

# Mục lục

<b>I</b>	<b>Cơ bản về vẽ hình bằng Tikz</b>	<b>3</b>
I.1	Khai báo và môi trường vẽ . . . . .	3
I.2	Các đối tượng hình học . . . . .	3
I.2.1	Hệ trục tọa độ . . . . .	4
I.2.2	Vẽ Điểm . . . . .	5
I.2.3	Đoạn thẳng . . . . .	6
I.2.4	Đường tròn . . . . .	7
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 . . . . .	11
I.3.1	Khai báo các điểm dùng nhiều lần . . . . .	11
I.3.2	Giao điểm của các đường trong Tikz . . . . .	13
I.3.3	Vẽ ký hiệu vuông góc . . . . .	18
I.3.4	Phép Quay . . . . .	19
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 . . . . .	21
I.3.8	Vẽ hình tròn và tam giác nội tiếp . . . . .	22
I.3.9	Vẽ hình trụ . . . . .	24
I.4	Vẽ hình nón . . . . .	24
I.4.1	Vẽ mặt cầu . . . . .	25
I.4.2	Hình lăng trụ . . . . .	26
I.5	Làm đẹp hình vẽ . . . . .	26
I.5.1	Tô màu cho đường nét . . . . .	26
I.5.2	Tô màu cho một miền kín . . . . .	27
<b>II</b>	<b>Bảng biến thiên-Đồ thị hàm số</b>	<b>30</b>
II.1	Bảng biến thiên . . . . .	30
II.2	Kết hợp Tikz và tkz-tab vẽ bảng biến thiên tùy chỉnh . . . . .	33
II.3	Đồ thị hàm số . . . . .	36
II.4	Đồ thị hàm số trong hệ tọa độ cực . . . . .	38
<b>III</b>	<b>Tô miền đồ thị hàm số với Tikz</b>	<b>40</b>
III.1	Vẽ đồ thị hàm số . . . . .	40
III.2	Tô miền đồ thị hàm số . . . . .	42
III.2.1	Khi đã biết hoành độ các điểm cận của miền cần tô . . . . .	42
III.2.2	Tô miền đồ thị không biết hoành độ giao điểm . . . . .	45
III.2.3	Sử dụng even odd rule . . . . .	48

III.3 Các kiểu tô miền . . . . .	50
<b>IV Đường cong</b>	<b>53</b>
IV.1 Controls . . . . .	53
IV.1.1 Cú pháp lệnh vẽ . . . . .	53
IV.1.2 Kinh nghiệm vẽ . . . . .	55
IV.1.3 Phân tích hình và lựa chọn code vẽ . . . . .	56
IV.1.4 một vài đồ thị vẽ bằng controls . . . . .	60

# I Cơ bản về vẽ hình bằng Tikz

## I.1 Khai báo và môi trường vẽ

Khai báo gói lệnh:

```
\usepackage{tikz}
\usetikzlibrary{calc,angles}
```

Môi trường vẽ:

```
\begin{tikzpicture}
.....
\draw ....
.....
\end{tikzpicture}
```

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

```
\documentclass{standalone}
\usepackage{tikz}
\usetikzlibrary{calc,angles}
\begin{document}
\begin{tikzpicture}
.....
\draw ....
.....
\end{tikzpicture}
\end{document}
```

## I.2 Các đối tượng hình học

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

`\draw [parameters] (point) object;`

trong đó **parameters** là các tham số tham gia vào việc vẽ hình, **point** là tọa độ điểm xuất phát bút vẽ và **object** là đối tượng cần 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.

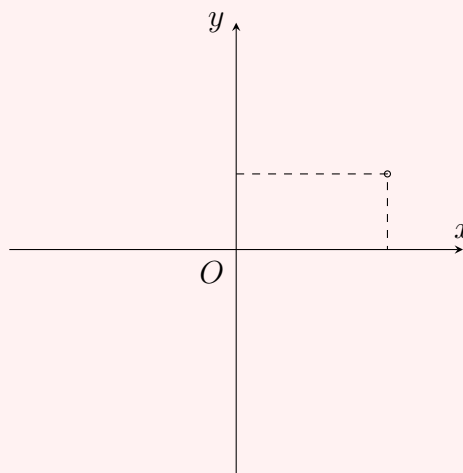
- **color=**: nếu muốn tô màu nét vẽ.
- **rotate=**: nếu muốn xoay đối tượng vẽ quanh góc tọa độ một góc nào đó.
- .....

Tọa độ điểm có thể dùng tọa độ Đề-các thông thường dạng  $(a, b)$  hoặc tọa độ cực dạng  $(\alpha : r)$ .

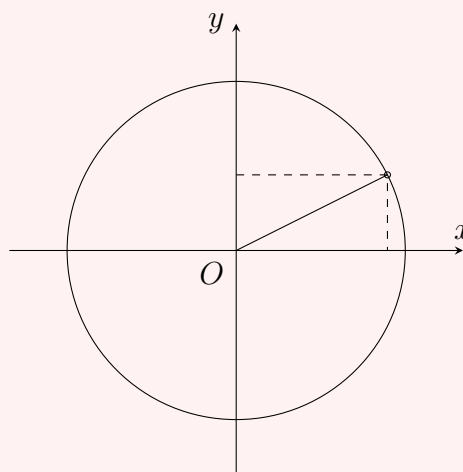
### I.2.1 Hệ trục tọa độ

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.

```
\begin{tikzpicture}[>=stealth]
\draw[->] (-3,0)--(3,0);
\draw[->] (0,-3)--(0,3);
\draw[dashed] (0,1)--(2,1)--(2,0);
\draw (2,1) circle (0.04);
\draw (3,0) node[above]{$x$} (0,3)
  node[left]{$y$} (0,0) node[below
left]{$O$};
\end{tikzpicture}
```



```
\begin{tikzpicture}[>=stealth]
\def\r{sqrt(5)}
\draw[->] (-3,0)--(3,0);
\draw[->] (0,-3)--(0,3);
\draw[dashed] (0,1)--(2,1)--(2,0);
\draw (0,0) circle (\r);
\draw (0,0)--(2,1);
\draw (2,1) circle (0.04);
\draw (3,0) node[above]{$x$} (0,3)
  node[left]{$y$} (0,0) node[below
left]{$O$};
\end{tikzpicture}
```



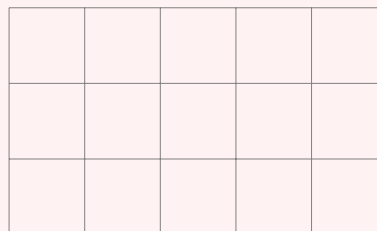
Để dễ dàng cho bạn khi bắt đầu tập vẽ, ta nên tạo lưới tọa độ vùng vẽ để khi xem kết quả ta dễ dàng hình dung ra cách thức và rút kinh nghiệm cho việc sử dụng lệnh bằng cú pháp:

```
\draw[step=1,gray,thin] (lower left corner) grid (upper right corner);
```

trong đó **lower left corner** là tọa độ điểm góc trái phía dưới và **upper right corner** là tọa độ điểm góc phải bên trên của vùng vẽ lưới

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

```
\begin{tikzpicture}
\draw[step=1,gray,very thin]
  (-2,-1) grid (3,2);
\end{tikzpicture}
```



### 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  $(\alpha : r)$ .

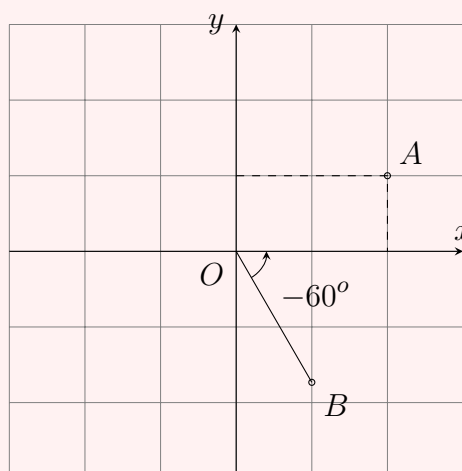
Ví dụ ta muốn vẽ điểm  $A(3; -1)$  ta dùng lệnh sau: `\draw (3,-1) circle (1pt);`

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 (1pt);`. Khi đó ta sẽ vẽ được điểm  $B$  sao cho  $OB = 2$  và góc giữa  $OB$  và  $Ox$  là  $60^\circ$ .

Hãy xem ví dụ dưới đây:

```
\begin{tikzpicture}[>=stealth]
\draw[step=1,gray,very thin]
  (-3,-3) grid (3,3);
\draw[>] (-3,0)--(3,0);
\draw[>] (0,-3)--(0,3);
\draw[dashed] (0,1)--(2,1)--(2,0);
\draw (2,1) circle (0.04);
\draw (3,0) node[above]{$x$} (0,3)
  node[left]{$y$} (0,0) node[below
    left]{$O$};
\draw (-60:2) circle (0.04);
\draw (-60:2) -- (0,0);
\draw (2,1) node[above right]{$A$}
  (-60:2) node[below right]{$B$};
\draw[>] (-60:4mm) arc (-60:0:4mm);
\draw (-30:5mm) node[below
  right]{$-60^\circ$};
\end{tikzpicture}
```

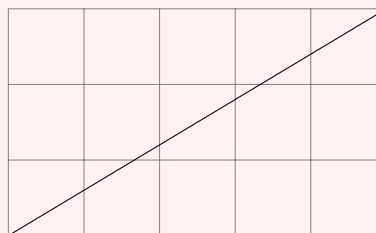


Chú ý rằng lệnh vẽ điểm là `\draw (point) circle (radius);` trong đó **point** là tọa độ điểm và **radius** là bán kính điểm (thường dùng là 1pt hoặc 2pt).

### 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 (point one) - (point two);`, trong đó **point one** và **point two** là tọa độ hai điểm đầu mút của đoạn thẳng. Ví dụ muốn vẽ đoạn thẳng  $AB$  với  $A(2;1)$  và  $B(-3,-1)$  ta dùng lệnh:

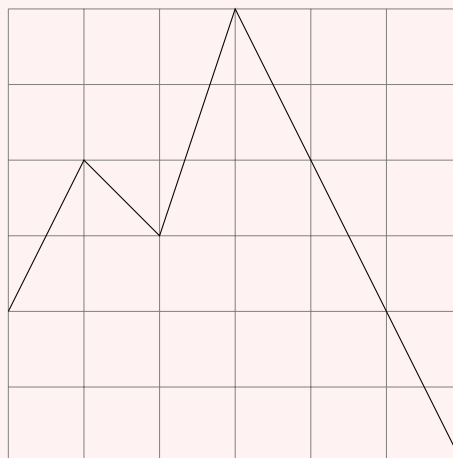
```
\begin{tikzpicture}
\draw[step=1,gray,very thin]
  (-3,-2) grid (2,1);
\draw (2,1)--(-3,-2);
\end{tikzpicture}
```



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

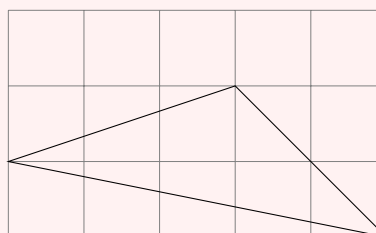
Chẳng hạn: `\draw (0,1)-(1,3)-(2,2)-(3,5)-(4,-1);`. Ta sẽ được đường gấp khúc sau:

```
\begin{tikzpicture}
\draw[step=1,gray,very thin] (0,-1)
  grid (6,5);
\draw
  (0,1)--(1,3)--(2,2)--(3,5)--(6,-1);
\end{tikzpicture}
```



Hơn thế nữa, nếu muốn vẽ đa giác khi biết các đỉnh, cũng thật là dễ dàng với việc vẽ đường khép kín khi sử dụng `cycle` như sau: `\draw(0,1)-(3,2)-(5,0)-cycle;`

```
\begin{tikzpicture}
\draw[step=1,gray,very thin] (0,0)
  grid (5,3);
\draw (0,1)--(3,2)--(5,0)--cycle;
\end{tikzpicture}
```



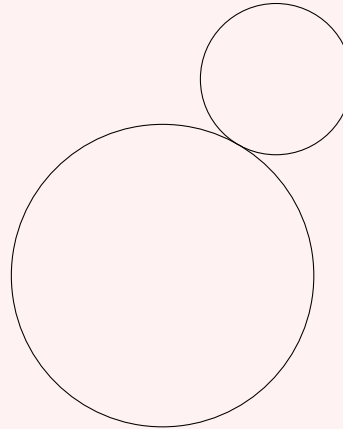
### I.2.4 Đường tròn

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 (point) circle (radius);` trong đó **point** là tọa độ tâm và **radius** là bán kính của đường tròn.

Chẳng hạn ta dùng đoạn lệnh sau, 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:

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

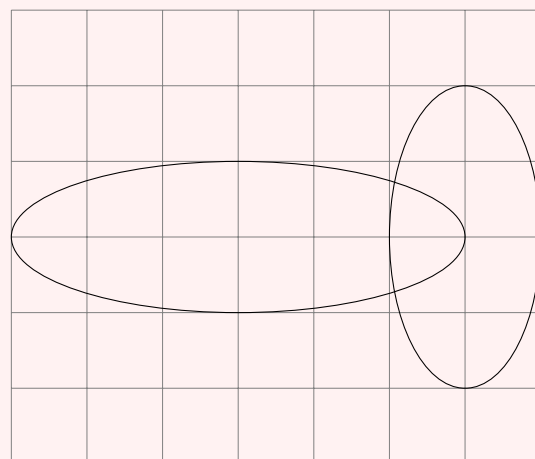


### I.2.5 Đường Ellip

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

`\draw (point) ellipse ({xradius} and {yradius});` trong đó **point** là tâm, **xradius** là bán trục hoành và **yradius** là bán trục tung của Ellipse.

```
\begin{tikzpicture}
\draw[step=1,gray,very thin]
(-3,-3) grid (4,3);
\draw (0,0) ellipse ({3} and {1});
\draw (3,0) ellipse ({1} and {2});
\end{tikzpicture}
```



### I.2.6 Cung tròn, Ellip - Đường cong

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

`\draw (point) arc [start angle= ,end angle= ,x radius= ,y radius= ];`

trong đó **point** là điểm xuất phát để vẽ cung, **start angle** là góc xuất phát, **end angle**

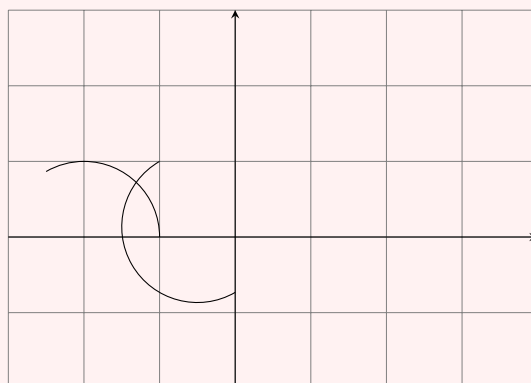
là góc kết thúc, và **x radius**, **y radius** lần lượt là bán trục hoành và bán trục tung của cung.

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

Ta được hình vẽ như sau:

```
\begin{tikzpicture}[>=stealth]
\draw[step=1,gray,very thin]
  (-3,-2) grid (4,3);
\draw[>-] (-3,0)--(4,0);
\draw[>-] (0,-2)--(0,3);
\draw (-1,0) arc [ start angle=0,
  end angle=120, x radius=1,y
  radius=1];
\draw (-1,1) arc [ start angle=120,
  end angle=300, x radius=1,y
  radius=1];
\end{tikzpicture}
```



Ở hình vẽ trên, lệnh vẽ cung tròn xuất phát từ điểm  $(-1;0)$ , góc vẽ xuất phát từ  $0^\circ$  và kết thúc ở  $120^\circ$ , 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  $120^\circ$  và kết thúc ở  $300^\circ$ . 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 (point) arc (start angle: end angle: radius);
```

Tương tự, **Tikz** cũng cho ta lệnh vẽ cung Ellipse rút gọn:

```
\draw (point) arc (start angle : end angle : {x radius} and {y radius});
```

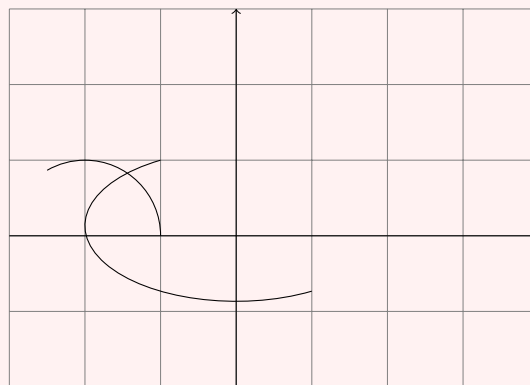
Quý vị hãy xem thử đoạn lệnh dưới đây và quan sát kết quả:



```

\begin{tikzpicture}
\draw[step=1,gray,very thin]
  (-3,-2) grid (4,3);
\draw[->] (-3,0)--(4,0);
\draw[->] (0,-2)--(0,3);
\draw (-1,0) arc (0 :120:1);
\draw (-1,1) arc(120 : 300 : {2}
  and {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 (point A) .. controls (point one) and (point two) .. (point B);
```

trong lệnh vẽ trên, **point A** và **point B** là hai điểm đầu và cuối của đường cong. Các điểm **point one** và **point two** là hai điểm điều khiển đường cong.

- Lệnh **to**. Cú pháp:

```
\draw (point one) to [bend left=] {point two};
```

trong đó **bend left** hoặc **bend right** độ cong của đường.

- Lệnh **parabola**. Cú pháp:

```
\draw (A) parabola (B);
```

với lệnh **parabola** ta có hai dạng là **parabola** và **parabola [bend at end]**. Quý vị hãy thử sử dụng từng lệnh để thấy rõ hiệu quả. Ngoài ra, khi muốn vẽ một đoạn parabol có chứa đỉnh của parabol, quý vị cũng có thể sử dụng:

```
\draw (A) parabola bend (B) (C);
```

Lệnh này sẽ cho ta đoạn parabol xuất phát từ điểm *A*, có đỉnh là điểm *B* và kết thúc ở điểm (*C*).

- Lệnh **plot coordinates**. Cú pháp:

```
\draw[smooth] plot coordinates {(A) (B) (C)...};
```

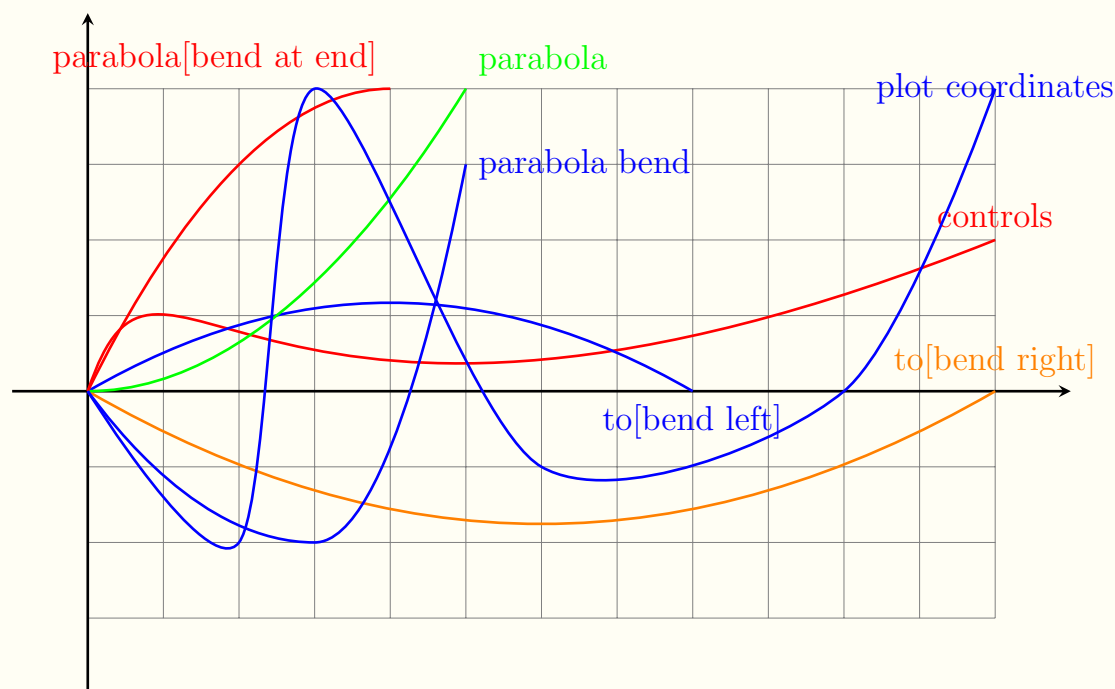
Với lệnh này quý vị sẽ thấy một đường cong đi qua tất cả các điểm *A*, *B*, *C* và có thể hơn thế nữa nếu quý vị thêm tiếp nhiều điểm khác nữa. Chú ý rằng nếu quý vị không dùng tham số **smooth** thì lệnh này sẽ vẽ cho quý vị một đoạn đường gấp khúc chứ không còn là đường cong nữa.

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:

```

\begin{tikzpicture}[line width=1]
\draw[step=1,gray,very thin] (0,-3) grid (12,4);
\draw[-stealth] (-1,0)--(13,0);
\draw[-stealth] (0,-4)--(0,5);
\draw[red] (0,0) .. controls (1,3) and (2,-2) .. (12,2)node[above]{controls};
\draw[blue] (0,0) to [bend left =30] (8,0)node[below]{to[bend left]};
\draw[orange] (0,0) to [bend right =30] (12,0)node[above]{to[bend right]};
\draw[green] (0,0) parabola (5,4)node[above right]{parabola};
\draw[red] (0,0) parabola[bend at end] (4,4)node[above left]{parabola[bend at end]};
\draw[blue] (0,0) parabola bend (3,-2) (5,3)node[right]{parabola bend};
\draw[blue,smooth] plot coordinates {(0,0) (2,-2) (3,4) (6,-1) (10,0)
(12,4)}node{plot coordinates};
\end{tikzpicture}

```



### 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 (point) node[parameters]{Content notes};
```

trong lệnh trên **point** là điểm đặt nội dung ghi chú, **parameters** là các tham số và **Content notes** chính là 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

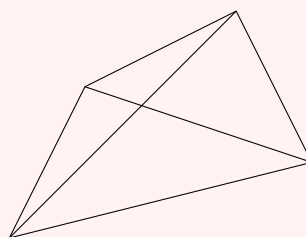
Để vẽ một hình vẽ hình học phẳng thuần túy, thông thường trong các hình có những điểm liên quan đến nhau. Từ đó xuất phát nhu cầu cần có những điểm phải lặp lại, những điểm do yêu cầu bài toán liên quan. Khi đó ta đi tìm tọa độ từng điểm rất mệt mỏi. Bạn không cần lo lắng vì **Tikz** đã biết trước điều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

#### I.3.1 Khai báo các điểm dùng nhiều lần

Để khai báo một điểm ta dùng lệnh:

`\coordinate (point name) at (point);` trong đó **point name** là tên điểm do ta tự đặt (chẳng hạn  $A, B, C\dots$ ) và **point** là 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  $k$  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  $\overrightarrow{AM} = k\overrightarrow{AB}$ . Hẳn nhiên nếu  $M$  là trung điểm  $AB$  thì lệnh khai báo sẽ là:

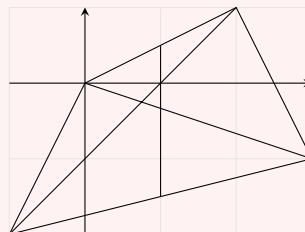
`\coordinate (m) at ($(A)!0.5!(B)$);`

Ta hãy thực hành lệnh này xem sao:

```

\begin{tikzpicture}
\draw[step=1,thin,gray!20] (-1,-2)
  grid (3,1);
\draw[-stealth] (-1,0)--(3,0);
\draw[-stealth] (0,-2)--(0,1);
\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:

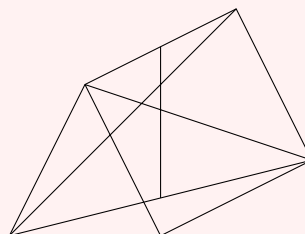
```
\coordinate (E) at ($(A)+(1,-2)$);
```

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) (M)--(N)
  (A)--(E)--(C);
\end{tikzpicture}

```

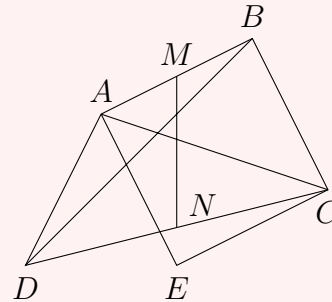


Và bây giờ, ta có thể hiển thị tên các điểm cho hình vẽ của mình với đoạn mã lệnh đầy đủ như sau:

```

\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) (M)--(N)
      (A)--(E)--(C);
\draw (A) node[above]{$A$} (B)
      node[above]{$B$}
      (M) node[above]{$M$} (C)
      node[below]{$C$}
      (D) node[below]{$D$} (N) node[above
        right]{$N$}
      (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 nói 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.

Trước hết, quý vị cần gọi thư viện **intersections** của tikz bằng `\usetikzlibrary{intersections}`. Sau đó ta bắt đầu gán tên đường.

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

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

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

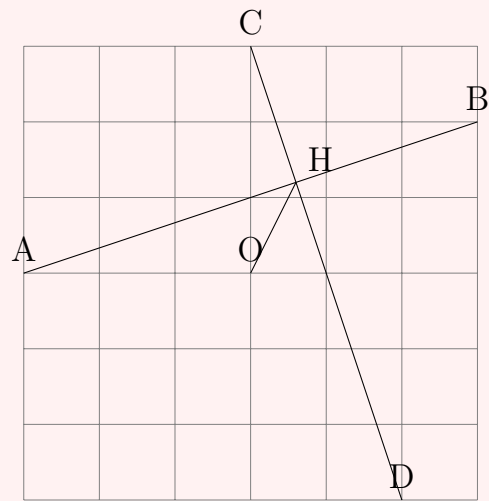
```
\path [name intersections={of=one and two,by=name point}];
```

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}

```

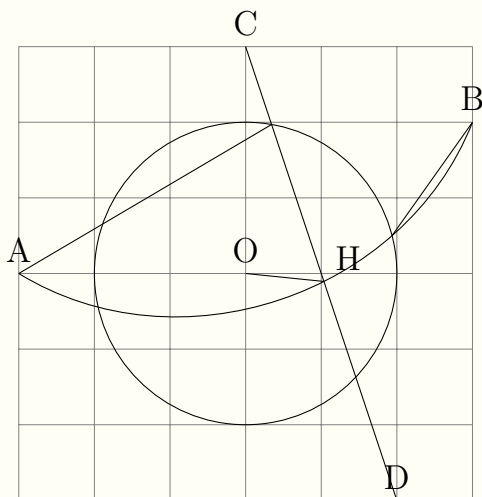


Hơn thế nữa, giao điểm của đường thẳng và đường cong cũng thật sự chính xác:

```

\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 ng 1 and tên ng 2}];
\coordinate[label=Tên giao im 1] (m) at (intersection-1);
\coordinate[label=Tên giao im 2] (k) at (intersection-2);

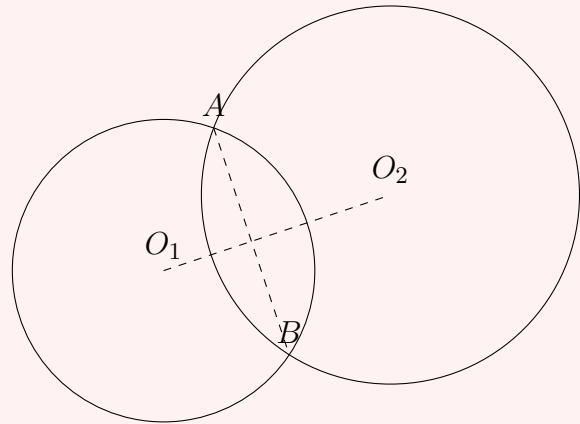
```

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);
\coordinate[label=$O_2$] (O2) at
  (3,1);
\draw [name path=done] (O1) circle
  (2);
\draw[name path=dtwo] (O2) circle
  (2.5);
\path [name intersections = {of =
  done and dtwo}];
\coordinate[label=$A$] (A) at
  (intersection-1);
\coordinate[label=$B$] (B) at
  (intersection-2);
\draw[dashed] (A)--(B) (O1)--(O2);
\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 {...}`).

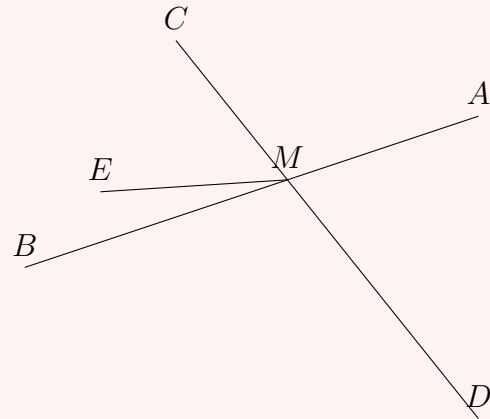
Chẳng hạn với code vẽ:



```

\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}];
\coordinate[label=$M$] (M) at
  (intersection-1);
\draw (E)--(M);
\end{tikzpicture}

```

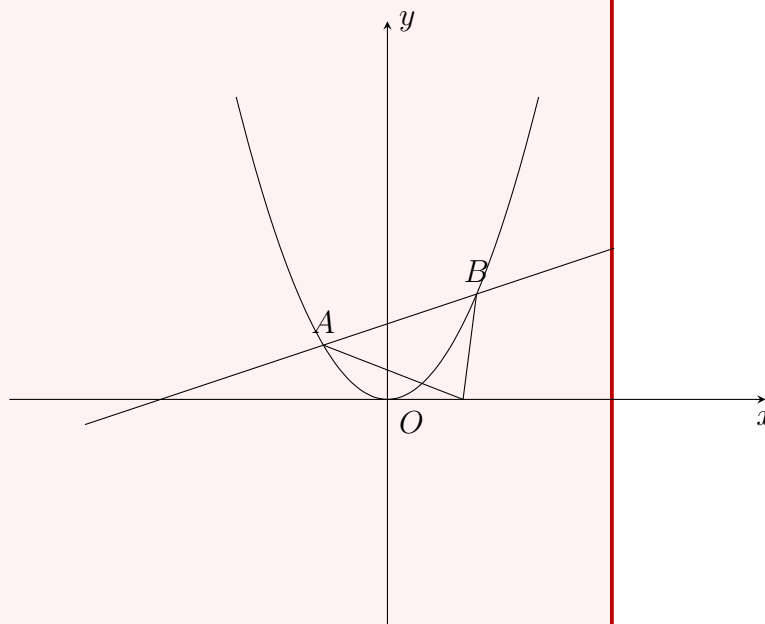


Hơn thế nữa, bạn cũng hoàn toàn có thể tìm giao điểm của các đồ thị hàm số với cách vẽ tương tự. Chẳng hạn bạn cần vẽ đồ thị hàm số  $y = x^2$  và đồ thị hàm số  $y = \frac{1}{3}x + 1$ . Thêm nữa, bạn cần lấy các giao điểm của các đồ thị này để nối với một điểm nào đó, bạn hãy xem ví dụ sau:

```

\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=one,smooth]
plot[domain=-2:2]
(\x,{(\x)^2});
\draw [name path=two,smooth]
plot[domain=-4:3]
(\x,{(1/3)*(\x)+1});
\path [name intersections={of =one
and two}];
\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:

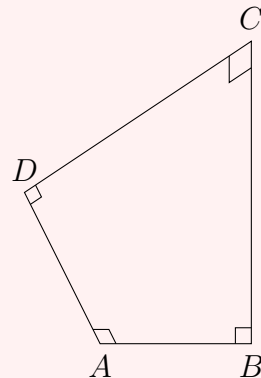
```
\khvuong[size= ](A,B,C)
```

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)
\khvuong[size=10pt] (B,C,D)
\khvuong[size=5pt] (C,D,A)
\khvuong[size=6pt] (D,A,B)
\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  $60^\circ$  thì ta có thể dùng lệnh:

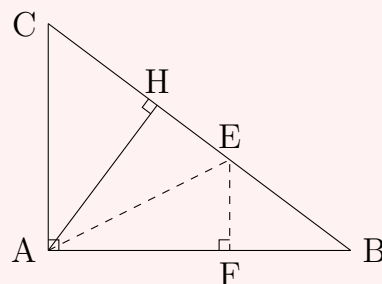
```
\coordinate (C) at ($(A)!1!60:(B)$);
```

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

```

\begin{tikzpicture}[>=stealth]
\pgfmathsetmacro{\b}{4}
\pgfmathsetmacro{\c}{3}
\pgfmathsetmacro{\h}{(\b
    *\c)/sqrt((\b)^2+(\c)^2)}
\coordinate[label=left:A] (A) at
    (0,0);
\coordinate[label=right:B] (B) at
    (\b,0);
\coordinate[label=left:C] (C) at
    (0,\c);
\draw[name path=ab] (A)--(B);
\draw[name path=ac] (A)--(C);
\draw[name path=bc] (B)--(C);
\draw[name path=tib,opacity=0] (A)
    arc (180:90:\b/2);
\path [name intersections={of=bc
    and tib}];
\coordinate[label=H] (H) at
    (intersection-1);
\draw (A)--(H) ;
\pgfmathsetmacro{\gocc}{atan(\b/\c)}
\coordinate[label=E] (E) at
    ($(C)!1!\gocc:(A)$);
\coordinate[label=below:F] (F) at
    ($(A)!1!-\gocc:(H)$);
\draw[dashed] (A)--(E)--(F);
\khvuong[size=4pt] (A,H,C)
\khvuong[size=4pt] (E,F,A)
\khvuong[size=4pt] (C,A,B)
\end{tikzpicture}

```



### 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?

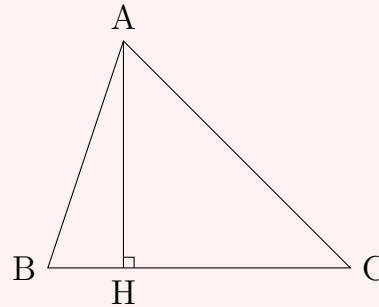
Thật đơn giản với lệnh khai báo sau: `\coordinate (H) at ($(B)!(A)!(C)$);`. Lúc này bạn đã có thể tìm thấy  $H$  là chân đường cao kẻ từ  $A$  tới  $BC$  rồi. Việc của bạn lúc đó chỉ còn là dùng lệnh `\draw` để vẽ.

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** ( $\alpha$ ): Trả về giá trị  $\sin \alpha$ .
- **cos** ( $\alpha$ ): Trả về giá trị  $\cos \alpha$ .
- **tan** ( $\alpha$ ): Trả về giá trị  $\tan \alpha$ .
- **asin** ( $b$ ): Trả về giá trị  $\arcsin b$ .
- **pow** ( $\alpha, a$ ): Trả về giá trị  $a^\alpha$ .
- **ln** ( $a$ ): Trả về giá trị  $\ln a$ .
- **exp** ( $\alpha$ ): Trả về giá trị  $e^\alpha$ .
- **log<sub>a</sub>** ( $b$ ) : 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

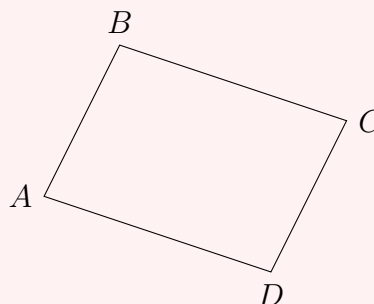
Ví dụ vẽ hình bình hành rất có ích cho bạn. Khi bạn muốn vẽ một hình nào đó, quá trình tư duy của bạn phải thông qua các bước: phân tích hình; đặt tọa độ những điểm cơ bản dựa vào mối liên hệ với các điểm còn lại; cuối cùng là đặt lệnh vẽ.

- 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 (B) at (5,3);
\coordinate (C) 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}
```



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.

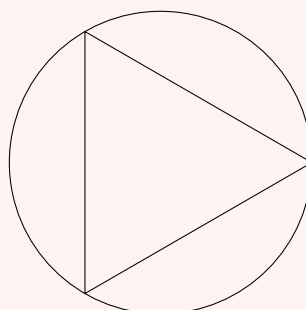
### I.3.8 Vẽ hình tròn và tam giác nội tiếp

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 vẽ đường tròn và một tam giác đều nội tiếp đường tròn đó:

```
\begin{tikzpicture}
\draw (0:0) circle (2);
\draw
  (0:2)--(120:2)--(240:2)--cycle;
\end{tikzpicture}
```

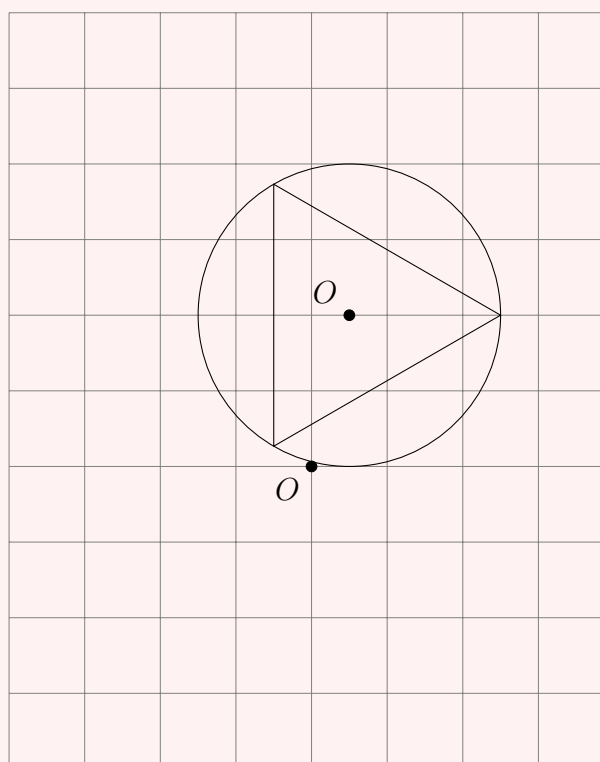


Thật là tuyệt vời với một vài lệnh đơn giản. Nhưng trong thực tế, có voi đòi hai bà trưng, thế nên bạn không muốn cái đường tròn kia có tâm là  $(0;0)$  thì ta làm thế nào? Dịch chuyển đường tròn thì không hề khó khăn, nhưng dịch chuyển cái tam giác bên trong với bạn lại có vẻ không dễ dàng. Không sao, 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;
```

Ta sẽ có như thế này:

```
\begin{tikzpicture}
\draw [step=1,gray,very thin]
  (-4,-4) grid (4,6);
\draw[shift={(0.5,2)}] (0:0) circle
  (2);
\draw[shift={(0.5,2)}]
  (0:2)--(120:2)--(240:2)--cycle;
\draw[fill=black,shift={(0.5,2)}]
  (0,0) node[above left] {$O$}
  circle (2pt);
\draw[fill=black] (0,0) node[below
  left] {$O$} circle (2pt);
\end{tikzpicture}
```



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.

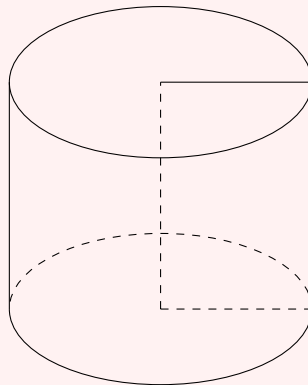
### I.3.9 Vẽ hình trụ

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];
\draw (-2,0) arc [ start angle=180,
  end angle=360, x radius=2,y
  radius=1];
\draw (-2,0)--(-2,3) (2,3)--(2,0)
(0,3)--(2,3);
\draw[dashed] (0,0)--(0,3)
(0,0)--(2,0);
\draw (0,3) ellipse ({2} and {1});
\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.4 Vẽ hình nón

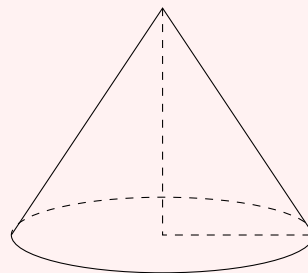
Cũng giống như vẽ hình trụ. Hình nón cũng có đáy được vẽ bằng hai nửa ellipse như sau:



```

\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}

```



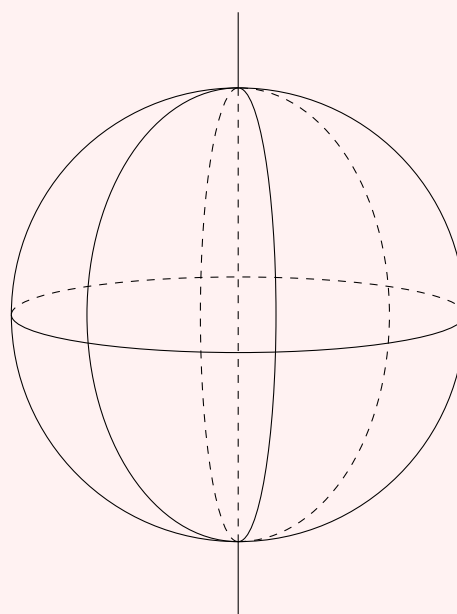
### I.4.1 Vẽ mặt cầu

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

```

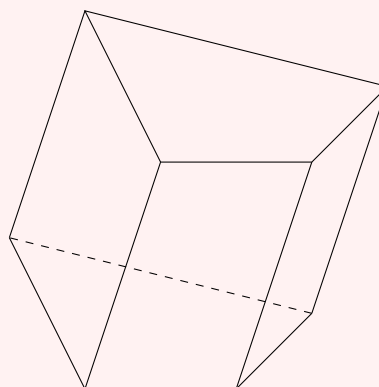


Hoàn toàn đơn giản phải không bạn? Bạn hãy tập thay đổi các thông số để tìm hiểu kỹ hơn.

### I.4.2 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.5 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.5.1 Tô màu cho đường nét

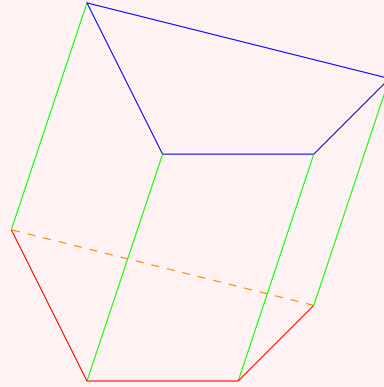
Trong lệnh vẽ `\draw`. Tham số màu là một tham số thường được dùng. Với tên gọi các màu hoặc dùng mã màu trong bảng màu [rgb] là thực sự cần thiết. Chẳng hạn muốn vẽ một đoạn thẳng màu xanh ta có thể dùng lệnh `\draw[blue] (A)-(B);`. Hơn thế nữa ta có thể thay đổi độ đậm nhạt cho màu của đường nét hoặc 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 biểu 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 (AA) at ($(A)+(1,3)$);
\coordinate (BB) at ($(B)+(1,3)$);
\coordinate (CC) at ($(C)+(1,3)$);
\coordinate (DD) at ($(D)+(1,3)$);
\draw[blue]
  (AA)--(BB)--(CC)--(DD)--cycle;
\draw[green] (A)--(AA) (B)--(BB)
  (C)--(CC) (D)--(DD);
\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.5.2 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:

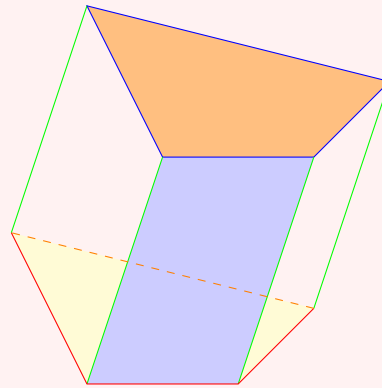
- **Fill:** Hãy dùng lệnh `\fill[color] object;` bạn sẽ thấy kết quả thật đẹp mắt.
- **Shade:** Tương tự như **Fill**, nhưng thật sự tô màu bằng lệnh này quả thật hiệu quả đẹp hơn rất nhiều.

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 (AA) at ($(A)+(1,3)$);
\coordinate (BB) at ($(B)+(1,3)$);
\coordinate (CC) at ($(C)+(1,3)$);
\coordinate (DD) at ($(D)+(1,3)$);
\fill[yellow!20]
  (A)--(B)--(C)--(D)--cycle;
\fill[orange!50]
  (AA)--(BB)--(CC)--(DD)--cycle;
\fill[blue!20]
  (A)--(AA)--(BB)--(B)--cycle;
\draw[blue]
  (AA)--(BB)--(CC)--(DD)--cycle;
\draw[green] (A)--(AA) (B)--(BB)
  (C)--(CC) (D)--(DD);
\draw[color=red] (D)--(A)--(B)--(C);
\draw[dashed, color=orange]
  (C)--(D);
\end{tikzpicture}

```

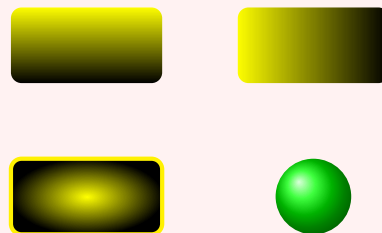


Với các lệnh tô màu, ta cũng cần quan tâm đến các tham số màu. Đặc biệt là các chế độ tô:

```

\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] (0,-2)
  rectangle +(2,1);
\shade[ball color=green] (4,-1.5)
  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.

## 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}
```

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

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}
```

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

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}

```

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

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

```

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

```

Bạn sẽ có ngay kết quả:

```

\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{+/$+\infty$ , -/$-4$ , +D+/$+\infty$ , -/$-4$ , +/$+\infty$}
\end{tikzpicture}

```

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

Hoặc sử dụng lệnh:

```

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

```

Ta sẽ được:



```

\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{ +/$+\infty$, -/$-4$, +D-$+\infty$/-$-\infty$, +/$-4$, -/$-\infty$ }
\end{tikzpicture}

```

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

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  $-H$ ,...

```

\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, h, d, -, z, +, }
\tkzTabVar{ +/$+\infty$, -H/$-4$, D-$+\infty$/-$-\infty$, +/$-4$, -/$-\infty$ }
\end{tikzpicture}

```

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

## II.2 Kết hợp Tikz và tkz-tab vẽ bảng biến thiên tùy chỉnh

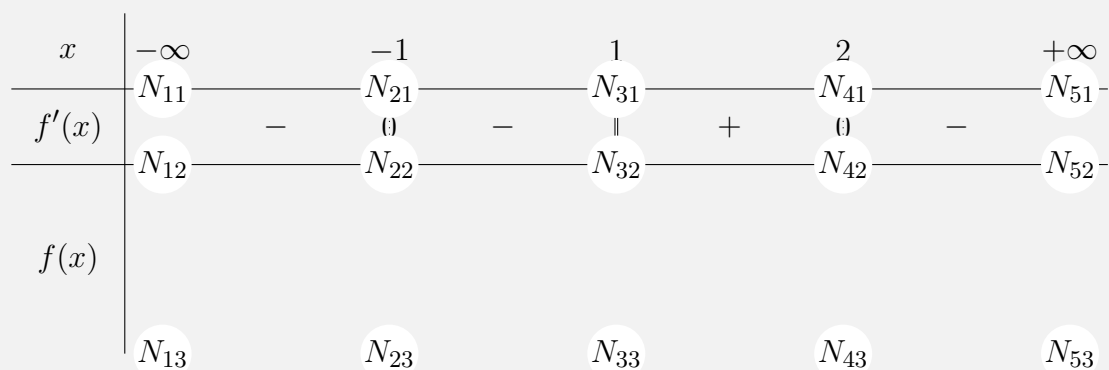
Khi vẽ bảng biến thiên của các hàm số thực tế. Trong nhiều trường hợp quý vị cần tùy chỉnh bảng biến thiên theo ý mình. Tất nhiên trong bảng biến thiên đó thì hai dòng  $x$  và  $f'(x)$  hầu như không có gì cần tùy chỉnh. Chỉ cần tùy chỉnh dòng  $f(x)$  mà thôi.

Để tùy chỉnh dòng  $f(x)$ , trước hết ta tìm hiểu về cách định nghĩa các điểm trong bảng biến thiên. Gói **tkz-tab** đã định nghĩa các điểm cơ bản của bảng theo vị trí như hình sau:

```

\begin{tikzpicture}
\tkzTabInit[nocadre,lgt=1.5,espcl=3,deltacl=0.5]
{$x$/1, $f'(x)$/1, $f(x)$/2.5}
{$-\infty$, $-1$, $1$, $2$, $+\infty$}
\tkzTabLine{-,z,-,d,+,z,-,}
\foreach \i in {1,...,5}{
\foreach \j in {1,...,3}{
\draw (N\i\j) node[circle,fill=white,inner sep=0.1]{$N_{\i\j}$};
}
}
\end{tikzpicture}

```



Như vậy Quý vị hoàn toàn có thể vẽ tùy ý dòng  $f(x)$  theo ý mình dựa vào các điểm đã được định nghĩa  $N_{ij}$ . Số điểm này phụ thuộc bảng của quý vị dài hay ngắn (có nhiều giá trị  $x$  hay không). Chẳng hạn như ở bảng trên, khi ta vẽ chiều biến thiên từ  $-\infty$  đến 1 là nghịch biến. Nếu bình thường vẽ bằng lệnh `\tkzTabVar` thì sẽ vẽ thẳng một mũi tên từ điểm  $N_{12}$  đến điểm  $N_{33}$ . Tại hạ lại muốn vẽ kiểu mũi tên từ  $+\infty$  đến  $f(-1) = -2$  rồi tiếp tục lại vẽ mũi tên thứ hai từ  $f(-1)$  đến  $f(1) = -4$ . Như vậy ta sẽ vẽ bằng tikz thuần dựa vào các điểm có sẵn:

```

\begin{tikzpicture}
\tkzTabInit[nocadre,lgt=2,espcl=3,deltacl=0.5]
{$x$/1, $f'(x)$/1, $f(x)$/2.5}
{$-\infty$, $-1$, $1$, $2$, $+\infty$}
\tkzTabLine{, -, z, -, d, +, z, -, }
\draw (N12) node[below] (A) {$+\infty$} ($ (N22)!0.5!(N23)$) node (B) {$-2$} (N33)
node[above left] (C) {$-4$};
\draw[-stealth] (A)--(B);
\draw[-stealth] (B)--(C);
\end{tikzpicture}

```

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

Tiếp theo, quý vị muốn vẽ đường thẳng đứng biểu diễn hàm số không xác định tại  $x = 1$ . Ta hãy vẽ đoạn thẳng từ  $N_{32}$  đến  $N_{33}$  với tham số **double** như sau:

```
\draw[double] ([yshift=-1mm]N32)-(N33);
```

Quý vị nên chú ý **[yshift=-1mm]** ở điểm  $N_{32}$  chính là để đường thẳng đứng cách vạch kẻ ngang 1mm. Nếu quý vị không muốn tách rời ra thì cứ việc bỏ nó đi là xong.

Phía bên phải của bảng biến thiên, Quý vị vẫn vẽ bình thường hai mũi tên biểu diễn đồng biến từ  $x = 1$  đến  $x = 2$  và nghịch biến từ  $x = 2$  đến  $+\infty$ . Tuy nhiên giá trị  $f(2) = -3$  lại nhỏ hơn  $-2$  nên quý vị muốn nó thấp hơn cả  $f(-1) = -2$ . Vậy thì quý vị hãy xác định điểm đặt  $f(2) = -3$  bằng công thức  **$(N42)!0.75!(N43)$$**  như sau:

```

\draw (N33) node[above right] (D) {$-\infty$} ($ (N42)!0.55!(N43)$)
node (E) {$-3$} (N53) node[above] (F) {$-\infty$};

```

Chú ý rằng con số 0.55 ở trên quý vị có thể tùy ý chọn sao cho phù hợp. Hơn thế nữa là thông số của  **$(N42)!0.75!(N43)$$**  ở **\tkzTabInit** quý vị cũng nên chọn thay 2.5 bằng các thông số khác để nhìn bảng hài hòa hơn (độ cao của dòng  $f(x)$ ).

Cuối cùng thì quý vị vẽ nốt hai mũi tên còn lại để hoàn thành bảng biến thiên:

```

\draw[-stealth] (D)--(E);
\draw[-stealth] (E)--(F);

```

Kết quả cuối cùng, quý vị hoàn thành bảng biến thiên theo tùy chỉnh của mình:

```

\begin{tikzpicture}
\tkzTabInit[nocadre,lgt=2,espcl=3,deltacl=0.5]
{$x$/1, $f'(x)$/1, $f(x)$/2.5}
{$-\infty$, $-1$, $1$, $2$, $+\infty$}
\tkzTabLine{, -, z, -, d, +, z, -, }
\draw (N12) node[below] (A) {$+\infty$} ($ (N22)!0.5!(N23)$ ) node (B) {$-2$} (N33)
node[above left] (C) {$-4$};
\draw[-stealth] (A)--(B);
\draw[-stealth] (B)--(C);
\draw[double] ([yshift=-1mm]N32)--(N33);
\draw (N33) node[above right] (D) {$-\infty$} ($ (N42)!0.55!(N43)$ )
node (E) {$-3$} (N53) node[above] (F) {$-\infty$};
\draw[-stealth] (D)--(E);
\draw[-stealth] (E)--(F);
\end{tikzpicture}

```

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

## II.3 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[smooth,samples...] plot[domain=xmin:xmax] (\x ,{f(x)});
```

trong đó **xmin** và **xmax** là các hoành độ xuất phát và kết thúc khi vẽ đồ thị

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

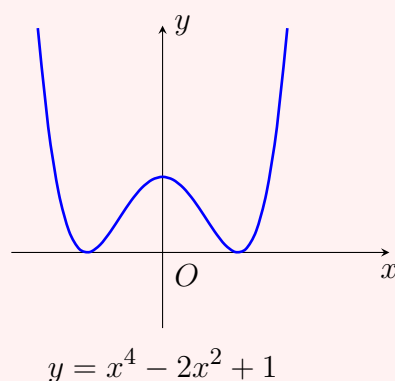
- **domain= a:b** Tham số này quy định cho ta vùng vẽ đồ thị. Với giá trị cụ thể của  $a$  và  $b$  thì **Tikz** vẽ cho ta phần đồ thị của hàm số với  $a \leq x \leq b$ . Việc xác định  $a, b$  cho phù hợp hình vẽ là điều rất cần thiết để hình vẽ không quá lớn hoặc tràn ra ngoài vùng cần vẽ.
- **smooth**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

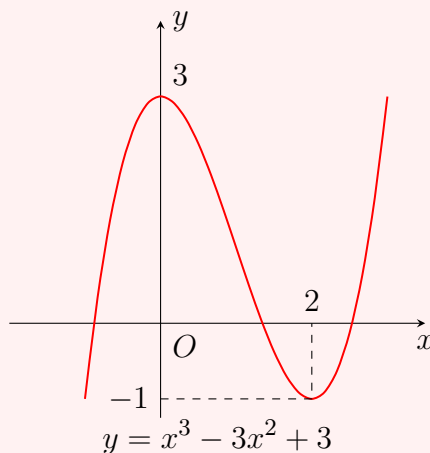
Ta hãy bắt đầu vẽ một vài đồ thị hàm số quen thuộc. Một đoạn lệnh đơn giản như dưới đây đã vẽ được đồ thị hàm số  $y = x^4 - 2x^2 + 1$ :

```
\begin{tikzpicture}[>=stealth]
\draw[>-] (-2,0) --(0,0) node[below
right]{$0$} -- (3,0)
node[below]{$x$};
\draw[>-] (0,-1)--(0,3)
node[right]{$y$};
\draw[smooth,blue,line width=1]
plot[domain=-1.65:1.65]
(\x,{(\x)^(4)-2*(\x)^(2)+1});
\draw (0,-1.5) node
{$y=x^4-2x^2+1$};
\end{tikzpicture}
```



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]{$x$};
\draw[>->] (0,-1.25)--(0,4)
node[right]{$y$};
\draw[smooth,red,line width=0.75]
plot[domain=-1:3]
(\x,{(\x)^3-3*(\x)^2+3});
\draw[dashed] (0,-1)
node[left]{$-1$} -- (2,-1) --
(2,0) node[above]{$2$};
\draw (0,3) node[above right]{$3$};
\draw (0.75,-1.5) node
{$y=x^3-3x^2+3$};
\end{tikzpicture}
```



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

## 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:

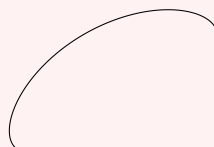
```
\draw[smooth] plot[domain=0:6.3] (canvas polar cs:angle=\x r,radius= {
r=f(x)});
```

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

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

Khi đó ta sẽ được:

```
\begin{tikzpicture}[>=stealth,scale=5]
\draw[domain=0:pi/3,samples=200,smooth]
plot (canvas polar cs:angle=\x
r,radius= {10*cos(3*\x r)});
\end{tikzpicture}
```



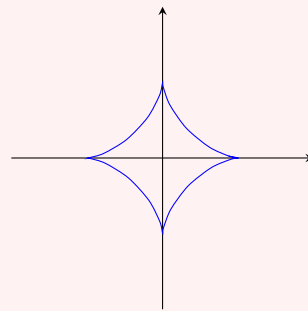
Ngoài ra, ta cũng có thể vẽ đồ thị hàm số với hệ tọa độ đề các và phương trình tham số  $\begin{cases} x = u(t) \\ y = v(t) \end{cases}$  bởi lệnh vẽ đồ thị như sau:

```
\draw [smooth,variable=\t] plot[domain=mint:maxt] ({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,smooth,variable=\t]
  plot[domain=0:120*pi ]
    ({(cos(\t))^3},{(sin(\t))^3});
\end{tikzpicture}
```



### III Tô miền đồ thị hàm số với Tikz

Qua quá trình trao đổi cùng các bạn vẽ hình trong và ngoài nhóm **TikZ - Asymptote** tôi thấy khá nhiều vướng mắc khi các bạn vẽ phần tô miền của đồ thị hàm số trong các bài toán tính diện tích hình phẳng và thể tích khối tròn xoay. Chính vì vậy tôi viết tạm bài viết ngắn này mong các bạn dễ dàng tháo gỡ khó khăn khi vẽ hình.

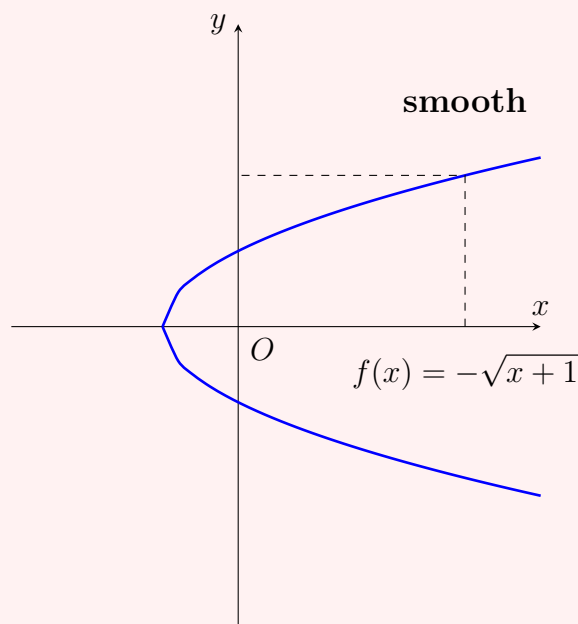
#### III.1 Vẽ đồ thị hàm số

Thông thường khi vẽ đồ thị hàm số, các bạn vẫn dùng lệnh:

```
\draw [samples=100,smooth] plot[domain=0:120*pi] (\x,{\f(\x)});
```

Trong lệnh trên đây thì **smooth** quyết định độ trơn (nhẵn) của đồ thị hàm số, còn **samples** quyết định số điểm chia để vẽ. Do đó việc sử dụng tham số nào cũng nên lưu ý. Thông thường những hàm đa thức thường gặp ta chỉ cần dùng **smooth** là đủ, một số hàm đặc biệt có độ gấp khúc cao thì ta cần sử dụng **samples** và tăng giá trị này để tạo độ cong hợp lý khi zoom hình. Ví dụ sau đây cho các bạn thấy sự khác biệt ở hai hình để giúp bạn lựa chọn cho phù hợp, bởi giá trị **samples** càng cao thì tốc độ biên dịch sẽ tăng thêm đáng kể:

```
\begin{tikzpicture}[>=stealth]
\def\f(#1){sqrt(#1+1)}
\draw[->] (-3,0)--(0,0)node[below
right]{$0$}--(4,0)
node[above]{$x$};
\draw[->]
(0,-4)--(0,4)node[left]{$y$};
\draw[smooth,blue, line width=1]
plot[domain=-1:4] (\x,{\f(\x)})
plot[domain=-1:4] (\x,{-\f(\x)});
\draw (3,-1)
node[above]{$f(x)=-\sqrt{x+1}$};
\draw[dashed] (3,0)--(3,2)--(0,2);
\draw (3,3) node{\bf smooth};
\end{tikzpicture}
```

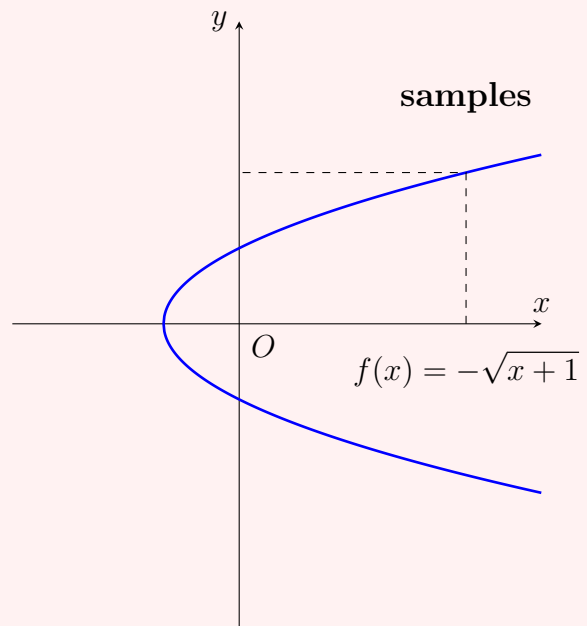




```

\begin{tikzpicture}[>=stealth]
\def\f(#1){sqrt(#1+1)}
\draw[>-] (-3,0)--(0,0)node[below
right]{$0$}--(4,0)
node[above]{$x$};
\draw[>-]
(0,-4)--(0,4)node[left]{$y$};
\draw[samples=500,blue, line
width=1]
plot[domain=-1:4] (\x,{f(\x)})
plot[domain=-1:4] (\x,{-f(\x)});
\draw (3,-1)
node[above]{$f(x)=-\sqrt{x+1}$};
\draw[dashed] (3,0)--(3,2)--(0,2);
\draw (3,3)node{\bf samples};
\end{tikzpicture}

```



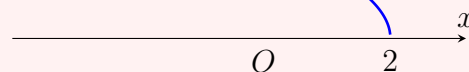
Một điều khác nữa tôi lưu ý các bạn, thông thường các bạn hay để **domain** ở ngay tham số sau lệnh `\draw`. Điều này gây khó khăn khi ta vẽ hai đồ thị liên tiếp để tô miền (mà tôi sẽ nói ở phần sau). Thông thường ta nên để **domain** ở sau `plot[domain=]` thì sẽ chủ động hơn trong việc quyết định khoảng vẽ đồ thị giống như hai ví dụ trên.

Qua kinh nghiệm vẽ đồ thị hàm số. Việc lựa chọn **domain** cũng cần sự tinh tế nhất định. Chẳng hạn với hàm số  $y = \sqrt{2-x}$ . Bạn thông thường vẽ đồ thị hàm số với **domain**  $=-2:2$  tức là điểm xuất phát là bên trái, điểm kết thúc là bên phải. Nhưng bạn hãy thử xem hình dưới đây để thấy sự lựa chọn **domain** cần sự tinh tế như thế nào:

```

\begin{tikzpicture}[>=stealth]
\draw[>-] (-3,0)--(0,0)node[below
right]{$0$}--(3,0)
node[above]{$x$};
\draw[samples=500,blue, line
width=1]
plot[domain=-2:2] (\x,{sqrt(2-\x)});
\draw (2,0)node[below]{$2$};
\end{tikzpicture}

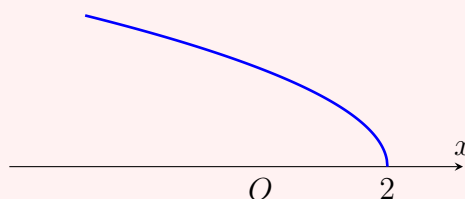
```



Nếu zoom lên, bạn sẽ thấy tại điểm  $x = 2$  đồ thị bị cách trục hoành một khoảng nhỏ mặc dù trên phương diện lý thuyết thì hàm số vẫn xác định tại  $x = 2$ . Lý do chính là cách vẽ của **Tikz** chia thành các khoảng nhỏ để vẽ các đoạn thẳng thông qua tham số **samples** nên ở một lân cận nào đó của điểm  $x = 2$  sẽ kết thúc việc vẽ dẫn đến bị đứt đoạn.

Để giải quyết vấn đề này bạn chỉ cần đổi chiều **domain**. Tại sao lại vậy? Vì khi đổi chiều **domain** thì **Tikz** sẽ vẽ từ điểm bắt đầu là  $x = 2$  nên đồ thị sẽ liền nét từ đó, phần cách đoạn là phần cuối cùng tại một lân cận của  $x = -2$  nên bạn sẽ hầu như thấy không thay đổi tại  $x = -2$  (vì không vẽ tiếp đồ thị nữa). Bạn hãy so sánh code dưới đây và code phía trên cùng hai hình vẽ để có kinh nghiệm cho bản thân:

```
\begin{tikzpicture}[>=stealth]
\draw[>] (-3,0)--(0,0)node[below
right]{$0$}--(3,0)node[above]{$x$};
\draw[samples=500,blue, line
width=1
plot[domain=2:-2](\x,{sqrt(2-\x)})];
\draw (2,0)node[below]{$2$};
\end{tikzpicture}
```



### III.2 Tô miền đồ thị hàm số

Thông thường trong các bài toán, ta có một số dạng tô miền đồ thị như sau:

- Tô miền đồ thị hàm số  $y = f(x)$ ,  $x = a$ ,  $x = b$  và trục hoành.
- Tô miền đồ thị hàm số giao bởi hai đồ thị hàm số  $y = f(x)$  và  $y = g(x)$

Với cả hai dạng trên, ta quy về việc có sẵn các hoành độ để tô miền hay không. Với mục đích code càng ngắn gọn, đơn giản và đặc biệt là độ chính xác càng cao thì càng tốt. Mỗi trường hợp ta có cách giải quyết cụ thể khác nhau.

#### III.2.1 Khi đã biết hoành độ các điểm cận của miền cần tô

Khi đã biết hoành độ các điểm cận của miền cần tô, ta dùng lệnh ngắn gọn như sau:

Chẳng hạn cần tô miền giới hạn bởi đồ thị hàm số  $y = f(x)$ ,  $y = g(x)$ ,  $x = a$  và  $x = b$ , khi đó ta sử dụng lệnh:

```
\draw[smooth] plot[domain=a:b](\x,{f(x)})--plot[domain=b:a](\x,{g(x)})--cycle;
```

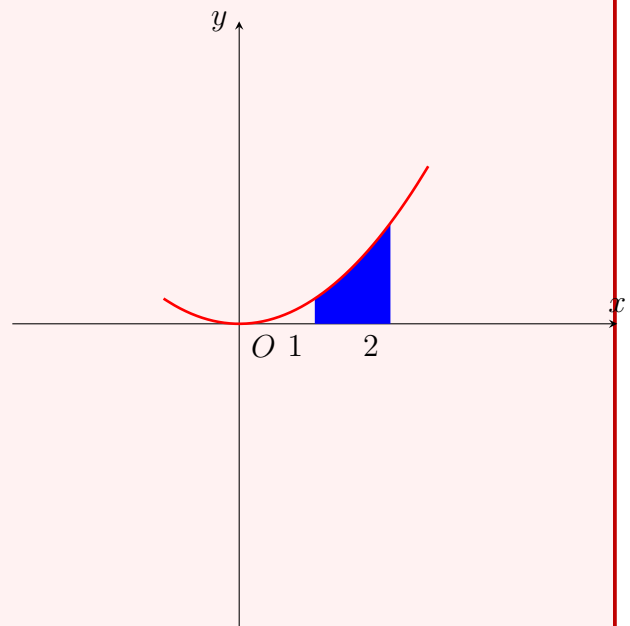
Ở đây ta để ý đầu bút vẽ. Hiểu đơn giản sẽ là bút vẽ đồ thị hàm số  $f(x)$  từ  $x = a$  đến điểm  $x = b$  và sau đó vẽ tiếp đồ thị hàm số  $g(x)$  từ điểm  $x = b$  đến  $x = a$ . Và như vậy bút vẽ sẽ đi một miền kín để tô màu ưng ý.

Ví dụ cụ thể, chẳng hạn tô miền đồ thị hàm số  $y = \frac{x^3}{3}$ ,  $x = 1$ ,  $x = 2$  và trục hoành:

```

\begin{tikzpicture}[>=stealth]
\def\f(#1){((#1)^2)/3}
\draw[>-] (-3,0)--(0,0) node[below
right]{$0$}--(5,0)
node[above]{$x$};
\draw[>-] (0,-4)--(0,4)
node[left]{$y$};
\fill[blue,smooth] (1,0) --
plot[domain=1:2] (\x,{\f(\x)})
-- (2,0) -- cycle;
\draw[smooth,red, line width=1]
plot[domain=-1:2.5]
(\x,{\f(\x)});
\draw (1,0) node[below left]{$1$}
(2,0) node[below left]{$2$};
\end{tikzpicture}

```



Như vậy việc còn lại bạn chỉ cần vẽ hai đường thẳng  $x = 1$  và  $x = 2$  nữa là việc tô miền đã hoàn thành.

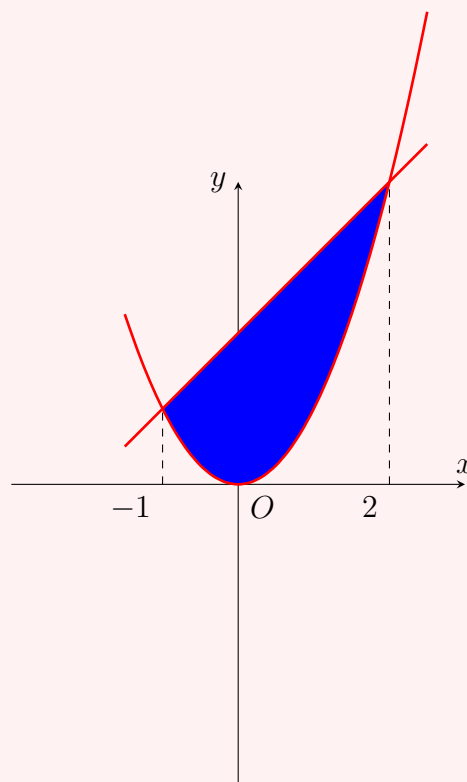
Một ví dụ khác là tô miền phần giao hai đồ thị mà ta có thể dễ dàng tìm thấy các hoành độ giao điểm bằng cách giải phương trình  $f(x) = g(x)$ . Chẳng hạn tô miền phần giới hạn bởi hai đồ thị hàm số  $y = x + 2$  và  $y = x^2$

Ta dễ dàng tìm được hoành độ giao điểm của hai đồ thị này là nghiệm phương trình  $x^2 = x + 2 \Rightarrow x = -1$  hoặc  $x = 2$ . Khi đó ta thực hiện vẽ đồ thị  $y = x^2$  từ  $x = -1$  đến  $x = 2$  rồi tiếp tục vẽ  $y = x + 2$  từ  $x = 2$  đến  $x = -1$ :

```

\begin{tikzpicture}[>=stealth]
\draw[->] (-3,0)--(0,0) node[below
right]{$0$} -- (3,0)
node[above]{$x$};
\draw[->] (0,-4)--(0,4)
node[left]{$y$};
\fill[blue,smooth]
plot[domain=-1:2] (\x,{(\x)^2})
-- plot[domain=2:-1] (\x,{\x+2})
-- cycle;
\draw[smooth,red, line width=1]
plot[domain=-1.5:2.5]
(\x,{(\x)^2});
\draw[smooth,red, line width=1]
plot[domain=-1.5:2.5] (\x,{\x
+2});
\draw[dashed] (-1,0)--(-1,1)
(2,0)--(2,4);
\draw (-1,0) node[below left]{$-1$}
(2,0) node[below left]{$2$};
\end{tikzpicture}

```



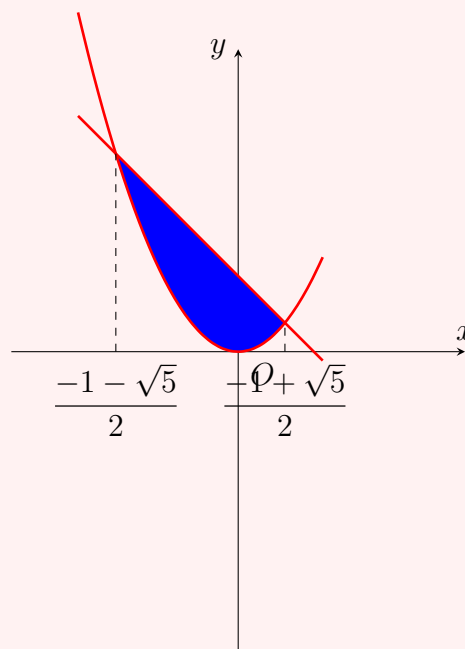
Chú ý rằng các bài toán có thể tìm hoành độ giao điểm nhưng hoành độ đó là số vô tỉ hoặc các số thập phân vô hạn. Ta không nên lấy các giá trị xấp xỉ của nó mà hãy gán cho giá trị đó vào một biến. Khi đó ta hoàn toàn có thể dùng biến đó đưa vào **domain** để chính xác đồ thị.

Chẳng hạn như hoành độ giao điểm của hai đồ thị  $f(x) = x^2$  và  $g(x) = -x + 1$ . Khi giải phương trình  $f(x) = g(x)$  ta có hai nghiệm lần lượt là  $x_1 = \frac{-1 - \sqrt{5}}{2}$  và  $x_2 = \frac{-1 + \sqrt{5}}{2}$ . Nếu bạn bấm máy tính để lấy xấp xỉ các con số để đưa vào **domain** dẫn đến đồ thị không hoàn toàn chính xác và đồng thời việc nhớ các con số dài ngoằng để nhập vào cũng làm code của bạn bớt đi tính thẩm mỹ. Giải pháp là bạn sẽ gán hai biến  $a = x_1$  và  $b = x_2$  sau đó đưa  $a, b$  vào **domain**

```

\begin{tikzpicture}[>=stealth]
\pgfmathsetmacro{\a}{(-1-sqrt(5))/2}
\pgfmathsetmacro{\b}{(-1+sqrt(5))/2}
\draw[>-] (-3,0)--(0,0) node[below
right]{$0$} -- (3,0)
node[above]{$x$};
\draw[>-] (0,-4)--(0,4)
node[left]{$y$};
\fill[blue,smooth]
plot[domain=\a:\b](\x,{(\x)^2})
-- plot[domain=\b:\a]
(\x,{-\x+1}) -- cycle;
\draw[smooth,red, line width=1]
plot[domain=\a-0.5:\b+0.5]
(\x,{(\x)^2});
\draw[smooth,red, line width=1]
plot[domain=\a-0.5:\b+0.5]
(\x,{-\x+1});
\draw[dashed] (\a,0) -- (\a,-\a+1)
(\b,0) -- (\b,-\b+1);
\draw (\a,0) node[below]
{$\dfrac{-1-\sqrt{5}}{2}$}
(\b,0) node[below]
{$\dfrac{-1+\sqrt{5}}{2}$};
\end{tikzpicture}

```



Nếu cả hai hàm số đều có công thức phức tạp mà không phải là bậc nhất như trên, bạn hãy gán thêm hai biến  $fa$  và  $fb$  là các tung độ giao điểm tương ứng với lệnh **pgfmathsetmacro** như khi gán biến  $a$  và  $b$  rồi đưa các tung độ tương ứng vào điểm cần vẽ.

### III.2.2 Tô miền đồ thị không biết hoành độ giao điểm

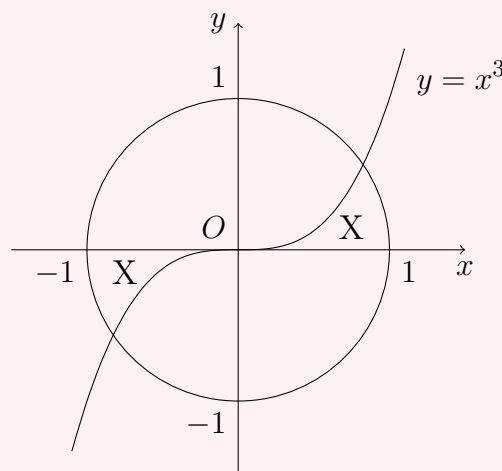
Một vấn đề đặt ra là nhiều khi ta có hàm số  $f(x)$ , hàm số  $g(x)$  nhưng việc giải phương trình  $f(x) = g(x)$  gặp khó khăn. Khi đó đến cả công thức nghiệm ta cũng chẳng có thì làm thế nào? Bạn hãy dùng môi trường **scope** với lệnh **clip**

Một ví dụ đơn giản là tô miền phần giới hạn của đồ thị hàm số  $f(x) = x^3$  và đường tròn có tâm là gốc tọa độ, bán kính bằng 1 được đánh dấu **X** như hình vẽ sau:

```

\begin{tikzpicture}[x=2cm,y=2cm]
\draw[>-] (-1.5,0)--(1.5,0)
  node[below]{$x$};
\draw[>-] (0,-1.5) --(0,1.5)
  node[left]{$y$};
\draw plot[domain=-1.1:1.1,smooth]
  (\x,{(\x)^3});
node[below right]{$y=x^3$};
\draw (0,0) circle(1);
\path
(0,0) node[above left]{$0$}
(1,0) node[below right]{$1$}
(-1,0) node[below left]{$-1$}
(0,1) node[above left]{$1$}
(0,-1) node[below left]{$-1$};
\draw (0.75,0.15) node{X}
  (-0.75,-0.15)node{X};
\end{tikzpicture}

```



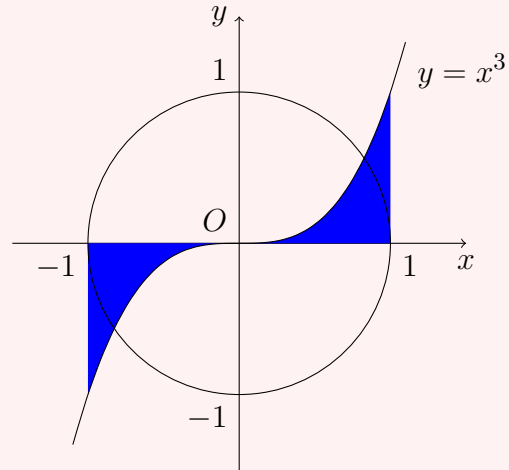
Bình thường bạn sẽ nghĩ đến việc dùng phương trình đường tròn  $x^2 + y^2 = 1$  để suy ra hàm số  $y = \sqrt{1 - x^2}$ . Tuy nhiên việc giải phương trình  $x^3 = \sqrt{1 - x^2}$  đã quá sức bạn. Hoặc có thể bạn giải được nhưng mất rất nhiều thời gian.

Bạn hãy phân tích phần được tô miền sẽ nằm trong đường tròn và nằm ở phần giới hạn bởi đồ thị  $f(x) = x^3$  với trục hoành và đường thẳng  $x = 1$ ,  $x = -1$ . Khi đó bạn hãy tô toàn bộ phần giới hạn bởi đồ thị hàm số  $f(x) = x^3$  với các đường  $x = 1$ ,  $x = -1$  và trục hoành. Sau đó cắt đi chỉ lấy phần nằm bên trong đường tròn:

```

\begin{tikzpicture}[x=2cm,y=2cm]
\draw[>-] (-1.5,0)--(1.5,0)
  node[below]{$x$};
\draw[>-] (0,-1.5) --(0,1.5)
  node[left]{$y$};
\fill[blue,smooth] (-1,0) --
  plot[domain=-1:1] (\x,{(\x)^3})
  -- (1,0) -- cycle;
\draw plot[domain=-1.1:1.1,smooth]
  (\x,{(\x)^3}) node[below right]
  {$y=x^3$};
\draw (0,0) circle(1);
\path
(0,0) node[above left]{$O$}
(1,0) node[below right]{$1$}
(-1,0) node[below left]{$-1$}
(0,1) node[above left]{$1$}
(0,-1) node[below left]{$-1$};
\end{tikzpicture}

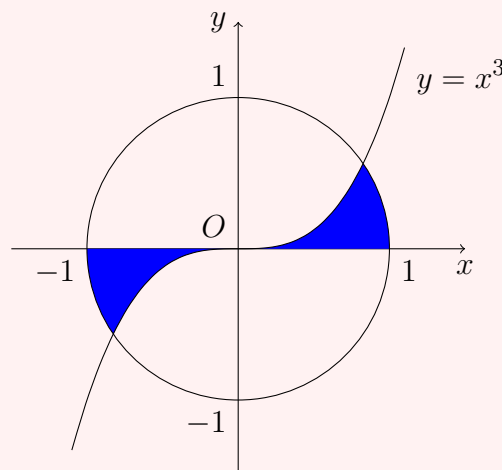
```



```

\begin{tikzpicture}[x=2cm,y=2cm]
\draw[>-] (-1.5,0)--(1.5,0)
  node[below]{$x$};
\draw[>-] (0,-1.5) --(0,1.5)
  node[left]{$y$};
\begin{scope}
\clip (0,0) circle (1);
\fill[blue,smooth] (-1,0) --
  plot[domain=-1:1] (\x,{(\x)^3})
  -- (1,0) -- cycle;
\end{scope}
\draw plot[domain=-1.1:1.1,smooth]
  (\x,{(\x)^3}) node[below right]
  {$y=x^3$};
\draw (0,0) circle(1);
\path
(0,0) node[above left]{$O$}
(1,0) node[below right]{$1$}
(-1,0) node[below left]{$-1$}
(0,1) node[above left]{$1$}
(0,-1) node[below left]{$-1$};
\end{tikzpicture}

```



Và bây giờ bạn đã thấy thành quả. Bạn chỉ việc xem hai code trên bạn đã hiểu việc dùng **scope** và **clip** hiệu quả như thế nào.

Bạn hãy thử thực hành với các hình vẽ khác xem sao.

### III.2.3 Sử dụng even odd rule

Một cách khác để tô một miền kín là sử dụng tham số **even odd rule**. Hiểu đơn giản **even** là chẵn, **odd** là lẻ. Và tham số **even odd rule** chính là lựa chọn miền tô không giao nhau của các miền kín.

Ví dụ ta có ba miền kín coi như ba tập hợp  $A$ ,  $B$  và  $C$ . Khi biểu diễn ba tập hợp này sẽ có các miền giao nhau và không giao nhau. Miền nào thuộc số chẵn trong số ba tập hợp  $A, B, C$  thì sẽ không tô và chỉ tô miền lẻ.

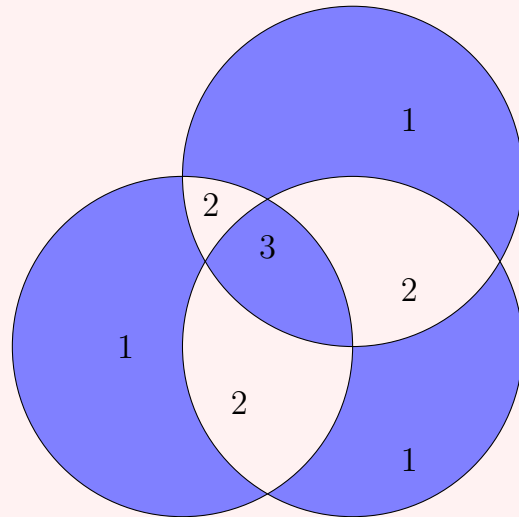
Các bạn hãy xem hình dưới đây để hiểu thế nào là miền **chẵn** (even) và **lẻ** (odd):



```

\begin{tikzpicture}[scale=0.75]
\fill[blue!50, even odd rule] (0,0)
  circle (3) (3,0) circle (3)
  (3,3) circle (3);
\draw (0,0) circle (3);
\draw (3,0) circle (3);
\draw (3,3) circle (3);
\draw (1,-1)node{2} (4,-2)node{1}
  (4,1)node{2} (4,4)node{1}
  (1.5,1.75)node{3}
  (0.5,2.5)node{2} (-1,0)node{1};
\end{tikzpicture}

```



Nhìn hình vẽ trên, các miền là giao của số lẻ miền sẽ được tô, các miền là giao của số chẵn miền không được tô. Tác động này là do sử dụng lệnh:

```

\fill[blue!50,even odd rule](0,0) circle (3)
(3,0) circle (3) (3,3) circle (3);

```

Mặc định của tikz (khi không sử dụng even odd rule) là tô kín mọi miền kể cả số chẵn và số lẻ. Khi đó ta sử dụng khéo léo **even odd rule** sẽ tô được miền ta ưng ý. Như vậy, khi tô các miền lẻ thì ta sử dụng **even odd rule**. Vậy muốn tô miền chẵn thì làm thế nào? Ta hãy đánh lừa **Tikz** bằng cách thêm một miền bao bên ngoài tất cả các miền trên, khi đó các miền đang lẻ sẽ thành chẵn và ngược lại. Hoặc là bạn có thể sử dụng một cách khác, tức là bạn tô sẵn nền, sau đó tô các miền lẻ bằng **white**. Vậy là các miền chẵn sẽ được tô. Chú ý rằng muốn dùng thủ thuật đánh tráo này, bạn sẽ phải clip hình bởi nếu không sẽ có những miền lẻ mới và miền chẵn mới như hình dưới đây tôi đã thêm một miền mới bằng lệnh:

```

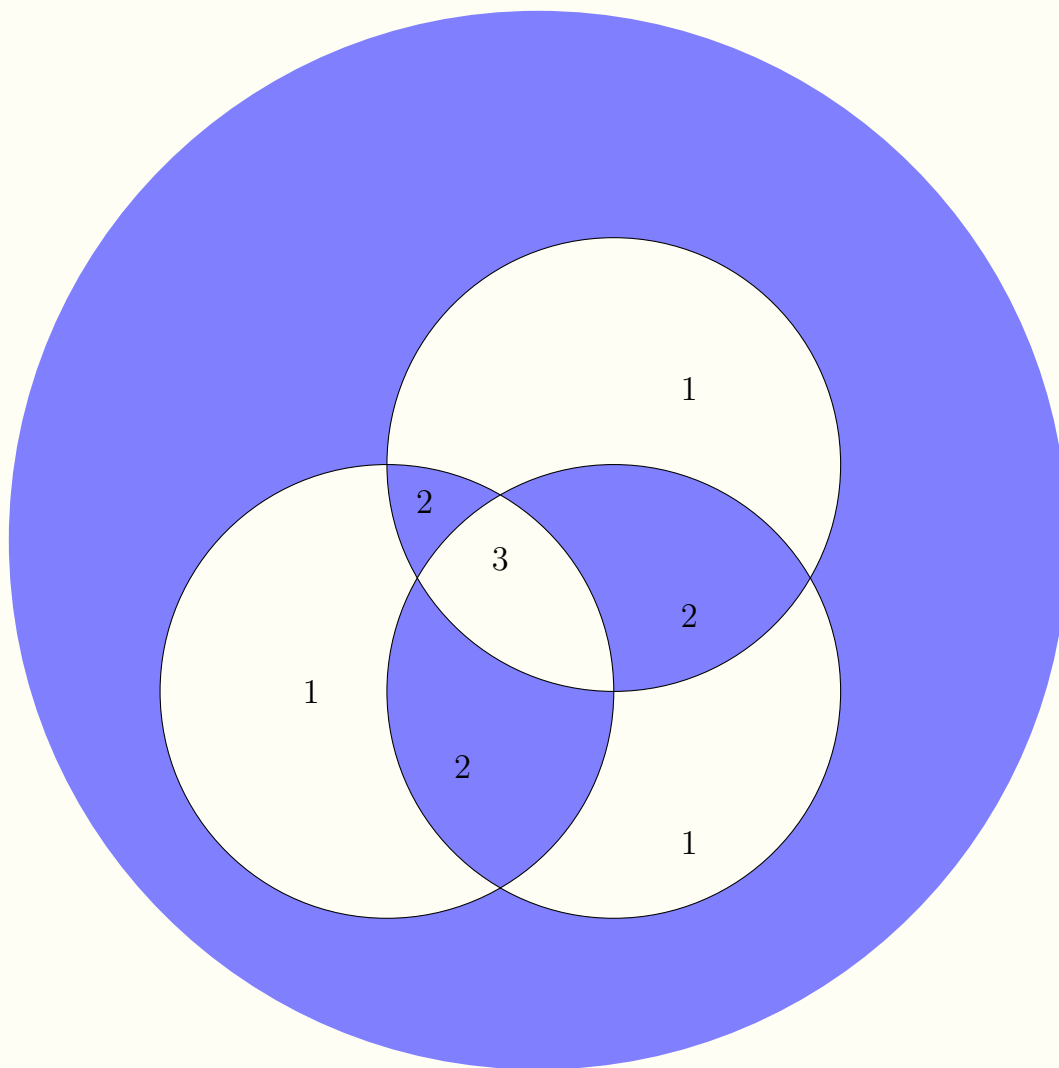
\fill[blue!50, even odd rule] (0,0) circle (3) (3,0) circle (3)
(3,3) circle (3) (2,2) circle (7);

```

```

\begin{tikzpicture}
\fill[blue!50, even odd rule] (0,0) circle (3) (3,0) circle (3) (3,3) circle
  (3) (2,2) circle (7);
\draw (0,0) circle (3);
\draw (3,0) circle (3);
\draw (3,3) circle (3);
\draw (1,-1)node{2} (4,-2)node{1} (4,1)node{2} (4,4)node{1} (1.5,1.75)node{3}
  (0.5,2.5)node{2} (-1,0)node{1};
\end{tikzpicture}

```



### III.3 Các kiểu tô miền

Bạn thích tô màu các miền thì thực hiện các lệnh **fill** như trên đây tôi trình bày, chỉ việc đổi tên màu là bạn đã đạt được ý nguyện. Nếu bạn không muốn tô miền bằng màu sắc bạn cũng có các kiểu tô miền khác mà **Tikz** đã thiết kế sẵn gọi là **pattern**.

Để sử dụng **pattern**, trong file biên dịch bạn cần gọi thư viện này ra bằng

```

\usetikzlibrary{patterns} trước \begin{document}

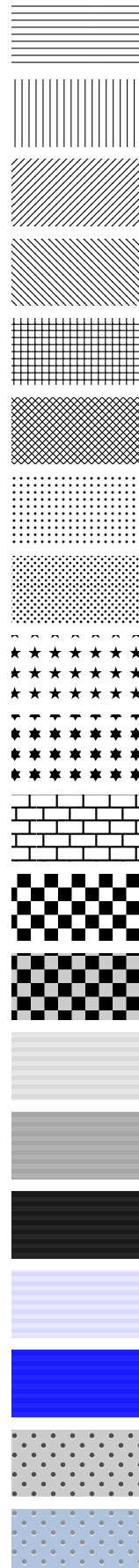
```

Tại lệnh **fill** bạn sử dụng như sau:

`\fill[pattern=name pattern]`.

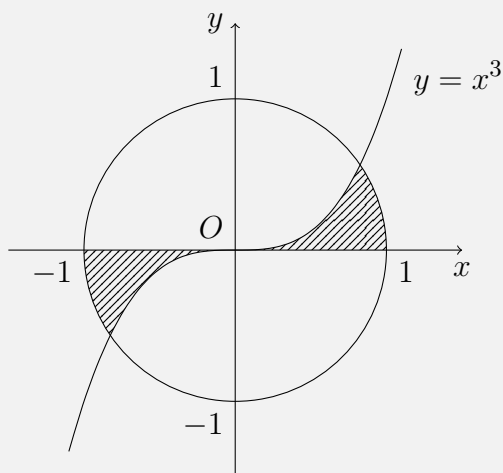
Một số lựa chọn kiểu tô:

- 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



Chẳng hạn, ở ví dụ trên tôi sẽ dùng kiểu gạch sọc:

```
\begin{tikzpicture}[x=2cm,y=2cm]
\draw[>-] (-1.5,0) -- (1.5,0) node[below]{$x$};
\draw[>-] (0,-1.5) -- (0,1.5) node[left]{$y$};
\begin{scope}
\clip (0,0) circle (1);
\fill[pattern=north east lines,smooth] (-1,0) -- plot[domain=-1:1]
(\x,{(\x)^3}) -- (1,0) -- cycle;
\end{scope}
\draw plot[domain=-1.1:1.1,smooth] (\x,{(\x)^3}) node[below right]{$y=x^3$};
\draw (0,0) circle(1);
\path
(0,0) node[above left]{$O$}
(1,0) node[below right]{$1$}
(-1,0) node[below left]{$-1$}
(0,1) node[above left]{$1$}
(0,-1) node[below left]{$-1$};
\end{tikzpicture}
```



Vậy là bạn đã hoàn thành được việc vẽ tô miền cho các hàm số trong các bài tích phân rồi đó. Chúc bạn có thể đơn giản hơn một phần việc không hề dễ dàng này trong tương lai!

## IV Đường cong

Như ở phần trên, tại hạ đã trình bày chúng ta có các cách vẽ đường cong từ điểm  $A$  đến điểm  $B$  có nhiều cách. Nếu là một cung tròn hoặc ellipse, parabol,... hoặc các đoạn cong là đường đồ thị của một hàm số đã cho sẵn thì hoàn toàn ta đã có cách vẽ chuẩn xác. Những đường cong mà ta chưa biết được phương trình đường thì ta thường sử dụng **controls** hoặc **plot coordinates** hay **to[in= ,out= ]**. Sau đây tại hạ cùng quý vị hãy tìm hiểu về những cách vẽ này:

### IV.1 Controls

#### IV.1.1 Cú pháp lệnh vẽ

Cú pháp để vẽ đường cong bằng **controls** có hai dạng:

```
\draw (A) .. controls (D) ..(B);
```

hoặc:

```
\draw (A) .. controls (DA) and (DB) ..(B);
```

Để hiểu về cú pháp và cách thực hiện của **Tikz**, tại hạ xin mô tả bằng lời các ý nghĩa của lệnh vẽ trước khi ta thực hành. Trước hết, **controls** có ý nghĩa là "điều khiển". Như vậy trong hai lệnh trên đây quý vị hiểu như sau:

Lệnh thứ nhất: Vẽ đường cong từ điểm  $A$  đến điểm  $B$  với điểm điều khiển là  $D$ .

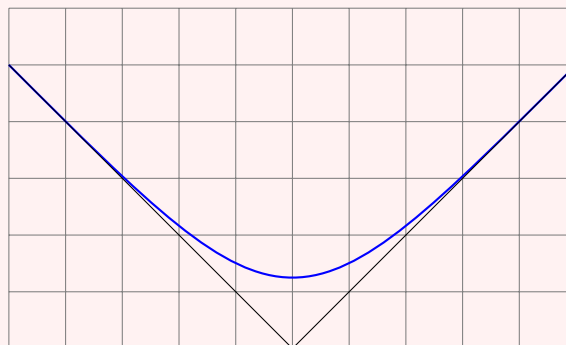
Lệnh thứ hai: Vẽ đường cong từ điểm  $A$  đến điểm  $B$  với hai điểm điều khiển là  $DA$  và  $DB$ .

Quý vị hãy hiểu đơn giản là khi dùng `(A)..controls (D)..(B)` nghĩa là từ điểm  $D$  có thể vẽ được tiếp tuyến với đường cong tại  $A$  và  $B$  hoặc khi dùng

`A..controls (DA) and (DB)..(B)` nghĩa là từ điểm  $DA$  có thể vẽ được tiếp tuyến với đường cong tại  $A$  và từ điểm  $DB$  có thể vẽ tiếp tuyến với đường cong tại  $B$ .

Như vậy, khi hai điểm  $A, B$  là cố định thì việc lựa chọn điểm điều khiển  $D$  hoặc  $DA$  và  $DB$  là tối quan trọng để quyết định hình dạng đường cong. Quý vị hãy phân tích ví dụ sau:

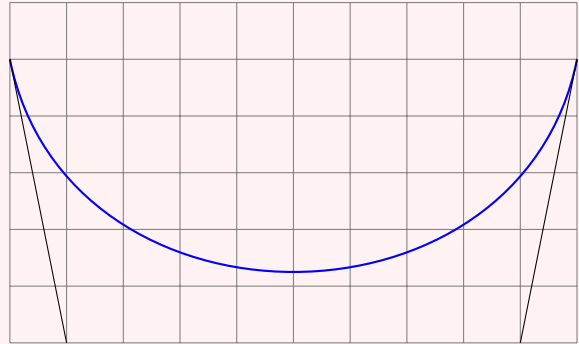
```
\begin{tikzpicture}[scale=0.75]
\draw[step=1,thin,gray] (0,0) grid
(10,6);
\coordinate (A) at (0,5);
\coordinate (B) at (10,5);
\coordinate (D) at (5,0);
\draw[blue,thick] (A)..controls
(D)..(B);
\draw (A)--(D)--(B);
\end{tikzpicture}
```



```

\begin{tikzpicture}[scale=0.75]
\draw[step=1,thin,gray] (0,0) grid
  (10,6);
\coordinate (A) at (0,5);
\coordinate (B) at (10,5);
\coordinate (DA) at (1,0);
\coordinate (DB) at (9,0);
\draw[blue,thick] (A)..controls
  (DA) and (DB) ..(B);
\draw (A)--(DA) (DB)--(B);
\end{tikzpicture}

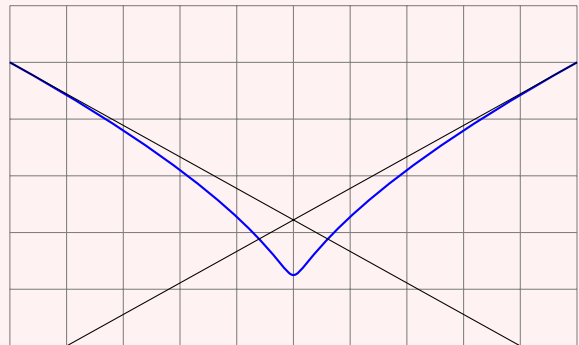
```



```

\begin{tikzpicture}[scale=0.75]
\draw[step=1,thin,gray] (0,0) grid
  (10,6);
\coordinate (A) at (0,5);
\coordinate (B) at (10,5);
\coordinate (DA) at (1,0);
\coordinate (DB) at (9,0);
\draw[blue,thick] (A)..controls
  (DB) and (DA) ..(B);
\draw (A)--(DB) (DA)--(B);
\end{tikzpicture}

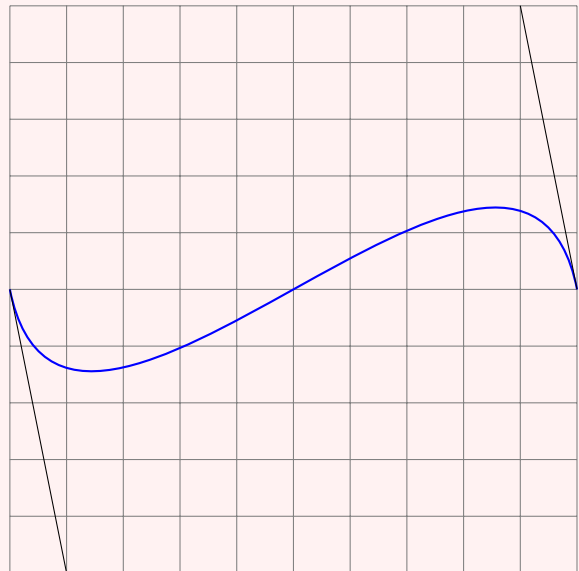
```



```

\begin{tikzpicture}[scale=0.75]
\draw[step=1,thin,gray] (0,0) grid
  (10,10);
\coordinate (A) at (0,5);
\coordinate (B) at (10,5);
\coordinate (DA) at (1,0);
\coordinate (DB) at (9,10);
\draw[blue,thick] (A)..controls
  (DA) and (DB) ..(B);
\draw (A)--(DA) (DB)--(B);
\end{tikzpicture}

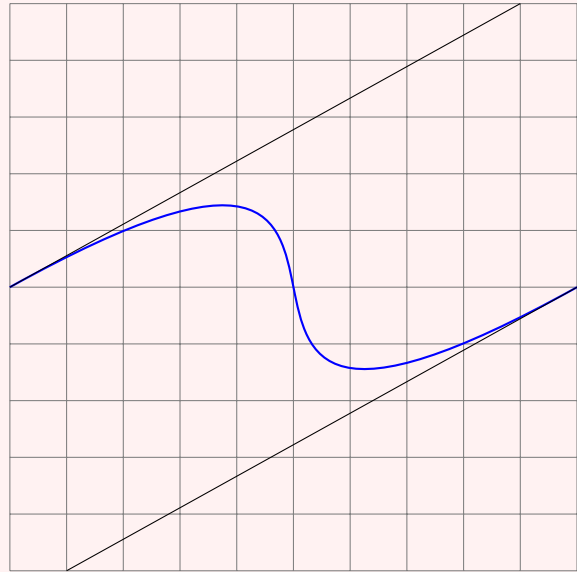
```



```

\begin{tikzpicture}[scale=0.75]
\draw[step=1,thin,gray] (0,0) grid
  (10,10);
\coordinate (A) at (0,5);
\coordinate (B) at (10,5);
\coordinate (DA) at (1,0);
\coordinate (DB) at (9,10);
\draw[blue,thick] (A)..controls
  (DB) and (DA) ..(B);
\draw (A)--(DB) (DA)--(B);
\end{tikzpicture}

```



Quý vị hãy thử dùng code này và thay đổi thông số tọa độ các điểm  $D$ ,  $DA$ ,  $DB$  để thấy sự thay đổi của đường cong, khi đó quý vị sẽ thấy rõ hơn tác động của các điểm điều khiển đến đường cong như thế nào.

#### IV.1.2 Kinh nghiệm vẽ

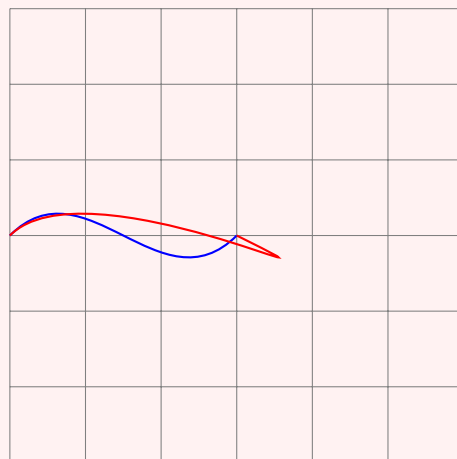
Thông thường, để vẽ một đường cong theo ý cho trước, tại hạ cần chốt điểm đầu và điểm cuối của đường cong. Tiếp theo là việc chọn điểm điều khiển. Khi đã hình dung ra hình dạng đường cong cần vẽ thì đồng thời trong đầu tại hạ dường như ra các điểm vẽ tiếp tuyến đến điểm đầu và điểm cuối (đó chính là điểm điều khiển cần chọn).

Ta hãy phân tích về việc vẽ một đường cong từ điểm  $A(0,0)$  đến điểm  $B(3,0)$  trong các ví dụ sau:

```

\begin{tikzpicture}
\draw[step=1,thin,gray] (0,-3) grid
  (6,3);
\draw[blue,thick] (0,0)..controls
  (1,1) and (2,-1) ..(3,0);
\draw[red,thick] (0,0)..controls
  +(1,1) and +(2,-1) ..(3,0);
\end{tikzpicture}

```



Quý vị sẽ thấy sự khác biệt của hai đường cong xanh và đỏ. Nếu nhìn vào tọa độ các điểm trong code, quý vị sẽ thấy

`controls (1,1) and (2,-1)` và

`controls +(1,1) and +(2,-1)`

trong hai code này khác nhau chỗ dấu `+`. Vậy dấu `+` ở đây mang ý nghĩa như thế

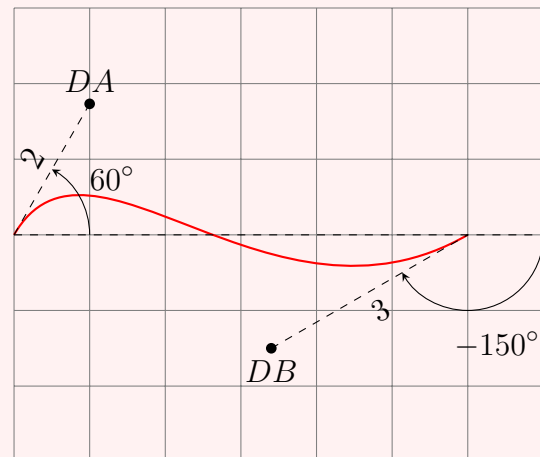
nào?

Ngay bây giờ quý vị sẽ hiểu. Trên đường màu xanh, điểm điều khiển chính là điểm  $(1, 1)$  và  $(2, -1)$ . Còn trên đường màu đỏ, điểm điều khiển không phải là hai điểm này. Mà điểm điều khiển thứ nhất sẽ có tọa độ  $(0, 0) + (1, 1) = (1, 1)$  và điểm điều khiển thứ hai chính là điểm  $(3, 0) + (2, -1) = (5, -1)$ . Như vậy, khi thêm dấu  $+$  trước tọa độ gán cho điểm điều khiển thì điểm điều khiển sẽ thay đổi như phép toán trên.

Chốt lại rằng thì là mà. Nếu ta dùng `(A)..controls +(DA) and +(DB)..(B)` thì hai điểm điều khiển phải là  $\$(A)+(DA)\$$  và  $\$(B)+(DB)\$$ . Để thuận tiện, tại hạ khuyên dùng tọa độ của  $(DA)$  và  $(DB)$  dưới dạng tọa độ cực thì sẽ dễ dàng hơn cho việc lựa chọn điểm điều khiển.

Quý vị hãy xem hình mô tả dưới đây:

```
\begin{tikzpicture}
\draw[step=1,thin,gray] (0,-3) grid
  (7,3);
\draw[red,thick] (0,0)..controls
  +(60:2) and +(-150:3)..(6,0);
\draw[dashed] (0,0)--(60:2)
  node[midway,sloped,above]{2}
  ($(6,0)+(-150:3)$)--(6,0)
  node[midway,sloped,below]{3}
  (0,0)--(7,0);
\draw[-stealth] (1,0) arc (0:60:1);
\draw[-stealth] (7,0) arc(0:-150:1);
\draw (30:1.5) node{$60^\circ$}
  ($(6,0)+(-75:1.5)$)
  node{$-150^\circ$};
\fill[black]
  (60:2)node[above]{$DA$} circle
  (2pt) ($(6,0)+(-150:3)$)
  node[below]{$DB$} circle (2pt);
\end{tikzpicture}
```

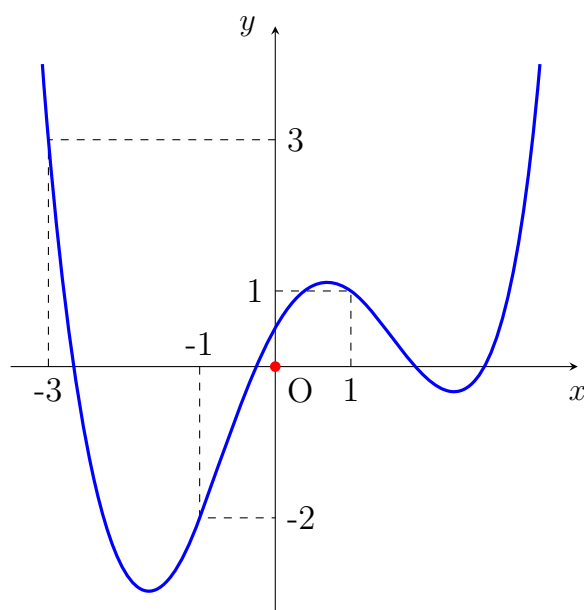


Ở ví dụ này, quý vị đọc kỹ dòng code `controls` để hiểu việc sử dụng tọa độ cực có lợi thế ra sao. Do vậy, trong hầu hết hình vẽ thì tại hạ đều dùng tọa độ cực và phép "cộng" để nhanh gọn và dễ dàng trong việc tinh chỉnh hình vẽ.

#### IV.1.3 Phân tích hình và lựa chọn code vẽ

Ví dụ ta muốn vẽ lại một đồ thị đã có sẵn như sau:

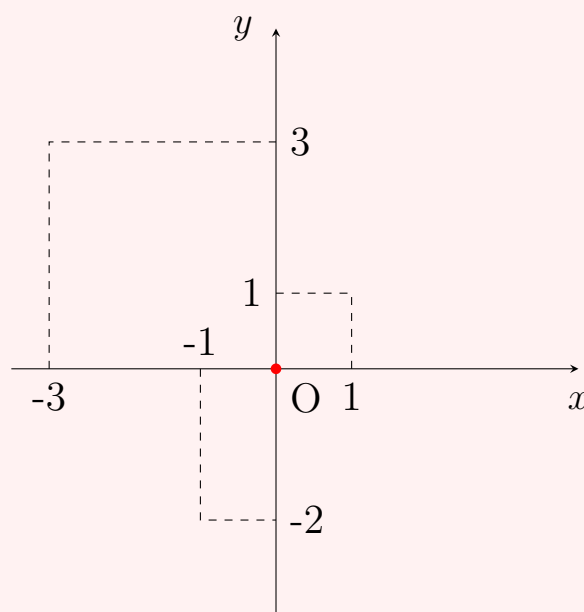




Ta hãy phân tích đường cong trên. Đường cong bắt buộc đi qua các điểm  $A(-3, 3)$ ,  $B(-1, -2)$ ,  $C(1, 1)$ . Hơn nữa ta nhìn hình có thể phán đoán đường cong qua điểm thứ tư  $E(3.5, 4)$  (riêng điểm này thì chỉ cần phán đoán tương đối vì trên hình vẽ gốc không thể hiện). Đó chính là các điểm mấu chốt để vẽ. Ta hãy chia đường cong thành các đoạn cong: từ  $A$  đến  $B$ , từ  $B$  đến  $C$  và từ  $C$  đến  $E$ . Khi đó ta sẽ vẽ dần từng đoạn cho chính xác.

Thêm một ý nữa, do đồ thị còn một đoạn phía bên trái của  $A$  nên ta cần dùng một đoạn đường cong nhỏ thể hiện điều này. Tại hai bắt đầu vẽ hệ trục tọa độ và các điểm mấu chốt:

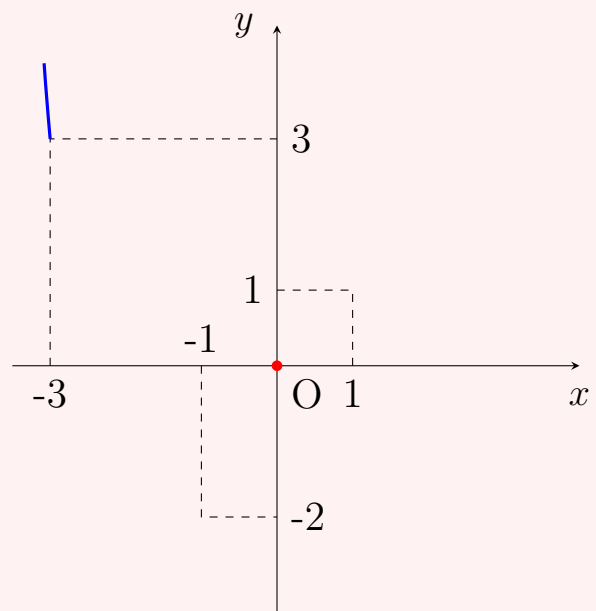
```
\begin{tikzpicture}[every
  node/.style={scale=1.2}]
\draw[-stealth] (-3.5,0)--(4,0)
  node[below=1mm]{$x$};
\draw[-stealth] (0,-3.3)--(0,4.5)
  node[left=1mm]{$y$};
\clip (-3.5,-3.3) rectangle (4,4.5);
\fill[red] (0,0) circle(2pt)
  node[below right,black]{0};
\draw[dashed]
(-3,0) node[below]{$-3$}|-(0,3)
  node[right]{3}
(-1,0) node[above]{$-1$}|-(0,-2)
  node[right]{-2}
(1,0) node[below]{$1$}|-(0,1)
  node[left]{1};
\end{tikzpicture}
```



tiếp theo, ta hãy vẽ từng đoạn để theo dõi thì quý vị sẽ hiểu thêm. Đầu tiên vẽ đoạn

bên trái điểm  $A(-3, -3)$  bằng `to[in= , out= ]`:

```
\begin{tikzpicture}[every
  node/.style={scale=1.2}]
\draw[-stealth] (-3.5,0)--(4,0)
  node[below=1mm]{$x$};
\draw[-stealth] (0,-3.3)--(0,4.5)
  node[left=1mm]{$y$};
\fill[red] (0,0) circle(2pt)
  node[below right,black]{0};
\draw[dashed]
(-3,0) node[below]{$-3$}|-(0,3)
  node[right]{3}
(-1,0) node[above]{$-1$}|-(0,-2)
  node[right]{-2}
(1,0) node[below]{1}|-(0,1)
  node[left]{1};
\draw[blue,very thick]
(-3.08,4) to[out=-86,in=95] (-3,3);
\end{tikzpicture}
```

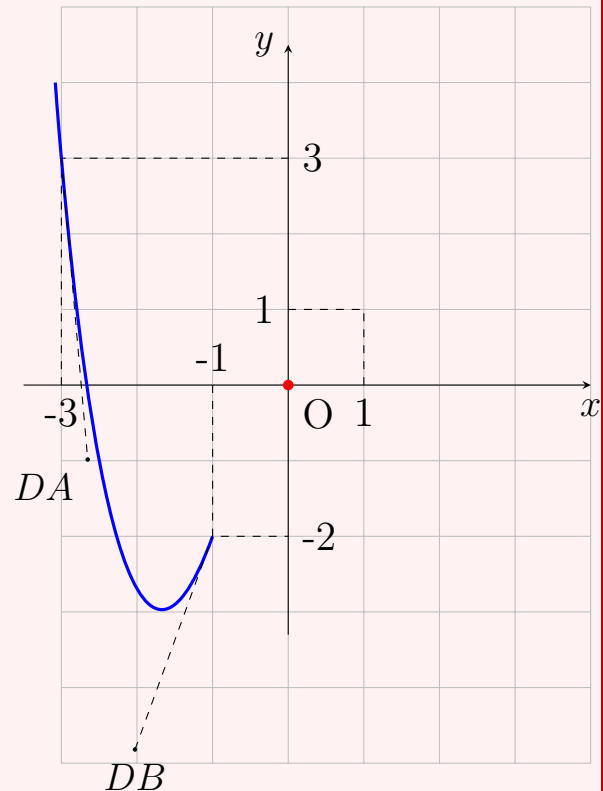


ta bắt đầu vẽ đoạn cong từ  $A$  đến  $B$  bằng controls. Quý vị hãy hình dung ra các điểm điều khiển:

```

\begin{tikzpicture}[every
  node/.style={scale=1.2}]
\draw[step=1,thin,gray!50] (-3,-5)
  grid (4,5);
\draw[-stealth] (-3.5,0)--(4,0)
  node[below]{$x$};
\draw[-stealth] (0,-3.3)--(0,4.5)
  node[left]{$y$};
\fill[red] (0,0) circle(2pt)
  node[below right,black]{0};
\draw[dashed]
(-3,0) node[below]{$-3$}|-(0,3)
  node[right]{3}
(-1,0) node[above]{$-1$}|-(0,-2)
  node[right]{-2}
(1,0) node[below]{1}|-(0,1)
  node[left]{1};
\draw[blue,very thick]
(-3.08,4) to[out=-86,in=95] (-3,3)
..controls +(-85:4) and +(-110:3)..
  (-1,-2);
\draw[dashed]
  (-3,3)--($(-3,3)+(-85:4)$)
  coordinate(DA)
  ($(-1,-2)+(-110:3)$)
  coordinate(DB)--(-1,-2);
\fill[black] (DA)node[below
  left]{$DA$} circle (0.03)
  (DB)node[below]{$DB$} circle
  (0.03);
\end{tikzpicture}

```

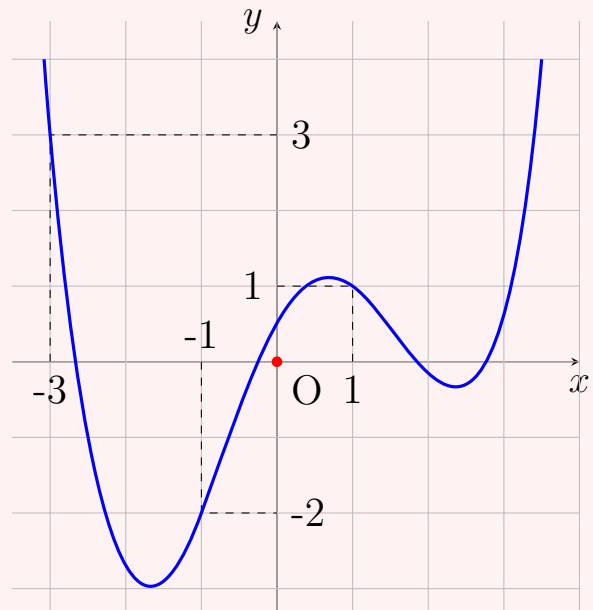


Tương tự như vậy, quý vị có thể vẽ nốt các đoạn đồ thị còn lại cho đủ hình như sau:

```

\begin{tikzpicture}[every
  node/.style={scale=1.2}]
\draw[-stealth] (-3.5,0)--(4,0)
  node[below]{$x$};
\draw[-stealth] (0,-3.3)--(0,4.5)
  node[left]{$y$};
\clip (-3.5,-3.3) rectangle (4,4.5);
\draw[step=1,thin,gray!50] (-4,-4)
  grid (4,5);
\fill[red] (0,0) circle(2pt)
  node[below right,black]{0};
\draw[dashed]
(-3,0) node[below]{$-3$}|-(0,3)
  node[right]{3}
(-1,0) node[above]{$-1$}|-(0,-2)
  node[right]{$-2$}
(1,0) node[below]{1}|-(0,1)
  node[left]{1};
\draw[blue,very thick]
(-3.08,4) to[out=-86,in=95] (-3,3)
..controls +(-85:4) and +(-110:3)..
  (-1,-2)
..controls +(70:2) and +(145:1)..
  (1,1)
..controls +(-35:1) and +(-95:7)..
  (3.5,4);
\end{tikzpicture}

```



Qua việc phân tích ví dụ trên, hi vọng quý vị có thể dùng **controls** để vẽ các đồ thị bất kì mà quý vị chưa biết đồ thị hàm số.

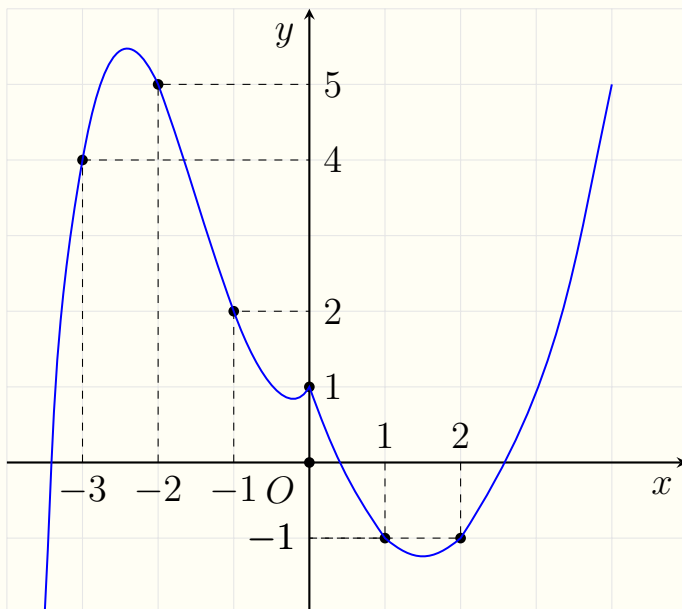
#### IV.1.4 một vài đồ thị vẽ bằng controls

##### Đồ thị số 01

```

\begin{tikzpicture}[every node/.style={scale=1.2}]
\draw[step=1,thin,gray!20] (-4,-2) grid (5,6);
\draw[-stealth,thick] (-4,0)--(5,0) node[below left]{$x$};
\draw[-stealth,thick] (0,-2)--(0,6) node[below left]{$y$};
\foreach \i /\j in {-3/4,-2/5,-1/2,1/-1,2/-1}{
\draw[dashed] (\i,0)|-(0,\j);
\fill[black] (\i,\j) circle (2pt);
}
\foreach \i /\j in {-3/4,-2/5,-1/2}{\draw (\i,0) node[below]{$\i$}
(0,\j)node[right]{$\j$};}
\foreach \i /\j in {1/-1,2/-1}{\draw(\i,0) node[above]{$\i$}
(0,\j)node[left]{$\j$};}
\fill[black] (0,1)node[right]{$1$} circle (2pt)
(0,0) node[below left]{$0$} circle (2pt);
\draw[blue,line width=0.75] (-3.5,-2) to[in=-100,out=87] (-3,4)
..controls +(80:1.5) and +(115:1) ..(-2,5)
to[in=110,out=-70] (-1,2)
..controls +(-70:1) and +(-120:0.5) .. (0,1)
to [in=125,out=-70] (1,-1)
..controls +(-40:0.5) and +(-140:0.5).. (2,-1)
to[in=-102,out=57] (4,5);
\end{tikzpicture}

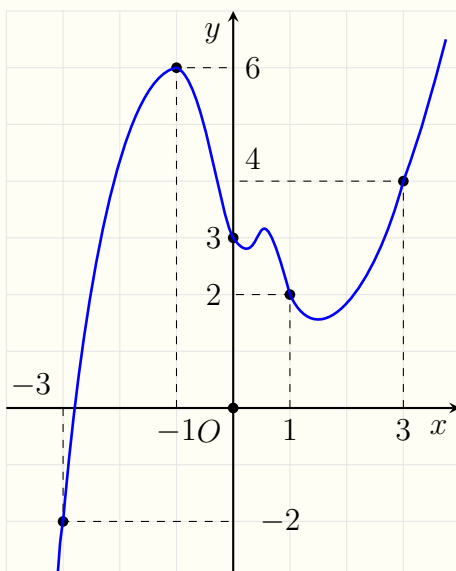
```



```

\begin{tikzpicture}[x=0.75cm,y=0.75cm]
\draw[step=1,thin,gray!20] (-4,-3) grid (4,7);
\clip (-4,-3) rectangle (4.2,7.2);
\draw[-stealth,thick] (-4,0)--(4,0)node[below left]{$x$};
\draw[-stealth,thick] (0,-3)--(0,7) node[below left]{$y$};
\foreach \i/\j in {-3/-2,-1/6,1/2,3/4}{
\draw[dashed] (\i,0)|-(0,\j);
\fill[black] (\i,\j) circle (2pt);}
\draw (-3,0) node[above left]{$-3$}
(-1,0) node[below]{$-1$}
(1,0)node[below]{$1$}
(3,0)node[below]{$3$}
(0,-2)node[right=2mm]{$-2$}
(0,2)node[left]{$2$}
(0,4)node[above right]{$4$}
(0,6)node[right]{$6$};
\fill[black] (0,0)node[below left]{$0$} circle (2pt)
(0,3)node[left]{$3$}circle (2pt);
\draw[blue,line width=1]
(-3.1,-3) to[in=-100,out=85] (-3,-2)
..controls +(85:4) and +(-170:1)..(-1,6)
..controls +(-20:0.5) and +(120:0.5)..(0,3)
..controls +(-55:1) and +(105:2.5)..(1,2)
..controls +(-65:1.25) and +(-105:2) ..(3,4)
to [out=70,in =-105] (3.75,6.5)
;
\end{tikzpicture}

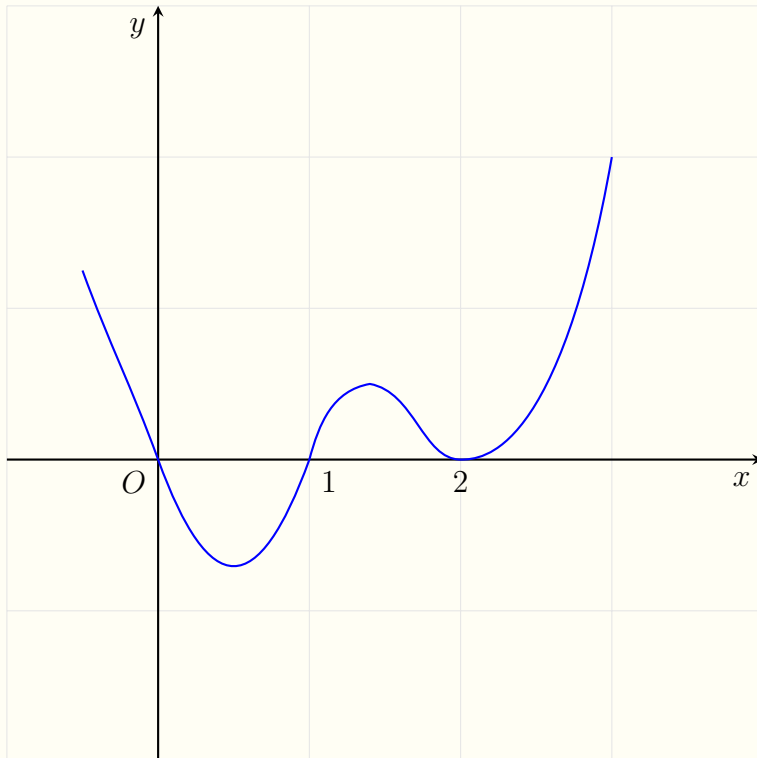
```



```

\begin{tikzpicture}[scale=2]
\draw[step=1,thin,gray!20] (-1,-2) grid (4,3);
\draw[-stealth,thick] (-1,0)--(4,0) node[below left]{$x$};
\draw[-stealth,thick] (0,-2)--(0,3) node[below left]{$y$};
\draw (0,0)node[below left]{$0$}
(1,0)node[below right]{$1$}
(2,0)node[below]{$2$};
\draw[blue,thick] (-0.5,1.25) to[out=-70,in=110] (0,0)
..controls +(-70:1) and +(-110:1)..(1,0)
..controls +(75:0.25) and +(-170:0.25)..(1.4,0.5)
..controls +(-10:0.3) and +(180:0.25)..(2,0)
..controls +(0:0.15) and +(-100:2)..(3,2)
;
\end{tikzpicture}

```



```

\begin{tikzpicture}
\draw[step=1,thin,gray!20] (-1,-2) grid (9,6);
\draw[-stealth,thick] (-1,0)--(9,0) node[above left]{$x$};
\draw[-stealth,thick] (0,-2)--(0,6) node[below left]{$y$};
\foreach \i/\j in {0.5/-1,3/5}{\draw[dashed] (\i,0)|-(0,\j); \fill[black]
    (\i,\j) circle (2pt);}
\draw (-1,0)node[left]{$-1$}
(0,5)node[left]{$5$}
(0.5,0)node[above]{$\frac{1}{2}$}
(5,0)node[below left]{$5$}
(0,0)node[above left]{$0$};
\fill[black] (0,0) circle (2pt) (5,0) circle (2pt) (7,0)circle (2pt);
\clip (-1,-1.5) rectangle (8,6);
\draw[blue,thick] (-0.5,-1.5) ..controls +(20:0.5) and +(-140:0.5)..(0.5,-1)
..controls +(30:1.5) and +(-180:1) ..(3,5)
..controls +(0:1) and +(120:2)..(5,0)
..controls +(-60:3) and +(-160:0.5)..(7,0)
..controls +(0:0.25) and +(100:1)..(7.5,-1.5)
;
\end{tikzpicture}

```

