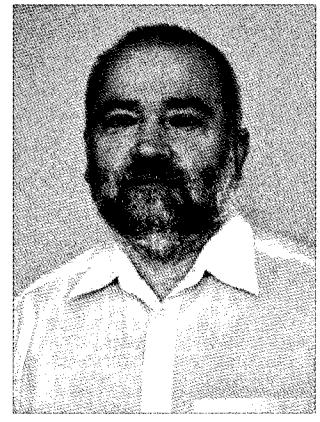
ВЕКТОРНА ГРАФІКА В LATEX ЗАСОБАМИ TIKZ



Рудик О. Б.

Ця публікація є безпосереднім продовженням попередніх публікацій автора, що описували можливості LaTeX 90-х років XX століття. Того, що давало можливість зручно для автора підготувати текст із численними математичними формулами зі складною структурою. Підготувати у формі, максимально доступній для сприйняття читачем. Водночас можна було, наприклад, точно подати графіки функцій і легко подати зображення розгорток многогранників. Для простих ілюстрацій планіметричних і стереометричних задач цього було достатньо. Але потреба подавати складні схеми, діаграми та ще й у кольорі потребувала нововведень. Таким нововведенням, що долучило до LaTeX графічний векторний редактор, став пакет Pgf/TikZ (входить до складу MiKTeX). Ця публікація є спробою подати стислий опис можливостей пакета Pgf/TikZ без «занурення до дна» детального опису — 725 сторінок формату A4 англійською мовою. Наш стислий опис неповний, але достатній для успішної роботи в більшості випадків. Посилання на ресурси глобальної мережі дозволить читачам, образно кажучи, «не лише пірнути до дна, але й покопатися». У кінці публікації описано процес розширення вільнопоширюваного редактора векторної графіки Inkscape до експорту результатів у TikZ.

Публікацію призначено для учнів класів із поглибленим вивченням математики чи фізики, студентів математичних і фізичних спеціальностей, учителів математики й фізики, наукових працівників.

1. Спосіб використання

Трансляція — перетворення формату з tex на pdf — результат виконання такої вказівки: pdflatex *назва файлу*.tex. Тут і далі курсивом виділено назви понять, замість яких користувач має записати назви об'єктів або числові величини.

У преамбулі потрібно замовити:

- використання пакета вказівкою:
  \usepackage{tikz};
- використання бібліотек пакета, наприклад, так:
  \usetikzlibrary{positioning,arrows,through,
  patterns,calc}

Параметри вказують на можливість такого:

- positioning — відносного позиціонування;
- arrows — креслення стрілок;
- through — проведення кола через задану точку;
- patterns — штрихування;
- calc — обчислень.

Всередині середовища document вказівки рисунка розташовують у середовищі tikzpicture або використовують як аргумент вказівки \tikz. Наприклад, і код:

```
\tikz{\draw (-1,0) — (1,0);
\draw (0,-1) — (0,1);}
```

і код:

```
\begin{tikzpicture}\draw (-1,0) — (1,0);
\draw (0,-1) — (0,1);
\end{tikzpicture}
```

призведуть до одного й того самого зображення двох взаємно перпендикулярних відрізків (рис. 1).

При цьому кожну вказівку завершують крапкою з комою. У поданому прикладі використано вказівку \draw. Згідно із замовчуванням TikZ тлумачить усі розміри в сантиметрах, але їх можна задати й стандартним способом. Розташування центра координат (0,0) є відносним.

Рис. 1

Середовище tikzpicture має параметр scale, за домомогою якого можна збільшувати чи зменшувати розмір рисунка без зміни величин координат об'єктів на рисунку.

2. Синтаксис вказівок

Загальний синтаксис вказівок такий:

\command [parameters] (name) {contents} arguments;

Тут

command — вказівка;

parameters — перелік параметрів через кому;

name — назва створюваного об'єкта (на уподобання користувача);

contents — вміст об'єкта (може містити інші об'єкти);

arguments — аргументи, наприклад, координати точок шляху, розміри чи інші вказівки без \.

Обов'язковою є наявність лише власне вказівки. У tikz багато вказівок, але в багатьох випадках достатньо використання лише path і node.

Вказівка \path створює контур. Її параметри описують, як цей контур використовують:

draw — лише накреслити контур;

fill — лише залити;

fill,draw — накреслити контур і залити;

use as bounding box — використати контур як обмеження зображення.

Замість

\path[draw], \path[fill], \path[shade,draw], ...

використовують скорочення

\draw, \fill, \shadedraw.

Як параметр \path також вказують на наявність і напрям стрілок, колір, товщину лінії тощо.

Колір можна задати так:

- **назвою кольору**, попередньо означеною в LaTeX і пакеті **xcolor**. Наступні назви є завжди досяжними: black, blue, brown, cyan, darkgray, gray, green, lightgray, lime, magenta, olive, orange, pink, purple, red, teal, violet, white, yellow;
- **змішуванням кольорів** з попередньо означеними назвами. Синтаксис запису змішування такий: *колір*$_1$*!відсоток!колір*$_2$. У результаті отримують

колір, що містить *відсоток*% *кольору*$_1$ і (100 – *відсоток*)% *кольору*$_2$. Наприклад, red!20!yellow означає змішування червоного й жовтого кольорів у пропорції 20:80. Білий колір можна не вказувати. Дії змішування виконують у порядку їх запису зліва направо — див. далі приклад задання стилю вузла для вказівки \node;

- **запровадження нових кольорів** здійснюють:

вказівкою \definecolor{*своя назва*}{rgb}{*інтенсивність червоного, інтенсивність зеленого, інтенсивність блакитного*}, де інтенсивності кольорів задано дійсними числами в межах від 0 до 1 включно;

вказівкою \colorlet{*своя назва*}{*код кольору, заданий допустимим чином*}.

Змінити колір тексту можна такою вказівкою \textcolor{*колір*}{text}.

Щодо решти параметрів — див. далі опис вказівки \draw та інших оболонок над \path.

Аргументами є координати точок, через які має проходити контур. У tikz можна використати такі способи задання координат (після назви подано приклади використання):

- декартові відносні (1,5);
- декартові абсолютні на аркуші (11pt, 22mm);
- полярні кут-відстань (3:2);
- декартові відносно попередньої точки ++(0,1);
- декартові відносно початкової точки +(–1,2);
- барицентричні відносно двох точок (1,2)!.3!(3,4) — координати точки, розташованої на відстані 3/10 шляху з (1,2) до (3,4);
- відстань на промені (1,2)!3cm!(3,4) — координати точки, розташованої на відстані 3 см від початку променя (1,2) на промені, що містить точку (3,4);
- проекція на дану пряму (1,2)!(3,0)!(3,4) — координати проекції точки (3,0) на пряму, що містить точки (1, 2) і (3, 4).

Координати точок, між якими потрібно провести лінію, розділяють символами – –.

Назви об'єктів (у круглих дужках), межі яких потрібно сполучити, розділяють символами edge:

\path (*об'єкт*$_1$) edge (*об'єкт*$_2$);

Вказівка \node створює вузол, що, зазвичай, містить текст. Її параметрами можуть бути стиль тексту, колір, інформація щодо наявності, форми й кольору меж, розташування відносно інших об'єктів тощо. Розташовують вузол у точці з координатами (*x*, *y*) за допомогою аргумента at (*x*, *y*). Розташування відносно інших об'єктів здійснюють, використовуючи бібліотеку positioning. Наприклад, вказівка:

\node[right of=*назва*$_1$] (назва$_2$) {*текст*}

створить вузол *назва*$_2$ праворуч від вузла *назва*$_1$ і розташує в ньому *текст*.

Стиль оформлення вузла задають, якщо кілька вузлів потрібно зобразити однаково. Для цього використовують вказівку \tikzstyle. Подамо приклад такого використання з коментарями після символа % (табл. 1).

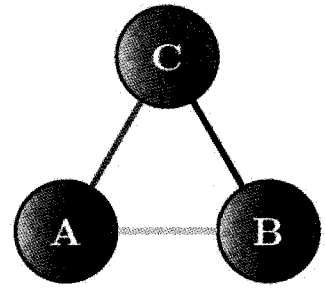


Рис. 2

Після такого означення стиль format можна вказати як параметр відповідного вузла \node. Зображення на рис. 2 створено за допомогою такого коду (тут і далі подано лише вміст середовища tikzpicture):

```
\tikzstyle{format} = [
 circle,
 thick,
 minimum size=1cm,
 draw=green!50!red!50!black,
 right color=red,
 left color=green,
 font=\bfseries
]
\node[format] (A) at (0,0) {\textcolor{white}{A}};
\node[format] (B) at (2,0) {\textcolor{white}{B}};
\node[format] (C) at (1,1.732) {\textcolor{white}{C}};
\path[green, ultra thick] (A) edge (B);
\path[blue, ultra thick] (B) edge (C);
\path[brown, ultra thick] (C) edge (A);
```

3. Лінії

Для зображення ліній використовують вказівку \draw. Довільну накреслену криву можна замкнути так: \draw ... – – cycle;. У квадратних дужках після \draw через кому можна вказати:

товщину (подано у порядку зростання):
ultra thin;
very thin;
thin;
semithick;
thick;
very thick;
ultra thick;
або вказати товщину явно, наприклад:
line width=4pt ;

тип:
solid;
dotted;
densely dotted;
loosely dotted;
dashed;
densely dashed;
loosely dashed;
dashdotted;
densely dashdotted;
loosely dashdotted;
dashdotdotted;

Таблиця 1

\tikzstyle{format} = [	
rounded rectangle,	% прямокутник зі зглаженими краями
thick,	% контур жирною лінією
minimum size=1cm,	% мінімальний розмір
draw=red!50!black!50,	% контур має колір: 25% червоного + 25% чорного + 50 білого
top color=white,	% білий згори для градієнтної заливки згори донизу
bottom color=red!50!black!20,	% 10% червоного + 10% чорного + 80% білого знизу
font=\itshape]	% похилий шрифт тексту

densely dashdotdotted;
loosely dashdotdotted.

Подамо зображення згаданих типів у тому самому порядку переліку (рис. 3).

Стрілку:

Рис. 3

- стрілка ліворуч або праворуч (точніше кажучи, спрамована на початкову або на кінцеву точку): <– або –>
- стрілки на обох кінцях лінії: <–>;
- подвійна стрілка ліворуч або праворуч: <<– або –>>;
- подвійні стрілки на обох кінцях лінії: <<–>>;

Подамо зображення, отримане за допомогою такого коду (рис. 4):

```
\draw[->] (0,0) -- (2,0);
\draw[>->>] (0,-.5) -- (2,-.5);
\draw[|<->|] (0,-1) -- (2,-1);
\draw[o-*] (0,-1.5) -- (2,-1.5);
\draw[) -)] (0,-2) -- (2,-2);
```

Колір (див. вище пункт 2. Синтаксис вказівок).

Задані величини параметрів застосовують до всіх фігур, що виводять вказівкою \draw до наступного переозначення параметрів у квадратних дужках.

Після вказівки \draw (з визначенням величин параметрів у квадратних дужках чи без цього) у круглих дужках потрібно вказати початкову точку кривої, центр фігури (для кола, еліпса або дуги) або одну із точок прямокутної області. Наприклад, \draw(2,3).

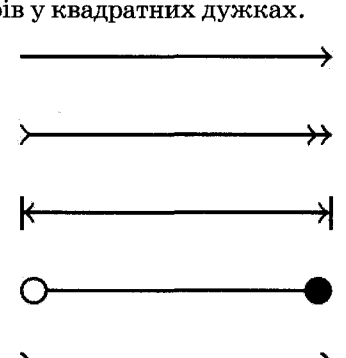


Рис. 4

Далі записують модифікатор — тип лінії. Тлумачення чисел у круглих дужках після модифікатора залежить від модифікатора. Різні модифікатори можна розташовувати послідовно всередині однієї вказівки \draw.

Проведення відрізка до точки (x, y) задають так: — (x, y). Ламану з кількох ланок можно накреслити, вказавши координати послідовних вершин у дужках через кому. Наприклад, вказівка:

```
\draw (0,0) -- (1,1) -- (0,1) -- (1,0) -- (0,0);
```

зобразить ламану, утворену діагоналями квадрата й парою його протилежних сторін, паралельних осі абсцис Ox (рис. 5).

Породження вузлів до проведення ламаної подамо прикладом такого коду:

```
\tikzstyle{every
 node}=[draw,shape=circle,thick];
\node (v1) at (360/7:2) {$V_1$};
\node (v2) at (2*360/7:2) {$V_2$};
\node (v3) at (3*360/7:2) {$V_3$};
\node (v4) at (4*360/7:2) {$V_4$};
\node (v5) at (5*360/7:2) {$V_5$};
\node (v6) at (6*360/7:2) {$V_6$};
\node (v7) at ( 0:2) {$V_7$};
\draw[thick] (v1) -- (v2) -- (v3) --
 (v4) -- (v5) -- (v6) -- (v7) -- (v1);
```

що породжує таке зображення (рис. 6).

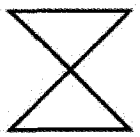


Рис. 5

Породження вузлів при проведенні ламаної подамо прикладом такого коду:

```
\draw (0,0) node [below left] {$A_1$} --
 (1,0) node [below right] {$A_2$} --
 (1,1) node [above right] {$A_3$} --
 (0,1) node [above left] {$A_4$} --
 cycle;
```

що породжує таке зображення (рис. 7).

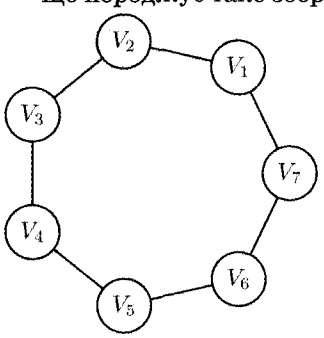


Рис. 6

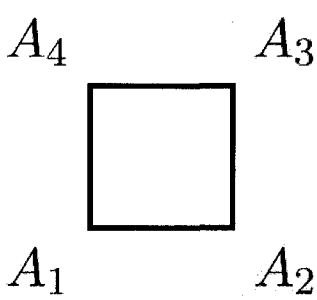


Рис. 7

Ґратку записують, наприклад, так:

```
\draw[step=.1cm] (1,2) grid (3,4);
```

У поданому прикладі параметр step визначає крок ґратки, а пари чисел до й після модифікатора grid — область [1,2]Ч[3;4] для покриття ґраткою.

Круг з радіусом r записують так: circle (r).

Еліпс з осями a й b записують так: ellipse (a and b).

Дугу круга з кутовим аргументом від α до θ й радіусом r записують так: arc (α : θ : r).

Дугу еліпса з кутовим аргументом від α до θ й осями a й b записують так: arc (α : θ : a and b). При зображенні дуг центр кола чи еліпса розраховується автоматично, а координати arc задають початкову точку дуги.

Прямокутник, у якому координати однієї вершини задано безпосередньо після запису вказівки \draw до модифікатора, а протилежна вершина має координати (x, y), записують так: rectangle (x, y). Заокруглити кути прямокутника дугою кола з радіусом r можна, вказавши параметр [rounded corners=r].

Кубічну криву Без'є задають такою вказівкою (A) .. controls (X) and (Y) .. (B). Тут (A) та (B) — кінці кривої, (X) та (Y) — опорні точки, розташування яких впливає на форму кривої. Відсутність частини and (Y) означає (Y) = (X). Водночас можна створювати вузли у точках параметризованої кривої (параметр зростає від 0 до 1 при русі точки від початку до кінця), задаючи величину параметра — див. результат виконання такого коду (рис. 8):

```
\draw[line width=2pt] (0,0) .. controls (1,3) .. (4, 0)
  node[pos=0,left] {0}
  node[pos=.2,left] {0.2}
  node[pos=0.7,above]{0.7}
  node[pos=1,right]{1}
```

Кубічну криву можна задати, вказавши початкову й кінцеву точки та кутові аргументи дотичних векторів у цих точках. Наприклад, так (рис. 9):

```
\draw[line width=2pt,->] (0,0) to [out=150, in=180] (4,2);
\draw[line width=2pt,->] (0,0) to [out=−90, in=0] (4,2);
```

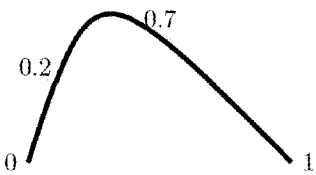


Рис. 8

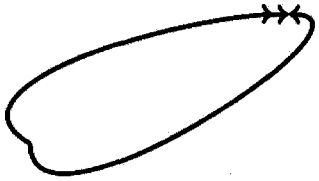


Рис. 9

4. Заливка

Замкнений контур із заливкою того самого кольору можна намалювати з допомогою вказівки \fill, синтаксис використання якої такий самий, як для вже розглянутої вказівки \draw.

При різних кольорах контуру й заливки використовують таку вказівку:

\filldraw[fill=*колір заливки*, draw=*колір контуру*] *запис контуру такий самий, як для* \draw.

Прозорість визначають параметром opacity, що набуває дійсних величин у межах від 0 до 1. Повній непрозорості відповідає 1.

Різні прозорості тла й контуру задають відповідно параметрами fill opacity й draw opacity.

Для прикладу подамо зображення (рис. 10), створене за допомогою такого коду:

```
\draw[step=1,gray] (−4,−3) grid (4,3);
\draw[thick,violet,densely dashed] (0,0)
 ellipse (4 and 3);
\draw[thick,red,densely dotted] (0,0) circle (3);
\fill[thick,green] (0,1.5) arc (90:0:2 and 1.5)
 arc (180:−90:.5) — cycle;
\fill[thick,cyan,opacity=0.5] (−1.5,1.5)
 rectangle (−2.5,−1.5);
\draw[thick,blue] (0,0) .. controls (1,−1)
 and (−1,−1) .. (0,−3);
```

5. Перетворення зображення

Поворот фігури здійснюють, вказавши у градусах параметр rotate вказівки виведення зображення — \draw, \fill або \filldraw)

Паралельне перенесення на вектор $v(x, y)$ можна здійснити, вказавши параметри: xshift=x, yshift=y. Якщо перенесення здійснюють у вертикальному чи горизонтальному напрямку, то відповідне надання нульової величини *shift=0 можна не записувати.

Вирізання частини зображення, обмеженого контуром, без зображення цього контуру здійснюють вказівкою \clip, синтаксис використання якої щодо запису контуру такий самий, як для розглянутої вказівки \draw. Ця вказівка нічого не малює, але впливає на зображення фігур, заданих наступними за нею вказівками.

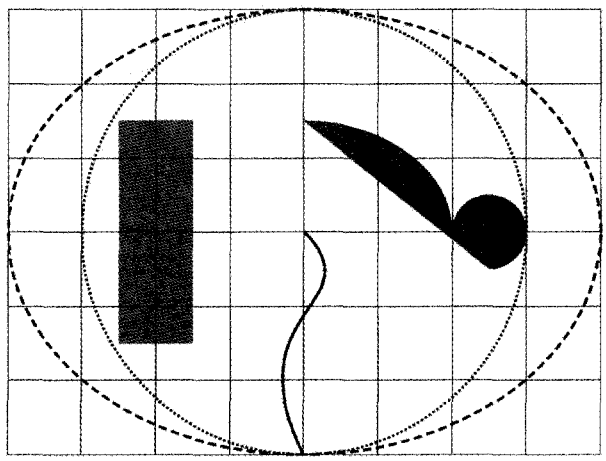


Рис. 10

Вирізання частини зображення, обмеженого контуром, разом із зображенням цього контуру здійснюють вказівкою \draw з опцією clip (вказаною у квадратних дужках).

Обмеження дії вирізання зображення здійснюють розташуванням вказівки вирізання й наступних вказівок побудови фрагментів зображення всередині середовища \begin{scope}...\end{scope}.

Для прикладу подамо зображення (рис. 11), створене за допомогою такого коду:

```
\draw[rotate=60,clip] (0,0) ellipse (4 and 3);
\draw[step=.5,gray] (−4,−4) grid (4,4);
\fill[black] (0,0) circle (.1);
\fill[thick,green,rotate=30] (0,0) rectangle (1,1);
\fill[thick,green,rotate=30,xshift=1cm,
 yshift=2cm] (0,0) rectangle (1,1);
\fill[thick,red,xshift=1.5cm,yshift=2cm] (0,0)
 rectangle (.5,.5);
\fill[thick,red,xshift=1.5cm,yshift=2cm,
 rotate=120] (0,0) rectangle (.5,.5);
```

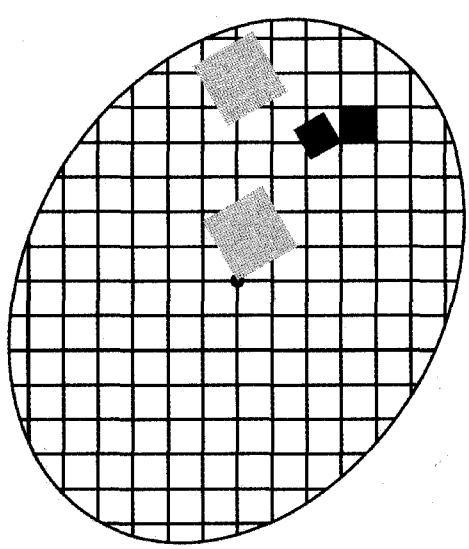


Рис. 11

Поданий приклад наочно переконує, що результат повороту й паралельного перенесення залежить від порядку виконання дій.

(Далі буде)