

УДК 655.533:004.915

ВИКОРИСТАННЯ ВИДАВНИЧОЇ СИСТЕМИ LaTeX ДЛЯ СТВОРЕННЯ ІЛЮСТРАЦІЙ У НАУКОВИХ РОБОТАХ

Влащенко Л.Г.

заступник директора Наукової бібліотеки,
Харківський національний університет радіоелектроніки

Дейнеко Ж.В.

к.т.н., професор, кафедра «Медіасистеми та технології»,
Харківський національний університет радіоелектроніки

Нікітенко О.М.

к.т.н., адміністратор баз даних Наукової бібліотеки,
Харківський національний університет радіоелектроніки

***Анотація.** Результати будь-якого дослідження як наукового, так і дослідницького спрямування для кращого розуміння вимагають графічної інтерпретації. Пакети tikz та pgfplots видавничої системи LATEX за допомогою команд plot та addplot допомагають будувати графіки різноманітних функцій. Здійснено порівняльний аналіз побудови графіків за допомогою plot та addplot. Наведено приклади побудови різноманітних графічних об'єктів за допомогою пакетів tikz та pgfplots.*

***Ключові слова:** LATEX, TEX, TIKZ, PGFPLOTS, PLOT, ADDPLOT.*

Вступ

Видавнича система LaTeX є зразком, до якого має наближатися будь-яка інша видавнича система. Таке висловлювання у тому чи іншому вигляді можна зустріти у багатьох виданнях, що присвячені опису системи LaTeX.

Отже, TeX-система надає повний контроль над структурою майбутнього документу, однак не створює ілюзій щодо того, що складні проблеми можуть бути вирішені просто: хоча перевизначені макети мають велику кількість параметрів, які слід налаштувати, створення повністю нового макета документу «з нуля» може виявитися непростою задачею й триватиме достатньо довго [1-5].

Відносно повільне розповсюдження TeX як засобу для наукових публікацій в Україні означає лише погане ознайомлення з можливостями цієї системи, здебільшого ідеальною для своєї області застосування.

Ілюстрації додають у наукову роботу для збільшення наочності матеріалу, який розглядають, та його кращого розуміння. Кількість графічного матеріалу в роботі визначають виключно доцільністю викладання.

До якості ілюстрації висувають такі вимоги:

– зображення не повинне мати спотворень, які отримано масштабуванням чи стисканням. Перевагу слід віддавати векторним зображенням перед растровими;

- під час друку зображення має бути чітким, тому слід враховувати можливості друкувального пристрою;
- креслення мають виконувати за допомогою спеціалізованих застосувань чи засобів;
- діаграми мають бути такими, що легко читаються. Тривимірні діаграми спотворюють інформацію, тому не рекомендуються;
- зображення, які сканують, повинні мати роздільну здатність, що дорівнює чи перевищує роздільну здатність друку [1].

Мета та задачі дослідження

Під час підготовки наукових статей, звітів та збірників конференцій, існує певна і значна ніша наукових публікацій, які містять складні таблиці, формули, графіки, гістограми, що ілюструють результати наукових досліджень або іншу важливу інформацію в зручному вигляді. Метою цього дослідження є порівняння побудови графіків за допомогою команд `plot` та `addplot` пакетів `tikz` та `pgfplots` видавничої системи `LaTeX`. Для досягнення цієї мети необхідно розв'язати такі задачі:

- продемонструвати можливості написання коду мовою `TEX` для наукових робіт різного спрямування;
- побудувати графічні залежності за допомогою команди `plot`;
- побудувати графічні залежності за допомогою команди `addplot`;
- порівняти результати побудови графічних залежностей за вище згаданими командами пакетів `tikz` та `pgfplots`.

Комбінація зазначених пакетів можна використовувати для створення високоякісної графіки за допомогою синтаксису системи `LaTeX`. `TikZ` генерує графіку на льоту, без потреби в окремих файлах для кожного зображення. Він також має дуже природний та інтуїтивно зрозумілий синтаксис

Основна частина

Є дві принципово різні можливості роботи з графікою в документі `LaTeX`. З одного боку – це створити зображення за допомогою спеціалізованих програмних пакетів, а потім додати його в документ. З іншого боку – створити її за допомогою базового чи спеціалізованих пакетів `LaTeX`.

Створювати малюнки прямо в документі `LaTeX` дозволяє оточення **`picture`**. Це оточення має значні обмеження на значення параметрів малювання відрізків та діаметрів кіл – вони можуть набувати тільки набір певних значень. Малюнок може бути вставлений у текст документу, або сформований як окремий об'єкт.

У документ можна вставити графіки, які створені безпосередньо засобами `LaTeX`. Зі всього розмаїття графічних можливостей слід розглянути два пакети:

- **`tikz`** – пакет для програмування векторної графіки;

– **pgfplots** – доповнення до попереднього пакету для побудови графіків на основі табличних даних. Для використання базових можливостей пакетів вони мають бути підключені в преамбулі документу.

```
\usepackage{tikz}
\usepackage{pgfplots}
```

Сам графік формується в межах спеціального оточення **tikzpicture**.

```
\begin{tikzpicture}
```

Код графіка [2].

```
\end{tikzpicture}
```

Отже метою цієї роботи є огляд можливостей побудови ілюстрацій за допомогою пакетів **tikz** та **pgfplots**, зокрема команд **plot** та **addplot**.

Результати будь-якого дослідження як наукового, так і дослідницького спрямування для кращого розуміння вимагають графічної інтерпретації. Такий підхід потребує побудови графічних залежностей різного роду.

Побудова графіків функцій – одна з найважливіших задач наукового графічного редактора.

Змінну величину y називають **функцією** від змінної величини x (аргументу), якщо кожному припустимому значенню x відповідає певне значення y .

Будь-яка функціональна залежність між двома величинами може бути зображена **графіком**. Для цього на площину наносять осі координат: горизонтальна – вісь абсцис і вертикальна – вісь ординат. По осі абсцис відкладають у певному масштабі різноманітні значення аргументу x – «абсциси» різноманітних точок графіка, по осі ординат – відповідні їм значення функції y – «ординати» тих же точок графіка. Кожна пара координат, абсциса та ордината утворюють одну точку графіка.

Існує велика кількість різноманітних систем для побудови графіків функцій та візуалізації даних [1, 6, 7]. Однак, існує певна і значна ніша, в якій застосовувати PGFPlots зручно: навчальні матеріали; різноманітні звіти, від звітів про виконання лабораторної роботи до звітів про наукові роботи; найпростіша візуалізація даних тощо.

Ілюстрації додають до наукової роботи для наочності матеріалу, який викладають, та його кращого розуміння. Кількість графічного матеріалу в роботі визначають виключно доцільністю викладення.

До якості ілюстрацій висувають такі вимоги:

- зображення не повинно мати спотворень, що отримано через масштабування чи стискання. Перевагу варто надавати векторним зображенням перед растровими;

- під час друку зображення має бути чітким, тому слід враховувати можливості друкувального пристрою;

- креслення мають створювати за допомогою спеціалізованих застосунків чи засобів. Рисунки від руки, а тим більше «мишею в Paint» не дозволяються;
- діаграми мають бути такими, що легко читаються. Тривимірні діаграми спотворюють інформацію, тому не рекомендуються;
- зображення, які сканують, повинні мати роздільну здатність, що дорівнює чи перевищує роздільну здатність друку.

Існує два способи подання графіки в пам'ять комп'ютера: растровий – де зображення подають матрицею точок, з певними значеннями кольору і векторний – де зображення створюється з графічних примітивів: відрізків прямих, дуг кіл чи еліпсів, сплайнів, а також фігур, що обмежені цими кривими. В пам'яті комп'ютера при цьому зберігається набір параметрів, які задають кожен фігуру та алгоритм побудови цих кривих [7, 8].

Перевагою векторного формату перед растровим є можливість задовільного масштабування чи обертання без спотворення зображення. Тому для ілюстрацій друкованих робіт надають перевагу векторним форматам перед растровими.

У видавничій системі LaTeX існує геть інша парадигма роботи с графікою.

TeX дозволяє будувати зображення безпосередньо в тілі документа. Для цього в місці розташування ілюстрації описують сам алгоритм побудови зображення, який будується під час компіляції. Отже, LaTeX бере на себе функції графічного редактора.

Існує кілька модулів для побудови зображень.

Одним з пакетів, що активно розвивається і є зручним, є **PGF/TikZ**, що написано професором Тілем Тантау (Till Tantau), співробітником інституту теоретичної інформатики Любекського університету, у 2007 році.

Ця система має три інтерфейсних рівні:

- системний рівень (PGFSys) складають команди, які безпосередньо здійснюють вимальовування зображення. Команди цього рівня можуть використовуватися в модулях, але не в кінцевому документі. Тобто користувач не взаємодіє з цим рівнем безпосередньо;

- базовий рівень (PGF – назва PFG є аббревіатурою перших літер фрази portable graphics format (графічний формат, який переносять)) – це надбудова над системним рівнем, яка реалізує всі можливості модуля PGF/TikZ. Команди цього рівня мають TeX синтаксис і доступні користувачеві. Область застосування інтерфейсу PGF – створення нових макросів і процедур, а також взаємодія з іншими модулями. Модуль PGF підключають командою `\usepackage{pgf}`;

- рівень користувача (TikZ – це рекурсивний акронім, що розшифровують як TikZ ist kein Zeichenprogramm. TikZ не є програмою для малювання. TikZ це лише оболонка. Малює PGF) – це оболонка, що забезпечує зручний інтерфейс для кінцевого користувача. TikZ підключають за допомогою команди `\usepackage{tikz}`. Під час підключення також стають доступними всі команди PGF [1].

TikZ надає кілька різних інтерфейсів для побудови графіків.

Тут ми порівнюємо команди **plot** та **addplot**, які дозволяють будувати криві та ламані за аналітично чи таблично заданими функціями. Приклади та опис параметрів взято з [1, 2, 7, 8].

1 Команда plot

Команда **plot** дозволяє доєднати до путі лінію чи криву, яка проходить через велику кількість точок. Ці точки або задають у простому списку точок, який читається з певного файлу, або обчислюється в реальному часі.

Існує багато потужних програм для побудови графіків функцій, наприклад, **gnuplot**, **maple** або **mathematica**.

Такі програми можуть створювати два різних типи:

– вивід повного графічного зображення у певному форматі (наприклад, pdf), який містить всі команди низького рівня, що є необхідними для того, щоб намалювати повний графік (включаючи осі координат та мітки);

– вивід простих таблиць даних у формі довгого списку координат [1-4].

Використовуючи код [3]:

```
\begin{tikzpicture}[domain=0:4]
\draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
\draw[->] (-0.2,0) -- (4.2,0) node[right] {$x$};
\draw[->] (0,-1.2) -- (0,4.2) node[above] {$f(x)$};
\draw[color=red] plot (\x,\x) node[right] {$f(x) = x$};
% \x r means to convert '\x' from degrees to _r_adians:
\draw[color=blue] plot (\x,{sin(\x r)}) node[right]
{$f(x) = \sin x$};
\draw[color=orange] plot (\x,{0.05*exp(\x)}) node[right]
{$f(x) = \frac{1}{20} \mathrm{e}^x$};
\end{tikzpicture}
```

можна побудувати графіки різноманітних функцій (рис. 1).

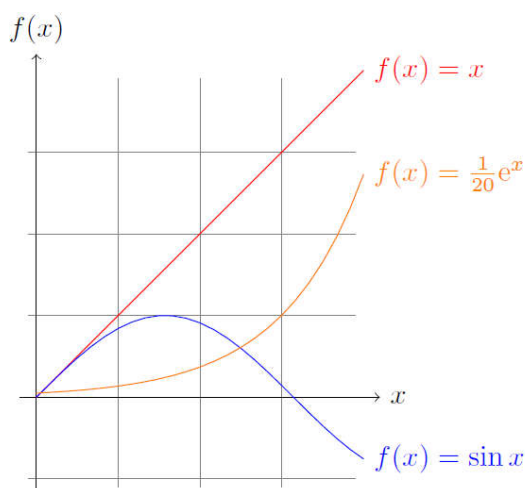


Рисунок 1 – Побудова графіків трьох функцій, заданих формулами

TikZ будує графіки функцій по точках. За замовчанням точки послідовно з'єднуються відрізками прямих, утворюючи ламану лінію. За допомогою параметрів примітиву **plot** можна зробити її гладкою, чи залишити лише вузлові точки.

Побудова функції, що задано аналітично

Функції, що задано аналітично, TikZ будує у два етапи:

- спершу обчислює точки вигляду $((x(t), y(t)))$;
- потім відтворює їх, за необхідності з'єднуючи відрізками.

Отже, необхідно визначити три параметри:

- ім'я змінної (variable);
- інтервал побудови графіка (domain);
- кількість вузлових точок (samples).

Змінною може бути будь-яка комбінація символів латинської абетки з початковим `\`.

Інтервал побудови графіка задають параметром **domain** у вигляді **domain = start:end**.

Кількість чи множина вузлових точок визначає параметр **samples** в одній з наступних форм.

```
samples=value
samples at ={list of value}
```

Перша форма визначає множину рівновіддалених точок, а друга дозволяє перелічити всі вузлові точки.

Розглянемо кілька прикладів [3].

Приклад 1.1. Побудова графіка функції.

Побудувати графік функції $y = \frac{x^2 - 3}{x^2 + 1}$ на відріжку $[-3; 3]$ (рис. 2).

```
\begin{tikzpicture}
\draw[->] (-3.2,0) -- (3.2,0)
node[right] {$x$};
\draw[->] (0,-3.2) -- (0,3.2)
node[above] {$y$};
\draw plot[domain=-3:3,variable=\x,samples=55,smooth]
(\x, {(pow(\x,2)-3)/(pow(\x,2)+1)});
\end{tikzpicture}
\end{verbatim}
```

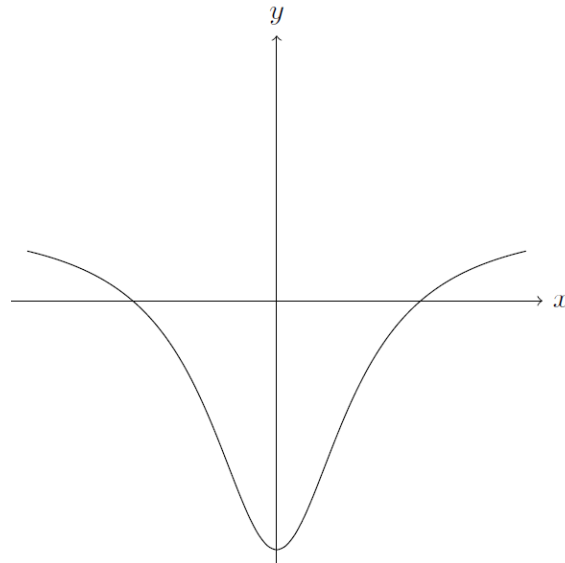


Рисунок 2 – Графік наведеної у прикладі функції

Приклад 1.2. Побудова графіка функції, що задано параметрично (рис. 3).

Дельтоїдою називають множину точок (x, y) , які задовольняють такі рівняння:

$$\begin{cases} x = 2 \cos t + \cos 2t, \\ y = 2 \sin t - \sin 2t. \end{cases}$$

```
\begin{tikzpicture}
\draw plot[domain=0:360,variable=\t, samples=360]
({2*cos(\t) + cos(2*(\t))},{2*sin(\t)
- sin(2*(\t))});
\end{tikzpicture}
```

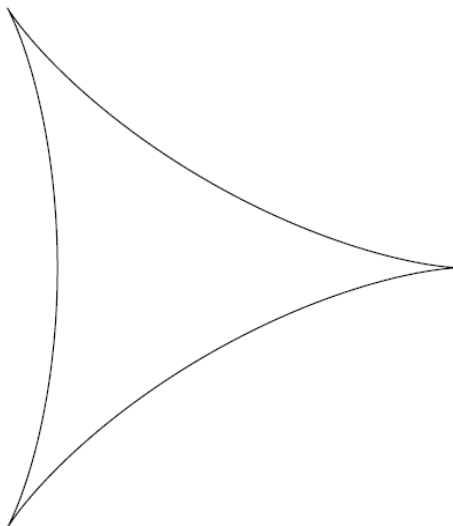


Рисунок 3 – Графік функції, наведеної у прикладі

У певних випадках **TikZ** некоректно опрацьовує довгі вирази. Уникнути цього можна, визначивши функцію за допомогою параметра **declare**.

```

declare function={ \
  F1(value list) = expression; \
  F2(value list) = expression; \
  ...}.

```

Приклад 1.3. Застосування визначених функцій бібліотеки TikZ (рис. 4).

```

\begin{tikzpicture}[
% визначення функції
declare function={
excitation(\t,\w) = sin(\t*\w);
noise = rnd - 0.5;
source(\t) = excitation(\t,20) + noise;
filter(\t) = 1 - abs(sin(mod(\t, 90)));
speech(\t) = 1 + source(\t)*filter(\t);
}
]
% побудова графіка
\draw [thick] plot [domain=0:500,
variable=\x, samples=501, smooth]
({\x/100},{speech(\x)});
\end{tikzpicture}

```



Рисунок 4 – Побудова графіку періодичної функції

Визначити функцію можна або в параметрах оточення **tikzpicture**, або в параметрах команди **\draw**.

Побудова функції, що задано таблично

Під час наукових досліджень аналітичне зображення залежностей, що вивчають, не є відомими. Однак можна виявити певні її значення під час експериментальних досліджень. У цьому випадку ми матимемо множину експериментальних точок у вигляді таблиці

x	x_1	x_2	\dots	x_n
y	y_1	y_2	\dots	y_n

У такому випадку побудову графіка здійснює така форма примітиву **plot**

plot[параметри побудови]

coordinates {(x_1, y_1) (x_2, y_2) ... (x_n, y_n)};

Приклад 1.4. Побудова багатокутника розподілу (полігона).

Випадкову величину X задано законом розподілу

x_i	1	2	3	4	5
y_i	0.1	0.2	0.4	0.1	0.2

Багатокутник розподілу є ламаною лінією. Через те, що ймовірність набуває значень на відрізку $[0; 1]$, визначимо розмір одиничного відрізка по вертикальній осі таким, що дорівнює 5 см (параметр $y = 5\text{cm}$)

```
begin{tikzpicture}[y=5cm]
% координатна сітка
\draw[ultra thin, color=gray] (0,0)
grid[xstep=1,ystep=0.1] (5,1);
% креслення осей
5 \draw[->] (-0.2,0) -- (5.2,0)
node[right] {$x$};
\draw[->] (0,-0.2) -- (0,1.2)
node[above] {$p$};
% нанесення шкал
\foreach \x in {0,...,5}
\draw[shift={(\x,0)}] (0pt,2pt) --
(0pt,-2pt) node[below] {\tiny $\x$};
10 \foreach \y in {0.1, 0.2, 0.4, 1}
11 \draw[shift={(0,\y)}] (2pt,0pt) --
(-2pt,0pt) node[left] {\tiny $\y$};
% Побудова багатокутника розподілу
\draw[thick] plot coordinates
{(1,0.1) (2,0.2) (3,0.4) (4,0.1)
(5,0.2)};
\end{tikzpicture}
```

Зазвичай таблицю значень формує спеціальна програма, яка фіксує результати експерименту або обчислень. Отримані значення доцільно розташовувати у файлі у вигляді таблиці. За допомогою наступного примітиву.

plot[параметри побудови] file {ім'я файлу}

Можна за таким файлом з визначеними параметрами, побудувати графік (рис. 5).

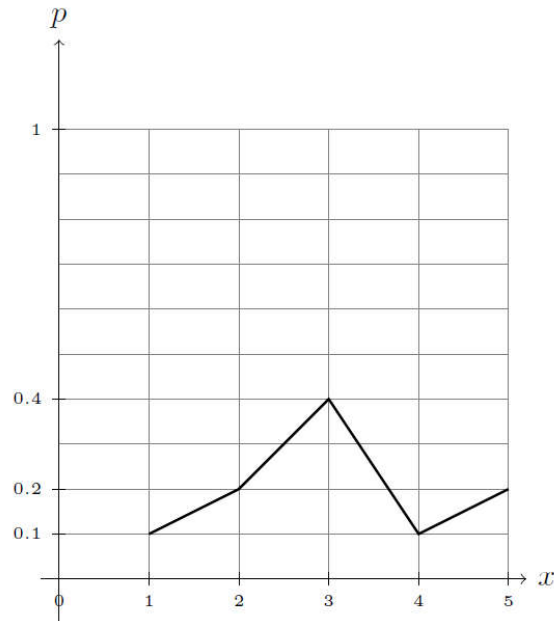


Рисунок 5 – Побудова багатокутника розподілу

Приклад 1.5. Читання даних для графіка з файлу (рис. 6.).
Функцією Гауса називають функцію

$$\varphi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}}.$$

У теорії ймовірностей вона є функцією щільності нормального розподілу $N(a, \sigma)$ з математичним сподіванням a та середнім квадратичним відхиленням σ . Побудуємо графік щільності нормального розподілу $N(0,1)$.

Спершу побудуємо таблицю значень функції, наприклад, за допомогою Excel і збережемо ці результати у файлі gauss.csv.

```
\begin{tikzpicture}[y=2cm]
% Креслення осей
\draw[->] (-2.1,0) -- (2.1,0) node[right] {$x$};
\draw[->] (0,-0.05) -- (0,0.6) node[above] {$p$};
% Нанесення шкал
\foreach \x in {-2,-1,...,2}
\draw[shift={(\x,0)}] (0pt,2pt) -- (0pt,-2pt)
node[below] {\tiny $\x$};
\foreach \y in {0.1,0.2,...,0.5}
\draw[shift={(0,\y)}] (2pt,0pt) -- (-2pt,0pt);
% Побудова графіка
\draw plot[thick] file {gauss.csv} node[above] {\tiny $p=\varphi(x)$};
\end{tikzpicture}
```

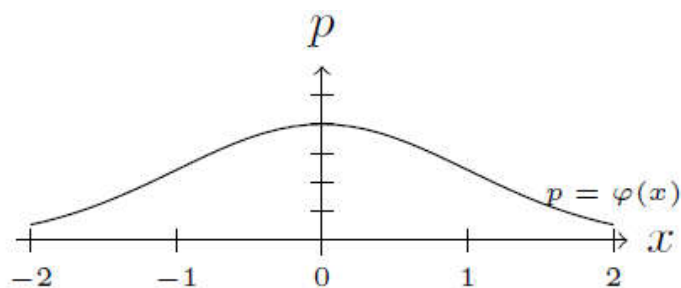


Рисунок 6 – Графік, побудований за даними із файлу

Гістограми

Гістограма – це специфічний вигляд подання графіка, тому для її зображення можна використовувати примітив **plot** зі спеціальними параметрами.

- **ybar** – вертикальна стовпчикова гістограма;
- **xbar** – горизонтальна гістограма;
- **ybar interval** – вертикальна інтервальна гістограма;
- **xbar interval** – горизонтальна інтервальна гістограма.

Діаграми характеризують такими параметрами:

- **bar width = value** – ширина прямокутника;
- **bar shift = value** – зсув прямокутників для суміщення кількох діаграм;
- **color = колір** – колір межі прямокутника;
- **fill = колір** – колір заливки;
- **pattern = штрихування** – вид штрихування;
- **pattern color = колір** – колір штрихування.

Приклад 1.6. Побудова гістограми за табличними даними (табл. 1, рис. 7).

Таблиця 1 – Табличні дані

Діапазон тестових балів	Відсоток робіт з дисципліни	
	Математика	Інформатика
0 – 10	3,5 %	1,2 %
11 – 20	2,7 %	2,5 %
21 – 30	10,5 %	3,2 %
31 – 40	18,6 %	3,7 %
41 – 50	13,0 %	14,5 %
51 – 60	26,4 %	17,5 %
61 – 70	17,3 %	18,7 %
71 – 80	5,1 %	20,3 %
81 – 90	2,2 %	12,4 %
91 – 100	0,7 %	6,0 %

Відобразимо дані з математики на гістограмі.

```
\begin{tikzpicture}[scale=0.125]
% координатний прямокутник
\draw (0,0) rectangle (100,30);
% нанесення шкал
```

```

\foreach \x in {0,10,...,100}
\draw[shift={(\x,0)}] (0pt,0pt) -- (0pt,-2pt)
node[below] {\tiny $\x$};
\foreach \y in {5,10,...,30}
\draw[shift={(0,\y)}] (0pt,0pt) -- (-2pt,0pt)
node[left] {\tiny $\y$};
% побудова діаграми
\draw plot[ybar interval,
color=black, fill=blue]
coordinates{
(0,3.5) (11,2.7) (21,10.5) (31,18.6) (41,13.0)
(51,26.4) (61,17.3) (71,5.1) (81,2.2) (91,0.7)
};
\end{tikzpicture}

```

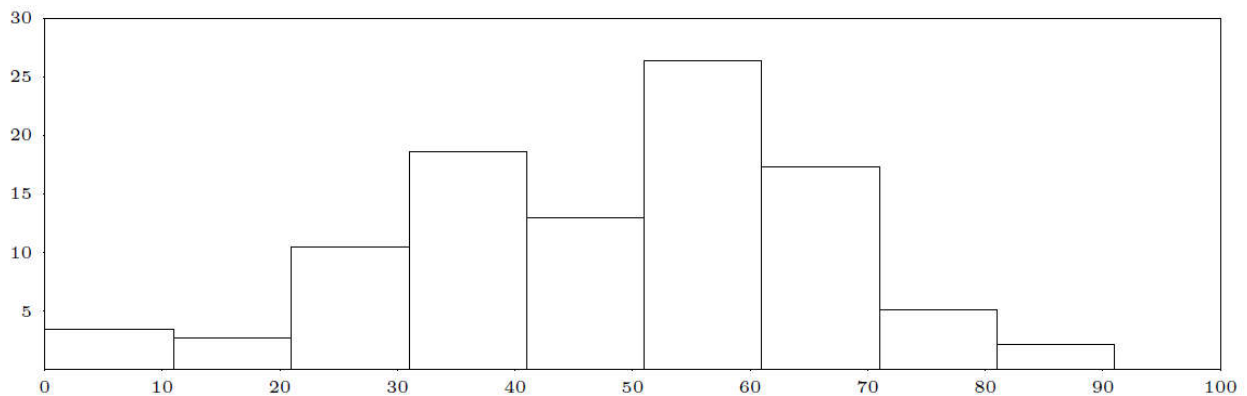


Рисунок 7 – Побудова гістограми за табличними даними

2 Команда `addplot`

В цьому розділі повторюються побудови з розділу Команда `plot` за допомогою команди `\addplot`.

Параметри цієї команди наведено в [2, 4].

```

\begin{tikzpicture}[domain=0:4]
\begin{axis}
[xlabel={$x$}, ylabel={$y$}, grid=major, legend pos={north west}]
\addplot +[draw=red,mark=none, ultra thick]{x};
\addlegendentry{$y=x$}
\addplot +[draw=blue,mark=none, ultra thick]{sin(deg(x))};
\addlegendentry{$y=\sin(x)$}
\addplot +[draw=orange,mark=none, ultra thick]{0.05*exp(x)};
\addlegendentry{$y=e^x$}
\end{axis}
\end{tikzpicture}

```

Будувати функції та графіки на основі математичного виразу або заданих даних в LaTeX дуже просто. Крім того, для якісної візуалізації даних на графіках необхідно створювати осі, додавати мітки та легенди та змінювати стиль графіків. Один із параметрів при визначенні функції `domain = a:b` визначає діапазон зміни даних. В наведеному вище прикладі, `domain = 0:4`, тобто область визначення функції $[0, 4]$. Область визначення функції не залежить від меж осей, але зазвичай для отримання графіків, які заповнюють вісь, потрібні однакові значення (рис. 8).

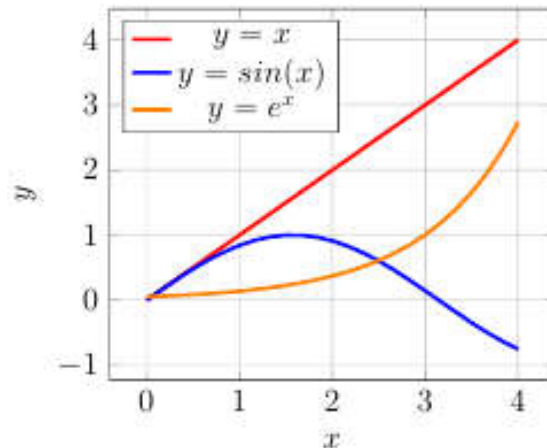


Рисунок 8 – Графіки трьох функцій, побудовані командою `\addplot`

Приклад 2.1. Побудова графіка функції.

Побудувати графік функції $y = \frac{x^2 - 3}{x^2 + 1}$ на відрізку $[-3; 3]$ (рис. 9).

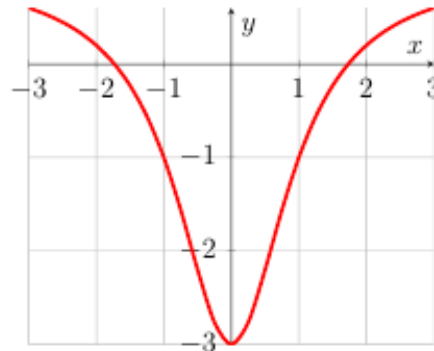


Рисунок 9 – Графік функції з використанням команди `\addplot`

```
\begin{tikzpicture}
\begin{axis}
[xlabel={\$x\$}, ylabel={\$y\$}, axis lines=middle, grid=major,
legend pos={north west}]
\addplot+[domain=-3:3,draw=red,mark=none,smooth,ultra thick]
{(x^2-3)/(x^2+1)};
\end{axis}
\end{tikzpicture}
```

Приклад 2.2. Побудова графіка функції, що задано параметрично (рис. 10).

```
\begin{tikzpicture}
\begin{axis}
\addplot+[domain=0:2*pi, draw=red,mark=none,smooth, ultra
thick, samples=360]
({2*cos(deg(x)) + cos(2*deg(x))},{2*sin(deg(x))- sin(2*deg(x))});
\end{axis}
\end{tikzpicture}
```

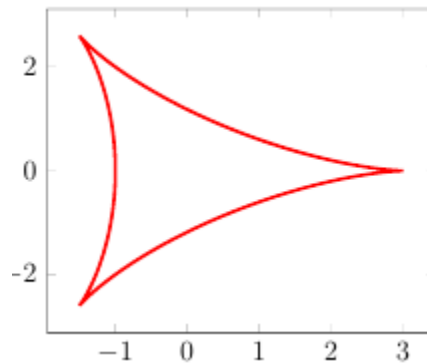


Рисунок 10 – Графік функції з використанням команди \addplot

Приклад 2.3. Застосування визначення функції (рис. 11).

```
\begin{tikzpicture}
\begin{axis}
% побудова графіка
\addplot+[domain=0:500, samples=501, mark=none,smooth]
{
(sin(deg(0.2*x))+rnd-0.5)*(1-abs(sin(deg(mod(x/30,90))))))
};
\end{axis}
\end{tikzpicture}
```

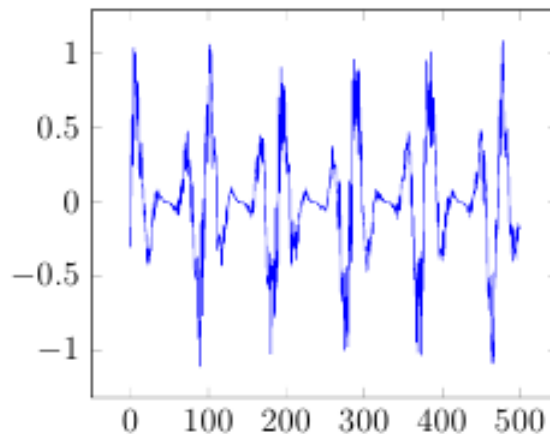


Рисунок 11 – Використання команди \addplot для побудови графіку

Приклад 2.4. Побудова багатокутника розподілу (полігонів) (рис. 12).

```
\begin{tikzpicture}
\begin{axis}[xlabel={\$x\$}, ylabel={\$p\$}, axis lines=middle, xstep=1,
ystep=0.1, grid=major, mark=none]
% задаються координати точок для побудови графіка
% координати точок задаються у форматі: (x,y)
\addplot coordinates
{(1,0.1) (2,0.2) (3,0.4) (4,0.1) (5,0.2)};
\end{axis}
\end{tikzpicture}
```

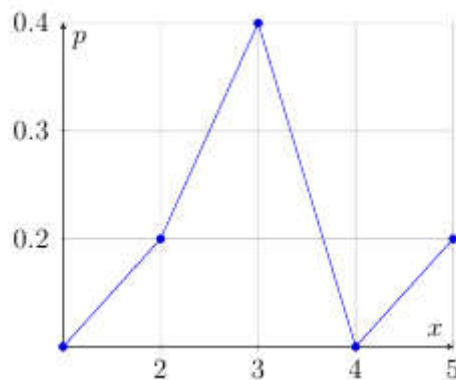


Рисунок 12 – Побудова багатокутника розподілу командою `\addplot`

Приклад 2.5. Читання даних для графіка з файлу (рис. 13).

```
\begin{tikzpicture}[scale=0.9]
% задано розмір графіка height=10cm,width=10cm
\begin{axis}[xlabel={\$x\$}, ylabel={\$p\$},
axis lines=middle,grid=major,mark=none,
legend style={at={(0.5,0.2)}, anchor=north,legend columns=-1}]
]
% дані читаються з файлу gauss.csv
\addplot file {gauss.csv};
% до графіка додається легенда: Експеримент
\addlegendentry{\$p=\varphi(x)\$}
\end{axis}
\end{tikzpicture}
```

Застосування файлів в форматі *.csv або визначення функції CSV (від англійської Comma-Separated Values – значення, розділені комами) – текстовий формат, призначений для представлення даних у вигляді таблиці. У таких файлах для розділення тексту на стовпчики використовується спеціальний символ – роздільник. Як роздільник, як правило, виступає кома, точка з комою або табуляція.

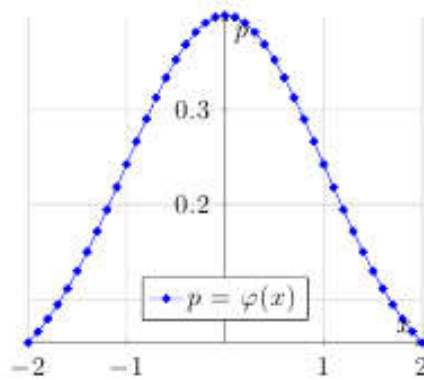


Рисунок 13 – Використання команди `\addplot` для створення графіку по даним із файлу

Приклад 2.6. Побудова гістограми (рис. 14).

```
\begin{tikzpicture}
\begin{axis}[ybar interval,width=15cm]
\addplot
coordinates {(0,3.5) (11,2.7) (21,10.5) (31,18.6) (41,13.0)
(51,26.4) (61,17.3) (71,5.1) (81,2.2) (91,0.7)};
\end{axis}
\end{tikzpicture}
```

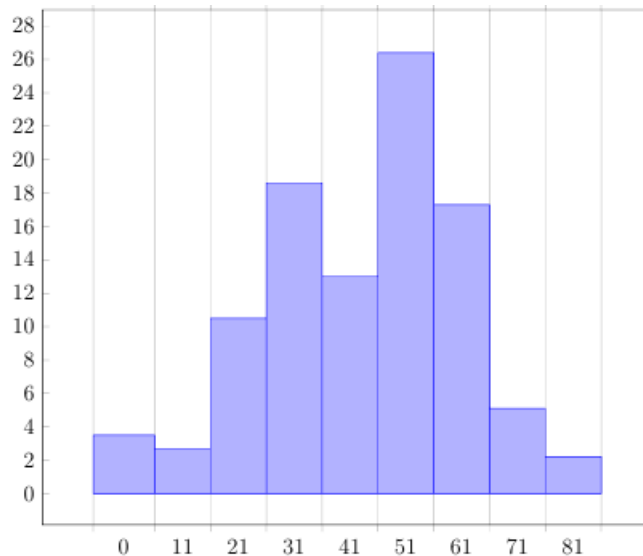


Рисунок 14 – Побудова гістограми з використанням команди `\addplot`

Деякі інші застосування пакету `tikz`

Візуальне пояснення теореми Піфагора представлено наступним кодом (рис. 15).

```
\begin{tikzpicture}[scale=0.75]
\fill[left color=blue, right color=yellow]
(0,0) -- node[below=3pt] {$a$} (4,0) --
node[right=5pt] {$b$} (4,3) --
cycle node[midway,above,sloped] {$c=\sqrt{a^2+b^2}$};
```



```

\node[below left] at (0,0) {\color{blue}$B$};
\node[below right] at (4,0) {\color{blue}$C$};
\node[above right] at (4,3) {\color{blue}$A$};
\end{tikzpicture}

```

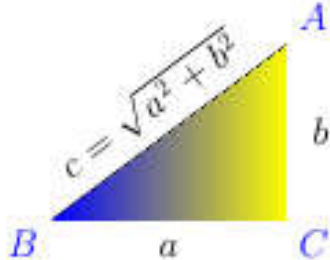


Рисунок 15 – Графічне представлення теореми Піфагора

Геометрична інтерпретація квадрата суми

Квадрат суми двох чисел дорівнює квадрату першого числа плюс подвоєний добуток першого числа на друге плюс квадрат другого числа. З правила випливає, що загальна формула квадрата суми, без проміжних перетворень, матиме такий вигляд: $(a + b)^2 = a^2 + 2ab + b^2$ (рис. 16).

```

\begin{tikzpicture}
\fill[red] (0,0) rectangle (3,3);
\fill[blue] (3,3) rectangle (4,4);
\fill[green] (0,3) rectangle (3,4);
\fill[green] (3,0) rectangle (4,3);
\path (0,0) -- node[left] {$a$} (0,3) -- node[left] {$b$} (0,4);
\path (0,0) -- node[below=3pt] {$a$} (3,0) -- node[below] {$b$} (4,0);
\node[circle,fill=white,inner sep=2pt]
at (1.5,1.5) {$a^2$};
\node[circle,fill=white,inner sep=2pt]
at (3.5,3.5) {$b^2$};
\node[ellipse,fill=white,inner sep=2pt]
at (1.5,3.5) {$a \cdot b$};
\node[ellipse,fill=white,inner sep=2pt,rotate=90]
at (3.5,1.5) {$a \cdot b$};
\end{tikzpicture}

```

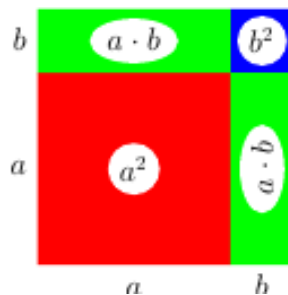


Рисунок 16 – Графічна інтерпретація квадрата суми

Висновки

Розглянуто можливості побудови графіків функцій за допомогою двох команд з пакету `tikz – plot` та `addplot`. Проведено порівняння роботи цих команд для побудови різноманітних графічних зображень, які зустрічаються у наукових публікаціях. У результаті порівняння їхньої роботи виявлено, що команда `addplot` є гнучкішою за команду `plot`. Наведено додаткові можливості побудови ілюстрацій, які не є функціональними залежностями.

Список літератури.

1. Нікітенко, О.М. (2021). LaTeX в дії. Рекомендації з використання видавничої системи LATEX для студентів, науковців, викладачів. Харків: ХНУРЕ.
2. Грищенко, Т.Б., Дейнеко, Ж.В., & Нікітенко, О.М. (2019). Використання системи LaTeX під час підготовки наукових публікацій. PRINT, MULTIMEDIA & WEB: тези доп. IV Міжнар. наук.-техн. конф. (14-17 травня 2019, м. Харків). Т. 1, 96-99.
3. Tantau, T. (2015). The TikZ and PGF Packages. Manual for version 3.0.1a. Institut fur Theoretische In-formatik Universitat zu Lubeck. <http://elib.ict.nsc.ru/jspui/handle/ICT/1487/1/pgfma-nual.pdf>.
4. Orobinskyi, P., Deineko, Z., & Lyashenko, V. (2020). Comparative Characteristics of Filtration Methods in the Processing of Medical Images. American Journal of Engineering Research, 9(4), 20-25.
5. Deineko, Z., Zeleniy, O., Lyashenko, V., & Tabakova, I. (2021). Color space image as a factor in the choice of its processing technology. Abstracts of I International scientific-practical conference «Problems of modern science and practice» (September 21-24, 2021). Boston, USA, 389-394.
6. Lyashenko, V., Deineko, Z., Zeleniy, O., & Tabakova, I. (2021). Wavelet ideology as a universal tool for data processing and analysis: some application examples. International Journal of Academic Information Systems Research (IAISR), 5(9), 25-30.
7. Нікітенко, О.М., Дейнеко, Ж.В., & Грищенко, Т.Б. (2021). Дослідження переваг застосування графіків у системі LaTeX при оформленні наукових праць. Актуальні проблеми теорії керуючих систем у комп'ютерних науках: праці науково-практичної конференції (21-24 грудня 2021 р. Слов'янськ), 79-84.
8. Грищенко, Т.Б., Нікітенко, О.М., Дейнеко, Ж.В. (2021). Створення електронних підручників засобами видавничої системи LaTeX. У В.П. Ткаченко, О.В. Вовк, І.Б. Чеботарьова (Ред.), Поліграфічні, мультимедійні та web-технології: колективна монографія (с. 80-96). Харків: ТОВ «Друкарня Мадрид».