max} + R_{\min} \cdot \cos(a_i)) \cdot \sin(b_j)$$

$$Z_{i,j} := R_{\min} \cdot \sin(a_i)$$

- в полі тримірного графіка ввести в дужках назви матриць X, Y, Z. Отримане зображення наведено на рис. 3.18а.

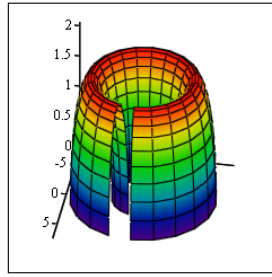
3. Спосіб вбудованої функції.

- визначити векторну функцію користувача $h(a,b)$

$$h(a,b) := \begin{bmatrix} (R_{\max} + R_{\min} \cdot \cos(a)) \cdot \cos(b) \\ (R_{\max} + R_{\min} \cdot \cos(a)) \cdot \sin(b) \\ R_{\min} \cdot \sin(a) \end{bmatrix}$$

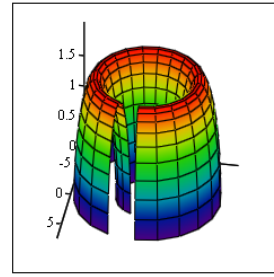
- в полі тримірного графіка ввести функцію **CreateMesh(h, 0, π , 0, $2\pi - \pi/8$, 20, 20)**. Отримане зображення наведено на рис. 3.18б.

Для обох способів графік функції має однаковий вигляд.



(X, Y, Z)

а



CreateMesh(h, 0, pi, 0, 2*pi - pi/8, 20, 20)

б

Рис.3.18 Зображення торичної поверхні

а) параметричне б) функцією **CreateMesh**

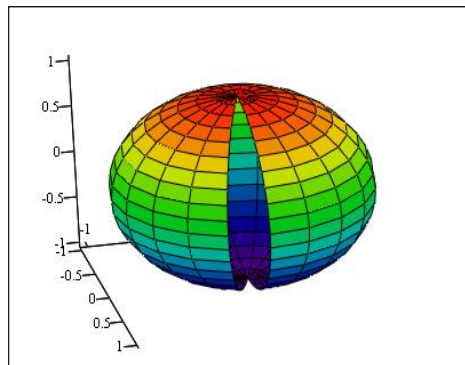
Завдання 3.7.1. Побудувати двомірний, тримірний та контурний графіки, що ілюструють «кільця Ньютона» згідно наступного виразу:

$$f(r) = I_0 \cdot \cos^2\left(\frac{2 \cdot \pi}{\lambda} \cdot \delta(r)\right), \text{ де}$$

$I_0=1, \lambda=0.5 \cdot 10^{-3}, R=5000, 0 < r < 4$, крок 0.02.

$$\delta(r) = \sqrt{R^2 + r^2} - R.$$

Завдання 3.7.2. Намалювати зображення сфери із секторним вирізом (рис. 3.19) параметрично та функцією *CreateMesh*.



(X, Y, Z)

Рис. 3.19. Сфера завдання 3.7.2

Завдання 3.7.3. На інтервалі -1 . . . 1 зобразити каркасно, з кольоровим фарбуванням, лініями рівня поверхню

$$f(x, y) = (\sin(x^2) + \cos(y^2))^{xy}.$$

Завдання 3.7.4 Зобразити поверхню, яка визначена параметрично
 $x(u,v)=\cos(u)\cdot\sin(v)$ $y(u,v)=\sin(u)\cdot\sin(v)$ $z(u,v)=u\cdot v$ $u,v\in[0,3]$.

Завдання 3.7.5. Намалювати гвинтову лінію $x(t)=t\cdot\cos(t)$ $y(t)=t\cdot\sin(t)$ $z=t$.

Завдання 3.7.6. Намалювати векторне поле

$$x(x,y)=\cos(3\pi x) \quad y(x,y)=\sin(3\pi y).$$

Завдання 3.7.7. Намалювати циліндричну поверхню, яка визначена в циліндричних координатах

$$R(a,b)=2 \quad \varphi(a,b)=a \quad q(a,b)=b.$$

3.8. Розрахунки з розмірностями

Приклад 3.8.1. Визначити за законом Ньютона силу тіла масою 1кг, яке має прискорення 2m/s^2 .

РОЗВ'ЯЗАННЯ

1. Визначити змінну для маси тіла, після чисельного значення ввести позначку множення $\text{mas} := 1 \cdot \blacksquare$, на місце маркеру ввести розмірність маси – kg.

2. Ввести змінну прискорення a, після чисельного значення поставити позначку множення $\text{mas} := 1 \cdot \text{kg} \quad | \quad a := 2 \cdot \blacksquare$ на місце маркеру ввести розмірність прискорення – m/s^2 .

3. Записати формулу закону Ньютона та поставити знак виведення результату «= \Rightarrow ». Отримуємо результат обчислення. Розмірність виведена в фундаментальних одиницях $\text{m}\cdot\text{kg}\cdot\text{s}^{-2}$

$$\text{mas} := 1 \cdot \text{kg} \quad a := 2 \cdot \frac{\text{m}}{\text{s}^2}$$

$$\boxed{\text{mas} \cdot a = 2 \cdot \text{m} \cdot \text{kg} \cdot \text{s}^{-2} \blacksquare}$$

4. На місці маркеру ввести «newton» $\text{mas} \cdot a = 2 \text{ newton}$

$$\boxed{\text{mas} \cdot a = 2 \cdot \text{N} \blacksquare}$$

Приклад 3.8.2. Визначити щільність та питому вагу сталі, якщо 2см^3 її мають масу 15.7 г.

РОЗВ'ЯЗАННЯ

Щільність визначається діленням об'єму речовини на масу. Питома вага – множенням прискоренням вільного падіння на щільність.

1. Ввести вихідні значення для об'єму в кубічних сантиметрах, маси в грамах

$$V := 2 \cdot \text{cm}^3 \quad \text{mas} := 15.7 \cdot \text{gm}$$

2. Розраховувати щільність та питому вагу

$$\rho := \frac{\text{mas}}{V} = 7.85 \times 10^3 \frac{\text{kg}}{\text{m}^3}$$

$$\text{uv} := \rho \cdot g = 7.698 \times 10^4 \frac{\text{kg}}{\text{m}^2 \cdot \text{s}^2}$$

Зверніть увагу, на незвичне позначення грамів та на той факт, що стала вільного падіння є вбудованою в систему.

Приклад 3.8.3. Визначити енергію вантажу за законом Гука $E = k \cdot \frac{l_0^2}{2}$, що підвішено на пружині при початковому зсуві $l_0 = 8\text{см}$, якщо відомо, що сила $F = 2\text{кГс}$ викликає подовження пружини на $L = 10\text{мм}$. Коефіцієнт Гука $k = \frac{F}{L}$.

РОЗВ'ЯЗАННЯ

$$F_0 := 2 \cdot \text{kgf} \quad L_0 := 8 \cdot \text{cm} \quad \text{Len} := 10 \cdot \text{cm} \quad - \quad \text{визначення}$$

вихідних даних сили в одиницях «кілограм сили», подовження в одиницях «сантиметр», початкового зсуву в одиницях «сантиметр». Для змінної сили, подовження застосовано імена F_0 , Len , бо в системі назва F вже використано для одиниці виміру «фарада», L - одиниця виміру «літр».

$$k := \frac{F_0}{L_{en}} = 196.133 \frac{\text{kg}}{\text{s}^2} \quad \text{- розрахунок коефіцієнта Гука.}$$

$$E := k \cdot \frac{L_0^2}{2} = 0.628 \text{m}^2 \cdot \text{kg} \cdot \text{s}^{-2} \quad E = 0.628 \text{J} \quad \text{- обчислення енергії.}$$

Перший результат виведено в базових одиницях. Другий - в «джоулях» за рахунок увімкнення режиму спрощення розмірностей.

Приклад 3.8.4. Визначити променевий потік від абсолютно чорного тіла у вигляді диску діаметром 1см в діапазоні λ (1-3)мкм для температури $T=300\text{K}$ інтегруванням формули Планка.

РОЗВ'ЯЗАННЯ

Формула Планка визначає спектральну щільність світності для визначеної температури випромінювача

$$M(\lambda, T) = \frac{C_1}{\lambda^5 e^{\left(\frac{C_2}{\lambda T} - 1\right)}} \left[\frac{W}{\text{cm}^2 \text{mkm}} \right], \text{ де}$$

$C_1 := 37400 \text{ Вт} \cdot \text{мкм}^4 \cdot \text{см}^{-2}$, $C_2 := 14400 \text{ мкм} \cdot \text{К}$ - сталі формули Планка, λ - довжина хвилі випромінювання, T - температура.

Потік випромінювання визначається як світність, помножена на площу випромінювача:

$$F = M \cdot A, \text{ де } A = \frac{\pi d^2}{4} \text{ - площа випромінювача.}$$

1. Визначити вихідні дані: діаметр d , довжини хвилі початку λ_1 та кінця діапазону λ_2 , температуру T

$$d := 1 \cdot \text{cm} \quad \lambda_1 := 1 \text{micron} \quad \lambda_2 := 2 \cdot \text{micron} \quad T := 300 \cdot \text{K}$$

2. Визначити сталі формули Планка

$$C_1 := 37400 \cdot \text{W} \cdot \text{cm}^{-2} \cdot \text{micron}^4 \quad C_2 := 14400 \cdot \text{micron} \cdot \text{K}$$

3. Визначити розрахункову формулу Планка як функцію користувача

$$M(\alpha, \beta, t) := \int_{\alpha}^{\beta} \frac{C1}{\lambda^5 \cdot \exp\left(\frac{C2}{\lambda \cdot T} - 1\right)} d\lambda$$

4. Розрахувати площу випромінювача

$$A := \frac{\pi \cdot d^2}{4} = 7.854 \times 10^{-5} \text{ m}^2$$

5. Розрахувати променевий потік

$$\text{Flux} := M(\lambda_1, \lambda_2, T) \cdot A = 8.916 \times 10^{-9} \text{ W}$$

Відповідь отримано в одиницях системи СІ. Система сама провела масштабування значень, які були визначені в різних одиницях, до однієї (мкм, см, м). Підкреслена зеленим змінна А показує, що відбулося перевизначення вже існуючих даних. В даному випадку – одиниці вимірювання – «Ампер».

Приклад 3.8.5. Розрахувати кількість квартир та будинків в мікрорайоні з населенням 1368 людей. В квартирі мешкає 2 особи, в будинку 60 квартир.

РОЗВ'ЯЗАННЯ

1. Визначити базову позасистемну розмірність «людина» та похідні від неї розмірності «квартира», «дім»

$$\text{людина} := 1 \quad \text{квартира} := 2\text{-лк} \quad \text{дім} := 60\text{-квартира}$$

2. Ввести змінну x з кількістю людей. В маркер вписати базову розмірність, Вивести значення змінної. На місці маркера вписати розмірності «квартира», «дім»

$$x := 1320\text{-людина} \quad x = 660\text{квартира} \quad x = 11\text{дім}$$

Завдання 3.8.1. Обрахувати щільність, питому вагу за умовами прикладу 3.8.2 в одиницях «Г», «см», «м», «сек», «кг».

Завдання 3.8.2. Обрахувати момент інерції

а) стрижня масою $M=5$ kg, довжиною $L=2$ m, який обертається відносно осі, що проходить через його кінець:

$$I = \int_0^L x^2 \cdot \frac{M}{L} dx$$

б) прямокутної пластини масою $M = 5$ kg, довжиною $L = 2$ m, шириною $h=10$ cm, відносно вісі, яка проходить через центр пластини перпендикулярно до неї:

$$I = \frac{M(h^2 + L^2)}{12}.$$

Завдання 3.8.3 Розрахувати кількість теплоти Q в кілоджоулях, яке виділиться при нагріванні праски масою $M=1$ kg на $dT=200^\circ\text{C}$ від 20°C до 220°C , якщо питома теплоємність матеріалу праски $C=0.54$ кДж/кг·К за виразом $Q=C \cdot (T_2-T_1) \cdot M=C \cdot dT \cdot M$.

3.9. Символьні обчислення

Завдання 3.9.1. Спростити вираз

$$\text{а) } \frac{3x^2 - 2x}{6 - 7x - 3x^2} \quad \text{б) } \frac{a - \frac{4a-4}{a}}{\frac{2}{a} - 1} \quad \text{в) } \frac{x-3}{4x^2 + 24x + 36} \cdot \left(\frac{x}{3x-9} - \frac{3}{9+3x} + \frac{x^2+9}{27-3x^2} \right).$$

Завдання 3.9.2. Розкласти вираз на множники

$$\text{а) } x^4 - 10x^3 + 35x^2 - 50x + 24 \quad \text{б) } a^2b + b^2a + 2abc + b^2c + a^2c + ac^2 + bc^2.$$

Завдання 3.9.3. Розгорнути вирази

$$\text{а) } \cos(5x) \quad \text{б) } (2x^3 + 3x^2 + 4x + 1)(x-1)(x-2)(x-3).$$

Завдання 3.9.4. Розкласти в ряд Тейлора функцію до 5-го ступеню

$$\text{а) } \sin^2(x) \quad \text{б) } \text{tg}(x).$$

Завдання 3.9.5. Визначити, чи є числа 123456789, 2127 -1 простими.

Завдання 3.9.6. Розрахувати інтеграл, похідну

$$\text{а) } \int x \cdot \operatorname{arctg}(x) dx \quad \text{б) } \int_a^b \frac{x^2}{x^6 + 4} dx \quad \text{в) } \int \frac{dx}{\sqrt{4x - 3 + x^2}}$$

$$\text{г) } \frac{d}{dx} \frac{e^x}{1+x^2} \quad \text{д) } \frac{d}{dx} \frac{\sqrt[9]{4x^5+2}}{3x^4} .$$

Завдання 3.9.7. Визначити сходимість ряду. Ряд сходиться, якщо $|a_n| < 1$ за $n \rightarrow \infty$. В разі сходимості, розрахувати суму ряду.

$$\text{а) } \sum_{n=1}^{\infty} \frac{2^{n+1}(3n+2)}{n!} \quad \text{б) } \sum_{n=1}^{\infty} \frac{n!}{n^2 2^n} \quad \text{в) } \sum_{n=1}^{\infty} \frac{n^3}{e^n} \quad \text{г) } \sum_{n=1}^{\infty} \frac{n}{(2n+1)!} .$$

Завдання 3.9.8. Розкласти функцію e^x в ряд Тейлора до 3-го ступеню та обрахувати значення ряду для $x=1$.

4. СКМ MATLAB

4.1. Загальні характеристики

Система *Matlab* (Matrix Laboratory) була розроблена С. Молером (C. Moler) і з кінця 70-х років широко використовувалась на великих ЕОМ. З'явився пакет *Matlab* як засіб доступу до бібліотек програм LINPACK і EISPACK, призначених для матричних обчислень. На початку 80-х років Джон Літл (John Little) з фірми *MathWorks Inc* розробив версії системи PC *Matlab* для комп'ютерів класу IBM PC, VAX і Macintosh. Далі були розроблені версії для робочих станцій Sun, комп'ютерів з операційною системою UNIX та інших типів великих і малих ЕОМ.

Система орієнтована на роботу з матрицями. Матриці широко застосовуються в складних математичних розрахунках, наприклад, при розв'язанні задач лінійної алгебри та математичного моделювання статичних та динамічних об'єктів. Вони є підґрунтям складання та розв'язання рівнянь стану динамічних систем.

Можливості системи *Matlab*:

- символічне та чисельне розв'язування різноманітних задач математичного аналізу (обчислення границь, диференціювання й інтегрування функцій, перетворення Фур'є тощо);
- реалізація алгоритмів обчислювальної математики (сплайн-інтерполяція, метод Рунге-Кутта для розв'язування звичайних диференціальних рівнянь тощо);
- розв'язування задач умовної і безумовної оптимізації (зокрема задач лінійного, дискретного та нелінійного програмування);
- розв'язування задач лінійної алгебри (додавання, множення, обчислення оберненої, транспонованої матриць, обчислення визначників і мінорів, множення матриці на вектор, пошук власних значень і векторів, розв'язування матричних рівнянь тощо);
- опрацювання статистичних даних;
- імпортування й експортування даних у формати ASCII, Binary, Excel, Word, PowerPoint;
- імпортування й експортування графіки форматів PS, EPS, Bitmap, EMF, GIF, AVI, JPG, TIFF;

- вбудована мова програмування та діалогу з користувачем.

Система відрізняється наявністю великої кількості розширень і доповнень, які дозволяють використовувати її не тільки як розрахунковий математичний пакет, а й застосовувати для проведення вимірювань, обробки результатів, розв'язання спеціалізованих інженерних задач моделювання. Спектр завдань, дослідження яких може бути здійснено за допомогою *Matlab* і його розширень «*Toolbox*», охоплює: матричний аналіз, обробку сигналів та зображень, задачі математичної фізики, оптимізаційні задачі, фінансові завдання, роботу з картографічними зображеннями, нейронні мережі, нечітку логіку і багато іншого.

Близька сорока спеціалізованих *Toolbox* можуть бути вибірково встановлені разом з *Matlab*, наприклад: бібліотека методів чисельної оптимізації *Optimization Toolbox*, бібліотека засобів символьних перетворень *Symbolic Math Toolbox*, бібліотеки засобів для підключення плат введення/виведення сигналів *Data Acquisition Toolbox*, *Image Acquisition Toolbox*, *Instrument Control Toolbox*, бібліотеки фінансово - економічних розрахунків *Financial Derivatives Toolbox*, *Financial Time Series Toolbox*, бібліотеки засобів обробки зображень *Image Processing Toolbox*, *Wavelet Toolbox*, бібліотеки розробки та аналізу високочастотних ланцюгів *Neural Network Toolbox*, *Filter Design Toolbox*, *Signal Processing Toolbox*, бібліотека побудови моделей на основі експериментальних даних *RF Toolbox*, бібліотеки статистичної, регресійної обробки даних *Spline Toolbox*, *Statistics Toolbox*, *Curve Fitting Toolbox*.

Наявність опціонального модуля *Simulink* дозволяє проводити візуальне моделювання частотних і часових характеристик на системному рівні. Спеціалізовані розширення модуля *Simulink* доповнюють вбудовані засоби, наприклад:

SimPowerSystems (раніше *Fixed-Point Blockset*) - призначено для моделювання електроенергетичних систем;

SimMechanics – призначено для моделювання механічних систем;

Real-Time Windows Target – призначено для проведення експериментів із введенням даних через пристрої введення/виведення;

Stateflow – призначено для моделювання кінцевих автоматів;

Aerospace Blockset – бібліотека елементів аерокосмічних систем;

CDMA Reference Blockset – бібліотека елементів бездротового зв'язку стандарту CDMA IS-95A;

DSP Blockset – бібліотека елементів систем цифрової обробки сигналів;

Communications Blockset – бібліотека елементів систем зв'язку тощо.

Matlab має добре розвинені можливості візуалізації двомірних і тримірних даних. Повний доступ до зміни властивостей графіків дають низькорівневі функції, застосування яких вимагає розуміння принципів комп'ютерної графіки та володіння прийомами об'єктного програмування.

Візуальне середовище *GUIDE* призначене для програмного створення графічного інтерфейсу користувача. Наявність певної навички роботи в середовищі *GUIDE* надає можливість створити візуальну оболонку для проведення власних "віртуальних" експериментів.

Matlab інтегрується з багатьма додатками і середовищами програмування. Зв'язок *Matlab* і *MS Word* за допомогою модуля *Notebook* забезпечує можливість написання в редакторі *MS Word* інтерактивних документів – "М-книг". Надбудова *MS Excel Link* забезпечує доступ користувача до функцій *Matlab* з *MS Excel*. Інформація, що зберігається в базах даних багатьох форматів, може бути імпортована в *Matlab*. Для обміну даними використовуються команди мови запитів SQL. Підтримується, зокрема, зв'язок з *Microsoft Access*, *Microsoft SQL Server*, *Oracle*.

Програмний інтерфейс програми (API) реалізує зв'язок середовища *Matlab* з програмами, що написані на мовах C, Fortran або Java.

Символьні обчислення в *Matlab* базуються на бібліотеці, яка є ядром пакета *MuPad*.

На відміну від *Mathcad*, в *Matlab* реалізовано командний діалог за допомогою вбудованої мови програмування.

Оболонка пакета

Після завантаження пакета *Matlab* на екрані виводиться діалогове середовище (рис. 4.1). Вікно є стандартним для *Windows* застосунків. Нагорі

розташовані меню та контекстні панелі керування, знизу розташований статусний рядок. Центральну область займають робочі вікна системи: командне вікно, вікно історії дій, вікно стану робочого простору, вікно поточних файлів, вікно редактора, яке відкривається при редагуванні М-файлів. При виведенні графіків, зображень, запуску програм з графічним інтерфейсом вони також виводяться в додаткові власні вікна в центральну область.

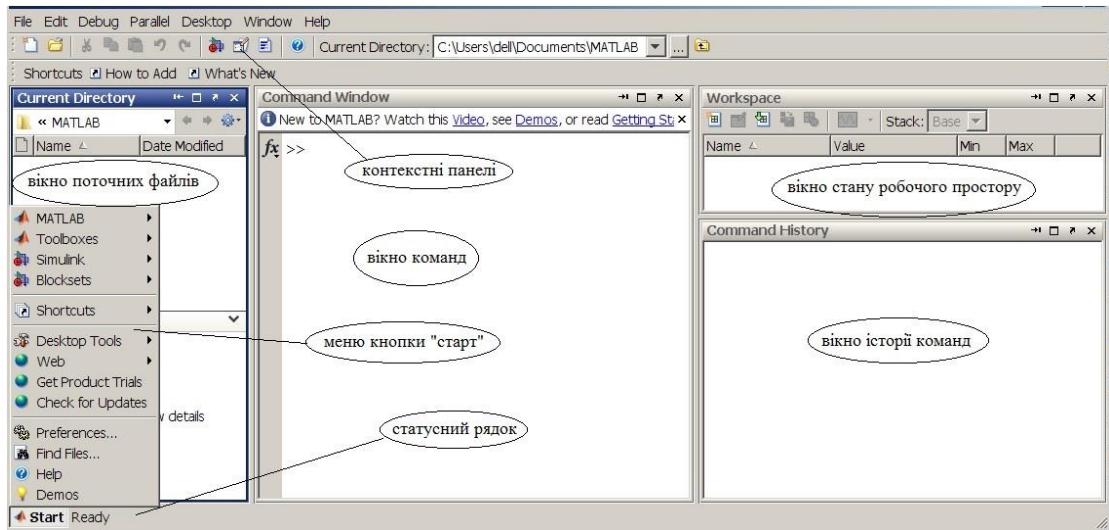


Рис. 4.1. Стандартний вигляд діалогового середовища *Matlab*

В статусному рядку відображається стан системи та закладки програм користувача, які в даний час завантажені в пакет. Кнопка **Start** статусного рядка подібна до однойменної кнопки *Windows*. Вона забезпечує через пункти випадаючого меню швидкий доступ до функцій пакетів розширення системи, вікон налаштування, допомоги тощо.

Command window – командне вікно. Основне вікно для введення команд і відображення результатів розрахунку. В ньому користувачем вводяться з клавіатури команди, відображаються обраховані результати та інформація про помилки.

Command History – вікно історії команд. Відображає по рядках всі уведені команди за час сеансу роботи з пакетом. Клацання на обраному рядку призводить до виконання команди в командному вікні.

Current directory – вікно, яке відображає дерево робочих файлів сеансу.

Workspace (робочий простір) – вікно, яке містить інформацію про склад та зміст даних під час роботи сеансу пакета. Всі данні системи за час сеансу розташовуються в області пам'яті, яку називають робочим простором.

Цим вікном слушно користуватися для контролю значень змінних та інших даних. В вікні по рядках відображаються імена, значення, тип даних та обсяг, який вони займають (рис. 4.2). Наприклад, в сеансі, який зображений на рис.4.2. введено три змінні. Змінна «x» зі значенням 5 та два вектори: «a» – зі значеннями 1, 2, 3 та «b» – 2, 3, 4. Подвійний щиглик на якомусь імені відкриває вікно редактора даних *Array Editor*. В редакторі можна змінювати значення. Крім того, на панелі вікна знаходяться кнопки для збереження даних у файл, зміни їхнього вигляду та команди малювання деяких двовірних графіків.

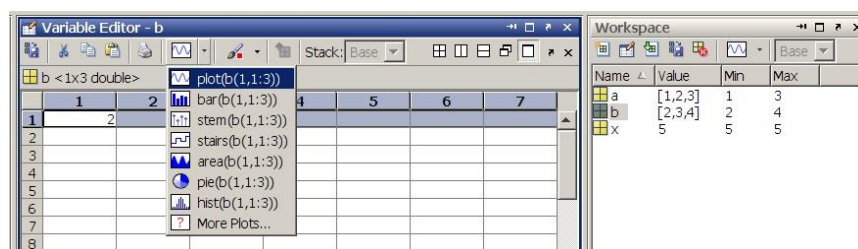


Рис. 4.2. Вікно робочого простору

Вигляд діалогового середовища задається пунктом **Desktop** головного меню. Можливі наступні варіанти: **Default** (*Command window, Command History, Current directory/Workspace*), **All Tabbed** (всі 6 вікон), **Command window only**, **Command window and Command History**, **All but Command minimized**.

Вигляд та розташування вікон можуть змінюватися в залежності від потреб користувача кнопками у верхній частині вікна (рис. 4.3). Кожне з вікон можна «вийняти» з середовища та зробити вікно незалежним кнопкою 1, розкрити на весь екран кнопкою 2, закрити кнопкою 3, згорнути до значка на краю екрана кнопкою 4. Клацання на значку знов відриває вікно. Зсув миші з натиснутою лівою кнопкою дозволяє зсунути вікно в інше місце.



Рис. 4.3. Засоби керування вікном

Команди керування робочим простором

- clear** - очищає робочий простір,
clear <x>, <y> - очищає обрані дані з пам'яті,
pack - дефрагментує робочий простір,
who - показує список даних робочого простору,
whos - показує список та зміст даних робочого простору.

Збереження даних робочого простору для його подальшого використання проводиться пунктом “**File – Save Workspace as...**” головного меню. Збереження проводиться в файлах *.MAT.

Завантаження даних з дискового файлу проводиться командою

load <ім'я файлу даних > <ім'я 1>,<ім'я 2>, ...

Команди керування командним вікном

- clc** - очищає екран та розташовує курсор в верхньому рядку;
home - розташовує курсор в верхньому лівому куту вікна;
echo <ім'я файлу ><on/off> - керує «ехо» відображенням команд з дискового *.m script файлу;
more<on/off> - керує посторінковим відображенням рядків файлів;
... - ставиться в кінці рядка, коли потрібно продовжити запис;
% - рядок або частина рядка після символу вважається текстовим коментарем;
↑ (↓) - прокручування рядків виконаних команд. Виконання команди проводиться натисканням клавіші уведення.

Примітка. Витерти зміст вікон можна пунктом «Clear» відповідного контекстного меню відповідного вікна або підменю «Edit» головного меню. Контекстні вікна відкриваються клацанням правою кнопкою миші.

Matlab має два режими роботи: *режим калькулятора* та *програмний режим*. Зовні режими нічим не відрізняються та можуть використовуватися в будь якій послідовності. *Режим калькулятора* передбачає введення в рядку командного вікна виразу для обчислення, команди системи та безпосередне

їхнє виконання відразу після натискання клавіші «*Enter*». *Програмний режим* передбачає створення скрипт-програм або підпрограм користувача в М-файлі з наступним її запуском для виконання. Більшість дій, які виконуються в режимі калькулятора, можливо виконати й в програмному режимі. Для цього потрібно перенести рядки коду в редактор програм та зберегти їх в М-файлі.

4.1.1. Контрольні питання

На обробку яких даних орієнтовано Matlab?

Перерахуйте основні можливості Matlab.

Що таке Toolbox? Для яких задач вони застосовуються?

Для чого слугує модуль Simulink?

Як в Matlab реалізовано символні перетворення?

Чим відрізняються реалізації діалогу в *Mathcad* та *Matlab*?

Назвіть складові діалогового середовища *Matlab*.

Що таке *Workspace*? Які можливості воно забезпечує?

Чим відрізняються два основні режими роботи *Matlab*?

4.2. Робота в режимі калькулятора

4.2.1. Змінні, вирази, вбудовані функції

Ознакою того, що *Matlab* готовий до прийняття команди є поява на початку останнього рядка командного вікна знака «*»*». Для виконання дій достатньо ввести вираз з числами або змінними та натиснути клавішу «**Enter**». В наступному рядку з'явиться відповідь.

Алгоритми обробки даних складаються з математичних виразів, команд керування, вбудованих та зовнішніх функцій. Вирази можуть вміщувати числа, імена змінних, математичних функцій, операторів дій.

Введення чисел з клавіатури проводиться за загальними правилами мов програмування високого рівня:

- для відділення дробової частини мантиси числа застосовується десяткова точка;

- для чисел в форматі з плаваючою комою десятковий показник задається літерою "e", показник ступеня числа записується як ціле число після символу "e". Між записом мантиси числа та символом "e", який відділяє мантису від показника, не повинно бути символів. Наприклад, $12,34 \cdot 10^{56} = 12.34e56$.

Введення інформації в командне вікно підпорядковується таким правилам:

- результат дії відразу виводиться в командне вікно. Якщо оператор закінчується символом «;», то результат його дії не відображається;
- якщо оператор не містить знака "=", тобто є просто записом послідовності дій над числами та змінними, значення результату присвоюється спеціальній системній змінній **ans**. Значення системної змінної **ans** змінюється після дії кожного оператора без знака "=" . Отримане значення змінної **ans** можна використовувати в наступних операторах;
- можна записувати кілька операторів в одному рядку з розділенням комою чи комою з крапкою. Виконання проводиться зліва - направо;
- для продовження запису в кількох рядках наприкінці рядка ставляться три крапки.

Для більш наочного представлення дій та програмування в системі застосовуються змінні. *Матлаб* не вимагає попереднього опису типів змінних.

Всі дані зберігаються у вигляді масивів (рис. 4.4): чисельних змінних (numeric), текстових рядків (char), комірок (cell), структур (struct). За замовчанням чисельні масиви мають тип *double*, розріджені матриці - *sparse*.

Зазвичай перетворення типів даних не потрібне. Перетворення типів даних потребується у випадках необхідності зменшення обсягу пам'яті, обробки спеціальних даних.

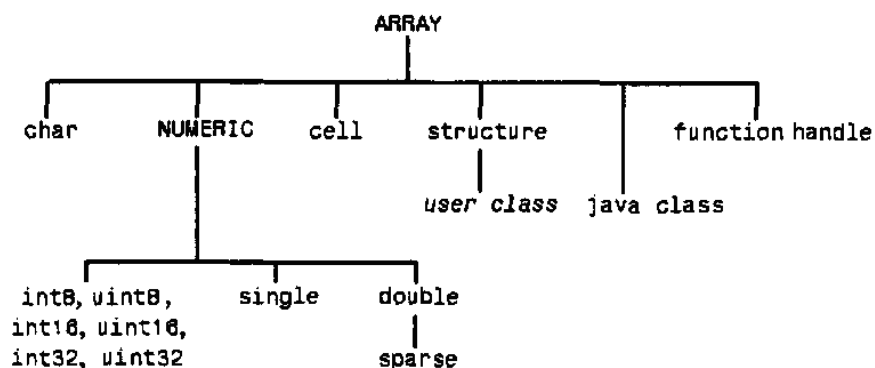


Рис. 4.4. Типи даних Matlab

Таблиця 4.1. Типи чисельних даних Matlab

Тип	Опис	Діапазон
int8/uint8	Однобайтове ціле з/без знака	-128... 127 / 0 ... 255
int16/uint16	Двобайтове ціле з/без знака	-32768...32767/0 ... 65535
int32/uint32	Чотирьохбайтове ціле з/без знака	-2147483648...2 47483647 0 ... 4 294 967 295
int64/uint64	Восьмибайтове ціле з/без знака	$-2^{63} \dots 2^{63} / 0 \dots 2^{64}$
single	Чотирьохбайтове дійсне	$1.175 \times 10^{-38} \dots 3.4 \times 10^{+38}$
double	Восьмибайтове дійсне	$2 \times 10^{-308} \dots 1.8 \times 10^{+308}$

За замовчанням всі чисельні дані обробляються як дійсні числа у вигляді восьмибайтового числа типу *double* у мові C.

Присвоєння має наступний вигляд:

<ім'я змінної > = <число, рядок або вираз >.

Ім'я змінної може мати до 30 символів та не повинно співпадати з іменами функцій, процедур системи та системних змінних. Система розрізняє великі та малі літери в змінних. Текстові рядки повинні бути забрані в апострофи.

Система *Матлаб* має вбудовані змінні, значення яких зарезервовано:

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

pi - число π (3.141592653589793);

inf - машинна нескінченність;

- NaN** - невизначений результат (0/0 чи inf/inf);
- eps** - похибка операцій над числами;
- ans** - результат останньої операції без знака "=";
- realmax/ intmax** - максимальна величина дійсного/цілого числа;
- realmin/ intmin** - мінімальна величина дійсного/цілого числа.

З чисел, змінних, функцій в командному рядку набирається обчислювальний вираз. Введення закінчується натисканням клавіші «**Enter**». Обчислення відбувається відразу після закінчення введення.

Наприклад, обчислення наступного виразу

$$0.7854 - \frac{0.7854^3}{1 \cdot 2 \cdot 3} + \frac{0.7854^5}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5} - \frac{0.7854^7}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7}$$

в командному вікні може мати наступний вигляд

```
>> 0.7854 - (0.7854)^3 / (1*2*3) + 0.7854^5 / (1*2*3*4*5) ...
- (0.7854)^7 / (1*2*3*4*5*6*7) (Enter)
ans = 0.7071
```

Три крапки в кінці рядка дозволяють перенести запис виразу на наступний рядок. Вираз немає знаку рівняння, тому система застосувала системну змінну **ans** та надала їй обчислене значення **0.7071**.

Пакет містить велику бібліотеку вбудованих функцій:

- оператори: +, -, *, \, /, >, >=, <, <=, =, ==,
- елементарні функції: **sqrt**, **logs**, **log10**, **exp**, **pows** тощо,
- тригонометричні прямі та зворотні функції,
- гіперболічні функції,
- спеціальні функції Беселя, Лагранжа тощо,
- функцій обробки рядків,
- функції роботи з часом,

- функції роботи з матрицями та векторами: `sum`, `prod`, `length`, `size`, `sort`, `ndims`, `max`, `min`, `mean`, `'`, `.`, `rot90`, `dot`, `fliplr`, `cross`, `inv`, `eig`, `det`,
- функції обробки даних та спектральних перетворень,
- функції обробки поліномів,
- функції двомірної та тримірної високорівневої графіки,
- функції низькорівневої дескрипторної графіки.

Виклик вбудованої або функції користувача проводиться наступним чином:

```
<ім'я (імена) результату>=  
<ім'я функції>(<перелік фактичних значень аргументів >)
```

Функції *Matlab* мають особливість. Вони можуть мати змінну кількість аргументів та результатів. Це значно розширює можливості обчислень. Наприклад, функція `x=find(A)` поверне вектор «x» порядкових номерів елементів матриці «A», значення яких не дорівнює нулю. Виклик `x=find(A,2)` поверне номери перших двох елементів. Виклик `[s d]=find(A)` поверне вектор, перший елемент «s» якого є вектором з номерами рядків елементів, другий «d» - вектор з номерами стовпців.

Аргументи більшості тригонометричних функцій задаються в радіанах. Функції, що приймають аргументи в градусах містять в кінці імені символ «d».

Наприклад, функція `sin(x)` потребує аргументу в радіанах, функція `asin(x)` повертає результат в радіанах, а функції `sind(x)`, `asind(x)` працюють з градусами.

Наприклад, обчислення виразу

$$e^{-2.5}(\ln(11.3))^{0.3} + \left(\frac{\sin(2.45\pi) + \cos(3.78\pi)}{\tan(3.3)}\right)^2 \text{ має наступний вигляд:}$$

```
>>exp(-2.5)*log(11.3)^0.3+((sin(2.45*pi)...  
+cos(3.78*pi))/tan(3.3))^2  
ans=121.2446
```


Арифметичні та логічні операції виконуються відповідно до пріоритету:

1. Дії в дужках;
2. Функції та логічні дії «так», «або», «ні», «виключаюче або» (and, or, not, xor) тому, що вони є функціями;
3. Логічне заперечення;
4. Транспонування, піднесення до ступеню;
5. Множення, ділення;
6. Складання, віднімання;
7. Логічні відношення: >, >=, <, <=, ==;
8. Логічне "і": «&»;
9. Логічне "або": «|»;
10. Логічне "виключаючи або": «|».

Дії одного пріоритету виконуються зліва – направо.

Наприклад, два вирази: **and (A, B) + C** та **A&B+C** не еквіваленти.

Matlab містить в собі вбудовану арифметику комплексних чисел. Більшість елементарних математичних функцій побудовані таким чином, що аргументи допускають комплексні числа, а результати також формуються як комплексні числа. Для позначення уявної одиниці в *Matlab* зарезервовано два імені "i" та "j".

Введення з клавіатури значення комплексного числа проводиться шляхом запису в командне вікно рядка:

<ім'я комплексної змінної>=<значення ДЧ>+i*<значення УЧ>,

де ДЧ - дійсна частина комплексного числа, УЧ - уявна частина.

Функції обробки комплексних чисел:

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

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

angle (Z) – обчислює значення аргументу φ комплексного числа Z (в радіанах від $-\pi$ до $+\pi$)

$$\varphi = \arctg\left(\frac{\text{imag}(Z)}{\text{real}(Z)}\right);$$

conj (Z) – повертає число, комплексно-спряжене відносно Z;

complex (A,) - повертає комплексне число з дійсною частиною A та уявною частиною B. В разі відсутності аргументу «B» повертається комплексне число з нульовою уявною частиною.

$$Z = A + jB = |Z|e^{j\varphi} = \sqrt{\text{imag}(Z)^2 + \text{real}(Z)^2} \cdot \exp(j \cdot \arctg(\frac{\text{imag}(Z)}{\text{real}(Z)})) = |Z|\exp(j \cdot \text{angle}(Z))$$

Логічні операції

Результатом логічної операції може бути "істина" чи "брехня". "Істині" в *Matlab* відповідає значення 1, "брехні" - 0.

В системі реалізовані *інтегральна та бітова логіки*. Інтегральні дії застосовуються до значення аргументу. Результатом можуть бути 1 або 0.

Наприклад,

```
» x > 2 ans = 1
```

```
» 2 & 3 ans = 1
```

Бітові операції проводяться над кожним бітом цілих аргументів. Результатом є ціле число.

Наприклад,

```
» a = 3; c = 1;
```

```
» bitand(a,c)
```

```
ans = 2
```

В *Matlab*, як і в *Mathcad*, використовуються логічні відношення (>, <, == і т.д.), логічні операції (or, and і т.д.). Оператори відношення, логічні та арифметичні операції можна поєднувати в одному виразі .

Наприклад, вираз **a + (x > c)** не є помилкою:

```
» a = 1; x = 5; c = 3; a = a + (x > c)
```

```
a = 2
```

Умова $x > c$ виконується, результатом є 1. Результат операції додається до значення a.

Результат виконання логічного виразу може бути збережено в змінній:

```
» f = a <= x
```

Основні вбудовані функції пакета наведено в додатку А.

Для отримання інформації про можливості та синтаксис вбудованих функцій призначені функції **help**, **lookfor**.

Функція **help fun** виводить в командне вікно опис синтаксису функції **fun**.

Список операторів можна вивести командою **help** з аргументом **ops**, список елементарних функцій – **elfun**, спеціальних - **specfun**, функцій обробки рядків - **strfun**, введення/виведення - **iofun**, роботи з часом - **timefun**, роботи з матрицями - **matfun**, обробки даних та спектральних перетворень - **datafun**, обробки поліномів - **polyfun**, двомірної графіки - **graph2d**, тримірної графіки - **graph3d**, дескрипторної графіки - **graphics**, графічного редактору - **uitools**.

Функція **lookfor word** проводить пошук слова **word** та виводить список функцій, в яких це слово зустрічається.

4.2.2. Функції користувача

В разі необхідності проведення в режимі калькулятора обчислень, що мають частини, які повторюються, можна застосувати вбудовану (**Inline**) та анонімну функції користувача. Функції користувача визначаються в командному вікні та викликаються як і всі інші функції *Matlab*. В разі потреби обчислити вираз, який визначено текстовим рядком, слушно застосувати функцію **eval ('вираз-рядок')**.

Наприклад, рядки

```
» eval('sin(2)');  
» x=2; eval('sin(x)');
```

дають однакову відповідь - розрахунку функції синуса з аргументом 2

```
ans =      0.9093
```

Inline-функція

Вбудована функція користувача визначається словом «**inline**»:

```
ім'я_функції=inline('формула', <'arg1', ..., argM'>),
```

де

'формула' є текстовим рядком з виразом для обчислення. Рядок з переліком аргументів використовується, якщо потрібно змінити порядок аргументів при виклику функції.

Наприклад, для обчислення наступного виразу

$$z = \frac{(\sin(y) + \sin(2y))^3}{1 + \frac{\sin(y) + \sin(2y)}{1 + e^x}} + \sqrt{1 + \frac{\sin(y) + \sin(2y)}{1 + e^x}}$$

потрібно декілька разів визначати значення $\sin(y) + \sin(2y)$ та $1 + e^x$. Можна визначити ці частини у вигляді функцій користувача

```
>>myexp=inline('1+exp(x)')
myexp= Inline function: myexp(x) = 1+exp(x)
>>mysin=inline('sin(y)+sin(2*y)')
mysin=Inline function: mysin(y) = sin(y)+sin(2*y)
>>mysin(2)^3/(1+mysin(2)/myexp(1)). . .
+sqrt(1+mysin(2)/myexp(1))
ans = 1.0237
>> y=2; x=1;
>>mysin(y)^3/(1+mysin(y)/myexp(x)). . .
+sqrt(1+mysin(y)/myexp(x))
ans = 1.0237
```

Якщо функція має кілька аргументів, то вони розташовуються в алфавітному порядку:

```
>> fun1 = inline ('sin(a*x) - x^2*cos(b*x) ')
fun1=inline function:fun1(a,b,x)=sin(a*x)- x^2*cos(b*x)
```

Для зміни порядку аргументів їх слід вставити через кому після рядка-виразу:

```
» fun2=inline('sin(a*x)-x^2*cos(b*x)', 'x', 'a', 'b')
fun2=inline function:fun2(x,a,b)=sin(a*x)- x^2*cos(b*x)
```

Анонімна функція

Анонімна функція користувача визначається символом «@»:

Ім'я_функції = @(список_аргументів) формула.

На відміну від **inline**-функції, аргументи та формула записуються в звичайному вигляді, а не як текстовий рядок в лапках. За способом використання анонімна функція подібна до **inline**-функції з тією відмінною, що створюється покажчик на функцію.

```
>>myexp1=@(x) 1+exp(x)
Myexp1= @(x)1+exp(x)
>>mysin1=@(y) sin(y)+sin(2*y)
Mysin1=@(y) sin(y)+sin(2*y)
>>mysin1(2)^3/(1+mysin1(2)/myexp1(1)). . .
+sqrt(1+mysin1(2)/myexp1(1))
ans = 1.0237
>> y=2; x=1;
>>mysin1(y)^3/(1+mysin1(y)/myexp1(x)). . .
+sqrt(1+mysin1(y)/myexp1(x))
ans = 1.0237
```

4.2.3. Формат виведення результатів

Чисельний результат виводиться системою у вигляді, який визначається попередньо заданим форматом чисел. Цей формат може бути встановлено за допомогою вікна **Preferences** (рис. 4.4) або введенням команди формату явно в рядок командного вікна. Пункт **Command Window** у вікні містить список **Numeric Format**, який визначає формат представлення чисел, що виводяться в командне вікно в процесі розрахунків. Відкрити вікно **Preferences** можна з меню **File** або контекстного меню кнопки «Старт».

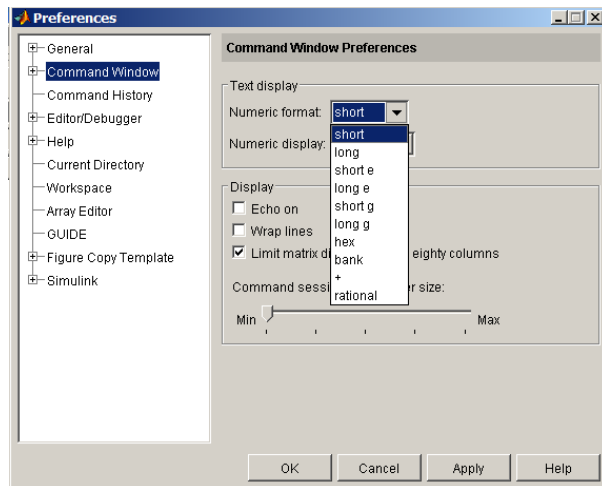


Рис. 4.4. Меню Preferences

Нижче наведено найбільш застосовувані формати:

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

Long - довгий запис;

Hex - запис у вигляді шістнадцятиричного числа;

Bank - запис з сотими частинами;

Plus - записується тільки знак числа;

Short E - короткий запис в форматі з плаваючою комою;

Long E - довгий запис в форматі з плаваючою комою;

Short G - форма короткого запису в форматі з плаваючою комою (5 значущих цифр);

Long G - форма довгого запису в форматі з плаваючою комою;

Rational - запис у вигляді раціонального дробу.

Наприклад, результат ділення $2/3$ в різних форматах має наступний вигляд:

0.6667	Short
0.666666666666667	Long
6.6667e-001	Short E
6.666666666666666e-001	Long E

0.66667	Short G
0.6666666666666667	Long G
3fe5555555555555	Hex
0.67	Bank
+	Plus
2/3	Rational

Для керування форматом результатів програмно та з командного рядка слугує команда

format <тип >, де

можливі значення <тип >:

- short** - запис з фіксованою комою 5 значущих цифр;
- long** - запис фіксованою комою 15 значущих цифр;
- hex** - запис шістнадцятиричного числа;
- bank** - запис з сотими;
- +** - запис тільки знаку числа;
- short e** - запис з плаваючою комою 5 значущих цифр;
- long e** - запис з плаваючою комою 15 значущих цифр;
- short g** - запис з плаваючою / фіксованою комою (5 значущих цифр);
- long g** - запис з плаваючою / фіксованою комою (15 значущих цифр);
- short eng** - запис з плаваючою комою (4 значущих цифри), ступінь кратний 3;
- long eng** - запис з плаваючою комою (16 значущих цифр), ступінь кратний 3;
- rational** - запис у вигляді раціонального дроби;
- compact** - видалення пробілів з запису;

loose - додавання пробілів в запис.

Примітка. Незалежно від формату виведення вихідні дані та результати всіх обчислень в системі *Matlab* зберігаються в пам'яті ПК у вигляді дійсних чисел об'ємом 8 байт з відносною похибкою $2 \cdot 10^{-16}$ з точними значеннями в 15 десяткових розрядах, діапазон представлення модуля дійсних чисел лягає в проміжку між 10^{-308} та 10^{+308} .

4.2.4. Масиви

Matlab є системою, яка спеціально призначена для проведення розрахунків з векторами, матрицями та поліномами.

Термінами «вектор, матриця, тензор» позначають математичні об'єкти. Термінами «одномірні, двомірні, багатомірні масиви» - спосіб збереження цих об'єктів в комп'ютері.

Вводити масиви в *Matlab* можна:

- повним списком елементів;
- поелементно;
- завантаженням з зовнішніх файлів;
- генеруванням із застосуванням спеціальних вбудованих матричних функцій;

Під вектором в *Матлабі* розуміють одномірний масив чисел, під матрицею - двомірний масив, під тензором – багатомірний масив. За умовчанням передбачається, що будь-яка змінна є вектором чи матрицею. Число система сприймає як матрицю розміром (1*1), а вектор – рядок з N елементами - як матрицю розміром (1*N).

Звернення до елементів матриць та векторів проводиться вказанням імені та індексу елемента. Індекс записується після імені в дужках з розділювачом у вигляді коми «,». Початковим індексом масивів є 1.

Наприклад, **a(1)** відповідає першому елементу вектора **a**, **c(2,2)** – елементу з другого рядка та другого стовпця матриці **c**.

Списком елементи вектора задаються через пробіл чи кому в квадратних дужках.

Наприклад, запис рядка $\mathbf{v} = [1.2 \ -0.3 \ 1.2e-5]$ задає вектор V , з трьома елементами з значеннями 1.2, -0.3, $1.2 \cdot 10^{-5}$. Поелементне введення буде мати наступний вигляд: $\mathbf{v}(1)=1.2$, $\mathbf{v}(2)=-0.3$, $\mathbf{v}(3)=1.2e-5$.

Довгий вектор можна вводити частинами, котрі потім об'єднати:

$$\mathbf{v} = [\mathbf{v1} \ \mathbf{v2}]$$

Вектор, елементи якого складають арифметичну прогресію, можна ввести у вигляді, який схожий з ранжованою змінною пакету *MathCAD*. Такий опис змінної називають *векторним*:

$$\mathbf{I} = \mathbf{nz} : \mathbf{h} : \mathbf{kz} , \text{ де}$$

\mathbf{nz} – початкове значення прогресії (значення першого елемента вектора); \mathbf{kz} - кінцеве значення прогресії (значення останнього елемента вектора); \mathbf{h} - крок прогресії. Якщо середній параметр (крок прогресії) не вказано, то він за замовчанням приймається таким, що дорівнює одиниці.

Таким чином вводяться вектори - рядки. Вектор - стовпець вводиться аналогічно, але значення елементів відокремлюються знаком крапки з комою «;».

Функція `linspace(first, last, n)` дозволяє визначити вектор з рівномірно розподіленими елементами першим **first**, останнім елементами **last** та їхньою кількістю **n**. Крок зміни система визначає автоматично.

Функція `logspace(x1, x2, n)` дозволяє визначити вектор з логарифмічно розподіленими **n** елементами в діапазоні від 10^{x1} до 10^{x2} .

Введення значень елементів матриці проводиться в *Matlab* в квадратних дужках по рядках. При цьому елементи рядка матриці один від одного відокремлюються пробілом чи комою, а рядки один від одного - знаком ";" .

Дії над векторами та матрицями проводяться за правилами матричної математики. Для проведення поелементних дій, необхідно перед символом дії поставити символ крапки «.».

Основні матричні функції пакета наведено в додатку А.

В *Matlab* використано дві невідомі в математиці функції ділення матриць. При цьому вводяться поняття ділення матриць "зліва направо" та ділення матриць "справа наліво". Перша операція записується за допомогою знака $/$, а друга - \backslash .

Операція \mathbf{B}/\mathbf{A} еквівалентна послідовності дій $\mathbf{B} \cdot \mathbf{A}^{-1}$. Її призначено для розв'язання матричного рівняння: $\mathbf{A} \cdot \mathbf{X} = \mathbf{B}$.

Операція $\mathbf{A} \backslash \mathbf{B}$ рівносильна операції $\mathbf{A}^{-1} \cdot \mathbf{B}$, що є розв'язанням матричного рівняння: $\mathbf{X} \cdot \mathbf{A} = \mathbf{B}$.

Іноді потрібно зробити меншу матрицю з більшої чи, навпаки, вставити меншу матрицю таким чином, щоб вона стала частиною матриці більшого розміру, або провести дії із завданням стовпцем чи рядком. Для подібних дій в *Matlab* застосовується знак (" $:$ "):

$\mathbf{A}(:, n)$ - виділяє n -ий стовбець з матриці \mathbf{A} .

$\mathbf{A}(n, :)$ - виділяє n -ий рядок з матриці \mathbf{A} .

$\mathbf{A}(n1:n2, m1:m2)$ – виділяє з матриці \mathbf{A} підматрицю, в яку входять рядки з $n1$ до $n2$ та стовпці з $m1$ до $m1$.

$\mathbf{A}(:)$ - перетворює матрицю в вектор.

Примітка. Якщо верхньою межею номерів елементів матриці є її розмір, замість числового значення можна використовувати службове слово *end*.

"Розширяти" матрицю можна також, додаванням існуючих матриць ("блоків"). Якщо існують кілька матриць - блоків $\mathbf{A1}, \mathbf{A2}, \dots, \mathbf{AN}$ з однаковою кількістю рядків, то їх можна об'єднати в одну матрицю \mathbf{A} :

$$\mathbf{A} = [\mathbf{A1}, \mathbf{A2}, \dots, \mathbf{AN}]$$

Таку операцію називають *горизонтальною конкатенацією* матриць. Аналогічно, *вертикальна конкатенація* матриць реалізується (за умови, якщо всі блоки – матриці мають однакову кількість стовпців) шляхом застосування для розділення блоків замість коми коми з точкою:

$$\mathbf{A} = [\mathbf{A1}; \mathbf{A2}; \dots; \mathbf{AN}]$$

Для видалення елемента матриці слід присвоїти відповідному елементу «пусте» значення, тобто « $[]$ » - квадратні дужки без пробіла між ними.

Наприклад, для видалення першого рядка з матриці **A** слід записати

A(1, :) = [] ;

Для видалення з матриці **A** другого та третього стовпців слід записати

A(:, 2:3) = [] ;

Примітка. Дії з розширення та скорочення векторів виконуються аналогічно діям з матрицями.

Логічні операції з масивами проходять поелементно. Результатом є масив того ж розміру, що і вихідні.

Наприклад:

» **A = [1 2 3 4; 5 6 7 8; 9 10 11 12];**

» **B = [3 2 3 3; 5 6 2 2; 4 10 11 11]; C = A == B**

C = 0 1 1 0 1 1 0 0 0 1 1 0

Matlab зберігає матриці по стовпцях. Для доступу до елементів матриці можна використовувати один індекс, який визначає номер елемента в векторі.

Наприклад, звернення до матриці **A(2,2)** еквівалентно **A(6)**:

» **A(2,2) ans = 6**

» **A(4) ans = 6**

Масиви комірок

Для зберігання матриць однакових розмірів можна застосовувати тримірні масиви (тензори). Для зберігання різнорідних об'єктів (масивів з різними розмірностями, даних різних типів) призначені масиви комірок. *Масиви комірок* - це багатомірні масиви, елементи яких є копіями інших вже існуючих даних.

Створюються масиви комірок визначенням даних в фігурних дужках.

Наприклад, для матриці **S** розміром 2x2 виразом

» **C = {S sum(S) min(max(S)) prod(prod(S))}** отримаємо масив комірок **C** з першим елементом [2x2 double] - у вигляді самої матриці,

другим елементом [2x2 double] у вигляді вектора результату функції додавання `sum(S)`, третім елементом [3] - числом 3: результатом дії функції `min(max(S))`, четвертим елементом – числом 24: результатом дії функції `prod(prod(S))`.

Порожній масив комірок розміром **MxN** створюється командою `cell(MxN)`.

Для доступу до елементів масивів комірок застосовується індексація ім'ям масиву з індексом, який забрано в фігурні дужки. Наприклад,

```
» S{2}
ans =     4     6
» S{3}
ans =     3
```

Структури

Структури (тип *struct*) також є різновидом багатомірних масивів. Аналогом структури є запис (*record*) в мовах програмування. Елементами структури можуть бути різномірні дані. Доступ до елементів структури здійснюється за ключовим словом, яке вказується після символу «.» після імені структури та визначають поля структури.

Наприклад, скалярну структуру S з трьома полями «name», «age», «grade» можна визначити по одному елементу як

```
»S.name = 'Al Gor'; S.age = 22; S.grade = 'A';
```

або всі разом функцією

```
struct('field1',VALUES1,'field2',VALUES2,...)
```

```
»S=struct('name','Al Gor','age',22,grade,'A')
```

Додавання елементів структури здійснюється поелементно. *Наприклад*,

```
»S(2).name = 'Gor Al'; S.age = 33; S.grade = 'C+';
```

```
»S(2)=struct('name','Gor Al','age',33,grade,'C+')
```

Для того, щоб дістатися до змісту поля структури слід ввести ім'я структури з індексом в круглих дужках та через крапку ввести назву поля. *Наприклад,*

```
»x=S (1) . age
```

```
x=22
```

Для того, щоб дістатися до змісту всіх значень поля структури слід ввести ім'я структури та через крапку назву поля забраними в дужки. Застосування фігурних дужок перетворює поле в масив комірок, квадратних – в вектор. Елементами вектора для чисельних полів будуть числа. Єдиним елементом вектора для текстових полів буде рядок, що є зчепленням всіх рядків полів.

Наприклад,

```
»x={S.age}
```

```
x = [22]      [33]          % <1x2 cell> масив комірок
```

```
»x=[S.age]
```

```
x = [22 33]          % <1x2 double> вектор
```

```
»x={S.name}
```

```
x = 'Al Gor'      'Gor Al'  % <1x2 cell> масив комірок
```

```
»x=[S.name]
```

```
x = ['Al Gor Gor Al']          % <1x10 char> вектор
```

Особливість логічних дій над масивами

При порівнянні двох скалярів результатом є 1 або 0. Порівняння двох масивів виконується поелементно. Результатом є логічний масив з 1 та 0 того ж розміру, що і вихідні масиви.

При порівнянні скаляру з масивом скаляр порівнюється з кожним елементом масиву. Результатом є логічний масив з 1 та 0 того ж розміру, що і вихідний масив.

Логічні масиви з 0 та 1 не є тотожними до математичних масивів з 0 та 1. Логічні масиви створюються в результаті логічних дій над масивами. Вони можуть бути використані для адресації масивів. З нього витягаються номери елементів, значення яких є 1.

Наприклад,

```
>> y=(6<10)+(7>8)+(5*3==60/4)   %y=1+0+1=2
>> b=[15 6 9 4 11 7 14]; c=[8 20 9 2 19 7 10];
>> d=c>=b                        %d =      0  1  1  0  1  1  0
>> b == c                          %ans =     0  0  1  0  0  1  0
>> r = [8 12 9 4 23 19 10]
>> s=r<=10                        %s =      1  0  1  1  0  0  1
>> t=r(s)                          %t =      8  9  4  10
>> w=r(r<=10)                      %w =      8  9  4  10
```

4.2.4. Контрольні питання

Що є ознакою готовності *Matlab* до прийняття команд?

За якими правилами слід вводити числа в *Matlab*?

Що означає наявність символу «;» наприкінці рядка?

Що таке змінна **ans**?

Які типи даних наявні в *Matlab* ?

Для чого застосовують три крапки?

В яких одиницях задаються параметри тригонометричних функцій?

За яким пріоритетом проводиться розрахунок виразів *Matlab* ?

Які два типи логічних дій реалізовано в *Matlab* ?

Назвіть види функцій користувача.

Що є аргументом функції **eval**?

Як задати обраний формат виведення результатів?

Як задаються вектори та матриці?

Що таке векторна змінна?

Що таке конкатенація матриць?

Назвіть особливість логічних дій над масивами.

4.3. Програмування в Matlab

Основним режимом роботи з пакетом для проведення обчислювальних експериментів є програмний режим. Програми пакета зберігаються у вигляді m-файлів.

В *Matlab* розрізняють два види програм: *Script* (файл – сценарію чи керуюча програма) і *function* (файл - функція). Обидва різновиди мають розширення '*.m'.

Файли – сценарії призначено для основних програм, які керують обчислювальним процесом. У вигляді *файл - функції* оформляють окремі функції, тобто частини програми, які призначені для багаторазового виклику при змінних значеннях вхідних параметрів і які не можуть бути виконані без попереднього визначення значень вхідних параметрів.

Для написання програм в середовищі *Matlab* призначено вбудований редактор (рис. 4.5). Редактор викликається пунктом *M-file* підменю *New*, пунктом *Open...* підменю *File*.

4.3.1. Редактор М-файлів

Редактор М-файлів виконує синтаксичну перевірку програмного коду при введенні тексту. При цьому використовуються наступні кольори:

- зарезервовані слова мови програмування - синій колір;
- оператори, константи та змінні - чорний колір;
- коментарі після знаку "%" - зелений колір;
- символічні змінні (в апострофах) - зелений колір;
- синтаксичні помилки – червоний колір.

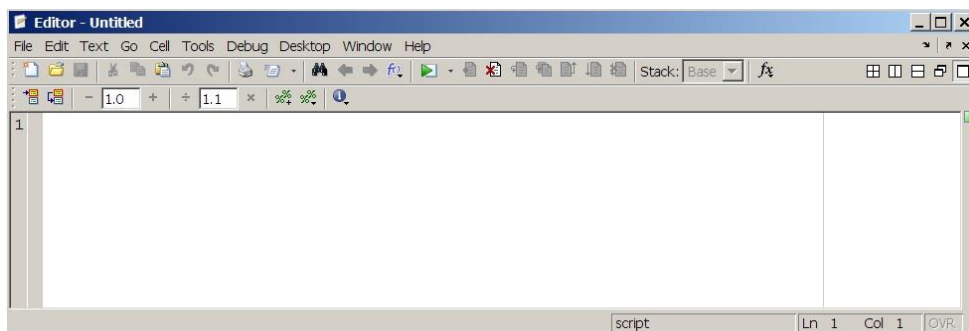



Рис. 4.5. Вікно вбудованого текстового редактора Matlab

Кнопки панелі редактора М-файлів

-  **New** - створення нового М-файлу;
-  **Open** - завантаження файлу з диску;
-  **Save** - запис файлу на диск;
-  **Print** - друк змісту поточного вікна редактора;
-  **Cut** - перенесення виділеного фрагмента в буфер обміну;
-  **Copy** - копіювання виділеного фрагмента в буфер;
-  **Paste** - вставка фрагмента з буфера в позицію курсору;
-  **Undo** - відміна попередньої операції;
-  **Redo** - повтор відміненої операції;
-  **Publish** - публікація файлу;
-  **Find text** - знаходження вказаного тексту;
-  **Show function** - показ функції;
-  **Set/Clear Breakpoint** – установка / скидання точки переривання;
-  **Clear All Breakpoints** - скидання всіх точок переривання;
-  **Step** - виконання одного кроку програми;
-  **Run** - запуск М-файлу на виконання;
-  **Step In** - покрокове трасування із заходженням в підлеглі М-файли;
-  **Step Out** - покрокова трасування без заходження в підлеглі М-файли;
-  **Exit Debug Mode** - завершення режиму налагодження.

Налагодження програм

Основним прийомом налагодження М-файлів є установка в них точок переривання (*breakpoints*). Вони установлюються/скидаються кнопкою *Set/Clear Breakpoint*. Скидання всіх точок переривання виконується кнопкою *Clear All Breakpoints*. Для установки точки переривання необхідно розмістити курсор напроти потрібного виразу та натиснути кнопку *Set/Clear Breakpoint*. При запусканні програма буде виконана до точки переривання, після чого поточні значення змінних будуть виведено у вікні робочого простору *Matlab*.

За допомогою кнопки *Step* можна виконати один крок обчислювань. Жовта стрілка вказує, в якому місці програми стався останов. При трасуванні

циклічних дій кожен крок виконання циклу фіксується у вікні командного режиму *Matlaby*.

При зупинці в точці переривання можна проконтролювати значення змінних «вручну» (шляхом введення імені змінної в командному рядку) або за допомогою перегляду точки переривання.

Кнопка *Exit Debug Mode* завершує режим налагодження.

Розбиття М-файлу на комірки

Редактор дозволяє перевірити виконання програми частинами. Для цього використовується розбиття програми на *комірки* (Cell).

Текст розбивається на *комірки* за допомогою рядків коментарів, які починаються з двох знаків відсотків (%%).

Для запуску засобів редактора роботи з *комірками* слід увімкнути режим *Cell Mode* пунктом *Enable Cell Mode* меню *Cell*. При цьому стають доступними інші пункти цього меню та з'являється панель інструментів для роботи з *комірками*.

Дії пунктів меню Cell

Insert Cell Divider - вставляння позначки *комірок* з двох символів відсотка (%%).

Insert Cell Divider around Selection - вставляння позначки %% до і після виділеного попередньо фрагменту.

Insert Text Markup - вставляння рядків зразків коментарів в місце знаходження курсору.

Previous Cell, Next Cell - зсув по *коміркам* вгору та вниз.

Evaluate Current Cell - виконання рядка поточної *комірки*. *Комірка* залишається поточною.

Evaluate Current Cell and Advance - виконання рядка поточної *комірки*. Поточною стає наступна *комірка*.

Evaluate Entire File - виконання всього файлу.

Примітка. Виконання *комірок* М-файлу проходить без попереднього збереження файлу на диск.

При виконанні файлу потрібно, щоб змінні були доступні в робочому середовищі (*Workspace*).

4.3.2. Файли сценарію

В режимі калькулятора оператор системи вводиться з клавіатури та відразу виконується після натискання клавіші “Enter”. Команди в командному вікні не можуть бути збережені. В режимі калькулятора не можна виконати кілька команд автоматично. Для здолання вказаних особливостей призначені файли-сценарію.

Файл сценарію – це записана на диск послідовність команд *Matlab*. При виконанні файлу команди виконуються автоматично рядок за рядком в тій послідовності, в якій вони були записані в файл. Результати дії команд виводяться в командне вікно. Якщо наприкінці рядка в файлі сценарію стоять точка з крапкою «;», то результат дії в командне вікно не виводиться.

Запуск сценарію з редактора *Matlab* здійснюється натисканням кнопки *Run*, запуск сценарію з командного вікна здійснюється введенням імені файлу без розширення.

Для того, щоб пакет знаходив файл сценарію, потрібно занести шлях до файлу в список робочого оточення.

Примітка. При завантаженні *Matlab* виконується команда *matlabrc*, яка завантажує для виконання файли *matlabrc.m* та *startup.m*, якщо вони існують.

4.3.3. Файл-функції

Файл-функція має наступну структуру:

рядок опису;

рядки коментарю;

тіло функції.

Перший рядок – опис має наступний синтаксис:

function <ПВР> = <ім'я процедури>(<ПВВ>) , де

<ПВР> - перелік вихідних результатів, <ПВВ> - перелік вхідних величин.

Якщо вихідних результатів декілька, вони забираються в квадратні дужки. Елементи переліку вхідних даних та вихідних результатів відокремлюються комами.

Поточні коментарі можуть бути вставлені в будь-якому місці файл-функції.

Виклик довідки **help** <ім'я функції> виводить на екран коментарі, які розташовано між рядком опису функції та першим виконавчим оператором чи першим порожнім рядком.

Перший рядок коментарів називається **Hi-line** рядком. Його зміст виводиться командою пошуку функцій **lookfor**, тому рекомендується в цьому рядку вводити коротку інформацію про дію функції.

Всередині файл-функції можуть використовуватись внутрішні файл-функції. При використанні внутрішніх функцій і основна, і внутрішні функції повинні завершуватися службовим словом **end**.

Рекомендується оформляти коментар файл-функції за наступною схемою [14]:

```
function [<ПВР>] = <ім'я функції>(<ПВВ>)
% Коротке пояснення дії функції.
% Детальне пояснення змісту, типу, розміру кожної змінної,
% що вказана в переліку вхідних параметрів ПВВ.
% Детальне пояснення змісту, типу, розміру кожної змінної,
% що вказана в переліку вихідних значень ПВР та змінних,
% які використані як глобальні.
% Використання інших функцій. Цей розділ заповнюється якщо
%процедура містить звернення до інших функцій, крім
%вбудованих.
% Автор : Вказується розробник, дата розробки.
< Порожній рядок >
```

Примітка. Запустити файл-функцію з редактора натисканням кнопки *Run* неможливо. При збереженні файл-функції слід називати файл таким же ім'ям, як ім'я функції.

Доступність змінних

Застосування змінних в файлах-сценаріях та файлах-функціях відрізняється.

В *файлах-функціях* змінні всередині файлу та в заголовку сприймаються як *локальні*, тобто всі значення після завершення процедури витираються та область пам'яті, яка була відведена під ці змінні, вивільняється. Змінні робочого простору та командного вікна всередині файл-функції не видимі та навпаки.

В *файлах-сценаріях* всі змінні є *глобальними* та зберігаються весь час сеансу пакета в області пам'яті "робочий простір" (*Workspace*). Робочий простір є спільним для всіх Script-файлів та вікна команд.

При необхідності доступу до змінних файл-функції ззовні, вона повинна бути описана як глобальна у всіх файл-функціях та файлі-сценарію, де передбачається її використання

global <var1> ... <varN>.

Різновиди функцій

Реалізація алгоритму в одній файл-функції не завжди є оптимальною. Деякі дії доцільно оформити у вигляді окремих службових функцій, які обслуговують основний алгоритм. Для цього в *Matlab* призначені підфункції (*subfunction*) та вкладені функції.

Підфункції

Підфункція розташовується в тому ж файлі, що і основна функція, після основної функції. Підфункція є залежною від основної функції та може викликатися тільки з основної функції.

Кожне звернення до підфункції з основної функції призводить до переходу до операторів підфункції та поверненню до точки виклику в основній функції.

Файл-функція може містити кілька підфункцій зі своїми вхідними і вихідними параметрами. Основна функція може бути тільки одна. Заголовок нової підфункції є ознакою кінця попередньої. Основна функція обмінюється інформацією з підфункціями тільки за допомогою вхідних і вихідних

параметрів. Змінні в кожній підфункції і в основній функції є локальними, вони доступні тільки в межах своєї функції.

Вкладені функції

Вкладені функції розташовується в тому ж файлі, що і основна функція безпосередньо в тілі основної функції. Якщо підфункція є зовнішньою по відношенню до основної функції, то вкладена функція є внутрішньою. Змінні основної функції доступні в вкладеній функції.

Основна та вкладена функції завершуються оператором **end**. В одному М-файлі можливо одночасне використання підфункцій та вкладених функцій. Рівень вкладеності функцій не обмежений. Функція може звертатися до своєї вкладеної функції, але не може використовувати вкладену функцію нижнього рівня. Вкладена функція може звертатися до інших вкладених функцій того ж рівня. Функція нижнього рівня може викликати функцію верхнього рівня, в яку вона вкладена.

Змінні основної та вкладених функцій є спільними. Змінні вкладених функцій одного рівня є недосяжними між функціями, тобто є локальними.

4.3.4. Керування обчислювальним процесом

Для керування обчислювальним процесом система має вбудовані оператори розгалуження, циклу, введення/виведення.

Оператори розгалуження

Оператори умовного переходу побудовані у вигляді складеного оператора, який починається з одного з службових слів **if**, **switch**, **try** та завершується службовим словом **end**.

В загальному випадку синтаксис *оператора умовного переходу if* має наступний вигляд:

```
if <умова 1>  
<операції1 >  
elseif <умова 2>  
    <операції 2>  
...
```

```

elseif <умова K>
    <операції K>
...
else
    <операції N> end

```

Рядки **elseif** можуть повторюватися або бути відсутніми, рядки **else** можуть бути відсутніми.

Якщо результат перевірки **умови1** вірний, програма виконує оператори **<операції1>**. В іншому випадку виконується перевірка **elseif**. Якщо умова виконана, виконується відповідна послідовність операторів **<операції K>**. Якщо жодна з умов не є вірною, виконуються **<операції N>**, які вказано після слова **else**.

Перевірка умов завершується після першої істинної умови.

Якщо логічна умова вміщує векторну змінну, то умова буде істиною, якщо вона істинна для ВСІХ елементів змінної.

Примітка. Повна кількість гілок розгалуження оператора if-elseif-else дорівнює трьом.

Оператор перемикання **switch** має наступний синтаксис:

```

switch <вираз, скаляр чи рядок символів >
case <значення 1>    <оператори 1>
case <значення 2>    <оператори 2>
...
otherwise            <оператори >
end

```

Оператор реалізує розгалуження обчислювань в залежності від значення змінної чи виразу, шляхом порівняння вказаних значень в рядку **switch** зі значеннями, які вказано в рядках **case**. Відповідна група операторів **case** виконується, якщо значення виразу співпадає з заданим

значенням. Якщо значення не співпадає ні з одним із значень в групах **case**, то виконуються оператори, які розташовані за словом **otherwise**.

Вираз в рядку **switch** повинен мати скалярний результат.

Перевірка умов завершується після першої вірної умови.

Оператор спроби має наступний синтаксис

```
try  
    оператори 1  
catch  
    оператори 2  
end
```

Команда намагається виконати «оператори 1». Якщо при виконанні виникає помилка або результатом є невизначеність, то будуть виконані «оператори 2».

Оператори циклу

Всі оператори циклу побудовані у вигляді складеного оператора, який починається з одного з службових слів **while**, **for** та завершується службовим словом **end**.

В системі реалізовано два різновиди циклу: *цикл з передумовою та арифметичний цикл*. Цикл з передумовою призначено для алгоритмів, в яких невизначена кількість повторювань. Арифметичний цикл використовується, коли кількість кроків циклу визначена до початку циклу.

Синтаксис оператора *циклу з передумовою* має наступний вигляд:

```
while <умова >  
    <оператори >  
end
```

Оператори всередині циклу виконуються тільки в разі виконання умови. В якості умови можна використовувати логічні або арифметичні вирази. Цикл повторюється доки значення виразу умови не дорівнює нулю.

Синтаксис *арифметичного оператора циклу* має наступний вигляд:

```
for <змінна> = <ПЗ>:<К>:<КЗ>
```

<оператори>

```
end
```

 , де

<змінна > - ім'я керуючої змінної циклу - лічильника циклу,

<ПЗ> - початкове значення змінної циклу, <К> - значення кроку змінної циклу, <КЗ> - кінцеве значення змінної циклу.

Якщо параметр <К> не вказано, його значення приймається рівним одиниці.

В якості діапазону змінної циклу можна явно вказати весь набір значень у вигляді вектору - рядка чи матриці. Наприклад:

```
1) for m = [ 2, 5, 7, 8, 11, 23 ]
```

```
2) A = [ 1 2; 3 4] ; for k = A
```

Цикл буде повторюватися стільки разів, скільки стовпців є в матриці А, тобто два рази для наведеного прикладу. В кожній ітерації змінна циклу буде приймати значення чергового стовпця матриці. В такому випадку вона є вектором-стовпцем.

Наприклад, фрагмент

```
S = 0; a = [ 1 2; 3 4] ;
```

```
for k = a ; S=S+sqrt(k(1)^2+k(2)^2); end
```

обчислює суму «довжин» стовпців матриці А.

Переривання дій

Для дострокового виходу з циклу застосовуються оператори **break**, **continue**. Оператор **break** достроково завершує цикл та передає керування команді, що стоїть після слова кінця циклу **end**. Оператор **continue** завершує виконання операторів, які знаходяться нижче оператора **continue**, та починає наступну ітерацію циклу.

Оператор **return** завершує виконання функції та передає керування в точку виклику – командне вікно, іншу функцію. Схожу дію виконує оператор **error** ('message'). Він також зупиняє виконання програми та виводить в командне вікно текст 'message' червоним кольором.

4.3.5 Введення – виведення інформації

Для забезпечення взаємодії з користувачем між М-файлом та командним вікном в системі *Matlab* призначені команди **disp**, **sprintf**, **fprintf**, **input**, **menu**, **keyboard**, **pause**.

Команда **disp(X)** проводить виведення значення вказаної змінної або тексту в рядок командного вікна. Функція може мати тільки один аргумент:

```
disp (<змінна або текст в апострофах >)
```

Щоб вивести значення кількох змінних за допомогою команди **disp**, потрібно створити єдиний об'єкт, який би містив всі значення. Це можна зробити об'єднанням відповідних змінних в вектор, використовуючи операцію створення вектора-рядка: **x = [x1 x2 ... xN]**.

Тоді виведення значень кількох змінних в один рядок буде мати наступний вигляд:

```
disp ([x1 x2 ... xN])
```

Аналогічно можна об'єднувати кілька текстових змінних, наприклад:

```
>> x1='psi'; x2='fi'; x3='teta'; x4='w1';
```

```
>> disp([x1 x2 x3 x4])
```

*Примітка. Елементи вектора не можуть мати різний тип. Тобто вони можуть бути або всі числами, або всі - рядками. Для виведення функцією *disp* комбінованої інформації з чисел та тексту потрібно використати функції перетворення чисел в рядки та сформувати єдиний рядок.*

Схожу дію виконує оператор **warning('message')**. Він виводить в командне вікно текст '**message**' червоним кольором.

Функція **sprintf** створює текстовий рядок, який містить пояснюючий текст та значення числових змінних у визначеному вигляді. Вигляд рядка задається форматом.

```
[S,ERRMSG] = sprintf(format,A, B,...), де
```

A, B,... – послідовність аргументів. Кожен аргумент може бути скаляром, вектором, матрицею, виразом. Елементи відокремлюються комою.

Якщо аргумент **A** – вектор, результуючий рядок формується контактеною всіх елементів;

format – опис форми виведення аргументів (дивись додаток A);

S – результуючий текстовий рядок;

ERRMSG - опціональне поле. Містить повідомлення про помилку або пусту матрицю.

Наприклад, фрагмент

```
x=4; a=0.5; sprintf('x= %i a= %f', x, a)
```

поверне рядок `'x= 4 a= 0.500000'`

```
sprintf('x= %i a= %3.1f', x, a)
```

поверне `'x= 4 a= 0.5'`

```
sprintf('x= %i a= %e', x, a)
```

поверне `'x= 4 a= 5.000000e-001'`

Замість використання для виведення інформації двох функцій **disp**, **sprintf** можна застосувати одну функцію **fprintf**. Ця функція призначена для виведення форматованої інформації в текстові файли.

Функція має наступний синтаксис

```
fprintf(FID, format, A, B, ...),
```

Якщо замість дескриптора файлу **FID** вказати в якості першого аргументу 1 або його взагалі не вказувати, то виведення буде проведено на екран.

Функція **input('<запрошення >', '<s'>)** виводить "**запрошення**" на екран командного вікна та зчитує введене користувачем з клавіатури. Наприклад, **c=input('c=')**.

В разі використання опціонального параметру **'s'**, функція сприймає введене значення як текстовий рядок. Масиви слід вводити в квадратних дужках.

Функція **menu** виводить на екран вікно меню користувача вибору з альтернатив майбутніх дій.

```
k=menu('Перший рядок заголовка', 'Альтернатива1',  
'Альтернатива2', 'Альтернатива n')
```

Виконання програми тимчасово переривається. Система чекає дії у вигляді вибору мишею однієї з "кнопок" меню з альтернативами. Після вибору функція повертає номер обраної альтернативи (1, 2 чи n). Кількість альтернатив може бути до 32. При зачиненні вікна без вибору альтернативи повертається 0. Для організації циклічного процесу потрібно застосувати разом з командою **menu** команду циклу з передумовою **while**.

Наприклад, запис

```
k=menu('Приклад', 'Вибір 1', 'Вибір 2')
```

призводить до появи на екрані меню, зображення рис. 4.6.

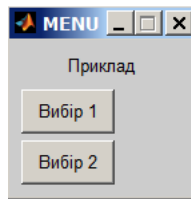


Рис. 4.6. Приклад команди **menu**

Функція **pause (< n>)** тимчасово перериває виконання програми до тих пір, доки користувач не натисне будь яку клавішу клавіатури. Якщо задано параметр "n", то буде проведено затримку на n секунд.

Функція **keyboard** перериває виконання програми з М - файлу та передає керування клавіатурі (командному вікну). Ознакою передачі керування є символ «k>>».

Користувач може здійснити будь які дії, при цьому йому доступні всі команди системи *Matlab*. Для повернення до виконання програми необхідно набрати команду **return**.

Більш детальний опис функцій введення – виведення наведено в додатку А.

4.3.6. Контрольні питання

Назвіть різновиди програм *Matlab*.

Назвіть можливості редактора М-файлів.

Для чого застосовуються комірки при програмуванні?

Як запускається скрипт?

Яку структуру має файл-функція?

Як запустити файл-функцію?

Для чого застосовуються підфункції, вкладені функції?

Яка повна кількість логічних гілок в повному операторі if?

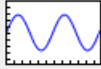
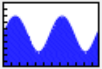
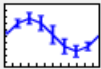
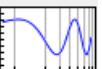
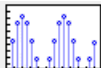
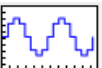
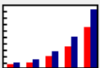



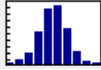

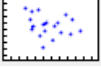
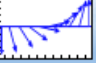
Якими операторами реалізуються цикли?

Якими засобами реалізується введення-виведення інформації?

4.4. Двомірні графіки

Для дослідження функцій однієї змінної, наочного представлення результатів обчислень система дозволяє створити значну кількість типів двомірних графіків за допомогою вбудованих засобів. Основні різновиди двомірних графіків наведено в таблиці 4.2.

Таблиця 4.2. Основні типи двомірних графіків

Вигляд	Зміст	Вигляд	Зміст
	Графік функції лініями в декартових координатах в лінійному масштабі		Графік функції лініями з фарбуванням зони під графіком
	Графік функції з діапазоном відхилень лініями в декартових лінійних координатах		Графік функції лініями в декартових координатах в напівлогарифмічному чи логарифмічному масштабі
	Графік функції вертикальними лініями		Графік горизонтальними відрізками ліній
	Стовпова діаграма		Секторна діаграма
	Графік функції в полярних координатах		Векторна діаграма в полярних координатах
	Гістограма в декартових координатах		Гістограма в полярних координатах
	Точкова діаграма		Векторна діаграма в лінійних координатах

Графіки в *Matlab* виводяться в спеціальні окремі графічні вікна **figure** (Рис. 4.7). За умовчанням відкривається одне вікно, в якому відображається один графік. Наступний графік за умовчанням виводиться в те ж вікно замість останнього графіка. При необхідності можливо організувати виведення кількох графіків в одне вікно або відкрити кілька вікон з графіками.

Створення та оформлення графіків може проводитися з командного вікна чи скрипт-файлів, з вікна робочого простору або безпосередньо у вікні графіки.

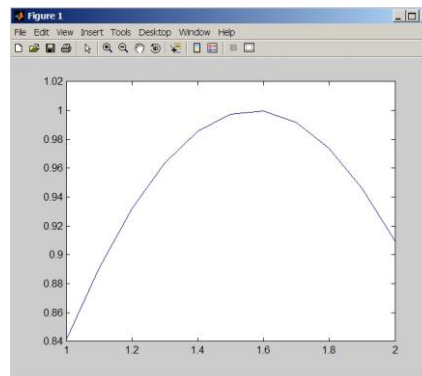


Рис. 4.7. Вигляд графічного вікна

4.4.1. Базові команди керування графікою

Для створення нового та перемикання між існуючими вікнами використовується команда **figure(<n>)**. При виклику без параметра відкривається нове вікно для графіка. Нове вікно стає поточним. Наступні після створення нового вікна графіки виводяться в поточне вікно. Новому вікну система присвоює номер, який є посиланням на вікно. При виклику з параметром **n** створюється або активується для виведення вікно з номером **n**.

Закрити вікно можна командою **close** або **close(n)**.

Основною функцією графічного відображення векторів є функція

<N>=plot(<n>,<x1>,y1,<s1>,<x2>,<y2>,<s2>,...), де

xi, yi - вектори, елементами яких є значення аргументу та функції, що відповідають *i*-й кривій графіка;

si - опціональні символні формати графіків. Формати можуть бути задано якостями лінії у вигляді рядка з трьох символів в апострофах 'лінія маркер колір' або послідовності "якість", "значення" в термінах низькорівневого програмування.

n – опціональний номер вікна, в яке буде виведено графік. В разі відсутності графік виводиться в поточне вікно.

Коди типів ліній: «-» - суцільна, «:» - точкова, «-.» –штрих-пунктирна, «--» - пунктирна.

Коди кольорів: **b** – синій, **g** – зелений, **r** – червоний, **c** – рожевий, **m** – блакитний, **y** – жовтий, **k** – чорний, **w** – білий.

Коди маркерів: «.» – точка, «o» – коло, «x» – хрест, «+» - плюс, «*» - зірка, «s» – квадрат, «d» – ромб, «v» – трикутник вниз, «^» - трикутник вгору, «<» - трикутник вліво, «>» - три, «p» – п'ятикутник, «h» – шестикутник.

Наприклад, запис

```
>>x=0:0.1:6; y=cos(x);  
>>plot(x, sin(x), '-c', x, y, 'xr')
```

будує в графічному вікні графіки функцій $f(x)=\sin(x)$ та $f(x)=\cos(x)$. Перший графік малюється безперервною лінією блакитним кольором, другий – пунктирною лінією з символом "x" червоним кольором.

Схожу дію виконує команда **area (<X>, Y, <LIMIT>)**. Результатом виклику функції **area** з двома аргументами – векторами X,Y - є графік, аналогічний функції **plot**, тільки з зафарбованою областю під графіком. В разі, коли Y є матрицею, будуються графіки стовпців матриці з накопиченням значень. Опційний параметр **LEVEL** визначає горизонтальну лінію нижньої межі графіка.

Команда **clf** очищує поле графіка.

Команда **<H>=loglog (...)** виводить графік функції в логарифмічному масштабі по обом осям. Синтаксис аналогічний команді **plot**.

Команди `<H>=semilogx(...)`, `<H>=semilogy(...)` виводять графік функції в напівлогарифмічному масштабі по осі X або Y. Синтаксис є аналогічним команді `plot`.

Команда `[AX, H1, H2]=plotyy(X1, Y1, X2, Y2, <F1>, <F2>)` аналогічна функції `plot`, але з відображенням двох вертикальних осей: ліва - для даних X1, Y1, права - для даних X2, Y2. `<F1>`, `<F2>` - опціональні рядки, що забрані в апострофи, вказують тип графіка: `plot`, `semilogx`, `semilogy`, `loglog`, `stem` тощо.

Команда `hold on` вмикає виведення графіків на вже відкрите вікно без витирання раніше намальованих. Команда `hold off` завершує дію `hold on`.

Команда `<H>=polar(T, R, <S>)` - виводить полярний графік залежності R(T). Форматується аналогічно функції `plot`.

Команда `[X, Y]=fplot(F, L, <T>, <N>, <S>)` виводить графік функції `F` в діапазоні `L` з точністю `<T>` типом лінії `<S>` та числом точок `<N>`. Функція `F` може задаватися рядком з іменем функції - `fplot('myfun(x)', [-1 1]);` посиланням на функцію - `fplot(@myfun, [-1 1]);` безпосереднім записом функції - `fplot(@x*sin(1./x), [-1 1]).`

Діапазон є вектором зі значеннями `[XMIN XMAX]` чи `[XMIN XMAX YMIN YMAX]`. Тип лінії визначається аналогічно команді `plot`. Чисельне значення `0<T<1` визначає відносну точність малювання шляхом зміни кількості точок та кроку між ними. Чисельне значення `N >= 1` визначає мінімальну кількість точок на графіку.

Наприклад,

```
>>fplot('1+sin(x)', [0 4], ':', 0.1)
```

малює точками графік виразу `1+sin(x)` з функцією користувача `sin(x)` для значень аргументу від 0 до 4 з точністю 0.1.

Спрощеним аналогом команди `fplot` є команда

```
ezplot(FUNX<, FUNY>, <[TMIN, TMAX]>, <FIG>).
```

Функція малює графік виразу **FUNX** в діапазоні $-2\pi < X < 2\pi$. В разі розташування після коми виразу **FUNY** будується параметричний графік **FUNX (T)** та **FUNY (T)**. Опціональний вектор **[TMIN, TMAX]** визначає діапазон зміни аргументу. При існуванні вектору діапазону значенням посилення (номеру) на вікно **FIG** можна направити виведення графіку в визначене вікно.

4.4.2. Оформлення графіків

Оформлення графіків проводиться ПІСЛЯ створення вікна та виведення графіків в нього. На полі графіку можна додати сітку, визначити діапазон осей, нанести написи на осі, нанести пояснення на графік.

Функція **grid <mode>** малює сітку на полі графіка. Сітка координат завжди відповідає "цілим" крокам.

Опції можливих режимів:

on \ off - включає/виключає відображення основної сітки,

без параметра – перемикає стани відображення основної сітки,

minor - переключає відображення мілкої сітки.

Функція **axis (<[xmin, xmax, ymin, ymax]>, <mode>)** додає (змінює) розмірні осі графіка. Параметр **<mode>** :

OFF/ ON – вмикає/вимикає зображення,

auto - встановлює масштаби по обом осям за замовчанням,

ij - пересуває початок відліку в лівий верхній кут (матрична система координат),

xу - встановлює декартову систему координат з початком відліку в лівому нижньому куту,

equal- встановлює однаковий масштаб по обох осях графіка,

SQUARE – встановлює квадратну форму поля вікна,

NORMAL – поновлює поле графіку до повного розміру.

Без визначення вектора масштабу ключові слова слід писати без дужок. Визначення масштабу потребує написання команди з аргументами в дужках,

а ключових слів – в апострофах. Додаткове слово **tight** стягує зображення до меж графіка. Може використовуватися разом з іншими режимами.

За замовчанням діє режим **'auto'**, **'on'**, **'xy'**. Визначення вектора масштабу **[xmin, xmax, ymin, ymax]** встановлює режим **manual**.

Наприклад,

```
>>axis equal tight
```

```
>>axis ([-1 1 2 2])
```

Функція **title('текст')** виводить заголовок над графіком.

Функція **xlabel('текст')** виводить напис осі X.

Функція **ylabel('текст')** виводить напис осі Y.

Функція

```
text(x,y,'текст',<'FontName','Font','FontSize','Size'>)
```

розміщує текст на полі графіка, при цьому початок тексту розміщується в точку с координатами x та y. Координати повинні бути задано в одиницях величин, в яких будується графік та знаходиться всередині діапазону цих величин.

Функція **legend** виводить коментарі у вказане місце.

```
<[L,OB,OU,OT]> = legend <(M)>, де
```

```
M - 'string1','string2',..., 'OFF/SHOW', 'BOXOFF/BOXON',  
'Location', LOC, 'Orientation', OR ;
```

string1 є поясненням до графіка 1, **string2** до графіка 2 і т.д.;

OFF/ SHOW - керує видимістю коментаря;

BOXOFF/BOXON - керує відображенням контуру навколо коментаря;

'Location','LOC' - визначає положення коментаря. Положення описані в таблиці 4.3 ;

'Orientation', 'vertical/horizontal' - визначає орієнтацію коментаря.

Функція **subplot(m, n, p)** розбиває область виведення графічної інформації на кілька підобластей, в кожену з яких можна вивести графіки функцій. Діапазони змін змінних по осях координат цих підобластей незалежні одне від одного. Аргументи: *m* – кількість стовпців, *n* – кількість рядків, *p* - номер вікна, в якому буде малюватися графік. Підвікна нумеруються зліва направо та згори вниз. Можливо в якості номера підвікна *p* використовувати вектор з об'єднаними в одне кількох підвікон.

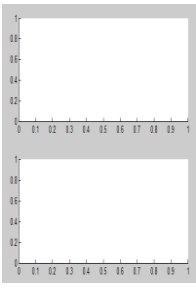
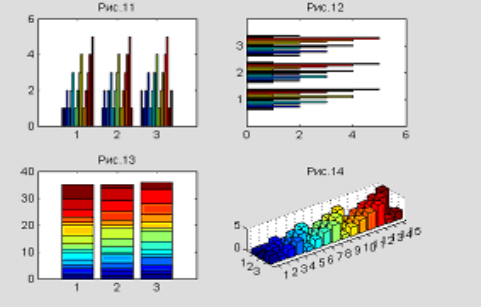
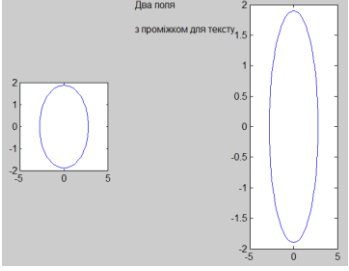
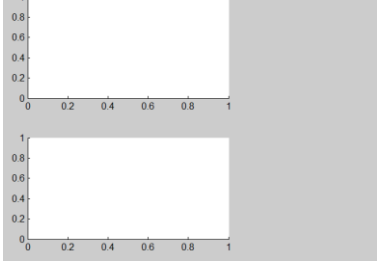
В таблиці 4.4 наведено приклади організації графіків в одному вікні.

Таблиця 4.3. Положення коментаря

LOC	Положення	LOC	Положення
'North'	Всередині згори	'BestOutside'	Оптимально ззовні
'South'	Всередині знизу	'Best'	Оптимально всередині
'East'	Всередині праворуч	'WestOutside'	Ззовні ліворуч
'West'	Всередині ліворуч	'EastOutside'	Ззовні праворуч
'NorthEast'	Всередині згори праворуч	'SouthOutside'	Ззовні знизу
'NorthWest'	Всередині згори ліворуч	'NorthOutside'	Ззовні згори
'SouthEast'	Всередині знизу праворуч	'SouthWest'	Всередині знизу ліворуч
'NorthEastOutside'	Ззовні згори праворуч	'SouthEastOutside'	Ззовні згори праворуч
'NorthWestOutside'	Ззовні згори ліворуч	'SouthWestOutside'	Ззовні знизу ліворуч

Функція **annotation(<fig,>TYPE<,pos>)** виводить на поле графіка вікна **fig** елемент оформлення типу **TYPE** в місце з координатами **pos**. Координати задаються у відносних величинах розміру поля графіка від 0 до 1. Можливо виведення прямокутника - '**rectangle**', еліпса - '**ellipse**', тексту - '**textbox**', лінії - '**line**', стрілки - '**arrow**', двосторонньої стрілки - '**doublearrow**', '**textarrow**' – стрілки з текстом.

Таблиця 4.4. Приклади виведення графіків в одному вікні.

	
<pre>subplot (2,1,1) subplot (2,1,2)</pre>	<pre>subplot (2,2,1);bar(y) title('Рис.11','FontName','Courier') subplot(2,2,2);barh(y) title('Рис.12','FontName','Courier') subplot(2,2,3); bar(y,'stacked') title('Рис.13','FontName','Courier') subplot(2,2,4); bar3d(y) title('Рис.14','FontName','Courier')</pre>
<p>Поля для двох графіків одне під одним</p>	<p>Поля для чотирьох графіків у вигляді матриці 2x2 з підписами</p>
	
<pre>subplot(1,3,1) plot(x,y);axis square subplot(1,3,3) plot(x,y) subplot(1,3,2) axis off text(0,1,'Два поля') text(0,0.9,'з проміжком для тексту')</pre>	<pre>subplot(2,3,[1 2]) subplot(2,3,[4 5]) subplot(2,3,[3,6]) axis off</pre>
<p>Поля для рядка з двох графіків та поля для тексту між ними</p>	<p>Поля для двох графіків збільшеної ширини один під одним та поля для тексту праворуч на два рядки</p>

Опційне поле визначає вектор для прямокутника, еліпса, тексту: координати x , y нижнього лівого кута, довжина, висота, - для стрілки, лінії: два вектори x , y - $x(1)$, $y(1)$ початку, $x(2)$, $y(2)$ кінця. Рядок тексту визначається парою `'String'`, `'Value'`.

Наприклад,

```
>>rh=annotation('rectangle',[.1 .1 .3 .3]);
>>ah=annotation('arrow',[.9 .5],[.9,.5]);
```

В текстові рядки можливо вводити спеціальні символи за допомогою модифікаторів. Модифікатором слугує символ зворотній слеш «\». Наприклад, літеру π вводять як `\pi`. Коди деяких модифікаторів наведено в таблиці 4.5. Дія модифікаторів розповсюджується на один символ, до кінця рядку, на групу символів, яка забрана в фігурні дужки.

Таблиця 4.5. Модифікатори символів

Вигляд	Код	Вигляд	Код
→	<code>\rightarrow</code>	text	<code>\rm{text}</code>
text	<code>\bf {text}</code>	text	<code>\fontname{name}</code>
text	<code>\fontsize{size}{text}</code>	text	<code>\it {text}</code>
x_a	<code>x_{a}</code>	x^b	<code>x^{b}</code>
α	<code>\alpha</code>	φ	<code>\Phi</code>
β	<code>\beta</code>	Δ	<code>\Delta</code>
γ	<code>\gamma</code>	Γ	<code>\Gamma</code>

4.4.3. Спеціальні графіки

Функція `<H> =bar (<x>, y, <w>, <LINESPEC>)` виводить стовпову діаграму $y(x)$ з шириною стовпця W . Якщо x не задано, аргументом вважається номер елемента вектора « y ». **LINESPEC** – задає колір стовпців (один з 'rgbycmkw').

Функція `barh (<x>, y, <w>, <LINESPEC>)` виводить стовпцеву горизонтальну діаграму.

Функція **stem(<x>,y,<w>,<LINESPEC>,'filled')** виводить «лінійну» діаграму з стовпцями нульової товщини та маркерами на кінці ліній. Формат є аналогічним формату функції **bar**. Додаткове поле **'filled'** замальовує маркери кінця ліній.

Функція **errorbar(X,Y,L,U)** виводить графік з нанесенням діапазону зміни значень **Y** від **U** (вектор верхніх відхилень) до **L** (нижніх). Якщо задано один аргумент, вважається, що діапазон є симетричним. Формат аналогічний функції **plot**.

Функція **hist(Y,<X>)** будує графік гистограми. Відображає кількість елементів вектора **Y**, які потрапили в піддіапазони, що задані вектором **X** або скаляром **X** (кількістю підінтервалів). При відсутності **X**, приймається розбиття на 10 інтервалів. Повертає розподіл **N** та центри розподілу **X**.

Функція **pie(Y,X)** виводить секторну діаграму.

Функція **scatter(X,Y,<S>,<C>,<M>,<filled>)** малює точкову діаграму точок по векторах **X,Y** розміром **S** кольором **C** маркером **M**. Опціональний параметр **'filled'** замальовує маркер. Колір точки визначається назвою або RGB вектором: жовтий: [1 1 0] або 'y', або 'yellow'. Червоний: [1 0 0] або 'r', або 'red'. Зелений: [0 1 0] або 'g', або 'green' тощо.

Наприклад, точкова діаграма зафарбованими точками розміром 60 пікселів синього кольору у вигляді ромба може бути намальована виразом

scatter(x,y,60,'blue','d','filled') .

Для зображення векторів та комплексних чисел призначено команди векторних діаграм в полярних координатах **compass(X,<Y>,<LINESPEC>)** та декартових координатах **feather(X,<Y>,<LINESPEC>)** .

Функція **compass** малює вектори, які мають початкову точку (0,0) та кінцеву точку, яка визначена елементами векторів **X,Y**. Команда **feather** малює вектори, початкові точки яких лежать на осі абсцис, а кінцеві визначені елементами векторів **X,Y**. Якщо застосовано один комплексний параметр, координати беруться з дійсної та уявної частин. Визначення

параметрів ліній в опційному параметрі **LINESPEC** аналогічно команді **plot**.

Рекомендації по організації графічних результатів обчислювального експерименту

Інженерні графіки повинні мати такий вигляд [14,15], щоб з них легко було зчитувати значення функції при будь-яких значеннях аргументу з відносною похибкою до кількох процентів. Для цього координатна сітка графіків повинна відповідати цілим числам десяткового розряду.

Інженерна графічна інформація повинна супроводжуватися текстом, з якого повинно бути зрозуміло, що зображено на графіку, яка математична модель застосована, які чисельні значення мають параметри досліджуваного об'єкту.

Додатково, можуть бути наведена інформація про деякі значення обрахованих інтегральних параметрів, назву програми, автора, дату експерименту.

Якщо потрібно порівняти графіки функцій, значення яких відрізняються не більш, ніж на один порядок, графіки доцільно виводити в одному графічному полі, якщо діапазони значень функцій значно розрізняються, функції, що описують подібні фізичні процеси та мають невід'ємні значення, доцільно виводити в одне графічне поле в логарифмічному масштабі. Функції однієї моделі, що мають різну фізичну природу та залежать від одного аргументу, доцільно виводити в одному графічному вікні, але в різних графічних полях. При цьому слід розміщувати графіки один під одним таким чином, щоб однакові значення аргументу на всіх графіках розташовувались на одній вертикалі.

Побудова графіків з робочої області

Якщо попередніми розрахунками сформовано векторні дані, які у вигляді змінних розміщені в робочому середовищі, замість ручного написання команд слушно скористатися кнопкою графіки робочого середовища.

Для цього слід виділити вектори аргументу та функції мишею з затисненою клавішею <Ctrl> та обрати тип графіка з випадаючого списку

вікна робочого середовища *Workspace* (рис. 4.8). Система сформує відповідну команду графіки та виконає її. Робоче середовище формує команду з використанням засобів дескрипторної графіки.

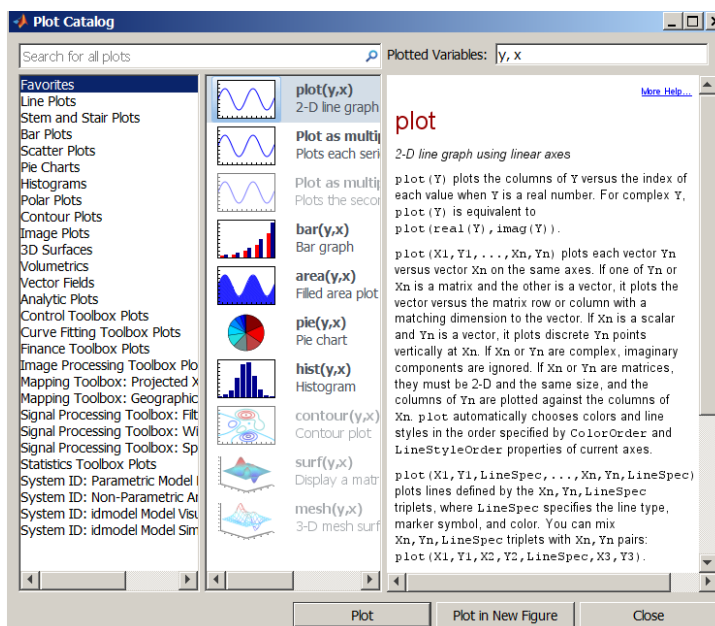


Рис. 4.8. Список графіків робочого середовища

Побудова та редагування графіків в графічному вікні

Графічне вікно (рис. 4.7) не тільки пасивно відображає графічну інформацію. Воно має потужні інструменти для налаштування вигляду зображення. З включенням всіх засобів вікно має наступний вигляд (рис. 4.9.).

Вікно містить поле графіка, меню вікна, кнопкові панелі: вікна (*Figure Toolbar*), камери 3D перегляду(*Camera Toolbar*), анотацій (*Plot Toolbar*), - редактор властивостей графічних елементів (*Property editor*), панель вікна (*Figure palette*), браузер графічних елементів (*Plot browser*), інспектор властивостей графічних елементів (*Inspector*).

Меню графічного вікна містить команди для керування самим вікном та інструментами налаштування вигляду графіків. Команди поєднані в логічні групи.

Група *File* містить дії з вікном: *New...*, *Open...*, *Close*, *Save*, *Save As...*, *Generate code*, *Import Data*, *Preferences...*, *Save Workspace*, *Page Setup...*, *Export Setup...*, *Print Preview...*, *Print*.


Група *Edit* містить стандартні команди Windows застосунків: *Undo*, *Cut*, *Copy*, *Paste*, *Delete*, *Select All*, *Copy Figure*, *Copy Options*, - для перенесення графіків між графічними вікнами. Звідси опцією *Figure properties* активується редактор властивостей *Property editor* в режимі редагування вікна **figure**, опцією *Axis properties* - в режимі редагування поля графіка **axis**, опцією *Current object properties* - в режимі редагування елемента графіка, який обере користувач щигликом миші на елементі. Опція *Colormap* активує редактор карти кольорів *Colormap editor*, яка використовується для розфарбування тримірних графіків.

Група *View* керує виведенням на екран кнопоквих панелей, редактора властивостей графічних елементів, панелі вікна, браузера графічних елементів.

Група *Insert* повторює кнопокву панель анотацій та панель вікна. В ній розміщені пункти керування вставлянням на поле графіка елементів оформлення: підписів осей (**X Label**, **Y Label**, **Z Label**), підпису графіка (**Title**), пояснень (**Legend**), карти кольорів (**Colorbar**), додаткових елементів оформлення у вигляді стрілок, ліній, прямокутників, еліпсів, текстів.

Група *Tool* повторює кнопокву панель вікна та керує опціями перегляду графіка: масштабуванням, зсувом, поворотом.

Групи *Debug*, *Desktop*, *Window*, *Help* – повторюють пункти меню системи.

Зміна властивостей графіків може проводитись безпосередньо в графічному вікні. Виклик редактора проводиться через пункт меню *Edit* або кнопкою вмикання режиму редагування  з панелі вікна, командою **plottools** командного вікна. Вигляд та можливості редактора визначаються типом графічного елемента.

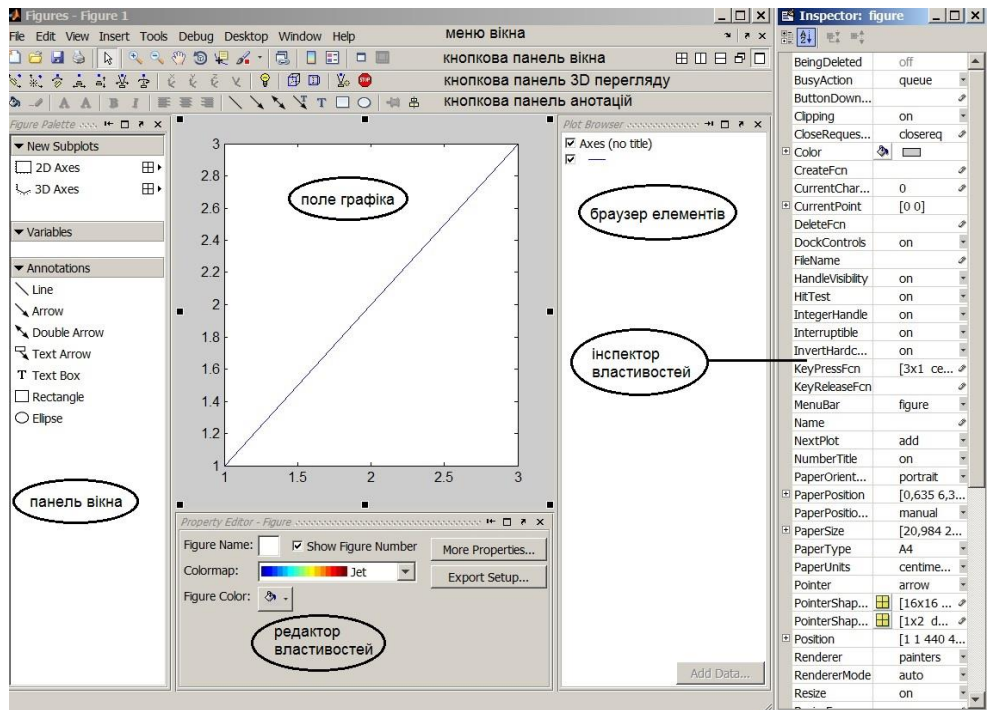


Рис. 4.9. Графічне вікно

На рис. 4.10 показано вигляд редактора для редагування властивостей осей: написів на осях, кольору та шрифту написів, типу осі (лінійна, логарифмічна), діапазоном аргументів осі, виведенням сітки.

На рис. 4.11 показано вигляд редактора для редагування властивостей графіків: типу, кольору, товщини ліній, типу, кольору маркерів. В полях *Data source* можливо змінити зміст з списку визначених в *Workspace* даних по осях.

Для додавання елементів оформлення - підписів осей та графіка, пояснень, карти кольорів, додаткових елементів оформлення у вигляді стрілок, ліній, прямокутників, еліпсів, текстів - можна використати групу *Insert* меню вікна, кнопки панелі анотації, панель вікна *Figure Palette*. Розміщення елементів та визначення їхніх розмірів проводиться безпосередньо на полі графіка в діалоговому режимі за допомогою миші.

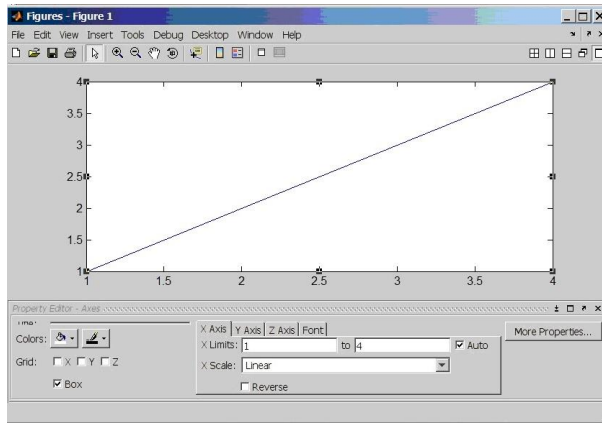


Рис. 4.10. Вигляд редактора в режимі редагування осей

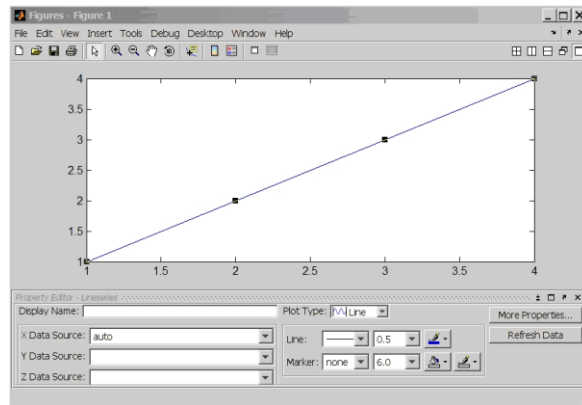


Рис. 4.11. Вигляд редактора в режимі редагування ліній

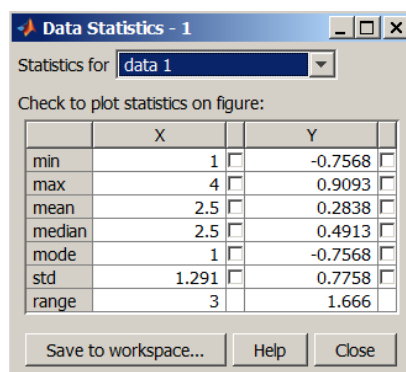
Зручною є можливість виконання дій з графіками за допомогою контекстних меню. Контекстне меню викликається натисканням правої кнопки миші після того, як об'єкт обрано клацанням лівою кнопкою миші. Ознакою обрання об'єкта є поява чорних прямокутних маркерів на ньому. Зміст пунктів меню залежить від типу об'єкта (рис. 4.12).



Рис. 4.12. Контекстні меню

а - вікна, б - поля графіка, в - графіка

Статистичну інформацію щодо властивостей графіка виводить пункт *Data Statistics* меню *Tools*. Вікно (рис. 4.13) містить дані про діапазон значень, середнє, медіану тощо.



	X		Y	
min	1	<input type="checkbox"/>	-0.7568	<input type="checkbox"/>
max	4	<input type="checkbox"/>	0.9093	<input type="checkbox"/>
mean	2.5	<input type="checkbox"/>	0.2838	<input type="checkbox"/>
median	2.5	<input type="checkbox"/>	0.4913	<input type="checkbox"/>
mode	1	<input type="checkbox"/>	-0.7568	<input type="checkbox"/>
std	1.291	<input type="checkbox"/>	0.7758	<input type="checkbox"/>
range	3		1.666	

Рис. 4.13. Вікно статистики

Інтерактивно керувати кількістю графіків можна за допомогою браузера друку. Простим клацанням миші можна прибрати з поля графіки або кнопкою *Add Data* додати.

4.4.4. Контрольні питання

Куди виводяться графіки *Matlab*?

Перерахуйте основні типи графіків *Matlab*.

Яка функція вважається основною для виведення графіків?

В чому полягає різниця функцій **plot** та **plotyy**?

Як вивести кілька графіків в одне вікно?

Назвіть засоби оформлення графіків.

Як вивести кілька графічних полів в одне вікно?

Яку інформацію рекомендується розташовувати в графічному вікні?

Назвіть складові графічного вікна.

Назвіть можливості редагування графічного вікна.

4.5. Тримірні графіки

Для візуалізації багатовимірних моделей, аналізу матричних даних, функцій двох змінних в *Matlabi* використовуються засоби тримірної графіки. Базові типи тримірних зображень наведено в таблиці 4.6.

Таблиця 4.6. Базові види просторових зображень

Вигляд	Тип	Вигляд	Тип
	Зображення матриці у вигляді поверхні		Зображення матриці у вигляді каркасної поверхні
	Вид згори матриці у вигляді поверхні з ізоліній		Зображення матриці у вигляді поверхні з сіткою по ізолініях
	Зображення просторових кривих		Зображення матриці у вигляді поверхні з сіткою по рядках чи стовпцях

Положення точки в просторі визначається трьома координатами. Наприклад, в декартовій системі координат - x , y , z . Сукупність точок, які належать до просторової кривої визначається трьома векторами, які містять координати точок по відповідних координатах. Тримірні поверхні описуються функцією двох змінних та потребують визначення двомірної сітки, в вузлах якої буде розраховуватися значення функції. Для опису такої сітки в *Matlab* використовуються дві матриці. Кількість рядків цих матриць дорівнює кількості вузлів розбиття відрізка $[y_{\min} y_{\max}]$, а кількість стовпців - кількості вузлів розбиття відрізка $[x_{\min} x_{\max}]$. Рядки матриці X складаються з однакових рядків, що містять координати вузлів на відрізку $[x_{\min} x_{\max}]$, а матриця Y - з однакових стовпців з координат вузлів відрізка $[y_{\min} y_{\max}]$.

Для перетворення векторів x , y в матриці X , Y призначена функція $[X, Y] = \text{meshgrid}(x, y)$.

Рядки матриці X є копіями вектора x , стовпці матриці Y є копіями вектора y .

4.5.1. Графіки просторових кривих

Для побудови графіків просторових кривих слугує функція

`plot3(x,y,z,<linestyle>)`.

Аргументами функції є вектори $\mathbf{x}, \mathbf{y}, \mathbf{z}$, елементами яких є координати точок кривої. Зазвичай вектори містять результати розрахунку параметричного опису кривої $x=f(t)$, $y=g(t)$, $z=s(t)$. Можна замість трьох векторів визначити матрицю, в трьох стовпцях якої розмістити вектори $\mathbf{x}, \mathbf{y}, \mathbf{z}$. Опціональний параметр **linestyle** визначає тип лінії кривої аналогічно функції **plot**.

Примітка. Функцію **plot3** можна використовувати і для отримання зображення функції двох змінних у вигляді сітчастої поверхні. При цьому в якості аргументів мають бути застосовані матриці сітки X, Y та значень функції Z . Малюються лінії паралельно площині YOZ . Кольори ліній змінюються послідовно відповідно списку кольорів команди **plot**.

4.5.2. Графіки просторових поверхонь

Для створення зображень поверхонь призначені функції **mesh**, **meshc**, **meshz** та **surf**, **surfz**, **surfl**. Функції **mesh** будують зображення сітчастої поверхні, функції **surf** - поверхні у вигляді оболонки. Колір точки поверхні змінюється відповідно до значення функції в цій точці.

Синтаксис всіх функцій однаковий:

mesh (X, Y, Z, <C>) .

Аргументами функцій є дві матриці \mathbf{X}, \mathbf{Y} сітки та матриця \mathbf{Z} значень функції. Опціональний параметр C визначає тип фарбування поверхні. За замовчанням $C=Z$. Параметр повинен бути матрицею з розміром, який дорівнює розміру матриць інших аргументів.

Функція **mesh** малює сітчасту поверхню з ребрами по двох координатах. Ребра розфарбовуються відповідно значенню функції.

Функція **meshz** додатково заштриховує поверхню в площині ZY для першого значення x_{\min} та останнього – x_{\max} .

Функція **meshc** додатково малює карту ізоліній під зображенням поверхні.

Функція **surf** малює оболонку-поверхню з ребрами та гранями по двох координатах. Грані розфарбовуються відповідно значенню функції.

Функція **surf1** додає джерело освітлення для фотореалістичного ефекту.

Функція **surfс** додатково малює карту ізоліній під зображенням поверхні.

Контурні графіки

Для кількісного аналізу графіків поверхонь корисно мати наочну інформацію про розподіл значень функції. Таку інформацію містять контурні графіки. На контурних графіках зображуються лінії, в яких функція має однакове значення – ізолінії.

В *Matlab* контурні графіки реалізовані функціями типу **contourX**.

Функція **[c h]=contour(X,Y,Z,<N>,<linestyle>)** зображує двомірну карту ізоліній поверхні. Вона є видом згори на поверхню Z у вузлах сітки X, Y . При виклику без опійного параметра N система сама визначає кількість ізоліній. Застосування параметра натуральним числом визначає графік з кількістю ізоліній N . Застосування параметра вектора визначає значення, для яких будуть намальовані ізолінії. Кількість ізоліній буде дорівнювати **length(N)**. Якщо визначити вектор як $[v \ v]$, то буде намальовано одну ізолінію для значення v . Опціональний параметр **linestyle** визначає колір, тип ізоліній аналогічно функції **plot**.

Для того, щоб додати на графік кількісні значення ізоліній слід викликати функцію **clabel(c,h)** з результатами виконання функції **contour** з матрицею **c** та вектором **h** в якості аргументів.

Функція **[c h]=contourf(X,Y,Z,<N>,<linestyle>)** малює контурний графік та зафарбовує зони між ізолініями відповідним кольором.

Функція **[c h]=contour3(X,Y,Z,<N>,<linestyle>)** малює аксонометричне зображення ізоліній. Значення Z відповідає висоті, на якій зображується ізолінія.

4.5.3. Оформлення тримірних графіків

Палітра фарбування граней поверхні визначається функцією **colormap('mode')**. Параметр-рядок **'mode'** визначає передумовлені палітри:

- з плавною зміною кольорів: **'hsv'** - червоний – жовтий – зелений – синій – червоний, **'hot'** - чорний – червоний – жовтий – білий, **'gray'** - відтінки сірого, **'bone'** - відтінки синьо-сірого, **'copper'** - відтінки мідного, **'pink'** - відтінки рожевого, **'jet'** - від синього до червоного, **'cool'** - від фіолетового до рожевого, **'autumn'** - від червоного до жовтого, **'spring'** - від рожевого до жовтого, **'winter'** - від синього до зеленого, **'summer'** - від зеленого до жовтого.
- з різкою зміною кольорів: **'flag'** - червоний, білий, синій, чорний, **'lines'** - відповідно списку кольорів ліній в **plot**, **'vga'** - Windows 16 кольорів, **'prism'**, **'white'** – білий.

Функція **colobar** виводить стовпець з поточною палітрою у вікно. Опціонально в якості параметра можна визначити положення палітри **'North'**, **'South'**, **'East'**, **'West'**, **'NorthOutside'**, **'SouthOutside'**, **'EastOutside'**, **'WestOutside'** або вимкнути зображення палітри: **'off'**, **'hide'**.

Для сітчастих поверхонь (**plot3**, **mesh**) можна командою **hidden on** вимкнути невидимі ребра, тобто з каркасної моделі зробити непрозору оболонку. Команда **hidden off** повертає каркасну модель.

Виглядом видимих ребер керує команда **shading mode**. Значення параметра **'interp'** прибирає ребра та проводить плавне фарбування, значення **'flat'** прибирає ребра та рівномірно фарбує грані, значення **'faceted'** виводить ребра та рівномірно фарбує грані. Є значенням за замовчанням.

Для оформлення графіків поверхонь також можна застосувати функції **text**, **xlabel**, **ylabel**, **zlabel**, **title**, **axis**, **subplot**.

Тримірні графіки зображуються в аксонометричній проекції. Вигляд поверхні залежить від взаємної орієнтації поверхні та точки зору спостерігача. В *Matlab* точка зору визначається кутами азимута Az та піднесення Ei (рис. 4.14). Кут азимута рахується від осі U в площині XOY проти годинникової стрілки. Кут піднесення – кут між лінією, яка поєднує точку зору з початком системи координат, та площиною XOY .

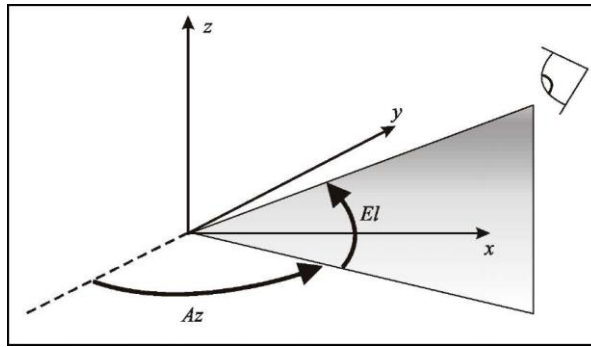


Рис. 4.14. Визначення точки зору

За замовчанням графіки поверхні зображуються для $Az = -37,5^\circ$, $El=30^\circ$. Ці значення в командному вікні можна змінити функцією

view([Az El]).

Змінити точку зору зручніше інтерактивно мишею в графічному вікні. Режим просторового повороту вмикається пунктом *Rotate 3D* меню *Tools*, кнопкою *Rotate 3D* кнопочної панелі вікна. При переміщенні миші з натиснутою лівою кнопкою зображення буде синхронно повертатися. В лівому куті вікна при цьому динамічно відображаються значення кутів орієнтації точки зору.

Графічне вікно забезпечує ще одну потужну можливість налаштування вигляду просторових поверхонь – це засоби кнопочної панелі 3D перегляду. Зображення у вікні імітується роботою фотографічної камери. В панелі можна керувати режимом відображення об'єкта. В *Matlab* використано два режими: ортографічний режим, в якому будується паралельна проекція, та перспективний режим, в якому будується перспективна проекція.

Функції 3D перегляду: *Orbit Camera* – обертанням камери навколо зображення. Дія є аналогом опції *Rotate 3D*, *Orbit Scene Light* – орієнтацією джерела світла відносно об'єкта, *Pan/Tilt Camera* – кутовий зсув об'єкта відносно камери, *Move Camera Horizontally/Vertically* – переміщення об'єкта в полі зору камери, *Move Camera Forward/Back* – зміна відстані між камерою та об'єктом, *Zoom Camera* – зміна збільшення камери, *Roll Camera* - поворот камери навколо своєї осі.

4.5.4. Контрольні питання

Назвіть види тримірних графіків *Matlab*.

Як створюється сітка для побудови тримірних графіків?

Якими способами визначаються вектори для побудови графіка кривих?

Що є аргументами функцій побудови графіків поверхонь?

Для чого призначені контурні графіки?

Як задається палітра фарбування графіків поверхонь?

В якій проекції зображуються графіки поверхонь?

Як змінити точку зору?

4.6. Символьні обчислення

В базовій комплектації *Matlab* не може виконувати символьні обчислення. Для проведення символьних перетворень потрібна інсталяція розширення *Symbolic Math Toolbox*, який призначений для проведення перетворення виразів, аналітичного розв'язання задач лінійної алгебри, диференціальних та інтегральних обчислень, отримання чисельного результату з підвищеною точністю. *Symbolic Math Toolbox* реалізує інтерфейс між *Matlab* та символьним ядром. В попередніх версіях *Matlab* використовувалося ядро найпотужнішої CAS *Maple*. В останніх версіях *Matlab* використовуються можливості CAS *MuPAD*, яку розробники *Matlab* придбали та вбудували в *Matlab*. Запустити *MuPAD* можна командою **mupad**.

4.6.1. Символьні змінні та функції

Символьні змінні є об'єктами класу *sym object*. Перед тим, як проводити символьні перетворення змінні та вирази слід об'явити як символьні. Для визначення одної змінної чи виразу застосовується функція **sym('x', <'type'>)**. Аргументом є вираз/ім'я змінної, забраний в апосторфи. Опціональний параметр <'type'> визначає тип змінної: **'real'** – дійсна змінна, **'positive'** – додатна дійсна змінна, **'unreal'** – комплексна змінна, **'clear'** – скидання типу до передумовленого.

Для визначення кількох змінних одночасно застосовується функція **syms x y, <' type' >**. Вирази, в яких використані символічні змінні, теж є символічними.

Наприклад, запис

```
>> x=sym('x','real'); syms a b;
>> c=sym('(sin(x)+a)^2*(cos(x)+b)^2');
>> f = (sin(x)+a)^2*(cos(x)+ b)^2/sqrt(abs(a+b))
```

в першому рядку визначає символічну змінну «x» як дійсну та дві символічні змінні «a», «b» без визначення типу. В другому рядку визначено змінну «c» як символічний вираз. В третьому рядку визначено символічний вираз «f» простим поєднанням арифметичними діями символічних змінних «x», «a», «b».

```
» syms a b real ; p = conj (a + i*b)   p =a - i*b
» syms a b unreal; q = conj (a + i*b)   q = conj(a + i*b)
```

В першому рядку змінні визначені як дійсні, тому змінна «p» розраховується як комплексно пов'язана. В другому рядку змінні визначені як комплексні, тому дія комплексного пов'язання не виконується. Змінна «q» є символічним виразом.

Для отримання запису виразу в майже «природньому» вигляді призначена функція **pretty(ім'я виразу)**.

```
» pretty(f)
(sin(x)+a)2(cos(x)+b)2
|a + b|1/2
```

4.6.2. Обчислення з символічними змінними

Символічні змінні можуть бути елементами матриць та векторів. В результаті створюються символічні матриці та вектори, до яких можна застосовувати матричні та поелементні операції.

```
» syms a b c d e f g h; A = [a b; c d]
» B = [e, f; g, h] ;
```

$$\gg C = A * B ; C = \begin{bmatrix} a * e + b * g & a * f + b * h \\ c * e + d * g & c * f + d * h \end{bmatrix}$$

$$\gg F = A .* B \quad F = \begin{bmatrix} a * e & b * f \\ g * c & d * h \end{bmatrix}$$

Символьний процесор дозволяє знаходити чисельне значення з підвищеною точністю. Для отримання чисельних значень символьних виразів призначена функція **vpa(S,D)**, де **S** - символьний вираз, **D** - опціональне значення кількості цифр у відповіді. За замовчанням виводиться 32 цифри.

Наприклад,

```
\gg c=sym('exp(1)'); vpa(c)  
2.71828182845904523536028747135
```

Параметр **D** функції **vpa** діє тільки для поточного виклику функції. Для глобального визначення кількості цифр слід використовувати функцію **digits(D)**. Виклик функції **digits** без аргументу повертає поточне значення **D**.

Більшість чисельних перетворень над матрицями можуть бути проведені в символьному вигляді також. Імена функцій символьних перетворень матриць співпадають з іменами функцій для чисельних дій: **det** – визначення детермінанта матриці, **inv** – знаходження зворотної матриці, **poly** – знаходження характеристичного полінома.

Наприклад,

```
\gg A = sym('[a b c; d e f; g h j ] ');  
\gg det(A)      j*a*e-a*f*h-j*d*b+d*c*h+g*b*f-g*c*e  
\gg inv(A)  
[(f*h-e*j)/(a*f*h-b*f*g-c*d*h+c*e*g-a*e*j+b*d*j), -(c*h-...  
b*j)/(a*f*h-b*f*g-c*d*h+c*e*g-a*e*j+b*d*j), -(b*f-c*e)/...  
(a*f*h-b*f*g-c*d*h+c*e*g-a*e*j + b*d*j)]  
[-(f*g-d*j)/(a*f*h-b*f*g-c*d*h+c*e*g-a*e*j+b*d*j), (c*g-...  
a*j)/(a*f*h-b*f*g-c*d*h+c*e*g-a*e*j+b*d*j), (a*f-c*d)/...  
(a*f*h - b*f*g - c*d*h + c*e*g - a*e*j + b*d*j)]  
[-(d*h-e*g)/(a*f*h-b*f*g-c*d*h+c*e*g-a*e*j+b*d*j), (a*h-...  
b*g)/(a*f*h-b*f*g-c*d*h+c*e*g-a*e*j+b*d*j), -(a*e-b*d)...  
/(a*f*h - b*f*g - c*d*h + c*e*g - a*e*j + b*d*j)]
```

» `poly(A)`

$x^3 + (-a - e - j) * x^2 + (a * e - b * d - c * g - f * h + j * (a + e)) * x + (a + e) * (c * g + f * h) - g * (a * c + b * f) - j * (a * e - b * d) - h * (c * d + e * f)$

Символьні вирази обраховуються максимально точно, на відміну від чисельної математики. Кількісні значення символьного та чисельного розрахунків відображаються на екрані по різному. Символьні результати вирівнюються по лівому краю, чисельні – зображуються з відступом.

Наприклад, обчислення виразу $\frac{a}{x} + c$ для $a=2, x=3, c=2$

СИМВОЛЬНО

```
>> a=sym(2); x=sym(3); c=sym(2);
```

```
>> a/x+sqrt(c) ans = 2^(1/2) + 2/3
```

чисельно

```
>> a=2; x=3; c=2; a/x+sqrt(c) ans = 2.0809
```

Перетворення виразів

Складні математичні вирази можуть бути приведені до еквівалентних шляхом спрощення та перетворення.

Для групування символьних поліномів «р» по ступенях визначеної змінної призначена функція `collect(p, <'a'>)`. За замовчанням змінною вважається «х». Для проведення дії з іншою змінною, її ім'я в апострофах слід поставити другим аргументом функції.

Наприклад,

```
>>p=sym(' (x+a)^4+(x-1)^3-(x-a)^2-a*x+x-3 ');
```

```
>>pretty(collect(p))
```

```
x^4+(1+4a)x^3+(6a-4)^2x^2+(4+a+4a)^3x+a^4 - 4a^2
```

```
>>pretty(collect(p,'a'))
```

```
a^4+4xa^3+(-1+6x2)a^2+(4x3+x)a+x^4+(x-1)^3-x^2-3+x
```

При виклику функції без опціонального параметра за замовчанням групує поліном за зниженням ступеню «х» від 4 до 0. Виклик цієї функції з

додатковим параметром «а» проводить групування та ранжування доданків відносно вказаної змінної «а».

Розкрити дужки та обраховувати значення символьних функцій в виразах можна за допомогою функції **expand(p)** :

» **pretty(expand(p))**

$$x^4+4ax^3+6x^2a^2+4xa^3+a^4+x^3-4x^2+4x-4+ax-a^2$$

Розкладається не тільки алгебраїчний поліном, а і тригонометричні, експоненційні, логарифмічні функції. Наприклад,

»**pretty(expand(sym('sin(arccos(3*x))+exp(2*log(x))')))**

$$(1 - 9x^2)^{1/2}+x^2$$

Для визначення множників символьного полінома «р», якщо множники мають раціональні коефіцієнти, слугує функція **factor(p)**:

»**pretty(factor(sym('x^5+13*x^4+215/4*x^3+275/4*x^2- . . . 27/2*x-18')))**

$$1/4 (2x + 1) (2x - 1) (x + 6) (x + 4) (x + 3)$$

Якщо вираз є натуральним числом, то функція проводить його розкладання на добуток простих чисел:

»**factor(sym('230010'))** (2)*(3)*(5)*(11)*(17)*(41)

Спрощення символьних виразів проводиться функцією **simplify(f,<n>,<Seconds>,<T>)**. Функція спрощує вирази, які вміщують тригонометричні, експоненціальні, логарифмічні, спеціальні: гіпергеометричну, Беселя, гамма-функцію. Опціональний параметр **<n>** визначає максимальну кількість кроків спрощення. За замовчанням дорівнює 50. Другий опціональний параметр визначає максимальний час на дію функції.

Наприклад,

>> syms alpa beta

>>simplify(exp(c*log(sqrt(alpa+beta))))

ans=(alpa+beta)^(c/2)

Подивитись результати застосування різних способів спрощення виразу дозволяє функція **simple**. Виклик з прирівнюванням ховає проміжні результати.

Наприклад,

```
» z=sym('cos(x)^2+sin(x)^2)^(1/2); simple(z^2)
simplify: 1
radsimp: cos(k)^2+sin(k)^2
combine(trig): 1
factor: cos(k)^2+sin(k)^2 expand: cos(k)^2+sin(k)^2
combine: 1
convert(exp):
(1/2*exp(i*k)+1/2/exp(i*k))^2-1/4*(exp(i*k)-1/exp(i*k))^2
convert(sincos): cos(k)^2+sin(k)^2 convert(tan):
(1-tan(1/2*k)^2)^2/(1+tan(1/2*k)^2)^2+4*tan(1/2*k)^2/...
(1+tan(1/2*k)^2)^2
collect(k): cos(k)^2+sin(k)^2
ans = 1
>> c=simple(z^2)
c =1
```

Для спрощення громіздких виразів, розрахунку символьних виразів при визначених значеннях обраної змінної потрібно проводити підстановку в вираз. Функція **subs(f,<OLD>,<NEW>)** проводить заміну в символьному виразі **f** змінної **<OLD>** на змінну **<NEW>**.

Наприклад, підстановка для отримання чисельного результату:

```
>> f = sym('exp(x^3 + 2*x^2 + x + 5)');
>> subs(f,'x',1.3) ans = 1.4392e+005
```

послідовна підстановка двох змінних:

```
>> f=sym('cos(a)+sin(b)');
>> f1=subs(f,'a','x^2') f1 =cos(x^2) + sin(b)
>> subs(f1,'b',pi/3) ans = cos(x^2) + 3^(1/2)/2
```

підстановка двох змінних із застосуванням комірок:

```
>>subs(f,{'a','b'},{'x^2',pi/3}) ans=cos(x^2)+ 3^(1/2)/2
```

Представити функцію у вигляді розкладання в ряд Тейлора дозволяє функція `taylor(f, <n>, <a>, <x>)`. За замовчанням розраховується шість членів розкладання. Опціональний параметр `<n>` дозволяє визначити нову кількість. Без наявності третього параметра `<a>` проводиться розкладання в ряд Маклорена, наявність параметра призводить до розкладання в ряд Тейлора навколо точки `<a>`. Опціональний параметр `<x>` застосовується для функцій кількох змінних та визначає змінну, по якій проводиться розкладання.

```
>>syms x;taylor(exp(-x))-x^5/120+x^4/24-x^3/6+x^2/2-x+1
>>taylor(exp(-x),3)          x^2/2 - x + 1
>>taylor(exp(-x),3,1)      1/exp(1) - (x-1)/exp(1) + (x-1)^2 ...
/ (2*exp(1))
```

Примітка. В склад *Symbolic Math Toolbox* входить функція `taylortool` з віконним графічним інтерфейсом для наочної демонстрації дії функції `taylor`.

Знаходження символічних значень для сум виконується функцією `symsum(z, <k>, <kmin>, <kmax>)`. Добутків - `symprod(z, <k>, <kmin>, <kmax>)`. Перший параметр – символічний вираз доданка/множника, другий – ім'я індексу, третій та четвертий – нижня та верхня межі. Нескінченості визначаються вбудованими змінними `Inf`, `-Inf`.

Наприклад, значення суми $\sum_{k=1}^{\infty} \frac{(-1)^k}{k^2}$:

```
» syms k
```

```
» symsum('(-1)^k/k^2', k, 1, Inf) ans=-1/12*p1^2
```

значення добутку $\prod_{k=1}^{\infty} \frac{(-1)^k}{k^2}$:

```
» symprod('(-1)^k/k^2', k, 1, Inf) ans=0
```

Ряд функцій *Symbolic Math Toolbox* призначено для розв'язання задач диференціального та інтегрального обчислення. Границі функцій та послідовностей знаходить функція `limit(F, <x>, <a>, <'mode'>)`. За замовчанням функція шукає границю символічної послідовності `F(x)`, в точці `x=0`. Опціональний параметр `<x>` явно визначає змінну, опціональний

параметр **<a>** визначає точку, де шукати границю, опціональний параметр **'mode'** може мати значення **'left'** або **'right'**.

Наприклад,

```
» syms x
» limit((10 + x)^(1/x), x, 0)           ans = NaN
» limit((10 + x)^(1/x), x, 0, 'left')  ans = 0
» limit((10 + x)^(1/x), x, 0, 'right') ans =inf
```

Обчислення похідних проводиться функцією **diff(f, <x>, <n>)**. Символьна функція записується в першому аргументі. За замовчанням обчислюється перша похідна для змінної з іменем в зворотному порядку алфавіту (z, y, x, ...). Опціональний аргумент **<x>** визначає змінну, для якої шукається похідна, опціональний аргумент **<n>** визначає порядок похідної.

Наприклад,

```
>>syms x;
>>diff(sin(x^2))  ans=2*x*cos(x^2)
>>diff(x^6,6)    ans=720
```

При застосуванні функції диференціювання, похідна обраховується для кожного елемента матриці.

Наприклад,

```
» syms a x
A = [cos(a*x), sin(a*x); -sin(a*x), cos(a*x)];
» diff(A)
ans=[-sin(a*x)*a, cos(a*x)*a] [-cos(a*x)*a, -sin(a*x)*a]
```

Symbolic Math Toolbox працює як з невизначеними, так із визначеними інтегралами.

Символьне інтегрування виконує функція **int(f, <x>, <a>, , <mode>, <val>)**.

За замовчанням шукається невизначений інтеграл від функції **f(x)**. Визначені інтеграли потребують визначення нижньої **<a>** та верхньої **** меж інтегрування. Опціональний параметр **<x>** визначає змінну

інтегрування. Опціональна пара параметрів `<mode>`, `<val>` застосовується в спеціальних випадках. `'IgnoreAnalyticConstraints'` контролює рівень суворості для обмежень типу розривів функції, ділення на нуль, тощо. Значення `TRUE` послаблює перевірки. За замовчанням – `FALSE`. `'IgnoreSpecialCases'` контролює, як відноситися до спеціальних значень функції. `TRUE` ігнорує особливі випадки. За умовчанням – `FALSE`. `'PrincipalValue'`, використовується для запиту головного значення визначеного інтеграла.

Кратні інтеграли обчислюються повторним викликом функції `int`.

Наприклад,

```
>> syms k x; int(sin(x*k)) ans = -cos(k*x)/k
>> int(sin(x*k), k) ans = -cos(k*x)/x
>> int(sin(x*k), k, 0, pi/3) ans = (2*sin((pi*x)/6)^2)/x
>>int(asin(sin(x)), x, 0, 5) Integral could not be found.
>>int(asin(sin(x)), x, 0, 5, 'IgnoreAnalyticConstraints', true)
ans=25/2
>>int(x^t, x)
ans=piecewise([t=-1, log(x)], [t<>-1, x^(t+1)/(t+1)])
>>int(x^t, x, 'IgnoreSpecialCases', true)
ans=x^(t + 1)/(t + 1)
>>int(1/x, -1, 1) ans=NaN
>>int(1/x, -1, 1, 'PrincipalValue', true) ans=0
```

В разі, якщо результат розрахунку визначеного інтеграла потрібно отримати в чисельному вигляді, слід натиснути на відповіді.

Наприклад, обчислення інтеграла $\int_1^5 \frac{x dx}{1+x^2}$

```
» syms x; int('x/(1+x^2)', 1, 5) ans= log(13)/2
ans =1.2825
```

4.6.3. Контрольні питання

Що потрібно для увімкнення можливості символьних перетворень?

Який тип мають символні змінні? Як вони визначаються?

Що таке «природній» вигляд виразу?

Які символні дії проводяться для перетворення виразів?

Для чого застосовується функція **subs**?

Які задачі інтегрального та диференціального обчислення символно вирішує *Matlab*?

5. ПРАКТИКУМ ДО РОЗДІЛУ 4

5.1. Розрахунки в режимі калькулятора

Завдання 5.1.1.

а) Обчисліть заданий арифметичний вираз, відформатуйте відповідь. Порівняйте отриманий результат з наведеним [15] .

$\frac{(6.6 - 3\frac{3}{14}) \cdot 5\frac{5}{6}}{(21 - 1,25) : 2,5}$	$2,5 \quad \frac{2,625 - \frac{2}{3} \cdot 2\frac{5}{14}}{(3\frac{1}{12} + 4,375) : 19\frac{8}{9}}$	$2,8095$
--	---	----------

б) Обчисліть наведені вирази при заданих значеннях змінних, відформатуйте відповідь. Порівняйте отриманий результат з наведеним.

$\frac{\sqrt[3]{\sin(a) + b}}{\lg^2(b)}$	$a=3,4^0 \quad b=0.89$	383.7039
$\frac{b^3 + \sin^2(a)}{\arccos(b^2) + e^{-\frac{b}{2}}}$	$a=83^0 \quad b=0.12$	$239/605$

Завдання 5.1.2. Задайте значення змінним $x=0.5$, $y=0.1$ та розрахуйте значення змінної z з використанням «**inline**» функцій:

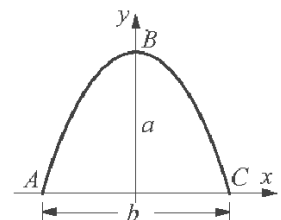
а)
$$z = \arctg\left(\frac{\sqrt[3]{x - \sin(y)}}{\sqrt{1 - x^2}} - \frac{|x|\sqrt{1 - x^2}}{\sqrt[3]{x - \sin(y)}}\right)$$
 б)
$$z = \frac{\sqrt{\cos(2y) + \sin(4y) + \sqrt{e^x + e^{-x}}}}{(e^x + e^{-x})^3 \cdot (\cos(2y) + \sin(4y) - 2)^2}$$

в)
$$z = \sinh\left(\frac{(x + \ln(y))^3}{\sqrt{|x - \ln(y)|}}\right) \cdot \cosh(x + \ln(y) \cdot \sqrt{|x - \ln(y)|})$$

Завдання 5.1.3. Довжина дуги ABC еліпсу з напівосями a , b визначається як

$$L_{ABC} = \frac{\sqrt{b^2 + 16a^2}}{2} + \frac{b^2}{8a} \ln\left(\frac{4a + \sqrt{b^2 + 16a^2}}{b}\right).$$

Розрахуйте довжину з використанням анонімної функції для $a=11$, $b=9$.



Завдання 5.1.4 Введіть довільну матрицю розміром (3*3) з одним найменшим елементом:

- а) знайдіть суму найбільших елементів її рядків;
- б) знайдіть суму елементів рядка, в якому розташовано елемент з найменшим значенням;
- в) знайдіть зворотну матрицю. Покажіть, що знайдена матриця є зворотною;
- г) введіть дві матриці розміром (2*2). Побудуйте нову матрицю розміром (2*4) так, щоб перші 2 стовпця були рядками першої матриці, а в інші - стовпцями другої матриці.

Завдання 5.1.5. Визначити, в скільки разів змінюється час розрахунку при розрахунку виразу $\sqrt{|\sin^{0.89}(\sqrt[3]{x})|}$ з використанням «векторизації» та при доступі до елементів векторів через індекс. Діапазон зміни x : від $1/e$ до $10000/e$ з кроком $1/e$.

5.2. Програмування

Завдання 5.2.1. Корпус бака складається з циліндру діаметром 24 м, висотою 10 м та конуса висотою 4 м, верхнім діаметром 46 м. Поплавок показує рівень рідини. Написати скрипт-файл, який вводить рівень рідини в метрах, обчислює об'єм рідини в кубічних метрах, виводить результат: «Об'єм рідини=XXX кубічних метрів». Передбачити перевірку введення значення рівня. В разі помилки вивести повідомлення червоним кольором.

Для $h \leq 10$ м об'єм визначається як $V = \pi \cdot 12^2 \cdot h$. Для $10 < h \leq 14$ об'єм визначається як $V = \pi \cdot 12^2 \cdot 10 + \pi \cdot (h-10) \cdot (12^2 + 12r_h + r_h^2)/3$, де

$$r_h = 12 + (46-24) \cdot (h-10)/(4 \cdot 2);$$

Завдання 5.2.2. Розробити М-функцію перерахунку координат X_0, Y_0 в координати X_r, Y_r при повороті системи координат на кут φ :

$$X_r = X_0 \cos(\varphi) - Y_0 \sin(\varphi) \quad Y_r = X_0 \sin(\varphi) + Y_0 \cos(\varphi) .$$

Завдання 5.2.3. Напишіть М-функції, які обчислюють наближене значення згідно наступних виразів. Аргументом функцій є кількість елементів виразу K .

$$\pi = \sqrt{6 \sum_{n=1}^K \frac{1}{n^2}} \quad \frac{2}{\pi} = \frac{\sqrt{2}}{2} \frac{\sqrt{2+\sqrt{2}}}{2} \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \dots$$

Завдання 5.2.4. Розробити М-функцію розрахунку «прямокутника» $\text{myrect}(x)$ (рис. 5.1)

$$\text{myrect}(U, \tau, \tau_0, x) = \begin{cases} U, & \tau_0 - \frac{\tau}{2} < x < \tau_0 + \frac{\tau}{2} \\ 0, & \tau_0 - \frac{\tau}{2} \geq x \geq \tau_0 + \frac{\tau}{2} \end{cases}$$

Вихідні дані: тривалість імпульсу - τ , зсув імпульсу - τ_0 , амплітуда імпульсу - U , точка відліку - x . Результат: значення функції «прямокутника».

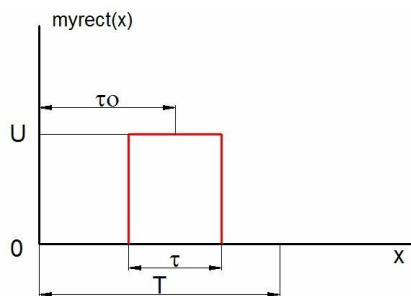


Рис. 5.1. Функція «прямокутника»

Завдання 5.2.5.

а) Розробити М-функцію для розрахунку за формулою Герцбергера:

$$n(\lambda) = A - \frac{B}{\lambda^2 - 0.028} + \frac{C}{(\lambda^2 - 0.028)^2} + D \cdot \lambda^3$$

$A=1.44902$, $B=0.004604$, $C=-0.000381$, $D=-0.0025262$.

б) Розробити скрипт-файл для розрахунку в діапазоні аргументу від 1 до 2 з кроком 0.1 з виведенням на екран результатів без форматування. Використати векторизацію.

в) Розробити скрипт-файл для розрахунку арифметичним циклом в діапазоні від 1 до 2 з кроком 0.1 з виведенням на екран результатів у вигляді таблиці наступного вигляду:

:	x	:	n(x)	:
:	1.0	:	-	:
		...		
:	2.0	:	-	:

г) Розробити скрипт- файл для розрахунку циклом з передумовою в діапазоні від 1 до 2 з кроком 0.1 з виведенням на екран результатів з форматуванням в наступному вигляді:

Аргумент: початок=старт, кінець=стоп, крок=крок, кроків=X

n (1.0) = -

...

n (2.0) =-

Завдання 5.2.6. Розробити файл-функцію розв'язання квадратного рівняння та скрипт-файл для введення коефіцієнтів, виклику розробленої функції та виведення відповіді. Скрипт-файл повинен циклічно запитувати значення коефіцієнтів та завершувати роботу при обранні користувачем пункту «Завершити».

1) Вибір користувача проводить рядку командного вікна:

Дальші дії: Продовжити -1 Завершити -2

2) Вибір користувача проводиться через меню (рис. 5.2).

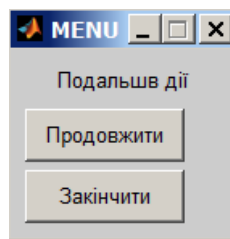


Рис. 5.2. Меню завдання 5.2.6

Завдання 5.2.7. Напишіть скрипт-файли, які вводять матрицю та

а) розраховують суму всіх додатних елементів матриці;

б) замінюють в матриці елементи, які більші за середнє значення на середнє значення;

в) визначають максимальний стовповий та рядковий індекси від'ємних елементів матриці.

Завдання 5.2.8. Напишіть скрипт, який вводить вектор оцінок, розраховує та виводить наступну інформацію:

Середній бал=XX

Оцінка А (100-96) XX студентів (XX%)

Оцінка В (86-95) XX студентів (XX%)

Оцінка С (75-85) XX студентів (XX%)

5.3. Двомірні графіки

Завдання 5.3.1. За допомогою функції **fplot** намалювати графік функції $f(x) = \frac{x^2}{2 + \sin(x) + x^2}$ для $-4 < x < 4$.

Завдання 5.3.2. На одному полі намалювати графіки функції $f(x) = x^3 - 2x^2 - 9\sin^2(x) - e^{0.4x}$ та її похідної в діапазоні $-2 < x < 3$. Графік функції – червоний суцільною лінією, похідної – синьою штриховою. Додати легенду.

Завдання 5.3.3. Намалювати в одному вікні графіки функції $y(x) = 2^{10-0.5x}$ в діапазоні $0.1 < x < 1000$ в лінійному, напівлогарифмічному, логарифмічному масштабі. Графіки підписати. Зобразити сітку.

Завдання 5.3.4. Для значень часу $10 < a < 20$ секунд, відповідних значень функції відстані $y = (920\ 650\ 420\ 335\ 221\ 146)$ метрів та похибки $d = (14\ 20\ 25\ 12\ 10\ 11)$ метрів намалювати графік. Похибка симетрична. Осі підписати.

Завдання 5.3.5. Намалювати в полярних координатах графік функції $r(x) = 2.5\cos^2\left(\frac{x}{3}\right) + x$ для 150 точок в діапазоні $0 < x < 2\pi$.

Завдання 5.3.6. Закон ідеального газу пов'язує тиск P , об'єм V та температуру T : $PV = nRT$, де n – кількість речовини в молях, $R = 8.3245$ Дж/(К·моль).

Побудувати на одному полі ізотермічні графіки залежності тиску в Паскалях від об'єму для одного моля газу при $T = 100, 200, 300, 400$ К. Діапазон зміни об'єму від 1 м^3 до 10 м^3 . Підпишіть графік, осі. Виведіть легенду.

Завдання 5.3.7.

а) Намалювати графік циклоїди $x=12\cos(t)-2\cos(6t)$ $y=12\sin(t)-2\sin(6t)$ для $-0 < t < 4\pi$.

б) Намалювати параметричний графік $x=(3.3-0.4t^2)\sin(t)$

$y=(2.5-0.3t^2)\cos(t)$ для $-2\pi < t < 2\pi$.

Завдання 5.3.8. Розробіть скрипт-файл для побудови графіків коефіцієнта передачі амплітуди та фази електричного ланцюга в напівлогарифмічному масштабі один під другим в одному вікні (рис. 5.3) згідно наступного виразу

$$K(\omega) = \frac{n}{1 + a_1 + a_2 - \omega^2 \frac{\tau_1 \tau_2}{a_1} + j\omega(\tau_1 + \tau_2) + \frac{1}{j\omega \tau_3}}$$

де n - коефіцієнт передачі трансформатора; R_f – опір перетворювача; R_t – опір навантаження; C_t – ємність; L_1 – індуктивність первинної обмотки; L_2 – індуктивність вторинної обмотки трансформатора.

$$a_1 = \frac{R_f}{R_t} \quad a_2 = \frac{L_1}{L_2} \quad \tau_1 = C_t R_f \quad \tau_2 = \frac{L_1}{R_t} \quad \tau_3 = \frac{L_2}{R_f} \quad n = \frac{L_2}{L_1}$$

Вихідні данні задати присвоєнням. Текстові коментарі повинні змінюватися автоматично при зміні вихідних даних. Графіки амплітудної та фазової характеристик намалювати в "напівлогармічному" масштабі.

Для форматування вікна графіків рекомендується застосувати розбиття вікна командою **subplot**. В текстовій частині сховати поле графіка командою **axis**, коментарі вивести командою **text**. Рядок тексту сформувати командою **sprintf**. Аргументи графіків доцільно формувати командою **logspace**.

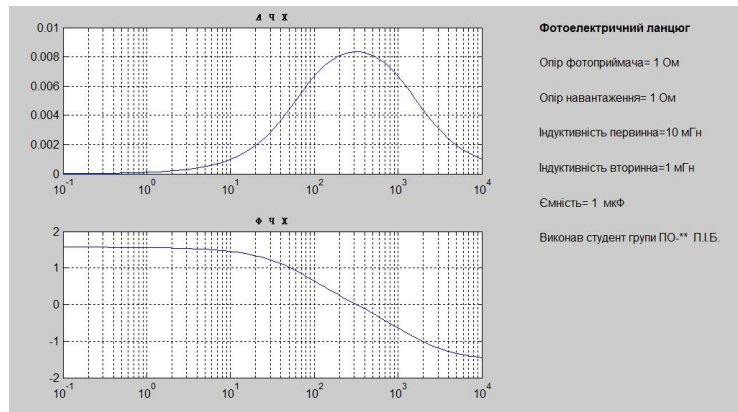


Рис. 5.3. Вигляд вікна завдання 5.3.8

Завдання 5.3.9. Розробити функцію розрахунку $\cos(x)$ за формулою Тейлора
$$csm(x,n) = \sum_{i=0}^n (-1)^i \frac{x^{2i}}{(2i)!}$$
. Для діапазону аргументу $-4 < x < 4$ сформуванати вектори: функцій $\cos(x)$, $csm(x,2)$, $csm(x,3)$, $csm(x,4)$. За допомогою засобів графічного вікна сформуванати зображення відповідно до рис. 5.4 .

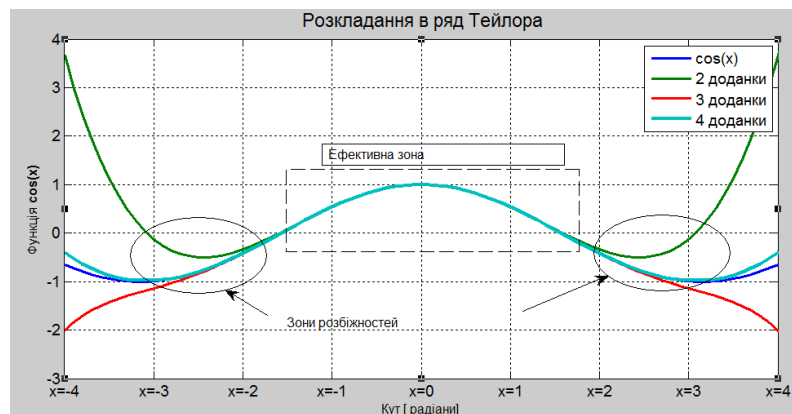


Рис. 5.4. Зображення завдання 5.3.9

5.4. Тримірні графіки

Завдання 5.4.1. Положення рухомого об'єкту визначається як

$$x = \left(\frac{t-15}{100} + 1\right) \sin(3t) \quad y = \left(\frac{t-15}{100} + 1\right) \cos(0.6t) \quad z = 0.3t^{1.6}$$

Зобразити траєкторію руху точки для $0 < t < 30$.

Завдання 5.4.2. Для функції $z(x, y) = (\sin(y^2) + \cos(x^2))^{xy}$ намалювати в одному вікні в діапазоні аргументів від -1 до 1

- а) каркасну поверхню з невидимими лініями,
- б) каркасну поверхню без невидимих ліній,
- в) каркасну поверхню з ізолініями.

Завдання 5.4.3. Для функції $z(x, y) = \frac{y^2}{3} - 1.5\sin(2x)$ намалювати в одному вікні в діапазоні аргументів від -3 до 3

- а) поверхню-оболонку з ребрами,
- б) поверхню без ребер та плавним фарбуванням граней,
- в) поверхню з додатковим джерелом світла.

Завдання 5.4.4. Для функції $z(x, y) = 3y + y^3 - x^3$ намалювати в одному вікні контурні графіки в діапазоні аргументів від -1 до 1

- а) графік з кількістю ізоліній 20,
- б) графік з фарбуванням,
- в) ізолінію $5 = 3y + y^3 - x^3$,
- г) аксонометричний графік ізоліній.

5.5. Символьні перетворення

Завдання 5.5.1. Спростити вираз

- а) $\log(x) - x \log(x) - x$ групуванням відносно $\log(x)$,
- б) $xy + axy + yx^2 - aux^2 + ax$ групуванням відносно x .

Завдання 5.5.2. Розкрити вираз

- а) $\sin(x + y)$ б) $\cos(3 \arccos(x))$.

Завдання 5.5.3. Розкласти вираз на множники

- а) $x^3 + 2$ б) $x^4 - y^4$.

Завдання 5.5.4. Спростити вираз

- а) $(1-x^2) / (1-x)$ б) $\cos(x)^2 + \sin(x)^2$ в) $\cos(2x) - \sin(2x)$.

Завдання 5.5.5. Отримати символічний результат та чисельне значення при $x=3$

а) $A \cdot B$, б) A/B , в) $A+B$, де

$$A=x(x^2+6x+10)+6, \quad B=(x-3)^2+10x-5.$$

Завдання 5.5.6. Розрахуйте

а) e^π та π^e , з точністю до 25 знака. Яке значення більше?

б) $2709/1024$, $10583/4000$, $2024/765$. Який результат ближчий до 1.7?

Завдання 5.5.7. Розрахувати значення сум

$$\text{а) } \sum_1^{10} \frac{1}{k^4}, \quad \text{б) } \sum_1^{\infty} \frac{1}{k^4}$$

Завдання 5.5.8. Розкласти вираз в ряд Тейлора навколо точки c з n доданками

а) e^x , $n=4$, $c=0$ б) $\operatorname{tg}(x)$, $n=6$, $c=0$ в) $\ln(x)$, $n=4$, $c=1$

Завдання 5.5.9. Знайти похідні функцій

$$\text{а) } \frac{2x-1}{x^2+1}, \quad \text{б) } \sqrt{1+x^4}, \quad \text{в) } \operatorname{arctg}(x^2+1)$$

Завдання 5.5.10. Визначити границю

$$\text{а) } \lim_{x \rightarrow \infty} \frac{1-e^{-x}}{x}, \quad \text{б) } \lim_{x \rightarrow 1} \frac{1-x}{\ln(x)}, \quad \text{в) } \lim_{x \rightarrow 1} (1-x)^{\frac{1}{1-x}}, \quad \text{г) } \lim_{x \rightarrow 1} \left(\frac{1}{1-x}\right)$$

Завдання 5.5.11. Знайти інтеграл

$$\text{а) } \int_0^1 x \sin(x) dx, \quad \text{б) } \int_0^1 \frac{x^2 dx}{\sqrt{1-x^2}}, \quad \text{в) } \int_0^{\infty} \frac{x dx}{\sqrt{e^x-1}},$$

$$\text{г) } \int \frac{x dx}{1-x^2}, \quad \text{д) } \int \frac{dx}{\operatorname{sh}(x)\operatorname{ch}(x)}, \quad \text{е) } \int x \cos(x^2) dx$$

Література

1. Дьяконов В. П., Абраменкова И. В., Пеньков А. А. Новые информационные технологии: Учеб. Пособие, Ч. 3: Основы математики и математическое моделирование / Под ред. В.П. Дьяконова; Смол. гос. пед. ун-т. - Смоленск, 2003. - 192 с.
2. Дьяконов В. П. Компьютерная математика. Теория и практика / Владимир Павлович Дьяконов. — М. : Нолидж, 2001. — 1296 с.
3. Кобильник Т. П., Когут У. П. Системи комп'ютерної інформатики у навчанні студентів напрямку «ІНФОРМАТИКА» ISSN Online: 2076-8184. Інформаційні технології і засоби навчання, 2014, Том 40, №2. С.50-64
4. Триус Ю. В. Комп'ютерно-орієнтовані методичні системи навчання математики: монографія / Юрій Васильович Триус. — Черкаси: Брама-Україна, 2005. — 400 с.
5. Жалдак М. И., Горошко Ю. В., Винниченко Е. Ф. Математика с компьютером. Пособие для учителей. – 3-тье изд. – К.: Изд-во НПУ имени М. П. Драгоманова, 2015. – 308 с.
6. Воскобойников Ю. Е. [и др.] Основы вычислений и программирования в пакете MathCAD : учеб. пособие / под ред. Ю. Е. Воскобойникова; Новосиб. гос. архитектур.-строит. ун-т (Сибстрин). – Новосибирск: НГАСУ (Сибстрин), 2012. – 212 с.
7. Половко А. М., Ганичев И. В. Mathcad для студента. – СПб.: БХВ-Петербург, 2006. – 336с.
8. Васильев А. Н. Mathcad 13 на примерах. – СПб.: БХВ-Петербург, 2006. – 528с.
9. Очков В. Ф. MathCAD 14 для студентов, инженеров и конструкторов.- СПб.: БХВ-Петербург, 2007. - 368с.
10. Кирьянов Д. В. Самоучитель MathCAD 13.- СПб.: БХВ-Петербург, 2006. - 528с.
11. Макаров Е. Г. MathCAD: учебный курс.- СПб.: БХВ-Петербург, 2009. - 384с.
12. Глушаков С. В., Жакин И. А., Хачиров Т. С. Математическое моделирование. Учебный курс.-Харьков: Фолио, 2001. - 524с.
13. Струтинський В. Б. Математичне моделювання процесів та систем

механіки: Підручник, Житомир: Ж1П, 2001. - 612 с.

14 Лазарев Ю. Ф. Моделювання на ЕОМ. Навчальний посібник. - К.: Політехніка, 2007. - 290 с.

15. Лазарев Ю. Ф., Початки програмування у середовищі MatLAB: Навч. посібник. - К.: Корнійчук, 1999. - 160 с.

16. Дьяконов В., Круглов В., Математические пакеты расширения MATLAB. Специальный справочник.- СПб.: Питер, 2001. - 480 с.

17. Кетков Ю. Л., Кетков А. Ю., Шульц М. М., MATLAB 7: программирование, численные методы. - СПб.: БХВ-Петербург, 2005. - 752с .

18. Дащенко А. Ф., Кириллов В. Х., Коломиец Л. В., MATLAB в инженерных и научных расчетах. - Одесса: «Астропринт», 2003. - 214 с.

19. Амос Гилат, MATLAB. Теория и практика. 5-е изд. / Пер. с англ. Смоленцев Н. К. – М.: ДМК Пресс, 2016. – 416 с.: ил.

Додаток А. Основні функції Matlab

Алгебраїчні оператори та функції

+	- додавання	(plus (a,b)) ;
-	- віднімання	(minus (a,b)) ;
*	- множення	(times (a,b)) ;
/	- ділення ліворуч	(mrdivide (a,b)) ;
\	- ділення праворуч	(mldivide (a,b)) ;
^	- зведення в ступінь	(power (a,b)) .

Тригонометричні та гіперболічні функції

sin(Z) , **cos(Z)** , **tan(Z)**, **cot(Z)** - тригонометричні;
sinh(Z) , **cosh(Z)**, **tanh(Z)**, **coth(Z)** - гіперболічні;
asin(Z) , **acos(Z)**, **atan(Z)**, **acot(Z)** - зворотні тригонометричні (в радіанах, в діапазоні від $-\pi/2$ до $+\pi/2$);
asinh(Z) , **acosh(Z)**, **atanh(Z)**, **acoth(Z)** - зворотні гіперболічні;
atan2(X,Y) - арктангенс (кут в діапазоні від $-\pi$ до $+\pi$);
sec(Z), **csc(Z)** - секанс/косеканс;
sech(Z), **csch(Z)** - гіперболічний секанс/косеканс;
asec(Z), **acsc(Z)** - арксеканс/косеканс;
asech(Z), **acsch(Z)** - зворотний гіперболічний секанс/ косеканс;

Функції обчислення

exp(Z) - експонента;
log(Z) - натуральний логарифм;
log10(Z) - десятковий логарифм;
sqrt(Z) - квадратний корінь;
abs(Z) - модуль числа.
log2(Z) - логарифм по основі 2;
pow2(Z) - 2 в ступені Z;
sqrt(Z) - квадратний корінь з Z;
nthroot(Z,n) - корінь n-го ступіню з Z;
factorial(Z) - факторіал Z

Функції приведення

fix(Z) - округлення до найближчого цілого в бік нуля;
floor(Z) - округлення до найближчого цілого в від'ємний бік;

ceil(Z) - округлення до найближчого цілого в додатній бік;
round(Z) - звичайне округлення до найближчого цілого;
mod(X,Y) - цілочислене ділення X на Y;
rem(X,Y) - залишок від ділення X на Y;
sign(Z) - знак числа (0 при Z=0, -1 при Z<0, 1 при Z>0).
realmax(‘тип’) - максимальне число «тип»: ‘single’, ‘double’.
realmin(‘тип’) - число «тип»: ‘single’, ‘double’
intmax(‘тип’) - максимальне число цілого типу. (Див. табл. 1.1)
single (X) – приведення аргументу до типу single.
double (X) – приведення аргументу до типу double.
isfloat(X) – перевірка аргументу. Вірно, якщо аргумент дійсний.
isinteger (X) – перевірка аргументу. Вірно, якщо аргумент цілий.

Спеціальні функції

besselj(n, Z) - функція Бесселя першого роду n порядку;
bessely(n, Z) - функція Бесселя другого роду n порядку;
besseli(n, Z) - модифікована функція Бесселя першого роду n порядку;
besselk(n, Z) - модифікована функція Бесселя другого роду n порядку.
beta(Z,W) - бета-функція;
betainc(X,Z,W)- неповна бета-функція;
gamma(Z) - гама-функція;
gammainc(Z,A) - неповна гамма-функція;
gamma1n (Z,A) - логарифм гамма-функція;
legendre(n,X) - узагальнена функція Лежандра;
ellipj(U,M) - еліптична функція Якобі;
ellipke(M) - повний еліптичний інтеграл;
erf(X) - функція похибки (гаусов розподіл);
erfc(X) - додаткова функція похибок;
erfcx(X) - нормована додаткова функція похибок;
erfinv(Y) - зворотна функція похибок.

Інші функції

gcd(A,B) - найбільший загальний дільник;
lcm(A,B) - найменше загальне кратне;
rat(X, tol) - число у вигляді раціонального дробу;
tic - запуск таймера;

toc - зупинка таймера, повернення часу з моменту запуску в секундах;

clock - поточний час. Результат – вектор з 6 чисел: рік, місяць, день, година, хвилина, секунда;

etime(t1, t2) - час між моментами, які завдано векторами t2, t1. Аргументи - вектори формату clock;

cputime - час роботи процесора в мілісекундах зі старту пакета;

fzero(fun, x0, op) - чисельний розв'язок нелінійного рівняння fun. Рівняння записується в нормалізованому вигляді. Способи опису аналогічні функціям оптимізації. Аргументи: x0 – початкова точка, опціональний параметр «ор» - структура умов, аналогічна структурі OPTIMSET в функціях оптимізації. Можливі властивості: *Display, TolX, FunValCheck, OutputFcn, PlotFcns*. Виклик з двома результатами повертає окрім кореню значення функції в точці кореня. Виклик з трьома результатами повертає додатково код закінчення розрахунку. Можливі значення: 1 нормальне завершення, -1 завершення по умовам функції OUTPUT, -3 отримано значення NaN чи Inf, -4 в процесі розрахунку отримано комплексні значення, -5 процес не сходиться, -6 функція не може визначити знак рівняння. Виклик з чотирма результатами додатково повертає структуру, яка вміщує поля OUTPUT.funcCount – кількість ітерацій, OUTPUT.algorithm – кількість ітерацій пошуку інтервалу, OUTPUT.intervaliterations – кількість ітерацій, що вміщують нульове значення всередині OUTPUT.iterations, OUTPUT.message;

QUAD(FUN, A, B, TOL) – обчислення інтегралу функції FUN від A до B з точністю TOL методом парабол. За замовчання точність 1.e-6;

DBLQUAD - обчислення двократного інтегралу. Дія аналогічна дії функції **QUAD**.

Логічні функції

== - перевірка на рівність **(eq(a, b))** ,

~= - перевірка на нерівність **(ne(a, b))** ,

< - менше **(lt(a, b))** ,

> - більше **(gt(a, b))** ,

<= - менше чи дорівнює **(le(a, b))** ,

>= - більше чи дорівнює **(ge(a, b))** ,

& - логічне І **(and(a, b))** ,

| - логічне ЧИ **(or(a, b))** ,

~ - логічне НІ **(not(a, b))** ,

xor(a, b) - логічне виключаюче ЧИ,

any(v) - вірно, якщо всі елементи вектору дорівнюють нулю,

all(v) – вірно, якщо всі елементи вектору НЕ дорівнюють нулю,
bitand(a,b) – побітове логічне І,
bitcmp(a,b) - побітове логічне ЧИ,
bitset(a,N<,v>) – встановлює N-й біт аргументу а в 1(y),
bitshift(a,N) - зсуває на N позицій аргумент а. N>0 еквівалентно множенню, N<0 – діленню,
bitget(a,N) - зчитує N-й біт аргументу,
bitxor(a) - побітове логічне виключаюче ЧИ.

Матричні функції

zeros(M,N) - створення матриці розміром (M*N) з нульовими елементами,
ones(M,N) - створення матриці розміром (M*N) з одиничними елементами,
eye(M,N) - створення одиничної матриці розміром (M*N) ,
rand(M,N) - створення матриці розміром (M*N) з випадкових чисел, які рівномірно розподілені в діапазоні від 0 до 1,
randn(M,N) - створює матрицю розміром (M*N) з випадкових чисел, які розподілені по гаусову закону с нульовим середнім,
fliplr(A) - перестановка стовпців матриці А відносно вертикальної осі,
flipud(A) - перестановка рядків матриці А відносно горизонтальної осі,
rot90(A) - поворот матриці А на 90⁰ проти годинникової стрілки,
tril(A) - формування нижньої трикутної матриці обнуленням елементів вище головної діагоналі,
triu(A) - формування верхньої трикутної матриці обнуленням елементів нижче головної діагоналі,
diag(x, <n>) - формування діагоналі матриці. Якщо x - вектор, то створюється квадратна матриця з вектором x на діагоналі з вказаним номером (діагоналі рахуються від головної n=1 вгору),
zeros(M,N) - формування матриці (zeros(1, N) - вектор-рядок, zeros(N,1) - вектор-стовпець) з нульових елементів,
A' - транспонування матриці,
det(A) - визначник матриці на основі трикутного розкладання методом виключення Гауса,
trace(A) - слід матриці (сума діагональних елементів),
rref(A) - трикутна матриця на основі методу виключення Гауса,
inv(A) - зворотна матриця,

cross(a,b) - векторне множення двох трикомпонентних векторів: вектор, довжина якого дорівнює площі паралелограма з сторонами a , b , перпендикулярний до площини цих векторів

$$a \times b = \begin{vmatrix} i & j & k \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} = i(a_y b_z - a_z b_y) + j(a_x b_z - a_z b_x) + k(a_x b_y - a_y b_x). \text{ Результатом є вектор,}$$

dot(a,b) - скалярне множення двох векторів: добуток довжин векторів на косінус кута між ними

$$c = |a||b| \cos(ab) = a_x b_x + a_y b_y + a_z b_z = a^T \cdot b. \text{ Результатом є скаляр,}$$

size(a) - кількість рядків та стовбців матриці. Результат - вектор $[n, p]$,

max(a), **min(a)** - значення максимального/мінімального елемента вектору. Визначення результату вектором з двох елементів, додає інформацію про індекс елемента з максимальним/мінімальним значенням.

mean(a) - обраховує середнє значення елементів вектору,

sort(a, mode) - впорядкування вектору як зростаючого (**mode** - **'ascend'**) чи зменшуючогося (**mode** - **'descend'**),

sum(a) - сума елементів вектору,

prod(a) - множення елементів вектору,

Функції **size**, **max**, **min**, **sort**, **sum**, **prod** можуть бути застосовані до матриць. В цьому випадку відповідні дії проводяться з кожним із стовпців матриці. Кожен стовпець матриці розглядається як вектор. Функція **sort** сортує елементи кожного стовпця матриці. Результатом є матриця того ж розміру. Функції **sum** та **prod** формують вектор-рядок, кожен елемент якого є сумою чи множенням елементів відповідного стовпця матриці.

roots(P) - вектор, елементи якого є коренями поліному,

poly(r) - вектор коефіцієнтів поліному по заданому вектору його коренів,

eig(A) - повертає вектор особистих значень матриці, тобто коренів характеристичного поліному матриці. Якщо виклик має вид **[R,D]=eig(A)**, то в результатом є діагональна матриця D особистих значень та матриця R правих особистих векторів, які задовольняють умові $A \cdot R = R \cdot D$,

find(X<,K>) - повертає вектор з номерами елементів аргументу - вектора (матриці), які не дорівнюють нулю. Для матриці нумерація

проводиться по стовпцях згори вниз. Виклик **[I,J,V] = FIND(X,...)** повертає вектори I,J номерів стобців та рядків з номерами елементів, які не дорівнюють нулю. Опціональний вектор V є вектором значень аргументу, які не дорівнюють нулю. В якості аргументу може стояти умова. Наприклад, для матриці A запис **find(A > 5)** поверне вектор з номерами елементів, які більші за 5. Опціональний аргумент «K» обмежує виведення «K» елементів, які задовольняють умові,

reshape(A,M,N) – вирізає з матриці A матрицю розміром MxN.