<u>Лабораторна робота 1</u>

Комп'ютерна система Mathematica

Мета роботи: ознайомитися з основами роботи в програмному пакеті системи *Mathematica*

За багато років в світі накопичено великі бібліотеки комп'ютерних підпрограм, написаних, в першу чергу, мовами *FORTRAN* та *C*, і призначених для розв’язку типових математичних задач (розв’язок систем лінійних та нелінійних рівнянь, розв’язок диференційних рівнянь, наближення функцій і т. д.). Крім того, є цілий ряд різних універсальних математичних пакетів, які реалізують різноманітні чисельні методи, а також здатні виконувати аналітичні перетворення і розв’язувати багато відомих математичних задач аналітично.

Найбільш відомими сьогодні є наступні пакети: *Mathematica* (фірма *Wolfram Research*), *Maple* (фірма *Waterloo Maple Inc*), *Matlab* (фірма *MathWorks*), *Mathcad* (фірма *MathSoft Inc*). Вони застосовуються навіть в інтелектуальній сфері діяльності вчених різних спеціальностей, які займаються розв’язком задач теорії поля, аеродинаміки, космонавтики, математичного моделювання.

Пакет *Mathematica* є найпопулярнішим в наукових колах, особливо серед теоретиків . Пакет надає широкі можливості в проведенні символічних (аналітичних) перетворень, проте вимагає значних ресурсів комп'ютера. Пакет *Mathematica* дозволяє швидко і ефективно розв’язувати багато задач лінійної алгебри, теорії чисел, дискретної математики, математичного аналізу, диференційних рівнянь. Вміє виконувати складні алгебраїчні перетворення і спрощення з різними математичними виразами: багаточленами, раціональними виразами, елементарними і спеціальними функціями, алгебраїчними і тригонометричними виразами. Знаходить кінцеві і нескінченні суми, добутки, границі та інтеграли. Розв’язує в символьному вигляді і чисельно алгебраїчні і трансцендентні системи рівнянь, системи звичайних диференційних рівнянь і деякі класи рівнянь в частинних похідних. У пакеті *Mathematica* розв’язок більшості задач проводиться в діалоговому режимі, без традиційного програмування, з використанням стандартних операторів. Однак, в цей пакет також інтегровано функціональну мову програмування високого рівня. Характерною особливістю пакету є те, що він дозволяє конвертувати документи у формат *LaTeX* — стандартний формат написання наукових робіт переважної більшості наукових видавництв світового класу.

Пакет *Maple* – один з перших серед пакетів символьної математики, також дуже популярний в наукових колах . Окрім аналітичних перетворень пакет може розв’язувати задачі чисельно. Наприклад, ядро пакету *Maple–6* в студентській версії містить понад три тисячі математичних функцій і правил їх перетворення. Графічний редактор пакету дозволяє одержувати зображення тривимірних фігур з перетинами з функціональним забарвленням. Також він дозволяє конвертувати документи у формат *LaTeX*. Крім того, ряд інших програмних продуктів використовують інтегрований символічний процесор *Maple*. Наприклад, пакет підготовки наукових публікацій *Scientific WorkPlace* (фірма *TCI Software Research*) дозволяє звертатися до символьного процесора *Maple*, проводити аналітичні перетворення і вбудовувати отримані результати в документ.

Подібно згаданим вище пакетам, *Matlab* є мовою програмування високого рівня, орієнтовану переважно на інженерні розрахунки теорії управління, електро– і радіотехніки, а також на моделювання технічних систем . Характерною особливістю пакету є те, що він дозволяє зберігати документи у форматі мови програмування *C*.

Виник *MatLab* як пакет для матричних обчислень. Потім він був оснащений сучасним графічним редактором і доповнений символічним процесором *Maple. Matlab* став фактично міжнародним стандартом сучасного учбового програмного забезпечення і використовується в багатьох провідних університетах світу. Можливості щодо символьних перетворень у пакеті *MatLab* є дещо скромнішими у порівнянні з *Matematica* та *Maple*, однак цей пакет має значно вищу швидкодію.

Пакет *Mathcad* теж популярний, однак, більше в інженерному, ніж в науковому середовищі. Характерною особливістю пакету є використання звичних стандартних математичних позначень, тобто документ на екрані виглядає так само, як звичайний математичний розрахунок. Пакет орієнтований, в першу чергу, на проведення чисельних розрахунків, але має вбудований символічний процесор *Maple*, що дозволяє виконувати аналітичні перетворення. В останніх версіях передбачена можливість створювати зв'язки документів *Mathcad* з документами *MatLab*. На відміну від згаданих вище пакетів, *Mathcad* є середовищем візуального програмування, тобто не вимагає знання специфічного набору команд.

Останнім часом намітилася тенденція до зближення та інтеграції різних математичних пакетів. Наприклад, останні версії пакетів *Mathematica* і *Maple* мають потужні засоби для візуального програмування; в *Matlab* інтегрована бібліотека аналітичних перетворень *Maple*; *Mathcad* дозволяє працювати спільно з *MatLab* і т.д. Тому для організації практичних занять з чисельних методів можливий вибір будь–якого з перелічених вище пакетів, виходячи з можливостей і традицій конкретного навчального закладу.

В лабораторних роботах з курсу „Чисельні методи” використовується математичний пакет *Mathematica*, перш за все, через простоту виконання обчислень і наочного представлення результатів розв’язку.

Слід зазначити, що розглянуті математичні пакети (*Mathematica*, *Maple*, *MatLab*, *Mathcad*) дозволяють ефективно розв’язувати порівняно невеликі за розмірами математичні задачі, що містять десятки змінних, але не більше. Тому для науковців та інженерів вони є незамінним *персональним* інструментарієм і не можуть замінити собою *інженерні програмні комплекси* комп'ютерного моделювання і проектування виробів і об'єктів сучасної науки і техніки, такі як системи *Cadence, AutoCAD, ANSYS, HSPICE* і багато інших, які не тільки обробляють математичні задачі великої і дуже великої розмірності, але і самі їх формують за вхідною інформацією про структуру об'єкту і параметри його компонентів.

При першому запуску пакету *Mathematica* ми бачимо наступне вікно:

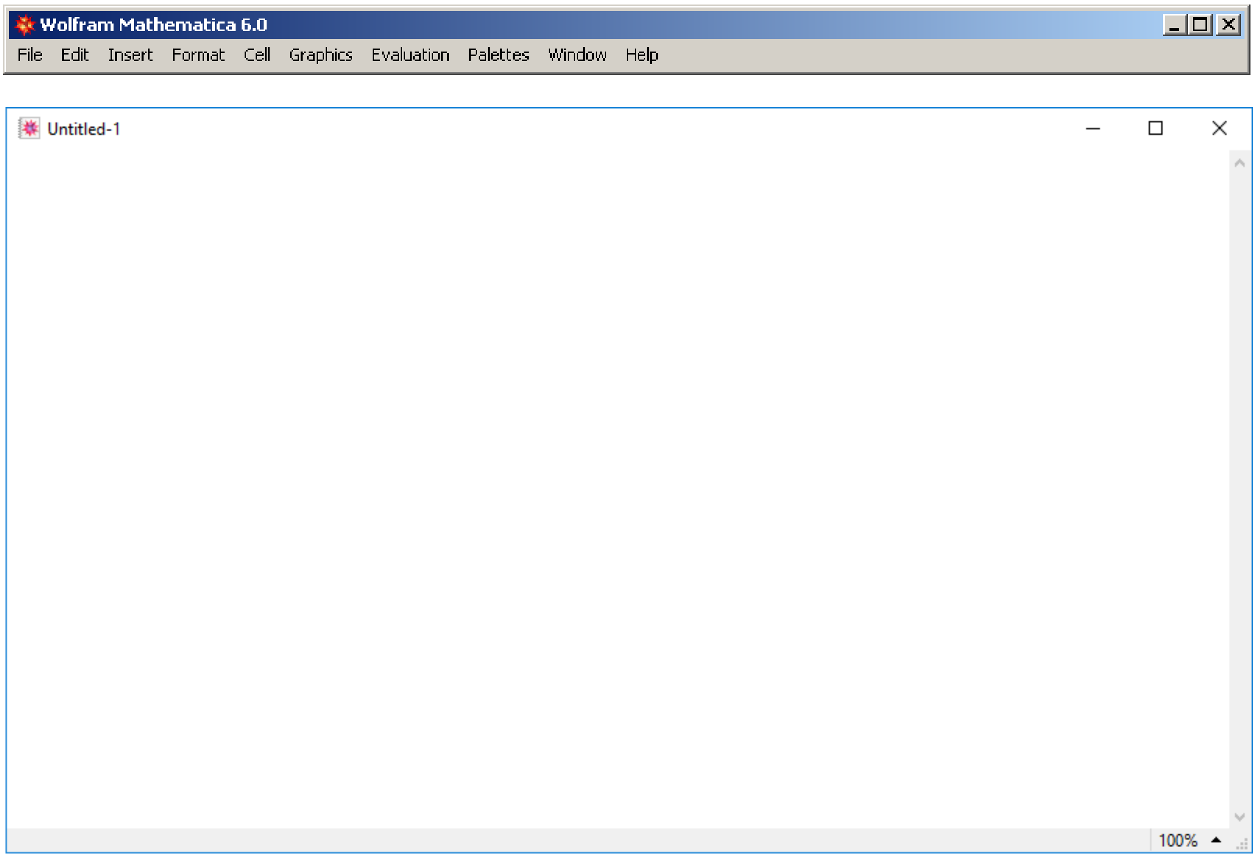


Рис. 1.1. Стартова сторінка

Неважко помітити, що призначений для користувача інтерфейс системи Mathematica реалізує окреме виведення своїх елементів - вікон (включаючи основне вікно редагування), панелей, палітр знаків і т.д. Головне вікно системи це блокнот для введення даних, призначених для математичних перетворень. Справа і знизу великого блокноту знаходяться лінійки прокрутки з характерними повзунками. Вони призначені для скролінгу текстів великих документів, якщо останні не поміщаються у видимій частині вікна. У самому низу на початку лінійки прокрутки є рядок стану (*Status bar*) з інформацією про поточний режим роботи. Ця інформація (якщо вона є в даний момент) корисна для оперативного контролю в ході роботи з системою. Головне меню системи містить наступні позиції:

•**File** - робота з файлами: створення нового файлу, вибір файлу з каталогу, закриття файлу, запис поточного файлу, запис файлу із зміною імені, друк документа і завершення роботи;

•Edit - основні операції редагування (відміна операції, копіювання виділених ділянок документа в буфер з їх видаленням і без видалення, перенесення виділених ділянок, їх стирання);

•Cell - робота з осередками (об'єднання і роз'єднання осередків, установка статусу осередку, відкриття і закриття);

•Format - управління форматом документів;

•Input - завдання елементів введення (графіків, матриць, гіперпосилань і т.д.);

•Kernel - управління ядром системи;

•Find - пошук заданих даних;

•Window - операції з вікнами і їх розташуванням;

•Help - управління довідковою системою.

Коротко розглянемо основні палітри, необхідні для вирішення практичних задач:

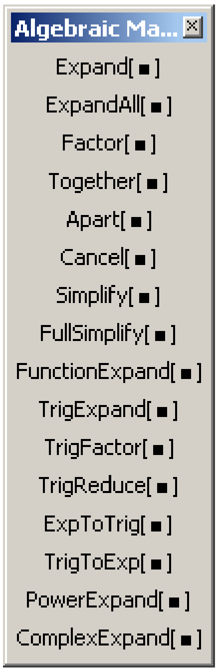


Рис. 1.2. Алгебраїчна палітра

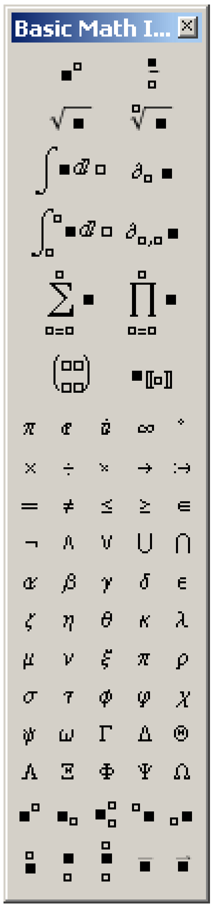


Рис. 1.3 – Базовий математичний ввід

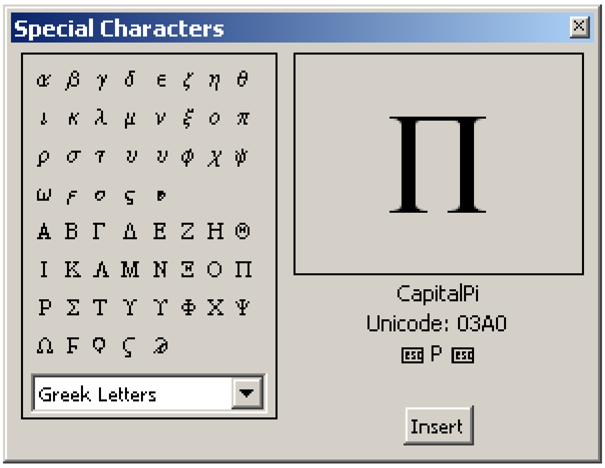


Рис. 1.4. Спеціальні символи

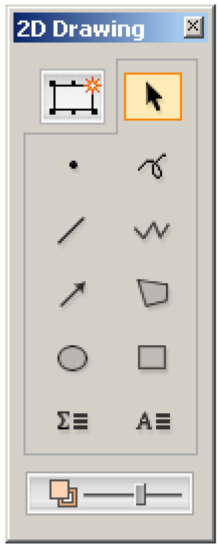


Рис. 1.5. Палітра для 2х-вимірного малювання

1. Графічне представлення функцій

В Mathematica використовується звичний формат для позначення елементарних і спеціальних функцій. Розглянемо спочатку, як в Mathematica визначаються функції

```
f[x_]:=x^2  Sin[x]+Exp[-2 (x-1.5)];
```

У лівій частині визначено ім'я функцій `f` з аргументом `x`, причому знак підкреслення ‘ _’ означає, що `x` є параметром функцій. Права частина є аналітичним виразом функції.

Mathematica використовує різні види дужок : [] – для функцій і процедур, {} – для списків і звичайні асоціативні дужки () – для алгебраїчних операцій.

1.1. Графіків функцій, заданих аналітично

Для графічного представлення функції однієї змінної використовується процедура

```
Plot[f[x], {x,0,2 Pi}, AxesLabel ->{"x", "f(x)"}]
```

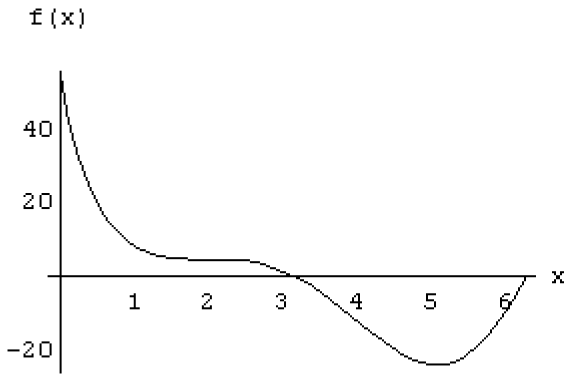


Рис. 1.6. Графік функції $y=x^2\sin x+e^{-2(x-2)}$

Для створення рамки навколо графіку і нанесення сітки на нього використовуються такі налаштування:

```
Show[%, Frame->True, FrameLabel->{"Argum","Func"},
```

```
GridLines-> Automatic]
```

Тут % означає звернення до попереднього результату, тобто до графіка f(x).

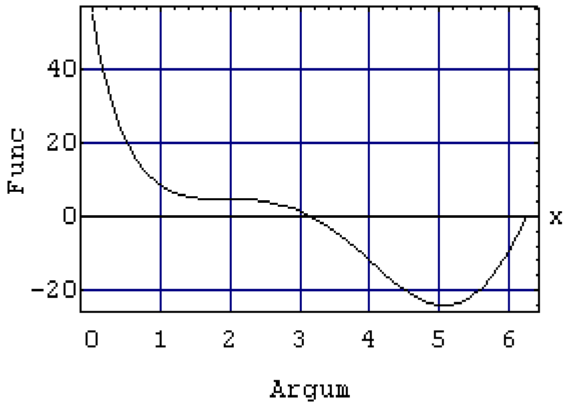


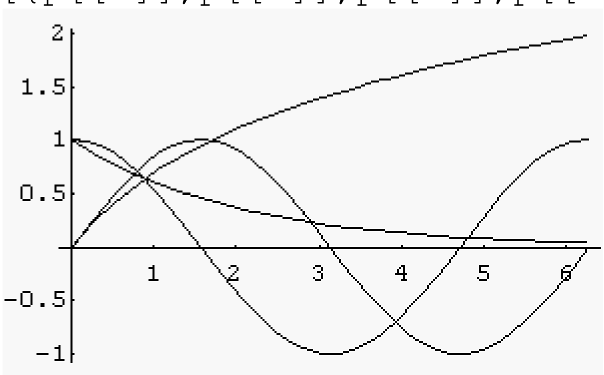
Рис. 1.7. Графік функції $y=x^2\sin x + e^{-2(x-2)}$

Визначимо список функцій наступним оператором

```
p={Sin[x],Cos[x],Exp[-0.5 x],Log[1+x]}
```

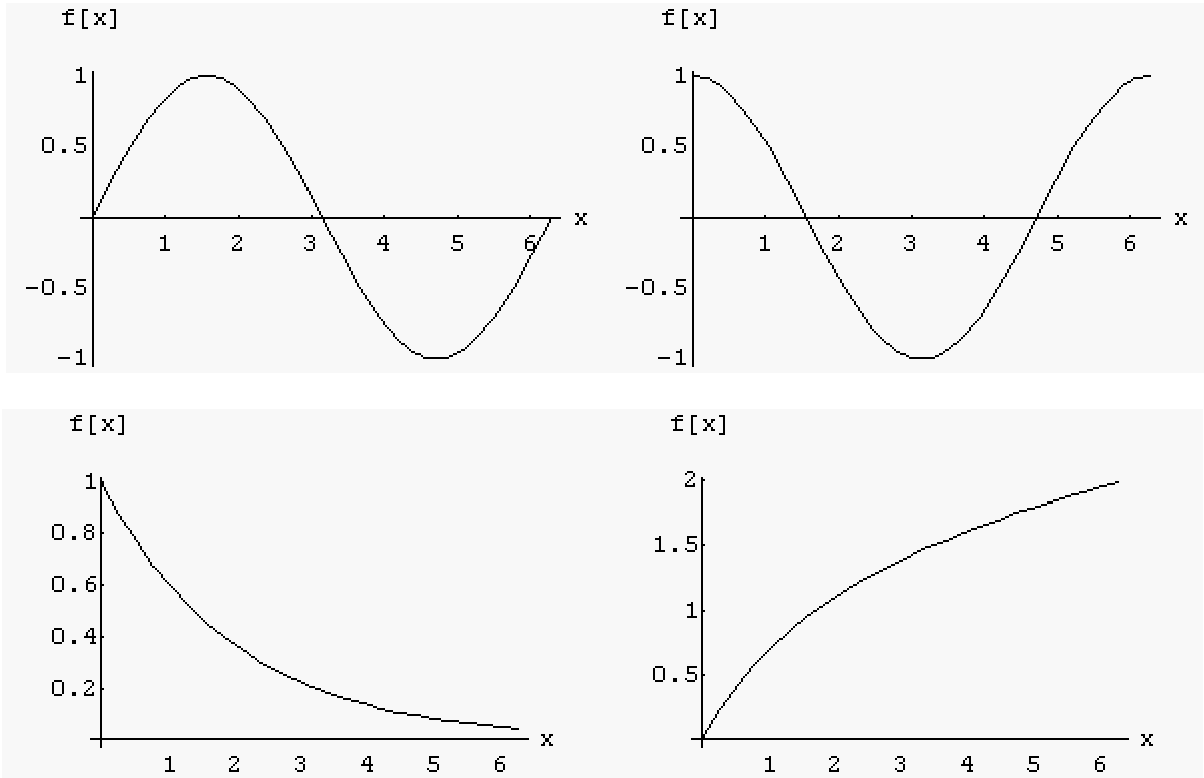
Побудуємо графіки функцій, що входять в список, на одному малюнку, використовуючи наведену нижче процедуруЖ

```
Plot[{p[[1]],p[[2]],p[[3]],p[[4]]},{x,0,2 Pi}]
```



Для побудови кожного графіка на окремому полотні використовується наступна процедура

```
g=Table[Plot[p[[i]],{x,0,2 Pi},AxesLabel->
{"x","f[x]"}],{i,1,4}]
```



Якщо необхідно нанести на графіках найменування кривих, то для цього вводиться список координат, які визначають положення назв графіків:

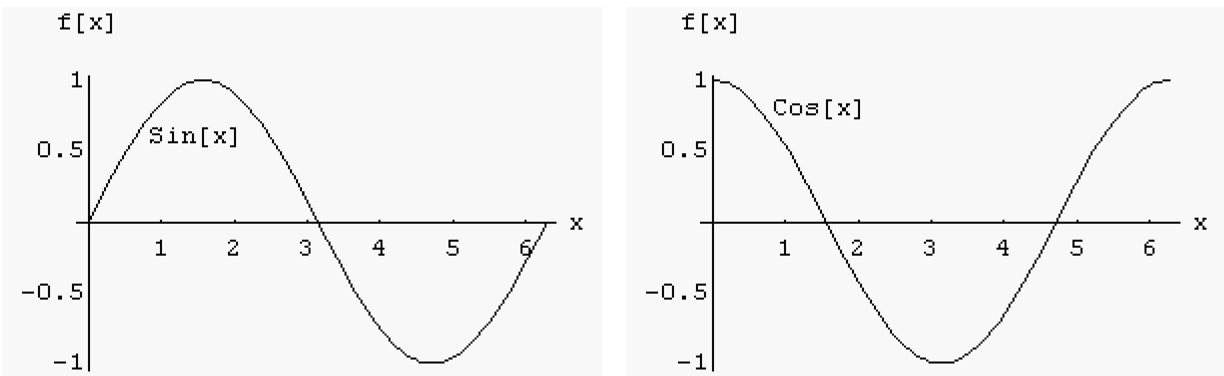
```
deskr={{1.5,0.6},{1.5,0.8},{2,0.8},{2,1.8}};
```

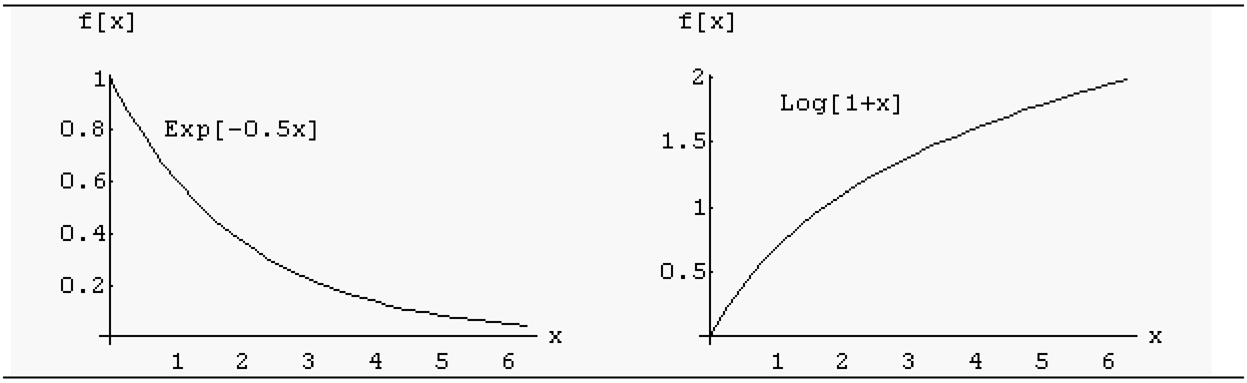
і самі назви

```
txt={"Sin[x]","Cos[x]","Exp[-0.5x]","Log[1+x]"};
```

а потім виводяться графіки з відповідними найменуваннями

```
g=Table[Plot[p[[i]],{x,0,2 Pi},AxesLabel->{"x","f[x]"},
      Epilog->Text[txt[[i]],deskr[[i]]]],{i,1,4}]
```

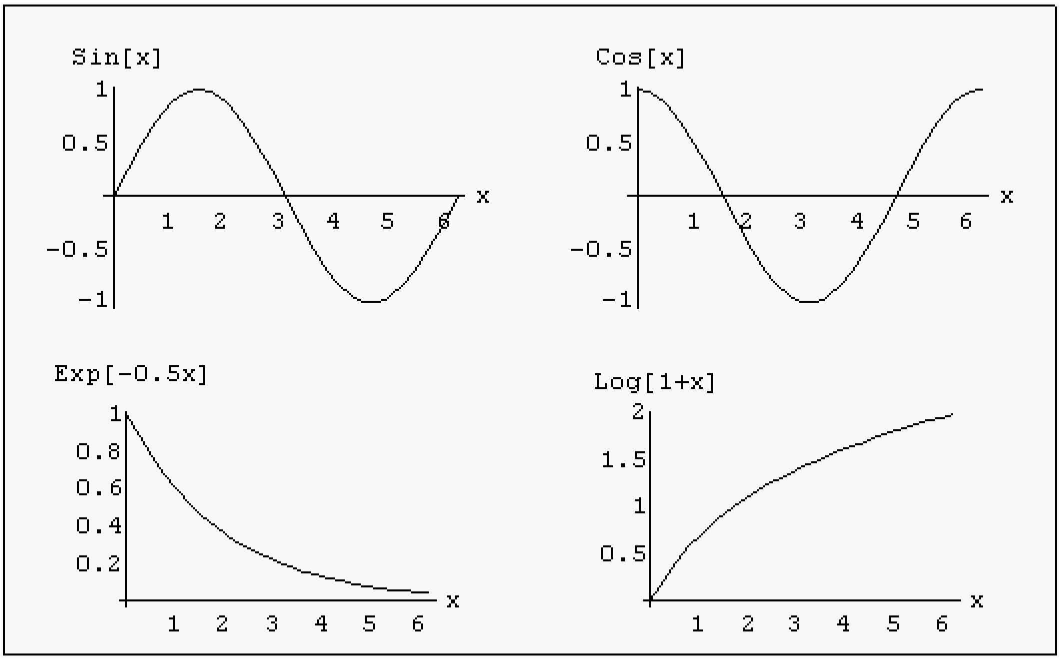




Графіки можна розмістити в таблиці, так як це зроблено нижче:

```
p={Sin[x],Cos[x],Exp[-0.5 x],Log[1+x]};
deskr={{0.1,1.3},{0.1,1.3},{0.1,1.2},{0.1,2.3}};
txt={"Sin[x]","Cos[x]","Exp[-0.5x]","Log[1+x]"};
g=Table[Plot[p[[i]],{x,0,2 Pi},AxesLabel->{"x"," "},
Epilog->Text[txt[[i]],deskr[[i]]],DisplayFunction-
>Identity],{i,1,4}];
Show[GraphicsArray[{{g[[1]],g[[2]]},{g[[3]],g[[4]]}}],
Frame->True,FrameTicks-> None]
```

Спочатку за допомогою опції DisplayFunction ->Identity буде скасована видача графіків на екран дисплея, а потім в наступній процедурі Show.



Наступний оператор дозволяє зобразити лінії рівного рівня функції двох змінних

```
ContourPlot[x^2-y^2,{x,-2,2},{y,-2,2}]
```

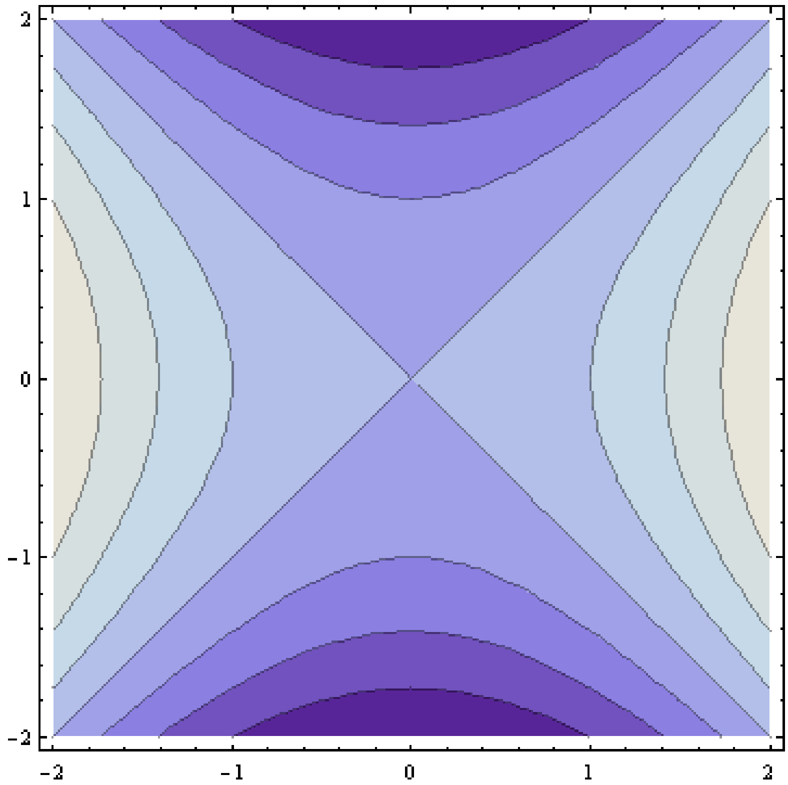


Рис. 1.8. Лінії рівного рівня функції $f(x,y)=x^2-y^2$

Для отримання просторового зображення поверхні, визначуваної рівнянням $z=x^2+y^2/16$, можна використовувати процедуру

```
Plot3D[x^2+y^2/16,{x,-2,2},{y,-4,4}]
```

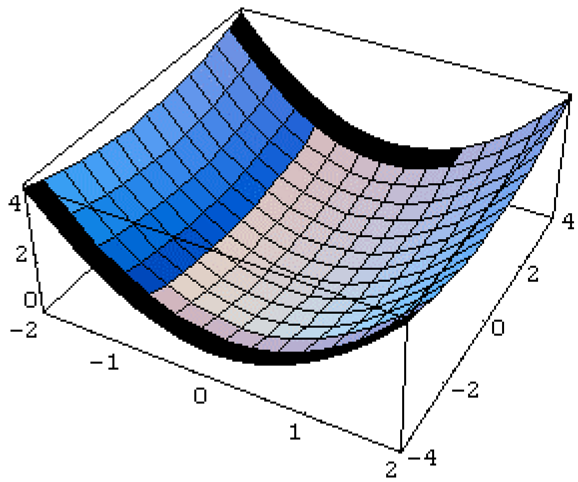


Рис. 1.9. Поверхня рівняння $z=x^2+y^2/16$

При параметричному завданні функції для її просторового зображення використовується процедура

```
ParametricPlot3D[{u Sin[t],u Cos[t], t/2}, {t,0,2 Pi},
{u,-3,3}, Ticks-> None]
```

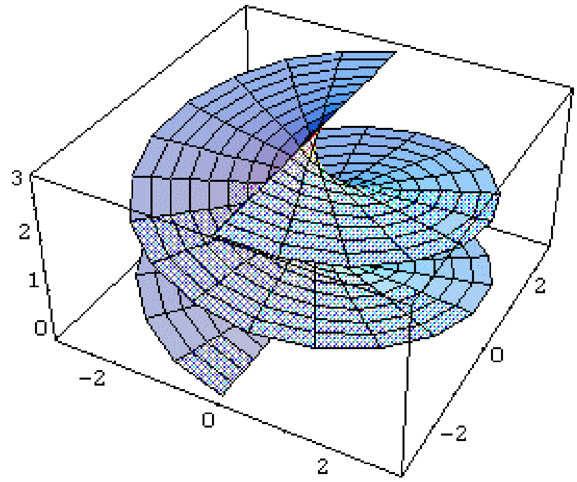


Рис. 1.10. Поверхня, задана параметрично x=uSint, y=uCost, z=t/2

Можна зображати і складніші поверхні, задані параметрично:

```
ParametricPlot3D[{Sin[t], Sin[u] Sin[2 t],Sin[2 t]
Cos[u]}, {t,-Pi/2, Pi/2},{u,0,2 Pi},Ticks->Automatic]
```

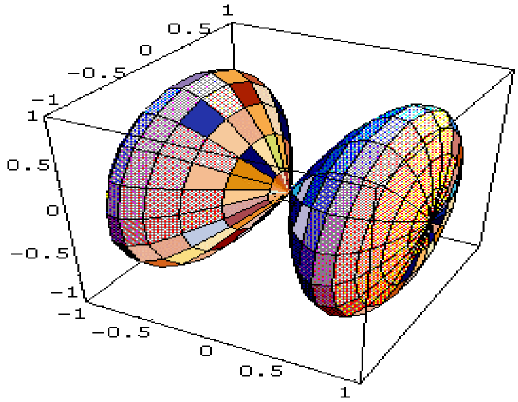


Рис. 1.11. Поверхня, задана рівняннями x=Sint, y=Sin2tSinu, z=Sin2tCosu

У останній процедурі використана опція автоматичної розмітки осей.

Окрім наведених вище графічних операторів Mathematica містить загальний графічний оператор формату Graphics3D[], що описує тривимірний графічний об'єкт на основі заданих графічних примітивів і директив. Після опису графічного об'єкту, Graphics3D виводиться за допомогою Show функції. Наведемо приклад використання примітиву:

```
Cuboid [{x,y,z },{x1,y1,z1}]
```

що описує паралелепіпед з протилежними кутами в точках (x,y,z) і (x1,y1,z1).

```
f2=Graphics3D[{Cuboid[{0,0,0},{1,1,3}],Cuboid[{1,0,0},{
2,1,2}],Cuboid[{2,0,0},{3,1,1}]}];
```

За допомогою останнього оператора створені три, що примикають один до одного паралелепіпеда, які візуалізуються наступним оператором

Show[f2]

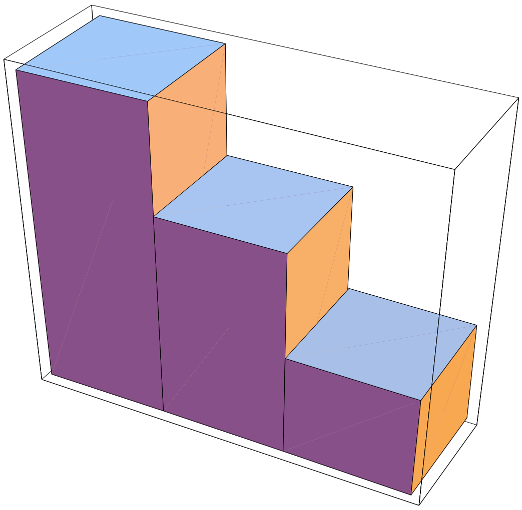


Рис. 1.12. Створення графічного об'єкту Graphics3D- оператором і виведення його функцією Show

Використання графічного примітиву для двовимірної графіки ілюструється наступним прикладом. Побудуємо стовпчикову діаграму, застосувавши двомірний графічний примітив

`Rectangle[{xmin, ymin}, {xmax, ymax}]` - який представляє заповнений прямокутник, орієнтований паралельно осям координат.

```
Show[Graphics[{Hue[.01],Rectangle[{0,0},{1,4}],Hue[.1],
Rectangle[{1,0},{2,7}],Hue[.82],Rectangle[{2,0},{3,3}],
Hue[.55],Rectangle[{3,0},{4,2}]}]]
```

Рис. 1.13. Створення графічного об'єкту Graphics- оператором і виведення його функцією Show.

1.2. Побудова графіків функцій, заданих таблицею.

Для побудови графіків табличних функцій можна використовувати функції:

а) функцію ListPLot

```
l=ListPlot[{{0,0},{0.25,0.15},{0.5,0.25},{0.75,0.6},{1.
,1.5}},FrameLabel->{"x","f(x)"},GridLines->Automatic]
```

де `ListPlot[{{x1,y1},{x2,y2}, ...{xi,yi}}]` – функція виведення точкового графіку змінних x і y.

б) за таблицею значень функції, записуваної у вигляді наступного списку

```
d={{0,0},{0.75,0.6},{1.,1.5},{1.5,2.5},{2.,4.}};
```

можна отримати інтерполяційний багаточлен. Mathematica використовує для цих цілей інтерполяційний багаточлен Ньютона, виклик відповідної процедури здійснюється таким чином. Якщо таблиця містить *n* значень, то будується багаточлен n-1 ступеня

```
ind=InterpolatingPolynomial[d,x]
```

В результаті на екран виводиться шуканий багаточлен

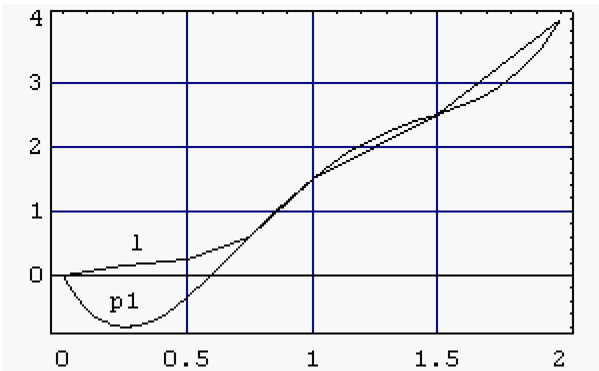
```
(4. +(-2.+x) (2. +(0.5 +(0.333333 +2.89778 (-1.5+x)) (-1.+x)) x)
```

У разі потреби можна побудувати його графік, використовуючи процедуру

```
p1=Plot[ind,{x,0,2},Frame->True,GridLines->Automatic]
```

Для порівняння графіки табличної функції і інтерполяційного многочлена наведено на одному малюнку

```
t=Graphics[{Text["l",{0.3,0.5}],Text["p1",{0.25,-0.4}]}];
Show[p1,l,t,Frame->True,GridLines->Automatic]
```



в) табличну функцію можна представити, інтерполюючи її за допомогою лінійних або кубічних сплайнів. Для побудови сплайн-функції по таблиці використовується наступна процедура. Опція указує на вибір кубічного сплайна.

```
Tr1=Interpolation[d,InterpolationOrder->3]
```

Значення інтерполяційної функції Tr, можна знайти таким чином:

```
Tr1[1.2]
```

результат надрукований нижче

```
1.96032
```

Графік отриманої функції можна побудувати оператором

```
Plot[Tr1[x],{x,0,2.},Frame->True,GridLines->Automatic]
```

Графік функцій, побудований таким чином, практично не відрізняється від графіка, побудованого за допомогою багаточлена Ньютона. Того ж результату можна досягти наступним оператором, який будує інтерполяційну функцію, використовуючи кубічні сплайни.

```
Plot[Evaluate[Interpolation[d]][x],{x,0,2.},Frame-
>True,GridLines->Automatic]
```

Отриману інтерполяційну функцію можна використовувати у виразі і, наприклад, проінтегрувати в заданих межах:

```
NIntegrate[Tr1[x]^2,{x,0,2}]
```

в результаті отримаємо

```
7.47993
```

1.3. Рішення лінійних систем стандартними операторами пакету

Систему лінійних рівнянь алгебри можна вирішити, представивши її в матричній формі. Введемо матрицю вирішуваної системи

```
m={{5,2,1,0.5,-1},{0,-
6,4,0.1,1},{0,2,10,1,4},{1,0,2,8,-3},{2,0,0,3,-9}};
```

Матриця представляється порядковою послідовністю списків. Її можна вивести на екран в звичайному вигляді наступною процедурою

```
MatrixForm[m]
```

$$\begin{pmatrix} 5 & 2 & 1 & 0.5 & -1 \\ 0 & -6 & 4 & 0.1 & 1 \\ 0 & 2 & 10 & 1 & 4 \\ 1 & 0 & 2 & 8 & -3 \\ 2 & 0 & 0 & 3 & -9 \end{pmatrix}$$

Введемо значення вектора невідомих x, вільних членів рівнянь `b`. Знайдемо потім зворотну матрицю `mi` і з її допомогою знайдемо рішення `x=mi.b`. Крапка в останній формулі позначає множення матриці на вектор.

```
b={2,-4,-3,5,8};
mi=Inverse[m];
x=mi.b

{0.0529219, 0.46216, -0.127254, 0.367172, -0.754738}
```

Останній рядок усередині дужок містить значення компонент шуканого вектора x.

Компоненту *j* вектора можна отримати через `x[[j]]`, а елемент матриці $m_{i,j}$ відповідно через `m[[i,j]]`. Є можливість обчислити визначник матриці:

```
di=Det[mi]
0.0000534456
```

Добуток двох матриць розраховується наступним чином:

```
E=m.mi
```

яка в даному прикладі повинна бути рівна одиничній матриці.

Систему лінійних рівнянь краще вирішувати за допомогою функції `LinearSolve`:

```
LinearSolve[m,b]
```

Остання процедура дозволяє отримати рішення як для числових, так і для символьних значень. Змінні `b`, `d`, `g` вже використовувались, раніше, тому потрібно їх очистити від раніше присвоєних значень функцією Clear[]:

```
Clear[b,d,g];
M={{a,b},{c,d}};
m={g,h};
x=LinearSolve[M,m]
```

$$\left\{\frac{dg-bh}{-bc+ad},\frac{cg-ah}{bc-ad}\right\}$$

Наприклад, можна отримати символьне рішення той же системи наступним чином:

```
U=Inverse[M].m
```

$$\left\{\frac{dg}{-bc+ad}-\frac{bh}{-bc+ad},-\frac{cg}{-bc+ad}+\frac{ah}{-bc+ad}\right\}$$

спростивши яке отримаємо ті ж вирази для компонент вектора рішення.

```
Simplify[%]
```

$$\left\{\frac{dg-bh}{-bc+ad},\frac{cg-ah}{bc-ad}\right\}$$

Для вирішення системи лінійних рівнянь, записаних у вигляді:

```
V={v1,v2};
W=M.V;
eq=W==m
```

можна використати процедуру

```
Solve[eq,V]
```

яка формує таке символьне рішення системи рівнянь

$$\left\{\left\{v1\to-\frac{-dg+bh}{-bc+ad},v2\to-\frac{-cg+ah}{bc-ad}\right\}\right\}$$

ПОРЯДОК ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Ознайомиться з принципом роботи та головними командами пакету Mathematica.
2. Повторити команди, наведені у теоретичній частині.
3. Зайти на сайт *www.wolfram.com* і повторити кілька виконаних прикладів в режимі он-лайн.
4. Скласти звіт з отриманих результатів