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

> I. В. Кравченко, В. І. Микитенко

# ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Підручник

Затверджено Вченою радою КПІ ім. Ігоря Сікорського як підручник для здобувачів ступеня бакалавра за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології»

Електронне мережне навчальне видання

Київ КПІ ім. Ігоря Сікорського 2022

Рецензенти:	Тягур Володимир Михайлович, доцент, доктор техн. наук, заступник начальника НТК КП СПБ «Арсенал»
	Цісарж Вячеслав Вікторович, канд. техн. наук., заступник директора з науки ДП НДІ «Квант-Радіолокація»
Відповідальний редактор	Колобродов Валентин Георійович, доктор техн. наук, проф.

Гриф надано Вченої радою КПІ ім. Ігоря Сікорського (протокол №8 від 12.12.2022 р.)

Розглянуто історію створення, класифікацію, сучасний стан та термінологію інформаційних технологій, комп'ютерних мереж та хмарних обчислень. Викладено відомості щодо технології розв'язання математичних та технічних задач в системах комп'ютерної математики (СКМ) MathCAD та Matlab, оформлення текстової наукової та технічної документації відповідно до вимог ДСТУ та інших нормативних документів України. Може бути корисний студентам та фахівцям технічних спеціальностей для набуття навичок застосування інформаційних технологій у вигляді офісних систем та СКМ в навчальній та науково-технічній практиці.

Реєстр. № П 22/23-003. Обсяг 24,9 авт. арк.

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» проспект Перемоги, 37, м. Київ, 03056 https://kpi.ua Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

© І. В. Кравченко, В. І. Микитенко 2022

© КПІ ім. Ігоря Сікорського, 2022

ПЕРЕДМОВА	6
1. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ	
1.1. Визначення, склад	
Контрольні запитання	12
1.2. Історія розвитку інформаційних технологій	12
Контрольні запитання	40
1.3. Комп'ютерні мережі	40
Контрольні запитання	50
1.4. «Хмарні» технології	50
Контрольні запитання	74
2. ОФІСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ	75
2.1. Правила розроблення електронних документів	76
Контрольні запитання	89
2.2. Текстовий процесор MS Word	89
Контрольні запитання	117
2.3. Табличний процесор MS Excel	
Контрольні запитання	144
3. СИСТЕМИ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ	145
3.1. Класифікація СКМ	146
3.2. Типова структура СКМ	148
3.3. Історія розвитку СКМ-СКА	149
3.4. Функції СКМ	
3.5. Сучасні СКМ	
Контрольні запитання	
4. CKM MathCAD	167
4.1. Загальні характеристики	167
Контрольні запитання	

# 3MICT

	4.2. Робота з документами MathCAD	183
	Контрольні запитання	189
	4.3. Організація обчислень	190
	Контрольні запитання	211
	4.4. Перевірка умов	211
	Контрольні запитання	216
	4.5. Робота з векторами та матрицями	216
	Контрольні запитання	223
	4.6. Програмування в MathCAD	223
	Контрольні запитання	238
	4.7. Графічне оформлення результатів	238
	4.7.1. Двовимірні графіки	239
	4.7.2. Тривимірні графіки	254
	Контрольні запитання	275
	4.8. Розрахунки з розмірностями	275
	Контрольні запитання	281
	4.9. Символьні обчислення	282
	Контрольні запитання	303
	4.10. Обмін даними	303
	Контрольні запитання	325
5.	. CKM MATLAB	327
	5.1. Загальні характеристики	327
	Контрольні запитання	334
	5.2. Робота в режимі калькулятора	334
	Контрольні запитання	357
	5.3. Програмування в Matlab	358
	Контрольні запитання	373
	5.4. Двовимірні графіки	373

Контрольні запитання	
5.5. Тривимірні графіки	
Контрольні запитання	
5.6. Символьні обчислення	
5.6.1. Засоби СКМ Matlab	
5.6.2. Символьні обчислення в СКА MuPAD	
Контрольні запитання	
5.7. Експорт-імпорт даних	
5.7.1. Програмно керований експорт-імпорт даних	
5.7.2. Засоби інтерактивного файлового обміну	
Контрольні запитання	
ВИКОРИСТАНІ ДЖЕРЕЛА	

#### ПЕРЕДМОВА

Підручник є складовою інформаційно-методичного забезпечення дисципліни «Інформаційні технології в приладобудуванні».

Зважаючи на фаховий напрям підготовки студентів як майбутніх інженерів та обсяг і місце дисципліни в освітній програмі, основна увага приділена матеріалу, необхідному для початку роботи студентами першого курсу з такими засобами інформаційних технологій, як застосунки комп'ютерної математики та офісної автоматизації, які є необхідними для використання в науково-технічних документах. Теоретичні положення комп'ютерних наук викладаються вибірково. Висвітлюються тільки загальні питання комп'юторики в цілому, мережевих інформаційних технологій, «хмарних» обчислень та ефективного використання систем комп'ютерної математики (СКМ) та офісних застосунків.

Перший розділ підручника розкриває базові визначення інформаційних технологій, склад, напрями комп'ютерних наук, термінологію галузі, історії розвитку в світі та в Україні, побудову комп'ютерних мереж, стан та можливості «хмарних» обчислень.

Розділ 2 присвячено застосуванню засобів офісної автоматизації для оформлення текстової документації за вимогами системи конструкторської документації України та оформлення науково-технічних документів відповідно діючим нормативним документам, зокрема ДСТУ 3008:2015 та ДСТУ 8302:2015. Увагу приділено моментам, які є необхідними для ефективної роботи в науково-технічній галузі: використанню стилів, засобам обміну даними, цитуванню, розробленню складених документів тощо, – в середовищі *MS Office*.

В розділі 3 розглядаються запитання історії виникнення, класифікації, типової структури, можливостей СКМ. Наводиться інформація про поширені сучасні СКМ.

В розділах 4-5 детально розглянуті запитання одиночних чисельних та символьних розрахунків, програмування алгоритмів користувача, роботи з матрицями, графічного виведення результатів в СКМ *MathCAD*, *Matlab*. Матеріал забезпечений практичними прикладами.

Розділ 4 охоплює запитання, що стосуються засобів керування, можливостей, особливостей застосування, методики розв'язання

6

математичних та технічних задач чисельно та аналітично, графічного відображення інформації в СКМ *MathCAD*.

Розділ 5 присвячено роботі в СКМ *Matlab* в режимі калькулятора та в програмному режимі.

Приклади в підручнику базуються на версіях програмного забезпечення *MS Office 2013, MathCAD 15, MathCAD Prime 7, Matlab 9X.* Види вікон діалогу та синтаксис команд в інших версіях пакетів можуть дещо відрізнятись від наведених.

Підручник в частині опису конкретних застосунків, зокрема *MS Office* CKM *MathCAD* та CKM *Matlab*, доповнює, розширює та деталізує інформацію посібника авторів з практикуму з дисципліни [1]. Разом видання створюють комплект начально-методичної літератури, яка повністю забезпечує дидактичним матеріалом дисципліну.

Метою видання є допомога студентам в самостійному вивченні навчальної дисципліни, в набутті навичок застосування інформаційних технологій офісних застосунків та сучасних СКМ в навчальній та науковотехнічній практиці. Підручник може використовуватись для самостійної роботи та дистанційного навчання.

## 1. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Інформатизація стала провідною тенденцією в переважній більшості галузей сучасної фундаментальної теоретичної та прикладної науки, виробництва, комерції, громадського життя. Даний процес є закономірним наслідком розвитку свідомості суспільства, зростання інформаційних потоків, вимог пришвидшених темпів росту виробництва, виникнення нових наук тощо. Підґрунтя для успіхів інформатизації складає розвиток математики та комп'ютерної техніки.

Математика та інформатика, з одного боку, спираються на розумову діяльність людини, використовують абстракті символічні мови. З іншого боку математика є універсальним неперевершеним інструментом для опису, вивчення прогнозування властивостей процесів, явищ оточуючої дійсності, інформатика стає необхідним інструментом розвитку інших фундаментальних та технічних наук та суспільства в цілому.

Сучасні підприємства не можуть вижити у конкурентній боротьбі, якщо не будуть випускати нові продукти кращої якості, більш низької вартості за менший час. Тому вони прагнуть використати можливості комп'ютерних технологій для того, щоб автоматизувати і зв'язати в єдиний процес частини своєї діяльності. Для досягнення цієї мети використовуються інформаційні технології.

### 1.1. Визначення, склад

Офіційні терміни та визначення, які повинні використовуватися в Україні в науково-технічній та навчальній царинах викладені в міжнародних стандартах та державних стандартах України.

Слід зазначити, що термінологія інформаційних технологій, зокрема інженерних, наразі не може вважатися остаточно оформленою. Різні джерела дають досить суперечливі тлумачення. Наприклад, загальновживаний в Європі термін *«informatics»* не використовується в США, де вживають термін *«computer science»*. А в ДСТУ 2382:2017 говориться про *«комп'ютерні науки»*, для яких рекомендовано вживати нову назву – *«комп'ютерні ка*» [3].

*Інформаційні системи* та *інформаційні технології* є елементуми загального процесу пізнання та вдосконалення навколишнього середовища.

ДСТУ 2382:1993 підкреслює, що «кожен, хто має справу з інформаційними технологіями має знати та розуміти міжнародні визначення засадничих термінів "знання", "інформація", "дані"».

8

ДСТУ 2226-94 надає найбільш загальне тлумачення, що «інформація (*information*) – це відомості про суб'єкти, об'єкти, явища та процеси» [4].

За ISO-IEC 2382-1:1993 «*дані* – надання інформації у формалізованому вигляді, придатному для її передавання, інтерпретації або обробки.

Знання – організоване та інтегроване зібрання (collection, собрание) фактів та узагальнень (generalization)» [5].

Термін *інформація* в цьому документі визначається через категорії *даних* та *знань* як «будь-який факт, поняття або значення (meaning), які отримані з даних, а також контекст, який обраний зі знань, або контекст, який асоційований зі знаннями» [5].

Взаємний зв'язок між *знанням*, *інформацією* та *даними* в ISO-IEC 2382-1:1993 пояснюється схемою співвідношень (рис. 1.1).



Рис.1.1. Інформаційні терміни [5]

Зі схеми видно, що знання оброблюються розумом людини. Це є важливим моментом логіки визначення термінів. На відміну від знань, інформація та дані можуть оброблятися зовні свідомості людини – системами обробки.

Поняття *даних* в сучасній редакції цього стандарту ДСТУ ISO/IEC 2382:2017 трактується як «переосмислене подання інформації формалізованим способом, придатним для обміну, інтерпретації чи опрацювання» [3], а *знання* як «колекція фактів, подій, переконань та правил, організованих для систематичного використання» [3]. При цьому знання ототожнюється зі штучним інтелектом (artificial intelligence). Найбільш вдалим можна вважати визначення знань за ДСТУ 2481-94 як «сукупності фактів, закономірностей, відношень та евристичних правил, що відображає рівень обізнаності з проблемами деяких предметних галузей» [7].

Комп'ютерні «оброблення системи проводять ланих систематизоване виконання операцій з даними. Наприклад, арифметичні або логічні операції з даними, злиття або сортування даних, складання (assembling) або компіляція програм, операції з текстом, такі як редагування тексту, сортування, поєднання, зберігання, відновлення, виведення на дисплей або друк» [5], «виконання впорядкованих операцій з інформацією, які містять обробку даних. Може містити операції передавання (communication) та офісну автоматизацію» [5].

В редакції стандарту ДСТУ ISO/IEC 2382:2017 терміни *оброблення даних*, *оброблення інформації* замінені на терміни *аналіз даних* та *аналіз інформації* як «систематичне дослідження інформації та її потоку в реальній або запланованій системі», термін *офісна автоматизація* вилучений.

Відповідно до ДСТУ 2226-94 *«інформаційна технологія* (information technology) – технологічний процес, предметом перероблення й результатом якого є інформація» [4].

ДСТУ 5034-2008 конкретизує визначення *інформаційних технологій* як «сукупність методів, процесів і програмно-технічних засобів, об'єднаних у технологічний процес, що забезпечує добирання, зберігання, накопичування, оброблення, пошук, виведення, копіювання, передавання та розповсюджування інформації» [6].

ДСТУ ISO/IEC 2382:2017 говорить про комп'ютерні науки як «галузь науки й технологій, що займається опрацюванням інформації за допомогою комп'ютерів» [4] та на відміну від англомовної версії документу вводить нову назву комп'ютерних наук «комп'юторика».

Як наукова дисципліна комп'ютерні науки виникли на початку 40-х років XX століття. Протягом півстолітньої історії комп'ютерних наук неодноразово виникали та зникали ті чи інші напрямки.

Предметну область комп'ютерних наук загалом можна розділити на дві великі підобласті. Перша з них включає вивчення до структур, механізмів та схем обробки інформації. Друга має відношення конкретних процесів обробки інформації та пов'язані з ними запитання подання даних. В редакції видання «Питер» 2004 р. відомого підручника [8] американського професора Дж. Брукшира (J. Glenn Brooksheer) виділяється дванадцять основних розділів комп'ютерних наук:

*Теорія алгоритмів та структур* даних охоплює теорію обчислюваності, теорію обчислювальної складності, теорію паралельних обчислень, теорію баз даних, теорію розпізнавання образів, криптографію тощо.

Мови програмування вивчає системи позначень, призначених для виконання алгоритмів та способи формальної запису самих алгоритмів і даних. Результатами цього напрямку є автоматизовані програмні середовища, які за введеним описом мови автоматично створюють компілятор для перекладу програм з цієї мови на машинну мову. Наприклад, YACC та LEX.

*Теорія архітектури комп'ютерів* охоплює цифрову логіку, булеву алгебру, теорію кодування та теорію кінцевих автоматів, методи виявлення та виправлення помилок, включаючи засоби відновлення після відмов. Результатом цього напрямку є сучасні модулі виконання арифметичних операцій, кеш-пам'ять, RISC-комп'ютери та CISC-комп'ютери тощо.

Розділ *операційних систем та комп'ютерних мереж* пов'язаний із дослідженнями механізмів управління, що дозволяють ефективно координувати роботу обчислювальних ресурсів при проведенні обчислень.

Розділ *розроблення програмного забезпечення* пов'язаний зі створенням великих програмних систем, які повинні задовольняти заданим програмним специфікаціям, бути безпечними, захищеними, надійними, зручними для користувачів.

Бази даних та інформаційно-пошукові системи. Цей розділ комп'ютерних наук пов'язаний з організацією наборів даних, що зберігаються, постійно оновлюються і спільно використовуються, ефективного виконання запитів методами реляційної алгебри та теорії паралельної обробки даних.

Штучний інтелект та робототехніка включає моделювання процесів пізнання світу з кінцевою метою створення компонентів машин, здатних імітувати чи посилювати їх. Основні об'єкти вивчення в цій галузі включають розпізнавання звуків, мов, зображень та образів, навчання, процеси прийняття рішень.

Комп'ютерна графіка пов'язана з процесами візуального представлення реальних та віртуальних об'єктів, а також імітації їх рухів на двовимірному екрані комп'ютера або в тривимірному середовищі засобами обчислювальної геометрії та інших галузей науки.

Взаємодія людини та комп'ютера вивчає запитання ефективної координації дій і передачі інформації між людьми і машинами (інтерфейс користувача) за допомогою різних чутників і пристроїв, що імітують діяльність людини, а також інформаційні структури, які відображають людські уявлення про навколишнє світі.

Обчислювальна математика пов'язана з проведенням наукових та технічних досліджень.

*Ділова інформатика* (офісна автоматизація) вивчає запитання обміну інформацією та створення програмних систем, що забезпечують роботу організацій і координацію дій їх співробітників.

Біоінформатика є напрямком комп'ютерних наук на межі обчислювальної математики та біологічних наук.

#### Контрольні запитання

- 1. Що таке «інформація»?
- 2. Що таке «дані»?
- 3. Що називають «обробленням даних»?
- 4. Що називають «інформаційною технологією»?
- 5. Як за ДСТУ ISO/IEC 2382:2017 називають комп'ютерні науки?
- 6. Назвіть напрями ком'ютерних наук.

#### 1.2. Історія розвитку інформаційних технологій

Перші елементи, які тим чи іншим чином застосовуються в сучасних інформаційних системах та технологіях, відносяться до часів другої інформаційної революції. На початку нашої ери з Індії та Китаю почалося використання систем числення, цифр, в тому числі цифри нуль, як засобів кількісного представлення інформації. Наприкінці першого тисячоліття нашої ери вченим Аль-Хорезмі введені поняття «алгоритм» та «алгебра». Тоді ж з'являється для торгових операцій перший обчислювальний пристрій – рахівниці. В Європі – під назвою *абак* (давньоєврейск. «пил, пісок»), в Китаї – *суан-пан*, в Японії – *соробан* (рис. 1.2).



Рис. 1.2. Перші рахівниці: а – абак; б – десяткові рахівниці

Значна кількість теоретичних розробок обчислювальних пристроїв з'явилося під час третьої інформаційної революції (XV ст. – XVI ст.)

Однією з перших обчислювальних машин був механічний пристрій Леонардо да Вінчи зі зубчастих коліс та стрижнів, які приводилися до руху набором вантажів.

В 1614 р. шотландський барон Джон Непер в Англії видав книгу «Опис дивовижних таблиць логарифмів» з даними у вісім розрядів, а в 1617 р. він придумав пристрій для множення чисел – «*пальці Непера*», який являв собою набір сегментованих стрижнів.

На основі його винаходу в 1620 р. професор Едмунд Гюнтер розробив логарифмічну шкалу, яка стала першим варіантом *логарифмічної лінійки*.

В 1645 р. французький вчений Блез Паскаль для полегшення роботи батька, який проводив багато розрахунків на посаді інтенданта, створив прототип арифметичного калькулятора — *«колесо Паскаля, паскаліна»* (рис. 1.3). Роботи були завершені виробництвом 50 машин на замовлення уряду Франції. Закладений Паскалем принцип став основою конструкцій наступних розробок на наступні три сторіччя.



Рис. 1.3. Арифмометр Паскаля

Перша відносно зручна у використанні механічна машина з чотирма арифметичними діями (рис. 1.4) була розроблена в Німеччині Готфрідом

Лейбницем в 1673 р. Свою машину Лейбніц розробив, коли по справах дипломатичної служби в Парижі побачив величезний обсяг обчислень, які проводив його знайомий астроном Христіан Гюйгенс.



Рис. 1.4. Машина Лейбніца

Ближчим за всіх до створення обчислювального пристрою, подібного до сучасного комп'ютера вважається англієць Чарлз Беббідж. В 1812 р. він видав статтю з описом різницевої машини та побудував її механічний екземпляр з зубчастих коліс, валів, які приводилися в дію важелем. В 1834 р. він розпочав розбудову рахувального пристрою «*analityc engine*», який повинен був проводити не одну операцію, а послідовність обчислення автоматично без участі людини згідно з інструкціями у вигляді кодованих карток (рис. 1.5). В складі машини передбачався механічний арифметичний пристрій та пам'ять на 100 чисел. Беббідж першим запропонував та частково реалізував *ідею програмно-керованих обчислень*.



Рис. 1.5. Прототип машини Беббіджа

Автором програм для аналітичної машини Беббіджа була графиня Ада Лавлейс. Вона вважається першою жінкою-програмістом. Одна з популярних мов програмування *Ada* була названі на її честь.



Ада Лавлейс

В період з 1790 по 1804 роки Жозеф Жаккар, який працював ткачем, розробив та виготував на замовлення уряду Франції перший програмований ткацький верстат (рис. 1.6 а) для виробництва крупновізерукових тканин (килими, скатертини тощо) з керуванням програмою у вигляді кодованої стрічки (рис. 1.6 б) – прообразу *перфострічок* XX століття.



Рис. 1.6. Програмований верстат: а – вигляд; б – кодована стрічка

Вперше практичне кодування інформації *перфокартами* в значних обсягах для обчислень було реалізовано в статистичному табуляторі американцем Германом Холлерітом для перепису населення в США в 1890 р. В 1896 р. Холлеріт заснував фірму «Tabulating Machine Company» з виробництва табуляційних машин (рис. 1.7), які продавав залізницям та державним установам.



Рис. 1.7. Табулятор Холлеріта: а – перфокарта; б – табулятор; в – перфоратор

З часом у 1911 р фірма Холлеріта була поглинута фірмою «Computer Tabulating Recording Company», яка з 1924 р. має назву «IBM – International Business Machines Corporation».

Наприкінці XIX століття петербургський інженер-механік, співробітник Експедиції державних паперів Вільгодт Однер виготовив механічний арифмометр на основі зубчастих *«колес Однера»* зі змінною кількістю зубців. Схема стала найрозповсюднішою на багато років (рис. 1.8).

Колесо мало дев'ять зубців, з кутом між двома зубцями, який прирівнювався до обчислювальної одиниці. Кожному розряду відповідало окреме колесо. Введення чисел проводилося важелями. При цьому на колесі висувалася відповідна кількість зубців. Після повертання рукоятки колеса через проміжні шестерні поверталися на кут, який відповідав числу-результату.



Рис. 1.8. Арифмометр Однера

В середині XIX століття англієць Джордж Буль написав дві статті, в яких розробив універсальну систему правил, яка мала три основні операції – AND, OR, NOT, як основні двійкові дії, які оперують поняттями: «істина/брехня», «так/ні», «відкрито/закрито», «одиниця/нуль». Завдяки участі американського вченого Чарльза Пірса теорія Буля стала основою логічних функцій майбутніх комп'ютерних систем.

«Спосіб електричної передачі зображень» був запатентований професором Петербургського технологічного університету Борисом Львовичем Розингом в 1907 році. 22 травня 1911 року Розинг отримав зображення геометричної статичної фігури на екрані кінескопу, що стало першим сеансом телебачення.

В березні 1925 року шотландський вчений Джон Лоуги Берд вперше продемонстрував динамічне зображення в електромеханічному телевізорі на основі дику Ніпкова. З липня 1928 року в США фірмою Western Television було продано перший в світі телевізор «Віжнетт» (Visionette) розробки

Володимира Кузьмича Зворикіна з 45-рядковою механічною розгорткою. з ціною біля 100 доларів. Масове застосування телебачення почалося в Німеччині в 1934 році телестанцією *DFR* (*«Deutscher Fernseh-Rundfunk»*) на 180-рядковій системі з телевізорами *Telefunken* (рис. 1.9).



Рис. 1.9. Серійний телевізор В2 з механічною розгорткою 12.5 кадр/сек, 30 рядків (https://ru.wikipedia.org/wiki/Телевизор)

«Батьком» сучасних універсальних електронних обчислювальних машин вважається науковець з Гарвардського університету США Говард Ейкен. На грант ВМС США та фірми ІВМ в 1943 р. він представив перший екземпляр машини «*Марк*-1» (офіційна назва – *ASCC*, *Automated Sequence Controlled Calculator*) на електромеханічних реле. Машина мала розміри 17 м х 2,5 м, містила біля 750 тис. деталей, 800 км дротів та працювала в десятковій системі числення (рис. 1.10). Інформація вводилася перфокартами *IBM*.



Рис. 1.10. Обчислювальна машина «Марк 1» (https://commons.wikimedia.org/w/index.php?curid=601387)

Юридично першість в створенні сучасних комп'ютерів належить професору болгарського походження Конраду Цузе. В 1938 р. він в Німеччині виготовив електромеханічну машину Z1 (рис. 1.12). Машина займала приблизно 4 м<sup>2</sup>. Для інформація вводилася перфострічками у вигляді 35-міліметрової кіноплівки з пробитими в ній отворами. Машина робила множення 22-бітних даних з плаваючою комою за 5 секунд, мала

пам'ять на 64 слова, клавіатуру, виведення на лампову індикаторну панель в десятковій системі, масу 500 кг, споживала 1 КВт.



Рис. 1.11. Машина Z1 (https://arxiv.org/pdf/1406.1886)

Наступний варіант – машина Z2 мала зчитування з паперової *перфострічки* (рис. 1.12).



Рис. 1.12. Перфострічка

В 1941 р. Цузе виготовив електромеханічну обчислювальну машину Z3. Машина виконувала вісім команд в двійковій системі: 4 арифметичні дії та обчислення квадратного кореня 22-розрядних чисел з плаваючою комою. Час додавання складав 0,3 с, множення – 4 с.

В 1946 р. під керівництвом Джона Моучлі та Джона Еккерта в Пенсільванському університеті США на замовлення та фінансування Балістичної лабораторії збройних сил США створили першу електронну обчислювальну машину ENIAC – Electronic Numerical Integrator and Computer.

Машина мала масу 30 т, займала 170 м<sup>2</sup> (довжина 26 м, висота – 6 м), містила 17468 ламп, 7200 діодів, 70 тисяч резисторів, 60 тисяч конденсаторів, 1500 реле, 5 млн паяних з'єднань. Машина проводила в десятковій системі 5 тис операцій додавання або 300 операцій множення в секунду. Дані кодувалися перфокартами *IBM*. Програма вводилася перекомутацією контактів, як на ручному телефонному комутаторі (рис. 1.13). Лампи нагрівали приміщення до 50 °C.



Рис. 1.13. Комп'ютер ENIAC (https://uk.wikipedia.org/wiki/ENIAC)

Саме в її назві вперше було застосовано термін «комп'ютер». До цього часу «комп'ютерами» ( computus, латин. – «обраховувати») називали людей, які професійно займалися розрахунками для математики, астрономії (рис. 1.14) тощо.



Рис. 1.14. Професійні «калькулятори»

Наприкінці 40-х років Джорж Стибиц запропонував називати калькуляторами пристрої, які призначені для проведення простих арифметичних дій, а комп'ютерами – машини, здатні автоматично виконувати послідовності операцій та зберігати проміжні результати. Людей, які працюють на таких машинах, він запропонував називати операторами (operator).

Сьогодні *калькулятором* називають «пристрій для виконання арифметичних операцій, який під час цього потребує втручання людини для зміни його збереженої програми, якщо така є, та ініціації кожної операції або послідовності операцій» [3].

Сьогодні комп'ютером називають «функційний блок, який може виконувати значні обчислення, зокрема багаточислові арифметичні та логікові операції без втручання людини» [3]. Першим серійним електронним комп'ютером вважається UNIVAC – Universal Automatic Computer (рис. 1.15), розроблений в 1951 р. Дж. Эккертом та Дж. Моучли на кошти Армії США. Він містив 5200 ламп, мав масу 13 тонн, займав площу 36 м<sup>2</sup>, споживав 125 кВт. Продуктивність складала 1900 операцій в секунду. Процесор та оперативна пам'ять мали розміри 4,3 м × 2,4 м × 2,6 м.

На UNIVAC вперше застосували 10 стрічкових накопичувачів UNISERVO з бронзовою стрічкою 1500 футів х 0,5 дюйми х 0,002 дюйми на 8 доріжок (6 доріжок даних, одна доріжка для контролю парності, одна — для синхронізації, загальною ємністю 1 440 000 шестибітних символів.



Рис. 1.15. Комп'ютер UNIVAC (https://uk.wikipedia.org/wiki/UNIVAC)

Теоретичні засади побудови сучасних універсальних комп'ютерів були розроблені Джоном фон Нейманом під час роботи в групі розробників нащадка машини «*Марк*-1» – ЕОМ *EDVAC*.

В 1945 р. він підготував звіт «Попередня доповідь про машину EDVAC», в якому описав формальну логічну організацію комп'ютера, яку сьогодні ми називаємо архітектурою.

В доповіді Нейман виділив наступні складові універсального комп'ютера: центральний арифметико-логічний пристрій, центральний пристрій керування операціями, запам'ятовуючий пристрій – пам'ять, пристрої введення-виведення.

Принципи фон Неймана:

- 1. Використання двійкової системи числення з причини простої реалізації арифметичних та логічних операцій.
- 2. Керування комп'ютером програмою, яка зберігається в пам'яті комп'ютера разом з даними та кодуються за одним принципом.
- 3. Комірки пам'яті мають послідовні чисельні адреси.

4. Можливість умовного переходу до будь-якої ділянки коду програми.

Сам термін «архітектура» комп'ютера було вперше використано співробітниками фірми IBM Лайлом Джонсоном та Фредеріком Бруксом в 1958 р.

Принципи фон Неймана сьогодні знайшли реалізацію у вигляді:

Архітектури CISC (Complete Instruction Set Computer). Була обрана фірмою IBM для родини IBM-360 в 1964 р. Реалізована в системах VAX, Intel x86, Pentium. Ознаками CISC-архітектури (комп'ютер з повним набором команд) є невелика кількість регістрів загального призначення, велика кількість машинних команд, які виконуються за кілька тактів, велика кількість способів адресації, команди різної розрядності.

Архітектури *RISC* (*Reduced Instruction Set Computer*). Була обрана фірмою *IBM* для машини IBM 801 в 1979 р. Ознаками *RISC*-архітектури (комп'ютер зі скороченим набором команд) є однакова довжина команд та формат, в якості операндів команд використовуються тільки регістри, наявність команд тільки простих дій, тільки пряма адресація.

Архітектури VLIW (Very Long Instruction Word), яка передбачає пакування кількох команд в єдине машинне слово. Реалізована системах *Itanium* Intel, *IA64*.

Архітектури *EPIC* (*Explicitly Parallel Instruction Computing*) є розвитком архітектури *VLIW* (обчислення з явним паралелізмом команд). Концепція *EPIC* розроблена для підвищення ефективності паралельних обчислень за рахунок предикатних полів та виключення логічних переходів. Реалізована в системах *Itanium 2 Intel*.

Через 20 років після першого патенту на миттєву фотографію в 1923 році Самуеля Шлафрока в липні 1948 р. з'явився перший ч/б фотоапарат з миттєвим друком (1 хв) «Polaroid Land Camera» виробництва *Agfa – Polaroid* (рис. 1.16).



Рис. 1.16. Один з перших фотоапаратів миттєвого друку (http://camera-wiki.org/wiki/Polaroid\_Highlander)

Після початку виробництва в 1950 році германієвих транзисторів в 1954 р. на ринок США вийшов перший транзисторний приймач *Regency* TR-1 на транзисторах *Texas Instruments*.

В 1954 р. фірма *IBM* продемонструвала прототип транзисторного калькулятора IBM-608 (рис. 1.17) з 3000 германієвими транзисторами. Калькулятор виконував 4500 операцій додавання в секунду, програми до 80 кроків. Випуск розпочався з 1957 року. Продано було 59 калькуляторів.



Рис. 1.17. Калькулятор IBM-608 (https://sites.google.com/site/itnewstimeua/in-the-news)

В тому ж 1957 році розпочався серійний випуск калькулятора на 4 дії арифметики над 14-розрядними десятковими числами Casio 14-A (рис. 1.18). Калькулятор мав масу 140 кг та споживання 300 Вт.



Рис. 1.18. Калькулятор Casio 14-A (https://sites.google.com/site/itnewstimeua/in-the-news)

В 1956 р. фірма *IBM* для комп'ютерів IBM-305 випустила першу комерційну систему жорстких магнітних дисків *RAMAC «Random Access Method of Accounting and Control»* IBM-350 (рис. 1.19) ємністю 5MB, масою 970 кг, на 50-ти дисках діаметром 24 дюйми.



Рис. 1.19. Перший комерційний магнітний диск IBM\_305 (https://en.wikipedia.org/wiki/IBM\_305\_RAMAC)

В 1956 році фірма *Mototrola*, США випустила перші комерційні пристрої для обміну текстовими повідомленнями (рис. 1.20).



Рис. 1.20. Перші пейджери

LGP-30 (рис. 1.21 а), розроблений Стенлі Френкелем з фірми *Librascope*, США в 1956 році, називають першим персональним комп'ютером. Машина призначалася для монопольної роботи одного оператора, мала розміри письмового стола, споживала 1,5 кВт, мала пам'ять на магнітному барабані. Було продано біля 500 машин за ціною 47000 доларів.



Рис. 1.21. Перші ПК: a – LGP-30 (https://en.wikipedia.org/wiki/LGP-30); б – IBM 610 (https://habr.com/ru/company/ua-hosting/blog/405753/)

ЕОМ IBM 610 була, розроблена Джоном Ленцем в 1954 році, під назвою *PAC «Personal Automatic Computer»*. Керування та введення даних проводилося з перфострічки та клавіатури, запам'ятовуючим пристроєм виступав магнітний барабан, результати виводилися на електричну

друкарську машинку. В машині вперше застосували курсор – спеціальну мітку на екрані монітора, яка вказувала місце, в якому проводилися дії. Значення sin(x) обчислювалося 20 секунд. Було реалізовано з 1957 р. 180 екземплярів з ціною 55000 доларів.

В 1957 р. інженер фірми *IBM* Джон. Бекус розробив алгоритмічну мову високого рівня *FORTRAN* (*FORmula TRANslation*).

12 вересня 1957 р. Джек Кілбі в компанії *Texas Instruments* продемонстрував першу мікросхему. 27 вересня 1960 р. під керівництвом Джея Ласта компанія *Fairchild Semiconductor* випустила перші напівпровідникові мікросхеми.

В 1959 р. під керівництвом Грейс Хоппер була розроблена основна мова програмування в 1960–70-ті *COBOL* (*COmmon Business Oriented Language*).

В період 1958-1963 років Джон Маккарті з Массачусетского технологічного інституту США розробив функціональну мову LISP (List PRocessing).

В 1960 р. вийшов перший серійний транзисторний телевізор SONY TV8-301.

В 1963 р. побачив світ перший масовий калькулятор ANITA MK VIII (рис. 1.22) фірми *Bell Punch Company*, Англія. Калькулятор мав розміри 376 мм х 450 мм х 255 мм, кнопкову клавіатуру, 12-тирозрядний дисплей та не міг виконувати програми.



Рис. 1.22. Перший масовий калькулятор Anita MK VII (https://en.wikipedia.org/wiki/Sumlock\_ANITA\_calculator)

В 1963 р. фірма *Philips*, Бельгія представила носій інформації у вигляді компактної магнітної стрічки Compact Cassette (рис. 1.23).



Рис. 1.23. Компакт-касета (https://en.wikipedia.org/wiki/Cassette\_tape)

В 1964 р. професорами Дартмутського коледжу Томасом Курцем та Джоном Кемені була розроблена мова BASIC (Beginner's All-purpose Symbolic Instruction Code)

В 1965 р. фірма *Digital Equipment Corporation* виробила мінікомп'ютер PDP-8 (Programmable Digital Processor) з вартістю біля 20 тис. доларів

(рис. 1.24). Машина мала пам'ять на 4 тисячі 12-бітних слів, час звернення до пам'яті 1,2 мксек, швидкодію 385 тис. додавань в секунду (190 тис. віднімань, 4000 множень). Було реалізовано біля 300 тис. екземплярів за ціною 18000 доларів з великим вибором периферійних пристроїв – накопичувачів на магнітних дисках та стрічках, телетайпів, перфострічок, моніторів тощо. Фірма була поглинута в 1998 році корпорацією *Compaq*, а та в 2002 році *Hewlett-Packard*.



Рис. 1.24. EOM PDP-8 (https://www.computercollection.net/index.php/pdp8and12/)

7 квітня 1964 р. відбулися 77 прес-конференцій в 15 країнах світу, де фірма *IBM* оголосили родини машин «Система-360» (System/360) (рис. 1. 25). Серія дебютувала шістьма моделями різної потужності.



Рис. 1.25. EOM S/360 (https://en.wikipedia.org/wiki/IBM\_System/360)

2 вересня 1969 р. в Каліфорнійському університеті під керівництвом професора Лена Клейнрока студенти Стівен Крокер та Вінтон Серф передали повідомлення по 5 м кабелю між двома комп'ютерами.

21 листопада 1969 р. відбулася перша передача по мережі між Каліфорнійським та Стенфордським університетами. Комерційний варіант мережі запровадила фірма *IBM* в 1972 р.

В 1969 р. було презентовано перший прототип електронного годинника Hamilton Pulsar-1 (рис. 1.29) на чипі *Texas Instruments*. В 1970 р був випущений перший серійний годинник Hamilton Pulsar-400d.

В 1970 р. Марсіан Хофф з фірми Intel створив першій мікропроцесор.

В 1971р. побачив світ перший кишеньковий калькулятор 901В фірми *Bomwar* (рис. 1.26). Калькулятор мав габарити 131 мм × 77 мм × 37 мм, виконував 4 арифметичні операції, мав 8-розрядний червоний індикатор, коштував \$240.



Рис. 1.26. Перший ручний калькулятор (https://ru.wikipedia.org/wiki/Калькулятор)

В 1971 р. Алан Шугарт з фірми *IBM* для пересилання клієнтам програм застосував магнітну дискету розміром 8 дюймів на 80 Кбайт одноразового запису (рис. 1.27).



Рис. 1.27. Перша дискета

В 1973 р. фірма *Motorola* представила перший стільниковий телефон Dyna TAC-8000 (рис. 1.28) масою 800 г та вартістю \$4000 (перша мережа 1G була запущена в 1984 р.).



Рис. 1.28. Перший стільниковий телефон

В період 1969 – 1973 років Денніс Рітчі з фірми *Bell Labs* розробив мову високого рівня *С*.

В 1973 р. в США використали волоконно-оптичну лінію зв'язку на борту корабля «Little Rock».

В 1974 р. Едвард Робертс випустив набір для домашнього збирання «персонального комп'ютера» Altair 8800 (рис. 1.29). Комп'ютер містив 256 байт пам'яті, не мав клавіатури та екрана. Введення програм та даних проводилося в двійковому коді перемикачами, результати зчитувалися в двійковому коді по лампочках. Це не завадило Білу Гейтсу та Полу Алену розробити на ньому свій першій програмний продукт.



Рис. 1.29. Комп'ютер Altair 8800 (https://uk.wikipedia.org/wiki/Altair\_8800)

В 1975 р. фірма *Hamilton* випустила перший комбінований електронний годинник Pulsar 901з калькулятором (рис. 1.30).



Рис. 1.30. Перший годинник з калькулятором (https://en.wikipedia.org/wiki/Pulsar\_(watch))

В 1975 р. фірма *IBM* випустила прототип персонального комп'ютера (ПК) IBM 5100 (рис. 1.31).



Рис. 1.31. ПК IBM 5100 ( https://ru.wikipedia.org/wiki/IBM\_5100) В 1976 р. в продаж надійшов перший ПК *Apple Computer* – Apple 1 (рис. 1.32).





В 1976 р. Ан Вэнг з Wang Laboratories та Джим Адкіссон, Дон Массаро з *Shugart Associates* запропонували дискету 5,25 дюймів (серветка – mini-floppy) (рис. 1.33).



Рис. 1.33. Міні флоппі

В 1976 р. розпочався серійний випуск лазерних принтерів IBM-3800 та Xerox 9700 (рис. 1.34) з швидкістю друку 120 стор. за хвилину з ціною \$350000.



Рис. 1. 34. Перші лазерні принтери (https://en.wikipedia.org/wiki/IBM\_3800)

В 1977 р. розпочато серійний випуск фотоапарату з кольоровим миттєвим друком One Step фірми *Polaroid* (рис. 1.35), з'явилися перші серійні струменеві принтери *Siemens*. В Лонг-Біч, штат Каліфорнія, США компанія *General Telephone and Electronics* вперше використала систему передавання трафіка на оптичному кабелі.



Рис. 1.35. Фотоапарат Опе Step В 1979 р. фірма DC Hayes, США випустила перший модем. В серпні 1981 р. почалося виробництво ПК IBM 5150 (рис. 1.36).



Рис. 1.36. ПК ІВМ РС

В 1981 р. фірма *Sony* випустила міні дискети 3,5'' (рис. 1.37 а) ємністю 720 Кбайт. Наступного року почався серійний випуск програвачів компакт дисків CDP-101 (рис. 1.37 б).

В 1983 р. Б'єрн Страуструп з фірми *Bell Telephone Laboratories*, США розробив мову програмування високого рівня С++.

В 1983 р. на годиннику Casio AT-550 (рис.1.38) вперше застосовано введення торканням до екрану *touchscreen*.



Рис. 1.37. Винаходи SONY: а – міні-дискети; б – програвач дисків



Рис. 1.38. Годинник з touchscreen

(https://www.microsoft.com/buxtoncollection/detail.aspx?id=227)

В 1984 р. фірма Apple випустила перший моноблочний ПК МАС (рис. 1.39).



Рис. 1.39. ПК МАС

В 1984 р. з'явився перший кишеньковий ПК (*personal digital assistant PDA*, *handheld PC*, *pocket PC*) *PSION-ORGANIZER-1* з об'ємом пам'яті 2 кБ (рис. 1.40). Того ж року фірма *Hewlett-Packard* випустила першій лазерний принтер LaserJet за \$3500.



Рис. 1.40. Перший кишеньковий ПК (https://oldcomputers.net/psion-organiser.html)

В 1987 р. побачив світ перший планшет LINUS WRITE-TOP (рис. 1.41). Планшет мав масу 4 кг та працював з операційною системою DOS.



Рис. 1.41. Перший планшет (https://oldcomputers.net/linus.html)

В 1984 р. в підрозділі *Ataritel* фірми *Atari Video Game Company*, Японія (з 1984 р. – *Mitsubishi Electronic*) почалися роботи з розроблення відеотелефона Luma LU -1000 (рис. 1.42).



Рис. 1.42. Відеотелефон Luma

(https://keddr.com/2016/07/evolyutsiya-kamer-mobilnyih-telefonov/)

В 1989 р. Національний науковий фонд США створив мережу *NSFNET* для зв'язку між університетами та обчислювальними центрами. Передавання мережі NSFNET в комерційне користування народило сучасний Інтернет.

1991 р. став роком народження мови програмування *Python*, яку з мови *ABC* розробив співробітник голландського інституту *CWI* Гвідо ван Россум, та мови *Visual Basic*.

1 липня 1991 р. в Фінляндії відбувся перший дзвінок 2G (2 Generation) – GSM (Groupe Spécial Mobile, Global System for Mobile Communications). В 1992-му році першу мережу в цьому стандарті побудували компанії Telenokia i Siemens.

В 1993 р. в продаж надійшов перший планшет APPLE NEWTON (рис. 1.43).



Рис. 1.43. APPLE NEWTON

В 1994 р. результатом співпраці фірм *Apple* та *KODAK* стала перша комерційна цифрова камера QuickTake 100 на 8 кадрів формату VGA з роздільною здатністю 0.3 МП (рис. 1.44). Камера працювала тільки з комп'ютерами *Macintosh* та коштувала \$749.

В 1995 р. консорціум Intel, Microsoft, Philips, US Robotics розробили специфікацію та випустили перші пристрої USB «Universal Serial Bus».



Рис. 1.44. QuickTake 100 (https://sncmuseum.org/en/apple-quicktake-100)

В 1996 р. вийшов перший комунікатор – пристрій, який суміщав кишеньковий ПК та телефон – NOKIA-9000 communicator (рис. 1.45), перший мобільний телефон з трикольоровим екраном Siemens S-10, перший комерційний PDA PALM PILOT (рис. 1.46) фірми *Palm Computing* (підрозділ *U.S. Robotics – 3Com – Palm, Inc – Hewlett Packard*) з операційною системою Palm OS.



Рис. 1.45. NOKIA-9000 (https://medium.com/people-gadgets/the-gadgetwe-miss-the-nokia-9000-communicator-ef8e8c7047ae)



Рис. 1.46. PALM PILOT (https://www.alamy.com/palm-pilotimage5830792.html)

В 1997 р. співробітник *Intel* Джим Кардаш (Jim Kardach) розробив систему для зв'язку між мобільним телефоном та комп'ютером. Вважається, що він надав назву системі за прізвищем героя роману Франса Г. Бенгтссона «Рудий Орм» про короля Данії Гаральда Синьозубого (норв. Harald Blåtann) – Bluetooth. Перші літери імені короля створили логотип **\***, **§**.

В 1998 р. фірма *HP* захопила масовий ринок лазерних принтерів з моделлю Laser Jet, співробітники *Microsoft* Андерс Хейлсберг та Скотт Вільтаумот для платформи Microsoft .NET Framework розробили мову програмування C#, в радіоастрономічній лабораторії *CSIRO* (*Commonwealth Scientific and Industrial Research Organization*) в Канберрі, Австралія інженер Джон O'Caлліван розробив технологію бездротового передавання даних *WiFi «Wireless Fidelity»* («бездротова точність»). Вже в наступному році на ринку з'явилися роутери.

18 травня 1999 р. після двох років досліджень компанія *Kyocera Corporation* випустила перший мобільний телефон з вбудованою відеокамерою 0,11 МП та двухдюймовим ТFT екраном за \$325 (рис. 1.47 а). Відео передавалося з швидкістю 2 кадри/сек. Перший відеодзвінок було проведено в 2003 році через 3G мережу.

Перший мобільний телефон з фотокамерою Sharp J-SH04 (рис. 1.47 б) було випущено в листопаді 2000 р.



Рис. 1.47. Перші телефони: а – з відеокамерою; б – з фотокамерою

В 2000 р. фірма Benefon представила мобільний телефон з GPS (Global Positioning System, система глобального позиціонування) Benefon ESC, фірма Ericsson випустила перший смартфон Ericsson R380S (рис. 1.48), ізраільска фірма M-System почала виробництво флеш накопичувачів, фірма Sony реалізувала поліфональний звук в телефоні Sony J5.



Рис. 1.48. Перший смартфон (https://www.gsmchoice.com/ru/catalogue/ericsson/r380s/gallery/ericssonr380s-match/)

В 2001 р. фірма *Apple* випустила мультимедійний IPOD, вийшов перший смартфон з кольоровим екраном Nokia 9201.

В 2002 р. співробітник *Microsoft Research*, Кембридж Дон Сайм для платформи Microsoft .NET Framework розробив мову F#.

В 2003 р. почалася експлуатація стільникових мереж EDGE (G2.5).

В 2004 р. на ринку з'явилися бездротові навушники *Ericsson* BlueTake-Iphono з Blootooth 2.

В 2005 р. почалася комерційна експлуатація стільникових мереж 3G, було випущено першу відеокамеру *GoPro Hero*.

В 2007 р. вийшов перший «сучасний» смартфон з операційною системою *iOS*, через рік фірма *HTC* випустила смартфон HTC Dream (рис. 1.49) з системою *android*.



Рис. 1.49. HTC Dream (https://en.wikipedia.org/wiki/ HTC\_Dream)

В 2009 р. почалася комерційна експлуатація стільникових мереж 4G *LTE «Long Term Evolution»*.

В 2010 р. вийшов перший пристрій читання електронних книг Amazon Kindle з екраном за технологією «електронний папір», планшет iPad.

## Розвиток інформаційних технологій в Україні

В 1945 р. професора МВТУ Сергія Олексійовича Лєбєдєва запросили в Київ на посаду директора інституту енергетики АН УРСР (Інститут електротехніки АН УРСР), де в 1948 р. під його керівництвом було започатковано роботи по створенню ЕОМ. Незалежно від Дж. фон Неймана



Сергій Олексійович Лєбєдєв

Лєбєдєв висунув та реалізував принципи побудови ЕОМ з програмою, яка зберігається в пам'яті машини.

Перша ЕОМ МЭСМ «малая электронная счетная машина» (рис. 1.50) почала роботу в 1951 р. в Києві. МЭСМ містила 6000 ламп, виконувала 50 операцій в секунду, мала ОЗП на діодних тригерах на 31 число та 63 команди по 16 двійкових розрядів з фіксованою комою, робочу частоту 5 кГц, штекерний ПЗП на 31 число та 63 команди, ЗЗП на магнітному барабані в 5000 слів, споживала 15 кВт, займала площу 60 кв. м. Машина виконувала чотири арифметичні операції та порівняння чисел.

В 1955 р. Лєбєдєв переїхав в Москву. Колектив розробників передали в Інститут математики АН УРСР.

В 1952 р. під його керівництвом в інституті точної механіки та обчислювальної техніки була розроблена ЕОМ БЭСМ «большая электронная счетная машина» (рис. 1.51). Машина мала швидкодію 2 тис. оп./с, складалася з 5000 ламп, споживала 35 кВт. БЭСМ серійне вироблялася з 1958 р. Модель БЭСМ-6 мала швидкодію 1 млн оп./сек.



Рис. 1.50. EOM MЭCM (https://www.engadget.com/2011-12-26-mesmsoviet-computer-project-marks-60-years.html)



Рис. 1.51. EOM БЭСМ (https://en.wikipedia.org/wiki/BESM)
В 1956 році в інститут математики АН УРСР (сьогодні – інститут кібернетики імені академіка Глушкова НАН України) було запрошено з Москви Віктора Михайловича Глушкова, де він очолив групу Лєбєдєва.



Віктор Михайлович Глушков

Першим проєктом групи в 1956 р. була ЕОМ «Киев» (рис. 1.52). Вироблено було два зразки: один для інституту математики, другий – для інституту ядерних досліджень (м. Дубна).

Машина мала ферит-трансформаторний ЗП об'ємом в 512 слів, зовнішню пам'ять на магнітних барабанах об'ємом 9000 слів. Обчислювальна потужність в 300 разів перевищувала потужність МЭСМ.



Рис. 1.52. EOM «Киев» (http://www.icfcst.kiev.ua/MUSEUM/PHOTOS/KIEV\_r.html)

Вперше в світі за допомогою «Киева» були проведено дистанційне керування технологічними процесами металургійного комбінату в Дніпродзержинську з Києва. Машина використовувалася в експериментах зі штучного інтелекту, машинного розпізнавання геометричних фігур, автоматичного проєктування функціональних схем, в апробації однієї з перших в світі СУБД «Автодиректор».

Першою напівпровідниковою керуючою машиною та первістком серійного виробництва стала ЕОМ «Днепр» (УМШН, управляющая машина широкого назначения) (рис. 1.53).



Рис. 1.53. EOM «Днепр» (http://www.icfcst.kiev.ua/MUSEUM/GL\_HALL2/COMPUTERS/Dnepr\_r.html

Машину було запущена в серію в 1961 р. Перші ЕОМ випускав Київський завод «Радиоприбор», для серійного виробництва було збудовано завод (ВУМ, вычислительных управляющих машин, сьогодні – «Електронмаш») в Києві.

«Днепр» використовувався на підприємствах ливарного, хімічного, суднобудівного виробництва. Аналогом машини стала американська ЕОМ RW300. В 1972 р. «Днепр» використовувався для контролю польоту «Союз 19-Аполлон».

Машина працювала в двійково-десятковій системі, мала ОЗП на 140 слів, введення даних з металізованих перфокарт, буфер команд на 100 команд, виконувала 32 операції, мала обчислювальну потужність 1000 операцій за хвилину (100 множень).

В 1959 р. під керівництвом В. М. Глушкова почалися роботи по машинам для інженерних розрахунків. В 1963 р. заводі в Сєверодонецьку почали виробляти ЕОМ «Промінь» (рис. 1.54).



Рис. 1.54. EOM «Промінь» (http://www.icfcst.kiev.ua/MUSEUM/PHOTOS/ /Promyn-1\_r.html)

В 1965 р. з'явилась ЕОМ МИР-1 (рис. 1.55) «машина для инженерных расчетов». В 1967 р. на всесвітній виставці в Лондоні МИР-1 отримала перший приз та була придбана IBM.

За рахунок новітньої архітектури МИР успішно змагалася з більшими універсальними ЕОМ, які формально мали більшу швидкодію та обсяги ОЗП. На машині вперше було застосовано діалоговий режим роботи на дисплеях зі світловим пером.

ЕОМ працювала в двійково-десятковій системі з даними з фіксованою та плаваючою точкою довільної розрядності, операції проводилися на основі вбудованих мікропрограм. Введення даних проводилося з електричного друкарського клавіатурного пристрою Zoemtron та перфострічок.

Обчислювальна потужність складала 1000-2000 операцій в секунду (300 множень), споживання 1,5 кВт, маса 400 кг. Швидкодія наступних моделей МИР складала: МИР-2 12000 операцій в секунду, МИР-3 – 400 000.

Машини мікропрограмну реалізацію мали апаратну MOB програмування високого рівня, можливість виконання чисельних та аналітичних розрахунків. Для них були розроблені мови високого рівня «МИР» і «АНАЛИТИК», які давали змогу працювати з дійсними та цілими розрядності, точні операції числами довільної над дробовими раціональними числами тощо. Система «АНАЛИТИК» була однією з перших систем комп'ютерної алгебри, а в мові «АНАЛИТИК» вперше була використана техніка запису алгебраїчних виразів у вигляді співвідношень, яка в наш час є основою технології декларативного програмування.



Рис. 1.55. EOM «МИР-2» (http://www.icfcst.kiev.ua/MUSEUM/PHOTOS/MIR2\_r.html)

Алфавіт складався з кириличних та латинських літер, алгебраїчних знаків, цифр тощо. Опис алгоритмів проводився термінами «заменить»,

«разрядность», «вычислить», «если», «то», «таблица» тощо. Всі необхідні параметри виведення програмувалися.

## Контрольні запитання

- 1. Хто вважається розробником першого механічного обчислювального пристрою, прототипу арифметичного калькулятора?
- 2. Чим примітні здобутки Чарльза Беббіджа?
- 3. Хто вважається розробником логічних функцій?
- 4. Які параметри мала перша електромеханічна обчислювальна машина?
- 5. В чому полягає відмінність комп'ютера від калькулятора?
- 6. В чому полягають принципи фон Неймана?
- 7. Що таке МЭСМ?
- 8. Хто в Україні очолював розроблення ЕОМ?
- 9. Що означає акронім МИР?
- 10. Які засоби аналітичних розрахунків були розроблені в Україні?

# 1.3. Комп'ютерні мережі

Ефективність сучасних інформаційних технологій багато в чому завдячує мережевим засобам та технологіям, які забезпечують доступ споживачів до ресурсів інформаційних систем.

В середині XX ст. характеристики ЕОМ надали змогу долучитися до комп'ютерних обчислень достатньо широкому загалу фахівців, для вирішення задач яких були потрібні великі обчислювальні потужності.

Концепція колективних обчислень почала реалізовуватися на великих EOM «mainframe» в «пакетному» режимі. Фахівці готували завдання для EOM у вигляді кодів програм на паперових носіях. Завдання збиралися в пакети, які виконувалися на EOM. Замовники забирали паперові носії з результатами розрахунків. Поява OC «розподіленого, реального часу» дозволила комплектувати EOM кількома терміналами. Термінали не мали власних обчислювальних ресурсів та являли собою пристрій введеннявиведення (комплект монітора та клавіатури). Всі обчислення проводилися центральною EOM. Співробітники великих підприємств отримували доступ до термінальних залів обчислювального центру організації, Установи, які називалися машинними бюро надавали в оренду потужності EOM через термінали зовнішнім споживачам. ДСТУ 2382:2017 [3] визначає *мережу* (network), як «організування об'єктів та їх взаємозв'язок. У топології мережі або в абстрактному розташуванні взаємопов'язані об'єкти є точками на схемі, а взаємоз'єднання – лініями на схемі.

Об'єкти мережі називають *вузлами* (node). *Вузол* є об'єктом, з'єднаним з одним чи більше об'єктами. У мережі пересилання даних це є точкою, де один або кілька функціональних блоків з'єднують канали пересилання чи схеми пересилання даних.

Тобто *мережа* є організованим розташуванням *вузлів* та з'єднаних гілок.

Відповідно, *комп'ютерна мережа* (computer network) – це «мережа, вузлами якої є комп'ютери та обладнання обмінювання даними, а гілками – канали пересилання даних» або «мережа вузлів опрацювання даних, які взаємопов'язані для обмінювання даними» [3].

Мережі можна класифікувати за наступними ознаками:

- за фізичним середовищем передавання: з коаксіальним кабелем, з кабелем «кручена пара», з волоконно-оптичним кабелем, з ефірним каналом тощо;
- за топологією: шинна, кільцева, зіркоподібна тощо;
- за територіальним розташуванням: локальні, глобальні тощо;
- за сферою застосування: офісні, промислові, військові тощо;
- тощо.

Важливою ознакою мережі є їхня архітектура. *Мережна архітектура* (network architecture) – це «логікова структура і принципи функціонування комп'ютерної мережі. Принципи функціонування комп'ютерної мережі охоплюють сервіси, функції та протоколи» [3].

Протокол – це «набір семантичних та синтаксичних правил, які визначають поведінку об'єктів одного рівня під час виконання функцій обмінювання даними» [3].

Слід зазначити, що у визначенні не уточнюється, що правила стосуються опису логічних лій та фізичної реалізації обміну даними.

Можна сказати, що *протокол* - це набір сигналів та правил, процедур їхнього оброблення, які регламентує взаємодію між двома пристроями в мережі. Протоколи можуть реалізовуватися апаратно, програмно та апаратно-програмно.

Найбільше розповсюдження отримали мережі архітектури *Ethernet* за стандартом IEEE 802.3 ETHERNET WORKING GROUP – технології пакетного передавання даних між вузлами. Назва архітектури складається зі слів *ether* «ефір» та *network* «мережа», бо в перших мережах все, що передавалося одним вузлом могло бути прийнято всіма вузлами мережі подібно до мережі радіомовлення.

В мережах промислової автоматики застосовується архітектура LonWorks (Local Operating Network) за стандартом EN14908.

Найчастіше мережі виокремлюють за назвами локальні комп'ютерні мережі (LAN, Local Area Networks) та глобальні комп'ютерні мережі (WAN, Wide Area Networks).

*Локальна мережа* (ЛМ) (local area network; LAN) – це «комп'ютерна мережа, розташована в приміщеннях користувача в межах обмеженої географічної зони. Обмін даними в локальній мережі не регулюється зовнішніми нормами» [3].

Використовують ЛМ для ефективного використання ресурсів: дисковий простір, принтери тощо, та колективного опрацювання даних.

Глобальні комп'ютерні мережі не обмежені якоюсь певною територією можуть охоплювати значні території. Основним елементом, який визначає вартість мережі та швидкість передавання даних в цих мережах – обладнання та лінії зв'язку.

Історично першими з'явилися глобальні мережі. В 1966 р. для потреб міністерства оборони США почалася розробка проєкту ARPANET (Advanced Research Projects Agency Network) під головуванням Джозефа Ліклайдера. Результатом проєкту стала комп'ютерна мережа між провідними університетами та науковими центрами США та Англії (рис. 1.56).



Рис. 1. 56. ARPANET (https://en.wikipedia.org/wiki/ARPANET)

Останнім часом в літературі можна зустріти випадки виділення в окрему групу *регіональних* мереж (MAN, Metropolitan Area Networks) – мереж масштабів району, міста, області, регіону.

## Базова еталонна модель мережі

Базова еталонна модель взаємодії відкритих систем (БМВВС) (Open Systems Interconnection Basic Reference Model) мережі визначена в міжнародному стандарті, розробленому спільно Міжнародним телекомунікаційним союзом (ITU, International Telecommunication Union) та Міжнародною організацією по стандартизації (ISO, International Standards Organization) ISO/IEC 7894-1:1994.

Термін «відкритості» систем не пов'язаний з їхньою реалізацією технічними засобами, а визначає їхню відповідність з дотриманням вимог міжнародних стандартів.

БМВВС грунтується на наступних засадах:

- функції обміну в мережі є складними, тому для їх визначення доцільно використовувати ієрархічний підхід з розділенням за рівнями (level);
- кожен рівень обслуговує свою частину процесу взаємодії та виконує скінчений набір завдань;
- на межі між рівнями обмін даними повинен бути мінімальним;
- рівні повинні описуватись так, щоб зміни на одному з них не викликали необхідності внесення змін на інших рівнях.



Рис.1.57. Рівні БМВВС

БМВВС виділяє сім ієрархічних рівнів обробки даних (рис. 1.57) під час сеансу зв'язку [9]:

- прикладний (application), рівень 7;
- відображення (presentation) рівень 6;
- сеансовий (session), рівень 5;
- *транспортний* (transport), рівень 4:
- мережевий (network), рівень 3;
- канальний (data link), рівень 2;
- фізичний (physical). рівень 1.

Протоколи рівнів 1 - 3 керують фізичною доставкою даних мережею тому їх називають рівнями середовища передавання даних або базовими.

На кожному рівні N два вузли обмінюються протокольними блоками daнux (PDU, Protocol Data Unit) у форматі та за правилами відповідного протоколу. Кожен PDU складається з блока сервісних (службових) daнux (SDU, Service Data Unit) від верхнього або нижнього рівня та протокольної управляючої інформації (PCI, Protocol Control Information) свого рівня.

Обробка даних у взаємодії між двома OSI-сумісними пристроями проходить наступним чином. Дані, що передаються на верхньому рівні складаються в протокольний блок даних PDU рівня N та передаються на рівень N-1. На рівні N-1 передані дані стають сервісним блоком даних SDU. До нього додається протокольна управляюча інформація PCI. Разом вони створюють протокольний блок даних рівня PDU N -1.

Процес продовжується до самого нижнього рівня, з якого дані передаються на приймаючий пристрій.

На приймальному пристрої дані передаються від нижчого рівня до найвищого у вигляді SDU. При цьому на кожному рівні проводиться послідовне видалення PCI відповідного рівня (рис. 1.58).



Рис. 1.58. Обробка даних між рівнями БМВВС [9]

Кожному рівню відповідає свій типовий елемент даних. На фізичному рівні – біт, на канальному рівні – кадр даних (frame), на мережевому – в пакет (packet), на транспортному – сегмент (segment) та датаграма (datagram). Дані, логічно поєднані для передавання називаються повідомленнями (message). Повідомлення є типовими елементуми даних на сеансовому рівні, рівнях представлення та прикладному.

#### Прикладний рівень

*Рівень застосунків* є верхнім рівнем моделі, який забезпечує доступ до таких мережевих служб як-от віддалений доступ до файлів, електронна пошта тощо. На цьому рівні формуються запити до рівня відображення, повідомлення про помилки тощо.

Протоколами прикладного рівня є TELNET, HTTP, FTP, POP3, SMTP, WebSocket, Modbus, ISO 8583 тощо.

#### Рівень відображення

*Рівень відображення* забезпечує перетворення протоколів, форматування, пакування, кодування та шифрування даних. На цьому рівні дані з прикладного рівня перетворюються на формат, зручний для передавання мережею. Рівень відображення є рівнем «перекладання» між різнорідними системами.

Протоколами даного рівня є ICA (Independent Computing Architecture), LPP (Lightweight Presentation Protocol), NDR (Network Data Representation), XDR (eXternal Data Representation), ASCII, EBCDIC (Extended Binary Coded Decimal Interchange Code), JPEG (Joint Photographic Expert Group), MIDI (Musical Instrument Digital Interface) тощо.

#### Сеансовий рівень

Сеансовий рівень забезпечує безпечне та надійне з'єднання «точка точка», підтримку сеансу зв'язку на тривалий час. Він керує початком та завершенням сеансу, синхронізацією обміну, визначенням прав на передавання даних.

Протоколами даного рівня є: H.245 (Call Control Protocol for Multimedia Communication), ISO-SP (OSI Session Layer Protocol (X.225, ISO 8327)), iSNS (Internet Storage Name Service), PAP (Password Authentication Protocol), PPTP (Point-to-Point Tunneling Protocol), RPC (Remote Procedure Call Protocol), RTCP (Real-time Transport Control Protocol), SMPP (Short Message Peer-to-Peer) тощо.

# Транспортний рівень

Транспортний рівень забезпечує налагодження зв'язку між кінцевими споживачами та визначає ступінь надійності передавання даних. На цьому рівні використовуються протоколи від тих, що надають тільки базові транспортні функції без підтвердження прийняття та контролю даних, протоколів з контролем цілісності даних в рамках тільки однієї датаграми з можливістю втрати пакета даних в цілому, дублювання пакетів, порушення порядку передавання пакетів (наприклад, UDP) до протоколів з гарантованою доставкою пакетів даних в первинній послідовності (наприклад, TCP).

Протоколами даного рівня є CUDP (Cyclic UDP), DCCP (Datagram Congestion Control Protocol), FCP (Fibre Channel Protocol), SCTP (Stream Control Transmission Protocol), SPX (Sequenced Packet Exchange), SST (Structured Stream Transport), TCP (Transmission Control Protocol), UDP (User Datagram Protocol) тощо.

## Мережевий рівень

*На мережевому рівні* формуються мережеві адреси та визначається маршрут передавання даних. На цьому рівні проводиться переведення логічних адрес на фізичні, визначення найкоротших маршрутів, відслідковування колізій при передаванні даних в мережі.

Протоколами цього рівня є: IP/IPv4/IPv6 (Internet Protocol), IPX (Internetwork Packet Exchange,), IPsec (Internet Protocol Security), RIP (Routing Information Protocol) тощо.

На цьому рівні застосовуються програмні засоби мережевих шлюзів та мережевих екранів, функціонують апаратні *маршрутизатори* (router).

*Маршрутизатор* – функціональний пристрій, який передає пакети даних між сегментами локальної мережі або мереж (шлюз локальної мережі, LAN gateway) на базі правил та таблиць маршрутизації.

Маршрутизатором може бути апаратний пристрій, або спеціалізований застосунок, зазвичай на ОС Linux BSD, наприклад, Quagga.

Для організації зв'язку за протоколом TCP/IP кожному вузлу надається логічна адреса у вигляді чотирьох чисел, розділених крапками, наприклад, 192.168.126.10. IP-адреса складається з двох компонентів: адреси мережі та адреси вузла (хоста). Адреса мережі може займати 3, 6 або 9 розрядів. Решта цифр визначають адресу вузла (табл. 1.1)

Таблиця 1.1. Типи мереж

Клас мережі	Значення	Формат	Формат	Кількість	Кількість
	першого	адреси	адреси	мереж	вузлів
	числа адреси	мережі	вузлів		
А	1 - 126	W	x.y.z	126	16777214
В	128 - 191	W.X	y.z	16384	65534
C (LAN)	192 - 223	w.x.y	Z	2097151	254

Адреси мереж з номерами 127, 224 є зарезервованими. Адреси 192.168.у.z призначені для локальних мереж. Значення 255 використовується як позначка мережі в масках мереж.

Маска мережі є шаблоном, який визначає формат адрес вузлів мережі. В мережі класу А вузли мають маску 255.0.0.0, в мережі класу В – маску 255.255.0.0, а мережі класу С – маску 255.255.255.0.

Таблиця маршрутизації складається із записів-маршрутів, кожна з яких містить ідентифікатор мережі отримувача, адресу наступного вузла в ланцюгу передавання, «адміністративну відстань» – число, яке показує якість до джерела та метрику - число, яке показує «вагу» маршруту тощо.

Наприклад:

192.168.64.0/18 [110/49] via 192.168.1.2, 00:34:34, FastEthernet0/0.1

де 192.168.64.0/18 — мережева адреса приймача; 110 — адміністративна відстань; 49 — метрика маршруту; 192.168.1.2 — адреса наступного вузла; 00:34:34 — час існування маршруту; FastEthernet0/0.1 — інтерфейс гілки зв'язку.

Алгоритм вибору маршруту має наступний вигляд. З пакету читається блок мережевих адрес. В блоці адрес виділяється адреса мережі призначення. Якщо адреса мережі призначення знаходиться в локальній мережі відправника, то пакет надсилається безпосередньо адресату. Якщо адреса призначення не з локальної мережі відправника, але є в таблиці маршрутизації, то пакет надсилається за відповідним маршрутом. Якщо ні один із варіантів неможливий, то видається повідомлення про помилку маршрутизації.

В мережі часто виникають ситуації, коли мережеву послугу замовляють кілька застосунків в одному вузлі. Наприклад, два примірника веб-браузера з одного клієнтського комп'ютера звертаються до одного сайту (рис. 1.59 а) або до вузла провайдера надходять одночасно запит електронної пошти та файлового обміну з одного клієнтського комп'ютера (рис. 1.59 б). В таких ситуаціях звичайних мережевих адрес недостатньо.



Рис. 1.59. Порти: а – два клієнти; б – два сервери

Для розв'язання подібних проблем застосовується додаткова мережева адреса – *порт*.

Для протоколу TCP/IP – це ціле число в межах від 1 до 65535. Для серверів виділяються номери від 1 до 1023, для клієнтів – від 1024 до 65535.

Серверу конкретного типу призначається єдиний незмінний номер порту. Цей номер називають стандартним портом. Він є стандартизованим, наприклад, 80 – порт веб-серверів, які обмінюються даними за протоколом http, 21 – порт FTP-серверів; 25 – порт поштових серверів, в яких застосовується протокол SMTP. Деякі веб-сервери використовують порт 8080, проксі-серверам виділений порт 3128.

Клієнтські номери портів є тимчасовими, присвоюється на час взаємодії з сервером та не мають фіксованих значень. Під час встановлення зв'язку ОС автоматично призначає номер порту з тих, які в даний момент не використовується.

Тобто для опису з'єднання використовуються чотири параметри: адреса сервера та номер порту сервера, адреса клієнта та номер порту клієнта.

В позначеннях URL (Uniform Resource Locator) повний формат адреси має наступний вигляд:

«протокол://адреса\_web-сторінки/#закладка:порт».

#### Канальний рівень

Канальний рівень забезпечує обмін даними між вузлами одного сегменту локальної мережі адресацією на фізичному рівні.

На канальному рівні дані вищих рівнів розбиваються на кадри фіксованого розміру для передавання на фізичний рівень та біти даних фізичного рівня групуються в кадри при прийманні даних.

Важливою функцією канального рівня є отримання доступу до середовища передавання.

Адресація на канальному рівні проводиться додаванням в заголовок кадру *апаратних адрес* (МАС-адрес) відправник та отримувача даних.

*MAC-адреса* (Media Access Control) – унікальний шестибайтний номер, який надається кожній одиниці обладнання виробником.

Для контролю цілісності даних на канальному рівні застосовуються алгоритми контрольних сум. В протоколах цього рівня використовуються алгоритми завадостійкого кодування, як-от код Хемінга, код Рида-Соломона, та механізми запобігання колізій кадрів.

В програмуванні канальний рівень є драйвером мережевої плати.

Протоколами цього рівня є: ARCnet, Controller Area Network (CAN), IEEE 802.3 (Ethernet), Ethernet Automatic Protection Switching (EAPS), Frame Relay, High-Level Data Link Control (HDLC), IEEE 802.11 wireless LAN, Point-to-Point Protocol (PPP), Point-to-Point Protocol over Ethernet (PPPoE), Token ring, IEEE 802 MAC (Media Access Control) тощо.

На канальному рівні функціонують комутатори (switch), мости (bridge), мережеві карти комп'ютерів, точки доступу (access point).

# Фізичний рівень

Фізичний рівень є найнижчим в ієрархії рівнів БМВВС. Він забезпечує фізичне передавання цифрових даних. В протоколах фізичного рівня виробниками визначаються для обраних середовищ передавання тип модуляції, часові параметри, сигнальні рівні дискретних станів, тип з'єднань тощо.

Функції фізичного рівня реалізуються на всіх мережевих пристроях.

Протоколами фізичного рівня є: IEEE 802.15 (Bluetooth), IRDA, RS-232, EIA-422, RS-485, DSL, ISDN, 802.11 Wi-Fi, Etherloop, V.35 тощо. На цьому рівні використовують *мережеві концентратори* (hub, за ДСТУ ISO/IEC 7498-1:2004 – «габ»), повторювачі та конвертери сигналів.

Слід відзначити, що в повному обсязі БМВВС не була реалізована, і, очевидно, не буде реалізована ніколи. Сьогодні використовується тільки деяка підмножина БМВВС. Вважається, що модель занадто складна. Практичного вжитку набули більш прості та менш формалізовані моделі, як-то Open-system environment reference model (OSE RM) за ISO/IEC TR 14252 «Information technology. Guide to the POSIX Open System Environment (OSE)», регламенти мережі Інтернет комісії з питань діяльності Інтернет (IAB, Internet Activities Board) та Інженерної ради Інтернету (IETF, Internet Engineering Task Force) або спрощена чотирирівнева модель міністерства оборони США (DoD, Department of Defence of USA).

#### Контрольні запитання

- 1. Що таке мережа?
- 2. За якими ознаками класифікують комп'ютерні мережі?
- 3. Що називають архітектурою мережі?
- 4. Що таке протокол?
- 5. Що таке відкрита система?
- 6. Назвіть принципи базової еталонної моделі мережі.
- 7. Які рівні розрізняє базова еталонна модель мережі?
- 8. Опишіть алгоритм взаємодії рівнів базової еталонної моделі мережі.
- 9. Назвіть функції прикладного рівня.
- 10. Назвіть функції рівня відображення.
- 11. Які рівні називають базовими?
- 12.Що таке маршрутизатор?
- 13.Що таке таблиця маршрутизації?
- 14.Що таке порт?
- 15.Що таке МАС-адреса?
- 16.Що таке комутатор?

## 1.4. «Хмарні» технології

Соціологічне опитування в першій декаді 21-го століття в США показало, що 54% опитаних впевнені, що вони не користуються

«хмарними» технологіями (https://myblaze.ru/oblachnyie-tehnologii-chtotakoe-oblako-v-internete/).

Насправді ситуація є дещо іншою, майже всі користувачі Інтернету використовують «хмарні» технології. Застосунки Youtube, Netflix, IpTv, iTunes, Facebook, Telegramm, Viber, iDrive, Google disk, OneDrive, електронна пошта Gmail, Ukr.net, поштові клієнти The bat, Microsoft Outlook (Microsoft Exchange Online) тощо сьогодні працюють за «хмарними» технологіями.

Сучасна тенденція організації праці за концепцією «BYOD – Bring Your Own Device», фрілансінг роблять «хмарні» технології трендом розвитку IT-індустрії сьогодення.

Концепція віддалених обчислень в 1960-х pp. peaлiзовувалася за базі великих ЕОМ «mainframe». До них під'єднувалися термінали, які не мали власних обчислювальних ресурсів, та по суті являли собою пристрій введення-виведення (комплект монітора та клавіатури). Всі обчислення проводилися центральною ЕОМ. Машинні бюро надавали в оренду потужності серверів споживачам.

Вважається, що ідею «хмарних» обчислень висунули наприкінці 60-х років 20-го століття Джон Маккатрі та Джозеф Ліклайдер.

Джон Маккарті (John McCarthy) використовував термін «ресурсні» обчислення. Він вважав, що «обчислення будуть проводитися за допомогою суспільних допоміжних засобів на зразок комунальних послуг», «обчислювальні функції коли-небудь можуть бути організовані як комунальна система телефонії».

Джозеф Карл Робнетт Ліклайдер (J. C. R. Licklider) вважав, що будьякий користувач отримає доступ до програм та даних незалежно від місця свого знаходження.

Терміном «*хмарні обчислення*» називають технологію розподіленої обробки даних, в якій динамічно масштабовані комп'ютерні ресурси надаються користувачам як інтернет-сервіс.

Джорж Різ (Риз Дж. Облачные вычисления: пер. с англ. / Дж. Риз. – Санкт-Петербург : БХВ-Петербург, 2011. – 288 с.) визначив ознаки, за якими сервіс можна кваліфікувати як «хмарний»: «Якщо для доступу до сервісу ви можете зайти в будь-яку бібліотеку або інтернет-кафе, сісти за будь-який комп'ютер, без особливих вимог до операційної системи, браузера, та отримати доступ до сервісу – то такий сервіс є хмарним.»

Передумови появи «хмарних» обчислень склали досягнення в розробці комп'ютерних мереж та *віртуалізації* обчислень.

Після завершення проєкту ARPANET та передавання його нащадка – мережі NSFNET в комерційне користування з'явився Інтернет.

В 1972 р. фірма *IBM* запровадила технологію *віртуалізації*, яка дозволяла починати та завершувати роботу застосунку в будь-який момент незалежно від конкретного обладнання.

Першою появою терміну «хмарні» обчислення вважається доповідь *Compaq Computer* в 1996 р. на тему майбутнього Інтернету [10].

Першим випадком надання «хмарних» послуг вважають 1999 р., коли фірма *Salesforce* перевела систему взаємовідносин з клієнтами на інтернетсайт. Вперше була запропонована послуга користування, а не звичайного придбання програмного забезпечення. Фірма впровадила концепцію отримання клієнтами корпоративних CRM-застосунків через свій веб-сайт за підпискою (модель *SaaS*).

В 2002 р. книжковий онлайн-магазин *Атаzon* надав своїм клієнтам можливість зберігати дані в якості веб-сервісу *Атаzon Web Services* та проводити обчислення через *Атаzon Mechanical Turk*.

В 2006 р. *Атагоп* запустив сервіс *Elastic Computing Cloud (Amazon EC2)* для надання обчислювальних ресурсів в «хмарі» шляхом оренди комп'ютерів для запуску власних програм (модель *IaaS*).

В 2008 р. *Microsoft* презентувала «хмарну» технологію Azure Services Platform.

Ідея надання комп'ютерних потужностей в оренду *Google* почала займатися з 2008 р., коли вперше дозволила стартапам створювати власні застосунки на базі своєї розгалуженої мережі дата-центрів.

В 2008 р. OpenNebula NASA в проєкті Європейської комісії RESERVOIR стала першим ПЗ з відкритим кодом для розгортання приватних та гібридних «хмар» (https://ru.wikipedia.org/wiki/Облачные вычисления#cite\_note-\_24a75d224ad02435-1).

Офіційним вважається визначення «*хмарних*» обчислень Національного інституту стандартів та технологій США (NIST) як «моделі

надання можливості повсюдного та зручного мережевого доступу за вимогою одночасного до всіх або роздільного до виділених конфігурованих обчислювальних ресурсів (серверів, мереж передавання даних, пристроям зберігання даних, застосункам та сервісам), які можуть бути оперативно надані та вивільнені з мінімальними витратами або зверненнями до провайдерів» [11].

В Україні термінологія та характеристики «хмарних обчислень» визначено ДСТУ ISO/IEC 17788:2017.

ISO/IEC 17788 визначає «*хмарні обчислення* (cloud computing) як парадигму для надання можливості мережевого доступу до масштабованого еластичного пулу спільних фізичних та віртуальних ресурсів (сервери, операційні системи, мережі, програмне забезпечення, застосунки та обладнання для зберігання даних) з наданням самообслуговування та адмініструванням "за вимогою"» [12].

Термін «хмарні обчислення» запозичений з англійської мови. Англомовне походження має дві версії:

Перша версія говорить про аналогію за значенням з словом *cloud* (хмара) – група предметів, деталі яких не важливі для опису суті явища. Наприклад, cloud of dust, cloud of mosquito.

Друга версія говорить про аналогію із схематичним зображенням з'єднання серверів комп'ютерної мережі на діаграмах в США (рис. 1.60).



Рис. 1.60. Схема комп'ютерної мережі

Типовими способами використання «хмарних» технологій є розгортання веб-проєктів та аутсорсинг ІТ-структури.

Операторам інформаційних порталів, баз даних, інтернет-магазинів потрібна потужна та гнучка площадка для розгортання проєкту, в якості якої використовується «хмара».

Аутсорсинг IT-структури є спосіб для будь-яких компаній від виробничих до логістичних перевести IT-сервіси: веб-портал, пошту, CRM

(Customer Relationship Management), ERP (Enterprise Resource Planning) тощо, - в хмару.

За даними агенції Spiceworks (https://aif.ru/boostbook/oblachnyetekhnologii-i-reshenija.html) 75 % користувачів «хмар» використовують «хмарний» хостінг, 54 % розміщують пошту, 45 % використовують безпекові функції «хмари».

За даними агенції Forrester обсяг «публічних» «хмарних» обчислень з 2015 р. до 2020 р. зріс з \$91 млрд в 3 рази та досяг значення в \$235 млрд.

Найбільш активні на ринку «хмарних» обчислень компанії розширюють мережу великих центрів обробки даних (більше 100.000 серверів) по всьому світу (рис.1.61). Станом на 2020 р. компанія Google мала 19 центрів в США, 12 в Європі, один в Росії, один в Південній Америці, 3 в Азії.



Рис. 1.61. Центр в Чикаго площею більше 82 тис. м<sup>2</sup> (http://swsysweb.ru/cloud-computing-basic-concepts-problems.html)

Лідерами за кількістю ІТ-фахівців, які працюють в «хмарній» галузі вважають Китай, Індія, США та Індонезія, Бразилія, Японія, Німеччина, Великобританія, Мексика та Корея.

# До достоїнств «хмарних» обчислень слід віднести:

- доступ до інформації незалежно від типу пристрою та програмного забезпечення споживача;
- доступ до інформації незалежно від місця знаходження користувача;
- спільний доступ до інформації та колективна робота над документами;
- зниження вартості ПЗ на ТЗ. Для доступу до «хмарних» сервісів достатньо web-браузера;

- збільшення потужності комп'ютерів для розв'язання задач споживача. Кількість процесорів, об'єму оперативної пам'яті, дискового простору в «хмарних» системах теоретично нічим не обмежені;
- зниження вимог до кількості та кваліфікації ІТ-фахівців користувача;
- зниження ймовірності втрати інформації в разі виходу з ладу пристрою користувача, оскільки вона зберігається в «хмарі» і доступ до неї можна отримати з іншого пристрою;
- використання останньої версії ПЗ та відсутність необхідності постійного оновлення;
- оптимізація обміну інформацією;
- покращення сумісності форматів документів;
- економія часу та витрат поточних (ЗП обслуговування, енергія, площі) на виконання задач. Оплата тільки за використані ресурси.

За результатами опитування компанії Dell Technologies в 2020 р. 87 % респондентів заявили про скорочення часу випуску нових продуктів, 81 % — про зростання темпів впровадження інновацій (https://www.novostiitkanala.ru/news/detail.php?ID=149117).

# Недоліками «хмарних» обмежень є:

 можливе порушення конфіденційності. Безпринципний провайдер. Політика. Технічні несправності безпеки сервера. Крадіжка. Інформація користувача зберігається та аналізується провайдерами без його відому. Користувач не в змозі змінити або видалити інформацію або її частину.

Ще в 2010 р. керівник ІТ-видання Strategic News Service Марк Андерсон попереджав, що через значне зростання користувачів «хмарних» сервісів збільшується вартість помилок та витоків інформації, неминучі великі катастрофи типу виходу з ладу або пов'язані з безпекою [13].

B Google користувач не може видалити деякі групи даних FeedBurner, Google Friend Connect) [13].

2020 р. Twitter – витік даних. 2019 р. Sony Playground – витік даних 77 млн. користувачів. 2019 р. Dropbox – витік 25 млн. облікових записів. За 2019 р. в США компанія Privacy Rights Clearinghouse зареєструвала 175 випадків витоку даних 13 млн. записів (http://ieee.tpu.ru/another/oblako.html).

2020 р. в Інтернеті знайдено базу з 53 млн записів особистих даних громадян України (https://www.dw.com/uk/ didzhytalizatsiia-khaosu-chomu-v-ukraini-techut-personalni-dani/a-59192549).

2021 р. у мережі знайшли дані 533 мільйонів користувачів Facebook (https://www.dw.com/uk/ dani-533-milioniv-korystuvachiv-facebook-znaishly-u-merezhi/a-57096054).

 ризик масової втрати даних багатьма користувачами з причини технічного збою у провайдера.

В 2009 р. сервіс Magnolia втратив всі свої дані [13].

За даними Лабораторії Касперського 13 % компаній за 2016 рік стикалися з проблемою безпеки «хмари». При цьому третина компаній (32 %) втратила дані (https://www.tadviser.ru/index.php/ Информационная\_безопасность).

За даними компанії Symantec 40 % компаній стикалися з загрозою конфіденційності інформації, 25 % – з крадіжкою облікових записів, псуванням веб-власності. 43 % компаній втрачали дані в «хмарі», 68 % стикалися з проблемами її відновлення (https://www.tadviser.ru/index.php/ Статья:Экономика облачных сервисов).

- необхідність постійного широкосмугового з'єднання з Інтернетом;
- зниження можливостей налаштувань ПЗ;
- залежність користувача від форми системи оплати.

Атаzon за останні 10 років 51 разів знижував ціни, раптово почав їх збільшувати (https://www.forbes.ru/tekhnologii/internet-i-svyaz/327741-voina-tekhnologii-kak-google-i-microsoft-pytayutsya-dognat-amazo);

- Сукупні «хмарні» витрати далеко не завжди нижче класичних, що особливо проявляється при розширенні бізнесу.

За результатами опитування компанії Dell Technologies в 2020 р. тільки 5 % організацій отримали вигоду від впровадження «хмарних» технологій. (https://www.novostiitkanala.ru/ news/detail.php? ID=149117).

За даними Symantec з прихованими витратами при використанні «хмарних» технологій стикалися 77 % всіх компаній (83 % великих, 70 % малих) (https://www.tadviser.ru/index.php/ Статья:Экономика\_облачных\_сервисов#cite\_note-a-0).

- можливість закриття сервісу;

9 серпня 2020 р. влада Білорусії вимкнула Інтернет. Збитки оцінювалися в \$156 млн на день;

- залежність формату збереження даних пропозиціями провайдера;
- втрата свободи. Більшість сервісів не має чітких стандартів. За зміни провайдера можливі серйозні проблеми. Вони можуть виникнути також за умови зміни провайдером інтерфейсу доступу, тощо.
- важкість повернення до попередньої схеми роботи (повернення власних даних, розгортання власної інфраструктури, протидія провайдера).

За аналізом агенції Forrester біля 51 % користувачів невдоволені процесом міграції (on-boarding), з них 26 % відмітили великі часові витрати, 21 % – відсутність підтримки провайдера (https://www.tadviser.ru/index.php/Статья:Главные\_угрозы\_безопа сности\_в\_облаке#cite\_note-2).

## Різновиди «хмарних» обчислень

В [12] «хмарна» модель описана п'ятьма основними характеристиками, трьома моделями сервісів та чотирма моделями розгортання.

# Характеристики (ознаки) хмарних обчислень за визначенням NIST

1. Самообслуговування за вимогою (On-demand self-service). Користувач може самостійно без взаємодії з постачальником послуг змінювати потрібні йому обчислювальні потужності (серверний час, обсяг сховища тощо).

2. Універсальний (широкий, Broad Network Access) доступ в мережі. Доступ до обчислювальних можливостей забезпечується незалежно від місця знаходження, незалежно від характеристик термінальних засобів (тонких або товстих клієнтів) стандартними механізмами.

3. Об'єднання ресурсів постачальника в єдиний пул (Resorce pooling) для спільного використання розподілених ресурсів великою кількістю користувачів за багатоорендною моделлю (multi-tenant). Пули містять різноманітні фізичні та віртуальні ресурси, які можуть бути динамічно перепризначені.

*Мультиоренда (multi-tenancy*) – «розподіл фізичних або віртуальних ресурсів, таким чином, що кілька орендаторів та їхні обчислення та дані є ізольованими одне від одного та недоступними один одному» [15].

4. Еластичність (швидка масштабованість, Rapid elasticity). Обсяг послуг може змінюватися додаватися, виключатися, стискатися, розширюватися виходячи з потреб користувача.

5. Облік споживання (Measured service). Автоматична оцінка постачальником обсягу наданих послуг та білінг. Система керування автоматично контролює та оптимізує виділення ресурсів.

# Моделі розгортання «хмар» (deployment models)

За ISO/IEC 17788 «модель розгортання хмарних обчислень – це спосіб організації (cloud deployment model) хмарних обчислень, який ґрунтується на способі керуванні та спільного використання фізичних або віртуальних ресурсів».

Розрізняють наступні чотири типи розгортання:

Приватна «хмара» (Private cloud) – інфраструктура, призначена для використання в межах однієї організації. Інфраструктура може належати самій організації або зовнішньому провайдеру.

Це спосіб розгортання, коли служби «хмарних» обчислень (cloud service) використовуються виключно єдиним користувачем служби «хмарних» обчислень і ресурсами керує тій же користувач служби «хмарних» обчислень. Приватна «хмара» може належати, перебувати під керівництвом і управлятися самою організацією або третьою особою. Ресурси для функціонування «хмари» можуть перебувати як у власника, так і зовні. Споживач служби «хмарних» обчислень може також дозволити доступ іншим сторонам для власної вигоди. Приватні «хмари» прагнуть встановити суворо керований кордон навколо приватного «хмари», заснований на обмеженнях споживачів співробітниками єдиної організації [12].

*Суспільна «хмара»* (Community cloud) – інфраструктура, яка призначена для виключного використання окремою спільнотою, яка вирішує спільні задачі.

Суспільною вважається розгортання, де служби «хмарних» обчислень виключно підтримують визначене співтовариство споживачів служби «хмарних» обчислень та обслуговуються цим співтовариством та де ресурсами керує принаймні один з членів цього співтовариства. Споживачі служби «хмарних» обчислень, які входять в співтовариство, мають спільні вимоги та відносини один з одним. Суспільна «хмара» може належати, перебувати під керівництвом і управлятися однією або більше організаціями в співтоваристві, третьою особою або деякої їх комбінацією. Ресурси для функціонування «хмари» можуть належати спільноті або зовнішньому власнику. Суспільні «хмари» обмежують доступ до служби «хмарних» обчислень споживачами, у яких є спільний набір потреб і завдань, на відміну від відкритих публічних «хмар», при цьому в громадських «хмарах» бере участь більш широке коло споживачів, ніж в приватних «хмарах». Спільні потреби і завдання включають, зокрема, місію, вимоги інформаційної безпеки, політику і етику [12].

Публічна «хмара» (Public cloud) – інфраструктура, яка призначена для вільного використання широким колом споживачів. Знаходиться як правило у власності постачальника послуг.

Публічною є модель розгортання, в якій служби «хмарних» обчислень потенційно є «доступними будь-якому користувачеві служби «хмарних» обчислень, при цьому ресурсами керує постачальник служби «хмарних» обчислень. Публічна «хмара» може належати, перебувати під керівництвом і управлятися академічною, урядовою або бізнес-організацією, або будьякої ïχ комбінацією. Дане «хмара» функціонує на потужностях постачальника служби «хмарних» обчислень. Фактична доступність для «хмарних» обчислень може підлягати споживачів служби певних юридичному регламентування. Публічні «хмари» мають дуже широкі межі, де споживачеві служби «хмарних» обчислень надано доступ до послуг публічного «хмари» з небагатьма обмеженнями, якщо такі є [12].

*Гібридна «хмара» (*Hybrid cloud) – комбінація різних «хмарних» інфраструктур, які залишаються унікальними об'єктами, які пов'язані між собою стандартизованими або пропрієтарними технологіями для можливості обміну даними та застосунками.

Гібридна «хмара» – це модель розгортання «хмарних» обчислень, яка використовує принаймні дві різні моделі розгортання «хмарних» обчислень. Включені розгортання залишаються при цьому унікальними одиницями, але пов'язані відповідною технологією, яка забезпечує функціональну сумісність (інтероперабельність), переносимість даних і переносимість додатків. Гібридна «хмара» може належати, перебувати під керівництвом і управлятися самою організацією або третьою стороною, і може функціонувати на ресурсах організації або поза нею. Гібридні «хмари» представляють ситуації, в яких взаємодії між двома різними розгортання не є необхідними, але залишаються пов'язаними через відповідні технології. Межі, як такі, встановлені гібридною «хмарою», відображають два його основних розгортання [12].

# Моделі сервісів (типи можливостей) «хмар»

NIST визначає наступні три базові моделі служб «хмар» (Service Models): *платформа як послуга* (Paas, Platform as a Service), *інфраструктура як послуга* (IaaS, Infrastructure as a Service), *програмне забезпечення як послуга* (SaaS, Software as a Service, Application-as-a-Service).

Програмне забезпечення як послуга (SaaS, Software as a Service) – можливості споживачу використовувати сервіс надання програми постачальника, що працюють у хмарній інфраструктурі. Програми постачальника можуть бути доступні з різних клієнтських пристроїв через інтерфейс «тонкого» клієнта, як-то веб-браузер (наприклад, електронна пошта в Інтернеті), або інтерфейс програми. Споживач не керує і не контролює базову «хмарну» інфраструктуру, включаючи мережу, сервери, операційні системи, сховище або навіть можливості окремих додатків, за винятком обмежених налаштувань конфігурації програм, що стосуються користувача.

Платформа як послуга (Paas, Platform as a Service) – сервіс надання можливості споживачу розгортати в «хмарній» інфраструктурі створені або придбані споживачем програми, створені за допомогою мов програмування, бібліотек, служб та інструментів, які підтримує постачальник. Споживач не керує базовою «хмарною» інфраструктурою, включаючи мережу, і не контролює її. серверів, операційних систем або сховища, але має контроль над розгорнутими програмами та, можливо, налаштуваннями конфігурації середовища розміщення програм.

Інфраструктура як послуга (IaaS, Infrastructure as a Service) – сервіс надання споживачу доступу до обробки, зберігання, мереж та інших фундаментальних обчислювальних ресурсів, де споживач може розгортати та запускати довільне програмне забезпечення, яке може включати операційні системи та програми. Споживач не керує і не контролює базову «хмарну» інфраструктуру, але має контроль над операційними системами,

сховищем і розгорнутими застосунками і, можливо, обмежений контроль окремих мережевих компонентів (наприклад, брандмауерів хоста).

Слід відзначити, що з часу публікації первинного «хмарного» документа з'явилися моделі сервісів, які в своїй більшості є частинами трьох базових моделей.

З урахуванням змін більш сучасний ISO/IEC 17788 вводиться замість терміну *Service Models* два терміни: *тип можливостей* та *категорії служб* хмарних обчислень. Термін *тип можливостей* визначає типовий набір ресурсів сервісу, термін *категорії служб* – конкретні ресурси сервісу.

«*Тип можливостей* хмари (cloud capabilities type) – це класифікація функціональності, що надається службою хмарних обчислень (cloud service) користувачеві служби хмарних обчислень, яка ґрунтується на ресурсах, які вона використовує».

«Категорії служб хмарних обчислень (cloud service category) – групи служб хмарних обчислень, які мають набір спільних якостей. Одна категорія служб хмарних обчислень може містити можливості одного або кількох типів можливостей хмари».

Виділяються наступні три базові типи можливостей:

- тип можливостей застосунків (application capabilities type) тип можливостей «хмари», в якому користувач служби «хмарних» обчислень може використовувати за стосунки постачальника служби «хмарних» обчислень;
- тип можливостей платформи (platform capabilities type) тип можливостей «хмари», в якому користувач служби «хмарних» обчислень може встановлювати, керувати та запускати застосунки, які створені або придбані користувачем, використовуючи одну або декілька мов програмування та одне або кілька середовищ виконання, які підтримуються постачальником служби «хмарних» обчислень;
- тип можливостей інфраструктури (infrastructure capabilities type) – тип можливостей «хмари», в якому користувач служби «хмарних» обчислень може отримати та використовувати обчислювальні ресурси, ресурси для зберігання даних або мережеві ресурси.

Як видно, зміст визначення *типу можливостей* відповідає визначенню *моделі сервісу*.

Відмінності базових моделей «хмарних» сервісів пояснюються на рис. 1.62.



Рис. 1.62. Базові «хмарні» сервіси

# Категорії служб «хмарних» обчислень

ISO/IEC 17788 виділяє 13 категорій служб «хмарних» обчислень:

- Обчислення (CompaaS, Computing as a Service);
- Обмін інформацією (CaaS, Communication as a Service, Communication Platform as a Service);
- Зберігання даних (DSaaS, DataSaving as a Service, Data as a Service, StaaS, Storage as a Service);
- Інфраструктура (IaaS, Infrastructure as a Service);
- Mepeжa (NaaS, Network as a Service);
- Платформа (Paas, Platform as a Service);
- Програмне забезпечення (Software as a Service, SaaS);
- СУБД (Database as a Service);
- Робочий стіл (Desktop as a Service);
- Електронна пошта (Email as a Service);
- Ідентифікація (Identity as a Service);
- Управління (Management as a Service);
- Безпека (Security as a Service).

# Програмне забезпечення як послуга

За ISO/IEC 17788 програмне забезпечення як послуга (SaaS, Software as a Service, Application-as-a-Service) – «це категорія служб хмарних обчислень, в якій споживачеві служби хмарних обчислень надається тип можливостей хмари – тип можливостей застосунків».

Користувач використовує застосунки, які надає постачальник через веб-інтерфейс або API (Application programming interface). Застосунок запускається в хмарній структурі постачальника. Користувач не може керувати та контролювати параметри застосунку та інфраструктуру «хмари» (мережу, сервери, OC, сховища даних).

Основним споживачем *SaaS* є бізнес-користувачі малого та середнього бізнесу та приватні користувачі.

Більшість SaaS-застосунків призначені для підтримки взаємодії між співробітниками, які разом працюють над спільними задачами (Collaboration). Apxiteкtypa SaaS-застосунків є мультиорендною (Multi– tenant). Один екземпляр застосунку, який запускається на хмарному сервері, обслуговує багато користувачів, тобто, кожному споживачеві в процесі виконання задач надається свій екземпляр віртуального застосунку.

За схемою SaaS використовуються застосунки як-от:

- керування бізнесом Business Apps (IBM B2B Cloud Services, Axway Cloud B2B, amoCRM SaaS сервіс для B2B, Google Apps for Business, Oracle Business Intelligence Managed Cloud Service, Anaplan/ADE Professional Solutions, Brand Analytics тощо),
- офісної автоматизації Office Apps (Google Docs, Office365, Zoho Docs, IBM SmartCloud Docs тощо),
- послуги сервісу E-mail as Service;
- послуги сервісу Communications as Service;
- послуги сервісу Security as Service;
- тощо.

Серед сегментів застосунків ринку *SaaS* лідерство належить CRM (Customer Relationship Management) з 31 %, HRM (Human Resources Management), ERP (Enterprise Resource Planning), CCC (Communication Collaborating Content), офісним застосункам, створенню цифрового контенту (DCC, Digital Content Creating).

Визначними вендорами сервісу *SaaS* вважаються компанії Salesforce (Salesforce1 Sales Cloud, Salesforce CRM), Oracle (Oracle Cloud Applications: HR, CX, ERP, EMP, SCP, Business Intelligence), Google (Google Apps/Google Docs, Google Drive, Google Sites, Hangouts, Gmail, Google Calendar), IBM (IBM SmartCloud Docs), Microsoft (Dynamics CRM, Microsoft OneDrive, Office 365, Lync Online, Exchange Online, SharePoint Online), Zoho (Zoho Docs, Zoho Reports, Zoho Business Intelligence, Zoho CRM).

Популярними  $\epsilon$  «хмарні» застосунки Netflix, Photoshop.com, Acrobat.com, Intuit QuickBooks Online, IBM LotusLive, Matlab Cloud, Feng Office Community Edition, Simple Groupware, Zarafa, Autodesk Fusion, Dassault Systemes.

За дослідженням компанії Forrester в 2020 р. лідерами SaaS названо компанії Salesforce, Amazon, Alibaba, Microsoft, Adobe, Google. За даними агенції РАС безумовним лідером користувачів SaaS є США з 60 %, далі – Великобританія та Німеччина по 6 %, Японія та Канада — по 4 % (https://www.crn.com/news/cloud/microsoft-leads-101b-saas-market-followed-by-salesforce-adobe).

Достоїнствами SaaS є:

- відсутність необхідності придбання та інсталяції ПЗ.
- оновлення. ПЗ оновлюється постачальником.
- зниження вимого до апаратного забезпечення користувача.
- зниження витрат на придбання ПЗ та розширення ПЗ.
- швидке розгортання.
- зниження вимог до кваліфікації користувача.

Недоліки SaaS:

- обмеженість вибору ПЗ;
- проблеми з отриманням конкретних версій ПЗ;
- недостатні інтеграційні можливості;
- залежність швидкості від якості зв'язку та її зменшення порівняно з локальним використанням.

Частиною сервісу програмного забезпечення як послуги є однойменний сервіс пошуку як послуги SaaS (Search as a Service). Послуга є спеціалізованою службою IT-аутсорсінгу, яка проводить пошук за допомогою публічних сервісів, як-от Google, Bing та по корпоративним ресурсам компанії зі створенням внутрішньої бази пошукової індексації. Дії проводяться обчислювальними засобами провайдера.

## Платформа як послуга

За ISO/IEC 17788 *платформа як послуга* (Paas, Platform as a Service) – це «категорія служб хмарних обчислень, в якої споживачеві служби хмарних обчислень надається тип можливостей хмари: тип можливостей платформи».

Користувач замовляє та отримує програмну платформу: ОС, СУБД, власне або надане прикладне ПЗ, засоби налагодження ПЗ (IDE, Integrate Development Equipment, framework). Користувач не може керувати та контролювати параметри інфраструктури «хмари» (мережу, сервери, сховища даних), але може налаштовувати застосунки на деякі параметри середовища.

Більшість *PaaS*-платформ призначені для задоволення інтересів розробників ПЗ. Вони дозволяють створювати масштабовані вебзастосунки зі зменшеними витратами, коли задача є типовою та повинна бути розв'язана в найкоротший термін. За це доводиться розплачуватися свободою вибору технологій та контролю за низькорівневими компонентами системи.

*PaaS* надає інтегровану платформу для розроблення, тестування, розгортання та підтримки веб-застосунків як сервісів. Клієнт РааS використовує може налаштовувати надані програмні інструменти, бібліотеки провайдера. Різновиди РааS містять середовища для хостінгу, розроблення, тестування, розгортання застосунків, сервіси масштабування, керування версіями тощо.

Створення сайту Java «з нуля» може зайняти кілька місяців. Інструменти та готові компоненти з *PaaS*-платформ Spring або Tapestry дозволяють знизити часові витрати в кілька разів.

Зазвичай PaaS-сервіс містить програмну платформу та засоби її обслуговування в складі:

- мережевої ОС (Unix-системи, включаючи Ubuntu Server, BSD/OS Family, Solaris/SunOS, Microsoft Azure);
- СУБД (MySQL, Microsoft SQL, SQL Database, PostgreSQL, Oracle),
- програмне забезпечення для забезпечення взаємодії між різними застосунками, системами та компонентами (Middleware);
- прикладне програмне забезпечення, фреймворки, бібліотеки тощо (Software development and testing tools (IDE for: Python, Java, PHP, Ruby, JS);

- сервер застосунків для розробки, тестування, налагодження та запуску веб-застосунків (Application server).

Основними *PaaS* вендорами є:

- Amazon (AWS Elastic Beanstalk: Java, .NET, PHP, Node.js, Python, Ruby and Apache HTTP Server, Apache Tomcat, Passenger, IIS);
- IBM (IBM Bluemix: Liberty for Java, SDK for Node.js, Ruby on Rails, Ruby Sinatra);
- Microsoft (Microsoft Asure: ASP.NET, Java, PHP, Python, Django, Node.js, Azure SQL Database);
- Google (Google App Engine: Python, Java, PHP, MySQL).
  Корпоративна версія коштує біля \$50 за одного користувача за 500 Мб дискового простору та 5 млн переглядів в місяць;
- Salesforce (Salesforce1 Platform Cloud application development: Force.com, ExactTarget, Salesforce1 Mobile App, Salesforce Database.com), Heroku: Ruby, Java, Node.js, Scala, Clojure, Python, PHP, PostgreSQL);
- Oracle (Oracle Cloud Platform Services, Oracle Database Cloud Service, Oracle Java Cloud Service);
- VMware (Cloud Foundry: Java Spring, Ruby on Rails, Sinatra, NodeJS, .NET, MySQL Redis, MongoDB);
- OpenSource (Xen Cloud Platform, Apache Hadoop, Apache Hive).

Достоїнствами *PaaS* є:

- можливість невеликих розробників ПЗ (SMB) та фрілансерів створювати продукт з мінімізацією витрат на придбання сучасних програмних продуктів та проблем з ініціалізацією продуктів;
- надання єдиного середовища для створення програмних продуктів;
- доступ до IDE великої кількості територіально віддалених розробників;
- вбудовані функції обміну повідомленнями, групового спілкування;
- деталізована звітність по використанню програмних та апаратних ресурсів, спрощення адміністрування;
- автоматичне оновлення ПЗ;
- скорочення часу на розгортання ПЗ;
- зниження вимог до терміналів споживачів;
- захист інформації.

До недоліків *PaaS* можна віднести:

- меншу гнучкість та менший контроль в порівнянні з *IaaS*;
- зменшення швидкості в порівнянні з локальними системами;
- прив'язку до провайдера *PaaS*;
- обмеженість функціоналу ПЗ.

Частиною сервісу *платформи як послуги* є *тестування як послуга* (Testing-as-a-Service), яка надає можливість тестування локального або хмарного ПЗ з «хмари».

# Інфраструктура як послуга

За ISO/IEC 17788 *інфраструктура як послуга* (IaaS, Infrastructure as a Service) – це «категорія служб хмарних обчислень, в якій споживачеві служби хмарних обчислень надається тип можливостей хмари: тип можливостей інфраструктури».

В інфраструктурі як послузі споживач отримує можливість у формі віртуалізації керувати засобами обробки та зберігання, серверами та мережевою структурою, на яких він може самостійно встановлювати ОС та прикладне ПЗ. Користувач орендує обчислювальні можливості (серверний час, дисковий простір, пропускну спроможність мережевих каналів) або використовує аутсорсінгові послуги. Користувач не має доступу до керування фізичною інфраструктурою «хмари», але може розпоряджатися наданою віртуальною інфраструктурою.

Тобто споживачу надаються абстрактні еластичні обчислювальні ресурси. Наприклад, не фізичні сервери, а серверний час, не фізичні диски, а необхідний дисковий простір, на фізичний канал зв'язку, пропускна спроможність каналу.

*IaaS* – це сервіс корпоративного рівня для підприємств та установ різного масштабу, користувачами є служби ІТ підприємств. Сервіс може бути розміщений в приміщеннях споживача або в зовнішньому центрі обробки даних. IaaS може використовуватися для з'єднання локальних мереж споживача з мережею «хмари».

Засоби автоматизації сервісу забезпечують динамічний розподіл ресурсів «хмари» (зміну кількості віртуальних серверів, дискового простору для зберігання даних, пропускну спроможність каналів тощо).

Достоїнствами «хмарного» хостінгу є економічність, бо оплата проводиться тільки за фактично використані ресурси: кількість

процесорного часу, об'єму дискового простору тощо, та еластичність, бо проводиться автоматичне масштабування ресурсів в залежності від завантаження.

Основними *IaaS* вендорами вважаються: Amazon (Amazon EC2), IBM (IBM SmartCloud, SoftLayer IaaS), Microsoft (Azure Virtual Machines), Google (Google Compute Engine), HP (HP Cloud), Oracle (Oracle Cloud Infrastructure Services), VMware (open-source OpenStack), OpenSource (Eucalyptus, OpenNebula, OpenStack, Nimbus).

Слід відзначити значну активізацію в сфері *IaaS* «стільникових» кампаній, наприклад, платформа FusionSphere від Huawei.

Тенденцією ринку є інтеграція послуг *PaaS* та IaaS в єдиний сервіс хмарних інфраструктур та платформ (CIPS, Cloud Infrastructure and Platform Services).

За даними агенції Gartner (https://aws.amazon.com/resources/analystreports/gartner-mq-cips-2021) лідерами надання послуг IaaS є Amazon з ~40 % ринку, на другому місці – Microsoft з ~15 %. Відмічається, що п'ять головних постачальників IaaS займають 80 % ринку.

Частиною сервісу *інфраструктури* є послуга *надання технічних ресурсів* (HaaS, Hardware-as-a-service), коли споживачу надається віртуальний еквівалент сервера із замовленням об'єму RAM, об'єму та типу HDD, кількості ядер та частоті CPU тощо.

## Зберігання даних як послуга

Зберігання даних як послуга (DSaaS, DataSaving as Service, DaaS, Data as a Service, StaaS, Storage as a Service) – категорія служб «хмарних» обчислень, в якій споживачеві служби «хмарних» обчислень надаються можливості використання ресурсів для зберігання даних і пов'язані з цим можливості. DSaaS може забезпечити будь-який з трьох типів можливостей «хмари». З боку споживача «хмарні» диски виглядають як звичайні папки або диски.

Сервіс Dropbox був першим *DSaaS*. Він забезпечує шифрування даних SSL, AES, синхронізацію даних, керовані рівні доступу, інтеграцію з Microsoft Office 365. Споживачу надається безкоштовно 2 Гб, додаткові послуги є комерційними: план Plus (1 Тб, \$10 в місяць), Professional (2 Тб за \$20), Business (\$12,5 в місяць за 3 Тб).

Cepsic Microsoft OneDrive (Microsoft SkyDrive) є інтегрованим з Office 365. Безкоштовно надається 5 Гб з розширенням до 5 Тб за 339 грн в місяць.

Сервіс iCloud Drive забезпечує шифрування AES, SSL та інтеграцію з Аррle застосунками. Безкоштовно надається 5 Гб з розширенням до 2 ТБ за \$9,99 в місяць.

Google Cloud Storage, Google Drive. Сервіс Google Drive забезпечує інтеграцію з офісними «хмарними» за стосунками Google Docs. Безкоштовно надається 15 Гб з розширенням до 30 Тб за 1900 грн в місяць.

Amazon Simple Storage Service (Amazon S3) – онлайн веб-служба файлового хостінгу  $\epsilon$  найшвидшою серед Amazon S3, Windows Azure, Rackspace.

Оскільки важливим є запитання захисту даних, розповсюдженими додатковими послугами категорії DSaaS є *резервне копіювання як сервіс* та *відновлення даних як сервіс*. Послуги призначені для компаній для яких побудова власної надійної системи резервування та відновлення є нерентабельною.

Резервне копіювання як сервіс (BaaS,Backup as a Service) – послуга по наданню засобів для процедури резервного збереження даних в «хмарі». Наприклад, Commvault®, Acronis Infoprotect, Carbonite, VEEAM Backup & Replication.

Аварійне відновлення як послуга (DRaaS, Data-recovery-as-a-service, Disaster Restoring as a Service) – послуга відновлення працездатності «хмарної» структури споживача в разі аварії тощо. В разі виходу з ладу клієнтських сервісів, вони на протязі кількох хвилин автоматично відновлюються в «хмарі». Послуги призначені для компаній, для яких безперервність роботи сервісів є критичною.

## Обмін інформацією як послуга

За ISO/IEC 17788 обмін інформацією як послуга (CaaS, Communication as a Service, Communication Platform as a Service) – це є «категорія служб хмарних обчислень, в якій споживачеві служби хмарних обчислень надаються можливості взаємодії в реальному часу і спільна робота. *CaaS* може забезпечити тип можливостей застосунків та платформи». Сьогодні має назву «UCaaS, Unified Communications Platform as a Service».

CaaS є послугою розгортання в «хмарі» провайдера комунікаційної інфраструктури компанії споживача. Послуга призначена для споживачів,

які бажають отримати можливість мовного VoIP спілкування, обміну миттєвими повідомленнями, відеоконференцій без розгортання власних засобів.

*IP телефонія* (VoIP, Voice over IP). За такою технологією зв'язок встановлюється через Інтернет, Апаратна або віртуальна (Cloud PBX) АТС знаходиться у провайдера. За додатковим замовленням споживач може отримати доступ до адміністрування АТС. Реалізуються багатоканальні лінії, голосове меню, автоматична переадресація, маскінг номерів, віртуальні асистенти, автоматизовані речові повідомлення, колл-трекінг, інтегрувати телефонію з CRM. Наприклад, Vonage Business Solutions Cloud PBX.

*Інтерактивна мовна відповідь* (IVR, Interactive Voice Response) – технологія, яка дозволяє абоненту натисканням клавіш або засобами мовного розпізнавання отримати доступ до інформації, контролювати якість роботи персоналу.

*Миттєве передавання повідомлень* (IM, Immediately Messaging) реалізується корпоративними месенжерами з двохфакторною ідентифікацією, ботами автоматичної підтримки чатів відповідями. Наприклад, Ekiga, iLBC, Speex, Microsoft Lync Online, Mango Office.

Сервіс керування мобільними пристроями (MDM, Mobile Device Management) призначений для корпоративної мережі мобільних пристроїв. Агенти MDM на пристроях забезпечують централізоване налаштування та доступ до корпоративної мережі.

Сервіс відеоконференції Cisco WebEx, ZOOM, Google Meet, Microsoft Team.

Можливості додати функції комунікації, як-от SMS (Short Message Service), MMS (Multimedia Messaging Service), VoIP, Video Communication, сповіщення, (click-to-call), виклик одним дотиком багатофакторну ідентифікацію (MFA, multifactor authentication), безпосередньо В корпоративні застосунки споживача надає комунікаційна платформа як послуга.

Комунікаційна платформа як послуга (CPaaS, Communication Platform as a Service) є частиною PaaS, яка дозволяє розробникам ПЗ додавати функції комунікації «реального часу» в застосунки без необхідності побудови інфраструктури зв'язку та використання окремих каналів уніфікованих засобів наданням доступу до SDK (Software Development Kit) та бібліотек API (Application Programming Interface) постачальника. Комунікаційні сервіси також постачає провайдер.

# База даних як послуга

База даних як послуга (DBaaS, Database as a Service) – категорія служб «хмарних» обчислень, в якій споживачеві служби «хмарних» обчислень надаються можливості використання функціональності бази даних, СУБД якої розгорнута та обслуговується постачальником служби хмарних обчислень. Є різновидом *SaaS*.

# Робочий стіл як послуга

*Робочий стіл як послуга* (DaaS, Desktop as a Service) – категорія служб «хмарних» обчислень, в якій споживачеві служби «хмарних» обчислень надаються можливості сформувати, побудувати, керувати, зберігати та встановлювати свій робочий стіл дистанційно.

Достоїнствами послуги є суттєве зменшення вимог до ТЗ споживача. Для роботи достатньо «тонкого» клієнта, бо обчислення та зберігання даних відбувається на віртуальних «хмарних» серверах, та зручність адміністрування IT-служб.

Подібні функції виконує *послуга робочого місця* (WaaS, Workspace/Workplace as a Service) – надання комплекту *SaaS*, яке призначене для створення робочого оточення. На відміну від *DaaS* споживач отримує тільки ПЗ. Обчислення проводяться на обладнанні споживача. Може вважатися гібридом *SaaS* та *PaaS* з орієнтацією на офісну роботу.

# Електронна пошта як послуга

*Електронна пошта* як послуга (Email as a Service) – категорія служб хмарних обчислень, в якій споживачеві служби хмарних обчислень надаються можливості у вигляді повної поштової служби, разом з послугами підтримки, такими як зберігання, отримання, передавання, резервна копія та відновлення електронної пошти. Є різновидом *Saas*.

# Безпека як послуга

*Безпека як послуга* (SecaaS, Security as a Service) – категорія служб «хмарних обчислень», в якій споживачеві служби хмарних обчислень надаються сукупність служб IT-безпеки в інтегрованому вигляді з існуючою ОС, наданою постачальником. Послуга може містити ідентифікацію,

антивірус, антивірус/антишпгун, виявлення вторгнення та керування подіями безпеки.

Послуги SecaaS у вигляді «хмарних» антивірусів (Panda Cloud Antivirus, McAfee SaaS Email Protection & Continuity, Comfortway Mobile Security), *хмарної ідентифікації*, хмарного відеоспостереження (SpaceCam), моніторингу тощо, можуть надаватися поодинці або інтегрованим пакетом.

Додаткова послуга *моніторинг як сервіс* (MaaS, Monitoring-as-a-Service,) передбачає надання користувачеві програмного забезпечення для моніторингу власної інфраструктури та безпеки.

## Ідентичність як послуга

*Ідентичність як послуга* (IdaaS, Identity as a Service) – категорія служб «хмарних» обчислень, в якій споживачеві служби «хмарних» обчислень надаються можливості керування ідентифікацією та доступом до IT-засобів, за рахунок розширення та централізації існуючих в ОС. Вони містять надання та керування довідником та роботу служби єдиної точки входу. Є функцією послуг *SaaS*, *SecaaS*, *PaaS*.

Подібні функції виконує *криптографія як сервіс* CaaS (Cryptography as a Service). Послуга є частиною ПЗ *Paas*. Є набором мікросервісів у вигляді бібліотек для створення та перевірки електронного підпису з використанням файлового контейнера на «хмарному» сервері у відповідності до Європейської директиви CEN/TS 419 241 «Security Requirements for Trustworthy Systems Supporting Server Signing».

# Менеджмент як послуга

Менеджмент як послуга (Management/Governace-as-a-Service) – категорія служб «хмарних» обчислень, в якій споживачеві служби «хмарних» обчислень надаються можливості керування застосунками, конфігурацією, ресурсами, супроводженням (служба підтримки), керування портфелем проєкту, рівнем обслуговування тощо. Дає змогу керувати та визначати параметри «хмарних» сервісів: топологією, використанням ресурсів, віртуалізацією. Є частиною послуг WaaS, IaaS.

## Обчислення як послуга

*Обчислення як послуга* (CompaaS, Computing as a Service) – категорія служб «хмарних» обчислень, в якій споживачеві служби хмарних обчислень надаються можливості отримання і використання обчислювальних
ресурсів, необхідних для розгортання та виконання програмного забезпечення.

## Мережа як послуга

Мережа як послуга (NaaS, Network as a Service) – категорія служб «хмарних» обчислень, в якій споживачеві «служби» хмарних обчислень надаються можливості транспортної зв'язності і пов'язані з нею мережеві можливості. *NaaS* може забезпечити будь-який з трьох типів можливостей «хмари».

Провідні компанії надають комплексні «хмарні» послуги «all-in-one».

Компанія Amazon пропонує «хмару» Amazon Web Services (AWS), яка містить: Elastic Compute Cloud (Amazon EC2) – послугу Infrastructure as a Service з функцією Amazon Machine Image (AMI), Relational Database Service (Amazon RDS) – послугу Database as a Service, Elastic Kubernetes Service (Amazon EKS) – послугу Container as a Service, Simple Storage Service (Amazon S3) – послугу Storage as a Service.

Компанія Google пропонує наступні «хмарні» послуги: Google Compute Engine – послугу Computing as a Service, Google Cloud SQL – послугу DataBase as a Service, Google App Engine – послугу Platform as a Service та хостингу застосунків з 1 Гб дискового простору, по 10 Гб вхідного та вихідного трафіка в день, 200 міліонів гігациклів CPU в день тощо, Google Cloud Storage – послугу Storage as a Service, Google App – послугу Software as a Service.

Компанія Windows пропонує «хмару» Windows Azure на базі Windows Server 2008 та містить: Windows Azure Compute – послугу Compute as a Service, Windows Azure Storage – послугу Storage as a Service, SQL Azure – послугу Database as a Service, Windows Azure Fabric – послугу Platform as a Service з функціями моніторингу в «реальному» часі, розподілу обчислювальних потужностей, розгортання віртуальних серверів, балансування навантаження тощо, Net Services – послугу Platform as a Service, Access Control – послугу Identity as a Service.

Компанія Oracle пропонує публічну «хмару» Oracle Exalogic Elastic Cloud, ядром якої є СУБД ORACLE. Пропонується ПЗ Oracle VM, Oracle Middleware, Oracle WebLogic, Oracle Exalogic, WebCenter Page Composer, BI Composer, BPM Composer, Oracle BPM Suite, Oracle Data Integration, Oracle Identity тощо, апаратно-програмні комплекси Oracle Exadata. Компанія НР пропонує публічну «хмару» НР Cloud Compute на базі відкритого ПЗ OpenStack.

Не дивлячись на появу нових видів мережевих обчислень, як-от fog computing, edge computing, «хмарні» технології займають суттєву частину ІТ-ринку (рис. 1.63).



Рис. 1.63. Стан «хмарних» обчислень (https://tadviser.com/index.php/Article:ICT\_(Global\_Market)#The\_global\_ICT\_ market\_grew\_by\_9.25.2C\_to\_.244.24\_trillion)

## Контрольні запитання

- 1. Що називають «хмарними» обчисленнями?
- 2. Що вважають передумовами появи «хмарних» обчислень?
- 3. В чому полягають достоїнства «хмарних» обчислень?
- 4. Назвіть недоліки «хмарних» обчислень.
- 5. Назвіть ознаки «хмарних» обчислень.
- 6. Назвіть моделі розгортання «хмар».
- 7. Назвіть типи сервісів «хмар».
- 8. Що таке тип можливостей «хмари»?
- 9. Назвіть основні відмінності типів можливостей «хмари».
- 10.Назвіть категорії служб «хмари».

## 2. ОФІСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

*Офісною автоматизацією* називають «інтегрування офісної діяльності за допомогою систем обробки інформації» [3].

Переважний обсяг офісних даних займає *текст* – «дані у формі знаків, символів, слів, фраз, абзаців, речень, таблиць або інших символів, що мають на меті передати певний сенс та інтерпретація яких загалом ґрунтується на знаннях читача про певну природну чи штучну мови» [5].

Відповідно, офісні інформаційні системи в своїй більшості призначені для *опрацювання* тексту (text processing) *редагування* тексту (word processing) – «операцій з опрацювання текстових даних, такі як введення, редагування, сортування, злиття, завантаження, зберігання, відображення чи друкування» [5].

Типовими застосунками офісної автоматизації для оброблення тексту є текстові редактори (текстові процесори) та програми для роботи з електронними таблицями.

*Текстовий редактор* (text editor) - це «програмне забезпечення, яке дає змогу користувачеві створювати й переглядати текст» [5].

Процесор опрацювання текстів, текстовий процесор (text processor) процесор редагування тексту (word processor) – це «програмний пакет або пристрій з відповідним програмним забезпеченням, що дає змогу користувачеві виконувати опрацювання тексту» [5].

Програма для роботи з електронними таблицями, табличний процесор (Spreadsheet program) – це «програма, яка показує на дисплеї комірки таблиці, які впорядковані в стовпці та рядки, таким чином, що зміна вмісту однієї комірки може приводити до зміни вмісту однієї або кількох комірок у відповідності до визначених користувачем залежностей між комірками» [3].

Офісні інформаційні технології реалізуються в бізнес середовищі засобами офісних пакетів. Офісні пакети програм (Office suite) призначені для створення різноманітних електронних документів звітного, презентаційного, ілюстративного, освітнього напряму тощо.

На поточний час стандартною вважається комплектація офісного пакету в складі засобів розроблення текстових документів – *текстового редактора* (Word processor) з можливостями *формульного редактора* (Formula editor) та текстового *редактора нотаток* (Notetaking software),

*табличного процесора* (Spreadsheet processor), програми створення *презентацій* (Presentation program).

Додатковими компонентами офісних пакетів є засоби розроблення *ілюстративної графіки* – універсальні програми растрової, векторної графіки, засоби розроблення *технічної графіки* (Diagramming software), програми комп'ютерного верстання (Desktop publishing software), програми управління проєктами (Project management software), програми роботи з базами даних (системи управління базами даних, Database software), програми електронного зв'язку (Communication software, E-mail client) тощо.

Незважаючи на те, що на ринку пропонується велика кількість комерційних та безкоштовних продуктів (*Open Office, Libre Office, iWork, Google Docs* тощо), з моменту своєї появи і до сьогодні одним з лідерів серед офісних застосунків вважається пакет програм *Microsoft Office*.

## 2.1. Правила розроблення електронних документів

Процеси наукових та науково-технічних досліджень, проєктування, готування та власне виробництва, здобуття освіти супроводжуються необхідністю оформлення значної кількості різноманітних текстових документів. В науковій галузі такими є науково-технічні звіти, протоколи дослідів, тези та статті, монографії тощо. В технічній галузі такими є технічні пропозиції, записки до проєктів, формуляри, паспорти, інструкції, інша виробнича документація. Здобувачі освіти оформлюють звіти з лабораторних робіт та практики, реферати, курсові роботи та проєкти, випускні дисертації тощо.

Текстові наукові, науково-технічні, технічні документи оформлюються відповідно до певних вимог. Вимоги до оформлення часто називають форматом документа. Складовими вимог до документу або компонентами формату документа є вимоги до способу збереження, вимоги до структури документа, вимоги до оформлення документа.

Типовими форматами збереження текстової інформації є формат текстового процесора *MS Word* у вигляді файлів з розширення .*doc*, .*docx* та портативний формат *Adobe PDF* у вигляді файлів з розширенням .*pdf*.

В оформленні документів регламентуються розмір та орієнтація аркушів, розміри берегів аркуша, спосіб нумерації сторінок документа. Розмір, колір та гарнітура шрифтів тексту, відстані між рядками абзаців та між абзацами, розташування слів в рядках та рядків в абзаці, правила написання формул, застосування розмірностей, цитувань та списки використаних джерел тощо створюють стиль оформлення документа.

## Проєктна текстова документація

В Україні правила оформлення для проєктної текстової документації: пояснювальних записок, розрахунків, пропозицій, інструкцій тощо регламентуються ГОСТ 2.105-95, ГОСТ 2.106-96, ДСТУ ГОСТ 2.104:2006, ДСТУ ЕN ISO 7200:2005.

Курсові та дипломні проєкти оформлюються згідно з вимогами відповідних галузевих стандартів до проєктно-конструкторської та проєктно-технологічної документації, тобто ГОСТ 2.105-95, ГОСТ 2.106-96, ДСТУ ГОСТ 2.104:2006 тощо.

Проєктна текстова документація повинна виконуватися на аркушах стандартних форматів за ДСТУ ISO 5457:2006 «Документація технічна на вироби. Кресленики. Розміри та формати». На аркуші повинні міститися основний напис за ДСТУ ГОСТ 2.104:2006 у формі 2 (рис. 2.1) для першого аркуша, у форматі 2а (рис. 2.2) – для всіх подальших.



Рис. 2.1. Форма 2 основного напису [14]



Рис. 2.2 Форма 2а основного напису [14]

В Україні сьогодні одночасно чинні два нормативні документи, які регламентують вигляд основного напису: ДСТУ ГОСТ 2.104:2006 та ДСТУ ЕN ISO 7200:2005. Вибір форми основного напису проводиться виконавцем.

ГОСТ 2.105-95 визначає, «що колір текстів має бути чорним, розмір символів не менше за 2.5 мм, горизонтальна відстань від тексту до рамки не менше за 3 мм, вертикальна – 10 мм, абзац – 5 символів (15-17) мм».

В текстових редакторах розмір шрифтів називається *кеглем* та визначається в поінтах (*pt*). Один поінт дорівнює одній сімдесят другій дюйма. Розмір символів в 2.5 мм відповідає кеглю 8.

«На першому аркуші розташовують зміст документа. Назва "Зміст" пишеться посередині рядка з прописної літери. Пункти змісту пишуться з прописної літери.

Розділи документа мають позначатися арабськими цифрами без крапки в кінці. Номери підрозділів мають складатися з номеру розділу та номеру підрозділу, які розділені крапкою. Розташовуються назви розділів та підрозділів з абзацного відступу. Назви розділів пишуться з прописної літери без крапки в кінці. Відстань між назвою та текстом має складати 3-4 інтервали (15 мм). Відстань між назвами розділів та підрозділів – 2 інтервали (8 мм).

Переліки (списки) без посилань на них починаються з дефіса, переліки з посиланнями – з малої літери з дужкою. Розташовуються переліки з абзацного відступу. Підпереліки позначаються арабськими цифрами з дужкою з додатковим абзацним відступом.

Фізичні розмірності пишуться через пробіл після значення. Для діапазонів значень розмірність пишеться після останнього значення.

Роз'яснення формул проводиться з нового рядка без абзацного відступу зі слова "де" без двокрапки. Послідовність роз'яснень проводиться в порядку написання позначень в формулі через крапку з комою.

Декілька формул в рядку розділяються комою.

Формули нумеруються по правому краю рядка арабськими цифрами в круглих дужках.

Примітки записуються після тексту з абзацного відступу. Одна примітка пишеться через тире, кілька – нумеруються арабськими цифрами без крапки.

Ілюстрації підписують "Рисунок 1" з наскрізним номером арабським цифрами. За необхідності ілюстрації можуть мати пояснювальний напис: "Рисунок 1 – Схема".

Таблиці позначаються аналогічно ілюстраціям: "Таблиця 1 – Перелік". Заголовки граф та рядків таблиці пишуть з прописної літери в однині без крапки. Мінімальна висота рядка таблиці – 8 мм. В разі продовження таблиці на кілька аркушів тільки на першому позначають "Таблиця 1". Дозволяється на інших аркушах писати "Продовження таблиці 1".

Виноски розташовуються з абзацного відступу наприкінці сторінки з відділенням короткою горизонтальною лінією від тексту. Позначка виноски ставиться в тексті арабською цифрою з дужкою або символом "\*" надрядково.

Перелік спеціальних термінів з відповідними роз'ясненнями наводиться в кінці документа.

Перелік літератури повинен міститися на останньому аркуші документа. Оформлення списку та посилань повинно проводитись відповідно до ГОСТ 7.32» [15].

Слід зазначити, що в Україні ГОСТ 7.32-2001 не є чинним. Його положення замінені на положення ДСТУ ГОСТ 7.1:2006.

ГОСТ 2.105-95 не містить вимог щодо оформлення цитувань, місця розташування в рядку ілюстрацій, таблиць, пояснювальних написів.

#### Наукова та науково-технічна текстова документація

В Україні правила оформлення документації в усіх сферах наукової діяльності, зокрема під час оформлення дисертацій, складання звітів за різними напрямами науково-технічної діяльності, посібників, підручників тощо регламентуються ДСТУ 3008:2015 та ДСТУ 8302:2015.

Магістерські роботи, реферати, звіти тощо оформлюються відповідно до ДСТУ 3008:2015, який висуває наступні вимоги:

«Текст друкують шрифтом *Times New Roman* чорного кольору прямого накреслення через півтора-два міжрядкові інтервали кеглем 14. Розмір шрифту для написання заголовків у рядках і колонках таблиць і пояснювальних даних на рисунках і в таблицях встановлює виконавець.

Рекомендовано на сторінках використовувати береги такої ширини: верхній і нижній – не менше ніж 20 мм, лівий – не менше ніж 25 мм, правий – не менше ніж 10 мм. Абзацний відступ має дорівнювати п'яти знакам.

Заголовки структурних елементів звіту заголовки розділів треба друкувати з абзацного відступу або посередині рядка великими літерами напівжирним шрифтом без крапки в кінці.

Заголовки підрозділів, пунктів і підпунктів звіту потрібно друкувати з абзацного відступу з великої літери без крапки в кінці. Відстань між заголовком, приміткою, прикладом і подальшим або попереднім текстом має бути не менше ніж два міжрядкових інтервали.

Сторінки звіту нумерують наскрізно арабськими цифрами. Номер сторінки проставляють праворуч у верхньому куті сторінки без крапки в кінці.

Розділи, підрозділи, пункти, підпункти нумерують арабськими цифрами без крапки, починаючи з цифри "1".

Підрозділи як складові частини розділу нумерують у межах кожного розділу окремо. Номер підрозділу складається з номера відповідного розділу та номера підрозділу, відокремлених крапкою. Після номера підрозділу крапку не ставлять.

Пункти нумерують арабськими цифрами в межах кожного розділу або підрозділу. Номер пункту складається з номера розділу та порядкового номера пункту, або з номера розділу, порядкового номера підрозділу та порядкового

номера пункту, які відокремлюють крапкою. Після номера пункту крапку не ставлять.

Усі графічні матеріали звіту (ескізи, діаграми, графіки, схеми, фотографії, рисунки, кресленики тощо) повинні мати однаковий підпис "Рисунок". Рисунок подають одразу після тексту, де вперше посилаються на нього, або якнайближче до нього на наступній сторінці, а за потреби – в додатках. Якщо рисунки створені не автором звіту, подаючи їх у звіті, треба дотримуватися вимог чинного законодавства України про авторське право. Рисунки нумерують наскрізно арабськими цифрами, крім рисунків у додатках.

Дозволено рисунки нумерувати в межах кожного розділу. "Рисунок 3.2" – другий рисунок третього розділу.

Назву рисунка друкують з великої літери та розміщують під ним посередині рядка, наприклад, "Рисунок 2.1 – Схема устатковання". Назва рисунка має відображати його зміст, бути конкретною та стислою. Якщо з тексту зрозуміло зміст рисунка, його назву можна не наводити.

За потреби пояснювальні дані до рисунка подають безпосередньо після графічного матеріалу перед назвою рисунка.

Таблицю подають безпосередньо після тексту, у якому її згадано вперше, або на наступній сторінці. На кожну таблицю має бути посилання в тексті звіту із зазначенням її номера. Таблиці нумерують наскрізно арабськими цифрами, крім таблиць у додатках. Дозволено таблиці нумерувати в межах розділу. У цьому разі номер таблиці складається з номера розділу та порядкового номера таблиці, відокремлених крапкою, наприклад, "Таблиця 2.1" – перша таблиця другого розділу. Назву таблиці друкують з великої літери і розміщують над таблицею з абзацного відступу. Якщо з тексту можна зрозуміти зміст таблиці, її назву можна не наводити. У разі поділу таблиці на частини дозволено її головку чи боковик заміняти відповідно номерами колонок або рядків, нумеруючи їх арабськими цифрами в першій частині таблиці. Слово "Таблиця" подають лише один раз над першою частиною таблиці. Над іншими частинами таблиці з абзацного відступу друкують "Продовження таблиці" або "Кінець таблиці" без повторення її назви. Заголовки та самостійні підзаголовки колонок таблиці починають з великої літери.

Перед переліком ставлять двокрапку (крім пояснювальних переліків на рисунках). Якщо подають переліки одного рівня підпорядкованості, на які немає посилань, то перед кожним із переліків ставлять знак "тире". Якщо в

тексті є посилання на переліки, підпорядкованість позначають малими літерами української абетки, далі – арабськими цифрами, далі – через знаки "тире". Після цифри або літери певної позиції переліку ставлять круглу дужку. Дозволено користуватися можливостями текстових редакторів автоматичного створення нумерації переліків (наприклад, цифра-літера-тире). Текст кожної позиції переліку треба починати з малої літери з абзацного відступу відносно попереднього рівня підпорядкованості.

Примітки подають безпосередньо за текстом, під рисунком (перед його назвою), під основною частиною таблиці (у її межах). Одну примітку не нумерують. Слово "Примітка" друкують кеглем 12 через один міжрядковий інтервал з абзацного відступу з великої літери з крапкою в кінці. У тому самому рядку через проміжок з великої літери друкують текст примітки тим самим шрифтом. Якщо приміток дві та більше, їх подають після тексту, якого вони стосуються і нумерують арабськими цифрами.

Найвище та найнижче розташування запису формул(и) та/чи рівняння(нь) має бути на відстані не менше ніж один рядок від попереднього й наступного тексту. Нумерують лише ті формули та/чи рівняння, на які є посилання в тексті. Формули та рівняння треба нумерувати наскрізно арабськими цифрами. Дозволено їх нумерувати в межах кожного розділу. Номер формули чи рівняння друкують на їхньому рівні праворуч у крайньому положенні в круглих дужках, наприклад (3). У багаторядкових формулах або рівняннях їхній номер проставляють на рівні останнього рядка. Пояснення познак, які входять до формули чи рівняння, треба подавати безпосередньо під формулою або рівнянням у тій послідовності, у якій їх наведено у формулі або рівнянні. Пояснення познак треба подавати без абзацного відступу з нового рядка, починаючи зі слова "д" без двокрапки. Познаки, яки встановлюють визначення чи пояснення, рекомендовано вирівнювати у вертикальному напрямку.

Виноски позначають над рядком арабськими цифрами з круглою дужкою. Виноски нумерують у межах кожної сторінки. Дозволено виноску позначати зірочкою (\*). Дозволено на одній сторінці тексту застосовувати не більше ніж чотири виноски. Знак виноски проставляють безпосередньо після слова, числа, символу або речення, до якого дають пояснення. Цей самий знак ставлять і перед поясню вальним текстом. Пояснювальний текст виноски пишуть з абзацного відступу у тексті – у кінці сторінки, на якій зазначено виноску; у таблиці – під основною частиною таблиці, але в її межах. Виноску відокремлюють від основного тексту звіту чи таблиці тонкою горизонтальною лінією завдовжки від 30 мм до 40 мм з лівого берега. Текст виноски друкують кеглем 12 через один міжрядковий інтервал.

Фізичні формули подають з обов'язковим записом у поясненні познаки одиниці виміру відповідної фізичної величини. Між останньою цифрою та одиницею виміру залишають проміжок (крім позначення одиниць плоского кута – кутових градусів, кутових хвилини і секунд, які пишуть безпосередньо біля числа вгорі). У формулах верхні та нижні індекси, показники степеня мають бути меншими за букву чи символ, якого вони стосуються. Переносити формули чи рівняння на наступний рядок дозволено лише на знаках виконуваних операцій, які пишуть у кінці попереднього рядка та на початку наступного. У разі перенесення формули чи рівняння на знакові операції множення застосовують знак «х». Перенесення на знаку ділення ":" слід уникати.

Числові значення величин з допусками наводять так:  $(65 \pm 3)$  %; 80 мм  $\pm 2$  мм або  $(80 \pm 2)$  мм. Діапазон чисел фізичних величин наводять, використовуючи прикметники "від" і "до": Від 1 мм до 5 мм (а не від 1 до 5 мм).

Якщо треба зазначити два чи три виміри, їх подають так: 80 мм х 25 мм х 50 мм (а не 80 х 25 х 50 мм)» [17].

Зазвичай, кожне видавництво висуває свої власні вимоги до наукових матеріалів для їх публікації. Наприклад, видання «Наукові вісті КПІ» вимагає наступне форматування:

«Параметри сторінки: відступ зверху і знизу – 2 см, зліва – 2,5 см, справа – 1,5 см. Нумерація сторінок – у правому верхньому куті.

Текст набирається в текстовому редакторі *MS WORD* з дотриманням таких вимог: шрифт *Times New Roman Cyr* (розмір кегля – 14). УДК – прописними літерами, назва – великими напівжирними літерами. Друк – через один інтервал з абзацу, автоматичні переноси не застосовувати, вирівнювати по лівому краю, абзац починати за допомогою клавіші "TAB". Авторські акценти виділені курсивом, а не підкресленням.

Формули набираються в редакторі формул *MathType*. Дужки – (); [], {} – набирати з клавіатури. Дужки з шаблону редактора формул використовуються тільки для дробів. Ширина надрукованої формули не перевищує 8 см. Розмір формул – 100 %. Рисунки мають бути виконані за допомогою графічного редактора WORD, CorelDRAW будь-яких версій, VIZIO, EXCEL. Розміри рисунків по горизонталі – 82 або 170 мм. Кожний рисунок повинен мати підпис. Товщина контурних ліній – 0,8-1,0 мм, допоміжних – 0,5 мм, масштабної сітки – 0,3 мм. Розмір таблиць по горизонталі – 85 або 170 мм. Максимальний розмір по горизонталі – 210 мм. Кожна таблиця повинна мати заголовок.

Література подається двома списками. Перший оформлюється згідно з ГОСТ 7.1-84, другий – згідно з IEEE Citation Reference» (http://scinews.kpi.Ua/about/submissions#authorGuidelines).

## Правила цитування

У чинному Законі України «Про вищу освіту» академічний плагіат розглядається як порушення академічної доброчесності. За цим законом «академічний плагіат – це оприлюднення (частково або повністю) наукових результатів, отриманих іншими особами, як результатів власного дослідження та/або відтворення опублікованих текстів інших авторів без відповідного посилання». Тому важливим є безумовне вказання джерел матеріалів, які не є результатом авторських досліджень.

Цитування джерел проводиться застосуванням бібліографічних посилань на них. Правила оформлення бібліографічних посилань в Україні регламентують ДСТУ ГОСТ 8302:2006 та ДСТУ 3008:2015.

ДСТУ 3008:2015 вказує, що «посилання на джерело інформації, наведене в переліку джерел посилання, рекомендовано подавати так: номер у квадратних дужках, за яким це джерело зазначено в переліку джерел посилання, наприклад, "у роботах [2]-[3]". Дозволено наводити посилання на джерела інформації у виносках. У цьому разі оформлення посилання має відповідати його бібліографічному опису за переліком посилань із зазначеного номера.

Цитата в тексті "... тільки 36 % респондентів відносять процес створення інформаційного суспільства до пріоритетних" [3] <sup>1</sup>.

Посилання внизу сторінки:

<sup>1)</sup>[3] Пархоменко, В. Д. Інформаційна аналітика у сфері науковотехнічної діяльності : Монографія / В. Д. Пархоменко, О. В. Пархоменко. – К. : УкрІНТЕІ, 2006. – 224 с.

Відповідний опис у переліку джерел посилання:

3. Пархоменко, В. Д. Інформаційна аналітика у сфері науково-технічної діяльності : Монографія / В. Д. Пархоменко, О. В. Пархоменко. – К. : УкрІНТЕІ, 2006. – 224 с.

Перелік джерел, на які є посилання в тексті, наводять у кінці документа перед додатками на наступній сторінці. У переліку джерел посилання бібліографічні описи подають у порядку, за яким джерела вперше згадують у тексті. Порядкові номери бібліографічних описів у переліку джерел мають відповідати посиланням на них у тексті звіту (номерні посилання). Бібліографічні описи джерел у переліку наводять згідно з ДСТУ ГОСТ 7.1:2006» [17].

ДСТУ 8302:2006 визначає, що «за складом елементів бібліографічного запису розрізняють повне та коротке бібліографічне посилання. Повне бібліографічне обов'язкові посилання містить yci елементи, ЩО використовують для загальної характеристики, ідентифікування й пошуку об'єкта посилання. Коротке бібліографічне посилання містить частину обов'язкових елементів, які використовують тільки для пошуку об'єкта посилання. В більшості міжнародних стилів цитування коротке посилання містить пару "Прізвище – Дата". Повне бібліографічне посилання може містити такі елементи: заголовок бібліографічного запису (ім'я автора); основну назву документа; відомості про відповідальність (містять інформацію про осіб і/або організації, які брали участь у створенні документа); відомості про повторність видання (містять інформацію про зміни й особливості цього видання відносно попереднього); вихідні дані (містять відомості про місце видання (випуску), видавця та рік випуску документа); позначення та порядковий номер тому, номера чи випуску документа, якщо є посилання на твір або публікації з багаточастинного (багатотомного чи серіального) документа; відомості про обсяг (кількість сторінок) документа (у разі посилання на нього загалом); назву документа (журналу, збірника, газети тощо), в якому опубліковано об'єкт посилання (наприклад, статтю); відомості про місцезнаходження об'єкта посилання — номер сторінки в документі (у разі посилання на його частину); примітки (у посиланні на електронний ресурс, депоновану наукову роботу тощо).

За місцем розташування в документі розрізняють такі бібліографічні посилання: внутрішньотекстове; підрядкове; позатекстове.

*Внутрішньотекстове* бібліографічне посилання застосовують, якщо значну частину відомостей про об'єкт посилання внесено до тексту документа.

Внутрішньотекстове бібліографічне посилання розміщують безпосередньо в тексті документа. Внутрішньотекстове бібліографічне посилання подають у круглих дужках. Знак "крапка й тире" (".–") у внутрішньотекстовому бібліографічному посиланні замінюють знаком "крапка".

Внутрішньотекстове посилання може бути у повній або короткій формі (якщо частину відомостей про джерело цитування подано в тексті). Рекомендовано складати внутрішньотекстове бібліографічне посилання в короткій формі.

Підрядкове бібліографічне посилання на джерела інформації використовують за умов, якщо всередині тексту документа його розмістити неможливо або небажано, щоб не переобтяжувати текст та не ускладнювати його читання. Підрядкове бібліографічне посилання розміщують як примітку в нижній частині сторінки (полоси набору), відмежовуючи від основного тексту горизонтальною рискою. Підрядкове бібліографічне посилання пов'язують із текстом документа за допомогою знаків виноски, які подають на верхній лінії шрифту після відповідного фрагмента в тексті (наприклад: Текст <sup>29</sup>) та перед підрядковим посиланням (наприклад: <sup>29</sup>Посилання). Знаки виноски відокремлюють від тексту проміжком. Під час нумерування кількох підрядкових бібліографічних посилань можна застосовувати наскрізне нумерування в межах усього документа чи в межах його окремої глави (розділу, частини тощо) або нумерування в межах певної сторінки тексту (арабськими цифрами).

Підрядкове бібліографічне посилання можна наводити у повній або короткій формі.

У підрядковому бібліографічному посиланні на електронний ресурс віддаленого доступу за наявності в тексті бібліографічних відомостей, що його ідентифікують, дозволено зазначати тільки електронну адресу, використовуючи замість слів "Режим доступу" абревіатуру "URI" або "URL" (відповідно до 7.4.4.4).

У тексті: Маніфест ІФЛА про Internet: прийнято Сесією Ради ІФЛА 23 серпня 2002 р. / пер. з англ. В. С. Пашкова \*.

У підрядковому посиланні: \*URL: http://archive.ifla.org/III/misc/imua.pdf. (дата звернення: 15.09.2002).

Позатекстове бібліографічне посилання використовують переважно у наукових виданнях у разі багаторазових посилань на одні й ті самі документи задля уникнення повторного подання однакових бібліографічних записів або через їхню велику кількість, або за браком місця для підрядкових посилань.

Позатекстові бібліографічні посилання нумерують у межах усього документа або в межах окремих глав, розділів, частин тощо, застосовуючи наскрізне нумерування (арабськими цифрами).

Позатекстові бібліографічні посилання наводять як перелік бібліографічних записів і розміщують наприкінці основного тексту (або після заключної статті, післямови, коментарів – за їх наявності) документа або його складника (зазначаючи, наприклад, "Список бібліографічних посилань").

Сукупність позатекстових бібліографічних посилань, оформлених як перелік бібліографічних записів, не можна вважати бібліографічним списком (списком використаної літератури) чи покажчиком, що мають самостійне значення як бібліографічні посібники. Проте текст документа можна пов'язувати знаками виноски з бібліографічним списком використаної літератури.

Позатекстове бібліографічне посилання пов'язують із фрагментом тексту документа, до якого воно належить, за допомогою знаків виноски, які або виносять на верхню лінію шрифту після відповідного тексту та перед позатекстовим посиланням, або складають в одну лінію зі шрифтом основного тексту (у квадратних дужках у тексті та без дужок перед позатекстовим посиланням)» [22].

У 2016 році Міністерство освіти та науки України рекомендувало для використання в наукових публікаціях разом із правилами ДСТУ 8302:2015 перелік з 9-ти міжнародних стилів, які відображають всі галузі наукових досліджень: MLA (Modern Language Association), APA (American Psychological Association), Chicago/Turabianstyle, Harvard, ACS (American Chemical Society), AIP (American Institute of Physics), IEEE (Institute of Electrical and Electronics Engineers), Vancouver, OSCOLA.

Рекомендується застосовувати: *MLA* для гуманітарних галузей, *APA* для соціальних галузей, *Chicago/Turabianstyle*, *Harvard* для наукових. мистецьких, бізнесових публікацій, *ACS* для публікацій з хімії, *AIP* для публікацій з фізики, *IEEE* для інженерних, електротехнічних галузей.

Слід зауважити, що вищезгадані міжнародні стилі регламентують не тільки оформлення цитувань, але й всіх параметрів документів.

## WEB засоби оформлення посилань

Для спрощення роботи з бібліотекою джерел на оформлення цитувань та списків літератури можуть застосовуватися спеціальні програми – *бібліографічні менеджери* та *генератори посилань*.

http://www.bibme.org – безкоштовний онлайн сервіс. Дозволяє створити окремі цитування та бібліографію за стилями MLA, APA, Chicago/Turabian.

**Zotero** www.zotero.org – локальним клієнт-застосунок з можливістю хмарного зберігання документів. Працює зі стилями ААА, APA, Chicago, IEEE, Harvard, MLA, Vancouver. Безкоштовний обсяг 300 Мб. Дозволяє експорт в MS Word, OpenOffice, Google.

**Сіte This For Me** www.citethisforme.com – безкоштовний онлайн сервіс автоматичної та ручної роботи зі стилями *AMA*, *MLA*, *Harvard*. Має пошук джерел за назвою.

# Mendeley

ини www.mendeley.com – безкоштовний в межах 2Гб обсягу бібліографічний менеджер та академічна соціальна мережа. Має пошук авторів, які цитують такі ж джерела.

refmewebsite-ui.herokuapp.com – безкоштовний онлайн менеджер для Windows, iOS, Android. Має можливість ідентифікації джерел за штрих-кодом. Є сумісним з Mendeley, Zotero, RefWorks. Підтримує стилі APA, MLA, Harvard, Chicago, ASA, IEEE, AMA.

## 1 Citation

**Machine** www.citationmachine.net – простий онлайн сервіс за стилями *AMA*, *MLA*, *Chicago/Turabian*.

## VAK.in.ua

vak.in.ua – сайт для оформлення наукових праць у відповідності з вимогами Вищої атестаційної комісії (ВАК) України та перевірки курсових, дипломних, дисертацій тощо за стилем ДСТУ ГОСТ.



орепясіепсе.in.ua – онлайн сервіс, який дозволяє автоматично скорегувати цитування за стилями ДСТУ 8302:2015, ГОСТ 7.1:2003, *АРА*, *MLA* 7, ISO 690, *Chicago*. Сайт проводить транслітерацію з українського, російського та експорт у формат *Microsoft Word*.

## Контрольні запитання

- 1. Якими мають бути параметри абзацу\*?
- 2. Якими мають бути параметри сторінки\*?
- 3. Як встановлюються параметри абзацного відступу\*?
- 4. Як нумеруються сторінки\*?
- 5. Як позначаються рисунки\*?
- 6. Як позначаються таблиці\*?
- 7. Як позначаються формули\*?
- 8. Як типи посилань встановлює стандарт\*?
- 9. Які типи посилань встановлює ДСТУ 8302:2006?
- 10.Які вимоги до оформлення сторінок текстів висуває ДСТУ 2.105-95?

## 2.2. Текстовий процесор MS Word

### Інтерфейс програми

Інтерфейс *MS Word* є типовим для всіх застосунків *MS Office*.

Вікно програми містить органи керування режимами роботи, діями та робоче поле документа (рис. 2.3). Для керування редактором призначені *панель швидкого доступу* (Quick Access Toolbar), *стрічка* (Ribbon) та *рядок стану* (Status bar).

Для оперативної зміни вигляду органу керування слугують спливаючі меню. Такі меню називають контекстними меню (Shortcut menu), бо їхній склад змінюється в залежності від дій, які можна виконати з обраним елементом (рис. 2.4). Відкривається контекстне меню комбінацією клавіш «SHIFT+F10» або кліком правою кнопкою миші над обраним елементом.

<sup>\*</sup> Відповідно до вимог ДСТУ 3008:2015



## Рис. 2.3. Інтерфейс користувача MS Word



Рис. 2.4. Контекстні меню: а – колекції стилів; б – рядка стану

## Стрічка

Стрічка є головним засобом керування редактором. Замінити стрічку на органи керування панель чи меню, які застосовувалися в версіях програми до 2007 неможливо.

Елементи керування на *стрічці* згруповані по відповідних *вкладках* (Tabs). В кожній вкладці містяться елементи, які визначають дії однієї спрямованості. На вкладках елементи поєднані в групи за типами дій. Для полегшення орієнтації внизу кожної групи виведено її назву. Деякі групи вкладок стрічки мають стрілку в нижньому правому куті (Dialog box launcher) (рис. 2.5). Кліком мишею на позначці відкриває діалогове вікно налаштування відповідних режимів або параметрів.



Рис. 2.5. Dialog box launcher

При наведенні курсору на елемент керування на екрані спливає *інформаційне вікно* (ScreenTip), яке містить опис дії елементу.

Стрічка може відображатися з усіма елементуми, тільки вкладками або не відображатися зовсім. Режим відображення стрічки змінюється пунктами кнопки Параметри відображення стрічки (Ribbon view control) (рис. 2.6 a), контекстним меню стрічки (рис. 2.6 б) або комбінацією клавіш «CTRL+F1»,

Автоматично приховати стрічку Приховати стрічку. Торкніться у верхній частині застосунку, щоб відобразити її.	
Відобразити вкладки Відображати лише вкладки на стрічці. Торкніться вкладки, щоб відобразити команди. Відобразити вкладки й команди Відображати вкладки й команди на стрічці весь час.	Н <u>а</u> строїти панель швидкого доступу Відображати панель швидкого доступу під стрічкою <u>Н</u> астроїти стрічку Згор <u>н</u> ути стрічку
а	б

б



Кнопка **Файл** (File) стрічки не містить вкладки. Вона відкриває діалогове *вікно файлових дій* (Backstage view) (рис. 2.6). В лівій частині вікна знаходиться меню основних дій з файлом. Вибір пункту меню призводить до виведення в праву частину відповідної інформації.

		hormone trace (Leway elancaer)	eren.	
Відомості	Віломос	ті		
Створити	рідошос			
Відкрити	praktika2021			
Зберегти	G: » inf_t » pidr2021			
26		Режим сумісності	Властивості -	
зоерегти як	MET	Для уникнення проблем під час роботи з попередніми версіями системи	Розмір	9.34Мбайт
Друк	Перетворити	отпсе вимкнуто кльках нових функції. Гісля перетворення фаилу ці функції буде активовано, проте це може призвести до змін у макеті.	Сторінки	
Надати спільний			Слова	
доступ	0	Saviact JORVMOUTS	Загальний час редагування	2541 хв
Експорт	<b></b>	Серування типами змін, які можна вносити в цей документ.	Назва	Глава 1
	Захист		Теги	Додати тег
Закрити	документа *		Примітки	Додати примітки
Обліковий запис	Ā	Перевірка документа	Пов'язані дати	
Параметри	<u>e</u>	Перш ніж опублікувати цей файл, переконайтеся, що в ньому містяться такі	Востаннє змінено	23.06.2022 10:44
	Перевірити на наявність проблем -	елементи:	Створено	05.11.2021 13:47
		триває пошук помилок	Надруковано	01.05.2022 19:13
		Bepcii	Пов'язані користувачі	
	Керування версіями	Немає попередніх версій цього файлу.	Автор	dell
				Додати автора

Рис. 2.6. Backstage view

### Панель швидкого доступу

Панель швидкого доступу містить кнопки команд, які виконуються найчастіше. За замовчанням на панелі розташовані кнопки збереження документу, відміни останніх дій, повторення останньої дії. Склад панелі визначається в діалоговому вікні налаштувань. Вікно налаштувань активується пунктом **Hacтpoїти панель швидкого доступу** (Customize Quick Access Toolbar) контекстного меню стрічки (рис. 2.5 б) або пунктом Файл-Параметри-Панель швидкого доступу (File-Options-Quick Access Toolbar) стрічки.

#### Рядок стану

В рядку стану виводиться основна інформація про документ та містяться деякі елементи керування. Склад елементів, які виводяться в рядок стану, можна змінювати в контекстному меню рядка (рис. 2.4 б). За замовчанням в рядку виводяться кількість сторінок та номер поточної сторінки *Сторінка* (Page) та слів *Кількість слів* (Words) документа, позначка помилок правопису, кнопка визначення мови документу, кнопки зображення документу (View controls) кнопки керування масштабом зображення документу (Zoom controls).

#### Робоче поле

На *робочому полі* редактора зображується сторінка документу. Документ може бути зображений в **режимі читання (Read More)**, **Вебдокумент (Web Layout)**, **розмітки сторінки (Print Layout)**, **режимі структури** складеного документу (**Outline**), **чернетки Draft**).

Встановлення поточного режиму відображення документа проводиться кнопками панелі стану (рис. 2.3) або кнопками Вигляд-Подання (View-Views) стрічки.

Для переміщення по великих документах застосовується *панель навігації* (Navigation pane) (рис. 2.3). Виводиться на екран *панель навігації* пунктом **Вигляд-Відображення** (View-Show) стрічки. Панель відображає список розділів документа або сторінки документа. Відображення відповідного фрагменту документа на екрані проводиться кліком по обраному розділу або сторінці на панелі.

Найбільш інформативним при створенні та редагуванні документа є режим *розмітки сторінки*. В цьому режимі аркуш документа відображається найбільш реалістично, а для спрощення компонування тексту можуть застосовуватися такі допоміжні елементи, як *розрив аркушів* (white space), мірильні *лінійки* (Rulers), *сітка* (Gridlines), *відображення недрукованих символів* (Nonprinting characters).

Відокремлення сторінок між собою в режимі *розмітки сторінки* за замовчанням проводиться смужкою фону сірого кольору. Наведення курсору

92

на смужку призводить до появи маркеру у вигляді двох стрілок (up-down arrow) (рис. 2.7 а). Оперативно прибрати/показати смужку можна подвійним кліком на маркері смужки. Глобально для всіх документів видимість смужки визначається пунктом Файл-Параметри-Відображення-Параметри відображення сторінок-Відображати пробіли між... (File-Options-Display-Page display options-Show white space...) стрічки (рис. 2.7 б).

Double-click to hide white space	Відображення Правопис Збереження Мова	Параметри відображення сторінки Гараметри відображення сторінки Відображати пробіли між сторінками в поданні розмітки сторінки ••••••••••••••••••••••••••••••••••••
а	б	

Рис. 2.7. White space: а – зображення; б – керування

### Мірильні лінійки

Мірильні лінійки (рис. 2.8 а) відображаються в робочому вікні в *режимі* розмітки сторінки та веб-документа.

Горизонтальна мірильна лінійка розташована нагорі робочого вікна. На ній показуються позначки абзацу, лівого та правого берегів, табуляції. Вертикальна лінійка розташована на лівому краю вікна та показує верхній та нижній береги. Відображенням обох лінійок керує пункт Вигляд-Відображення-Лінійки (View-Show) стрічки (рис. 2.8 б). Видимість вертикальної лінійки Відображати вертикальну лінійку... Show verticsl ruler...) та одинці виміру на лінійках Відображати вимірювання в одиницях (Show measurements in units of) можна в діалоговому вікні з пункту Файл-Параметри-Додатково-Відображення (File-Options-Advanced-Display) стрічки (рис. 2.8 в).



Рис. 2.8. Лінійки та сітка: а – вигляд; б – відображення; в – налаштування

## Сітка

Сітка використовується для точного позиціонування тексту та елементів оформлення на сторінці документа.

Параметри *сітки* визначаються у вікні Сітка і напрямні (Gridlines Navigation Pane) (рис. 2.9 б). У вікні можна задати кроки Крок сітки (Gridlines step) сітки в поточних одиницях вимірювань та кроки відображення сітки на екрані в цілих величинах періодів кроку сітки Показати сітку (Show gridlines). Викликається вікно пунктом Розмітка сторінки-Вирівняти-Параметри сітки (Layout-Align-Gridlines options) стрічки.

## Параметри текстів

Текст документа складається з набору символів. Символи поєднуються в шрифти за ознакою подібності контуру символів. Виділять шрифти *пропорційного* та *моно* типів. В *пропорційних* шрифтах символи мають змінну ширину, в *моно* шрифтах ширина всіх символів є сталою. В *пропорційних* шрифтах розрізняють родину *Serif*, в яких на кінцях ліній символів є декоративні засічки, та родину *San Serif*, контури символів яких не мають декоративних засічок.

Конкретний набір символів з відповідним способом малювання контуру називають *гарнітурою* або просто шрифтом відповідного імені. Наприклад, пропорційним шрифтом родини *Serif*, тобто з декорацією контуру, є шрифт гарнітури *Times New Roman* або просто *Times New Roman*. Прикладом пропорційного шрифту родини *San Serif* є *Arial*, моно шрифту – *Courier*.



Рис. 2.9. Сітка: а – виклик вікна налаштувань; б – вікно налаштувань

Для шрифтів в документі визначається розмір або кегль, колір, стиль написання: нормальний (*Normal*), напівжирний (*Bold*), з нахилом (*Italic*) тощо.

*Кегль* – це одиниця виміру розміру шрифтів, який визначається в «точках» (*point size*). Одна точка, кегль 1 має висоту в одну сімдесят другу дюйма (1 pt=1/72 дюйма). Розмір символу 14 кегля відповідає висоті 5 мм.

Основні параметри поточного шрифту визначаються полями Основне-Шрифт (Main-Fonts) стрічки. Для детального визначення параметрів шрифту слугує вікно властивостей шрифтів. Вікно відкривається комбінацією клавіш «*Ctrl+D*», пунктом Щрифт (Font) контекстного меню виділеного тексту або Dialog box launcher групи Основне-Шрифт (Main-Fonts)стрічки.

Хоча ДСТУ 2.105-95 не вимагає для документів, які виготовляються версткою на комп'ютері, обов'язково використовувати креслярський шрифт за ГОСТ 2.304-81 (ДСТУ ISO 3098:2006), для повного виконання вимог ДСТУ можна використати *TTF* шрифт ОС Windows, наприклад *isocpuer.ttf*.

Крім символів тексту в документі можуть відображатися *недруковані* символи.

*Недруковані символи* – це елементи сторінки, які зазвичай не відображається на екрані, як-то абзацний відступ, табуляція, пробіл, позначки структури документа, позначка кінця рядка тощо.

*Недруковані символи* відображаються як «·» для пробілу, «→» для табуляції «¶» для кінця параграфу, натискання клавіші «*Enter*».

Вмикається показ недрукованих символів кнопкою <sup>¶</sup> закладки Основне-Абзац (Main-Paragraph) стрічки.

Відображення символів відбувається відповідно до його коду в застосованій кодовій таблиці. Найбільш розповсюдженими є коди UTF та ANSI. Коди абеток та найбільш вживаних символів призначені клавішами клавіатури та вводяться автоматично однократним натискання відповідних клавіш. Для введення спеціальних символів, як то математичних, грецьких призначено пункт стрічки. В меню пункту зберігаються останні використані в документах спеціальні символи. Додати нові символи можна у вікні вставляння символів пунктом Вставлення-Символ-Інші символи... (Insert-Symbol-More Symbols...) стрічки. Корисні для науково-

95

технічних документів символи знаходяться в наборах *грецькі та коптські* символи, знаки пунктуації, стрілки, математичні оператори, технічні знаки.

Символ в коді UTF вставляється за його шістнадцатеричним кодом комбінацією клавіш «чотиризначний шістнадцятирічний код+ALT+X». Наприклад, символ і може бути вставлений комбінацією «20AA+ ALT+X».

Символ в коді ANSI вставляється за його десятковим кодом комбінацією клавіш на цифровій клавіатурі «*ALT+тризначний десятковий код*». Наприклад, символ <sup>Ц</sup>, може бути вставлений комбінацією «*ALT+188(цифровий)*».

Український правопис [23] передбачає застосування дефісу в складених словах та тире – в інших випадках.

Редактор дозволяє введення в текст дефісу «-», тире «--» та довгого тире «---».

Символ тире вводиться комбінацією клавіш «*Ctrl+-»*. Символ «-» вводиться на цифровій клавіатурі.

Символ довгого тире вводиться комбінацією клавіш «*Ctrl+Alt+-»*. Символ «-» вводиться на цифровій клавіатурі.

За умови активації *режиму автозаміни* символ дефісу автоматично перетворюється на символ тире після введення символу пробілу після слова за дефісом, якщо перед дефісом теж був поставлений символ пробілу.

Режим автозаміни активується пунктом Файл-Параметри-Правопис-Параметри автозаміни-Автоформат під час введення тексту-дефіси та тире (File-Options-Proofing-AutoCorrect Options-AutoFormat as you type-Hyphens with Dash) стрічки (рис. 2.10).



Рис. 2. 10. Режим автозаміни

У випадках, коли необхідно забезпечити нерозривність послідовних слів або складеного слова застосовують *нерозривний пробіл* та *нерозривний дефіс*. Наприклад, *нерозривний пробіл* між прізвищем та ініціалами – «Куций О. О.», *нерозривний дефіс* в назві «ПО-21».

*Нерозривний пробіл* (Nonbreaking space) вставляється комбінацією клавіш «*Ctrl+Shift+пробіл*».

*Нерозривний* дефіс (Nonbreaking hyphen) вставляється комбінацією клавіш «*Ctrl+Shift+дефіс*».

Позначка трьох крапок вставляється комбінацією клавіш «*Ctrl+Alt+крапка»*.

Переведення рядка без розривання абзацу проводиться комбінацією клавіш «Shift+Enter».

Цитати, власні назви, терміни тощо в документах забираються в лапки.

В різних країнах, в різних документах застосовують різні типи лапок. Ресурс Вікіпедія (https://uk.wikipedia.org/wiki/Лапки) наводить велику кількість різновидів лапок. Серед них:

- ялинки, кутові, шеврони, французькі, guillemets «Текст»;
- прямі, машинописні, програмувальні, double prime "Текст";
- німецькі "Текст";
- особливі, польські, угорські "Текст";
- подвійні, англійські, quotation marks "Текст";
- непарні, скандинавські "Текст";
- марровскі 'Текст'.

Редактор дозволяє введення кількох різновидів лапок:

- лапки-ялинки «» виводяться комбінацією клавіш «Shift+2» за розкладки клавіатури «рус/укр»;
- марровські лапки '' виводяться клавішею «`» за розкладки клавіатури «eng»;
- подвійні верхні лапки "" виводяться комбінацією, клавіш «*Shift*+'» за розкладки клавіатури «eng».

Натискання комбінації клавіш «*Ctrl+Z*» безпосередньо після друку лапок переводить лапки в прямі.

За українським правописом в документах слід «використовувати лапкиялинки. У випадку цитати всередині іншої цитати слід вживати лапки-ялинки як зовнішні лапки, а "лапки-ла́пки" (англійські) – як внутрішні» [23].

## Абзаци

Шрифт символів тексту, порядок вирівнювання слів в рядку (Alignment), режим проставляння переносів в словах (Hyphenation), горизонтальне положення абзацу на сторінці (Indentation), міжрядковий інтервал (Line spacing), інтервали між абзацами (Spacing), контроль першого та останнього рядків (Widow, orphan) тощо створюють стиль абзаців (Paragraph styles).

Редактор дозволяє встановлювати вирівнювання слів в рядку по лівому краю, по правому краю, по центру, по ширині рядка. Стиль вирівнювання задається кнопками Основне-Абзац (Home-Paragraph) стрічки або полем Загальні-Вирівнювання (General-Alignment) вікна налаштувань абзацу Абзац (Paragraph) (рис. 2.11 а). Вікно активується пунктом контекстного меню абзацу або *Dialog Box launcher* стрічки Основне-Абзац (General- Paragraph).

Рекомендованим є вирівнювання по ширині рядка По ширині (Justify).

Проставляння знаку перенесення в словах між рядками може бути встановлене як Ручне (Manual), Автоматичне (Automatic), Немає (None). Автоматичне розташування слів в рядку абзацу без переносів називається Word Wrap. Спосіб проставляння переносів задається полями Розмітка сторінки-Розставляння переносів (Layout-Hyphenation) стрічки.

Абзац			?	×
Відступ <u>и</u> та інт	тервали <u>Р</u> озташуванн	я на сторінці		
Загальні				
Durpinungenu	Demonstra	~		
одривнюван	ня. по центру			
Рівень конту	уру: Основний текст	<ul> <li>Згорнуто за промов</li> </ul>	чанням	
Відступ				
Діворуч:	0 см	Спеціал <sub>ін</sub> ний:	Ha:	
Праворуч:	0 см 🌩	Перший рядок	√ 1.25 см	-
Інтервал Перед:	0 nr 🔶	Міжрядковий інтервал:	<u>H</u> a:	•
Після:	6 nr 🖵	Одинарний	~	-
Не дода	вати інтервал між абзаца	ми одного стилю		
Попередній пе	ерегляд			
1	Зоперадній аблац Поперадній аби	n Hompanik afan Hompanik afan	Попередній абнац	
Попярядз	ния абнац Поперадній абнац Поперадо Рис. 2.11. Напакиту вання абнацу: а -	ни комп, Шопередник кбим, Попередній кбик вікно налаштувань абзаду; б — керування п	ц Поперадний абнац впоженных рядків	
E afron Ha	Haerymmit aðsan Haerymmit aðsan H servmmit aðsan Haerymmit aðsan Ha	laerymmil aðsan Haerymmil aðsan Haerymm erymmil aðsan Haerymmil aðsan Haerymm	nli aðsan Haerymmik sk aðsan Haerymmik	
a6mm Ha a6mm Ha	ступний абын Наступний абын На ступний абын Наступний абын На	erymuli aðsan Haerymuli aðsan Haerymu erymuli aðsan Haerymuli aðsan Haerymu	ik aðsan Hacrymusk ik aðsan Hacrymusk	
				_
Табилаціа	23 0004000	OK	Craom	
Toolynuffur	24 UKOWOB4	OK	CKaCya	

Рис. 2.11. Налаштування абзацу: а – вікно налаштувань абзацу; б – керування положенням рядків

Рекомендованим режимом  $\varepsilon$  спосіб без перенесення **Немає** (None).

Параметри горизонтальних відступів рядків абзаців помічаються на горизонтальній лінійці (рис. 2.8 а). Відступ першого рядка (first-line indent) від лівого краю абзацу помічається маркером . Відступ краю абзацу від лівого берега (left indent) позначається маркером . Відступ краю абзацу від правого берега (right indent) позначається маркером .

Змінити значення горизонтальних відступів можна безпосереднім зсувом маркерів на лінійці. Зсув маркера **Відступ зліва (Left** intend) за нижній край переміщує лівий край всього абзацу. Зсув маркера **Нависаючий відступ (Hanging indent)** за верхній край переміщує абзац без зміни положення маркера відступу першого рядка. Положення першого рядка абзацу відносно краю аркуша при цьому не змінюється.

Кількісно визначити відступи абзацу можна у вікні налаштувань абзацу (рис. 2.11 а) в полях Відступ-Ліворуч/Праворуч/Перший рядок (Intendation-Left/Right/Special).

Редактор забезпечує автоматичне уникнення такого розташування абзацу на сторінці, коли на ній знаходиться тільки один рядок абзацу. Такий режим називається контролем першого та останнього рядків на сторінці. Режим вмикається на сторінці Розташування на сторінці-Заборона нависаючих рядків (Line and Page Breaks-Widow/Orphan control) вікна налаштування абзаців (рис. 2.11 б).

Оформлення абзацу з виділенням першої літери можна провести пунктом Вставка-Текст-Буквиця (Insert-Text-Drop cap) стрічки.

Для визначення міжрядкового інтервалу на вкладинці Конструктор-Інтервал між абзацами (Design- Paragraph Spacing) стрічки містяться кілька вбудованих форматів. Серед них практичними з урахуванням вимог ДСТУ 3009:2015 можна назвати Щільно (Tight, перед 0pt, після брt, міжрядкова відстань 1.15) та Розріджено (Relaxed, веред 0pt, після брt, міжрядкова відстань 1.5).

#### Стилі

Визначати параметри тексту можна безпосередньо під час введення або використанням визначених *стилів абзаців*.

При безпосередньому форматуванні кожна зміна формату автоматично додається в список стилів документа та збільшує його. Такий спосіб введення

текстів суттєво збільшує витрати на редагування документа, може призвести до проблем відображення при перенесенні інформації між документами, тому його не можна вважати ефективним.

Застосування стилів передбачає попереднього визначення типових *стилів абзаців*, які передбачається використовувати в документі. Зазвичай таких стилів небагато. Визначені стилі розміщуються в *експрес-колекції стилів* (Styles gallery) **Основне-Стилі** (Main-Styles) на головній вкладці стрічки (рис. 2.12 а). В такому випадку для визначення формату всього абзацу слід встановити курсор в обране місце, обрати стиль абзацу кліком на відповідному стилі з колекції та ввести текст. Змінити стиль вже введеного тексту можна встановив курсор на вже введений текст та клікнути на відповідному стилі з колекції.

Визначити стиль можна і в повному списку стилів. Повний список стилів виводиться на екран в плаваючому вікні позначкою *Dropbox* колекції стилів (рис. 2.12 а). При безпосередньому форматуванні документа список містить до сотні рядків. Працювати з таким списком досить незручно. Дещо покращити ситуацію можна змінивши стиль виведення інформації у вікно стилів. Для налаштування списку виведених стилів використовується діалогове вікно налаштувань (рис. 2.12 б). Вікно активується кнопкою **Параметри** внизу списку. Для видалення зі списку стилів, які не використовуються в документі слід в полі **Вибір відображуваних стилів** вікна налаштувань обрати значення **Використовуване**. Для того, щоб залишити в списку тільки базові стилі для всіх елементів тексту та прибрати окремі рядки для стилів тексту, списків, абзаців слід прибрати відповідні позначки в полі **Відображати вказані типи форматів як стилі вікна**.

Створення нового стилю передбачає визначення шрифту символів тексту, порядку вирівнювання слів в рядку, режиму проставляння переносів в словах, горизонтального положення абзацу на сторінці, міжрядкового інтервалу, інтервалів між абзацами тощо. Параметри стилю задаються та редагуються в діалоговому вікні Створення стилю за допомогою форматування (рис. 2.12 в). Вікно активується кнопкою створення стилю списку стилів (рис. 2.12 а) або пунктом контекстного меню **Створити стиль** галереї стилів. Меню відкривається позначкою *з*, яка знаходиться в правому нижньому куті галереї.

I Oberwaii Voor vir toom v	АаБбВв АаБбВв АаБбВі	АаБбВв Стилі	🕶 : Параметри області стилів		?	×
Image: Construction of Constru	1 Звичайн 1 Обычн Строгий	1 Абзац с	<u>в</u> ибір відображуваних стилів:			
Bo to	Стилі	я Земчайний	Використовуване			$\sim$
A second dealer of the second dealer dea	10 · · · 11 · · · 12 · · · 13 · · ·	14 - 1 15 - Маркований списо	вибір способу сортування спи	ис <u>к</u> у:		
A concernent doorwary name.          A concernent doorwary name.       A processand drucos + anape.         A georeanded drucos + anape.       A processand drucos + anape.         A georeanded drucos + anape.       A processand drucos + anape.         A georeanded drucos + anape.       A processand drucos + anape.         A georeanded drucos + anape.       A processand drucos + anape.         A georeanded drucos + anape.       A processand drucos + anape.         A georeanded drucos + anape.       A processand drucos + anape.         A georeanded drucos + anape.       A proposand seguration and seguration and seguration and seguration.         A georeanded drucos + anape.       A proposand seguration and seguration and seguration.         A georeanded drucos + anape.       A proposand seguration and seguration.         A georeanded drucos + anape.       A proposand seguration and seguration.         A georeanded drucos + anape.       A proposand seguration.         A ge		Маркований списо	ок + Camb Як рекомендовано			$\sim$
a <sup>1</sup>	Эт Створити стила	Маркований списо	ок – Camb Відображати вказані типи фо	рмату як стилі:		
Action       Imaginación (maginación	🏘 Очистити форм	латування Маркований списо	ок + налов 🗸 абзаци			
a	🏘 Застосувати ст	илі надрядкові	✓ шр <u>и</u> фти			
a Buildo caccodo sido do paxenta incenta sido consola intervenzio en la consola dan traccada and traccada a		Обычный 0 без 6	🔹 🗹 форматування у вигля	ді маркірованого та нумерованого спис	ків	
a a a b b b b c c c c c c c c c c c c c		Попередній пере	вибір способу відображення	імен вбудованих стилів		
a		Вимкнути зв'язан	ні стилі 📃 Відображати наступни	і <u>й</u> заголовок під час використання попер	реднього	о рівня
a b b b b b b b b b b b b b b b b b b b		2 3 3	Параметри	імена, як <u>щ</u> о існують альтернативні		
Compose crows at generation of the physical particular distance       1       X         Decrementing       Image:	a	Створення стил	КО О Лише в цьому документі	О Нові документи на основі цього ша	аблону	
Bacremoci         With       We added         With       We added         With Control       We         Control       We added         Control       Decontrol         Contro       Decontrol		Створення стилю за допомогою форм	матування	? ×		
Image:		Властивості				
té coogé (ranse Criss agrymours afairs Criss agrymours afairs		тідп стилно: Зв'язанняй (аб	бзац і символ)	~		
Concerner         Programmi divent         Threes Hores Rooma         I III IIIIIIIIIIIIIIIIIIIIIIIIIIIIIII		На основі стилю: Т Звичайний	й			
Тетек Влема 1 14 2 X K 0 2		Стиль наступного аbsauj: Стильс		×		
		Times New Roman 💙 14 💟 🕅	К К П			
Паворский абон Поорский обла Паворский обла Поорский Обла Поорскии Обла Поорскии Обла Поорскии Обла Поорскии Обла Поорскии Обл		= = = <b>=</b> = =	=   # #   I I			
Додата до доленції силія   Денальовати автоматично     Дина в цали долушино Повід допуметаті на основі цало цалбону      Фонта на основі доля цало цалбону     Ок      Селонати		Поперений абили Поперений Поперений абили Поперений Зразок тексту Зразок тексту зото то то то то то	ай абаап Полерский абаап. Полерский абаап. Полерский абаап. Зразок тексту Зразок тексту Зразок тексту Зразок тек к тексту Зразок тексту тексту зразок тексту зразок тексту зразок тексту зразок тексту зразок тексту тексту зразок тексту зразо	жсту разок кксту		
Compare to Look any system or the construction of Lo		🗹 Додати до колекції стилів 🔽 Qновлюг	OHI-PITEMOTEL WIG			
		<ul> <li>Дише в цьому документі Нові доку</li> <li>Формат •</li> </ul>	ОК	Скасувати		

Рис. 2.12. Стилі: а – колекція; б – налаштування списку; в – введення нового стилю

## Табуляція

Табуляція застосовується для того, щоб отримати регулярне розташування інформації на зразок таблиці, але без створення самої таблиці. Якщо при звичайному введенні відстань між словами задається символами пробілу, то в режимі табуляції слова відстоять одне від одного на крок табуляції. Відокремлення слів проводиться клавішею табуляції. За замовчання крок табуляції – 1.27 см. Позиції табуляції позначаються умовними позначками на горизонтальній мірильній лінійці.

Слова можуть бути вирівняні відносно позначки табуляції з лівого боку (Left ), з правого боку (Right ), по центру (Center ) тощо.

*Тип вирівнювання* табуляції оперативно задається кнопкою табуляції у верхньому лівому куті сторінки над вертикальною лінійкою (рис. 2.3). Перемикання між типами вирівнювання табуляції проводиться послідовними кліками на кнопці. Розмітка табульованого рядка може проводитися парами дій «вибір типу вирівнювання — клік на обраному місці горизонтальної лінійки» або створенням списку табуляції в діалоговому вікні **Табуляція (Tabs)** (рис. 2.13). Вікно активується кнопкою **Табуляція...(Tabs...)** у вікні налаштування абзаців (рис. 2.11 а) або подвійним кліком на горизонтальній лінійці для редагування попередньо визначеної розмітки.

Табуляція		? ×
Позиції та <u>б</u> уляції:		За промов <u>ч</u> анням:
		1.27 см 💂
	~	Буде видалено:
	~	·
вирівнювання		
) з <u>л</u> іва	○ по цен <u>т</u> ру	🔵 с <u>п</u> рава
<u>з</u> а розділювачем	) з риско <u>ю</u>	
Заповнювач		
<u>1</u> (немає)	<u>2</u>	<u>3</u>
<u></u>		
Установ	ити Видалити	Видалити вс <u>е</u>

Рис. 2.13. Вікно табуляції

При розмітці табульованого рядка на лінійці відображаються позначки обраного типу у відповідних місцях.

Створення табульованого рядка може проводитися кількома способами:

- спочатку створити шаблон табульованого рядка, а потім внести в нього слова. Введення слів за таким способом починається з зсуву курсору в потрібне місце натисканням клавіші табуляції.
- спочатку набрати слова, потім провести розмітку рядка, а потім зліва направо ставити курсор на початок слова та натискати клавішу табуляції.

За будь-якого способу заново створювати шаблон для кожного нового рядка не потрібно. Переведення курсору на наступний рядок зберігає розмітку табуляції.

### Редагування

Типовими діями для виділення інформації для редагування є:

- подвійний клік на слові для виділення слова;
- клік на лівому березі для виділення рядка;
- потрійний клік на слові або подвійний клік на лівому березі для виділення абзацу;

- встановлення курсору на початок фрагменту та натискання клавіш «*Shift+клік»* в кінці фрагменту – для виділення довільного за обсягом фрагменту тексту;
- виділення мишею з натиснутою клавішею «*Ctrl*» (режим multiselection) — для виділення кількох довільних несуміжних областей.

Вставляння інформації (Paste) з буфера обміну (Clipboard) може проводитися в режимах Зберегти вихідне форматування (Кеер source formatting), Об'єднати форматування (Merge formatting), Зберегти лише текст (Keep text only).

В режимі Зберегти вихідне форматування стилі з буферу додаються до існуючих стилів. В режимі Об'єднати форматування для нового документа інформації з буферу призначається стиль Звичайний, в існуючий документ стилі з буферу додаються до існуючих стилів. В режимі Зберегти лише текст інформації з буферу призначається стиль Звичайний.

Тип вставляння з буферу клавішами «*Ctrl+V»* за замовчанням визначається установками групи стрічки **Файл-Параметри-Додатково-Вирізання, копіювання та вставляння (File-Options-Advanced-Cut, copy and paste**) (рис. 2.14 а). За замовчанням проводиться вставляння **Зберегти вихідне форматування**.





Провести вставляння можна спочатку обравши спосіб вставляння в одному з пунктів Основне-Вставити (Home-Paste) або можна спочатку вставити фрагмент з буферу клавішами «*Ctrl+V*». Потім натиснути клавішу «*Ctrl*». На місці вставленого фрагменту з'явиться меню (рис. 2.14 б). Обрати спосіб вставляння можна кнопками меню або клавішами: «В» – «зберегти вихідне форматування», «J» – «об'єднати форматування», «С» – «зберегти лише текст».

При виділенні фрагменту тексту над ним за замовчанням автоматично виводиться невеличке вікно (Mini toolbar), яке містить основні дій форматування (рис. 2.15 а). Автоматичним виведенням цього вікна керує пункт налаштувань файл-Параметри-Загальне-Параметри інтерфейсу користувача-Відображати міні-панель... (File-Options-General-Show mini toolbar on selection) (рис. 2.15 б) стрічки.



Рис. 2.15. Mini toolbar: а – вигляд; б – налаштування

### Шаблони

Для того, щоб не повторювати визначення стилів в кожному новому документі застосовуються *шаблони* документа (Template). В шаблоні зберігаються формати шрифтів графіки, параметри аркуша, стилі абзаців, макрокоманди, елементи автотексту, налаштування стрічки тощо. Файли шаблонів мають розширення .*dot*, .*dotx*, .*dotm*.

При створенні нового документа *Word* за замовчанням використовує шаблон *Normal.dotm*, який знаходиться за адресою *«c: \ Users\XXX\AppData\Roaming\Microsoft\Templates\»*.

Найзручнішим можна вважати спосіб створення документа із застосуванням вже підготовленого шаблону. Слушним можна вважати створення шаблонів користувача для типових документів. Наприклад, шаблон для подання статті в журнал «Наукові вісті НТУУ КПІ» Vesti-KPI.dotm, шаблон для технічних документів СКД ESKD.dotm, шаблон документів за ДСТУ 3008:2015 DSTU3008.dotm тощо.

В документації виробника вказано, що для застосування шаблону користувача потрібно перейти на вкладинку Особисті/Персональні (Personal) стартового екрана (рис. 2.16).

В версіях редактора, починаючи з *Word 2013*, розробник ускладнив роботу з шаблонами користувача. В режимі створення файлів виводиться екран з вбудованими шаблонами *Microsoft*. За умови підключення Інтернету виводяться шаблоні з сайту *Office*, в автономному режимі виводяться шаблони з каталогу «*c:\Program Files\Microsoft Office XX\root\Templates\XXX\»*, де

XXXX – номер мовної сторінки редактора (1033 – англійська, 1049 – російська тощо). Змінювати списки вбудованих шаблонів користувачеві неможливо.

Створити		
Пошук онлайнових шаблонів		
Рекомендовані пошукові запити:	Бизнес	Карточк
СПЕЦІАЛЬНІ ОСОБИСТІ		

Рис. 2.16. Вкладинка Особисті стартового екрана

Перехід на сторінку шаблонів користувача активується тільки після того, як в налаштуваннях редактора в пункті **Файл-Параметри-**Збереження-Збереження документів- Розташування за замовчанням особисті шаблонів (File-Options-Save-Default personal templates location) стрічки буде вказано шлях до папки з шаблонами (рис. 2.17).

За замовчанням шаблони зберігаються за адресою

«c:\Users\XXX\Documents\User templates Office».

Розташування за промовчанням для локального файлу: С:\U: Розташування за промовчанням для особистих шаблонів: О\U: Параметри редагування файлів в автономному режимі на се

Рис. 2.17. Шлях до шаблонів

Шаблони, що автоматично завантажуються, знаходяться за адресою

«c: \ Users\XXX\AppData\Roaming\Microsoft\Word\Startup\».

Іншим способом використання шаблонів є створення нового стандартного документа та під'єднання до нього потрібного готового шаблону. Це можна зробити в діалоговому вікні налаштування шаблонів (рис. 2.18). Вікно викликається пунктом Файл-Параметри-Надбудови-Керування-Шаблони-Перейти (File-Options-Add ins-Manage-Templates-Go) або полем вкладки Розробник-Шаблон документа (Developer-Templates) стрічки.

Шаблони	XML-схема	Пакети розширення XML	Зв'язані CSS			
<u>Ш</u> аблон до	кумента					
Normal					При <u>є</u> днати…	
	матично оновли	овати стилі				
Приє	днувати до нові	их повідомлень				
2010/10/10	аблони та налби	UTORIA				
Поририон		2202020202000				
Word	Cmds dot	sabania, kenu.			0	
	emasiaot			$\sim$	додати <u>.</u>	
					Видалити	
					-	
				~		
	) Dreener Files	(u06)) Mathatiana) Office Current	a) 20\\WardCarda dat	$\vee$		

Рис. 2.18. Вікно шаблонів

## Оформлення аркушів

При оформленні документів застосовуються *горизонтальні лінії*, які відокремлюють частини документу (рис. 2.19).



Рис. 2.19. Оформлення документу горизонтальною лінією

Для креслення горизонтальних ліній в редакторі передбачені вбудовані засоби автоформатування символів, властивостей абзацу або таблиці, створення лінії як графічного елементу.

Рекомендованим виробником є спосіб малювання горизонтальної лінії шляхом автоформатування символів. В режимі автоформатування лінія зображується на всю ширину рядка після трикратного введення символу та натискання клавіші «*Enter*». Символи «\*» генерують пунктирну лінію, символи «=» – подвійну тонку лінію, символи «~» – хвилясту лінію, символи «#» – подвійну товсту лінію, символи «-» – товсту лінію, символи «\_» – тонку лінію.

Особливістю режиму автоформатування є те, що зміна розмірів такої лінії є неможливою.

Прибрати лінію автоформатування. відразу після її введення можна клавішами відміни дії «*Ctrl+Z*».

Прибрати автоформатовану лінію в будь-який час можна встановивши курсор на рядок над лінією та обравши пункт Основне-Межі —-Без меж (Main-Paragraph group menu Borders- No border) стрічки (рис. 2.20 а).

Керування активацією режиму автоформатування символів проводиться пунктом Файл-Параметри-Правопис-Параметри автовиправлення-Автоформат під час введення тексту-Застосовувати під час введення тексту-лінії меж (Options-Proofing-Autocorrect options-Autoformat-Apply as you type-Boder lines) стрічки (рис. 2.20 б).



Рис. 2.20. Керування лініями автоформатування: а – прибирання; б – режим автозаміни

Горизонтальна лінія може бути введена як частина рамки абзацу тексту, таблиці.

Рамка навколо абзацу або виділеного фрагменту тексту малюється пунктом OchoBhe-Mexi — тип рамки (Main-Paragraph group menu-Borders). Лінії рамки навколо абзацу можуть бути намальовані з будь-якого боку. Навколо фрагменту тексту рамка може проводитися тільки з усіх боків. Рамка малюється поточною шириною, кольором та візерунком. Поточні параметри визначаються в діалоговому вікні рамок Mexi (Borders) (рис. 2.21 а). Вікно активується пунктом стрічки OchoBhe-Mexi — Mexi та тіні...(Main-Paragraph group menu-Borders-Borders and Shading).

Горизонтальна лінія може вводитися як підкреслення (**Underline**) фрагменту тексту. Візерунок та колір лінії підкреслення тексту обираються з меню кнопки підкреслення (рис. 2.21 б).

Горизонтальна лінія може бути намальована пунктом Основне-Межі — Горизонтальна лінія (Main-Paragraph group menu Borders-Horizontal line) стрічки. Така лінія за замовчанням

107

малюється як тонка. Вона є графічним примітивом, який можна пересунути в інше місце, змінити його розміри, видалити в будь-який час. Довжина, ширина, колір, стиль позиціонування горизонтальної лінії можуть бути налаштовані пунктом **Параметри горизонтальної лінії** контекстного меню лінії (рис. 2.21 в).

Межі й зал	ливка		? X			
Межі	<u>С</u> торінка	а <u>З</u> аливка		$\blacksquare \bullet abe X_2 X^2   \land \bullet ab  \bullet X_2 X^2   \land \bullet ab  \bullet A$	Формат горизонтальної лінії	? X
Тип:		Ст <u>и</u> ль:	Попередній перегляд		Горизонтальна лінія	
	нема <u>с</u>	^ ^	Щоб додати межі, клацніть зразок або скористайтеся кнопками		Ширина 100% 🔶 одиниці: Відсотки	~
	рам <u>к</u> а				Висота	
	тјнь		АаБбВвЮю		Колір	
	об'ємна	Кодір:			використовувати чистий колір	~
		Автоматично 🗸 Ширина:	88 83	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	Вирівнювання	
	інша	1,5 nr — ~	Застосувати до:	немає		_
			тексту 🗸	<u>Інші підкреслення</u>	здіва по центру	справа
			э	б	D	

Рис. 2.21. Меню: а – рамок; б – підкреслення тексту; в – горизонтальної лінії

Горизонтальна лінія є одним з вбудованих графічних примітивів Фігури (Shape). Вставлення фігури проводиться пунктом Вставлення-Фігури-Лінії (Insert-Shapes-Lines) стрічки (рис. 2.22 а). Визначення розмірів, положення на сторінці, кольору, візерунку фігури тощо проводиться полями контекстної сторінки формат (Format) стрічки (рис. 2.22 б), яка автоматично активується при виділенні фігури.



Рис. 2.22. Фігури: а – малювання: б – редагування

## Колонтитули

Для автоматизації введення незмінної інформація, яка повторюється на всіх сторінках розділу документа або всього документа, як номери сторінок, назви розділів тощо, застосовують *колонтитули* (Header and Footer).

Колонтитули – це горизонтальні смуги згори та знизу сторінки.
Активуються колонтитули для редагування пунктами групи Вставлення-Колонтитули (Insert-Header&Footer) стрічки або подвійним кліком на верхньому/нижньому березі аркуша.

Ввести номери сторінок з позиціонуванням в рядку найпростіше пунктом номеру сторінки (рис. 2.23 а) Вставлення-Колонтитули-Номер сторінки (Insert-Header&Footer-Page number) стрічки або контекстної вкладки колонтитулів Конструктор-Колонтитули-Номер сторінки (Design-Header&Footer-Page number) стрічки в разі активації колонтитулів.

В такий спосіб номер проставляється Згори сторінки (**Top of Page**), Внизу сторінки (**Bottom of Page**), На полях сторінки (**Page Margins**) або в рядку поточного положення курсору Поточне положення (**Current Position**).

В рядку поточного положення курсору номер сторінки може бути вставлений як *поле* (field) з назвою *PAGE*. Для цього слід розмістити курсор в потрібне місце, відкрити діалогове вікно полів (рис. 2.23 б) пунктом Вставлення-Текст-Огляд експрес-блоків-Поле... (Insert-Text-Express block-Field...) та зі списку полів обрати поле *PAGE*.



Рис. 2.23. Номери сторінки: а – стрічка; б – поле; в - фігура

Вставити номер сторінки в довільне місце аркуша без прив'язки до рядків можна використанням такого різновиду фігури, як текстове поле. Вставлення номеру проводиться за наступною схемою: Вставлення-Фігури-Основні фігури-Текстове поле (Insert-Shapes-Basic Shapes-Inscription) - окреслення габаритів фігури – вставлення всередину поля Вставлення-Колонтитули-Номер сторінки-Поточне положення (Insert-Header & Footer-Page Number-

# Current position) aбо Текст-Огляд експрес-блоків-Поле...-PAGE (Text-Express block-Field-PAGE).

#### Рамка та основний напис

Введення в документ розділів та застосування колонтитулів дозволяє оформити вбудованими засобами редактора аркуш текстового проєктного документа у відповідності до вимог Системи конструкторської документації.

Вставлення рамки аркуша та основного напису за ДСТУ ГОСТ 2.104:2006 можна провести кількома способами:

- використанням файлу малюнку рамки в якості *фону* (подложка, background);
- вставлянням малюнку рамки в нижній колонтитул аркуша;
- вставлянням в колонтитул рамки у вигляді таблиці.

Використання фону не дозволяє повністю виконати умови ДСТУ, бо фон є однаковим для всього документа.

Малюнок може бути вставлений в колонтитул із зовнішнього файлу або намальований вбудованими засобами редактора. Слід зазначити, що намальовані елементи виглядають природніше.

Для дотримання вимого ДСТУ слід для першої сторінки документа призначити окремий розділ, в який внести основний напис типу 2. Основний напис типу 2a слід вносити в колонтитули інших розділів документа.

## Зміст

Зміст (Table of contents, Оглавление) є одним з розділів документа. Розміщується зміст на першому аркуші документа. Умовою для автоматичного створення є наявність в структурних частинах документа стилів ЗаголовокХХХ (HeadingXXX). Створюється зміст вибором потрібного оформлення змісту з меню поля зміст Посилання-Зміст (References-Table of contents).

## Цитування

Редактор дозволяє напівавтоматично проставити внутрішньотекстові, позатекстові цитування за якими автоматично згенерувати список літератури.

Введені бібліографічні дані джерел зберігаються у базі даних редактора та можуть бути використані в інших документах.

Керування позначками цитуванням та базою бібліографічних даних проводиться полями групи Посилання-Посилання та бібліографія (Citations & Bibliography) (рис. 2.24).



Рис. 2.24. Вкладка цитувань

Перед вставленням позначки цитування в документ слід перенести бібліографічні дані з бази редактора в документ або ввести нові дані.

Перенесення даних з бази в документ проводиться в діалоговому вікні диспетчера джерел (рис. 2.25) шляхом копіювання відповідних даних з основного списку зліва вікна в поточний список справа вікна.

Диспетчер джерел	? ×
Знайти:	Сортування за автором
Джерела, доступні в: Основний список	П <u>о</u> точний список
Положення про випускну атестацію студентів КПІ імені Ігоря Сікорськ Аnthony, Carl Sterrazza, America's First Families: An Inside View of 200 Yr, Meringolo, Avis; First Ladies Political Role and Public Image (1990) Брукцир Глен, Брилов Денніс; Комп' істерні науки. Базовий курс: 13-е Зигель Р., Хауэлл Дж.; Теплообмен излучением (1975) Редагувати	Положення про випускну атестацію студентів КПІ імені Ігоря Сікорсь 🧄 Брукшир Глен, Брилов Денніс Комп' ютерні науки. Базовий курс. 13-
	<ul> <li>цитоване джерело</li> <li>джерело покажчика місця заповнення</li> </ul>
Попередній перегляд (IEEE):	
Посилання: [[index]]	^
Елемент бібліографії:	

Рис. 2.25. Диспетчер джерел

Диспетчер активується пунктом Посилання-Посилання і бібліографія-Керування джерелами (References-Citation&Bibliography-Manage Sources-New...) (рис. 2.24).

Створення бібліографічного опису нового джерела проводиться в діалоговому вікні (рис. 2.26).

Вікно створення активується пунктом Посилання-Посилання і бібліографія-Вставити елемент таблиці посилань-Додати нове джерело (References-Citation&Bibliography-Add new Source) (рис. 2.27) або пунктом Створити (New...) вікна джерел.

Позначки цитування оформлюються відповідно до вимог нормативних документів, які визначаються полем Стиль (Style) стрічки.

Створення джерела						?	×
<u>Т</u> ип,	джерела <mark>Книга</mark>	~ ·	<u>М</u> ова	За промовчанням			~
Поля бібліографії для ІЕ	EE						
Автор					Pe	дагувати	
	Організація Автор						
Заголовок							
Рік							
Місто							
Видавець							
Відображати всі біблі	ографічні поля						

Рис. 2.26. Вікно створення джерела

Слід зауважити, що вбудований пункт оформлення цитат **ГОСТ** ... НЕ ВІДПОВІДАЄ вимогам. Найближче до вимог ДСТУ 3008-2015 є стиль **ІЕЕЕ**.

Для проставляння позначки цитування слід встановити курсор в потрібне місце та обрати з поточного списку поля Посилання-Посилання і бібліографія-Вставити елемент таблиці посилань (References-Citation&Bibliography-Insert Citation) (рис. 2.27) потрібне джерело.



Рис. 2. 27. Джерела для цитування

Редагувати кількість інформації позначки цитування та змінити джерело можна редагувати пунктом **Змінити джерело (Edit Citation)** контекстного меню посилання (рис. 2.28 а). Меню дозволяє керувати інформацією про авторів, рік видання, сторінку цитати в джерелі (рис. 2.28 б).

	Змінити елемент таблиці п	? X
Змінити елемент таблиці посилань	Додати <u>С</u> торінки:	
<u>З</u> мінити джерело	Приховати Автор <u>Р</u> ік	<u>З</u> аголовок
Перетворити елемент таблиці посилань на статичний текст 	ОК	Скасувати
a	б	

Рис. 2.28. Редагування цитування: а – виклик; б – меню

За умови позначення цитування вбудованими засобами список використаної літератури створюється напівавтоматично пунктом Посилання-Посилання і бібліографія -Бібліографія (References-Bibliography) стрічки.

Список літератури та позначки цитування можна оновити після внесення змін в документ пунктом Оновити посилання в бібліографію (Update Citations and Bibliography) (рис. 2.28 a) контекстного меню позначки цитування.

Правила зображення елементів у вбудованих стилях цитування розташовуються за адресою

«...\Users\[UserName]\AppData\Roaming\Microsoft\Bibliography\Style»

Підрядні позначки цитування проставляються в редакторі як виноски (сноска, footnote). Вставлення виноски проводиться пунктом Посилання-Виноски-Вставити виноску (References-Footnotes) стрічки (рис. 2.29).

Виноски НЕ ВРАХОВУЮТЬСЯ АВТОМАТИЧНО В БАЗУ ДЖЕРЕЛ та СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.



## Рис. 2.29. Виноски

Формат виносок редактора не відповідає вимогам ДСТУ 3008:2015 по розміру шрифту, міжрядковому інтервалу, позначці виноски та потребує після вставляння ручного коригування.

#### Таблиці

*Таблиці* є способом візуалізації регулярної інформації, яка може бути структурована по рядках та стовпцях.

Операції роботи з таблицями в редакторі є зрозумілими та уніфікованими з табличним процесором *MS Excel*. Компонентам таблиці: стовпцям, рядкам, окремим коміркам можуть мати власні формати значень, рамки, кольори та інші елементи декору.

Особливістю редактора є те, що таблиці є не тільки способом візуалізації введених, але й засобом проведення розрахунків. Над значеннями в комірках таблиці можна проводити математичні операції.

Створення таблиць проводиться визначенням потрібної кількості рядків та стовпців з випадаючого меню стрічки Вставлення-Таблиці-Вставити таблицю (Insert-Tables-Add a Table). Форматування, редагування таблиці проводиться пунктами контекстного меню таблиці або з вкладок Конструктор (Design) та Макет (Layout) стрічки, які автоматично активуються при виділенні таблиці.

Для активації контекстних вкладок стрічки достатньо клікнути на будьякому компоненті таблиці. Виділити всю таблицю можна кліком на позначці «хрест» +в правому верхньому куті таблиці.

Арифметичні вирази в таблиці вводяться у вікні формул (рис. 2.30 а). Вікно активується пунктом **Макет-Дані-Формула** (Layout-Dataformula) стрічки (рис. 2.30 б). Адреси комірок таблиці не відображаються на екрані, їх треба визначати за правилами адресації А1 (літера стовпця-число рядка) *MS Excel* та вводити вручну. Формульні вирази вводяться в поле **Формули** після знаку рівності в синтаксисі *MS Excel*. Арифметичні дії позначаються звичними символами, імена вбудованих функцій обираються зі списку Вставлення функцій або вводяться вручну латиницею.

Автоматично розраховується значення в комірці тільки після введення формули. Запуск повторних розрахунків проводиться клавішею «*F9*» вручну після виділення потрібної комірки або всієї таблиці.

Формула			×
<u>Ф</u> ормула:			
=			
Ф <u>о</u> рмат номерів:			
			~
<u>В</u> ставити функцію:	Вставити закладку:		
×			$\sim$
ABS			
AND		Скасу	вати

Рис. 2.30. Розрахунки в таблиці: а – вікно формул; б – виклик вікна формул

# Складені документи

Орієнтуватися в документі, обсяг якого сягає кількох сотень сторінок, стає важко. Для роботи з такими документами в редакторі реалізовано технологію *складених документів* (Complex document).

Складений документ – це документ, який складається з кількох файлів. Зберігається складений документ у вигляді головного файлу та файлів, кожен з яких містить один розділ. При цьому файли розділів можуть редагуватися незалежно від основного файлу.

Операціями зі складеним документом проводяться з контекстної вкладки Структура (Outline) (рис. 2.31) стрічки. Вкладка активується полем Вигляд-Структура (View-Outline) стрічки.



Рис. 2.31. Вкладка структури

Складений документ може створюватися декількома способами.

Створити складений документ можна розбиттям файлу загального первинного документа на головний та файли розділів вбудованими засобами редактора. При цьому головний файл отримує назву файлу первинного документа, а файли розділів набувають назву відповідних розділів.

Створити складений документ можна додаванням до первинного головного файлу зовнішніх файлів. При цьому документи зовнішніх файлів можуть і не мати розділів, самі зовнішні файли можуть мати довільні імена. Вставляння кожного зовнішнього файлу призводить до створення в головному документі нового розділу. Зміст розділу співпадає зі змістом документу зовнішнього файлу.

Складений документ можна редагувати без усяких обмежень.

Зображується складений документ в редакторі у вигляді тексту головного документа з гіперпосиланнями на файли розділів.

Вигляд складеного документу в режимі структури нагадує веб-документ з включеними недрукованими символами, якими на екрані зображуються межі розділів. Інформація в документі складається з основного тексту та заголовків розділів рівня від 1 до 9. Розділи мають рівень 1, підрозділи рівень 2 тощо.

Рядки заголовків розділів всіх рівнів (розділи, підрозділи, параграфи тощо) помічаються на екрані маркером 👻. Клік на маркері заголовку виділяє весь розділ. Подвійний клік на маркері відкриває або ховає текст відповідного розділу на екрані. Аналогічну дію виконують кнопки «+» та «-» стрічки.

Деталізацію виведення на екран інформації визначає елемент списку Структура-Знаряддя структури-Показати рівень (Outline tools-Show level). Він визначає мінімальний рівень розділів, заголовки яких будуть виведені на екран. Визначення цього значення призводить до того, що на екран будуть виведені тільки заголовки розділів визначеного та вищих за визначений рівнів. Для того, щоб вивести на екран весь текст всього документа слід обрати елемент списку Всі рівні (All levels).

Кнопками стрічки у вигляді трикутних стрілок можна пересувати обраний розділ вгору та вниз по дереву структури документа.

Поле списку **Рівень** (Level) стрічки показує рівень розділу. Стрілками зліва та справа від поля можна підвищити або знизити рівень розділу.

Перші рядки абзаців тексту позначаються на екрані сірим колом зліва абзацу. Для зменшення обсягу інформації та спрощення навігації полем Структура-Знаряддя структури-Показати лише перший рядок (Outline tools-Show first line only) стрічки можна виводити на екрані тільки перший рядок абзаців.

Кнопка Головний документ-Показати документ (Master Document-Show Document) активує заблоковані за замовчанням кнопки групи Головний документ: Створити (Create), Вставити (Insert), Заблокувати документ (Lock Document)

Кнопки Скасувати зв'язок (Unlink), Об'єднати (Merge), Розділити (Divide) є контекстними. Вони активуються за умови виділення якогось розділу.

Кнопка Згорнути вкладені документи (Collapse Subdocuments) керує виглядом на екрані головного документа. Якщо вийти з режиму структури з вимкненою кнопкою, то документ буде показаний у

116

вигляді повного тексту. Вихід з режиму з включеною кнопкою покаже документ у вигляді гіперпосилань на файли розділів.

Кнопка Створити використовується для перенесення обраного розділу в зовнішній файл. Розділ перетворюється на зовнішній файл з іменем заголовка розділу.

Кнопка Вставити вставляє зовнішній файл формату .*docx* в структуру документа. Зовнішні файли помічаються позначкою поруч з заголовком розділу.

Кнопка **Розірвати зв'язок (Unlink**) розриває зв'язок головного документа із зовнішнім файлом. При цьому інформація зовнішнього файлу переноситься в головний файл документа.

# Контрольні запитання

- 1. Що таке Backstage view ?
- 2. Як швидко згорнути/розгорнути Ribbon?
- 3. Наведіть синонім до терміну KeyTips.
- 4. Що таке і для чого призначений dialog box launcher?
- 5. Що таке shortcut menu ?
- 6. Що таке nonprinting characters?
- 7. Для чого призначено Auto Complete?
- 8. Що таке Word Wrap?
- 9. Що таке *template*?
- 10.Які існують типи відступів абзаців?

11.Що таке "кегль"?

- 12. Які є режими вставляння?
- 13.Назвіть типи та родини шрифтів.
- 14.Як вставити основний напис?
- 15.Як проставити найпростіше нумерацію сторінок?
- 16.В чому невідповідність MS Word та ДСТУ 3008:2015 для виносок?
- 17.Як оформити посилання в MS Word за ДСТУ 3008:2015?
- 18.Як змінити тип табуляції?
- 19. Як ввести горизонтальну лінію?
- 20.Що таке widow, orphan?
- 21. Як ввести арифметичні вирази в таблицю?
- 22. Як автоматично створити зміст?
- 23.За якою ознакою розбивається складений документ?

# 2.3. Табличний процесор MS Excel

Практично у всіх царинах бізнесової або науково-технічної діяльності людини виникає потреба в обробці інформації у вигляді таблиць, в яких частина даних може змінюватися, частина є розрахованою по формулах.

Для проведення такої обробки на комп'ютері слугують електронні таблиці. Електронні таблиці – це двовимірні масиви даних, які згруповані в стовпці та рядки. Електронна таблиця є комп'ютерним еквівалентом звичайній таблиці, в комірках якої знаходяться або первинні дані (числа, тексти, дати, формули тощо), або результати розрахунків по формулах. При змінах первинних даних перерахунок по формулах проводиться автоматично. Автоматичний миттєвий перерахунок всіх даних, які пов'язані формульними залежностями при змінах будь-якого операнда, є безперечним достоїнством електронних таблиць.

Обробляння електронних таблиць проводять спеціальні програми – *табличні процесори* (Spreadsheet program).

В першу чергу табличні процесори орієновані на використання в економічних, бухгалтерських розрахунках тощо.

Табличні процесори забезпечують:

- введення, зберігання та редагування табличних даних;
- автоматичні перерахунки по формульних залежностях;
- наочну та природню форму надання документів на екрані;
- ілюстрацію табличних даних у вигляді діаграм та графіків:
- створення підсумкових та зведених таблиць;
- сортування та вибірку даних по різноманітних запитах;
- обмін даними з іншими застосунками;
- розробку та виконання програм автоматизації дій з даними в таблицях тощо.

Піонерами в розробці табличних процесорів вважаються Ден Бриклін (Dan Bricklin) та Боб Франкстон (Bob Frankston). В 1979 році тоді студенти Гарвардського університету для спрощення бухгалтерських робіт розробили для комп'ютерів *Apple* програму *VisiCalc*.

Популярність програми VisiCalc та її нащадка – програми SuperCalc, привело до появи в 1982 р. табличного процесора нового покоління Lotus 1-2-3 для комп'ютерів фірми IBM. В Lotus 1-2-3 вперше було інтегровано графічні можливості та засоби обміну з СУБД. Нащадками Lotus 1-2-3 стали табличні процесори VP Planner від компанії Paperback Software та Quattro Pro від компанії Borland International.

Табличний процесор *Excel* від фірми *Microsoft* з'явився в 1987 р.. Простий графічний інтерфейс та функціональні можливості зробили сьогодні *MS Excel* світовим лідером в даному класі програмного забезпечення з обсягом ринку біля 80 %.

# Інтерфейс програми

Інтерфейс *MS Excel* є типовим для всіх застосунків *MS Office*.

Вікно програми містить органи керування режимами роботи, діями та робоче поле документа (рис. 2.32). Для керування процесором призначені *панель швидкого доступу* (Quick Access Toolbar), *стрічка* (Ribbon), *рядок стану* (Status bar), *поле назв* (Name Box), *рядок формул* (Formula bar).

ΦΑΪ	5	OCHOBHE	> г	Танель ня роз	<b>ШВИДКОГО Д</b> МІТКА СТОРІНІ	оступу и форм	УЛИ ЛАН	K I PELIE	нига1 - Excel нзування	виглял	РОЗРОБНИ	IK .	Стріч	ка	?	<b>A</b> –	□ × Увійти
Встав Буфер	ити 💉	- Calibri Ж К	-  11 □ -   ⊞ +   4 Шрифт	• A* A* • <u>A</u> •	= = <b>■</b> » ≡ = = •	• 🔐	Загальний 😨 - % осо Число	*   :08 :08	умовне форматуван	Формат ня - таблиці Стилі	Стилі клітинок *	<ul> <li>Встави</li> <li>Видали</li> <li>Форма</li> <li>Кліти</li> </ul>	ти т ити т атувати т	∑ - Ал ↓ - Я ↓ - Сортув фільтру Реда	т найтий затий Знайтий вати • виділити • агування	Макроса ту	
G7		>	$\times \checkmark$	$f_x$			Рядок	формул									~
1	A	В	С	D	E	F	G	н	1	J	к	L	м	N	0	P	Q *
1 2 3 4 5 6 7 8		Поле	назв				Робо	че поле	•								
9 10 11 12 13																	
готов	30 🛅	Аркуш	1 (+)			Рядо	ок стану								<u> </u>	+	100%

Рис. 2.32. Інтерфейс *MS Excel* 

Стрічка може відображатися з усіма елементуми, тільки вкладками або не відображатися зовсім. Режим відображення стрічки змінюється пунктами кнопки Параметри відображення стрічки (Ribbon view control), контекстним меню стрічки (рис. 2.6 б) або комбінацією клавіш «CTRL+F1»,

Для оперативної зміни вигляду органу керування слугують спливаючі контекстні меню елементів (Shortcut menu).Склад меню змінюється в залежності від дій, які можна виконати з обраним елементом. Відкривається контекстне меню комбінацією клавіш «SHIFT+F10» або кліком правої кнопки миші над обраним елементом.

*Рядок стану* знаходиться внизу робочого вікна. В рядку стану виводиться основна інформація про документ та містяться деякі елементи керування. Склад елементів, які виводяться в рядок стану, можна змінювати в контекстному меню рядка. За замовчанням в рядку знаходяться кнопки зображення документу (View controls) та кнопки керування масштабом зображення документу (Zoom controls). Склад елементів рядка статусу можна змінювати в контекстному меню рядка.

Панель швидкого доступу містить кнопки команд, які виконуються найчастіше. За замовчанням на панелі розташовані кнопки збереження документу, відміни останніх дій, повторення останньої дії. Склад кнопок панелі можна змінювати. Наприклад, сюди зручно додавати кнопки запуску *макросів користувача*, для автоматизації запуску часто використовуваних власних додатків. Налаштовується панель в діалоговому вікні налаштувань. Вікно налаштувань активується пунктом **Настроїти панель швидкого доступу (Customize Quick Access Toolbar)** контекстного меню стрічки або пунктом **Файл-Параметри-Панель швидкого доступу (File-Options-Quick Access Toolbar)** стрічки.

Документи *MS Excel* називаються *робочими книгами* (workbook). Документи зберігаються в файлах (spreadsheet file) з розширенням *.xls*, *.xlsx*, *.xlsm*, *.xlt*, *.xltm*, *.xltx*, *xlsb*. Кожен документ може містити кілька аркушів (worksheet) з таблицями, діаграмами, макросами тощо.

На робочому полі редактора зображуються сторінки документа. Документ може бути зображений в режимі **Звичайний** (Normal), режимі **Размітка сторінки** (Page Layout), режимі **Макет сторінки** (Page Break Preview). Режим **Звичайний** вважається найбільш зручним для повсякденної роботи без необхідності друку документів.

Сторінки документа помічаються внизу робочого поля закладками. За замовчанням сторінки мають імена **Аркуш1**, **Аркуш2** (Sheet1, Sheet2) тощо. Ім'я та положення сторінки можна змінювати через відповідний пункт контекстного меню сторінки.

В полі назв Ім'я (Name Box) відображається адреса поточної комірки та власні імена комірок.

В рядку формул (Formula bar) відображається зміст поточної комірки та вводяться формульні вирази для проведення обчислень в активній комірці.

## Адресація комірок

Електронну таблицю створюють рядки (row) та стовпці (column), які мають свої імена – адреси. Перетин рядка та стовпця показує комірку таблиці (cell). Для вказання на комірку вказують її адресу – імена рядка та стовпця, на перетині яких знаходиться комірка.

Група суміжних комірок називається блоком комірок (range of cells).

Комірка, в якій знаходиться курсор, називається *поточною* або *активною* (active cell). *Активна* комірка виділяється на екрані товстою рамкою. Зміст *активної* комірки показується в рядку формул.

Адреси в *MS Excel* можуть бути визначені двома способами: «традиційним» або «A1» та «програмним» або «R1C1». В обох способах адреса комірки отримується поєднанням адрес відповідних рядка та стовпця.

В «програмному» стилі спочатку вказується ім'я рядка, потім – стовпця. Імена рядків складаються з літери R (Row) та номера рядка. Імена стовпців складаються з літери C (Column) та номера стовпця. Нумерація рядків та стовпців починається з 1.

Завдяки універсальності абсолютної адресації та відносної адресації комірки відносно поточної саме «програмним» стилем визначаються адреси в макросах та додатках. В номерах число без дужок визначає абсолютний номер рядка/стовпця. Наприклад, адреса комірки на перетині другого стовпця та другого рядка записується як **R2C2**. Число в квадратних дужках визначає зсув відносно поточної комірки. Наприклад, адреса **R**[1]C[-1] визначає комірку, яка зсунута на 1 рядок вниз та 1 стовпець вліво відносно поточної.. Символ без індексу визначають весь поточний рядок/стовпець. Наприклад, **RC** визначає поточну комірку, **RC**[3] визначає комірку в поточному рядку, яка зсунута на 3 стовпці вправо.

«Традиційна» адреса на першому місці містить ім'я стовпця, на другому – рядка. Імена рядків визначаються їхніми номерами, починаючи з 1. Імена стовпців визначаються літерами латинського алфавіту від A до Z, потім від AA до AZ, потім від BA до BZ тощо.

Ідентифікація комірок проводиться їхнею адресою перетину відповідних рядка та стовпця, наприклад «традиційна» адреса, **B2**.означає комірку на перетині другого стовпця з адресою **B** та другого рядка з адресою **2**, що еквівалентно адресі **R2C2** «програмним» стилем.

121

Адреса блоку визначається адресами комірки лівого верхнього кута блоку та комірки нижнього правого кута блоку. Між адресами ставиться знак двокрапки «:». Наприклад, **A1:C3**.

Повна адреса комірки складається з назви книги, назви аркуша та адреси комірки або групи комірок на аркуші. Наприклад, Sheet3!R3C3, Аркуш1!A1:C3, [Книга.xlsx]Аркуш!B3.

Встановлюється стиль адресації полем Файл-Параметри-Формули-Робота з формулами-Стиль посилань (File-Options-Formulas-R1C1 reference style) стрічки (рис. 2.33 а).



Рис. 2.33. Визначення способу адресації: а – визначення; б – «традиційна»; в – «програмна»

Поточний стиль адресації легко визначити по вигляду робочого поля. Для «традиційної» адресації в рядку адрес стовпців виводяться латинські літери (рис. 2.33 б). Для «програмної» адресації в рядку адрес стовпців виводяться числа (рис. 2.33 в).

Використання стилю адресації, який відмінний від поточного, викликає помилку.

#### Формули

Формули в MS Excel являють собою сукупність операторів дій та операндів. Операторами слугують символи арифметичних та логічних дій, імена вбудованих функцій та функцій користувача. В якості операндів (аргументів) використовуються числа та адреси комірок, в яких знаходяться дані.

Результат обчислення за формулою розміщується в тій же комірці, в якій розміщено формулу. В комірках з формулами відображається тільки результат, самі формули відображаються за замовчанням в разі виділення комірки в рядку формул.

Порядок дій в формулах без дужок є традиційним: логічне заперечення, вбудована функція, піднесення до степеню, множення та ділення, додавання

та віднімання, конкатенація рядків, порівняння (=, <>, <=, >=, <, >). Дії проводяться зліва-направо.

Розрахункова залежність в комірку вводиться після знаку прирівнювання. Тому наявність символу прирівнювання першим елементом в комірці слугує ознакою проведення обчислень в комірці.

Наприклад, для «традиційної» адресації формула в комірці **АЗ** з добутком вмістів комірок **А1** та **А2** має вигляд **=А1\*А2**.

Для програмного стилю комірка **A3** має адресу **R3C1**, формула має вигляд =**R**[-2]C+**R**[-2]C[1]. Перший множник шукається в комірці на два рядки вище поточного стовпця (комірка **A1~R1C1**). Другий множник шукається в комірці на один рядок вище поточного стовпця (комірка **A2~R1C2**).

Імена вбудованих функцій можуть вводитися вручну в рядку формул або безпосередньо в комірці або обиратися зі списків вікна вбудованих функцій (рис. 2.33 а) чи списків функцій відповідного типу групи Формули-Бібліотека функцій (Formulas-Function Library) стрічки (рис. 2.33 б). Вікно вбудованих функцій активується кнопкою Бетавити функцію (Formulas-Insert Function) стрічки (рис. 2.33 б).

ставлення функції	? ×	
Пошук функції:		
Введіть короткий опис дії, яку бажаєте виконати, і натисніть кнопку "Знайти"	З <u>н</u> айти	
Категорія: Усі 🗸		
<u>В</u> иберіть функцію:		
ABS	^	
ACCRINT		ФАЙЛ ОСНОВНЕ ВСТАВЛЕННЯ РОЗМІТКА СТОРІНКИ ФОР
ACCRINTM		
ACOS		🖌 🕹 Автосума 👻 🔢 Логічна 👻 Посилання та м
ACOSH		🕖 🗸 Недавно використані т 🔼 Текстова т 🛛 🖯 Математична т
ACOT		Вставити
ACOTH	*	функцію 💷 Фінансова т 🔛 Дата та час т 🔛 Інші функції т
АВS(число)		Бібліотека функцій
Повертає модуль (абсолютне значення) числа, тобто число без знака.		$-$ A1 $\cdot$ : $\times \sqrt{f_x}$

Рис. 2.33. Списки функцій: а – віконний; б – стрічки

Імена функцій є локалізованими. Тобто вводяться мовою, яка співпадає з мовою локалізації *MS Excel*. Нажаль для української мови локалізація не проведена та імена функцій вводяться англійською або російською мовою.

Формульні залежності можуть обраховуватися автоматично або вручну. За замовчанням формули розраховуються автоматично при їхньому введенні та автоматично перераховуються при зміні будь-якого з операндів формули. Для поточного сеансу режим обчислення визначається полями формулиобчислень (Formulas-Calculation-Обчислення-Параметри Calculation **Options**) стрічки (рис. 2.34 а). Глобальний режим Файл-Властивості-Формулиобчислення визначається полями Параметри обчислень-Обчислення в книзі (File-Options-Formulas-Calculation Options-Calculations in Workbook) стрічки (рис. 2.34 б).



Рис. 2.34. Визначення режимів обчислення: а – локально; б – глобально

#### Особливості адресації комірок в формулах

Незалежно від стилю адресації в формулах використовується *відносна* від поточної адресація комірок.

При копіюванні комірок в нове місце в формулах автоматично перераховуються адреси комірок таким чином, що ЗСУВИ КОМІРОК ОПЕРАНДІВ ВІДНОСНО ПОТОЧНОЇ КОМІРКИ НЕ ЗМІНЮЮТЬСЯ.

Для «традиційних» адрес операндів в формулах введені абсолютні (absolute reference), абсолютні по рядку або стовпцю та відносні (relative reference) адреси.

*Абсолютна адреса* – це незмінна при копіюванні адреса. Позначається абсолютна адреса символом «\$». Наприклад, **\$B\$5** – абсолютне посилання, **B\$5** – абсолютне по рядку 5, **\$B5** – абсолютне по стовпцю В.

*Відносна* – це адреса, яка змінюється при копіюванні. Наприклад, «традиційна» адреса В5 є відносною та буде змінюватися при копіюванні.

Швидко змінити тип адреси можна натискання клавіші «F4» на операнді. Зміна проводиться циклічно. Наприклад, \$A\$1-\$A1-\$A\$1.

Подвійний клік на комірці з формулою виділяє кольоровими рамками комірки операндів на екрані.

Аргументи функцій можуть визначатися послідовністю адрес комірок через крапку з комою (наприклад, A1;B3;E6), адресою блока комірок (наприклад, A1:F5), поєднанням блоків комірок через крапку з комою (наприклад, A1:A5;C1:C5), перетином блоків комірок через пробіл (наприклад, B1:B7 A7:D7).

#### Вхідні дані

В комірках можуть міститися символьні (текстові), чисельні дані, формули, дати.

Вигляд змісту комірок визначається типом даних та форматом відображення, який задано для них.

Основними є наступні формати:

- Загальний (General). Формат є універсальним, бо може застосовуватися і для чисел, і для текстів. Дані відображаються в тому вигляді, як вводилися. Формат Загальний є форматом комірок за замовчанням.
- **Числовий (Number)**. Формат застосовується для числових даних. Дані відображаються з фіксованою кількістю десяткових знаків
- Грошовий (Currency). Формат застосовується для числових даних. Грошовий формат є різновидом загального. Для чисельних даних кожні три розряди числа відокремлюються комою. Після значення проставляється позначка грошової одиниці.
- Відсотковий (Percentage). Формат застосовується для числових даних. В відсотковому форматі виводиться помножене на 100 чисельне значення. Після значення проставляється позначка відсотків.
- Експоненційний (Scientific). Формат застосовується для числових даних. Виводить дані у *стандартній* формі з плаваючою точкою х.ххх·10<sup>xx</sup>. Застосовується для показу дуже великих або дуже маленьких чисел.
- **Текстовий (Text)**. Формат застосовується для текстових даних. Дані відображаються так, як вводяться. Обробляються дані як рядки символів.

Оперативно змінюється формат даних в комірці елементом списку групи стрічки Основне-Число-Формат (Home-Number) (рис. 2.35 а). Більш детально формат даних в комірці визначається в діалоговому вікні форматування комірок Формат комірок (Cell Format) (рис. 2.35 б). Вікно активується відповідним пунктом контекстного меню комірки (рис. 2.35 в), кнопкою запуску діалогового вікна *Dialog Box Launcer* групи стрічки Основне-Число (Home-Number) або елементом Інші формати списку Основне-Число-Формат (рис. 2.35 а).



Рис. 2.35. Формати: а – діалогове вікно; б – стрічка; в – контекстне меню

Користувач може задати свій власний формат через пункт (усі формати) (Custom). Формат визначається шаблоном у вигляді послідовності символів діезу для символу, нуля для цифри, коми, пробілу, знаку відсотків тощо.

Колір записується службовим словом мовою локалізації в квадратних дужках. За наявності символу пробілу за шаблоном числа виводиться значення в тисячах. Символ «\_» розсуває символи на екрані в місці свого встановлення. Інші символи з шаблону виводяться на додаток від значення числа.

За замовчанням приймається:

- ширина комірки 9 символів;
- ліве вирівнювання для символьних даних;
- праве вирівнювання для чисельних даних та основного формату;
- дані вводяться в основному форматі.

Швидко змінити кількість знаків в числах в комірках для відображення можна кнопками <sup>3</sup> <sup>3</sup> <sup>3</sup> групи Основне-Число (Home-Number) стрічки.

В якості десяткового роздільника дійсних чисел за замовчанням використовується системний символ, який обрано в налаштуваннях

регіональних стандартів системи. Символ легко змінюється кнопкою **г**рупи Основне-Число (Home-Number) стрічки.

Слід пам'ятати, що використання роздільника у вигляді символу коми суттєво зменшує сумісність файлів для використання в розрахункових системах на кшталт СКМ та АСП.

#### Введення послідовностей

Суттєве спрощення введення послідовностей на кшталт арифметичних прогресій за рахунок автоматичного визначення змісту комірок та їхнього форматування забезпечує *режим заповнення* (Auto Fill).

Вмикається режим заповнення полем Основне-Редагування-Заповнити (Home-Editing-Fill) стрічки (рис. 2.36 а) або позначкою заповнення (fill handle) у вигляді хреста «+» в правому нижньому куті виділених комірок (рис. 2.36 б).



Рис. 2.36. Режим заповнення: а – стрічка; б – позначка; в – контекстне меню

Для числових послідовностей або послідовностей термінів (natural series) найпростішим є використання позначки заповнення. Вручну потрібно заповнити дві комірки, наприклад, «Понеділок Вівторок» чи «1 2», та потягнути за позначку заповнення. Значення в нових комірках вносяться автоматично. Наприклад, «Понеділок Вівторок Середа Четвер» чи «1 2 3 4». Режим заповнення працює тільки для термінів мовою локалізації пакета.

Після проведення дії користувач має змогу в контекстному меню (рис. 2.36 в) обрати внесення в нові комірки змінених значень та форматів (Fill Series), внесення в нові комірки значень без змін з форматами Копіювати клітинки (Copy Cells), внесення в нові комірки тільки форматів даних Заповнити лише формати (Fill Formatting Only) або внести тільки значення Заповнити лише значення (Fill Without Formatting). Для перенесення формату обраної комірки чи блока комірок на нову комірку чи блок комірок слугує *дія формату за зразком*. Режим активується

кнопкою **с**трічки Основне-Буфер обміну-Формат за зразком (Home-Clipboard-Format Painter). Ознакою режиму є курсор у вигляді пензля. Дія проводиться в два кроки. Спочатку активація комірки, з якої буде братися формат та натискання кнопки режиму. Далі кліками миші в обраних комірках формат переноситься на нові комірки.

Вставляння інформації з буферу обміну може проводитися значно гнучкіше, ніж повне копіювання даних, формул, формату та коментарів клавішами «*Ctrl+V*».

Пункт спеціального вставляння Основне-Буфер обміну-Спеціальне вставляння (Home-Clipboard-Paste special...) відкриває діалогове вікно, в якому можна провести тонке налаштування параметрів вставляння.

Дії кнопок вставляння наведені нижче:

- *Б* вставляння тільки формули, формат та коментарі не переноситься
- 123 вставляння тільки результату, формула, формат та коментарі не переносяться
  - вставляння тільки формату комірки
- 123

**%** 

- вставляння результату та формату комірки, формула та коментар не переносяться
- вставляння посилання у вигляді абсолютної адреси комірки

١

**2**1

Ċ.

вставляння малюнка

вставляння малюнка у вигляді адреси комірки-джерела зі збереженням зв'язку

вставляння блоку комірок з транспонуванням

# Оформлення аркуша

Для приведення формул до більш традиційного та у випадках багаторазового використання комірок, слушно надати коміркам *імена* (Naming a Range).

Наявні в книзі імена комірок виводяться в *поле імен* (рис. 2.32) у вигляді списку. З цього списку кліком можна обрати ім'я комірки для вставляння в формулу.

Для надання імені комірці достатньо зробити її поточною та в полі імен замість адреси ввести потрібне ім'я.

Вбудовані інструменти для роботи з поіменованими коміркам розміщені в група **Формули-Визначені імена (Formulas-Defined Names**) стрічки (рис. 2.37 а).



Рис. 2.37. Поіменовані комірки: а – стрічка; б – вікно надання імені; в – контекстне меню

Ввести поіменовану комірку та визначити для неї область доступності імені Область (Scope), адресу Посилання (Refers to) та додати коментар Примітка (Comment) можна в діалоговому вікні Нове ім'я (New name) (рис. 2.37 б). Активується вікно полем Визначити ім'я (Define name) стрічки, пунктом контекстного меню комірки (рис. 2.47 в).

Повний контроль над усіма поіменованими комірками здійснюється в диспетчері імен (рис. 2.38). Диспетчер активується полем Формули-Визначені імена-Диспетчер імен (Formulas-Defined Names-Names Manager) стрічки (рис. 2.37 а).

Імена комірок не мають мати довжину більше 255 символів та містити в собі символи «R», «C» та пробілу.

Імена, які визначені для аркуша вважаються локальними. Імена для робочої книги вважаються глобальними.

Диспетчер ім	ен					?	×	
<u>С</u> творити	<u>С</u> творити <u>Р</u> едагувати <u>В</u> идалити							
Ім'я	Значення Посилання		Область	Примітка				
🗐 a	5	=Аркуш1!\$Н	Робоча					
💷 b	{"1","2","3"}	=Аркуш1!\$D	Робоча					
<u>П</u> осилання:								
× 🗸 =Ap	окуш1!\$H\$3							
						Закр	ити	

Рис. 2.38. Диспетчер поіменованих комірок

Таблиці з великою кількістю рядків та/або стовпців не вміщуються на екран. Для їхнього перегляду потребується пересувати аркуш. Таблиці можуть містити другорядні дані та проміжні результати. Такі комірки бажано не показувати на екрані в готовій таблиці.

Щоб тимчасово прибрати рядки або стопці з таблиці можна застосувати дію «сховати/показати». Дія виконується для обраних рядків/стовпців відповідним пунктом контекстного меню.

Вказаний спосіб має недолік. Сховані елементи не помічаються в таблиці. Знайти їх можна аналізом номерів рядків/стовпців.

Іншим способом прибирання елементів з екрану є їхнє *групування*. Груповані комірки позначаються на екрані позначкою «±» в полі номерів комірок (рис. 2.39). Натискання позначки циклічно ховає/показує елементи групи.

	9		
	10	Actual Expenses	
	11		Totals
+	18	Total Expenses per Day	\$1,634.50
	19	the second data and the second data	
	20	Reimbursable Expenses	
	21		Totals
+	28	Total Expenses per Day	\$1,041.10
	29		

Рис. 2.39. Групування комірок

Проводиться групування полем Дані-Структура-Групувати (DATA-Outline-Group) стрічки.

## Умовне форматування

Умовне форматування (Conditional formatting) є засобом візуального виділення інформації в таблиці за деяким критерієм кольором фону, стилем шрифту комірки тощо без зміни вмісту самої таблиці (рис. 2.40).

	A		В		с	D		E	F	G	
1	City ,	đ	Jan 🔤	•	Feb 💌	Mar	•	Apr 💌	May 💌	Jun 💌	
2	Barstow		8	80	84	8	34	97	95	98	
3	California City		7	8	86	8	34	96	98	102	
4	Cinco		8	33	86	8	36	97	95	103	
5	Hesperia		7	8	85	8	37	98	97	102	
6	Lancaster		7	8	85	8	36	99	95	101	
7	Mojave		8	32	85	8	36	98	96	99	
8	Palmdale		8	31	84	8	35	97	95	101	
9	Ridgecrest		8	81	87	8	37	97	96	98	
10	Rosamond		8	32	86	8	38	99	97	101	
11	Santa Clarita		7	9	85	8	37	95	96	103	

Рис. 2.40. Умовне форматування

Керується умовне форматування меню кнопки Основне-Стилі-Умовне форматування (Home-Style-Conditional Formatting) стрічки (рис. 2.41 а).

Умовне Формат Стилі Формат Стилі Форматувати - Стилі Форматувати - Стилі	А Виберіть тип правила:	? X
равила виділення клітинок фило Правила виділення клітинок   Правила для визначення перших і останніх елементів	• Форматувати всі клітинки на основі їх значень     • Форматувати лише клітинки, які містать     • Форматувати лише перші або останні значення     • Форматувати лише значення, більші або менші за середнє     • Форматувати лише значальні або повтоокрані значення	
Гістограми	Використовувати формулу для визначення клітинок для форматування     Редагувати опис правила:     Форматувати всі клітиник на основі їх значень:	
Кольорові шкали Набори піктограм	<ul> <li>Стұль формату: Двоколірна шкала</li> <li>Мінімальне значення</li> <li>Тип: Найменше значення</li> <li>Найбілі</li> </ul>	альне значення
<ul> <li>Створити правило</li> <li>Очистити правила</li> <li>Керувати правилами</li> </ul>	у Снайменше значення) 🛣 (Найблика) Колір: У Попередній перегляд;	њше значення) 🔝 🗸
a	б	

Рис. 2.41. Управління умовним форматуванням: а – стрічка; б – створення правил користувача

Можна обрати одне з вбудованих правил або створити та застосувати своє одне чи кілька правил одночасно. Більшість вбудованих критеріїв орієнтована на аналіз комірок з числовим вмістом.

Правила застосовуються лише до раніше виділеного діапазону комірок.

Вбудованими правилами є:

- вставлення позначки біля комірок Набори піктограм (Icon Sets);
- вставлення діаграми значень біля комірок Гістограми (Data Bars);
- градієнте фарбування комірок Кольорові шкали (Color Scales) в залежності від їхніх значень;
- форматування комірок в залежності від результатів перевірки Правила виділення клітинок (Highlight Cells Rules)

з умовою Більше (Greater Then...), Менше (Less Then...), Дорівнює (Equal To...), Між (Between...), Текст містить (Text That Contains...), Повторювані значення (Dublicate Values...) тощо;

виділення частини комірок Правила для визначення перших
 і останніх елементів (Top|Bottom Rules); 10
 перших/останніх елементів (Top/Bottom 10
 Items...), Перші/останні 10% (Top/Bottom 10%...),
 Більше/менше середнього (Above/Below Average...), які вдовольняють критерію перевірки.

Всі правила в аркуші книги відображаються в менеджері правил (Conditional Formatting Rules Manager). Менеджер активується полем Основне-Стилі-Умовне форматування-Управління правилами... (Home-Style-Conditional Formatting-Manage Rules) стрічки (рис. 2.41 а). З менеджера можна застосувати, видалити, змінити існуючи правила форматування та викликати вікно створення правил користувача.

Диспетчер правил умовного форматування					?	×			
ображати правила форматування для: Поточний фрагмент 🔍									
<u>Створити правило</u> <u>Редагувати п</u>	равило 🗙 В <u>и</u> далити правило								
Правило (застосовується в указаному порядку)	Формат	Застосовується до		Припинити, якщо зна	чення ІСТИ	1HA			
Градуйована шкала кольорів		=\$D\$3:\$D\$8	•						

Рис. 2. 42. Менеджер правил

Правила користувача вводяться у вікні правил (рис. 2.41 б). Вікно активується полем Створити правило... (New Rule...) стрічки або кнопкою менеджера правил (рис. 2.42). У вікні з списку шаблонів правил Виберіть тип правила (Select a Rule Type), який відображається в верхній частині вікна, в полях групи Редагувати опис правила (Edit the Rule Description) описується нове правило.

## Перевірка даних при введенні

В процесорі передбачені засоби захисту від отримання невірних результатів та від аварійного завершення роботи застосунку у вигляді *перевірки даних при їхньому введенні*. Наприклад, значення довжин, маси мають бути додатними, в числах не може бути символів, телефонні номери мають мати 13 цифр тощо.

Вмикання режиму перевірки проводиться полем Дані-Знаряддя даних-Перевірка даних (DATA-Data Tools-Data Validation) (рис. 2.43).

ДАНІ	РЕЦЕНЗ	УВАННЯ	вигл	іяд	РОЗРОБ	ник		
ити сувати п	овторно	Текст за	₩ P	Литтєв Видали	е заповн ти дублік	ення ати	📴 Консолідація	
ково гр		стовпцями		1ереві Пере	рка дани: вірка дан	к т их	в¦а Зв'язки	
			ø	<u>О</u> бве	сти непри	пуст	имі дані	
;	Н	I.	Ş	<u>В</u> идал	ити обві,	цку н	еприпустимих даних	

Рис. 2.43. Перевірка даних

Параметри перевірки даних визначаються у діалоговому вікні Data Validation dialog box. Вікно має три сторінки. На сторінці Параметри (Settings) (рис. 2.44 а) в полі Тип даних (Allow) з вбудованого списку обирається тип даних, для яких потрібно проводити перевірку: цілі або дійсні числа, текстові рядки, дати, списки тощо. В полі Значення (Data) 3 вбудованого списку обирається логічна умова: в межах, за межами, дорівнює, більше, менше тощо. В залежності від обраної умови в полях Мінімум (Minimum), Максимум (Maximum) тощо задаються відповідні значення. Для типу даних **Інший** (Custom) визначається розрахункова залежність в (Formulas). Для типу даних Список полі Формула (List) в полі Джерело (Source) задаються елементи випадаючого стиску, який буде з'являтися при введенні даних.



Рис. 2.44. Діалог перевірки: а – параметри; б – повідомлення; в – помилка

На сторінці Повідомлення для вводу (Input Message) (рис. 2.44 б) записується повідомлення, яке може автоматично виводитися на екран, коли комірка стає поточною.

На сторінці **Повідомлення про помилку (Error Alert**) (рис. 2.44 в) записується повідомлення, яке буде виводитися на екран в разі помилки

# введення та в полі Вид (Style) визначається вигляд повідомлення: Зупин (Stop), Попередження (Warning), Повідомлення (Message).

# Обмін даними

Для розв'язання науково-технічних задач часто виникає необхідність обміну даними, зокрема імпорт даних, між *MS Excel* та іншими застосунками. Наприклад, зчитування експериментальних даних з системи автоматизації для їхньої обробки в *MS Excel*.

Текстові файли з розділенням даних комою, які мають розширення .*csv*, є «рідними» для *MS Excel* та зчитуються як і файли робочих книг .*xlsx*.

Керування імпортом даних в проводиться з групи **Дані-Отримання зовнішніх даних (Data – Get External Data)** стрічки (рис.2.45). *MS Excel* має вбудовані засоби імпорту даних з баз даних – MS Access, SQL Server, *MS Query*, імпорту даних стандарту *xml*, імпорту даних з текстових файлів.



Рис. 2.45. Імпорт даних

Імпорт текстових файлів проводиться в діалоговому режимі з «помічником імпорту» Text Import Wizard полем Дані-Отримання зовнішніх даних- З тексту (Data-Get External Data-From Text) стрічки.

Перший екран «помічника» містить кнопки вибору типу розділювача з розділювачем або фіксована ширина (Delimited, Fixed width), поле визначення номеру рядка, з якого буде почато зчитування Почати імпорт із рядка (Start import at row) та список форматів кодування Формат файлу (File origin). В центрі вікна відображається інформація з файлу у вигляді, який розпізнаний *MS Excel* (рис. 2.46 а).

Майстер імпорту тексту - крок 1 із 3	? >>	Майстер імпорту тек	сту - крак 2 із 3	? X
Майстер текстів розпізнав дані як список значень із розділювачам	1.	Lie giasorone sizuo and	змоги установити роздільники для текстових данну. Разил	ьтат позділення можна побацити у вікні внизи
Якщо це правильно, натисніть кнопку "Далі", якщо ні - укажіть біль	ш придатний тип даних.	Dession units	аногу установити роздоловики для тексторих даних гезул	атат роздовения вожна поовчити у акти анизу.
Формат вихідних даних		Роздивники		
Укажіть формат даних:		крапка з комою	Вважати послідовні роздільники одним	
<ul> <li>з роздільн<u>и</u>ками</li> <li>– для розділення полів використовуї табуляції.</li> </ul>	оться символи-роздільники, такі як коми та символи	Бома		
<ul> <li>фіксованої ширини - поля мають указану ширину.</li> </ul>		_ пробіл	Обмежувач рядків:	
-		інший:		
Поцати імпорт із рядка: 1 🗘 Формат файлу: 12	i1 : кирилиця (Windows)			
<u>М</u> ої дані із заголовками.				
Dependentiñ poperane deŭer GAfile tut		Зразок аналізу даних		
попереднии перегляд филлу стапеска.				
2 11, 33333333333331, 66666666666672	2222	1	1,333333333333333333333333333333333333	\$\$\$\$\$\$\$\$\$\$\$\$
1,66666666666666722,333333333333332,6666666666	6667	1,6666666666666666	7 2 2,333333333333333 2,666	566666666666
6				
	>	6	1 1 1	>
	маистер імпорту тексту - крок з із з		· ^	
	Це діалогове вікно дає змогу призначити формати дани	их для кожного стовпця.		
	Формат даних стовпця			
	<ul> <li>загальний</li> <li>"Загальний" фор</li> </ul>	мат перетворює числові значення на	исла, значення дати - на дати, а	
	О текстовий решту значень - н	на текст.		
	Одата: ДМР	<u>До</u> датково		
	О пропустити стовпець			
	Зразок аналізу даних		1	
	Вагальний Вагальний Вага 0,666666666666666666666666666666666666	альний Вагальний 3333333333333 1,66666666666666	67	
	1 <b>1,33333333333333</b> 1,66	666666666666667 2 2,3333333333333333	33	
	1,6666666666667 2,33	3333333333333 2,6666666666666	667	
			~	
B	<		>	

Рис. 2.46. Вікна імпорту: а – кодування; б – розділювач; в – формат

Другий екран «помічника» містить перемикачі вибору типу розділювача знак табуляції (Tab), крапка з комою (Semicoln), кома (Comma), пробіл (Space), інший (Other) та список символів розділювача рядків Обмежувач рядків (Text qualifier). В центрі вікна відображається інформація з файлу у вигляді, який розпізнаний *MS Excel* (рис. 2.46 б).

На третьому екрані «помічника» перемикачами **загальний** (General), текстовий (Text), дата (Date) визначається формат даних для комірок обраного мишею стовпця таблиці. В центрі вікна відображається інформація з файлу у вигляді, який розпізнаний *MS Excel* (рис. 2.46 в).

# Сортування даних

Таблиці можна *відсортувати* (Sorting) або виділити для показу тільки ті комірки, які вдовольняють якомусь критерію – *відфільтрувати* таблицю (AutoFilter). Сортування рядків чи стовпців таблиці полягає в зміні порядку їхнього взаємного розташування.

Дії сортування та фільтрація можуть використовуватися тільки для наперед виділеного діапазону комірок.

Сортування може здійснюватися в *порядку зменшення* (Sort Largest to Smallest) або в порядку збільшення значень (Sort Smallest to Largest). Для текстів порядок визначається алфавітом мови.

Проводиться сортування в порядку збільшення в наступному порядку: числа, текст, логічні значення, значення помилок, порожні комірки.

Управління режимами проводиться кнопками групи Данні-Сортування й фільтр (Data-Sort&Filter) стрічки (рис. 2.47).



Рис. 2.47. Сортування та фільтрація

Запускається сортування кнопкою Сортувати (Sort) стрічки. Кнопка активує діалогове вікно (рис. 2.48), в якому в полі Стовпець (Column) визначається стовпець, по даних якого буде проводитися сортування, в полі Сортування за (Sort On) визначається ознака даних: Значення (Values), Колір комірки (Cell Color), Колір шрифту (Font Color),- за якою буде проводитися сортування, в полі Порядок (Order) визначається порядок сортування.

Кнопки Додати рівень (Add level), Видалити рівень (Delete Level) дозволяють додати, прибрати додаткові умови сортування.

Сортування									?	×
<b>†</b> А↓ <u>Д</u> одати рі	вень	<u>Х</u> видал	ити рівен	нь	<u>Копіювати рівень</u>	-	<u>П</u> араметри		Д <u>а</u> ні з заго.	ловками
Стовпець				Сор	тування за		Порядок			
Сортувати за	Стовп	ець В		Зна	чення		Від найменшо	ого зн	ачення до н	айб 🖂

Рис. 2.48. Вікно сортування

Фільтрація запускається кнопкою **Фільтр** (**Filter**) стрічки. Ознакою готовності до фільтрації є поява знаків трикутника на комірках першого рядка виділеного діапазону. Натискання значка виводить на екран контекстне меню (рис. 2.49 а), в якому можна призначити проведення сортування, обрати з наведеного списку значення або значень для фільтрації, задати в полі

**Фільтри чисел (Number Filters**) логічну умову (=, <, > тощо ) порівняння значень комірок з якимось порогом.

Більш складні критерії фільтрації можна задати опцією **Користувацький фільр... (Custom Filters...**). Опція активує діалогове вікно (рис. 2.49 б), в якому можна визначити критерій із застосуванням логічних дій «AND» та «OR» з кількома значеннями комірок стовпця.

Пакет автоматично розпізнає випадки, коли перший рядок таблиці є рядком заголовків стовпців.

Треба мати на увазі, що комірки, які не виводяться на екран за рахунок фільтрації, беруть участь в розрахунках, для яких аргументи задаються діапазонами, які вміщують сховані комірки.

1	*			
А́↓	Сортування від найменшого до найбільшого	)	Лорівнює	
¢A	Сортування від найбільшого до найменшого	)	Не дорівнює…	
	Сор <u>т</u> ування за кольором	+	Більша	Користувацький автофільтр
T <sub>×</sub>	Видалити фільтр із "1"		Більше або дорівнює…	Показати лише ті рядки, значення яких:
	Фільтрування за кольором	Þ	Менше	1
	Фідьтри чисел	•	<u>М</u> енше або дорівнює…	не дорівнює 🗸 4
	Пошук	2	<u>М</u> іж	• 1 О АБО
	<ul> <li>✓ (Виділити все)</li> <li>✓ 2</li> </ul>	^	<u>П</u> ерші 10	менше ✓ 7
	🗹 3		<u>Б</u> ільше середнього	
	-⊻4 -▼5	0	<u>М</u> енше середнього	Знак питання "?" позначає один будь-який символ
		•	<u>К</u> ористувацький фільтр	Символ "*" позначає послідовність будь-яких символів
	a			б
	a			U

Рис. 2.49. Фільтрація: а – вибір критерію; б – фільтр користувача

## Макроси

Одні з перших програм для роботи з електронними таблицями для багатокрокових обчислень використовували списки функцій та вводили тимчасові заховані стовпці на невидимому краю аркуша. Такі програми вважалися «великими функціями» та отримали назву *макроси*.

Сьогодні в табличних процесорах *макросом* називається записана послідовність команд або дій користувача на клавіатурі. Використовуються макроси для автоматизації проведення однотипних послідовностей операцій, які часто виконуються, та створення функцій користувача з можливостями, які відсутні у вбудованих функціях *MS Excel*.

Основне призначення макросів – автоматизація обробки документа, яка зазвичай виконується послідовністю дій користувачем засобами інтерфейсу *MS Excel*. Наприклад, форматування нової таблиці, готування таблиці до друку

хованням комірок, які не потрібно друкувати, очищення полів бланку тощо. Такі макроси називають *макросами-послідовностями* (Basic Macro).

Іншим різновидом макросів є *макроси-функції*. Наприклад, функції користувача, які розширюють бібліотеку вбудованих функцій *MS Excel*. У вигляді *макросів-функцій* можна створювати додатки до *MS Excel*, які мають діалоговий віконний інтерфейс та виглядають як повноцінні *Windows* застосунки.

Макроси-послідовності можуть бути записані (Recording) або розроблені у вбудованому середовищі мови програмування Visual Basic Application (VBA).

Для запису *макросів-послідовностей MS Excel* має вбудований макрорекордер, який сприймає послідовність дій користувача, переводить їх в програмний код та зберігає. В макросах-послідовностях можливо записати тільки лінійний алгоритм, тобто записати тільки послідовність визначених дій. Циклічні операції, розгалуження алгоритмів в макросах-послідовностях реалізувати неможливо.

Зберігаються обидва типи макросів у вигляді програм мовою Visual Basic for Application, VBA (рис. 2.50). Макроси-послідовності записуються у вигляді підпрограм без вхідних аргументів (Sub), макроси-функції записуються у вигляді підпрограм-функцій з вхідними аргументам (Function)



Рис. 2.50. Макрос в середовищі VBA

Основне керування макросами проводиться з групи Розробник-Код (Developer-Code) стрічки (рис. 2.51 а)



Рис. 2.51. Макроси: а – керування та запис; б – визначення параметрів; в – закінчення запису

Для запису макросу-послідовності слід натиснути кнопку початку запису Розробник-Код-Записати макрос (Developer-Code-Record Macro) (рис. 2.51 а). У виведеному на екран діалоговому вікні Запис макросу (Record Macro) (рис. 2.51 б) потрібно визначити ім'я макросу в полі Ім'я макросу (Macro name), місце зберігання в полі Зберегти в (Store macro in) та, за потреби, додати коментар в полі Опис (Description), призначити «гарячі» клавіші запускання макросу в полі Сполучення клавіш (Shortcut key).

Записані макроси можуть бути збережені в поточній книзі або в книзі макросів користувача на комп'ютері, на якому встановлений *MS Excel*.

Макроси, для яких вказане місце зберігання Ця книга (This Workbook), є доступними тільки коли робоча книга, в якій вони створювалися, є відкритою в *MS Excel*.

Макроси, для яких вказане місце зберігання Особиста книга макросів (Personal Macro Workbook), є доступними в будь-яких робочих книгах на тому комп'ютері, на якому вони створювалися. Зберігаються такі макроси в окремому файлі, зазвичай, *Personal.xlsb* в папці «..\*AppData\Roaming\Microsoft\Excel\XLSTART\»*. Саме цей файл називається *особистою книгою макросів*. Він завантажується автоматично під час відкриття застосунку *MS Excel*.

Для завершення процесу запису макросу слід натиснути кнопку Розробник-Код-Зупинити запис (Developer-Code-Stop Record Macro) на місці кнопки початку запису (рис. 2.51 в).

За замовчання макрос-послідовність відтворює дії користувача буквально, тобто з адресами комірок, які було вказано при запису. Для того, щоб починати дії макроса-послідовності з поточної комірки на момент його

запуску, слід перед записом макросу активувати поле Відносні посилання (Use Relative References) стрічки (рис. 2.51 в)

Запустити (Run a Macro) *макрос-послідовність* можна «гарячими» клавішами, які призначені макросу під час створення (рис. 2.51 б) або кнопкою **Виконати (Run**) у вікні менеджера макросів (рис. 2.52).

З вікна менеджера макросів можна також:

- змінити в полі Макроси з (Macros in) місце пошуку макросів: Усі відкриті книги, Ця книга, Персональна книга макросів;
- видалити макрос кнопкою Видалити (Delete);
- перейти в режим редагування макросу в середовищі VBA кнопкою Змінити (Edit).

Макрос1	1	<u>В</u> иконати
Иакрос1	<u>^</u>	
		<u>Крок із заходом</u>
		<u>З</u> мінити
		В <u>и</u> далити
	~	Параметри

Рис. 2.52. Вікно макросів

Менеджер макросів активується кнопкою Макроси (Macros), яка знаходиться в групі Розробник-Код (Developer-Code) та Основне (Home) стрічки.

Запускати макрос можна також кнопками з панелі швидкого запуску, стрічки або безпосередньо на аркуші робочої книги.

Кнопка на аркуші книги є вбудованим елементом управління. Вставляється вона в аркуш полем (рис. 2. 53 а) Розробник-Вставити-Елементи керування форми-Кнопка (Developer-Controls-Insert-Form Controls-Button) стрічки.

Відразу після вставляння кнопки в аркуш на екран виводиться вікно макросів, в якому можна визначити макрос, який буде прив'язаний до кнопки. Налаштувати властивості кнопки, прив'язку макросу в тому числі, можна відповідними пунктами контекстного меню кнопки (рис. 2.53 б).

Прив'язка макросу до кнопки панелі швидкого доступу або стрічки проводиться у вікні налаштувань, яке відкривається полем Файл-Параметри-Панель швидкого доступу/Стрічка (File-Options-Customize Ribbon/Quick Access Toolbar) стрічки або відповідними пунктами контекстного меню панелі швидкого доступу/стрічки.

Вставити Режим Конструктора Відобразити	- 🐰 Вирізати 🖶 Копідвати 🖺 Вставити	Вибрати <u>к</u> оманди: <sup>①</sup> У	Настроїти <u>п</u> анель швидкого доступу: <sup>©</sup> Для всіх документів (за промовчанням)
Консрукций — Консрукций – Консрукций форми Мата – Солона (слемент керування форми) Кнопка (слемент керування форми) — Конска (слемент керування форми) — Конска (слемент керування форми)	<ul> <li>Зыйнити текст</li> <li>[рупування</li> <li>Порядок</li> <li>Призначити макрос</li> <li>Формат елемента керування</li> </ul>	«Роздільник» 🖧 Макрос1	п 3берегти
	а	б	В

Рис. 2.53. Прив'язка макросу: а – контекстне меню кнопки; б – вікно налаштувань

Для введення кнопки слід обрати в полі Вибрати команди: (Choose command from) значення Макроси (Macros) та кнопкою Додати перенести виділений макрос, який при цьому виводиться в лівій частині вікна під полем, на праву частину вікна в визначене місце стрічки/панелі швидкого доступу (рис. 2.53 в).

Макроси-функції створюються в середовищі VBA.

Середовище активується кнопкою Розробник-Код-Visual Basic (Developer-Code- Visual Basic) стрічки (рис.2.51 а).

У вікні середовища відображається вгорі головне меню, зліва вгорі дерево активних проєктів. Під ним знаходиться вікно властивостей об'єктів. Праву частину оболонки займає редактор коду програм (рис. 2.50).

Найпростіший спосіб створення *макросу-функції* — це використання вбудованих засобів середовища.

В проєктах компоненти програм містяться в модулях. Додавання модулів проводиться з головного меню або контекстного меню проєкту пунктом Insert-Module (рис. 2.54 а).

Подвійний клік на модулі в дереві проєктів активує редактор коду відповідного модуля.

Створення нової підпрограми в обраний модуль можна провести пунктом **Insert-Procedure** з редактора коду в діалоговому вікні (рис. 2.54).

У вікні в полі **Name** вводиться ім'я компонента, зі списку обирається тип компонента: підпрограма/макрос-послідовність **Sub**, підпрограма-функція **Function**, властивість об'єкту **Property**. Список **Scope** визначає доступність компонента. Компоненти, які повинні бути видимі ззовні середовища, наприклад, в *MS Excel*, мають мати тип **Public**, службові компоненти проєкту мають тип **Private**.

着 Microsoft Visual Bas	sic for Applications - Книга	1.	
😽 <u>F</u> ile <u>E</u> dit <u>V</u> iew	Insert Format Debug	R Add Procedure	>
i 🔣 🚾 🗕 🛃 🛛 🖉	Procedure		, i
Project - VBAProject	🖳 UserForm	er Name:	OK
🔳 💷 🛅	💐 <u>M</u> odule	Туре	
🗄 😹 VBAProject (PEF	🖄 🖸 Class Module	• Sub	Cancel
🖯 😻 VBAProject (Кни	Fi <u>l</u> e	C Function	
— 🖶 Містоsoft Excel Лист1 (Лист — 🕷 ЭтаКнига	T1)	C Property	
Hereit Modules		Scope	
Properties - Лист1	×	C Private	
Лист1 Worksheet	•		
Alphabetic Categorized (Name) Ли DisplayPageBreaks Fal	cr1 ^	All Local variables as Statics	
a		· ·	б

Рис. 2.54. Створення елементів: а – модуля; б – функції У вікні редактора повинен з'явитися шаблон підпрограми:

```
Public Function Name()
```

## End Function

В круглих дужках через кому вводяться імена змінних, які слугуватимуть вхідними параметрами функції. Між службовим словами початку опису **Public Function** та кінця опису **End Function** функції вставляються необхідні рядки програмного коду.

За правилами програмування змінним можна призначити тип. Найбільш поширеним типом даних є дійсний тип. Тип призначається додаванням опису типу після першої згадки змінної. Наприклад,

```
Public Function Name(x As Double) As Double
Dim y As Double
y=Sqr(x)
Name=x+y
End Function
```

За замовчанням програми зберігаються в тому файлі *MS Excel*, з якого відкривалося середовище.

Розроблений макрос-функція стає доступним разом із вбудованими функціями в менеджері функцій в категорії Визначені користувачем (User Defined) (рис. 2.55 а).



Рис. 2.55. Макроси-функції: а – вигляд в менеджері функцій; б – властивості

Для *макросу-функції* за замовчанням в менеджері функцій відсутній опис. Зробити опис засобами програмування неможливо. Ввести опис можна з урахуванням того, що розроблена функція є макросом, у вікні макросів. У вікні макросів макроси-функції не відображаються – ім'я слід вводити вручну в поле **Ім'я Макросу (Мастоз Name**). Після цього стає доступним кнопка властивостей **Параметри** (Options). У відкритому кнопкою вікні (рис. 2.55 б) в полі Опис (Description) можна ввести текст, які буде відображатися як довідка в менеджері функцій.

Для того, щоб забезпечити перенесення функцій користувача та їхнє використання на інших комп'ютерах можна скористатися *надбудовами* (Addins) MS Excel. Надбудови є програмами додатками, які завантажуються автоматично та виконуються в середовищі MS Excel. Зберігаються надбудови в файлах з розширенням .xlam.

Під'єднання надбудови проводиться кнопкою **Файл-Параметри-**Надбудови-Керування-Надбудови Excel-Перейти (File-Option-Add-ins-Manage-Excel Add-ins...) стрічки. У відкритому кнопкою вікні (рис. 2.56 а) слід активувати потрібну надбудову. Список надбудов за замовчанням зчитується з папки «..\Users\xx\AppData\Roaming\Microsoft\AddIns». В разі відсутності надбудови в списку слід додати файл надбудови пошуком кнопкою Огляд... (Browse...).

Під списком надбудов виводиться опис виділеної надбудови. Інформація для опису береться з поля нотаток властивостей файлу надбудови

з поля Документ-Примітки (Summary-Comments) вікна додаткових властивостей (рис. 2.56 б). Вікно відкривається полем меню Файл-Параметри-Властивості-Додаткові властивості(File-Options-Info-Properties-Advanced Properties).



Рис. 2.56. Надбудови: а – вікно надбудов; б – вікно визначення опису

# Контрольні запитання

- 1. Які стилі адресації використовує Excel?
- 2. Як перемикаються стилі адресації?
- 3. Що означає, адреса A1, \$A\$1, A\$1 \$A1?
- 4. Що є ознакою наявності формули в комірці?
- 5. Що таке позначка заповнення?
- 6. Як ввімкнути доступність записів макросів?
- 7. Як записати макрос?
- 8. Як записати макрос відносно поточної активної комірки?
- 9. Як запустити макрос?
- 10.Як ввімкнути доступність макросів?
- 11. Як прив'язати макрос до кнопки?
- 12. Як прив'язати макрос до панелі швидкого запуску?
- 13.Яким типом визначається макрос в VBA?
- 14. Як зробити так, щоб макроси завантажувалися автоматично?
- 15.Як запустити функцію користувача?
- 16. Яким типом визначається функція в VBA?
- 17. Як зробити, щоб функції завантажувалися автоматично?
- 18.В якому форматі зберігаються надбудови add in?
- 19.Як активізувати надбудову add in?
# 3. СИСТЕМИ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ

Комп'ютерна математика – це напрямок в математиці на перетині класичної математики та інформатики. Виникнення комп'ютерної математики наприкінці двадцятого століття пов'язане з успіхами впровадження персональних комп'ютерів (ПК) в практику вирішення математичних задач.

За визначенням проф. В. П. Дьяконова «комп'ютерна математика є сукупністю 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».

Головним засобом комп'ютерної математики є *системи комп'ютерної* математики (СКМ). СКМ забезпечують можливість використання користувачем математичних методів без процедури програмування в зручному середовищі для роботи.

СКМ є програмними засобами, які автоматизують виконання як чисельних, так і аналітичних (символьних) обчислень і розрахунків. В англомовному середовищі СКМ називають *Computer Mathematics Systems* (*CMS*).

СКМ є поліфункціональними, універсальними програмними засобами, які призначенні 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.

Використання СКМ зменшує потребу в різноманітних довідниках і математичних таблицях, дозволяє 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 СКМ професійного призначення представлені в основному виробами таких великих західних фірм як *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.

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.

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, украй 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 [25].

## 3.1. Класифікація СКМ

На сьогодні не існує сталої класифікації СКМ. Різні автори пропонують свою класифікацію [24-27].

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

- По відношенню до форми отриманих результатів та застосованих математичних методів на системи для чисельних розрахунків, системи для аналітичних розрахунків системи комп'ютерної алгебри (computer algebra system, CAS) та комбіновані системи.
- По відношенню до **оплати за користування** на *безкоштовні* системи, *умовно-коштовні* системи, *комерційні* системи. Більшість СКМ є досить дороговартностними комерційними проєктами.
- По відношенню до **типу задач**, які розв'язує СКМ на *математичні* та *технічні*.
- По відношенню до кількості виконуваних математичних дій на спеціалізовані та універсальні. Серед спеціалізованих розрізняють системи статистичних розрахунків, математичні системи для розв'язання нелінійних рівнянь, розв'язання систем диференційних рівнянь, обчислення поліномів, аналітичної геометрії, задач теорії груп, програмні засоби візуалізації математичних даних тощо.
- По відношенню до **організації даних** на *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 математичних програмних систем:

- калькулятори застосунки, що реалізують прості арифметичні дії, як додавання, множення, піднесення до ступеню, тригонометричні обчислення;
- CMS системи чисельної комп'ютерної математики;
- CAS системи символьної комп'ютерної математики;
- засоби статистичного аналізу;
- засоби оптимізації;
- засоби геометрії;
- інтернет-засоби;
- математичні бібліотеки.

### 3.2. Типова структура СКМ

Сучасні СКМ представлені розробками різних виробників, тому природнім є відмінності в архітектурі кожної СКМ. Утім, всі сучасні універсальні СКМ мають типову структуру (рис. 3.1).



Рис. 3.1. Типова структура універсальних СКМ [26]

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 запитів та директив користувача, виклик потрібних процедур.

Ядро (kernel) системи містить коди функцій, які повинні виконуватися максимально швидко. Кількість вбудованих в ядро функції обмежена. Більш складні та не так часто виконувані функції реалізуються засобами зовні ядра бібліотеками. Загальна кількість функцій СКМ може досягати кількох тисяч (рис. 3.2). Наприклад, одна з найбільш потужних СКМ Mathematica містить інформацію про більше, ніж 5000 інтегралів, а використовує тільки кілька десятків функцій ядра.

Адаптація СКМ 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 користувачеві знань з чисельних методів, аналітичної математики та комп'ютерної графіки [27].



Рис. 3.2. Обсяг вбудованих функцій СКМ

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

### 3.3. Історія розвитку СКМ-СКА

Практичного вжитку математичне програмне забезпечення набуло в середині XX століття. На той час 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.

Розв'язання математичних та відповідно технічних задач проводилося лише на великих комп'ютерах засобами систем програмування колективного використання, застосування яких навіть для простих задач потребувало участі багатьох висококваліфікованих фахівців, обізнаних в математиці та програмуванні.

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:  $\pi$ , 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 на зразок дрібно-раціональних перетворень, підстановок, спрощення, розв'язання рівнянь, диференціювання тощо традиційними для математики способами, а результат отримувати не тільки у вигляді чисел, а й у вигляді формульних виразів. Такі перетворення називають перетвореннями в символьному вигляді або аналітичними перетвореннями.

Це призвело до буму досліджень в області комп'ютерної математики. Прообрази сьогоднішніх СКМ та СКА були розроблені саме в провідних університетах світу.

Ресурс Вікіпедія вважає першими успішними проєктами СКМ-СКА програму *Schoonschip* для символьних обчислень в галузі фізики, яка була розроблена Мартіусом Велтманом в 1963 році в ЦЕРНі, та програму

*MATHLAB*, яка була розроблена в 1964 році Карлом Енгельманом в рамках проєкту MITRE.

До першого покоління СКА відносять:

- СКА МАСЅҮМА, яка розроблена Джоелом Мозесом в Масачусетському технологічному інституті на замовлення Департаменту енергетики США. Сьогодні не виробляється. Потомком насьогодні є безкоштовна СКА МАХІМА.
- *СКА SCRATCHPAD*, яка розроблена Ричардом Димиком з фірми *Thomas J. Watson Research Center* за замовленням компанії *IBM*. Сьогодні не виробляється. Потомком насьогодні є безкоштовна система *Axiom*.
- *СКА REDUCE* (http://www.reduce-algebra.com/), яка розроблена Тоні Хірном. Є безкоштовною системою.

Працювали СКА першого покоління на великих комп'ютерах. Доступні такі системи були тільки вузькому колу користувачів.

СКМ як клас спеціалізованих програмних засобів індивідуального використання набули популярності в останній чветрі XX століття завдяки розвитку індустрії персональних комп'ютерів. Вони утворили друге покоління СКМ-СКА. Ознакою систем другого покоління став графічний інтерфейс, орієнтований на користувача та можливість реалізації на малих ЕОМ та ПК.

До систем другого покоління відносяться:

- *Maple*. Є універсальною СКМ-СКА. яка розроблена Кейтом Геддесом та Гастоном Гоннетом в Університеті Ватерлоо, Канада (University of Waterloo). Сьогодні виробляється фірмою Waterloo Maple Inc.
- *Mathematica*. Є універсальною СКМ-СКА. Прообраз системи SMP (Symbolic Manipulation Program) був розроблений Стівеном Вольфрамом в Каліфорнійському технологічному інституті, США (California Institute of Technology). Сьогодні виробляється фірмою Wolfram Research.
- *SageMath* потомок СКА *Sage* (System for Algebra and Geometry Experimentation).
- DERIVE СКА потомок системи *muMATH* фірми Soft Warehouse, підрозділу Texas Instruments. Сьогодні не виробляється.

- МАТLAВ (МАТгіх LABoratory). В перших версіях СКМ для чисельних розрахунків, сьогодні – універсальна СКМ-СКА., яка розроблена Клівом Моулером з університету Нью-Мехіко (University of New Mexico). Сьогодні виробляється фірмою MathWorks.
- МАТНСАД. В перших версіях СКМ для чисельних розрахунків, на поточний час – універсальна СКМ-СКА, яка розроблена Алленом Раздовом з Масачусетського технологічного інституту (Massachusetts Institute of Technology). Сьогодні виробляється фірмою РТС.

Третє покоління математичних систем характеризується графічноорієнтованим інтерфейсом та використанням сучасних мов програмування. Такі системи почали з'являтися на початку XXI століття.

До систем третього покоління відносять:

- *МUPAD*. Є універсальною СКА. яка розроблена Бенно Фуксштейнером
  з університету Падерборн, Німеччина (Universität Paderborn).
  Використовувалася самостійно та в складі СКМ SciLAB, MathCAD.
  Сьогодні не виробляється, є частиною СКМ MATLAB.
- Scilab. Є безкоштовною СКА. Випускається Національним інститутом досліджень з інформатики та автоматики, Франція (INRIA, Institut national de recherche en informatique et en automatique), Національною школою мостів та доріг, Франція (ENPC, École nationale des ponts et chaussées) та фіромою Scilab Enterprises.

Сьогодні СКМ активно використовуються в навчальному процесі у всьому світі. Тільки в системи Mathematica офіційними користувачами є більше тисячі університетів з 61 країн. 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.

На вітчизняному та світовому ринку сьогодні СКМ-СКА представлені лише закордонними виробами. Це пов'язано з необхідністю великих інтелектуальних, трудових та фінансових витрат.

Хоча слід відмітити, що вже на початку 60-х років XX століття у Києві під керівництвом академіка В. М. Глушкова були створені передвісники майбутніх персональних комп'ютерів – інженерні малі комп'ютери серії «Мир» з апаратною реалізацією мов програмування високого рівня та унікальними аналітичних можливостями виконання чисельних та розрахунків. Для цих ЕОМ були розроблені мови високого рівня «МИР» і «АНАЛИТИК», які давали змогу обробки дійсних чисел довільної розрядності, цілих чисел необмеженої розрядності, проведення точних операцій дробовими раціональними над числами тощо. Система «АНАЛИТИК» була однією з перших систем комп'ютерної алгебри та вперше використовувала техніку запису алгебраїчних виразів (застосування співвідношень), яка в наш час є основою технології декларативного програмування [1].

## 3.4. Функції СКМ

До можливостей чисельних розрахунків СКМ 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 степеневих поліномів;
- знаходження комплексних коренів поліномів;
- розв'язання систем нелінійних алгебраїчних рівнянь;
- розв'язання систем диференційних рівнянь;
- оптимізація функцій;
- лінійне програмування;
- одномірна та багатомірна інтерполяція;
- задачі теорії імовірності;
- задачі математичної статистики;
- операції над текстом;
- обчислення трикутників;

- визначення площин в тривимірному просторі;
- визначення кривих другого порядку
- задачі аналітичної геометрії.

Можливості різних СКА в частині *аналітичних перетворень* суттєво відрізняються. Серед типових дій можна виділити:

- поєднання символьних 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 елементурних та спеціальних математичних функцій:
- розв'язок рівнянь та систем рівнянь;
- розв'язання диференційних рівнянь та систем диференційних рівнянь;
- розв'язок задач лінійного та нелінійного програмування;
- операції з векторами та матрицями.

Графічні можливості СКМ дозволяють:

- будувати графіки функцій в декартовій, полярній системі координат;

- будувати графіки функцій, які визначені в параметричній формі;
- будувати графіки тривимірних поверхонь;
- будувати графіки контурні, векторні, діаграми;
- сторювати анімовані графіки;
- редагувати графіки.

Особливостями програмування вбудованою мовою СКМ є:

- 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.

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 [27]:

- 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;
- впровадження нових функцій, наприклад, для реалізації нечіткої логіки, нейронних мереж тощо;

- впровадження в СКМ засобів для створення електронних підручників у різних форматах;
- створення документів з текстами, формульними виразами, рисунками і графіками найвищої поліграфічної якості.

# 3.5. Сучасні СКМ

На сучасному ринку програмного забезпечення наявні всі різновиди математичних систем: безкоштовні та комерційні, універсальні та спеціалізовані, математичні та технічні, СКМ та СКА тощо.

В таблиці 3.1 наведений стислий огляд сучасних математичних систем. На рисунках 3.3 – 3.5 зображені приклади екранів оглянутих систем.

Назва	Опис	Вартість	Сайт
СКА ко	мерційні спеціалізовані		
Fermat	Розроблена Робертом Л'юісом (Robert H. Lewis). Розв'язує задачі лінійної алгебри.	\$70	http://home.bway.net/le wis/
Magma	Розроблена в Університеті Сіднею, Австралія (University of Sydney). Розв'язує задачі теорії груп	\$2275	http://magma.maths. usyd.edu.au/magma/
СКА комерці	ині універсальні		
	Випускається фірмою Wolfram Research of Champaign, Illinois у вигляді кількох систем: Mathematica, Wolfram One, Wolfram Alpha. Має хмарну версію. Вважається найбільш потужною СКМ-СКА.	\$2500/\$140 професійна/ студентська	https://www.wolfram.co m
Maple	Випускається фірмою Waterloo Maplesoft. Має хмарну версію MapleCloud.	\$2275/\$99 комерційна/ студентська	https://www.maplesoft.c om

Таблиця 3.1. Сучасні математичні системи

# Таблиця 3.1. Продовження

Назва	Опис	Вартість	Сайт
Matlab	СКА МАТгіх LABoratory виробляється фірмою MathWorks Inc. Стала універсальною СКМ-СКА завдяки придбанню СКА MuPad та включення її в склад Matlab. Має хмарну версію Matlab Online. Одна з найбільш розповсюджених в інженерних колах СКМ завдяки спеціалізованим пакетам	\$2350/\$20 стандартна/ студентська	https://www.mathworks. com/
Mathcad ptc mathcad	розширення. Реалізує інтерфейс WYSIWYG. До універсальних СКА відноситься завдяки ліцензіям на використання символьного процесора СКА MuPad. Має скорочені можливості символьних перетворень. Має безкоштовну версію Express Edition	\$650/\$105 комерційна/ студентська	https://www.ptc.com/
LiveMath	С найпростішою з комерційних СКА. Розроблена Карлом Спіцнагелем (Carl Spitznagel) в Університеті Джона Карролла, США (John Carroll University). Випускалася під назвами Theorist, MathView, MathPlus. Має засіб розробки інтернет застосунків LiveMath Maker	\$400	https://www.livemath. com
СКА бе	зкоштовні спеціалізовані		
Cadabra	Розроблена Каспером Питерсом (Kasper Peeters) в 2007 році. Розв'язує задачі тензорної теорії поля		https://cadabra.science
FORM	Розроблена в 1989 році Джосом Вермасереном (J.A.M. Vermaseren) в Датському інституті ядерної фізики (Dutch institute for subatomic physics). Розв'язує задачі квантової фізики		https://www.nikhef.nl/ ~form/

# Таблиця 3.1. Продовження

Назва	Опис	Вартість	Сайт
GAP	Groups, Algorithms and Programming. Розроблена в Рейнсько-Вестфальскому технічному університеті м. Ахен, Німеччина (Lehrstuhl D für Mathematik, RWTH Aachen, Germany) в 1986 році. Розповсюджується GAP Group. Розв'язує задачі теорії груп та комбінаторики		https://www.gap- system.org/
Macaulay2	Розроблена Даніелем Грайсоном (Daniel Grayson) в університеті Ілінойсу, США в 1994 році. Розв'язує задачі лінійної алгебри		https://faculty.math. illinois.edu/Macaulay2/
СКА бе	зкоштовні універсальні		
Axiom AXIOM	Розроблена Річардом Дженксом (Richard Jenks) на заміну СКМ Scratchpad		http://www.axiom- developer.org
FriCAS	С самостійним насьогодні відгалуженням СКА <b>Ахіот.</b> Розроблена Валдеком Хебішем (Waldek Hebisch) в 2007 році.		http://fricas.sourceforge. net/
Maxima Š	Розроблена Вільямом Шелтером (William Schelter) в Масачусетському технологічному інституті, США		https://maxima. sourceforge.io/
SageMath	(Software for Algebra and Geometry Experimentation). Розроблена в 2005 році Вільямом Штейном (William Stein) в Університеті Вашингтону, США (University of Washington) на базі бібліотек NumPy, SciPy, Sympy. Є хмарною СКА		https://www.sagemath. org/
Mathics	Розроблена в 2011 році Жаном Пошко (Jan Pöschko) подібно до СКА Mathematica на базі бібліотек SymPy, NumPy, SciPy. Є Інтернет орієнтованою СКА.		https://mathics.org

# Таблиця 3.1. Продовження

Назва	Опис	Вартість	Сайт
СКМ комерц	ійні спеціалізовані		
TK Solver	Розроблена в 1982 році Мілошем Кнопачеком (Milos Konopasek) з фірми Software Arts. Розв'язує системи нелінійних рівнянь. Виробляється фірмою Universal Technical Systems, Inc	\$599	https://www.uts.com
СКМ комерц	ійні універсальні		
	Розроблена фірмою Civilized Software, Inc.	\$2250/ \$50 стандартна/ студентська	http://www.civilized. com/mlabdesc.html
СКМ бе	езкоштовні універсальні		
Euler Math Toolbox	Розроблена Рене Гросманом (René Grothmann) в Університеті Інгольштату, Німеччина (Catholic University of Eichstätt- Ingolstadt, Germany) в 1988 році. Має можливість спільної роботи з безкоштовною СКА Махіта Розроблена Scilab Enterprises в		http://www.euler-math- toolbox.de/ https://www.scilab.org/
Scilab	1990 році як альтернатива комерційній СКМ Matlab. Поставляється з пакетом Xcos, який є аналогом Simulink в Matlab. Має хмарну версію		
RlabPlus	Розроблена на зразок СКМ MATLAB СКМ для ОС Linux		http://rlabplus. sourceforge.net
SMath Studio SMath	Розроблена в 2005 році Андрієм Івашовим в Санкт-Петербургському будівельному університеті як альтернатива СКА MathCAD. Має версії для мобільних пристроїв та україномовну локалізацію.		https://en.smath.com/ view/SMathStudio/ summary
GNU Octave	Середовище чисельного моделювання з мовою програмування високого рівня. Розроблено на зразок СКМ MATLAB		https://www.gnu.org/soft ware/octave

Таблиця 3.1. Закінчення

Назва	Опис	Вартість	Сайт	
СКМ безкош	СКМ безкоштовні спеціалізовані			
	AD Model Builder розроблена Девідом Фурн'є (D. Fournier). Розв'язує задачі диференційного обчислення.		http://www.admb- project.org/	
Dynamics Solver	Розроблена Хуаном Агурегабиром (Juan М. Aguirregabiria). Розв'язує системи диференційних рівнянь.		http://tp.lc.ehu.es/jma/ds/ ds.html	



Рис. 3.3. Приклади екранів СКА-СКМ: a – *Mathematica*; б – *Maple*; в – *Matlab*; г – *MathCAD*; д –*LiveMAth* 







а





В

Рис. 3.5. Приклади екранів СКА-СКМ: a – SciLab; б – SMath; в – Octave

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

Стислий огляд найбільш популярних систем цього класу наведений в таблиці 3.2. Приклади екранів оглянутих програм зображені на рис. 3.6–3.7

# Таблиця 3.2. Системи наукової графіки

Назва	Опис	Вартість	Сайт		
Комери	Комерційні програми візуалізації, аналізу та обробки даних				
DADiSP	Data Analysis and Display Розроблена в 1987 році фірмою DSP Development Corporation програма візуалізації та аналізу сигналів та зображень.	\$1995/\$129 комерційна/ академічна	https://www.dadisp.com/		
IGOR Pro	Розроблена фірмою WaveMetrics Inc програма візуалізації, обробки та аналізу сигналів та зображень	\$995/\$85 комерційна/ академічна	https://www.wavemetric s.com/		
MagicPlot	Розроблена фірмою Magicplot Systems, LLC програма візуалізації та регресійного аналізу сигналів	\$350/\$15 комерційна/ студентська	https://www.magicplot.c om/		
Advanced Grapher	Розроблена фірмою Alentium Software програма візуалізації, математичного та регресійного аналізу сигналів	\$1000/\$30 комерційна/ студентська	https://www.alentum.co m/agrapher/		
Безкош	товні програми візуалізації,	аналізу та о	бробки даних		
DataMelt	Розроблена на базі мов Python, Ruby, Java Сергієм Чекановим програма чисельних та символьних обчислень, аналізу та візуацізації даних		https://datamelt.org/		
Dataplot	Розроблена Американським інститутом стандартизації та технологій (National Institute of Standards and Technology).		https://www.itl.nist.gov/ div898/software/dataplot /index.htm		
SciDAVis	Scientific Data Analysis and Visualization Розроблена під керівництвом Расела Стандіша (Russell Standish) програма аналізу та візуалізації даних		http://scidavis.sourceforg e.net		
kst	Програма візуалізації та аналізу даних з можливістю програмування на мові Python		https://kst-plot.kde.org/		
Sysquake	Безкоштовна в діалоговій версії програма візуалізації даних. Професійна версія з можливістю програмування та бібліотеками є платною	\$1078	https://calerga.com		

Назва	Опис	Вартість	Сайт
Програ	ами побудови та аналізу граф	іків безкош	товні
MathPlot	Проста програма графічної візуалізації функцій		http://rigaux.org
GRAN	Програми      розробки        Національного      педагогічного        університету      України      [5]        графічної      візуалізації      та        елементурного      аналізу      функцій.        За      можливостями      є      аналогами        найпростіших      програм      типу        GraphAnder      (http://andron-        st.narod.ru),      FlatGraph        (http://old.exponenta.ru/soft/other      s/flatgraph/fg.asp).		http://www.zhaldak.npu. edu.ua/prohramnyi- zasib-gran
GeoGebra	Інтернет орієнтована проста програма візуалізації функцій		https://www.geogebra.or g/classic
OpenPlaG	Програма візуалізації функцій Юргена Куммера (Jürgen Китте). Має хмарну версію		https://rechneronline.de/ openPlaG/





Рис. 3.6. Приклади екранів програм візуалізації та обробки даних: а – DADiSP; б – IGOR Pro





Ж

Рис. 3.6. Приклади екранів програм візуалізації та обробки даних: a – MagicPlot; б – KST; в – Advanced Grapher; г – Sysquake; д – SciDAVis; e – OpenPlaG; ж – GeoGebra; з – MathPlot

3

Окремо слід виділити СКМ для статистичних розрахунків. Головна відмінність таких програм – велика кількість вбудованих спеціальних

функцій, які дозволяють без програмування виконувати багатоваріантний статистичний аналіз, наявність засобів для автоматичної фільтрації помилкових даних, спеціалізований (ad hoc) аналіз, перевірку гіпотез, прогнозуючу аналітику тощо.

Інформацію про деякі з таких програм наведено в табл. 3.3. На рисунку 3.7 зображені приклади екранів оглянутих систем.

Таблиця 3.3. Статистичні СКМ

Назва	Опис	Вартість	Сайт		
Універс	Універсальні комерційні статистичні програми				
SPSS Statistics	Statistical Package for the Social Sciences. Придбана фірмою IBM за 1,2 млрд. дол. США. Вважається провідною статистичною програмою в світі.	від \$99 в місяць	https://www.ibm.com/an alytics/spss-statistics- software		
Statgraphics	Розроблена в Прінстонському університеті, США (Prinston university). Має програмний інтерфейс з Python, R.	від \$990 в місяць	https://www.statgraphics .com/		
StatPlus		від \$135 на рік	https://www.analystsoft. com		
Універс	сальні безкоштовні статисти	чні програм	И		
PSPP	Розроблена Беном Пфаффом (Ben Pfaff) як безкоштовний аналог IBM SPSS Statistics		https://www.gnu.org/soft ware/pspp/		
ADaMSoft	Розроблена Марко Скарно (Marco Scarno) для UNESCO проста статистична програма		http://adamsoft.sourcefor ge.net/		
Спеціалізовані	комерційні статистичні програм	И			
MedCalc  easy-to-use statistical software	Розроблена фірмою MedCalc Software BVBA програма для медичних задач. Має специфічні методи аналізу медичних даних на зразок графіків Бленда- Альтмана (Bland-Altman), аналізу Пассінга-Баблока (Passing-Bablok), Демінга (Deming), Каплана-Мейера (Kaplan-Meier), Кокса (Cox) тощо	\$550	https://www.medcalc.org		
EViews	Розроблена фірмою HIS для економічних задач	\$1695	https://www.eviews.com		





Рис. 3.7. Приклади екранів статистичних систем: а – *SPSS*; б – *StatPlus*; в – *PSPP*; г – *MedCalc* 

#### Контрольні запитання

- 1. Що таке комп'ютерна математика?
- 2. Що таке СКМ?
- 3. За якими критеріями класифікують СКМ?
- 4. Що відомо про вітчизняні СКМ?
- 5. В чому різниця чисельних та символьних обчислень в СКМ?
- 6. Які СКМ вважаються найкращими?
- 7. З яких типових елементів складаються універсальні СКМ?
- 8. Які типові функції чисельних розрахунків СКМ?
- 9. Які типові функції аналітичних перетворень СКМ?
- 10.До якого типу мов програмування відносять мови СКМ?
- 11.В чому полягає особливість статистичних СКМ?

## 4. CKM MATHCAD

#### 4.1. Загальні характеристики

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



Розробник СКМ Аллен Раздов

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 оформлювати якісні тексти з наочним графічним наданням результатів.

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

### (-B1+KOPEHb(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 символьний процесор СКА *Maple*, що перевело *MathCAD* до класу універсальних систем комп'ютерної алгебри СКА. Останньою «авторською» версією став *MathCAD* 13.

В 2006 році саму систему та торгівельну марку *Mathcad* придбала фірма *PTC* (www.pts.com), яка спеціалізується у царині САD програм з основним продуктом *Creo*. Нові власники змінили колектив розробників та постачальника символьного ядра. Постачальником ядра стала фірма *SciFace Software* з СКА *MuPad*. Символьне ядро *Mathcad*, на відміну від оригінального *Maple* (*MuPAD*), має обмежений функціонал.

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

Новий продукт набув назву *MathCAD Prime*, при цьому сумісність з «класичним» *MathCAD* розробник не гарантує. Розробник не розвиває «класичну» версію, хоча продовжує реалізацію *MathCAD 15*.

*MathCAD* є інтегрованою системою, яка орієнтована на проведення математичних та інженерно-технічних розрахунків. До достоїнств пакета відносяться також можливості збереження документів в форматі Webсторінок (можливість відсутня в *MathCAD Prime*), *Microsoft Word*. *Adobe PDF*. 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, шістнадцатирічній 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 тощо);
- 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 символьні (аналітичні) перетворення;

- зберігати результати у форматах обміну з офісними пакетами;
- програмувати алгоритми обчислень.

Серцевину *MathCAD* складають власні обчислювальний та графічний процесори, ліцензійний символьний процесор (рис. 4.1).



Рис. 4.1. Структура СКМ MathCAD

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 деяких операцій в аналітичному вигляді. Графічний процесор слугує для побудови графіків функцій однієї та двох змінних.

*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 (можливість відсутня в *MathCAD Prime*).

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

Основними складовими *довідникової підсистеми MathCAD* (рис. 4.2) є довідник основних можливостей та функцій **Help**, посібники з використання **Tutorials**, заготовки розв'язання типових задач **QuickSheets** (відсутні в

MathCAD Prime), довідникові таблиці Reference Table (відсутні в *MathCAD Prime* в версіях старше за 5.0), електронні книги **E-books**.

Доступ до всіх елементів довідникової підсистеми в MathCAD забезпечується через пункт **Help** головного меню (рис. 4.2 а).

В MathCAD Prime доступ до довідника та посібників проводиться із закладки **Resources** стрічки (рис. 4.2 б). Посібники та довідникові таблиці доступні також з довідника **Help** (рис. 4.3 б).



а

Рис. 4.2. Структура довідникової системи: а – *MathCAD*, athCAD Prime

 $\delta - M$ 



Рис. 4.3. Довідник: а – *MathCAD*, б – *MathCAD Prime* 

QuickSheets – це MathCAD документи, які вміщують готові приклади розв'язання широкого кола математичних та науково-технічних задач (рис. 4.4).



Рис. 4.4. Заготовки типових задач MathCAD

**Reference Table** – розділ довідки (рис. 4.5), який вміщує довідники з даними про хімічні та фізичні властивості середовищ, математичні та технічні константи, математичні таблиці інтегралів та похідних, формули з механіки, електротехніки, радіотехніки тощо.

В MathCAD Prime Reference Table наявні тільки в версіях до 6.0.

Mathcad		Эртс.
Reference Table	S	
Basic Science	Properties of Liquids	Properties of Gases
Fundamental Constants	Density	Specific Gravity
Physics - Mechanics	Viscosity	Specific Heat
Periodic Table	Specific Gravity	Sound Velocity
Calculus	Sound Velocity	Molecular Weight
Derivative Formulas	Surface Tension	
Integral Formulas	Dielectric Constant	Properties of Metals
megrari omiaus	Index of Refraction	Thermal Conductivity
Geometry	Molecular Weight	Specific Gravity
Areas and Perimeters		Linear Expansion Coeffi
Volumes and Surface Areas	Properties of Solids	Electrical Resistivity
Polyhedra	Density	Poisson's Ratio
	Specific Gravity	Modulus of Elasticity
Mechanics	Specific Heat	Melting Point
Centroids	Thermal Conductivity	Temperature Coefficient
Mass Moments of Inertia	Dielectric Constant	of Resistivity
Electromagnetics		
Capacitance		
Oscillators		

Рис. 4.5. Довідникові таблиці MathCAD

**E-books** – 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 містять зміст, предметний покажчик та систему гіперпосилань. *Електронні книги* є версієзалежні. Для кожної версії системи застосовуються свої книги.

Виробник та спільнота користувачів пропонують багато безкоштовних та комерційних бібліотек, електронних книг з описами методів розв'язання

специфічних задач, діючими алгоритмами та готовими до використання документами системи.

## Електронні книги та бібліотеки для СКА MathCAD

Пакет аналізу даних (Data Analysis Extension Pack) (рис. 4.6 а).

Пакет обробки сигналів (Signal Processing Extension Pack). Містить біля 70 функцій для аналогової та цифрової обробки сигналів, аналізу та графічного представлення результатів (рис. 4.6 б).

Пакет обробки зображень (Image Processing Extension Pack) (рис. 4.6 в).



Рисунок 4.6. Електронні книги: а – аналізу даних, б – обробки сигналів, в – обробки зображень

Пакет для роботи з вейвлетами (Wavelets Extension Pack) містить біля 60-ти функцій ортогональних та біортогональних вейвлетів Хаара, Добеши, симлетів, койфлетів тощо.

Visual Electromagnetics for Mathcad (видавництво McGraw-Hill).

Mathcad E-Book on Wireless Channels.

## Електронні книги та бібліотеки для СКА MathCAD Prime

*Бібліотека будівництва* (Civil Engineering Library) коштує \$200. Містить біля 300 шаблонів розрахунків для будівельного проєктування та приклади теплових розрахунків.

*Електротехнічна бібліотека* (Electrical Engineering Library) коштує \$200. Містить біля 500 обчислювальних процедур, формул, довідникові таблиці.

Бібліотека машинобудування (Mechanical Engineering Library) коштує \$200. Містить біля 300 шаблонів розрахунків,

Довідник Роарка (Roark's Formulas for Stress and Strain) коштує \$425. Містить шаблони для розрахунку напружень і деформацій, моментів інерції, торсійного навантаження.

Бібліотека РТС том 1 (Mathcad Prime Worksheet Library - Volume 1) коштує \$520. Містить шаблони з алгебри, геометрії, статистики, аналізу даних, хімії, електротехніки, механіки, FEA.

Бібліотека РТС том 2 (Mathcad Worksheet Library - Volume 2) коштує \$520. 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) коштує \$200. Містить біля 400 шаблонів з диференційних рівнянь, статистики, оптимізації, аналізу даних.

### Хмарні обчислення в Mathcad

«Класичний» 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 .xmcd (MA/CS). Сервер приймає звертання «online» 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.

Методика розрахунків та проміжні дані в серверних розрахунках можуть бути відкриті або частково чи повністю закриті (рис.4.7).

Технологія MA/CS надає такі переваги:

*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.



Рис. 4.7. Документ MA/CS

*Мобільність*. Нові розрахункові методики стають моментально доступні всім авторизованим користувачам. Досить тільки повідомити їм відповідні інтернет-адреси сторінок.

*Комерціалізація*. Щоб розрахунки стали товаром, можна адмініструвати доступ до сайту, зробити його платним (помірно платним, частково платним), оточити посилання на них банерами тощо.

Будь-які помилки, недоробки в розрахунку швидко (і непомітно для користувачів) можуть бути виправлені. Непомітно для користувачів можна також *модернізувати* і розширювати розрахунки.

Зменшення вимог до кваліфікації користувачів. Проєктувальники розробляють документи, модифікують, виправляють помилки в повноцінному середовищі. Користувачі заносять в форми потрібні вихідні дані та отримують результат в звичайних Інтернет-браузерах.

3 березня 2016 р. РТС анонсувала *Mathcad Gateway* на базі *Mathcad Prime* 3.1 (https://www.ptc.com/en/mathcad-software-blog/announcing-ptc-mathcad-gateway). Вважалося, що користувачі можуть купити підписку на цей розрахунковий сервер та відсилати на нього свої файли, щоб проводити

розрахунки без встановлення самого пакета *Mathcad Prime*. Станом на 2021 рік хмарна реалізація *Mathcad Prime* споживачам не пропонується.

## Інтерфейс користувача MathCAD

Програмне вікно *MathCAD* має стандартне оформлення застосунків *Microsoft* стилю до 2007, тобто до введення *стрічки* (Ribbon) (рис. 4.8). В робочому вікні застосунка розміщені *рядок заголовка 1, головне меню 2, панелі інструментів 3, робочу область 4, рядок стану 5.* 



Рис. 4.8. Програмне вікно *MathCAD*: 1 – рядок заголовка; 2 – головне меню; 3 – панелі інструментів; 4 – робоча область; 5 – рядок стану [1]

*Рядок заголовка* розміщений нагорі вікна. Він виконує довідникову функцію. В нього виводиться ім'я відкритого файлу або назва *Untitled*, якщо відкрито новий документ.

*Рядок стану* розміщений нижній частині вікна. *Рядок стану* містить інформацію про режим обрахунків пакета: ручний/автоматичний, - та номер поточної сторінки документа.

Основними елементуми керування діями є головне меню та кнопкові панелі інструментів. Головне меню розташовано нагорі вікна під рядком заголовка. Кнопкові панелі за про мовчанням розташовані безпосередньо під *головним меню*. Вони є *плаваючими (*floating), тобто їх можна розмістити в будь-якій частині екрану. Саме в такій конфігурації показано рис. 4.8.

Головне меню має наступні пункти [1; 30]: File (Файл), Edit (Правка), View (Вигляд), Insert (Вставлення), Format (Формат), Tools (Математика), Symbolic (Символіка), Window (Вікна), Help (Допомога).

Меню File (Файл) визначає дії з документами пакета: створити новий, відкрити, зберегти, друкувати, – та дії для визначення параметрів сторінки документу тощо.

Меню **Edit** (Правка) визначає стандартні операції вирізки, копіювання, вставляння, OLE операції в документі.

Меню **Вид** (Вигляд) керує виглядом програмного вікна. Меню містить наступні пункти:

- **Toolbars** (Панелі). Пункт містить елементи керування видимістю кнопкових панелей (рис. 4.8), які можна вивести в програмне вікно або сховати;
- Ruler (Лінійка). Пункт керує видимістю мірильної лінійки над документом в робочій області. Одиниці вимірювання лінійки centimeters/inches/points обираються в її контекстному меню;
- Status bar (Рядок стану). Пункт керує видимістю рядка стану нижче робочої області;
- **Trace Window** (Трасування). Пункт керує видимістю вікна трасування програм;
- Header and Footer (Колонтітули). Пункт відкриває діалогове вікно, яке призначено для визначення розмірів, шрифтів, положення колонтитулів сторінок документа;
- **Regions** (Області). Пункт керує виділенням кольором існуючих в документі областей;
- Annotations (Анотації). Пункт керує видимістю приміток у вигляді кольорових дужок біля кожної області документа;
- **Refresh** (Оновити). Пункт оновлює інформацію на екрані;
- **Zoom** (Збільшення). Пункт виводить діалогове вікно для визначення масштабу зображення документа.

Меню Format (Формат) керує виглядом елементів документа в робочому вікні а також оформленням розташування частин документа вирівнюванням Align Regions та розсуванням Separate Regions. В цьому меню визначаються формати математичних виразів Equation, результатів розрахунків Result, текстів Text, Paragraph, Style, графіків Graph тощо.

Меню Tools (Математика) керує режимами обчислень Calculate, анімації Animation, налагодження програм Debug, містить засоби для визначення системних змінних точності, форматів даних, матриць Worksheet Options тощо.

Меню Symbolic (Символіка) запускає виконання символьних операцій та визначає вигляд результатів Evaluation Style.

Меню Window (Вікна) керує розташування кількох робочих вікон;

Меню **Help** (Допомога) містить елементи довідникова підсистеми. Для виклику довідника достатньо натиснути клавішу «F1». Якщо курсор встановлено на елемент документа, то натискання клавіші «F1» відкриває вікно довідникової системи з інформацією саме про цей елемент.

*Кнопкові панелі* можна вважати основним та найбільш зручним елементом оперативного керування *MathCAD* для введення операторів математичних дій, символів, підпрограм, графічних областей тощо.

Для вставки обраного елементу з панелі в документ достатньо встановити курсор в потрібне місце та клацнути мишею на відповідному значку панелі.

Кнопки виведення на екран більшості панелей згруповані на панелі Math (Математична). Призначення кнопкових панелей пояснено в таблиці 4.1.

Для виведення на екран потрібної панелі слід натиснути відповідну кнопку панелі **Math**. За потреби, панель можна розмістити в зручному для роботи місці робочого вікна простим перетягненням в нове місце (drag and drop). Коли панель виведено на екран, відповідна кнопка панелі **Math** відображається блакитною.

177

Значок	Призначення панелі	Значок	Призначення панелі
кнопки		кнопки	
	Calculator (Калькулятор). Введення знаків арифметичних операцій, цифр	<b>₩</b> <sub>M</sub>	Boolean (Логічна). Введення знаків логічних дій
	Matrix ( <i>Mampuui</i> ). Введення матриць, дії з матрицями	<u>∫ के</u>	Calculus      (Обчислення).        Введення      символів        алгебраїчних дій
Æ	<b>Graph</b> ( <i>Графіки</i> ). Побудова графіків	€	Symbolic (Символьна). Проведення обчислень в символьному вигляді
趵	<b>Programming</b> ( <i>Програмування</i> ). Введення операторів підпрограм-функцій	x =	Evaluation (Числення). Введення знаків обчислення
αβ	<b>Greek</b> ( <i>Грецька</i> ). Введення літер грецького алфавіту		

Таблиця 4.1. Кнопкові панелі MathCAD [2]

Панелі Standard, Formatting, Controls, Modifier, Debug, Custom Characters вивести на екран можна тільки пунктами меню View - Toolbars.

Вигляд та функції панелей **Standard** (Стандартна), **Formatting** (Форматування) (рис. 4.9) є подібними до відповідних панелі пакета текстового процесора *Microsoft Office*.



Рис. 4.9. Панелі стандартних дій

Панель Modifier (Модифікатори) слугує для тонкого налаштування типів даних при символьних перетвореннях. Елементи панелі Debug (Налагодження) призначені для налагодження програм. Панель Custom Characters (Обрані символи) містить деякі символи, які часто використовуються, як то: градуси Фаренгейта, Цельсія, символ плюс-мінус тощо.

## Програмне вікно MathCAD Prime

*MathCAD Prime* є багатомовним. Під час інсталяції системи одночасно встановлюються мовні пакети для англійської, німецької, французької, італійської, японської, корейської, китайської, російської мов. Мову інтерфейсу можна змінити без переустановлення системи зміною суфікса ярлика програми. Для англомовного інтерфейсу слід до ярлика, наприклад, «*C:\ProgramFiles(x64)\PTC\MathcadPrime XX\MathcadPrime.exe*» додати через пробіл суфікс *«/culture:en-US»*, щоб повний ярлик набув наступного вигляду «*C:\Program Files (x64)\PTC\Mathcad Prime XX\MathcadPrime.exe*" */culture:en-US*».

Для російськомовного інтерфейсу ярлик має мати суфікс *«/culture:ru-RU»*.

*MathCAD Prime* має інтерфейс на базі елементу типу стрічка (*ribbon*). Стрічка замінює меню та кнопкові панелі *MathCAD*.

В програмному вікні *MathCAD Prime* (рис. 4.10) розташовані стрічка 3, робоча область 4, рядок стану **Status Bar** 5. Над стрічкою містяться кнопка **PTC MathCAD** 1 та кнопки «швидкого доступу» **Quick Access Bar** 2.

Кнопка **PTC Mathcad Button** 1 є аналогом меню «класичного» *Mathcad*. Вона відкриває меню для дій з документами пакета: створити новий, відкрити, зберегти, друкувати, закрити.

Кнопки швидкого доступу **Quick Access Toolbar** 2 повторюють подібну панель *Microsoft Office*. Найбільш зручними можна назвати кнопки відміни невірної дії та повторення введеної дії.

Статусний рядок Status Bar 5 також є схожим на подібний елемент *Microsoft Office*. Він містить нумерацію поточної сторінки документа, індикатор режиму обчислень (зелений в разі ввімкнення автоматичного режиму обчислень), поля пошуку Find та заміни **Replace** інформації в робочому полі, повзунок масштабування зображення в робочій області, кнопки вигляду документа: повне відображення **Page View** чи спрощене відображення чернетки документа **Draft View**.

Базовим елементом керування є стрічка **Ribbon** 3. На закладинах (*tabs*) стрічки розташовані кнопки введення в робоче поле обчислювальних конструкцій або елементів оформлення, визначення форматів інформації тощо. Кнопки на вкладках згруповані в групи (*groups*).



Рис. 4.10. Програмне вікно MathCAD Prime

Вкладка Math (Математика) містить групи Regions (Області), Operators and Symbols (Оператори і символи), Style (Стиль), Units (Одиниці вимірювань), Clipboard (Буфер обміну) з елементуми для проведення математичних дій.

(Введення/Виведення) Вкладка Input/Output містить групи Worksheets (Документи), Data Import/Export (Імпорт/експорт ланих). Objects (Об'єкти). Controls (Елементи керування), (Інтеграція) з елементуми керування обміном даними з Integration файлами пакета та інших застосунків.

Вкладка Functions (Функції) містить кнопки виведення в робочу область вбудованих математичних функцій.

Вкладка Matrices/Tables (Матриці/Таблиці) містить групи Matrices and Tables (Матриці і таблиці), Rows and Columns (Рядки і стовпці), Result Format (Формат результатів) з кнопками керування створюваними матрицями та їх подальшого редагування.

Вкладка Plots (Графіки) містить групи Traces (Криві), Markers (Маркери), Styles (Стилі), Axes (Oci), Format Plot Values (Форматування значень графіків) з кнопками керування створюваними графіками та їх подальшого редагування.
Вкладка Math Formatting (Форматування формул) містить групи Math Font (Шрифт формул), Results (Результати), Label Styles (Стилі позначень) з полями визначення вигляду математичних виразів.

Вкладка Text Formatting (Форматування тексту) містить групи Text Font (Шрифт тексту), Paragraph (Параграф), Links (Посилання), Spelling (Правопис) з полями визначення вигляду текстів в робочій області.

Закладинка Calculation (Розрахунок) містить групи Controls (Елементи керування), Error Tracing (Трасування), Worksheet Settings (Параметри документа) з кнопками визначення режимів обчислення та системних змінних пакета.

Вкладка Document (Документ) містить групи Regions (Області), Spacing (Інтервал), Page (Сторінка), View (Вигляд), Header and Footer (Колонтітули), Clipboard (Буфер обміну) з полями визначення параметрів аркушів документу та розміщення областей на аркушах.

Вкладка Resources (Ресурси) містить групи Get Help (Допомога), Connect to PTC Community (Спільнота), Technical support (Технічна підтримка), Learning (Навчання) з елементуми для отримання доступу до локальної довідникової системи або інтернет ресурсів.

## Робоче поле

Робоче поле знаходиться в центрі програмного вікна. В ньому розташовуються одне чи декілька вікон документів. Документ відображає математичну, текстову, графічну інформацію, яку вводить користувач та результати розрахунків системи. Документи в робочому полі оформлюються на віртуальних аркушах на зразок текстового редактора *Microsoft Word*.

## Робоче поле MathCAD

Розміри, орієнтація та береги аркуша документа може бути визначені в діалоговому вікні (рис. 4.11) через меню File-Page Setup.

Колонтитули аркуша можуть бути визначені в діалоговому вікні (рис. 4.12), яке викликається з меню **View-Header and Footer**.

Page Setup X	Header and Footer	
Paper Size: A4	Header Footer	Right
Source: Auto ~ Orientation Поля (мм)	Use full width for each section	
Portrait     Left: 30.48 Right: 30.48     Landscape     Top: 30.48 Bottom: 30.48	Tools     Options       Image     Format     Start at page number     1 ÷       Image     Image     Different header and footer on first page	Frame Page Header Footer

Рис. 4.11. Меню параметрів аркуша Рис. 4.12. Вікно колонтитулів

Справа на робочому полі знаходиться вертикальна зелена риска. Вона розділяє документ на стовпці. Ширина стовбцю відповідає ширині аркуша, яка визначена в меню **File-Page Setup**. Область документа друкується по стовпцях згори вниз. По досягненню кінця інформації одного стовпця, система переходить до друкування сусіднього, який знаходиться праворуч.

Масштаб зображення на робочому полі найпростіше задавати перемикачем з панелі **Standard** (рис. 4.9). Збільшення більше 200 % або нестандартне збільшення, яке неможливо обрати стандартним перемикачем з панелі **Standard** можна задати з меню **View-Zoom**.

## Робоче поле MathCAD Prime

Зовнішній вигляд робочої області візуально є значно приємнішим для користувача. В області показується віртуальний аркуш документа. Розміри, орієнтація, береги та колонтитули легко задаються кнопками групи **Document-Page** стрічки (рис. 4.13)

Plots Math Formatting	Text Formatting	Calculation	Document	Resources
Letter (8.5" x 11")	Show Grid     Grid Size: Sta     Show Major (	indard •	Page Draft 30% - View	

Рис. 4.13. Група налаштувань аркуша документа

Для того, щоб було зручніше розташовувати елементи на аркуші передбачено можливість нанесення на аркуш допоміжної сітки (*grid*). Вигляд сітки: звичайна **Standard** з кроком 5 мм або мілка **Fine** з кроком 2.5 мм, – обирається кнопками групи **Document-Page** стрічки (рис. 4.13).

# Контрольні запитання

- 1. До якого класу програмного забезпечення відноситься MathCAD?
- 2. Які особливості пакета MathCAD?
- 3. Перерахуйте основні можливості пакета MathCAD?
- 4. Що розміщується в програмному вікні MathCAD?
- 5. Які функції довідникової системи MathCAD?
- 6. Що таке електронні книги MathCAD?
- 7. З яких елементів формується робоче поле *MathCAD*?
- 8. Для чого призначені панелі, які панелі присутні в пакеті MathCAD?
- 9. Що означає вертикальна лінія на робочому полі MathCAD?
- 10.3 яких елементів формується робоче поле MathCAD?
- 11.Назвіть складові СКМ *MathCad* та їхнє призначення.
- 12.Що таке Mathcad Application /Computation Server, для чого він використовується?

# 4.2. Робота з документами MathCAD

# Створення та редагування документа

В родині СКА *MathCAD* для використання форматів оформлення документів за замовчанням використовується як в офісних IT-системах принцип шаблонів (*template*). Вигляд документа за замовчанням визначається форматами з шаблону іменем **Normal**. Для *MathCAD* шаблони мають розширення .mctx, для MathCAD Prime – .xmct.

Документ *MathCAD* створюється шляхом введення в потрібне місце в визначеному форматі та редагування математичних залежностей, графіків, текстових описів тощо.

В залежності від типу інформації вона в документі вводиться в *математичні, текстові області, областіх графіків* та *оформлення. Області* (regions) в документі можуть розташовуватися в будь-якому порядку та кількості без обмежень

*Математична область* призначена для введення символів та операторів пакета, які описують математичну задачу: змінних, констант, формул, функцій, програм.

*Текстова область* призначена для введення коментарів, пояснень тощо. Інформація в *текстовій області* є рядками тексту, який може містити будь які

символи (кириличні, латинські тощо). Введення та редагування в *текстовій* області здійснюється як в звичайному текстовому редакторі.

В області графіків виводяться двовимірні та тривимірні графіки.

Область оформлення призначена для розміщення в документі бітових графічних зображень стандартних форматів .*jpg*, .*bmp*.

Для редагування існуюча область виділяється кліком миші. Кілька областей виділяються їх обведенням віртуальною рамкою рухами миші з натиснутою лівою кнопкою.

Для перенесення виділеної області в інше місце слід підвести курсор до рамки в місце, де курсор перетворюється на зображення руки *МаthCAD* або перехрестя для *МathCAD Prime*. 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 підвести курсор до рамки в місце, де курсор перетворюється на зображення стрілок та потягнути з натиснутою лівою кнопкою миші маркер рамки області

Зазвичай документ вміщує декілька областей. При їх зсуві чи масштабуванні одна область може частково або повністю перекрити іншу та закрити частину інформації. Для того, щоб візуально виділити межі областей в *MathCAD* призначено пункт меню **View-Regions**. Всі проміжки між областями зафарбовуються в сірий колір так, що світлі області стають виділеними на темному фоні.

Зсунути вже існуючі області, щоб вони були розташовані без перекриття можна в *MathCAD* пунктом меню Format-Separate Regions. Автоматично горизонтально або вертикально розсунути області в *MathCAD Prime* можна кнопкою стрічки Document-Spacing-Separate Regions.

Вирівняти області горизонтально чи вертикально в *MathCAD* можна пунктом меню Format-Align Regions. Дія над декількома областями проводиться з групуванням областей натисканням миші з утриманням клавіші «*Shift»* або рамкою виділення мишею.

Для визначення та редагування властивості областей застосовується контекстне меню. Контекстне меню активується кліком правої кнопки миші на області. Зміст меню визначається типом області.

При запуску системи за замовчанням встановлюється математична область введення. Ознакою математичної області є курсор у вигляді хреста червоного кольору в *MathCAD* та синього кольору в *MathCAD Prime*.

Для прибирання з екрану допоміжних, проміжних обчислень тощо області можуть «сховані» (collapse). Для обмеження доступу до інформації області можуть «блоковані» (lock), тобто закриті паролем.

Для застосування можливості схову та блокування областей, вони повинні бути розміщені в спеціальних зонах – *area*. *Area* являє собою контейнер, який займає весь рядок документа. Вводиться *area* в документ пунктом меню **Insert-Area** в *MathCAD* та кнопкою стрічки **Document-Region-Area** в *MathCAD Prime*.

Проведення прибирання та виведення *area* в *MathCAD* може проводитися кількома способами [2]:

- Подвійним кліком на верхньому маркері area (рис. 4.14 а);
- Пунктами **Collapse**, **Expand** контекстного меню *area* (рис.4.14 б);
- Пунктами Collapse, Expand меню Format (рис. 4.14 в).



Рис. 4.14. Керування групою областей: а – на маркері; б - контекстним меню; в – головним меню

В *MathCAD Prime* «сховати» область з екрану можна простим кліком на маркері (рис. 4.15 а) **Агеа** або пунктами **Collapse**, **Expand** контекстного меню **Агеа** (рис. 4.15 б).



Рис. 4.15. Керування групою областей: а – на маркері; б – контекстним меню

В *MathCAD* для блокування доступу до *Area* можна застосувати пункти **Lock**, **Unlock** контекстного меню верхнього маркера *Area* або пункти **Lock**, **Unlock** меню **Format**.

В *MathCAD Prime* блокування доступу до *Area* проводиться пунктом **Protect Area** контекстного меню *Area*.

Введення інформації в області робочого поля може бути проведено шляхом натискання комбінацій «гарячих» клавіш, написанням команди безпосередньо в тексті документа. вибором пункту меню або кнопки панелі в *MathCAD* / натисканням кнопки стрічки в *MathCAD Prime*.

# Робота з областями оформлення документа

Для оформлення документа в нього додаються тексти та ілюстрації. Прості тексти можуть складати однорядкові пояснення та коментарі. Для викладення змістовного матеріалу прості тексти поєднуються в абзаци. Ознакою створення текстової області є наявність прямокутної рамки навколо тексту.

Інформація в текстовій області може оформлюватися як в текстовому редакторі визначенням типу шрифту, кеглю, стилю та кольору символів, параметрів абзацу тощо за допомогою головного меню **Format-Text**, панелі **Formatting**, контекстного меню тексту.

## Засоби MathCAD

*Текстова область* може бути створена в *MathCAD* трьома способами. В усіх способах спочатку треба встановити курсор в потрібне місце документа [1; 2; 30]:

 просто почати вводити текст. Коли після першого слова буде введений пробіл, математична область перетвориться в текстову автоматично;

- почати введення інформації із символу " (подвійні комп'ютерні лапки). Відразу після введення символу на екрані з'явиться прямокутна рамка текстової області;
- застосувати пункт меню Insert-Text Region.

Текстова автоматично 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.

Корисною можливістю *MathCAD* є спроможність 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.

# Засоби MathCAD Prime

В MathCAD Prime для тексту визначено два елементи. Розміри текстового боксу (**Text box**) можна змінювати. Текстовий блок (**Text block**) займає цілі рядки документа.

Ввести текст можна тільки визначивши його тип в полі **Text** box/Text block-Region стрічки Document або Math (рис. 4.16).

$\sim$	<u>A</u>	Text Block
Chart	А	Text Box
nponent	<u>^</u>	Image
Regions		

Рис. 4.16. Вставляння тексту

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 вертикальної лінії.

Для текстової області можна задавати тип шрифту, його розмір, стиль, колір, параметри абзацу, колір фона. Визначення проводиться закладинки **Text Formatting** стрічки (рис. 4.17).

	Math	Input/Output	Function	ns Matrices/Tables	Plots	Math Formatting	Text Formatting	Calculatio
Taho B	oma I <u>U</u>	• <u>A</u> • <u>&amp;</u> •	11 •		CEP CEP	ABC Spell Check Dictiona	Proofing Languag ry English (United St	e: ates) 🔹
		Text Font		Paragraph	Links		Spelling	

Рис. 4.17. Вкладка форматування тексту

Вставлені в документ ілюстрації у вигляді бінарних файлів створюють області оформлення.

Зображення можуть бути вставлені в документ через буфер обміну (clipboard) або кнопками меню/стрічки.

Зображення, які вставлені через буфер обміну, стають звичайними ілюстраціями.

В MathCAD Prime кнопка стрічки Math/Documents-Regions-Ітаде виводить на екран поле Browse for Image (рис. 4.18 а), за яким відкривається стандартне вікно пошуку та завантаження графічного файлу. Різниці в поведінці та вигляді зображення, яке вставлене через буфер обміну та стрічку, відсутня. Зображення ніяк не виділяється на екрані і слугує простою ілюстрацією.



Рис. 4.18. Зображення: а – стрічкою; б – меню; в – контекстне меню обробки

В *MathCAD* меню **Insert-Picture** виводить на екран рамку з маркером. В маркер потрібно ввести текстовий рядок із специфікацією графічного файлу (рис. 4.18 б). Після завантаження зображення введений рядок залишається видимим в документі. Для завантаженого через меню зображення можна отримати дані про зображення та редагувати його. Клік мишею на такому зображенні відкриває контекстне вікно обробки зображення (рис. 4.18 в) [2; 30].

### Збереження документа

Документи пакета 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. Для СКА MathCAD Prime використовується особливий формат .mcdx. СКА MathCAD Prime може читати файли «класичного» СКА MathCAD в конвертуванням пунктом (рис. 4.19) XMCD Converter- MathCAD Worksheets- Input/Output стрічки.



Рис. 4.19. Конвертор файлів MathCad Prime

Зберегти документ (рис. 4.20) можна у внутрішньому форматі пакета як робочий файл з розширенням *.xmcd/.mcd* для *MathCad* та *.mcdx* для *MathCad Prime*, як шаблон з розширенням *.xmct/.mctx*, а також в форматі текстового форматованого файлу з розширенням *.rtf. MathCad* дозволяє зберегти документ як інтернет-сторінку в форматі *.htm*.



Рис. 4.20. Формати збереження документа

# Контрольні запитання

- 1. Яке розширення мають файли документів MathCad?
- 2. Які дії можна виконувати з областями робочого поля?
- 3. Як викликається контекстне меню?
- 4. Перерахуйте способи виклику команд *MathCad*.
- 5. Для чого області прибираються з екрану?
- 6. Як створити текстову область?

# 4.3. Організація обчислень

Дані користувача в пакет можуть вводитися як цілі, дійсні, комплексні числа, текстові рядки, одиниці вимірів фізичних величин.

Цілі числа містять тільки цифри (-1, 0, 1000 тощо). Дійсні числа можуть записуватися в формі фіксованої крапки (3.1515926) або в експоненційній формі через цілу частину у вигляді дійсного числа з фіксованою точкою **a** або цілого числа та основи системи числення з порядком **p** : **a**·10<sup>-p</sup>. Текстові рядки забираються в подвійні комп'ютерні лапки ("Це рядок").

Комплексні числа вводяться через дійсну та уявну частини як **a** + **bi**, де **a** – дійсна частина, **b** – уявна, **i** або **j** – уявна одиниця. Між уявною частиною та уявною одиницею знак множення НЕ СТАВИТЬСЯ.

Комплексний результат в *MathCad* виводиться через дійсну та уявну частину. Літера уявної одиниці «і», «ј» визначається користувачем у вікні форматування результату (рис. 4.21 а).

Комплексний результат в *MathCad Prime* виводиться через дійсну та уявну частину або через модуль числа та фазу числа. Форма відображення результату визначається користувачем в полі **Math Formatting-Result** стрічки (рис. 4.21 б).

Result Format		×	
Number Format Displ	ay Options Unit Display Tolerance		
Matrix display style	Automatic V		1 + 1i
Imaginary value Radix	i (1) V Decimal V		$1 + 1j$ $1.141 \angle 0.25\pi$ $1.141 \angle 0.785$ $1.141 \angle 45^{\circ}$
	a		б

Рис. 4.21. Формат комплексних чисел: a – MathCad; б – MathCad Prime

Описувати типи даних в пакеті не потрібно. Всі чисельні дані оброблюються пакетом як комплексні, з типом дійсної та уявної частини *double* в термінах мови С з точністю 15 знаків мантиси.

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 елементум буде надано тип **NaN**. При побудові графіків елементи з типом **NaN** ігноруються.

## Змінні

Дані, які можуть приймати різне значення, в пакеті називаються змінними (variables). Змінні використовуються у вигляді простих змінних, векторів та матриць, ранжованих змінних. Змінним може бути надана розмірність для розв'язання фізичних задач.

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

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, с – швидкість світла 2.998·10<sup>8</sup> м·с<sup>-1</sup>.

Для надання документу звичного вигляду ім'я змінної може містити підрядкові літери. Підрядкові літери НЕ Є ЗНАЧЕННЯМ ІНДЕКСУ МАСИВУ.

В *MathCad* підрядковий індекс вводиться за допомогою клавіші «.» (крапка). Наприклад, послідовність «С.і» дає змінну  $C_i$ , де «і» не є індексом, а є часткою імені.

В *MathCad Prime* підрядковий індекс вводиться кнопкою Math-Style-Subscript (рис. 4.22) стрічки.

a<sub>2</sub> Subscript Equation Break Style

Рис. 4.22. Підрядковий індекс

Великі та малі літери в іменах визначають різні імена.

Математичні дії в *MathCad* можуть проводитися в десятковій (decimal), восьмеричній (octal), двійковій (binary) та шістнадцятирічній (hex) системах числення. За замовчанням використовується десяткова система. Для проведення дій в цій системі ніяких додаткових дій не потребується. Визначення значення в восьмеричній системі проводиться введенням після чисельного значення літери «o», в шестнадцятиричній – «h», в двійковій – «b». Визначити систему числення результату можна в діалоговому вікні меню Format-Result-Display options-Radix

Для керування математичними діями в пакеті призначені кілька системних змінних [30]:

- ORIGIN змінна цілого типу. Визначає початковий індекс масивів.
   Значення за замовчанням 0;
- **TOL** змінна дійсного типу. Визначає точність чисельного обчислення рівнянь, інтегралів, похідних тощо. Значення за замовчанням 0.001;
- СТОL змінна дійсного типу. 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 та 8, відповідно.

Значення системних змінних можна задавати безпосередньо в робочому полі або у вікні меню Tools – Worksheet Options –Built-in-Variables (рис. 4.23 a) для *MathCad* та полів Calculation-Worksheet Settings стрічки (рис. 4.23 б) для *MathCad Prime*.

Worksheet Options			$\times$			
Unit System Built-In Variables	Dimensions Calculation	Compatibility Display				
Convergence Tolerance	(TOL) 0.001	(0.001)		Math Formatting	Text Formatting	Calculation
Seed value for random n	umbers 1	(1)		ORIGIN := 0 $\checkmark$ TOL := 10 <sup>-3</sup> $\checkmark$		
Precision (PRNPRECISIC Column Width (PRNCOL	N) 4 WIDTH) 8	<ul> <li>(4)</li> <li>(8)</li> </ul>		$\text{CTOL} := 10^{-3} \star$	Calculation Options	
		а		Worksheet S	Settings G	

Рис. 4.23. Системні змінні: a – MathCad; б – MathCad Prime

Грецькі літери вводяться комбінацією «латинська літера+Ctrl+G» (табл. 4.2) або з панелі Greece в MathCad та з панелі Math-Operator and Symbols-Symbols-Greece в MathCad Prime.

Таблиця 4.2. Клавіатурна відповідність літер грецького алфавіту [2]

Буква	α	β	δ	3	φ	Γ	Φ	λ	η	Ω	π	θ	ρ	σ	τ
Клавіша	a	В	d	e	F	G	Η	1	n	0	р	q	r	S	t

В пакет вбудовано кілька математичних констант (табл. 4.3).

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

Таблиця 4.3. Вбудовані математичні константи [2]

Для визначення змінних та функцій в пакеті застосовуються кілька знаків прирівнювання. *Локальне* прирівнювання позначається символами «:=». Його дія розповсюджується вправо-вниз від місця визначення. *Глобальне* прирівнювання позначається символом «≡». Його дія розповсюджується на весь документ незалежно від місця визначення знаку. Глобальне прирівнювання доцільно застосовувати для констант, які передбачається використовувати у всьому документі [30].

Для введення знаку локального прирівнювання з клавіатури слід ОДНУ клавішу «:». Наприклад, **x:=3.1415**.

Для введення знаку локального прирівнювання в *MathCad* можна використати панель **Evaluate**, в *MathCad Prime* – кнопку **Math-Operators and Symbols-Operators-Definition and Evaluation** стрічки.

Для введення знаку глобального прирівнювання в *MathCad* можна використати панель **Evaluate** або клавіші «*Shift*+~». Наприклад, **x=3.1415**.

Для введення знаку глобального прирівнювання в *MathCad Prime* можна використати кнопку **Math-Operators and Symbols-Operators-Definition and Evaluation** стрічки або клавіші «*Ctrl+Shift+~*».

Кількість прирівнювань однієї змінної в документі не обмежується.

Для виведення значень змінних та обрахунків призначено знак рівняння «=». Ввести знак рівняння можна клавішею «=» або кнопкою панелі Evaluate в *MathCad* та кнопкою Math-Operators and Symbols-Operators-Definition and Evaluation стрічки в *MathCad Prime*.

Знак рівняння може використовуватися в одному виразі після дії знаку прирівнювання. Наприклад, запис «c:=5+2=» присвоює змінній с значення обрахованого виразу та виводить на екран значення 7.

# Введення математичних виразів

## Засоби MathCAD

Ознакою готовності пакета сприймати математичні вирази є зображення курсору у вигляді червоного хреста. З початком введення в *математичній області* курсор набуває вигляд блакитного кута.

Орієнтація кута вправо або вліво показує напрям введення виразу. Якщо кут правий, то наступний символ буде вводитися праворуч, якщо кут лівий, то символ вводитиметься зліва.

Наприклад, символ дії ділення «/» для правої орієнтації кута введеться за операндом і фрагмент стане діленим, для лівої орієнтації – перед, та операнд стане дільником:



Зміна орієнтації кута проводиться натисканням клавіші Insert. Пересування по виразу проводиться клавішами стрілок «—»», «—».

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 клавішею пробілу **Space** так, щоб він охоплював весь фрагмент.

Знаки арифметичних операцій вводяться клавішами клавіатури або кнопками панелі **Calculator**. Клавіатурні коди деяких математичних операцій наведено в таблиці 4.4.

Наприклад, обрахування значення виразу  $\frac{13+15}{3\cdot(27-8)} \cdot \frac{56-34}{19} = 0.569$  може складатися з наступних дій [1]:

1) Набрати чисельник першого множника та клавішею **Space** охопити його правим кутом: ; **13+15** 

2) до виділеного виразу застосувати арифметичну операцію ділення:

 $\frac{13+15}{1}$ 

3) набрати знаменник та охопити весь дріб правим кутом: ;

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

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

13 + 15	56 - 34	]
3 · (27 – 8)	19	,

6) ввести оператор виведення «=»: .

13 + 15	56 - 34	- 0.569 -
3 · (27 - 8)	19	- 0.509

13 + 15

3 · (27 - 8)

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√3	Ctrl+\
степінь	$X^2$	^	комплексне пов'язання	$\overline{2+3j}$	Shift +"
додавання в два рядки	2 + 3	Ctrl+Enter	ділення в один рядок	2I3	Ctrl+/

Таблиця 4.4. Введення операцій з клавіатури [2]

## Шаблони виразів та функцій

Крім способу послідовного введення елементів математичних формул в пакеті можливо вводити вирази за допомогою шаблонів – заготовок арифметичних операцій з маркерами на місці операндів. Це дозволяє використовувати для запису виразів окрім звичного клавіатурного способу – *шаблонний* спосіб. Для шаблонного способу введення на першому етапі

створюється шаблон виразу. На другому етапі в поля маркерів необхідно вставити чисельні значення, імена змінних або інші математичні символи.

Шаблон виразу з позначеннями відповідної дії та маркерами виводиться на екран кнопками панелей **Evaluate**, **Calculator** (рис. 4.24).

sin(1)	cos(∎)	tan(∎)	ln(ı)	log(∎)	Calculator 🛛 🗙
<b>1</b> !	i		<b>√</b> ∎	<u>ار</u>	sin cos tan In log
	1				n!i  ×  √ "√"
e	÷	(1)	ı <sup>2</sup>	. '	$e^{X} \frac{1}{X}$ () $X^{2} X^{Y}$
					π 7 8 9 /
<sup>n</sup>	l '	0	,	•	<del> </del> 4 5 6 ×
1-	4	5	6	11	÷ 1 2 3 +
•					∷ · 0 - =
1 ÷ 1	1	2	3	I + I	
1 := 1		0	-1	= 1	

Рис. 4.24. Шаблони з панелі Calculator

Наприклад, клавіатурний порядок введення виразу

$$\sqrt{\frac{2}{3+4}} \cdot 5 = 2.673$$

може мати наступний вигляд:

$$\sqrt{\rightarrow}2\rightarrow/\rightarrow3\rightarrow+\rightarrow4\rightarrow$$
 Space $\rightarrow$  Space $\rightarrow$ Space $\rightarrow$  \* $\rightarrow$ 5 $\rightarrow$ =

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 
$$\frac{\ln\left(\frac{1+y}{1+y}\right)+y^{2}}{\sqrt[4]{1+y}} \cdot e^{1}$$
. Ця

структура заповнюється за 10 кроків: визначаються 7 змінних та 3 константи.

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.

 $x^{fy} \rightarrow \bullet \bullet \bullet 5$  atan2 6 та «*дерево»* у вигляді «дерева»:  $x^{fy} \rightarrow \bullet \bullet \bullet \to \bullet$ atan2 5 6 – для функцій двох аргументів [30].

Eval	×	ì			
=	:=	=	$\rightarrow$	•→	
fx	хf	xfy	$\mathbf{x}^{\mathbf{f}}\mathbf{y}$		

Рис. 4.25. Панель Evaluation

### Засоби MathCAD Prime

Ознакою *математичної області* є курсор у вигляді синього хреста. З початком введення курсор набуває вигляд блакитної вертикальної риски. Пересування по виразу проводиться клавішами стрілок «—», «—». Наступний символ вводиться з того боку виразу, де знаходиться маркер-риска: . 34 34+

Якщо маркер зліва лівий, то символ вводиться зліва: 34 +34

При введенні арифметичного знаку, дія якого відноситься до фрагменту математичного виразу, необхідно виділити фрагмент клавішею пробілу *«Space»*. Візуально виділений фрагмент виділяється затемненням

$$\frac{(12+34)\cdot 3}{6}$$

Знаки арифметичних операцій вводяться з клавіатури або кнопками панелі **Math-Operators and Symbols-Operators-Algebra** стрічки (рис. 4.26).

∛ Operators →	eta Symbols	Programming	π • Constants	x→ • Symbolics •	Labels (-) a <sub>2</sub> Subscript
Algebra					
+		-		÷	/
$x^y$		$\sqrt{x}$	x	x!	%
0		,			

Рис. 4.26. Панель операторів стрічки

Наприклад, обрахування значення виразу  $\frac{13+15}{3\cdot(27-8)} \cdot \frac{56-34}{19} = 0.569$  може складатися з наступних дій [1]:

1) Набрати чисельник першого множника та клавішею **Space** охопити його: (12+15);

2) до виділеного виразу застосувати арифметичну операцію ділення:

(12+	- 15)

3) набрати знаменник та виділити весь дріб: (12+15)
 4) ввести знак множення: (12+15)

4) ввести знак множення:

(12+15)
$3 \cdot (27 - 8)$

5) аналогічно пунктам 1-3 провести набір другого множника:

(12+15)	56 - 34
$3 \cdot (27 - 8)$	19

, 6) ввести оператор виведення « =»:	$\frac{(12+15)}{3\cdot(27-8)}\cdot\frac{56-34}{19}=0.548$

# Шаблони виразів та функцій

В MathCAD Prime також можна для запису виразів окрім звичного клавіатурного порядку використовувати шаблонний. Шаблони арифметичних дій вводяться кнопками панелі Math-Operators and Symbols-**Operators-Algebra** стрічки (рис. 4.26).

Шаблони префіксний, постфіксний, інфіксний, «дерево» функцій в MathCAD Prime не реалізовані.

# Формати виведення результатів [30]

Результати розрахунків можуть бути виведені на екран в наступних форматах Decimal, Scientific, Engineering, General, Fraction B *MathCAD*. В *MathCAD Prime* замість формату **Fraction** застосовано формат **Percent**.

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

Scientific – екпоненційний формат. Параметр Show exponents as E+000 визначає як виводити основу: у вигляді 10 або E  $(1.22 \cdot 10^2 \text{ або} 1.22\text{E}+2)$ .

**Engineering** – екпоненційний формат. Відмінність від формату **Scientific** полягає в тому, що показник степеню є кратним 3.

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

**Fraction** – формат у вигляді дробу. Параметр **Use mixed numbers** дозволяє виділити цілу частину дробу.

Наприклад, формат Fraction для виразу  $\frac{8}{6}$  без параметра Use mixed numbers показує результат  $\frac{4}{3}$ , з параметром –  $1\frac{1}{3}$ .

## Засоби MathCAD

Визначення формату результату проводиться в діалоговому вікні (рис. 4.27). Виклик вікна проводиться пунктом Format-Result-Number Format меню або подвійним кліком безпосередньо на результаті.

Визначення формату проводиться обраним пунктом діалогового вікна.

Result Format				$\times$				
Number Format	Display Options	Unit Display	Tolerance					
Format General Decimal Scientific Engineering Fraction	Number o	<sup>;</sup> decimal place trailing zeros exponents in e	s 3 🗘					
Exponential threshold 3								

Рис. 4.27. Вікно визначення форматів результатів

## Засоби MathCAD Prime

Визначення формату результату проводиться тільки полями панелі Math Formatting-Result стрічки (рис. 4.28).



## Рис. 4.28. Визначення форматів стрічкою

Верхнє поле панелі визначає формат: Decimal, Scientific, Engineering, General, Percent. Друге поле панелі кількість знаків в мантисі числа. Кнопка зліва внизу панелі перемикає показ нулів в числі (Show Trailing Zeros).

### Режими обчислень

Пакет дозволяє проводити обчислення в *автоматичному* режимі, коли всі обчислення виконуються безпосередньо під час введення розрахункових залежностей, та в *ручному* режимі, коли обчислення розрахункових залежностей запускається примусово вручну користувачем.

За замовчанням встановлюється автоматичний режим.

Коли вираз знаходиться в процесі розрахунку форма курсору змінюється, сам вираз позначається на екрані зеленою рамкою,. Будь-які дії користувача під час обчислень щодо подальшого редагування документа блокуються, крім клавіши переривання **Esc**. Після переривання вирази, які залишилися не вирахуваними, будуть позначені в документи червоним кольором.

Автоматичний режим є зручним, бо результати розрахунків виводяться в реальному часі безпосередньо після змін умов, що зменшує час на проведення аналізу. Складні обчислення можуть потребувати багато часу. Зміна однієї умови на початку великого документа призводить до перерахунку всіх залежностей заново. Корекція частини алгоритму програми також призводить до перерахунку всіх залежностей заново. В таких випадках рекомендується редагувати документ в ручному режимі, а обчислення включати в міру необхідності.

При редагуванні тексту в ручному режимі не виконуються ані обчислення виразів, ані побудова графіків, відповідні місця в виразах відзначаються маркерами (рис. 4.29).



Рис. 4.29. Позначки ручного режиму

Поточний режим розрахунків показується у відповідному полі в правій частині рядка стану.

В Mathcad в ручному режимі щоб обчислити увесь документ слід обрати пункт меню Tools-Calculate-Calculate Worksheet або натиснути клавіші «Ctrl+F9». Щоб обчислити всі формули на екрані слід обрати пункт меню Tools-Calculate-Calculate Now, клавішу «F9» або кнопкою Calculate із зображенням знака рівності на панелі Standard (рис. 4.29).

В *Mathcad Prime* керування режимами обчислень спрощене та проводиться тільки кнопками стрічки **Calculate-Controls** (рис. 4.30).

Натискання кнопки Auto Calculation вмикає автоматичний режим розрахунків. Кнопка Calculate проводить однократний перерахунок всього документа.

Щоб перервати тривалий процес обчислень слід натиснути клавішу Stop all Calculation. Перерване обчислення поновлюються натисканням кнопки Calculate.



Рис. 4.30. Керування режимами зі стрічки

## Оптимізація обчислень

«Особливістю *Mathcad* є прискорення чисельних обчислень за рахунок застосування елементів символьної математики. Безпосередньо перед чисельним розрахунком *Mathcad* автоматично намагається спростити вираз, використовуючи символьний 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 меню **Tools-Optimize**.

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

Ознакою того, що вираз знаходиться в режимі оптимізації є блакитний квадрат справа від виразу» [1; 2].

Оптимізація обчислень в Mathcad Prime не передбачена.

## Відключення обчислення окремих формул

Пакет дозволяє видалити з обчислень будь-який вираз. В *Mathcad* праворуч від відключених виразів встановлюється чорний прямокутний маркер. В *Mathcad Prime* відключені вирази помічаються напівпрозорим блакитним фоном. Наприклад,



# Засоби Mathcad

Найпростіше вимкнення обраного виразу проводиться пунктом **Disable Evaluations** в контекстному меню виразу:



Додатковим способом відключення обчислення обраного виразу є використання пункту **Disable Evaluations** сторінки **Calculation** діалогового вікна **Properties**. Вікно викликається однойменним пунктом контекстного меню виразу.

## Засоби Mathcad Prime

В *Mathcad Prime* керування режимами обчислень спрощене та проводиться тільки кнопкою **Disable Region** стрічки **Calculate-Controls** (рис. 4.30).

# Вбудовані функції

Пакет містить значну кількість *вбудованих* розрахункових функції. Їхня кількість сягає кількох сотень, що принципово більше за кількість вбудованих

функцій в мовах програмування високого рівня. Крім того, вбудовані функції, на відміну від мов програмування, є функціями комплексних змінних, тобто можуть мати в якості аргументів комплексні числа, результатами обрахунків таких функцій також є комплексні числа.

Вбудовані функції дозволяють вирішувати такі складні завдання, як задачі спектрального аналізу, статистичного аналізу, наближення функцій, розв'язання нелінійних та диференціальних рівнянь тощо.

Виклик вбудованої функції проводиться введенням її імені та заданням аргументів функції в дужках. Функцію можна ввести з клавіатури або засобами інтерфейсу.

В Mathcad ввести вбудовані функції можна пунктом меню Insert-Function, з панелі Calculate, клацнувши на кнопці *по панелі* інструментів Standard. При цьому відкривається діалогове вікно Insert Function (рис. 4.31), яке містить повний список функцій та опис їхнього синтаксису.



Рис. 4.31. Вікно вбудованих функцій Mathcad

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

В *Mathcad Prime* ввести вбудовані функції можна тільки з панелі **Functions-Functions** стрічки (рис. 4.32). Кнопка **All functions** відкриває діалогове вікно **Function** (рис. 4.33), яке містить повний список функцій і їх синтаксис.

	Math	Input/Output		Functions	Matrices/	Tabl	es	Plots	Math Fo	rmat	ting	Text Formattin	ng	Calculatio
<b>JO</b> E <b>C</b>	Bessel Curve Fitting ar Data Analysis	nd Smoothing	• • •	Design of $\vec{x}_{jj}$ Differentia File Acces	Experiments al Equations s	• • •		Image Proce Probability D Signal Proce	ssing Distribution ssing	• • •	₹} //\ [::]	Solving Statistics Vector and Matrix	• • •	<b>f</b> x All Functions
						Fu	nctior	IS						

# Рис. 4.32. Стрічка вбудованих функцій



Рис. 4.33. Список вбудованих функцій

Клік на позначці трикутника біля назви категорії функцій розкриває список функцій даної категорії. При наведенні курсору на ім'я функції виводиться коротке пояснення дії функції.

Всі вбудовані функції в списках діалогових вікон розбити на групи відповідно до спрямованості функцій [30]:

- 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 диференційних 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.

Наведемо деякі особливості найбільш уживаних вбудованих математичних функцій.

Аргументи тригонометричних функцій визначаються в радіанній мірі. Для позначення даних в градусах потрібно перевести аргумент з градусів в радіани вручну, використати вбудовану функцію переведення градусів в радіани DMS (x) або можна додати після значення аргументу в градусах службове слово **deg**.

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

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

#### Функції роботи з комплексними числами

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

Комплексне число **Z** визначається двома параметрами. Це можуть бути *diйсна* **C** і *уявна* **B** частини комплексного числа або його *модуль* A і *фаза* або *аргумент* – **ф**. Ці параметри є взаємопов'язаними. Опис комплексного числа через дійсну та уявну частини називають *алгебраїчною* формою, через амплітуду та фазу – *експоненційною*.

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

Пакет має вбудовані функції для оцінки параметрів комплексних чисел. Ці функції, як і більшість інших, мають загальноприйняті математичні назви: **Re(z)** для розрахунку дійсної частини, **Im(z)** – для уявної, **arg(z)** – для аргументу.

Слід мати на увазі, що «вбудовані функції *Mathcad* мають деякі особливості застосування (наприклад, обмежена область визначення, обробка комплексних аргументів тощо), незнання яких може привести до помилок. 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» [1].

#### Розширені арифметичні оператори

Пакет містить інструменти, які дозволяють спростити обчислення сум, добутків, похідних, визначених інтегралів, меж: *розширені арифметичні* оператори або операторі математичного аналізу.

При введенні таких операторів на екрані зображується шаблон для подальшого заповнення числовими або символьними значеннями. Ввести розширені арифметичні оператори можна комбінаціями клавіш з клавіатури або засобами інтерфейсу.

Розширені *арифметичні оператори* в *MathCAD* водяться з панелі Calculus (рис. 4.34 а), в *MathCAD Prime* – з панелі Math-Operators and Symbols-Operators-Calculus стрічки (рис. 4.34 б).



Рис. 4.34. Панелі розширених операторів: а – *MathCAD*; б – *MathCAD Prime* 

### Оператор обчислення суми

*MathCAD* має два шаблони обчислення суми. Один шаблон (рис. 4.35 а) потребує визначення всіх маркерів, тобто задання в операторі нижньої 2 та верхньої 3 меж, індексу суми 1, вираз доданку 4. Другий шаблон (рис. 4.35 б) орієнтований на застосування ранжованої змінної. В ньому діапазон підсумовування 1 задається іменем наперед визначеної змінної [30].

*MathCAD Prime* має один універсальний шаблони підсумовування (рис. 4.35 в). Він має три маркери. В разі безпосереднього визначення меж визначаються нижня межа 1 та верхня межа 3. Нижня межа 1 визначається у вигляді виразу прирівнювання (наприклад, i=1), верхня межа 3 – у вигляді числа, наперед визначеної змінної або виразу. В разі використання для визначення діапазону ранжованої змінної вказується тільки ім'я ранжованої змінної в маркері 1.



Рис. 4.35. Шаблони операторів сум та добутків: а – шаблон суми *MathCAD*; б – шаблон суми з ранжованою змінною *MathCAD*; в – шаблон *MathCAD Prime*; г – шаблон добутку

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

Для введення оператора в документ необхідно клікнути на відповідний кнопці панелі та заповнити поля шаблону.

## Оператор обчислення добутку

Дії для введення оператора множення в документ відрізняються від введення операторів підсумовування тільки кнопками, які треба натиснути.

## Функції користувача

Для потреб багаторазових розрахунків залежності з різними значеннями параметрів, модифікації вбудованих функцій в разі їхньої невідповідності вимогам користувача, створення нових функції, які відсутні серед вбудованих, призначені *функції користувача*.

Функції користувача є подібними до підпрограм-функцій в мовах програмування високого рівня (МПВР), з однією суттєвою особливістю – функція користувача може містити тільки один рядок коду. Найбільш близька до функції користувача пакета MathCad є функція DEFFN мови BASIC.

Застосування функцій користувача проводиться аналогічно до підпрограм-функцій МПВР та складається з двох етапів: опис або визначення та використання або виклик.

Опис *функції користувача* є обов'язковим етапом, який попередує наступним викликам функції. Опис *функції користувача* розміщується в робочому документі перед викликом функції.

Опис функції користувача складається з імені функції, списку формальних параметрів, за яким після знаку прирівнювання слідує рядок тіла функції:

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

Кожна *функція користувача* повинна мати оригінальне ім'я, по якому здійснюється звернення до цієї функції. Іменуються *функції користувача* аналогічно змінним пакета.

Після імені функції розміщується список *формальних* параметрів, взятий у круглі дужки. Через формальні параметри в функцію користувача передаються дані, необхідні для виконання обчислень функції. В якості формальних параметрів можуть використовуватися імена простих змінних, масивів і функцій. Формальні параметри записуються виключно іменами. Застосування виразів в якості формальних параметрів не дозволяється. Імена формальних параметрів є локальними. Вони не співвідносяться до таких же імен, які використовувалися в документі.

Елементи списку формальних параметрів відокремлюються один від одного комами.

Для виклику *функції користувача* необхідно записати ім'я функції з вказівкою в дужках списку *фактичних* параметрів.

Фактичні параметри визначають конкретні значення, для яких здійснюються обчислення в тілі функції. Імена фактичних параметрів при виклику функції можуть не збігатися з іменами її формальних параметрів.

В якості *фактичних параметрів* можна використовувати числа, імена змінних, арифметичні вирази, імена функцій.

Важливим є виконання умови, щоб кількість, типи і порядок запису фактичних і формальних параметрів співпадали.

Між фактичними і формальними параметрами має бути відповідність по кількості, порядку проходження і типу. Остання відповідність означає, що [1]:

- 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.

Наприклад, визначення функції користувача gipot(a,b), яка розраховує гіпотенузу прямокутного трикутника з катетами a, b може мати наступний вигляд:

gipot (a,b) :=  $\sqrt{a^2 + b^2}$ 

Виклик функції gipot (a, b) може проводитися як:

x := 3 y := 4

z := gipot(x, y) = 5 gipot(3, 2 + 2) = 5

### Розрахунки в діапазоні значень

Для випадків, коли потрібно проводити обчислення одного виразу не однократно, а багато разів при різних значеннях аргументів (наприклад, для

розрахунку масивів значень для побудови графіка) застосовуються інструменти для розрахунків в діапазоні значень. Цей вид розрахунків реалізується за допомогою *ранжованих* змінних (в деяких джерелах вони називаються *дискретними змінними*).

Дискретна змінна – змінна, яка приймає не одне, а декілька значень за законом арифметичної прогресії. Для визначення ранжованої змінної необхідні початкове значення змінної та кінцеве значення. На відміну від МПВР для ранжованої змінної задається не крок, а друге значення, яке дорівнює сумі першого значення та кроку зміни змінної [1].

Аналогом ранжованої змінної є операція зумовленого циклу **for** в МПВР.

Суттєвою відмінністю циклу ранжованої змінної від циклу програмування є те, що рядки циклу виконується по черзі послідовно для одного значення змінної циклу, а ранжований цикл виконується по черзі над рядками для всіх значень змінної.

Ранжована змінна задається в наступний спосіб:

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

Якщо крок змінної дорівнює одиниці, то задавати друге значення не обов'язково.

В *MathCAD* дві крапки між другим та останнім значеннями вводяться символом «:» (двокрапка) або кнопкою панелі **Matrix** (рис. 4.36 а).

В *MathCAD Prime* дві крапки між другим та останнім значеннями вводяться двома символами «.» (крапка). Шаблон оператора визначення ранжованої змінної вводиться кнопками панелі **Math-Operator and** Symbols-Operators-Vector and Matrix стрічки (рис. 4.36 б).

Matrix ×				
[] ×, × <sup>-1</sup>  ×	Vector and Matrix			
f(M) M <sup>C&gt;</sup> M <sup>T</sup> mn	×	$\ x\ $	["]	$M^{\langle\!\! \!\rangle}$
<b>\$•</b> \$ \$×\$ Σ∨ ∰	$M^{\widehat{\mathbb{L}}}$	$M^{\mathrm{T}}$	$1 \dots n$	1, 3n
а		б		



Наприклад, запис **x**:=0.5,0.7 .. 1.5 вводить ранжовану змінну **x**, яка змінюється від 0.5 до 1.5 з кроком 0.2.

Для виведення на екран всіх значень дискретної змінної слід набрати її ім'я та поставити після нього знак рівності «**х=**».

Вивести одне значення ранжованої змінної неможливо. Присвоїти значення ранжованої змінної іншій змінній НЕМОЖЛИВО. Наприклад, можливо розрахувати та вивести значення виразу, вбудованої функції чи функції користувача  $2 \times x$ , sin(x) з ранжованою змінною x, але присвоїти значення цього виразу іншій змінній  $y := 2 \times x$  неможливо.

# Контрольні запитання

- 1. Якими типами даних оперує MathCAD?
- 2. Які змінні існують в MathCAD?
- 3. Як вводяться грецькі літери в MathCAD?
- 4. Що означає символ «крапка» в імені змінної MathCAD?
- 5. В яких системах числення працює MathCAD?
- 6. Які дії прирівнювання реалізовані в MathCAD?
- 7. Що означає маркер типу блакитного кута?
- 8. Які є формати виведення результатів в MathCAD?
- 9. Що таке режими обчислень в *MathCAD*?
- 10.Що означає оптимізація обчислень в MathCAD?
- 11.Які основні типи вбудованих функцій MathCAD?
- 12. Для чого використовуються розширені арифметичні оператори?
- 13.Що таке шаблони виразів?
- 14.Що таке ранжована змінні?
- 15.Які особливості застосування дискретної змінної?

# 4.4. Перевірка умов

В розрахунках часто трапляється ситуація, коли вигляд розрахункового виразу або кроки алгоритму залежить від виконання якихось умов. Наприклад, розрахунок кусково-неперервних функцій, імпульсних сигналів, виборі потрібного з кількох коренів рівняння тощо. Такі дії називаються логічними або умовними. Вони реалізують виконання алгоритму з розгалуженням, які потрекують кількох гілок обчислювального процесу. Вибір конкретної гілки визначається результатом перевірки заданих умов.

Для перевірки умов в пакеті передбачені:

- унітарні логічні оператори (вирази відношень);
- логічні операції;

- логічні вирази;
- логічні функції.

Введення символів логічних умов може проводитися з клавіатури або в *MathCAD* кнопками панелі Boolean (рис. 4.37 a), в *MathCAD Prime* кнопками панелі Math-Operator and Symbols-Operators-Comparison стрічки (рис. 4.37 б).

Особливістю перевірки умов в пакеті є те, шо для вірна умова набуває значення 1, невірна умова — 0. Це обумовлено відсутністю в пакеті спеціального логічного типу даних «вірно»/ «невірно» (True/False).

Во	olea	n	×	Comparison				
=	<	>	≤	∈	=	$\oplus$	>	$\geq$
≥	¥	٦	$\wedge$	<	$\leq$	$\wedge$	7	$\vee$
۷	$\oplus$			$\neq$				
	a						б	

Рис. 4.37. Введення умов: а – панель **Вооlean**; б – стрічка

## Унітарні логічні оператори – вирази відношення

Унітарні оператори використовуються для порівняння значень двох арифметичних виразів між собою.

Вираз відношення записуються як два вирази або змінні зі знаком логічного відношення (логічної умови) між ними:

### <вираз 1> <знак відношення> <вираз 2>,

де в якості знака відношення виступають символи, наведені в таблиці 4.5.

Знак відношення	Введення	Знак відношення	Введення
=	Ctrl + =	≤	Ctrl + 0
<	<	2	Ctrl + 9
>	>	<i>≠</i>	Ctrl + 3

Таблиця 4.5. Символи логічних відносин [2]

Якщо задане відношення виконується, то вираз відносин набуває значення рівне 1, якщо не виконується – 0.

Знак операції рівності не співпадає «=» зі знаком виведення значень «=». Він вводиться символами «Ctrl + =».

Наприклад, запис виразу відношення може мати наступний вигляд:

х := 6 х≤2 = 0 Результат

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

виразів. Наприклад, x := 6  $2 \le x \le 8 = 1$ 

### Логічні операції

Логічні операції в пакеті реалізують булеву логіку: АБО (OR), I (AND), HI (NOT), ВИКЛЮЧАЮЧЕ АБО (XO).

Можливі значення логічних операцій наведено в таблиці 4.6

NOT ¬	AND $\land$	$OR \lor$	xor $\oplus$
0 - 1 = 1	$0 \wedge 0 = 0$	$0 \lor 0 = 0$	$0 \oplus 0 = 0$
1 - = 0	$0 \wedge 1 = 0$	$0 \lor 1 = 1$	$1 \oplus 0 = 1$
	$1 \wedge 0 = 0$	$1 \lor 0 = 1$	$0 \oplus 1 = 1$
	$1 \land 1 = 1$	$1 \lor 1 = 1$	$1 \oplus 1 = 0$

Таблиця 4.6. Значення логічних операцій

Логічна операція «АБО» (OR)

Позначається знаком «+» або «v» (Ctrl+Shift+6) і записується у вигляді

<Логічн.вир.1> + <Логічн.вир. 2>

<Логічн.вир.1> / <Логічн.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 «л» (Ctrl+Shift+7) і записується у вигляді

<Логічн.вир.1>. <Логічн.вир. 2>

### <Логічн.вир.1> < <Логічн.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*) і записується у вигляді

### ¬ <Логічн.вираз>

Результат дорівнює зворотним результату логічного виразу.

## Логічна операція «ВИКЛЮЧАЮЧЕ АБО» (XOR)

Позначається знаком «⊗» (Ctrl+Shift+5) і записується у вигляді

### <Логіч.вираз 1> ⊗ <Логіч.вираз 2>

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

Так як результатом логічних дій в пакеті є числа 0, 1, то *унітарні логічні* оператори, логічні операції та математичні операції можуть поєднуватися в один вираз – логічний вираз..

Логічні дії виконуються за наступними пріоритетами:

- 1. дії в дужках;
- 2. логічні операції (AND, OR, XOR);
- 3. вирази відношень (<,>,=).

Для однозначного обчислення логічного виразу слід використовувати круглі дужки. Наприклад,

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

## Умовна функція if

Функція **if** є повним аналогом таких функцій в МПВР та реалізує структуру «*якщо-то–інакше*», тобто *повну альтернативу*.

Функція вводиться з клавіатури в наступному вигляді:

## if (<умова>, <вираз 1>, <вираз 2>)

В якості умови можна застосовувати арифметичний вираз або логічний вираз. Арифметичний виразу вважається вірним, якщо його значення не дорівнює 0.

Якщо умова має результат більше 0, то функція **if** розраховує та повертає значення виразу 1, якщо умова має результат 0, то функція **if** набуває значення рівне значенню виразу 2. В якості виразів 1, 2 може використовуватися будь-яка припустима арифметична, логічна дія і сама функція **if**.

Використання в якості виразу 1, 2 самої функції **іf** називається *вкладеними умовними функціями* та дозволяє реалізовувати алгоритми з розгалуженням по трьох та більше умовах.

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 передбачені наступні корисні функції [30]:

**Ceil(x,y)** – розрахунок найменшого числа, яке кратне у та більше х.

- **ceil(x)** розрахунок найменшого цілого числа, яке більше х.
- **Floor** (**x**, **y**) розрахунок найбільшого числа, яке кратне у та менше х.
- **floor (x)** розрахунок найбільшого цілого числа, яке менше х
- Round (x, y) округлення х до найближчого числа, яке кратне у, за правилами математики.
- round (x, n) округлення х до п знаків після десяткової точки. Якщо другий аргумент відсутній, то округлення відбувається до найближчого цілого. Від'ємне значення п визначає кількість розрядів цілої частини числа.
- **Trunc (x, y)** розрахунок добутку від множення у на цілу частину ділення х на у. Якщо |x| < |y|, то повертається нуль.
- trunc (x) виділення цілої частини х.
- **signum (x)** функція повертає 1, коли x=0, в інших випадках x/|x|.
- **sign (x)** функція повертає 0, коли х=0, 1 коли х>0, -1 коли х <0.
- Ф(х) функція Хевісайда (функція сходинки). Повертає 0, коли x<0, та 1, коли x>0.
- **max(c)** пошук максимального елементу списку аргументів. В якості списку може бути вектор.
- **min(c)** пошук мінімального елементу списку аргументів.

## Контрольні запитання

- 1. Як ввести символи логічних умов?
- 2. Що таке унітарні логічні оператори?
- 3. Які значення приймають логічні оператори?
- 4. Які логічні операції реалізовані в MathCAD?
- 5. Яку структуру реалізує умовна функція?

### 4.5. Робота з векторами та матрицями

*Масивом* в комп'юториці називають структуру даних з кількох однотипних об'єктів (чисел, символів, рядків, тощо), яка розглядається як єдине ціле. Доступ до кожного елементу масиву здійснюється за допомогою індексу елементу. Індекс має порядковий тип, та слугує адресою елементу в масиві. Індекс виступає ідентифікатором при по елементних обчисленнях.

Термін масиви є терміном інформатики (комп'юторики). В математиці терміну масив відповідають терміни вектор, матриця, тензор. Одномірні масив відповідають векторам-рядкам, які містять елементи в одному рядку та векторам-стовпцям, які містять елементи в одному стовпці. Двовимірні масиви, елементи яких розташовані по рядках і стовпцях, відповідають матрицям, багатомірні масиви складної структури – тензорам.

Матрицею A розміром R х C вважається сукупність даних, які розташовані у вигляді таблиці з R рядків та C стовпців:

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

В середовищі *MathCAD* можливо використання векторів-рядків, векторів-стовпців, матриць та тензорів. Вектор вважається матрицею з одним стовпцем. На відміну від математичних визначень елементуми матриць можуть бути вирази, рядки тексту тощо.

Особливістю векторних та матричних розрахунків в пакеті є те, що не є обов'язковим обробляти масиви поелементно, більшість математичних дій виконуються за правилами матричної математики за замовчанням.

Для позначення елементу масиву використовується *індексована змінна*. На екрані індексована змінна виглядає так, як заведено в математиці – А<sub>і, і</sub>.
Початковий індекс матриць та векторів задається системною змінною **ORIGIN**. За замовчанням значення початкового індексу дорівнює 0.

В *MathCAD* індекс масивів може приймати не тільки натуральні, а й цілі значення, тобто бути ВІД'ЄМНИМИ. Визначити значення початкового індексу можна безпосередньо в тексті документу змінною **ORIGIN** або у вікні системних змінних (рис. 4.22).

В *MathCAD Prime* початковий індекс масивів може приймати значення 0 або 1. Змінити значення початкового індексу можна безпосередньо в тексті документу або в полі **Calculation-Worksheet Setting-ORIGIN** (рис. 4.22).

Ввести матрицю можна послідовним визначенням елементів як індексованих змінних або визначити матрицю за одне введення в шаблоні.

Для введення індексованої змінної слід ввести ім'я змінної та натиснути клавішу квадратних дужок «[». Це призводить до появи поруч з іменем маркеру індексу. В маркер слід ввести потрібний індекс чи індекси у вигляді чисел або імен раніше визначених змінних [30]:

$$\underline{A} \xrightarrow{} [\rightarrow \qquad A_{\bullet} \xrightarrow{} 1: \rightarrow \qquad A_{1}:= \bullet$$

Визначення елементу  $\mathbf{a}_{i,j}$  з індексами, більшими за раніше введені, призводить до автоматичного створення відповідного масиву розміром  $\mathbf{ixj}$ . Наприклад, введення нової індексованої змінної  $\mathbf{x}_2$  призведе до створення вектора-стовпця  $\mathbf{x}$  розміром 3х1, введення нової індексованої змінної  $\mathbf{A}_{2,3}$ призведе до появи матриці A розміром 3х4:

$$x_{2} \coloneqq 10 \qquad A_{2,3} \coloneqq 30$$
$$x = \begin{pmatrix} 0 \\ 0 \\ 10 \end{pmatrix} \qquad A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 30 \end{pmatrix}$$

Введення вектора (матриці) шаблоном проводиться заповненням маркерів шаблона вектора (матриці) відповідного розміру. Шаблон виводиться на екран діалоговим вікном.

В MathCAD вікно активується клавішами «Ctrl+M» клавіатури, пунктом меню Insert-Matrix, кнопкою 📖 панелі Matrix. У вікні потрібно визначити кількість рядків (rows) і стовпців (columns) матриці (рис. 4.38 а).

В *MathCAD Prime* вікно введення матриць (рис. 4.38 б) зроблено подібним до вікна створення таблиць в *Microsoft Office*. Виклик вікна проводиться з панелі **Matrices/Tables-Insert Matrix** стрічки. Для введення матриці достатньо виділити потрібну кількість комірок у вікні.

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



Рис. 4.38. Визначення матриць: а – у вікні; б – стрічкою

В *MathCAD* передбачено можливість відображення масивів у вигляді матриці та у вигляді таблиці (рис. 4.39).

	(100	0)			0	1	2	3
h	010	0		0	1	0	0	0
0 =	001	0	a =	1	0	1	0	0
	000	) 1)		2	0	0	1	0
	+			3	0	0	0	1

стиль «матриця» стиль «таблиця» Рис. 4.39. Стилі виведення масивів [1]

Стиль відображення масивів визначається на сторінці **Display** Options діалогового вікна **Result Format**. Вікно відкривається меню Format-Result. або подвійним кліком на самому масиві. В пункті **Matrix** display style на сторінці в списку містяться три стилі: Automatic, Matrix, Table (рис. 4.40).

Result Format					×
Number Format Display Matrix display style		y Options	Unit Display		
		Automatic	. ~		
Expand nes	ted arra	Matrix Table			

Рис. 4.40. Вікно визначення стилю зображення матриць

Матриця стилю **Matrix (Матриця)** завжди відображується у вигляді, який прийнято в математиці (рис. 4.40 а). Показуються завжди всі елементи матриці без нумерації рядків та стовпців. Для матриць великих розмірів зображення займає дуже багато місця на екрані [30].

Для стилю **Table (Таблиця)** перший рядок містить індекси стовпців, перший стовпець – індекси рядків подібно до вигляду таблиць MS Excel (рис. 4.39 б). За про мовчанням на екран виводиться тільки частина масиву в діапазоні індексів 1 ... 10. Для перегляду інформації, яка знаходиться поза екраном використовуються повзунки (Scrollbar) [30].

Значення максимального індексу елементу, який буде відображатися, можна визначити на сторінці **Data Range** вікна властивостей масиву-таблиці (рис. 4.41 а). Вікно відкривається пунктом **Properties** контекстного меню масиву-таблиці (рис. 4.41 б).

Display File Options Data Range	Properties	
Start at row:	Alignment	•Тор
Stop at row: 1	Copy Selection Select All	✓ <u>C</u> enter <u>B</u> ottom
ay     File Options     ∪ara Hange       ow	<u>D</u> eselect All	Above
Stop at end of data Stop at column: 1	<b>Export</b> <u>U</u> p One Level	Belo <u>w</u>

Рис. 4.41. Властивості таблиці: а – максимальний індекс; б – вирівнювання

В масиві-таблиці можна провести вирівнювання матриці відносно знаку прирівнювання пунктом **Alignment** контекстного меню масиву-таблиці (рис. 4.41 б).

Показ першого рядка та першого стовпця з індексами в масиві-таблиці можна відмінити. Для цього на сторінці **Display** вікна властивостей масивутаблиці слід змінити поле **Show column/row labels**.

Для стилю Automatic пакет сам визначає стиль виведення. Для розмірів масиву менше 10 використовується стиль Matrix, для більших – Table.

За замовчанням встановлено стиль Automatic.

MathCAD Prime матриця відображається тільки стилем Matrix.

В *MathCAD Prime* введений спеціальний тип даних *maблиця* (table), який являє собою різновид матриці (рис. 4.42). В таблиці дані згруповані по стовпцях. Кожен стовпець може мати ім'я, а елементи стовпця – на фізичну розмірність. Ім'я стовпців визначається в першому рядку таблиці. Одинці розмірності – в другому рядку. Введення таблиці проводиться з панелі **Matrices/Tables-Insert Tables** стрічки. В подальшому стовпці таблиці можуть ідентифікуватися по наданому імені та використовуватися як вектори-стовпці.



Рис. 4.42. Елемент «таблиця»

Вбудовані засоби *Mathcad*, які призначені для роботи з матрицями, зібрані на панелі **Matrix** (рис. 4.43 а), в *Mathcad Prime* – на панелі **Math**-**Operator and Symbols-Operators-Vector and Matrix** стрічки (рис. 4.43 б). Список вбудованих матричних функцій наведений в таблиці 4.7.



Рис. 4.43. Матричні оператори: а – *MathCAD [1]* ; б – *MathCAD Prime* Таблиця 4.7. Основні вбудовані матричні функції [2]

Оператор	Дія
length (v)	обчислює кількість елементів вектора v
last(v)	повертає індекс останнього елементу вектора v
rows (M)	повертає число рядків матриці М
cols(M)	повертає число стовпців матриці М
M <i></i>	виділяє і-й стовпець матриці М
identity(n)	створює одиничну квадратну матрицю розміром n x n

Таблиця 4.7. Продовження

Оператор	Дія
A-1	обернення матриці
v ,  A	довжина вектора v, детермінант матриці А
AT	транспонування матриці А (Ctrl+1)
A*	комплексне спряження (")
Σv	розрахунок суми елементів вектора v (Ctrl+4)
diag(v)	створює діагональну матрицю, на головній діагоналі котрої розташовані елементи вектора v.
matrix(m,nf)	створює матрицю, в якій (i,j)-й елемент дорівнює значенню функції f(i,j)
augment(A,B)	створює матрицю приєднанням горизонтально матриць А та В
stack(A,B)	створює матрицю приєднанням після останнього рядка матриці А матрицю В вниз
submatrix	вирізає з матриці А підматрицю з рядками з іг по јг та
(A,ir,jr,ic,j c)	стовицями з ис по је
rank (A)	розраховує ранг матриці: кількість лінійно незалежних рядків та стовпців
tr(A)	розраховує слід матриці: суму діагональних елементів
mean(A)	розраховує середнє значення елементів масиву
norme (M)	розраховує евклідову норму 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) rsort(M,n)	переставляє рядки матриці так, що п-й стовпець/рядок сортується за зростанням

# Арифметичні дії з масивами

Над векторами та матрицями можна проводити обчислення, як і над числами. Основні арифметичні матричні функції системи наведено в таблиці 4.8.

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

Таблиня 4.8.	Операто	ри арифмет	ичних лій з	масивами	[2]
таолици п.о.	Onepuio	րո սրություն	п ших диі э	Machbawin	

Позначення: матриця – А, В, М, вектор – v, u, скаляр – к.

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. Це можна реалізувати двома способами: послідовно виконуючи дії над кожним елементом масиву або використанням *оператора векторизації*. Векторизований об'єкт зображується зі стрілкою нагорі.

Оператор векторизації змінює зміст операцій. Наприклад, якщо А є матрицею, тоді запис **ехр (А)** є некоректним, оскільки аргументом функції **ехр** повинна бути проста змінна, а не матриця. Застосування до цієї функції

оператора векторизації дозволяє провести обчислення функції **ехр** від кожного елементу матриці і результатом також є матриця.

Для введення оператора векторизації в необхідно над виділеним об'єктом натиснути одночасно клавіші «Ctrl+-» або обрати символ « $\rightarrow$ » з панелі **Matrix** в *MathCAD*; одночасно натиснути клавіші « $Ctrl+Shift+^{}$ » або обрати символ « $\rightarrow$ » з панелі **Matrices/Tables-Vector/Matrices Operators** стрічки в *MathCAD Prime*.

# Контрольні запитання

- 1. Які масиви реалізовано в MathCAD?
- 2. Що таке індексовані змінні?
- 3. Як створюються масиви в MathCAD?
- 4. Що таке векторизація?
- 5. В чому особливість арифметичних дій з масивами?
- 6. Чим визначається початковий індекс масиву?

### 4.6. Програмування в MathCAD

Організувати обчислення в пакеті можна безпосереднім написанням відповідних операторів та функції в текст документа. Такий спосіб називається програмуванням в тексті документа. Він є прийнятним для простих алгоритмів, повноцінної реалізації циклічних дій та розгалуження алгоритмів він не забезпечує.

Іншим способом є використання підпрограм. В пакеті передбачено застосування підпрограм-функцій. Підпрограми-функції в пакеті є аналогами підпрограм-функцій в мовах програмування високого рівня.

Документ, в якому використовуються підпрограми-функції реалізує основний принцип *модульного програмування*, який полягає в написанні окремих незалежних програмних модулів. які викликаються алгоритмах користувача з різними аргументами.

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

# Етапи застосування підпрограм-функцій

Застосування підпрограм-функцій є аналогічним до застосування функцій користувача проводиться в два етапи: *опис підпрограми-функції та* виклик підпрограми-функції.

# Опис підпрограми-функції

Опис підпрограми є першим етапом. Опис підпрограми може бути розміщений в тому ж документі, де передбачається його використання, або в зовнішньому документі. Опис повинен бути розташований вище місця першого використання підпрограми.

Опис підпрограми-функції складається з імені підпрограми, розташованого після імені функції списку формальних параметрів в круглих дужках, знака прирівнювання та тіла підпрограми з міткою програмування за знаком прирівнювання.

В пакеті допускається, що підпрограма може не мати формальних параметрів. В такому разі дані в підпрограму передаються через імена змінних сеансу, визначених вище опису підпрограми. Правильність результатів підпрограм без формальних параметрів не може бути гарантована, бо результат залежить від оточення, в якому описано підпрограма. Тому ВИКОРИСТАННЯ ПІДПРОГРАМ-ФУНКЦІЙ БЕЗ ФОРМАЛЬНИХ ПАРАМЕТРІВ НЕ РЕКОМЕНДУЄТЬСЯ.

Міткою тіла підпрограми є вертикальна лінія. Лінія автоматично проставляється перед кожним новим доданим рядком тіла підпрограми. Всі операції та результати присвоєнь та розрахунків в тілі підпрограми є локальними. Тобто мають значення тільки всередині підпрограми. Складатися алгоритм підпрограми може з рядків присвоєння або розрахунку з локальним оператором присвоювання «←», умовних операторів **if** з оператором повного вибору **otherwise** (elseif), операторів передумовленого циклу **for**, ітераційного циклу **while**, спеціальних оператори переривання обчислень **break**, **continue**, **return**, **on error**.

Ввести нові рядки тіла підпрограми-функції можна клавішею «]» та кнопкою панелі **Programming** (рис. 4.46 а) в *MathCAD* або вкладки **Math-Programming** стрічки (рис. 4.44 б) в *MathCAD Prime*.



Рис. 4.44. Кнопки програмування: a – MathCAD; б – MathCAD Prime

В *MathCAD Prime* дещо змінено візуальне оформлення вбудованих операторів програмування та змінені деякі клавіатурні дії. Оператори відображаються більш зрозумілішими.

Введення вбудованих операторів в рядки підпрограми проводиться тільки кнопками панелі/стрічки.

Заготовка підпрограми **f** з ім'ям та одним формальним параметром **x** після введення двох нових рядків показана на рис. 4.45.



Рис. 4.45. Шаблон підпрограми

Оператори алгоритму розміщуються в маркерах шаблону. Додавання нових рядків відбувається нижче того маркера, на якому в той час знаходиться курсор. Для видалення рядка слід помістити на нього курсор і натиснути клавішу *Delete*.

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

Для визначення всередині підпрограми значення будь-якої змінної використовується локальний оператор присвоювання:

#### <ім'я змінної> - <вираз>

Використання «звичайного» оператора присвоювання «:=» в тілі програми-функції призводить до синтаксичної помилки.

### Виклик підпрограми-функції

Результат підпрограми можна вивести записавши її ім'я зі списком фактичних параметрів зі знаком виведення «=». Для застосування результатів підпрограми-функції необхідно використати її ім'я зі списком фактичних параметрів в розрахунковому виразі справа від знаку прирівнювання,

Співвідношення між фактичними і формальними параметрами для підпрограми-функції є таким же, як і для функцій користувача.

Наприклад, підпрограма для розрахунку відстані між двома точками, які визначені своїми декартовими координатами на площині, може виглядати наступним чином:



Підпрограма має назву dist та чотири формальні аргументи  $\mathbf{x}_{A}$ ,  $\mathbf{x}_{B}$ ,  $\mathbf{y}_{A}$ ,  $\mathbf{y}_{B}$ , які означають відповідні координати точок A та B. Наведений простий приклад має лінійний алгоритм, в якому необхідні операції необхідні операції виконуються послідовно. Оператори, що реалізують цей алгоритм у тілі підпрограми-функції також розміщуються послідовно і виконуються всі, починаючи з першого оператора і закінчуючи останнім.

Заготовка функції створюється введенням назви підпрограми dist(x<sub>A</sub>, x<sub>B</sub>, y<sub>A</sub>, y<sub>B</sub>,), знаку прирівнювання «:=» та двох маркерів рядків тіла підпрограми кнопками або клавішею «]».

Маркери заповнюються потрібними виразами із використанням символу локального прирівнювання.

Перший виклик підпрограми не зберігає її результат, він проводиться для виведення результату на екран. Другий виклик використовує результат підпрограми та зберігає дані в змінній total.

### Алгоритми, що розгалужуються

«У розгалужених алгоритмах присутні кілька гілок обчислювального процесу. Вибір конкретної гілки залежить від виконання (чи невиконання) заданих умов» [2].

Для розгалуження алгоритмів в підпрограмах пакета призначений програмний умовний оператор **if**.

Програмний умовний оператор застосовується тільки в тілі підпрограми-функції. Він реалізує «неповну» альтернативу (рис. 4.46 а). В разі виконання умови керування передається операторам 1, в разі невиконання умови керування передається операторам 2, які розташовані після умовного оператора.

В тіло підпрограми оператор вводиться клавішею «}» або кнопкою **if** панелі або стрічки. Шаблон оператора містить два маркера (рис.4. 46 б, в). Справа від службового слова **if** знаходиться маркер 1 умови. Сюди вписується логічний вираз (у найпростішому випадку це вираз відношення) умови. Умова вважається вірною якщо значення виразу не дорівнює нулю. В маркер 2 вводяться оператори, які повинні бути виконані якщо логічний вираз буде вірним.



Рис. 4.46. Оператор **if**: а – алгоритм; б – *MathCAD*; в – *MathCAD Prime* 

«Повна» альтернатива передбачає проведення визначених дій як для випадку виконання, так і для невиконання умови (рис. 4.47 а). В разі виконання умови керування передається операторам 1, в разі невиконання умови керування передається операторам 2. Продовження алгоритму операторами 3, які розташовані після **if**, відбувається тільки обов'язкового виконання операторів 1 або 2.



Рис. 4.47. «Повна альтернатива»: а – алгоритм; б – шаблон; в – реалізація

Для «повної» альтернативи в *MathCAD* застосовується оператор **otherwise**. Оператор вставляється в рядок маркера 1, наступний за оператором **if** (рис. 4.47 б). Вирази, що стоять перед словом **otherwise**, будуть виконуватися тільки в тому випадку, якщо не виконано умову попереднього **if** (рис. 4.47 в).

В разі використання кількох операторів **if** вираз **otherwise** буде виконуватися, якщо жодна з умов в попередніх **if** не виконується.

Для «повної» альтернативи в *MathCAD Prime* застосовується оператор else. Оператор вставляється в рядок, наступний за оператором if. Вирази, що стоять під словом else в маркері 1 (рис. 4.48 а), будуть виконуватися тільки в тому випадку, якщо не виконано умову if. Оператор else може бути тільки один для оператора if.



Рис. 4.48. «Повна» альтернатива **else**: а – шаблон; в – реалізація

За функціями оператор **elseif** він є аналогічним програмному логічному оператору **if** та має такий же синтаксис. Він також має два маркери. Маркер 1 призначений для розміщення умови для перевірки. В маркер 2 вводиться вираз для розрахунку. Оператор повинен розміщуватися ПІСЛЯ оператора **if**. Оператор **elseif** буде виконуватися в тому випадку, коли умова попереднього оператора **if** не виконана.

Оператор **else** може бути тільки один для оператора **if** та повинен стояти в кінці ланцюжка перевірок **if-elseif-..-elseif-else**.

В *MathCAD Prime* програмний умовний оператор **if** з додатковими умовними операторами **else**, **elseif** може реалізувати більш складну перевірку. Повний комплект операторів реалізує множинну перевірку за кількома умовами (рис. 4.49 а).



Рис. 4.49. Алгоритм дії оператора **іf** MathCAD Prime

Перша перевірка Умова 1 проводиться оператором if. В разі, коли Умова 1 є вірною, виконуються Оператори 1 оператора if. Якщо Умова 1 невірна, виконується наступний за if оператор elseif, який перевіряє Умову 2. В разі, коли Умова 2 є вірною, виконуються Оператори 2 оператора elsef. Якщо Умова 2 невірна, виконується наступний за elseif оператор elseif, який перевіряє Умову 3. В разі, коли Умова 3 є вірною, виконуються Оператори 3 оператора elsef. Якщо Умова 3 невірна, виконується наступний за elseif оператор else, який виконує Оператори 4.

В маркери введення виразів логічних операторів може бути додано декілька рядків аналогічно блоковому оператору *begin - end* мови Паскаль або складному оператору «{}» мови С.

Наприклад, підпрограма моделювання 2-бітного аналого-цифрового перетворювача, який працює в діапазоні від 0 В до 1 В, може мати вигляд, наведений на рис. 4.50. Чотири можливі значення вихідної напруги визначаються вбудованим компаратором за рівнем вхідної напруги. Мінімальне значення 0 – для від'ємного входу та 0 В, максимальне 3 – для входу більше 0.66 В.

Рис. 4.50. Модель АЦП: а – «повна» альтернатива *MathCAD*; б – «неповна» альтернатива; в – «повна» альтернатива *MathCAD Prime*;

Проаналізуйте наведений нижче варіант підпрограми. Чи видає він правильні результати для функції ?

$$z(t) = \begin{cases} t^{3}, t < -3; \\ \ln(t), t > 4; \\ t^{2}, -3 \le t \le 4. \end{cases}$$
$$z(t) \coloneqq \| \begin{array}{c} \text{if } t \le -3 \\ \| t^{3} \\ \text{if } -3 < t \le 4 \\ \| t^{2} \\ \text{else} \\ \| \ln(t) \\ \end{array}$$

Циклічні алгоритми

Фрагменти дій, які необхідно повторювати кілька разів створюють циклічний алгоритм (або цикл). Кількість повторень залежить від значення деякої змінної. Така змінна називається *параметром циклу*, а сам фрагмент складає *тіло циклу*. Розрізняють *зумовлені* та *ітераційні* цикли.

Ознакою зумовлених циклів є відома апріорі кількість повторення кроків циклу, тобто визначення параметра циклу. Параметр циклу може бути заданий у вигляді вектора явно, коли визначені всі елементи вектора. Кількість кроків циклу в такому випадку дорівнює кількості елементів вектора. Для циклу типу арифметичної прогресії параметр циклу визначається по закону арифметичної прогресії начальним значенням  $x_0$ , кінцевим  $x_k$  та кроком зміни **d**. В такому випадку кількість кроків циклу **N** можна визначити за виразом

Кількість повторень циклу N визначається за наступною формулою

$$N = \frac{x_k - x_0}{d} + 1.$$

В ітераційних циклах параметр циклу змінюється за більш складним законом. Визначити наперед його значення неможливо. Для закінчення ітераційний цикл має містити перевірку деякої умови закінчення. Кількість кроків повторень циклу залежить від значення умови закінчення.

#### Зумовлений цикл

Зумовлений цикл в обох версіях пакету виконується оператором циклу **for**. Іноді його називають оператором циклу з параметром

Ввести оператор можна клавішами «*Ctrl+Shift+*"» або кнопкою for панелі/стрічки.

Шаблон оператора містить три маркери (рис. 4.51).



Рис. 4.51. Шаблон оператора for

В маркер 1 вводиться ім'я змінної лічильника циклу, в маркер 2 – значення параметра циклу. Діапазон значень вводиться аналогічно діапазону ранжованої змінної. В маркер 3 вводяться оператори, що складають тіло циклу. За потреби в маркер 3 додається необхідна кількість додаткових рядків.

Наприклад, функція **f**, яка для аргументу-вектора x розраховує суму його елементів зумовленим циклом, може мати наступний вигляд (рис. 4.52).

Рис. 4.52. Цикл : а – *MathCAD*; б – *MathCAD Prime*; в – з векторним параметром циклу

В наведеному варіанті підпрограма має один формальний параметр – вектор х, для якого потрібно розрахувати суму елементів. Він використовується для визначення діапазону зміни параметра циклу. В варіанті з доступом до елементу вектора індексованою змінною параметром циклу виступає змінна і, яка містить можливі індекси вектора. Індекси проходять значення від 0 до N з кроком 1. В варіанті векторним параметром (рис. 4.52 в) циклу змінною циклу є z, яка на кожному кроці циклу приймає відповідне значення елементу вектора х. Необхідності в введенні індексної змінної немає.

#### Ітераційні цикли

Для програмування таких циклів використовується оператор циклу while.

Ввести оператор можна клавішами «*Ctrl+J*» або кнопкою while панелі/стрічки.

Шаблон оператора має два маркери (рис. 4.53). В маркер 1 вводиться вираз умови виконання циклу, в маркер 2 – оператори тіла циклу.



Рис. 4.53. Шаблон оператора while

Оператор циклу while  $\epsilon$  по суті послідовністю логічного оператора перевірки умови, рядків тіла циклу та оператора безумовного переходу. На першому кроці в рядку while перевіряється зазначена умова. Якщо умова  $\epsilon$ вірною, то виконується тіло циклу. Після завершення тіла циклу керування передається рядку while і знову перевіряється умова. Якщо умова не виконується, то цикл завершується.

Використання оператора **while**, потребує уваги. Застосування без додаткових засобів контролю може привести до зациклення, коли умова не буде виконана ніколи, тому в тілі циклу повинні бути присутніми оператори, які гарантовано надають параметру циклу значення, яке робить умову циклу невірною, інакше цикл буде продовжуватися нескінченно.

Наприклад, функція **f**, яка для аргументу-вектора x розраховує суму його елементів ітераційним циклом, може мати наступний вигляд (рис. 4.54).



Рис. 4.54. Ітераційний цикл : a – MathCAD; б – MathCAD Prime

В наведеному варіанті підпрограма має один формальний параметр – вектор х, для якого потрібно розрахувати суму елементів. Він використовується для визначення максимального індексу елементів вектора. В якості умови циклу використовується порівняння поточного індексу з довжиною вектора. Якщо поточний індекс менше довжин, то виконується тіло циклу в кому розраховується сума елементів S. Так як параметр циклу автоматично не змінюється, в тілі циклу передбачено збільшення поточного індексу на 1. Відсутність такої дії веде до нескінченного циклу.

В пакет введені оператори **break**, **continue**, **return**, **on error**, які призначені для розв'язання таких ситуацій, як вихід з циклу, продовження циклу, переривання дії, виявлення похибки, в пакеті

Оператор break призначений для виходу з циклу for або нескінченного циклу while. Застосовується він разом з логічним оператором if. При виконанні визначеної умови цикл переривається, керування передається наступному за циклом рядку.

Використання оператора **break** скоріш за все говорить про недолугий алгоритм розв'язання задачі.

Ввести оператор можна клавішами «*Ctrl*+{» або кнопкою break панелі/стрічки. Вставляється оператор break в маркер 2 логічного оператор if. Відповідно, виконується break коли умова припинення роботи циклу в логічному операторі є вірною.

233

Наступний фрагмент показує підпрограму розрахунку суми з оператором **break**:

$$f(x) \coloneqq \left| \begin{array}{c} N \leftarrow \operatorname{length}(x) \\ S \leftarrow 0 \\ i \leftarrow -1 \\ \text{while } 1 \\ \| \text{ if } i = N \\ \| \| \operatorname{break} \\ i \leftarrow i + 1 \\ S \leftarrow S + x_i \end{array} \right|$$

В алгоритмі використаний нескінченний цикл. Штучний вихід з циклу проводиться за умови досягнення поточного індексу максимального значення. Перевірка умови проводиться логічним оператором **if**.

Оператор **continue** перериває поточний крок циклу та повертає керування на перший рядок циклу без виконання рядків, які розташовані після оператора **continue**.

Наступний фрагмент показує підпрограму розрахунку суми непарних елементів вектора з оператором **continue**.

Підпрограма може мати наступний вигляд:

$$smodd(x) \coloneqq \left\| \begin{array}{l} N \leftarrow \operatorname{length}(x) \\ S \leftarrow 0 \\ \text{for } z \in x \\ \\ \| \begin{array}{l} \inf \operatorname{mod}(z, 2) = 0 \\ \\ \| \operatorname{continue} \\ \\ S \leftarrow S + z \end{array} \right\|$$

Для визначення парності в підпрограмі використовується вбудована функція **mod** (**x**, **2**), яка визначає залишок від ділення числа на 2. В циклі по всіх елементух вектора **x** проводиться перевірка додатності. Якщо елемент  $x_i$ є парним, то дії сумування нижче пропускаються і цикл повторюється для нового елементу вектора **x**.

Дія оператора переривання є подібною до дії оператора **break**. Оператор перериває виконання алгоритму, завершує підпрограму та передає керування модулю, який викликав підпрограму. Шаблон оператора містить один маркер. В нього вводиться вираз, значення якого стане результатом роботи підпрограми. Використовується оператор **return** в парі з логічним оператором if для перевірки умови, яка повинна викликати переривання підпрограми.

Наступний фрагмент показує підпрограму, яка знаходить та виводить перший від'ємний або перший додатний елемент в масиві з відповідними коментарями.

$$smd(x) \coloneqq \left\| \begin{array}{c} \operatorname{for} z \in x \\ \| \operatorname{if} z < 0 \\ \| \operatorname{return} [\operatorname{``First negative''} z] \\ \| \operatorname{if} z < 0 \\ \| \operatorname{return} [\operatorname{``First positive''} z] \\ \operatorname{``Only zeroes''} \end{array} \right\|$$

В наведеному варіанті підпрограма **smd** має один формальний параметр – вектор х. Зумовлений цикл переглядає по черзі всі елементи вектора. В разі виявлення першого ж від'ємного оператор **return** перериває підпрограму та повертає повідомлення **First negative** та відповідне значення. В разі виявлення першого ж від'ємного оператор **return** перериває підпрограму та повертає повідомлення **First positive** та відповідне значення. В протилежному випадку підпрограма формує повідомлення **Only zeroes**.

Оператор try/on error дозволяє виявити можливі обчислювальні помилки.

Оператор вставляється в порожній рядок в наступному синтаксисі:

#### <Bupas 1> on error <Bupas 2>.

Оператор намагається виконати **Вираз** 2. Якщо при виконанні **Виразу** 2 виникає помилка, то оператор виконує **Вираз** 1.

Наприклад, використання оператора **оп еггог** для обчислення функції  $\frac{\sin(x)}{x}$  може мати наступний вигляд:

$$sinc(x) \coloneqq try$$

$$\left\| \frac{\sin(x)}{x} \right\|$$
on error
$$\left\| 1 \right\|$$

# Модульне програмування в МАТНСАД

Одним з принципів структурного програмування є модульність. Сутність модульності, викладена вперше професором Ніклаусом Віртом полягає в тому, що обчислювальні процеси реалізуються у вигляді окремих універсальних програмних одиниць-модулів. Складні програми збираються з універсальних модулі. Для реалізації задачі алгоритма програма викликає модуль, передає в нього необхідні дані та отримує для обробки результати роботи модуля.

Застосування модульної архітектури зменшує обсяг лістингів програм, зробити їх більш чіткими, пришвидшити розроблення і тестування програм тощо програм.

СКМ *MathCAD* підтримує модульне програмування в межах одного документа та модульне програмування в декількох документах засобами підпрограм-функцій та зовнішніх посилань (*Reference*).

При модульному програмуванні в межах одного документа описи підпрограм-функцій та їхні виклики знаходяться в межах одного документа. Всі описи підпрограм та результати їх роботи зберігаються в одному файлі. Підпрограма-функція може містити виклики вбудованих функцій та інших підпрограм-функції документа.

Модульне програмування в одному документі унеможливлює колективну роботу одночасної розробки декількох модулів, унеможливлює налагодження модулів без наявності головного документу, унеможливлює використання готової підпрограми-функції в декількох документах без дублювання опису підпрограми-функції заново в кожному документі.

Модульне програмування в кількох документах долає вказані недоліки. За такої технології опис підпрограми-функції проводиться в одному документі, а використовується підпрограма-функція в іншому документі як зовнішня.

Уставляються зовнішні підпрограми-функції до файлу, в якому потрібно використати цю підпрограму засобами імпортування або зовнішніх посилань. В *MathCAD* імпортування проводить оператор **Reference**, в *MathCAD Prime* – оператор **Include**.

Застосування зовнішньої підпрограми функції в документі передбачає проведення трьох кроків. Попередньо необхідно розробити потрібну підпрограму та зберегти її у дисковому файлі. На другому кроці необхідно

вбудованим оператором імпортування уставити зовнішню підпрограму в робочий документ. Після уставляння виклик зовнішньої підпрограми нічим не відрізняється від використання підпрограм-функцій пакета.

Уставляння зовнішньої підпрограми в *MathCAD* проводиться пунктом меню **Insert-Reference** (рис. 4.55 а). Пункт відкриває діалогове вікно (рис. 4.56 а) для вибору зовнішнього файлу.



Рис. 4.55. Засоби уставляння зовнішніх підпрограм: a – *MathCAD*; б – *MathCAD Prime* 

Insert Reference	×	
Insert reference to file:		
	Browse	
Enter or locate the path to the document you want to reference. This can be a d on your hard drive, or a document on your company's network.	locument	Include <<
2	б	
a	0	

Рис. 4.56. Вибір файлу зовнішніх підпрограм: a – *MathCAD*; б – *MathCAD Prime* 

Після виконання цих кроків в документі з'явиться позначка імпортування (рис. 4.57 а) та зовнішня підпрограма стане доступною в відкритому документі.



Рис. 4.57. Позначка імпортування: а – *MathCAD*; б – *MathCAD Prime* 

Уставляння зовнішньої підпрограми в *MathCAD Prime* проводиться кнопкою **Input/Output-Include Worksheet** стрічки (рис. 4.55 б). Пункт виводить на екран діалогове вікно (рис. 4.56 б) для вибору зовнішнього файлу. Після виконання цих кроків в документі з'явиться позначка

імпортування (рис. 4.57 б) та зовнішня підпрограма стане доступною в відкритому документі.

# Контрольні запитання

- 1. Які способи програмування реалізовані в MathCAD?
- В чому різниця між функцією користувача та підпрограмоюфункцією?
- 3. Які правила опису підпрограми-функції?
- 4. Як реалізувати розгалуження в підпрограмі функції?
- 5. Які різновиди циклів є в підпрограмі-функції?
- 6. Які способи переривання дій існують в підпрограмі?
- 7. Що таке зовнішнє посилання?
- 8. Які недоліки модульного програмування в одному файлі?

# 4.7. Графічне оформлення результатів

Найбільш ефективним способом представлення інформації для сприйняття людиною є графічний спосіб. Графіки є найзручнішими для виявлення характерних точок функції, аналізу експериментальних та інших табличних даних тощо.

Пакет містить вбудовані засоби виведення графічної інформації у вигляді двовимірних графіків та тривимірних графіків.

В MathCAD можливо із статичних графіків зробити анімацію.

Графіки виводяться в спеціальну область графіків.

Створення області графіків в *MathCAD* здійснюється за допомогою комбінації клавіш, меню **Insert-Graph**, кнопковою панеллю графіки (рис. 4.58 а). В *MathCAD Prime* створення області графіків проводиться комбінацією клавіш та полями стрічки **Plots-Insert Plot** (рис. 4.58 б).



Рис. 4.58. Створення області графіків: а – MathCAD; б – MathCAD Prime

Розміри та місцеположення області графіків можуть бути динамічно змінені як описано в п. 4.2.

В MathCAD / MathCAD Prime з панелей можлива побудова:

- графіків функції **у=f(x)** в декартових координатах **X-Y Plot** (клавіші «@/Ctrl+2»);
- графіків функції ρ=f(φ) в полярних координатах Polar Plot (клавіші «Ctrl+7/Ctrl+4»);
- об'ємного 3D зображення функції z=f(x,y) у вигляді поверхні
   Surface Plot/ 3D Plot (клавіші «*Ctrl*+2»);
- діаграми ліній рівня функції z=f(x,y) Contour Plot (клавіші «*Ctrl+5*»). Діаграма є зображенням ізоліній ліній, в яких функція приймає фіксовані значення z=const.

В MathCAD додатково реалізовані:

- 3D точкове зображення матриці або функції z=f(x,y) по заданих точках (x<sub>i</sub> y<sub>j</sub>) 3D Scatter Plot. Застосовується для побудови просторових кривих;
- 3D зображення матриці або функції z = f(x,y) у вигляді стовпцевої діаграми 3D Bar Plot;
- зображення двовимірних векторних полів Vector Field Plot.

Створення, редагування та оформлення графіків в *MathCAD* проводиться через діалогові вікна контекстного меню графіків. В *MathCAD Prime* вказані дії проводяться спрощено безпосередньо на полі графіка або в панелі стрічки.

### 4.7.1. Двовимірні графіки

На двовимірних графіках зображують залежності функцій користувача, виразів одного аргументу, векторів. Пакет будує графіки функцій однієї змінної в декартових y = f(x) та полярних координатах  $\rho = f(\varphi)$ . де  $y = \rho \sin(\varphi), x = \rho \cos(\varphi)$ .

Функції можуть бути записані у вигляді залежностей від однієї спільної змінної в абсолютній формі y = f(x),  $\rho = f(\varphi)$  або в параметричній формі y = f(t),  $x = \varphi(t)$ .

## Графіки в декартових координатах

### Засоби MathCAD

Двовимірні декартові графіки в пакеті будуються в шаблоні декартових графіків. Створення шаблону в *MathCAD* проводиться комбінацією клавіш «*Shift*+@», полями меню **Insert- Graph-X-Y Plot**, кнопками панелі графіки **Graph** (рис. 4.58 а). В шаблоні декартових графіків можуть бути створені різноманітні зображення (рис. 4.59).



Рис. 4.59. Різновиди плоских графіків: а – графік на двох осях в логарифмічному масштабі з лініями сітки, підписами графіка та осей, легендою; б – графік двох функцій по двох аргументах з маркерами; в – графік похибок; г – точкова діаграма для вектору; д – лінійна діаграма для двох векторів; є – точкова діаграма

Шаблон графіка містить в центрі рамку для графіка та 6 маркерів навколо рамки (рис. 4.60). В маркери вводиться інформація для побудови графіків. Обов'язковим для заповнення є маркер 2. Маркери 4, 1, 3 визначають вісь абсцис графіка, маркери 6, 2, 5 визначають вісь ординат графіка.



Рис. 4.60. Шаблон двовимірної декартової графіки [1]

В маркер 1 вводиться ім'я ранжованої змінної-аргументу або вектора-аргументу графіка. Крайні маркери призначені для визначення границь графіків. В маркери 3,4 вводяться межі діапазону зміни аргументу. Введені значення повинні знаходитись всередині значень ранжованої змінноїаргументу або вектора-аргументу.

За замовчанням без заповнення маркера 1 двовимірний графік в декартових координатах будується для значень аргументу в діапазоні від -10 до 10 із кроком 1. Ім'я аргументу визначається пакетом автоматично з введеного в маркер 2 виразу або імені функції користувача з іменем аргументу в дужках.

В маркер 2 вводиться ім'я функції користувача або вираз, графіки яких планується намалювати. Для побудови графіків декількох функцій їхні імена вводяться в маркер 2 через кому. В такому разі можливо визначення кількох змінних-аргументів, кожного для своєї функції. Кілька аргументів записуються через кому (рис. 4.59 б).

В маркери 5,6 вводяться мінімальне та максимальне значення функції для відображення. Якщо маркери не заповнені, то значення визначаються пакетом автоматично.

За замовчанням графік зображується червоною суцільною лінією без сітки.

Якщо потрібний діапазон аргументу знаходиться всередині діапазону за замовчанням, можна побудувати графік за замовчанням, а потім уточнити потрібні значення аргументу графіків в маркерах 3, 4.

Для усунення необхідності додаткових дій та отримання зображення з потрібним кроком між сусідніми точками доцільно визначати діапазон змін аргументу через ранжовану змінну.

Для зображення графіків функцій на зразок замкнених геометричних фігур, які мають декілька значень функції для одного значення аргументу, слушно застосовувати параметричні графіки. На параметричному графіку на двох осях координат задаються дві функції, які залежать від одного аргументу.

Наприклад [1], в звичайному режимі функція  $y(x) = \sin(3\frac{\arcsin(x)}{2})$  зображується в декартових координатах тільки однією гілкою графіка (рис. 4.61 а). Якщо записати цю функцію в параметричному вигляді  $x = a(t) = \sin(2t) \ y = b(t) = \sin(3t)$ , то графік стає значно інформативнішим (рис. 4.61 б).



Рис. 4.61. Графіки: а – звичайний; б – параметричний [1]

Пакет дозволяє візуалізувати табличні (векторні) дані.

Для зображення вектору (рис. 4.59 г) ім'я вектору вводиться в маркер 2 імені функції, ім'я змінної індексу вектора вводиться в маркер 1 аргументу графіка. Діаграма векторів будується визначенням імен векторів в обидва маркери 1 та 2 шаблону (рис. 4.59 д). Один вектор слугує абсцисою графіка, другий вектор слугує ординатою графіка. Точкова діаграма (рис. 4.59 д) є різновидом діаграми векторів. В якості аргументів в точкової діаграми використовують два вектори. Один вектор містить координати х точок та слугує абсцисою графіка, другий вектор містить координати у точок слугує ординатою графіка.

Редагування плоских декартових графіків проводиться в діалоговому вікні форматування графіків (рис. 4.62). Викликати вікно редагування можна подвійним кліком миші в області графіка, пунктом **Format** контекстного меню графіків або пунктом меню **Format–Graph-X-Y Plot**.

Вікно містить п'ять вкладок X-Y Axes, Traces, Labels, Defaults, Number Format.

Для форматування осей графіків на вкладці **Х-У Ахез** (рис. 4.63 а) можна провести наступні дії:

- додати другу вісь «Y» для відображення двох графіків з різними значеннями ординат в полі Enable second Y axis. При малюванні кількох графіків перший малюється відносно першої осі, всі інші – відносно другої;
- задати логарифмічний масштаб осей незалежно по кожній з осей в полі Log Scale (рис. 4.59 а);
- задати видимість та колір допоміжних ліній сітки на полі графіка в полі **Grid Lines** (рис. 4.59 а);
- вивести на поле графіка одну або дві лінії з визначеним положенням та кольором в полі **Show markers**;
- керувати виведенням чисельних значень по лініях сітки всередині діапазону в полі **Numbered**;
- керувати режимом автоматичного масштабування графіка в полі **Autoscale**;
- керувати режимом автоматичного нанесення сітки в полі Autogrid;
- керувати виглядом осей в полі Axis Style. Припустимими типами осей є: Boxed прямокутник навколо графіка, Crossed осі у вигляді хреста в центрі, None графік без осей, Equal scale осі рівного масштабу.

Логарифмічний масштаб є корисним:

- в разі необхідності детального аналізу ділянок графіка, діапазон змін аргументу або значень функції на якому є дуже великим;
- в разі необхідності відображення на одному полі графіків, які мають дуже різні значення.

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

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

Результатом функції logspace (min, max, N) є вектор з N елементів, які розподілені за логарифмічним законом від min до max. Наприклад, виклик logspace (1,10,5) генерує вектор [1, 1.1778, 3.162, 5.623, 10].

Результатом функції logpts (minexp,dec,dN) є вектор з розраховує N=dec ·dN елементів, які лінійно розподілені від  $10^{\text{minexp}}$  на dec декад з dN інтервалами на декаду. Функція не розраховує останню точку декади. Наприклад, виклик logpts (-1,1,9) генерує вектор [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]. Перший елемент  $0.1=10^{-1}=10^{\text{minexp}}$ , дев'ять елементів займають одну декаду від 0.1 до 1.

Виклик **logpts** (-1,1,10) генерує не вектор [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0], а вектор [0.1, 0.19, 0.28, 0.37, 0.46, 0.55, 0.64, 0.73, 0.82, 0.91].

На вкладці (рис. 4.63 в) Labels можна ввести титульний напис до графіка (рис. 4.59 а) в полі Title, задати підпис осі X (рис. 4.59 а) в полі X-Axis, задати підпис осі Y (рис. 4.59 а) в полі Y-Axis.

X-Y Axes Traces Number For	mat Labels Defaults		X-Y Axes Tr	aces Nu	umber For	mat Lab	els Del	faults				
Enable secondary Y axis  X-Axis  Grid Ines Grid Ines Auto scale Show markers Auto scale Number of grids:	Primary Y Axis     Secondary Y Axis       Log scale     Grid lines       VNumbered     Auto scale       Show markers     Auto grid       Auto grid     Xumber of grids:     2		egend labe trace 1 trace 2 trace 3 trace 4 trace 5 trace 6 trace 7 trace 8	Symbol requenc 1 1 1 1 1 1 1 1 1 1 1 1	Symbol +	Symbol Weight 1 1 1 1 1 1 1 1	Line	Line Veigł 1 1 1 1 1 1 1 1	Color	Type points lines lines lines lines lines lines lines	Y-axis	
Boxed     Crossed     None	Equal scales	б		Juments			iop-left lottom-le lelow	eft		○ Top-riq ○ Botton	ght n-right	
	Formatting Curren	tly Selected X-Y Plo	ot				×					
	X-Y Axes Traces	Number Format	Labels Defaults	;								
	Above	◯ Below	Show 2	Title								

Рис. 4.63. Вікно форматування: а – осей; б – графіків; в – написів

На вкладці (рис. 4.63 б) **Traces** визначається вигляд графіка [30], а саме:

- текст опису ліній графіка в полі Legend (рис. 4.59 а);

- символ, який буде зображено на лінії графіка: хрест, коло, квадрат, в полі Symbol (рис. 4.59 а, г, є);
- кількість точок графіка між символами в полі Symbol Frequency;
- товщину символів в полі Symbol Weight;
- візерунок лінії: суцільна, точкова, штрихова тощо, в полі Line;
- колір лінії графіка в полі **Color**;
- тип графіка в полі Туре. Припустимими є: звичайна лінія (line) (рис. 4.59 а), стовпцева діаграма (bar), лінійна діаграма (stem) (рис. 4.59 д), лінія з точок (points) (рис. 4.59 є), лінія сходів (step). графік без продовження до осей (draw). Для двох ліній можна застосувати тип «похибка» (error) (рис. 4.59 в). При цьому на графіку буде заштриховано область, що окреслена лініями графіка.
- товщину лінії графіка в полі Weight.

### Засоби MathCAD Prime

Графіки в декартових координатах в пакеті будуються в шаблоні декартових графіків. Створення шаблону проводиться комбінацією клавіш «*Ctrl+2*» або кнопкою **Plots-Traces-Insert Plots-XY Plot** стрічки.

Шаблон містить осі для графіка та 2 маркери (рис. 4.64). В маркери вводиться інформація для побудови графіків.



Рис. 4.64. Поле графіка в декартових координатах: 1 – маркер **X ахі** аргументів, 2 – крок міток осі X, 3 – максимальне значення по осі X, 4 – маркер **Y ахі** функції, 5 – одиниці вимірювання, 6 – максимальне значення по осі Y, 7 – крок міток по осі Y [1]

В маркер 1 вводиться ім'я ранжованої змінної-аргументу або вектора-аргументу графіка. В маркер 4 вводиться ім'я функції користувача або вираз, графіки яких планується намалювати. Для побудови графіків декількох функцій слід



використати кнопку Trace Plots-Traces-Add Trace стрічки.

За замовчанням двовимірний графік в декартових координатах будується суцільною синьою лінією для 500 точок в діапазоні аргументу від -10 до 10. Ім'я аргументу пакет автоматично не визначає.

Редагування діапазонів зображення графіків та кроку міток на осях проводиться безпосередньою зміною значень на осях. Наведення курсору на крайні та передостанні значення на осях супроводжується виділенням значень та появою вікна пояснень (рис. 4.65). Крайні значення (п. 3, 6 рис. 4.64) відповідають мінімальним та максимальним значенням діапазону, передостанні значення (п. 3, 7 рис. 4.64) відповідають положенню міток на осі. Значення змінюються редагуванням наявних.



Рис. 4.65. Редагування осей

Клік мишею на маркерах 1, 4 розкриває маркер та додає поле 5 для введення одиниці вимірювання.

Розташування маркерів аргументів 1, функцій 4 та осей може бути змінено простим перетягуванням в потрібне місце.

На кінцях осей, протилежних маркерам 3, 6 знаходяться маркери мінімальних значень відповідних осей.

Редагування ліній декартових графіків проводиться засобами вкладки **Plots** стрічки (рис. 4.66).



Рис. 4.66. Стрічка керування графіками

Для вже намальованого графіка полями панелі **Styles** стрічки можна визначити:

- символ, який буде зображено на лінії графіка в полі Symbol;

- візерунок лінії в полі Line Style;
- колір лінії графіка в полі **Trace Color**;
- товщину лінії графіка в полі WTrace Thickness.

Обрання будь-якого з маркерів осей активує панель **Axes** стрічки. З панелі **Axes** можна:

- показати/сховати мітки осі кнопкою **Tick Marks**;
- показати/сховати числа біля маркерів осі кнопкою Tick Marks Values;
- показати/сховати самі маркери осей кнопкою Axis Expression;
- задати логарифмічний масштаб по осі кнопкою Logarithmic Scaling.

Тип плоского графіка визначається списком (рис. 4.67 а) Plots-Traces-Change Type. Пакет дозволяє малювати графіки у вигляді ліній (Line Trace), стовпцевих діаграм (Bar Trace), горизонтальних рядкових діаграм (Column Trace), постовпцевих графіків залежності векторівстовпців матриці Y від значень векторів-стовпців матриці X (Waterfall) (рис. 4.67 б), графіків відрізків, вертикальні координати яких беруться з рядків ДВОХСТОВПЩЕВОЇ матриці Y (Error) (рис. 4.67 в), графіків статистичної обробки експериментальних даних (Effect, Box) (рис. 4.67 г, 4.67 д).

### Графіки в полярних координатах

Особливістю полярних графіків є те, що аргумент функції повинен задаватися в радіанах, а на графіку він відображається в градусах.

### Засоби MathCAD

Полярні графіки в пакеті будуються в шаблоні полярних графіків. Створення шаблону проводиться комбінацією клавіш «*Ctrl*+7», полями меню Insert-Graph-Polar Plot, кнопкою Polar Plot панелі графіки Graph.

Шаблон містить в центрі контурне коло для графіка та 4 маркери навколо рамки (рис. 4.68 а). В маркери вводиться інформація для побудови графіків. Обов'язковим для заповнення є маркер 2.



Рис. 4.67. Типи плоских графіків: а – список типів графіків; б-графік Waterfall; в-графік Error; г-графік Effect; д-графік Box



Рис. 4.68. Графіка в полярних координатах: а – шаблон; б – приклад В маркер 1 вводиться ім'я ранжованої змінної-аргументу або вектора-аргументу графіка.

За замовчанням без заповнення маркера 1 полярний графік будується для значень аргументу в діапазоні від 0 до 2*π*. Ім'я аргументу визначається пакетом автоматично з введеного в маркер 2 виразу або імені функції користувача з іменем аргументу в дужках. В маркер 2 вводиться ім'я функції користувача або вираз, графіки яких планується намалювати. Для побудови графіків декількох функцій їхні імена вводяться в маркер 2 через кому.

В маркери 3,4 вводяться мінімальне та максимальне значення функції для відображення. Якщо маркери не заповнені, то значення визначаються пакетом автоматично.

За замовчанням графік зображується червоною суцільною лінією без сітки (рис. 4.68 б).

Форматування графіка в полярних координатах проводиться аналогічно форматуванню графіка в декартових координатах. Параметри графіка редагуються у діалоговому вікні полярних графіків.

Викликати вікно редагування можна подвійним кліком миші в області графіка, пунктом Format контекстного меню графіків або пунктом меню Format-Graph-Polar Plot.

Bikho полярних графіків містить вкладники Polar Axes, Traces, Labels, Defaults, Number Format. Зміст дій та поля вкладинок в цілому співпадають з полями вікна декартових графіків.

Поле **Show markers** виводить лінії-маркери за заданими користувачем радіусами.

Для спрощення розрахунків можна застосовувати функції переведення декартових координат в полярні та навпаки: **xy2pol(<apr>)** та **pol2xy(<apr>)**. Аргументом функцій можуть бути два числа: координати х,у/радіус, кут, - або двоелементний вектор. Перший елемент вектора містить значення х/радіуса, другий — у/кута в радіанах. Результатом дії функцій є вектор полярних координат/декартових координат. Перший елемент вектора результату містить значення радіуса/х, другий — кута в радіанах/у.

Наприклад, звернення до функції **ху2ро1** з аргументами "1, 1" або

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

виведе результат

## Засоби MathCAD Prime

Полярні графіки в пакеті будуються в шаблоні полярних графіків. Створення шаблону проводиться комбінацією клавіш «*Ctrl+4*» або кнопкою **Plots-Insert Plot- Polar Plot** стрічки.

Шаблон містить контурне коло для графіка та два маркери 1, 3 (рис. 4.69). В маркери вводиться інформація для побудови графіків.



Рис. 4.69. Графік в полярних координатах

В маркер 3 Angular Axes вводиться ім'я ранжованої змінноїаргументу або вектора-аргументу графіка.

За замовчанням полярний графік будується суцільною синьою лінією для значень аргументу в діапазоні від 0 до 2*π*.

В маркер 1 Radial Axes вводиться ім'я функції користувача або вираз, графік якої планується намалювати.

При наведенні миші на маркери 1, 3 відкривається поле 2 для одиниць вимірювання.

Для додавання графіків на поле використовується кнопка Read Plots-Traces-Add Trace стрічки.

Редагування діапазонів зображення графіків та кроку міток на осях проводиться безпосередньою зміною значень в полях 4, 5 на осі аналогічно декартовим графікам.

Редагування ліній полярних графіків проводиться засобами закладинки **Plots** стрічки (рис. 4.66).

Наприклад, для того щоб відобразити графік виразу **cos(θ)+1** в полярних координатах, слід в потрібному місці вставити графічну область

полярного графіка. В маркер **Radial axis** 1 ввести розрахунковий вираз, в маркер **Angular axis** ввести ім'я аргументу θ, обрати червоний колір в полі **Plots-Style-Trace Color** стрічки. Графік набуде вигляд, зображений на рис. 4.69. Діапазон аргументу 0...2π система визначила автоматично.

# Додаткові графічні можливості MathCAD

Пакет дозволяє змінювати область зображення графіків без пере визначення діапазону аргументів в режимі збільшення **Zoom**. Область збільшення задається її координатами в полях х, у діалогового вікна режиму збільшення або простим виділенням прямокутної області на полі графіка мишею з натиснутою лівою кнопкою. Після натискання кнопки «*OK*» у вікні збільшення графік автоматично перемальовується в новому масштабі. Режим активується кнопкою **Zoom** панелі **Graph** (рис. 4.58 а) та полем **Zoom** контекстного меню контекстного меню області графіків (рис. 4.70 а).

Пакет дозволяє отримання кількісні дані для точок графіків в режимі аналізу **Trace**. В режимі аналізу на полі графіка зображуються дві пунктирні лінії у вигляді хреста. Точка їх перетину відповідає положенню миші (рис. 4.70 в). При русі миші по полю графіка у вікні трасування відображаються її координати (рис. 4.70 б). Перемикач **Track data points** вмикає режим, в якому миша може знаходитися тільки на зображеному графіку.

Режим активується кнопкою **Trace** панелі **Graph** (рис. 4.58 a) та полем **Trace** контекстного меню контекстного меню області графіків (рис. 4.70 a).



Рис. 4.70. Режими аналізу двовимірних графіків: а – контекстне меню; б – вікно аналізу; в – поле графіка в режимі аналізу

В *Mathcad* передбачена можливість створення *анімації* графіків у вигляді AVI-файлів. Анімація створюється покадровим записом. Збережений анімаційний кліп є стандартним відеофайлом. Переглянути його можна будьяким відео програвачем. Для ідентифікації номерів кадрів використовується вбудована системна змінна **FRAME** цілого типу. Змінна **FRAME** має буди

включена у формульні вирази таким чином, щоб при її зміні змінювався б графік. Діапазон значень **FRAME** задається при створенні анімації в діалоговому вікні меню **Tools-Animate-Record**.

Анімування здійснюється в два етапи: запис кліпу та програвання анімації.

Для запису кліпу слід виконати наступні дії:

 записати вираз функції графіка з використанням змінної **FRAME** та побудувати графік (рис. 4.71). Рекомендується вимкнути автоматичне масштабування графіка та визначити чисельні значення на осях графіка вручну.



Рис. 4.71. Заготовка анімації

- Відкрити вікно анімації з меню **Tools-Animation-Record**. Після відкриття діалогового вікна анімації (рис. 4.72) мишею рамкою обвести графік.



Рис. 4.72 – Вікно анімації

 У вікні анімації визначити значення початкове From та кінцеве To значення змінної FRAME (кадри фільму) та швидкість запису At (рекомендується швидкість 2-3 кадри в секунду). Після визначення значень натиснути кнопку Animate.
- Готовий кліп можна зберегти кнопкою Save As. Кнопка Options... вікна анімації відкриває доступ до налаштувань запису (кодек, опорні кадри тощо).
- В разі успішного створення кліпу відкривається вікно вбудованого в *MathCAD* програвача (рис. 4.73). Для програвача можна налаштувати розмір **Вид**, швидкість програвання **Скорость**.

Використати збережений кліп в *MathCAD* можна прямим програванням вбудованим плеєром або вставлянням кліпу в документ. Можливості некоректно реалізовані для Windows 10.

Плеєр активується з меню **Tools-Animation-Record**.



Рис. 4.73. Вбудований програвач

Кліп вставляється в документ з меню Insert-Object (Создать из файла) (рис. 4.74).

Вставка объекта		×
<ul> <li>Создать новый</li> <li>Создать из файла</li> </ul>	Файл: D:\Program Files (x86)\Mathcad\Mathca	ОК Отмена
	Обзор Связь	🗌 В виде значка

Рис. 4.74. Вставляння кліпу в документ

Кнопка **Связь** дозволяє зберегти зв'язок з зовнішнім дисковим файлом. Його зміни автоматично повторюються у відеопрогравачі *Mathcad*.

Встановлення прапору **В виде значка** виводить в документ позначку AVI-файлу. Подвійний клік на позначці запускає відеопрогравач з обраним файлом.

Без прапору **В виде значка** в документ виводиться зображення першого кадру кліпу. Подвійний клік на зображення запускає відеопрогравач з обраним файлом.

# 4.7.2. Тривимірні графіки

Тривимірна графіка в системі базується на вбудованих матричних перетвореннях. 3D графіки є зображенням елементів матриці. Координатами по осях X, Y виступають індекси елементів матриці, по осі Z відображаються значення елементів матриці з відповідними індексами.

# Засоби MathCAD

Система спроможна відображати поверхневі графіки **Surface Plot** (рис.4.75 а), контурні графіки **Contour Plot** (рис.4.75 б), 3D точкові графіки **Data Points/Scatter Plot** (рис.4.75 в), графіки векторного поля **VectorField Plot**, 3D діаграми **Bar Plot** (рис.4.75 г), діаграми рівнів **Patch Plot** (рис.4.75 д).



Рис. 4.75. Типи тривимірних графіків: а – поверхні; б – контурний; в – точковий; г – діаграма; д – рівнів

Створення графічної області 3D графіків здійснюється клавішами «*CTRL*+2», відповідними пунктами меню **Insert-Graph** (рис.4.9) або відповідних кнопок панелі **Graph** (рис. 4.59 а).

Результатом є відображення на екрані шаблону 3D графіки з єдиним маркером (рис. 4. 76)



Рис. 4. 76. Шаблон тривимірних графіків [30]

В маркер вводиться ім'я об'єкта, графік якого передбачається побудувати. Об'єктом може бути матриця або наперед визначена функція користувача від двох змінних. Функція вводиться самим ім'ям без імені аргументу в дужках.

Наприклад, на рис. 4.77 зображені два ідентичні графіки. Один графік (рис. 4.77 а) побудований прямим введенням імені функції **ff** в маркер поля графіки, інший графік (рис. 4.77 б) побудований введенням імені матриці **h**.



Рис. 4.77. Тривимірний графік: а – функції користувача; б – матриці

Для побудови кількох графіків імена вводяться через кому. Не обов'язково, щоб імена відносилися до об'єктів одного типу. Можливо використання одного імені функції, іншого – матриці.

Безпосередньо на графічному полі можна повернути зображення та змінити його масштаб. Рухи миші з натиснутою лівою кнопкою повертають зображення. Рухи миші з натиснутими лівою кнопкою та натиснутою клавішею «*Ctrl*» або поворот колеса миші змінює масштаб зображення.

За замовчанням аргументи функції користувача вважаються заданими в діапазоні від -5 до 5 в 20-ти точках.

Явне визначення аргументів функції користувача вектором або ранжованою змінною не змінює діапазон аргументів графіка.

Коригування діапазонів графіків, які створені без програмного готування вбудованими функціями, можливо тільки в діалоговому вікні налаштувань 3D графіка.

Введення замість імені функції користувача розрахункового виразу не дозволяється.

# Налаштування тривимірних графіків

За замовчанням рідко вдається отримати потрібний вигляд тривимірного графіка. Графіки поверхонь зображуються чорно-білими у вигляді дротової моделі, контурні графіки зображуються чорно-білими без чисельних значень та виділення зон ізоліній (рис. 4.78).



Рис. 4.78. Зображення тривимірних графіків за замовчанням

Налаштування вигляду тривимірних графіків проводить в діалоговому вікні **3D Plot Format**. Вікно викликається подвійним кліком миші на полі графіка або з меню **Format–Graph** (рис. 4.8).

Bikho містить вкладки для визначення точки зору, виду графіка General; для визначення вигляду координатних осей Axis; для визначення кольорів, вигляду точок графіка Appearance; для визначення додаткового освітлення Lighting; для визначення написів на осях та полі графіка Title; для визначення сітки, кольору фонових площин Backplanes; для визначення типу ліній, контурних ліній, стовпових діаграм Special; для установки прозорості, «задимлення», якості друку Advanced; для визначення діапазонів аргументів, системи координат QuickPlotData.

Вкладка General (рис. 4.79) містить групи для визначення орієнтації зображення View, визначення типу осей Axes Style, визначення типу рамки навколо графіка Frames, визначення типу графіка Plot.

General Axes	Appearance	Lighting	Title	Backplanes	Special	Advanced	QuickPlot Data
View Rotation: Tilt: Twist: Zoom:	35 ÷ -10 ÷		s Style rimeter orner one ual Scale	25	Fra	ames Show Border Show Box	
Plot 1							
Surface Plot		OData	Points		C	) Bar Plot	
O Contour Plot		○ Vect	or Field F	Plot	C	) Patch Plot	

Рис. 4.79. Вкладка General вікна налаштувань

Поле Rotation групи View задає кут повороту осей XY горизонтальної площини XOY відносно вертикальної осі Z в градусах в діапазоні від -360° до 360°. Значення 0° відповідає YOZ площині, 90° – площині XOZ.

Поле **Tilt** групи **View** задає *кут підвищення* осі зору спостерігача відносно горизонтальної площини ХОУ в діапазоні від 0° до 180°. Значення 0° відповідає горизонтальній проекції площини (вид збоку), 90<sup>0</sup> – вертикальній проекції (вид згори).

Поле **Twist** групи **View** задає *кут кручення* відносин вертикальної осі Z в діапазоні від -360° до 360°.

Поле **Zoom** групи **View** задає масштаб в діапазоні від 0.0001 до 10000.

Радіокнопки групи **Axes Style** визначають розташування осей поля тривимірних графіків та визначення осей рівного масштабу **Equal Scales**. Припустимими є значення розташування по периметру **Perimeter**, в куті **Corner**, без осей **None**.

Перемикачі групи **Frames** керують видимістю та кольором рамки навколо поля графіка **Show Border** та власне зображення графіка **Show Box**.

Типи можливих для малювання графіків (рис.4.75) визначають радіокнопки групи **Plot**.

Вкладка **Axis** (рис. 4.80) визначає вигляд осей X, Y, Z. Дія її елементів  $\epsilon$  аналогічною дії вкладинці двовимірних графіків. З вкладники проводиться керування видимістю сітки **Draw Lines**, кількістю **Number**, шириною **Line Weight**, кольором **Line Color** ліній сітки, маркерами **Draw Ticks**, видимістю чисельних значень **Show Numbers**, кольором **Axis Color**, товщиною **Axis Weight**, підписом **Label** осей, максимальним **Maximum Value** по осях.

X-Axis Y-Axis Z-Axis		
Grids Draw Lines	Label	
Draw Ticks Auto Grid Line Color	Axis Format	Axis Limits
4   +     1   +     Line Weight	Axis Weight	-1 Minimum Value 1 Maximum Value

Рис. 4.80. Вкладка Ахез

Елементи закладинки **Appearance** (рис. 4.81) містить групи, які визначають спосіб фарбування графіка **Fill options**, вигляд контурних ліній графіка **Lines options**, вигляд додаткових позначок на контурних лініях графіка **Points Options**, кольорову палітру **Color options**.



Рис. 4.81. Вкладка Арреагалсе

Припустимими режимами фарбування графіки є заповнення графіка кольором за схемою, яка обрана в групі Color Options, Fill Surface; заповнення кольором у напрямку, який визначений на вкладинці Special, Fill Contours; без фарбування No Fill. Заповнення може проводитися кольоровими трикутниками Alternate Mesh та покращеним Smooth Shading.

Зображення графіка може бути у вигляді прозорого каркасу Wireframe, у вигляді горизональних ліній рівня Contour, у вигляді тіла без контурних ліній No Lines. Лінії, які знаходяться на задній поверхні графіка можуть бути сховані Hide Lines.

На контурних лініях графіка можуть бути нанесені позначки Draw **Points** визначеного вигляду **Symbol** та визначеного розміру **Size**.

Фарбування графіка може проводитися обраним суцільним кольором **Solid Color** або градієнтно **Colormap** з параметрами, які визначаються на вкладинці **Advanced**.

Із закладинки **Backplanes** (рис. 4.82 а) визначаються такі параметри фонових площин графіка. як фарбування **Fill Backplane** та колір **Color** фарбування, наявність ліній **Draw Lines**, міток **Draw Ticks**, ширини ліній **Line Width** основної сітки **Grid** та мілкої сітки **Sub-Grid**, наявність рамки **Backplane Border** навколо фонової площини.

Використання форматованої фонової площини (рис. 4.82 б) додає наочності та покращує сприйняття графічної інформації.

X-Y Backplane	Y-Z Backplane	X-Z Backplane			Δ
Fill Backplane	2	C <u>o</u> lor	Backplane Border		50-1
	Grids		S	ub-Grids	40-
X-Axis	S	Y-Axis	X-Axis	Y-Axis	30-
Draw Lines	Dr	aw Lines	Draw Lines	Draw Lines	
Draw Ticks	Dr	aw Ticks	Draw Ticks	Draw Ticks	
🔲 Li <u>n</u> e Color		li <u>n</u> e Color	🔲 Line Go[or	🔲 Line Color	
Line Weight	Line V	Veight	Line Weight	Line Weight	* - <u>4</u>
1 ‡	1		0.5 🗘	0.5 ‡	2 4

Рис. 4.82. Фонові площини: а – визначення; б – використання Закладинка Special (рис. 4.83) містить поле Line Style для визначення суцільного solid, штрихового dashed, точкового dotted, штрихово-точкового da-dot відображення контурних ліній графіка та групи для керування параметрами контурних графіків Contour Options, 3D діаграм Bar Plot Layout, 3D точкових діаграм Mesh, Connectivity.

Contour Options Fill Draw Lines Auto Contour Numbered 15 ‡ Number	Bar Plot Layout Matrix Stacked Side by Side Spacing	Mesh Rows 21 + Columns	Connectivity <ul> <li>Row Order</li> <li>Increasing X</li> <li>Increasing Y</li> <li>Increasing Z</li> <li>Line Style</li> </ul>
Z-Contours V	20 🚔	21 🚔	solid 🗸 🗸

Рис. 4.83. Вкладка Special

В групі Contour Options проводиться керування фарбуванням Fill, нанесенням ізоліній Draw Lines, виведенням чисельних значень на ізолінії Numbered, кількістю ізоліній Number або їх автоматичним нанесенням Auto Contour контурних графіків.

В групі **Bar Plot Layout** проводиться визначення звичайного матричного **Matrix**, вертикального **Stacked**, в один рядок **Side by Side** розташування елементів та відстані **Spacing** між ними для 3D діаграм.

В групі **Mesh** проводиться визначення кількості рядків **Rows** та стовпців **Columns**, які будуть брати участь в інтерполяції для створення графіка точкової діаграми.

В групі Connectivity визначається незмінний, що відтворює порядок вхідних даних Row Order, зростаючий по X Increasing X, зростаючий по Y Increasing Y, зростаючий по Z Increasing Z спосіб проведення ліній між точками точкової діаграми. Група активується в разі обрання опції Lines на вкладці Appearance.

Вкладка Advanced (рис. 4.84) містить групи для керування оптичними

ефектами Advanced View Options, якістю друку графіків Printing, кольоровими палітрами Colormap.

Gene	ral	Axes	Appearance	e Lighting	Title	Backplanes	Special	Advanced	QuickPlot Data
Adv	ance nable erspi	ed View e Fog ective	Options 100 18	Vertical Sc Viewing Di	ale stance		Printin High	g I Quality Prin as default	ting
Plot	1						Colorm	nap easing X	
4	-	Shinir	ness				Incr	easing Y	
0	÷	% Tr	% Transparency						
1 Polygon Offset					Rainbow ~				

Рис. 4.84. Закладинка Advanced

Група View Options містить поля для керування режимом туману Enable Fog, режимом перспективної проекції Perspective на визначеній відстані Viewing Distance, масштабування графіка в вертикальному напрямку Vertical Scale, віддзеркалюванням від поверхні графіка Shininess, прозорістю графіка Transparency, відступом зафарбованої ділянки від лінії сітки Polygon Offset.

Група Colormap керує напрямом градієнтного фарбування та його типом [30]: веселка rainbow, відтінки сірого grayscale, відтінки синього blue, відтінки зеленого green, відтінки червоного red, обернена веселка fire, освітлені відтінки сірого gamma, жовто-синій neon, жовтобузковий royal, жовто-зелено-синій topograf.

В групі **Printing** визначається висока **High Quality Printing** якість друку графіків.

Поля start, end, of Grids закладинки QuickPlot (рис. 4.85) дозволяють визначити діапазон аргументів, відмінний від значень за замовчанням для малювання графіка функції без застосування функції CreateMesh. Радіокнопки Coordinate System визначають декартову Cartesian, сферичну Spherical, циліндричну Cylindrical систему координат графіка.

Plot 1				
Range 1		Range 2		Coordinate System
start	-5	start	-5	Cartesian
end	5	end	5	O Spherical
# of Grids	20 🛓	# of Grids	20 🔺	○ Cylindrical

Рис. 4.85. Закладинка QuickPlot

#### Поверхневі графіки

Для програмного готування даних для поверхневих графіків призначена функція

# [X,Y,Z]:=CreateMesh(fun<,[x0,x1,y0,y1],... [xN,yN],[fmap]>)

Обов'язковим ім'я аргументом E наперед визначеної fun. Функція-користувача функції-користувача використання для в CreateMesh може записана в звичайному виглялі бути або В параметричному вигляді через вектор-стовпець з трьох елементів, які містять параметричні залежності по трьох координатах. Наприклад,

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

Звичайною формою описуються прості незамкнені поверхні. Для замкнених геометричних фігур використовується параметрична форма запису.

Опціональні аргументи визначають додаткові параметри, які відрізняються від значень за замовчанням. Вектор [x0,x1,y0,y1]визначає діапазон зміни аргументів функції: від x0 до x1 для першої координати, від y0 до y1 - для другої. Зміною початку та кінця діапазону<math>[x0,x1,y0,y1] можна створювати графіки поверхонь із вирізом. Елементи вектора [xN,yN] визначають кількість точок в інтервалах.

Результатом функції є вектор з трьох матриць **х, у, z**, які містять координати **х, у, z** відповідно.

Опційний параметр **fmap** задається в разі визначення функції не в декартових координатах. Параметр може визначатися у вигляді вектора-стовпця із значеннями трьох координат або у вигляді імені функції перерахунку координат.

Наприклад, виклик функції для побудови тривимірного графіка функції *Sinc* для діапазонів аргументів від **-1** до **1** та кількості точок 30 по кожній координаті має наступний вигляд:

$$\underline{F}(x,y) := \frac{\sin(x^2 + y^2)}{x^2 + y^2} \quad B := CreateMesh(F, -1, 1, -1, 1, 30, 30)$$

При введенні в маркер тривимірної графіки імені функції користувача пакет сховано викликає функцію **CreateMesh** та перетворює значення функції в матрицю.

Для побудови тривимірного графіка для функції, яку задано в параметричному вигляді, слід ввести *в дужках* імена трьох функцій/матриць опису через кому або ім'я вектора-стовпця, елементуми якого є три параметричні функції.

Пакет дозволяє проводити опис функцій в декартовій Cartesiam, сферичні Spherical та циліндричній Cylindrical системі координат.

В сферичних координатах поверхня визначається радіусом  $\mathbf{r}$  від точки початку системи координат до точки поверхні, кутом місця  $\boldsymbol{\theta}$ , який вимірюється від площини ХОУ до точки на поверхні та може змінюється від - $\pi$  до  $\pi$ , та кутом азимута  $\boldsymbol{\varphi}$ , який вимірюється в площині ХОУ та може змінюватися змінюється від 0 до  $2\pi$ .

В циліндричних координатах поверхня визначається радіусом  $\mathbf{r}$ , від точки z на осі OZ до точки поверхні в площині XzУ, кутом азимута  $\boldsymbol{\varphi}$ , який вимірюється в площині XzУ та може змінюватися змінюється від 0 до  $2\pi$ , та вертикальною координатою від площі XOУ до точки поверхні.

Для полегшення перерахунку значень координат з однієї системи в іншу призначені наступні вбудовані функції [1; 30]:

- **xyz2cyl (x,y,z)** для перерахунку декартових координат в циліндричні;
- суl2хуг(r, φ, z) для перерахунку циліндричних координат в декартові;
- **xyz2sph(x,y,z)** для перерахунок декартових координат в сферичні;
- sph2xyz (x, φ, λ) для перерахунок сферичних координат в декартові.

В функціях перерахунку координат кутові координати φ, λ задаються в радіанах. Для введення кутів в градусах слід застосувати одиницю вимірювання **deg** праворуч від числа.

Результатом дії функцій перерахунку координат є вектор-стовпець, який містить координати у відповідній системі координат.

Наприклад, для точки Р з декартовими координатами **x=1**, **y=2**, **z=3** в циліндричні координати можна перерахувати наступним чином:

x := 1 y := 2 z := 3 p := 
$$(1 \ 2 \ 3)^{T}$$
  
xyz2cyl(x,y,z) =  $\begin{pmatrix} 2.236\\ 1.107\\ 3 \end{pmatrix}$  xyz2cyl(p) =  $\begin{pmatrix} 2.236\\ 1.107\\ 3 \end{pmatrix}$ 

Розглянемо декілька способів готування даних та побудови графіків сферичної поверхні з центром в точці початку координат та радіусом **r**. Описати сферу звичайною формою неможливо, тому застосовується параметричний опис. В декартових координатах він має наступний вигляд:

$$\begin{split} X(u,v) &:= r \cdot \sin(u) \cdot \cos(v) \quad Z(u,v) := r \cdot \cos(u) \\ Y(u,v) &:= r \cdot \sin(u) \cdot \sin(v) \end{split}$$

Опис сфери в сферичних координатах суттєво спрощується. В сферичних координатах параметрично сфера визначається радіусом **r**, кутом місця  $\theta$  та кутом азимута  $\varphi$ :  $\theta$  (**a**, **b**) := **b**  $\varphi$  (**a**, **b**) := **a R**(**a**, **b**) := **r**.

Графіки за замовчанням сфери через параметризовані функції-вектор **h (u, v)** в декартових координатах та сферичних координатах введенням імен в маркер тривимірної графіки показані на рис. 4.86. Тип системи координат задається на вкладці **Axes-Coordinate System** вікна налаштувань.

Для малювання без функції **CreateMesh** діапазон значень аргументів та кількість точок графіка можна змінити ТІЛЬКИ у вікні налаштувань.



Рис. 4.86. Параметризоване зображення за замовчанням: а – декартові координати; б – сферичні координати

Особливістю цього способу можна вважати те, що програмно змінити параметри графіка формульними залежностями, неможливо.

Графік сфери в декартових координатах з використанням функції **CreateMesh** для діапазону від -3 до 3 та 40 точок показано на рис. 4.87 а.

Графік сфери в сферичних координатах з використанням функції **CreateMesh** для діапазону кута місця від 0 до  $\pi$ , діапазону кута азимута від 0 до  $2\pi$  по 20 точках наведений на рис. 4.87 б.

За такого визначення даних вікно редагування стає недоступним (рис. 4.87 а).

Слід взяти до уваги, що функція **CreateMesh** в сферичних координатах потребує зміненого порядку запису імен параметричних функцій (визначення елементів функції-вектора). Першим елементом є радіус, другим – кут місця **θ**, третім – азимут **φ**.



Рис. 4.87. Зображення сфери функцією **CreateMesh:** а – декартові координати; б – сферичні координати

Застосування функції **CreateMesh** дозволяє програмно визначити діапазони аргументів, кількість точок та тип системи координат.

# Контурні графіки

Контурний графік є ортогональною проекцією поверхні уздовж осі Z на площину XOY.

Таку проекцію можна отримати зміною напряму зору на поверхню. Для цього слід у вікні налаштувань визначити вид згори General-Rotation 0, General-Tilt 90, General-Twist 0. На графік можна вивести контурні лінії, які відповідають визначеним значенням функції — ізолінії. Для цього слід у вікні налаштувань визначити Special-Contour Options-Draw Lines (рис. 4.88 а). Вбудований контурний графік створюється з панелі **Graph** або меню **Insert** обранням типу **Contour Plot** в один клік. Додатково для кількісного аналізу функції на контурному графіку на зображення можна пунктом **Special-Contour Options-Numbered** вікна налаштувань нанести на контурні лінії чисельні значення функції (рис. 4.88 б).

Комплексні графіки, на яких зображується один і той же об'єкт у вигляді поверхні та у вигляді контурного графіка суттєво підвищують інформативність зображення. В такому разі ім'я функції в маркер графічного поля вводиться двічі (рис. 4.88 в).



Рис. 4.88. Контурний графік: а – поверхні; б – вбудований одиночний; б - комплексний

## Точкові графіки

Точкові графіки є різновидом поверхневих графіків, в яких зображуються тільки точки значень функції у вузлах сітки. Грані полігонів та їхні ребра не малюються та не зафарбовуються.

Таке зображення можна отримати налаштуванням зображення поверхні. вікні налаштувань прибрати малювання ребер Для цього слід у Appearance-Line Options-No Lines фарбування граней та Options-No Fill. Вмикнути зображення точок Appearance-Fill Appearance-Point Options-Draw Points (рис. 4.89 a).

Вбудований контурний графік створюється з панелі **Graph** або меню **Insert** обранням типу **3D Scatter Plot** в один клік. Шаблон графіка векторного поля є стандартним (рис. 4.75) та містить один маркер.



Рис. 4.89. Точковий графік: а – поверхні; б – вбудований

Саме через точкові діаграми **3D Scatter Plot** відображаються графіки просторових кривих.

Умовою створення графіка просторової кривої є наявність її математичного опису в параметричному вигляді.

Крива може бути задана у вигляді трьома параметричними функціями користувача з визначенням аргументів **x**, **y**, **z**, трьома векторами явно. Три функції користувача можуть бути замінені на одну функцію користувача. Функція повинна бути вектором-стовпцем, три елементи якого містять параметричні вирази визначення аргументів **x**, **y**, **z**.

В маркер шаблону записуються параметрично три імені функцій користувача або імена трьох векторів, або ім'я функції-вектора користувача.

За замовчанням аргумент параметричних функцій користувача вважаються заданими в діапазоні від -5 до 5 в 20-ти точках.

Незалежно від способу визначення опису кривої графік потребує форматування. За замовчанням точковий графік зображується у вигляді точок (рис. 4.90 а). Для надання йому звичного вигляду слід вмикнути відображення ліній між точками графіка пунктом **Appearance-Line options-Lines** вікна налаштувань. За потреби в групі на вкладниці **Appearance-Point options** вікна налаштувань можна визначити опорні точки та позначити їх розміром та кольором (рис. 4.90 б).



Рис. 4.90. Графік кривої: а – первинний; б – форматований

Наприклад, гіперболічна спіраль з радіусом **r** параметрично описується наступними виразами: **x=r·ch(t) y=r·sh(t) z=r·t** 

Спосіб опису кривої трьома функціями користувача наведено на рис. 4.91 а. В маркер записуються параметрично три імені функцій.

Спосіб опису кривої однією функцією-вектором користувача наведено на рис. 4.90 б. Відмінністю є те, що в маркер записуються тільки одне ім'я функції.

Для обох способів кількість точок графіка можна змінити тільки на вкладці **Quick Plot Data** вікна налаштувань.

Спосіб опису кривої трьома векторами наведено на рис. 4.90 в. В маркер записуються параметрично три імені векторів Відмінністю є те, що кількість точок графіка можна визначати зовні поля графіка розміром векторів **N**.

Спростити визначення векторів опису кривої та забезпечити програмне керування кількістю точок графіка (рис. 4.91 г) можна вбудованою функцією

#### CreateSpace(fun, min, max, N, fmap).

Аргументи функції є аналогічними аргументам функції **CreateMesh** для поверхонь. В параметр **fun** вводяться через кому імена функційкористувача або ім'я функції-вектора опису кривої.



Рис. 4.91. Опис кривої: а – функціями; б – функцією-вектором; в – векторами; г – функцією **CreateSpace** 

## Засоби MathCAD Prime

*MathCAD Prime* має дещо скорочену кількість типів тривимірних графіків, зменшені можливості візуальних ефектів та спрощене налаштування графіків.

Система має вбудовані засоби для малюванні тільки двох типів тривимірних графіків: поверхневих графіки **Surface Plot** (рис.4.92 a) та контурних графіків **Contour Plot** (рис.4.92 б). Як окремі вбудовані типи тривимірних графіків точкові графіки, векторні діаграми, графіки просторових кривих, стовпцеві діаграми, графікі рівнів в пакеті не реалізовані.



Рис. 4.92. Вбудовані тривимірні графікі: а – поверхні; б – контурний

Створення тривимірних графіків проводиться заповненням маркерів шаблону (рис. 4.93) графічної області тривимірних графіків.

Шаблон виводиться на екран клавішами «*CTRL*+3» або пунктом меню **Plot-Insert Plot-3D Plot** (рис. 4.9) стрічки.

Шаблон має один маркер **Z-ахіs**.



Рис. 4. 93. Шаблон тривимірних графіків

Тривимірна графіка в системі базується на вбудованих матричних перетвореннях. 3D графіки є зображенням елементів матриці. Координатами по осях X, Y виступають індекси елементів матриці, по осі Z відображаються значення елементів матриці з відповідними індексами.

В маркер вводиться ім'я об'єкта, графік якого передбачається побудувати. Для поверхонь об'єктом може бути ім'я відомої матриці, ім'я наперед визначеної функції користувача від двох змінних чи функції-вектора

з трьома елементуми параметричного опису або безпосередньо вираз розрахунку функції двох змінних. Для просторових кривих об'єктом може бути ім'я наперед визначеної функції-вектора з трьома елементуми параметричного опису.

За замовчанням графіки будуються в 41-й точці у вигляді дротової чорно-білої моделі з аргументами функції діапазоні від -10 до 10.

Визначити кілька об'єктів в маркері для побудови кількох графіків на одному графічному неможливо. Графіки на поле додаються кнопкою Plots-Traces-Add

Безпосередньо графічному полі можна повернути, зсунути зображення та змінити його масштаб кнопками 💀 💠 🔍, які розташовані в лівому верхньому куті шаблону (рис. 4.93).

В разі створення зображення матриці кількість точок графіка та діапазони аргументів є незмінними та визначеними самою матрицею.

В разі створення графіка функції, яка явно залежить від змінних х,у поведінка пакета різниться. За умови введення імені функції та аргументів **f**(**x**, **y**) (рис. 4.94 б) пакет сприймає визначені зовні ранжованими змінними значення кількості точок графіка та діапазони аргументів та дозволяє редагувати значення діапазону безпосередньо на осях, кількість точок – полем **Plots-Traces-Number of Points** стрічки. За умови введення імені функції без аргументів **f** (рис. 4.94 в) пакет НЕ сприймає визначені зовні ранжованими змінними значення кількості точок графіка. Змінити значення діапазону можна тільки безпосередньо на осях, кількість точок – полем **Plots-Traces-Number of Points** стрічки.

В разі створення графіка матриці кількість точок графіка та діапазони аргументів є незмінними та визначені самою матрицею.

Наприклад, на рис. 4.94 а показано графік функції  $fzf(x,y) = x^2 + y^2$ , який створено за замовчанням з вказанням в маркері тільки імені функції. Як видно, визначені ранжовані змінні аргументів ніяк не вплинули на зображення. На рис. 4.94 б показано графік тієї ж функції, який створено за замовчанням з вказанням в маркері імені функції з аргументами. Вигляд графіка обумовлений визначеними ранжованими змінними аргументів. На рис. 4.94 в показано графік тієї ж функції, який створено за замовчанням з вказанням в маркері імені функції з аргументами. Вигляд графіка обумовлений визначеними ранжованими змінними аргументів. На рис. 4.94 в показано графік тієї ж функції, який створено за замовчанням з вказанням в маркері імені функції. Вигляд створений редагуванням значень

осей та поля **Plots-Traces-Number of Points** стрічки. Різниця між зображеннями 4.94 б та 4.94 в полягає в кількості міток на осях.



Рис. 4.94. Визначення діапазонів: а – за замовчанням; б – за замовчанням з аргументами; в – редагуванням

Редагування діапазонів аргументів та мітками на осях проводиться аналогічно редагуванню плоских графіків безпосередньо на самих осях.

Активна вісь графіка на екрані виділяється синім кольором. Для зміни активної вісі застосовується перемикач осей , який розташований в правому верхньому куті графічного поля. Активація осі проводиться кліком миші на зображенні на перемикачі обраної осі. За умови активації осі на правому боці поля виводиться вертикальна шкала. Для зміни значення максимального/мінімального значення осі слід клікнути на відповідному значенні шкали та ввести потрібне значення. Кількість міток на осі визначається автоматично за двома крайніми значеннями шкали. Для зміни кількості міток на осі слід змінити попереднє перед останнім або наступне після першого значення шкали.

Наприклад, якщо на шкалі відображаються значення –10 …. 5 10, то на осі будуть встановлені 5 міток. Зміна передостаннього значення 5 на 0 встановить 3 мітки. Зміна значень шкали на 0 … 0.8 1 встановить 6 міток.

Для тривимірних графіків з групи **Style** стрічки **Plots** можна визначити суцільний колір фарбування граней **Surface Fill**, додати у вузли графіка символи **Symbol** точки та ребра граней **Line Style** визначеного кольору **Trace Color** та розміру **Trace Thickness**.

Побудова тривимірного поверхневого графіка функції, яку задано в параметричному вигляді НЕМОЖЛИВА трьома параметричними функціями користувача. Пакет сприймає тільки одну функцію користувача у вигляді вектора-стовпця.

Наприклад, спроба задати в маркері графічного поля три імені параметричних функцій видає невірний результат (рис. 4.95 а). Введення в маркері одного імені функції-вектора створює зображення (рис. 4.95 б).

Повне програмне керування діапазоном аргументів графіка, кількістю точок, системою координат опису забезпечує вбудована функція **CreateMesh**. Синтаксис функції **CreateMesh** в *MathCAD Prime* є аналогічним синтаксису в *MathCAD*.



Рис. 4.95. Параметричний графік: а – трьох функцій; б – функції-вектора

Наприклад, на рис. 4.96 показано використання функції **CreateMesh** для створення графіків сферичної поверхні в діапазоні аргументів від **0** до **n** та від **0** до **2n** по 20-ти точках. Графік 4.96 а створений по трьох параметричних функціях **R**, **\phi**, **\theta** в сферичних координатах. Графік 4.96 б створений по одній параметричній функції-вектору **h** в сферичних координатах.



Рис. 4.96. Використання **CreateMesh**: а – для трьох функцій; б – для однієї

# Точкові графіки

Точкові графіки є різновидом поверхневих графіків, в яких зображуються тільки точки значень функції у вузлах сітки. Грані полігонів та їхні ребра не малюються та не зафарбовуються.

Малюється точковий графік налаштуванням первинного зображення поверхні. На стрічці слід вимкнути зображення ребер поверхні полем **Plots-Styles-Line Style-none**, вмикнути зображення точок полем **Plots-Styles-Symbol-dot**(рис. 4.97 а). Графік зображується тільки вузлами у вигляді замальованих кіл. Інші форми точок не передбачені. Колір зображення змінюється полем **Plots-Styles-Trace Color** стрічки, розмір точок – полем **Plots-Styles-Trace Thickness** стрічки (рис. 4.97 б).



Рис. 4.97. Точковий графік: а – первинний; б – змінений

# Графіки просторових кривих

Умовою створення графіка просторової кривої є наявність її математичного опису в параметричному вигляді.

Крива може бути задана у вигляді параметричної функції користувача або за допомогою вбудованої функції **CreateSpace**. Функція повинна бути вектором-стовпцем, три елементи якого містять визначення аргументів **x**, **y**, **z**.

Графіки просторових кривих в пакеті вважаються різновидом поверхневих графіків. Для їхньої побудови використовується поле та шаблон поверхневих графіків.

В маркер графічного поля записується ім'я функції-вектора користувача або результат функції **CreateSpace**.

Розглянемо способи побудови графіка гіперболічної просторової кривої з розглянутого прикладу для *MathCAD*.

Спосіб опису кривої іменем функції користувача без аргументів наведено на рис. 4.98 а. Кількість точок та діапазон графіка не залежать від ранжованої змінної. Значення можна змінити тільки діалогом редагування.

Спосіб опису кривої іменем функції користувача з аргументами наведено на рис. 4.98 б. Кількість точок графіка та діапазон графіка визначені ранжованою змінною.

Діапазон аргументів змінити точно вкрай важко, бо первинним аргументом опису функції є змінна t. Саме її діапазон визначає вигляд графіка, а по осях показані діапазони функцій x, y, z.

Спростити визначення векторів опису кривої та забезпечити програмне керування кількістю точок графіка призначена вбудована функція **CreateSpace**.

Застосування функції **CreateSpace** наведено на рис. 4.98 в діапазоні аргументу від -5 до 5 по 5-ти точках.



Рис. 4.98. Графіки кривих: а – іменем з аргументами; б – іменем; в – **CreateSpace** 

# Контурні графіки

Шаблон контурного графіка (рис. 4.99 а) виводиться на екран клавішами «*CTRL*+5» або пунктом меню **Plot-Plot-Insert Plot-Contour Plot** (рис. 4.9) стрічки.

На графік можна вивести контурні лінії, які відповідають визначеним значенням функції — ізолінії та для кількісного аналізу функції нанести на контурні лінії чисельні значення функції (рис. 4. 99 в). Контурні графіки є єдиним типом графіків в пакеті, для яких передбачені різнокольорові палітри фарбування.

Шаблон графіка має один маркер.



Рис. 4. 99. Контурні графіки: а – шаблон; б – вигляд за замовчанням; в – відредагований

В маркер вводиться ім'я відомої матриці, ім'я наперед визначеної функції користувача від двох змінних чи функції-вектора з трьома елементуми параметричного опису або безпосередньо вираз розрахунку функції двох змінних.

За замовчанням графіки будуються в 41-й точці у вигляді кольорової мапи в палітрі райдуги з аргументами функції в діапазоні від -10 до 10.

Побудувати кілька контурних графіків на одному графічному полі неможливо.

Зручним інструментом аналізу є шкала кольорів графіка. Горизонтальна шкала виводиться знизу від контурного графіка. За замовчанням шкала має 10 градацій зі сталим кроком. На шкалу нанесені чисельні значення функції та відповідні їм кольори.

Параметри зображення визначаються вибором з наявних в списках в наступних групах стрічки :

- Line Style візерунок ізолінії;
- Trace Color, Trace Thickness колір та товщина ізолінії;
- Contour Values нанесення чисельних значень на ізолінії;
- **Color Scheme** палітра фарбування.

Змінити діапазон аргументів контурного графіка, значення шкали кольорів можна редагуванням відповідних значень по осях аналогічно двовимірним графікам та графікам поверхні. Для зміни значення слід клікнути на обраному значенні та ввести потрібне значення. Кількість міток на осі та кольорів в шкалі кольорів визначається автоматично за двома крайніми значеннями. Для зміни кількості слід змінити попереднє перед останнім або наступне після першого значення.

# Контрольні запитання

- 1. В якому діапазоні за замовчанням будується графік функції однієї змінної?
- 2. Як будуються графіки кількох функцій в режимі додаткової осі?
- 3. Що визначає режим Show markers?
- 4. Для чого використовується логарифмічний режим осей графіка?
- 5. Який режим призначено для отримання інформації про координати точок кривої?
- 6. В якому діапазоні за замовчанням будується графік функції двох змінних?
- 7. Що вводиться в поле маркера графіка двох змінних?
- 8. Як впливає визначення аргументу функції двох змінних у вигляді ранжованих змінних на вигляд графіка?
- 9. Що таке графік векторного поля?
- 10. Як враховується система координат для тривимірних графіків?
- 11. Яким типом графіків зображуються просторові криві?
- 12. Яка особливість контурних графіків?

## 4.8. Розрахунки з розмірностями

*Mathcad*, на відміну від більшості СКМ може проводити не тільки абстрактні математичні, але й науково-технічні розрахунків. Особливістю таких розрахунків є наявність розмірних величин в залежностях.

Змінну, значенням якої є величина з розмірністю, називають *розмірною змінною*.

Bci версії системи дозволяють роботу в системі розмірностей *CI* (SI, International), в системі *СГС* (CGS, сантиметр-грам-секунда), в американській система одиниць (USCS, United States Customary System of Units).

*Mathcad* додатково працює в системі *MKC* (**MKS**, метр-кілограмсекунда) та системі користувача (**Custom**). Остання система є комбінацією одиниць розмірності з вищенаведених систем. Додати власні або змінити існуючі одиниці в систему користувача неможливо.

В основу розрахунків з розмірностями покладені фундаментальні метрологічні величини. До таких величин відносять довжину L(length), *масу* M(mass), *час* T(time), *силу струму* C(current). *температуру* K(temperature), *силу світла* I(luminocity) та *одинцю речовини* S(substance).

В системі СІ базовими одиницями розмірності фундаментальних **meter)**, кілограм величин є метр m (length, kα (mass, kilogramm), секунда s (time, second), amnep A (current, Ampere), кельвін Κ (temperature, Kelvin), кандела cd (luminocity, candella), MOJL (mole).

Визначення поточної системи розмірностей для документа *MathCAD* проводиться на сторінці **Unit System** діалогового вікна **Worksheet Options** (рис. 4.100 а). Вікно активується головним меню **Tools-Worksheet Options**. Для *MathCAD Prime* визначення системи проводиться зі списку **Math-Units-Unit System** стрічки (рис. 4.100 б).

Для проведення суто математичних обчислень та за необхідності збільшити продуктивність системи рекомендується вимкнути розмірності пунктом **None**.



Рис. 4.100. Системи розмірностей: а – *MathCAD*; б – *MathCAD Prime* [1]

Розмірність може призначатися будь-яким даним в документі: чисельним значенням в розрахункових виразах, одиночним змінним, ранжованим змінним, векторам та матрицям, функціям користувача. Обмеження стосується умовних операторів, результати яких повинні бути тільки безрозмірними.

Для додавання розмірності до змінної або чисельного значення достатньо після числа ввести назву одиниці розмірності. Вбудовані розмірності вводяться АНГЛІЙСЬКОЮ МОВОЮ.

Вірність перетворень та розрахунків з вбудованими розмірностями контролюється автоматично, в результати вбудовані розмірності проставляються автоматично.

Одиниці розмірності, які відсутні в пакеті, вводяться як розмірні змінні через відомі одиниці.

Наприклад, dm:=10 ·cm day:=24 ·hr month:=31 ·day

Саме так можна перекласти назви розмірностей на іншу мову або створити власні позасистемні одиниці. Наприклад,

# см:=0.01 ·m мм:=mm B:=V година:=3600 ·sec BT:=watt квартира:=60 ·м<sup>2</sup> дім:=100 ·квартира людина:=1

Слід мати на увазі, що в результати розмірности користувача автоматично НЕ ПРОСТАВЛЯЮТЬСЯ. Такі розмірності потребують ручного проставляння.

На графіках *MathCAD* розмірності користувача не передбачені, на них відображаються тільки вбудовані розмірності. В *MathCAD Prime* одиниці розмірності на графіках задаються без обмежень.

Розмірність при визначенні ранжованої змінної вказується для кожного із значень виразу. Наприклад, path:=1mm,1.11mm..2.0mm

Елементи масивів можуть мати тільки однакову розмірність. Розмірність масивів вводиться додаванням імені розмірності ззовні після масиву.

Наприклад,  $path := \begin{pmatrix} 1.5 \\ 2.2 \end{pmatrix} mm$  area :=  $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} cm^2$ 

Ввести документ вбудовані одиниць розмірності можна не тільки з клавіатури, а й з інтерфейсу пакета. Введення з інтерфейсу полягає в кліці мишею на обраному елементі списку доступних розмірностей. Частина списку вбудованих розмірностей наведено в таблиці 4.9.

В *MathCAD* список вбудованих розмірностей знаходиться в діалоговому вікні **Insert Unit** (рис. 4.101 а). Вікно активується пунктом меню

Insert-Unit або кнопкою Inateлi Standard. В *MathCAD Prime* список вбудованих розмірностей (рис. 4.101 б) викликається кнопкою Math-Units-Units стрічки.



Рис. 4.101. Списки вбудованих розмірностей: а – *MathCAD*; б – *MathCAD Prime* 

Таблиця 4.9. Ти	ипи розмірних	величин	Mathcad	[2]
-----------------	---------------	---------	---------	-----

Категорія	Опис
Acceleration	Одиниці розмірності пришвидшення
Angle	Одиниці розмірності кутів
Area	Одиниці розмірності площ
Capacitance	Одиниці розмірності електричної ємності
Charge	Одиниці розмірності електричного заряду
Conductance	Одиниці розмірності електричної провідності
Current	Одиниці розмірності сили електричного струму
Energy	Одиниці розмірності енергії
Force	Одиниці розмірності сили
llluminosity	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
Substance	Одиниці розмірності кількості речовини
Temperature	Одиниці розмірності температури
Time	Одиниці розмірності часу
Velocity	Одиниці розмірності швидкості
Mass	Одиниці розмірності маси
Permeability	Одиниці розмірності магнітної проникненості
Permittivity	Одиниці розмірності електричної проникненості
Magnetic Flux	Одиниці розмірності магнітного потоку

У вікні Insert Unit всі розмірності, які є в системі, згруповані по категоріях. Вікно містить два списки: Dimension та Unit. В списку Dimension відображаються категорії розмірностей. Список Unit показує доступні вбудовані розмірності відповідної категорії.

В версіях системи до 14 працездатними були підпрограми-функції, результатом яких могли бути різні фізичні величини. Наприклад, підпрограма-функція могла розраховувати периметр та площу геометричної фігури. В останніх версіях така підпрограма неможлива. Всі результати підпрограми-функції повинні мати однакову розмірність.

В *MathCAD* розмірність підпрограм-функцій можна перевірити. Для визначення розмірності підпрограми-функції слід ввести ім'я функції без параметрів. В *MathCAD Prime* перевірка розмірності підпрограм-функцій не реалізована.

Результат містить ознаку функції літерою **f** та перелік аргументів в дужках. Після правої стрілки розташовуються результати перевірки.

Якщо результат функції може мати розмірність, то аргументи та результат позначаються словами **any1. алуN**. Для аргументу вказується розмірність відносно результату з позначкою степеню арифметичних дій відносно розмірності результату. Наприклад,

$$a(x) \coloneqq \sqrt{x}$$
  $a = f(any1^2) \rightarrow any1$ 

Якщо в тілі функції явно вказана розмірність, то аргументи та результат позначаються словами **any1.. anyN** з додаванням степеню арифметичних дій розмірності та тип розмірності. Наприклад,

 $b(x,y) := (2 \cdot x \cdot cm + y)^2$   $b = f(any1, any1 \cdot Length) \rightarrow any1^2 \cdot Length^2$ 

Якщо результат функції НЕ може мати розмірність, то він позначається словом **unitless**. Для аргументу вказується розмірність відносно результату. Наприклад,

 $d(x) := \sqrt{x + x}$   $d = f(\text{Unitless}) \rightarrow \text{Unitless}$   $z(x) := \sin(x \cdot g)$   $z = f(\text{Length}^{-1} \cdot \text{Time}^{-2}) \rightarrow \text{Unitless}$ 

Можливість проведення перевірки розмірностей привела до того, 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 тривимірних графіків [2].

### Розмірність результатів

Результати розрахунків *MathCAD* за замовчанням можуть виражатися через фундаментальні величини, наприклад,

$$F := 1 \cdot \text{kg} \cdot \text{g} = 9.807 \frac{\text{mass-length}}{\text{time}^2}$$

або через базові одиниці обраної системи одиниць:

$$F := 1 \text{kg} \cdot \text{g} = 9.807 \frac{\text{kg} \cdot \text{m}}{s^2}$$

Визначення способу відображення результату проводиться пунктом Display dimensions вікна Worksheet Options-Dimensions (рис. 4.102).

Vorksheet Option	S				>
Built-In Vari	ables		Calculation	Display	
Unit System	1	Dim	ensions	Compatibility	
Display dimen Dimension Nam Mass	es mass		]		
Length	length		Į		
Time	time		]		
Current	current		]		
Temperature	temperature	•	]		
Luminosity	luminosity		]		
Substance	substance		]		

Рис.4.102. Визначення одиниць результату

Результат з розмірністю може надаватися в базових одиницях у вигляді добутку, наприклад,  $\underline{F} := 1 \cdot \text{kg} \cdot \text{g} = 9.807 \text{ s}^{-2} \cdot \text{kg} \cdot \text{m}$ ,

в базових одиницях у вигляді дробу, наприклад,

,

$$F := 1 \text{kg} \text{g} = 9.807 \frac{\text{kg} \text{m}}{s^2}$$

Визначення формату розмірностей результатів розрахунків визначається на сторінці **Unit Display** (рис. 4.103) діалогового вікна **Result Format** форматування результатів.

Поле Format units встановлює режим відображення. Без активізації цього поля одиниці вимірювання відображаються у вигляді добутку.

Поле Simplify units when possible встановлює режим відображення розмірностей через похідні одиниці.

Поле **Show unit exponents as a fraction** встановлює для всього документа режим відображення одиниць розмірностей у вигляді раціональних дробів.

Γ	Result Format				×			
	Number Format	Display Options	Unit Display	Tolerance				
q	C Exception							
	Format units							
	Show unit exponents as a fraction							
		Aponones as a ma	coon					

Рис. 4.103. Сторінка форматування розмірностей результату

Результати розрахунків *MathCAD Prime* виражаються через базові одиниці обраної системи розмірності у вигляді дробу:

$$F \coloneqq 1 \cdot kg \cdot g = 9.807 \frac{kg \cdot m}{s^2}$$

або в похідних одиницях, наприклад,

 $F \coloneqq 1 \cdot kg \cdot g = 9.807 \ N$ 

Для *MathCAD Prime* вибір базових чи похідних одиниць проводиться полем **Math-Units-Base Units** стрічки (рис. 4.100 б).

Користувач має змогу змінити одиниці розмірності результату. Клік миші на результаті в *MathCAD* показує позаду значення та розмірності маркер, В цей маркер вводиться потрібна одиниця розмірності. Після зміни результат перераховується автоматично. Розмірність результату в *MathCAD Prime* редагується заміною існуючої розмірності на потрібну.

## Контрольні запитання

- 1. Що таке розмірна змінна?
- 2. Що таке базові одиниці вимірювання?
- 3. Які системи вимірювання підтримує MathCAD ?
- 4. Як вводиться розмірна змінна?
- 5. Як змінити одиницю вимірювання?
- 6. Як ввести одиницю вимірювання, яка відсутня в системі?

#### 4.9. Символьні обчислення

Символьні або аналітичні обчислення є комп'ютерними обчисленнями, які призначені для надання результатів математичних операцій на кшталт тотожних перетворень, підстановок, розв'язання рівнянь тощо не тільки у вигляді чисел, а й у вигляді формульних виразів. Символьні обчислення є ознакою систем комп'ютерної алгебри (CAS).

На відміну від чисельних методів комп'ютерні символьні обчислення ще не є досконалими. Часто отримані результати символьних обчислень не придатні для практичного застосування, а для деяких випадків відповіддю є повторення вхідних даних.

Символьний обчислення в пакеті проводяться ліцензованим придбаним зовнішнім символьним процесором. В версіях до 12-ї ним був скорочений процесор найпотужнішої спеціалізованої СКА *Maple*. З версії 13 по сьогодення застосовується процесор СКА *MuPad*.

Наприклад [2], символьне обчислення суми  $\sum_{n=2}^{\infty} \frac{n^3 - 1}{n^3 + 1}$  в *MathCAD 11,12* дає простий для використання вірний результат  $\frac{2}{3}$ , результат в *MathCAD 14* є громіздким і незручним:

$$\frac{2 \cdot \Gamma \left[ \frac{3}{2} - \left( \frac{\sqrt{3}}{2} \right) \cdot i \right] \cdot \Gamma \left( \frac{3}{2} + \frac{1}{2} \cdot \sqrt{3} \cdot i \right)}{\Gamma \left[ \frac{5}{2} - \left( \frac{\sqrt{3}}{2} \right) \cdot i \right] \cdot \Gamma \left( \frac{5}{2} + \frac{1}{2} \cdot \sqrt{3} \cdot i \right)}$$

а результат в MathCAD 15 вже зовсім не можна вважати практичним:

$$\frac{\lim_{i \to \infty} \frac{\left(n^{3} - 1\right)^{i}}{\left(n^{3} + 1\right)^{i}} \cdot (n + 1)^{2} \cdot \left(n^{2} - n + 1\right)^{2}}{\left(n - 1\right)^{2} \cdot \left(n^{2} + n + 1\right)^{2}} \quad \text{if } \neg \left(n \in \text{DOM\_SET}\left(1, -\frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i, -\frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i\right)\right)$$
$$\prod_{i=2}^{\infty} \frac{n^{3} - 1}{n^{3} + 1} \quad \text{if } n \in \text{DOM\_SET}\left(1, -\frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i, -\frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i\right)$$

Тільки MathCAD Prime 7 повертається до можливостей MathCAD 12

$$\prod_{n=2}^{\infty} \left( \frac{n^3 - 1}{n^3 + 1} \right) \rightarrow \frac{2}{3}$$

Пакет дозволяє виконати символьно велику кількість математичних операцій. Алгоритми різних операцій символьних обчислень є

спеціалізованими, тому в пакеті дії символьних обчислень згруповані по типах.

Група «спектральних» прямих та зворотних перетворень **Transform** містить перетворення Фур'є Fourier та Inverse Fourier, Лапласа Laplace та Inverse Laplace, дискретне z-перетворення Z та Inverse Z.

Група матричних перетворень Matrix містить дії з обернення матриць Invert, транспонування Transpose, обчислення визначника матриці Determinant.

Група символьного обчислення математичної залежності **Evaluate** включно з обчисленням сум, добутків, границь.

Група тотожних математичних перетворень містить дії зі спрощення виразу **Simplify**, розкриття дужок в виразі **Expand**, розкладання залежності на множники **Factor**, переведення добутку в доданок **Collect**,

За спеціальними алгоритмами проводяться дій обчислення коефіцієнтів полінома, на який може бути розкладена залежність Polynomial Coefficients, підстановки змінних в вирази Substitute, диференціювання Differentiate, інтегрування Integrate, розкладання в ряд Тейлора Expand to Series, розкладання на частини Convert to Partial Fraction, розв'язання рівнянь та систем рівнянь Solve, отримання числового результату з довільною кількістю знаків мантиси Floating point.

Символьні дії в пакетах можна провести «*швидким»* **menu** способом та способом «*реального часу»* **live** [1; 2; 30]. Способи відрізняються відображенням на екрані, введенням даних для обчислення та сприйняттям оточуючих змінних.

В *MathCAD* реалізовані обидва способи, в *MathCAD Prime* використовується тільки спосіб *«реального часу»*.

Символьні обчислення «*швидкий*» способом проводяться тільки через меню **Symbolics** (рис. 4.104 а). Може бути проведено тільки одне перетворення одноразово. Результат немає зв'язку з первинною залежністю. Ланцюги дій, як для числових обчислень не дозволяються. Вигляд результату може бути визначений користувачем полями діалогового вікна (рис. 4.104 б), яке виводиться пунктом **Evaluation style** меню.



Рис. 4.104. «Швидкі» символьні перетворення: а – меню; б – вікно форматування результату

Результат «швидких» символьних перетворень може бути виведений на місці виразу-умови полем **Evaluate in Place**, праворуч від виразу-умови полем **Horizontally** (рис. 4.105 а), під виразом-умовою в наступному рядку полем **Vertically** (рис. 4.105 б), до результату може бути доданий коментар з назвою проведеної дії Show Comments (рис. 4.105 в). Слід зауважити, що вигляд пунктів Vertically, inserting lines та Vertically, without inserting lines не відрізняється, додатковий рядок не додається.

sin(2·x) 2·cos(x)·sin(x) sin(2·x) sin(2·x) 2·cos(x)·sin(x) expands to 2·cos(x)·sin(x) a δ Β

Рис. 4.105 – Відображення «швидкого» перетворення: а – горизонтально; б – вертикально; в – вертикально з коментарем

«Швидкі» символьні обчислення потребують обов'язкового виділення мишею виразу, до якого передбачається застосування символьних дій. «Швидкі» символьні обчислення не можуть бути застосовані до функцій користувача.

Послідовність дій «швидким» способом має наступний порядок:

- введення виразу та для функцій однієї змінної виділення всього виразу охопленням синім кутом курсору, для функції кількох змінних виділення потрібної змінної;
- 2. клік на обраному пункті меню Symbolics.

Символьні обчислення «реального часу» можуть проводитися комбінацією клавіш клавіатури або кнопками панелі **Symbolics** (рис. 4.106 а) в *MathCAD*, полями панелі **Math-Operators and Symbols-Symbolics** стрічки (рис. 4.106 б) в *MathCAD Prime*. Спосіб дозволяє проводити ланцюги дій та дії з додатковими умовами. Він не потребують обов'язкового виділення мишею виразу, до якого передбачається застосування символьних дій. Символьні обчислення «реального часу» можуть бути застосовані до функцій користувача та змінних. Результат залишається пов'язаним з первинною залежністю.

Ознакою символьного обчислення «реального часу» є наявність в рядку виразу символу правої стрілки «→».

Для символьного обчислення виразу «реальним часом» достатньо замість символу виведення числового результату знаку рівності «=» застосувати символ правої стрілки «->». Символ може бути введений з клавіатури клавішами «*Ctrl*+.» або кнопками панелей **Evaluation**, **Symbolic** в *MathCAD*, **Math-Operators and Symbols-Symbolics-Operators** стрічки в *MathCAD Prime*.

			Symbolics •		Units +	Olits	- Contra
			Operators				
umbolio			lim	$\rightarrow$			
ymbolic		~	Keywords				
$\rightarrow$	$\bullet \rightarrow$	Modifiers	assume	cauchy	coeffs	collect	combine
float	rectangular	assume	confrac	expand	explicit	factor	float
nour	rectangular	assame	fourier	fully	invfourier	invlaplace	invztrans
solve	simplify	substitute	laplace	parfrac	rectangular	rewrite	series
factor	expand	coeffs	simplify	solve	substitute	using	ztrans
			Modifiers				
collect	series	partrac	ALL	acos	acot	asin	atan
fourier	laplace	ztrans	cauchy	complex	cos	cosh	cot
fourior	inulanlasa	insettono	coth	degree	domain	even	exp
wouner	inviapiace	Invztrans	fraction	fully	gamma	generated	integer
$M^{T} \rightarrow$	$M^{-1} \rightarrow$	M  →	ln	log	matrix	max	odd
volicit	combino	confrac	raw	real	RealRange	signum	sin
sapiren	combine	connac	sincos	sinh	sinhcosh	tan	tanh
ewrite			using				

Рис. 4.106. Панелі «реального часу»: a – MathCAD; б – MathCAD Prime

Для інших дій крім математичного виразу та знаку символьного виведення потребується введення уточнюючого ключового слова визначення типу операції. Перетворення складних залежностей, ланцюгові перетворення після ключового слова потребують додатково уточнення імені змінної та режиму проведення перетворень.

Для створення виразу символьних обчислень «реального часу» призначено універсальний шаблон (рис. 4.107). Маркер 1 призначений для введення математичного виразу, в маркер 2 вводяться ключові слова та

додаткові умови. Створити шаблон можна клавішами «*Ctrl+Shift+.*» або кнопкою символьної панелі.



Рис. 4.107. Універсальний шаблон «реального часу»: а – *MathCAD*; б – *MathCAD Prime* 

Заповнювати шаблон можна в будь-якій послідовності. Можна спочатку вводити математичний вираз, потім ключове слово. Можна навпаки. Ключове слово можна вводити з клавіатури, можна кнопкою символьної панелі. Натискання кнопки ключового слова без наявності універсального шаблона в документі призведе до створення шаблона дії із вставленим ключовим словом в маркер 2.

Дії вбудованих символьних операцій наведено в таблиці 4.10.

### Додаткові можливості способу «реального часу»

Для способу реального часу пакет має два додаткові символьні перетворення: **rewrite**, **combine**. Перетворення **rewrite** полягає наданні вхідному виразу форми, коли залежність описується комбінацією базових функцій. Перетворення **combine** схоже на перетворення приведення подібних **collect**, але призначено не для поліномів, а для виразів, в яких використовуються базові функції.

В якості базових функцій можуть обиратися тригонометричні sincos (sin, cos),tan, зворотні тригонометричні acos, asin, atan,acot, гіперболічні sinhcosh (sinh, cosh),tanh,coth, логарифмічні ln, log, експоненційній функції еxp. Тип базової функції, через яку буде виражатися вхідна залежність, задається ім'ям через кому після ключового слова або відповідною кнопкою Symbolics-Modifier стрічки.

Наведемо приклади перетворень.

Перетворення **rewrite** для виразу гіперболічної функції через логарифмічну:

$$\operatorname{asinh}(x) \text{ rewrite, } \ln \to \ln \left[ x + \sqrt{\left[ x^2 + 1 \right]} \right], \qquad \operatorname{asinh}(x) \xrightarrow{rewrite, \ln} \ln \left( \sqrt{x^2 + 1} + x \right)$$

Таблиця 4.10. Дії символьних функцій

Д1я	«Швидкий»	«Live»	«Live»
	спосіб	MathCAD	MathCAD Prime

**matrix** – стандартні параметри матриць.

Обчислення не потребує уточнень. Для «швидких» обчислень проводиться відповідним пунктом меню після введення матриці в документ. Для «реального часу» відрізняється від числового розрахунку символом стрілки «—» замість символу «=».

транспонування	$ \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} $ by matrix transposition, yields $ \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix} $ $ \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}^{\mathrm{T}} \rightarrow \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix} $ $ \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{\mathrm{T}} \rightarrow \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} $	
обернення	$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ by \text{ matrix inversion, yields} \end{pmatrix} \begin{pmatrix} \frac{a_{22}}{a_{11} \cdot a_{22} - a_{12} \cdot a_{21}} & -\frac{a_{12}}{a_{11} \cdot a_{22} - a_{12} \cdot a_{21}} \\ -\frac{a_{21}}{a_{11} \cdot a_{22} - a_{12} \cdot a_{21}} & \frac{a_{11}}{a_{11} \cdot a_{22} - a_{12} \cdot a_{21}} \end{pmatrix}$	
	$ \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} \rightarrow \begin{bmatrix} \frac{a_{22}}{a_{11} \cdot a_{22} - a_{12} \cdot a_{21}} & \frac{-a_{12}}{a_{11} \cdot a_{22} - a_{12} \cdot a_{21}} \\ \frac{-a_{21}}{a_{11} \cdot a_{22} - a_{12} \cdot a_{21}} & \frac{a_{11}}{a_{11} \cdot a_{22} - a_{12} \cdot a_{21}} \end{bmatrix} $	
визначник	$ \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}^{\text{has determinant}} \xrightarrow{a_{11} \cdot a_{22} - a_{12} \cdot a_{21}} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \rightarrow a_{11} \cdot a_{22} - a_{12} \cdot a_{21} $	
	$\ [a_{21} \ a_{22}]\  = a_{11} \ a_{22} \ a_{12} \ a_{21}$	

Таблиця 4.10. Продовження

Дія	«Швидкий» спосіб	«Live»	«Live»
		MathCAD	MathCAD Prime

evaluate – обчислення.

Обчислення за замовчанням не потребує уточнень. Для «швидких» обчислень проводиться відповідним пунктом меню після введення виразу в документ. Для «реального часу» відрізняється від числового розрахунку символом стрілки «—» замість символу «=».

обчислення	$x := 1 + x \cdot (sin(asin(y \cdot z)))$ yields $1 + x \cdot (y \cdot z)$			
виразу				
	$\mathbf{x} \coloneqq 1 + \mathbf{x} \cdot (\sin(\operatorname{asin}(\mathbf{y} \cdot \mathbf{z}))) \to \mathbf{y} \cdot \mathbf{z} + 1$ $\mathbf{x} \coloneqq 1 + \mathbf{x} \cdot \sin(\operatorname{asin}(\mathbf{y} \cdot \mathbf{z})) \to \mathbf{y} \cdot \mathbf{z} + 1$			
	$\sin\left(\frac{\pi}{4}\right)  \frac{\sqrt{2}}{2} \qquad x \cdot \cos(0)  \frac{\pi \cdot x}{2} \qquad \qquad x \cdot \cos(0) \rightarrow \frac{\pi \cdot x}{2} = \qquad \sin\left(\frac{\pi}{4}\right) \rightarrow \frac{\sqrt{2}}{2} = 0.707$			
	$x \cdot \operatorname{acos}(0) \rightarrow \frac{x \cdot \pi}{2} = ?$ $\operatorname{sin}\left(\frac{\pi}{4}\right) \rightarrow \frac{\sqrt{2}}{2} = 0.707$			
	Обчислення виконується разом зі спрощенням виразу.			
	На «швидкі» обчислення не впливає зміна значень змінних, результат обчислень «реального часу» залежить від значень оточуючих змінних.			
	Результат обчислення надається в формульному вигляді. В разі відсутності у виразі змінних послідовно чисельним обчисленням введенням символу «=» отримати числове значення.			
обчислення	$f(\mathbf{x},\mathbf{y},\mathbf{z}) \coloneqq 1 + \mathbf{x} \cdot \sin(\operatorname{asin}(\mathbf{y} \cdot \mathbf{z}))$			
функції користувача	$ \begin{array}{ccc} f(\mathbf{x},\mathbf{y},\mathbf{z}) \text{ yields } f(\mathbf{x},\mathbf{y},\mathbf{z}) & f(\mathbf{x},\mathbf{y},\mathbf{z}) \to \mathbf{x} \cdot \mathbf{y} \cdot \mathbf{z} + 1 \\ f(\mathbf{x},1,1) \text{ yields } f(\mathbf{x},1,1) & f(\mathbf{x},1,1) \to \mathbf{x} + 1 \end{array} \begin{array}{c} f(x,y,z) \to x \cdot y \cdot z + 1 \\ f(x,1,1) \to x + 1 \end{array} $			
	$\begin{aligned} \mathbf{f}(\mathbf{x}) &\coloneqq \mathbf{x}^2 \cdot \cos(\mathbf{x})  \mathbf{f}(\mathbf{x}) \text{ yields } \mathbf{f}(\mathbf{x}) \\ &\qquad \mathbf{f}(1) \text{ yields } \mathbf{f}(1) \end{aligned} \qquad \mathbf{f}(\mathbf{x}) &\coloneqq \mathbf{x}^2 \cdot \cos(\mathbf{x})  \mathbf{f}(\mathbf{x}) \to \mathbf{x}^2 \cdot \cos(\mathbf{x}) \qquad \mathbf{f}(3) = -8.91  \mathbf{f}(3) \to 9 \cdot \cos(3) \end{aligned}$			
	«Швидкі» обчислення не проводять символьне обчислення			
	функцій користувача обчислення «реального часу»			
	wymanin acpuerybund, oo menemin apearbhoro naey			
	перетворюють функції користувача та символьно			
Дія	«Швидкий» спосіб	«Live»	«Live»	
-----------------------	--	--	--	
		MathCAD	MathCAD Prime	
evaluate -	– обчислення.			
обчислення суми	$\sum_{k=1}^{\infty} \frac{1}{4}$ yields $\frac{\pi}{90}$ $\sum_{k=1}^{\infty}$	$\frac{1}{1}\frac{1}{k} \rightarrow \frac{\pi^4}{90} \qquad \qquad \sum_{k=1}^{4}$	$\sum_{i=1}^{\infty} \frac{1}{k^4} \to \frac{\pi^4}{90}$	
	$\sum_{k=0}^{\infty}$	$\frac{1}{2^{k} \cdot k!} \rightarrow \sqrt{\mathbf{e}} = 1.649 \qquad \sum_{k=0}^{\infty} \frac{1}{2^{k} \cdot k!}$	$\frac{1}{k!} \rightarrow \sqrt{e} = 1.649$	
	$\sum_{k=0}^{\infty} \left[ \frac{(-1)^k \cdot x^{2k}}{(2k+1)!} \right] \to \frac{1}{2}$	$\frac{\sinh\left(2\cdot\sqrt{-\frac{x^2}{4}}\right)}{2\cdot\sqrt{-\frac{x^2}{4}}} \qquad \sum_{k=0}^{\infty} \frac{(-1)^k \cdot x}{(2\ k+1)^k}$	$\frac{x^{2k}}{1)!} \to \frac{\sin\left(\sqrt{x^2}\right)}{\sqrt{x^2}}$	
обчислення добутку	$\prod_{n=1}^{\infty} \left( \frac{4n^2}{4n^2 - 1} \right) \text{ yields } \frac{\pi}{2} \qquad \qquad \prod_{n=1}^{\infty}$	$\prod_{n=1}^{\infty} \left( \frac{4n^2}{4n^2 - 1} \right) \rightarrow \frac{\pi}{2} \qquad \qquad \prod_{n=1}^{\infty}$	$\prod_{i=1}^{\infty} \frac{4 n^2}{4 n^2 - 1} \rightarrow \frac{\pi}{2}$	
	$\prod_{k=1}^{\infty} \frac{1}{k^3 + 1} \rightarrow \Gamma\left(\frac{1}{2} - \frac{\sqrt{3} \cdot i}{2}\right) \cdot \Gamma\left(\frac{1}{2} + \frac{\sqrt{3} \cdot i}{2}\right)$	$\frac{\sqrt{3} \cdot \mathbf{i}}{2} \cdot \lim_{\mathbf{k} \to \infty} \frac{1}{\mathbf{k}! \cdot \Gamma\left(\mathbf{k} - \frac{1}{2} - \frac{\sqrt{3} \cdot \mathbf{i}}{2}\right)}$	$\Gamma\left(\mathbf{k}-\frac{1}{2}+\frac{\sqrt{3}\cdot\mathbf{i}}{2}\right)$	
			$\prod_{k=1}^{\infty} \frac{1}{k^3 + 1} \to 0$	
		$\prod_{k=0}^{m} (n+k) \to$	$ \begin{aligned} & \text{if } n \in \mathbb{Z} \land n \ge -m \land n \le 0 \\ & \parallel 0 \\ & \text{else if } n \in \mathbb{Z} \land n < -m \\ & \parallel \frac{(-n)! \cdot (-1)^{m+1}}{(-n - (m+1))!} \\ & \text{else if } \neg (n \in \mathbb{Z}) \lor -n < 0 \\ & \parallel \frac{\Gamma(n+m+1)}{\Gamma(n)} \end{aligned} $	
	Символьне обчислення замовчанням додаткових	сум та добуткін дій. Результат обч	в не потребує за ислення надається	
	в формульному вигляді. Е	мпляд результату з	залежить від версіі	

Таблиця 4.10. Продовження

Символьне обчислення сум та добутків не потребує за замовчанням додаткових дій. Результат обчислення надається в формульному вигляді. Вигляд результату залежить від версії програмного забезпечення. Слід бути готовим, що результат буде надано у вигляді, який не є практичним. Навести результат до практичного вигляду іноді можна додатковими умовами та ланцюгом символьних дій.

Таблиця 4.10. Продовження

Дія	«Швидкий» спосіб	«Live»	«Live»
		MathCAD	MathCAD Prime
evaluate	– обчислення.		
обчислення	sin(x) 1	$\lim_{x \to 1} \frac{\sin(x)}{\cos(x)} \to 1$	$\sin(x)$ 1
границі	$x \to 0$ $x$ $x$	$\rightarrow 0$ x	$\lim_{x \to 0} \frac{1}{x} \to 1$
	$\lim_{x \to 1} \frac{1}{1-x} \qquad \qquad \lim_{x \to -\infty} \frac{1}{x-x}$	$ m \frac{1}{1-x} \rightarrow undefined $	$\lim_{x \to 1} \frac{1}{1-x} \to undefined$
	undefined lin lim x-	$\underset{\rightarrow}{m} \frac{1}{1-x} \rightarrow -\infty$	$\lim_{x \to 1^{-}} \frac{1}{1-x} \to \infty$
	$\begin{array}{c} x \to 1^{+} 1 - x \\ -\infty \end{array} \qquad $	$\underset{\rightarrow}{m} \frac{1}{1-x} \rightarrow \infty$	$\lim_{x \to 1^+} \frac{1}{1-x} \to -\infty$
	В MathCAD Prime вве	едення позначки г	раниці lim з панелі
	Symbolics Cyttebo II	ідвищує швидкість	обчислень.
	Застосування одноб	ічних границь	дозволяє провести
	детальний аналіз пове	едінки функції. В <i>І</i>	MathCAD вибір типу
	границь проводиться к	нопками панелі С <b>а</b>	lculus, в MathCAD
	Prime – введенням позн	начки «-» для лівобі	чної границі, «+» для

правобічної границі в маркер аргументу границі.

simplify – спрощення шляхом групування на приведення подібних.

Перетворення потребує уточнюючого слова **simplify**. Для «швидких» обчислень проводиться відповідним пунктом меню після введення виразу в документ. Для «реального часу» проводиться кнопкою символьної панелі або введенням слова з клавіатури в універсальний шаблон. Дія є зворотною до дії розкриття виразу **expand**.

спрощення функції  $f_{\infty}^{(a,b,c)} := \frac{(a+b)^2 - c^2}{a+b+c}$  ff(x.v.z) simplifies to f(x,y,z)f(x,y,z) simplify  $\rightarrow x + y - z$   $f(1,1,1) \xrightarrow{simplify} 1$ «Швидкі» обчислення не проводять спрощення функцій користувача, обчислення «реального часу» перетворюють функції користувача та символьно розраховують значення виклику функції.

Таблиця 4.10. Продовження

Пія	«Шрилкий» спосіб	// ive	"I ive
Дія	«швидкии» спосто		
		MathCAD	MathCAD Prime
simplify-	спрощення шляхом групува	ння на приведенн	я подібних.
спрощення виразу	$\frac{\frac{x^2 - 3 \cdot x - 4}{x - 4} + 2 \cdot x - 5 \text{ simplifies to } 3 \cdot x - 4}{x - 4}$		
	$\frac{x^2 - 3 \cdot x - 4}{x - 4} + 2 \cdot x - 5 \text{ simplify } \rightarrow 3 \cdot x - 4$		
	$\frac{x^2 - 3 x - 4}{x - 4} + 2 x - 5 \xrightarrow{simplify} 3 \cdot x - 4$		
	Слід мати на увазі, що на результат спрощення суттєво впливає форма запису аргументів. Результатом дії для виразів з мантисою в операндах дійсних числах може бути вхідний вираз з операндами з максимальною кількістю знаків після коми:		
	$\frac{3-z^2}{\sqrt{3}-z} \text{ simplify } \rightarrow z + \sqrt{3} \qquad \frac{3.0-z^2}{\sqrt{3}-z}$	simplify $\rightarrow \frac{z^2 - 3}{z - 1.732050807}$	0 5688772935
expand - p	озкриття дужок, перетворе	ення виразу на с	суму та добуток
функцій одного аргументу.			
Перетворення потребує уточнюючого слова <b>ехрапd</b> . Лля «швилких»			
обчислень проводиться відповідним пунктом меню після ввелення виразу в			
локумент. Лля «реального часу» проволиться кнопкою символьної панелі			
або ввеленням слова з клавіатури в універсальний шаблон та може			
застосовуватися до виразів і функцій користувача.			
	exp(x + y) $exp(x + y)$ even	and $\rightarrow e^{X}e^{Y}$	

$$exp(x + y)$$
  $exp(x + y)$   $exp(x + y)$   $expand \rightarrow e^{x} \cdot e^{y}$   
 $exp(x + y)$   $exp(x + y)$   $expand \rightarrow e^{x} \cdot e^{y}$   
За потреби залишити визначений операнда залежності в  
результаті в способі «реального часу» дія уточнюється  
додаванням виразу операнда через кому після ключового  
слова. Для «швидких» перетворень можливість не  
реалізовано.

Дія	«Швидкий» спосіб	«Live»	«Live»
		MathCAD	MathCAD Prime
<b>expand</b> – po3	криття дужок.		
	Наприклад: (x - 1)-	$(x^2 + 1) \cdot (x + 1)$ expand $\rightarrow x^4 -$	1
	$(x-1)\cdot (x^2+1)\cdot (x+1) \text{ expand}, x^2+1 \rightarrow x^2\cdot (x^2+1)-x^2-1$		
	$(x-1) \cdot (x+1) \cdot (x^2+1) \xrightarrow{expand} x^4 - 1$ $(x-1) \cdot (x+1) \cdot (x^2+1) \xrightarrow{expand} x^2 + 1 x^2 \cdot (x^2+1) - (x^2+1)$		
	Є зворотною до спроще	ення simplify:	
	$\tan(3x) + \cos(2x) \text{ expand } \rightarrow \cos(x)^2 - \sin(x)^2 - \frac{3 \cdot \tan(x) - \tan(x)^3}{3 \cdot \tan(x)^2 - 1}$		
	$\cos(x)^2 - \sin(x)^2 - \frac{3 \cdot \tan(x) - \tan(x)^3}{3 \cdot \tan(x)^2 - 1} \text{ simplify } \rightarrow \cos(2 \cdot x) + \tan(3 \cdot x)$		
	«Швидкий» на відміну	від «реального ча	асу» не проводить
	операцію для функцій к	ористувача.	
<b>float</b> – о практично нес	тримання числового зн обмеженою кількістю зна	ачення символьно ків після коми.	ого розрахунку з
Дія потребує проводиться в Для «реально введенням сло	уточнюючого слова з відповідним пунктом мен ого часу» проводиться ова з клавіатури в універс	float. Для «шы ю після введення н кнопкою симво. альний шаблон.	идких» обчислень зиразу в документ. льної панелі або
числовий	Для «швидких» обчисл	ень кількість ци	фр визначається в
розрахунок	спливаючому діалоговог	му вікні	
	Floating Point Evaluation     X       Floating Point Precision:     Image: Comparison of the precision	loating point evaluation yields +	1.5707963·x
	Для «реального часу» з	за замовчанням м	антиса складає 20
	знаків. Уточнення пров	одиться введенням	и через кому після
			float 6
	x-acos(0) fl	oat, $5 \rightarrow 1.5708 \cdot x \cdot a \cos x$	$f(0) \xrightarrow{f(0)ac} 0 1.5708 \cdot x$

Таблиця 4.10. Продовження

Дія	«Швидкий» спосіб	«Live»	«Live»
		MathCAD	MathCAD Prime

**factor** – розкладання виразу на множники.

Перетворення потребує уточнюючого слова **factor**. Для «швидких» обчислень проводиться відповідним пунктом меню після введення виразу в документ. Для «реального часу» проводиться кнопкою символьної панелі або введенням слова з клавіатури в універсальний шаблон та може застосовуватися до виразів і функцій користувача.

$$\begin{split} f(a,b,y) &:= a \cdot y^4 - a \cdot y^3 - 2 \cdot a \cdot b \cdot y^3 + 2 \cdot a \cdot b \cdot y^4 & f(a,b,y) & by factoring, yields & f(a,b,y) \\ f(a,b,y) factor &\to a \cdot y^3 \cdot (y-1) \cdot (2 \cdot b + 1) \\ & f(a,b,y) \frac{factor}{\longrightarrow} a \cdot (2 \cdot b + 1) \cdot (y-1) \cdot y^3 \end{split}$$

Перетворення надає практичний результат для залежностей, які мають раціональні корені. В іншому випадку як результат виводиться сам вихідний вираз. Для отримання практичного результату в способі «реального часу» дія може уточнюватися додаванням значення потрібного кореня через кому після ключового слова. Для «швидких» перетворень можливість не реалізовано. Наприклад:

$$x^2$$
 - 5 factor  $\rightarrow x^2$  - 5  $x^2$  - 5 factor,  $\sqrt{5} \rightarrow (x - \sqrt{5}) \cdot (x + \sqrt{5})$ 

 $x^2 - 5 \xrightarrow{factor, \sqrt{5}} (-\sqrt{5} + x) \cdot (\sqrt{5} + x)$ 

Розкладанням на множники можна визначити, чи є ціле число простим. Просте число матиме два множники: одиницю та саме число. Непросте числі матиме більше множників. Наприклад. Число 113 є простим, розкладання <sup>113 factor  $\rightarrow$  113</sup> показує множники 1, 113. Число 2022 не є простим, бо розкладання  $2022 \xrightarrow{factor} 2 \cdot 3 \cdot 337$  показує множники 1, 2, 3, 337.

Дія	«Швидкий» спосіб	«Live»	«Live»
		MathCAD	MathCAD Prime

**collect** – перетворення алгебраїчного виразу на степеневий поліном, доданки якого розміщено в порядку зниження степеню аргументу.

Перетворення потребує уточнюючого слова **collect**. Для «швидких» обчислень проводиться відповідним пунктом меню після введення виразу в документ. Для «реального часу» проводиться кнопкою символьної панелі або введенням слова з клавіатури в універсальний шаблон та може застосовуватися до виразів і функцій користувача.

$$(x-5)\cdot(2-x)^2 + 3 \qquad (x-5)\cdot(2-x)^2 + 3 \text{ collect } \rightarrow x^3 - 9\cdot x^2 + 24\cdot x - 17$$
  
by collecting terms, yields  
$$x^3 - 9\cdot x^2 + 24\cdot x - 17$$
  
$$(x-5)\cdot(2-x)^2 + 3 \xrightarrow{collect} x^3 - 9\cdot x^2 + 24\cdot x - 17$$

Для функцій кількох змінних або за потреби залишити в степеневому поліномі якісь операнди аргумент степеню може бути уточнений. Для «швидких» обчислень уточнення проводиться охопленням синім кутом курсору потрібної частини виразу. Для «реального часу» уточнення проводиться введенням через кому після ключового слова імені змінної або довільного операнду. Наприклад,

уточнення розкладання за змінними **х** або **у** для функції двох змінних:

$$(2 - x)^2 \cdot (y - 5) + 3 \text{ collect}, x \rightarrow (y - 5) \cdot x^2 + (20 - 4 \cdot y) \cdot x + 4 \cdot y - 17$$
  
 $(2 - x)^2 \cdot (y - 5) + 3 \text{ collect}, y \rightarrow (x - 2)^2 \cdot y + 3 - 5 \cdot (x - 2)^2$ 

уточнення розкладання за операндом **х-1**:

 $(2 - x)^2 \cdot (x - 5) + 3 \text{ collect}, x - 1 \rightarrow (x - 1)^3 - 6 \cdot (x - 1)^2 + 9 \cdot (x - 1) - 1$ 

Дія	«Швидкий» спосіб	«Live»	«Live»
		MathCAD	MathCAD Prime

**coeffs** – символьний розрахунок коефіцієнтів степеневого поліному перетворення алгебраїчного виразу.

Перетворення потребує уточнюючого слова **coeffs**. Для «швидких» обчислень проводиться відповідним пунктом меню після введення виразу в документ. Для «реального часу» проводиться кнопкою символьної панелі або введенням слова з клавіатури в універсальний шаблон.

розрахунок Результатом дії є вектор-стовпець, в якому за збільшенням коефіцієнтів ступеню. Перший елемент містить коефіцієнт аргументу в нульовому степеню, останній – в найбільшому. Для «реального часу» результуючий вектор може бути присвоєний змінній пакету для подальшого використання.

$$\begin{array}{c} (2-x)^2 \cdot (x-5) + 3 \\ \text{has coefficients} \\ 1 \end{array} \right) \qquad \mathbf{a} := (2-x)^2 \cdot (x-5) + 3 \text{ coeffs} \rightarrow \begin{pmatrix} -17 \\ 24 \\ -9 \\ 1 \end{pmatrix} \qquad \mathbf{a} = \begin{pmatrix} -17 \\ 24 \\ -9 \\ 1 \end{pmatrix}$$

За потреби після слова **coeffs** через кому можна додати уточнююче слово **degree**. В такому випадку вектор результату перетворюється на матрицю. До нього додається другий стовпець, який містить відповідні степені поліному.

$$a := (x-5) \cdot (2-x)^2 + 3 \xrightarrow{coeffs, degree} \begin{bmatrix} -17 & 0 \\ 24 & 1 \\ -9 & 2 \\ 1 & 3 \end{bmatrix} a = \begin{bmatrix} -17 & 0 \\ 24 & 1 \\ -9 & 2 \\ 1 & 3 \end{bmatrix}$$

**complex/rectangular** – переведення комплексного виразу в алгебраїчну форму.

Перетворення потребує уточнюючого слова. Для «швидких» обчислень проводиться пунктом меню **complex** після виділення виразу в документі. Для «реального часу» проводиться кнопкою символьної панелі або введенням слова **rectangular** з клавіатури в універсальний шаблон.

```
e^{x+2i}
evaluation over the complex plane yields

\cos(2) \cdot e^{x} + e^{x} \cdot \sin(2) \cdot i
e^{x+2i} rectangular \rightarrow \cos(2) \cdot e^{x} + e^{x} \cdot \sin(2) \cdot i
e^{x+2i} \xrightarrow{rectangular} \cos(2) \cdot e^{x} + \sin(2) \cdot e^{x} \cdot 1i
```

Дія	«Швидкий» спосіб	«Live»	«Live»
		MathCAD	MathCAD Prime

series – розкладання функції в ряд Тейлора.

Перетворення потребує уточнюючого слова **series**. Для «швидких» обчислень проводиться відповідним пунктом меню після виділенні в виразі синім кутом курсору необхідної змінної в виразі. Для «реального часу» проводиться кнопкою символьної панелі або введенням слова з клавіатури в універсальний шаблон та може застосовуватися до виразів і функцій користувача.

Для «швидких» обчислень кількість доданків визначається в спливаючому діалоговому вікні



Для «реального часу» за замовчанням розкладання проводиться в ряд Маклорена, тобто в околі точки 0 до 5 степеню.

 $\frac{\sin(x)}{x} \quad \text{converts to the series} \quad 1 - \frac{x^2}{6} + \frac{x}{120} \quad \frac{\sin(x)}{x} \text{ series } \rightarrow 1 - \frac{x^2}{6} + \frac{x}{120} \quad \frac{\sin(x)}{x} \quad \frac{\operatorname{series}}{x} - \frac{1}{6} \cdot x^2 + \frac{1}{120} \cdot x^4 = \frac{$ 

Кількість доданків в розкладанні та координата точки можуть бути змінені введенням числового значення та імені змінної через кому після ключового слова. Наприклад, уточнення розкладання в околі точки 2 до третього степеню поліному виглядає наступним чином:

 $\frac{\sin(x)}{x} \text{ series}, x = 2, 3 \rightarrow \frac{\sin(2)}{2} + (x - 2) \cdot \left(\frac{\cos(2)}{2} - \frac{\sin(2)}{4}\right) - (x - 2)^2 \cdot \left(\frac{\cos(2)}{4} + \frac{\sin(2)}{8}\right)$  $\frac{\sin(x)}{x} \xrightarrow{\text{series}, 3, x = 2}{3} \xrightarrow{\sin(2)}{2} + \frac{2 \cdot \cos(2) - \sin(2)}{4} \cdot (x - 2) - \frac{\sin(2) + 2 \cdot \cos(2)}{8} \cdot (x - 2)^2$ 

	«Швидкий» спосіб	«Live»	«Live»
Дія		MathCAD	MathCAD Prime

substitute – заміна змінної в символьному виразі.

Перетворення потребує уточнюючого слова **substitute**. Для «швидких» обчислень потребує попереднього введення та копіювання в буфер обміну виразу нової змінної, після чого проводиться пунктом меню **complex** з виділенням змінної для заміни в виразі в документі. Для «реального часу» проводиться кнопкою символьної панелі або введенням слова **substitute** з клавіатури в універсальний шаблон. Змінна для заміни додається через кому після ключового слова.

Прямим призначенням перетворення є аналіз поведінки функцій за різних умов.

Заміна «швидким» способом в виразі  $(sin(x)+y)^2 \cdot (cos(x)+y)^2$ змінної **у** на змінну **x**<sup>2</sup> складаються з двох кроків. На першому кроці в документ вводиться змінна **x**<sup>2</sup> та зберігається в буфері обміну. На другому кроці курсор встановлюється на змінну та в меню обирається пункт **Symbolics-Variable-Substitute**:

 $(\sin(x) + y)^2 \cdot (\cos(x) + y)^2$  by substitution, yields  $(\cos(x) + x^2)^2 \cdot (\sin(x) + x^2)^2$ 

Для «реального часу» заміна проводиться введенням уточнюючого запису **у=x**<sup>2</sup> через кому після ключового слова **substitute**:

$$(\sin(x) + y)^2 \cdot (\cos(x) + y)^2$$
 substitute,  $y = x^2 \rightarrow (\cos(x) + x^2)^2 \cdot (\sin(x) + x^2)^2$ 

 $(\sin(x)+y)^2 \cdot (\cos(x)+y)^2 \xrightarrow{substitute, y=x^2} (\cos(x)+x^2)^2 \cdot (\sin(x)+x^2)^2$ 

Для «реального часу» значення символьної залежності виводиться в числовому вигляді за умови введення через кому після ключового слова уточнюючого запису надання змінній числового значення.



Перетворення rewrite для виразу гіперболічної функції через експоненційну:

 $1 + \sinh(x)^2 \text{ rewrite}, \exp \rightarrow \left(\frac{e^x}{2} - \frac{e^{-x}}{2}\right)^2 + 1 \qquad 1 + \sinh(x)^2 \xrightarrow{\text{rewrite}, \exp} \frac{e^{2 \cdot x} + e^{-(2 \cdot x)} + 2}{4}$ 

Перетворення **combine** для спрощення зворотної тригонометричної функції:

 $\begin{aligned} & \operatorname{atan}(\mathbf{x}) + \operatorname{atan}(\mathbf{y}) \text{ combine, atan } \to -\operatorname{atan}\left[\frac{1}{(\mathbf{x} \cdot \mathbf{y} - 1)} \cdot (\mathbf{x} + \mathbf{y})\right], \\ & \operatorname{atan}(x) + \operatorname{atan}(y) \xrightarrow{combine, \operatorname{atan}} -\operatorname{atan}\left(\frac{y + x}{x \cdot y - 1}\right) \end{aligned}$ 

Перетворення combine для спрощення гіперболічної функції:

 $\begin{aligned} \sinh(\mathbf{x})\sinh(\mathbf{y}) \text{ combine, sinhcosh } &\to \frac{\cosh(\mathbf{x}+\mathbf{y})}{2} - \frac{1}{2} \cdot \cosh(\mathbf{x}-\mathbf{y}) \\ \sinh(\mathbf{x})\cdot\sinh(\mathbf{y}) \xrightarrow{combine, sinhcosh} \frac{\cosh(\mathbf{y}+\mathbf{x}) - \cosh(\mathbf{y}-\mathbf{x})}{2} \end{aligned}$ 

#### Ланцюги перетворення

Вбудовані символьні функції не завжди надають потрібного результату за один крок. Часто отриманий на першому кроці перетворень результат доводиться додатково спрощувати, робити в ньому підстановки, перетворювати на числовий тощо.

Такі дії можна провести послідовно вручну. Для цього треба в буфер пам'яті скопіювати результат. В документ ввести шаблон наступної дії, в який в маркер вхідного виразу з буфера вставити попередній результат. Можна здійснити наступне перетворення з використанням результату попереднього перетворення в якості аргументу.

Зменшити витрати на проведення кількох символьних перетворень дозволяє в способі «реального часу» поєднання дій в один блок ланцюга перетворень. Додавання наступної дії можна провести тільки з панелі символьних дій. Для додавання слід встановити курсором на останнє слово попередньої дії та клікнути на кнопці обраної наступної дії.

Наприклад, при перевірці символьного розв'язання рівняння функцією substitute, яка наведена в табл. 4.9, потрібні були дві операції – підстановки та спрощення:

$$\int_{0}^{x} \sin(t) dt \xrightarrow{substitute, x = a\cos(1-a)} -\cos(a\cos(a-1)-\pi) + 1 \xrightarrow{simplify} a$$

Блок у вигляді ланцюга виглядає компактніше:

$$\int_{0}^{x} \sin(t) dt \xrightarrow{simplify} a$$

Наприклад, результат функції **rewrite**, яка наведена в табл. 4.9,

$$1 + \sinh(x)^2 \xrightarrow{rewrite, \exp} \frac{e^{2 \cdot x} + e^{-(2 \cdot x)} + 2}{4}$$

стає більш практичним з додатковим спрощенням:

 $1 + \sinh(x)^{2} \xrightarrow{rewrite, \exp} \cosh(x)^{2}$ 

Результат розкладання функції **sinc** в ряд Тейлора з трьома доданками в околі точки 2 наведено в табл. 4.9:

$$\frac{\sin(x)}{x} \text{ series, } x = 2,3 \rightarrow \frac{\sin(2)}{2} + (x-2) \cdot \left(\frac{\cos(2)}{2} - \frac{\sin(2)}{4}\right) - (x-2)^2 \cdot \left(\frac{\cos(2)}{4} + \frac{\sin(2)}{8}\right)$$

Отримати числове значення коефіцієнтів розкладання можна блоком series та float:

Отримати числове значення коефіцієнтів розкладання та ряду в точці x=2 можна блоком series, float та substitute:

$$\frac{\sin(x)}{x} \begin{vmatrix} \sec i \sec 3, x = 2 \\ \ substitute, x = 2 \rightarrow 0.4546 \\ \ float, 4 \end{vmatrix} \xrightarrow{series, 3, x = 2} \\ substitute, x = 2 \rightarrow 0.4546 \\ \ \frac{\sin(x)}{x} \xrightarrow{float, 4} \\ 0.4546 \end{vmatrix}$$

Результати символьних перетворень «реального часу» залежать від значень змінних, які вже введені в документ. В співпадіння імен в виразі

символьного перетворення та зовнішніх змінних результат символьного перетворення може виявитись не символьним, а числовим.

Перетворення **explicit** дозволяє виключити такі ситуації. Аргументами перетворення є імена змінних, значення яких потрібно взяти ззовні. Імена вводяться через кому після ключового слова. Застосування ключового слова без уточнення виконує перетворення вважаючи всі змінні виразу символьними.

Наприклад, для змінних a=1.1, b=2.2,  $c=sin(0.8\pi)$ , результат символьного обрахування буде числовим, бо процесор знайде числові значення для змінних з іменами перетворення **a**, **b**, **c**::

 $a + b + c \rightarrow 3.88778525229247312917$ 

Для того, щоб зробити символьною змінну с та числовими зовнішніми змінні а, b слід застосувати explicit з аргументами а, b:

```
a + b + c explicit, a, b \rightarrow 1.1 + 2.2 + c
```

щоб вважати всі змінні символьними слід застосувати **explicit** без аргументів:

 $a + b + c explicit \rightarrow a + b + c$ 

Перетворення може застосовуватися в блоці ланцюгових перетворень. Наприклад, для того, щоб отримати символьну відповідь виразу **x<sup>3</sup>-3x+2** розкладання залежності на множники за умови наявності зовнішньої змінної **x=1**:

 $x \coloneqq 1 \qquad x^3 - 3x + 2 \text{ factor } \rightarrow 0 \qquad x^2 - 3x + 2 \quad \begin{vmatrix} explicit \\ factor \\ \end{pmatrix} (x-1) \cdot (x-2)$ 

#### Обчислення за додаткових умов

Часто результат символьних перетворень не задовольняє практичних вимог або взагалі перетворення за замовчанням не можуть бути проведені. Наприклад, в наведених вище прикладах символьного розрахунку добутків були отримані наступні результати:

$$\prod_{k=1}^{\infty} \frac{1}{k^3 + 1} \rightarrow \Gamma\left(\frac{1}{2} - \frac{\sqrt{3} \cdot i}{2}\right) \cdot \Gamma\left(\frac{1}{2} + \frac{\sqrt{3} \cdot i}{2}\right) \cdot \lim_{k \to \infty} \frac{1}{k! \cdot \Gamma\left(k - \frac{1}{2} - \frac{\sqrt{3} \cdot i}{2}\right) \cdot \Gamma\left(k - \frac{1}{2} + \frac{\sqrt{3} \cdot i}{2}\right)}$$



Схожа ситуація виявляються при спробі спростити наступний вираз:

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

Для подолання подібних випадків потрібно додати умови, які математично уточнюють умови розрахунку.

В «реальному часі» описує додаткові умови ключове слово **assume**. Для операцій обчислення ключове слово вставляється в маркер дії універсального шаблону, для інших перетворень слово **assume** в блок ланцюга перетворень після ключового слова основної дії.

Умови можуть накладатися на області значень змінних або тип змінних. Значення обмежуються логічних операторами >, <, ≥, ≤, =, a<x< b (RealRange(a,b)). Тип змінних визначається належністю до можливих типів real, integer, complex.

Умови значень вводяться з клавіатури через кому після ключового слова. Типи змінних вводяться кнопками панелі **Modifier** (рис. 4.108) для *MathCAD* та панелі **Math-Symbolics** (рис. 4.104 б) стрічки для *MathCAD Prime*.

Modifier		×
integer	real	RealRange
complex	fully	

Рис. 4.108. Панель типів змінних

Декілька умов записуються через кому. Якщо умова стосується всіх змінних виразу, застосовується службове слово **ALL**. Наприклад,

для отримання практичних результатів наведених вище прикладів можна застосувати наступні обмеження;

$$\begin{split} &\prod_{k=0}^{m} (n+k) \text{ assume, } n > 0 \rightarrow \frac{\Gamma(m+n+1)}{\Gamma(n)} \\ &\prod_{k=0}^{m} (n+k) \xrightarrow{assume, ALL > 0} \frac{\Gamma(n+m+1)}{\Gamma(n)} \\ &\sqrt{x^2} \quad \left| \substack{\text{simplify} \\ \text{assume, } x = \text{real}} \right| |x| \qquad \sqrt{x^2} + \sqrt{y^2} + \sqrt{z^2} \quad \left| \substack{\text{simplify} \\ \text{assume, } ALL > 0} \right| x + y + z \\ &\text{assume, } ALL > 0 \\ \end{split}$$

#### Контрольні запитання

- 1. Що таке символьні обчислення?
- 2. Які способи виконання символьних обчислень реалізовані в системі?
- 3. Як визначається вигляд запису при «швидких» перетвореннях?
- 4. Перерахуйте команди, які запускаються з панелі Symbolic.
- 5. Як згруповані дії в меню Symbolics?
- 6. Яким чином в «реальному часі» провести інтегрування, диференціювання?
- 7. В чому особливість спрощення дійсних чисел?
- 8. Для яких чисел застосовується розкладання на множники?
- 9. Для чого проводиться заміна змінної?
- 10.В чому різниця дій функції collect та coeffs?
- 11. Як визначається кількість доданків в розкладанні в ряд Тейлора?
- 12.Що таке ланцюги обчислень?
- 13.Для якого способу призначена функція **explicit**?
- 14.Для чого призначена функція assume?

#### 4.10. Обмін даними

При проведенні технічних розрахунків та наукових досліджень часто виникає ситуація, коли вхідні дані формуються одними застосунками, обробляються другими, звіт з результатами оформлюється третіми. Типовим є використання засобів офісних систем для оформлення звітів та засобів СКМ-СКА для обробки даних. «Зовнішня» інтеграція застосунків в інформаційних технологіях називається *імпортом-експортом*.

В пакетах *MathCAD* обмін для обміну даними можуть бути застосовані вбудовані функції файлового обміну та технології зв'язування та передавання об'єктів (*Object Linking and Embedding*). Вбудовані функції пакету дозволяють проводити обмін даними на кшталт векторів-стовпців, векторів-рядків, матриці з текстовими *ASCII* файлами з розділенням даних пробілами, знаком табуляції, комою (формат .*CSV*), з файлами *Excel* формату *Open XML* (файли .*xlsx*).

Вбудованими функціями може бути реалізовано *програмний файловий* обмін, коли функції безпосередньо вставляються в документ або *інтерактивний обмін*, коли обмін проводиться вбудованими інтерактивними діалоговими віконними засобами діалогу.

## Програмний файловий обмін

## Програмні засоби обміну з текстовими файлами

Імпорт-експорт даних з текстових файлів в пакеті може проводитися наступними вбудованими функціями: WRITEPRN, READPRN, APPENDPRN, READCSV, WRITECSV, WRITEFILE, READFILE.

Вбудовані функції програмного обміну даними з текстовими файлами наведено в таблиці 4.11.

Тип файлу	Функція імпорту	Функція експорту
Текстові ASCII файли з чисельними даними, які розділені пробілами	READPRN	WRITEPRN APPENDPRN
Текстові ASCII файли з чисельними та текстовими даними	READCSV READFILE	WRITECSV WRITEFILE

Таблиця 4.11. Вбудовані функції обміну з текстовими файлами

Деякі функції мають однаковий синтаксис в усіх версіях пакета, Деякі відрізняються для *MathCAD* та *MathCAD Prime*.

Функція **M:=READPRN(file)** зчитує з текстового ASCII файлу з розділенням пробілами або знаками табуляції, який визначений обов'язковим аргументом **file**, дані.

Обов'язковий аргумент file є текстовим рядком в лапках з повною специфікацією файлу. Наприклад, "c:\user\gatafile.dat".

Зчитані в масив м дані можуть бути вектором або матрицею.

Функція **var** : = **APPENDPRN (file, M)** дописує вектор або матрицю **M** з розділенням пробілами після останнього рядка існуючих даних в текстовий файл **file** формату ASCII.

Кількість символів для елементу даних та кількість символів після коми даних для функцій **WRITEPRN** та **APPENDPRN** визначається значеннями системних змінних **PRNPRECISION** та **PRNCOLWIDTH** відповідно.

```
Функція
M:=READFILE(file, type, <colwidths>, <rows>,...
<cols>, <emptyfill>)
```

зчитує в масив М з текстового файлу file дані одного формату або з розділювачем.

Обов'язковими є аргументи **file** та **type**.

Опціональні параметри уточнюють спосіб зчитування:

- type визначник типу організації даних в файлі у вигляді текстовий рядок в подвійних лапках. Можливі значення: "delimited" для даних з розділювачем даних в рядку комою або пробілом, "fixed" для числових даних сталої ширини
- emptyfill- символ, який замінюватиме в результуючому масиві м незаповнені комірки файлу. Можливі значення: число, текстовий рядок в подвійних лапках, символ NaN. За замовчанням NaN.
- colwidths- розмір числа в символах для файлів типу **fixed** у вигляді вектора-рядка розміром [1× n], де n– кількість даних в рядку
- **dec** (https://support.ptc.com/help/mathcad/en/index.html#page/PTC\_M athcad\_Help/data\_files\_2.html), але на практиці не є працездатним.

#### Функції MathCAD

Функція **var**:=**WRITEPRN (file,M)** записує вектор або матрицю **M** в текстовий файл **file** формату ASCII з розділенням пробілами.

Оператор **WRITECSV (M, file)** записує вектор або матрицю **M** в текстовий файл **file** формату ASCII з розділенням комою (формат .*CSV*).

Функція var:=APPENDFILE (M, file, type) додає вектор або матрицю М до існуючого файлу file. Розділювач визначається аргументом type.

Всі аргументи в наведених функціях *MathCAD* є обов'язковими.

Функція

M:=READCSV (file, <rows>, <cols>, <emptyfill>, <delimiter>) зчитує з текстового файлу file формату ASCII дані з розділителем у вигляді символу, який визначено опціональним аргументом delimiter. Обов'язковими параметрами є M та file.

Опціональні параметри уточнюють спосіб зчитування:

- rows/cols- число або двоелементний вектор-рядок, яке визначає діапазон рядків/стовпців для зчитування. Індекси рахуються з 1 незалежно від значення ORIGIN. Число означає кількість рядків/стовпців, які будуть зчитані з початку файлу. Перший елемент вектора означає індекс першого зчитаного рядка/стовпців, другий елемент – індекс останнього зчитаного рядка/стовпця.
- delimiter- односимвольний текстовий рядок, який визначає символ-розділювач даних. За замовчанням за відсутності аргументу delimiter розділювачем вважається символ коми.

### Функції MathCAD Prime

Функція var:=WRITEPRN (file,M <,row,col>) записує вектор або матрицю M в текстовий файл file формату ASCII з розділенням пробілами. Аргументи є аналогічними функціям *MathCAD*.

Функція **WRITECSV (file, M, <row, col, dec>)** записує вектор або матрицю **М** в текстовий файл **file** формату ASCII з розділенням комою.

Опціональний аргумент **dec** є односимвольним рядком в подвійних лапках, який визначає символ десяткової крапки в числах. Можливими значеннями є кома "," або крапка "." За замовчанням за відсутністю аргументу **dec** символом вважається крапка. Символ десяткової крапки «кома» робить символом-розділювачем **delimeter** символ крапки з комою ";".

Інші аргументи є аналогічними функціям *MathCAD*.

Функція **M:=READCSV(file)** зчитує з текстового файлу **file** формату ASCII дані з розділителем у вигляді коми.

Функція **WRITEFILE (file, M)** записує вектор або матрицю **M** в текстовий файл **file** формату ASCII з розділителем у вигляді коми.

Функція

**M=READTEXT (file, <type, delimiter, colwidths, row,... col, emptyfill, blank, options>)** зчитує дані з текстового файлу **file** формату ASCII дані.

Спосіб зчитування визначаються опціональними параметрами:

- rows/cols- текстовий рядок в подвійних лапках, який визначає діапазон рядків/стовпців для зчитування. Один рядок/стовпець задається індексом ("5"), діапазон задається індексами початку та кінця через дефіс ("4-5"). Діапазон від рядка/стовпця до кінця задається індексом з дефісом ("3-"). Індекси можуть задаватися через кому.
- blank- текстовий рядок в подвійних лапках, який визначає дії з незаповненими рядками. Можливі значення: пропустити "skip", зчитати "read" (за замовчанням), зупинитися "stop".
- options- триелементний вектор-стовпець, елементи якого визначають розділювачі. Перший елемент розділювач тексту. Можливі значення: ", ' або **none**. Другий елемент десятковий розділювач. Можливі значення: , або . (за замовчанням). Третій елемент розділювач порядків числа. Можливі значення: . або , або пробіл.

Інші аргументи є аналогічними функціям MathCAD.

Функція **WRITETEXT (file, M,<rows, cols, dec>)** записує вектор або матрицю **M** в текстовий файл **file** формату ASCII з розділителем у вигляді табуляції.

Наведемо приклади використання функцій MathCAD.

Проведемо запис чисельних даних з матриці М в текстовий файл dat.prn з розділенням пробілами та додамо в існуючий файл дані з матриці. Для запису використаємо найпростішу функцію, яка спроможна це зробити, **writeprn**.

$$M := \begin{pmatrix} 1 & 2.2 \\ 3.33333 & 4.4 \end{pmatrix}$$
  
out := WRITEPRN("e:\dat.prn",M)" out := APPENDPRN("e:\dat.prn",M)"  
$$\bigwedge_{mm} := READPRN("e:\dat.prn") = \begin{pmatrix} 1 & 2.2 \\ 3.333 & 4.4 \\ 1 & 2.2 \\ 3.333 & 4.4 \end{pmatrix}$$

Дані в записаному файлі відділені пробілами. Елементи A<sub>2,1</sub> та A<sub>4,1</sub> записані з трьома знаками після коми:

1	2.2
3.333	4.4
1	2.2
3.333	4.4

Такий формат даних в файлі пояснюється тим, що за замовчанням під час запису змінна **PRNCOLWIDTH**, яка визначає загальну ширину поля даних дорівнювала 8, змінна **PRNPRECISION**, яка визначає кількість символів в числі, дорівнювала 4.

Дані функцією **READPRN** зчитуються в тому вигляді, який мають в файлі.

$$NS := READCSV("e:\dat.pm") NN := READCSV\left["e:\dat.pm", \binom{2}{2}, 1, "N", ""\right]$$
$$NS = \begin{pmatrix} "1 & 2.2" \\ "3.333 & 4.4" \\ "1 & 2.2" \\ "3.333 & 4.4" \end{pmatrix} NN = (3.333 & 4.4)$$

NR := READFILE("e:\dat.prn","delimited")

$$NR = \begin{pmatrix} 1 & 2.2 \\ 3.333 & 4.4 \\ 1 & 2.2 \\ 3.333 & 4.4 \end{pmatrix}$$

Зчитування з того ж файлу функцією **READSCV** з параметрами за замовчанням не можна вважати успішним. Функція за замовчанням вважає розділювачем символ кому, тому результат **NS** є вектором-стовпцем 4х1.

Елементами вектора є текстові рядки, які відповідають рядкам в файлі. Визначення символу-розділювача в аргументах функції **READSCV** виправляє ситуацію. Результат зчитування **NN** другого рядка з файлу є вектором-рядком з двома чисельними елементуми. Значення елементів відповідають файловим даним.

Зчитування того ж файлу функцією **READFILE** з уточненням в аргументах функції типу файлів з розділювачем забезпечує вірний результат.

Проведемо зчитування даних з текстового файлу, в якому містяться неоднорідні дані. Перший рядок – текстові назви. Інші рядки містять текст, цілочисельні дані, дійсні числа у фіксованій та експоненційній формі, комплексні числа, службові слова Inf, NaN:

Level	Data1	Data2	Res1	Corr
Level1	12	1.23e4	Inf	Yes
Level2	45.11	16.1+3i	2e6	No
Level3	34.12	51.5-2.1	i NaN	Yes

Зчитування функцією **READPRN** є безрезультатним не дивлячись на те, що файл є текстовим та розділювачем є пробіл.

Зчитування з того ж файлу функцією **READSCV** з параметрами за замовчанням не можна вважати успішним. Результат **NS** є вектором-стовпцем 5х1. Елементуми вектора є текстові рядки, які відповідають рядкам в файлі. Визначення символу-розділювача в аргументах функції **READSCV** покращує ситуацію. Результат зчитування **NSD** є матрицею 5х5. Дані першого рядка розпізнані вірно. В інших рядках розпізнані текстовими комплексні числа, службове слово **Inf**. Доданий зайвий п'ятий рядок з текстовими позначками нерозпізнаних даних.

Зчитування того ж файлу функцією **READFILE** з уточненням в аргументах функції типу файлів з розділювачем забезпечує вірний результат. Розпізнані цілочисельні, дійсні, комплексні значення та текст.

Наведемо приклади використання функцій MathCAD Prime.

Проведемо запис першого рядка даних з матриці М в текстовий файл dat.prn з розділенням пробілами та додамо в існуючий файл дані з матриці. Для запису використаємо найпростішу функцію, яка спроможна це зробити, WRITEPRN.

$$\begin{split} M \coloneqq \begin{bmatrix} 1 & 2.2 \\ 3.333333 & 1.5 \cdot 10^{-6} \end{bmatrix} \quad out \coloneqq \text{WRITEPRN} \begin{pmatrix} \text{``e:\dat.prn''}, M, \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix} \end{pmatrix} \\ out \coloneqq \text{APPENDPRN} \begin{pmatrix} \text{``e:\dat.prn''}, M \end{pmatrix} \end{split}$$

Дані в записаному файлі відділені пробілами. Елементи A<sub>1,1</sub> та A<sub>3,1</sub> записані з трьома знаками після коми:

3.333 1.5e-06 1 2.2 3.333 1.5e-06

Змінити формат даних в файлі неможливо.

Дані функцією **READPRN** зчитуються в тому вигляді, який мають в файлі.

Рядок  $a \coloneqq \text{WRITECSV}\left(s, M, \begin{bmatrix} 2\\2 \end{bmatrix}, 1, ", "\right)$ 

записує в текстовий файл із специфікацією, яка задана змінною **s**, дані з матриці **M**. Записуються всі стовпці другого рядка. Десятковим розділювачем є кома, розділювачем даних – крапка з комою:

3,333333;1,5e-06

Проведемо зчитування даних з текстового файлу, в якому містяться неоднорідні дані. Перший рядок – текстові назви. Інші рядки містять текст, цілочисельні дані, дійсні числа у фіксованій та експоненційній формі, комплексні числа, службові слова Inf, NaN:

Level	Data1	Data2	Res1	Corr
Level1	12	1.23e4	Inf	Yes
Level2	45.11	16.1+3i	2e6	No
Level3	34.12	51.5-2.1i	L NaN	Yes

Зчитування функцією **READPRN** є безрезультатним не дивлячись на те, що файл є текстовим та розділювачем є пробіл.

Зчитування з того ж файлу функцією **READSCV** з параметрами за замовчанням не можна вважати успішним. Результат **NS** є вектором-стовпцем 4x1, елементуми якого є текстові рядки, які містять рядки файлу.

Зчитування того ж файлу функцією **READFILE, READTEXT** з уточненням в аргументах функції типу файлів з розділювачем забезпечує вірний результат. Розпізнані цілочисельні, дійсні, комплексні значення, **NaN** та текст.

			1			
	["Lev	el Data1 I	Data2 Re	es1 Corr"		
	"L	$\operatorname{evel1}12$	1.23e4 In	f Yes"		
$\operatorname{READCSV}(s) =$	"Le	$vel2 \ 45.11$	16.1+3i 2	2e6 No"		
	VaN Yes"					
			<b>66 ?</b> ?			
$NF \coloneqq \operatorname{READFIL}$	E(s, "c	lelimited"	, 1, 1)			
		Level"	"Data1"	"Data2"	"Res1"	"Corr"]
	NF =	"Level1"	12	$1.23 \cdot 10^4$	"Inf"	"Yes"
		"Level2"	45.11	16.1 + 3i	$2 \cdot 10^6$	"No"
		"Level3"	34.12	51.5 - 2.1i	NaN	"Yes"

Рядок

$$NR \coloneqq \text{READTEXT} \left( s, \text{``delimited''}, \text{``}, \text{''}, \text{`'}2-\text{''}, \text{`'}2,4\text{''}, \text{``N''}, \text{``read''}, \left| \begin{array}{c} \dots \\ \dots \\ \dots \\ \dots \\ \dots \end{array} \right)$$

проводить зчитування в матрицю даних з текстового файлу. Вважається, що в файлі застосований розділювач у вигляді коми. Зчитуються дані з другого рядка по останній рядок, другий та четвертий стовпці. Нерозпізнані елементи замінюються на символ "**N**". Порожній рядок зчитується. Ознакою тексту вважається апостроф, Десятковим розділювачем є крапка. Розряди тисяч відокремлюються пробілом.

Функції запису **WRITECSV**, **WRITEFILE**, **WRITETEXT** записують в файл текст в лапках. Наприклад, запис матриці **M** має наступний вигляд:

```
"Level","Data1","Data2","Res1","Corr"
"Level1",12,12300,"Inf","Yes"
"Level2",45.11,16.1 + 3i,2000000,"No"
"Level3",34.12,51.5 - 2.1i,NaN,"Yes"
```

Текст, який є забраний в лапки функція зчитує некоректно, але функції **READFILE**, **READTEXT** однаково вірно розпізнають текст в лапках та без лапок:

```
\begin{aligned} \text{READCSV}(s1) = \begin{bmatrix} \text{``Level'''Data1'''Data2''''Res1''''Corr''''} \\ \text{``Level1 12 12300 ''Inf'' ''Yes''''} \\ \text{``Level2 45.11 16.1 + 3i2.1e-05 ''No''''} \\ \text{``Level3 34.12 51.5 - 2.1i NaN ''Yes''''} \end{bmatrix} \\ \\ \text{READFILE}(s1, \text{``delimited''}) = \begin{bmatrix} \text{``Level'' ``Data1'' ``Data2'' ``Res1'' ``Corr''} \\ \text{``Level1'' 12 1.23 \cdot 10^4 ``Inf'' ``Yes''} \\ \text{``Level2'' 45.11 16.1 + 3i 2.1 \cdot 10^{-5} ``No''} \\ \text{``Level3'' 34.12 51.5 - 2.1i NaN ''Yes''} \end{bmatrix} \\ \\ \text{READTEXT}(s1) = \begin{bmatrix} \text{``Level'' ``Data1'' ``Data2'' ``Res1'' ``Corr''} \\ \text{``Level1'' 12 1.23 \cdot 10^4 '`Inf'' ``Yes''} \\ \text{``Level1''' 12 1.23 \cdot 10^4 '`Inf'' ``Yes''} \\ \text{``Level2''' 45.11 16.1 + 3i 2.1 \cdot 10^{-5} ``No''} \\ \text{``Level2''' 45.11 16.1 + 3i 2.1 \cdot 10^{-5} ``No'''} \\ \text{``Level3''' 34.12 51.5 - 2.1i NaN ''Yes''} \end{bmatrix} \end{aligned}
```

### Програмні засоби обміну з MS Excel

Синтаксис функції програмного імпорту даних з файлу **file** *MS Excel* формату *.xlsx* є однаковим в *MathCAD* та *MathCAD* Prime:

```
M=READEXCEL(file, range <, emptyfill>).
```

Аргумент **range** є текстовим рядком, який визначає діапазон даних. Записується у форматі *MS Excel*. Наприклад, "**Sheet1!A1:B3**". Аркуш книги замість імені може визначатися своїм номером в квадратних дужках. Наприклад, "[1]A1:B3". Одна адреса означає зчитування однієї вказаної комірки.

#### MathCAD

Оператор **WRITEEXCEL (M, file, range)** записує дані з масиву М в файл *MS Excel*.

#### **MathCAD Prime**

Оператор WRITEEXCEL (file, M <, rows, cols, range>) записує дані з масиву М в файл *MS Excel*.

Опціональний аргумент **rows/cols** визначає діапазон рядків для запису. Задається числом або вектором аналогічно функціям *MathCAD*.

Наведемо приклади використання функцій MathCAD.

Створимо файл **exdat.xlsx**, який матиме дві однакові сторінки **Лист1** та **Data**. Перший рядок таблиці містить назви стовпців. Рядки даних складаються з текстової інформації, чисел цілого, комплексного та дійсного типів, службових слів **NaN**, **Inf**. Дійсні числа наведені в фіксованому та експоненційному форматах:

	А	В	С	D	E	F	G	Н	I			
1	Level	Date	Data1	Data2	Data3	Res1	Res2	Corr	Compl			
2	Level1	29.09.2021	12.34	45	1.23E+04	Inf	NaN	Yes	5.1+3i			
3	Level2	13.10.2021	45.11	16	2.00E+06	25.30	0.10	No	1.5-2.1i			
4	Level3	19.10.2021	34.12	521	-543.10	12.00	0.10	Yes	6.1+3i			
5												
4	(											

Проведемо зчитування функцією **READEXCEL** з параметрами за замовчанням та з визначенням аркуша книги **Data** та діапазону зчитування в області дійсних чисел **B2:E4**.

```
ME := READEXCEL("e:\exdat.xisx") ME1 := READEXCEL("e:\exdat.xisx", "Data!B2:E4")
```

	("Level"	"Date"	"Data1"	"Data2"	"Data3"	"Res1"	"Res2"	"Corr"	"Compi"
ME =	" Levell"	$4.447  imes 10^4$	12.34	45	$1.23 \times 10^4$	"Inf"	NaN	"Yes"	"5.1 <b>+</b> 3i"
	" Level2"	$4.448 \times 10^4$	45.11	16	$2 \times 10^6$	25.3	0.1	"No"	"1.5-2.1i"
	(" Level3"	$4.449  imes 10^4$	34.12	521	-543.1	12	0.1	"Yes"	"6.1 <b>+</b> 3i" )
	(4.447 ×	10 <sup>4</sup> 12.34 4	15 1.23 ×	104					
ME1	= 4.448 ×	10 <sup>4</sup> 45.11 1	6 2×1	.06					
	(4.449 ×	10 <sup>4</sup> 34.12 5	21 -543	3.1					

Результат показує, що функція **READEXCEL** коректно зчитує цілочисельні, дійсні, текстові дані та обробляє службове слово **NaN**, розуміє адресацію за аркушами книги та комірками таблиці *MS Excel*. Визначимо матрицю M 3x3. Перший рядок матриці містить текстові назви. Останні рядки містять чисельні дані цілого, дійсного, комплексного типу та вираз. Запишемо дані функцією WRITEEXCEL з параметрами за замовчанням та в діапазоні A2:B3, який містить числа, на аркуш Dt в файли *MS Excel*.



Результат є незадовільним. Функція запису **WRITEEXCEL** не прийняла комплексні дані та діапазон.

Змінимо комплексне значення на дійсне та вкажемо діапазон однією адресою.

 $M := \begin{pmatrix} "Data1" & "Data2" & "Data3" \\ 1 & NaN & 2.5 \\ \frac{3}{34} & 1.5 \cdot 10^{-6} & 33.3 \\ b := WRITEEXCEL(M, "e:\wrt1.xlsx") \\ b := WRITEEXCEL(M, "e:\wrt1.xlsx", "Sheet2!A2") \end{pmatrix}$ 

Результатом є запис:

```
        A
        B
        C

        1
        2
        Data1
        Data2
        Data3

        3
        1
        2.5
        4
        0.088235
        1.5E-06
        33.3
        5
        6

        4
        4
        >>
        Sheet1
        Sheet2
        Sheet2
        Sheet3
        4
```

Функції обміну **WRITEEXCEL**, **READEXCEL** працюють тільки з тими типами даних, які обробляє *MS Excel*, вони не сприймають комплексні числа. Діапазон для запису слід задавати однією адресою, яка означає адресу початку вставляння даних в таблицю *MS Excel*. Значення **NaN** записуються в як порожня комірка, вирази перед записом обраховуються. Для запису можна використовувати тільки англомовні імена аркушів книги за замовчанням на зразок **Sheet3**.

Наведемо приклади використання функцій MathCAD Prime.

Проведемо зчитування функцією **READEXCEL** з параметрами за замовчанням та з визначенням аркуша книги **Data** та діапазону зчитування в області дійсних чисел **B2:E4** з файлу **exdat.xlsx**.

```
\begin{aligned} a &\coloneqq \text{READEXCEL} (\text{``e:} \setminus \text{exdat.xlsx''}) \\ a &= \begin{bmatrix} \text{``Level'' ``Date'' ``Data1'' ``Data2'' ``Data3'' ``Res1'' ``Res2'' ``Corr'' ``Compl''} \\ \text{``Level1'' 4.447 \cdot 10^4 12.34 45 1.23 \cdot 10^4 ``Inf'' NaN ``Yes'' 5.1 + 3i \\ \text{``Level2'' 4.448 \cdot 10^4 45.11 16 2 \cdot 10^6 25.3 0.1 ``No'' 1.5 - 2.1i \\ \text{``Level3'' 4.449 \cdot 10^4 34.12 521 - 543.1 12 0.1 ``Yes'' 6.1 + 3i } \end{bmatrix} \\ a &\coloneqq \text{READEXCEL} (\text{``e:} \setminus \text{exdat.xlsx'', ``Data!B2:E4''}) \\ &= \begin{bmatrix} 4.447 \cdot 10^4 12.34 45 1.23 \cdot 10^4 \\ 4.448 \cdot 10^4 45.11 16 2 \cdot 10^6 \\ 4.449 \cdot 10^4 34.12 521 - 543.1 \end{bmatrix} + \end{aligned}
```

Результат показує, що функція **READEXCEL** коректно зчитує всі дані, з комплексним включно, та обробляє службове слово **NaN**, розуміє адресацію за аркушами книги та комірками таблиці *MS Excel*.

Визначимо матрицю M 3x3. Перший рядок матриці містить текстові назви. Останні рядки містять чисельні дані цілого, дійсного, комплексного типу та вираз. Запишемо дані функцією WRITEEXCEL з параметрами за замовчанням та в діапазоні A2:B3, який містить числа, на аркуш Dt в файли *MS Excel*.



Результат є неповним. Функція запису **WRITEEXCEL** не прийняла назву аркуша книги. Вкажемо стандартну назву аркуша.



Функції обміну **WRITEEXCEL**, **READEXCEL** працюють з всіма типами чисел, з комплексними включно. Значення **NaN** записуються в як порожня комірка, вирази перед записом обраховуються. Для запису можна використовувати тільки англомовні імена аркушів книги за замовчанням на зразок **Sheet3**.

#### Інтерактивні засоби обміну

Інтерактивно в *MathCAD* можна проводити обмін з текстовими файлами, файлами баз даних та файлами *MS Excel* засобами імпорту File Input, експорту File Output, універсального обміну Data Wizard.

#### Засоби файлового обміну

Результат передається в змінну документа через створюваний спеціальний компонент. Компонент відображається на екрані у вигляді позначки з іменем зовнішнього файлу даних (рис. 4.109 а). Між компонентом та зовнішнім файлом зберігається зв'язок. Клік на компоненті оновлює передавання даних між документом *MathCAD* та файлом. Пункт **Properties** контекстного меню компоненту відкриває вікно властивостей компоненту. Вигляд вікна властивостей змінюється для текстових файлів та файлів *MS Excel*. Сторінки вікна є подібними до сторінок вікон імпорту-експорту. Для текстових файлів можна відредагувати ім'я файлу, тип файлу text, MATLAB, Microsoft Excel тощо, десятковий розділювач чисел, розділювач даних та діапазон даних для обміну (рис. 4.109 б).

	1	Component Properties	?	Component Properties	?
		File Options Data Range		File Options Data Range	
		File format: Text	~	Row Start at row: Stop at end of data	٥
		Enter the name of the file or data source which will b	e	Stop at row:	1
		associated with this component.		Column	
v :=		G:\data.dat		Start at column:	1
·		Browse		Stop at end of data	
	D:\q.bt	Use comma as decimal symbol		Stop at column:	1
а		б			

Рис. 4.109. Компонент інтерактивного обміну: а – позначка; б – вікно властивостей

В MathCAD Prime залишено один інтерактивний засіб імпорту з MS Excel. Засіб є надбудовою на функцією **READEXCEL**. Результатом є сформований в діалозі виклик функції.

# Засоби MathCAD

Запуск інтерактивних засобів файлового обміну проводиться пунктами **Insert-Data** меню (рис. 4.110).



Рис. 4.110. Вибір інтерактивних засобів файлового обміну

### Імпорт з текстових файлів

Віконний діалог зчитування даних з дискового текстового файлу запускається пунктом Insert-Data-File Input меню. Параметри зчитування визначаються в двох діалогових вікнах. В першому вікні (рис. 4.111 а) зі списку можливих текстових файлів **Text**, файлів *MS Excel*, файлів баз даних dBase III, файлів *Matlab* обирається тип файлу, змінюється за потреби десятковий розділювач в числах з крапки на кому use comma as decimal separator. Друге вікно (рис. 4.111 б) використовується у випадку, коли зчитувати потрібно не весь файл, а частину даних з нього. У вікні визначається діапазон рядків та стовпців даних в файлі для зчитування.



Рис. 4.111. Інтерактивний файловий імпорт: а – вікно визначення файлу; б – вікно діапазону текстових файлів; в – вікно діапазону файлів;

В разі визначення типу файлу як *MS Excel* в першому вікні **File Input** вигляд другого вікна діалогу контекстно змінюється. У вікні (рис. 4.111 в) визначається діапазон комірок в файлі для зчитування. Діапазон комірок визначається у форматі *MS Excel* на зразок **Sheet1!A1:C4**. За наявності в книзі *MS Excel* аркушів з іменами *англійською мовою за замовчанням* та поіменованих блоків комірок аркуш для зчитування можна обрати із списку **Worksheet** блок комірок – зі списку **Named range** вікна.

### Файловий експорт

Віконний діалог експорту даних в дисковий файл запускається пунктом Insert-Data-File Output меню. Параметри запису визначаються в двох діалогових вікнах, аналогічних вікнам діалогу імпорту. Список типів файлів першого вікна додатково містить форматований текст Formatted Text, текст з розділенням табуляцією Tab Delimited Text та текст з розділенням комою Comma Separated Text. Друге вікно використовується у випадку, коли записати в файл потрібно частину масиву. У вікні визначається діапазон рядків та стовпців даних масиву для запису.

Інтерактивний експорт не сприймає імена аркушів книги та адреси на аркуші *MS Excel*.

### Універсальний імпорт

Універсальний імпорт керується додатком Data Import Wisard. Результат універсального імпорту передається в змінну компонентом типу *таблиця*. Використання елементів таблиці в документі не відрізняється від використання елементів матриць. Зображується таблиця як матриця в стилі «таблиця» (рис. 4.112 а).

Пункт **Properties** контекстного меню компоненту відкриває вікно властивостей компоненту. Сторінки вікна є подібними до сторінок вікон додатку **Data Import Wisard**. Можна відредагувати ім'я файлу, формат виведення даних в комірках таблиці подібно до аналогічного вікна форматування результатів обчислення та діапазон даних для обміну. Для текстових файлів додатково можна відредагувати розділювач даних, десятковий розділювач чисел, символ позначки текстової інформації тощо (рис. 4.112 б).

						Component Properties	?	Component Properties	? ×
						File Options Data Range		Display Options Delimited Text Options Text Options Data Range	
						Row Start at row:	1	File format:         Delimited Text         Number Format           File format:         Delimited Text         Image: Comparison of the file which will be i	3 🔹
f :=						Stop at row:	1	associated with this component: Codebaged	
		0	1	2	3			Exponential threshold	3 🖨
	0	"Level"	"Data1"	"Data2"	"Res1"	Column		Complex threshold	10 🜲
	1	"Level1"	12	1.23.104	"Inf"	Start at column:	1	Zero threshold	15 🜩
	2	"Level2"	45.11	16.1+3i	2.1.10-5	Stop at end of data		Display as icon	
	3	"Level3"	34.12	51.5-2.1i					
	4					Stop at column:	1	Show column/row labels	Font
			2				б		

Рис. 4.112. Компонент таблиці універсального імпорту: а – позначка; б – вікно властивостей

Додаток Data Import Wisard запускається пунктом Insert-Data-Data Import Wisard або Insert-Component-Data Import Wisard меню.

Перше вікно діалогу має вигляд, аналогічний стартовому вікну файлового імпорту. Воно призначено задання файлу для імпорту даних та обрання зі списку типу файлу: текстового файлу з розділювачем **Delimited Text**, текстового файлу зі стовпцями фіксованої ширини **Fixed Width Text**, файлу *MS Excel*, файлу баз даних **dBase III**, файлу *Matlab*. бінарного файлу **Binary**.

Вміст вікон наступних кроків діалогу визначається обраним типом файлу та є подібним до вікно імпорту даних табличним процесором *MS Excel*. Діалог текстових файлів має три додаткові вікна, діалог файлів *MS Excel* має одне додаткове вікно. В центрі всіх вікон виводиться таблиця з відображенням даних такими, як їх на поточний час розпізнає пакет.

#### Універсальний імпорт з текстових файлів

Друге вікно **Delimited Text Options** (рис.4.113 а) призначено для визначення символу розділення даних **Delimiter**, рядку в файлі, з якого потрібно починати імпорт, **Starting row**, дій з порожніми рядками **Blank rows**, кількості рядків колонтитулу **Footer rows**.

Третє вікно **Text Options** (рис. 4.113 б) призначено для визначення ознаки текстової інформації **Text qualifier**, символу десяткової точки **Decimal Symbol**, символу розділення порядків числа **Thousand separator**, символу заміни нерозпізнаних символів **Missing value**.

Четверте вікно Data Range (рис. 4.113 в) призначено для визначення діапазону рядків/стовпців для зчитування Read rows, Read columns. Рядки/стовпці задаються відповідними номерами. Діапазон визначається значеннями через дефіс. Наприклад, 1, 3, 5-8.



Рис. 4.113. Вікна універсального імпорту: а – вікно розділювача; б – вікно тексту; в – вікно даних; д – вікно Ехсеl

### Універсальний імпорт файлів MS Excel

Друге вікно **Excel Options** (рис. 4.113 г) призначено для обрання з вбудованих списків дій з порожніми рядками Blank rows, заміни нерозпізнаних символів **Missing value** та для визначення діапазону даних для зчитування **Range**. Діапазон задається повною адресою за типом A1 в синтаксисі. Для спрощення можна скористатися аркушів списком Worksheet, Обраний елемент списку автоматичного вводиться в повну адресу. Діапазон може задаватися іменем поіменованого блоку Named **range**. Наявні на аркуші поіменовані блок відображаються в таблиці даних синім кольором. Обраний елемент списку автоматичного вводиться в рядок адрес діапазону.

Пакет не дозволяє налаштовувати імпорт для файлів формату .xlsx. Зчитування може проводитися тільки всієї таблиці з першого аркуша MS Excel. Повний функціонал налаштувань імпорту працює тільки з файлами формату .xls.

# Засоби імпорту з MS Excel Mathcad Prime

В MathCAD Prime імпортувати таблицю з MS Excel можна в змінну типу таблиця ручним вставлянням (Ctrl+V, **Paste**) через буфер обміну (clipboard). Через буфер обміну з комірок, значення яких розраховані за якимось виразом, передаються тільки результати без збереження формульної залежності. MathCAD Prime розпізнає дійсні числа з десятковим роздільником чисел у вигляді символу крапки. В іншому випадку дійсні числа будуть імпортовані в документ MathCAD Prime як текстові рядки.

Алгоритм дій з інтерактивного формування аргументів функції **READEXCEL** *в MathCAD Prime* має наступний порядок дій:

- ввести ім'я змінної, в яку передбачається імпортування даних, та записати знак прирівнювання;
- кнопкою **Input/Output-READEXCEL** стрічки (рис. 4.114 а) активувати діалогове вікно імпорту;
- в діалоговому вікні (рис. 4.114 б) кнопкою Browse активувати діалог пошуку файлу та обрати файл *MS Excel*. Контролювати ім'я файлу в полі File name вікна;
- задати в форматі MS Excel стилю адрес «А1» діапазон рядків в полі Rows та стовпців в полі Columns. комірок таблиці MS Excel для зчитування. Адреси рядків та стовпців в блоці розділяються дефісом. Таблиця MS Excel відображається в центрі вікна в області Preview. Обрати потрібний діапазон комірок можна позначенням мишею потрібного діапазону комірок в області Preview.

			🥴 READEXCEL						
			File name: Columns: Rows:	E:\inf_t\dat.3 E - E 1 - 1	xlsx				Browse
			Preview:	B Date	C Data1	D Data2	100% (=) E Data3	F Res1	G Res2
ADEXCEL	Excel Component	Insert Object	2 Ser1 3 Ser2 4 Ser3	44468 44452 44488	12.34 45.11 34.1199999	45 16 521	12000 2000000 -43.1	Inf 25.3 12	NaN 0.1 0.1
a Import/Ex	Insert Excel C	Component Expression t Expression	Аркуш1	Аркуш	12		Cancel	)	•

Рис. 4.114. Імпорт READEXCEL: а – виклик; б – діалогове вікно

### Компонент Excel

Компонент Excel MathCAD є способом поєднання застосунків Excel та MathCAD за технологією зв'язування та передавання об'єктів (Object Linking and Embedding) шляхом спільного використання даних. Він поєднує доступ до таблиці робочої книги MS Excel одночасно з MathCAD та з MS Excel.

Необхідною умовою використання компонента є наявність на комп'ютері інстальованого *MS Excel* версії не менше за 2007. Обмін даними через компонент *Excel MathCAD* супроводжується одночасним відкриттям застосунку *MS Excel*.

Компонент *Excel MathCAD* забезпечує приєднання обчислювальних можливостей *MS Excel* у вигляді вбудованих функцій або макросів в спільний потік обчислень з *Mathcad*. З документа *MathCAD* можна надсилати дані в *MS Excel* та отримати назад результати обробки даних.

Можна редагувати та форматувати дані за допомогою всіх інструментів *MS Excel* в документі *MathCAD*. Компонент *Excel MathCAD* створює один аркуш із назвою *Sheet1*. Для посилання на інші аркуші потрібно створити аркуші в *MS Excel*.

Дані з компонента *Excel MathCAD* можна зберегти в *MS Excel* у дисковому файлі *.xlsx* без збереження зв'язку з джерелом даних в компоненті *Excel MathCAD*.

### Структура компонента Excel

Компонент *Excel MathCAD* розділений на три області: *входи* (Inputs), *таблиця*, *виходи* (Outputs).

Дані з області *входів* передаються з *MathCAD* до *MS Excel*. В якості вхідних даних можуть бути використані чисельні значення, змінні у вигляді скалярів, векторів, матриць, текстових рядків. Кожному входу має бути призначено одну комірку або діапазон комірок, які будуть приймати дані *Mathcad*.

Таблиця компоненту Excel MathCAD аналогом таблиці аркуша робочої книги MS Excel. В ній відображається вміст комірок MS Excel, які відповідають даним областей компонента. Подвійний клік на таблиці компонента активує MS Excel та передає йому керування. Зміни в таблиці MS Excel негайно відображаються в таблиці компонента Excel Mathcad.

Дані з області *виходів* повертаються з *MS Excel* в *Mathcad*. Використання даних забезпечується їхнім присвоєнням змінним *Mathcad*.

За замовчанням *входи* та *виходи* співставляються з коміркою A1 першого аркуша робочої книги *MS Excel*. Доки застосунок *MS Excel* відкритий оновлення вхідних та вихідних даних НЕ ВІДБУВАЄТЬСЯ Обчислення проводиться тільки при закритті *MS Excel*. При закритті *MS Excel* компонент послідовно обробляє залежності входів та виходів.

# Компонент Excel Mathcad

Компонент *Excel MathCAD* створюється як компонент типу Microsoft Excel у віконному діалозі компонентів. Діалог активується пунктом Insert-Component головного меню.

Компонент *Excel MathCAD* може бути пов'язаний з новим файлом *MS Excel* або з існуючим. Спосіб пов'язання та ім'я існуючого файлу *MS Excel* визначається в другому вікні діалогу (рис. 4.115 а)

Excel Setup Wizard	$\times$	Excel Setup Wizard	×
Create an empty Excel worksheet  Create from file:  E:\inf_t\dat.xlsx  Browse  Display as Icon		Inputs Starting Cell   1 A1   0utputs 1   8ange 1   1 A1:A1	]
а		б	

Рис. 4.115. Створення компонента *Excel*: а – вибір файлу; б – визначення входів та виходів

Кількість входів та виходів, діапазони відповідних їм комірок таблиці *MS Excel*, визначаються в наступному третьому вікні діалогу. Кількість входів/виходів обирається зі списків **Inputs/Outputs** відповідно. Діапазони входів вводяться в поля таблиці **Starting Cell** вікна, виходів – таблиці **Range** вікна у форматі стилю «A1» повною адресою з вказанням імені аркуша. Для виходів одна комірка повинна визначатися діапазоном.

Компонент *Excel* показується в документі у вигляді таблиці з маркерами входів та виходів (рис. 4.116 а). Нагорі в лівому верхньому куті розташований маркер виходів. Маркер супроводжується знаком прирівнювання. Кілька виходів позначаються як елементи вектора-стовпця. Внизу під таблицею розташований маркер входів. Кілька входів позначаються як елементи вектора-рядка.



Рис. 4.116. Компонент *Excel* : а – вигляд; б – меню налаштувань

Відредагувати існуючий компонента *Excel* можна через контекстне меню компонента (рис. 4.116 б). Пункти Add Input Variable, Add Output Variable, Remove Input Variable, Remove Output Variable додають/видаляють вхід/вихід компонента відповідно. Пункт Properties відкриває діалогове вікно налаштувань діапазонів значень, в якому можна змінити відповідність змінних входів/виходів в *MathCAD* коміркам робочої таблиці *MS Excel*. Поля вікон налаштувань мають вигляд на зображення аналогічне вікнам створення компонента.

### Компонент Excel Mathcad Prime

Робота з компонентом *Excel в MathCAD Prime* суттево зручніша. Створюється компонент пунктом **Input/Output-Excel Component**-**Insert Excel Component** стрічки (рис. 4.111 а). За замовчанням компонент зв'язується з новим порожнім файлом *MS Excel*, для зв'язку з існуючим файлом, він повинен бути відкритий в *MS Excel* під час створення компонента.

Компонент *Excel* показується в документі у вигляді таблиці з областями входів та виходів (рис. 4.117 а). Нагорі розташована область входів. Внизу під таблицею розташована область виходів.



Рис. 4.117. Компонент *Excel MathCAD Prime*: а – вигляд; б – контекстне меню
Входи та виходи визначаються шляхом введення виразів прирівнювання у відповідних областях. Вирази можуть бути введені вручну, пунктами кнопки Import/Export-Excel Component стрічки (рис. 4.111 а), пунктами контекстного меню областей входів та виходів (рис. 4.117 б).

Якщо у відкритому застосунку *MS Excel* перед використанням пункту стрічки виділити діапазон комірок таблиці, то цей діапазон буде вставлений за замовчанням у відповідний вираз компонента *Excel*.

Комірки книги *MS Excel* позначаються індексованою змінною на ім'я *excel*. Індексом змінної є повна адреса комірки в форматі *MS Excel* стилю «A1» у вигляді текстового рядка в парних лапках. (наприклад, "**Sheet! A1**"). Для входів може визначатися одна комірка або блок комірок, для виходів завжди визначається блок комірок. Використання для введення елементів інтерфейсу виводить в область шаблон змінної. За замовчанням адресується комірка A1 для входів, блок A1:A1 – для виходів. Наприклад, *excel*\_sheet1/A1" = 123.

Редагувати таблицю компонента можна лише в *MS Excel*. Відкривається застосунок *MS Excel* подвійним кліком на таблиці компонента.

Закрити застосунок *MS Excel* при роботі з компонентом *Excel* можна пунктом **File-Close** стрічки в *MS Excel* або подвійним кліком на таблиці компоненту *Excel* в документі *Mathcad Prime*. При цьому закривається лише книга компонента *Excel*. Інші відкриті книги *MS Excel* залишаються відкритими.

#### Контрольні запитання

- 1. Яка функція дозволяє імпортувати дані з текстових файлів з довільним розділенням?
- 2. В чому різниця між функціями **READCSV** та **READPRN** ?
- 3. Як обробляються комплексні числа при програмному файловому обміні?
- 4. Як нумеруються комірки діапазонів в функціях імпорту-експорту у випадку їх чисельного визначення?
- 5. Як програмно провести обмін з MS Excel?
- 6. Як оновити інформацію, яку імпортовано інтерактивно як файлову?
- 7. Як визначити аркуш книги при файловому інтерактивному експорті?
- 8. Який тип обміну реалізовано інтерактивно?
- 9. Які засоби імпорту реалізовані в Data Import Wisard?

- 10.Як змінити діапазон входу/виходу компонента Excel?
- 11.Як поводиться компонент *Excel*, який вставлений в колонтитул?
- 12.Коли відбувається синхронізація даних таблиці *MS Excel* та компонента *Excel*?
- 13.Як впливає порядок рядків входів на значення таблиці компонента *MS Excel*?

## 5. CKM MATLAB

#### 5.1. Загальні характеристики

Система *Matlab* (Matrix Laboratory) була розроблена Клівом Молером (Cleve Moler) як засіб доступу до бібліотек програм *LINPACK* і *EISPACK*, призначених для матричних обчислень на великих ЕОМ.



Клів Молер

На початку 80-х років Джон Літл (John Little) з фірми *MathWorks Inc* розробив версії системи PC *Matlab* для комп'ютерів класу IBM PC і Macintosh. Далі були розроблені версії для робочих станцій Sun, комп'ютерів з операційною системою UNIX та інших типів великих і малих EOM.

Система *Matlab* орієнтована на роботу з матрицями, які широко застосовуються в складних математичних розрахунках, наприклад, при розв'язанні задач лінійної алгебри та математичного моделювання статичних та динамічних об'єктів. Матриці є підґрунтям рівнянь стану динамічних систем.

Символьні обчислення в *Matlab* проводяться вбудованою СКА *MuPad*, що робить *Matlab* універсальною СКМ-СКА.

Дії користувача в *Matlab* проводяться командним діалогом та за допомогою вбудованої мови програмування.

*Matlab* має добре розвинені вбудовані засоби візуалізації двовимірних і тривимірних графіків. Тонке налаштування графіків проводиться засобами низькорівневого програмування, яке вимагає розуміння принципів комп'ютерної графіки та володіння прийомами об'єктного програмування.

Вбудована мова програмування та діалогу з користувачем дозволяє створювати комп'ютерні моделі складних технічних систем.

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.

СКА Matlab дозволяє проводити:

- символьне та чисельне розв'язування різноманітних задач математичного аналізу, серед яких – обчислення границь, диференціювання й інтегрування функцій, спектральні перетворення тощо;
- розв'язування звичайних диференціальних рівнянь методами Рунге-Кутта тощо;
- розв'язування задач умовної і безумовної оптимізації лінійного, дискретного та нелінійного програмування;
- розв'язування задач матричної 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);
- обробку експериментальних даних та наближення функцій методами апроксимації, поліноміальної та сплайн інтерполяції;
- обмін та обробку даних у формати ASCII, Binary, Excel, Word, PowerPoint;
- імпортування й експортування бінарної графіки форматів PS, EPS, Bitmap, EMF, GIF, AVI, JPG, TIFF.

Для системи розроблена велика кількість розширень і доповнень, які дозволяють використовувати її не тільки для розрахунковий математичний інструмент, а й застосовувати *Matlab* як засіб 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 задачі, роботу з картографічними зображеннями, нейронні мережі, нечітку логіку тощо.

Близька п'ятдесяти спеціалізованих розширень *Toolbox* можуть бути вибірково встановлені в *Matlab*, наприклад, бібліотека методів чисельної оптимізації *Optimization Toolbox*, бібліотека засобів символьних перетворень *Symbolic Math Toolbox*, бібліотеки засобів для керування модулями

введення/виведення сигналів та зображень Data Acquisition Toolbox, Image Acquisition Toolbox, Instrument Control Toolbox, фінансово-економічні бібліотеки 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, бібліотеки статистичної, регресійної обробки даних Spline Toolbox, Statistics Toolbox, Curve Fitting Toolbox тощо.

Популярний серед інженерів опціональний модуль *Simulink* дозволяє проводити візуальне моделювання частотних і часових характеристик динамічних систем.

Базовий модуль *Simulink* може бути доповнений спеціалізованим засобами моделювання:

- Simscape Electric (panime Fixed-Point Blockset, SimPowerSystems) електричних та енергетичних систем;
- Simscape Multibody (panime SimMechanics) статичних механічних систем;
- Simscape Fluids процесів в рідинах;
- SimBiology біологічних систем;
- Simscape Driveline кінематики механічних систем;
- *Stateflow* кінцевих автоматів.

*Matlab* інтегрується з популярними офісними застосунками та мовами програмування *C, Fortran, Java*.

Модуль Notebook забезпечує передавання даних з Matlab в MS Word та створення інтерактивних документів — "М-книг". Надбудова MS Excel Link забезпечує доступ користувача до функцій Matlab з MS Excel та імпорт таблиць в MS Excel з Matlab.

В Matlab може бути імпортована інформація з таких популярних баз даних, як Microsoft Access, Microsoft SQL Server, Oracle. Для обміну даними використовується мова SQL.

## Оболонка пакета

Програмне вікно *Matlab* (рис. 5.1) має стандартне оформлення застосунків *Microsoft* стилю вище за 2007. Вікно містить *стрічку керування, рядок стану* та *робоче поле*.

HOME PLOTS									
Troning. TEOTO	APPS EDIT	OR PUBLISH				🖧 🔚 🔏 🗞 i	1 9 C 🗗	Search Documentation	Log
tew New New Open Live Script Live Script	Find Files Compare Import San Data Works	e Open Variable Copen Variable Pace Ocean Workspace VARABLE	Analyze Code	Simulink Layout	Preferences     Preferences     Set Path     Parallel      ENVFROMMENT	Add-Ores Help	Community	ан Стрічка	
🔷 📥 🔄 🔄 📘 🕨 D: 🕨 Program	Files + MATLAR + R201	l7h ⊁ hin ⊁							
Current Folder	Editor - D:\Program	Files\MATLAB\R2017b\bin\p	rimer.m				Θx	Workspace	
Маге -     Маге	<pre>phasem % pri 1 - function 2 - n=length ( 3 - Cor i=lin 4 - y(1)= 8 - end MINB</pre>	menm % + (y)=primer(a,b) a); a(i)*b(i) (	вікно редактора робоче поле	>			Вікно	Name - Value a [1 2 3] b [1 2 3] c стану робочого просто соттану Histoy b - 24,09,2021 8107 %	opy
Details ~								a=[1 2 3];	
Select a file to view details	Command Window New to MATLAB See >> a=[1 2 3]; >> b=[1 2 3]; ft >>   pagoc crany	resources for Getting Started.	вікно команд				×	b=[1 2 3]; а*b а*b вікно історії команд ргіляг ргіляг ргіляг р- 01.10.2021 19:26%	

Рис. 5.1. Оболонка Matlab

В Matlab версій вище за 8.Х всі елементи керування зосереджені на стрічці **Toolstrips**. Стрічка є аналогом елементу керування *Ribbon* фірми *Microsoft*. В більш старих версіях керування пакетом проводиться засобами меню та контекстних кнопкових панелях керування, подібно до СКМ *MathCAD*. Розташована стрічка **Toolstrips** згори оболонки. Стрічка є контекстною. Склад елементів на ній змінюється автоматично в залежності від виконуваних пакетом дій.

Слід відмітити, що оновлення версій пакету супроводжується суттєвим збільшенням обсягу пам'яті та не завжди забезпечує розширення функціональних можливостей. Наприклад, версія 7х пакету займає приблизно 4 ГБ на диску, а версія 9.х – вже біля 20 ГБ, з версії 2022 виключений пакет символьних перетворень *Мираd*.

Внизу оболонки розташований рядок стану. В *рядку* відображається поточний стан пакета та кнопка керування паралельним режимом обчислень (рис. 5.2). В разі активації редактора пакета з правого боку відображаються тип файлу, який редагується та координати місця, в якому знаходиться курсор.



Рис. 5.2. Рядок стану

В центрі оболонки розташовуються командне вікно, вікно історії команд, вікно робочого простору, вікно поточних файлів, вікно редактора.

Кількість вікон в оболонці, їхні розміри та положення можуть бути вибрані із стандартних конфігурацій або визначені користувачем вручну. Для керування вікнами оболонки призначено меню **Layout** стрічки (рис. 5.3). Користувач має змогу обрати конфігурацію вікон з стандартних варіантів **Default**, **Three Column**, **Command window only**, **All but Command minimized**.



Рис. 5.3. Меню налаштування вікон оболонки

Кожне з вікон можна «вийняти» з оболонки та зробити вікно незалежним **Undocking**, розкрити на весь екран, закрити, згорнути до значка на краю екрана. Зсув миші з натиснутою лівою кнопкою дозволяє зсунути вікно в інше місце. Керування виглядом робочих вікон здійснюється кнопками керування, які розташовані в правому верхньому куті вікна або відповідними пунктами контекстного меню вікна.

В пакеті реалізовано змішаний тип діалогу керування. Розрахункові та керуючі оператори вводяться в командному вікні Command window, налаштування оболонки, перемикання між режимами роботи пакета проводиться елементуми стрічки. В командному вікні також відображаються результати розрахунків та інформація про помилки. Тому командне вікно можна вважати основним вікном пакета.

Для керування командним вікном пакет має спеціальні команди. Найчастіше застосовуються команди **clc**, **home** та клавіатурні клавіши стрілок « $\uparrow(\downarrow)$ ».

Команда **clc** очищає командне вікно та розташовує курсор на початку першого рядка;

Команда **home** пересуває курсор в верхній лівий куті вікна;

Символами стрілок «вгору/вниз» проводиться прокручування рядків раніше введених в командне вікно команд. Виконання команди при цьому запускається натисканням клавіші **Enter**.

Вікно *iсторії команд* **Command History** є засобом спрощення повторного проведення дій в пакеті. В ньому відображається список всіх введених за час сеансу роботи з пакетом команд. Для повторного виконання команди не потрібно знов вводити її з клавіатури, команда вводиться в командний рядок *командного вікна* кліком на обраному рядку списку вікна *iсторії команд*.

Вікно **Current directory** відображає дерево робочих файлів сеансу та використовується для спрощення введення файлів програм в редактор.

В пакеті реалізовано зручний спосіб контролю та редагування введених даних. Всі данні, які введені користувачем або є результатами розрахунків розміщуються в області пам'яті, яку називають *робочим простором* **Workspace**. Зміст цієї області показується у вікні *робочого простору*. Вікно містить таблицю з інформацію про імена та зміст даних. Наприклад, в сеансі, який зображений на рис. 5.4, введено три змінні. Змінна z є числом зі значенням 1.7431, змінна у є вектором-рядком (одновимірним масивом) зі значеннями 0, 0.2, 0.4 ..., змінна є матрицею (двовимірним масивом). У вікні *робочого простору* можна змінювати значення даних та перейменовувати змінні.

Виділення якогось даного подвійним кліком активує вікно *редактора даних* **Array Editor** (рис. 5.4). В редакторі даних відображаються ім'я змінної, структура та значення даних. За необхідності в редакторі можна змінити значення та ім'я змінної, додати елементи до масиву або видалити елементи з масиву. Наприклад, на рис. 5.4 показано вміст редактора при виділенні змінної **x**. Показано її ім'я **x**, тип – масив розміром 2 рядки та 3 стовпці елементів дійсного типу **double**. Для підвищення зручності навігації та редагування одночасно з редактором даних активується контекстна вкладка **Variable** стрічки (рис. 5.4). Вкладка містить поля індексів поточного елементу масиву, інструменти додавання, прибирання рядків/стовпців масивів, редагування матриць.

лs	APPS		VIEW					1 <b>6</b> 6
Pows 1	Columns 1 SELECTION		e all Sort ▼	>	контекстна вк стрічки	ладка		
inf_t	pm11 b bondaro	nko			0.			
vork	y x x	×			, ©	Name -	Value	
arenk arenk	2x3 double	2	3 4	5	6	y y	[1 2 3;-1 0 1] [0 0.2000 0.4 1 7221	
sx	1 1 2 -1 3	0	1			табли	ця змінних	
sx sx	4							_
sx L	6	реда	ктор даних					
onda enko enko	2				> \	/		

Рис. 5.4. Вікно робочого простору

Дані робочого простору можуть бути збережені для подальшого використання. Дані зберігаються в ANSI файлах з розширенням \*. МАТ

# кнопкою Home-Save Workspace 🗔 стрічки.

Завантаження даних робочого простору з дискового файлу може проводитися кнопкою **Home-Open** стрічки або введенням команди **load** в командному вікні.

Команда

#### load 'ім'я файлу даних' <ім'я 1>,<ім'я 2>

має один обов'язковий аргумент – ім'я файлу даних. Інші аргументи є опціональними. Виклик в команди без опціональний аргументів призводить до завантаження в робочий простір всіх даних з файлу. Визначення після імен змінних приводить до завантаження вказаних змінних..

Видалити змінну з робочого простору можна відповідним пунктом контекстного меню робочого простору або командою clear <x>, ... Виклик команди без аргументів очищає весь робочий простір.

Очистити будь-яке робоче вікно можна пунктом **Clear** відповідного контекстного меню вікна.

Matlab може виконувати дії в двох режимах: режимі калькулятора та програмному режимі. Режими можуть використовуватися в будь якій послідовності.

*Режим калькулятора* є діалоговим режимом. За такої технологією передбачається послідовне введення в рядку командного вікна виразу для обчислення, команди системи та безпосереднє їхнє виконання відразу після натискання клавіші *Enter*.

Програмний режим є пакетним режимом. Він передбачає написання програм та їхній запуск на виконання. Програми можуть бути створені у вбудованому редакторі програм або завантажені із зовнішніх файлів.

Більшість дій, які виконуються в режимі *калькулятора*, можливо виконати й в *програмному* режимі.

#### Контрольні запитання

- 1. На обробку яких даних орієнтовано Matlab?
- 2. Перерахуйте основні можливості Matlab.
- 3. Що таке **Toolbox**? Для яких задач вони застосовуються?
- 4. Для чого слугує модуль Simulink?
- 5. Як в Matlab реалізовано символьні перетворення?
- 6. Чим відрізняються реалізації діалогу в Mathcad та Matlab?
- 7. Назвіть складові діалогового середовища Matlab.
- 8. Що таке Workspace? Які можливості воно забезпечує?
- 9. Чим відрізняються два основні режими роботи Matlab?

#### 5.2. Робота в режимі калькулятора

#### Змінні, вирази, вбудовані функції

Ознакою того, що *Matlab* готовий до прийняття команди є наявність на початку останнього рядка командного вікна позначки "»".

Введення після позначки обчислювального виразу або команди та натискання клавіши *Enter* призведе до виконання дій та виведенню в наступному рядку відповіді.

Введення інформації в командне вікно підпорядковується таким правилам [28]:

- результат дії відразу виводиться в командне вікно. Для того, щоб результат його дії не відображався в командному вікні, слід в кінці рядка поставити символ крапки з комою «;»;
- якщо оператор не містить знака «=», тобто є просто записом послідовності дій над числами та змінними, то значення результату присвоюється спеціальній системній змінній ans. Значення системної змінної ans змінюється після дії кожного оператора без знака «=». Поточне значення змінної ans можна використовувати в наступних операторах;
- кілька операторів можуть бути записані в одному рядку. Розділення операторів провожиться комою «,» чи крапкою з комою «;». Виконання операторів проводиться зліва-направо;
- рядки, які містять коментарі, та не повинні виконуватися помічаються символом відсотків «%» на початку рядка;
- для продовження запису в кількох рядках наприкінці рядка ставляться три крапки.

Всі дані в пакеті можуть бути представлені у вигляді масивів (рис. 5.5) чисельних змінних (numeric), текстових рядків (char), комірок (cell), структур (struct). За замовчанням чисельні масиви мають тип **double**, розріджені матриці мають тип **sparse**.



Рис. 5.5. Типи даних Matlab [2]

Тип	Опис	Діапазон
int8/uint8	Однобайтове ціле з/без знака	-128 127 / 0 255
int16/uint16	Двобайтове ціле з/без знака	-3276832767/0 65535
int32/uint32	Чотирьохбайтове ціле з/без	-21474836482 47483647
	знака	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	$2x10^{-308}\dots 1.8x10^{+308}$

Таблиця 5.1. Типи чисельних даних Matlab [2]

Введення чисел з клавіатури проводиться за загальними правилами мов програмування високого рівня [29]:

- для відділення дробової частини мантиси числа застосовується десяткова точка;
- для чисел в форматі з плаваючою комою десятковий показник задається літерою «е», показник степеня числа записується як ціле число після символу «е». Між записом мантиси числа та символом «е», який відділяє мантису від показника, не повинно бути символів. Наприклад, 12,34·10<sup>5</sup> =12.34e5.

Додаткових дій для опису даних за звичай проводити не потрібно. За замовчанням всі чисельні дані обробляються як комплексні числа з дійсною та уявною частинами у вигляді восьмибайтових чисел типу **double** у мові С з 15-ма десятковими розрядами. Діапазон значень модуля чисел, що можуть оброблюватися, лежить в проміжку між 10<sup>-308</sup> та 10<sup>+308</sup>.

Визначення нової змінної проводиться введенням її імені зліва від знаку рівності та потрібного значення праворуч від знаку:

## <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. В пакеті великі та малі літери в іменах розрізняються.

Пакет має вбудовані змінні, з зарезервованими іменами та змістом:

і, ј – уявна одиниця (корінь квадратний з -1);

**рі** – квадратура кола, число *π* (3.141592653589793);

**inf** – позначка нескінченності;

**NaN** – позначка невизначеного результату, наприклад при діленні 0/0 чи inf/inf;

ерs – похибка операцій над числами;

**ans** – змінна, в якій зберігається результат останньої операції, якщо остання записана без знака «=»;

**realmax/intmax** – максимальна величина дійсного/цілого числа;

realmin/intmin – мінімальна величина дійсного/цілого числа.

Пакет містить в собі вбудовану арифметику комплексних чисел. Більшість вбудованих математичних функцій написані таким чином, що аргументи допускають комплексні числа, а результати також формуються як комплексні числа. Для позначення уявної одиниці в пакеті використовуються звичні в математиці імена «і» та «j».

Комплексні числа вводяться в алгебраїчній формі визначенням дійсної та уявної частин числа шляхом запису:

<ім'я комплексної змінної>=<значення ДЧ>+і\*<значення УЧ>,

де **дч** – дійсна частина комплексного числа, **уч** – уявна частина.

Обчислювальні вирази набирається з чисел, імен змінних, імен вбудованих функцій або функцій користувача. Синтаксис виразів є аналогічним синтаксису структурних мов програмування. Вирази записуються послідовно в один рядок. За потреби зміст рядка може бути продовжений в наступному рядку. Алгоритми набираються послідовністю розташованих один за одним рядків.

Пакет містить велику бібліотеку вбудованих операторів логічних та арифметичних дій, функцій різного призначення [31]:

- оператори арифметичних та логічних дій +, −, \*, \, /, >, >=,
   <, <=, =, ==;</li>
- математичні функції sqrt, pows тощо;

- логарифмічні та експоненційні функції logs, log10, exp тощо;
- тригонометричні прямі та зворотні функції;
- гіперболічні функції прямі та зворотні функції;
- спеціальні функції Беселя, Лагранжа тощо;
- функцій обробки текстових рядків;
- функції роботи з часом;
- функції роботи з матрицями та векторами;
- функції статистичної обробки даних;
- функції спектральних перетворень;
- функції обробки поліномів;
- функції двовимірної та тривимірної високорівневої графіки
- функції низькорівневої дескрипторної графіки.

Отримати довідкову інформацію про можливості та синтаксис вбудованих функцій можна у вікні довідки натисканням клавіши «F1» або в командному вікні введенням функцій **help** або **lookfor**.

Функція **help fun** виводить в командне вікно опис синтаксису функції **fun**.

Функція **help орs** виводить в командне вікно список операторів.

Функція **help elfun** виводить в командне вікно список елементурних функцій.

Функція **help specfun** виводить в командне вікно список спеціальних.

Функція **help iofun** виводить в командне вікно список функцій введення/виведення.

Функція **help matfun** виводить в командне вікно список функцій роботи з матрицями.

Функція **help polyfun** виводить в командне вікно список функцій обробки поліномів.

Функція **help graph2d** виводить в командне вікно список функцій двовимірної графіки.

Функція **help graph3d** виводить в командне вікно список функцій тривимірної графіки.

Функція **help graphics** виводить в командне вікно список функцій дескрипторної графіки.

Функція lookfor word проводить пошук слова word та виводить список функцій, в яких це слово зустрічається.

Виклик вбудованої або функції користувача проводиться типовим для мов структурного програмуванням чином введення імені функції та вказанням в дужках переліку значень фактичних параметрів:

#### <ім'я функції>(<перелік фактичних значень аргументів >).

Слід мати на увазі, що аргументи базових тригонометричних функцій задаються в радіанах. Функції, що приймають аргументи в градусах мають таке ж ім'я, але містять в кінці імені символ d. Наприклад, функція тангенса tan(x) потребує запису значення аргументу в радіанах, а функція тангенса tan(x) – в градусах.

Комплексні числа в пакеті можливо задавати в алгебраїчній формі через дійсну та уявну частини або в експоненційній формі через амплітуду та фазу (аргумент) числа:

$$Z = A + iB = |Z|e^{i\varphi} = \sqrt{real(Z)^2 + image(Z)^2 *} \exp(i * arctg(\frac{imag(Z)}{real(Z)}) = abs(Z) \cdot \exp(i * angle(Z)) \quad \varphi = arctg(\frac{imag(Z)}{real(Z)}).$$

Для визначення параметрів комплексних чисел пакет має спеціальні вбудовані функції: **real (Z)** для знаходження дійсної частини, **imag (Z)** для знаходження уявної частини, **angle (Z)** для знаходження аргументу  $\varphi$  в радіанах від  $-\pi$  до  $+\pi$ , **conj (Z)** для знаходження комплексно-спряженого комплексного числа, **complex (A**, **<B>)** для створення комплексного числа з дійсною частиною А та уявною частиною В.

Функції пакета мають на відміну від функцій більшості МПВР можуть мати та сталу, а змінну кількість аргументів та результатів, що дозволяє використовувати функції значно гнучкіше.

Наприклад, вбудована функція **x=xlsread('file.xlsx')** з одним аргументом та одним результатом проведе зчитування всіх даних з першого аркуша книги ексель з файлу **file.xlsx** в матрицю **x**. Та ж функція зі

двома аргументами x=xlsread('file.xlsx','Sheet1') проведе зчитування всіх даних з аркуша Sheet1 книги ексель з файлу file.xlsx в матрицю x, a виклик з трьома аргументами x=xlsread('file.xlsx','Data', 'D2:H4') проведе зчитування всіх даних діапазону 'D2:H4 з аркуша Data книги ексель з файлу file.xlsx в матрицю x. Виклик функції у вигляді

[x,t,r]=xlsread('file.xlsx') поверне як результат вектор, перший елемент x якого буде матрицєю з числовими даними з книги вексель, другий елемент t буде масивом комірок з назвами стовпців даних на аркуші вексель, третій елемент r буде масивом комірок з текстовими даними.

Порядок виконання арифметичних та логічні операції в виразах визначається наступним пріоритетом дій [31]:

1. Дії в дужках.

2. Вбудовані функції та функції користувача, логічні функції **and**, **or**, **not**, **xor**.

3. Логічне заперечення.

4. Транспонування, піднесення до степеню.

5. Множення, ділення.

6. Складання, віднімання

7. Логічні відношення: >, >=, <, <=, ==.

8. Логічне «і» (символ «**&**»).

9. Логічне «або» (символ « | »).

10. Логічне «виключаючи або» (символ «| |»).

Дії одного пріоритету виконуються зліва – направо.

Наприклад, обчислення виразу для знаходження кореня квадратного

рівняння  $\frac{-2.2 + \sqrt{2.2^2 - 4 \cdot 1.33 \cdot 2}}{2 \cdot 1.33}$  має наступний вигляд:

>>(-2.2+sqrt(2.2<sup>2</sup>-4\*1.33\*2))/(2\*1.33)

ans= -0.82707 + 0.90538i121.2446

#### Формат виведення результатів

Вигляд чисельного результату на екрані визначається попередньо заданим операторами форматування або пунктами інтерфейсу форматом чисел. Формат виведення результату може бути установлено за допомогою вікна **Preferences** (рис. 5.6). Активується вікно **Preferences** кнопкою

Home-Environment-Preferences Crpiчки. Вузол дерева Command Window у вікні містить список Numeric Format, який містить список можливих форматів чисел для їх виведення в командне вікно.

A Preferences	
ΜΑΤΙΑΒ	MATIAR Commond Mindow Desfavores
Add-Ons	Text display
App Designer Code Analyzer	Numeric format: Long V
Colors	Numeric display: short
Command History	Datetime format short e
Comparison	Locale: long e n_US ~
Current Folder	Default date and short g
Editor/Debugger Figure Copy Template	Default date-only short eng
Fonts	Display long eng V
General	Wrap lines
GUIDE Heln	Set matrix display width to eighty columns

Рис. 5.6. Вікно налаштування форматів

В списку налаштувань вікна доступними є наступні формати [31]:

- Short формат з фіксованою комою з 4-ма знаками (застосовується за умовчанням);
- Long формат з фіксованою комою з 15-ма знаками;
- **Нех** виведення в шістнадцятирічній системі числення;
- **Bank** формат з фіксованою комою з 4-ма знаками;
- **ShortE** експоненційний формат з 4-ма знаками;
- **LongE** експоненційний формат з 15-ма знаками;
- ShortEng експоненційний формат з 4-ма знаками з показником степеня кратним 3;
- LongEng експоненційний формат з 15-ма знаками з показником степеня кратним 3;
- ShortG універсальний формат з 5-мя знаками. Виведення з фіксованою комою або в експоненційному вигляді визначається значенням числа;

- LongG універсальний формат з 15-мя знаками. Виведення з фіксованою комою або в експоненційному вигляді визначається значенням числа;
- **Rational** запис числа у вигляді раціонального дробу.

Наприклад, результат ділення 2.0 на 3.0 в різних форматах має наступний вигляд [2]:

0.6667	Short
0.6666666666666666666666666666666666666	Long
6.6667e-001	ShortE
6.66666666666666666e-001	Long E
0.66667	ShortG
666.6667e-003	ShortEng
0.6666666666666666666666666666666666666	LongG
3fe555555555555555555555555555555555555	Hex
0.67	Bank
2/3	Rational

Визначення формату результатів програмно проводиться командою format <тип>. Обов'язковий аргумент тип є службовим словом визначення формату: short, long, hex, bank, shortE, longE, shortG, long, shortEng, longEng, rational, compact (запис числа з видаленням пробілів з запису), loose (запис числа без видалення пробілів з запису).

Службові слова потрібно вводити без лапок. Визначене командою значення формату діє з місця визначення до наступної дії зміни формату результату.

Формат чисел впливає лише на форму виведення числа на екран. В пам'яті значення залишаються незмінними, з 15-ма знаками мантиси.

## Логічні операції

В пакеті відсутній спеціальний логічний тип даних «вірно» (true)/ «не вірно» (false). Результатами логічних перевірок є цілі числа 1 (true) або 0 (false).

Пакет дозволяє проводити *інтегральні* логічні операції зі значенням всього аргументу та *бітові* логічні операції над кожним бітом цілих аргументів окремо.

Для перевірки умов в пакеті передбачені:

- унітарні логічні оператори (вирази відношень);
- логічні операції;
- логічні вирази;
- логічні функції.

*Інтегральні* операції перевірки можуть вводитися за допомогою унітарного логічного оператора у вигляді конструкції перевірки або вбудованих логічних функції (табл. 5.2).

Унітарні оператори використовуються для порівняння значень двох арифметичних або логічних виразів між собою.

Вираз відношення записуються як два вирази або змінні зі знаком логічного відношення (логічної умови) між ними:

# <вираз 1> <знак відношення> <вираз 2>

Логічні операції в пакеті реалізують булеву логіку АБО (OR), I (AND), НІ (NOT), ВИКЛЮЧАЮЧЕ АБО (ХО) (табл. 4.5). Вводяться вирази логічних операцій тільки за допомогою вбудованих логічних функцій (табл. 5.2).

Результат *інтегральної перевірки* є ціле число. Якщо задане відношення виконується, то вираз відносин набуває значення рівне 1, якщо не виконується – 0.

*Бітові* логічні операції проводяться над кожним бітом цілих аргументів окремо. Для бітової перевірки доступними є тільки логічні операції мулевої логіки (табл. 5.2). Результатом є ціле число, яке створюється зі значень бітової операції. Записуються бітові операції тільки через вбудовані логічні функції.

Наприклад,

# » a=3; c=2; bitand(a,c) ans=2

Додаткові інструменти для бітової роботи з даними надають такі вбудовані функції: функція **bitset(a,N<,v>)** встановлює N-й біт аргументу а в 1(y), функція **bitget(a,N)** зчитує N-й біт аргументу.

Знак відношення	Символи	Функція
=	==	eq(a,b)
<	<	lt(a,b)
>	>	gt(a,b)
≤	<=	le(a,b)
2	>=	ge(a,b)
≠	~=	ne(a,b)
логічне I	&	and(a,b)
логічне АБО		or(a,b)
логічне HI	~	not(a,b)
виключаюче АБО		xor(a,b)
побітове І		bitand(a.b)
побітове АБО		<pre>bitcmp(a,b)</pre>
побітове XOR		bitxor(a)

Таблиця 5.2. Символи логічних відносин

Так як результатом логічних дій в пакеті є числа 0, 1, то *унітарні логічні* оператори, логічні операції та математичні операції можуть поєднуватися в один вираз – логічний вираз.

Слід обов'язково пам'ятати про пріоритет проведення дій в виразах. Наприклад, два вирази: and (A, B)+с та A&B+c не є еквівалентними. В першому виразі логічне «і» записано функцією та буде виконуватися першим (має перший пріоритет), додавання буде виконуватися другим (має шостий пріоритет). В другому виразі логічне «і» записано символом (має восьмий пріоритет) та буде виконуватися другим, першим буде проводитися додавання (має шостий пріоритет).

Результат виконання логічного виразу може бути збережено в змінній, наприклад, **f=a<=x**.

## Функції користувача в режимі калькулятора

В режимі калькулятора може виникнути потреба в багаторазовому проведенні обчислення одного або кількох виразів для різних значень аргументів. Для таких випадків в пакеті передбачено використання вбудованих (Inline function) та анонімних (Anonymous function) функції користувача (@). Вбудована та анонімна функції користувача застосовуються в два кроки. На першому кроці функції описуються в командному вікні, на другому кроці викликаються як інші вбудовані функції пакета своїм іменем з дійсним аргументами.

Вбудована функція користувача описується наступним чином:

# ім'я\_функції=inline('формула', <'арг1',..., 'аргМ'>),

де 'формула' є текстовим рядком з виразом для обчислення. Обов'язковим аргументом є тільки тестовий рядок з обчислювальним виразом 'формула'. Для функції з одним аргументом його вказувати не потрібно, для кількох аргументів застосовується їхній перелік через кому 'apr1',..., 'aprM'. За замовчанням кілька аргументів розташовуються в алфавітному порядку. Наприклад, опис

#### >> fun1=inline('sin(a\*x)-x^2\*cos(b\*x)')

визначає вбудовану функцію користувача з іменем **fun1** та аргументами a,b,x:

#### funl=inline function:funl(a,b,x)=sin(a\*x) - $x^2*\cos(b*x)$

Для зміни порядку аргументів їх слід записати через кому після виразу для обчислень **'формула'** в потрібному порядку:

```
» 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)
```

Наприклад, застосування вбудованої функції для обчислення кореня квадратного рівняння слід провести її опис в командному вікні:

```
>>myquad=inline('(-b+sqrt(b^2-4*a*c))/(2*a)')
```

```
myquad =
```

```
Inline function:
```

```
myquad(a,b,c) = (-b+sqrt(b^2-4*a*c))/(2*a)
```

Після цього можна проводити виклик функцій:

```
>> myquad (1,2,3)
ans = -1 +1.4142i
aбo
>> x=1; y=2; z=3;
>> myquad (a,b,c)
ans = -1 +1.4142i
Aнонімна функція користувача визначається наступним чином:
```

```
Ім'я функції = @(список аргументів) формула.
```

На відміну від *вбудованої* функції користувача, аргументи та формула *анонімної* функції записуються в звичайному вигляді без апострофів.

Наприклад, застосування вбудованої функції для обчислення кореня квадратного рівняння слід провести її опис в командному вікні:

```
>> myquad1=@(f,b,c) '(-b+sqrt(b^2-4*a*c))/(2*a)'
myquad1 =
```

```
function handle with value:
```

 $@(f,b,c)'(-b+sqrt(b^2-4*a*c))/(2*a)'$ 

Після цього можна проводити виклик функцій:

```
>> myquad1(1,2,3)
ans = -1 +1.4142i
a60
>> x=1; y=2; z=3;
>> myquad1(a,b,c)
ans = -1 +1.4142i
```

#### Масиви

*Масивом* в комп'юториці називають структуру даних з кількох однотипних об'єктів (чисел, символів, рядків, тощо), яка розглядається як єдине ціле. Доступ до кожного елементу масиву здійснюється за допомогою індексу елементу. Індекс має порядковий тип, та слугує адресою елементу в масиві. Індекс виступає ідентифікатором при по елементних обчисленнях.

Термін масиви є терміном комп'юторики. В математиці терміну *масив* відповідають терміни *вектор, матриця, тензор*. Одномірні масиви відповідають векторам-рядкам, які містять елементи в одному рядку та векторам-стовпцям, які містять елементи в одному стовпці. Двовимірні масиви, елементи яких розташовані по рядках і стовпцях, відповідають матрицям, багатомірні масиви складної структури – тензорам.

*Matlab* як система комп'ютерної математики є системою, яка спеціально призначена для проведення розрахунків саме з векторами, матрицями та поліномами.

В пакеті масиви реалізовані у вигляді матриць, масивів комірок (cell array), структур (struct), таблиць (table).

Матрицею A розміром R x C в пакеті вважається сукупність однотипних числових даних, які розташовані у вигляді таблиці з R рядків та C стовпців

В пакеті передбачено використання векторів-рядків, векторів-стовпців, матриць та тензорів. Вектор вважається матрицею з одним стовпцем. Число вважається вектором розміром 1\*1.

Особливістю векторних та матричних розрахунків в пакеті є те, що не є обов'язковим обробляти масиви поелементно, більшість математичних дій виконуються за правилами матричної математики за замовчанням.

За потреби проведення саме поелементних дій з матрицями слід явно вказувати поелементні дії додаванням символу крапки «.» перед кожним знаком арифметичної дії або використовувати циклічні алгоритми поелементної обробки зі звертанням до кожного елементу матриці окремо по його індексу.

Звернення до елементів матриць та векторів проводиться вказанням імені та індексу елементу. Індекси записуються після імені в дужках з розділювачем у вигляді коми. Першим вказується індекс рядка, другим – стовпця. Початковим індексом масивів є 1.

Наприклад, запис A (1) відповідає першому елементу вектора а, B (2, 2) – елементу з другого рядка та другого стовпця матриці B.

Визначити матрицю в пакеті можна однократним введенням повного списку елементів, поелементно вручну або програмно в циклі, завантаженням з зовнішніх файлів, генеруванням матриць із застосуванням спеціальних вбудованих матричних функцій. Однократним введенням списку елементи вектора-рядка задаються через пробіл або кому в квадратних дужках.

Наприклад, запис  $V = [1 \ 2 \ 3]$  або V = [1, 2, 3] визначає вектор-рядок V, з трьома елементуми зі значеннями 1, 2, 3.

Поелементне введення цього ж вектора буде мати наступний вигляд: V(1)=1; V(2)=2; V(3)=3;

Для векторів-рядків, елементи якого складають арифметичну прогресію, оптимальним з точки зору швидкодії можна вважати спосіб визначення, який називають *векторним* [29; 31]:

 $\mathbf{V} = \mathbf{n} : \mathbf{h} : \mathbf{k} ,$ 

де  $\mathbf{n}$  – значення першого елементу вектора, яке складає початкове значення прогресії;  $\mathbf{k}$  – значення останнього елементу вектора, яке складає кінцеве значення прогресії;  $\mathbf{h}$  – крок прогресії. Якщо крок прогресії дорівнює одиниці, то його можна не вказувати.

Наприклад, запис **v=1:1:3** або **v=1:3** визначає вектор-рядок V, з трьома елементуми зі значеннями 1, 2, 3.

Вектор-рядок з великою кількістю елементів можна вводити частинами з наступним поєднанням частин в єдиний вектор. Наприклад, **W**=[V V].

Елементи в векторі-стовпці відокремлюються між собою при введенні знаком крапки з комою «;».

Наприклад, запис **V1=[1; 2; 3]** визначає вектор-стовпець V, з трьома елементуми зі значеннями 1, 2, 3.

Вектор-стовпець може бути отриманий з вектора-рядка транспонуванням: **v1=v'**.

Вектор-стовпець з великою кількістю елементів також можна вводити частинами з наступним поєднанням частин в єдиний вектор за умови додавання операції транспонування. Наприклад, **W=[V V]**'.

Для визначення розміру вектора призначена функція

#### l=length(V).

Обов'язковим аргументом є ім'я вектора V. Результатом функції є число 1 – кількість елементів в векторі. Однократне введення матриці списком проводиться в квадратних дужках по рядках. Елементи рядка матриці один від одного відокремлюються пробілом чи комою, рядки відокремлюються між собою знаком крапки з комою «;».

Наприклад, запис M=[1 2; 3 4] визначає матрицю А розміром 2\*2 з чотирма елементуми M(1,1)=1; M(1,2)=2; M(2,1)=3; M(2,2)=4.

Для визначення розмірів матриці призначена функція

Обов'язковим аргументом є ім'я матриці **a**. Результатом функції є вектор-рядок. Перший елемент вектора **r** містить кількість рядків матриці **a**, другий **c** – кількість стовпців стовбців матриці **a**.

Операція виділення з матриці визначеного стовпця/рядка в пакеті застосовується позначка двох крапок « :». Символ ставиться на місці тих елементів, які не потрібно чіпати. Наприклад, запис **A(:,c)** виділяє **c**-ий стовбець з матриці **A**, запис **A(r,:)** виділяє **r**-ий рядок з матриці **A**.

Запис **A(:)** перетворює матрицю **A** в вектор. Слід мати на увазі, що матриці в пакеті зберігаються по стовпцях у вигляді вектора-стовпця. Елемент матриці можна ідентифікувати не по двох, а по одному індексу, який визначає номер елементу в векторі.

Наприклад, звернення до матриці з наведеної вище матриці **М** звернення **М(1,2)** еквівалентно зверненню **М(3)**:

#### M(1,2) ans = 3 M(3) ans = 3

Операція виділення з матриці визначеної під матиці проводиться вказанням діапазону адрес в індексах через позначку двох крапок « :». Наприклад, запис A(r1:r2, c1:c2) виділяє з матриці A підматрицю, в яку входять рядки з r1 до r2 та стовпці з c1 до c1.

Матрицю можна створити поєднанням існуючих матриць в єдину. Така операція поєднання називається *конкатенацією* матриць.

Для конкатенації матриць призначені вбудовані функції cat, horzcat, vertcat..

Функція **M=horzcat (A, B <, C, ...>)** поєднує матриці A, В тощо з однаковою кількістю рядків в одну матрицю М зліва направо. Таку операцію називають *горизонтальною конкатенацією* матриць.

349

Функція **M=vertcat(A,B <,C,...>)** поєднує матриці A, В тощо з однаковою кількістю стовпців в одну матрицю М згори вниз. Таку операцію називають *вертикальною конкатенацією* матриць.

Функція M=cat (dim, A, B <, C, ...>) поєднує матриці A, В тощо в одну матрицю M в напрямку dim. Значення dim 1 проводить вертикальну конкатенацію, значення dim 2 – горизонтальну.

Операцію конкатенації можна провести і безпосереднім введенням списку матриць.

Запис A=[A1, A2,..., AN] реалізує горизонтальну конкатенацію.

Запис A = [A1; A2; ...; AN] реалізує вертикальну конкатенацію.

Для видалення елементу матриці слід присвоїти йому «порожнє» значення. Для цього використовуються символи квадратних дужок без пробілу між ними «[]». Можна видалити окремий елемент або рядок/стовпець. Наприклад, запис A(:,3:5)=[] видаляє з матриці A стовпців з третього до п'ятого, запис A(1,:)=[] видаляє перший рядок з матриці A.

Дії з розширення та скорочення векторів виконуються аналогічно діям з матрицями.

Для розрахунку параметрів матриць та операцій матричної математики пакет має цілий ряд вбудованих функцій [2; 32]: sum, prod, length, size, sort, ndims, max, min, mean, ', ., rot90, dot, fliplr, cross, inv, eig, det тощо

Слід мати на увазі, що разом з загальновживаними символами арифметичних матричних операцій, як то «+», «-», «\*», пакет дозволяє застосовувати символи ділення «/» та «\». В матричній математиці ОПЕРАЦІЇ ДІЛЕННЯ МАТРИЦЬ НЕМАЄ. Результатом використання вказаних символів буде зовсім інша математична дія.

Перевірка логічних умов з масивами має деякі особливості. Логічні дії над двома масивами або над масивом та скаляром виконуються поелементно. Результатом є логічний масив з логічних нулів та одиниць.

Результуючі масиви з логічних нулів та одиниць мають додаткові можливості. Вони можуть бути використані для адресації масивів по номерах елементів, значення яких є логічною одиницею.

Наприклад, результатом порівняння на не перевищення значення 10 вектора **x=[8 12 8 4]** буде вектор **y**:

```
>>y=x<10
y = 1×4 logical array
1 0 1 1
```

За допомогою вектора логічних значень можна сформувати з вектора х новий вектор, який би містив тільки елементи, які вдовольняють перевіреній умові:

>> z=x(y) z = 8 9 4

В разі необхідності зберегти в одному масиві елементів різних типів, застосовуються *масиви комірок*.

Обробка комірок проводиться також, як і матриць. Відрізняються на екрані масиви комірок від матриць символами позначення індексів. Індекси в масиви комірок позначаються фігурними дужками.

Наприклад, для матриці даних А розміром 2х2 можна поєднати в масиві комірок саму матрицю, вектор, елементуми якого є суми значень експериментів з рядків матриці, значення мінімального елементу даних в матриці та текстовий коментар.

Запис

#### »C = {A sum(A') min(min(A)) 'data matrix'}

створює масив комірок C, першим елементом [2x2 double] є сама матриця, другим елементом [1x2 double] є вектор результату функції додавання sum (A'), третім елементом [1x1 double] є число – результат дії функції min (min (A)), четвертим елементом [1x11 char] – рядок символів.

Порожній масив комірок розміром **M\*N** створюється командою cell(MxN).

Для випадків, коли зручніше описувати дані за принципами об'єктного програмування та баз даних у вигляді «кортеж.відношення» в пакеті реалізований складний тип даних *структури* structures. Структури є найбільш подібними до записів **record** в мовах програмування та можуть містити в собі дані різних типів. Елементи структури називаються *полями* структури. Доступ до елементів структури здійснюється за іменем поля, яке вказується після символу крапка «.» після імені структури.

Описувати типи даних та склад структури попередньо не потрібно, вони готові до обробки відразу після задання першого екземпляра структури.

Наприклад, для опису корпусу з параметрами довжини, ширини, висоти, маси, матеріалу

Найменув.	Довжина	Ширина	Висота	Матеріал
К-1.1	12	2	5.5	Ст3
К-1.2	6	3	1.5	АБС

можна застосувати структуру К з відповідними полями: K.name, K.length, K.width, K.height, K.mater.

Структуру можна визначити через введення її екземпляру значенням кожного поля по черзі вручну або вбудованою функцією створення структури struct('field1', VALUES1, 'field2', VALUES2,...).

```
»K.name='K-1.1'; K.length=12; K.width=2;
```

```
»K.height=5.5;K.mater='CT3';
```

```
або
```

```
»K=struct('name','K-1.1.','length',12'width',2,...
'height',5.5,'mater','CT3')
```

Екземпляри структури можна поєднувати в масиви. Елементи масиву структур позначаються як звичайні масиви числовим індексом в дужках після імені структури. Введення одного екземпляру структури автоматично дозволяє звертатися до нього як до першого елементу масиву структури.

```
K(1) = struct with fields:
    name: K-1.1
    length: 12
    width: 2
    height: 5.5000
    mater: 'CT3'
```

Введення елементів масиву структури проводиться аналогічно одиночному екземпляру структури з додаванням після імені структури в дужках потрібного значення індексу елементу. Наприклад,

```
»K(2)=struct('name','K-1.2','length',6,...
'width',3,'height',1.5,'mater','AEC')
```

Для того, щоб отримати зміст поля елементу структури слід ввести ім'я структури з індексом в круглих дужках та через крапку ввести назву поля. Наприклад, »**x=K(2).length x=6**.

Запис без індексу завжди повертає зміст першого елементу структури. Наприклад, »**x=K.length x=12**.

Отримати для обробки всі значення визначеного поля масиву структури можна у вигляді масиву комірок або звичайного масиву – вектора. Для того, щоб отримати зміст всіх полів масиву структури оператором прирівнювання слід ввести ім'я структури з назвою поля через крапку забраними в дужки. Фігурні дужки повертають результат у вигляді масиву комірок, квадратні – у вигляді вектора. Елементуми вектора для чисельних полів є числа. Результатом для текстового поля масиву структури буде вектор з одним елементом. Текстові поля зчіплюються в єдиний рядок. Елементи масиву комірок приймають тип даних поля структури.

Наприклад,

```
»x={K.length}
x = {[12]} {[6]} % <1x2 cell> масив комірок
»x=[K.mater]
x = ['Cт3AEC'] % <1x6 char> вектор
```

Обробка структур забезпечується вбудованими функціями на кшталт fieldnames(K), isstruct(K), getfield(K,'field'), isfields(K,'field'), setfield(K,'field',V)тощо

Для роботи з даними у вигляді таблиць, які мають головку, а стовпці яких можуть містити як числову, так і текстову інформацію, в *Matlab* введено тип даних *maблиця* **table**. Саме такий вигляд мають *worksheet MS Excel*. Таблиця в *Matlab* є сукупністю змінних типу вектор-стовпець або одномірний масив-стовпець комірок. Саме в таблицю вбудовані функції *Matlab* проводять імпорт даних з *MS Excel*. Для створення нової таблиці використовується функція

## T=table(var1<,...,varN,Name,Value>).

Обов'язковим аргументом є ім'я наперед визначеної змінної **var1**, яка містить дані майбутнього стовпця таблиці, або явно записані дані для стовпця у вигляді вектора або масиву комірок. Елементи масивів аргументів заповнюватимуть відповідні стовпці таблиці. Комірки таблиці наслідують тип елементів аргументів. Комірки таблиці з векторів набувають тип елементів векторів, комірки таблиці з масивів комірок мають тип комірки (**cell**). Якщо дані в таблицю беруться зі змінних, то за замовчанням стовпцям таблиці надаються імена відповідних змінних, якщо дані задаються явно, то стовпцям надаються імені **Var1**, **Var2** тощо.

Наприклад, для наведених вище табличних даних за замовчанням може бути створена таблиця

В першому випадку буде створена таблиця з поіменованими іменами користувача стовпцями

```
T = 2 \times 5 table
```

name	length	width	height	mater
'K-1.1'	12	2	5.5	' C <b>±</b> 3 '
'K-1.2'	6	3	1.5	'AEC'

В другому випадку буде створена таблиця з іменами стовпців Vari

```
T = 2 \times 5 table
```

Var1	Var2	Var3	Var4	Var5
'K-1.1'	12	2	5.5	'C <b>±</b> 3'
'K-1.2'	6	3	1.5	'ABC'

Змістовна частина таблиці в обох випадках є однаковою.

Опціональними парами «властивість-значення» можна визначити додаткові параметри таблиці при її створенні. Імена стовпців визначає властивість VariableNames, рядків – RowNames. Значення властивостей вводяться як масиви-рядки комірок.

Наприклад, запис

T=table({'K-1.1';'K-1.2'},[12;6],[2;3],[5.5;1.5],... {'Cτ3';'AEC'},'VariableNames',{'N','L','W','H','M'}) αδο

T=table(name, length, width, height, mater,... 'VariableNames',{'N','L','W','H','M'})

створить таблицю з іменами стовпців 'N', 'L', 'W', 'H', 'M':

```
T = 2 \times 5 table
```

N	느	W	H	M
'K-1.1'	12	2	5.5	'C亚3'
'K-1.2'	6	3	1.5	'ABC'

Доступ до комірок таблиці проводиться індексованими змінними на зразок масивів з використанням круглих дужок та фігурних дужок. Замість другого індексу може використовуватися ім'я змінної відповідного стовпця. Результат звернення з круглими дужками буде мати тип таблиця. Результат звернення в фігурних дужках буде мати тип відповідного елементу таблиці, тобто буде значенням елементу.

Наприклад, звернення до комірки таблиці Т з першого рядка другого стовпця може бути проведено як T(1,2), T(1,'L') або  $T\{1,2\}$ ,  $T\{1,'L'\}$ . Результатом першого звернення буде таблиця

```
ans =table
<u>L</u>
12
```

Результатом першого звернення буде таблиця другого звернення буде число **ans = 12**.

Перетворення масивів, комірок, структур в таблицю проводиться функціями array2table, cell2table, struct2table відповідно.

## Оформлення та збереження розрахунків

Пакет дозволяє зберегти вмісту командного вікна у вигляді pdf файлу з контекстного меню командного вікна або у вигляді скріншоту.

Іншим засобом збереження та повторного виконання команд командного рядка є запис дій у вигляді скрипт-файлу в редакторі. В пакет вбудовано два редактори: редактор м-файлів (програм) та редактор динамічних («живих», Live Script) скриптів *Live Editor*.

Дії сеансу режиму калькулятора можна записати в редакторі м-файлів та виконати як файл-сценарій. Обов'язковою умовою є повне програмне визначення форматів та режимів. Дії через інтерфейс користувача на зберігаються та не відтворюються редактором. Особливістю такого виконання є роздільне розташування команд у вікні редактора, результатів – в командному вікні.

*Live Editor* дозволяє поєднати на екрані команди, функції, результати розрахунків пакету та форматований текст з поясненнями, коментарями тощо, ілюстрації для надання документу презентаційного вигляду на кшталт вигляду документів *MathCAD*. За можливостями є подібним до простого редактора *MS WordPad*.

Документи *Live Editor* зберігаються в архівованих XML файлах з розширенням .*mlx*.

Активується редактор автоматично при створенні нового документа кнопкою **New Live Script** стрічки або відкритті існуючого .*mlx* файлу.

Керування редактором проводиться з контекстної вкладки стрічки (рис. 5.7). Вкладка автоматично виводиться на екран.



Рис. 5.7. Вкладка Live Editor

Документ може мати два типи областей: текстову та програмну. Області займають цілий рядок та не перекриваються. Текстова область призначена для введення пояснень, коментарів, ілюстрацій тощо. В програмній області розташовуються команди, вирази, функції пакета, результати розрахунків, графіки. Визначається тип областей відповідними кнопками **Text**, Code стрічки.

Визначення типу області активує поля налаштувань відповідної групи стрічки. Для текстової області доступне форматування стилю тексту, абзацу, списків. Для програмної – введення коду або коментарів.

Фрагменти програм в документі можуть бути запущені на виконання

безпосередньо з редактора кнопкою **Run** стрічки.

Результати розрахунків можуть бути виведені безпосередньо в програмну область документа (рис. 5.8 а) або на окрему сторінку праворуч від тексту (рис. 5.8 б). Спосіб відображення визначається кнопками режиму, які знаходяться справа вгорі повзунка екрана (рис. 5.8 а).

Dynamic_Systems_Example.mbx X +	Dynamic, Systems_Example.mbx 🔀 +	
y = lsim(A, B, C, D, forces_N, time_s, x0) % This data will have	time_s = [0:0.01:75]; forces N = zeros(length(time_s), 2); % This makes a matrix with $x^0 = [0.5, 0, 0, 0];$ $x^0 = [0.5, 0, 0, 0];$	0 83e-05 024287
y = 0.5 0 0.49946 -3.5283-05 0.49911 -0.00024287 0.49611 -0.00024287 0.49617 -0.00024526 0.49377 -0.0024977 0.49906 -0.0024515 0.48779 -0.004664	Now that we've set up those things, we can run the "lisim" command. We can         0.49517         -0.88           let it generate a plot directly, or - if we want to have some control over the         0.49377         -0.8           appearance of the plot - we can make it generate data and then plot that         0.49377         -0.8           data.         0.49387         -0.8	071526 014977 026115 040664 058655 080091 010495
0.48/1/3 -0.0040505 0.49123 -0.005855 0.49038 -0.005855 0.4716 -0.81245 0.4716 -0.81245 0.4716 -0.81245 0.4676 -0.81245 0.4676 -0.81246 0.45567 -0.81248 0.45567 -0.8	<pre>% Simulation of the system and generation of graphs % Simulation of the system and generation of graphs 9 = 1sim(A, B, C, D, forces_N, time_s, x0) % This data will have figure(1), clf, hold on, plot(time_s, y(:,1), 'k-', 'LineWidth', 0.43976 - 0.</pre>	013322 016485 .01998 023804 027952 032418 037197
0.4902/ -0.02/952 0.44416 -0.032418 •	<pre>// figure(1), plot(time_s, y(:,2), 'r', 'LineWidth', 2), xlabel(') 6</pre>	

Рис. 5.8. Виведення результатів: а – в тексті; б – в окремому вікні

# Контрольні запитання

- 1. Що є ознакою готовності *Matlab* до прийняття команд?
- 2. За якими правилами слід вводити числа в Matlab?
- 3. Що означає наявність символу «;» наприкінці рядка?
- 4. Що таке змінна ans?
- 5. Які типи даних наявні в Matlab ?
- 6. Для чого застосовують три крапки?
- 7. В яких одиницях задаються параметри тригонометричних функцій?
- 8. За яким пріоритетом проводиться розрахунок виразів Matlab ?
- 9. Які два типи логічних дій реалізовано в Matlab ?

10. Назвіть види функцій користувача.

11.Що є аргументом функції **eval**?

- 12. Як задати обраний формат виведення результатів?
- 13.Як задаються вектори та матриці?
- 14.Що таке векторна змінна?

15.Що таке конкатенація матриць?

16. Назвіть особливість логічних дій над масивами.

## 5.3. Програмування в Matlab

Основним режимом роботи з пакетом для проведення обчислювальних експериментів є програмний режим. Програми пакета зберігаються у вигляді зовнішніх дискових текстових файлів з розширенням . *m*.

В пакеті розрізняють два види програм – *файл–сценарій* і *файл-функція* 

Файл–сценарії є незалежними модулями. Запускаються на виконання оператором з ім'ям сценарію з командного рядка або іншого модуля. Після завершення виконання сценарію пакет переходить в режим очікування. Сценарії не мають аргументів, Дані визначаються в самих програмах або вводяться користувачем. Зазвичай файл–сценарії зберігають основні програми, які керують обчислювальним процесом. В пакеті файл–сценарій називають типово для програмування скрипт-файлом (script), на відміну від спільноти Office, де так називають і файл–сценарій, і файл-функцію.

Файл-сценарій (script) – це записана на диск послідовність команд пакета. При виконанні файлу-сценарію 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 програми не стоїть точка з крапкою «;».

Запуск сценарію з редактора програм здійснюється натисканням кнопки **Run** стрічки редактора. Запуск сценарію з командного вікна здійснюється введенням імені файлу без розширення або команди **run <iм'я файлу>**.

Для того, щоб пакет знаходив файл сценарію, потрібно занести шлях до файлу в список робочого оточення. Редагування шляхів пошуку файлів проводиться в діалоговому вікні **Set path** (рис. 5.9) або програмно.



Рис. 5.9. Вікно шляхів пошуку файлів

Вікно активується Set path кнопкою P Home-Envirinment-Set path стрічки.

При завантаженні пакета автоматично виконується команда **matlabrc**. Команда завантажує для виконання файл **matlabrc.m**, завантажує файл **startup.m**, та запускає скрипт з нього. В ці файл-сценарії можна додавати власні програми.

Зберігаються файли-сценарію на диску у вигляді текстових файлів з розширенням .*m*. Імена файлів можуть задаватися довільно.

Файл-функція є аналогом підпрограми-функції в мовах програмування високого рівня. В пакеті передбачена на порушення правил структурного програмування можливість створення файл-функцій, які не мають вхідних аргументів та вихідних параметрів. Такі файл-функції можна запускати введенням одного імені функції програмно або вручну з командного рядка.

Файл-функція складається з рядка заголовку, рядків коментарів та тіла функції.

Рекомендується завершувати опис функції в файлі службовим словом end.

Обов'язковим є виконання першого рядка у вигляді заголовка функції з наступним синтаксисом:

#### function out=im's(in),

де out – перелік імен вихідних результатів, in – перелік імен вхідних аргументів.

Якщо вихідних результатів декілька, їх перелік забирається в квадратні дужки. Елементи переліків вхідних аргументів та вихідних результатів відокремлюються комами.

За рядком заголовку розміщуються не обов'язкові рядки коментарів. Рядки коментарів відокремлюються від тексту програми порожнім рядком.

Рядки коментарів є рекомендованим елементом оформлення файл-функцій, бо команда допомого **help** використовує для виведення на екран саме текст, який оформлено як коментарі та розташовано між рядком заголовку функції та першим оператором функцій або першим порожнім рядком.

Перший рядок блоку коментарів називається **Hi-line** рядком. Зміст цього рядка виводиться командою пошуку функцій **lookfor**, тому рекомендується в цьому рядку вводити коротку інформацію про дію функції.

Зберігаються файли-функції на диску у вигляді текстових файлів з розширенням .*m*. Імена файлів повинні збігатися з іменами описаних в них функцій.

Блок коментарів файл-функції рекомендується оформляти за наступною схемою [29]:

```
function [out] = ім'я функції(in)
```

% 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 >

#### Змінні в програмах Matlab

Застосування змінних в файлах-сценаріях та файлах-функціях відрізняється.

В файлах-сценаріях всі змінні є спільними для файлів-сценаріїв та командного вікна пакету. Такі змінні називаються *глобальними*. Вони зберігаються весь час сеансу пакета в робочому просторі (WorkSpace). Тобто робочий простір є спільним для всіх файлів-сценаріїв та командного вікна.

В файлах-функціях змінні всередині файлу доступні тільки в файл-функції. Такі змінні називаються *локальними*. На початку виконання
функції для змінних, імена яких згадуються в її описі виділяється тимчасова область пам'яті. Після завершення функції всі локальні змінні функції прибираються з пам'яті та область пам'яті, яка була відведена під ці змінні, вивільняється. Змінні *робочого простору* та *командного вікна* всередині *файлфункції* не видимі та навпаки.

За необхідності спільного використання змінних в файл-функції та командному вікні або файлі-сценарію, змінні повинні бути визначені як глобальні у всіх файл-функціях та файлах-сценарію. Визначення глобальних змінних проводиться оператором global varl ... varN.

# Вкладені функції

Написання функції таким чином, щоб вона описувала всі програмні дії не завжди є оптимальним. Фрагменти, які повторюються, доцільно оформляти у вигляді окремих службових функцій. Опис службових функцій може проводитися в зовнішніх файл-функціях або безпосередньо в файлі головної функції. Службові функції в одному файлі з головною можуть бути виконані як *підфункції* (subfunction) або *вкладені* функції.

В одному m-файлі можливо одночасне використання підфункцій та вкладених функцій.

# Підфункції

Підфункція розташовується в файлі основної файл-функції після опису основної функції. В файлі файл-функції може міститися кілька підфункцій зі своїми вхідними і вихідними параметрами. Основна функція в файлі може бути тільки одна.

Підфункція є залежною від основної функції та може викликатися *тільки* з основної функції. Виклик підфункції ззовні з іншої файл-функції, файлу-сценарію, командного вікна є неможливим.

Заголовок нової підфункції є ознакою кінця попередньої. Помічати кінець основної функції та підфункцій службовим словом **end** не потрібно. Змінні в кожній підфункції і в основній функції є локальними, вони доступні тільки в межах опису своєї функції.

# Вкладені функції

Вкладені функції розташовується перед тілом головної функції після блоку коментарів.

Опис головної та кожної з вкладених функцій повинен завершуватися оператором **end**. Вкладена функція може мати свої вкладені функції. Основна функція може звертатися до своєї вкладеної функції, але не може викликати вкладену функцію нижчого рівня. Вкладена функція може звертатися до інших вкладених функцій свого рівня. Вкладена функція нижнього рівня може викликати функцію верхнього рівня, в яку вона вкладена.

Змінні головної та вкладених і неї функцій є спільними. Змінні основної функції доступні у *вкладеній* функції. Змінні вкладених функцій одного рівня є невидимими між ними, тобто є *локальними*.

Для введення текстів програм в пакеті можна використовувати вбудований редактор EDIT. Редактор активується оператором EDIT в командному вікні або пунктами Home-New-Script, Home-New-Function, Home-Open стрічки.

## Редактор М-файлів

Активація редактора призводить до появи контекстно чутливої вкладки стрічки **Editor** з елементуми керування редактором (рис. 5.10). Наповнення закладинки автоматично змінюється в залежності від режиму роботи редактора.

-		_											
́нс	ЭМΕ			APPS	EDIT				- 1 L B C	6	2?	Search Docum	entation
Nev	v	Open	Save	Find Files	<ul> <li></li></ul>	To 👻 Ca	Insert 🔜 ) omment % 🖇 Indent 🛐 🚽	x F4 → 2 27 1 E4	Breakpoints	► Run	Run and Advance	Nun Section	Run and Time
			FILE		NAVIGA	ATE	EDIT		BREAKPOINTS			RUN	
4	•	1		F: ▶ inf_t ▶									
	dite	or - F:\	inf_t∖pi	dr2021\fed.m									
	fed.	m ×	+										
1		8	8										
2 -	-	c.	lc;										
3 -	-	k=	=5;				8 initia	l bloc	k				
4 -	-	aı	nswe	r = 0;									
5		81	8										
6.	-	x=	=0;										
7 ·	-	. <b>₽f</b> ¢	or i	= 1 : k									
8 -	-		x	= x + (1	./(i^2))	;							
9.		L ei	nd										
10 .	-	aı	nswe	r <mark>=</mark> sqrt	(6*x)								
11													

Рис. 5.10. Редактор програм

Для полегшення сприйняття текстів редактор виділяє інформацію різних типів різними кольорами: оператори мови програмування — синій колір; функції, константи та змінні — чорний колір; коментарі (після символу «%») — зелений колір; символьні та текстові дані (в апострофах або лапках) — синій колір. Кольори можна визначити у вікні налаштувань **Home-Preferences-Colors**.

Для навігації по документу в редакторі реалізовано пошук по змісту та переходи по закладках (Bookmarks). Режим пошуку тексту активується кнопкою Find стрічки редактора, керування закладками проводиться з поля GoTo стрічки.

Швидко отримати доступ до довідникової системи пакета з інформацією про синтаксис написання та особливості застосування забезпечує кнопка **Function** групи **Insert** стрічки редактора.

Запуск програми проводиться кнопкою 🕨 Run стрічки редактора.

Перед виконанням сценарію редактор М-файлів проводить синтаксичну перевірку програмного коду. В разі виявлення помилок в командне вікно виводиться сповіщення та виконання сценарію переривається.

#### Налагодження програм

Обов'язковим етапом програмування є налагодження (debugging) програм. Для налагодження м-файлів в пакеті передбачене покрокове виконання програми, виконання програми по точках переривання (breakpoints) або виконання програми по фрагментах (sections).

Ознакою того, що пакет знаходиться в режимі налагодження програм є позначка «**К>>**» на початку рядка в командному вікні.

Перехід в режим налагодження відбувається автоматично за наявності в програмі точок переривання. Вихід з режиму налагодження проводиться

кнопкою 🧶 Quit Debugging стрічки.

#### Точки переривання

Точки переривання встановлюються на початку програмного рядка. В редакторі точки переривання позначаються червоною великою крапкою (рис. 5.11 а). Найпростіше встановити/прибрати точку переривання кліком обраного миші на номері рядка програми. Точки можна також встановити/прибрати з меню (рис. 5.11 б) кнопки **Breakpoints** стрічки редактора. З меню можна встановити/прибрати одну точку Set/Clear, активувати або дезактивувати точку **Enable/Disable**, прибрати всі точки одночасно Clear All. При виконанні програма буде виконана до точки переривання та призупинена.



Рис. 5.11. Налагодження програм: а – точки переривання; б – меню точок переривання; в – стрічка

Рядок, на якому програма призупинена позначається стрілкою. Після призупинення програми поточні значення змінних є доступними у вікні робочого простору та при наведенні курсора на відповідне місце рядка програми (рис. 5.11 а).

При призупиненні в точці переривання можна проконтролювати значення змінних у вікні робочого простору або шляхом введення імені змінної в командному вікні.

При призупиненні програми вміст вкладки стрічки редактора змінюється (рис. 5.11 в). Кнопка Run змінюється на кнопку Continue, з'являються кнопки виконання одного кроку програми Step, виконання одного кроку із заходженням в m-файли підпрограм-функцій Step In, виконання одного кроку без заходження в m-файли підпрограм-функцій Step Out, виконання до курсору Run to Cursor, кнопка виходу з налагодження Quit Debugging.

В версіях *Matlab* більше 8.х програма для налагодження може бути запущена тільки до першої точки переривання. Покрокове виконання програми стає доступним після призупинення програми в першій точці переривання.

#### Виконання фрагменту програми

Для перевірки може бути запущена не вся програма, а тільки її частина. Для цього програма може бути розбита на *секції* (Section).

Секція програми позначається розміщенням в рядку перед початком фрагменту двох знаків відсотків «%%». Фрагментом вважається частина програми, яка розташована після позначки до кінця тексту або до наступної

позначки. Проставити позначки можна вручну або кнопкою 🗟 Section групи Insert стрічки.

Виконання фрагмента без зміни виділення поточного фрагмента запускається кнопкою **Run Section** стрічки. Виконання поточного фрагменту із зсувом виділення поточного фрагменту до наступної позначки запускається кнопкою **Run and Advance** стрічки. Вибір поточного фрагменту проводиться кліком миші на обраному рядку програми. Поточний фрагмент виділяється жовтим кольором.

## Засоби реалізації алгоритмів

Керування обчислювальним процесом в пакеті проводиться вбудованими операторами *розгалуження*, *циклу* та *введення/виведення*.

## Керування розгалуженням

Розгалуження в пакеті здійснюється складеними операторами умовного nepexody if, множинної nepesipku switch, nepexody в разі помилки try. Оператори розгалуження починаються з одного з службових слів if, switch, try, містять після службового слова умову критерію перевірки та завершується службовим словом end.

В якості **умов** в операторах зазвичай використовують вирази логічних відносин або логічних операцій. Умова вважається виконаною, вірною, в разі, якщо результат виразу **умова** не дорівнює нулю.

Синтаксис оператора умовного переходу **іf** має наступний вигляд:

```
if умова 1
операції 1
<elseif> умова 2
операції 2
...
<elseif> умова К
операції К
<else>
операції K+1
end
```

Рядки перевірки додаткових умов **elseif** можуть повторюватися або бути відсутніми. Рядок **else** повинен розміщуватися в кінці оператора або бути відсутнім.

Якщо умова 1 є вірною, то оператор виконує рядки операції 1. Якщо умова 1 не виконується, послідовно проводяться перевірки elseif. Якщо якась умова elseif є вірною, то оператор виконує відповідну послідовність рядків операції К.

Якщо жодна з умов не є вірною, виконуються **<операції N**>, які вказано після слова **else**.

Перевірка умов завершується після першої виконаної умови.

Для логічної умови векторної змінної умова вважається виконаною, якщо вона є вірною для BCIX елементів вектора.

Синтаксис *оператора множинної перевірки* **switch** має наступний вигляд:

```
switch ім'я змінної або вираз
case значення 1 операції 1
<case> <значення 2> <операції 2>
...
<otherwise> <операції K>
end
```

Оператор множинної перевірки проводить розгалуження обчислювань по визначеному набору **значення i**. Оператор послідовно порівнює значення змінної або виразу з рядка **switch** зі значеннями, які вказані в рядках **case**. Відповідні рядки **операції і** виконується, якщо значення змінної або виразу з рядка **switch** співпадає зі значенням **значення і**, яке вказано після відповідного службового слова **case**. Якщо значення змінної або виразу з рядка **switch** не співпадає з жодним із значень в групах **case**, то виконуються рядки **операції К**, які розташовані за службовим словом **otherwise**.

Вираз в рядку **switch** повинен мати скалярний результат.

Перевірка умов завершується після першої вірної умови.

Синтаксис оператор *переходу в разі помилки* **try** має наступний вигляд:

```
try
оператори 1
catch
оператори 2
end
```

Оператор **try** виконує рядки **оператори 1**. Якщо при виконанні виникає помилка або результатом є невизначеність, то оператор виконує **оператори 2**.

## Оператори циклу

В пакеті можливо використання двох типів циклу – *циклу з передумовою* та *арифметичного циклу*. *Цикл з передумовою* призначено для алгоритмів, в яких заздалегідь невідома кількість повторювань. *Арифметичний цикл* використовується, коли кількість повторень циклу відома до початку циклу.

Оператори циклу є складеними операторами, які починається з одного з службових слів while, for та завершуються службовим словом end.

Синтаксис оператора циклу з передумовою має наступний вигляд:

while умова оператори end

Оператори перевіряє умову. В разі, якщо умова є вірною, виконуються оператори.

В якості **умови** можна використовувати логічні або арифметичні вирази. Умова вважається вірною, якщо її значення не дорівнює нулю.

Синтаксис оператора арифметичного циклу має наступний вигляд:

```
for count=vector
оператори
```

end

Цикл визначається змінною count, яка є лічильником циклу. Діапазон зміни лічильника циклу задається вектором vector.

Вектор може бути визначений векторизацією count=start:<step>:stop, де start  $\epsilon$  початковим значенням, step  $\epsilon$ 

значенням кроку, **stop** та кінцевим значенням. Для одиничного кроку параметр **step** можна не вказувати.

Вектор може бути визначений явно у вигляді вектору-рядка або матриці. Для матриці цикл повторюватиметься стільки разів, скільки стовпців має матриця. Наприклад, цикл for k=[2,5,7,1,1,3] або c=[2,5,7,1,1,3]; for k=c буде повторюватися 6 разів зі значеннями зі списку.

Наприклад, цикл **A=[1 2;3 4;5 6]; for k=A** буде повторюватися за кількістю стовпців в матриці А три рази. На кожному кроці змінна циклу **k** буде вектором-стовпцем зі значенням відповідного стовпця матриці.

Наприклад, фрагмент

S=zeros(2,1); A=[1 2;3 4];i=1;

for j=A; S(i)=S(i)+sqrt(j(1)^2+j(2)^2); i=i+1;end

обчислює суму квадратів елементів стовпців матриці А. Цикл повторюється два рази.

## Переривання дій

Для дострокового переривання циклу або дострокового виходу з функції передбачені вбудовані оператори break, continue, return.

Оператор **break** достроково перериває циклу та передає керування першому за циклом рядку. Зазвичай використовується разом з умовним оператором **if**, який визначає умову переривання циклу. Слід зауважити, що використання оператора скоріш за все показує, що алгоритм складений невдало.

Оператор **continue** перериває крок циклу. Він обминає виконання операторів, які знаходяться нижче оператора **continue**, та починає наступну ітерацію циклу. Зазвичай використовується разом з умовним оператором **if**, який визначає умову переривання циклу.

Оператор **return** перериває виконання функції та передає керування в точку виклику функції. Зазвичай використовується разом з умовним оператором **if**, який визначає умову переривання функції. Схожу з оператором return дію виконує оператор error ('message'). Він зупиняє виконання програми та виводить в командне вікно текст 'message' червоним кольором.

## Організація діалогу з корисувачем

Взаємодія між користувачем та М-файлом або командним вікном в пакеті проводиться за допомогою операторів введення/виведення disp, sprintf, fprintf, input, menu, keyboard, pause.

Оператор **disp (аргумент**) виводить на екран в командне вікно значення **аргумент**. Оператор **disp** сприймає тільки один аргумент, яким може бути ім'я відомої змінної, число або текстовий рядок.

В *Matlab* крім «класичних» текстових рядків, які позначаються апострофами та являють собою *масиви символів* (*character array*), припустиме використання текстових рядків, які позначаються подвійними лапками та являють собою *масиви текстів* (*string array*).

Наприклад, рядки **a='aa'**; **b='bb'**; є масивами символів [1x2 char], а їхнє поєднання в вектор c=[a b] є масивом символів [1x4 char] та виводиться як єдине ціле '**aabb**'. Сформувати з масивів символів один текстовий рядок арифметичним додаванням *неможливо*, потрібно використовувати вбудовані функції обробки символів.

Рядки **a="aa"**; **b="bb"**; є масивами текстів [**1x2 string**], а їхнє поєднання в вектор **c=[a b]** є масивом текстів [**1x2 string**] та виводиться поелементно **"aa" "bb"**. З масивів текстів можна сформувати один текстовий рядок арифметичним додаванням.

В текстові рядки можливо вводити спеціальні символи за допомогою модифікаторів. Ознакою модифікації слугує символ зворотного слешу «\». Наприклад, літеру  $\pi$  вводять як **\pi**. Коди деяких модифікаторів наведено в таблиці 5.2. Дія модифікаторів розповсюджується на один символ, до кінця рядку, на групу символів, яка забрана в фігурні дужки.

Щоб вивести значення кількох змінних за допомогою команди **disp**, потрібно створити єдиний об'єкт, який би містив всі значення.

Для однотипних даних це можна зробити об'єднанням в вектор-рядок.

Таблиця 5.2. Модифікатори символів [2]

Вигляд	Код	Вигляд	Код
$\rightarrow$	\rightarrow	text	\rm{text}
text	\bf {text}	text	\fontname{name}
text	\fontsize{size}{text}	text	\it {text}
Xa	x_{a}	x <sup>b</sup>	x^{b}
α	\alpha	φ	<b>\Phi</b>
β	\beta	Δ	\Delta
γ	\gamma	Γ	\Gamma

Наприклад, запис

>> x1=5; x2=5; x3=5; x=[x1 x2 x3];

abo >>x=[5 5 5] abo >>x(1)=x1; x(2)=x2; x(3)x3;

>>disp(x) або >>disp([5 5 5]) або >>disp([x1 x2 x3]) виведе на екран

5 5 5

Наприклад, запис

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

виведе на екран

#### sifitetaw1

Елементи вектора не можуть мати різний тип, вони можуть бути або всі числами, або текстовими рядками. Для виведення оператором **disp** комбінованої інформації з чисел та тексту за допомогою вектора потрібно перевести всі чисельні значення в текстові за допомогою вбудованих функцій перетворення чисел в рядки або використати вбудовану функцію створення форматованого за зразком текстового рядка **sprintf**.

Функція **sprintf** має наступний синтаксис:

## <[> S <, ERRMSG]> = sprintf(format, A<, B,...>).

Результатами функції є текстовий рядок символів **S** та опціональне поле **ERRMSG**, яке містить повідомлення про помилку.

Функція **sprintf** створює текстовий рядок, який містить значення числових та текстових аргументів **A**, **B** у визначеному вигляді. Вигляд рядка задається аргументом текстовим рядком **format**. Кожен з аргументів може бути скаляром, вектором, матрицею, виразом. Аргументи відокремлюються комами. Якщо аргумент є вектором, то результуючий рядок формується конкатенацією всіх елементів.

Аргумент **format** є рядком символів в апострофах або рядком текстів в лапках. Він містить вміст текстового рядка, який буде виведено на екран. На місці, де в рядку повинні розташовуватися значення аргументів **A**, **B** вставляються позначки формату числа.

Позначки формату чисел мають наступну структуру **%FW. PST**.

Поле **F** містить ознаку виведення додаткових символів перед числом: символ «-» означає ліве вирівнювання, символ «+» означає обов'язкове виведення знаку числа.

Поле W містить число, яке позначає загальну кількість символів в числі.

Поле Р містить число, яке позначає кількість символів в числі після десяткової точки.

Поле S містить ознаку формату числа подібно до функції format: d(decimal),i(integer), f(fixed point), e(exponential), g (general).

Наприклад, фрагмент

x=1/4; y=5; sprintf('x= %f a= %i', x, y)

створить рядок 'x=0.25000 y=5',

sprintf('x= %4.2f y=%i', x, y)

створить рядок 'x=0.25 y=5'

sprintf('x= %e y= %i', x, y)

створить рядок 'x= 2.500000e-001 a= 5'

Вивести комбіновану інформацію з чисел та масивів тексту оператором **disp** можна також поєднанням в один масив текстів операцією арифметичного додавання чисельних значень та текстів в лапках.

Oператор warning ('message') виводить в командне вікно текст 'message' червоним кольором.

Замість використання для виведення інформації двох функцій disp, sprintf можна застосувати одну функцію fprintf.

Функція має наступний синтаксис

## fprintf(FID, format, A, B, ...).

Функція призначена для виведення інформації аргументів **A**, **B** у вигляді, який визначений аргументом **format**, в дисковий текстовий файл, який визначений аргументом **FID**. Якщо ввести для аргументу **FID** значення 1 або взагалі не вводити його, то виведення буде проведено на екран. Інші аргументи **fprintf** функції співпадають з аргументами функції **sprintf**.

Функція input (<'запрошення',> < 's'>) виводить рядок запрошення на екран командного вікна та зчитує введене користувачем з клавіатури. Масиви вводяться списком в квадратних дужках.

Наприклад, **c=input('c=')**.

В разі наявності опціонального параметру 's', функція сприймає введене значення як текстовий рядок.

Функція **menu** виводить на екран вікно з меню користувача для вибору із запропонованих варіантів.

Функція має наступний синтаксис:

k=menu('Перший рядок заголовка','Варіант1',... 'Варіант2','Варіант n').

Після виклику функції виконання програми тимчасово переривається. На екран виводиться діалогове вікно (рис. 5.11). Пакет чекає натискання однієї з кнопок меню з варіантами. Функція повертає номер обраного варіанту. Кількість варіантів обмежена значенням 32. Зачинення вікна без вибору варіанта повертає 0.

Функція тимчасово перериває виконання програми до тих пір, доки користувач не натисне будь яку клавішу клавіатури. Виклик **pause** (n) проведе затримку на **n** секунд.

Функція **keyboard** перериває виконання програми з m-файлу та передає керування клавіатурі (командному вікну). Ознакою передачі керування є наявність символів **K>>** на початку командного рядка. Користувач може проводити в командному вікні будь-які дії. Повернення до виконання програми здійснюється введенням команди **return**.

## Контрольні запитання

- 1. Назвіть різновиди програм Matlab.
- 2. Назвіть можливості редактора М-файлів.
- 3. Для чого застосовуються комірки при програмуванні?
- 4. Як запускається скрипт?
- 5. Яку структуру має файл-функція?
- 6. Як запустити файл-функцію?
- 7. Для чого застосовуються підфункції, вкладені функції?
- 8. Яка повна кількість логічних гілок в повному операторі if?
- 9. Якими операторами реалізуються цикли?
- 10. Якими засобами реалізується введення-виведення інформації?

## 5.4. Двовимірні графіки

Пакет дозволяє створити багато типів двовимірних графіків вбудованими функціями. Основні типи двовимірних графіків пакета наведено в таблиці 5.3.



Рис. 5.12. Графічне вікно: a – plot: б – area; в – semilogx

Створення та налаштування графіків може проводитися програмно з повним визначенням вигляду графіка з командного вікна, скрипт-файлів, файл-функцій або в діалоговому режимі зі спрощеним визначенням вигляду графіка з вікна робочого простору та стрічки.

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	$\sim$	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

Таблиця 5.3. Основні типи двовимірних графіків [1]

# Діалогове керування графікою

За умови наявності в робочому просторі векторних змінних, для яких можуть бути побудовані графіки, автоматично активуються кнопки графіків вкладки **PLOTS** стрічки (рис. 5.13).



Рис. 5.13. Стрічка графіки

Алгоритм побудови графіка з робочого простору є прости та полягає в наступному:

 виділити в робочому просторі змінні аргументу та функції або функції для графіка. Виділені змінні повинні бути векторами однакової довжини. Множинний вибір проводиться стандартним для застосунків способом – мишею із затисненою клавішею «*Ctrl*». Змінні стають аргументами відповідної команди малювання графіка, тому їх слід обирати в тому порядку, в якому вони повинні бути вписані в графічні функції;

- клікнути на стрічці кнопку потрібного виду графіка.

Пакет автоматично згенерує в командному вікні потрібну вбудовану графічну функцію та в графічному вікні виведе визначений графік. Місце виведення графіка в поточне вікно **Reuse Figure** або в нове вікно **New Figure** можна задати відповідними полями стрічки. Поля знаходяться з правого боку стрічки (рис. 5.12).

Слід відзначити, що буде намальовано типовий графік з параметрами за замовчанням та без елементів оформлення. Для налаштування вигляду графіка слід додатково застосувати програмні засоби або скористатися редагуванням з вікна графіки.

## Редагування графіків в графічному вікні

Графічне вікно не тільки відображає графіки, в нього вмонтовані потужні інструменти для оформлення графіків. Вікно з активованими вбудованими засобами має вигляд, зображений на рис. 5.14.

Вікно містить поле графіка, меню керування вікном, редактор властивостей графічних елементів **Property editor**, панель вікна **Figure palette**, браузер графічних елементів **Plot browser**, інспектор властивостей графічних елементів **Inspector**, кнопкові панелі вікна **Figure Toolbar**, камери 3D перегляду **Camera Toolbar**, анотацій **Plot Toolbar**.

▲ Figures - Figure 1		- 🗆 🗙	🛤 Inspector: matla	- 🗆	$\times$
File Edit View Insert Tools Debug Desktop Window Help меню вікна		X 5 K	E <b>2↓</b> € * = \$		
🗅 😅 🚽 🌭 🔍 🔍 🕲 🥷 🖌 🔜 🔲 📰 🔲 кнопкова па	нель вікна		BeingDeleted	Off	^
🗙 🕷 🛊 🚉 🖑 🛊 й й й й ү 💡 🗐 🛄 🗓 🤷 кнопкова па	нель 3D перегляду		BusyAction	queue	*
	анель анотацій		ButtonDownFcn	_	0
	Dist Province		Clipping	✓ On	
	Plot browser	U	CloseRequestFcn	closereq	
			Color	(2)	
L 2D Axes B			CreateFcn		
Se 3D Axes III I	Engineen reachi		CurrentCharacter		0
1.5	елементів		dia CurrentPoint	[24,333 221	d
▼ Variables	Chericinia		DeleteFcn	_	- 1
х 1x51 1 поле графіка	-		DockControls	⊻ On	- 1
y 1x51			FileName	_	Ø
▼ Annotations	інспектор		GraphicsSmoothing	⊻ On	- 1
Line 0.5 панель вікна	властивос	теи	HandleVisibility	on	۳
Arrow			InnerPosition	[1 1 425,333	3 26
> Double Arrow			IntegerHandle	✓ On	
		Add Data	Interruptible	🗹 On	
		7 tot Dutti	InvertHardcopy	🗹 On	
Property Editor - Figure		۲	KeyPressFcn	3×1 cell arra	a 🏿
Figure Name:	Show Figure Number	More Properties	KeyReleaseFcn		0
Colormap: Parula Y		Export Setup	MenuBar	figure	٣
Figure Color:			Name		Ø
			NextPlot	add	*
			Number	1.0	

Рис. 5.14. Графічне вікно

Кнопкова панель 3D перегляду дозволяє налаштувати просторову орієнтацію зображення у вікні шляхом обертання камери навколо зображення Orbit Camera, орієнтації джерела світла відносно об'єкта Orbit Scene кутового зсуву об'єкта відносно камери Pan/Tilt Light. Camera, переміщення об'єкта В полі зору камери Move Camera Horizontally/Vertically, зміни відстані між камерою та об'єктом Move Camera Forward/Back, зміни збільшення камери Zoom Camera, повороту камери навколо осі зору Roll Camera.

Меню графічного вікна містить команди для керування самим вікном та інструментами налаштування вигляду графіків. Команди поєднані в групи [31].

Група File містить стандартні файлові дії Windows застосунків.

Група **Edit** містить:

- стандартні команди для перенесення графіків між графічними вікнами;
- пункт Figure properties активації редактора властивостей Property editor в режимі редагування вікна;
- пункт Current object properties активації редактора властивостей Property editor в режимі редагування елементу графіка, який обере користувач кліком миші на елементі;
- пункт **Colormap** активації редактора карти кольорів, яка використовується для розфарбування тривимірних графіків **Colormap editor**.

Група **View** керує виведенням на екран кнопкових панелей, редактора властивостей графічних елементів, панелі вікна, браузера графічних елементів.

Група Insert повторює кнопкову 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 кнопкову панель вікна та керує опціями перегляду графіка: масштабуванням, зсувом, поворотом.

Вигляд та можливості редактора визначаються типом виділеного графічного елементу. Виділення елементу для редагування проводиться кліком миші на ньому. Ознакою обрання об'єкта є поява чорних прямокутних маркерів на ньому. При цьому вміст полів редактора автоматично налаштовується на параметри виділеного об'єкта. Обрати параметр та дію для редагування можна також за допомогою контекстного меню виділеного об'єкта.

Для редагування властивостей вікна **figure** є доступними ім'я вікна та фоновий колів вікна.

На рис. 5.16 а показано вигляд редактора в режимі редагування властивостей осей **axis**. Для редагування є доступними написи на осях, колір та шрифт написів, тип осі лінійна або логарифмічна, діапазон аргументів осі, сітка.



Рис. 5.16. Редактор: а – редагування осей; б – редагування ліній

На рис. 5.16 б показано вигляд редактора в режимі редагування властивостей графіка. Для редагування є доступними тип, колір, товщина ліній, тип та колір маркерів тощо.

Додавання елементів оформлення на поле графіка, таких як підписи осей та графіка, описи ліній, карта кольорів, стрілки, прямокутники тощо, можна провести через групу Insert меню вікна, панелі анотації, панель вікна Figure Palette. Розміщення елементів та визначення їхніх розмірів проводиться безпосередньо на полі графіка в діалоговому режимі за допомогою миші. Додавання елементів з панелі вікна Figure Palette проводиться методом «перетягування» відповідних зображень зі списку панелі на поле графіка. З групи **Subplots** панелі вікна **Figure Palette** також можливо керувати кількістю підобластей з полями графіка.

З браузера графічних елементів **Plot browser** можна додавати нові графіки на існуюче поле графіка та прибирати з поля непотрібні графіки. Пунктом **Delete** контекстного меню графік прибирається з поля графіки, кнопкою **Add Data** браузера – додається.

Відредаговані елементи не мають програмного коду, тому зберігати їх для перенесення та подальшого використання потрібно в самому графічному вікні у файлах з розширенням .fig.

## Програмне керування графікою

Для створення нового та перемикання між існуючими графічними вікнами використовується функція <**H>=figure** <(n)>.

Виклик без параметра **n** створює нове вікно для графіка. Створеному вікну надається автоматично номер, який може бути використаний для програмної ідентифікації вікна. Кожен наступний виклик без параметра створює нове вікна зі збільшеним на одиницю номером. Перший виклик створює вікно з номером 1, другий виклик – 2 тощо. Останнє створене вікно стає поточним. Графіки, які будуть генеруватися після створення поточного вікна, будуть виводитися за замовчанням в поточне вікно.

Виклик з номером вікна **n** в разі відсутності створює нове вікно з вказаним номером, в разі наявності вікна з вказаним номером – воно стає поточним.

Опціональний результат **н** містить посилання на графічне вікно для програмної ідентифікації.

Закрити вікно можна командою **close(<n>)**. В разі відсутності аргументу закривається поточне вікно, в разі наявності закривається вікно з вказаним номером або ідентифікатором.

Команда **clf** очищує поточне графічне вікно.

Створення графіків в декартових координатах проводиться функцією

## <H>=plot(<x1>,y1,<s1>,<x2>,<y2>,<s2>,...).

Функція виводить в поточне вікно графіки з координатами абсцис з векторів **хі** та координатами ординат з векторів **уі**. Вектори можуть

визначатися явно, векторно, іменами змінних. Кілька графіків визначаються почерговим визначенням через кому пар векторів **х**, **у**. В разі відсутності аргументу **х** будується графік залежності елементів вектора **у** від номера елементу вектора. Вигляд графіка визначається опціональним аргументом **si** або послідовністю пар «властивість,значення» в термінах низькорівневого програмування. Аргумент **si** є текстовим трьохелементним рядком 'лінія маркер колір'.

Типи ліній позначаються наступними символами: «-» – суцільна, «:» – точкова, «-.» – штрих-пунктирна, «--» – пунктирна.

Кольори ліній позначаються наступним символами: «b» – синій, «g» – зелений, «r» – червоний, «c» – рожевий, «m» – блакитний, «y» – жовтий, «k» – чорний, «w» – білий.

Типи символів на лінії графіка позначаються наступними символами: « .» – точка, «о» – коло, «х» – хрест, «+» – плюс, «\*» – зірка, «s» – квадрат, «d» – ромб, «v»– трикутник вниз, «^» – трикутник вгору, «<» – трикутник вліво, «>» – трикутник вправо, «р» – п'ятикутник, «h» – шестикутник.

Наприклад, запис

## >>x=0:0.1:6; y=cos(x); plot(x,sin(x),'-g',x,y,':xr')

будує в графічному вікні графіки функцій f(x) = sin(x) та f(x) = cos(x) для значень аргументу х від 0 до 6 з кроком 0.1. Перший графік малюється безперервною лінією зеленим кольором, другий графік малюється червоною пунктирною лінією з символом «х» в проміжках (рис. 5.12 а).

Вбудована функція linspace(first, last, n) дозволяє визначити вектор з рівномірно розподіленими елементуми від першого first до останнього last з їхньою кількістю n. Наприклад, аргумент для попереднього прикладу: x=linspace(0,6,).

Функція < [X, Y] >= fplot (F, <L>, <S>) малює декартовий графік функції **F** в діапазоні значень аргументу **L** типом лінії **<S>**. Функція **F** може залаватися рядком символів 3 іменем файл-функції користувача посиланням на файл-функцію **@myfun**, безпосереднім 'myfun(x)', записом функції як анонімної @**x\*sin(1./x)**. Діапазон значень аргументу **L** визначається двоелементним вектором [**XMIN XMAX**]. В разі відсутності аргументу **L** графіки малюється в діапазоні аргументу -5<x< 5. Тип лінії в аргументі **S** визначається аналогічно команді **plot**.

Вектори опціональних результатів **х**, **х** містять значення координат графіка.

Наприклад, запис

## >>fplot('exp(sin(x))',[0 pi],':r')

малює точками графік виразу  $e^{\sin(x)}$  для значень аргументу **x** від 0 до  $\pi$  точковою лінією червоним кольором.

Спрощеним аналогом команди fplot с команда

## ezplot(FUNX<,FUNY>,<[TMIN,TMAX]>,<FIG>).

Функція малює параметричний графік виразів **FUNX (T)** та **FUNY (T)** в діапазоні [**TMIN, TMAX**] у графічному вікні з ідентифікатором **FIG**. В разі відсутності аргументу **FUNY** малюється графік виразу функції **FUNX (X)**. В разі відсутності аргументу **[TMIN, TMAX]** малюється графік в діапазоні  $-2\pi < X < 2\pi$ . В разі відсутності аргументу **FIG** малюється графік в поточному вікні.

Декартовий графік без ліній між точками (точкову діаграму) малює функція <H>=scatter(<p,>X,Y,<S>,<C>,<M>,<'filled'>). Точки графіка малюються розміром S кольором C маркером M.. Опціональний аргумент 'filled' замальовує маркер. Колір точки визначається символом назви ('r', 'g' тощо). За замовчанням розмір маркера 36.

Наприклад, запис scatter(x,y,10,'r','filled','d') малює залежність у вигляді зафарбованих червоних ромбів розміром 10.

Декартовий графік, подібний до графіку функції plot, із зафарбованою областю під графіком малюється функцією **(X>, Y, (Level>)** В разі, коли аргумент **Y** є матрицею, будуються графіки стовпців матриці з накопиченням значень. Опційний параметр **Level** є числом в одиницях абсцис графіка, яке визначає горизонтальну нижню межу фарбування графіка. Наприклад, запис **>>area**(**x**, **y**+1, 0.1, 'FaceColor', 'red') малює зафарбований декартовий графік з межею фарбування 0.1 червоним кольором (рис. 5.12 б).

Декартовий графік функції в логарифмічному масштабі по обом осям, в напівлогарифмічному масштабі по осі X або Y малюють функції <H>=loglog(...),<H>=semilogx(...),<H>=semilogy(...) відповідно. Синтаксис функції є аналогічним команді plot. Наприклад, запис >>semilogx(x,y,'--k') малює декартовий графік з логарифмічним масштабом по осі X за умовами попереднього прикладу чорним кольором штриховою лінією.

Функція logspace (X1, X2, n) дозволяє визначити вектор з логарифмічно розподіленими n елементуми в діапазоні від 10<sup>X1</sup> до 10<sup>X2</sup>.

Малювання декартових графіків на двох осях проводиться із застосовуванням команду **ууахів**. Команда визначає бік, в якому буде намальовано вісь графіка. Записується команда перед функцією малювання та діє до наступної зміни боку осі. Для виходу з режиму двох осей слід застосувати команду очищення графічного вікна **clf**.

Команда має наступний синтаксис:

yyaxis(<H>, 'mode') ado yyaxis mode.

Аргумент mode задається службовим словом left або right.

Наприклад, наступний фрагмент

```
x=0:0.1:10; y=sin(3*x); z=sin(3*x).*exp(0.5*x);
yyaxis left; plot(x,y)
yyaxis right; plot(x,z)
```

визначає ліве розташування осі та малює графік y(x), потім визначає праве розташування осі та малює графік z(x) (рис. 5.18 а).



Рис. 5.18. Двомірні графіки: а – **ууахіs**; б – **polar**; в – **bar**; г – **errorbar** 

Аналогічну дію за один крок виконує функція [AX,H1,H2]=plotyy(X1,Y1,X2,Y2,<F1>,<F2>). Опціональні аргументи <F1>,<F2> у вигляді текстових рядків, що забрані в апострофи вказують тип графіка: plot, semilogx, semilogy, loglog, stem тощо.

Полярний графік на осях з посиланням р залежності Ri (Ai) для кута Ai, визначеного в радіанах, малюють функції <H>=polar(<p,> A,R,<S>) та <H>=polarplot(<p,Al>,Rl,<Sl,...,AN,RN,SN>). Аргументи функцій визначаються аналогічно функції plot. Функція polar потребує обов'язкового задання аргументу кута Al та малює один графік. Функція polarplot не вимагає обов'язкового задання аргументів кута Ai та може малювати кілька графіків. В разі відсутності аргументу кута функція малює залежність для діапазону 0 ...  $2\pi$  з рівномірним кроком. Значення кутів на графіку виводяться в градусах.

Наприклад, запис

## >>A=0:0.01:2\*pi;R=sin(2\*A).\*cos(2\*A); polar(A,R)

виведе графік, який зображений на рис. 5.18 б.

Стовпцеву діаграму y(a) на осях з посиланням p з підписами a, шириною стовпця w малює функція  $\langle H \rangle = bar(\langle p, x \rangle, y, \langle w \rangle, \langle col \rangle)$ . Якщо аргумент a не заданий, то малюється стовпова діаграма y(i), де i номер елементу вектора y. Опціональний аргумент col задає колір стовпців діаграми символом назви ('r', 'g' тощо). Ширина визначає дійсним числом долю від максимальної.

Функція **<H>=barh (<p, x>, y, <w>, <color>)** виводить стовпову горизонтальну діаграму.

Наприклад, запис

>>y=[7 9 23 31 50 79 103 126 149 181];bar(y,0.3,'r') виведе графік, який зображений на рис. 5.18 в.

Лінійну діаграму на осях з посиланням **p** зі стовпцями нульової товщини та маркерами на кінці ліній малює функція **<H>=stem(<p,x>,y,<linespec>,<'filled'>)**. Функція малює лінії з ординатою **y** та абсцисою **x**, тобто залежність **y(x)** у вигляді вертикальних ліній. Аргумент **linespec** визначає спосіб малювання ліній діаграми аналогічно функції plot. Додаткове слово 'filled' замальовує маркери кінця ліній.

Лінійний графік y(x) з нанесенням діапазону зміни значень **Y** малює функція **<H>=errorbar(X,Y<,L,U, linespec, type>**). Верхні відхилення визначаються вектором **U**, нижні відхилення визначаються вектором **L**. Якщо заданий один аргумент **L**, вважається, що діапазон є симетричним. Аргумент **linespec** визначає спосіб малювання ліній діаграми аналогічно функції **plot**. Опціональний параметр **type** змінює вигляд діапазону змін. Без параметра зміни відображаються вертикально, значення **'horizontal'** задає горизонтальне зображення, **'both'** – обидва разом.

Наприклад, запис

>>x = 1:10:100; y = [20 30 45 40 60 65 80 75 95 90]; >>err=[1 3 5 3 5 3 6 4 3 3]; errorbar(x,y,err,'horizontal') виведе графік, який зображений на рис. 5.18 г.

Гістограму розподілу Y(X) у вигляді стовпової діаграми залежності кількості потраплянь значень Y в кожен піддіапазон малює функція [n,xc]=hist(<p,>Y,<,X>). Аргумент X може бути вектором зі значеннями піддіапазонів розподілу, або числом, яке задає кількість підінтервалів. За відсутності X графік малюється для 10 інтервалів. Опціональний результат n містить кількість інтервалів, xc – вектор координат центрів інтервалів. Розширені налаштування парами «властивість-значення» дозволяє функція histogram.

*Секторну діаграму* малює функція **<H>=pie (<p,>Y,X)**.

Для зображення векторів та комплексних чисел у вигляді векторних полів призначено команди compass та feather.

Функція **<H>=compass (<p,>X,<Y,linespec>)** малює графік в полярних координатах. Вважається, що вектори мають початкову точку (0,0) та кінцеву точку, яка визначена елементуми векторів **X,Y**.

Функція **<H>=feather(<p,>X,<Y,linespec>)** малює векторне поле в декартових координатах. Вважається, що початкові точки векторів лежать на осі абсцис, а кінцеві визначені елементуми векторів **X,Y**. Якщо застосовано один комплексний аргумент X, координати беруться з його дійсної та уявної частин. Деякі функції дозволяють вивести на одні осі кілька графіків безпосередньо під час виклику функції. Накласти декілька будь-яких графіків без витирання раніше намальованих можна командою-перемикачем hold. Команда hold on вмикає режим малювання графіків зверх вже намальованих. Команда hold off повертає режим малювання графіків з витиранням вже намальованих.

Для створення в одному графічному вікні кількох підобластей, в кожну з яких можна вивести графіки функцій, призначена функція <h>=subplot(m,n,p).

Аргументами функції є кількість стовпців **m**, кількість рядків **n** та номер поточного поля графіка **p**, в якому буде малюватися графік. Підобласті нумеруються зліва направо та згори вниз. Кілька підобластей можна поєднувати в одну. Для цього номер **p** визначається у вигляді вектора, елементуми якого є номери поєднаних підобластей.

В таблиці 5.4 наведено приклади організації графіків в одному вікні.

В деяких версіях для розділення графічного вікна на кілька під областей може використовуватися функція

## <H>=tiledlayout(<p,> m,n<,'flow'>)

Функція розділяє графічне вікно з ідентифікатором **p** на **m** рядків та **n** стовпців. Виклик функції без опціонального аргументу '**flow**' виводить **mxn** полів осей графіка відразу після виклику функції. Опціональний аргумент '**flow**' призводить до того, що поля осей графіків з'являються на екрані не одночасно, а по мірі малювання графіків.

На першому етапі підготовки функцією вікно розбивається на під області. Другий етап малювання складається з послідовних кроків активації підобласті функцією **nexttile** та дій малювання. Крок активація, малювання повторюється для кожної підобласті вікна.

Наприклад, малювання функцій **y=sin(3\*x)** та **z=cos(2\*x)** для аргументу від 0 до 10 з кроком 0.1 в двох областях в одному рядку в одному вікні (рис. 5.19) може мати наступний запис:

```
x=0:0.1:10; y=sin(3*x); z=cos(2*x);
tiledlayout(1,2);
nexttile; plot(x,y)
nexttile; plot(x,z)
```

або

```
x=0:0.1:10; y=sin(3*x); z=cos(2*x);
subplot(2,1,1); plot(x,y)
subplot(2,1,2); plot(x,z)
```

Таблиця 5.4. Приклади виведення графіків в одному вікні [2]



## Додаткові графічні елементи оформлення

На графіки можна програмно додати елементи оформлення на кшталт сітки, написів на осях, назви графіків.

Оформлення графіків програмно проводиться ПІСЛЯ малювання графіків у вікні.

Сітку на полі графіка малює команда grid <mode>. Крок сітки координат завжди відповідає цілим значенням аргументу. Виклик функції без аргументу перемикає видимість сітки. Виклик grid on включає відображення сітки, виклик grid off виключає відображення сітки. Додаткове службове слово minor включає відображення мілкої сітки.

За замовчанням автоматично на вісь абсцис виводяться всього діапазону аргументу графіка, на вісь ординат виводяться розраховані значення функції, початок відліку системи координат розташовується в нижньому лівому куті ( режими **auto**, **on**, **xy**).

Змінити зображення осей графіка з ідентифікатором р можна функцією ((<p,><[xmin, xmax, ymin, ymax]>,<mode>).

Аргумент-вектор [xmin, xmax, ymin, ymax] явно задає діапазони значень по осях графіка. Запис аргументу встановлює режим manual, який відміняє автоматичний розрахунок діапазонів по обом осям.

Опціональний аргумент встановлення режиму mode значенням:

- **OFF/ON** вмикає/вимикає відображення осей;
- **auto** встановлює автоматичний розрахунок діапазонів по обом осям;
- іј встановлює відлік по осі у згори вниз;
- ху встановлює відлік по осі у знизу вгору;
- equal встановлює однаковий масштаб по обох осях графіка;
- square встановлює квадратну форму поля вікна.

Додатковий аргумент tight стягує зображення до меж графіка.

Без визначення вектора масштабу ключові слова слід писати без дужок. Визначення масштабу потребує написання команди з аргументами в дужках, а ключових слів – в апострофах.

Наприклад, axis ij tight ado axis([-0 pi 0 1],'xy').

Назву графіка Title над полем графіка над графіком виводить функція <h>=title ('Title').

Підпис **Title X** під віссю X виводить функція **<H>=xlabel(<p,>'Title X')**.

Підпис Title Y зліва від осі Y виводить функція <H>=ylabel(<p,>'Title Y').

Текст **Text** на полі графіка з посиланням **р** в точку з координатами **х**, **у** виводить функція

# <H>=text(,x,y,'Text',<'FontName', 'Font',... ''FontSize', 'Size'>)

Координати **х**, **у** повинні бути задано в одиницях величин, в яких будується графік та знаходитись всередині діапазону цих величин. Опційні аргументи є парами «властивість-значення» застосовуються для визначення параметрів шрифту тексту.

Опис ліній (легенду) на поле графіка виводить функція

## <[L,OB,OU,OT]>=legend(string <,mode,'Property','Value'>).

Аргумент string є текстовими поясненнями до ліній графіка у вигляді символьних рядків через кому 'Line1','Line2',... або масивом комірок {'Line1','Line2',...}.

Символьний рядок mode вмикає значенням 'SHOW' видимість опису або вимикає видимість значенням 'OFF'.

Властивість 'вох 'вмикає 'boxon' видимість рамки навколо опису або вимикає видимість 'boxoff'.

Властивість 'Orientation' визначає горизонтальне 'horizontal' або вертикальне 'vertical' розташування опису.

Властивість 'Location' визначає положення опису на полі графіка: 'North' — всередині згори, 'BestOutside' — автоматично ззовні, 'South' — всередині знизу, 'Best' — автоматично всередині, 'East' всередині праворуч, 'WestOutside' — ззовні ліворуч, 'West' всередині ліворуч, 'EastOutside' — ззовні праворуч, 'NorthEast' всередині згори праворуч, 'SouthOutside' — ззовні внизу, 'NorthWest' — всередині згори ліворуч, 'NorthOutside' — ззовні згори, 'SouthEast' - всередині внизу праворуч, 'SouthWest' всередині внизу ліворуч, 'NorthEastOutside' - ззовні згори праворуч,
'SouthEastOutside' - ззовні згори праворуч, 'NorthWestOutside' ззовні згори ліворуч, 'SouthWestOutside' - ззовні внизу ліворуч.

Додаткові графічні елементи оформлення, як то прямокутник, еліпс, текст в рамці тощо на поле графіка з посиланням р виводить функція <h>=annotation(<p,>TYPE<,pos>).

Символьний рядок **TYPE** визначає тип елементу у вигляді прямокутника 'rectangle', еліпса 'ellipse', тексту 'textbox', лінії 'line', стрілки 'arrow', двосторонньої стрілки 'doublearrow', стрілки з текстом 'textarrow'. Координати елементу визначаються у векторі **pos** у відносних величинах розміру поля графіка від 0 до 1. Для прямокутника, еліпса, тексту координати описуються 4-ма елементуми вектора **pos** координатами х, у нижнього лівого кута, довжиною та висотою елементу. Для стрілки, лінії аргумент **pos** складається з двох двоелементних векторів. Перший містить координати х початку та кінця стрілки. Другий – координати у. Елемент 'textbox' визначається парою 'String', 'Value'.

Наприклад, програмне оформлення графіків з попереднього опису функції **plot** (рис. 5.12) з використанням горизонтальної легенди по центру вгорі графіків, сітки, назви поля «Приклад оформлення», підписів осі X «Вісь абсцис», Y – «Вісь ординат», прямокутника з написом «Зона нуля» в центрі, стрілки та тексту «Максимуми» в зоні максимуму може виглядати наступним чином (рис. 5.19):

```
>>legend('Line 1','Line 2','Location','north',...
'Orientation','horizontal')
>>grid on; xlabel('Bicь абсцис'); title('Приклад
формлення'); ylabel('Bicь ординат')
>>annotation('textbox',[0.4 0.5 0.3 0.1],'String',...
'Зона нуля')
>>annotation('arrow',[.5 .6],[0.8 0.9])
>>text(-1,0.6,'Максимуми')
```



Рис. 5.19. Приклад програмного оформлення графіку

# Рекомендації по організації графічних результатів обчислювального експерименту [28, 29]

Інженерні графіки повинні мати такий вигляд, 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.

## Контрольні запитання

- 1. Куди виводяться графіки Matlab?
- 2. Перерахуйте основні типи графіків Matlab.
- 3. Яка функція вважається основною для виведення графіків?
- 4. В чому полягає різниця функцій **plot** та **plotyy**?
- 5. Як вивести кілька графіків в одне вікно?
- 6. Назвіть засоби оформлення графіків.
- 7. Як вивести кілька графічних полів в одне вікно?
- 8. Яку інформацію рекомендується розташовувати в графічному вікні?
- 9. Назвіть складові графічного вікна.
- 10. Назвіть можливості редагування графічного вікна.

# 5.5. Тривимірні графіки

Для малювання просторових графіків тривимірних поверхонь та просторових кривих пакет має вбудовані засоби тривимірної графіки. Базові типи тривимірних зображень пакета наведено в таблиці 5.5.

Таблиця 5.5. Базові види просторових зображень [1]

Вигляд	Тип	Вигляд	Тип			
	Тривимірні поверхні		Каркасні поверхні			
	Вид поверхні у вигляді з ізоліній згори	Mo.	Поверхні з сіткою по ізолініях			
	Просторових кривих		Поверхні з сіткою по			
			рядках чи стовпцях			

Положення точки в просторі визначається трьома координатами. Для декартової системи координатам застосовуються координати **x**, **y**, **z**, для сферичної – радіус **r** та два кути **θ**, **φ**.

Просторові криві в пакеті визначається трьома векторами координат точок кривої. Тривимірні поверхні описуються функцією двох змінних та потребують визначення двовимірної сітки, у вузлах якої розраховується значення функції. Тримірні графіки зображуються в графічному вікні в аксонометричній проекції. Сприйняття поверхні визначається взаємною орієнтацію поверхні та точки зору спостерігача. Точка зору спостерігача визначається кутами азимута Az та піднесення Ei (рис. 5.20). Кут азимута рахується від осі ОУ в площині ХОУ проти годинникової стрілки. Кут піднесення рахується від проекції лінії, яка поєднує точку зору з початком системи координат в горизонтальній площині ХОУ та самою лінією.



Рис. 5.20. Кути точки зору [2]

За замовчанням азимут дорівнює Az = -37,5°, кут піднесення El=30°. Програмно визначити кути точки зору можна вбудованим засобом <[ Az El]>=view <([point])>.

Виклик як функції без аргументів визначає поточні кути в елементи результуючого вектора. Виклик як команди задає нові значення кутів явно через вектор [Az El] або через вектор декартових координат точки зору [x y z].

Для опису сітки, на якій зображується тривимірна поверхня, в пакеті використовуються дві матриці **X**, **Y**. Кількість рядків цих матриць дорівнює кількості точок на відрізку діапазону значень аргументу  $[y_{min} y_{max}]$ , кількість стовпців матриць дорівнює кількості точок на відрізку діапазону аргументу  $[x_{min} x_{max}]$ . Матриця **X** складається з однакових рядків, що містять координати вузлів на відрізку  $[x_{min} x_{max}]$ , матриця **Y** складається з однакових стовпців з координат вузлів відрізку  $[y_{min} y_{max}]$ .

Матриці **х**, **х** можуть бути створені за вищенаведеними правилами користувачем.

Для автоматичного розрахунку матриць X, Y призначена функція [X, Y]=meshgrid(x, y). Для аргументів векторів x довжиною M та у довжиною N результатами будуть матриця X розміром N\*M з рядками у вигляді копій вектора x та матриця Y розміром N\*M зі стовпцями у вигляді копій вектора y. Тривимірні графіки розфарбовуються кольорами, які визначаються поточною палітрою. В пакеті вбудовані палітри з плавною зміною кольорів та з різкою зміною кольорів.

Палітри з плавною зміною кольорів [32]: 'hsv' – червоний-жовтийзелений-синій-червоний (за замовчанням), 'hot' – чорний-червонийжовтий-білий, 'gray' – відтінки сірого, 'bone' – відтінки синьосірого, 'copper' – відтінки жовтого, '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.

Палітри з різкою зміною кольорів: 'flag' — червоний-білий-синійчорний, 'lines' — відповідно списку кольорів ліній в функції plot, 'vga' — кольорова схема *Windows* з 16 кольорів, 'white' — білий.

Поточна палітра встановлюється функцією [cmap]=colormap(mode). Аргумент mode є символьним рядком, який визначає передумовлені палітри кольорів. Опціональний результат стар зберігає значення поточних кольорів в матрицю.

Поруч з тривимірним графіком можна вивести стовпець, розфарбований поточною палітрою. Виводить стовпець функція [H]=colobar(<loc>). Опціональний аргумент loc є символьним рядком, який визначає положення стовпця 'North', 'South', 'East', 'West', 'NorthOutside', 'SouthOutside', 'EastOutside', 'WestOutside' або видимість стовпця палітри: 'off', 'hide'.

#### Графіки просторових кривих

Просторові криві на осі з посиланням р малює функція

#### <H>=plot3(<p,>arg,<linespec>).

Аргумент **arg** функції може задаватися параметрично трьома векторами **x**, **y**, **z** або матрицею **M**. Елементуми векторів **x**, **y**, **z** є координати точок кривої, які містять результати розрахунку параметричного опису кривої x=f(t), y=g(t), z=s(t). Матриця **M** в трьох стовпцях повинна містити значення з векторів **x**, **y**, **z**. Опціональний аргумент **linespec** є символьним рядком з визначенням лінії кривої аналогічно функції **plot**. Функцію **plot3** можна використовувати і для малювання функції двох змінних у вигляді сітчастої поверхні. При цьому в якості аргументів мають бути застосовані три матриці: сітки **х**, **х** та значень функції **z**. Поверхня малюється лініями паралельними площині YOZ. Кольори ліній змінюються послідовно відповідно списку кольорів функції **plot**.

Наприклад, запис

```
>>z=0:pi/5:2*pi;x=sin(z);y=cos(z); plot3(x,y,z,'--ro')
```

намалює графік спіралі пунктирною червоною лінією з маркерами у вигляді кіл, який зображений на рис. 5.21 а.

Запис

```
>>x=-2:0.2:2;y=x;[X Y]=meshgrid(x,y);
>>z=exp(-(X.^2.+Y.^2)); plot3(X,Y,z)
```

намалює графік кривих, який зображений на рис. 5.21 а.



Рис. 5.21. Графік : а – векторів; б – матриць

#### Графіки просторових поверхонь

Тривимірні поверхні в пакеті малюються функціями **mesh**, **meshc**, **meshz** та **surf**, **surfc**, **surfl**. Функції **mesh** малюють сітчасте зображення поверхні, функції **surf** малюють поверхню у вигляді оболонки.

Синтаксис всіх вищенаведених функцій є подібним:

## <H>=mesh(<p,X,Y,> Z,<C, 'Property','Value'>).

Обов'язковим аргументом є матриця значень функції у вузлах сітки  $\mathbf{Z}$ . Виклик тільки з аргументом  $\mathbf{Z}$  малює графік з вказанням по осях індексів матриці. Опціональні аргументи матриці  $\mathbf{X}, \mathbf{Y}$  виводять по осях **ОХ**, **ОУ** значення аргументів  $\mathbf{x}, \mathbf{y}$ . Опціональний параметр **С** є матрицею, значення якої визначають кольори поверхні. Розміри всіх матриць повинні бути однаковими. За замовчанням C=Z та графік фарбується за поточною палітрою. Змінити колір можна зміною палітри або парами «властивість-значення». Властивість FaceColor значеннями 'flat' (суцільний, за замовчанням), 'interp' (з градієнтом), 'none' (без фарбування), назвами кольорів аналогічно функції plot 'r', 'g', 'b' тощо або трикомпонентним вектором-рядком RGB в діапазоні значень від 0 до 1

[0.2 1 0.5] визначає колір граней.

Властивість **EdgeColor** аналогічними значеннями визначає колір ребер.

Змінити вигляд видимих ребер на тривимірних графіках можна командою shading <mode>. Аргумент mode є символьним рядком. Значення 'faceted' виводить ребра та рівномірно фарбує грані, воно є значенням за замовчанням. Значення 'interp' прибирає ребра та проводить плавне фарбування. Значення 'flat' прибирає ребра та рівномірно фарбує грані. Команда застосовується для налаштування попередньо намальованого графіка.

Функція **surf** малює за про мовчанням поверхню-оболонку з чорними ребрами та гранями, розфарбованими відповідно значенню функції.

Наприклад, запис

## >> [X,Y] = meshgrid(-3:.125:3); Z = peaks(X,Y);

```
surf(X,Y,Z,'Facecolor','r')
```

малює поверхню червоного кольору (рис. 5.22 а).

Функція **surfl** малює поверхню-оболонку з ребрами та гранями по двох координатах та додає джерело освітлення для фотореалістичного ефекту. Аргументи **C**, **'Property'**, **'Value'** в функції не застосовуються.

Додаткові параметри зображення визначаються опціональними аргументами:

- службове слово 'light' враховує параметри поточного джерела світла;
- службове слово 'cdata' враховує відбиття від поверхні;
- вектор [Az El] визначає напрям до джерела світла;

 вектор визначає відносну нерівномірність розподілу світла, коефіцієнт дифузного розсіяння, коефіцієнт дзеркального відбиття, коефіцієнт яскравості [ka kd ks shine] із значеннями за про мовчанням [.55,.6,.4,10].

Функція **surfc** малює поверхню-оболонку з ребрами та гранями та малює контурний графік під зображенням поверхні.

Функція **mesh** малює сітчасту поверхню з ребрами по двох координатах. За замовчанням ребра розфарбовуються відповідно значенню функції.



Рис. 5.22. Тримірні графіки: а – поверхні; б – сітки; в – дротяної поверхні

Функція **meshz** малює сітчасту поверхню з ребрами по двох координатах та заштриховує поверхню в площині ZOY для першого значення аргументу та останнього (рис. 5.22 б). Аргументи **C**, **'Property'**, **'Value'** в функції не застосовуються.

Функція **meshc** малює сітчасту поверхню з ребрами по двох координатах та малює карту ізоліній під зображенням поверхні. Аргументи **C**, '**Property'**, '**Value**' в функції не застосовуються.

Слід відзначити, що функції **mesh** не малюють повноцінну сітку. Відображається поверхня, бо передні грані є непрозорими. Зображення у вигляді справжньої дротяної моделі створюється без фарбування граней та кольорами ребер згідно значенням функції (рис. 5. 22 в).

Зробити з сітчастих поверхонь **mesh** дротяну каркасну модель можна командою **hidden off**. Команда зображує ребра із зворотного боку зображення, тобто з непрозорої оболонки робить каркасну. Виклик **hidden on** прибирає ребра із зворотного боку зображення, тобто відображає непрозору оболонку (є режимом за замовчанням).

## Контурні графіки

Контурні графіки надають наочну інформацію про розподіл значень функції для кількісного аналізу графіків поверхонь. Вони являють собою вид згори на поверхню. На контурних графіках зображуються ізолінії – лінії, в яких функція має однакове значення.

В пакеті контурні графіки малюються вбудованими функціями contour, contourf, contour3.

Φункція [c h]=contour(<p, X,Y,> Z,<N>,<linespec>) створює вид згори поверхні **Z** у вузлах сітки **X**, **Y**. Обов'язковим аргументом є матриця значень функції **z**. Виклик тільки з аргументом **z** малює графік з вказанням по осях індексів матриці. Опціональні аргументи матриці х, у виводять по осях ОХ, ОУ значення аргументів х, у. Опціональний аргумент **N** ізолінії на зображенні. Визначення натуральним числом задає кількість ізоліній. Визначення у явно вигляді вектора визначає значення, для яких намальовані ізолінії. Кількість ізоліній дорівнювати будуть буде length (N). Якщо визначити вектор з двома однаковими елементуми v **v**], то буде намальовано одну ізолінію для значення **v**. За відсутності аргументу кількість ізоліній визначається автоматично. Опціональний параметр linesspec визначає тип ізоліній аналогічно функції plot.

Опціональні результати **с**, **h** містить дані для виведення на графік чисельних значень ізоліній. Виводить чисельні значення ізоліній функція **clabel(c,h)**.

Наприклад, запис

>> [X,Y] = meshgrid(-3:.125:3); Z = peaks(X,Y); >> [f b]=contour(X,Y,Z,3,'-.b'); clabel(f,b)

малює контурний графік з трьома штрих-пунктирними блакитними ізолініями та нанесеними числовими значеннями на них (рис. 5.23 а).

Функція **contourf** малює контурний графік та зафарбовує зони між ізолініями відповідним кольором. Аргументи функції є аналогічними аргументам функції **contour**.

Наприклад, запис

>> [X,Y] = meshgrid(-3:.125:3); Z = peaks(X,Y);

```
>> [f b]=contourf(X,Y,Z); colormap('gray')
```
малює контурний графік з фарбуванням зон між ізолініями в градаціях сірого (рис. 5.23 б).

Функція contour3 малює аксонометричне зображення ізоліній. Значення **Z** визначає висоту, на якій зображується відповідна ізолінія (рис. 5.23 в). Аргументи функції є аналогічними аргументам функції contour.



Рис. 5.23. Контурні графіки: a – contour; б – contourf; в – contour 3

Для оформлення тривимірних графіків програмно можна додати назву (функція title text), підписи на осях (функції xlabel, ylabel, zlabel).

#### Контрольні запитання

- 1. Назвіть види тривимірних графіків Matlab.
- 2. Як створюється сітка для побудови тривимірних графіків?
- 3. Якими способами визначаються вектори для побудови графіка кривих?
- 4. Що є аргументами функцій побудови графіків поверхонь?
- 5. Для чого призначені контурні графіки?
- 6. Як задається палітра фарбування графіків поверхонь?
- 7. В якій проекції зображуються графіки поверхонь?
- 8. Як змінити точку зору?

#### 5.6. Символьні обчислення

Ядро пакета на містить засобів для проведення символьних перетворень. Для здійснення символьних обчислень потрібна наявність додатку *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 пакетом та символьним ядром. В попередніх версіях пакета використовувалося ядро однієї 3 найпотужніших СКА Maple. В останніх версіях пакета використовуються можливості СКА *МиРАD*, яку розробники пакета придбали та вбудували в Matlab. СКА MuPAD інсталюється разом з розширенням Symbolic Math Toolbox та може використовуватися автономно. Запустити *МиРАD* можна з командного вікна командою **тираd**.

#### 5.6.1. Засоби СКМ Matlab

#### Символьні змінні та функції

Символьні змінні є об'єктами класу sym object.

Для того, щоб пакет відрізняв дані чисельних розрахунків та аналітичних, перед тим, як проводити символьні перетворення змінні та вирази слід описати як символьні.

Рекомендованою для визначення кількох змінних одночасно використовувати команду syms x1 x2 ...<, type>. Обов'язковими аргументами команди є імена змінних, які вводяться без апострофів. Опціональний аргумент type визначає службовим словом тип змінних як дійсний real, цілий integer, додатній дійсний positive.

Наприклад, запис

#### >> syms b c; >> syms d e integer;

визначає змінні **b c** символьними, змінні **d e** символьними, які повинні оброблятися як цілі.

В якості імен можна використовувати імена функцій з аргументами в дужках. Функція визначить відповідну функцію як абстрактну символьну. Наприклад, syms s(t) f(x,y).

Автоматично визначити змінні в виразі можна функцією **symvar(x)**. Аргумент функції є рядком символів. Результатом функції є масив комірок, елементуми якого є імена символьних змінних у вигляді масивів символів. Функція є корисною для розробки застосунків, в яких вирази не є статичною частиною програми, а імпортуються ззовні або вводяться користувачем. Наприклад, запис C = symvar('cos(pi\*x - beta1)') створить масив комірок розміром 2x1C = 2x1 cell array {'beta1'} {'x'}.

Для опису однієї змінної або виразу символьним можна також використовувати функцію **C=sym(x<,type>)**. Аргументи функції є рядками символів. Обов'язковим аргументом є ім'я змінної.

Наприклад, запис

>> x=sym('x'); y=sym('y','real');

```
>> d=sym('(sin(x)+y)^2*(cos(x)+y)^2');
```

описує символьну дійсну змінну **x**, дійсну символьну змінну **y**, визначає змінну **d** як символьний вираз.

Рекомендованим засобом опису виразу як символьного є функція str2sym.

Вирази описувати символьними не обов'язково, Вирази, в яких використані символьні змінні, автоматично стають символьними.

Символьні вирази використовуються як аргументи функцій символьних перетворень та символьних методів розв'язання задач.

Додаткове визначення типу символьної змінної може знадобитися для уточнення вигляду отриманого результату перетворення.

Наприклад, для змінних без типу комплексне пов'язання не виконується:

>>syms x y; b=conj(x+i\*y) a=conj(x)-conj(y)\*1i

для змінних, визначених як дійсні комплексне пов'язання набуває очікуваного вигляду:

>>syms x y real; a=conj(x+i\*y)a=x-y\*1i

#### Обчислення з символьними змінними

Для отримання символьного значення виразу достатньо ввести в командний рядок його ім'я або сам вираз через символьні змінні.

```
>> d
d=(sin(x)+y)^2*(cos(x)+ y)^2
>>symsf(x,y); f=inline('(sin(x)+y)^2+(cos(x)+y)^2')
```

f=Inline function:  $f(x, y) = (sin(x)+y)^2 + (cos(x)+y)^2$ 

>> f(x,y)

#### ans = $(cos(x)+y)^{2}+(sin(x)+y)^{2}$

Наближення вигляду результатів символьних операцій до звичних позначень степеню, модуля тощо виконує команда

```
pretty(ім'я символьного виразу).
```

Наприклад,

```
>z=(abs(x+y))^(1/2)
> pretty(d)
(sin(x)+y)<sup>2</sup> (cos(x)+y)<sup>2</sup>
> pretty(z)
|x + y|<sup>1/2</sup>
```

Функція **subs (p, <old>, <new>)** проводить заміну в символьному виразі **p** змінної **<old>** на змінну **<new>**. Вона призначена для спрощення громіздких виразів та перевірки результатів перетворень.

Наприклад, заміна змінної **х** на змінну **х**<sup>2</sup> в виразі **d** може мати наступний вигляд:

#### >> subs(d,x,x^2) ans=(y+cos(x^2))^2\*(y+sin(x^2))^2

Але її можна використовувати і для розрахунку символьних виразів при визначених значеннях обраної змінної.

Наприклад, розрахунок символьного виразу d для визначеного чисельного значення змінної **x=1 y=2** функцією **subs** може проводитися послідовно за два кроки:

>> c=subs(d,x,1) c=(y + cos(1))^2\*(y + sin(1))^2 >> subs(c,y,2) ans=(cos(1)+2)^2\*(sin(1)+2)^2

або за один з використанням комірок:

>>subs (d, { 'x', 'y' }, {1,2}) ans=  $(cos(1)+2)^{2*}(sin(1)+2)^{2}$ 

До аналогічного результату веде перевизначення змінних:

>> x=sym(1); y=sym(2);

>>f(x,y) ans=(cos(1)+2)^2+(sin(1)+2)^2

>>  $(y+\cos(x))^{2}(y+\sin(x))^{2} ans=(\cos(1)+2)^{2}(\sin(1)+2)^{2}$ 

Для розрізняння способу розрахунку кількісні значення символьного та чисельного розрахунків відображаються на екрані по різному. Символьні результати вирівнюються по лівому краю екрана, чисельні значення виводяться з відступом [33].

Наприклад, обчислення виразу  $\frac{a}{b} + \sqrt{c}$  для **a=2**, **b=3**, **c=2** 

символьно:

Числове значення символьних виразів з визначеною точністю розраховує функція **vpa (S<, D>)**. Обов'язковим аргументом є символьний вираз/змінна **S**. Опціональний аргумент **D** визначає потрібну кількість цифр у відповіді. За замовчанням виводиться 32 цифри.

Наприклад, для введеної вище символьної змінної z=2^ (1/2) +2/3 застосування функції лає наступний результат:

#### >> vpa(z) ans = 2.0808802290397617154683553908764

Параметр **D** функції **vpa** діє тільки для поточного виклику функції. Для визначення кількості цифр в результаті функції **vpa** на весь сеанс роботи пакету слід використовувати командою **digits(D)**. Виклик **digits** без аргументу повертає поточне значення **D**.

Символьні вирази в пакеті обраховуються максимально точно, на відміну від чисельної математики.

Матричні операції для матриць, елементуми яких є символьні змінні, є символьними.

Наприклад,

>> syms all al2 al3 a21 bl1 bl2 b21 b22; >> A=[all al2; a21 a22]; »B=[bl1 bl2; b21 b22]; >> C=A\*B C = [ all\*bll + al2\*b2l, all\*bl2 + al2\*b22] [ a21\*bl1 + a22\*b2l, a21\*bl2 + a22\*b22] >> F=A.\*B F= [ all\*bl1, al2\*bl2] [ a21\*b2l, a22\*b22] >> det(A) ans = al1\*a22 - al2\*a21 >> charpoly(A) ans=[1, -all -a22, al1\*a22-al2\*a21] >> >> inv(A) ans =[ a22/(al1\*a22-al2\*a21), -al2/( al1\*a22-al2\*a21)] [-a21/( al1\*a22-al2\*a21), al1/( al1\*a22-al2\*a21)] Иеретворення символьних виразів [32]

Пакет має вбудовані функції для проведення спрощення виразу, розкриття дужок в виразі, розкладання залежності на множники, переведення добутку в доданок, обчисленням сум та добутків рядів, границь тощо.

Базові функції символьних перетворень наведені в таблиці 5.6.

Таблиця 5.6. Базові символьні перетворення

Перетворення	Функція
Тотожні перетворення	
степеневий поліном	collect
розкриття дужок	expand
розкладання на множнинки	factor
спрощення	simplify
розкладання в ряд	taylor
Алгебраїчні операції	
обчислення суми ряду	symsum
обчислення добутку ряду	symprod
обчислення границі	limit

Функція symsum (z, <k>, <kmin>, <kmax>) розраховує символьне значення суми ряду. Доданок ряду визначається символьним виразом z, індекс суми визначається символьною змінною k. Границі суми визначаються аргументами kmin, kmax. Нескінченості визначаються вбудованими змінними Inf, -Inf.

Наприклад, для ряду  $\sum_{n=1}^{\infty} \frac{1}{2^n n!}$  застосування функції може мати наступний вигляд:

» syms κ
» symsum(1/2<sup>n</sup>/factorial(n), n, 0, Inf)
ans=exp(1/2)

Функція symprod(z, <k>, <kmin>, <kmax>) розраховує символьне значення добутку. Множник визначається символьним виразом z, індекс добутку визначається символьною змінною k. Границі добутку визначаються аргументами kmin, kmax.

Наприклад, для ряду  $\prod_{n=1}^{\infty} \frac{1}{n^3+1}$  застосування функції може мати наступний вигляд:

» symprod(1/(n^3+1), n, 1, Inf)
ans=0

Функція limit(z, <a>, , <mode>) шукає границю символьного виразу z. За замовчанням вважається, що аргументом виразу є змінна x. Опціональний аргумент a явно задає ім'я змінної. За замовчанням границя шукається в точці x=0, опціональний параметр p явно задає чисельне значення точки границі. Опціональний аргумент mode задає пошук границі зліва 'left' або справа 'right'.

Наприклад, для функції *sinc* в точці 0 застосування функції може мати наступний вигляд:

» syms x; limit( $\sin(x)/x$ ) ans = 1

Наприклад, знаходження границі  $\lim_{x\to 1} \frac{1}{1-x}$  застосування функції може мати наступний вигляд:

```
» syms x; limit(1/(1-x)) ans = NaN
» syms x; limit(1/(1-x),x,1,'left') ans = inf
» syms x; limit(1/(1-x),x,1,'right') ans = -inf
```

Функція simplify(p, < Properties, Value>) спрощує символьні вирази p. Вирази можуть бути не тільки поліномами, але і містити тригонометричні, експоненціальні, спеціальні функції тощо. Опціональний аргумент у вигляді пари «властивість.значення» визначає додаткові умови перетворення. Можливими властивостями є 'Criterion', 'IgnoreAnalyticConstraints', 'Seconds', Steps'.

Наприклад,

```
>> syms x;
>>simplify((x^2-3*x-4)/(x-4)+2*x-5)
ans = 3*x - 4
>>pretty(simplify(exp(c*log(sqrt(a+b)))))
ans=(a+b)<sup>(c/2)</sup>
```

Дія не є універсальною. Наприклад для виразу  $\frac{3-x^2}{\sqrt{3}-x}$ , який в *MathCAD* спрощується, *Matlab* не надає практичного результату:

```
>>simplify((3.0-x^2)/(sqrt(3.0)-x),'Criterion',...
'PreferReal','IgnoreAnalyticConstraints',true)
ans = (x^2 - 3)/(x - 3^(1/2))
```

Функція **expand (p, <Properties, Value>)** розкриває дужки в символьних виразах **p**. Опціональний аргумент у вигляді пари «властивість.значення» визначає додаткові умови перетворення. Можливими властивостями є 'IgnoreAnalyticConstraints', 'ArithmeticOnly'.

Наприклад,

```
>> expand(exp(a+b)) ans=exp(a)*exp(b)
>>pretty(expand((x-1)*(x+1)*(x^2+1)))
ans=x<sup>4</sup> - 1
```

Функція **factor(p)** визначає множники символьного полінома **p**. Результатом є вектор-рядок, елементуми якого є множники розкладання виразу.

Наприклад,

»factor(factor( $a*x^4-a*x^3-2*a*b*x^3+2*a*b*x^4$ ) ans = [ a, x, x, x, x - 1, 2\*b + 1]) Якщо вираз є натуральним числом, то функція проводить його розкладання на добуток простих чисел. Таким чином можна визначити, чи є ціле число простим. Просте число матиме два множники: одиницю та саме число. Непросте числі матиме більше множників. Наприклад. Число 113 є простим, розкладання показує »factor(sym('113')) ans= 113 множники 1, 113. Число 2022 не є простим, бо розкладання показує множники 1, 2, 3, 337

#### »factor(sym('2022')) ans= [2 3 337].

Функція не проводить розкладання, якщо функція має ірраціональні нулі. Наприклад,

## >> factor( $x^2-9.0*a^2$ ) ans = [-1, 3\*a - x, 3\*a + x] >> factor( $x^2-19.0*a^2$ ) ans = [-1, 19\*a^2 - x^2].

Функція collect(p, <a>) проводить перетворення алгебраїчного виразу **p** на степеневий поліном, доданки якого розміщено в порядку зниження степеню аргументу. За замовчанням змінною вважається змінна **x**. Опціональний аргумент **a**  $\epsilon$  іменем змінної, вбудованої чи функції користувача або виразом символьного типу та визначає групування по степенях вказаної змінної.

Наприклад,

>> pretty(collect((2-x)^2\*(x-5)+3))

 $X^3 - 9 x^2 + 24 x - 17$ 

>> pretty(collect((2-x)^2\*(a-5)+3,a))

(x - 2)a + 3 - 5 (x - 2).

Функція taylor (p, <x>, <a>, <Properties, Value>) розкладає символьний вираз p в степеневий ряд Тейлора. За замовчанням розкладання проводиться до п'ятого степеню в околі точки 0, тобто проводиться розкладання в ряд Маклорена. Опціональний аргумент x явно визначає змінну, по якій проводиться розкладання. Опціональний аргумент a є дійсним числом. Він визначає точку, навколо якої проводитиметься розкладання в ряд Тейлора. Опціональний аргумент y вигляді пари «властивість.значення» визначає додаткові умови перетворення. Можливими властивостями є 'ExpansionPoint', 'Order', 'OrderMode'. Наприклад, розкладання функції *sinc* за замовчанням, тобто в околі точки х=0 до шостого степеню може мати наступний вигляд:

```
>> pretty(taylor(sin(x)/x))
```

Розкладання функції *sinc* в околі точки х=2 з трьома доданками ряду може мати наступний вигляд:

```
>> taylor(sin(x)/x,x,2,'Order',3)
ans =
sin(2)/2+(x-2)*(cos(2)/2-sin(2)/4)-(x-2)^2*(cos(2)/4+...
sin(2)/8)
a60
>> taylor(sin(x)/x,x,'ExpansionPoint',2,'Order',3)
ans =
sin(2)/2+(x-2)*(cos(2)/2-sin(2)/4)-(x-2)^2*(cos(2)/4+...
sin(2)/8)
```

Розширення Symbolic Math Toolbox містить функцію taylortool з віконним графічним інтерфейсом для наочної демонстрації дії функції taylor. У вікні (рис. 5.24) можна по графіку оцінити поведінку вихідної та наближеної функцій. Вираз функції та параметри розкладання вводяться у відповідних полях вікна в діалоговому режимі.



Рис. 5.24. Функція taylortool

## 5.6.2. Символьні обчислення в СКА MuPAD

Розділ опису символьних обчислень в СКА *MuPAD* базується на матеріалі навчального посібника авторів [1] та документації розробника [34].

СКА *MuPAD* інсталюється разом з розширенням *Symbolic Math Toolbox Matlab* та може використовуватися автономно. Запустити *MuPAD* можна з командного вікна командою **mupad**.

Так як в *Matlab* символьні перетворення здійснюються ядром СКА MuPAD, відповідно, можливості СКА MuPAD  $\epsilon$  не меншими за ті, що реалізовані засобами розширення *Symbolic Math Toolbox*.

Символьні дії в можна проводити введенням команд в робоче поле пакета або за допомогою кнопкових панелей. Використання кнопкових панелей суттєво підвищує зручність проведення дій.

## Оболонка пакета

Вікно *MuPAD* відкривається з назвою Notebook (рис. 5.25). Вигляд вікна є стандартним для *Windows* застосунків. Оболонка пакета містить розташоване згори вікно застосунку *меню, кнопкові панелі* **Standard Bar**, **Format Bar**, **Command Bar**, **Find and Replace Bar**. В центрі оболонки розташовується *робоче командне вікно*. Внизу вікна розміщений *статусний рядок*. В *статусному рядку* відображається тип активної області робочого вікна та параметри введення.



Рис. 5.25. Оболонка МиРАД

Інформація в робочому вікні розміщується в текстовій **Text Paragraph** або командній області **Calculation**. Область займає один або кілька рядків. Кількість та порядок областей не обмежується. В текстову область вводяться текстові пояснення, коментарі тощо. Візуальних позначок, які позначають область як текстову не передбачено. Абзаци тексту в текстовій області можуть бути відформатовані засобами кнопкової панелі **Format**.

Для введення тексту достатньо встановити курсор за межами командної області та почати вводити текст.

Створити нову текстову область можна кнопкою **Isert Text** панелі Standard або пунктом Insert-Text Paragraph меню.

Командна область є основною для введення команд і відображення результатів розрахунку. В ній користувачем вводяться команди з клавіатури або кнопкової панелі **Command** та відображаються обраховані результати. Візуально командна область позначається сірим символом квадратної дужки на початку рядка. Командна область розділена на два регіони. Згори знаходиться регіон введення команд, під ним знаходиться регіон виведення результатів. За замовчанням введені дані мають червоний колір, результати – синій. Формат шрифтів командної області може бути налаштований пунктом **Format-Characters** меню.

Створити нову командну область можна кнопкою **Test** Insert Calculation панелі Standart або пунктом Insert-Calculation меню.

#### Робота в пакеті

Знаком прирівнювання для надання значень слугують символи «:=». В одному рядку можуть записуватися кілька дій. Дії розділяються в рядку символом крапки з комою «;». Всі дії, які записані в рядку виводяться в регіон виведення. Для того, щоб не виводити дію в регіон виведення, слід в кінці опису дії поставити символ двокрапки «:».

Натискання клавіші в кінці рядка виводить результат дій рядка в регіон виведення.

Зміна даних в одній командній області враховується автоматично в інших командних областях. Обрахувати командні області можна кнопкою

**Evaluate** панелі або пунктами Notebook-Evaluate From Beginning/Evaluate To End/ Evaluate All меню.

Введені змінні та вирази витираються з пам'яті командою delete. В якості аргументів через кому перераховуються визначені імена.

Для введення матриць призначена функція **matrix**. Рядки матриці забираються в квадратні дужки. Між собою рядки розділяються комами.

Наприклад,

Отримати довідку про використання функцій пакета можна функціями info(im'я функції) та ?im'я функції. Функція info виводить коротке пояснення в регіон виведення (рис. 5.26 а). Функція ?... відкриває відповідну сторінку *html help* пакета. Утримання на імені функції в регіоні введення виводить спливаючу підказку (рис. 5.26 б).

```
info(sqrt)<br/>sqrt -- the square rootsqrt(56)<br/>2 sqrt - the square root function<br/>sqrt(z)a6
```

Рис. 5.26. Підказки: а – функції **іпfo**; б – спливаюча

За замовчання пакет виводить результат в компактному вигляді. За потреби отримати чисельне значення слід використати функцію **float**. Кількість знаків чисельного значення визначається системною змінною **DIGITS**. За замовчанням **DIGITS**=10.

Наприклад,

```
sqrt(56); float(sqrt(56)); DIGITS:=3: float(sqrt(56))
2 \sqrt{14
7.483314774
7.48
```

За замовчанням ознакою уявної одиниці в регіоні введення є символ прописної літери і – «І».

Назва більшості функцій символьних перетворень в пакеті не відрізняється від синтаксису відповідних команд в СКА *Matlab*. Для більшості функцій синтаксис аргументів теж є аналогічним. Найзручніше проводити символьні перетворення введенням назв функцій з панелі Command. На панелі дії General Math згруповані в логічні групи (рис. 5.27) Expand, Factor, Normalize, Evaluate, Simplify, Rewrite, Combine, Calculus, Matrices, Polynomials. Кнопки найбільш часто вживаних дій виведені безпосередньо на панель.

Define Standard Matrix Domain Characteristic Polynomial Concatenate Decomposition / Determinant Dimension Eigenvalues / Fiagenvectors / Fraction-Free Gaussian Elimination Gauss-Jordan Elimination Hermite Normal Form	Integrate (Definite) Integrate (Definite) Integrate by Parts Integrate Numerically Change Variable Partial Fraction Differentiate n times Differentiate n times Differential Operator Sum	Expand Factor Normalize Evaluate Simplify Combine Rewrite Sclve	Numerically Boolean General Logical	General Arc Tangent Exponential Logarithm Power Sine / Cosine Sine / Cosine Hyp Square Root Differential
Hilbert Matrix Inverse Nullspace Basis Random Matrix Rank Tests	Limit Limit (Left) Limit (Right)	Power Series Matrices + Polynomials +	<ul> <li>Divide</li> <li>Numeric</li> <li>Real Roots</li> <li>Integer Roots</li> </ul>	Factorial Gamma Heaviside Logarithm Sign Sine / Cosine
Trace Transpose				

Рис. 5.27. Склад панелі Command Bar

Натискання кнопки на панелі **Command** призводить до появи в регіоні введення шаблону відповідної функції

#### [diff(<mark>#f</mark>, #x)

В шаблоні аргументи позначені символом діезу «#». Для проведення дії в поля аргументів повинні бути вписані фактичні дані. Перемикання між аргументами можна проводити мишею або клавішею **Таb**.

Функція collect проводить групування символьних поліномів по степенях.

```
a := (x+y)^{4} + (x-1)^{3} - (x-y)^{2} - y^{*}x + x - 3: \text{collect}(a, x);

collect (a, y)

x^{4} + (4y+1)x^{3} + (6y^{2}-4)x^{2} + (4y^{3}+y+4)x + y^{4} - y^{2} - 4

y^{4} + (4x)y^{3} + (6x^{2}-1)y^{2} + (4x^{3}+x)y + x + (x-1)^{3} - x^{2} + x^{4} - 3
```

Матричні перетворення не потребують додаткових описів.

Наприклад, для квадратних матриць А, В добуток матриць, визначник матриці А, обернена матриця А, характеристичний поліном матриці В будуть мати наступний вигляд:

```
A := matrix([[a, b], [c, d]]):

B := matrix([[e, f], [g, k]]): A*B;det(A);1/A;

linalg::charpoly(A, x)

\begin{pmatrix} a e + b g a f + b k \\ c e + d g c f + d k \end{pmatrix}
a d - b c
\begin{pmatrix} \frac{d}{ad-bc} & -\frac{b}{ad-bc} \\ -\frac{c}{ad-bc} & \frac{a}{ad-bc} \end{pmatrix}
x^{2} + (-a - d) x + a d - b c
```

Функція **ехрапd** розкриває дужки в символьних виразах.

 $a := (x+y)^{4} + (x-1)^{3} - (x-y)^{2} - y^{*}x + x - 3 : expand (a)$  $x^{4} + 4x^{3}y + x^{3} + 6x^{2}y^{2} - 4x^{2} + 4xy^{3} + xy + 4x + y^{4} - y^{2} - 4$ 

Функція factor визначає множники символьного полінома.

```
\frac{(2 x + 1) (2 x - 1) (x + 6) (x + 4) (x + 3)}{4}
```

Функції simplify та Simplify спрощують символьні вирази. Функція simplify працює швидше. Функція Simplify дозволяє спрощувати більш складні вирази та має опціональні аргументи. Аргумент визначає Steps кількість кроків спрощення. Аргумент All дозволяє отримати інший варіант запису результату.

```
f:= \exp((\cos(x)^{2} - \sin(x)^{2}) * \sin(2*x) * (\exp(2*x) - 2*\exp(x) + 1) / (\exp(2*x) - 1)):

simplify(f);Simplify(f);Simplify(f,Steps=250)

e^{\frac{\sin(4x)(e^{x}-1)}{2(e^{x}+1)}}
e^{-\frac{\sin(4x) - \sin(4x)e^{x}}{2e^{x}+2}}
e^{\frac{\sin(4x)(e^{x}-1)}{2(e^{x}+1)}}
```

```
 \begin{aligned} f &:= (x - 1) * (x + 1) * (x^{2} + x + 1) * (x^{2} + 1) * (x^{2} - x + 1) * (x^{4} - x^{2} + 1) : \\ & \text{simplify(f); Simplify(f, All)} \\ x^{12} &- 1 \\ & \left[ x^{12} - 1, (x^{2} + 1) (x - 1) (x + 1) \sigma_{2} \sigma_{1} \sigma_{3}, (x - 1) (x + 1) \sigma_{2} \sigma_{1} \sigma_{3} + x^{2} (x - 1) (x + 1) \sigma_{2} \sigma_{1} \sigma_{3} \right] \\ & \text{where} \\ & \sigma_{1} &= x^{4} - x^{2} + 1 \\ & \sigma_{2} &= x^{2} - x + 1 \\ & \sigma_{3} &= x^{2} + x + 1 \end{aligned}
```

Функція subs (p, old=new) проводить заміну в символьному виразі р змінної old значенням. Отримати обчислений результат перетворення можна застосуванням опціонального аргументу EvalChanges або викликом після функції обчислення eval (%).

```
f:=sin(x^2) + 1:subs(f, x = 0): subs(f, x=0);eval(%);subs(f, x=0, EvalChanges)
sin(0) +1
1
1
```

Функція series (p, x=a, n) розкладає символьний вираз р в степеневий ряд Тейлора.

```
\begin{bmatrix} \text{series}(\exp(-x), x=1, 3) \\ e^{-1} - e^{-1}(x-1) + \frac{e^{-1}(x-1)^2}{2} + O((x-1)^3) \end{bmatrix}
```

Функція **sum** розраховує символьне значення суми.

```
\begin{bmatrix} \operatorname{sum}((-1)^k/k^2, k=1..\inf(n) \end{bmatrix}-\frac{\pi^2}{12}
```

Функція **product** символьне значення добутку.

```
[ product((-1)^k/k^2, k=1..infinity)
0
```

Функція limit шукає границю символьного виразу.

## Контрольні запитання

- 1. Що потрібно для увімкнення можливості символьних перетворень?
- 2. Який тип мають символьні змінні? Як вони визначаються?
- 3. Що таке "природній" вигляд виразу?
- 4. Які символьні дії проводяться для перетворення виразів?
- 5. Для чого застосовується функція **subs**?
- 6. Які задачі інтегрального та диференціального обчислення символьно вирішує *Matlab*?

## 5.7. Експорт-імпорт даних

Пакет має вбудовані засоби для файлового обміну. Обробляються зовнішні дані з мультимедіа файлів, наукові дані, дані з текстових ASCII файлів та файлів офісних застосунків. Керування обміном може проводитися **програмно** або **інтерактивно**. Програмні засоби забезпечують як імпорт, так і експорт даних. Інтерактивні засоби дозволяють проводити тільки імпорт даних.

## 5.7.1. Програмно керований експорт-імпорт даних

## Низькорівневий файловий обмін.

Алгоритм низькорівневого обміну складається з наступних типових етапів:

- формування специфікації файлу.
- відкриття файлу.
- проведення дій з файлом.
- закриття файлу.

## Визначення специфікації файлу

Повний шлях до файлу (специфікація файлу) для діалогового режиму роботи застосунків в пакеті найзручніше проводити засобами віконного діалогу вбудованими функціями uigetfile, uiputfile

Системний віконний діалог для відкриття файлу в модальному вікні реалізує функція

[FileName,Path<,Filter>]=
uigetfile<(FilterSpec,Title,Default)>.

Опціональний аргумент FilterSpec визначає тип файлів, які будуть відображатися у вікні. Аргумент може бути масивом символів або масивом комірок. Якщо значенням аргументу є ім'я файлу, то це ім'я буде вказано за замовчанням в полі вибору файлу вікна. Аргумент може містити символи адресації папок «.», «..», «\», «/», «~» та автозаміни імен «\*». Наприклад, запис '../\*.dat' говорить про всі файли з розширенням dat в папці над поточною. Для визначення адресою пошуку папки слід в аргументі вказати ім'я папки з символом «\» або «/» в кінці.

Перелік з кількох альтернатив обрання типів файлів задається у вигляді комірок в фігурних лапках. Типи розширень файлів записуються масивами символів з розділенням символом крапки з комою. Коментар до альтернативи, який буде виведено в поле типів файлів вікна, записується через кому в кінці альтернативи як масив символів та завершується символом крапки з комою.

Наприклад, запис

{'\*.jpg;\*.jpeg';'\*.png','JPEG Family files';
'\*.bmp;\*.pcx', 'BitMap Family files';
'\*.\*', 'All Files (\*.\*)'}

визначає три рядки в полі визначення типів файлів вікні. В поле буде виведено список з трьох рядків: 'JPEG Family files', 'BitMap Family files' та 'All Files (\*.\*)'. Для пошуку будуть застосовані розширення файлів .*jpg*, .*jpeg*, .*png* для першого рядка, . *bmp*, .*pcx* – для другого, будь-які – для третього.

В разі відсутності аргументу або порожньому значенні у вигляді двох апострофів функція використовує список типів за замовчанням, тобто список *Matlab*.

Опціональний аргумент **Title** є назвою модального діалогового вікна.

Опціональний аргумент **Default** є ім'ям файлу або папки за замовчанням.

Обов'язковими результатами є **FileName**, який містить ім'я файлу та **Path**, який місить шлях до функції. Опціональний результат **Filter** є індексом типу обраного файлу зі списку фільтрів аргументу **FilterSpec**. В разі закриття вікна кнопкою **Cancel** функція повертає чисельне значення всім результатам 0.

Виклик функції без аргументів відкриває поточну папку.

Наведемо приклади відкриття файлів.

Дія зчитування з вікна з назвою «Введіть ім'я файлу даних» файл з розширенням .*dat* може виглядати наступним чином:

[FileName,Path]=uigetfile('\*.dat','Введіть ім'я... файлу даних ').

Зчитування з вікна файл *Matlab* з альтернативами вибору може виглядати наступним чином:

```
[filename, pathname]=uigetfile({'*.m;*.mat',...
'MATLAB Files'; '*.fig','Figures';...
'*.mn','MuPad Files';'*.*', 'All Files (*.*)'};
```

Системний віконний діалог для запису файлу на диск в модальному вікні реалізує функція

# [FileName,Path<,Filter>]= uiputfile<(FilterSpec,Title,Default)>

виводить модальне системне діалогове вікно для введення імені файлу для збереження. Якщо значенням аргументу є ім'я файлу, то це ім'я буде вказано за замовчанням в полі вибору файлу вікна.

Опціональний аргумент **FilterSpec** визначає тип файлів, які будуть відображатися у вікні в полі **Save as**. В разі відсутності аргументу або порожньому значенні у вигляді двох апострофів функція використовує список типів за замовчанням, тобто список *Matlab*.

Опціональні аргументи Title, Default результати FileName, Path, Filter  $\epsilon$  поведінка вікна кнопкою Cancel  $\phi \epsilon$  аналогічними функції uigetfile.

Виклик функції без аргументів відкриває поточну папку.

Наведемо приклади визначення файлу для збереження.

Визначення у вікні з назвою Зберегти специфікації файлу для збереження файлу priklad.dat може виглядати наступним чином:

## [FileName,Path]=uigetfile('priklad.dat','Зберегти дані ');

Визначення у вікні специфікації файлу робочого простору з пропонуванням типу файлу за замовчанням може виглядати наступним чином:

[file,path]=uiputfile('\*.mat',' Зберегти дані ').

Для формування повної специфікації файлу слід поєднати інформацію про ім'я (специфікації) файлу та шляху до нього функцією конкатенації масивів символів strcat(st1,...,strN) або функцією fullfile(filepart1,...,filepartN) в єдиний масив символів.

## Відкриття-закриття файлу

Відкриття файлу проводиться функцією **fopen**.

Функція має дві форми виклику. Основною формою є

## <[> fileID <,errmsg]>=

## fopen(filespec <,permission,fmt,encoding>).

Обов'язковим аргументом є масив символів **filespec**, який містить повну специфікацію файлу.

Опціональні аргументи визначають додаткові умови відкриття файлу [35]:

- permission визначає режим, в якому може бути використаний файл. Можливими режимами є зчитування 'r', запис 'w', додавання інформації в кінець існуючого файлу 'a', зчитування та запис 'r+', 'w+', зчитування та додавання інформації 'a+'. Для уточнення, що буде проводитися обробка текстових даних, може бути додавана літера 't' в кінці назви режиму. Наприклад, 'rt', 'wt+'. За замовчанням файл відкривається в режимі зчитування 'r'. Задається текстовим рядком масиву символів в апострофах.
- encoding визначає кодову сторінку текстових даних. Можливими значеннями є кодові сторінки 'UTF-8', 'ISO-8859-1', 'windows-1251', 'GBK', 'IBM866', 'KOI8-U', 'Macintosh'. За замовчанням визначається поточна кодова сторінка. Задається текстовим рядком масиву символів в апострофах.
- fmt визначає порядок обробки байтів в словах для бінарної інформації. За замовчанням застосовується 'n', 'native' поточна обробка системи. Задається текстовим рядком масиву символів в апострофах.

Результатом є файлова змінна **fileID** з системним посиланням на файл, яка стає ідентифікатором файлу в пакеті.

Опціональний результат **errmsg** містить системне повідомлення про помилку. В разі успішної операції є порожнім рядком.

Друга форма використовується в разі потреби отримати інформацію про стан файлу, який має ідентифікатор **fileID**:

## [filename,permission,fmt,encoding] = fopen(fileID).

Результатом функції є вектор-рядок з чотирма елементуми типу масив символів. Перший елемент вектора **filename** містить специфікацію файлу, другий елемент **permission** – режим, в якому відкритий файл, третій – порядок обробки байтів в словах для бінарної інформації, четвертий – назву кодової сторінки. Непотрібні елементи в векторі результату позначаються символом тильди «~».

Наведемо приклади відкриття файлів.

Відкриття файлу **priklad.dat** з параметрами за замовчанням може виглядати наступним чином:

## fileID = fopen('priklad.dat');

Файл відкрито для зчитування в бінарному режимі інформації з поточною кодовою сторінкою та поточним порядком обробки байтів.

Приклад. Ввести з клавіатури назву файлу та відкрити файл для запису з звичайною обробкою байтів в кодовій сторінці **Unicode**. В разі невдалої операції вивести системне сповіщення та повторити операцію. Вивести інформацію про відкритий файл.

Розв'язок

```
fileID = -1; errmsg = 'Помилка';
while fileID < 0
filename = input('Open file: ', 's');
[fileID,errmsg]=fopen(filename,'w','n','UTF-8');
if fileID < 0
disp(errmsg); end
end
[fname,mode,bytes,code] = fopen(FileID)
disp([fname mode bytes code])
```

Закриття файлу з внутрішнім ідентифікатором fileID проводиться функцією

#### <s> = fclose(fileID <all>).

Виклик функції з аргументом '**all**' замість ідентифікатора файлу викликає закриття всіх відкритих файлів. Опціональний результат **s** має значення 0, якщо операція пройшла успішно та -1 в інших випадках.

## Зчитування даних

Зчитування інформації проводиться функціями fgetl, fgets, textscan.

Функція line = fgetl (fileID) зчитує в змінну рядок з текстового файлу як текстовий рядок з прибиранням службових символів переведення рядка. В разі досягнення кінця файлу повертається чисельне значення -1.

Функція line = fgets (fileID <,nchar>) зчитує в змінну рядок з текстового файлу як текстовий рядок БЕЗ прибирання службових символів переведення рядка. В разі досягнення кінця файлу повертається чисельне значення -1. За потреби кількість символів в рядку, які будуть зчитуватися, можна визначити числом в опціональному аргументі nchar.

## Функція

## <[> C <, position]>=textscan(fileID, format<, N, Name, Value>)

зчитує дані з текстового файлу в масив комірок **C**. Обов'язковим аргументами є ідентифікатор файлу **fileID** та масив символів в апострофах **format**, який в синтаксисі опису формату функціям **sprint**, **fprintf** визначає формат даних рядка для зчитування.

Опціональний аргумент **N** є додатним цілим числом, яке визначає кількість разів зчитування визначеного формату. Для продовження зчитування слід повторити виклик функції з тим же ідентифікатором файлу.

Опціональні пари аргументів **Name, Value** визначають додаткові налаштування зчитування в термінах об'єктного програмування «властивість-значення» [35].

Дія
всі дані рядка повинні збиратися в єдиний масив символів. Припустимими значеннями є true або false.
символи, між якими дані вважаються коментарем та НЕ зчитуються. Наприклад, 'CommentStyle', {'/*', '*/'}, 'CommentStyle', '%'.
символ розділення даних в рядку. Без вказання символу розділення таким вважається символ пробілу.
Наприклад, 'Delimiter',{';'}, 'Delimiter','\t'.
число замість <b>Nan</b> заміни порожнього поля в рядку.
символ позначення кінця рядка замість стандартних '\n', '\r', '\r\n'.
кількість рядків з початку файлу, які не будуть зчитуватися.
значення, які будуть замінені на <b>NaN</b> . Значення записуються як масив комірок або вектор-рядок символів. Наприклад, 'TreatAsEmpty', {'NA', 'N/A', '0.99'} або 'TreatAsEmpty', ['NA', 'N/A', '-0.99'].

## Формат даних кодується записом у вигляді %<\*n>L.МТ

Поле **T** визначає тип даного значенням для цілого типу **d**, цілого типу без знака **u**, дійсного типу одинарної точності **f**, дійсного типу подвійної точності **%n**, одиночні символи **c**, слова **s**, дати **D** тощо.

Поле L визначає загальну кількість символів в даному, поле М – кількість символів після коми. Опціональне поле **\*** показує, що дане описаного коду НЕ буде зчитуватися. Вказання для нечисельних типів даних числа **n** після показує, що в даному НЕ буде зчитуватися **n** символів.

Явно вказаний в описі формату текст НЕ зчитується в результат.

Опціональний результат **position** містить номер останнього зчитаного рядка

Наведемо приклади застосування функції.

Наприклад, застосування функції до текстового файлу з ідентифікатором id в якому міститься рядок **1.0 2.0 3.0 4.0** 

#### C = textscan(id, '%f');

результатом дає масив комірок розміром 1x1. Елемент масиву є вектором-стовпцем [1;2;3;4].

Застосування функції до того ж текстового файлу

C = textscan(id, '%f %f %f %f');

результатом дає масив-рядок комірок розміром 1х4. Елементуми масиву є дійсні числа.

Наприклад, застосування функції до текстового файлу з ідентифікатором id в якому міститься стовпець 1.0 2.0 3.0 4.0

C = textscan(id, '%f');

результатом дає масив комірок розміром 1x1. Елемент масиву є вектором-стовпцем [1;2;3;4]. Тобто повторює зчитування рядка.

Застосування функції до того ж текстового файлу

#### C = textscan(id, '%f %f %f %f');

результатом дає масив-рядок комірок розміром 1х4. Перший є векторомстовпцем [1;2;3;4]. Останні три є векторами-стовпцями з елементуми NaN [NaN; NaN; NaN; NaN].

Наприклад, застосування функції до текстового файлу з ідентифікатором іd в якому міститься рядок 'Приклад не є вдалим'

C = textscan(id, '%s%\*s%s%s');

результатом дає масив комірок розміром 1x3 з елементуми 'Приклад' 'є' 'вдалим'. Наприклад, застосування функції до текстового файлу з ідентифікатором id в якому міститься рядок 'Багато2Мало'

```
C = textscan(id, 'Багато%uМало');
```

результатом дає масив комірок 1х1 з одним елементом числом 2.

Приклад. В текстовому файлі **dat.txt** містяться експериментальні дані :

Series Date Data1 Data2 Data3 Res1 Res2 Corr Compl
Ser1, 29/09/2021, 12.34, 45, 1.2e4, Inf, NaN, Yes, 5.1+3i
/\* not correct
Ser2,13/19/2021, 45.11, 16, 2e6, 25.3, 0.1, No, 1.5-2i
\*/

```
Ser3, 19/10/2021, 34.12, 521, -43.1, ,0.1, Yes, 1+3.1i
```

Перший рядок є текстом з назвами стовпців даних. В другому рядку містяться два текстові, дійсне, ціле, два дійсні, текстове та комплексне поля. Другий рядок даних помічено коментарем як хибний. Зчитати тільки вірні експериментальні дані без текстових включень, невизначені значення замінити на 0.

Спроба зчитування з форматом '%s %s %f %d %f %f %f %s %f' не показує помилки, але в результуючому масиві комірок розміром 1x9 заповненими є тільки перші два елементи. Перший елемент є рядком 'Series', другий – рядком 'Date'.

```
f=fopen('dat.txt','r');
C=textscan(f,'%s %s %f %d %f %f %f %s %f');
fclose(f);
> celldisp(C)
C{1}{1} = Series
C{2}{1} = Date
C{3} =
            []
C{4} =
            []
C{5} =
            []
C{6} =
            []
C{7} =
            []
C{8} =
            { }
C{9} =
            []
```

421

Для виправлення слід вказати кількість рядків заголовку. Для виключення коментарів потрібно вказати використані в файлі символи коментарів, явно вписати текст для прибирання в другому текстовому полі, прибрати третє текстове поле та змінити значення **NaN**:

```
f=fopen('e:\dat.txt','r');
C=textscan(f,'%sSer%s%f%d%f%f%f%f%*s%f',...
'CommentStyle', { '/*', '*/' },...
    'HeaderLines',1,'EmptyValue',0,'Delimiter',',');
fclose(f);
Результатом є масив комірок розміром 1x8
C{1}{1} = 1
C{1}{2} = 2
C{2}{1} = 29/09/2021
C{2}{2} = 29/09/2021
C{3} =
  12.3400000000000000
  34.119999999999
C{4} =
   45
   521
C{5} =
   1.0e+04 *
   1.20000000000000
  -0.0431000000000
C{6} =
   Inf
    0
C{7} =
   NaN
   0.100000000000000
C{8} =
  5.1000000000000 + 3.000000000000000
  5.1000000000000 + 3.000000000000000
```

Скидання лічильника зчитаних рядків на початок файлу проводиться функцією **frewind(fileID)**.

## Програмні засоби високого рівня файлового обміну

Кожна вбудована функція обміну високого рівня поєднує в собі дії з даними, разом з операціями відкриття та закриття файлу, що суттєво спрощує програмування файлового обміну.

## Обмін з текстовими файлами

Вбудовані функції високого рівня програмного обміну даними з текстовими файлами наведено в таблиці 5.7.

Тип файлу	Розширення	Функція імпорту	Функція експорту
Текстові ASCII файли з чисельними даними, які розділені комами	довільне, CSV, TXT	csvread	csvwrite
Текстові ASCII файли з чисельними даними	довільне, ТХТ	dlmread	dlmwrite
Текстові ASCII файли зі стовпцями чисельних та текстових даних	довільне, ТХТ	readtable	writetable

Таблиця 5.7. Вбудовані функції обміну з текстовими файлами

Зчитування з текстового файлу табличних числових даних з розділенням комою в масив **М** проводиться функцією

## M = csvread (filespec <,R1,C1,[R1 C1 R2 C2]>)

Обов'язковим аргументом функції є повна специфікація файлу **filespec** у вигляді масиву символів.

Опціональні аргументи R1, C1, R2, C2 задають цілими числами діапазон зчитування. R1(C1) визначає номер рядка (стовпця) початку зчитування, R2(C2) – номер рядка (стовпця) кінця зчитування. Нумерація комірок проводиться з нуля. Для зчитування з рядка R1 стовпця C1 до кінця файлу другий опціональний аргумент-вектор не записується. Для зчитування частини даних всередині файлу повинні задаватися обидва опціональні аргументи. В разі відсутності опціональних аргументів проводиться зчитування всього файлу.

Наведемо приклади застосування функції.

Наприклад, застосування функції до текстового файлу priklad.dat, в якому містяться наступні чисельні дані:

81.1, 52.0, 73.3, 34.2
36.0, 67.5, 6.48, 10.8
58.5, 93.6, 2.7, 3.8
M = csvread('priklad.dat')

результатом дає матрицю M розміром 3х4 з дійсними елементуми подвійної точності:

м =

81.0999999999999994	52.0000000000000000	73.299999999999999
34.20000000000003		
36.0000000000000000	67.5000000000000000	6.4800000000000000
10.800000000000000		
58.5000000000000000	93.5999999999999994	2.7000000000000000
3.800000000000000		

Зчитування з того ж файлу даних від першого рядка третього стовпця до кінця файлу

```
N = csvread('priklad.dat' ,0,2)
```

результатом дає матрицю N розміром 3x2 з дійсними елементуми подвійної точності:

N = 73.299999999999997 34.200000000000 6.480000000000 10.800000000000 2.700000000000 3.800000000000 Запис M=csvread ('priklad.dat' ,1,0,[1,0,2,2])

результатом дає матрицю M розміром 2x3 з елементуми значень файлу з другого рядка по третій рядок та з першого стовпця по третій стовпець.

м =

Записує в текстовий файл з повною специфікацією файлу filespec з розділенням комою числові дані з масиву **M** оператор csvwrite(filespec,M <,R1,C1>).

Опціональні аргументи **R1**, **C1** задають числом номер рядка (стовпця) початку запису. Нумерація проводиться з нуля. В результуючому файлі матриці на місцях до **R1**, **C1** записуються коми.

Наведемо приклади застосування функції.

Запис csvwrite('scv.dat',M) результатом дає текстовий файл scv.dat з наступним вмістом

```
52, 73.3
67.5, 6.48
93.6, 2.7
```

Запис csvwrite('scv.dat',M, 2,2) результатом дає текстовий файл scv.dat з наступним вмістом

```
,,,
,,52, 73.3
,,67.5, 6.48
,,93.6, 2.7
```

Зчитування в масив **M** з текстового файлу з розділителем даних, який можна визначати, табличних числових даних функція форматованого зчитування

```
M = dlmread (filespec, <del,R1,C1>).
```

Обов'язковим аргументом функції є повна специфікація файлу **filespec** у вигляді масиву символів. В разі відсутності опціональних аргументів проводиться зчитування всього файлу.

Опціональний **del** визначає в апострофах символ, який слугує розділювачем даних в файлі. Для знака табуляції застосовують символ '\t'. Без вказання аргументу функція автоматично визначає розділювач.

Пара опціональних аргументів **R1**, **C1** визначає індекси комірки, з якої треба починати зчитування даних. Записуватися може у вигляді вектору або символуми за правилами адресації A1 в *MS Excel* 'літера стовпця номер рядка'. В разі визначення аргументу вектором [**R1 C1 R2 C2**] зчитування

проводиться в діапазоні рядків від **R1** до **R2** та стовпців – від **C1** до **C2**. Комірка 0,0 має адресу **A1**.

Наприклад, зчитування з файлу 'priklad.dat' даних з рядка 2 по 3 стовпців з 2 по 3 може мати наступний вигляд:

```
M=dlmread('priklad.dat',',','B2:C3')a60
M=dlmread('priklad.dat',',', [1 1 2 2])
M=
67.5000000000000
6.4800000000000
93.599999999994
2.700000000000000
```

Запис в текстовий файл, який визначений рядком повної специфікації файлу **filespec** в апострофах числові дані з масиву **M** з розділювачем даних, який можна задати, оператором

```
dlmwrite(filespec,M<,-'append'><,del,R1,C1,Name,Value >)
```

Опціональні аргументи дозволяють тонко налаштувати запис [40]:

- del визначає в апострофах символ розділення. В разі відсутності аргументу del вважається, що для розділення використовується кома.
- '**append**' проводить запис в існуючий файл з додаванням даних.
- пара чисел **R1, C1** визначає стартову комірку, з якої в файлі будуть розміщені дані з матриці **М**.
- пари «властивість-значення» Name, Value є аналогами інших опціональних аргументів в термінах об'єктного програмування: 'delimiter' – del, 'roffset' – R1, 'coffset' – C1

Наприклад, запис

A=[95 94 93 92; 85 84 83 82; 75 74 73 72; 65 64 63 62]; dlmwrite('num.dat',A)

створить текстовий файл 'num.dat' з 4-х рядків з даними матриці A 4х4 з розділювачем у вигляді коми.

Запис dlmwrite('num.dat', A, '\t', 2, 1) створить текстовий файл з даними матриці A 4x4 з розділювачем у вигляді знаку табуляції. В файл

перед даними матриці **A** будуть додані 2 рядки зі знаками розділення, на початку кожного рядка буде додано один знак табуляції.

Зчитує таблично організовані дані з текстового файлу, який визначений забраною в апострофи повною специфікацією файлу filespec, в таблицю **т** функція

#### T = readtable(filespec <,Name,Value>)

Опціональні пари аргументів **Name**, **Value** визначають додаткові налаштування зчитування в термінах об'єктного програмування «властивістьзначення» [35]:

Властивість	Дія		
'FileType'	<b>'text'</b> для	текстових	файлів,
	<b>'spreadsheet'</b> .xls, .xlsb, .xlsm, .xls	для файлів sx, .xltm, .xltx, .c	MS Excel ods.
'ReadVariableNames'	true для зчиту вигляді імен змін зчитування всіх ст в них. В разі відсу змінним стовпц замовчанням 'va:	вання першог них таблиці та овпців з другог лтності функція ів таблиці <b>r1</b> '	о рядка у значенням го елементу призначає імена за
'ReadRowNames '	true для зчитув вигляді імен рядкі замовчанням) лоя з	ання першого в та значенням зчитування всіх	стовпця у , <b>false</b> (за срядків без

У випадку, коли обидві властивості 'ReadVariableNames' та 'ReadRowNames' є true, функція зберігає елемент файлової таблиці як першу назву властивості таблиці \*.Properties.DimensionNames.

виділення першого елементу з них.

Властивості 'CollectOutput', 'CommentStyle', 'Delimiter' (в разі відсутності функція намагатиметься визначити символ автоматично), 'EmptyValue', 'EndOfLine', 'HeaderLines' (для текстових файлів), 'TreatAsEmpty', 'TextType', 'Format' є аналогічними функції textscan.

Властивість **Encoding** є аналогічною функції **fopen**.

Наведемо приклади використання функції.

Наприклад, для файлу 'dat.txt', який описано вище для функції , запис T=readtable('dat.txt') не дозволить зчитати дані, бо в другому та п'ятому рядках дані не є табличними, в останньому рядку є комірка без даних.

Якщо прибрати наведені особливості в файлі та зробити стовпець 'Seriesl' першим, то функція зчитає в таблицю Т розміром 3х9 дані з текстового файлу. Змінні стовпців таблиці набули імен з першого рядка файлу.

```
T = 3×9 table
Series Date Data1 Data2 Data3 Res1 Res2 Corr Compl
```

'Serl'	29/09/21	12.34	45	12000	Inf NaN	'Yes'	5.1+3i
'Ser2'	13/10/21	45.11	16 2	000000	25.3 0.1	'No '	1.5-2.1i
'Ser3'	19/10/21	34.12	521	-43.1 (	0.1 14	'Yes'	5.1+3i

Для того, щоб дані внесені в таблицю дані містили перший рядок файлу слід відмінити зчитування з першого рядка в список назв стовпців:

```
T=readtable('e:\dat.txt','ReadVariableNames',false)
```

В такому разі функція зчитає в таблицю Т розміром 4х9 дані з текстового файлу.

```
T = 4×9 table
Var1 Var2 Var3 Var4 Var5 Var6 Var7 Var8 Var9
```

```
'Series''Date''Data1''Data2''Data3''Res1''Res2''Corr''Comp'
'Ser1''29/09/21''12.34''45''1.2e4''Inf''NaN''Yes''5.1+3i'
'Ser2''13/19/21''45.11''16''2e6''25.3''0.1''No''1.5-2i'
'Ser3''19/10/21''34.12''521''-43.1''12''0.1''Yes''1+3.1i'
```

Для того, щоб таблиця містила назви і стовпців, і рядків слід вмикнути властивості **ReadVariableNames** та **ReadRowNames**:

```
T=readtable('e:\dat.txt','Delimiter',',','ReadRowNames',...
true, 'ReadVariableNames', true)
```

## T = 3×8 table Date Data1 Data2 Data3 Res1 Res2 Corr Compl

Ser129/09/2112.344512000InfNaN'Yes'5.1+3iSer229/10/2145.1116200000025.30.1'No'1.5-2iSer319/10/2134.12521-43.1120.1'Yes'1+3.1i

Дані з таблиці **Т** пакета в текстовий файл записує оператор writetable(**T** <, filespec, Name, Value>).

В разі відсутності повної специфікації файлу пакет створює в робочому каталозі файл з іменем таблиці та розширенням *.txt*. За про мовчанням запис проводиться з розділенням комою.

Опціональні пари аргументів **Name,Value** застосовуються для форматування даних у файлі. Значення властивостей є аналогічним функції readtable. Припустимими є наступні властивості: 'FileType', 'WriteVariableNames', 'WriteRowNames', 'DateLocale', 'Encoding', 'Delimiter'.

Наведемо приклади використання функції.

Запис writetable(T) запише в файл T.txt дані з таблиці T з розділенням комою. Назви стовпців будуть записані в файл, назви рядків – ні:

Date,Data1,Data2,Data3,Res1,Res2,Corr,Compl 29/09/2021,12.34,45,12000,Inf,NaN,Yes,5.1+3i 29/10/2021,45.11,16,2000000,25.3,0.1,No,1.5+-2i 19/10/2021,34.12,521,-43.1,12,0.1,Yes,1+3.1i Рядок

```
writetable(T,'wdata.txt','Delimiter','space',...
'WriteRowNames',true,'QuoteString',true)
```

запише в файл wdata.txt дані з таблиці **T** з розділенням пробілом з назвами стовпців та назвами рядків, взяттям текстових полів в лапки:

```
Series Date Data1 Data2 Data3 Res1 Res2 Corr Compl
"Ser1" "29/09/2021" 12.34 45 12000 Inf NaN "Yes" 5.1+3i
"Ser2" "NaT" 45.11 16 2000000 25.3 0.1 "No" 1.5+-2i
"Ser3" "19/10/2021" 34.12 521 -43.1 12 0.1 "Yes" 1+3.1i
```

## Обмін з файлами MS Excel

Вбудовані функції високого рівня програмного обміну даними з *MS Excel* файлами наведено в таблиці 5.8.

Тип файлу	Розширення	Функція імпорту	Функція експорту
Текстові ASCII файли зі стовпцями чисельних та текстових даних	довільне, ТХТ	readtable	writetable
Файли <i>MS Excel</i>	xls, xlsx	xlsread	xlswrite

Таблиця 5.8. Вбудовані функції обміну з MS Excel

Функція **readtable** в режимі **spreadsheet** зчитує дані з *MS Excel* файлів. Слід взяти до уваги, що СКМ не може мати імена змінних з кириличними символами, тому назви стовпців таблиці потрібно давати латиницею.

Властивість 'Sheet' визначає аркуш *MS Excel*, з якого буде проводитися зчитування. Значення може задаватися натуральним числомномером аркуша або масивом символів з його ім'ям.

Властивість 'Range' визначає масивом символів діапазон даних для зчитування. Визначається в термінах адресації *MS Excel*. Одна адреса 'Corner1' означає, що зчитування буде проводитися з неї до кінця файлу. Дві адреси 'Corner1:Corner2' визначають прямокутну область з початком 'Corner1' та закінченням 'Corner2'.

Наведемо приклади використання функції.

Запис **T=readtable('xl.xlsx')** зчитує з файлу **'xl.xlsx'**, який має наведений нижче зміст

	А	В	С	D	E	F		A	В	С	D	E	F
1	Name	Sex	Age	Height	Weight	HealthLife	1	Name	Sex	Age	Height	Weight	HealthLife
2	Бейден	М	38	171.22	82.5	1	2	Вейден	М	66	166	47	1
3	Мукран	М	43	169	68.9	0	3	Куркуль	F	28	148	44	1
4	Муркуль	F	38	158.91	51	1	4						

Лист 1



дані з першого аркуша файлу в таблицю Т:

т = 3	3×6 tab	ole				
Name	Sex	Age	Heig	ght W	eight	HealthLife
 'Бейде	⊇н′	' <u>M'</u>	38	171.2	2 82.	5 1
' Мукра	ан'	'M'	43	16	9 68.	9 0
' Мурку	уль'	'F'	38	158.9	1 5	51 1

Запис **T=readtable('xl.xlsx', 'ReadRowNames', true)** зчитує з першого аркуша файлу **'xl.xlsx'** дані в таблицю **T** з назвами рядків:

```
T = 3 \times 5 table
```

	Sex	Age	Height	Weight	HealthLife
Бейден	<u>'</u> M'	38	171.22	82.5	1
Мукран	'M'	43	169	68.9	0
Муркуль	'F'	38	158.91	51	1

Властивість DimensionNames таблиці показує:

```
T.Properties.DimensionNames
ans = 1×2 cell array
{'Name'} {'Variables'}
```

'**Name** ' є ім'ям першого стовпця таблиці.

Запис

```
T=readtable('e:\xl.xlsx','Sheet',2,'Range','C2:F3',...
'ReadVariableNames',false)
```

зчитує з другого аркуша файлу 'xl.xlsx' дані в діапазоні C2:F3 в таблицю T:

т	=	2×4	table		
		Var1	Var2	Var3	Var4
		66	166	47	1
		28	148	44	1

Функція writetable в режимі spreadsheet записує дані з таблиці в файл *MS Excel*. За замовчанням запис проводиться з комірки **A1** першого аркуша таблиці *MS Excel*. Наведемо приклади використання функції.

Рядок writetable (T, 'xp.xlsx') записує в файл 'xp.xlsx' дані з таблиці T, яка наведена вище, починаючи з комірки A1 першого аркуша:



Рядок

writetable(T,'e:\xp.xlsx','Sheet','Запис','Range',... 'E2','WriteVariableNames',false)

записує в файл 'xp.xlsx' дані з таблиці **т** без назв стовпців з комірки **E3** аркуша **'Запис '**:

	D		Е	F	G	Н	
1							
2			66	166	47	1	
3			28	148	44	1	
		Лист1		<u>Запис / Sl</u>			

Спеціально для роботи з файлами *MS Excel* призначені функції **xlsfinfo, xlsread, xlswrite**.

Функція

## <[> status <, sheets, xlFormat] >= xlsfinfo(filename)

є інформаційною. Вона визначає, чи в змозі *Matlab* прочитати дані з файлу. Масив символів з повною специфікацією файлу **filename** є обов'язковим аргументом. Основним результатом є масив символів **status**. В разі сумісності даних повертається рядок '**Microsoft Excel Spreadsheet**', в разі несумісності – порожній текстовий рядок (' ').

Опціональний результат **sheets** є 1xn масивом комірок з назвами аркушів книги *MS Excel*.

Опціональний результат **xlFormat** є масивом символів, який визначає формат файлу: '**xlOpenXMLWorkbook**' для файлів Excel 2007 та вище, '**xlWorkbookNormal**' або '**xlExcel8**' для файлів Excel 97-2003, '**xlCsV**' для CSV файлів, '**xlHtml**' або '**xlWebArchive**' для HTML файлів.

Наприклад, запис

[s,sh,Form] = xlsfinfo('myExample.xlsx')
може видати наступний результат:

```
s = Microsoft Excel Spreadsheet
sh = 'Sheet1' 'Sheet2' 'Sheet3'
Form = xlOpenXMLWorkbook
Функція
```

```
<[>num <,txt,raw]>=
xlsread(filename <,sheet,xlRange,'basic', -1>)
```

зчитує в масив **num** дані з файлу *MS Excel* з повною специфікацією файлу **filename**, забраною в апострофи.

Опціональні аргументи уточнюють спосіб зчитування.

Аргумент **sheet** визначає аркуш книги *MS Excel*, з якого треба проводити зчитування у вигляді імені аркуша в апострофах або його номера;

**xlRange** є масивом символів, який визначає діапазон комірок *MS Excel* з яких буде проводитися зчитування. за правилами адресації A1 комірок *MS Excel* блоком комірок через символ двокрапки в апострофах (**'A1:B2**').

Аргумент **basic** записується у випадку відсутності на комп'ютері *MS Excel*. Не визначені попередні аргументи записуються апострофами.

Наприклад,

```
num = xlsread(filename,'','','basic').
```

Аргумент **-1** відкриває файл **filename** в *MS Excel* та дозволяє користувачеві обрати дані для зчитування інтерактивно. Не може застосовуватися разом з іншими опціональними аргументами.

Опціональний результат **txt** є масивом комірок з текстовими даними з *MS Excel*.

Опціональний результат **raw** є масивом комірок з даними з *MS Excel*. Отримання опціональних результатів значно збільшує час виконання операції. Рекомендується за потреби отримання результатів у вигляді масивів комірок застосовувати режим **basic**.

Наведемо приклади використання функції.

Запис **a=xlsread('excl.xlsx')** зчитує всі чисельні дані з першого аркуша книги *MS Excel* з файлу **'excl.xlsx'** 

	А	В	С	D	E
1	Number	First	Second	Third	
2	One	1	2	3	
3	Two	4	x	5	
4	Three	6	7		
5					

в матрицю **a**. Дані нечислових типів замінюються в матриці **a** не значення **NaN**:

a =			
	1	2	3
	4	NaN	5
	6	7	NaN

Запис [a b c]=xlsread('excl.xlsx','','','basic') зчитує чисельні дані з першого аркуша книги *MS Excel* з файлу 'excl.xlsx' в матрицю **a**, текстові дані – в масив комірок **b**, всі дані – в масив комірок **c**:

a =							
	1 2	3					
	4 NaN	5					
	6 7	NaN					
b =	4×4 cell	array					
	{'Number'	} {'Fi	.rst' }	{ 'Sec	ond'}	{ ' Th	hird' }
	{'One' }	{ 0×0	char}	{0×0}	char}	{0×0	) char}
	{'Two' }	{ 0×0	char}	{'x'	}	{0×0	) char}
	{ 'Three ' }	{ 0×0	char}	{0×0}	char}	{0×0	) char}
c =	4×4 cell	array					
	{'Number'	} {'E	'irst'}	{ 'Sec	ond'}	{ ' Th	nird'}
	{ 'One '	]} {[	1]}	] }	2]}	]}	3]}
	{ 'Two '	]} {[	4]}	{ <b>'x</b> '	}	]}	5]}
	{ 'Three '	]} {[	6]}	] }	7]}	]}	NaN]}

Запис columnB = xlsread('excl.xlsx', 'B:B') зчитує чисельні дані зі стовпця В першого аркуша книги *MS Excel* з файлу 'excl.xlsx' в вектор columnB:

```
columnB =
2
NaN
7
```

Записує дані з масиву **A** в файл *MS Excel*, який визначений забраною в апострофи повною специфікацією файлу **filename** функція

# <[status,message]=>

# xlswrite (filename ,A <,sheet,xlRange>)

За замовчанням дані записуються з комірки A1 першого аркуша книги *MS Excel*. Опціональні аргументи **sheet**, **xlRange** мають значення та визначаються аналогічно функції **xlsread**.

При створенні нового аркуша книги функція видає на екран повідомлення:

# Warning: Added specified worksheet. > In xlswrite>activate\_sheet In xlswrite/ExecuteWrite In xlswrite

Для того, щоб не виводити повідомлення можна застосувати функцію warnings: warning('off', 'MATLAB:xlswrite:AddSheet').

Опціональний результат **status** є цілим числом, яке визначає статус операції. Значення 1 показує, що операція запису була успішною, значення 0 показує помилку запису.

Опціональний результат **message** є структурою, яка містить системне повідомлення про результат операції.

Наведемо приклади використання функції.

Рядок **xlswrite** ('**excwr.xlsx**') записує дані з матриці **a** в файл '**excwr.xlsx**' з початком в комірці **A1** першого аркуша книги *MS Excel*:



Рядок **xlswrite** ('excwr.xlsx', A, 'Data', 'B2') записує чисельні та текстові дані з масиву комірок **A** 

A =	4×	2 cell	array	
	{ ' T	'ime'}	{ 'Tempe	erature'}
	<b>{</b> [	12]}	{ [	12]}
	<b>{</b> [	13]}	{ [	13]}
	{[	14]}	{[	15]}

в файл 'excwr.xlsx' з початком в комірці B2 аркуша 'Data' книги *MS Excel*:

	A	В	С	D
1				
2		Time	Temperati	ure
3		12	12	
4		13	13	
5		14	15	
6				
7				
	⊧ ► ► Ли	ст1 Data	<u>@</u> ]	

### 5.7.2. Засоби інтерактивного файлового обміну

Для інтерактивного імпортування даних із зовнішніх файлів слугує додаток **Import Tool**. Додаток можна активувати з клавіатури введенням команди **uiimport** в командному рядку або кнопкою crpiчки **HOME**. Якщо у вікні **Current Folder** обрано файл, то імпортувати дані з обраного файлу можна пунктом контекстного меню **Import Data**... (рис. 5.28 а ) файлу. Додаток автоматично розпізнає дані з файлів з розширенням .*txt*, .*dat*, .*csv*, .*asc*, .*tab*, .*dlm*,.*xls*,.*xlsx*.

Виклик додатка з командного рядка супроводжується виведенням допоміжного діалогового вікна уточнення (рис. 5.28 б). Для того, щоб заблокувати етап уточнення, слід викликати додаток командою uiimport('-file').



Рис. 5.28. Вікна уточнення інтерактивного обміну: а - Import Data; б – уточнення

В центрі вікна (рис. 5.29) виводиться таблиця розпізнаних імпортованих даних. Адресація комірок таблиці наводиться в стилі «A1» *MS Excel*. Під адресами комірок виводиться ім'я змінної, яку буде передано в робочий простір пакету. За замовчанням змінній надається ім'я файлу. Під іменем змінної виводиться рядок з іменами змінних таблиці пакету, які розпізнані

пакетом в файлі та тип даних у відповідних стовпцях файлу. Таблиця є інтерактивною. Імена змінних, тип даних стовпців можна змінювати. Редагування імен проводиться після подвійним кліку на обраному імені. Тип даних обирається зі списку у вікні типів даних (рис. 5.29 б). Вікно відкривається натискання позначки трикутника ліворуч від вказаного для стовпця типу даних.

IMPORT	VIEW	/				S#4.1	$\times$			
<ul> <li>Delimited</li> <li>Fixed Width</li> </ul>	Column delimite Space @ Delimiter	variable N	Range: A1 James Row: 1	1:16 -	tput Type: Table • Text Options •	UNIMPORTA				
DELIN	DELIMITERS SELECTION IMPORTED DATA									
A	A B C D				F	G	G H			
Series	Date Text	Data1	Data2	Data3	Res1	Res2 ▼Number	Corr •Text	Compl •Text		
1 Series	Date	Data1	Data2	Data3	Res1	Res2	Corr	Compl		
2 Ser1,29/09/										
3 /*	not	correct								
Ser2,13/19/										
5*/										
5 Ser3,19/10/	,0.1,Yes,1+3									
	Number	✓Number	<b>-</b> Text	<b>-</b> ⊤						
	Text (string	)								

a

ext (string)		
Text	Column delimiters:	
ext like 1.234 will convert to string "1.234"	Comma 🗸	Output Tupo:
lumber (double)	STANDARD DELIMITERS	Output Type.
<b>Jumber</b> ext like "1.234" will convert to number 1.234	Tab	🔳 Table 🗖
Categories (categorical)	📲 🖾 Comma	🖽 Table
Categorical	Space	🖣 Column vectors
ext like 'orange' will convert to categorical orange Dates and Times (datetime)	Semicolon	🔲 🗄 Numeric Matrix
Id-MMM-yyyy HH:mm:ss	CUSTOM DELIMITERS	🔤 String Array
nere data formata	🔲 Enter Custom Delimiter	Cell Array

Рис. 5.29. Додаток **dImport tool**: а – діалогове вікно; б – визначення типів даних; в – визначення розділення; г – визначення типу результату

Над таблицею на вкладці **ІМРОКТ** розташовані поля керування імпортом даних.

Група DELIMETERS визначає розділювач даних та містить перемикач типу файлу Delimeted/Fixed Width, список можливих розділювачів Column delimeters (рис. 5.29 в).

Група **SELECTION** визначає діапазон комірок для імпорту та містить поле діапазону **Range** та поле номеру рядка з іменами змінних стовпців таблиці **Variable Names Row**. Визначити адреси блока комірок діапазон

даних, які планується імпортувати, можна виділенням комірок рамкою в зображеній таблиці.

Група **IMPORTED DATA** визначає тип вихідної змінної. Тип результату визначається пунктом випадаючого списку **Ouput Type** (рис. 5.29 г). Можливими типами є матриця **Numeric Matrix**, таблиця **Table**, масив комірок **Cell Array**, текстовий масив **String Array**, вектори стовпців **Column vectors**.

Список кнопки UNIMPORTABLE CELLS визначає дії з нерозпізнаними даними (рис. 5.30). Можливими діями є заміна символів Replace нерозпізнані (є режимом за замовчанням) та відсутні дані unimportable cells, виключення рядків Exclude rows with та/або стовпців Exclude columns with.



Рис. 5.30. Група unimportable cells

Пояснимо роботу з додатком. В якості даних використаємо описаний вище для функції **textscan** ASCII файл *dat.txt*. Файл містить числові та текстові дані з розділенням комою. Між елементуми в файлі наявні пробіли. Два рядки в файлі позначені як некоректні.

Додаток за замовчанням розпізнав файл як текстовий з розділювачем пробілом, про що свідчить активований перемикач **Delimited** на стрічці та слово в списку розділювачів, перший рядок розпізнаний як рядок з іменами змінних стовпців 'Level', 'Date', 'Datal' тощо. Дані задано з першим рядком включно.

Зчитування всіх рядків файлу можливо тільки у вигляді текстового масиву **String Array**, імпорт даних, в яких наявні рядки з відмінною інформацією як матриць, масивів комірок інтерактивно неможливий.

Навіть в текстовому вигляді файл за замовчанням зчитується некоректно. Результатом є текстовий масив 6х9 (рис. 5.31 а).

6x9 string				<ul> <li>Delimited</li> <li>Fixed Width</li> </ul>	Column delimi Comma @ Delimiter	variabl	Range: A	A2:16 •	utput Type: String Array Text Options	UNIMPO	ORTABLE CELLS	Import Selection -
1	2	3	4	DELI	ATERS		SELECTION		IMPORTED DATA			IMPORT
1 Series	Date	Data1	Data2	dat.txt 🛪	P	C	D	r	c	G	ч	
2 Ser1,29/09/2021,				A	D	C	U	dat	r	9	п	
3 /*	not	correct		Text	Text	•Text	-Text	■Text	- Text	Text	Text	Text
4 Ser2,13/09/2021,				2 Ser1	29/09/2021	12.34	45	1.2e4	Inf	NaN	Yes	5.1+3i
5 /*				3 /* not correct	42/00/2024	45.44	46	2-6	25.2	0.4	N	4.5.01
5 Ser3.19/10/2021	.0.1.Yes.1+3.1i			4 Ser2 5 */	13/09/2021	45.11	10	260	25.5	0.1	NO	1.5-21
7				6 Ser3	19/10/2021	34.12	521	-43.1	0.1	Yes	1+3.1i	

Рис. 5.31. Інтерактивне зчитування: а – в текстовий масив за замовчанням; б – в текстовий масив;

В першому рядку в 9 комірках містяться назви стовпців, в другому та четвертому в першій комірці – вміст всіх комірок відповідних рядків файлу, в третьому в першій комірці – позначка коментаря, в другій слово **not**, в третій – слово **correct**. В п'ятому в першій комірці – частина рядка файлу до порожньої комірки, в другій комірці – залишок рядка.

Зміна розділювача на використану в файлі кому покращує, але не виправляє ситуацію (рис. 5.31 б). Дані з рядків вірно рознесені по комірках, але порожню комірку в останньому рядку файлу додаток не враховує.

Приберемо закоментовані рядки та порожню комірку з файлу.

Зчитування в таблицю без обов'язкового визначення дій з нерозпізнаними даними неможливо. Додаток не дозволяє перенести дані в робочий простір.

Результатом імпорту з визначенням дій **Replace** є таблиця 4х9. Інформацію перенесено невірно (рис. 5.32 а). Дев'ять комірок першого рядка містять викривлені назви стовпців файлової таблиці. В перших комірках інших рядків міститься вся інформація з відповідних рядків файлу. Інші комірки рядків містять порожній символ або **NaN**.

Визначимо вірний розділювач, тип першого стовпця **text** та область зчитування без першого рядка. Результатом імпорту є таблиця 3х9. Дані вірно рознесені по комірках та розпізнані. Хоча додаток показав, що не розпізнав вбудовані змінні **inf**, **NaN** та комплексне число з дійсним значенням уявної частини (рис. 5.32 б), імпорт проведено вірно (рис. 5.32 в).

								<ul> <li>Delimited</li> <li>Fixed Width</li> </ul>	Column delim Comma,Spa. Ø Delimiter	iters:	Range: A Names Row: 1	2:14 • Out; • 0 ti • 0 ti	out Type: Fable ext Options	▼ UNIMPORT	ABLE CELLS	Import Selection V
								DELIN dat tyt	AITERS		SELECTION		MPORTED DATA			IMPORT
		4x9 tab	le					A	В	С	D	E	F	G	н	1
		1 Serie	2 s Date	D	3 ata1	4 Data2		Series	Date Datetime	Data1	Data2	dat Data3	Res1	Res2	Corr	Compl
	1	'Series'	'Date'		1	2	1	Series	Date	Data1	Data2	Data3	Res1	Res2	Corr	Compl
	2	'Ser1,29,	/09 ''		NaN	NaN	2	Ser1	29/09/2021	12.34	45	1.2e4	Inf	NaN	Yes	5.1+3i
	3	'Ser2,13,	/09 ''		NaN	NaN	3	Ser2	13/09/2021	45.11	16	2e6	25.3	0.1	No	1.5-2i
a	4	'Ser3.19	/10'.0.1.Yes.	1+	NaN	NaN	6 4	Ser3	19/10/2021	34.12	521	-43.1	12	0.1	Yes	1+3.1i
		3x9 <u>tab</u>	le													
		1	2	3	4	5	6	7	8	9	)					
		Series	Date	Data1	Data2	Data3	Res1	Res2	Corr	Cor	npl					
	1	'Ser1'	29/09/2021	12.3400	45	12000	Ir	nf N	laN 'Yes'	5.1000 +	3.0000i					
	2	'Ser2'	13/09/2021	45.1100	16	2000000	25.300	0 0.1	000 'No'	1.5000 -	2.0000i					
B	3	'Ser3'	19/10/2021	34.1200	521	-43.1000	1	2 0.1	000 'Yes'	1.0000 +	3.1000i					

Рис. 5.32. Інтерактивне зчитування: а – в таблицю за замовчанням; б – в таблицю; в – результат

Відмітимо область числових даних C2:G4, I2:I4 та збережемо дані з типом вихідної змінної Numeric Matrix як матрицю А. Додаток показав, що не розпізнав вбудовані змінні inf, NaN та комплексне число з дійсним значенням уявної частини (рис. 5.33 а). Проте результатом імпорту є матриця 3x6. Дані вірно рознесені по комірках. Всі елементи матриці набули комплексного типу (рис. 5.33 б) з причини того, що елементи останнього стовпця файлу задані як комплексні. Вбудовані змінні inf, NaN розпізнані вірно.



Рис. 5.33. Зчитування в матрицю: а – вікно; б – результат

Проведемо імпорт в вектори Column Vectors. Додаток показав, що не розпізнав вбудовані змінні **inf**, **NaN** та комплексне число з дійсним значенням уявної частини (рис. 5.34 а). Проте результатом є сім векторів 4x1, масив комірок та масив дат (рис. 5.34 б). Імена змінних відповідають назвам стовпців файлової таблиці. Першим елементом всіх змінних є некоректно розпізнані дані першого рядка файлової таблиці. Масив комірок призначено стовпцю **Corr** з текстовими даними, масив дат – стовпцю дат файлової таблиці.



Рис. 5.34. Зчитування в матрицю: а – вікно; б – результат за замовчанням; в – відредаговане зчитування

Відмітимо область з числових та текстових даних C2:G4, I2:I4 без першого рядка з іменами та збережемо дані в вектори. Додаток показав, що не розпізнав вбудовані змінні **inf**, **NaN** та комплексне число з дійсним значенням уявної частини. Проте результатом є 5 векторів 3x1, та масив комірок 3x1 (рис. 5.34 в). Імена змінних відповідають назвам стовпців файлової таблиці. Всі дані, включно з вбудованими змінними **inf**, **NaN** та комплексними числами розпізнані вірно.

Проведемо імпорт в масив комірок **Cell Array**. Додаток показав, що не розпізнав вбудовані змінні **inf**, **NaN**, комірку **Compl** та комплексне число з дійсним значенням уявної частини (рис. 5.35 а). Проте результатом є масив комірок векторів 4х9 (рис. 5.35 б). Імпорт не можна вважати успішним. Дані по комірках рознесені вірно. Елементуми першого рядка масиву є некоректно розпізнані дані першого рядка файлової таблиці. Вбудована змінна **inf** та комплексне число з дійсним значенням уявної частини замінені на **NaN**.

Відмітимо область без першого рядка з іменами та збережемо дані. Додаток показав, що не розпізнав вбудовані змінні **inf**, **NaN** та комплексне число з дійсним значенням уявної частини. Проте результатом є масив комірок 3x1 (рис. 5.35 в). Всі дані, включно з вбудованими змінними **inf**, **NaN** та комплексними числами розпізнані вірно. Масив комірок не може містити дані типу «дата», тому такі дані розпізнаються як текстові.

<ul> <li>Delimited</li> <li>Fixed Width</li> </ul>	Comma,Spa	Variable N	Range: A1:I ames Row: 1	4 • 0 c	ell Array 👻	UNIMPORTA	BLE CELLS	Import Selection 👻
DELIN	MITERS		SELECTION	IN	IPORTED DATA			IMPORT
dat.txt 🛛								
А	В	С	D	E	F	G	Н	I.
				dat				
Text •	Text	Number	Number	Number	-Number	-Number	•Text	-Number
Series	Date	Data1	Data2	Data3	Res1	Res2	Corr	Compl
Ser1	29/09/2021	12.34	45	1.2e4	Inf	NaN	Yes	5.1+3i
Ser2	2 13/09/2021 45.11 16		16	2e6 25.3		0.1	No	1.5-2i
Sor3	3 19/10/2021 34.12 521		521	-43.1	12	0.1	Yes	1+3.1i

1 4x9 <u>ce</u>	<u> </u>								{]	3x9 <u>ce</u>	<u>I</u>							
1	2	3	4	5	6	7	8	9		1	2	3	4	5	6	7	8	9
1 'Series'	'Date'	1	2	3	1 NoN	2 NoN	'Corr'	NaN 5 1000 + 2	1	'Ser1'	29/09/2021	12.3400	45	12000	Inf	NaN	'Yes'	5.1000 + 3
3 'Ser2'	13/09/2021	45.1100	45	2000000	25.3000	0.1000	'No'	1.5000 - 2	2	'Ser2'	'13/09/2021	45.1100	16	2000000	25.3000	0.1000	'No'	1.5000 - 2
4 'Ser3'	'19/10/2021	34.1200	521	-43.1000	12	0.1000	'Yes'	NaN	3	'Ser3'	'19/10/2021	34.1200	521	-43.1000	12	0.1000	'Yes'	1.0000 + 3
													1	1	1		1	
							6				-							
							0				B							

Рис. 5.35. Зчитування в масив комірок: а – вікно; б – результат за замовчанням; в – відредаговане зчитування

Покажемо функціонування додатку на прикладі зчитування файлу dat.xslx *MS Excel*, в який імпортований текстовий файл dat.txt. Файл містить дві однакові сторінки. Комірки таблиці відформатовані відповідно до даних в них (рис. 5.36). *MS Excel* не працює з комплексними числами, тому для таких комірок обрано загальний формат.

	А	В	С	D	Е	F	G	н	1
1	Series	Date	Data1	Data2	Data3	Res1	Res2	Corr	Compl
2	Ser1	29.09.2021	12.34	45	1.20E+04	Inf	NaN	Yes	5.1+3i
3	Ser2	13.09.2021	45.11	16	2.00E+06	25.3	0.1	No	1.5-2i
4	Ser3	19.10.2021	34.12	521	-43.1	12	0.1	Yes	1+3.1i
5									

Рис. 5.36. Файл dat.xslx

Додаток розпізнав файл як книгу MS Excel та обидві сторінки.

Режим матриці, масиву комірок та масиву тексту прибирають рядок з іменами змінних стовпців та можливість зміни типу стовпця. Зчитування в таблицю, вектори, матрицю без обов'язкового визначення дій з нерозпізнаними даними неможливо. Додаток не дозволяє перенести дані в робочий простір.

Результатом зчитування в матрицю є матриця 3х9. Додаток автоматично відкинув перший рядок з іменами стовпців. За результатами імпорту в матриці збережені всі дійсні числові значення. Текстові, комплексні значення та дати замінені на **NaN** (рис. 5.37). Примусове розширення діапазону на всю файлову таблицю результатом дасть матрицю 4х9. Перший рядок матриці з іменами сповпців буде заповнено символами **NaN**.



Рис. 5.37. Імпорт з файлу MS Excel в матрицю: а – вікно; б – результат

Зчитування в змінну типу масив комірок не показує нерозпізнаних коміок (рис. 5.38 а). За результатами імпорту в масив комірок збережені всі текстові та дійсні чисельні значення. Комплексні значення та вбудовані константи **Inf**, **NaN** замінені на текст (рис. 5.38 б).



Рис. 5.38. Імпорт з файлу *MS Excel* в масив комірок: : а – вікно; б – результат

Режим таблиці залишає рядок з назвами стовпців та можливість зміни типу стовпця (рис. 5.39 а). Хоча додаток вказав як нерозпізнані вбудовані змінні **Inf**, **NaN**, результатом імпорту є таблиця 3х9. За результатами імпорту в таблицю комірок збережені всі текстові, дійсні числові значення, дати. Комплексні значення та вбудована змінна **Inf** замінені на текст (рис. 5.39 б).



Рис. 5.39 – Імпорт з файлу *MS Excel* в таблицю: а – вікно; б – результат

# Контрольні запитання

- 1. Чим відрізняються програмний та інтерактивний файловий обмін?
- 2. Назвіть етапи низькорівневого файлового обміну?
- 3. Шо таке повна специфікація файлу?
- 4. Як проводять діалог функції **uigetfile**, **uiputfile**?
- 5. В яких режимах може відкриватися файл за низькорівневого обміну?
- 6. Що показує опціональний результат функції **textscan**?
- 7. Як визначається формат даних для функції textscan?
- 8. Що означає властивість 'CommentStyle'?
- 9. Що означає властивість 'Delimiter'?
- 10.Який тип має зчитана функцією **textscan** інформація за замовчанням?
- 11.В чому полягає особливість функцій обміну високого рівня?
- 12.В чому полягає різниця між функціями **сsvread** та **dlmread**?
- 13.Як визначається діапазон зчитування в функціях обміну високого рівня?
- 14.Що означає властивість '**ReadVariableNames**'?
- 15.Що означає властивість ' **ReadRowNames** '?
- 16.Як прочитати функцією **readtable** дані з файлів, які мають нестандартне розширення?
- 17. Яким функціями проводиться програмний обмін з файлами MS Excel?
- 18.Як визначити аркуш книги MS Excel для обміну?
- 19. Яку інформацію видає функція **xlsfinfo**?
- 20.Для чого призначено аргумент basic функції xlsread?
- 21.Для чого призначено аргумент -1 функції **xlsread**?
- 22.Що є результатом функції **xlsread**?
- 23.Як викликати додаток Import Tool?
- 24.В якому вигляді можна імпортувати дані додатком Import Tool?

## ВИКОРИСТАНІ ДЖЕРЕЛА

- 1. Кравченко, І. В. Інформаційні технології в приладобудуванні. Практикум: Навчальний посібник / І. В. Кравченко, М. С. Мамута. – К. : КПІ ім. Ігоря Сікорського, 2021. – 300с.
- Кравченко, І. В. Інформаційні технології. Системи комп'ютерної математики: Навчальний посібник / І. В. Кравченко, В. І. Микитенко. К. : КПІ ім. Ігоря Сікорського, 2018. 233с.
- ДСТУ ISO/IEC 2382:2017 (ISO/IEC 2382:2015, IDT) Інформаційні технології. Словник термінів. Чинний від 2019-01-01. Офіц. вид.. – К. : Держстандарт України, 2017. – 468с.
- 4. ДСТУ 2226-93. Автоматизовані системи. Терміни та визначення. Чинний від 1994-01-07. Офіц. вид.. – К. : Держстандарт України, 1994. – 91с.
- 5. ISO-IEC\_2382-1:1993 Information technology Vocabulary. Part 1: Fundamental terms [Електронні дані]. – Режим доступу: https://www.iso.org/obp/ui/#iso:std:iso-iec:2382:-1:ed-3:v1:en.
- ДСТУ 5034:2008. Інформація та документація. Науковоінформаційна діяльність. Терміни та визначення понять. Чинний від 2009-01-01. Офіц. вид.. – К. : Держстандарт України, 2008. – 41с.
- 7. ДСТУ 2481-94 Системи оброблення інформації. Інтелектуальні інформаційні технології. Терміни та визначення. Чинний від 1995-01-01. Офіц. вид.. К. : Держстандарт України, 1994. 72с.
- 8. Brooksheer, J. Computer Science. An overview / J. Glenn Brooksheer. 7th edition – Pearson Education, Inc, 2003. – 599 p.
- ДСТУ ISO/IEC 7498-1:2004 Інформаційні технології. Взаємозв'язок відкритих систем. Базова еталонна модель. Частина 1. Базова модель (ISO/IEC 7498-1:1994, IDT). Чинний від 2006-01-01. Офіц. вид.. – К. : Держстандарт України, 1994. – 67с.
- 10.A history of cloud computing / Arif Mohamed. Режим доступу: http://www.computerweekly.com/feature/A-history-of-cloud-computing/ (Дата звернення: 10.10.2022).
- 11. The NIST Definition of Cloud Computing / P. Mell, T. Grance. Режим<br/>доступу: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/<br/>nistspecialpublication800-145.pdf.
- 12. ISO/IEC 17788:2014 Information technology. Cloud computing. Overview and vocabulary. ISO/IEC, 2014. 16 p.

- 13. URL: https://ru.wikipedia.org/wiki/Облачные вычисления (Дата звернення: 15.10.2022).
- 14. ДСТУ ГОСТ 2.104:2006 Єдина система конструкторської документації. Основні написи. Чинний від 2007-07-01. Київ: ДП «УкрНДНЦ». 2006. 23с.
- 15. ГОСТ 2.105-95 Единая система конструкторской документации. Общие требования к текстовым документам. Введен 1997-07-01. – Минск: Межгосударственный совет по стандартизации, метрологии и сертификации. – 1996. – 45с.
- 16.Tanenbaum, A. Computer networks / Andrew S. Tanenbaum, David J. Wetherall. – 5th edition. — Prentice hall, 2003. — 962 p.
- 17. ДСТУ 3008:2015 Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Чинний від 2017-07-01. К. : ДП «УкрНДНЦ». 2016. 31с.
- 18. ДСТУ EN ISO 7200:2005 Розроблення технічної документації. Графи у штампах та основних написах. Чинний від 2007-12-01. К. : ДП «УкрНДНЦ». 2006. 11с.
- ДСТУ ГОСТ 7.1:2006 Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання. Чинний від 2008-04-01. – К. : ДП «УкрНДНЦ». – 2007. – 56с.
- 20. ГОСТ 2.106-96 Единая система конструкторской документации. Текстовые документи. Введен 1999-01-01. – Минск: Межгосударственный совет по стандартизации, метрологии и сертификации. – 1996. – 59с.
- 21. URL: http://scinews.kpi.Ua/about/submissions#authorGuidelines (дата звернення 10.11.2022).
- 22. ДСТУ 8302:2015 Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Чинний від 2016-07-01. К. : ДП «УкрНДНЦ». 2016. 20с.
- 23. Український правопис. [Електронний ресурс]. Затверджено Українською національною комісією з питань правопису (протокол № 5 від 22 жовтня 2018 р.) 2019. 282 с. Назва з екрана. Режим доступу https://mon.gov.ua/storage/app/media/zagalna%20serednya/ 05062019-onovl-pravo.pdf
- 24. Danaila, Ionut An Introduction to Scientific Computing / I. Danaila,P. Joly, S. M. Kaber, M. Postel. Springer, 2007. 296 p.

- 25. Кобильник, Т. П. Системи комп'ютерної інформатики у навчанні студентів напрямку «Інформатика» / Т. П. Кобильник, У. П.Когут // Інформаційні технології і засоби навчання. 2014. Том 40. №2. С.50-64.
- 26. Триус, Ю. В. Комп'ютерно-орієнтовані методичні системи навчання математики: монографія / Юрій Васильович Триус. – Черкаси: Брама-Україна, 2005. – 400 с.
- 27. Жалдак М. І. Математика з комп'ютерною підтримкою : Навч. посіб.
  / М. І. Жалдак, А. В. Грохольська, О. Б. Жильцов; Міжрегіон. акад. упр. персоналом. – К., 2004. – 456 с.
- 28. Лазарєв, Ю. Ф. Моделювання на ЕОМ. Навчальний посібник / Ю. Ф. Лазарєв. К. : Політехніка, 2007. 290 с.
- 29. Лазарєв, Ю. Ф., Початки програмування у середовшщ MatLAB: Навч. посібник / Ю. Ф. Лазарєв. К. : Корнійчук, 1999. 160 с.
- 30. User's Guide MathCAD 15.0 Parametric Technology Corporation, 2011. 168 p.
- 31. MATLAB® Desktop Tools and Development Environment. The MathWorks, Inc., 2010. 1063 p.
- 32. MATLAB Function Reference The MathWorks, Inc., 2016. 11052 p.
- MATLAB® Symbolic Math Toolbox The MathWorks, Inc., 2016. 1800 p.
- 34. MuPAD® User's Guide The MathWorks, Inc., 2016. 1004 p.
- 35. MATLAB® Data Import and Export The MathWorks, Inc., 2016. 444 p.