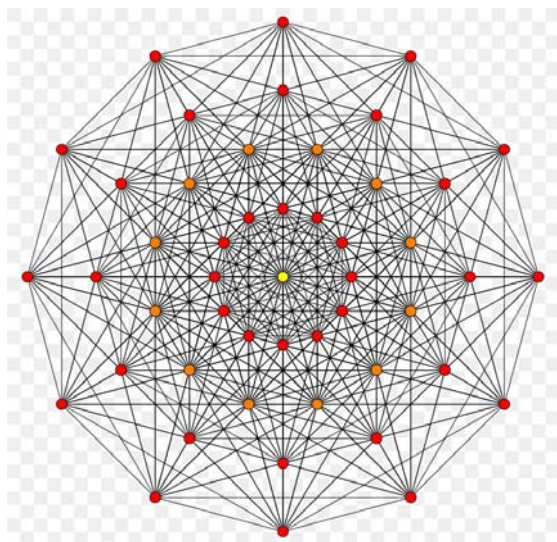




А.М. Грищук, П.П. Корнійчук

**Використання системи символьного
числення «Mathematica»
при проходженні навчальної практики з
теоретичної фізики**

**Використання системи символьного
числення «Mathematica»
при проходженні навчальної практики з
теоретичної фізики**

**Навчально-методичний посібник
для студентів старших курсів
спеціальності
«104 Фізика-Астрономія», «014 Середня освіта
(Фізика)».**

Грищук А.М. Корнійчук П.П.

Житомир
ЖДУ
2022

УДК 519.67/.68

ББК 22.183.49

Г-62

*Рекомендовано до друку Вченою радою Житомирського державного
університету імені Івана Франка
(протокол №10 від 24.06.2022 р.).*

Рецензенти:

О.М. Маханець, доктор фізико-математичних наук, професор кафедри теоретичної фізики та комп'ютерного моделювання, Чернівецького національного університету імені Ю.Федьковича

С.П. Семенець, доктор педагогічних наук, професор, кафедри фізики та вищої математики державного університету «Житомирська політехніка».

В.Л. Рудницький, директор Наукового ліцею Житомирського державного університету імені Івана Франка.

Г Гришук А.М., Використання системи символьного числення «Mathematica» при проходженні навчальної практики з теоретичної фізики [метод.] / Гришук А.М., Корнійчук П.П. – Житомир: ЖДУ, 2022. – 62с.

Навчально-методичний посібник складається з трьох частин. В першій частині обговорюються загальна проблематика «Навчальна та науково-дослідна практика на кафедрах університету», надаються методичні рекомендації для проведення практики, опису теоретичної частини практики, постановки та розв'язку задачі, оформленню звітної документації.

В другій частині приділена особлива увага наглядності описаних методик розв'язку задач теоретичної фізики за допомогою програми символьного числення Mathematica 8-12. Наведено приклади використання основних функцій, які значно полегшують розуміння багатьох абстрактних понять складної комп'ютерної системи. Розв'язки всіх приведених прикладів отримані безпосередньо в системі Mathematica 8-12.

В третій частині містяться контрольні завдання та можливі завдання з навчальної практики теоретичної фізики, які може вибрати здобувач для використання в процесі проходження практики.

ББК 22.183.49

УДК 519.67/.68

© Гришук А.М.

© ЖДУ

ВСТУП

Розв'язування задач теоретичного змісту є невід'ємною складовою навчального процесу, що сприяє формуванню фізичних понять, розвитку логічного мислення, навичок практичного застосування знань, допрофільній підготовці та професійній орієнтації студентів. У практиці навчально-виховної діяльності прикладні задачі використовуються як метод засвоєння, закріплення, перевірки і контролю теоретичних знань; засіб реалізації принципу політехнізму, екологічного й економічного виховання, набуття навичок професійного самовизначення.

Бурхливий розвиток комп'ютеризації суспільства розпочався з початком масового виробництва і впровадження персональних комп'ютерів (ПК). Довгий час їх обмежені можливості не дозволяли реалізовувати на них потужні системи символьної математики, і тому їх використовували як потужні калькулятори або не менш потужні друкарські машинки. Але з початку 90х років ситуація почала змінюватися не лише у кількісному, але й у якісному плані. Ріст потужності ПК і поява графічних операційних систем дали поштовх розвитку **систем комп'ютерної символьної математики (СКСМ)**, які вже існували на великих ЕОМ і були доступні лише представникам наукової еліти розвинутих країн.

Еру створення СКСМ прийнято відрховувати з початку 60-х років. Саме тоді в обчислювальній техніці виник новий напрямок комп'ютерної математики, який назвали **комп'ютерною алгеброю**. Мова йшла про можливість створення комп'ютерних систем, здатних здійснювати типові алгебраїчні перетворення, підстановки у виразах, спрощення виразів, розв'язувати рівняння та системи рівнянь, розраховувати похідні та інтеграли. При цьому передбачалась можливість отримання аналітичних, символьних результатів всюди, де це можливо. Зрозуміло, що здійснення символьних операцій – процес набагато тонший і складніший, ніж реалізація навіть складних числових розрахунків. Відомо, що лише таблиці інтегралів, похідних, сум та формул перетворення і зображення спецфункцій займають велику кількість об'ємних книжок. Ось чому висока ефективність символьних розрахунків стала реальною лише в останні роки.

На теперішній час лідерами серед систем комп'ютерної алгебри є Mathematica 12 і Maple 9. Поступаються їм системи MATLAB та

Mathcad, яка стала міжнародним стандартом для технічних числових розрахунків. Блок символічної математики у ці системи доданий на основі ядра Maple V.

Системи символічної математики у провідних країнах вивчаються не лише в університетах, але й у школах. На основі ядра системи Mathematica реалізовані електронні інтерактивні математичні енциклопедії та довідники.

МЕТА, ЗАВДАННЯ ТА ОСНОВНІ ТЕМИ ПРАКТИКИ

Мета: закріпити і поглибити знання студентів, здобуті при вивченні навчальних дисциплін теоретичного курсу фізики «Електродинаміка», «Нанофізики», «Квантова механіка» та елементів із загальної теорії відносності, набуті глибших навичок з розв'язування задач за даним розділами фізики, в тому числі і задач поглибленого теоретичного курсу фізики, здобути початкові вміння розв'язання фізичних задач за допомогою прикладних програмних пакетів символічних обчислень та мов програмування. Навчальна практика також покликана забезпечити закріплення одержаних знань та теоретичних навичок практичної роботи в умовах лабораторій кафедри фізики та охорони праці.

Завдання:

закріпити вміння розв'язувати задачі з розділів Електродинаміка, Нанофізика та Загальна теорія відносності.

вивчити спеціальні підходи до розв'язання задач підвищеної складності, дослідницьких задач.

здобути навички розв'язування фізичних задач з Нанофізики та Електродинаміки за допомогою чисельних методів, що реалізовані прикладних програмних додатках

навчитися використовувати стандартні алгоритмічні структури для моделювання розв'язків фізичних задач дослідницького типу.

За час проходження практики здобувач вищої освіти має набути таких компетентностей:

ІК. Здатність особи розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі освіти (предметна спеціалізація фізика) або у процесі навчання, що передбачає застосування певних теорій та методів фізики і характеризується комплексністю та невизначеністю умов.

ЗК 3. Здатність навчатися та самонавчатися.

ЗК 4. Здатність бути критичним та самокритичним.

ЗК 5. Здатність до абстрактного та аналітичного мислення й генерування ідей (креативність).

ЗК 6. Здатність застосовувати знання на практиці.

СК 1. Володіння уявленнями про фізику як науку і навчальний предмет, визначення її місця у сучасному світі і в системі наук.

СК 2. Володіння математичним апаратом фізики у межах, достатніх для вивчення предметів загальної та теоретичної фізики.

СК 3. Здатність здійснювати експериментальне дослідження фізичних об'єктів і на їх основі будувати математичні моделі, застосовувати сучасні інформаційні технології для розв'язання прикладних задач фізики.

СК 5. Здатність використовувати систематизовані теоретичні й практичні знання з фізики та методики навчання фізики у вирішенні професійних завдань.

СК 6. Здатність до формування в учнів ключових і предметних компетентностей та здійснення міжпредметних зв'язків.

СК 8. Здатність розв'язувати задачі курсу фізики різного рівня складності та пояснювати їх розв'язання учням.

СК 12. Здатність до критичного аналізу, діагностики й корекції власної педагогічної діяльності, оцінки педагогічного досвіду.

Програмні результати навчання:

ПР 6. Знати та розуміти основні поняття, закони, теорії, загальну структуру сучасної фізики, експериментальні методи дослідження та методику навчання фізики, її місце і зв'язки в системі наук, етапи історії розвитку.

ПР 9. Розв'язувати задачі різних рівнів складності курсу фізики на рівні базової середньої освіти, чітко й раціонально пояснювати їх розв'язання учням.

ПР 11. Володіти основами наукових досліджень, здійснювати самостійну експериментальну діяльність з фізики та методики навчання фізики з описом, аналізом та критичним оцінюванням експериментальних даних.

ПР 22. Мати навички самостійного прийняття рішень стосовно своїх освітньої траєкторії та професійного розвитку.

Загальні теми програми практики

Тематичний зміст практики

№ модуля	Зміст
Модуль 1. Методика розв'язування задач з розділу теоретичної фізики «Електродинаміка»	
1	Методи розв'язання стандартних задач з розділу Електродинаміка та загальна теорія відносності, в тому числі і задач шкільного курсу
2	Вивчення спеціальних підходів до рішення задач підвищеної складності та дослідницьких задач з Електродинаміки
Модуль 2. Методика розв'язування задач з розділу теоретичної фізики «Нанофізика»	
3	Методи розв'язання стандартних задач з розділу Нанофізика, та початків Квантової механіки
4	Вивчення спеціальних підходів до рішення задач підвищеної складності та дослідницьких задач з Нанофізики та початків Квантової механіки
Модуль 3. Розв'язування задач з теоретичної фізики за допомогою програмних засобів. Захист результатів практики	
5	Знайомство з основними програмними пакетами для чисельного розв'язування задач теоретичної фізики.
6	Програмування чисельних розв'язків теоретичних задач дослідницького типу.
7	Звіт і захист результатів практики

Розподіл часу індивідуальної роботи з переліком завдань

Вид роботи
Використання методів математичного аналізу для розв'язку задач з теоретичної фізики
Використання чисельних методів інтегрування диференціальних рівнянь при розв'язанні фізичних задач з Нанофізики.
Методи математичної фізики, та векторного і тензорний аналізу в задачах з Електродинаміки та загальної теорії відносності.
Підготовка звіту

Розподіл часу індивідуальної роботи з переліком завдань

Вид роботи
Постановка індивідуального завдання з теоретичної фізики.
Опрацювання літератури та статей які стосуються індивідуального завдання.
Аналітичний розв'язок поставленої задачі.
Моделювання отриманих результатів за допомогою програмних пакетів для чисельного розв'язування задач
Аналіз отриманих результатів, висновки.

До керівництва практикою здобувачів залучаються досвідчені викладачі кафедр, а також завідуючі відділеннями навчальних закладів, які брали безпосередню участь в навчальному процесі, по якому проводиться практика. Відповідальний за проведення практики: перед початком практики контролює підготовленість баз практики та до прибуття

здобувачів проводить відповідні заходи з практики. Розподіл здобувачів на практику проводиться кафедрами і відділом практики. В період проходження практики студент зобов'язаний дотримуватись вимог календарного індивідуального плану практики, трудової дисципліни на робочому місці, правил техніки безпеки і протипожежної безпеки, виконувати всі розпорядження керівника практики.

Керівник практики, який призначається з числа викладачів кафедри, або ведучих спеціалістів підприємств і установ, зобов'язаний скласти затверджений на засіданні кафедри індивідуальний план практики для кожного здобувача, здійснювати контроль за виконанням цього плану, надавати здобувачам своєчасну допомогу в оволодінні ними кваліфікаційними навичками в роботі з експериментальною апаратурою та під час роботи зі спеціальною і науковою літературою.

Форми та методи контролю

Під час практики керівники здійснюють поточний контроль своєчасного виконання індивідуальних завдань.

Підсумковий контроль за роботою практикантів:
перевірка звіту;
захист звіту.

Підсумки практики з теоретичної фізики підводяться шляхом складання студентами заліку.

Результати складання заліку з практики заносяться до залікової відомості, залікової книжки та в журнал обліку успішності.

Контроль успішності студентів з урахуванням поточного і підсумкового оцінювання здійснюється відповідно до індивідуального плану практики.

Вимоги до звітної документації

Звітними документами є звіт з практики з викладом результатів виконання індивідуальних завдань. Звіт з практики складається з двох частин. Перша частина повинна містити детальний розв'язок фізичних задач, що має зробити студент згідно із своїм індивідуальним завданням. Звіт з практики може оформлятися як в рукописному вигляді так і в електронному вигляді. Однак для допуску до захисту, електронні звіти мають бути роздруковані. Друга частина містить виконання дослідницьких

завдань з використанням комп'ютерних технологій. Ця частина може бути оформлена повністю в електронному вигляді і не потребує друкування. Друга частина звіту повинна містити повний розв'язок завдання в одному із прикладних програмних пакетів для чисельного моделювання фізичних явищ або програмний код, написаний на довільній мові програмування, що дає розв'язок поставленого завдання. Звіт здається керівнику не пізніше ніж за тиждень після закінчення практики. На захисті звіту студент повинен вміти пояснити отримані ним розв'язки фізичних задач, згідно до його індивідуального завдання. Усний звіт у формі повідомлення заслуховується комісією із захисту практики.

МЕТОДИКА РОЗВ'ЯЗУВАННЯ ЗАДАЧ ІЗ ТЕОРЕТИЧНОЇ ФІЗИКИ

Розв'язування задач теоретичного змісту є невід'ємною складовою навчального процесу, що сприяє формуванню фізичних понять, розвитку логічного мислення, навичок практичного застосування знань, допрофільній підготовці та професійній орієнтації студентів. У практиці навчально-виховної діяльності прикладні задачі використовуються як метод засвоєння, закріплення, перевірки і контролю теоретичних знань; засіб реалізації принципу політехнізму, екологічного й економічного виховання, набуття навичок професійного самовизначення.

Теоретичну фізичну задачу розглядаємо як уявну модель певної виробничої ситуації, що відображає систему зв'язків, утворених у процесі професійної діяльності людини. Її розв'язання потребує від студентів розумових і практичних дій на основі законів і методів фізики, спрямованих на оволодіння теоретичними знаннями та розвиток мислення.

З метою профорієнтації студентів на основі аналізу змісту курсів теоретичної фізики, здійснено класифікацію задач прикладного характеру:

- за змістом (абстрактні й конкретні, з виробничим і історичним наповненням);
- за дидактичними цілями (тренувальні, контролюючі, творчі);
- за способом подання умови (текстові, графічні, завдання-малюнки, завдання-досліди);
- за рівнем складності (прості, складні, комбіновані);
- за характером й методом дослідження (обчислювальні, якісні, експериментальні, дослідницькі).

Охарактеризуємо деякі види прикладних фізичних задач.

Обчислювальні - необхідні під час вивчення тем курсу, що містять числові відношення і залежності між величинами (закони кінематики, динаміки, збереження енергії, постійного струму тощо) з метою з'ясування й усвідомлення фізичного змісту відповідних явищ і процесів.

Якісні - це задачі, розв'язання яких не потребує обчислень. Застосування таких задач сприяє розвитку мовлення студентів, формуванню вмінь чітко, логічно й лаконічно висловлювати думку, «оживляє» виклад навчального матеріалу, підвищує пізнавальну активність здобувачів. Наприклад, чому зволожений ґрунт промерзає на меншу глибину, а вологі предмети замерзають і руйнуються швидше? Для

охолодження молока на фермах використовують лід, який зберігають у лідниках. Для чого його посипають сіллю?

Експериментальні - задачі, для розв'язання яких вихідні дані отримують дослідним шляхом або з експерименту. Перевага експериментальних задач полягає в тому, що вони не можуть бути розв'язані лише формально, без розуміння сутності певного фізичного процесу. Наприклад, під час вивчення дії фізичного приладу реостата учні усвідомлюють різницю між приладом, що регулює струм у ланцюзі, і дільником напруги (потенціометром).

Дослідницькі - це задачі, предметом яких є фізичні явища, процеси, факти й відповідні їм моделі. Розв'язання таких задач потребує застосування методів наукових досліджень або їх елементів.

Здійснимо класифікацію навчальних дослідницьких задач на основі методів наукового пізнання:

- дослідження на основі побудованих моделей;
- дослідження відмінностей між ідеальними моделями та їх реальними прообразами (об'єктами, явищами, процесами);
- розроблення моделі експерименту;
- формулювання висновків спостережень та експериментів;
- висунення, формулювання і перевірка гіпотез;
- добір засобів вимірювання тощо.

Графічні - дають змогу наочно й доступно виражати функціональні залежності між величинами, що характеризують фізичні процеси у природі й техніці (вивчення різних видів механічного руху, газових законів та ін.). Деякі фізичні закономірності спочатку можна представити лише графічно, а потім виразити аналітично (робота змінної сили).

Задачі з неповними даними найчастіше зустрічаються в житті, коли відсутні дані відшуковують у таблицях, довідниках або шляхом вимірювань. Розв'язування задач цього типу сприяє формуванню навичок самостійної роботи з довідковою літературою. Наприклад, який максимальний вантаж може витримати алюмінієвий (мідний, сталевий та ін.) дріт заданого перетину? При якій найменшій довжині він обривається?



Рис. 1. Динамічна модель розв'язування прикладної фізичної задачі

У навчальній практиці вироблено загальний алгоритм розв'язування різних типів прикладних фізичних задач:

- вивчення умови та з'ясування змісту нових термінів і виразів;
- короткий запис умови, виконання потрібних малюнків, схем, графіків (фізичні величини мають бути виражені у Міжнародній системі одиниць (СИ));
- аналіз умови задачі, у процесі якого з'ясовується її фізична сутність, встановлюються фізичні явища, процеси, стани системи та закони і закономірності, потрібні для розв'язку;
- складання плану розв'язування;
- вираження зв'язків між невідомими і відомими величинами у вигляді формул;
- розв'язування системи рівнянь для одержання невідомого;
- обчислення шуканої величини;
- аналіз одержаних результатів;
- пошук і аналіз інших шляхів розв'язування.

Під час розв'язування конкретних задач деякі етапи загального алгоритму можуть бути пропущені.

Виявлення зв'язків між елементами прикладної задачі здійснюється у структурно- функціональній послідовності (сукупності гностичних актів). Гностичні акти як компонент процесу розв'язування задач - це послідовне переміщення певного «уявного погляду» з одного елемента задачної ситуації на інший. Сукупність таких актів, у процесі яких розкриваються властивості елементів розв'язуваної задачі, називається гностичною динамікою. На основі гностичної динаміки процесу дослідження елементів задачі, зіставлення вихідної й кінцевої ситуацій, здобувач мисленнєво будує її динамічну модель, яку можна представити у вигляді схеми (рис. 1).

Без усвідомлення наведених вище етапів розумової діяльності під час розв'язування фізичних задач здобувачі не в змозі оволодіти необхідними навчальними прийомами і мисленнєвими операціями, що призводить до значних ускладнень.

Залежно від виду мисленнєвих операцій, розрізняють аналітичний, синтетичний та аналітико-синтетичний способи розв'язування прикладних фізичних задач.

Аналітичний - полягає в поділі складної задачі на простіші. Розв'язування розпочинається з відшукування закономірностей, що дають змогу знайти безпосередню відповідь на запитання задачі. Кінцева розрахункова формула утворюється шляхом синтезу окремих фізичних закономірностей. Завдяки аналізу учень осмислює умову складної задачі, розчленовуючи її на складові, кожна з яких досліджується окремо. Процес поділу лише тоді постає засобом пізнання, коли здійснюється неперервне співвіднесення простіших задач із загальною, умови задачі з вимогами та наявними знаннями.

Синтетичний - полягає у послідовному виявленні зв'язків вихідних величин з іншими, поки не одержимо рівняння з невідомою величиною. На відміну від аналітичного синтетичний спосіб передбачає початок розв'язування з вихідних величин. Аналіз умови задачі обов'язково супроводжується синтезом, мисленнєвим об'єднанням простіших задач, що виділені у процесі аналізу. Поглиблене пізнання умови здійснюється шляхом розкриття нових зв'язків шуканих фізичних величин з відомими та шляхом синтезу результатів аналізу. Нові знання, здобуті в результаті синтезу, постають об'єктом глибшого аналізу, що завершується синтезом вищого рівня. У цьому процесі аналіз і синтез постійно переплітаються,

розкриваючи суттєві зв'язки між явищами і фізичними величинами.

Аналітико-синтетичний - у «чистому вигляді» аналітичний і синтетичний способи майже не застосовуються. Під час розв'язування задач використовують, як правило, загальний аналітико-синтетичний.

Евристичний - використовується під час розв'язування якісних задач і полягає у формулюванні взаємозалежних цілеспрямованих запитань. На основі цього способу формуються навички логічного мислення, аналізу фізичних явищ, складання плану розв'язку задачі, узагальнення фактів, дослідження вірогідності результату.

Розрізняють такі форми евристичного способу розв'язування прикладних якісних задач у процесі навчання фізики:

- а) *«навідних» запитань* - передбачає формулювання запитань учителем;
- б) *питально-відповідальна* - передбачає формулювання запитань і відшукування відповідей здобувачем у письмовій формі;
- в) *оповідальна* - передбачає усне розв'язування задачі у вигляді логічно пов'язаних речень, що утворюють цілісне оповідання.

Знання особливостей структурних елементів прикладної задачі, аналіз функціональних залежностей між фізичними величинами дають змогу здійснити логічне обґрунтування, письмове оформлення її розв'язки.

На цьому етапі важливу роль у мисленнєвій діяльності відіграють проєктувальні вміння учня:

- планування дій, пов'язаних із розв'язуванням задачі;
- відбір раціональних і ефективних способів їх здійснення;
- передбачення можливих ускладнень під час самостійної роботи;
- прогнозування ефективності обраної методики розв'язку.

Невід'ємним компонентом продуктивної діяльності учня на етапі оформлення розв'язку задачі є конструктивні вміння:

- композиційна побудова навчально-пізнавальної діяльності;
- передбачення різних варіантів її реалізації;
- формулювання запитань;
- вираження функціональних залежностей між величинами у вигляді загальної формули.

Після знаходження основного співвідношення між фізичними величинами, а також значень невідомих процес розв'язування задачі не завершується. На цьому етапі домінує контролююча гностична діяльність здобувачів, що потребує аналізу результату, обґрунтування його вірогідності. Її ефективність залежить від сформованості таких умінь:

формулювання мети аналізу і висновків, вибір критеріїв якості розв'язування задачі, узагальнення й перенесення одержаних результатів на часткові випадки, оцінка доцільності обраних дій тощо.

Визначимо основні критерії сформованості вмінь розв'язувати прикладні фізичні задачі:

- знання основних операцій, з яких складається процес розв'язування;
- засвоєння структури сукупності операцій;
- перенесення засвоєного способу розв'язку з одного розділу на інші.

Залежно від рівня математичного апарату виокремлюють такі способи

розв'язування задач: арифметичний, алгебраїчний та геометричний.

Арифметичний - передбачає поетапне розв'язування із застосуванням математичних дій або тотожних перетворень виразів з фізичними величинами без складання рівнянь.

Алгебраїчний - ґрунтується на використанні фізичних формул для складання рівнянь, з яких визначається шукана фізична величина.

Геометричний - полягає у застосуванні геометричних властивостей фігур і тригонометричних залежностей між їх елементами.

Добір і складання прикладних задач здійснюється з урахуванням психофізіологічних та індивідуальних особливостей здобувачів, змісту і специфіки навчального матеріалу, що створює необхідні умови для ефективного навчання. Результативність навчально-виховного процесу залежить від методичної компетентності вчителя, важливою складовою якої є професійні вміння (гностичні, проєктувальні, конструктивні, комунікативні, організаторські).

Знання різних способів розв'язування прикладних фізичних задач сприяє ефективному формуванню фізичних понять, різнобічному, міцному й глибокому усвідомленню змісту навчального матеріалу, набуттю практичних умінь і навичок застосовувати фізичні закони і закономірності, створює умови для реалізації принципу політехнізму.

ТИПИ ДАНИХ І РОБОТА З НИМИ

1. Числові дані

Цілі числа. Розраховуються абсолютно точно, зберігаються усі значущі цифри. Наприклад, задамо такий вираз

$$5 + 4^2 + 20!$$

Отримаємо абсолютно точний результат

$$2432902008176640021.$$

Раціональні числа. Розраховуються абсолютно точно, результат отримаємо у вигляді дробу, який у разі необхідності можна перетворити у дійсне число з потрібною точністю.

$$\frac{1}{3} + \frac{1}{4} = \frac{7}{12}$$

Ірраціональні числа. Вираз, що містить ірраціональні числа в міру можливості спрощується. Отриманий результат абсолютно точний, без будь-яких округлень.

$$\text{Out}[2] = -3 + 4\sqrt{2} + 2\pi^2$$

При необхідності раціональні та ірраціональні числа можуть бути записані з потрібною точністю через дійсні числа з допомогою функції **N[вираз, точність]**. Наприклад, число π :

$$\text{N}[\pi, 50]$$

$$\text{Out}[3] = 3.1415926535897932384626433832795028841971693993751$$

Комплексні числа можуть бути записані через усі попередні типи чисел і тому володіють всіма їх властивостями

$$2 + 5\pi i + 4\sqrt{-3}$$

$$\text{N}\left[2 + 5\pi i + 4\sqrt{-3}, 4\right]$$

$$\text{Out}[4] = 2 + 4i\sqrt{3} + 5i\pi$$

$$\text{Out}[5] = 2.000 + 22.64i$$

2. Символьні дані. Задаються рядком символів у лапках, наприклад "ssssss". У символьних даних можна використовувати управляючі символи: **\n** – новий рядок, **\t** – табуляція.

3. Вирази. Під виразом в системі Mathematica розуміють математичні формули, які можуть містити як оператори, так і функції. При записі виразів необхідно дотримуватись такого синтаксису:

- знак множення може бути замінений пробілом;
- функції, вбудовані в систему, починаються з великої букви, параметри функції задаються у квадратних дужках. Наприклад: **Sin[x]**, **ArcTan[x]**, **BesselJ[n, x]**;
- круглі дужки () використовуються при виділенні частин виразів для зміни послідовності їх розрахунку;
- фігурні дужки { } використовуються **тільки** для запису списків.

4. Списки та масиви. Записуються з допомогою фігурних дужок. Кожний елемент відділяється комою. Двовимірні масиви можуть бути записані у табличній (матричній) формі. Елементом масиву може бути будь-який тип даних (числовий, символьний, графічний чи звуковий об'єкт)

In[7]:=

$$\{1, x^2 + y^2, \sin^2[x]\} + \{2, z^2, \sin^2[x]\} - 4$$

$$\text{Out}[7] = \{-1, -4 + x^2 + y^2 + z^2, -4 + 2\sin^2[x]\}$$

In[8]:=

$$\text{TableForm}[\{\{a1, a2, a3\}, \{b1, b2, b3\}, \{c1, c2, c3\}\}]$$

[8]//TableForm=

a1	a2	a3
b1	b2	b3
c1	c2	c3

5. Об'єкти й ідентифікатори. У загальному випадку Mathematica оперує з об'єктами. Під ними розуміють математичні вирази (*expr*), символи (*symbol*), рядки символів (*strings*), числа різного типу, константи, змінні, графічні та звукові об'єкти. Кожному об'єкту можна присвоїти ім'я – **ідентифікатор**, яке повинне бути унікальним, тобто єдиним. У системі вбудовано декілька тисяч імен різних об'єктів. Існують наступні правила для присвоєння імен:

- ім'я може містити будь-яку кількість букв та цифр (навіть кирилицю), не містити пробілів та спеціальних символів;
- ім'я об'єкта, заданого користувачем, записуються малими буквами ;
- ім'я об'єктів, що входять в ядро системи (функції, константи) починаються з великої літери, а якщо ім'я складається з двох слів, то кожне слово з великої букви без пробілу між ними, наприклад **FullSimplify**;
- ім'я системного об'єкта починається зі знака \$, наприклад **\$MaxPrecision**;

Не заборонено задавати ім'я об'єкта користувача з великої букви, але таке ім'я може випадково бути зайнятим об'єктом, що входить в ядро системи і таке присвоєння буде проігнороване. Наприклад, імена **E** і **I** – заборонені, а **e** та **i** – дозволені.

6. Константи. Константи – типові елементи системи, що несуть наперед визначене числове чи символічне значення. Це значення не може змінюватись під час розрахунків. До констант належать самі числа (наприклад: 2, 8.3). Константи-числа не мають ідентифікаторів, оскільки саме число є його іменем і значенням водночас. Існує ряд іменованих констант, які можна розглядати як функції без аргумента, вони повертають задане у системі значення. Наприклад:

Infinity – додатна безмежність;

E – основа натурального логарифма (2.71828....);

I – уявна одиниця;

Pi – число пі, можна задавати і буквою **π** (3.1415....);

True – логічна одиниця, або "Так";

False – логічний нуль, або "Ні".

7. Змінні. Змінні в системі Mathematica є іменованими об'єктами, які в ході виконання програми можуть неодноразово набувати різних значень – числових, символічних, математичних виразів, графічних об'єктів чи масивів будь-якого типу.

Для присвоєння змінній якогось значення використовують *оператор присвоєння*, знак рівності (=), наприклад **a=12; y=a/2**. Існує ще один оператор присвоєння, який називають *відкладеним присвоєнням* (:=) , наприклад **z:=a/2**. У чому ж різниця? В першому випадку змінній **y** відразу присвоюється значення виразу **a/2**, тобто число **6**. У другому випадку

змінній **z** присвоюється вираз, який буде розраховуватись лише пізніше, тоді, коли зустрінеться змінна **z** у подальших виразах. Якщо перед використанням змінних **y** і **z** змінити значення змінної **a**, то значення змінної **y** залишиться без змін, а змінна **z** видасть результат з урахуванням зміни **a**. Крім того, операція *присвоєння* супроводжується видачею результату розрахованого виразу у вихідній комірці, а операція *відкладеного присвоєння* виконується без такого супроводу.

8. Оператори і функції. Оператори – елементи запису математичних виразів, які вказують на те, які дії виконуються над символічними чи числовими даними. Дані, що використовуються разом з операторами, називають *операндами*. Дія більшості операторів може бути замінена відповідною функцією, але це часто буває незручно. Нижче наведені основні оператори математичних виразів і відповідні їм функції:

```
x + y + z Plus[x, y, z]
a = b Set[a, b]
x y z Times[x, y, z]
x^n Power[x, n]
{a, b, c} List[a, b, c]
```

Логічні оператори і функції. Логічними називають операції, що виражають чисто логічні відповідності між даними. Для здійснення логічних операцій використовуються наступні логічні оператори і відповідні їм функції:

== Рівність (наприклад, **a == b**), функція **Equal[a,b]**;

!= Нерівність (**a != b**);

> Більше, функція **Greater[x,y]**;

>= Більше або рівно, функція **GreaterEqual[x,y]**;

< Менше

<= Менше або рівно

! Логічне заперечення (наприклад, **!a**), функція **Not[x]**;

&& Логічне множення, функція **And[p,q]**

|| Логічне додавання, функція **Or[p,q]**.

Функції користувача. Для створення функцій користувача використовують такі конструкції:

fun[x_] := x^2 - відкладене задання функції користувача **fun**;

fun[x_] = x^2 - невідкладне задання функції користувача **fun**;

У позначеннях типу **x_** знак **_** застосовується для створення так званих *зразків*, що описують аргументи як локальні змінні в тілі функції. Після *зразка* можна вказати тип даних, які можна використовувати як аргумент функції. Наприклад, у функціях **fun1[x_Integer]**, **fun2[x_List]** аргументами можуть бути лише цілі числа та масив відповідно. Без використання *зразків* функція користувача не буде працювати.

При описі функції користувача багатьох змінних можна задати значення деяких аргументів *за замовчуванням*. При виклику значення функції у якійсь точці можна не задавати значення параметрів, якщо вони збігаються з їхніми значеннями за замовчуванням. Наприклад, задамо **0** як значення третього аргументу за замовченням.

```
f[x_, y_, z_ : 0] := x^2 + y^2 + z^2;
{f[2], f[2, 0], f[2, 0, 0]}
```

```
{f[2], 4, 4}
```

З результату видно, що третій аргумент функції є вже не обов'язковим, тоді як без другого аргументу функція не може повернути значення.

Як створювати функцію користувача за допомогою *чистої функції*, буде показано пізніше.

Оперативна довідка по функції, її аргументах та опціях.

Повний список об'єктів, заданих в ядрі системи, легко отримати, використовуючи команду **?***. Для отримання списку об'єктів, що починаються з букв **Arc**, – команда **?Arc*** і так далі.

? Name – довідка по заданому слову **Name**.

?? Name – розширена довідка по заданому слову **Name**.

Options[Name] – довідка про опції об'єкта **Name**

Наприклад, викличемо довідку по функції **ArcTan[x]**

```
? ArcTan
```

ArcTan[z] gives the arc tangent $\tan^{-1}(z)$ of the complex number z.

ArcTan[x, y] gives the arc tangent of $\frac{y}{x}$, taking into account which quadrant the point (x,

9. Підстановки. Важливе значення в числових та символічних перетвореннях мають операції *підстановки* (**rules**). Їх зміст полягає в

заміні одного об'єкта, чи його частини другим об'єктом або частиною другого об'єкта.

Операція підстановки виконується за допомогою комбінації символів **"/."**:

$$\frac{1}{x+1} /. x \rightarrow \sin[x]$$

$$\frac{1}{1 + \sin[x]}$$

Для використання розв'язків що отримуються у вигляді масивів у подальших розрахунках потрібно виконати підстановки, над виразом, що містить список невідомих, відносно яких знайдено розв'язки.

10. Генерація списків та виділення елементів списків.

Для генерації списків з елементами будь-якого типу використовують функцію **Table**:

Table[object, {imax}] – генерує список, що містить **imax** екземплярів об'єкта **object**;

Table[object, {i, imax}] – генерує список значень об'єкта **object** при **i**, що змінюється від **1** до **imax**;

Table[object, {i, imin, imax}] – генерує список значень об'єкта **object** при **i**, що змінюється від **imin** до **imax**;

Table[object, {i, imin, imax, di}] – генерує список значень об'єкта **object** при **i**, що змінюється від **imin** до **imax** з кроком **di**;

Table[object, {i, imin, imax, di}, {j, jmin, jmax, dj}, ...] – генерує багатовимірний масив;

```
spys = Table[i^j, {i, 1, 3}, {j, 1, 3}]
```

```
{{1, 1, 1}, {2, 4, 8}, {3, 9, 27}}
```

```
MatrixForm[spys]
```

```
MatrixForm=
```

```
( 1 1 1
  2 4 8
  3 9 27 )
```

Для виділення елементів списку використовують подвійні квадратні дужки. Наприклад, другий рядок масиву **spys** :

```
spys[[2]]
{2, 4, 8}
```

Перший елемент першого рядка:

```
= spys[[1, 1]]
1
```

Аналогом оператора `[[ ]]` є функція **Part**. Функція *Part*, як і оператор `[[ ]]`, можуть використовуватись для вибору заданого елемента із символьного виразу. Щоб продемонструвати це замінимо один елемент числового масиву символьним виразом:

```
spys[[3, 2]] = x^2 + y^2
MatrixForm[spys]
```

$x^2 + y^2$

```
MatrixForm=
( 1 1 1 )
( 2 4 8 )
( 3 x^2 + y^2 27 )
```

Виділимо другий елемент виразу:

```
: spys[[3, 2, 2]]
y^2
```

11. Чисті та анонімні функції.

Чистою функцією називають функцію, яка не має імені. Така функція, як правило, використовується один раз, відразу після її створення. Ця функція зображується тільки виразом без ідентифікатора, звідси і її назва – чиста функція (*pure function*). Використання *чистої функції* є одним з методів створення функції без використання *зразків*. Для створення такого об'єкта служить вбудована функція **Function**, яка використовується в одному з наступних варіантів:

Function[body] – створює чисту функцію з тілом **body**;

Function[{x}, body] – створює чисту функцію параметра **x** з тілом **body**;

Function[{x1, x2, ... }, body] – створює чисту функцію декількох параметрів **x1, x2,...** з тілом **body**;

Для розрахунку такої функції після неї задається список параметрів у квадратних дужках. Наприклад:

```
Function[{x}, x^2]
```

```
Function[{x}, x^2]
```

Чиста функція створена. Тепер розрахуємо її з конкретним параметром:

```
In[2]:= Function[{x}, x^2][5]
```

```
Out[2]= 25
```

Чисту функцію легко перетворити у функцію користувача

```
In[3]:= fun = Function[{x}, x^2]
```

```
Out[3]= Function[{x}, x^2]
```

Перевіримо, чи працює функція користувача

```
{fun[6], fun[10]}
{36, 100}
```

Замість функції **Function[]**, можна скористатись простішим записом чистої функції з допомогою оператора **&** та знака **#**, який позначає місце, куди підставляти аргумент функції. Такий запис називають анонімною функцією. Створимо функцію користувача таким методом:

```

fun = (#^2 + Sin[#]) &

: #1^2 + Sin[#1] &

fun[x] / fun[y]
fun[Pi / 4]

:  $\frac{x^2 + \sin[x]}{y^2 + \sin[y]}$ 

:  $\frac{1}{\sqrt{2}} + \frac{\pi^2}{16}$ 

```

Для випадку функції користувача багатьох змінних

```

fun1 = (#1^2 Sin[#2] Cos[#3]) &

: #1^2 Sin[#2] Cos[#3] &

fun1[r, 2, 1]

: r^2 Cos[1] Sin[2]

```

12. Методи підстановки параметрів для функції.

У системі *Mathematica*, крім стандартного методу виклику функції з її аргументами, наприклад **fun[a, b, c]**, існує ряд додаткових можливостей, які часто використовуються у функціональному програмуванні.

Ці додаткові можливості організовані у вигляді функцій **Apply**, **Map** та різних їх скорочених форм. Розглянемо їх форми запису та результати їх дії.

Скорочена форма	Повна форма	Приклад	Отриманий результат
f @ expr	f[expr]	f@{a, b, c}	f[{a, b, c}]
f @@ expr	Apply[f, expr]	f@@{a, b, c}	f[a, b, c]
f /@ expr	Map[f, expr]	f/@{a, b, c}	{f[a], f[b], f[c]}
f //@ expr	MapAll[f, expr]	f//@{a, b, c}	f[{f[a], f[b], f[c]}]

ОСНОВИ ПРОГРАМУВАННЯ

1. Використання процедур. В основі процедурного програмування лежить поняття *процедури* і типових засобів керування процесом виконання програми – *циклів, умовних та безумовних переходів*.

Процедури – повністю самостійні програмні модулі, які задаються своїми іменами і ототожнюються з виконанням деякої послідовності операцій, які можуть бути задані в одному рядку з використанням як роздільника - символу ”;”.

Для опису процедури із списком локальних змінних {a,b, ...} і тілом *procedure* можуть використовуватись функції

Block[{x, y, ...}, procedure]

Module[{x, y, ...}, procedure].

Ці функції повертають результат обчислення останнього в їх тілі виразу.

Розглянемо приклад побудови функції, яка буде повертати рівняння Лагранжа. Аргументами цієї функції будуть кінетична, потенціальна енергії і узагальнена сила. Будемо вважати, що система описується лише однією узагальненою координатою **x[t]**.

Використаємо функцію взяття похідної **D[f[x], x]**

```

LagrEq[T_, U_, Q_] := Module[{L}, L = T - U;
D[D[L, D[x[t], t]], t] == Q + D[L, x[t]];

```

Локальною змінною у процедурі **Module** є лише **L** – функція Лагранжа. У тілі процедури міститься лише два вирази розділені знаком ; . Результат розрахунку останнього виразу повертається як результат функції **LagrEq** . Тепер задаючи різні вирази кінетичної, потенціальної енергії та узагальненої сили, виражені через узагальнену координату **x[t]** можна автоматично отримати різні рівняння Лагранжа, тобто рівняння руху. Наприклад для гармонійного осцилятора із затуханням отримаємо

```

T = m x'[t]^2 / 2;
U = K x[t]^2 / 2;
Q = -c x'[t];
LagrEq[T, U, Q]

```

```

m x''[t] == -K x[t] - c x'[t]

```

2. Організація циклів. Багато задач у системі *Mathematica* розв'язуються з використанням лінійних алгоритмів і програм. Вони можуть бути записані неперервним рядком виразів, що виконуються послідовно від початку до кінця.

Проте в більшості випадків серйозні обчислення базуються на використанні циклічних і розгалужених алгоритмів та програм. При цьому, в залежності від проміжних чи початкових даних, обчислення можуть йти по різних гілках програми, циклічно повторюватись і т.д.

Для організації циклів використовують управляючі структури **Do**, **For**, **While**.

```
Do[expr, imax]
Do[expr, {i, imax}]
Do[expr, {i, imin, imax}]
Do[expr, {i, imin, imax, di}]
Do[expr, {i, imin, imax}, {j, jmin, jmax}, ... ]
```

Приклад циклу з виводом значення локальної змінної **i**

```
Do[Print[i], {i, 1, 3}]
1
2
3
```

Приклад побудови ланцюгового дробу на основі циклу **Do**

```
a = {a1, a2, a3, a4, a5};
x = y; Do[x = 1 / (1 + a[[k]] x), {k, 1, 5}]; x
```

$$1 + \frac{1}{1 + \frac{a_5}{1 + \frac{a_4}{1 + \frac{a_3}{1 + \frac{a_2}{1 + a_1 y}}}}}$$

Організація циклу з допомогою структури **For**

```
For[i = 1, i ≤ 3, Print[i]; i++]
```

```
1
```

```
2
```

```
3
```

Організація циклу з допомогою структури **While**

```
nn = 7; While[nn > 5, nn = nn - 1; Print[nn]]
```

```
6
```

```
5
```

У вказаних типах циклів і в інших керуючих структурах можна використовувати такі директиви-функції:

Abort[] – зупиняє розрахунок з повідомленням \$Aborted;

Break[] – виконує вихід із тіла циклу або рівня вкладеності програми, що містить даний оператор;

Continue[] – задає перехід на наступний крок даного циклу;

Interrupt[] – перериває розрахунки з можливістю їх відновлення;

Return[expr] – перериває виконання з виведенням значення виразу **expr**;

3. Умовні вирази. Умовні вирази задаються з допомогою функції **If**, форма її має вигляд

If[condition, t, f] – повертає **t**, якщо результат розрахунку **condition** дає **True** і **f**, якщо результат рівний **False**.

If[condition, t, f, u] – те саме, але повертає **u** у разі, якщо результат розрахунку **condition** не дає ні **True**, ні **False**.

```
a = Pi; b = E; If[a > b, x^2 + y^2, z]
```

```
x == y
```

```
x^2 + y^2
```

```
x == y
```

```
If[x == y, a, b, c]
```

```
c
```

4. Трасування програмних модулів. Щоб переглянути проміжні операції, які здійснює система, використовують засоби *трассування*. Розглянемо функцію **Trace**

`Trace[Log[Sin[Pi / 4]]]`

$$\left\{ \left\{ \left\{ \frac{1}{4}, \frac{1}{4} \right\}, \frac{\pi}{4}, \frac{\pi}{4} \right\}, \sin\left[\frac{\pi}{4}\right], \frac{1}{\sqrt{2}}, \log\left[\frac{1}{\sqrt{2}}\right], -\frac{\log[2]}{2} \right\}$$

Результат трасування видається вкладеними списками, що мають два елемента – вираз, що розраховується, і результат розрахунку.

ОПЕРАЦІЇ МАТЕМАТИЧНОГО АНАЛІЗУ

1. Суми та добутки. Для розрахунку сум в системі Mathematica передбачена функція `Sum`, яка може містити різну кількість аргументів

`Sum[f, {i, imax}]` – розраховує суму значень `f` при зміні індексу `i` від 1 до `imax` з кроком +1.

`Sum[f, {i, imin, imax}]` – розраховує суму значень `f` при зміні індексу `i` від `imin` до `imax` з кроком +1. Може бути замінена оператором суми. Для `f=i`:

$$\sum_{i=imin}^{imax} i$$

$$\text{Out[5]} = \frac{1}{2} (1 + imax - imin) (imax + imin)$$

`Sum[f, {i, imin, imax, di}]` – розраховує суму значень `f` при зміні індексу `i` від `imin` до `imax` з кроком `di`.

`Sum[f, {i, imin, imax}, {j, jmin, jmax, dj}, ....]` – розраховує багатократну суму при зміні індексів `i, j, ...`.

Якщо сума збіжна, то можна сумувати до безмежності. Наприклад, розклад експоненти система "впізнає"

$$\sum_{n=0}^{\infty} \frac{(x+y)^n}{n!}$$

$$: e^{x+y}$$

У розкладі синуса система повертає спрощений вираз

$$\sum_{n=1}^{\infty} (-1)^{n+1} \frac{x^{2n-1}}{(2n-1)!}$$

`Sin[x]`

Інколи для спрощення необхідно використати функцію спрощення виразів `FullSimplify[]` наприклад:

`FullSimplify[x2 + 2 * x * y + y2]`

$$(x + y)^2$$

Сума геометричної прогресії

$$\text{summa1} = \sum_{i=0}^{\infty} a * q^i$$

$$\text{Out[2]} = \frac{a}{1 - q}$$

Тут необхідно звернути увагу на те, що Mathematica видає вірний результат лише для `q < 1`, хоча не зауважує про це.

Для розрахунку сум в числовому вигляді використовують функцію `Nsum`, яка еквівалентна запису `N[Sum[...]]`.

Для розрахунку добутків в системі Mathematica передбачена функція `Product`.

`Product[f, {i, imax}]` – розраховує добуток значень `f` при зміні індексу `i` від 1 до `imax` з кроком +1.

`Product[f, {i, imin, imax}]` – розраховує добуток значень `f` при зміні індексу `i` від `imin` до `imax` з кроком +1. Може бути замінена оператором добутку. Для `f=i`:

$$\prod_{i=1}^n i$$

`Out[3] = n!`

`Product[f, {i, imin, imax, di}]` – розраховує добуток значень `f` при зміні індексу `i` від `imin` до `imax` з кроком `di`.

2. Похідні першого та вищих порядків. До найбільш вживаних математичних операцій належать розрахунки похідних. Для цього використовують наступні функції:

`D[f[x], x]` – повертає частинну похідну функції `f[x]` по змінній `x`. Можна скористатися операторами похідної, наприклад:

```
{D[Sin[x], x], Sin'[x], D[Sin[x], x]}
{Cos[x], Cos[x], Cos[x]}
```

Всі три записи дають однакові результати.

`D[f[x], {x, n}]` – повертає частинну похідну n-го порядку функції `f[x]` по змінній `x`. Цю функцію можна замінити n-кратним повторенням попередньої функції.

! Завдання. Розрахуйте похідну від виразу $x^2 \operatorname{tg}(x) e^x \ln(x)$, всіма трьома методами. (Для другого методу розрахунку похідної створіть функцію користувача).

3. Розрахунок інтегралів. Для інтегрування в системі Mathematica використовуються наступні функції

`Integrate[f, x]` – розраховує неозначений інтеграл від функції `f` за аргументом `x`.

`Integrate[f, {x, xmin, xmax}]` – розраховує означений інтеграл з межами від `xmin` до `xmax`

`Integrate[f, {x, xmin, xmax}, {y, ymin, ymax} ...]` – розраховує n-вимірний означений інтеграл у межах від `xmin` до `xmax` за змінною `x`, від `ymin` до `ymax` за змінною `y`, і т. д.

Для наглядності розрахунку інтегралів можна скористатися операторним виглядом:

$$\int \int x^3 y^2 dx dy$$

$$\frac{x^4 y^3}{12}$$

У випадку, коли інтеграл у символьному вигляді розрахувати не вдається, можна скористатися функцією для числового розрахунку інтегралів: `NIntegrate[f, {x, xmin, xmax}]`.

Розглянемо такий приклад:

```
Integrate[Exp[Sin[x]], {x, 0, 1}]
```

$$\int_0^1 e^{\sin[x]} dx$$

```
NIntegrate[Exp[Sin[x]], {x, 0, 1}]
```

```
1.63187
```

! Завдання. Розрахуйте неозначений інтеграл від виразу, що отримали в попередньому завданні. Розрахуйте означений інтеграл від того ж виразу у межах $[0, 1]$ за допомогою функцій аналітичного та чисельного інтегрувань і порівняйте результати.

4. Розв'язування рівнянь та системи рівнянь.

Для розв'язку рівнянь у символьному вигляді використовуються функції `Solve` і `Roots`. Для знаходження розв'язків чисельними методами використовуються функції `NSolve` і `FindRoot`.

Розв'язування рівнянь та систем за допомогою функції `Solve[eqns, vars]`,

`eqns` – рівняння, або система рівнянь, в яких як знак рівності використовується логічний оператор рівності (знак `=`). Система рівнянь задається у вигляді списку, тобто одновимірного масиву.

`vars` – змінна або список змінних, відносно яких розв'язується задана система рівнянь.

Функція `Solve[eqns, vars, elims]` розв'язує рівняння `eqns` за змінними `vars`, виключаючи змінні `elims`.

Приклади

```
Solve[a x^2 + b x + c == 0, x]
```

$$\left\{ \left\{ x \rightarrow \frac{-b - \sqrt{b^2 - 4ac}}{2a} \right\}, \left\{ x \rightarrow \frac{-b + \sqrt{b^2 - 4ac}}{2a} \right\} \right\}$$

```
Solve[x^3 + x^2 + x + 1 == 0, x]
```

$$\{\{x \rightarrow -1\}, \{x \rightarrow -i\}, \{x \rightarrow i\}\}$$

Із даних прикладів видно, що розв'язки рівнянь видаються у вигляді підстановок у двовимірному масиві (подвійні фігурні дужки). Для використання розв'язків у подальших розрахунках потрібно значення розв'язків присвоїти змінним. Зробимо це, виконавши підстановки над виразом, що містить змінну `x`, відносно якої знайдено розв'язки рівняння, і збережемо результати у масиві змінних `{x1, x2}`.

$$\{x1, x2\} = x /. \left\{ \left\{ x \rightarrow \frac{-b - \sqrt{b^2 - 4ac}}{2a} \right\}, \left\{ x \rightarrow \frac{-b + \sqrt{b^2 - 4ac}}{2a} \right\} \right\}$$

$$\left\{ \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \frac{-b + \sqrt{b^2 - 4ac}}{2a} \right\}$$

Далі ці розв'язки можна використовувати у подальших виразах:

`FullSimplify[x1 + x2]`

$$-\frac{b}{a}$$

Серед запропонованих розв'язків можуть бути і хибні, тому необхідно здійснювати перевірку коренів.

`soll = Solve[Cos[x]^2 + a Sin[x] Cos[x] == 0, x]`

$$\left\{ \left\{ x \rightarrow -\frac{\pi}{2} \right\}, \left\{ x \rightarrow \frac{\pi}{2} \right\}, \left\{ x \rightarrow -\text{ArcCos}\left[-\frac{a}{\sqrt{1+a^2}}\right] \right\}, \right. \\ \left. \left\{ x \rightarrow \text{ArcCos}\left[-\frac{a}{\sqrt{1+a^2}}\right] \right\}, \left\{ x \rightarrow -\text{ArcCos}\left[\frac{a}{\sqrt{1+a^2}}\right] \right\} \right\} \\ \left\{ x \rightarrow \text{ArcCos}\left[\frac{a}{\sqrt{1+a^2}}\right] \right\} \right\}$$

Перевіримо шостий корінь

`FullSimplify[(Cos[x]^2 + a Sin[x] Cos[x]) /. soll[[6]]]`

$$a^2 \left(\frac{1}{1+a^2} + \frac{\sqrt{\frac{1}{1+a^2}}}{\sqrt{1+a^2}} \right)$$

Видно, що отриманий вираз не дорівнює нулю, а отже, шостий корінь хибний.

Для розв'язку системи рівнянь задамо масив рівнянь eqns і розв'яжемо їх відносно списку змінних {x, y}:

`eqns = {x == 1 + 2 a y, y == 9 + 2 x};`
`Solve[eqns, {x, y}]`

$$\left\{ \left\{ x \rightarrow -\frac{1+18a}{-1+4a}, y \rightarrow -\frac{11}{-1+4a} \right\} \right\}$$

Для розв'язування системи лінійних рівнянь застосовують функцію `LinearSolve[ ]`, яка описана нижче у пункті *операції векторного та тензорного аналізу*.

Функція `Roots` використовується для знаходження символьних розв'язків поліноміальних рівнянь.

`kor = Roots[x^3 + x^2 + x + 1 == 0, x]`

$$x == i \mid x == -i \mid x == -1$$

Із даного прикладу видно, що розв'язки рівняння видаються у вигляді логічного виразу, з якого можна виділити кожний корінь окремо

`kor[[2]]`

$$x == -i$$

Для числового розв'язку рівнянь використовують функцію `NSolve`.

`NSolve[3 + 3 x - 7 x^2 - x^3 + 2 x^4 + 3 x^7 - 3 x^8 - x^9 + x^10 == 0, x]`

$$\left\{ \left\{ x \rightarrow -1.73205 \right\}, \left\{ x \rightarrow -0.868688 - 0.585282 i \right\}, \right. \\ \left\{ x \rightarrow -0.868688 + 0.585282 i \right\}, \left\{ x \rightarrow -0.496292 \right\}, \\ \left\{ x \rightarrow 0.0763556 - 1.14095 i \right\}, \left\{ x \rightarrow 0.0763556 + 1.14095 i \right\}, \left\{ x \rightarrow 1. \right\}, \\ \left\{ x \rightarrow 1.04048 - 0.56735 i \right\}, \left\{ x \rightarrow 1.04048 + 0.56735 i \right\}, \left\{ x \rightarrow 1.73205 \right\} \right\}$$

`soll = NSolve[Cos[x]^2 + Sin[x] Cos[x] == 0, x]`

$$\left\{ \left\{ x \rightarrow -1.5708 \right\}, \left\{ x \rightarrow -0.785398 \right\}, \left\{ x \rightarrow 1.5708 \right\}, \left\{ x \rightarrow 2.35619 \right\} \right\}$$

Якщо не вдається знайти розв'язки рівнянь описаними функціями, то використовують функцію `FindRoot`, яка знаходить розв'язки рівняння чи системи рівнянь будь-якої складності числовим методом. Але при цьому необхідно вказати початкові точки, починаючи з яких система шукає розв'язки. Функція `FindRoot` видає лише один знайдений розв'язок. Часто результат такого пошуку залежить від заданої початкової точки.

`FindRoot[lhs == rhs, {x, x0}]` - шукає розв'язок рівняння починаючи з початкової точки $x = x_0$

`FindRoot[lhs == rhs, {x, x0, xmin, xmax}]` - шукає розв'язок рівняння $lhs = rhs$, починаючи з початкової точки $x = x0$, не виходячи за межі інтервалу $xmin, xmax$.

Приклади:

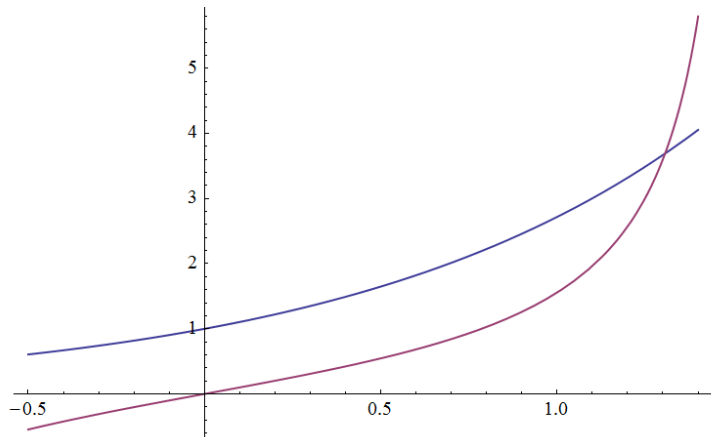
```
FindRoot[Exp[x] == Tan[x], {x, 1.0}]
```

```
{x -> 1.30633}
```

Для визначення початкових точок зручно користуватись графічним методом розв'язування рівняння.

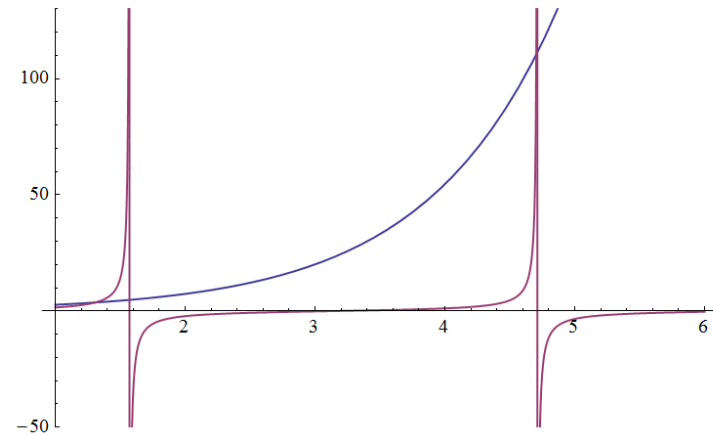
Побудова графіків здійснюється з допомогою функції `Plot`. (опис функцій побудови графіків приведений у наступному розділі)

```
Plot[{Exp[x], Tan[x]}, {x, -0.5, 1.4}]
```



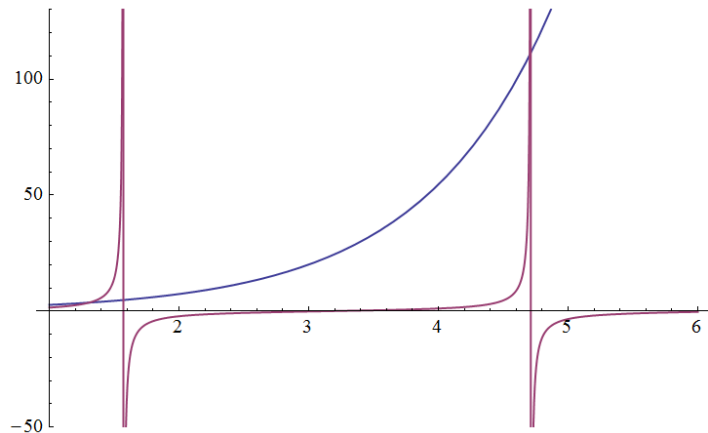
На графіку точка перетину двох функцій відповідає розв'язку рівняння. Але цей розв'язок не єдиний. Побудуємо продовження цих графіків

```
Plot[{Exp[x], Tan[x]}, {x, 1, 6}, PlotRange -> {-50, 130}]
```



На графіку видно ще одну точку перетину графіків $x=4,7$ (перетин з вертикальною лінією у точці $x=1,5$, яка відповідає точці розриву 2-го роду для функції $\tan(x)$ не відповідає розв'язку рівняння). Опція `PlotRange` задає область зміни значень функції, яка зображається на графіку. Без цієї опції система пробує помістити на графік усю область зміни функції, яку вона обчислила, і для розривних функцій отримуємо графік в дрібному масштабі. Щоб точніше з графіка визначити початкову точку $x0$ для знаходження розв'язку рівняння, зручно будувати графік різниці функцій. Тоді розв'язок рівняння є точка перетину графіка з віссю OX .

```
Plot[{Exp[x], Tan[x]}, {x, 1, 6}, PlotRange -> {-50, 130}]
```



На графіку видно дві точки перетину графіка з віссю ОХ. Розв'язком рівняння буде лише ліва точка, так як права точка є точкою розриву другого роду, тобто перехід від $-\infty$ до $+\infty$. Знайдемо другий розв'язок, задавши початкову точку $x_0=4.7$.

```
FindRoot[Exp[x] == Tan[x], {x, 4.7}]
```

```
{x -> 4.70332}
```

Продемонструємо, наскільки важливо задати близьке до кореня значення x_0 . Послідовно задамо значення $x_0=4.6$ та $x_0=4.8$:

```
FindRoot[Exp[x] == Tan[x], {x, 4.6}]
```

```
{x -> -9.4247}
```

```
FindRoot[Exp[x] == Tan[x], {x, 4.8}]
```

```
{x -> -12.5664}
```

У першому випадку методом Ньютона випадково знайдено один з коренів рівняння далеко від початкової точки, у другому - методом Ньютона після 15 ітерацій не знайдено корінь трансцендентного рівняння і видано останню точку, на якій припинився пошук.

Для збільшення кількості ітерацій, що виконуються при знаходженні коренів, збільшують значення опції MaxIterations. Для збільшення точності отриманого кореня збільшують значення опції WorkingPrecision.

```
FindRoot[Exp[x] == Tan[x], {x, 4.6}, MaxIterations -> 25,  
WorkingPrecision -> 20]
```

```
{x -> -9.4246972547387840782}
```

Щоб отримати повну інформацію про функції Plot та FindRoot, використайте такі звертання: ??Plot та ??FindRoot

5. Розв'язування диференціальних рівнянь. Розв'язування диференціальних рівнянь в аналітичному вигляді виконують за допомогою функції Dsolve.

Dsolve[eqn, y[x], x] - розв'язується диференціальне рівняння eqn відносно функції y, яка залежить від аргументу x;

Dsolve[{eqns1, eqns2, ...}, {y1[x], y2[x], ...}, x] - розв'язується система диференціальних рівнянь відносно списку функцій, які залежать від аргументу x;

Dsolve[eqn, y[x1, x2, ...], {x1, x2, ...}] - розв'язується система диференціальних рівнянь відносно списку функцій, які залежать від аргументу x.

Розв'яжемо диференціальне рівняння 2-го порядку

```
DSolve[y''[x] == y'[x] + y[x], y[x], x]
```

$$\left\{ \left\{ y[x] \rightarrow e^{\left(\frac{1}{2} - \frac{\sqrt{5}}{2}\right)x} C[1] + e^{\left(\frac{1}{2} + \frac{\sqrt{5}}{2}\right)x} C[2] \right\} \right\}$$

У отриманий загальний розв'язок входять дві невідомі постійні $C[1]$ та $C[2]$, які можна знайти, задавши додаткові умови на функцію та її похідну. Такими умовами будуть граничні чи початкові умови у залежності від фізичної задачі.

Деколи зручно відразу задати ці умови і знайти розв'язок, що задовольняє граничні чи початкові умови. Розв'яжемо диференціальне рівняння з урахуванням граничних умов і виконаємо спрощення отриманих розв'язків

```
DSolve[{y'[x] == y'[x] + y[x], y'[0] == dy0, y[0] == y0}, y[x], x]
{{y[x] -> 1/10 (-2 Sqrt[5] dy0 E^(1/2 - Sqrt[5]/2 x) + 2 Sqrt[5] dy0 E^(1/2 + Sqrt[5]/2 x) + 5 E^(1/2 - Sqrt[5]/2 x) y0 +
Sqrt[5] E^(1/2 - Sqrt[5]/2 x) y0 + 5 E^(1/2 + Sqrt[5]/2 x) y0 - Sqrt[5] E^(1/2 + Sqrt[5]/2 x) y0)}}
```

Зауваження. Для складних диференціальних рівнянь не рекомендується задавати декілька таких операцій одночасно, бо коли система довго не видає результату, неможливо встановити, яка з цих операцій виявилась надто складною. Лише перевіривши можливість виконання кожної з цих операцій, можна виконувати їх одночасно. Бувають випадки, коли операція FullSimplify набагато складніша за розв'язування диференціального рівняння. У такому разі можна обійтися і без спрощення або спростувати розв'язок частинами.

Функція DSolve може видавати розв'язок у вигляді підстановки з чистою функцією, якщо, зазначаючи відносно якої функції розв'язуємо рівняння не вказати у квадратних дужках її аргумент. Наприклад, порівняй два записи і результати, що їм відповідають

```
rozv1 = DSolve[{y'[x] == y[x], y[0] == 1}, y, x]
{{y -> Function[{x}, e^x]}}

rozv2 = DSolve[{y'[x] == y[x], y[0] == 1}, y[x], x]
{{y[x] -> e^x}}
```

Перший результат дозволяє створити іменовану функцію користувача без використання *зразків*

```
fun1 = y /. rozv1[[1]]
Function[{x}, e^x]
```

Продемонструємо як отримати функцію користувача без використання оператора підстановки "/.".

Для цього використаємо той факт, що вираз {{y -> Function[{x}, e^x]}} складається з двох частин "y" та "Function[{x}, e^x]". Щоб отримати другу частину цього виразу, який є

елементом двовимірного масиву rozv, задамо додатковий третій індекс, що дорівнює 2.

```
fun2 = rozv1[[1, 1, 2]]
Function[{x}, e^x]
Перевіримо, чи функції, побудовані різними методами, однакові
fun1[x] == fun2[x]
True
```

Отже обидва методи дають однаковий результат. Яким методом використовувати – справа смаку автора.

Для числового розв'язування диференціальних рівнянь застосовують функцію NDSolve.

NDSolve[eqns, y, {x, xmin, xmax}] – Знаходить числовий розв'язок диференціального рівняння на інтервалі від xmin до xmax за допомогою інтерполяційної функції. Такий розв'язок не можна подати у аналітичному вигляді. Його можна подати у вигляді таблиці, чи зобразити на графіку.

Для знаходження числового розв'язку обов'язково разом з диференціальним рівнянням задаються граничні чи крайові умови.

```
Розглянемо наступний приклад
solution1 = NDSolve[{y'[x] == y[x], y[1] == 2}, y[x], {x, 0, 3}]
{{y[x] -> InterpolatingFunction[{{0., 3.}}, <>][x]}}
```

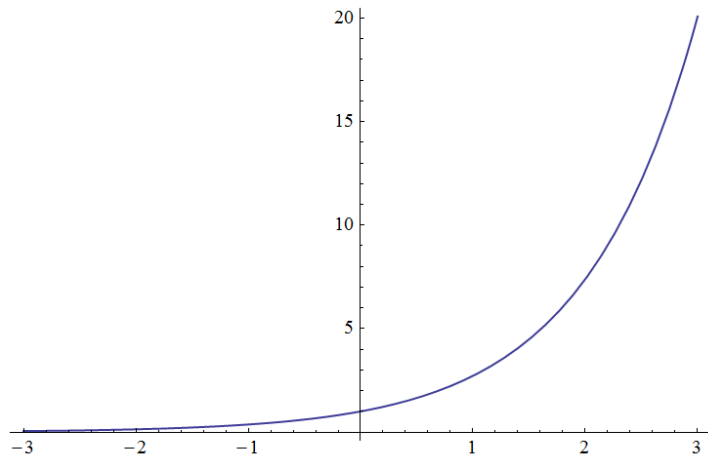
Розв'язок отримано. Але як ним тепер скористатись? Спочатку виділимо перший елемент масиву розв'язків (тобто побудуємо зовнішньої пари фігурних дужок), потім на основі одновимірного масиву solution1[[1]] виконаємо підстановку і результат присвоїмо деякій функції fun1[x], яку опишемо за всіма правилами, використовуючи знак зразка "_".

```
fun1[x_] = y[x] /. solution1[[1]]
InterpolatingFunction[{{0., 3.}}, <>][x]
Перевіримо, чи "працює" утворена нами функція, розрахуємо її
значення в точці x=0
fun1[0.5]
```

```
1.64872
```

Побудуємо графік отриманого розв'язку

```
Plot[fun1[x], {x, -3, 3}]
```



Отже функція, яка є результатом чисельного розв'язку диференціального рівняння - відома.

6. Операції векторного та тензорного аналізу у різних системах координат. Ядро системи Mathematica містить такі основні функції для дій над векторами та тензорами:

Dot[a, b], або a.b - скалярний добуток векторів a і b або добуток двох матриць a і b;

Cross[a, b] - векторний добуток двох векторів;

Inverse[m] – побудова оберненої матриці;

Transpose[m] – побудова транспонованої матриці;

Det[m] – розрахунок визначника квадратної матриці;

Tr[m] – розрахунок сліду квадратної матриці;

Eigenvalues[m] – знаходження власних значень квадратної матриці;

Eigenvectors[m] – знаходження власних векторів квадратної матриці;

Eigensystem[m] – знаходження власних значень та власних векторів квадратної матриці;

LinearSolve[m,b] – знаходження розв'язків лінійної системи рівнянь $m \cdot x = b$

Для дій над векторами у різних системах координат існує пакет розширення 'VectorAnalysis', який ініціюється так

```
<<Calculus`VectorAnalysis`
```

і містить наступні функції та змінні:

CoordinateSystem – повертає назву активної системи координат;

Coordinates[] - повертає координати активної системи координат;

Coordinates[system] - повертає координати системи координат system;

Змінна system може набувати таких значень:

Bipolar Bispherical

Cartesian ConfocalEllipsoidal

ConfocalParaboloidal Conical

Cylindrical EllipticCylindrical

OblateSpheroidal ParabolicCylindrical

Paraboloidal ProlateSpheroidal

Spherical Toroidal

Coordinates[system] – установлює систему координат system активною;

Coordinates[system] – установлює систему координат system активною;

Coordinates[system[vars]] - установлює систему координат system активною з координатними змінними vars ;

CoordinateRanges[system] - повертає межі змін координат системи system;

CoordinatesToCartesian[point, system] – повертає координати точки point з системи system в декартову систему координат;

CoordinatesFromCartesian[point, system] – повертає координати точки point з декартової системи координат в систему координат system;

DotProduct[a, b], або a.b - скалярний добуток векторів a і b або добуток двох матриць a і b;

CrossProduct[a, b] - векторний добуток двох векторів;

JacobianMatrix[system] – матриця Якобі у системі координат system.

JacobianDeterminant[system] – визначник Якобі у системі координат system;

Div[f] – дивергенція функції f в активній системі координат;

Curl[f] - ротор функції f;

Grad[f] - градієнт функції f;

Laplacian[f] – оператор Лапласа від функції f;

Розглянемо приклади використання наведених функцій над векторами $a = \{a1, a2, a3\}$, $b = \{b1, b2, b3\}$ і матрицею $m = \{\{m11, m12\}, \{m21, m22\}\}$

```

a = {a1, a2, a3};
b = {b1, b2, b3};
m = {{m11, m12}, {m21, m22}};
a.b

a1 b1 + a2 b2 + a3 b3
Cross[a, b]

{-a3 b2 + a2 b3, a3 b1 - a1 b3, -a2 b1 + a1 b2}

```

Inverse[m]

$$\left\{ \left\{ \frac{m22}{-m12 m21 + m11 m22}, -\frac{m12}{-m12 m21 + m11 m22} \right\}, \left\{ -\frac{m21}{-m12 m21 + m11 m22}, \frac{m11}{-m12 m21 + m11 m22} \right\} \right\}$$

Transpose[m]

```
{{m11, m21}, {m12, m22}}
```

Det[m]

```
-m12 m21 + m11 m22
```

Tr[m]

```
m11 + m22
```

Eigenvalues[m]

$$\left\{ \frac{1}{2} \left(m11 + m22 - \sqrt{m11^2 + 4 m12 m21 - 2 m11 m22 + m22^2} \right), \frac{1}{2} \left(m11 + m22 + \sqrt{m11^2 + 4 m12 m21 - 2 m11 m22 + m22^2} \right) \right\}$$

Eigenvectors[m]

$$\left\{ \left\{ -\frac{-m11 + m22 + \sqrt{m11^2 + 4 m12 m21 - 2 m11 m22 + m22^2}}{2 m21}, 1 \right\}, \left\{ -\frac{-m11 + m22 - \sqrt{m11^2 + 4 m12 m21 - 2 m11 m22 + m22^2}}{2 m21}, 1 \right\} \right\}$$

Тепер власні значення та власні вектори в одному списку

Eigensystem[m]

$$\left\{ \left\{ \frac{1}{2} \left(m11 + m22 - \sqrt{m11^2 + 4 m12 m21 - 2 m11 m22 + m22^2} \right), \frac{1}{2} \left(m11 + m22 + \sqrt{m11^2 + 4 m12 m21 - 2 m11 m22 + m22^2} \right) \right\}, \left\{ \left\{ -\frac{-m11 + m22 + \sqrt{m11^2 + 4 m12 m21 - 2 m11 m22 + m22^2}}{2 m21}, 1 \right\}, \left\{ -\frac{-m11 + m22 - \sqrt{m11^2 + 4 m12 m21 - 2 m11 m22 + m22^2}}{2 m21}, 1 \right\} \right\} \right\}$$

Розв'яжемо систему лінійних рівнянь $x+y=2$ і $3x+2y=1$ за допомогою функції `LinearSolve`, параметрами якої виступають матриця коефіцієнтів біля невідомих та масив вільних доданків

```
LinearSolve[{{1, 1}, {3, 2}}, {2, 1}]
```

```
{-3, 5}
```

Для виконання прикладів з пакету `VectorAnalysis` загрузимо цей пакет в оперативну пам'ять

```
<<Calculus`VectorAnalysis`
```

Уведемо та виконаємо для прикладу функцію з цього пакету

```
<<Calculus`VectorAnalysis`
```

```
JacobianMatrix[Spherical[r, θ, φ]]
```

```
{{Cos[φ] Sin[θ], r Cos[φ] Cos[θ], -r Sin[φ] Sin[θ]},
 {Sin[φ] Sin[θ], r Cos[θ] Sin[φ], r Cos[φ] Sin[θ]},
 {Cos[θ], -r Sin[θ], 0}}
```

7. Розклад у ряд Тейлора. Для розкладу функції $f(x)$ у ряд Тейлора біля точки x_0 із збереженням доданків до n -го порядку використовують таку функцію

```
Series[f(x), {x, x0, n}]
```

Наприклад,

```
Series[Sin[x], {x, 0, 6}]
```

$$x - \frac{x^3}{6} + \frac{x^5}{120} + O[x]^7$$

```
Series[1/Sqrt[1-x], {x, 0, 2}]
```

$$1 + \frac{x}{2} + \frac{3x^2}{8} + O[x]^3$$

Для функції двох змінних можна отримати розклад за двома змінними

```
Series[Sin[x y], {x, 0, 5}, {y, 0, 5}]
```

$$(y + O[y]^6) x + \left(-\frac{y^3}{6} + O[y]^6\right) x^3 + \left(\frac{y^5}{120} + O[y]^6\right) x^5 + O[x]^6$$

ПОБУДОВА ГРАФІКІВ РІЗНОГО ТИПУ

1. Побудова графіків функцій на площині. Для побудови графіків функцій $y=f(x)$ використовується функція Plot, яка вже зустрічалася при розв'язуванні рівнянь:

`Plot[f, {x,  $x_{\min}$ ,  $x_{\max}$ }]` — будує графік функції $y=f(x)$ в інтервалі зміни x від $x_{\min}$ до $x_{\max}$;

`Plot[{ $f_1, f_2, \dots$ }, {x,  $x_{\min}$ ,  $x_{\max}$ }]` — будує графіки ряду функцій $f_1, f_2, \dots$

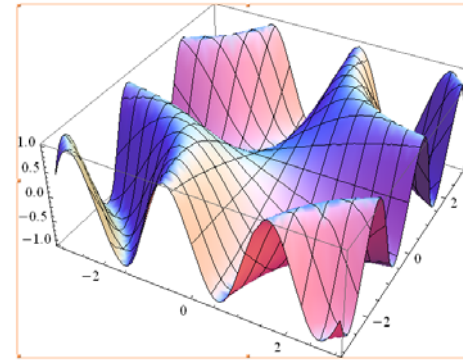
2. Побудова графіків функцій в просторі.

Для побудови графіків функцій $z=f(x,y)$ використовується функція Plot3D. Вона задається так:

`Plot3D[f, {x,  $x_{\min}$ ,  $x_{\max}$ }, {y,  $y_{\min}$ ,  $y_{\max}$ }]` — будує графік функції $z=f(x,y)$ в інтервалі зміни x від $x_{\min}$ до $x_{\max}$ та зміни y від $y_{\min}$ до $y_{\max}$.

Наприклад, побудуємо графік функції $f[x] = \sin(x y)$ при зміні x від -3 до 3 і y від -3 до 3 .

```
Plot3D[Sin[x y], {x, -3, 3}, {y, -3, 3}]
```

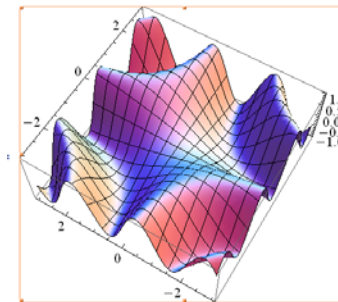


Для поліпшення результату збільшимо кількість точок, у яких розраховується функція, за допомогою опції PlotPoints.

Функція Plot3D має велику кількість опцій, повний перелік яких можна отримати за допомогою команди `??Plot3D`

Для вибору кращої орієнтації поверхні використовують опцію ViewPoint, яка задає координати точки спостереження. Щоб краще підібрати координати точки спостереження тривимірної фігури використовують 3D View Point Selector. Цю опцію задають за допомогою панелі інструментів Input. Вибравши потрібне положення поверхні, натискаємо Paste і отримаємо опцію ViewPoint з готовими координатами

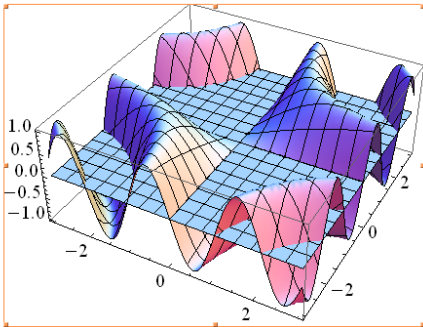
```
Plot3D[Sin[x y], {x, -3, 3}, {y, -3, 3}, PlotPoints -> 30,
ViewPoint -> {-1.426, -0.859, 2.946}]
```



За допомогою опції Mesh -> False можна зняти лінії каркасу, а опція Shading->False зображає лише лінії каркасу

Для об'єднання декількох графіків використовують функцію Show[]. Для цього присвоїмо ім'я кожному графіку і побудуємо їх в одній системі координат.

```
gr1 =
  Plot3D[Sin[x y], {x, -3, 3}, {y, -3, 3}, PlotPoints -> 30];
gr2 =
  Plot3D[0, {x, -3, 3}, {y, -3, 3}, PlotPoints -> 30];
Show[gr1, gr2]
```



3. Побудова графіків функцій, заданих неявно. Для побудови неявних функцій $f(x,y)=0$ необхідно підгрузити пакет ImplicitPlot стандартного доповнення. Для цього введіть наступну команду:

```
<<Graphics`ImplicitPlot`
```

Тепер можна використовувати функції з цього пакету

`ImplicitPlot[f, {x,  $x_{\min}$ ,  $x_{\max}$ }]`, де f - функція, задана неявно.

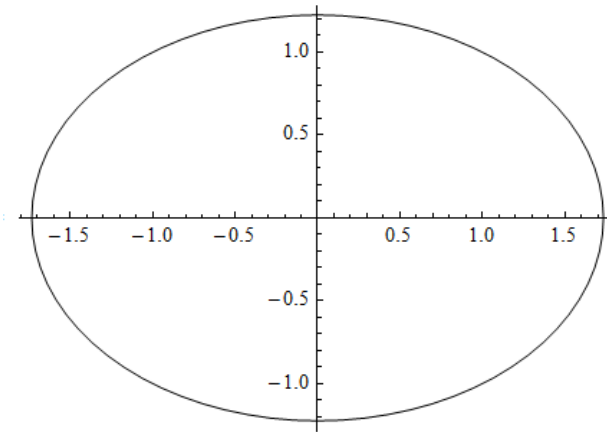
Функція f задається у вигляді рівняння (наприклад, $f(x,y)=0$), розв'язки якого будуються графічно у межах зміни аргументу $x, x_{\min}, x_{\max}$.

Для побудови декількох неявно заданих функцій записують так:

`ImplicitPlot[{ $f_1, f_2, \dots$ }, {x,  $x_{\min}, x_{\max}$ }]`, де $f_1, f_2, \dots$ - функції, задані неявно.

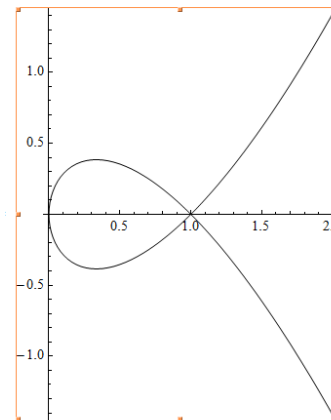
Побудуємо, наприклад, еліпс, та декілька складних заданих неявно функцій:

```
<< Graphics`ImplicitPlot`
ImplicitPlot[x^2 + 2 y^2 = 3, {x, -2, 2}]
```



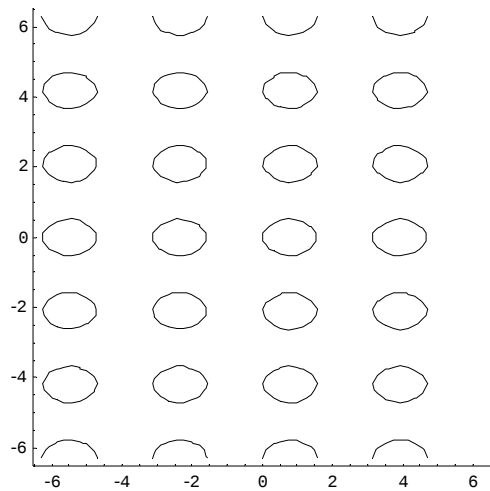
Побудуємо графік петльової параболи

```
<< Graphics`ImplicitPlot`
ImplicitPlot[y^2 = x (x - 1)^2, {x, -2, 2}]
```



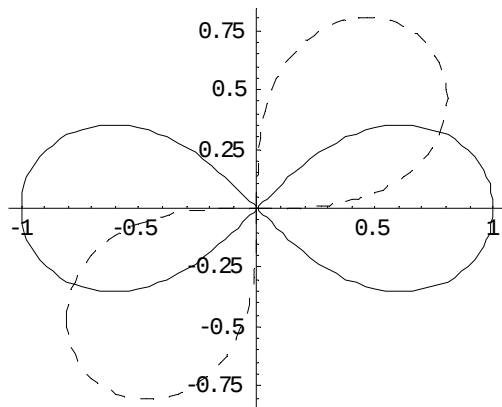
Побудуємо графік ще однієї функції, яка не може бути побудованою за допомогою стандартних засобів

```
ImplicitPlot[Sin[2*x]+Cos[3*y]==1, {x, -2*Pi, 2*Pi}, {y, -2*Pi, 2*Pi},
PlotPoints->60]
```



Побудуємо графіки масиву функцій, наприклад, двох функцій

```
ImplicitPlot[{(x^2+y^2)^2==(x^2-y^2), (x^2+y^2)^2==2*x*y}, {x, -2, 2}, PlotStyle->{GrayLevel[0], Dashing[{.03}]}]
```



Зверніть увагу на те, що для побудови графіків декількох функцій першим аргументом повинен бути масив, тобто всі неявно задані функції

повинні бути виділені фігурними дужками. Використано опцію PlotStyle, яка дозволяє зображати графіки різним стилем. Значення цієї опції задають у вигляді масиву, кожний елемент якого описує стиль відповідної кривої. Наприклад, для зображення другого графіка штриховою лінією використано функцію Dashing, яка може містити масив декількох чисел, що і будуть сприйматись послідовно і циклічно як довжини штрихів та відстані між ними.

4. Графіки функцій, заданих у полярних координатах.

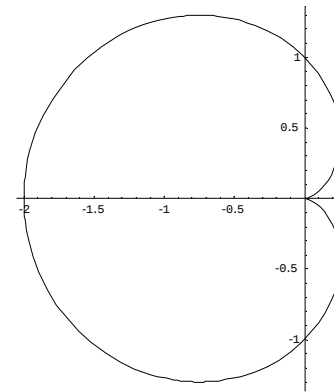
Підгрузимо пакет Graphics стандартного доповнення. Для цього уведіть наступну команду:

```
<<Graphics`Graphics`
```

Використаємо функцію PolarPlot[f(t), {t, t_{min}, t_{max}}].

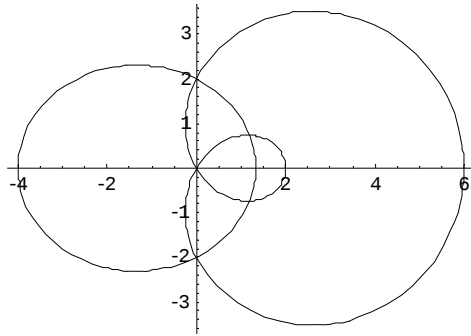
Побудуємо кардіоїду $r = 1 - \cos(t)$.

```
PolarPlot[1-Cos[t], {t, 0, 2*Pi}]
```



Побудуємо дві функції на одному графіку

```
PolarPlot[{4/(2 + Cos[t]), 4*Cos[t]-2}, {t, 0, 2*Pi}]
```



Функція `PolarPlot` містить стандартний набір опцій

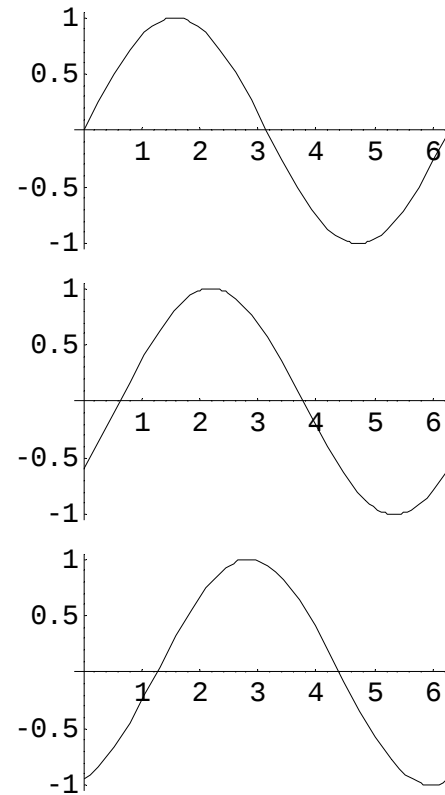
`Options[PolarPlot]`

```
{AlignmentPoint -> Center, AspectRatio -> Automatic, Axes -> Automatic,
AxesLabel -> None, AxesOrigin -> {0, 0}, AxesStyle -> {}, Background -> None,
BaselinePosition -> Automatic, BaseStyle -> {}, ColorFunction -> Automatic,
ColorFunctionScaling -> True, ColorOutput -> Automatic, ContentSelectable -> Automatic,
CoordinatesToolOptions -> Automatic, DisplayFunction -> $DisplayFunction,
Epilog -> {}, EvaluationMonitor -> None, Exclusions -> Automatic,
ExclusionsStyle -> None, FormatType -> TraditionalForm, Frame -> False,
FrameLabel -> None, FrameStyle -> {}, FrameTicks -> Automatic, FrameTicksStyle -> {},
GridLines -> None, GridLinesStyle -> {}, ImageMargins -> 0., ImagePadding -> All,
ImageSize -> Automatic, ImageSizeRaw -> Automatic, LabelStyle -> {},
MaxRecursion -> Automatic, Mesh -> None, MeshFunctions -> {#3 &}, MeshShading -> None,
MeshStyle -> Automatic, Method -> Automatic, PerformanceGoal -> $PerformanceGoal,
PlotLabel -> None, PlotPoints -> Automatic, PlotRange -> Automatic,
PlotRangeClipping -> True, PlotRangePadding -> Automatic, PlotRegion -> Automatic,
```

5. Мультиплікація.

Mathematika дозволяє виконувати візуалізацію динамічних процесів. Якщо створити серію графічних рисунків, то вони можуть бути оживлені подібно до того, як серія нерухомих картинок-кадрів об'єднується в мультфільм. Технологію створення мультфільму продемонструємо на простому прикладі: біжучої синусоїдальної хвилі. Спочатку зробимо серію з 11 рисунків, на кожному з яких зображено графік функції $\sin(x - t)$ на відрізку осі x від 0 до 4π для фіксованих $t = 0, 2\pi/10, 4\pi/10, \dots, 2\pi$. Ці рисунки розміщуються послідовно один за одним:

```
Table[Plot[Sin[x - t], {x, 0, 2 Pi}], {t, 0, 2 Pi, 2 Pi / 10}]
```



Для оживлення створених малюнків достатньо мишкою клікнути на будь-якому з них. Знизу екрана при цьому з'являється 6 кнопок, що дозволяють змінювати напрямок анімації, прискорити, сповільнити її чи зробити паузу.

При підготовці рисунків до анімації необхідно забезпечити однакову область значень функції. У приведеному прикладі область значень функції не змінювалась, оскільки не змінювалась амплітуда хвилі. Область значень функції устанавлюється опцією `PlotRange -> {ymin, ymax}`.

Для трьохвимірних рисунків необхідно установити опцію `SphericalRegion -> True`, яка забезпечує нерухомість коробочки з графічним об'єктом від рисунка до рисунка.

6. Символьні перетворення виразів

Mathematica дозволяє автоматизувати практично всі типи символьних перетворень, які зустрічаються у фізичних чи математичних задачах. Нижче приведено перелік основних функцій, що застосовуються для цих цілей, а також приклади їх використання.

`Expand[expr]` - розкладає вираз на доданки.

`Factor[expr]` - розкладає поліноми на множники (протилежна до `Expand`).

`Collect[expr, x]` - групує доданки з однаковою ступінню символу x .

```
z = Factor[x^4 + 2 x^3 - 6 x^2 + 2 x + 1]
```

```
(-1 + x)^2 (1 + 4 x + x^2)
```

```
Expand[z]
```

```
1 + 2 x - 6 x^2 + 2 x^3 + x^4
```

Для тригонометричних виразів необхідно вказати опцію `Trig`

```
expr = Expand[Sin[x + 2 y], Trig -> True]
```

```
Cos[y]^2 Sin[x] + 2 Cos[x] Cos[y] Sin[y] - Sin[x] Sin[y]^2
```

```
Factor[expr, Trig -> True]
```

```
Sin[x + 2 y]
```

Для розкладу раціональних виразів на дробі з простими доданками використовують функцію `Apart`:

```
z = Apart[x^4 + 2 x^3 - 6 x^2 + 2 x + 1 / (x^2 + 1)]
```

```
-7 + 2 x + x^2 + 8 / (1 + x^2)
```

Функція `Together` приводить суму раціональних виразів до спільного знаменника і скорочує спільні множники в чисельнику та знаменнику:

```
Together[-7 + 2 x + x^2 + 8 / (1 + x^2)]
```

```
1 + 2 x - 6 x^2 + 2 x^3 + x^4 / (1 + x^2)
```

Деколи виникає необхідність виключити частину змінних із системи рівнянь і знайти рівняння для змінних, що залишилися. Наприклад при заміні змінних у рівнянні. Для цього використовують функцію `Eliminate[eqns, x]`, де `eqns` – система рівнянь, x – змінна, яка виключається.

Наприклад, при переході від змінної x до змінної y за деяким правилом, записують так

```
Eliminate[{x^4 + 2 x^3 - 6 x^2 + 2 x + 1 == 0, y == x + 1 / x}, x]
```

```
2 y + y^2 == 8
```

Функція `Reduce[eqns, vars, elims]` спрощує рівняння `eqns`, розв'язує їх відносно `vars`, виключаючи при цьому змінні `elims`. Рівняння, які породжуються цією функцією еквівалентні початковим рівнянням і містять усі їх можливі розв'язки.

МОЖЛИВІ ПРИКЛАДИ ЗАВДАНЬ З ТЕОРЕТИЧНОЇ ФІЗИКИ, ДЛЯ ВИКОРИСТАННЯ В НАВЧАЛЬНІЙ ПРАКТИЦІ

1. Визначити напруженість поля нескінченного, рівномірно зарядженого з поверхневою густиною σ прямого круглого циліндру, радіус якого a .
2. Визначити напруженість поля рівномірно зарядженої з поверхневою густиною σ сферичної поверхні радіусом a .
3. Куля радіуса a рівномірно заряджена по всьому об'єму з густиною заряду ρ . Знайти потенціал і напруженість поля в середині і зовні кулі.
4. Знайти розподіл потенціалу в плоскому конденсаторі з відстанню між пластинами d , одну з яких заземлено, а друга перебуває під потенціалом φ_0 . Простір між пластинами заповнений зарядом з об'ємною густиною $\rho = \text{const}$.
5. Визначити плече диполя, якщо потужність його випромінювання дорівнює потужності випромінювання рамки зі струмом в задачі 4.22. Частота коливань диполя дорівнює частоті коливань сили струму в рамці, а кожен із зарядів диполя дорівнює $|q| = 10^{-4}$ Кл.
6. Пробій в повітрі відбувається при напруженості електричного поля $E=30$ кВ/см. При якій густині потоку енергії плоских електромагнітних хвиль не дуже великої частоти настає пробій у повітрі?
7. Повільний мюон розпадається за час $t_0=2 \cdot 10^{-6}$ с на електрон і нейтрино. Обчислити час життя швидкого мюона енергія якого $E=3,5 \cdot 10^9$ еВ.
8. Яку відстань проходить мюон попередньої задачі свого життя з точки зору власної системи відліку? Припустити що частинка рухається зі швидкістю світла.
9. Нерухомий пі-мезон розпадається на мюон і антинейтрино. Знайти кінетичну енергію мюона і антинейтрино, якщо маси спокою дорівнюють: пі-мезона – $273 m_e$, мю-мінус мезона – $207 m_e$, і антинейтрино ≈ 0 , де m_e – маса спокою електрона.

10. На прискорювачу протони розганяються до енергії $7,6 \cdot 10^4$ МеВ. Обчислити релятивістську масу (в масах спокою) і швидкість цих частинок.
11. Вантаж масою 1 кг підвішений до троса довжиною 2 м і здійснює коливання згідно рівняння $\varphi = \frac{\pi}{6} \sin(2\pi t)$, де φ – кут відхилення троса від вертикалі в радіанах. Визначити натяг T_1 і T_2 троса в верхньому і нижньому положеннях вантажу.
12. Тіло масою m рухається по еліпсу $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$. Прискорення тіла паралельно осі ОУ. При $t=0$ координати тіла були $x=0$; $y=b$; початкова швидкість v_0 . Визначити силу, що діє на тіло в кожній точці його траєкторії.
11. Знайти рівняння руху точки масою m , що падає без початкової швидкості на землю. Опір повітря пропорційний квадрату швидкості. Коефіцієнт пропорційності дорівнює k .
12. Тіло масою 1 кг рухається під дією сили $F = 10(1 - t)$ Н. Через скільки секунд тіло зупиниться, якщо початкова швидкість тіла $v_0 = 20$ м/с і сила збігалась за напрямком зі швидкістю тіла? Який шлях пройде тіло до зупинки?
13. Матеріальна точка масою m здійснює прямолінійний рух під дією сили $F = F_0 \cos(\omega t)$. Початкова швидкість точки v_0 . Знайти рівняння руху точки.
14. Матеріальна точка масою m відштовхується від силового центру з силою, що пропорційна відстані (коефіцієнт пропорційності mk_2). Опір середовища пропорційний швидкості руху (коефіцієнт пропорційності $2mk_1$). В початковий момент точка знаходилась на відстані a від силового центру, і її швидкість в цей момент дорівнювала нулю. Знайти закон руху точки.

15. Точка масою m починає рухатись без початкової швидкості з положення $x = b$ прямолінійно (вздовж осі X) під дією сили притягання до початку координат, яка змінюється за законом $F = \frac{a}{x^2}$. Знайти момент часу, коли точка перебуватиме в положенні $x_1 = \frac{b}{2}$. Визначити швидкість точки в цьому положенні.

16. Вантаж на пружині коливається так, що його рух описується диференціальним рівнянням:

$$m\ddot{y} + ky = 5 \cos(\omega t) + 2 \cos(3\omega t)$$

Знайти закон руху вантажу, якщо в початковий момент його зміщення і швидкість дорівнювали нулю, а також визначити, при яких значеннях ω наступить резонанс.

17. Знайти рівняння прямолінійного руху точки масою m , що знаходиться під дією сили $F_1 = kv$ та сили $F_2 = \alpha t$. В початковий момент часу $x_0 = 0$, $x'_0 = 0$.

18. Знайти рівняння прямолінійного руху точки масою m , що знаходиться під дією сили $F_1 = -kx$ та сили $F_2 = F_0 e^{-\alpha t}$. В початковий момент часу $x_0 = 0$, $x'_0 = 0$.

19. Матеріальна точка масою $m=3$ кг підвішена на пружині, коефіцієнт пружності якої $k=117,6$ Н/м. На точку діє вимушуюча сила $F_1 = F_0 \sin(6,26t + \phi)$ Н, та сила опору руху $F_2 = -\alpha v$ Н. Як зміниться амплітуда вимушених коливань точки, якщо коефіцієнт α збільшиться в три рази.

20. Матеріальна точка масою $m=0,1$ кг підвішена на пружині, коефіцієнт пружності якої $k=5$ кН/м. На точку діє вимушуюча сила $F_1 = 100 \sin(100t)$ Н та сила опору руху $F_2 = -50v$ Н. Написати рівняння руху вимушених коливань.

КОНТРОЛЬНІ ЗАВДАННЯ

1. Розрахуйте вирази $N[\text{Pi}, 3]$, $N[\text{Pi}, 6]$, $N[\text{Pi}, 100]$. Що вони означають? Дізнайтесь, скільки часу займе розрахунок на комп'ютері приблизне значення числа π з тисячу вірними цифрами, розрахувавши вираз

$N[\text{Pi}, 1000]// \text{Timing}$

2. Встановіть, що більше: e^π чи π^e ?
3. Запишіть розклад в ряд Тейлора функції **Cos[x]** використовуючи функцію суми і перевірте його правильність.
4. Запишіть у загальному випадку суму n доданків арифметичної прогресії. Перший доданок **a1**, крок **d**.
5. За допомогою вбудованої функції **D** розрахуйте змішану похідну від функції $y \exp(x^2 y^3)$ за змінними x і y . Скільки часу займе розрахунок похідної десятого порядку від цієї функції за змінною x ?
6. За допомогою функції **Integrate** розрахуйте визначений інтеграл від функції $x \sin(x)$ від **0** до π .
7. Встановіть, при якому значенні величини α інтеграл від функції $\sin(\alpha x)/\alpha$ у межах від **0** до π рівний інтегралу від функції $x \sin(x)$ у тих же межах? Із отриманим значенням величини α побудуйте графіки цих двох функцій у заданих межах і переконайтеся, що площі, обмежені кривими та віссю Ox , однакові.
8. Знайдіть розв'язки систем рівнянь $x^2 + y^2 = 1$, $x - y = 1/2$ та $x^2 + y^2 = 1$, $x - y = 0.5$. Поясніть, чому отримані результати відрізняються за формою?
9. Знайдіть числове значення найбільшого кореня рівняння $\sin(x) = 0.1x$. Попередньо побудуйте графіки функцій правої та лівої частини рівняння.
10. Знайдіть всі розв'язки рівняння $\sin(x^2) = 1$ на відрізку $0 < x < 4$. Скористайтесь функціями **NSolve**, **FindRoot**, **Plot**. Перевірте отримані розв'язки методом підстановки. Встановіть, яка функція знаходить точніші значення коренів. Для збільшення точності у функції **FindRoot** додайте опції **MaxIterations** $\rightarrow 30$ та **WorkingPrecision** $\rightarrow 20$

11. Створіть двохвимірний масив 5x5, кожний елемент якого задається виразом $x^i + y^j$, де i – номер рядка, j – номер стовпчика.
12. Знайдіть аналітичний і числовий розв'язки диференціального рівняння $y'[x] = y[x]^2$, $y[0] = 1$ на інтервалі $0 < x < 1$ і переконайтеся, що вони збігаються, побудувавши відповідні графіки.
13. Розрахуйте п'ять перших доданків розкладу функції $\ln(1-x)\ln(1+x)$ в ряд Тейлора в околі точки $x=0$.
14. Побудуйте в одній системі координат графіки таких функцій $y = e^x$, $y = x^2$, $y = \Gamma(x)$ у межах $\{x, -5, 5\}$. (Гамма функція записується так **Gamma[x]**). Усі функції повинні бути зображені різним кольором.
15. Побудуйте 3-вимірний графік функції $f(x,y)=\sin(x*y)$, у межах $\{x, -1, 1\}$, $\{y, -1, 1\}$
16. Побудуйте динамічну просторову картинку на основі графіка функції **BesselJ[n, x*y]**, у межах $\{x, 0, \text{Pi}\}$, $\{y, 0, \text{Pi}\}$ при зміні n від 0 до 8 з кроком 0.2. Не забудьте скористатись опціями **PlotRange**, **SphericalRegion**. Підберіть необхідну кількість точок в опції **PlotRange**.
17. За допомогою системи Mathematica складіть програму, яка створить таблицю основних математичних формул: похідних інтегралів, перетворення тригонометричних виразів та інші і запише їх у текстовий файл.

ЗМІСТ

ВСТУП	6
МЕТА, ЗАВДАННЯ ТА ОСНОВНІ ТЕМИ ПРАКТИКИ.....	7
МЕТОДИКА РОЗВ'ЯЗУВАННЯ ЗАДАЧ ІЗ ТЕОРЕТИЧНОЇ ФІЗИКИ	13
ТИПИ ДАНИХ І РОБОТА З НИМИ	19
ОСНОВИ ПРОГРАМУВАННЯ	28
ОПЕРАЦІЇ МАТЕМАТИЧНОГО АНАЛІЗУ	31
МОЖЛИВІ ПРИКЛАДИ ЗАВДАНЬ З ТЕОРЕТИЧНОЇ ФІЗИКИ, ДЛЯ ВИКОРИСТАННЯ В НАВЧАЛЬНІЙ ПРАКТИЦІ	57
КОНТРОЛЬНІ ЗАВДАННЯ	60

Навчально-методичний посібник

**Використання системи символьного числення «Mathematica»
При проходженні навчальної практики з теоретичної фізики**

*Методичні вказівки для проходження практики з теоретичної фізики та
роботи з програмою «Mathematica»*

Укладач **Грищук Андрій Миколайович,**
Корнійчук Платон Павлович
Відповідальний за випуск
Літературний редактор