

Mục lục

I	NHỮNG HÌNH VẼ CƠ BẢN	2
I.1	Khai báo và môi trường vẽ	2
I.2	Các đối tượng hình học	2
I.2.1	Hệ trục tọa độ	3
I.2.2	Vẽ Điểm	4
I.2.3	Đoạn thẳng	5
I.2.4	Đường tròn	6
I.2.5	Đường Ellip	7
I.2.6	Cung tròn, Ellip - Đường cong	7
I.2.7	Hiển thị tên đối tượng trên hình vẽ	10
I.3	Vẽ các hình đơn giản đầu tiên	10
I.3.1	Khai báo các điểm dùng nhiều lần	10
I.3.2	Giao điểm của các đường trong Tikz	14
I.3.3	Vẽ ký hiệu vuông góc	19
I.3.4	Phép Quay	20
I.3.5	Vẽ đường vuông góc của một đường	20
I.3.6	Tính toán trong Tikz	21
I.3.7	Vẽ hình bình hành	22
I.3.8	Vẽ hình tròn và tam giác nội tiếp	23
I.3.9	Vẽ hình trụ	25
I.3.10	Vẽ hình nón	26
I.3.11	Vẽ mặt cầu	26
I.3.12	Hình lăng trụ	27
I.4	Làm đẹp hình vẽ	28
I.4.1	Tô màu cho đường nét	28
I.4.2	Tô màu cho một miền kín	29
II	BẢNG BIẾN THIÊN-ĐỒ THỊ HÀM SỐ	32
II.1	Bảng biến thiên	32
II.2	Đồ thị hàm số	35
II.3	Tô một miền đồ thị	38
II.4	Đồ thị hàm số trong hệ tọa độ cực	41

PHẦN I. NHỮNG HÌNH VẼ CƠ BẢN

I.1 Khai báo và môi trường vẽ

Khai báo gói lệnh:

```
\usepackage{tikz}
\usetikzlibrary{calc,angles}%Gọi các thư viện cần thiết phục vụ vẽ hình
```

Môi trường vẽ:

```
\begin{tikzpicture}[Các tham số nếu cần]
.....
Các lệnh vẽ hình ...
.....
\end{tikzpicture}
```

Như vậy. Một file vẽ hình bằng **Tikz** có thể đầy đủ để bạn biên dịch:

```
\documentclass{standalone}%Khai báo lớp văn bản
\usepackage{tikz}%Khai báo gói Tikz để vẽ hình
\usetikzlibrary{calc,angles}%Gọi các thư viện phục vụ vẽ hình
\begin{document}
\begin{tikzpicture}
.....
Các lệnh vẽ hình
.....
\end{tikzpicture}
\end{document}
```

I.2 Các đối tượng hình học

Để vẽ một đối tượng trong **Tikz**, ta thường dùng lệnh:

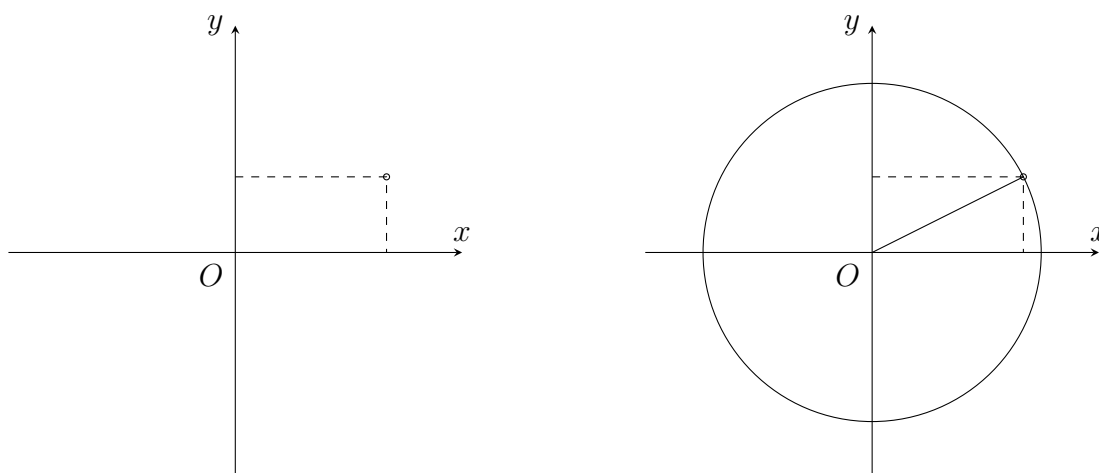
```
\draw [Các tham số ] (Tọa độ điểm vẽ) Đối tượng vẽ;
```

Ta có một vài tham số cơ bản:

- **dashed**: nếu muốn vẽ nét đứt
- **line width=**: nếu muốn thay đổi độ to, nhỏ của nét vẽ.
- **color=**: nếu muốn tô màu nét vẽ.
- **rotate=**: nếu muốn xoay đối tượng vẽ quanh góc tọa độ một góc nào đó.
-

I.2.1 Hệ trục tọa độ

Các đối tượng hình học luôn gắn với hệ tọa độ. Chúng ta có thể sử dụng một trong hai hệ tọa độ để xác định các đối tượng cần vẽ: Hệ tọa độ vuông góc Oxy hoặc tọa độ cực.



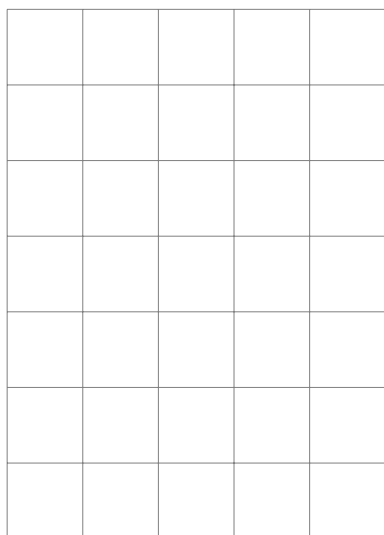
Để dễ dàng cho bạn khi bắt đầu tập vẽ, ta nên tạo lưới tọa độ vùng vẽ để khi xem kết quả ta dễ dàng hình dung ra cách thức và rút kinh nghiệm cho việc sử dụng lệnh bằng cú pháp:

`\draw[step=1,gray,very thin] (góc trái bên dưới) grid (góc phải bên trên);`

Chẳng hạn tạo một lưới tọa độ với vùng vẽ từ $(-2; -1)$ đến $(3; 6)$ ta dùng lệnh:

`\draw[step=1,gray,very thin] (-2,-1) grid (3,6);`

Ta được:



I.2.2 Vẽ Điểm

Ta xác định một điểm trên hình vẽ bằng tọa độ. Trên hệ Oxy ta xác định điểm bằng tọa độ $(x; y)$. Còn trên hệ tọa độ cực ta xác định điểm bằng $(\alpha : r)$.

Ví dụ ta muốn vẽ điểm $A(3; -1)$ ta dùng lệnh sau: `\draw (3,-1) circle (0.1);`

Chú ý rằng tọa độ của điểm trong **Tikz** ngăn cách hoành độ và tung độ bởi dấu "phẩy" (,) chứ không phải dấu "chấm phẩy" (;) như ta thường viết trong văn bản. Kết thúc lệnh vẽ luôn bằng dấu "chấm phẩy" (;).

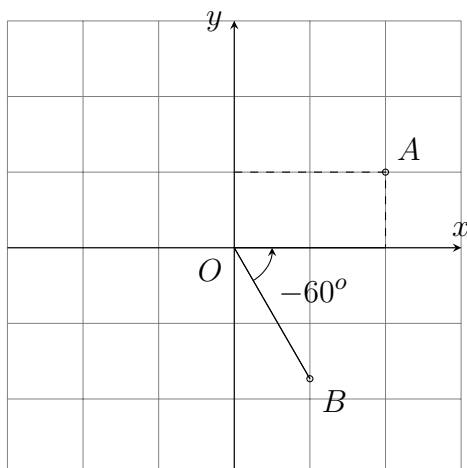
Ta cũng có thể vẽ điểm bằng lệnh `\draw (60:2) circle (0.1);` Khi đó ta sẽ vẽ được điểm B sao cho $OB = 2$ và góc giữa OB và Ox là 60° .

Hãy xem ví dụ dưới đây:

```
\begin{tikzpicture}[>=stealth]
\draw[->](-3,0)--(3,0);% Vẽ trục hoành
\draw[->](0,-3)--(0,3);%Vẽ trục tung
\draw[dashed] (0,1)--(2,1)--(2,0);%Vẽ đường giống tọa độ điểm A
\draw (2,1) circle (0.04);%Vẽ điểm A(2;1)
\draw (3,0) node[above]{$x$} (0,3) node[left]{$y$}
(0,0) node[below left]{$O$};%Hiển thị các chữ
\draw (-60:2) circle (0.04);%Vẽ điểm B(-60:2)
\draw (-60:2) -- (0,0);%Vẽ đoạn thẳng OB
\draw (2,1) node[above right]{$A$} (-60:2) node[below right]{$B$};
%Hiển thị tên điểm A và B
\draw(-60:2) coordinate (A)--(0:0) coordinate (B)--(0:2) coordinate (C)
```

```
pic [draw, ->] {angle=A--B--C};%Vẽ biểu diễn góc lượng giác
\draw (-30:0.5)node[below right]{$-60^{\circ}$};
\end{tikzpicture}
```

Ta sẽ được hình vẽ biểu thị điểm $A(2; 1)$ trên hệ tọa độ vuông góc và điểm $B(-60 : 2)$ trong tọa độ cực:



Chú ý rằng lệnh vẽ điểm là `\draw (Tọa độ điểm) circle (bán kính điểm);`

I.2.3 Đoạn thẳng

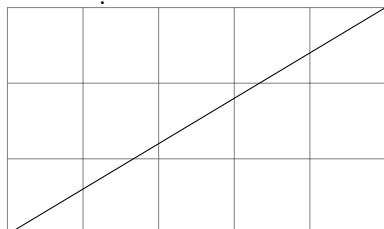
Để vẽ một đoạn thẳng, ta cần xác định tọa độ hai điểm đầu mút của đoạn thẳng và dùng lệnh vẽ:

```
\draw (Tọa độ điểm thứ nhất) -- (Tọa độ điểm thứ hai);
```

Ví dụ muốn vẽ đoạn thẳng AB với $A(2; 1)$ và $B(-3, -1)$ ta dùng lệnh:

```
\draw (2,1)--(-3,-2);
```

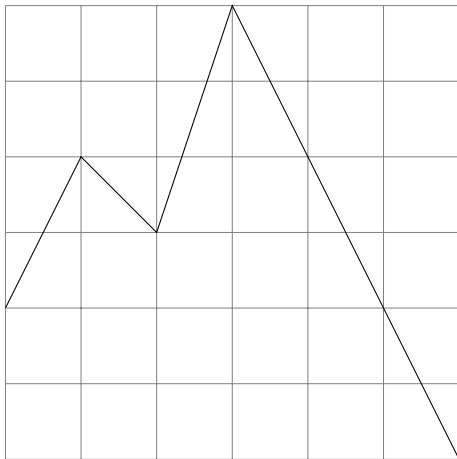
Ta được hình vẽ:



Như vậy, ta có thể vẽ đường gấp khúc qua nhiều điểm liên tiếp. Chẳng hạn:

```
\draw (0,1)--(1,3)--(2,2)--(3,5)--(4,-1);
```

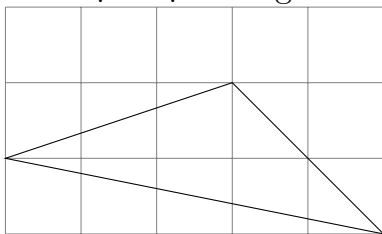
Ta sẽ được đường gấp khúc sau:



Hơn thế nữa, nếu muốn vẽ đa giác khi biết các đỉnh, cũng thật là dễ dàng:

```
\draw(0,1)--(3,2)--(5,0)--cycle;
```

Ta được một tam giác:



I.2.4 Đường tròn

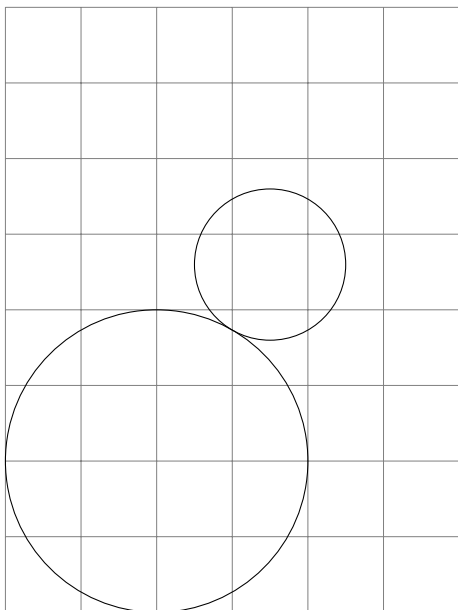
Để xác định một đường tròn, bạn cần tọa độ tâm đường tròn và bán kính đường tròn. Khi đó lệnh vẽ cũng rất đơn giản:

```
\draw (tọa độ tâm đường tròn) circle (bán kính đường tròn);
```

Chẳng hạn ta dùng đoạn lệnh:

```
\begin{tikzpicture}
\draw (0,0) circle (2);
\draw (60:3) circle (1);
\end{tikzpicture}
```

Ta sẽ được hai đường tròn. Đường thứ nhất có tâm là gốc tọa độ, bán kính có độ lớn bằng 2. Đường thứ hai có tâm là điểm có tọa độ (60:3) và bán kính bằng 1:



I.2.5 Đường Ellip

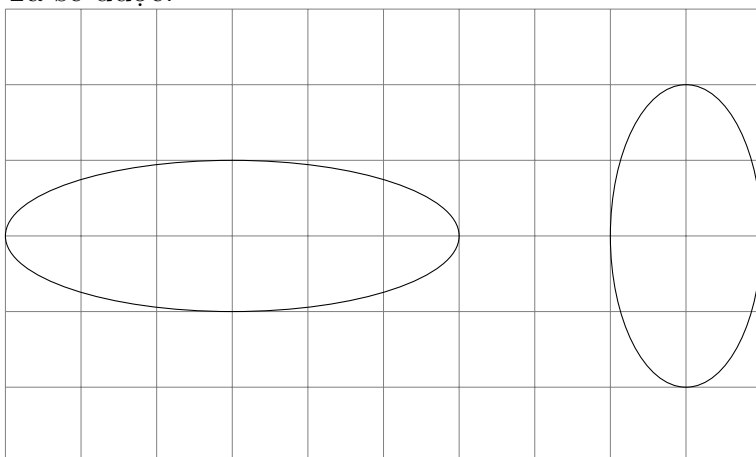
Giống như đường tròn, Ellip cần tọa độ tâm và độ dài hai bán trục. Ta dùng lệnh:

```
\draw (tâm Ellip) ellipse ({bán trục thứ nhất} and {bán trục thứ hai});
```

Chú ý rằng bán trục thứ nhất tính theo trục hoành và bán trục thứ hai tính theo trục tung. Chẳng hạn:

```
\draw (0,0) ellipse ({3} and {1});
\draw (6,0) ellipse ({1} and {2});
```

Ta sẽ được:



I.2.6 Cung tròn, Ellip - Đường cong

Để vẽ một cung tròn, cung Ellipse ta có thể dùng lệnh:

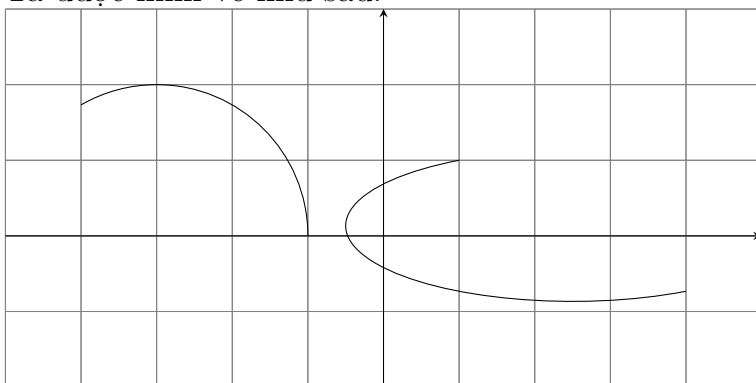
```
\draw (Điểm xuất phát cung)
```

arc [start angle=góc xuất phát cung, end angle=góc kết thúc cung,
x radius=bán kính trên trục hoành, y radius=bán kính trên trục tung];

Như vậy, nếu bạn muốn vẽ cung tròn thì giá trị **x radius** và **y radius** phải bằng nhau, nếu bạn vẽ cung Ellipse thì hai giá trị này khác nhau tùy mục đích vẽ của bạn. Ví dụ sau đây sẽ cho bạn thấy rõ hình ảnh của các cung qua các lệnh vẽ:

```
\draw (-1,0) arc [ start angle=0, end angle=120, x radius=2,y radius=2];
\draw (1,1) arc [ start angle=120, end angle=300, x radius=3,y radius=1];
```

Ta được hình vẽ như sau:



Ở hình vẽ trên, lệnh vẽ cung tròn xuất phát từ điểm $(-1; 0)$, góc vẽ xuất phát từ 0° và kết thúc ở 120° , bán kính hoành và tung là bằng nhau và bằng 2 (vì cung tròn). Lệnh vẽ cung Ellipse xuất phát từ điểm $(1; 1)$, góc vẽ xuất phát từ 120° và kết thúc ở 300° . Bán kính hoành là 3 và bán kính tung là 1.

Để đơn giản hơn, **Tikz** cung cấp cho ta lệnh vẽ cung tròn rút gọn:

```
\draw (Điểm xuất phát) arc (Góc xuất phát : Góc kết thúc : Bán kính cung);
```

Tương tự, ta cũng có thể tạo một lệnh tương tự cho việc vẽ cung Ellipse một cách ngắn gọn như vậy bằng cách định nghĩa mỗi một lệnh `\cunge` như sau:

```
\def\cunge(#1:#2:#3:#4){
  arc [ start angle=#1, end angle=#2, x radius=#3,y radius=#4]
}
```

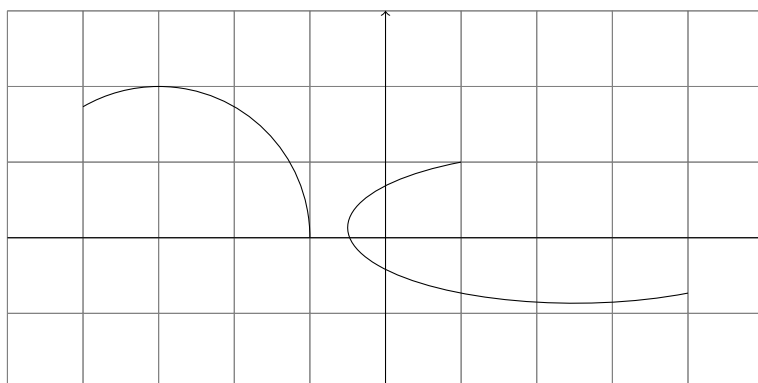
Quý vị hãy đưa đoạn lệnh này đặt sau (hoặc trước) `\begin{document}` rồi khi cần vẽ cung Ellipse quý vị chỉ cần gọi ra bằng lệnh:

```
\draw (Điểm) \cunge( a:b:x:y);
```


Trong đó **a** là góc xuất phát cung, **b** là góc kết thúc cung, **x** là bán kính hoành và **y** là bán kính tung.

Quý vị hãy xem thử đoạn lệnh dưới đây và quan sát kết quả:

```
\begin{tikzpicture}
\draw[step=1,gray,very thin] (-5,-2) grid (5,3);
\draw[step=1,gray,very thin] (-5,-2) grid (5,3);
\draw[->] (-5,0)--(5,0);
\draw[->] (0,-2)--(0,3);
\draw (-1,0) arc (0 :120:2);
\draw (1,1) \cunge (120 : 300 : 3 : 1);
\end{tikzpicture}
```



Ngoài việc vẽ các cung theo đường sẵn có (tròn, ellipse ...), ta còn có thể vẽ các đường cong theo ý mình bằng một số lệnh sau:

- Lệnh **controls**:

Cú pháp:

```
\draw (Điểm 1) .. controls (Điểm tiếp 1) and (Điểm tiếp 2) .. (Điểm 2);
```

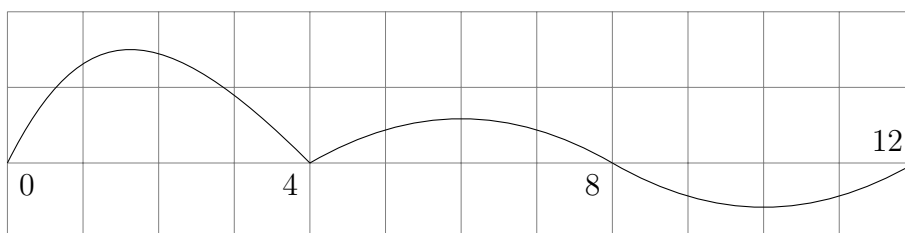
- Lệnh **to**:

Cú pháp:

```
\draw (điểm 1) to [bend left=] {điểm 2}; hoặc [bend right=]
```

Bạn hãy xem ví dụ dưới đây và thực hành các lệnh, với sự thay đổi các giá trị bạn sẽ nhận được các hình khác nhau, và từ đó chính bạn sẽ rút ra những kết luận cho mình khi muốn vẽ đường cong theo ý mình:

```
\draw (0,0) .. controls (1,2) and (2,2) .. (4,0);
\draw (4,0) to [bend left =30] (8,0);
\draw (8,0) to [bend right =30] (12,0);
```



I.2.7 Hiển thị tên đối tượng trên hình vẽ

Để hiển thị các chữ viết, các tên điểm, tên đường, hay có thể là ghi chú thích trên hình vẽ, ta có lệnh:

```
\draw (tọa độ đặt ghi chú) node[tham số]{Nội dung ghi chú};
```

Các tham số ở đây có:

- **above**: ghi chú ở phía trên điểm.
- **below**: ghi chú ở phía dưới điểm.
- **left**: ghi chú ở bên trái điểm.
- **right**: ghi chú bên phải điểm.

Hẳn nhiên bạn cũng có thể kết hợp các tham số này kiểu **above left**, **above right**, **below left** và **below right** mà **Tikz** cũng không hề phàn nàn gì. Bạn cứ yên tâm mà sử dụng.

I.3 Vẽ các hình đơn giản đầu tiên

Để vẽ một hình vẽ hình học phẳng thuần túy, thông thường trong các hình có những điểm liên quan đến nhau. Từ đó xuất phát nhu cầu cần có những điểm phải lặp lại, những điểm do yêu cầu bài toán liên quan. Khi đó ta đi tìm tọa độ từng điểm rất mệt mỏi. Bạn không cần lo lắng vì **Tikz** đã biết trước điều đó.

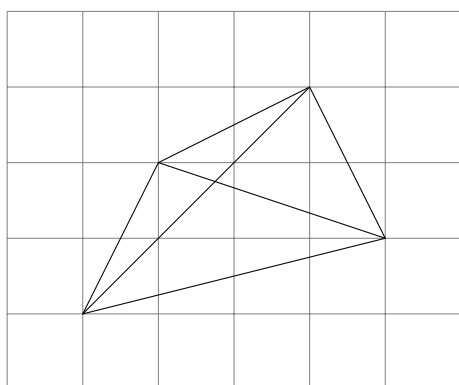
I.3.1 Khai báo các điểm dùng nhiều lần

Để khai báo một điểm ta dùng lệnh:

`\coordinate (tên điểm) at (tọa độ điểm);`

Sau khi ta khai báo điểm này, khi cần vẽ liên quan đến điểm này ta chỉ cần gọi ra:

```
\begin{tikzpicture}
\coordinate (a) at (0,0);
\coordinate (b) at (2,1);
\coordinate (c) at (3,-1);
\coordinate (d) at (-1,-2);
\draw (a)--(b)--(c)--(d)--cycle;
\draw (a)--(c) (b)--(d);
\end{tikzpicture}
```



Thật đơn giản phải không bạn? Tất nhiên bạn muốn hơn thế nữa, chẳng hạn bạn muốn nối trung điểm M của AB và trung điểm N của CD thì làm thế nào?

Cũng chẳng có gì phức tạp khi **Tikz** đã nghĩ hộ cho bạn, không chỉ là trung điểm mà còn là một phép vị tự với tỉ lệ bất kỳ:

`\coordinate (m) at ($(a)!k!(b)$);`

Lệnh này sẽ khai báo cho ta điểm M nằm trên đường thẳng AB với điều kiện $\overrightarrow{AM} = k\overrightarrow{AB}$. Hiển nhiên nếu M là trung điểm AB thì lệnh khai báo sẽ là:

`\coordinate (m) at ($(a)!0.5!(b)$);`

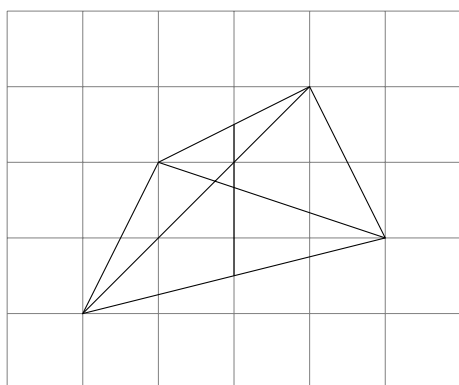
Ta hãy thực hành lệnh này xem sao:

```
\begin{tikzpicture}
\coordinate (a) at (0,0);
\coordinate (b) at (2,1);
\coordinate (c) at (3,-1);
\coordinate (d) at (-1,-2);
```

```

\coordinate (m) at ($(a)!0.5!(b)$);
\coordinate (n) at ($(c)!0.5!(d)$);
\draw (a)--(b)--(c)--(d)--cycle;
\draw (a)--(c) (b)--(d);
\draw (m)--(n);
\end{tikzpicture}

```



Ngoài ra, phép tịnh tiến cũng được **Tikz** quan tâm. Bạn muốn khai báo điểm E là ảnh của điểm A qua phép tịnh tiến theo véc tơ $\vec{v} = (1; -2)$ chẳng hạn:

```

\coordinate (e) at ($(a)+(1,-2)$);

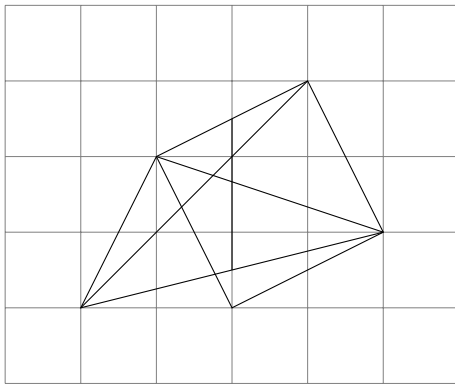
```

Vậy là bạn có thể dùng điểm E trong các lệnh vẽ của mình rồi:

```

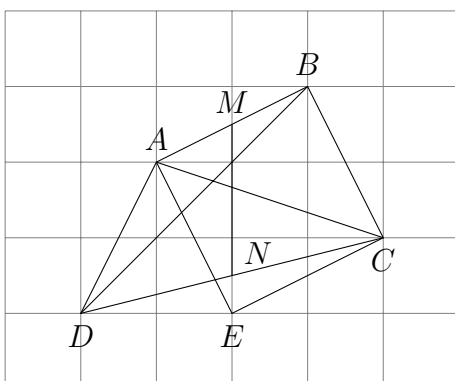
\begin{tikzpicture}
\coordinate (a) at (0,0);
\coordinate (b) at (2,1);
\coordinate (c) at (3,-1);
\coordinate (d) at (-1,-2);
\coordinate (m) at ($(a)!0.5!(b)$);
\coordinate (n) at ($(c)!0.5!(d)$);
\coordinate (e) at ($(a)+(1,-2)$);
\draw (a)--(b)--(c)--(d)--cycle;
\draw (a)--(c) (b)--(d);
\draw (m)--(n);
\draw (a)--(e)--(c);
\end{tikzpicture}

```



Và bây giờ, ta có thể hiển thị tên các điểm cho hình vẽ của mình với đoạn mã lệnh đầy đủ như sau:

```
\begin{tikzpicture}
\coordinate (a) at (0,0);
\coordinate (b) at (2,1);
\coordinate (c) at (3,-1);
\coordinate (d) at (-1,-2);
\coordinate (m) at ($(a)!0.5!(b)$);
\coordinate (n) at ($(c)!0.5!(d)$);
\coordinate (e) at ($(a)+(1,-2)$);
\draw (a)--(b)--(c)--(d)--cycle;
\draw (a)--(c) (b)--(d);
\draw (m)--(n);
\draw (a)--(e)--(c);
\draw (a) node[above]{$A$} (b) node[above]{$B$} (m) node[above]{$M$};
\draw (c) node[below]{$C$} (d) node[below]{$D$} (n) node[above right]{$N$};
\draw (e) node[below]{$E$};
\end{tikzpicture}
```



Bạn hãy cứ an tâm rằng, nếu như ngay từ ban đầu bạn quên khai báo một điểm nào đó, bạn cũng có thể khai báo ngay điểm đó ngay tại lệnh vẽ kiểu như:

```
\draw (0,1) \coordinate (a) --(2,-3) \coordinate (b)--(1,2) \coordinate (c);
```

Như vậy các lệnh sau này bạn cứ việc gọi các điểm A , B , C ra vẽ là **Tikz** cũng chấp nhận cho bạn một cách vui vẻ thôi.

I.3.2 Giao điểm của các đường trong Tikz

Khi vẽ hình, ta thường gặp các hình vẽ cần nối các giao điểm của đường này và đường kia. **Tikz** cũng rất ân cần giúp ta điều này. Tuy nhiên để có thể lấy giao điểm của các đường ta cần gán cho mỗi đường (đoạn, cung) những cái tên riêng cho nó để **Tikz** có thể nhận ra.

Để gán tên ta dùng lệnh `\draw` với tham số:

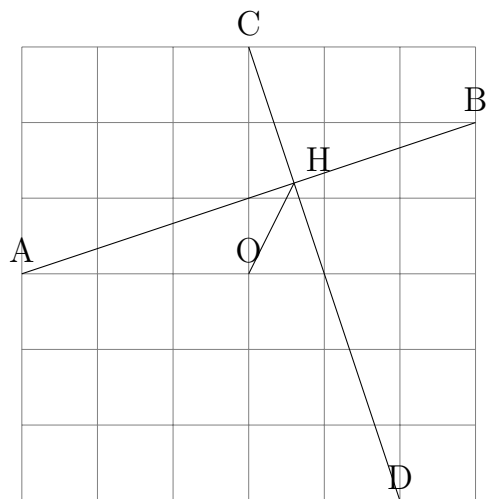
```
\draw[name path= tên đường] ....
```

Sau khi các đường đã được đặt tên, ta có thể tìm giao điểm của các đường ấy bằng lệnh:

```
\path [name intersections={of=đường 1 and đường 2,by=tên giao điểm}];
```

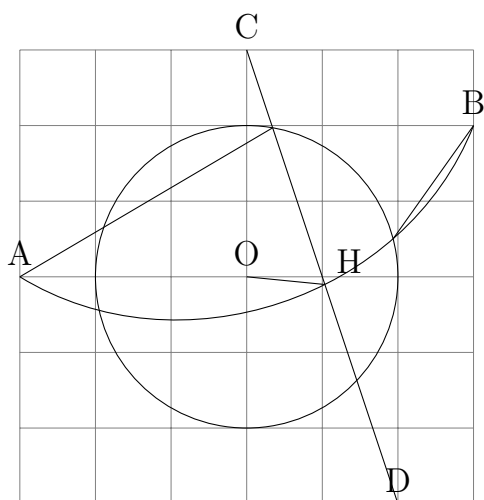
Và bây giờ ta hoàn toàn có thể dùng cái giao điểm vừa tìm để vẽ hình một cách thoải mái. Ta hãy thử với ví dụ sau nhé:

```
\begin{tikzpicture}
\draw [step=1,gray,very thin] (-3,-3) grid (3,3);
\coordinate[label=A] (a) at (-3,0);
\coordinate[label=B] (b) at (3,2);
\coordinate[label=C] (c) at (0,3);
\coordinate[label=D] (d) at (2,-3);
\coordinate[label=O] (o) at (0,0);
\draw[name path=done](a)--(b);
\draw[name path=dtwo] (c)--(d);
\path [name intersections={of=done and dtwo,by=h}];
\draw (o)--(h) ;
\draw (h) node[above right]{H};
\end{tikzpicture}
```



Hơn thế nữa, giao điểm của đường thẳng và đường cong cũng thật sự chính xác:

```
\begin{tikzpicture}
\draw [step=1,gray,very thin] (-3,-3) grid (3,3);
\coordinate[label=A] (a) at (-3,0);
\coordinate[label=B] (b) at (3,2);
\coordinate[label=C] (c) at (0,3);
\coordinate[label=D] (d) at (2,-3);
\coordinate[label=O] (o) at (0,0);
\draw[name path=done](a) to [bend right =50] (b);
\draw[name path=dtwo] (c)--(d);
\draw[name path=dthree] (o) circle (2);
\path [name intersections={of=done and dtwo,by=h}] ;
\path [label=M,name intersections={of=dtwo and dthree,by=m}];
\path [label=N,name intersections={of=dthree and done,by=n}];
\draw (o)--(h) (a)--(m) (b)--(n);
\draw (h) node[above right]{H};
\end{tikzpicture}
```

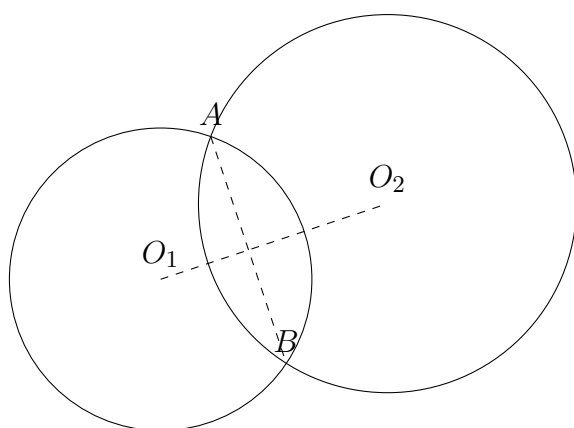


Vấn đề đặt ra ở đây là hai đối tượng nếu giao nhau bởi hai điểm trở lên thì ta sẽ phải làm thế nào? Cũng không hề khó khăn khi bạn dùng cụm lệnh như sau:

```
\path [name intersections={of=tên đường 1 and tên đường 2}];
\coordinate[label=Tên giao điểm 1] (m) at (intersection-1);
\coordinate[label=Tên giao điểm 2] (k) at (intersection-2);
```

Với cụm lệnh trên ta có thể khai báo luôn cả hai giao điểm của hai đối tượng rồi, và tiếp theo ta chỉ việc dùng lệnh vẽ với các điểm đó. Chẳng hạn tôi sẽ tìm giao điểm của hai đường tròn và nối chúng lại bằng một đường nét đứt như sau:

```
\begin{tikzpicture}
\coordinate[label=$O_{1}$] (o1) at (0,0);%Khai báo tâm một
\coordinate[label=$O_{2}$] (o2) at (3,1);%Khai báo tâm hai
\draw [name path=done] (o1) circle (2); %vẽ đường tròn một và đặt tên done
\draw[name path=dtwo] (o2) circle (2.5);%vẽ đường tròn hai và đặt tên dtwo
\path [name intersections={of=done and dtwo}];%Lấy giao điểm
\coordinate[label=$A$] (a) at (intersection-1);%Khai báo giao điểm thứ nhất
\coordinate[label=$B$] (b) at (intersection-2);%Khai báo giao điểm thứ hai
\draw[dashed] (a)--(b) (o1)--(o2);%Vẽ đoạn thẳng AB và O1O2 bằng nét đứt
\end{tikzpicture}
```

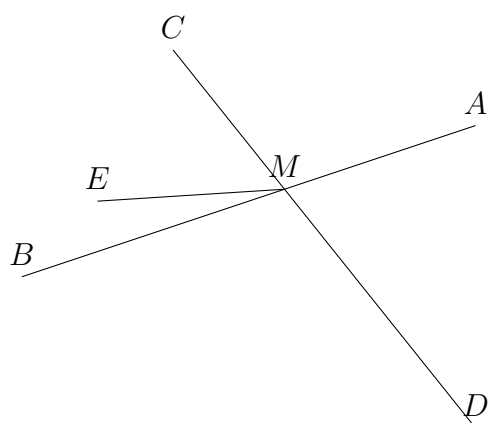



Hoàn toàn tương tự, nếu hai đường cắt nhau bởi ba điểm trở lên. Hoặc ngay chính hai đường chỉ có duy nhất một giao điểm, khuyến cáo bạn nên dùng cách này để khai báo điểm được chính xác và có thể gán nhãn luôn cho điểm khi vẽ hình (đỡ tốn công dùng lệnh `\node` hoặc `\draw ... node ...`).

Chẳng hạn với code vẽ:

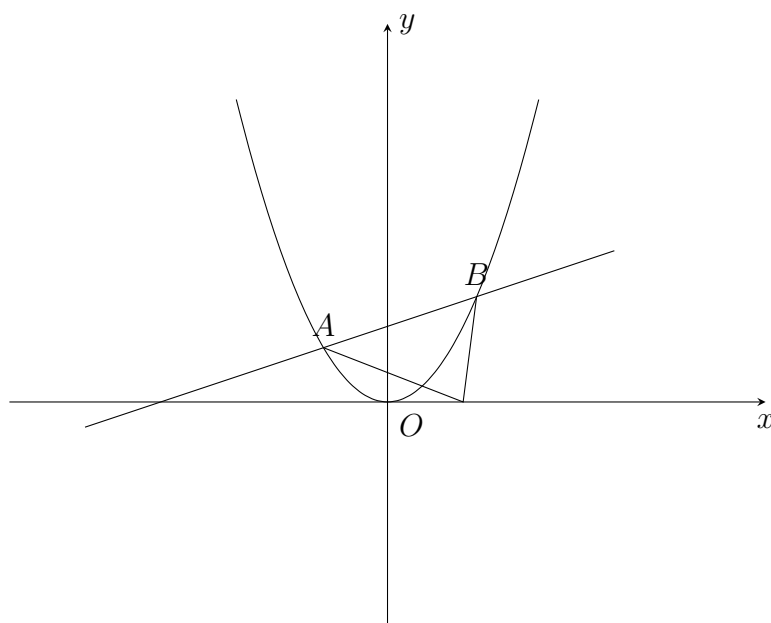
```
\begin{tikzpicture}
\coordinate[label=$A$] (a) at (3,1);
\coordinate[label=$B$] (b) at (-3,-1);
\coordinate[label=$C$] (c) at (-1,2);
\coordinate[label=$D$] (d) at (3,-3);
\coordinate[label=$E$] (e) at (-2,0);
\draw[name path=done] (a)--(b);
\draw[name path=dtwo] (c)--(d);
\path[name intersections={of =done and dtwo}]; %Lấy giao điểm
\coordinate[label=$M$] (m) at (intersection-1);
%Khai báo và gán nhãn giao điểm
\draw (e)--(m);
\end{tikzpicture}
```

Ta sẽ có hình vẽ như sau:



Hơn thế nữa, bạn cũng hoàn toàn có thể tìm giao điểm của các đồ thị hàm số với cách vẽ tương tự. Chẳng hạn bạn cần vẽ đồ thị hàm số $y = x^2$ và đồ thị hàm số $y = \frac{1}{3}x + 1$. Thêm nữa, bạn cần lấy các giao điểm của các đồ thị này để nối với một điểm nào đó, bạn hãy xem ví dụ sau:

```
\begin{tikzpicture}[>=stealth]
\draw[->] (-5,0)--(0,0) node[below right]{$0$}--(5,0) node[below]{$x$};
\draw[->] (0,-3)--(0,5) node[right]{$y$};
\draw[name path=done,smooth,domain=-2:2] plot (\x,{(\x)^(2)});
\draw[name path=dtwo,smooth,domain=-4:3] plot (\x,{(1/3)*(\x)+1});
\path[name intersections={of =done and dtwo}];
\coordinate[label=$A$] (a) at (intersection-1);
\coordinate[label=$B$] (b) at (intersection-2);
\draw (a)--(1,0) -- (b);
\end{tikzpicture}
```



Như vậy, bạn hoàn toàn có thể vẽ hình với các lệnh vẽ đơn giản và gần như đã đầy đủ cho một hình vẽ thông thường.

I.3.3 Vẽ ký hiệu vuông góc

Trong **Tikz**, tại hạ cũng đã tìm hiểu về các thư viện, chẳng hạn như thư viện **angles** để vẽ các vòng khuyên ký hiệu cho các góc, tuy nhiên ký hiệu cho góc vuông thì tại hạ chưa tìm thấy. Cũng không sao cả, tại hạ đã mày mò và hoàn thiện được một lệnh để bạn có thể vẽ ký hiệu này cho các hình của mình.

Đầu tiên, tại hạ khuyên quý vị hãy copy đoạn lệnh sau đặt ngay vào đầu file biên dịch vẽ hình (ngay sau `\begin{document}` càng tốt)

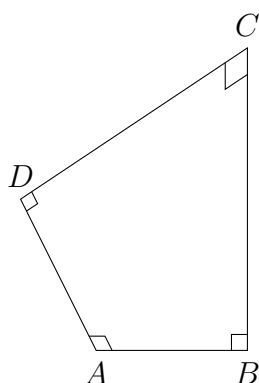
```
\def\khvuong[size=#1](#2,#3,#4){%
\draw ($(#3)!#1!(#2)$) --
($($(#3)!#1!(#2)$)+($(#3)!#1!(#4)$)-(#3)$)--
($(#3)!#1!(#4)$);
}
```

Sau đó, khi muốn vẽ ký hiệu góc vuông ở điểm nào ta dùng lệnh:

```
\khvuong[size=độ lớn ký hiệu](Điểm 1,Điểm 2,Điểm 3);
```

Ví dụ dưới đây sẽ cho quý vị thấy sự tùy biến này tốt như thế nào:

```
\begin{tikzpicture}
\coordinate[label=below:$A$] (a) at (0,0);
\coordinate[label=below:$B$] (b) at (2,0);
\coordinate[label=$C$] (c) at (2,4);
\coordinate[label=$D$] (d) at (-1,2);
\draw (a)--(b)--(c)--(d)--cycle;
\khvuong[size=6pt](a,b,c);%Vẽ ký hiệu vuông góc tại điểm B, cỡ là 6pt
\khvuong[size=10pt](b,c,d);%Vẽ ký hiệu vuông góc tại điểm C, cỡ 10pt (lớn hơn)
\khvuong[size=5pt](c,d,a);%Vẽ ký hiệu vuông góc tại điểm D
\khvuong[size=6pt](d,a,b);%Vẽ ký hiệu vuông góc tại điểm A
\end{tikzpicture}
```



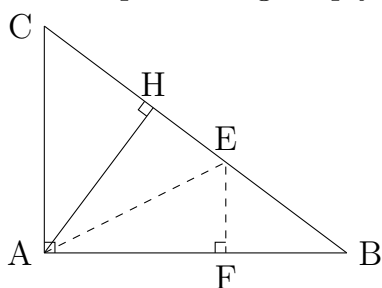
Nếu quý vị có nhã ý, hoàn toàn quý vị có thể dùng lệnh này để vẽ ký hiệu vuông góc khi vẽ hình không gian. Thật sự tuyệt vời phải không nào!

I.3.4 Phép Quay

Với **Tikz**, việc thực hiện phép quay để lấy ảnh của một đối tượng trong hình vẽ khi khai báo cũng rất cần thiết. Chẳng hạn ta cần lấy điểm C là ảnh của phép quay tâm A của điểm B với góc quay 60° thì ta có thể dùng lệnh:

```
\coordinate (c) at ($(a)!1!60:(b)$);
```

Vấn đề quan trọng là quý vị cần phải biết được góc quay cụ thể của phép quay đó.



I.3.5 Vẽ đường vuông góc của một đường

Trong các bài toán hình, việc lấy chân đường vuông góc từ một điểm đến một đoạn thẳng là việc thường gặp. Quý vị sẽ vẽ như thế nào?

Thật đơn giản với lệnh khai báo sau: `\coordinate (h) at ($(b)!(a)!(c)$);`. Lúc này bạn đã có thể tìm thấy H là chân đường cao kẻ từ A tới BC rồi. Việc của bạn lúc đó chỉ còn là dùng lệnh `\draw` để vẽ.

Bạn hãy vẽ một tam giác ABC và đường cao AH như sau:

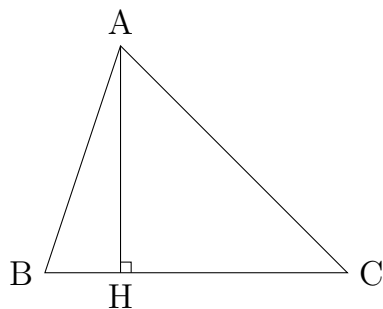
```
\begin{tikzpicture}
```

```
\coordinate[label=above:A] (a) at (1,3);
```

```

\coordinate[label=left:B] (b) (0,0);
\coordinate[label=right:C] (c) at (4,0);
\coordinate[label=below:H] (h) at ($(b)!(a)!(c)$);
\draw (a)--(b)--(c)--cycle;
\draw (a)--(h);
\khvuong[size=4pt](a,h,c)
\end{tikzpicture}

```



I.3.6 Tính toán trong Tikz

Một số hàm tính toán trong **Tikz**

- **sin** (α): Trả về giá trị $\sin \alpha$.
- **cos** (α): Trả về giá trị $\cos \alpha$.
- **tan** (α): Trả về giá trị $\tan \alpha$.
- **asin** (b): Trả về giá trị $\arcsin b$.
- **pow** (α, a): Trả về giá trị a^α .
- **ln** (a): Trả về giá trị $\ln a$.
- **exp** (α): Trả về giá trị e^α .
- **log_a** (b) : Trả về giá trị $\log_a b$.
- **sqrt** (a): Trả về giá trị \sqrt{a} .
- **abs**(a): Trả về giá trị tuyệt đối của a .
- **int** (a): Trả về phần nguyên của số a .

I.3.7 Vẽ hình bình hành

Ví dụ vẽ hình bình hành rất có ích cho bạn. Khi bạn muốn vẽ một hình nào đó, quá trình tư duy của bạn phải thông qua các bước: phân tích hình; đặt tọa độ những điểm cơ bản dựa vào mối liên hệ với các điểm còn lại; cuối cùng là đặt lệnh vẽ.

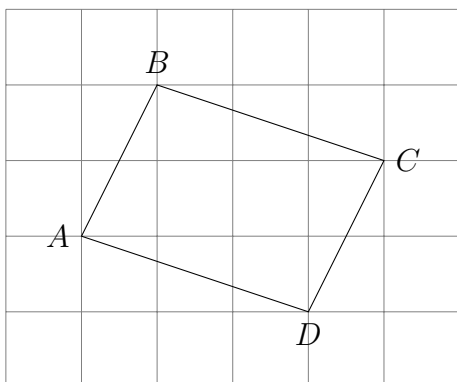
- Phân tích: Với hình bình hành, bạn sẽ phải đặt trước tọa độ ít nhất hai điểm, yếu tố thứ ba là tọa độ điểm thứ ba hoặc véc tơ cạnh hoặc độ dài một cạnh ... Như vậy, giả sử tôi sẽ đặt tọa độ hai điểm bằng lệnh khai báo điểm: `\coordinate (a) at ();`
- Vậy hai điểm còn lại sẽ liên quan thế nào? Hãy sử dụng điều kiện của hình bình hành: $ABCD$ là hình bình hành $\Leftrightarrow \overrightarrow{AB} = \overrightarrow{DC}$. Như vậy ta sẽ tìm ra cách khai báo hai điểm còn lại dựa theo hai điểm đã cho.
- Ta vẽ thôi: Giả sử như tôi đặt hai điểm trước là $A(1;2)$ và $C(5,3)$. Véc tơ $\overrightarrow{AB} = (1;2)$ thì ta có các lệnh khai báo như sau:

```
\coordinate (a) at (1,2);
\coordinate (c) at (5,3);
\coordinate (b) at ($(a)+(1,2)$);
\coordinate (d) at ($(c)-(1,2)$);
```

Vậy thì ngại gì ta không sắp luôn cả mã lệnh để hoàn thành hình vẽ chứ?

```
\begin{tikzpicture}
\coordinate (a) at (1,2);
\coordinate (c) at (5,3);
\coordinate (b) at ($(a)+(1,2)$);
\coordinate (d) at ($(c)-(1,2)$);
\draw (a)--(b)--(c)--(d)--cycle;
\draw (a) node[left]{$A$} (b) node[above]{$B$}
(c) node[right]{$C$} (d) node[below]{$D$};
\end{tikzpicture}
```

Và đây là thành quả của ta:



Như vậy, khi vẽ bất kỳ một hình bình hành ở vị trí khác, tôi và bạn chỉ cần thay đổi tọa độ điểm A , C và tọa độ véc tơ \overrightarrow{AB} là ta có một hình bình hành tùy ý rồi. Đây mới là mục tiêu chính khi ta tập vẽ hình bình hành này.

I.3.8 Vẽ hình tròn và tam giác nội tiếp

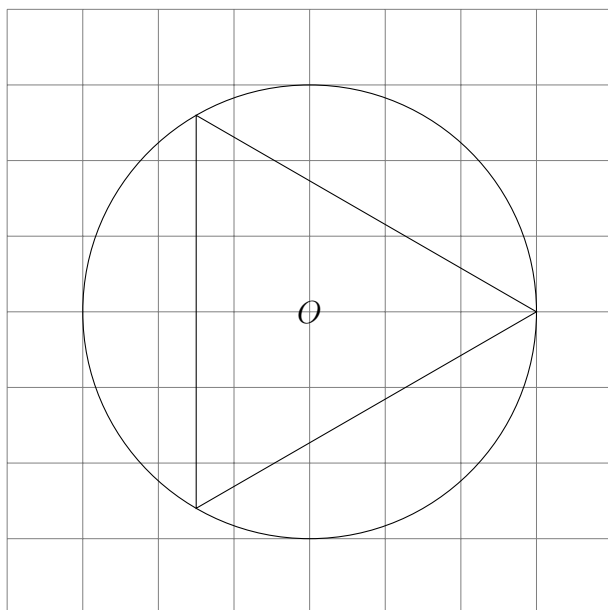
Nếu bạn phải vẽ một tam giác và hình tròn ngoại tiếp tam giác? Hoặc ngược lại bạn phải vẽ hình tròn và một tam giác (hay một đa giác) nội tiếp hình tròn đó? Thật vất vả nếu phải đi tìm tọa độ của các điểm nằm trên hình tròn vì các điểm nằm trên hình tròn cho dù bán kính có thật đẹp thì tọa độ của chúng cũng là những số thập phân dài ngoằng, nhiều khi là những số vô tỉ. Để đạt được sự chính xác, tôi khuyên bạn hãy dùng tọa độ cực.

Chẳng hạn, bạn có hình tròn có tâm là gốc tọa độ, bán kính $r = 3$. Bạn muốn lấy một điểm nào đó trên hình tròn thì tọa độ của điểm đó rất lẻ. Nhưng nếu bạn sử dụng tọa độ cực thì hoàn toàn khả thi vì chỉ liên quan đến góc và chính bán kính r đã có.

Ví dụ sau đây cho ta một giải pháp:

```
\begin{tikzpicture}
\draw (0:0) circle (3);
\draw (0:3)--(120:3)--(240:3)--cycle;
\end{tikzpicture}
```

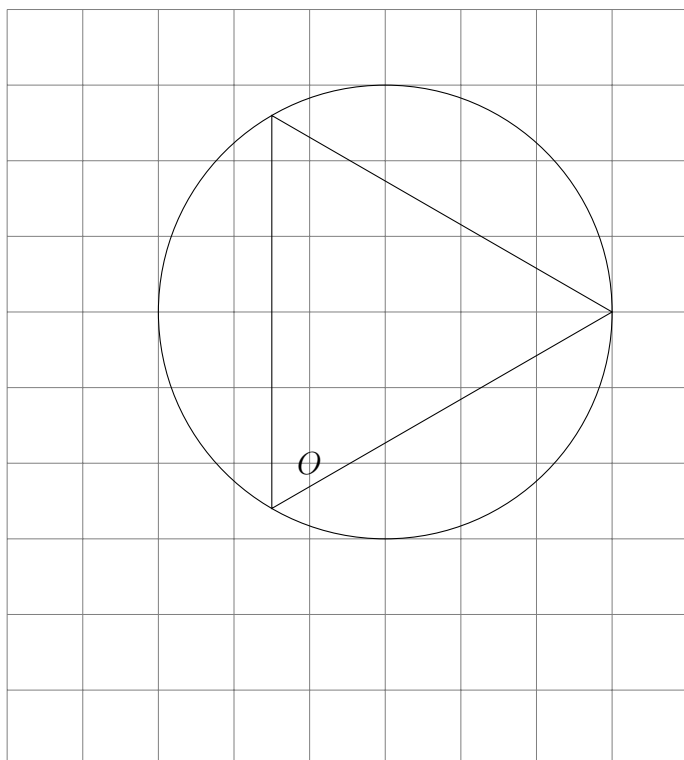
Chỉ cần đơn giản thế thôi ta đã có một tam giác đều nội tiếp đường tròn:



Thật là tuyệt vời với một vài lệnh đơn giản. Nhưng trong thực tế, có voi đòi hai bà trưng, thế nên bạn không muốn cái đường tròn kia có tâm là $(0;0)$ thì ta làm thế nào? Dịch chuyển đường tròn thì không hề khó khăn, nhưng dịch chuyển cái tam giác bên trong với bạn lại có vẻ không dễ dàng. Không sao, bạn sẽ có giải pháp dịch chuyển cả đường tròn và tam giác với tham số **shift=** trong lệnh `\draw` như sau:

```
\draw[shift={(1,2)}] (0:0) circle (3);
```

```
\draw[shift={(1,2)}] (0:3)--(120:3)--(240:3)--cycle;
```



Hoàn toàn tương tự như vậy khi vẽ hình. Bạn hãy dùng tư duy của chính mình để hoạch định kế hoạch và đặt lệnh vẽ sao cho tối ưu nhất có thể.

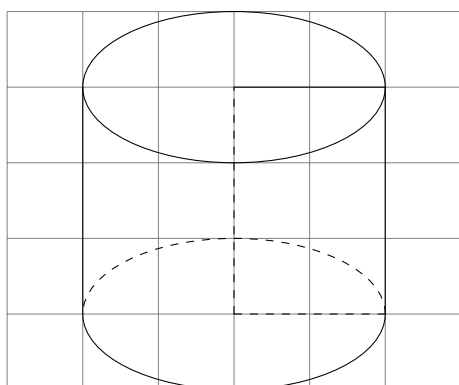
I.3.9 Vẽ hình trụ

Đối với hình học không gian, ta cần vẽ những hình có nét đứt và nét liền, có phần khuất và phần nhìn thấy nên có vẽ phức tạp hơn. Ta hãy bắt đầu từ hình trụ.

Biểu diễn hình trụ gồm hai đường sinh hai bên, hai ellipse làm hai đáy trong đó một đáy có một nửa nét liền và một nửa nét đứt. Điều này buộc tôi phải nghĩ đến việc vẽ cung ellipse (Cung tròn, Ellip - Đường cong).

Khi vẽ hình, để cho đơn giản ta thường đưa những điểm đặc biệt của hình về gốc tọa độ, khi đó những thao tác còn lại sẽ đơn giản hơn và các tính toán cũng sẽ nhẹ nhàng hơn. Ở đây tôi có thể đưa tâm một đáy về gốc tọa độ. Chẳng hạn là đáy dưới, vậy thì ta sẽ vẽ hai nửa ellipse, một nửa nét đứt và một nửa nét liền. Đáy trên sẽ vẽ nguyên cả ellipse vì đó là mặt nhìn thấy. Sau một thời gian phân tích lằng nhằng như thế ta có toàn bộ mã lệnh:

```
\begin{tikzpicture}
\draw[dashed] (2,0) arc [start angle=0,end angle=180, x radius=2,y radius=1];
%Vẽ nửa ellipse nét đứt
\draw (-2,0) arc [ start angle=180, end angle=360, x radius=2,y radius=1];
%vẽ nửa ellipse nét liền
\draw (-2,0)--(-2,3) (2,3)--(2,0) (0,3)--(2,3);
%Vẽ hai đường sinh và bán kính đáy trên
\draw[dashed] (0,0)--(0,3) (0,0)--(2,0);
%Vẽ trục và bán kính đáy dưới
\draw (0,3) ellipse ({2} and {1});
%Vẽ ellipse đáy trên
\end{tikzpicture}
```



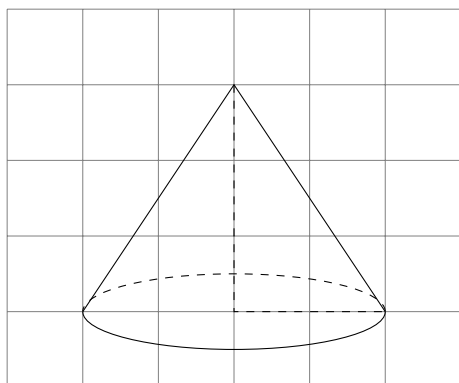
Một giải pháp khác cho bạn và tôi khi vẽ các nửa ellipse bằng lệnh

`\draw (A) to[bend left=] (B);` Khi đó ta cũng thu được một kết quả rất khả quan. Bạn có thể tự thực hành để nắm vững hơn kỹ thuật này.

I.3.10 Vẽ hình nón

Cũng giống như vẽ hình trụ. Hình nón cũng có đáy được vẽ bằng hai nửa ellipse như sau:

```
\begin{tikzpicture}
\draw[dashed] (2,0) arc [start angle=0, end angle=180, x radius=2,y radius=0.5]
\draw (-2,0) arc [start angle=180, end angle=360, x radius=2,y radius=0.5];
\draw (-2,0)--(0,3)--(2,0);
\draw[dashed] (0,3) --(0,0)--(2,0);
\end{tikzpicture}
```

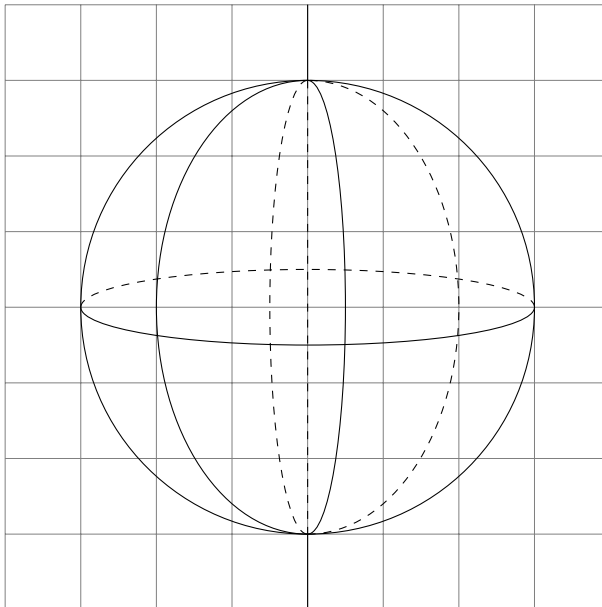


I.3.11 Vẽ mặt cầu

Đối với mặt cầu. Bạn biết chắc phải vẽ một hình tròn, ngoài ra vài cái ellipse làm kinh tuyến và vĩ tuyến với một cái trục để nhìn cho thật hơn. Đơn giản chỉ có vậy:

```
\begin{tikzpicture}
\draw (0,0) circle (3);
\draw[dashed] (3,0) arc [start angle=0,end angle=180,x radius=3,y radius=0.5];
\draw (-3,0) arc [start angle=180,end angle=360,x radius=3,y radius=0.5];
\draw[dashed] (0,3) arc [start angle=90,end angle=270,x radius=0.5,y radius=3];
\draw (0,-3) arc [start angle=-90,end angle=90,x radius=0.5,y radius=3];
\draw (0,3) arc [start angle=90,end angle=270,x radius=2,y radius=3];
\draw[dashed] (0,-3) arc [start angle=-90,end angle=90,x radius=2,y radius=3];
\end{tikzpicture}
```

```
\draw (0,4)--(0,3) (0,-3)--(0,-4);
\draw[dashed] (0,3)--(0,-3);
\end{tikzpicture}
```



Hoàn toàn đơn giản phải không bạn? Bạn hãy tập thay đổi các thông số để tìm hiểu kỹ hơn.

I.3.12 Hình lăng trụ

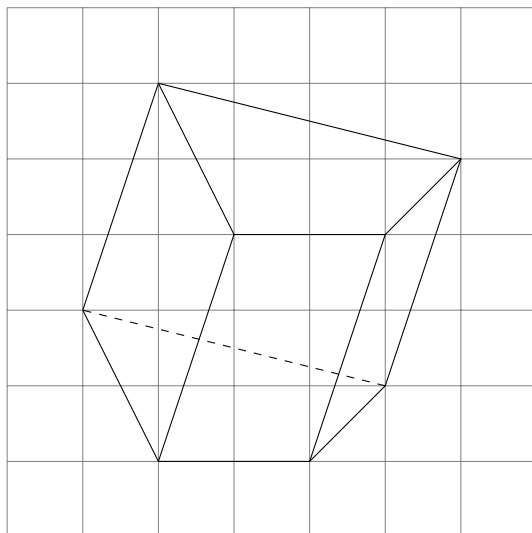
Vẽ hình lăng trụ quả thật không hề khó, bởi ta đã biết tính chất của lăng trụ là các cạnh bên song song và bằng nhau. Vậy thì bạn hãy khai báo các đỉnh của một đáy và từ đó suy ra các đỉnh của đáy kia. Việc còn lại của bạn chỉ còn là gõ lệnh vẽ các đoạn thẳng (nét đứt và nét liền).

```
\begin{tikzpicture}
\coordinate (a) at (0,0);
\coordinate (b) at (2,0);
\coordinate (c) at (3,1);
\coordinate (d) at (-1,2);
\coordinate (a') at ($(a)+(1,3)$);
\coordinate (b') at ($(b)+(1,3)$);
\coordinate (c') at ($(c)+(1,3)$);
\coordinate (d') at ($(d)+(1,3)$);
\draw (a')--(b')--(c')--(d')--cycle;
```

```

\draw (a)--(a') (b)--(b') (c)--(c') (d)--(d');
\draw (d)--(a)--(b)--(c);
\draw[dashed] (c)--(d);
\end{tikzpicture}

```



Trong mấy hình vẽ trên đây, việc của bạn là thêm tên các đỉnh trong hình. Và với những điều cơ bản trên, bạn có thể vẽ được khá khá hình vẽ thông thường rồi đấy.

I.4 Làm đẹp hình vẽ

Đối việc vẽ hình, vẽ chính xác là một tiêu chí. Hình vẽ cần sinh động, màu mè lại là một tiêu chí thứ hai không thể thiếu. Khi nhìn vào một hình vẽ đầy những đường với một màu đen huyền bí quả là chán ngắt. Do đó bạn cần tô màu cho hình vẽ bắt mắt hơn.

I.4.1 Tô màu cho đường nét

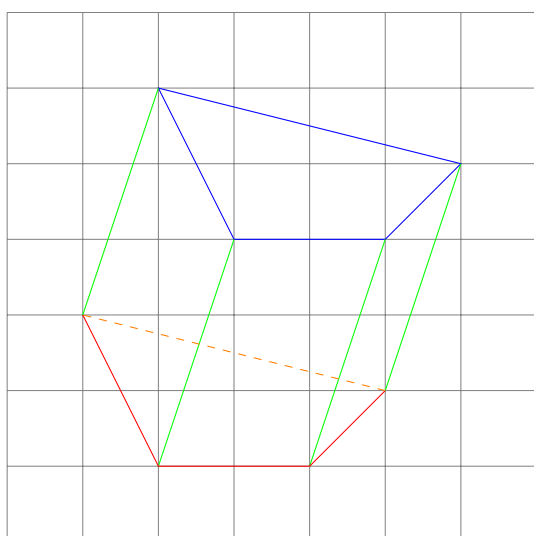
Trong lệnh vẽ `\draw`. Tham số màu là một tham số thường được dùng. Với tên gọi các màu hoặc dùng mã màu trong bảng màu `[rgb]` là thực sự cần thiết. Chẳng hạn muốn vẽ một đoạn thẳng màu xanh ta có thể dùng lệnh `\draw[blue] (a)--(b);`. Hơn thế nữa ta có thể thay đổi độ đậm nhạt cho màu của đường nét hoặc trộn phối màu khi sử dụng tham số `color=blue!50!yellow!50`. Các con số này thay đổi bạn sẽ thấy tỉ lệ màu thay đổi và bểu hình vẽ của bạn đẹp nghĩa là bạn đã thành công với việc phối trộn màu.

Hãy thực hành việc tô màu cho đường nét với hình lăng trụ ta vừa vẽ ở trên nhé:

```

\begin{tikzpicture}
\coordinate (a) at (0,0);
\coordinate (b) at (2,0);
\coordinate (c) at (3,1);
\coordinate (d) at (-1,2);
\coordinate (a') at ($(a)+(1,3)$);
\coordinate (b') at ($(b)+(1,3)$);
\coordinate (c') at ($(c)+(1,3)$);
\coordinate (d') at ($(d)+(1,3)$);
\draw[blue] (a')--(b')--(c')--(d')--cycle;
\draw[green] (a)--(a') (b)--(b') (c)--(c') (d)--(d');
\draw[color=red] (d)--(a)--(b)--(c);
\draw[dashed, color=orange] (c)--(d);
\end{tikzpicture}

```



Hãy thực hành nhiều để bạn có thể chủ động hơn với các màu vẽ.

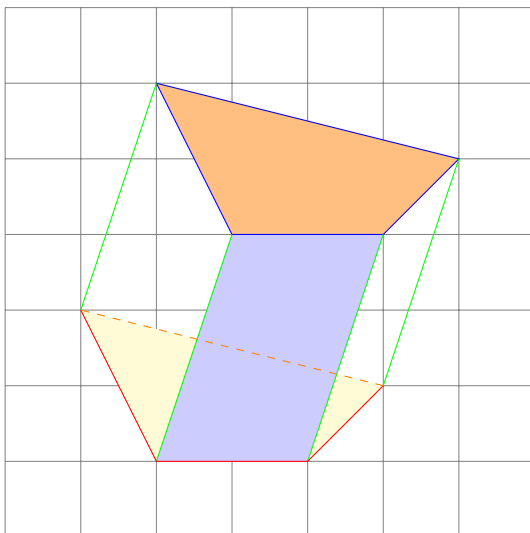
I.4.2 Tô màu cho một miền kín

Cũng với hình vẽ trên, nếu tô màu cho một miền kín, chẳng hạn như đáy trên của hình lăng trụ, ta có một vài cách sau:

- **Fill:** Hãy dùng lệnh `\fill[tên màu]` Vùng tô màu; bạn sẽ thấy kết quả thật đẹp mắt.
- **Shade:** Tương tự như **Fill**, nhưng thật sự tô màu bằng lệnh này quả thật hiệu quả đẹp hơn rất nhiều.

Ta tô màu hình lăng trụ như thế này:

```
\begin{tikzpicture}
\coordinate (a) at (0,0);
\coordinate (b) at (2,0);
\coordinate (c) at (3,1);
\coordinate (d) at (-1,2);
\coordinate (a') at ($(a)+(1,3)$);
\coordinate (b') at ($(b)+(1,3)$);
\coordinate (c') at ($(c)+(1,3)$);
\coordinate (d') at ($(d)+(1,3)$);
\fill[yellow!20] (a)--(b)--(c)--(d)--cycle;
\fill[orange!50] (a')--(b')--(c')--(d')--cycle;
\fill[blue!20] (a)--(a')--(b')--(b)--cycle;
\draw[blue] (a')--(b')--(c')--(d')--cycle;
\draw[green] (a)--(a') (b)--(b') (c)--(c') (d)--(d');
\draw[color=red] (d)--(a)--(b)--(c);
\draw[dashed, color=orange] (c)--(d);
\end{tikzpicture}
```



Với các lệnh tô màu, ta cũng cần quan tâm đến các tham số màu. Đặc biệt là các chế độ tô:

```
\begin{tikzpicture}[rounded corners,ultra thick]
\shade[top color=yellow,bottom color=black] (0,0) rectangle +(2,1);
```

```
\shade[left color=yellow,right color=black] (3,0) rectangle +(2,1);  
\shade[inner color=yellow,outer color=black,draw=yellow] (6,0) rectangle +(2,1)  
\shade[ball color=green] (9,.5) circle (.5cm);  
\end{tikzpicture}
```



Bạn hãy để ý đến các chế độ tô như **top color**, **bottom color**, **left color** và **right color**, **ball color** hoặc **inner color** hay **outer color**. Bạn hãy thực hành với những hình vẽ của bạn. Sự thú vị sẽ đến với bạn.

PHẦN II. BẢNG BIẾN THIÊN-ĐỒ THỊ HÀM SỐ

II.1 Bảng biến thiên

Cấu trúc của bảng biến thiên gồm ba dòng. Dòng giá trị x , dòng giá trị đạo hàm $f'(x)$ và dòng $f(x)$. Như vậy ta cần đến gói **Tkz-tab** để làm việc này.

Khai báo gói bằng lệnh sau: `\usepackage{tkz-tab}`

Sau khi khai báo gói rồi, ta bắt đầu vẽ bảng biến thiên nhé:

```
\begin{tikzpicture}
\tkzTabInit[nocadre,lgt=1.5,espcl=2,deltacl=.5]
{\$x\$/0.7, \$f'(x)\$/0.7, \$f(x)\$/2}
{\$-\infty\$, \$-1\$, \$1\$, \$2\$, \$+\infty\$}
\end{tikzpicture}
```

Với dòng lệnh như trên, ta có hình vẽ sau:

x	$-\infty$	-1	1	2	$+\infty$
$f'(x)$					
$f(x)$					

Như vậy ta đã có khung của một bảng biến thiên. Trong ba dòng lệnh trên, dòng đầu tiên `\tkzTabInit[...]` thì hầu như ta không cần động đến, hãy để mặc định như vậy cho ta hình vẽ cân đối nhất. Nếu bạn có ý định thay đổi thì cứ thử thay đổi từng thông số một để kiểm tra sự thay đổi của hình vẽ. Như vậy bạn sẽ hiểu thêm được ý nghĩa của các thông số.

Dòng thứ hai là

```
\$x\$ / 0.7, \$f'(x)\$ / 0.7, \$f(x)\$ / 2}
```

Thật dễ hiểu phải không bạn? chỉ là ghi các dòng tương ứng x , $f'(x)$ và $f(x)$ mà thôi. Tuy nhiên bạn có thể thay đổi các thông số 0.7, 2 để thấy được sự thay đổi độ cao của từng dòng.

Dòng thứ ba


```
{\$-\infty$, \$-1$, \$1$, \$2$, \$+\infty\$}
```

Đây chính là dòng quy định các giá trị của biến x trên bảng biến thiên, các điểm là nghiệm của đạo hàm và điểm hàm số không xác định. Có bao nhiêu điểm bạn hãy cứ điền cho đủ. Bảng biến thiên sẽ kéo dài đủ độ rộng cho bạn trong việc vẽ hình.

Bắt đầu đến dòng của đạo hàm. Dòng của đạo hàm thường chỉ có các điểm $f'(x) = 0$, các điểm $f'(x)$ không xác định và các dấu $+$, $-$. Dòng này có mã lệnh như sau:

```
\tkzTabLine{, -, z, +, z, -, z, +, }
```

Như vậy, mỗi khoảng có các dấu đã rõ. Tại các điểm $f'(x) = 0$ ta thấy được ký hiệu bằng chữ z . Bạn hãy xem:

```
\begin{tikzpicture}
\tkzTabInit[nocadre,lgt=1.5,espcl=2,deltacl=.5]
{\$x\$/0.7, \$f'(x)\$/0.7, \$f(x)\$/2}
{\$-\infty$, \$-1$, \$1$, \$2$, \$+\infty\$}
\tkzTabLine{, -, z, +, z, -, z, +, }
\end{tikzpicture}
```

x	$-\infty$	-1	1	2	$+\infty$		
$f'(x)$	$-$	0	$+$	0	$-$	0	$+$
$f(x)$							

Đến đây, nếu bạn thắc mắc rằng tại điểm $x = 1$ hàm số không xác định thì bạn phải làm gì? Tất nhiên bạn có thể thay chữ z bằng chữ d

```
\begin{tikzpicture}
\tkzTabInit[nocadre,lgt=1.5,espcl=2,deltacl=.5]
{\$x\$/0.7, \$f'(x)\$/0.7, \$f(x)\$/2}
{\$-\infty$, \$-1$, \$1$, \$2$, \$+\infty\$}
\tkzTabLine{, -, z, +, d, -, z, +, }
\end{tikzpicture}
```

x	$-\infty$	-1	1	2	$+\infty$	
$f'(x)$	$-$	0	$+$	$-$	0	$+$
$f(x)$						

Cuối cùng, bạn cần hoàn thành chiều biến thiên của dòng $f(x)$. Bạn hãy lưu ý rằng, tính từ trái sang phải, mũi tên đi lên hay đi xuống phụ thuộc vào bạn. Nếu bạn dùng $+/ \$+\infty\$$, $- / \$-4\$$ thì nghĩa là mũi tên sẽ từ trái qua phải đi xuống từ $+\infty$ đến -4 . Nếu bạn dùng $- / \$-4\$$, $+/ \$-3\$$ thì nghĩa là mũi tên từ trái qua phải đi lên từ -4 đến 3 . Bạn hãy chú ý dấu $-$ hoặc $+$ trước kí hiệu $/$ nhé. Hãy thử dòng lệnh sau với bảng biến thiên vừa tạo:

```
\tkzTabVar{+/$+\infty\$ , -/ \$-4\$ , +/$-3\$ , -/$-4\$ , +/$+\infty\$}
```

Bạn sẽ có mã hoàn chỉnh của bảng biến thiên như sau:

```
\begin{tikzpicture}
\tkzTabInit[nocadre,lgt=1.5,espcl=2,deltacl=.5]
{\$x\$/0.7, \$f'(x)\$/0.7, \$f(x)\$/2}
{\$-\infty\$, \$-1\$, \$1\$, \$2\$, \$+\infty\$}
\tkzTabLine{,-,z,+,d,-,z,+,}
\tkzTabVar{+/$+\infty\$ , -/ \$-4\$ , +/$-3\$ , -/$-4\$ , +/$+\infty\$}
\end{tikzpicture}
```

Thế thì bạn sẽ được bảng biến thiên:

x	$-\infty$	-1	1	2	$+\infty$	
$f'(x)$	$-$	0	$+$	$-$	0	$+$
$f(x)$	$+\infty$		-3		$+\infty$	
		-4		-4		

Hoặc nếu dòng $f(x)$, tại $x = 1$ cũng không xác định thì bạn có thể thay bằng dòng lệnh:

```
\tkzTabVar{+/$+\infty\$ , -/ \$-4\$ , +D+/$+\infty\$ , -/$-4\$ , +/$+\infty\$}
```

x	$-\infty$	-1	1	2	$+\infty$
$f'(x)$		$-$	0	$+$	
$f(x)$	$+\infty$		$+\infty$	$+\infty$	$+\infty$
		-4		-4	

Hoặc sử dụng lệnh

```
\tkzTabVar{+/$+\infty$ , -/$-4$ , +D-/$+\infty$/$-\infty$, +/$-4$, -/$-\infty$}
```

Ta sẽ được:

x	$-\infty$	-1	1	2	$+\infty$
$f'(x)$		$-$	0	$+$	
$f(x)$	$+\infty$		$+\infty$	-4	$-\infty$
		-4		$-\infty$	

Hơn thế nữa, nếu bạn muốn một khoảng nào đó $f'(x)$ hoặc $f(x)$ không xác định bằng cách gạch chéo khoảng thuộc dòng đó, bạn có thể thay $+D$ bằng $+H$ hoặc $-H, \dots$

x	$-\infty$	-1	1	2	$+\infty$	
$f'(x)$	$-$	0		$-$	0	$+$
$f(x)$	$+\infty$			-4		
		-4		$-\infty$		$-\infty$

II.2 Đồ thị hàm số

Để vẽ đồ thị một hàm số trong **Tikz**. Ta đã có lệnh `\draw[...] plot ()`;

Cụ thể lệnh như sau:

```
\draw[tham số vẽ] plot (biến số , {Công thức hàm số});
```

Ta cần chú ý một số tham số sau trong lệnh vẽ đồ thị:

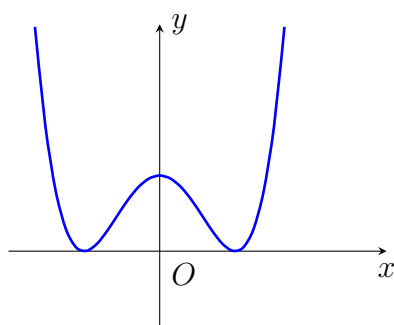
- **domain= a:b** Tham số này quy định cho ta vùng vẽ đồ thị. Với giá trị cụ thể của a và b thì **Tikz** vẽ cho ta phần đồ thị của hàm số với $a \leq x \leq b$. Việc xác định a, b cho phù hợp hình vẽ là điều rất cần thiết để hình vẽ không quá lớn hoặc tràn ra ngoài vùng cần vẽ.

- **smooth** Nếu ta không có tham số này, đồ thị của ta nhìn như các đường gấp khúc nối lại, vẽ rất xấu và nhìn không ra hệ thống công rãnh gì. Nhất là với các đồ thị là đường cong.
- **variable=\x** hoặc **variable=\y** Định dạng cho ta vẽ đồ thị theo biến x hay biến y . Với các đồ thị thông thường thì ta thường bỏ qua tham số này hoặc sử dụng **variable=\x** là chính.
- Ngoài ra, một số tham số màu (**color**), độ đậm nét của đường (**line width**) hoặc độ hòa trộn (**opacity**) ... cũng có thể tùy biến để được một đồ thị ưng ý.

Hơn nữa, quý vị cũng cần chú ý đến cách viết công thức hàm số để **Tikz** có thể hiểu được. Ví dụ như khi cần vẽ đồ thị hàm số $y = \frac{2}{3}x^3 - 3x + 1$ thì ta không viết như khi viết văn bản là `\dfrac{2}{3}x^3-3x+1` mà phải viết `(2/3)*(\x)^(3)-3*\x+1`. Như vậy các dấu phép tính viết bình thường, nhưng phân số khi tính toán ta phải dùng phép chia (dấu `/`), phép nhân phải dùng dấu nhân (dấu `*`) và phép lũy thừa thì vẫn dùng dấu `^` nhưng chú ý rằng các đối tượng này nằm trong cặp ngoặc đơn `()` chứ không phải cặp `{ }`.

Ta hãy bắt đầu vẽ một vài đồ thị hàm số quen thuộc. Một đoạn lệnh đơn giản như dưới đây đã vẽ được đồ thị hàm số $y = x^4 - 2x^2 + 1$:

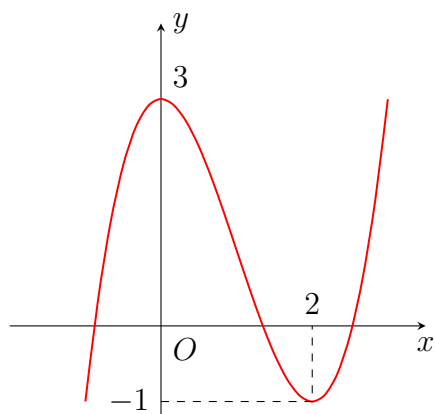
```
\begin{tikzpicture}[>=stealth]
\draw[->] (-2,0) --(0,0) node[below right]{$0$} --(3,0)node[below]{$x$};
%Trục hoành
\draw[->] (0,-1)--(0,3)node[right]{$y$};%trục tung
\draw[domain=-1.65:1.65,smooth,blue,line width=1]
plot(\x,{(\x)^(4)-2*(\x)^(2)+1});%Đồ thị
\draw (0,-1.5) node {Đồ thị hàm số  $y=x^4-2x^2+1$ };
%Chú thích bên dưới đồ thị
\end{tikzpicture}
```



Đồ thị hàm số $y = x^4 - 2x^2 + 1$

Quý vị cũng có thể thử vẽ đồ thị hàm bậc ba $y = x^3 - 3x^2 + 3$ với màu đỏ:

```
\begin{tikzpicture}[>=stealth]
\draw[->] (-2,0) --(0,0) node[below right]{$0$} --(3.5,0)node[below]{$x$};
%Trục hoành
\draw[->] (0,-1.25)--(0,4)node[right]{$y$};%trục tung
\draw[domain=-1:3,smooth,red,line width=0.75]
plot(\x,{(\x)^(3)-3*(\x)^(2)+3});%Đồ thị
\draw[dashed] (0,-1)node[left]{$-1$}--(2,-1)--(2,0)node[above]{$2$};
%Đường dóng nét đứt
\draw (0,3) node[above right]{$3$};%Hiển thị giá trị cực đại
\draw (0.75,-1.5) node {Đồ thị hàm số  $y=x^3-3x^2+3$ };
%Chú thích bên dưới đồ thị
\end{tikzpicture}
```



Đồ thị hàm số $y = x^3 - 3x^2 + 3$

Tương tự như vậy bạn hoàn toàn có thể vẽ đồ thị hàm số bất kỳ nào.

II.3 Tô một miền đồ thị

Khi vẽ hình, đặc biệt là đồ thị hàm số trong chương trình THPT, nhiều khi quý vị cần tô màu một miền giới hạn bởi các đường đồ thị nào đó (hoặc tô sọc kẻ, các ký tự khác để lấp đầy miền đó). Ta hãy sử dụng thư viện **Pattern**.

Để sử dụng thư viện này, trước hết ngay đầu file biên dịch hình vẽ, quý vị hãy gọi thư viện bằng lệnh:

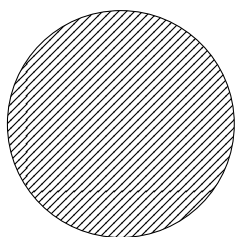
```
\usetikzlibrary{patterns}
```

Khi vẽ hình, bạn cần tô một vùng nào hãy dùng lệnh:

```
\draw[pattern = tham số tô] Miền cần tô;
```

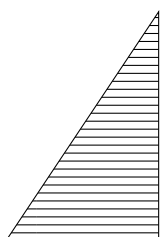
Ví dụ như bạn cần tô một hình tròn với đường kẻ sọc từ trái qua phải đi lên ta có thể dùng lệnh như sau:

```
\begin{tikzpicture}
\draw[pattern=north east lines] (0,0) circle (1.5);
\end{tikzpicture}
```



Tương tự, nếu cần tô một tam giác với kẻ sọc ngang ta lại có thể dùng lệnh:

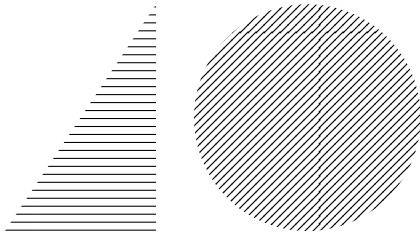
```
\begin{tikzpicture}
\draw[pattern=horizontal lines] (0,0) -- (2,0) -- (2,3)--cycle;
\end{tikzpicture}
```



Chú ý rằng nếu ta muốn tô sọc một vùng mà không cần đến đường biên ngoài thì ta có thể thêm tham số **draw=none** như ví dụ sau:

```
\begin{tikzpicture}
```

```
\draw[pattern=north east lines,draw =none] (4,1.5) circle (1.5);
\draw[pattern=horizontal lines, draw=none] (0,0) -- (2,0) -- (2,3)--cycle;
\end{tikzpicture}
```



Quý vị có thể quan tâm đến một số tham số có thể cần đến như dưới đây:

- horizontal lines
- vertical lines
- north east lines
- north west lines
- grid
- crosshatch
- dots
- crosshatch dots
- fivepointed stars
- sixpointed stars
- bricks
- checkerboard
- checkerboard light gray
- horizontal lines light gray
- horizontal lines gray



- horizontal lines dark gray
- horizontal lines light blue
- horizontal lines dark blue
- crosshatch dots gray
- crosshatch dots light steel blue

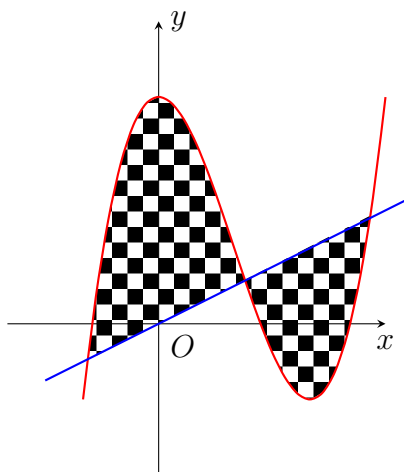


Ngoài ra quý vị hoàn toàn có thể tùy biến thêm bằng tham số **opacity** và **line width** để hình vẽ được theo như ý muốn.

Bây giờ ta hãy tô miền cho phần giới hạn bởi hai đồ thị hàm số nhé:

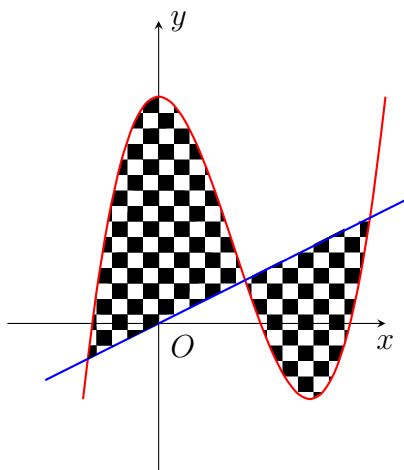
Giả sử ta có hai hàm số $y = x^3 - 3x^2 + 3$ và $y = \frac{1}{2}x$. Trước hết Quý vị hãy vẽ đồ thị hai hàm số này trên cùng một hệ trục tọa độ, sau đó quý vị hãy tìm hoành độ giao điểm của hai đồ thị này (nếu chính xác được là rất tốt, nếu không thì cũng tương đối). Chẳng hạn như bài toán này ta tìm ra hai hoành độ giao điểm (nhỏ nhất và lớn nhất (tương đối) xấp xỉ bằng -0.94 và 2.8) Từ đó ta dùng lệnh tô miền đồ thị:

```
\begin{tikzpicture}[>=stealth]
\draw[pattern=checkerboard,domain= -0.94:2.8 , draw=none]
plot (\x,{(\x)^3-3*(\x)^2+3});
\draw[->] (-2,0)--(0,0)node[below right]{$0$}--(3,0)node[below]{$x$};
\draw[->] (0,-2)--(0,4)node[right]{$y$};
\draw[name path=done,line width=0.8, red, domain=-1:3,smooth]
plot (\x,{(\x)^3-3*(\x)^2+3});
\draw[name path=dtwo,line width=0.8, blue, domain=-1.5:3.25,smooth]
plot (\x,{(1/2)*(\x)});
\end{tikzpicture}
```

Nếu quý vị muốn đẹp và chính xác tuyệt đối miền tô, quý vị hãy sử dụng môi trường **scope** kết hợp với lệnh **clip**.

Như ví dụ trên, ta cần biết chính xác tọa độ ba giao điểm của hai đồ thị hàm số, ta cần lấy giao điểm chính xác như dưới đây:



II.4 Đồ thị hàm số trong hệ tọa độ cực

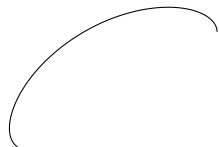
Cấu trúc lệnh vẽ đồ thị hàm số trong hệ tọa độ cực hoàn toàn đơn giản như sau:

```
\draw[domain=0:6.3,samples=200,smooth]
plot (canvas polar cs:angle=\x r,radius= {Biểu thức hàm  $r=f(x)$ });
```

Trong đó ta chú ý đến tham số **domain=0: 6.3** chính là góc ϕ trong hệ tọa độ cực với đơn vị là **rad**. Biến $\backslash x$ chính là góc θ và hàm số $r = f(x)$ chính là công thức tính r theo góc θ . Chẳng hạn ta muốn vẽ đồ thị hàm số $r = \cos(3\theta)$ với $0 \leq \theta \leq \pi$ thì ta dùng lệnh:

```
\draw[domain=0:6.3,samples=200,smooth]
plot (canvas polar cs:angle=\x r,radius= {\cos(3*\x r)});
```

Khi đó ta sẽ được:



Ngoài ra, ta cũng có thể vẽ đồ thị hàm số với hệ tọa độ đề các và phương trình tham số $\begin{cases} x = u(t) \\ y = v(t) \end{cases}$ bởi lệnh vẽ đồ thị như sau:

```
\draw [samples=100,smooth,domain=mint:maxt,variable=\t] plot ({u(t)},{v(t)});
```

Trong đó **mint** và **maxt** là giá trị nhỏ nhất, giá trị lớn nhất của tham số t cho vùng cần vẽ. Biểu thức $u(t)$ và $v(t)$ chính là hai biểu thức hoành độ và tung độ theo t .

Chẳng hạn ta vẽ đồ thị hàm số $\begin{cases} x = \cos^3 t \\ y = \sin^3 t \end{cases}$ trên vùng $0 \leq t \leq 120\pi$ như sau:

```
\begin{tikzpicture}[>=stealth]
\draw[->](-2,0)--(2,0);
\draw[->](0,-2)--(0,2);
\draw [blue, samples=100,smooth,domain=0:120*pi ,variable=\t]
plot ({(cos(\t))^3},{(sin(\t))^3});
\end{tikzpicture}
```

Ta được đồ thị:

