

Міністерство освіти і науки України  
Національний університет водного господарства та  
природокористування  
Навчально-науковий інститут автоматики, кібернетики та  
обчислювальної техніки  
Кафедра комп'ютерних наук та прикладної математики

**04-01-56М**

**КОНСПЕКТ ЛЕКЦІЙ**  
з навчальної дисципліни  
**КОМП'ЮТЕРНА МАТЕМАТИКА**  
для здобувачів вищої освіти першого (бакалаврського) рівня  
за освітньо-професійними програмами  
«Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» та  
«Прикладна математика» спеціальності 113 «Прикладна математика»  
денної та заочної форми навчання

Рекомендовано  
науково-методичною радою з  
якості ННІАКОТ

Протокол № 4  
від 11 лютого 2021 р.

Конспект лекцій з навчальної дисципліни «Комп'ютерна математика» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійними програмами «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» та «Прикладна математика» спеціальності 113 «Прикладна математика» денної та заочної форми навчання [Електронне видання] / Демчук О. С. – Рівне : НУВГП, 2021. – 62 с.

**Укладач:** Демчук О. С., доцент кафедри комп'ютерних наук та прикладної математики, кандидат технічних наук.

**Відповідальний за випуск:** Турбал Ю. В., в.о. завідувача кафедри комп'ютерних наук та прикладної математики, доктор технічних наук, професор.

**Керівник групи забезпечення спеціальності 122 «Комп'ютерні науки»:** Мартинюк П. М., професор кафедри комп'ютерних наук та прикладної математики, доктор технічних наук, професор.

**Керівник групи забезпечення спеціальності 113 «Прикладна математика»:** Прищеп О. В., доцент кафедри комп'ютерних наук та прикладної математики, кандидат фізико-математичних наук, доцент.

## ЗМІСТ

Передмова	4
Тема 1. Вступ у комп'ютерну математику. Огляд сучасних систем комп'ютерної математики	6
Тема 2. Основи роботи з системою комп'ютерної математики Mathcad	11
Тема 3. Символьні перетворення виразів у Mathcad	21
Тема 4. Диференціювання	28
Тема 5. Інтегрування. Інтегральні перетворення	32
Тема 6. Чисельне і символічне розв'язання нелінійних алгебраїчних рівнянь та їх систем	42
Тема 7. Задачі Коші та крайові задачі для звичайних диференціальних рівнянь	48
Тема 8. Задачі Коші та крайові задачі для диференціальних рівнянь у частинних похідних	56
Література	62

## ПЕРЕДМОВА

У сучасному інформаційному світі стрімко зростає попит на спеціалістів, які поєднують ґрунтовну математичну підготовку зі знанням сучасних комп'ютерних технологій. Комп'ютерна математика як навчальна дисципліна є дуже актуальною на сьогодні для підготовки кваліфікованих фахівців у галузі прикладної математики і комп'ютерних наук. Вона поєднує фундаментальну математику зі спеціалізованими програмними засобами для прогнозування, комп'ютерно-математичного моделювання та оптимізації природно-технічних і суспільно-економічних процесів та явищ.

Сучасні системи комп'ютерної математики надають можливість чисельно та символічно розв'язувати задачі практично з усіх розділів математики, підтримують інтерактивну візуалізацію даних, одну чи декілька мов програмування, дозволяють комбінувати в одному файлі текст, математичні формули та результати обчислень з алгоритмами, комп'ютерною графікою та анімацією.

Більшість універсальних систем комп'ютерної математики дозволяє спрощувати алгебраїчні вирази, диференціювати та інтегрувати функції, розв'язувати алгебраїчні, диференціальні та різницеві рівняння, виконувати інтегральні перетворення, операції з матрицями і векторами, проводити статистичний аналіз даних, автоматично доводити та перевіряти теореми, експортувати власні коди в інші мови програмування тощо.

Однією із найбільш популярних у світі та зручних у використанні є система комп'ютерної алгебри Mathcad. Програма має зручний, інтуїтивно зрозумілий інтерфейс, де математичні формули представлені у традиційному вигляді. При створенні електронних документів засобами Mathcad використовується принцип WYSIWYG (What You See Is What You Get – «що бачиш, те й отримувеш»).

Mathcad зручно використовувати для навчання математиці, виконання різноманітних обчислень над даними та їх графічного представлення, підготовці електронних документів і книг. Хоча дана програма орієнтована на широке коло користувачів, що не є фахівцями у галузі інформаційних технологій, Mathcad, завдяки вбудованій мові програмування, також може використовуватись у складних обчислювальних системах для візуалізації результатів математичного моделювання та інженерних розрахунків.

Метою вивчення дисципліни «Комп'ютерна математика» є формування у майбутніх фахівців сучасного рівня володіння інформаційними технологіями розв'язання задач з різних розділів математики, побудови математичних моделей процесів та явищ, а також інтерактивної візуалізації результатів обчислень; оволодіння основними принципами роботи з універсальними комп'ютерними математичними системами; набуття практичних навичок розв'язання математичних та інженерних задач на прикладі системи комп'ютерної математики Mathcad.

В рамках дисципліни «Комп'ютерна математика» вивчаються основні терміни і поняття комп'ютерної математики, класифікація систем комп'ютерної математики; принципи роботи з універсальними комп'ютерними математичними системами; стандартні засоби СКМ Mathcad; призначення та функції її основних бібліотек та пакетів розширення; основні можливості візуалізації результатів обчислень в системі Mathcad; можливості інтеграції даної СКМ з мовами програмування та рядом інших офісних і графічних програм.

Теоретичним фундаментом для вивчення дисципліни «Комп'ютерна математика» є математичний аналіз, алгебра та геометрія, дискретна математика, теорія ймовірностей та математична статистика, методи обчислень, методи оптимізації та інші. Дана дисципліна буде корисною у подальшому вивченні таких дисциплін як математичне моделювання економічних, природно-технічних та соціальних систем, аналіз даних, випадкові процеси, теорія керування тощо.

При підготовці даного конспекту лекцій було використано матеріали книги Кирьянова Д. В. «Mathcad 15/Mathcad Prime 1.0», інші літературні джерела та електронні ресурси. Ряд тем, що не увійшли у конспект, достатньо детально і на прикладах описано у методичних вказівках до виконання лабораторних робіт з даної дисципліни. Для більш глибокого вивчення системи Mathcad студентам рекомендовано список літератури.

## Тема 1. Вступ у комп'ютерну математику. Огляд сучасних систем комп'ютерної математики

### 1.1. Основні поняття комп'ютерної математики. Класифікація систем комп'ютерної математики

Озн.: (за Дьяконовим) *Комп'ютерна математика* – це сукупність теоретичних, методичних, алгоритмічних, апаратних і програмних засобів, які призначені для ефективного розв'язування за допомогою комп'ютерів широкого кола математичних задач з високим ступенем візуалізації всіх етапів обчислень.

Програмні засоби комп'ютерної математики за призначенням, структурою та функціями поділяють на кілька груп, а саме:

1. Системи числових розрахунків, в тому числі калькулятор Windows.
2. Табличні процесори (*MS Excel, LibreOffice Calc, OpenOffice Calc*).
3. Системи для статистичних розрахунків (*Statistica, SPSS, SAS, STATGRAPHICS*)
4. Програмні засоби візуалізації математичних даних:
  - Редактори формул: *MathCast, MathType, New Number*.
  - *LaTeX* – мова розмітки даних та пакет макросів системи комп'ютерної верстки *TeX* для високоякісного оформлення математичних і технічних текстів, є стандартом для публікації в багатьох провідних наукових виданнях.
  - Системи побудови графіків функцій і поверхонь: *Axum Graph* (інтегрується в *MathCAD*); *AceIT Grapher; Advanced Grapher; 3D Grapher; GnuPlot* (вільна програма для створення дво- та тривимірних графіків); *MathPlot; SigmaPlot* (для аналізу і візуалізації наукових та статистичних даних); *Surfer* (для створення і редагування всіх типів карт і цифрових моделей рельєфу).
  - Системи геометричного моделювання: *Autodesk 3ds Max* (для створення і редагування тривимірних моделей об'єктів), *ANSYS* – універсальна програмна система для автоматизованих інженерних розрахунків і розв'язання просторових задач механіки деформованого твердого тіла і конструкцій, рідини і газу, теплопередачі і теплообміну, електродинаміки, акустики і т.д.
  - *Jmol* – програма для перегляду структури молекул в просторі. Використовується як в навчальних цілях, так і при проведенні наукових досліджень в галузі молекулярної біології і хімії. У вільному доступі.

## 5. Математичні пакети вузької спеціалізації:

- *GAP* (від англ. Groups, Algorithms, Programming) – вільна кросплатформна система комп'ютерної алгебри для виконання обчислень з дискретної математики, особливо теорії груп.

- *Macaulay* – комп'ютерна алгебраїчна система, придатна для многочленних обчислень із акцентом на обчисленнях на основі базису Грьобнера – множини, яка породжує ідеал заданого кільця многочленів із спеціальними властивостями.

- *Singular* – для перестановочної алгебри, алгебраїчної геометрії і теорії сингулярності.

- *Diffg, Dynamic Solver* – для розв'язання диференціальних рівнянь та їх систем.

- *TK Solver* для розв'язання нелінійних рівнянь.

- *EViews* – професійний пакет, орієнтований на аналіз часових рядів.

- *SPSS* – аналіз просторових даних, кластерний аналіз. Популярний серед маркетологів і соціологів, має розвинутий Windows інтерфейс і гарну графіку.

## 6. Універсальні системи комп'ютерної математики: *Derive, Mathcad, Maple, Matlab, Mathematica, Maxima, Sage* та ін.

Далі ми вивчатимемо саме такі системи, які називатимемо скорочено *системами комп'ютерної математики* (СКМ). Ці системи мають дружній інтерфейс, реалізують велику кількість стандартних і спеціальних математичних операцій, забезпечені потужними графічними засобами і мають власні мови програмування. Усе це надає широкі можливості для ефективної роботи фахівців різних профілів.

Озн.: *Системи комп'ютерної математики* – це поліфункціональні, універсальні програмні засоби, 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 значно полегшує і скорочує час розв'язання типових математичних задач, таких як обчислення значень функцій і побудова їхніх графіків, розв'язування рівнянь, нерівностей і їх систем, обчислення границь, інтегралів, похідних функцій тощо.

Сучасні СКМ – це, перш за все, потужні електронні довідники і бази даних з усіх сучасних напрямів математики, у них передбачені ефективні засоби розв’язування більшості типових математичних задач.

### 1.2. Структурна схема системи комп’ютерної математики

У кожній СКМ можуть бути нюанси у своїй архітектурі. Утім, сучасні СКМ мають таку типову структуру:



Рис. 1. Структурна схема системи комп’ютерної математики

*Ядро системи* – це коди заздалегідь відкомпільованих вбудованих процедур, функцій та операторів системи.

*Інтерфейс* дає користувачеві можливість звертатися до ядра зі своїми запитамі й одержувати результат на екрані дисплею.

Створені окремі *бібліотеки* процедур і функцій, які використовуються рідше. Метою такого підходу є обмеження об’єму ядра для підтримки максимальної швидкості роботи вбудованих функцій і процедур.

Можливе використання *пакетів розширення* систем для покращення їх функціональності й адаптації до конкретних завдань. Ці пакети пишуться власною мовою програмування, наявною в системі, що дає можливість їх підготовки звичайними користувачами.

Наявна розвинена *довідкова система*.

### 1.3. Чисельні та символні розрахунки

Більшість перших систем комп’ютерної математики (Eureka, Mercury, Excel, Lotus 123, Mathcad для MSDOS, PC MATLAB та ін.) призначалися для чисельних розрахунків. Вони перетворювали комп’ютер у великий програмований калькулятор, здатний швидко й автоматично (за введеною програмою) виконувати арифметичні і логічні операції над числами або масивами чисел. Результат обчислень завжди конкретний: це або число, або набір чисел.

Результати обчислень рідко були абсолютно точними в математичному сенсі. Як правило, під час виконання операцій з дійсними числами



відбувається їх округлення. Реалізація більшості чисельних методів також базується на наближених алгоритмах. Часто через накопичення похибок ці методи втрачають обчислювальну стійкість, даючи помилкові результати чи, навіть, приводять до «зависання» обчислювальної системи.

Не було можливості отримувати результати у вигляді загальних формул, які описують процес отримання результату. Як правило, з результатів числових обчислень неможливо було зробити узагальнені теоретичні, а часом і практичні висновки. Тому для реалізації серйозних наукових проєктів доводилось звертатись за допомогою математиків-аналітиків, які розв'язували потрібні завдання в аналітичному вигляді і пропонували доцільні методи їх чисельного розв'язування за допомогою комп'ютера. Такий підхід був дорогим і не дуже оперативним, що й привело до створення сучасних СКМ.

Усі сучасні математичні пакети: Mathcad, Maple, Matlab, Matematica – дозволяють проводити як символічні обчислення, так і використовувати чисельні методи. Тепер такі системи є одним з основних обчислювальних інструментів комп'ютерного моделювання в реальному часі і знаходять застосування в різних галузях науки.

Термін «комп'ютерна алгебра» є синонімом термінів «символьні обчислення», «аналітичні обчислення», «аналітичні перетворення» і т. д.

Головне завдання Систем Комп'ютерної Алгебри (Computer Algebra System – CAS) – обробка математичних виразів у символічній формі. Символьні операції зазвичай включають: обчислення символічних або числових значень виразів, перетворення виразів, знаходження похідної однієї або декількох змінних, розв'язування лінійних і нелінійних рівнянь, розв'язування диференціальних рівнянь, обчислення границь, обчислення інтегралів, роботу з множинами, роботу з векторами і матрицями. На додаток до перерахованого, більшість CAS підтримують різноманітні чисельні операції: розрахунок значень виразів при певних значеннях змінних, побудову графіків на площині і в просторі. Більшість CAS включають високорівневу мову програмування, яка дозволяє реалізувати свої власні алгоритми.

Озн.: *Комп'ютерна алгебра* — розділ інформатики, який займається розробкою, аналізом, реалізацією і застосуванням алгоритмів алгебри.

Від інших алгоритмів алгоритми алгебри відрізняються наявністю простих формальних описів, існуванням доведення правильності і меж часу виконання, які можна отримати на основі добре розвинутої

математичної теорії. Крім того, об'єкти алгебри можна точно подавати в пам'яті обчислювальної машини, завдяки чому алгебраїчні перетворення можуть бути виконані без втрати точності.

Методи символічних обчислень і чисельні алгоритми зазвичай доповнюють один одного.

#### **1.4. Огляд найбільш популярних систем комп'ютерної математики**

У всьому світі створено десятки СКМ, але широко відомі лише декілька.

*Derive, MuPAD* – СКМ початкового рівня. Вони орієнтовані на середню та вищу освіту за спеціальностями, що не потребують розширеної математичної підготовки. Графіка недостатньо розвинута, хоча і постійно покращується.

*Mathcad* – система, орієнтована на вищу освіту, виконання помірно складних чисельних і аналітичних розрахунків. Основна її перевага – математичні вирази на екрані комп'ютера представлені в загальноприйнятому вигляді. Завдяки простому інтерфейсу і гарній графічній візуалізації обчислень є наймасовішою СКМ.

*Maple, Mathematica* – універсальні системи, орієнтовані на виконання аналітичних обчислень на будь-якому рівні, аж до професійного. Широко застосовуються у вищій освіті і практиці виконання наукових розрахунків.

*MATLAB+Simulink* – потужні і великі системи, орієнтовані на матричні і числові методи, реалізацію численних розрахунків підвищеної складності, математичне моделювання систем і пристроїв. Мають десятки пакетів розширення по різним галузям математики і найсучаснішим сферам її застосування.

Можна виділити наступні загальні напрями розвитку сучасних СКМ:

- перетворення СКМ в інтелектуальні системи представлення знань та їх експертної оцінки;
- інтеграція систем між собою та з рядом інших офісних і графічних програм;
- розширені можливості обчислень, що охоплюють всі практично важливі і фундаментальні галузі математики;
- розширена ступінь візуалізації обчислень;
- перетворення СКМ в універсальні системи;

- введення нових функцій, напр. для вейвлет-перетворень (вейвлет-перетворення розглядають функцію, взяту як функцією від часу у термінах коливань, локалізованих за часом (простором) і частотою), обробки сигналів, реалізації нечіткої логіки, нейронних мереж тощо;

- введення в СКМ засобів, що дозволяють на їх основі готувати повноцінні електронні підручники в різних форматах;

- можливість створювати документи з текстами, формулами, рисунками і графіками найвищої поліграфічної якості.

## **Тема 2. Основи роботи з системою комп'ютерної математики Mathcad**

### **2.1. Знайомство з Mathcad**

Назва системи походить від двох слів – MATHeMatics (математика) і CAD (Computer Aided Design – системи автоматизованого проектування, або САПР).

Mathcad був задуманий і спочатку написаний Алленом Раздовим з Массачусетського технологічного інституту (MIT), співзасновником компанії Mathsoft Inc., яка з 2006 року є частиною корпорації PTC (Parametric Technology Corporation).

Деякі з математичних можливостей Mathcad (версії до 13.1 включно) засновані на підмножині системи комп'ютерної алгебри Maple (МКМ, Maple Kernel Mathsoft). Версії 14 та 15 використовують символічне ядро MuPAD. Версія нового покоління – Mathcad Prime 1.0 – символічні обчислення не підтримує. Однак версія Mathcad Prime 3.0, що вийшла в 2013 році, має безліч нововведень та вдосконалень в тому числі і символічні обчислення.

До складу Mathcad входить декілька інтегрованих між собою компонентів:

- потужний текстовий редактор для введення, редагування і форматування як текстів, так і математичних виразів;
- обчислювальний процесор, що вміє проводити розрахунки по введеним формулам, з використанням вбудованих числових методів;
- символічний процесор, що дозволяє проводити аналітичні обчислення і являє собою, фактично, систему штучного інтелекту;
- величезне сховище довідкової інформації математичної та інженерної, оформленої в якості інтерактивної електронної книги.

## 2.2. Інтерфейс Mathcad

Mathcad має простий і інтуїтивний для використання інтерфейс. Для введення формул і даних можна використовувати як клавіатуру, так і спеціальні панелі інструментів.

Робота здійснюється в межах робочого аркуша, на якому рівняння і вирази відображаються графічно, на протизагу текстовому запису в мовах програмування. При створенні документів-програм використовується принцип WYSIWYG (What You See Is What You Get – «що бачиш, те й отримуєш»).

Складові частини інтерфейсу Mathcad (зверху донизу):

- заголовок вікна;
- рядок меню;
- панелі інструментів (toolbars) Стандартна (Standard), Форматування (Formatting), панель інструментів Математичні (Math) і доступні через неї додаткові математичні панелі інструментів та ін;
- робоча область (worksheet);
- допоміжне вікно трасування (Trace Window);
- рядок стану (status line или status bar).

Панелі інструментів:

- *Стандартна (Standard)* – для виконання більшості операцій, таких як дії з файлами, редакторська правка, вставка об'єктів і доступ до довідкових систем;
- *Форматування (Formatting)* – для форматування (зміни типу і розміру шрифту, вирівнювання і т. д.) тексту і формул;
- *Математичні (Math)* – для вставки математичних символів і операторів в документи;
- *Елементи управління (Controls)* – для вставки в документи стандартних елементів управління інтерфейсом користувача (прапорців перевірки, полів вводу і т. д.);
- *Ресурси (Resources)* – для швидкого виклику ресурсів Mathcad (прикладів, підручників, електронних книг і т. д.);
- *Відлагодження (Debug)* – для управління відлагодженням Mathcad-програм.

Панель *Математичні (Math)* призначена для виклику на екран ще дев'яти панелей, з допомогою яких, проходить вставка математичних операцій в документи:

- *Калькулятор (Calculator)* – служить для вставки основних математичних операцій;
- *Графік (Graph)* – для вставки графіків;
- *Матриця (Matrix)* – для вставки матриць і матричних операторів;
- *Обчислення (Evaluation)* – для вставки операторів управління обчисленнями;
- *Математичний аналіз (Calculus)* – для вставки операторів інтегрування, диференціювання, сумування і т.д.;
- *Булева алгебра (Boolean)* – для вставки логічних (булевих) операторів;
- *Програмування (Programming)* – для програмування засобами Mathcad;
- *Грецькі символи (Greek)* – для вставки грецьких символів;
- *Символьні (Symbolic)* – для вставки символьних операторів;
- *Модифікатор (Modifier)* – для вставки деяких операторів (напр., перетворення числа);
- *Додаткові символи (Custom Characters)* – для вставки спеціальних символів (одиниць вимірювання температури і т.д.).

Примітка: Дві останні (з перерахованих) панелі використовуються досить рідко і тому викликаються не через панель *Математичні (Math)*, а за допомогою меню Вид / Панелі інструментов (View / Toolbars).

### 2.3. Оператори чисельного і символьного виводу


Обчислити значення виразу в Mathcad можна за допомогою операторів чисельного (=) та символьного (-> або <Ctrl>+<.>) виводу. У першому випадку ми отримуємо наближений (до трьох знаків після коми), а у другому – точний результат.

$$\text{acos}(0) = 1.571$$

$$\text{acos}(0) \rightarrow \frac{\pi}{2}$$

По замовчуванню обчислення у документі проводяться у режимі реального часу. Іноді, у випадку складних і довгих розрахунків, буває корисно зупинити їх натисканням клавіши <Esc>, а потім (у потрібний момент) відновити натисканням клавіши <F9> або командою Tools / Calculate /Calculate Now.

## 2.4. Математичні вирази і вбудовані функції

Більш складні і громіздкі обчислення в Mathcad проводять, користуючись арсеналом вбудованих функцій, закладених розробниками в систему Mathcad (на відміну від користувацьких функцій, сконструйованих безпосередньо при розробці Mathcad-програми). Легше всього вводити імена функцій з клавіатури або за допомогою кнопки  на стандартній панелі інструментів.

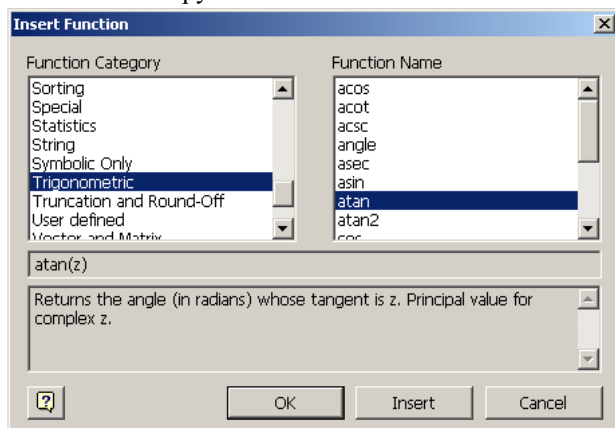


Рис. 2. Вікно вставки функції в системі Mathcad

## 2.5. Змінні і оператор присвоєння

Символ "=" у виразі говорить про обчислення значення зліва направо, а символ ":=" – про присвоєння значення справа наліво.

$x := 1$

$x = 1$

Якщо ввести знак "=" для змінної, яка вперше зустрічається в документі, він буде автоматично замінений символом присвоєння.

Для того щоб обчислити значення виразу, що містить деяку змінну, застосовують оператор чисельного виводу (змінна попередньо повинна бути задана). У випадку символічного виводу – це необов'язково.

$a := 3$

$$\frac{d}{dx} \sin\left(\frac{k \cdot x}{a}\right) \rightarrow \frac{k \cdot \cos\left(\frac{k \cdot x}{9}\right)}{9}$$

Існує можливість одночасного визначення змінної чи функції та числового виводу значення.

$$x := \sqrt{3} = 1.732$$

$$x \rightarrow 1.7320508075688772$$

$$y := \sqrt{3}$$

$$y \rightarrow \sqrt{3}$$

## 2.6. Функції користувача

Подібно присвоєнню числових значень змінним, можна задавати функції користувача від однієї або кількох змінних.

$$f(x) := x^2 - 3x - 2$$

$$f(0) = -2$$

$$f(10) = 68$$

$$g(a, y, \phi) := a \cdot \sin(y + \phi)$$

$$g(1, 0, \pi) = 0$$

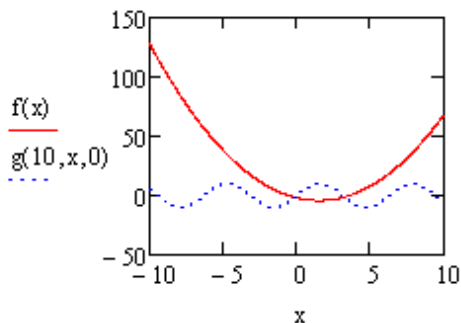


Рис. 3. Приклади побудови функцій користувача в Mathcad

## 2.7. Основні типи даних

**Дійсні числа.**

Крім десяткової форми у програмі Mathcad можна працювати з числами у двійковій (binary), вісімковій (octal) або шістнадцятковій (hexadecimal) системах числення.

$$\begin{array}{ll}
 a := 1000 & b := 1.4543 \\
 c := 3435.2 & d := 56.25 \cdot 10^{-15} \\
 a := 101011b & a = 43 \\
 b := 434o & b = 1.454 \\
 c := 1f3h & c = 499
 \end{array}$$

### Комплексні числа.

Більшість операцій в Mathcad по замовчуванню виконуються над комплексними числами.

$$a := i + 10$$

$$x := i \quad x = i$$

Замість символу  $i$  в уявній частині можна вводити також  $j$ .

Розглянемо ще кілька прикладів завдання комплексних чисел та операцій над ними.

$$\begin{array}{ll}
 a := 2i + 0.5 & a = 0.5 + 2i \\
 b := 1.77 \cdot e^{2i} & b = -0.737 + 1.609i \\
 c := 25j + 12 & c = 12 + 25i \\
 \text{Im}(a) = 2 & \text{Re}(a) = 0.5 \\
 |a| = 2.062 & \arg(a) = 1.326 \\
 \arg(b) = 2 & |b| = 1.77
 \end{array}$$

$$\text{Тут } |a| = \sqrt{\text{Re}^2 + \text{Im}^2}$$

$\arg(a)$  – кут в комплексній площині між дійсною віссю і  $a$ .  
Повертає результат між  $-\pi$  і  $\pi$  радіан.



## 2.8. Вбудовані константи

Деякі імена в Mathcad зарезервовані під системні змінні і називаються *вбудованими константами* (built-in constants). Вони діляться на два типи:

- *математичні константи* (math constants)

$$\infty = 1 \times 10^{307}$$

$$e = 2.718$$

$$\pi = 3.142$$

$$i = i$$

$$j = i$$

$$\% = 0.01$$

$$100 \cdot 25 \cdot \% = 25$$

- *системні змінні* (system variables), що визначають роботу більшості числових алгоритмів:

TOL – точність чисельних методів;

STOL – точність обчислення виразів, що використовуються в деяких чисельних методах;

ORIGIN – номер початкового індексу у масивах;

PRNPRECISION – установка формату даних при виводі у файл;

PRNCOLWIDTH – установка формату стовпця при виводі у файл;

CWD – рядкове представлення шляху до поточної робочої папки.

## 2.9. Рядкові змінні

Розміщуються в подвійних лапках. Системна константа ORIGIN може тепер також встановлювати початок відліку для функцій рядкового (текстового) аргументу. Щоб підключити цю опцію, встановлюють прапорець перевірки *Use ORIGIN for string indexing* на вкладці *Calculations* діалогового вікна *Worksheet options*.

```
s := "Hello,"          s = "Hello,"
concat(s, " world!") = "Hello, world!"
```

## 2.10. Ранжовані змінні та матриці

У програмі реалізовані масиви. Доступ до елемента здійснюється по індексу.

$$A := \begin{pmatrix} 0.1 & 0.2 & 0.3 \\ 4 & 5 & 6 \\ 7 & 8 & -9 \end{pmatrix}$$

$$A_{0,0} = 0.1$$

$$A_{2,0} = 7$$

Виділяють два типи масивів:

- *вектори* (одноіндексні), *матриці* (двоіндексні) та *тензори* (багатоіндексні);
- *ранжовані змінні* типу діапазон – вектори, значення яких певним чином залежать від індексу.

Примітка: Стартовий індекс масиву задається системною змінною ORIGIN, яка по замовчуванню рівна нулю. Щоб нумерувати елементи векторів і матриць з 1 необхідно ORIGIN~1.

В програмі можливий доступ до підмасиву, наприклад:

$$A^{(0)} = \begin{pmatrix} 0.1 \\ 4 \\ 7 \end{pmatrix}$$

$$(A^T)^{(2)T} = (7 \ 8 \ -9)$$

## 2.11. Розмірні змінні

В Mathcad числові змінні і функції можуть мати розмірність. У системі вбудовано велику кількість одиниць вимірювання. Щоб створити розмірну змінну, напр., силу струму в 10А, спочатку I:=10, потім символ множення <\*>, а потім літеру "A". Всі символи, що позначають одиниці вимірювання, зарезервовані. Проте, якщо раніше змінній A присвоювали якесь значення, то сприйматися як ампери вона вже не буде.

$$I := 10 \cdot A$$

$$U := 110 \cdot V$$

$$R := \frac{U}{I}$$

$$R = 11 \ \Omega$$

Інакше вставити одиницю вимірювання можна за доп. меню **Insert / Unit**. Потім в списку **Unit** діалогового вікна **Insert Unit** виберіть Ampere (A) **OK**.

Одиницю вимірювання в системі СІ будь-якої розмірної змінної можна вивести за доп. функції **SIUnitsOf()**.

$$v := 40 \cdot \frac{\text{km}}{\text{hr}} \quad v = 11.111 \frac{\text{m}}{\text{s}}$$

$$\text{SIUnitsOf}(v) = 1 \frac{\text{m}}{\text{s}}$$

## 2.12. Елементи інтерфейсу редактора формул

- Вказівник миші
- Курсор всередині документа може бути 3-ох видів:
  - курсор вводу (червоний хрестик) – позначає пусте місце, куди можна вводити текст або формулу;
  - лінії вводу (горизонт. і вертикал. лінії синього кольору), що виділяють в тексті чи формулі певну частину;
  - лінія вводу тексту (червона вертикал. лінія), для вводу текстових областей

Змінити лінії вводу можна мишкою або клавіатурою:

- стрілками;
- клавишою <Ins>, що переводить вертикальну лінію вводу з одного кінця горизонтальної лінії вводу на протилежний;
- пробілом, що призначений для виділення різних частин формули. При натисканні пробілу лінії вводу будуть почергово виділять одну із частин формули.
- Місцезаповнювачі – з'являються всередині незавершених формул:
  - місцезаповнювач символу – чорний прямокутник;
  - місцезаповнювач оператора – чорна прямокутна рамка.

## 2.13. Основи програмування у Mathcad

Традиційне програмування у дещо спрощеному вигляді реалізується у Mathcad за допомогою панелі інструментів **Programming**.

Основні можливості даної панелі:

- застосування циклів і умовних операторів;
- зручне завдання змінних і функцій;

– створення функцій, що містять закритий для останнього документу код, включаючи використання локальних змінних і обробку виняткових ситуацій.

Щоб почати створення програмного модуля, слід (у нашому випадку після символу присвоєння) натиснути на панелі **Programming** кнопку **Add Line**. Потім задати приблизну кількість рядків програми повторними натисканнями кнопки **Add Line**.

У першому рядку вводять "negative" і натискають кнопку **if** і т.д. Аналогічно у другому. У третьому натискають кнопку **otherwise**.

$$f(x) := \begin{cases} \text{"negative"} & \text{if } x < 0 \\ \text{"positive"} & \text{if } x > 0 \\ \text{"zero"} & \text{otherwise} \end{cases}$$

$$f(2) = \text{"positive"}$$

$$f(0) = \text{"zero"}$$

$$f(-5) = \text{"negative"}$$

Вставити рядок програмного коду можна у будь-якому місці вже створеної програми, наприклад:

$$f(x) := \begin{cases} \text{"negative"} & \text{if } x < 0 \\ \text{if } x > 0 \\ \quad \begin{cases} \text{"positive"} \\ \text{"big positive"} & \text{if } x > 1000 \end{cases} \\ \text{"zero"} & \text{otherwise} \end{cases}$$

$$f(2) = \text{"positive"}$$

$$f\{10^5\} = \text{"big positive"}$$

Ні оператор присвоєння :=, ні оператор чисельного виводу = у межах програм застосовувати не можна. Присвоєння у програмах Mathcad здійснюється за допомогою оператора **Local Definition**, який вставляється кнопкою стрілки ( $\leftarrow$ ) на панелі **Programming**.

$$x := \begin{cases} z \leftarrow 0 \\ \text{for } i \in 0..5 \\ \quad z \leftarrow z + i \end{cases} \quad \text{Fact}(n) := \begin{cases} (n \cdot \text{Fact}(n - 1)) & \text{if } n > 1 \\ 1 & \text{otherwise} \end{cases}$$

$$x = 15$$

$$\text{Fact}(5) = 120$$

## Тема 3. Символьні перетворення виразів у Mathcad

### 3.1. Загальні підходи до символьних обчислень

Символьні обчислення в Mathcad можна проводити двома способами:

- за допомогою команд меню;
- за допомогою оператора символьного виводу  $\rightarrow$ , ключових слів символьного процесора і звичайних формул (live symbolic evaluation).

Перший спосіб – меню **Symbolics** – використовується, коли обчислення потрібно провести 1 раз, без збереження ходу обчислень. Символьний процесор Mathcad вміє спрощувати вирази, розкласти їх на множники, символьно сумувати і множити. Аналітичні перетворення, що проводяться через меню, стосуються лише одного, виділеного в даний момент, виразу.

Другий спосіб більш наглядний, оск. дозволяє записувати вирази в традиційній математичній формі і зберігати в документах Mathcad. Використовується панель інструментів **Math**, зокрема ключові слова для символьних перетворень розміщені на панелі інструментів **Symbolic**. Наприклад, для таких операцій як розклад на множники, зведення подібних доданків та ін. в Mathcad не передбачено вбудованих функцій. Щоб звузити коло аналітичних перетворень, обмежившись дійсними або комплексними числами застосовують допоміжну панель **Modifier**.

### 3.2. Розклад виразів

#### 1 спосіб: меню **Symbolics / Expand**

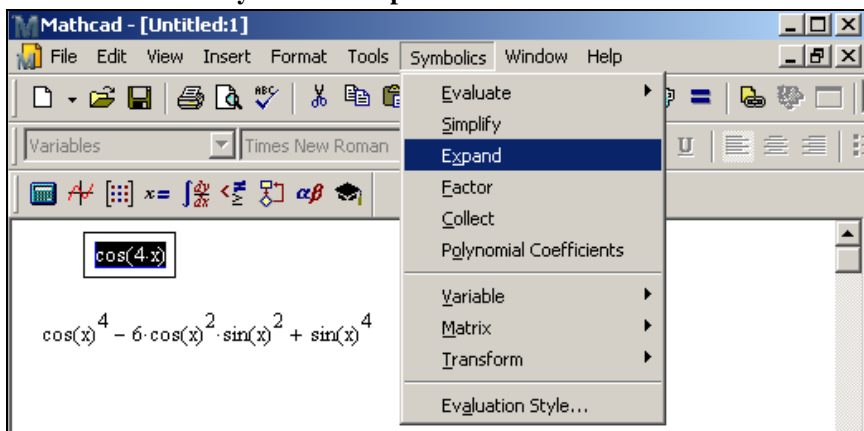
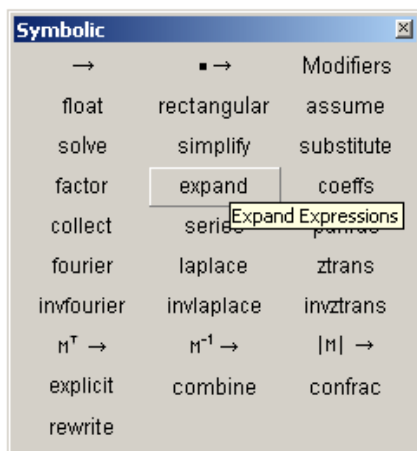


Рис. 4. Приклад розкладу виразу за допомогою меню Symbolics

**2 спосіб:** панель інструментів **Symbolic**, оператор **expand**.

$$\cos(4x) \text{ expand}, x \rightarrow \cos(x)^4 - 6 \cdot \cos(x)^2 \cdot \sin(x)^2 + \sin(x)^4$$



**Рис. 5.** Приклад розкладу виразу за допомогою панелі інструментів Symbolic

### 3.3. Об'єднання виразів

Зворотна операція до розкладу.

$$z^x \cdot z^{2y} \text{ combine} \rightarrow z^{x+2 \cdot y}$$

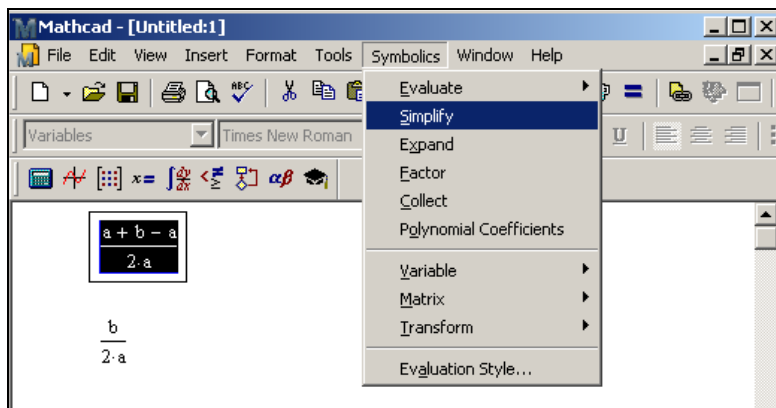
$$e^x \cdot e^{2y} \text{ combine, exp} \rightarrow e^{x+2 \cdot y}$$

$$\cos(x)^4 - 6 \cdot \cos(x)^2 \cdot \sin(x)^2 + \sin(x)^4 \text{ combine, sincos} \rightarrow \cos(4x)$$

$$\ln(x) + 4 \cdot \ln(2) + 7 \cdot \ln\left(\frac{1}{2}\right) \text{ combine, ln} \rightarrow \ln\left(\frac{x}{8}\right)$$

### 3.4. Спрощення виразів

**1 спосіб:** меню **Symbolics / Simplify**.



**Рис. 6.** Приклад спрощення виразу за допомогою меню Symbolics  
**2 спосіб:** панель інструментів **Symbolic**, оператор **simplify**.

$$\frac{a + b - a}{2a} \text{ simplify} \rightarrow \frac{b}{2 \cdot a}$$

$$\frac{a + b - a}{2a} \rightarrow \frac{b}{2 \cdot a}$$

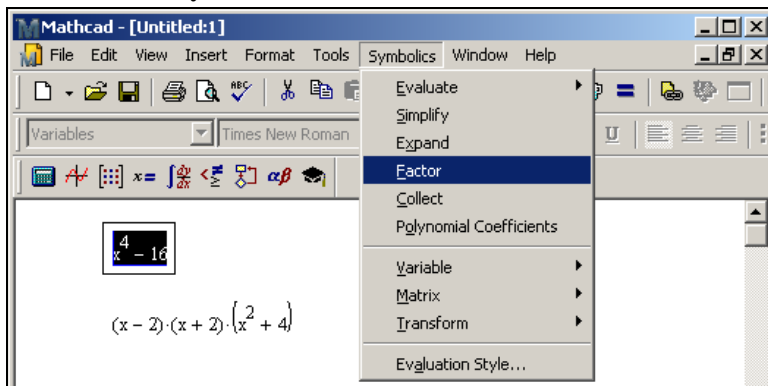
Спрощення виразів з числами:

$$\sqrt{3.01} \rightarrow 1.7349351572897472412$$

$$\text{acos}(0) \text{ simplify} \rightarrow \frac{\pi}{2}$$

### 3.5. Розклад на множники

**1 спосіб:** меню **Symbolics / Factor**



**Рис. 7.** Приклад розкладу виразу на множники за допомогою меню  
 Symbolics

**2 спосіб:** панель інструментів **Symbolic**, оператор **factor**.

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

$$28 \text{ factor} \rightarrow 2^2 \cdot 7$$

### 3.6. Розклад на елементарні функції

$$\cos(x) \text{ rewrite, exp} \rightarrow \frac{e^{-x \cdot i}}{2} + \frac{e^{x \cdot i}}{2}$$

$$\frac{e^{-x \cdot i}}{2} + \frac{e^{x \cdot i}}{2} \text{ rewrite, sincos} \rightarrow \cos(x)$$

### 3.7. Розклад числа на прості множники

Тест простих чисел:

$$\text{IsPrime}(191) \rightarrow 1$$

$$\text{IsPrime}(15) \rightarrow 0$$

Розклад цілого числа на прості множники

$$7654 \text{ factor} \rightarrow 2 \cdot 43 \cdot 89$$

$$191 \text{ factor} \rightarrow 191$$

### 3.8. Розклад на елементарні дроби

**1 спосіб:** меню **Symbolics / Variable / Convert to Partial Fractions**

**2 спосіб:** ключове слово **parfrac**

$$\frac{11 \cdot x^2 + 9x + 1}{x^2 - 3x + 2} \text{ convert, parfrac, x} \rightarrow \frac{63}{x - 2} - \frac{21}{x - 1} + 11$$

### 3.9. Розклад на неперервні дроби.

Ключове слово **confrac**

$$\frac{2 + \sqrt{5}}{19} \text{ confrac} \rightarrow \left( \begin{array}{c} 0 \\ 4 \\ 2 \\ 16 \\ 2 \\ 84 \\ 2 \end{array} \right)$$



$$\frac{2 + \sqrt{5}}{19} \text{ confrac, fraction} \rightarrow \left[ 4 + \frac{1}{2 + \frac{1}{16 + \frac{1}{2 + \frac{1}{\left(84 + \frac{1}{2}\right)}}}} \right]$$

### 3.10. Зведення подібних доданків

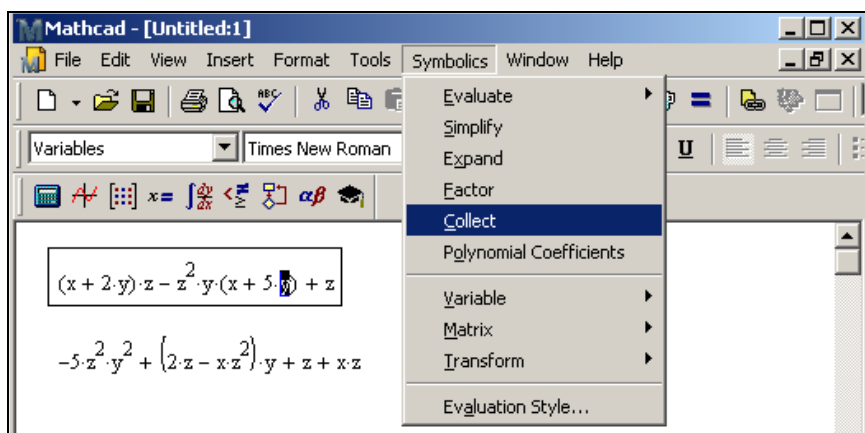
**1 спосіб:** меню **Symbolics / Collect**

**2 спосіб:** ключове слово **collect**

$$(x + 2y) \cdot z - z^2 \cdot y \cdot (x + 5y) + z \text{ collect, } x \rightarrow (z - y \cdot z^2) \cdot x + 2 \cdot y \cdot z - 5 \cdot y^2 \cdot z^2 + z$$

$$(x + 2y) \cdot z - z^2 \cdot y \cdot (x + 5y) + z \text{ collect, } y \rightarrow -5 \cdot z^2 \cdot y^2 + (2 \cdot z - x \cdot z^2) \cdot y + z + x \cdot z$$

$$(x + 2y) \cdot z - z^2 \cdot y \cdot (x + 5y) + z \text{ collect, } x, y, z \rightarrow \left[ (-z^2) \cdot y + z \right] \cdot x + -5 \cdot z^2 \cdot y^2 + 2 \cdot z \cdot y + z$$



**Рис. 8.** Приклад зведення подібних доданків за допомогою меню **Symbolics**

### 3.11. Обчислення коефіцієнтів полінома.

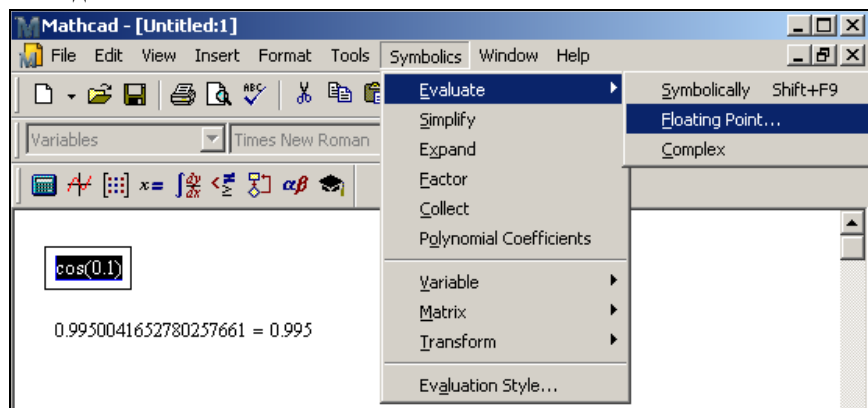
Якщо вираз є поліномом відносно деякої змінної  $x$ , але він заданий не у вигляді полінома  $a_0 + a_1x + a_2x^2 + \dots$ , а як добуток інших більш простих поліномів, то коефіцієнти  $a_0, a_1, a_2, \dots$  легко визначаються за допомогою меню **Symbolic / Polynomial Coefficients** або ключового слова **coeffs**.

$$(x + 2 \cdot y) \cdot z - z^2 \cdot y \cdot (x + 5y) + z \text{ coeffs}, z \rightarrow \begin{pmatrix} 0 \\ x + 2 \cdot y + 1 \\ -5 \cdot y^2 - x \cdot y \end{pmatrix}$$

$$(x - 4) \cdot (x - 7) \cdot x \text{ coeffs}, z \rightarrow \{x^3 - 11 \cdot x^2 + 28 \cdot x\}$$

### 3.12. Отримання числового значення виразу

За допомогою символічного процесора можна розрахувати числове значення виразу (дійсне чи комплексне). Для цього вибирають меню **Symbolics / Evaluate / Symbolically** або пункт **Symbolics / Evaluate / Floating Point** (з плаваючою крапкою). У другому випадку відкривається діалогове вікно **Floating Point Evaluation**, де задають точність виводу. Пункт меню **Symbolics / Evaluate / Complex** дозволяє представити вираз у вигляді  $a+bi$ .



**Рис. 9.** Приклад отримання числового значення виразу за допомогою меню Symbolics

Аналогічно діють ключові слова **float** і **complex** з панелі **Symbolic**.

Обчислення виразу з плаваючою крапкою:

$$\begin{aligned}
 x &:= 3 & k &:= 2.4 \\
 \cos(k \cdot x) + 4 \cdot x^{2-k} \text{ float}, 3 &\rightarrow 3.19 \\
 \cos(k \cdot x) + 4 \cdot x^{2-k} \text{ float}, 10 &\rightarrow 3.185927374 \\
 \cos(k \cdot x) + 4 \cdot x^{2-k} \text{ float}, 20 &\rightarrow 3.185927374441271673
 \end{aligned}$$

Аналогічно проводяться комплексні перетворення виразів (ключове слово – complex)/

### 3.13. Явні обчислення

Терміном явних (explicit) символьних обчислень позначають розрахунки, в яких відбувається проста підстановка у вираз чисельного значення змінної (без спрощення). Для цього використовують ключове слово **explicit** з с панелі **Symbolic**.

$$x := 2 \cdot m \quad y := 0.5 \cdot \text{sec}$$

$$z := 1.5 \cdot \frac{m}{\text{sec}}$$

$$\frac{x}{y} + z \text{ explicit}, x \rightarrow \frac{2 \cdot m}{y} + z$$

$$\frac{x}{y} + z \text{ explicit}, x, y, z \rightarrow \frac{2 \cdot m}{(0.5 \cdot \text{sec})} + 1.5 \cdot \left( \frac{m}{\text{sec}} \right)$$

### 3.14. Обчислення рядів

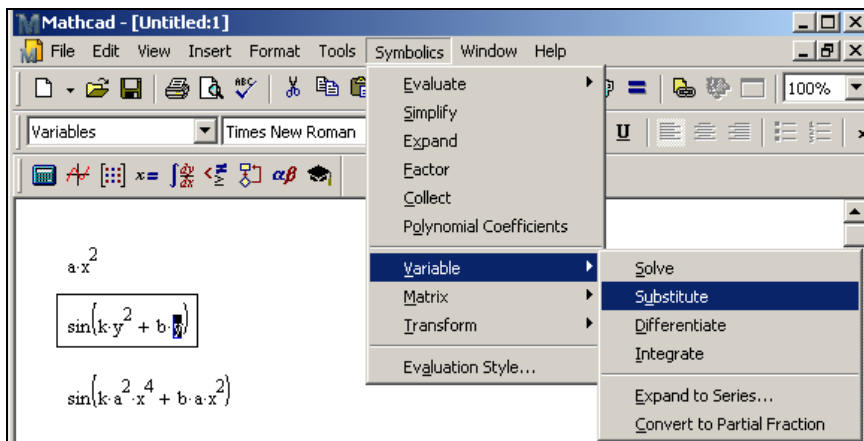
Для обчислення суми (добутку) скінченного чи нескінченного ряду використовують панель **Calculus**.

$$\sum_{n=0}^{\infty} \frac{x^n}{2^n \cdot n!} \rightarrow e^{\frac{x}{2}}$$

$$\sum_{n=0}^{100} \frac{1^n}{2^n \cdot n!} = 1.649$$

### 3.15. Підстановка значення змінної у вираз

**1 спосіб:** за допомогою меню **Symbolics**



**Рис. 10.** Приклад підстановки значення змінної у вираз за допомогою меню Symbolics

- 1) спочатку потрібно виділити і скопіювати у буфер вираз чи значення, що підставляється ( $ax^2$ );
- 2) потім в основному виразі виділити змінну, яка буде замінюватись ( $y$ );
- 3) виконати команду **Symbolics / Variable / Substitute**

**2 спосіб:** ключове слово **substitute**.

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

## Тема 4. Диференціювання

Операція диференціювання реалізована в Mathcad як в чисельній, так і в аналітичній формі і позначається операторами, записаними в традиційній математичній формі.

### 4.1. Аналітичне диференціювання

**1 спосіб:** за допомогою панелі інструментів **Calculus**.

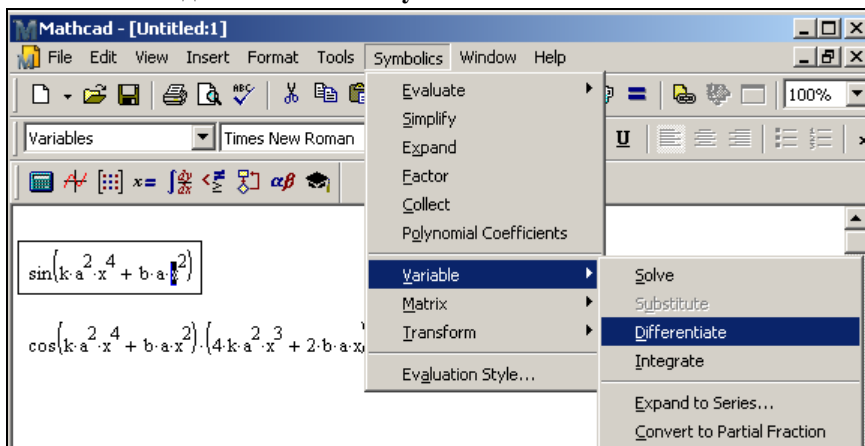
$$f(x) := x^4 + 2 \cdot x^3 - 7 \cdot x^2 + 3x - 1$$

$$g(x) := \frac{d}{dx} f(x)$$

$$g(x) \rightarrow 4 \cdot x^3 + 6 \cdot x^2 - 14 \cdot x + 3$$

$$g(1) \rightarrow -1$$

2 спосіб: за допомогою меню **Symbolics / Variable / Differentiate**



**Рис. 11.** Приклад диференціювання виразу за вказаною змінною за допомогою меню Symbolics

Для **чисельного диференціювання** Mathcad застосовує досить складний алгоритм (метод Ріддера) з точністю до 7—8-го знаку після коми. Похибка в цьому вип. не залежить від констант TOL чи CTOL.

#### 4.2. Частинні похідні

Похідні від функції багатьох змінних завдаються аналогічно звичайним:

$$f(x, k) := k \cdot \sin(x)$$

$$\frac{\partial}{\partial x} f(x, k) \rightarrow k \cdot \cos(x)$$

$$\frac{\partial}{\partial k} f(x, k) \rightarrow \sin(x)$$

Розрізняють символічне і чисельне обчислення частинних похідних.

$$f(x, k) := k \cdot \sin(x)$$

$$x := 10$$

$$\frac{\partial}{\partial x} f(x, k) \rightarrow k \cdot \cos(10)$$

$$k := 1$$

$$\frac{\partial}{\partial x} f(x, k) = -0.839$$

Розглянемо приклади обчислення другої частинної похідної.

$$f(x, y) := y^2 \cdot x^3 + y \cdot x^2$$

$$\frac{d^2}{dx^2} f(x, y) \rightarrow 6 \cdot x \cdot y^2 + 2 \cdot y$$

$$\frac{d^2}{dy^2} f(x, y) \rightarrow 2 \cdot x^3$$

$$\frac{\partial}{\partial x} \left( \frac{d}{dy} f(x, y) \right) \rightarrow 6 \cdot y \cdot x^2 + 2 \cdot x$$

Хоч це нічого по суті не міняє, можна змінити вигляд оператора диференціювання на представлення частинних похідних. Для цього потрібно: у контекстному меню області оператора диференціювання вибрати **View Derivative As**, а потім у підменю вибрати пункт **Partial Derivative**.

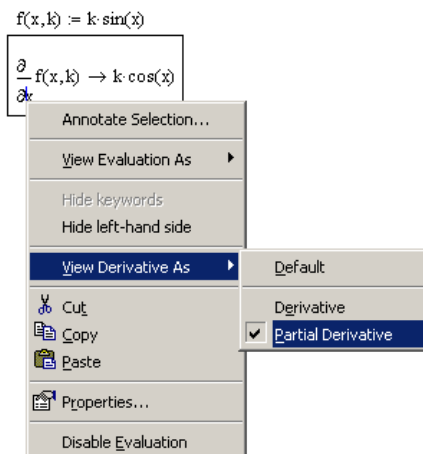


Рис. 12. Частинні похідні в Mathcad

### 4.3. Обчислення вектора градієнта функції

Здійснюється за допомогою наступного оператора з панелі Calculus:

$$f(x, y) := 0.12 \cdot x^2 + y \cdot x - 0.01 \cdot y^4$$

$$\nabla_{x, y} f(x, y) \rightarrow \begin{pmatrix} y + 0.24 \cdot x \\ x + -0.04 \cdot y^3 \end{pmatrix}$$

Обчислення вектора градієнта функції двох змінних “вручну”:

$$\text{grad}f(x, y) := \begin{pmatrix} \frac{d}{dx} f(x, y) \\ \frac{d}{dy} f(x, y) \end{pmatrix}$$

$$\text{grad}f(x, y) \rightarrow \begin{pmatrix} y + 0.24 \cdot x \\ x + -0.04 \cdot y^3 \end{pmatrix}$$

### 4.4. Розклад функції в ряд Тейлора

**1 спосіб:** за допомогою меню **Symbolics / Variable / Expand to Series**

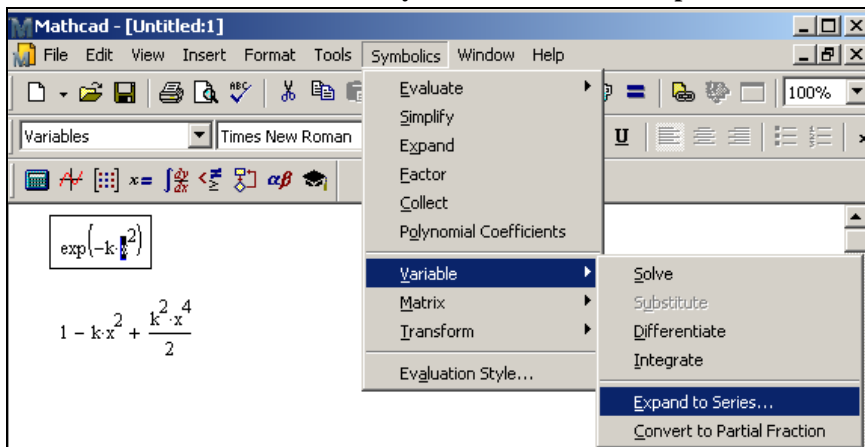


Рис. 13. Розклад функції в ряд Тейлора в Mathcad

**2 спосіб:** ключове слово **series**.

Розглянемо приклади розкладу функції кількох змінних в ряд по різним змінним:

$$f(x, k) := \exp\{-k \cdot x^2\}$$

$$f(x, k) \text{ series, } k, 4 \rightarrow 1 - k \cdot x^2 + \frac{k^2 \cdot x^4}{2} - \frac{k^3 \cdot x^6}{6}$$

$$f(x, k) \text{ series, } x, 4 \rightarrow 1 - k \cdot x^2$$

## Тема 5. Інтегрування. Інтегральні перетворення

З одного боку, чисельне інтегрування – одна з самих простих, з обчислювальної точки зору, операцій, а з іншої – аналітично проінтегрувати можна далеко не кожену функцію.

### 5.1. Чисельне інтегрування

Розглянемо приклади чисельного і символьного обчислення визначеного інтегралу:

$$\int_0^{\pi} \exp\{-x^2\} dx = 0.886$$

$$\int_0^{\pi} \exp\{-x^2\} dx \rightarrow \frac{\sqrt{\pi} \cdot \text{erf}(\pi)}{2} = 0.886$$

Обчислимо інтеграл з нескінченними межами:

$$\int_{-\infty}^{\infty} \exp\{-x^2\} dx \rightarrow \sqrt{\pi}$$

Інтегрування функції двох змінних по різних змінним:

$$\int_a^b \exp\{-x \cdot z^2\} dx \rightarrow \frac{e^{-a \cdot z^2} - e^{-b \cdot z^2}}{z^2} = \frac{e^{-a \cdot z^2}}{z^2} - \frac{e^{-b \cdot z^2}}{z^2}$$

Вираз виводиться червоним кольором, тому що значення змінних  $a$  і  $b$  не задане.

В результаті чисельного інтегрування отримуємо наближене значення в межах точності, заданої вбудованою константою TOL. По замовчужанню TOL=0.001. Це значення можна змінити. Крім цього, користувач має можливість вибирати сам алгоритм чисельного інтегрування. Для цього використовують контекстне меню, клацанням по лівій частині інтеграла, і вибирають один із чотирьох алгоритмів:



- *Метод Ромберга (Romberg)* – для більшості функцій, без особливостей;
- *Адаптивний метод (Adaptive)* – для функцій, що швидко змінюються на інтервалі інтегрування;
- *Границя в нескінченності (Infinite Limit)* – для інтегралів с нескінченними границями;
- *Особлива межа (Singular Endpoint)* – для інтегралів із сингулярністю на кінці (застосовується модифікований алгоритм Ромберга для функцій, не визначених на одному чи обох кінцях інтервалу інтегрування).

## 5.2. Символьне інтегрування

### 1 спосіб: за допомогою панелі інструментів Calculus

Аналітичне обчислення невизначеного інтегралу:

$$\int \exp(-a \cdot x^2) dx \rightarrow \frac{\sqrt{\pi} \cdot \operatorname{erf}(\sqrt{a} \cdot x)}{2 \cdot \sqrt{a}}$$

### 2 спосіб: за допомогою меню Symbolics / Variable / Integrate:

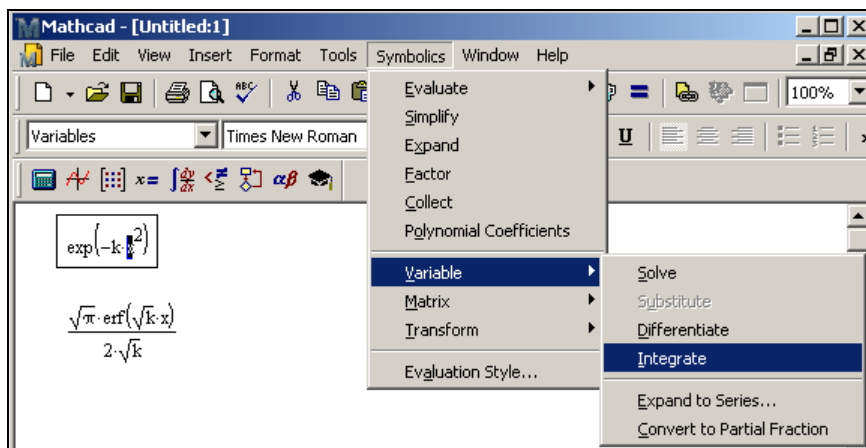


Рис. 14. Приклад інтегрування функції у системі Mathcad

Подивимося, що виведе система при обчисленні розбіжного інтегралу:

$$\int_0^{\infty} \frac{1}{\sqrt{x}} dx \rightarrow \infty$$

$$\int_0^{\infty} \frac{1}{\sqrt{x}} dx =$$

Виконаємо аналітичне обчислення інтегралу із змінною верхньою межею:

$$\int_0^z \frac{1}{\sqrt{x}} dx \rightarrow 2 \cdot \sqrt{z}$$

Розглянемо приклади символічного і чисельного обчислення кратних інтегралів:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x^2+y^2)} dx dy \rightarrow \pi$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x^2+y^2)} dx dy = 3.142$$

$$\int_a^b \int_{-1}^1 x + y^3 dx dy \rightarrow \frac{b^4}{2} - \frac{a^4}{2}$$

$$\int_a^b \int_{-1}^1 x + y^3 dy dx \rightarrow b^2 - a^2$$

### 5.3. Інтегральні перетворення в Mathcad

Існує багато класів проблем, що складно розв'язуються алгебраїчно, чи занадто громіздкі в своїх оригінальних завданнях. Інтегральне перетворення відображає рівняння з його оригінального "домену" (наприклад, функції де час є незалежною змінною) в інший домен. Потім отриманий розв'язок повертають назад в оригінальний домен за допомогою оберненого інтегрального перетворення.

Інтегральні перетворення працюють бо вони базуються на концепції спектральної факторизації над ортонормованим базисом. Це означає, що багато важливих, складних функцій можуть бути представлені як суми набагато простіших функцій.

Символьний процесор Mathcad дозволяє здійснювати три види інтегральних перетворень функцій – перетворення Фур'є, Лапласа і Z-перетворення.

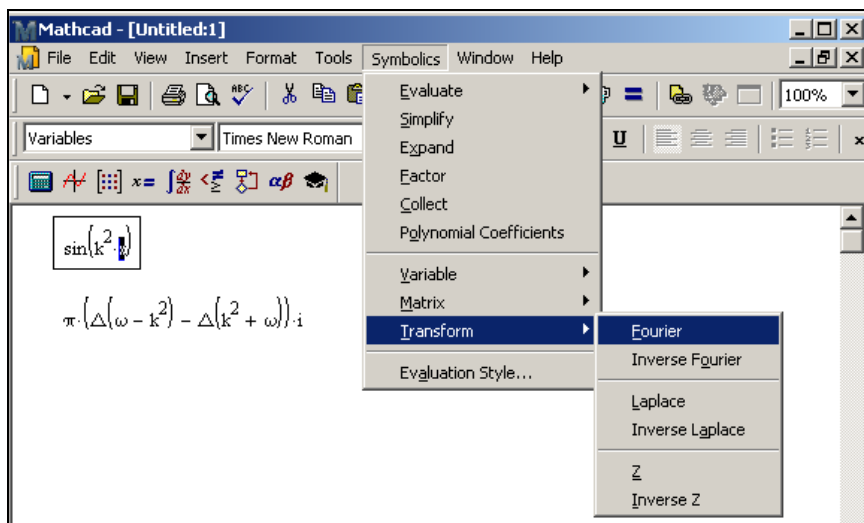
#### 5.4. Перетворення (інтеграл) Фур'є

Найбільш широко розповсюдженим інтегральним перетворенням є перетворення Фур'є, яке представляє функцію  $f(x)$  у вигляді інтеграла по гармонічним функціям, що називається інтегралом Фур'є:

$$F(\omega) = \int_{-\infty}^{\infty} f(x) \cdot \exp(-i\omega x) dx$$

Функція  $F(\omega)$  наз. також перетворенням Фур'є, або Фур'є-спектром функції  $f(x)$ . Її аргумент має зміст частоти відповідної гармонічної складової  $f(x)$ . Дана функція комплексна, навіть якщо  $f(x)$  дійсна.

**1 спосіб:** за допомогою меню **Symbolics / Transform / Fourier**



**Рис. 15.** Приклад прямого перетворення Фур'є з використанням меню Symbolics

## 2 спосіб: ключове слово **fouriar**.

Розглянемо приклад прямого перетворення Фур'є, реалізованого за допомогою панелі Symbolic:

$$\cos\{k^2 \cdot x\} \text{ fouriar, } x \rightarrow \pi \cdot \left( \Delta(\omega - k^2) + \Delta(k^2 + \omega) \right)$$

Аналогічно здійснюється обернене перетворення Фур'є:

$$\frac{1}{\omega} \text{ invfourier, } \omega \rightarrow -\Phi(t) \cdot i + \frac{1}{2} \cdot i$$

$$-\Phi(t) \cdot i + \frac{1}{2} \cdot i \text{ fourier} \rightarrow \frac{1}{\omega}$$

У багатьох прикладних задачах розраховується інтеграл Фур'є для функцій, або заданих таблично або функцій, проінтегрувати які аналітично не вдається. У даному випадку замість символівних перетворень приходится застосовувати чисельні методи інтегрування і дискретні Фур'є-перетворення.

У Mathcad воно реалізовано за доп. популярного алгоритму швидкого перетворення Фур'є, що представлений кількома вбудованими функціями, що відрізняються лише нормуванням:

- **fft(y)** – вектор прямого перетворення Фур'є;
- **FFT(y)** – вектор прямого перетворення Фур'є у другому нормуванні;
- **ifft(ω)** – вектор оберненого перетворення Фур'є;
- **IFFT(ω)** – вектор оберненого перетворення Фур'є у другому нормуванні.

Тут  $y$  – вектор дійсних значень, взятих через рівні проміжки значень аргументу;

$\omega$  – вектор дійсних значень Фур'є-спектру, взятих через рівні проміжки значень частоти.

Нижче наведено приклад дискретного перетворення Фур'є (алгоритм ШПФ – швидкого перетворення Фур'є) модельного сигналу

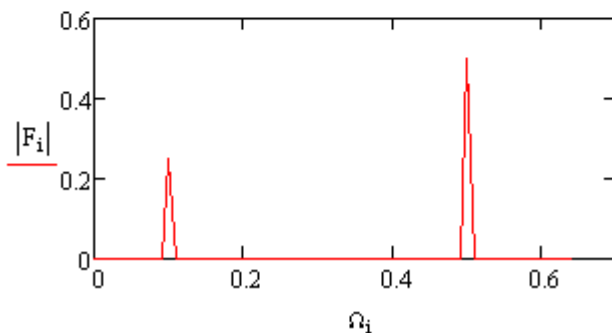
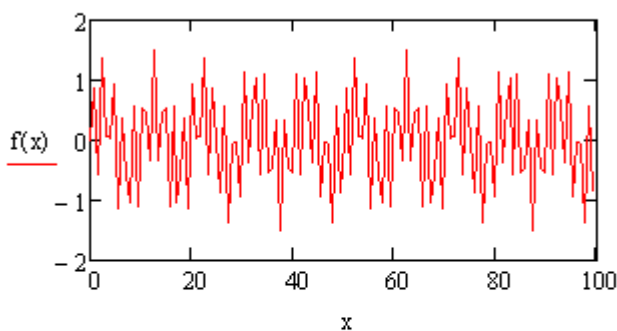
$$f(x) := 0.5 \cdot \sin(2 \cdot \pi \cdot 0.1x) + 1 \cdot \sin(2 \cdot \pi \cdot 0.5x)$$

$$\underline{L} := 100 \quad \underline{N} := 128 \quad h := \frac{L}{N} \quad h = 0.781$$

$$i := 0..N - 1$$

$$x_i := i \cdot h \quad y_i := f(x_i) \quad \underline{\Omega}_i := \frac{i}{L}$$

$$\underline{F}_i := \text{FFT}(y)$$



**Рис. 16.** Приклад реалізації дискретного перетворення Фур'є модельного сигналу у Mathcad

Алгоритм швидкого перетворення Фур'є для комплексних даних вбудовано в аналогічні функції, назви яких починаються з літери "с":  $\text{cff}(y)$ ,  $\text{CFFT}(y)$  і т.д.

Двовимірне перетворення Фур'є матиме, наприклад, наступний вигляд:

$$N := 64$$

$$i := 0..N - 1$$

$$j := 0..N - 1$$

$$y_{i,j} := \sin\left(\frac{i+j}{10}\right) + \sin\left(\frac{i-j}{10}\right)$$

$$F := \text{CFFT}(y)$$

$$F := \text{submatrix}(F, 7, N - 7, 7, N - 7)$$

### 5.5. Перетворення Лапласа

Описується формулою:

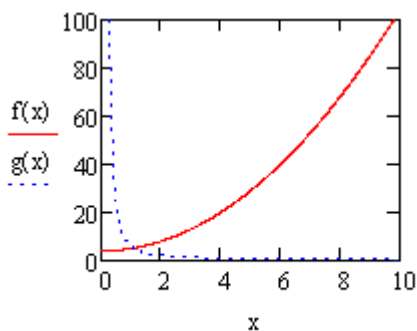
$$F(s) = \int_0^{\infty} f(x) \cdot \exp(-sx) dx.$$

Розраховується аналогічно перетворенню Фур'є.

$$f(x) := x^2 + 4$$

$$g(s) := f(x) \text{ laplace, } x \rightarrow \frac{4s^2 + 2}{s^3}$$

$$g(s) \text{ invlaplace, } s \rightarrow t^2 + 4$$



**Рис. 17.** Приклад реалізації прямого і оберненого перетворення Лапласа у системі Mathcad

## 5.6. Z-перетворення

Має наступний вигляд:

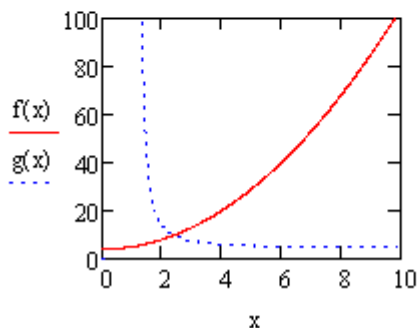
$$F(z) = \sum_{n=0}^{\infty} f(n) \cdot z^{-n}.$$

Приклад прямого та оберненого Z-перетворення, реалізованого засобами Mathcad, показано на наступному рисунку.

$$f(x) := x^2 + 4$$

$$g(z) := f(x) \text{ ztrans, } x \rightarrow \frac{z \cdot (4z^2 - 7z + 5)}{(z - 1)^3}$$

$$g(z) \text{ invztrans, } z \rightarrow 3 \cdot n + 2 \cdot \text{combin}(n - 1, 2) + 2$$



**Рис. 18.** Приклад реалізації прямого і оберненого Z-перетворення у системі Mathcad

## 5.7. Вейвлет-перетворення

Інша назва – дискретне хвильове перетворення для неперервного сигналу відносно вейвлет функції, визначається наступним чином:

$$[W_{\psi} f](a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} \overline{\psi\left(\frac{x-b}{a}\right)} f(x) dx$$

Тут  $\overline{\psi\left(\frac{x-b}{a}\right)}$  означає комплексне спряження для функції  $\psi$ , параметр  $b \in \mathbb{R}$  відповідає часовому зсуву і називається параметром положення, параметр

$a > 0$  задає масштабування і називається параметром розтягу. Тут

$$w(a) \equiv \frac{1}{\sqrt{a}} - \text{вагова функція.}$$

Застосовується для аналізу нестационарних сигналів і у багатьох випадках виявляється більш ефективним, ніж перетворення Фур'є. Основна відмінність – розклад не по синусоїдам (як Фур'є), а по іншим, так званим вейвлетоутворюючим функціям. На відміну від безкінечно осцилюючих синусоїд, вони локалізовані в деякій обмеженій області свого аргументу, а в даліні від неї рівні нулю або нескінченно малі. Приклад такої функції – функція "мексиканський капелюх" (див. рис. 19).

Крім частоти, другим аргументом  $b$  є місце локалізації вейвлетоутворюючої функції. Тому  $b$  має ту ж розмірність, що і  $x$ .

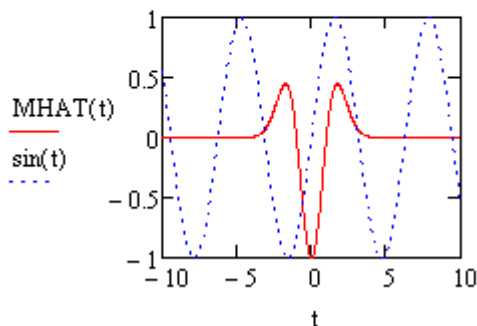
Mathcad має одну вбудовану функцію для розрахунку вейвлет-перетворення на основі вейвлетоутворюючої функції Добеши:

- **wave(y)** – вектор прямого вейвлет-перетворення Добеши;
- **iwave(v)** – вектор оберненого вейвлет-перетворення Добеши;

Тут  $y$  – вектор даних, взятих через рівні проміжки значень аргументу;

$v$  – вектор даних вейвлет-спектру.

$$\text{MHAT}(t) := \frac{d^2}{dt^2} e^{-\frac{t^2}{2}} \rightarrow t^2 \cdot e^{-\frac{t^2}{2}} - e^{-\frac{1}{2} \cdot t^2}$$



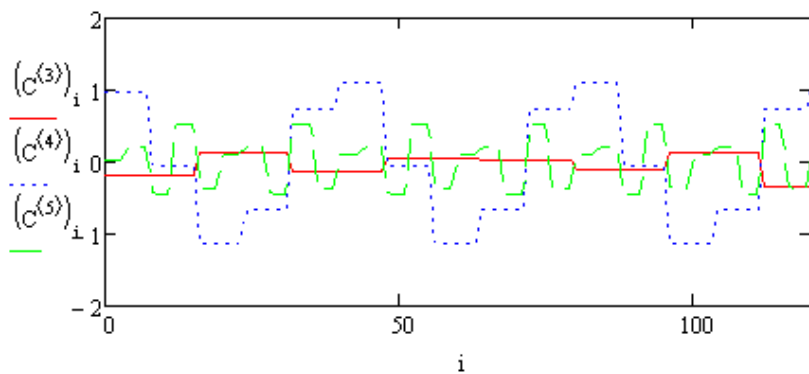
**Рис. 19.** Функція "мексиканський капелюх"



```

f(t) := 0.5·sin(2π·0.1·t) + 1·sin(2π·0.5·t)
N := 128
i := 0..N - 1
Yi := f(i)
W := wave(Y)
coeffs(level) := submatrix(W, 2level, 2level+1 - 1, 0, 0)
Nlevels :=  $\frac{\ln(N)}{\ln(2)}$  - 1      Nlevels = 6
k := 1..Nlevels
Ci,k := coeffs(k)
           floor( $\frac{i \cdot 2^k}{N}$ )

```



**Рис. 20.** Приклад реалізації вейвлет-перетворення у системі Mathcad

## Тема 6. Чисельне і символічне розв'язання нелінійних алгебраїчних рівнянь та їх систем

**6.1. Аналітичне розв'язання рівнянь та систем рівнянь.** Аналітичне розв'язання рівнянь здійснюється кількома способами.

**1 спосіб:** за допомогою блоку Given ...Find()

Given

$$3x^3 + 2x^2 - 7x = 0$$

$$\text{Find}(x) \rightarrow \left( \frac{\sqrt{22}}{3} - \frac{1}{3} \quad -\frac{\sqrt{22}}{3} - \frac{1}{3} \quad 0 \right) = (1.23 \quad -1.897 \quad 0)$$

Розглянемо приклад символічного розв'язання алгебраїчних рівнянь відносно різних змінних:

Given

$$ax^3 + bx^2 - cx = 0$$

$$\text{Find}(x) \rightarrow \left( -\frac{b - \sqrt{b^2 + 4ac}}{2a} \quad -\frac{b + \sqrt{b^2 + 4ac}}{2a} \quad 0 \right) = \left( -\frac{b - \sqrt{b^2 + 4ac}}{2a} \quad -\frac{b + \sqrt{b^2 + 4ac}}{2a} \quad 0 \right)$$

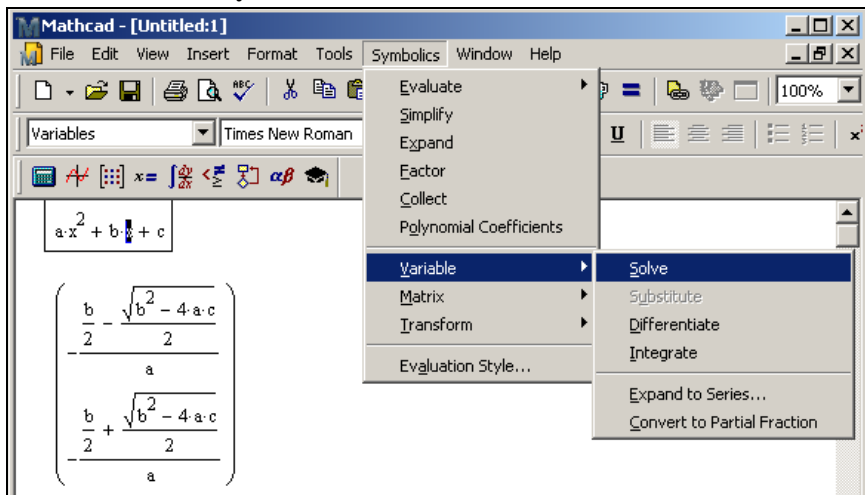
**2 спосіб:** ключове слово **solve**

$$(y \cdot x - 1) \cdot x + 1 \text{ solve, } x \rightarrow \begin{pmatrix} \frac{\sqrt{1-4y}}{2} + \frac{1}{2} \\ y \\ \frac{\sqrt{1-4y}}{2} - \frac{1}{2} \\ -\frac{1}{y} \end{pmatrix}$$

Аналітичний пошук всіх коренів рівняння здійснюється командою:

$$(y \cdot x - 1) \cdot x + 1 \text{ solve, } x, \text{fully} \rightarrow \begin{cases} 1 & \text{if } y = 0 \\ \begin{pmatrix} \frac{\sqrt{1-4y}}{2} + \frac{1}{2} \\ y \\ \frac{\sqrt{1-4y}}{2} - \frac{1}{2} \\ -\frac{1}{y} \end{pmatrix} & \text{if } y \neq 0 \end{cases}$$

### 3 спосіб: меню Symbolics/Variable/Solve



**Рис. 21.** Приклад розв'язання алгебраїчного рівняння з використанням меню Symbolics

Аналітичне розв'язання систем рівнянь також здійснюється за допомогою блоку Given ...Find().

Given

$$-x^2 + y + 1 = 0$$

$$x = 2y + 1$$

$$\text{Find}(x, y) \rightarrow \begin{pmatrix} 1 & -\frac{1}{2} \\ 0 & -\frac{3}{4} \end{pmatrix}$$

**6.2. Чисельне розв'язання рівнянь та систем рівнянь.** Як правило, розв'язання цієї задачі чисельними методами пов'язане з двома задачами:

- локалізація коренів, тобто виявлення їх існування в принципі, а також дослідження їх кількості і приблизного розміщення;
- власне відшукування коренів із заданою похибкою (в Mathcad її роль грає системна константа TOL).

Спочатку розглянемо другу задачу.

**1 спосіб:** за допомогою блоку Given ...Find().

Результатом чисельного розв'язання рівняння є один його корінь. Щоб отримати інші корені, потрібно задати інші початкові значення невідомої.

$$x := 1$$

Given

$$3 \cdot x^3 + 2 \cdot x^2 - 7x = 0$$

$$\text{Find}(x) = 1.23$$

Чисельне розв'язання систем рівнянь також приводить до одного з коренів.

$$x := 10 \quad y := 10$$

Given

$$-x^2 + y + 1 = 0$$

$$x = 2y + 1$$

$$\text{Find}(x, y) = \begin{pmatrix} 1 \\ -3.884 \times 10^{-10} \end{pmatrix}$$

Чисельне розв'язання рівняння, що має нескінченне число коренів, приводить до одного з них:

$$x := 10 \quad y := 10$$

Given

$$-x^2 + y + 1 = 0$$

$$\text{Find}(x, y) = \begin{pmatrix} 3.317 \\ 10 \end{pmatrix}$$

## 2 спосіб: функція **root()**

Може мати або два, або чотири аргументи:

- **root(f(x),x);**
- **root(f(x),x,a,b).**

Тут  $f(x)$  – скалярна функція, що визначає рівняння  $f(x)=0$ ;

$x$  – ім'я скалярної змінної, відносно якої розв'язується рівняння;

$a, b$  – границі інтервалу, всередині якого шукають корені.

Перший тип функції **root()** аналогічно **Find()** вимагає задати початкове значення  $x$  для уточнення приблизної локалізації кореня.

Розглянемо два варіанти розв'язання одного рівняння методом січних:

$$\text{root}(e^x - 1, x, -1, 1) = 0$$

$$x := 1$$

$$\text{root}(e^x - 1, x) = 2.174 \times 10^{-13}$$

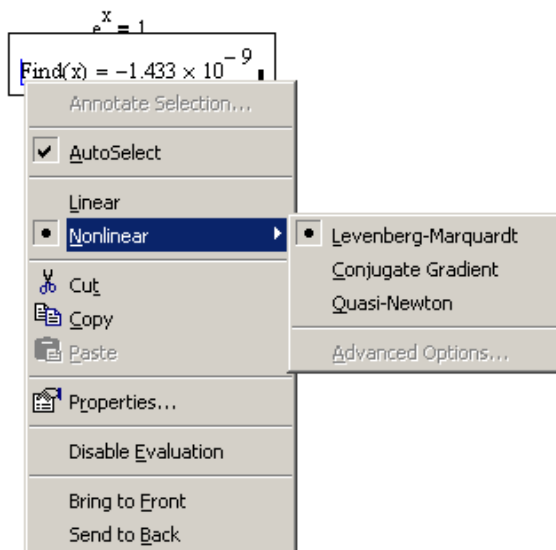
У чому ж відмінність вбудованої функції **Find()** від функції **root()**? Для розв'язання одних і тих же задач використовуються різні чисельні алгоритми (градієнтні методи і метод січних відповідно).

Mathcad пропонує три різних варіанти градієнтних методів. Щоб змінити чисельний метод, потрібно:

1. Клацнути правою кнопкою миші по назві функції **Find**.
2. Вибрати пункт **Nonlinear** у контекстному меню.
3. У підменю вибрати один з трьох методів:
  - *Метод спряження градієнтів (Conjugate Gradient)*;
  - *Метод квазі-Ньютона (Quasi-Newton)*;
  - *Метод Левенберга-Марквардта (Levenberg-Marquardt)*.

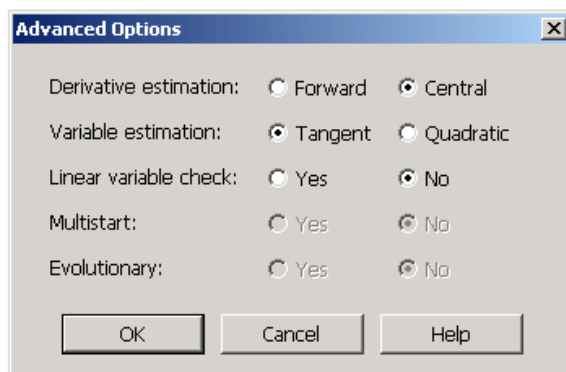
$$x := 1$$

Given



**Рис. 22.** Приклад чисельного розв'язання алгебраїчного рівняння з використанням блоку Given ...Find()

Крім того, для другого і третього методів є можливість встановлювати їх деякі параметри. Для цього у тому ж контекстному меню вибирають пункти Нелінійний / Додаткові параметри (**Nonlinear / Advanced options**). У діалоговому вікні є п'ять груп перемикачів по два у кожній.



**Рис. 23.** Вікно завдання додаткових параметрів для розв'язання нелінійних алгебраїчних рівнянь

1-й рядок – *Апроксимація похідної (Derivative estimation)* – методи обчислення похідної Вперед (**Forward**) або Центральна (**Central**). Вони відповідають апроксимації похідної або правою (двоточковою схемою "вперед"), або центральною (треточковою симетричною схемою) скінченною різницею.

2-й рядок – *Апроксимація змінних (Variable estimation)* – можна задати тип апроксимації рядом Тейлора. Для апроксимації дотичною (прямою лінією) вибирають Дотична (**Tangent**), для більш точної квадратичної апроксимації (параболою) вибирають Квадратична (**Quadratic**).

3-й рядок – *Перевірка лінійності (Linear variable check)* – дозволяє в специфічних задачах зекономити час обчислень. Якщо встановити перемикач **Yes**, то похідні будуть прийняті рівними константам і не будуть обчислюватися на кожному кроці.

4-й рядок – пара перемикачів *Сканування (Multistart)* задає опцію пошуку глобального чи локального екстремуму. Якщо виставлений перемикач **Yes**, Mathcad буде намагатися знайти найбільш глибокий екстремум з області, близької до початкового наближення.

5-й рядок – пара перемикачів *Еволюційний алгоритм (Evolutionary)*, якщо встановити **Yes**, то буде використовуватись модифікація чисельного

методу для розв'язання рівнянь, заданих не обов'язково гладкими функціями.

Якщо функція  $f(x)$  є поліномом, то всі його корені можна знайти за доп. вбудованої функції: **polyroots(v)**, де  $v$  – вектор із коефіцієнтів полінома.

$$v := \begin{pmatrix} 3 \\ -10 \\ 12 \\ -6 \\ 1 \end{pmatrix} \quad \text{polyroots}(v) = \begin{pmatrix} 0.992 \\ 1.004 + 7.177i \times 10^{-3} \\ 1.004 - 7.177i \times 10^{-3} \\ 3 \end{pmatrix}$$

Для функції **polyroots()** можна вибрати один з двох чисельних методів – метод поліномів Лаггера (по замовчуванню) або метод парної матриці.

Для зміни методу:

1. Викликають контекстне меню, правою кнопкою миші на слові **polyroots**.

2. У верхній частині контекстного меню виберіть або пункт **LaGuerre**, або **Companion Matrix**.

### 6.3. Локалізація коренів

У самих простих випадках можна використати графічне представлення функцій. У випадках функцій від багатьох змінних це практично не можливо. Для дослідження певної ділянки області визначення змінних рівняння на наявність коренів із визначенням їх приблизного розміщення, можна використовувати досить затратний спосіб – *сканування*. Воно полягає у послідовному пошуку кореня, починаючи з множини пробних точок, що покривають розрахункову область.

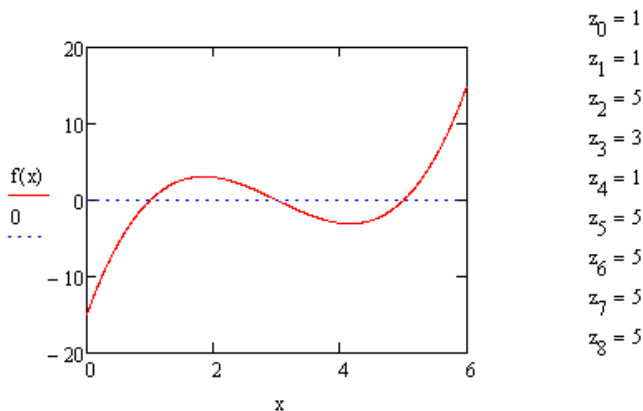
Зазвичай область визначення функції розбивається на елементарні області (прямокутники, куби і т.д.). Із центра кожної елементарної області запускається чисельний метод пошуку кореня, і, у випадку виходу ітерацій за її межі, розрахунки перериваються, а у протилежному випадку знаходять корінь.

Менш надійний (зате більш економний) спосіб – просте обчислення і порівняння між собою нев'язок системи рівнянь у центральних точках елементарних областей.

На тих ділянках області визначення, де норма нев'язки невелика, ймовірність локалізації кореня більша, а саме, з локальних мінімумів норми нев'язки можна запускати градієнтний метод для уточнення кореня.

Розглянемо приклад організації спрощеного варіанта сканування по одній змінній. Спочатку будується графік функції. Потім здійснюється розв'язання рівняння за допомогою функції **root()** для кількох послідовно розміщених вузлів. Видно, що на даному інтервалі рівняння має три корені: 1, 3 і 5.

$$f(x) := x^3 - 9x^2 + 23x - 15 \quad i := 0..8 \quad z_i := \text{root}(x^3 - 9x^2 + 23x - 15, i)$$



**Рис. 24.** Метод “сканування” для відшукування коренів алгебраїчного рівняння

## Тема 7. Задачі Коші та крайові задачі для звичайних диференціальних рівнянь

У диференціальних рівняннях невідомими є не змінні (числа), а функції від однієї чи багатьох змінних. Якщо у диференціальне рівняння входять похідні тільки по одній змінній, то такі рівняння називають *звичайними диференціальними рівняннями (ЗДР)*. У протилежному випадку говорять про *диференціальні рівняння у частинних похідних (ДРЧП)*.

### 7.1. Розв'язання задач Коші для ЗДР за допомогою Solve-блоку

ЗДР з невідомою функцією  $y(t)$ , в яке входять похідні цієї функції аж до  $y^{(N)}(t)$ , називається ЗДР  $N$ -го порядку.

Є два типи задач, які можна розв'язувати за допомогою Mathcad:

- *задачі Коші*, для яких визначені початкові умови на шукані функції, тобто задані значення функції і її похідних до  $(N-1)$ -го порядку в початковій точці інтервалу інтегрування рівняння;



– *крайові задачі*, для яких задані певні співвідношення відразу на обох границях інтервалу.

Приклад: Розглянемо задачу Коші для ЗДР, що описує *модель затухаючого гармонічного осцилятора*. Тут  $y(t)$  описує зміну кута його відхилення від вертикалі,  $y'(t)$  – кутову швидкість маятника,  $y''(t)$  – прискорення, а початкові умови, відповідно, початкове відхилення маятника  $y(0)=1.0$  і початкову швидкість  $y'(0)=0$ .

$$\omega := 0.5 \quad \beta := 0.2$$

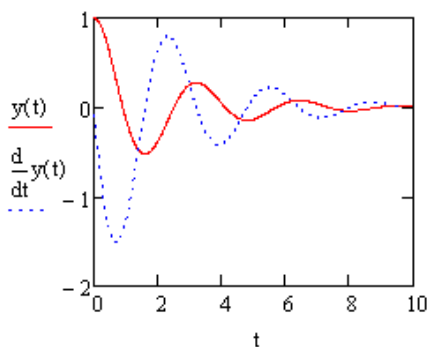
Given

$$\omega^2 \cdot \frac{d^2}{dt^2} y(t) + \beta \cdot \frac{d}{dt} y(t) + y(t) = 0$$

$$y'(0) = 0$$

$$y(0) = 1$$

$y := \text{Odesolve}(t, 10)$



**Рис. 25.** Модель затухаючого гармонічного осцилятора

Примітка: Взагалі кажучи, ЗДР (або система ЗДР) має єдиний розв'язок, якщо крім самого рівняння (системи рівнянь) певним чином задані початкові або граничні умови.

Для розв'язання ЗДР порядку  $N > 1$  в Mathcad передбачені дві можливості:

– блок рішення (обчислювальний блок) **Given/Odesolve** – в цьому випадку розв’язок має вигляд функції від  $t$ ;

– вбудовані функції розв’язання систем ЗДР, при чому рівняння вищих порядків необхідно попередньо звести до еквівалентної системи рівняння першого порядку, в цьому випадку розв’язок має формат вектора.

У даному прикладі використовується перший спосіб, перевагою якого є наочність представлення задачі і результатів, тоді як другий спосіб дає користувачу більше важелів впливу на параметри чисельного методу.

Обчислювальний блок для розв’язання ЗДР, реалізує чисельний метод Рунге–Кутта і складається з трьох частин:

- ключове слово **Given**;
- ЗДР і початкові умови у форматі  $y(t_0)=b$ , записані за допомогою логічних операторів з панелі інструментів Boolean;
- вбудована функція **Odesolve(t,t1)** для розв’язання ЗДР відносно змінної  $t$  на інтервалі  $(t_0,t_1)$ , при чому  $t_0 < t_1$ .

## 7.2. Вбудовані функції для чисельного розв’язання задачі Коші для системи ЗДР

За допомогою Mathcad можна розв’язувати системи  $N \geq 1$  ЗДР першого порядку, якщо вони записані в стандартній формі (Коші) у вигляді векторного співвідношення:

$$Y'(t)=F(Y(t),t)$$

В Mathcad існує декілька вбудованих функцій, які дозволяють розв’язувати задачу Коші різними чисельними методами. Для "хороших" нежорстких систем ЗДР застосовуються функції:

- **rkfixed(y0,t0,t1,M,D)** – метод Рунге–Кутта з фіксованим кроком;
- **Rkadapt(y0,t0,t1,M,D)** – метод Рунге–Кутта із змінним кроком;
- **Bulstoer(y0,t0,t1,M,D)** – метод Булірша–Штера.

Тут  $y_0$  – вектор початкових значень в точці  $t_0$  розміру  $N \times 1$ ;

$t_0$  – початкова точка;

$t_1$  – кінцева точка;

$M$  – число кроків, на яких чисельний метод знаходить розв’язок;

$D$  – векторна функція розмірності  $N \times 1$  двох аргументів – скалярного  $t$  і векторного  $y$ . При цьому  $y$  – шукана векторна функція аргументу  $t$  тієї ж розмірності  $N \times 1$ .

Кожна з наведених функцій видає розв’язок у вигляді матриці розмірності  $(M+1) \times (N+1)$ . В її лівому стовпці знаходяться значення

аргументу  $t$ , що ділять інтервал на рівномірні кроки, а в інших  $N$  стовпцях – значення шуканих функцій  $y_0(t), y_1(t), \dots, y_{N-1}(t)$ , розрахованих для цих значень аргументу. Оскільки всього точок (крім початкової)  $M$ , то рядків в матриці розв'язків буде  $(M+1)$ .

**Приклад:** Розглянемо приклад розв'язання того ж рівняння другого порядку моделі осцилятора  $\omega^2 y'' + \beta y' + y = 0$ . Якщо ввести позначення  $y_0(t) = y(t)$ ,  $y_1(t) = y'(t)$ , тоді  $y_1'(t) = y''(t)$  і дане рівняння 2-го порядку зведеться до еквівалентної системи двох рівнянь 1-го порядку:

$$\begin{aligned} y_0'(t) &= y_1(t); \\ \omega^2 y_1' + \beta y_1 + y_0 &= 0, \end{aligned}$$

Розв'язання даної системи ЗДР проведемо за допомогою функції **rkfixed()**. Результат розрахунків представлений вигляді графіка. Щоб використати інший чисельний алгоритм, достатньо змінити ім'я функції **rkfixed** на **Rkadapt** чи **Bulstoer**.

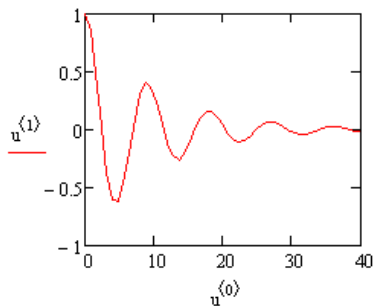
$$\omega := 0.5 \quad \beta := 0.2$$

$$y_0 := \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$M := 50$$

$$D(t, y) := \begin{pmatrix} y_1 \\ -\omega y_0 - \beta y_1 \end{pmatrix}$$

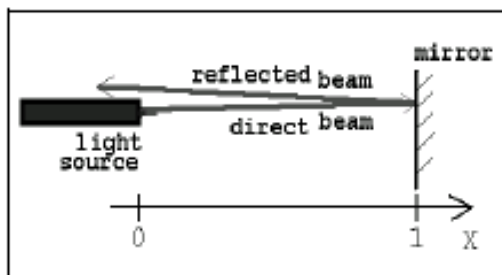
$$u := \text{rkfixed}(y_0, 0, 40, M, D)$$



**Рис. 26.** Приклад розв'язання задачі Коші для системи ЗДР

**7.3. Алгоритм стрільби.** Постановка крайових задач для ЗДР відрізняється від задач Коші тим, що граничні умови для них задаються і в початковій точці і в кінцевій.

Приклад: Розглянемо математичну модель взаємодії зустрічних світлових пучків. Потрібно знайти розподіл інтенсивності оптичного випромінювання в просторі між джерелом (лазером) і дзеркалом, заповненому деяким середовищем.



**Рис. 27.** Графічна ілюстрація задачі відбиття світлового променя від дзеркала

Нехай від дзеркала відбивається  $R$ -та частина падаючого випромінювання, а середовище як поглинає його з коефіцієнтом ослаблення  $a(x)$ , так і розсіює його назад з коефіцієнтом  $r(x)$ . Нехай  $y_0(x)$  – інтенсивність випромінювання, що поширюється вправо, а  $y_1(x)$  – ліво. Тоді матимемо систему рівнянь 1-го порядку:

$$\frac{dy_0(x)}{dx} = -a(x) \cdot y_0(x) + r(x) \cdot y_1(x);$$

$$\frac{dy_1(x)}{dx} = a(x) \cdot y_1(x) - r(x) \cdot y_0(x).$$

з крайовими умовами:

$$y_0(0) = I_0;$$

$$y_1(1) = R \cdot y_0(1).$$

Маємо крайову задачу (boundary value problem) на відрізку  $[0;1]$ . Надалі вважатимемо, що  $R=1$ ,  $a(x)=const=1$  і  $r(x)=const=0.1$ , це описує випадок ізотропного (не залежного від  $x$ ) розсіювання.

Для розв'язання крайових задач в Mathcad реалізований алгоритм, що називається методом *стрільби* або *пристрільки* (*shooting method*). Він

зводить розв'язання крайової задачі до розв'язання серії задач Коші з різними початковими умовами.

Приклад: Розв'яжемо відповідну задачу Коші. В нас не вистачає початкової умови для  $y_1$  в т. 0. Візьмемо, наприклад,  $y_1(0)=10$  – число додатне (випромінювання) і набагато менше від початкового 100 (відбивається).

$$r(x) := 0.1 \quad a(x) := 1$$

$$R := 1$$

$$D(x, y) := \begin{pmatrix} -a(x) \cdot y_0 + r(x) \cdot y_1 \\ a(x) \cdot y_1 - r(x) \cdot y_0 \end{pmatrix}$$

$$I_0 := 100$$

$$y := \begin{pmatrix} 10 \\ 10 \end{pmatrix}$$

$$M := 10$$

$$S := \text{rkfixed}(y, 0, 1, M, D)$$

$$S_{M,1} = 37.558$$

$$S_{M,2} = 15.372$$

Побудуємо графік. Бачимо, що значення шуканих функцій в крайній точці  $x=1$  не співпадають - не задовольняють другу умову:  $y_1(1) = R \times y_0(1)$ .

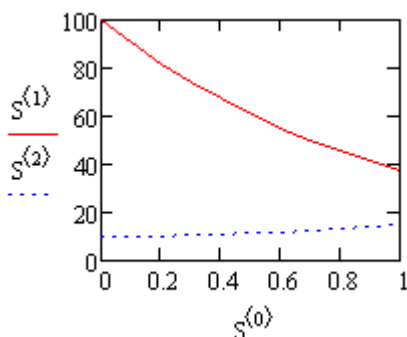


Рис. 28. Графік поширення променя при  $y_1(0)=10$

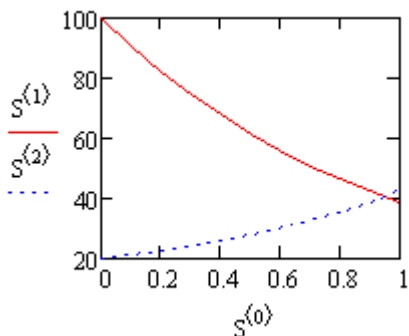


Рис. 29. Графік поширення променя при  $u_1(0)=20$

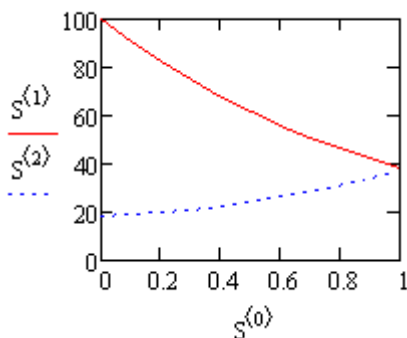


Рис. 30. Графік поширення променя при  $u_1(0)=18$

Збільшимо значення  $u_1(0)$  до 20 (рис. 29) – забагато. Бачимо, що шукане значення лежить десь в межах від 10 до 20, ближче до 20. Візьмемо  $u_1(0)=18$  (рис. 30) – промені співпадають. Це і є метод стрільби (пристрільки).

#### 7.4. Двоточкові крайові задачі

Автоматизоване розв'язання крайових задач для систем ЗДР методом стрільби у Mathcad досягається застосуванням вбудованої функції **sbval()**. Вона призначена для двоточкових задач с крайовими умовами, заданими на кінцях інтервалу:

Функція **sbval(z,x0,x1,D,load,score)** здійснює пошук вектору недостаючих  $L$  початкових умов для двоточної крайової задачі для системи  $N$  ЗДР:

Тут  $z$  – вектор розмірності  $L \times 1$ , що присвоює недостаючим початковим умовам (на лівій границі інтервалу) початкові значення;

$x_0$  – ліва границя розрахункового інтервалу;

$x_1$  – права границя розрахункового інтервалу;

**load**( $x_0, z$ ) – векторна функція розмірності  $N \times 1$  лівих граничних умов, при чому недостаючі початкові умови іменуються відповідними компонентами векторного аргументу  $z$ ;

**score**( $x_1, y$ ) – векторна функція розмірності  $L \times 1$ , що виражає  $L$  правих граничних умов для векторної функції  $y$  в точці  $x_1$ ;

$D(x, y)$  – векторна функція, що описує систему  $N$  ЗДР, розмірності  $N \times 1$  і двох аргументів – скалярного  $x$  і векторного  $y$ . При цьому  $y$  – це невідома векторна функція аргументу  $x$  тої ж розмірності  $N \times 1$ .

Примітка: Число елементів векторів  $D$  і **load** рівне кількості рівнянь  $N$ , а векторів  $z$ , **score** і результату функції **sbval** – кількості правих граничних умов  $L$ . Відповідно, лівих граничних умов в задачі повинно бути  $(N-L)$ .

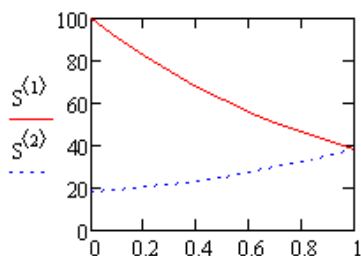
Приклад: Розглянемо попередню задачу, в якій: система складається з двох рівнянь ( $N=2$ ), одна ліва ( $L=1$ ) і одна права ( $N-L=2-1=1$ ) гранична умова (рис. 31).

$$\begin{aligned} r(x) &:= 0.1 & a(x) &:= 1 \\ R &:= 1 & I_0 &:= 100 \\ D(x, y) &:= \begin{pmatrix} -a(x) \cdot y_0 + r(x) \cdot y_1 \\ a(x) \cdot y_1 - r(x) \cdot y_0 \end{pmatrix} \\ z_0 &:= 10 \\ \text{load}(x_0, z) &:= \begin{pmatrix} 100 \\ z_0 \end{pmatrix} \\ \text{score}(x_1, y) &:= R \cdot y_0 - y_1 \\ I_1 &:= \text{sbval}(z, 0, 1, D, \text{load}, \text{score}) \\ I_1 &= (18.555) \\ S &:= \text{rkfixed} \left[ \begin{pmatrix} I_0 \\ I_1 \end{pmatrix}, 0, 1, 10, D \right] \end{aligned}$$

**Рис. 31.** Автоматизований метод стрільби для задачі відбиття світлового променя від дзеркала

Перші три рядки програми задають необхідні параметри задачі і саму систему ЗДР. У четвертому рядку визначається вектор  $z$ . Оскільки права гранична умова всього одна, то недостаюча початкова умова теж одна, відповідно, і вектор  $z$  має лише один елемент  $z_0$ . Йому необхідно присвоїти початкове значення ( $z_0=10$ ), щоб запустити алгоритм стрільби. Далі векторній функції **load**( $x,z$ ) присвоюються ліві граничні умови. Далі функція **score**( $x,y$ ) визначає праву граничну умову. Вона записується так само, як система рівнянь в функції D. Аргумент  $x_1$  функції **score** потрібен для тих випадків, коли гранична умова явно залежить від координати  $x$ .

Реалізований у функції **sbval** алгоритм стрільби шукає недостаючі початкові умови так, щоб розв'язання отриманої задачі Коши робило функцію **score**( $x,y$ ) як можна ближчою до нуля. Результат застосування **sbval** для інтервалу (0,1) присвоюється векторній змінній  $\Pi$ . Цей вектор схожий на вектор  $z$ , тільки він містить шукані початкові умови, замість наближених, заданих в  $z$ . Вектор  $\Pi$  містить, як і  $z$ , всього один елемент  $\Pi_0$ . З його допомогою можна знайти  $S$  – розв'язок крайової задачі  $y(x)$ . Тим самим функція **sbval** зводить розв'язання крайових задач до задач Коші.



**Рис. 32.** Графічна ілюстрація автоматизованого методу стрільби для двоточкових крайових задач

## Тема 8. Задачі Коші та крайові задачі для диференціальних рівнянь у частинних похідних

### 8.1. Види ДРЧП

Диференціальні рівняння у частинних похідних потребують знаходження функції не однієї, як для ЗДР, а декількох змінних, наприклад,  $f(x,y)$  чи  $f(x,t)$ . Постановка задач включає у себе саме рівняння (або систему рівнянь) з частинними похідними по різним змінним, а також певну кількість крайових умов на границях розрахункової області.



Mathcad має досить обмежені можливості по відношенню до ДРЧП (у ньому є декілька вбудованих функцій). Розв'язувати ДРЧП можна і шляхом безпосереднього програмування алгоритмів користувача.

ДРЧП поділяють на три типи:

– *параболічні* рівняння – використовуються для опису нестационарних процесів дифузії, теплопровідності тощо. Наприклад, одновимірне рівняння теплопровідності має вигляд:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}.$$

– *гіперболічні* рівняння – використовуються для опису хвильових процесів. Наприклад, одновимірне рівняння вільних коливань струни має вигляд:

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2},$$

– *еліптичні* рівняння – використовуються для опису стаціонарних процесів. Наприклад, тривимірні рівняння Лапласа і Пуассона мають вигляд відповідно:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = 0$$

$$\left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) \varphi(x, y, z) = f(x, y, z).$$

## 8.2. Скінченно-різнцева апроксимація задачі Коші для ДРЧП.

Приклад: Розглянемо одновимірне рівняння теплопровідності, пов'язану з ним крайову задачу, з початковою і крайовими умовами і метод сіток для її розв'язання.

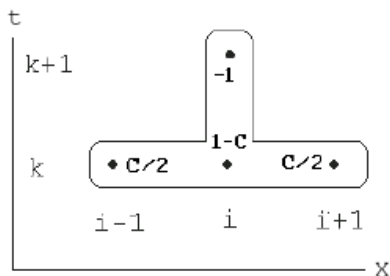
$$\frac{\partial u(x, t)}{\partial t} = D \frac{\partial^2 u(x, t)}{\partial x^2} + \phi(x, t, u),$$

$$u(x, 0) = u_0(x),$$

$$u(0, t) = f_0(t), \quad u(L, t) = f_1(t),$$

Спочатку розрахункову область  $(x,t)$  покривають сіткою з кроком  $\Delta t$  і  $\Delta x$  відповідно і використовують ці вузли для різницевої апроксимації рівняння. Розглянемо явну і неявну схеми Ейлера.

• **Явна схема Ейлера.** Для різницевої апроксимації використаємо наступний шаблон:



**Рис. 33.** Шаблон для явної схеми Ейлера

Замінивши похідні у вихідному ДРЧП їх скінченними різницями, отримаємо:

$$\frac{u_{i,k+1} - u_{i,k}}{\Delta_t} = D \frac{u_{i-1,k} - 2u_{i,k} + u_{i+1,k}}{\Delta_x^2} + \phi_{i,k}.$$

Зведемо подібні доданки і введемо коефіцієнт:

$$C_{i,k} = \frac{2 \cdot \Delta_t \cdot D_{i,k}}{\Delta_x^2}.$$

Це коефіцієнт Куранта ( $Co$ ). Після групування подібних доданків отримаємо:

$$u_{i,k+1} = \frac{C_{i,k}}{2} u_{i-1,k} + \{1 - C_{i,k}\} u_{i,k} + \frac{C_{i,k}}{2} u_{i+1,k} + \phi_{i,k}.$$

Тоді, програмна реалізація явної схеми Ейлера для лінійного рівняння теплопровідності може мати вигляд (у даному прикладі немає джерел тепла  $\phi(x, u) := 0$ , коефіцієнт дифузії постійний  $D(u) := 1$ ):

$$\tau := 0.0005$$

$$M := 20$$

$$\Delta := \frac{1}{M}$$

$$D(u) := 1$$

$$\phi(x, u) := 0$$

Init(x) :=  $\Phi(x - 0.45) - \Phi(x - 0.55)$

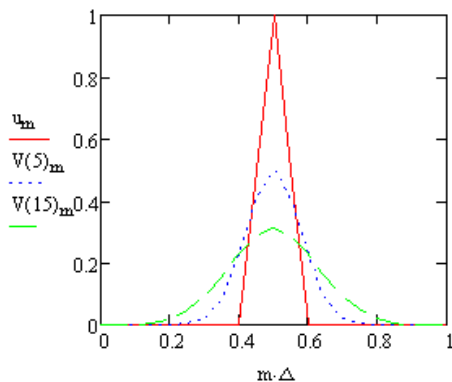
Border(t) := 0

$\underline{m} := 0..M$

$u_m := \text{Init}(m \cdot \Delta)$

$$F(v) := \begin{cases} v1_0 \leftarrow \text{Border}(\tau \cdot T) + 0 \cdot (v_0 + v_1) \\ v1_M \leftarrow \text{Border}(\tau \cdot T) + 0 \cdot (v_M + v_{M-1}) \\ \text{for } m \in 1..M-1 \\ v1_m \leftarrow \Phi(m \cdot \Delta, v_m \cdot \tau) + v_{m-1} \cdot \frac{D(v_{m-1}) \cdot \tau}{\Delta^2} + v_m \left( 1 - \frac{2D(v_m) \cdot \tau}{\Delta^2} \right) + v_{m+1} \cdot \frac{D(v_{m+1}) \cdot \tau}{\Delta^2} \end{cases}$$

$V(t) := \begin{cases} u & \text{if } t = 0 \\ F(V(t-1)) & \text{otherwise} \end{cases}$



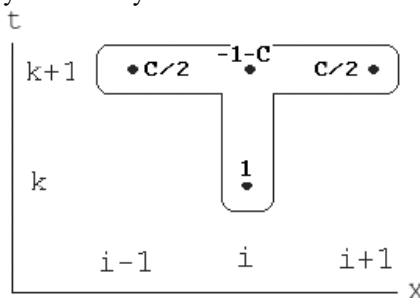
**Рис. 34.** Програмна реалізація явної схеми Ейлера для одновимірного рівняння теплопровідності

Недолік явної схеми – її нестійкість. Програма працює при  $\tau = 0.0005$ ;  $\tau = 0.0010$ , проте при подальшому незначному збільшенні кроку до  $\tau = 0.0015$ , отримуємо осцилюючі нереальні профілі температури.

З теорії чисельних методів відомо, що явна схема Ейлера для рівняння теплопровідності стійка при значеннях коефіцієнта Куранта, менших 1, і нестійка у протилежному випадку. Тобто, існує обмеження для вибору співвідношення між кроками: на більш частих просторових сітках необхідно використовувати також і малі кроки по часу.

- **Неявна схема Ейлера**

На відміну від явної неявна схема є безумовно стійкою, однак на кожному кроці по часу необхідно розв'язувати системи алгебраїчних рівнянь. Використовується наступний шаблон:



**Рис. 35.** Шаблон для неявної схеми Ейлера

В результаті одержимо наступне різницеве рівняння :

$$\frac{u_{i,k+1} - u_{i,k}}{\Delta_t} = D \frac{u_{i-1,k+1} - 2u_{i,k+1} + u_{i+1,k+1}}{\Delta_x^2} + \varphi_{i,k}.$$

Неявна схема для лінійного рівняння теплопровідності може бути реалізована в Mathcad наступним чином:

$$D := 1 \quad \varphi(x, u) := 0$$

$$\text{Border}(t) := 0$$

$$\text{Init}(x) := \Phi(x - 0.45) - \Phi(x - 0.55)$$

$$\tau := 0.005$$

$$M := 20$$

$$\Delta := \frac{1}{M} \quad \Delta = 0.05$$

$$m := 0..M$$

$$u_m := \text{Init}(m \cdot \Delta)$$

$$\text{Cou} := \frac{2D \cdot \tau}{\Delta^2} \quad \text{Cou} = 4$$

$$m := 1..M - 1$$

$$A_{m,m} := -\text{Cou} - 1$$

$$A_{m,m-1} := \frac{\text{Cou}}{2} \quad A_{m,m+1} := \frac{\text{Cou}}{2}$$

$$A_{0,0} := 1 \quad A_{M,M} := 1$$

$$V(n) := \begin{cases} u & \text{if } n = 0 \\ \text{lsolve}(A, -V(n-1)) & \text{otherwise} \end{cases}$$

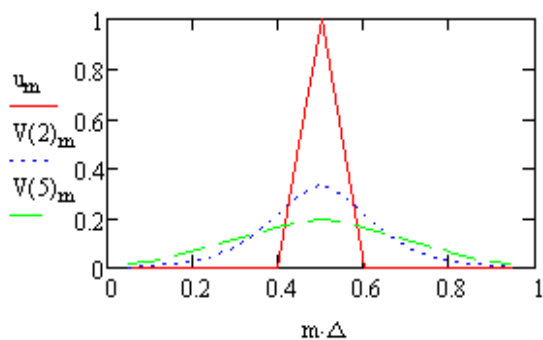


Рис. 36. Програмна реалізація неявної схеми Ейлера для одновимірного рівняння теплопровідності

## ЛИТЕРАТУРА

1. Кирьянов Д. В. Mathcad 15/Mathcad Prime 1.0. СПб. : БХВ-Петербург, 2012. 432 с.
2. Дьяконов В. П. Энциклопедия компьютерной алгебры. М. : ДМК Пресс, 2009. 1264 с.
3. Тарасевич Ю. Ю. Использование пакетов Maple, Mathcad и LATEX 2 $\epsilon$  при решении математических задач и подготовке математических и естественно-научных текстов. Информационные технологии в математике. М. : Либроком, 2016. 134 с.
4. Охорзин В. А. Прикладная математика в системе Mathcad. Издательство : Лань, 2009. 352 с.
5. Очков В.Ф. MathCAD 14 для студентов, инженеров и конструкторов. СПб. : БХВ-Петербург, 2007. 368 с.
6. Макаров Евгений. Инженерные расчеты в Mathcad 15. Учебный курс. М. : Питер, 2011. 400 с.
7. Каганов В.И. Компьютерные вычисления в средах Excel и Mathcad. М. : Горячая линия - Телеком, 2011. 741 с.
8. Штыков В. В. MathCAD. Руководство по решению задач для начинающих. М. : Либроком, 2013. 168 с.
9. Щепетов, А. Г. Автоматизация инженерных расчетов в среде Mathcad. М. : Стандартинформ, 2006. 264 с.