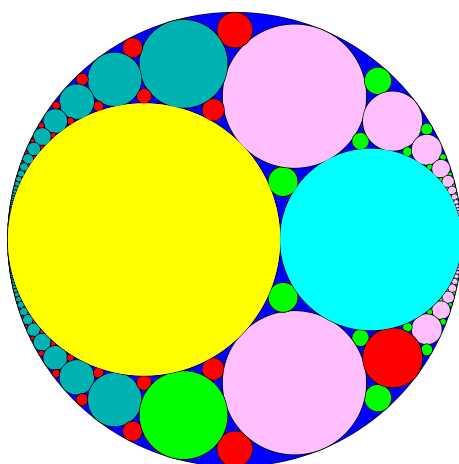


# VẼ HÌNH HỌC PHẪNG BẰNG GÓI GEOMETRY TRONG PHẦN MỀM ASYMPTOTE

Huỳnh Văn Thơ

*Ngày 11 tháng 8 năm 2018*



## Tóm tắt nội dung

Trong soạn thảo văn bản bằng  $\text{\LaTeX}$  có nhiều gói lệnh, phần mềm vẽ hình rồi xuất ra định dạng pdf, eps, jpg, png, ... hoặc mã  $\text{\LaTeX}$  để chèn vào văn bản nhằm đem lại bản in chất lượng nhất. Tôi xin giới thiệu **Vẽ hình phẳng bằng gói geometry trong phần mềm asymptote** dùng để vẽ hình phẳng bằng lệnh và chèn trực tiếp vào tài liệu. Trong tài liệu này, tôi lược dịch lại phần lớn từ tài liệu Euclidean geometry with asymptote.pdf bản tiếng Anh của tác giả Philippe Ivaldi có tại [đây](#) và tham khảo thêm vài trang web. Vì trình độ có hạn nên trong quá trình biên soạn không thể tránh khỏi những sai sót. Mọi ý kiến đóng góp xin liên hệ qua email: [thohv.ngoquyen@gmail.com](mailto:thohv.ngoquyen@gmail.com)

Huỳnh Văn Thơ

## Mục lục

1. Cài đặt – khai báo gói.....	4
1.1. Đối với hệ điều hành windows.....	4
1.2. Đối với hệ điều hành MacOS.....	4
1.3. Khai báo gói.....	4
2. Hệ trục tọa độ $Oxy$ .....	5
3. Màu sắc.....	6
4. Điểm.....	7
4.1. Định nghĩa điểm.....	7
4.2. Vẽ điểm.....	8
4.3. Nhãn – vị trí nhãn.....	9
5. Một số cú pháp cơ bản trong asymptote.....	10
6. Đường thẳng.....	11
6.1. Định nghĩa đường thẳng.....	11
6.1.1. Đường thẳng qua hai điểm.....	11
6.1.2. Vẽ đường thẳng và gắn nhãn.....	11
6.2. Định nghĩa đường thẳng bằng quan hệ song song, vuông góc, ... ..	11
6.3. Đường thẳng trung trực.....	14
7. Đánh dấu góc.....	14
7.1. Đánh dấu góc không vuông.....	14
7.2. Đánh dấu góc vuông.....	15
8. Định nghĩa, vẽ, ký hiệu đoạn thẳng.....	16
8.1. Định nghĩa đoạn thẳng.....	16
8.2. Vẽ – ký hiệu đoạn thẳng.....	16
9. Kiểu đường thẳng, đoạn thẳng, ... ..	17
10. Giao điểm của hai đường thẳng.....	18
11. Kiểu mũi tên.....	18
12. Tam giác.....	19
12.1. Định nghĩa tam giác.....	19
12.2. Vẽ tam giác – nhãn các đỉnh tam giác.....	20
12.3. Các lệnh liên quan đến đỉnh – cạnh của tam giác.....	21
12.4. Ảnh của tam giác qua phép biến hình.....	22
12.5. Một số lệnh thường dùng trong tam giác.....	22

13. Tô màu – lát nền .....	27
13.1. Lát nền.....	27
13.2. Tô màu.....	29
13.3. Phối màu.....	30
13.3.1. Đồ bóng – 3D .....	30
13.3.2. Pha trộn nhiều màu .....	30
13.3.3. Tạo màu mới chuẩn rgb.....	31
13.3.4. Biểu đồ Ven .....	32
14. Bổ sung gói mới từ L <sup>A</sup> T <sub>E</sub> X .....	33
15. Các phép biến hình phẳng .....	36
16. Đường tròn .....	40
16.1. Định nghĩa – vẽ đường tròn.....	40
16.2. Chuyển đường tròn thành path .....	42
16.3. Giao điểm của đường thẳng – đường tròn, đường tròn – đường tròn.....	42
16.4. Tiếp tuyến của đường tròn.....	42
16.5. Một số lệnh thường dùng với đường tròn.....	43
17. Đường ellip .....	44
17.1. Định nghĩa – vẽ đường ellip.....	44
17.2. Chuyển đường ellipse thành path .....	44
17.3. Giao điểm của đường thẳng – ellipse, đường tròn – ellipse, ellipse – ellipse.....	44
17.4. Tiếp tuyến của ellipse .....	45
17.5. Một số lệnh thường dùng với đường ellipse.....	46
18. Ảnh qua phép biến hình .....	46
18.1. Ảnh của điểm .....	46
18.2. Ảnh của đoạn thẳng, đường thẳng.....	47
18.3. Ảnh của đường tròn .....	51
18.4. Ảnh của đường ellipse .....	52
19. Cung – quạt – vàng trắng khuyết .....	52
20. Vòng lặp.....	57
21. Khai báo số thực .....	59
22. Phép nghịch đảo.....	59
22.1. Định nghĩa .....	59
22.2. Ứng dụng của phép nghịch đảo .....	60
23. Định nghĩa hàm .....	61
24. Hàm số .....	62
25. Đồ thị hàm số – tô miền tích phân .....	62

## 1. Cài đặt – khai báo gói

### 1.1. Đối với hệ điều hành windows

Có 3 phần mềm chính cần cài đặt: Asymptote, Ghostscript, Tex editor (Texmaker, textstudio,...), miktex hoặc texlive. Riêng đối với những người đã cài hệ thống  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  thì chỉ cần cài thêm Asymptote, Ghostscript.

- 1) Phần mềm miktex hoặc texlive cái này cài trước nhất với mục đích chính là gắn nhãn của điểm, đường,... và bổ sung thêm vài gói lệnh trong  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Miktex có tại <https://miktex.org> hoặc texlive có tại trang <https://ctan.org/tex-archive/systems/texlive/Images>
- 2) Phần mềm Asymptote phiên bản mới nhất là 2.42 có tại [đây](#)
- 3) phần mềm ghostscript 9.18 khoảng 15mb có tại trang <http://downloads.ghostscript.com/public/old-gs-releases/> và cài thêm phần mềm gsvie để xem file ảnh eps sau khi xuất từ code có tại trang <https://www.fshare.vn/file/OTTD2W6R54TL>
- 4) Phần mềm Tex editor. Tôi ưu tiên cài Texmaker vì nó hỗ trợ dịch trực tiếp và không cần phải cấu hình sau khi cài đặt. Texmaker có tại <http://www.xmlmath.net/texmaker/download.html>

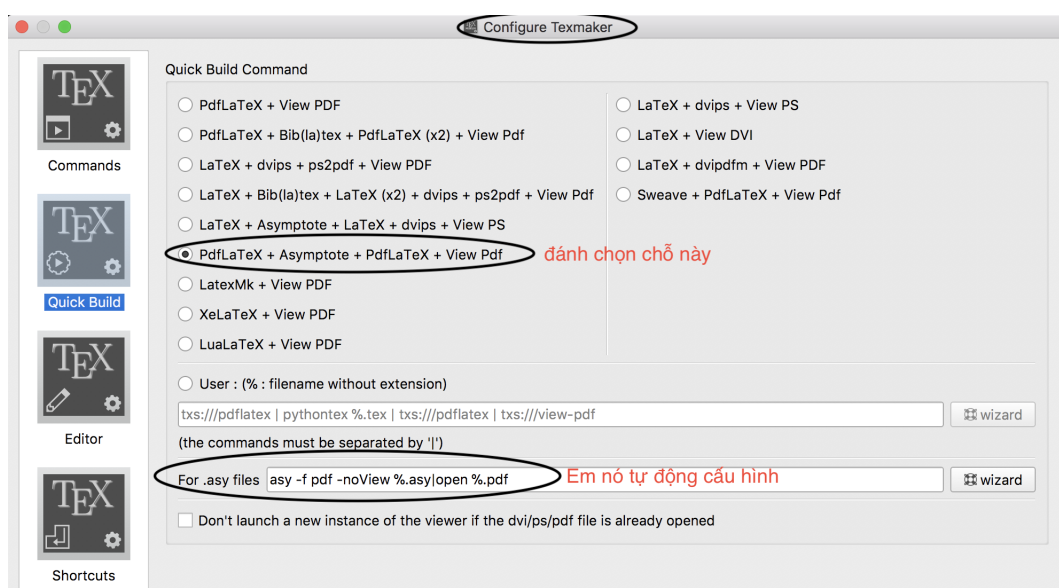
### 1.2. Đối với hệ điều hành MacOS

Có 2 phần mềm chính cần cài đặt: Mactex 2017 và tex editor (Texmaker hoặc Textstudio,...)

- 1) Trong Mactex 2017 đã chứa phần mềm asymptote có tại địa chỉ <https://tug.org/mactex/mactex-download.html>
- 2) Ưu tiên cài phần mềm Texmaker có tại <http://www.xmlmath.net/texmaker/download.html>

### 1.3. Khai báo gói

Asymptote hỗ trợ soạn thảo trực tiếp trong văn bản  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  bằng cách đặt trong môi trường asy và biên dịch qua 3 lần: `pdflatex -> asymptote -> pdflatex`. Cấu hình dịch trong Texmaker



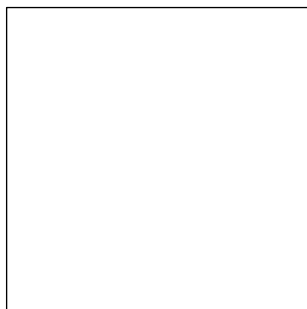
**Cú pháp.**

```

\documentclass[...]{...}
\usepackage[utf8]{vietnam}
\usepackage{graphicx}% cần có gói này
.....
\usepackage{asymptote} % gói chính
\begin{document}
.....
\begin{asy}
import geometry;
Các lệnh vẽ hình...
\end{asy}
.....
\end{document}

```

**Ví dụ 1.** Hình vẽ đầu tiên. Chèn lệnh vẽ hình trực tiếp vào văn bản  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ .



```

\documentclass[11pt]{article}
\usepackage[utf8]{vietnam}
\usepackage{graphicx}
\usepackage{asymptote}
\begin{document}
\begin{asy}
import geometry;
unitsize(2cm); // đơn vị độ dài.
draw ((0,0)--(2,0)--(2,2)--(0,2)--cycle);
\end{asy}
\end{document}

```

Nếu ta làm tài liệu lớn gồm nhiều hình vẽ soạn trực tiếp trong văn bản  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  thì quá trình dịch hơi lâu. Để quá trình dịch nhanh hơn và dễ kiểm soát, dễ sửa đổi hình vẽ thì ta sẽ vẽ hình đó ra một file riêng rồi dịch hình thành dạng pdf, eps, png và chèn vào tài liệu với cấu hình phiên dịch  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  là PdfLateX + View PDF ở trong Configure Texmaker. Các bước tạo 1 file hình riêng biệt.

1) Tạo 1 file hình trong phần mềm soạn thảo texmaker

```

import geometry;
Các lệnh vẽ hình...

```

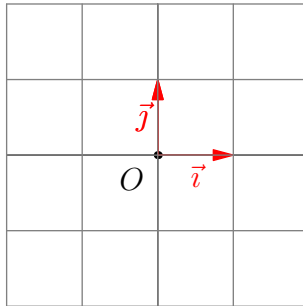
2) Lưu tập tin lại thành file.asy

3) Tiến hành phiên dịch bấm F1 hoặc chọn tools  $\rightarrow$  quick build trong texmaker. Khi đó asymptote tạo ra 1 file.pdf lưu trong thư mục chứa tập tin file.asy. Ta có thể dịch trong command prom hoặc trong terminal tham khảo thêm trong Asymptote: the Vector Graphics Language 2.42-1.pdf. Ở ví dụ tiếp theo minh họa cho vấn đề vẽ 1 hình riêng biệt.

## 2. Hệ trục tọa độ $Oxy$

- Để hiện hệ trục tọa độ mặc định trong gói geometry ta dùng lệnh `show(defaultcoordsys);` sau lệnh `import geometry;`
- Để hiện lưới tọa độ ta dùng lệnh `add(grid(a,b,gray));` với a, b là số ô lưới tương ứng hoành độ và tung độ được tính từ 0.

**Ví dụ 2.** Ở ví dụ này ta sẽ vẽ 1 hình riêng biệt chứ không phải chèn trực tiếp câu lệnh vào văn bản  $\text{\LaTeX}$ . Vẽ hệ trục và mặt phẳng tọa độ 4.4 ô lưới với 2.4 ô ở phần âm và 2.4 ô ở phần dương.



```
import geometry;
unitsize(1cm); // đơn vị
show(defaultcoordsys);
add(shift(-2,-2)*grid(4,4,gray)); // 16 ô: 8,8.
```

### 3. Màu sắc

Màu sắc được hỗ trợ trong Asymptote gồm:

Tham khảo trên trang <http://asy.marriss.fr/asymptote/Labels/index.html>

<span style="color: #f08080;">—</span> palered	<span style="color: #d3d3d3;">—</span> palegray
<span style="color: #ff6347;">—</span> lightred	<span style="color: #bdb76b;">—</span> lightgray
<span style="color: #ff4500;">—</span> mediumred	<span style="color: #808080;">—</span> mediumgray
<span style="color: #ff0000;">—</span> red	<span style="color: #696969;">—</span> gray
<span style="color: #8b0000;">—</span> heavyred	<span style="color: #3b3b3b;">—</span> heavygray
<span style="color: #654321;">—</span> brown	<span style="color: #191919;">—</span> deepgray
<span style="color: #2f4f4f;">—</span> darkbrown	<span style="color: #000000;">—</span> darkgray
<span style="color: #90ee90;">—</span> palegreen	<span style="color: #000000;">—</span> black
<span style="color: #90ee90;">—</span> lightgreen	<span style="color: #ffffff;">—</span> white
<span style="color: #32cd32;">—</span> mediumgreen	<span style="color: #ffa500;">—</span> orange
<span style="color: #008000;">—</span> green	<span style="color: #ff00ff;">—</span> fuchsia
<span style="color: #008000;">—</span> heavygreen	<span style="color: #00ff00;">—</span> chartreuse
<span style="color: #008000;">—</span> deepgreen	<span style="color: #00ff00;">—</span> springgreen
<span style="color: #008000;">—</span> darkgreen	
<span style="color: #add8e6;">—</span> paleblue	<span style="color: #800080;">—</span> purple
<span style="color: #6495ed;">—</span> lightblue	<span style="color: #0000ff;">—</span> royalblue
<span style="color: #4169e1;">—</span> mediumblue	
<span style="color: #0000ff;">—</span> blue	<span style="color: #00bfff;">—</span> Cyan
<span style="color: #0000ff;">—</span> heavyblue	<span style="color: #ff00ff;">—</span> Magenta
<span style="color: #0000ff;">—</span> deepblue	<span style="color: #ffff00;">—</span> Yellow
<span style="color: #0000ff;">—</span> darkblue	<span style="color: #000000;">—</span> Black
<span style="color: #7fffd4;">—</span> palecyan	<span style="color: #000000;">—</span> cmyk+red
<span style="color: #7fffd4;">—</span> lightcyan	
<span style="color: #40e0d0;">—</span> mediumcyan	
<span style="color: #00ced1;">—</span> cyan	
<span style="color: #00ced1;">—</span> heavycyan	
<span style="color: #00ced1;">—</span> deepcyan	
<span style="color: #00ced1;">—</span> darkcyan	
<span style="color: #ffb6c1;">—</span> pink	
<span style="color: #ff69b4;">—</span> lightmagenta	
<span style="color: #ff00ff;">—</span> mediummagenta	
<span style="color: #ff00ff;">—</span> magenta	
<span style="color: #ff00ff;">—</span> heavymagenta	
<span style="color: #ff00ff;">—</span> deepmagenta	
<span style="color: #ff00ff;">—</span> darkmagenta	
<span style="color: #fffacd;">—</span> paleyellow	
<span style="color: #fffacd;">—</span> lightyellow	
<span style="color: #fffacd;">—</span> mediumyellow	
<span style="color: #fffacd;">—</span> yellow	
<span style="color: #9acd32;">—</span> lightolive	
<span style="color: #808000;">—</span> olive	
<span style="color: #808000;">—</span> darkolive	

## 4. Điểm

### 4.1. Định nghĩa điểm

#### Cú pháp 1.

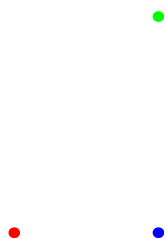
- 1) `point A=(x,y)`; Định nghĩa điểm  $A$  có hoành độ  $x$  và tung độ  $y$
- 2) `point A=(xA,yA), B=(xB,yB), C=(xC,yC)`; định nghĩa cho nhiều điểm.
- 3) `point Ax=conj(A)`;  $Ax$  là điểm đối xứng với  $A$  qua trục hoành.

#### Ví dụ 3.



```
import geometry;
unitsize(1cm);
point A=(0,0), B=(2,0), C=(2,3);
dot(A,linewidth(4bp)); // vẽ điểm A
dot(B,linewidth(4bp)); // có độ dày 4bp
dot(C,linewidth(4bp));
```

#### Ví dụ 4.

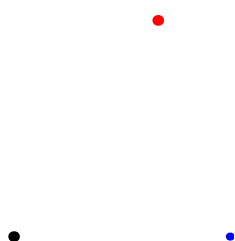


```
import geometry;
unitsize(1cm);
point A=(0,0), B=(2,0), C=(2,3);
dot(A,linewidth(4bp)+red); // vẽ điểm A màu đỏ
dot(B,linewidth(4bp)+blue);
draw(C,linewidth(4bp)+green); // vẽ điểm C
```

#### Cú pháp 2.

`point A[]; A[1]=(x1;y1); A[2]=(x2;y2); A[3]=(x3;y3);`

#### Ví dụ 5.



```
import geometry;
unitsize(3cm);
point A[];
A[1]=(0,0); A[2]=(3,0); A[3]=(2,3);
draw(A[1],linewidth(4bp));
draw(A[2],3bp+blue);
dot(A[3],red+4bp);
```

## Đơn vị độ dài được sử dụng trong L<sup>A</sup>T<sub>E</sub>X

1pt	1mm	1cm	1ex	1em	1bp	1dd	1pc	1in	
1	2.84526	28.45274	4.30554	10.00002	1.00374	1.07001	12	72.26999	pt
0.35146	0.99999	10	1.51322	3.5146	0.35277	0.37606	4.21752	25.4	mm
0.03515	0.1	1	0.15132	0.35146	0.03528	0.03761	0.42175	2.54	cm
0.23226	0.66084	6.6084	1	2.32259	0.23313	0.24852	2.78711	16.78534	ex
0.1	0.28453	2.84527	0.43055	1	0.10037	0.107	1.2	7.22699	em
0.99626	2.83463	28.34644	4.28946	9.96266	0.99999	1.06601	11.95517	71.99999	bp
0.93457	2.6591	26.59114	4.02384	9.34573	0.93807	1	11.21486	67.54151	dd
0.08333	0.23711	2.37106	0.3588	0.83333	0.08364	0.08917	1	6.0225	pc
0.01384	0.03937	0.3937	0.05958	0.13837	0.01389	0.01481	0.16604	1	in

Satuan ukuran dalam pengetikan (*typographic unit*)

pt = point    ex = X    em = M    bp = big point    dd = didot    pc = pica    in = inch

Diberdayakan oleh L<sup>A</sup>T<sub>E</sub>X © 2017 Sulaeman, S.Pd. — SMKN 2 Sumbawa Besar

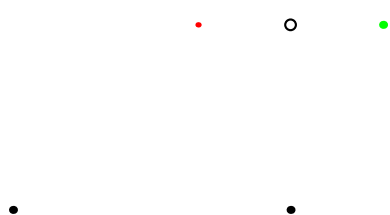
<http://menulislalax.blogspot.com/>

## 4.2. Vẽ điểm

### Cú pháp.

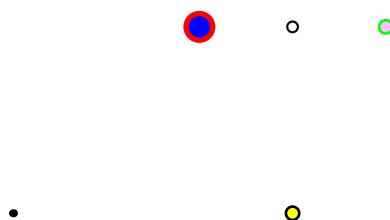
- 1) `draw(A)`; vẽ điểm  $A$  là một chấm đen với độ dày 0.5bp. Tham khảo thêm đơn vị độ dài được sử dụng trong L<sup>A</sup>T<sub>E</sub>X.
- 2) `draw(A,linewidth(2bp))`; vẽ điểm  $A$  là một chấm có độ dày 2bp
- 3) `draw(A,màu+độ dày)`; hoặc `draw(A,độ dày+ màu)`; tô màu cho điểm  $A$ .
- 4) `dot(A,độ dày+màu,Fill(màu))`; vẽ điểm điểm  $A$  tô màu viền.

### Ví dụ 6.



```
import geometry; unitsize(1cm);
point A,B,C,D,E; A=(0,0); B=(3,0); C=(2,2); D=(4,2);
E=(3,2); dot(A);
draw(B,linewidth(3bp)); draw(C,2bp+red);
draw(D,3bp+green);
dot(E,0.8bp+black,Fill(white));
```

### Ví dụ 7.



```
import geometry; unitsize(1cm); point A,B,C,D,E;
A=(0,0); B=(3,0); C=(2,2); D=(4,2); E=(3,2);
dot(A); dot(B,linewidth(1bp),Fill(yellow));
dot(C,2bp+red,Fill(blue));
dot(D,1bp+green,Fill(pink));
dot(E,0.8bp+black,Fill(white));
```



### 4.3. Nhãn – vị trí nhãn

#### Cú Pháp.

- 1) `dot("nhãn",điểm,vị trí,màu);`
- 2) `draw("nhãn",điểm,vị trí,màu);`
- 3) `label("nhãn",điểm,vị trí,màu);`
- 4) `labelmargin=2;` điều chỉnh khoảng cách từ điểm đến nhãn.
- 5) `defaultpen(fontsize(20pt));` điều chỉnh size mặc định của nhãn.

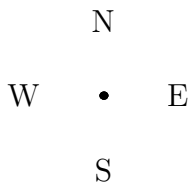
Vị trí bao gồm: N: Bắc, S:Nam, E: Đông, W: Tây, NE: Đông Bắc, SE: Đông Nam, NW: Tây Bắc, SW: Tây Nam. Xem các ví dụ

**Ví dụ 8.** Khoảng cách và size nhãn mặc định.



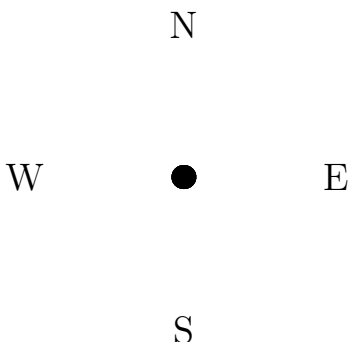
```
import geometry;
point O=(0,0);
dot("$N$",O,N);
draw("S",O,S);
dot("E",O,E);
label("W",O,W);
```

**Ví dụ 9.** Tăng khoảng cách từ điểm đến nhãn.



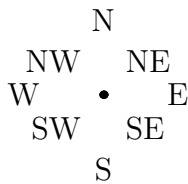
```
import geometry;
point O = (0,0);
labelmargin = 2;
dot("N",O,N);
dot("S",O,S);
dot("E",O,E);
dot("W",O,W);
```

**Ví dụ 10.** Kết hợp tăng khoảng cách, độ dày của điểm và size của nhãn.



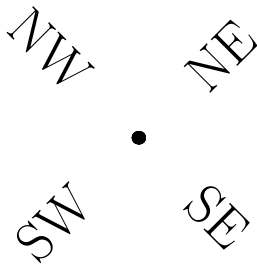
```
import geometry;
defaultpen(fontsize(15pt));
labelmargin=3.5;
dotfactor*=3;//tăng độ dày của điểm
point O=(0,0);
dot("N",O,N);
dot("S",O,S);
dot("E",O,E);
dot("W",O,W);
```

**Ví dụ 11.** Kết hợp tăng khoảng cách, độ dày của điểm và size của nhãn.



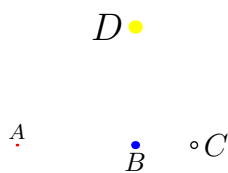
```
import geometry;
point O=(0,0);
labelmargin=2;
draw("N",O,N);
draw("S",O,S);
draw("E",O,E);
draw("W",O,W);
dot("NE",O,0.5NE);
dot("SE",O,0.5SE);
dot("NW",O,0.5NW);
dot("SW",O,.5SW);
```

**Ví dụ 12.** Quay nhãn, tăng size của nhãn và thay đổi độ dày mặc định của điểm



```
import geometry; unitsize(1cm);
defaultpen(linewidth(3bp)); //tăng độ dày điểm.
point O=(0,0);
draw(rotate(45)*Label("NE"),O,5NE);
draw(rotate(-45)*Label("SE"),O,5SE);
draw(rotate(-45)*Label("NW"),O,5NW);
draw(rotate(45)*Label("SW"),O,5SW);
```

**Ví dụ 13.** Kết hợp vẽ điểm, gắn nhãn, tô màu điểm, tô màu nhãn.

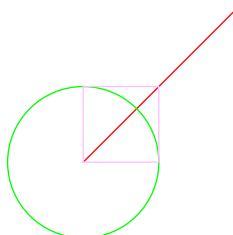


```
import geometry; unitsize(1cm);
point A,B,C,D; A=(0,0); B=(2,0); C=(3,0); D=(2,2);
dot(Label("$A$",align=N,fontsize(8pt)),A,1bp+red);
dot(Label("$B$",align=S,fontsize(10pt)),B,3bp+blue);
dot(Label("$C$",align=E,fontsize(12pt)),C,0.5bp+black,
Fill(white));
draw(Label("$D$",align=W,fontsize(15pt)),D,5bp+yellow);
```

## 5. Một số cú pháp cơ bản trong asymptote

- \* ; Kết thúc mỗi lệnh, ta dùng dấu ;
- \* // dùng để comment. Những lệnh nằm sau dấu // sẽ bỏ qua trong quá trình dịch.
- \* pair A=(x,y); pair là đối tượng cơ bản trong Asymptote, là một điểm trong mặt phẳng tọa độ  $Oxy$ .  
Còn trong geometry thì điểm được định nghĩa là point.
- \* draw(A--B); Vẽ đoạn thẳng nối giữa điểm A và điểm B.
- \* draw(unitsquare); Vẽ hình vuông đơn vị có đỉnh (0,0), (1,0), (1,1), (0,1).
- \* draw(unitcircle); Vẽ đường tròn đơn vị có tâm là điểm  $O(0,0)$ .
- \* draw(polygon(n)); Vẽ đa giác đều  $n$  cạnh.
- \* path p=A--B; là đoạn thẳng nối A và B.

**Ví dụ 14.** Vẽ đường tròn, hình vuông đơn vị và path A- -B.



```
import geometry;
unitsize(1cm);
point A=(0,0), B=(2,2);
draw(A--B,red); // Nối đoạn AB màu đỏ
draw(unitcircle,green); // Đường tròn đơn vị màu xanh
draw(unitsquare,pink); // hình vuông đơn vị màu hồng
```

## 6. Đường thẳng

### 6.1. Định nghĩa đường thẳng

#### 6.1.1. Đường thẳng qua hai điểm

Cú pháp.

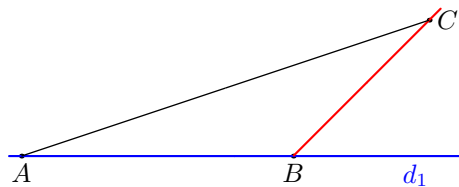
- 1) `line d=line(A,B)`; định nghĩa đường thẳng  $d$  đi qua hai điểm  $A$  và  $B$ .
- 2) `line d=line(A,false,B)`; định nghĩa đường thẳng  $d$  đi qua hai điểm  $A$  và  $B$  có không có phần nối dài từ điểm  $A$ .
- 3) `line d=line(A,false,B,false)`; định nghĩa đường thẳng  $d$  đi qua hai điểm  $A$  và  $B$  không có phần nối dài hai đầu mút.

#### 6.1.2. Vẽ đường thẳng và gắn nhãn

Cú pháp.

`draw("d",d,độ dày+màu)`; vẽ đường thẳng  $d$  và gắn nhãn cho đường thẳng  $d$  đã được định nghĩa. Xem các ví dụ.

Ví dụ 15.



```
import geometry; unitsize(1cm);
defaultpen(fontsize(10pt)); //size nhãn 10pt
dotfactor*=.6; // điều khiển độ dày của điểm.
point A,B,C,D;
A=(0,0); B=(2,0); C=(3,1); D=(-1,2);
dot("$A$",A,S); dot("$B$",B,S); dot("$C$",C,E);
line d1=line(A,B);
line d2=line(B,false,C);
line d3=line(C,false,A,false);
draw("$d_1$",d1,0.8bp+blue);
draw(d2,.8bp+red); draw(d3);
```

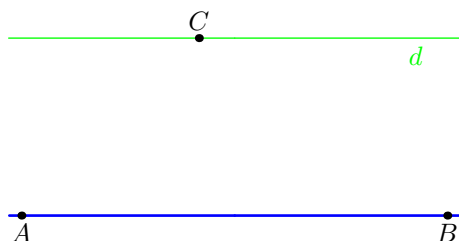
### 6.2. Định nghĩa đường thẳng bằng quan hệ song song, vuông góc, ...

Đường thẳng đi qua một điểm và song song với một đường thẳng

Cú pháp.

`line d=parallel(M,l)`; định nghĩa đường thẳng  $d$  đi qua điểm  $M$  và song song với đường thẳng  $l$  cho trước.

Ví dụ 16.



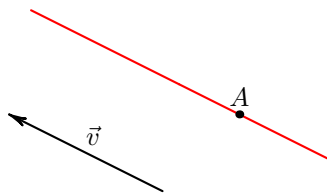
```
import geometry; unitsize(1cm);
defaultpen(fontsize(10pt)); // size nhãn 10pt
point A, B, C; A=(0,0); B=(2.4,0); C=(1,1);
line l=line(A,B); line d=parallel(C,l);
draw(l,bp+blue); draw("$d$", d,0.5bp+green);
dot("$A$",A,S); dot("$B$",B,S); dot("$C$",C,N);
```

## Đường thẳng đi qua một điểm và có một vectơ chỉ phương

### Cú pháp.

- 1) `line d=parallel(M,v)`; đường thẳng  $d$  đi qua điểm  $M$  và nhận vectơ  $\vec{v}$  làm vectơ chỉ phương với  $\vec{v}$  đã được định nghĩa.
- 2) `line d=parallel(M,(a,b))`; đường thẳng  $d$  đi qua điểm  $M$  và nhận vectơ  $\vec{v} = (a,b)$  làm vectơ chỉ phương.

### Ví dụ 17.



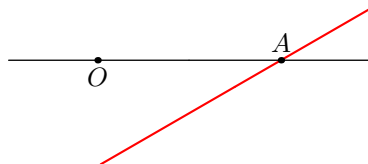
```
import geometry; unitsize(1cm);
linemargin=-1cm; //giãn lề của hình
point A=(1,1);
vector v=(-2,1); // Định nghĩa vectơ v
line d=parallel(A,v); draw(d,0.8bp+red);
show("$\vec{v}$",v,linewidth(0.8bp),
    Arrow(HookHead,size=1.2mm)); // hiện vectơ v
dot("$A$",A,N);
```

## Đường thẳng đi qua một điểm và có góc tạo bởi tia $Ox$ với nó

### Cú pháp.

`line d=line(M,góc)`; đường thẳng  $d$  đi qua điểm  $M$  tạo với tia  $Ox$  một góc (độ).

### Ví dụ 18.



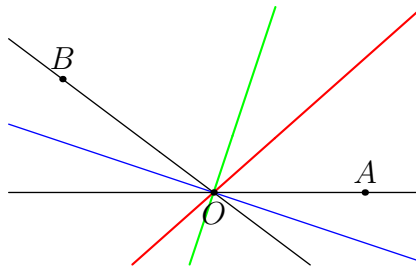
```
import geometry; unitsize(1cm);
linemargin=-1cm; dotfactor*=.8;
point O, A, B; O=(0,0); A=(2.4,0); B=(4,0);
line Ox=line(O,A); line l=line(A,30);
draw(l,0.8bp+red); draw(Ox);
dot("$O$",O,S); dot("$A$",A,N);
```

## Đường thẳng phân giác của góc tạo bởi hai đường thẳng cắt nhau

### Cú pháp.

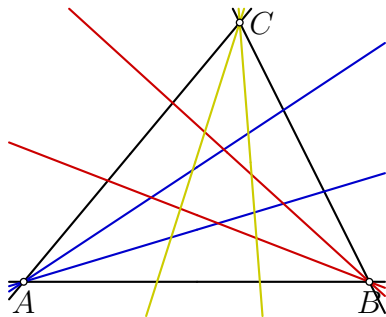
- 1) `line d=bisector(l1,l2)`; đường thẳng  $d$  là phân giác của góc nhọn tạo bởi hai đường thẳng  $l1$  và  $l2$ .
- 2) `line d=bisector(l1,l2,false)`; đường thẳng  $d$  là phân giác của góc tù tạo bởi hai đường thẳng  $l1$  và  $l2$ .
- 3) `line d=bisector(l1,l2,góc)`; quay đường thẳng  $l1$  quanh giao điểm của  $l1$  và  $l2$  với góc(độ).
- 4) `line d=sector(đối số 1,đối số 2 ,l1,l2)`: đường thẳng  $d$  qua giao điểm của  $l1$  và  $l2$  theo góc chia đều đối số 2.

### Ví dụ 19.



```
import geometry; unitsize(1cm); linemargin=-1cm;
dotfactor*=.8; point O=(0,0),A=(4,0),B=(-4,3);
line OA=line(O,A); line OB=line(O,B);
draw(OA); draw(OB); line nhon=bisector(OA,OB);
line tu=bisector(OA,OB,false);
line quay=bisector(OA,OB,60);
draw(nhon,.5bp+blue); draw(tu,.8bp+green);
draw(quay,.8bp+red); dot("$O$",O,S);
dot("$B$",B,N); dot("$A$",A,N);
```

### Ví dụ 20.



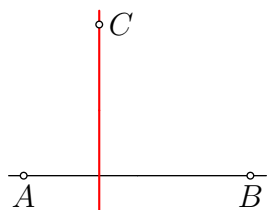
```
import geometry; unitsize(1cm);
point A=(0,0), B=(4,0), C=(2.5,3);
pen pb=0.8*blue, pr=0.8*red, py=0.8*yellow,
    bpp=linewidth(.5bp);
line AB=line(A,B), AC=line(A,C), BC=line(B,C);
draw(AB, bpp); draw(AC, bpp); draw(BC, bpp);
line bA1=sector(3,1,AB,AC), bA2=sector(3,2,AB,AC);
line bB1=sector(3,1,AB,BC), bB2=sector(3,2,AB,BC);
line bC1=sector(3,1,AC,BC,false),
    bC2=sector(3,2,AC,BC,false);
draw(bA1, bpp+pb); draw(bA2, bpp+pb);
draw(bB1, bpp+pr); draw(bB2, bpp+pr);
draw(bC1, bpp+py); draw(bC2, bpp+py);
dot("$A$",S,A,.5bp+black,Fill(white));
dot("$B$",S,B,.5bp+black,Fill(white));
dot("$C$",S,C,.5bp+black,Fill(white));
```

## Đường thẳng đi qua một điểm và vuông góc với một đường thẳng

### Cú pháp.

`line d=perpendicular(M,l);` đường thẳng  $d$  đi qua điểm  $M$  và vuông góc với đường thẳng  $l$  đã định nghĩa.

### Ví dụ 21.



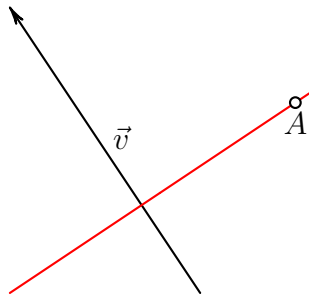
```
import geometry; unitsize(1cm);
pen p1=linewidth(0.5bp); filltype p2=Fill(white);
point A=(0,0),B=(3,0),C=(1,2); line AB=line(A,B);
draw(AB); line d=perpendicular(C,AB);
draw(d,.8bp+red); dot("$A$",A,S,p1,p2);
dot("$B$",B,S,p1,p2); dot("$C$",C,E,p1,p2);
```

## Đường thẳng đi qua một điểm và có một vectơ pháp tuyến

### Cú pháp.

- 1) `line d=perpendicular(M,v);` đường thẳng  $d$  đi qua điểm  $M$  nhận vectơ  $\vec{v}$  làm vectơ pháp tuyến.
- 2) `line d=perpendicular(M,(a,b));` đường thẳng  $d$  đi qua điểm  $M$  nhận vectơ  $\vec{v} = (a, b)$  làm vectơ pháp tuyến.

### Ví dụ 22.



```
import geometry; unitsize(1cm);
pen p1=linewidth(.8bp)+black;
point A=(1,2); vector v=(-2,3);
show("$\vec v$",v,p1,Arrow(HookHead,size=1.2mm));
line d=perpendicular(A,v); draw(d,.8bp+red);
dot(Label("$A$",align=S),A,.5bp+black,Fill(white));
```

## 6.3. Đường thẳng trung trực

### Cú pháp.

line d=bisector(A,B); đường thẳng trung trực của đoạn thẳng AB.

## 7. Đánh dấu góc

### 7.1. Đánh dấu góc không vuông

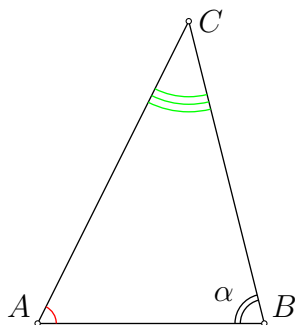
#### Cú pháp.

1) markangle("nhãn",đổi số 1,đổi số 2,đổi số 3,l1,l2,màu,tô màu,đánh dấu cung);

2) markangle("nhãn", đổi số1,đổi số2,đổi số3,A,B,C,mũi tên,màu+độ dày,tô màu);

- Đổi số 1: số cung.
- Đổi số 2: bán kính cung.
- Đổi số 3: khoảng cách cung.
- l1,l2: hai đường thẳng cần đánh dấu góc.
- A,B,C là góc cần đánh dấu  $\angle ABC$ .

### Ví dụ 23.

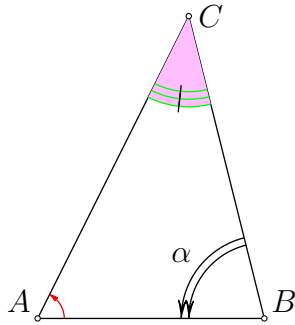


```
import geometry; unitsize(1cm);
pen p1=linewidth(.4bp), p2=linewidth(.8bp);
filltype p3=Fill(white), p4=Fill(pink);//đn tô màu
point A=(0,0), B=(3,0),C=(2,4);
line AB=line(A,B); line CB=line(C,B);
line AC=line(A,C); draw(A--B);
draw(C--B); draw(A--C); //đoạn thẳng
markangle(1,7,AB,AC,red);
markangle("$\alpha$",2,9,2,line(B,C),line(B,A));
markangle(3,line(C,A),line(C,B),green);
dot("$A$",A,NW,p1,p3); dot("$B$",B,NE,p1,p3);
dot("$C$",C,E,p1,p3);
```

**Ví dụ 24.** Thêm mũi tên và đánh dấu cung. Để đánh dấu cung, thêm lệnh `StickIntervalMarker(đổi số 1,đổi số 2,size=...,angle=...)`. Tham khảo thêm trên trang chủ <sup>1</sup>

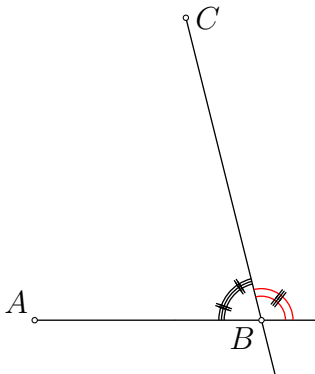
<sup>1</sup> trang chủ.

- đối số 1: Chia cung ra số phần để đánh dấu.
- đối số 2: Số gạch cần đánh dấu trong một phần đã chia.
- size=...: độ dài của gạch đánh dấu.
- angle=...: góc nghiêng của gạch.



```
import geometry; unitsize(1cm);
pen p1=linewidth(.4bp), p2=linewidth(.8bp);
filltype p3=Fill(white), p4=Fill(pink);
point A=(0,0), B=(3,0), C=(2,4);
draw(A--B); draw(C--B); draw(A--C);
markangle(1,10,B,A,C,Arrow(1mm),red);
markangle("$\alpha$",2,C,B,A,Arrow(HookHead,1.2mm));
markangle(3,A,C,B,green,p4,StickIntervalMarker(1,1,size=1.5mm));
dot("$A$",A,NW,p1,p3);
dot("$B$",B,NE,p1,p3);
dot("$C$",C,E,p1,p3);
```

Ví dụ 25.



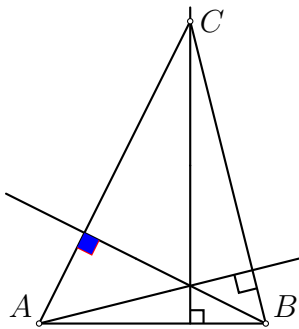
```
import geometry; unitsize(1cm);
pen p1=linewidth(.4bp), p2=linewidth(.8bp);
filltype p3=Fill(white), p4=Fill(pink);
linemargin=-0.3cm; point A=(0,0), B=(3,0), C=(2,4);
arrowbar ar=Arrow(HookHead,1.2mm);
marker mk1=StickIntervalMarker(1,2,1.5mm,red);
line AB=line(A,false,B), BC=line(B,C,false); draw(AB); draw(BC);
markangle(2,9,AB,BC,red,StickIntervalMarker(1,3,size=1.4mm,space=.3mm));
markangle(3,14,1,BC,line(B,A),StickIntervalMarker(2,2,size=1.4mm,space=.3mm));
dot("$A$",A,NW,p1,p3); dot("$B$",B,SW,p1,p3);
dot("$C$",C,E,p1,p3);
```

## 7.2. Đánh dấu góc vuông

Cú pháp.

- perpendicularmark(l1,l2,size,quater=...,màu, tô màu);
- markrightangle(A,B,C,màu,tô màu,size=,...); đánh dấu góc vuông  $\angle ABC$ .
  - l1,l2 là hai đường thẳng vuông góc.
  - quater là các phần tư gồm: 1, 2, 3, 4.

Ví dụ 26. xem ví dụ.



```
import geometry; unitsize(1cm);
defaultpen(linewidth(.8bp));
pen p1=linewidth(.4bp), p2=linewidth(.8bp);
filltype p3=Fill(white), p4=Fill(pink);
point A=(0,0), B=(3,0),C=(2,4);
arrowbar ar=Arrow(HookHead,1.2mm);
marker mk1=StickIntervalMarker(1,2,1.5mm,red);
line AB=line(A,B); line CB=line(C,B); line AC=line(A,C);
draw(A--B); draw(C--B); draw(A--C);
line hC=perpendicular(C,AB), hB=perpendicular(B,AC);
line hA=perpendicular(A,CB); draw(hC); draw(hB); draw(hA);
perpendicularmark(AB,hC,size=5,quarter=1);
perpendicularmark(AC,hB,size=6,quarter=3,red,Fill(blue));
perpendicularmark(CB,hA,size=7,quarter=4);
dot("$A$",A,NW,p1,p3); dot("$B$",B,NE,p1,p3);
dot("$C$",C,E,p1,p3);
```

## 8. Định nghĩa, vẽ, ký hiệu đoạn thẳng

### 8.1. Định nghĩa đoạn thẳng

Cú pháp.

- 1) `segment s=segment(A,B)`: đoạn thẳng  $AB$  với tên là  $s$ .
- 2) `segment s=l`: đoạn thẳng  $s$  được giới hạn bởi hai điểm đi qua của đường thẳng  $l$  đã định nghĩa.
- 3) `point M=midpoint(s)`:  $M$  là trung điểm của đoạn thẳng  $s$ .
- 4) `line d=bisector(s)`: đường trung trực của đoạn thẳng  $s$ .
- 5) `line l=bisector(s,angle=góc)`: đường thẳng  $l$  tạo bởi 1 góc với đường trung trực của đoạn thẳng  $s$ .
- 6) `draw(complementary(segment(A,B)))`: vẽ phần bù của đoạn thẳng  $AB$  trên đường thẳng  $AB$  (bỏ trống đoạn thẳng  $AB$  chỉ vẽ hai đầu mút).

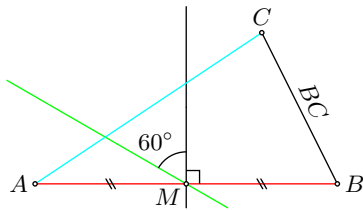
### 8.2. Vẽ – ký hiệu đoạn thẳng

Cú pháp.

- 1) `draw("nhãn",đoạn thẳng,màu,ký hiệu)`: ta thay |đoạn thẳng| bằng path nối giữa hai đầu mút của đoạn thẳng để tiện đánh dấu vì gói geometry không hỗ trợ tốt vấn đề này. Đặt vị trí nhãn cho đoạn thẳng gồm: Center, LeftSide, RightSide.
- 2) `StickIntervalMarker(ds1,ds2,size,angle=...,space=...mm)`: ký hiệu cho đoạn thẳng giống như ký hiệu của cung đánh dấu góc.

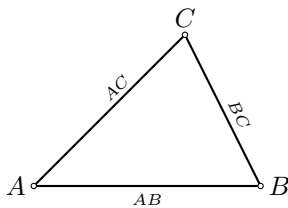
**Ví dụ 27.** Trong ví dụ này, gắn nhãn cho đoạn thẳng  $BC$  bằng mặc định "không đẹp lắm." Ta sẽ làm các khác.





```
import geometry; defaultpen(fontsize(9pt));
filltype to = Fill(white); unitsize(1cm); dotfactor*=.7;
point A=(0,0),B=(4,0),C=(3,2);
segment s =segment(A,B),t=segment(A,C);
path i =B--C; // đoạn BC ở dạng path
point M = midpoint(s);
draw(Label(rotate(-65)*"$BC$",align=RightSide),i);
draw(s,red,StickIntervalMarker(2,2,angle =30,size=1.5mm,
space=.5mm));
draw(bisector(s)); perpendicularmark(line(A,B),bisector(s),
quarter=1,size=5);
draw(bisector(s,60),green); draw(t,cyan);
markangle("$60^\circ$",1,12,bisector(s),bisector(s,60));
dot("$M$",M,SW,to); dot("$A$",A,W,to);
dot("$B$",B,E,to); dot("$C$",C,N,to);
```

**Ví dụ 28.** Gắn nhãn cho đoạn thẳng "đẹp hơn."

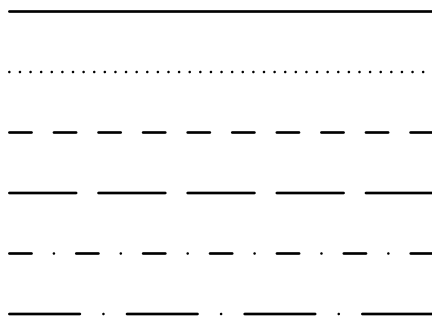


```
import geometry; unitsize(1cm);
defaultpen(linewidth(.8bp)); defaultpen(fontsize(11pt));
pen p1=linewidth(.4bp); filltype p2=Fill(white);
point A=(0,0), B=(3,0), C=(2,2);
draw(Label("\tiny $AB$",align=S),A--B);
draw(Label("\tiny $AC$",Rotate(dir(A--C)),align=N),A--C);
draw(Label("\tiny $BC$",Rotate(-dir(B--C)),align=E),B--C);
dot("$A$",A,W,p1,p2); dot("$B$",B,E,p1,p2);
dot("$C$",C,N,p1,p2);
```

## 9. Kiểu đường thẳng, đoạn thẳng,...

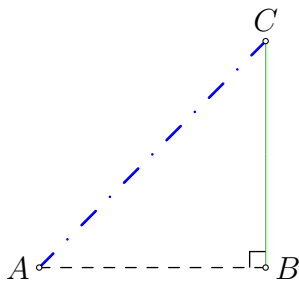
**Cú pháp.**

`draw(..., kiểu, kiểu đường,...);` vẽ đường theo kiểu. Một số kiểu đường thường dùng. Kết quả sau được lấy tại trang <http://asy.marris.fr/asympote/Lignes/index.html> và tham khảo thêm trên trang [piprime.fr](http://piprime.fr)



- 1) Kiểu solid mặc định.
- 2) Kiểu dotted.
- 3) Kiểu dashed (hay dùng).
- 4) Kiểu longdashed.
- 5) Kiểu dashdotted.
- 6) Kiểu longdashdotted.

**Ví dụ 29.**



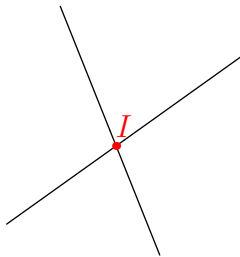
```
import geometry; unitsize(1cm);
pen p1=linewidth(.4bp),p2=linewidth(.8bp);
filltype p3=Fill(white);point A=(-1,0),B=(2,0),C=(2,3);
segment sAB=segment(A,B), sBC=segment(B,C),
      sCA=segment(C,A); markrightangle(C,B,A,size=3);
draw(sAB,dashed); draw(sBC,solid+green);
draw(sCA,dashdotted+blue+1bp);
dot("$A$",A,W,p1,p3);
dot("$B$",B,E,p1,p3);
dot("$C$",C,N,p1,p3);
```

## 10. Giao điểm của hai đường thẳng

### Cú pháp.

- 1) point  $I = \text{intersectionpoint}(l1, l2)$ ;  $I$  là giao điểm của hai đường thẳng  $l1, l2$ .
- 2) point  $I = \text{intersectionpoint}(s1, s2)$ ;  $I$  là giao điểm của hai đoạn thẳng  $s1, s2$ .

### Ví dụ 30.

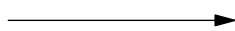


```
import geometry; unitsize(1cm);
linemargin=-2cm; point A=(-1,-1), B=(5,2);
point C=(0,3), D=(2,-3);
line l1=line(A,B); line l2=line(C,D);
draw(l2); draw(l1);
point I=intersectionpoint(l1,l2);
dot("$I$",I,dir(65),red);
```

## 11. Kiểu mũi tên

### Cú pháp.

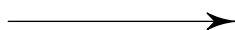
`draw(..., Arrow hoặc Arrow(kiểu mũi tên), ...)`; Các kiểu mũi tên thường dùng và tham khảo thêm trên [web này](#). Để có mũi tên hai đầu mũi, ta thêm `s` ở sau Arrow.



1) Arrow mặc định.



2) SimpleHead.

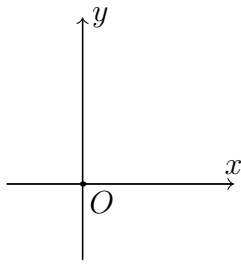


3) HookHead.



4) TeXHead.

### Ví dụ 31.

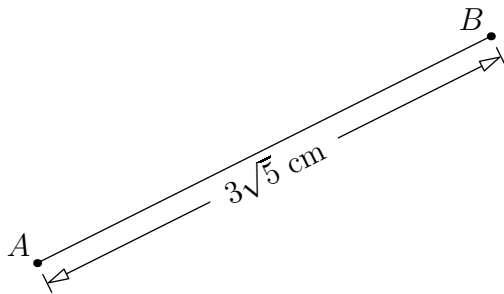


```
import geometry; unitsize(1cm);
DefaultHead=TeXHead;//Mặc định mũi tên kiểu TeXHead.
pen p1=linewidth(0.4bp),p2=linewidth(2bp);
filltype p3=Fill(white);
point O=(0,0),A=(-1,0),B=(2,0),C=(0,-1),D=(0,2.2);
segment sAB=segment(A,B), sCD=segment(C,D);
draw(Label("$x$",EndPoint,N),sAB,Arrow());
draw(Label("$y$",EndPoint,E),sCD,Arrow());
dot(Label("$O$",align=SE),O,p2);
```

## Ký hiệu khoảng cách.

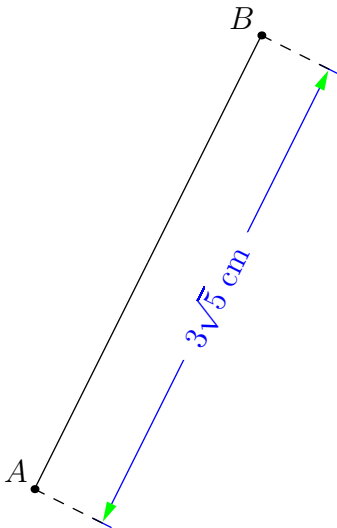
`distance("nhãn", A,B, các tùy chọn)`; kí hiệu và gắn nhãn cho khoảng cách giữa 2 điểm  $A$  và  $B$ .

**Ví dụ 32.** Xem hai ví dụ này và tham khảo thêm trên [trang chủ](#)



```
import geometry;
unitsize(1cm);
point A=(-1,2), B=(5,5);
draw(segment(A,B));
distance("$3\sqrt{5}$ cm",A,B);
dot("$A$",A,NW);dot("$B$",B,NW);
```

**Ví dụ 33.** Có các tùy chọn



```
import geometry; unitsize(1cm);
point A=(0,-2), B=(3,4);
draw(segment(A,B));
dot("$A$",A,NW);dot("$B$",B,NW);
distance("$3\sqrt{5}$ cm", // nhãn
A,B, // điểm A,B
rotated=false, // quay nhãn: true, false
offset=10mm, // điểm đến mũi tên: 3mm
p=blue, // màu: currentpen
joinpen=dashed, // đường nối: invisible
Arrows(Fill(green))// Mũi tên: Arrows(NoFill)
);
```

## 12. Tam giác

### 12.1. Định nghĩa tam giác

**Cú pháp.**

- 1) `triangle t=triangle(point A, point B, point C)`;  $t$  là tam giác  $ABC$ .
- 2) `triangle t=triangleabc(a, b, c, angle=0, A=(0,0))`;  $t$  là tam giác  $ABC$  với  $AB = b$ ,  $BC = a$ ,  $CA = c$  và  $(\vec{i}, \vec{AB}) = \text{angle}$ . Mặc định là  $\text{angle} = 0$  và điểm  $A$  là gốc tọa độ  $O = (0,0)$ .

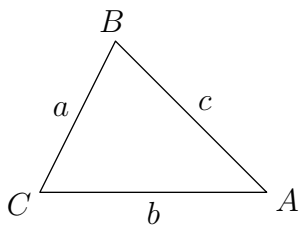
- 3) `triangle t=triangleAbc(alpha, b, c, angle=0, A=(0,0));`  $t$  là tam giác  $ABC$  thỏa mãn  $(\overrightarrow{AB}, \overrightarrow{AC}) = \alpha$ ,  $AB = c$ ,  $AC = b$  và  $(\vec{i}, \overrightarrow{AC}) = \text{angle}$ . Mặc định  $\text{angle} = 0$  và  $A$  là gốc tọa độ  $O$ .
- 4) `triangle t=triangle( l1, l2, l3);`  $t$  là tam giác  $ABC$  có 3 cạnh tương ứng nằm trên 3 đường thẳng  $l1, l2, l3$ .

## 12.2. Vẽ tam giác – nhãn các đỉnh tam giác

### Cú pháp.

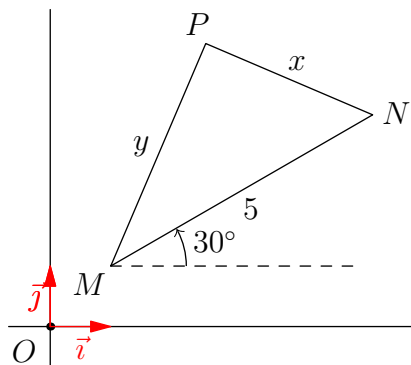
- 1) `label(LA="$A$", LB="$B$", LC="$C$", t, alignAngle=0, alignFactor=1, pen, fill);`  
hiện nhãn cho tam giác  $t = \triangle ABC$  ở đỉnh  $A, B, C$  lần lượt là  $A, B, C$  và  $\text{alignAngle}=0$  là góc lệch của nhãn so với đường phân giác trong của góc ở các đỉnh còn  $\text{alignFactor}=1$  là khoảng cách từ các đỉnh đến nhãn, `pen`, `fill` màu và tô màu.
- 2) `show(LA="$A$", LB="$B$", LC="$C$", La="$a$", Lb="$b$", Lc="$c$", t, pen, fill);`  
Vẽ tam giác  $t$  và nhãn ở các đỉnh, các cạnh tương ứng  $A, B, C$  và  $a, b, c$ . Nếu bỏ `LA, ..., La` thì geometry mặc định là  $A, B, C$  ở các đỉnh và  $a, b, c$  là các cạnh.
- 3) `draw(t, màu, đánh dấu);` vẽ tam giác  $t$  (các cạnh) và đánh dấu các cạnh.
- 4) `drawline(t, màu);` vẽ tam giác  $t$  là ba đường thẳng chứa ba cạnh.

### Ví dụ 34.



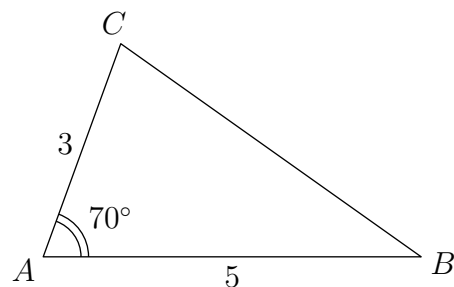
```
import geometry; unitsize(1cm);
point A=(3,0), B=(1,2), C=(0,0);
triangle t=triangle(A,B,C);
show(t); // mặc định.
```

### Ví dụ 35.



```
import geometry; unitsize(1cm);
show(currentcoordsys); // trục tọa độ.
triangle t=triangleabc(3,4,5,angle=30,(1,1));
show(LA="$M$", LB="$N$", LC="$P$", La="$x$",
      Lb="$y$", Lc="$5$", t); //
markangle("$30^\circ$", (2,1), t.A, t.B, Arrow(TeXHead));
draw((1,1)--(5,1), dashed);
```

### Ví dụ 36.



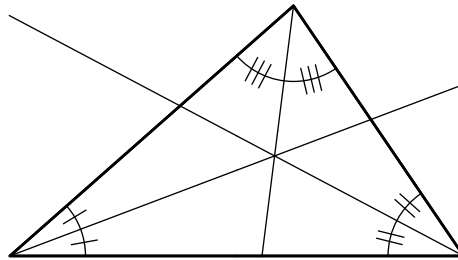
```
import geometry; unitsize(1cm);
triangle t=triangleAbc(70,3,5);
show(La="", Lb="3", Lc="5", t);
markangle("$70^\circ$", 2, radius=.5cm, t.B, t.A, t.C);
//t.B, t.A, t.C là 3 đỉnh của tam giác t.
// Sẽ nói ở phần sau
```

### 12.3. Các lệnh liên quan đến đỉnh – cạnh của tam giác

Giả sử tam giác  $t$  đã được định nghĩa. Các  $t.VA$ ,  $t.VB$ ,  $t.VC$ ,  $t.AB$ ,  $t.BC$ ,  $t.CA$ ,  $t.BA$ ,  $t.CB$ ,  $t.AC$  là các đỉnh, cạnh của tam giác  $t$ . Geometry cung cấp một số lệnh liên quan.

- 1) `line ba=bisector(t.VA)`; đường phân giác trong của góc tại đỉnh  $A$  của tam giác  $t$ . Tương tự cho  $t.VB$ ,  $t.VC$ .
- 2) `line l=line(t.AB)`; đường thẳng chứa cạnh  $AB$ .
- 3) `point P=operator cast(t.VA)`; hoặc `point P=point(t.VA)`; mặc định điểm  $P$  tại  $A$ .

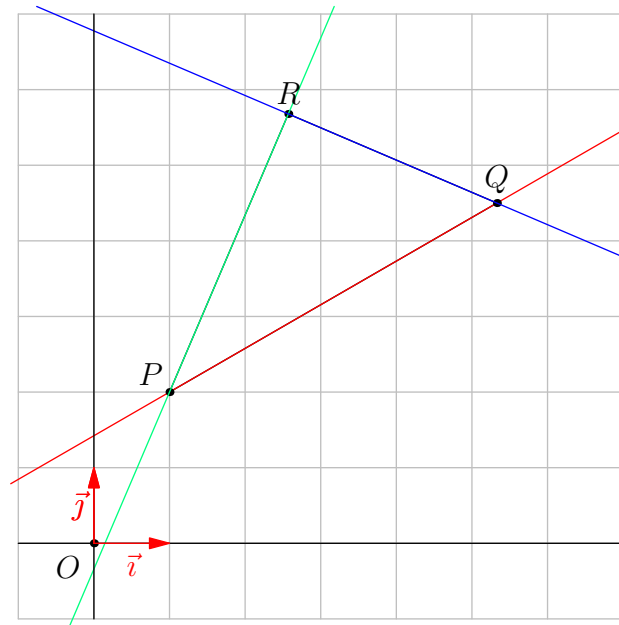
**Ví dụ 37.**



```
import geometry; unitsize(1cm);
triangle t=triangleabc(4,5,6); drawline(t, linewidth(bp));
line ba=bisector(t.VA), bb=bisector(t.VB);
line bc=bisector(t.VC); draw(ba^^bb^^bc);
markangle(line(t.AB), line(t.AC), StickIntervalMarker(2,1));
markangle(line(t.BC), line(t.BA), StickIntervalMarker(2,2));
markangle(line(t.CA), line(t.CB), StickIntervalMarker(2,3));
```

- 4) `circle c=excircle(t.AB)`; đường tròn bàng tiếp tiếp xúc với cạnh  $AB$ .
- 5) `line l=operator cast(t.AB)`; đường thẳng chứa cạnh  $AB$ .
- 6) `segment s=segment(t.AB)`; đoạn thẳng chứa cạnh  $AB$ .
- 7) `side a=opposite(t.VA)`; cạnh đối của đỉnh  $A$ .
- 8) `vertex P=opposite(t.AB)`; đỉnh đối của cạnh  $AB$ .

**Ví dụ 38.**



```
import geometry; import math; unitsize(1cm); linemargin=-.1cm;
add(shift(-1,-1)*grid(8,8,mediumgray));
triangle t=triangleabc(3,4,5,angle=30,(1,2)); draw(t);
line ba=bisector(t.VA),bb=bisector(t.VB);
line l=line(t.AB); draw(l,red);
point P=operator cast(t.VA); dot("$P$",P,NW);
point Q=point(t.VB); dot("$Q$",Q,N);
point R=point(t.VC);dot("$R$",R,N); line m=operator cast(t.AC);
draw(m,springgreen); side a=opposite(t.VA); draw(a,blue);
show(currentcoordsys);
```

## 12.4. Ảnh của tam giác qua phép biến hình

Cú pháp.

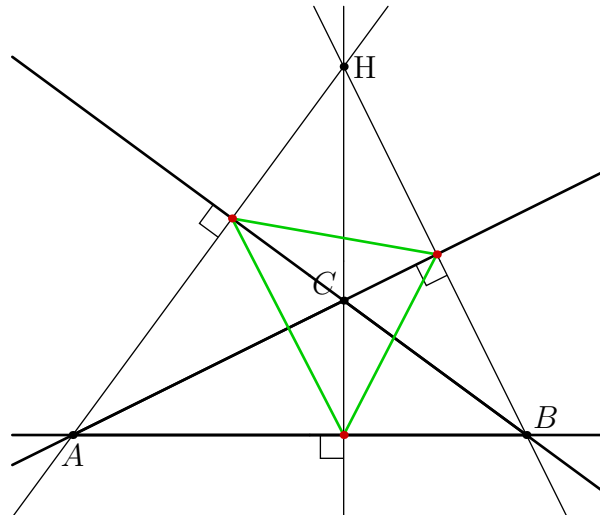
`triangle pt=operator *(p,t);` ảnh của tam giác `t` qua phép biến hình `p`.

## 12.5. Một số lệnh thường dùng trong tam giác

Cú pháp.

- 1) `point H=orthocentercenter(t);`  $H$  là trực tâm của tam giác `t`.
- 2) `point Ah=foot(t.VA);` hoặc `point Ah=foot(t.BC);`  $Ah$  là chân đường cao hạ từ đỉnh  $A$  của tam giác `t`.
- 3) `line ha=altitude(t.VA);` hoặc `line ha=altitude(t.BC);`  $ha$  là đường cao xuất phát từ đỉnh  $A$  của tam giác `t`.
- 4) `triangle tHC=orthic(t);` `tHC` là tam giác có 3 đỉnh là 3 hình chiếu của 3 đỉnh trên 3 cạnh của tam giác `t`.

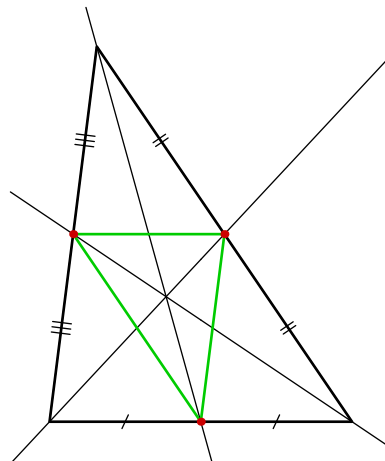
**Ví dụ 39.**



```
import geometry; unitsize(1cm); triangle t=triangleabc(3,4,6);
drawline(t, linewidth(bp));dot(t);
label(LA=Label("$A$",align=S),LB=Label("$B$",align=NE),
LC=Label("$C$",align=NW),t);
line hc=altitude(t.AB), hb=altitude(t.AC); line ha=altitude(t.BC);
draw(hc^^hb^^ha); dot("H", orthocentercenter(t));
perpendicularmark(t.AB,hc,quarter=3);
perpendicularmark(t.AC,hb,quarter=3); perpendicularmark(t.BC,ha);
triangle ort=orthic(t); draw(ort,bp+0.8*green); dot(ort, 0.8*red);
addMargins(.6cm,.6cm); //cạnh lề của hình vẽ
```

- 5) point  $M=\text{midpoint}(t.AB)$ ;  $M$  là trung điểm của cạnh  $AB$ .
- 6) point  $G=\text{centroid}(t)$ ;  $G$  là trọng tâm của tam giác  $t$ .
- 7) line  $ma=\text{median}(t.VA)$ ; hoặc line  $ma=\text{median}(t.BC)$ ;  $ma$  là đường trung bình xuất phát từ đỉnh  $A$  của tam giác  $t$ .
- 8) triangle  $tMC=\text{medial}(t)$ ;  $tMC$  là tam giác có 3 đỉnh là 3 trung điểm của 3 cạnh của tam giác  $t$ .

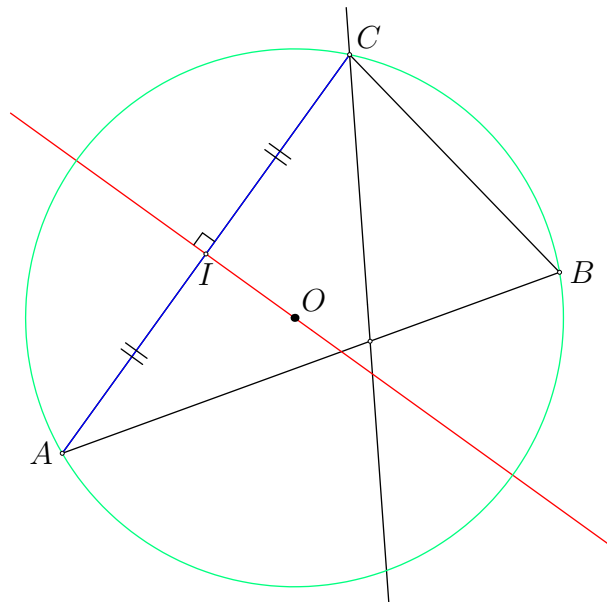
Ví dụ 40.



```
import geometry; unitsize(1cm);
triangle t=triangleabc(6,5,4); drawline(t, linewidth(bp));
line ma=median(t.VA), mb=median(t.VB);
line mc=median(t.VC); draw(ma^~mb^~mc);
draw(segment(t.AB), StickIntervalMarker(2,1,size=2mm,angle=-25));
draw(segment(t.BC), StickIntervalMarker(2,2,size=2mm,space=.6mm));
draw(segment(t.CA), StickIntervalMarker(2,3,size=2.5mm,space=.7mm));
triangle med=medial(t); draw(med,bp+0.8*green); dot(med, 0.8*red);
addMargins(.5cm,.5cm);
```

- 9) `triangle tm=anticomplementary(t)`; `tm` là tam giác chứa 3 đỉnh của tam giác `t` là trung điểm trên 3 cạnh của `tm`.
- 10) `line ba=bisector(t.VA,angle=alpha)`; là ảnh của đường phân giác trong tại đỉnh `t.VA` qua phép quay tâm `t.VA` và góc `alpha`. Mặc định là `alpha=0` và cũng chính là đường phân giác trong của góc tại đỉnh `t.VA`
- 11) `point D=bisectorpoint(t.AB)`; chân đường phân giác trong của góc tại đỉnh `t.VC`
- 12) `line ba=bisector(t.VA)`; `ba` là đường phân giác trong của góc tại đỉnh `t.VA`
- 13) `line d=bisector(t.AB)`; `d` là đường trung trực của cạnh `t.AB`
- 14) `point O=circumcenter(t)`; `O` là tâm đường tròn ngoại tiếp tam giác `t`.
- 15) `circle c=circle(t)`; hoặc `c=circumcircle(t)`; `c` là đường tròn ngoại tiếp tam giác `t`.

Ví dụ 41.

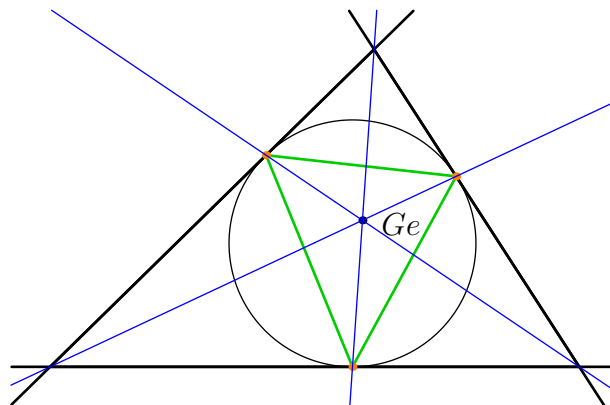




```
import geometry; import math;
unitsize(1cm); linemargin=-0.2cm;
filltype p2=Fill(white); pen p1=linewidth(.3bp);
triangle t=triangleabc(4,6.5,7,angle=20);
draw(t);
point A=point(t.VA), B=point(t.VB),C=point(t.VC);
line bc=bisector(t.VC,angle=0); draw(bc);
point D=bisectorpoint(t.AB);
line bb=bisector(t.CA); draw(bb,red);
draw(A--C,blue,StickIntervalMarker(2,2));
perpendicularmark(t.AC,bb,quarter=1,size=2mm);
point O=circumcenter(t); dot("$O$",O,NE);
draw(circle(t),springgreen); dot("$I$",scale(0.5,A)*C,S,p1,p2);
dot("$A$",A,W,p1,p2); dot("$B$",B,E,p1,p2); dot("$C$",C,NE,p1,p2);
dot(D,p1,p2);
```

- 16) `triangle tp=tangential(t)`; `tp` là tam giác nhận 3 đỉnh là 3 giao điểm của 3 tiếp tuyến tại 3 đỉnh của đường tròn ngoại tiếp tam giác `t`.
- 17) `point I=incenter(t)`; `I` là tâm đường tròn nội tiếp tam giác `t`.
- 18) `circle c=incircle(t)`; `c` là đường tròn nội tiếp tam giác `t`.
- 19) `real r=inradius(t)`; `r` là bán kính đường tròn nội tiếp tam giác `t`.
- 20) `triangle tg=intouch(t)`; `tg` là tam giác nhận 3 tiếp điểm của đường tròn nội tiếp với 3 cạnh của tam giác `t` làm 3 đỉnh.
- 21) `point T=intouch(t.AB)`; `T` là tiếp điểm của đường tròn nội tiếp với cạnh `AB` của tam giác `t`.
- 22) `point P=gergonne(t)`; `P` là điểm đối trung của tam giác nhận 3 tiếp điểm của đường tròn nội tiếp làm 3 đỉnh. Điểm đối trung là giao điểm của 3 đường thẳng đối xứng của 3 đường trung tuyến qua 3 đường phân giác trong.

#### Ví dụ 42.



```
import geometry; unitsize(1cm);
triangle t=triangleabc(5,6,7); drawline(t, linewidth(bp));
draw(incircle(t)); // vẽ đường tròn nội tiếp tam giác t.
triangle itr=intouch(t); draw(itr,bp+0.8*green); //đn và vẽ tam giác
//nhận 3 tiếp điểm là 3 đỉnh
dot(itr,orange); // vẽ 3 đỉnh của tam giác itr
point Ge=gergonne(t); // Ge là điểm đối trung của tam giác t.
dot("$Ge$", Ge, 2*dir(-8)); // vẽ và gán nhãn cho điểm đối trung
draw(line(Ge,t.A), blue); draw(line(Ge,t.B),blue);
draw(line(Ge,t.C), blue); // vẽ 3 đường đối trung
addMargins(.5cm,.5cm);
```

23) circle c=excircle(t.AB); c là đường tròn bàng tiếp tiếp xúc với cạnh  $AB$ .

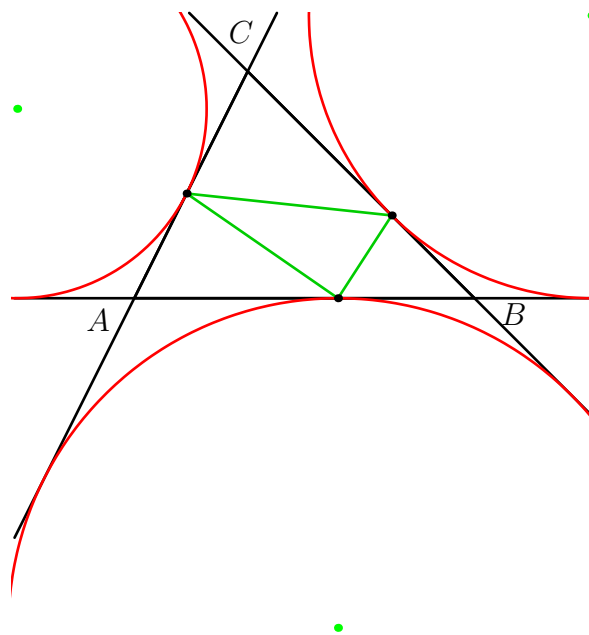
24) point I=excenter(t.AB); I là tâm của đường tròn bàng tiếp tiếp xúc với cạnh  $AB$ .

25) real r=exradius(t.AB); r là bán kính của đường tròn bàng tiếp tiếp xúc với cạnh  $AB$ .

26) triangle txo=extouch(t); txo là tam giác nhận 3 tiếp điểm của 3 đường tròn bàng tiếp làm 3 đỉnh của tam giác t.

27) point P=extouch(t.AB); P là tiếp điểm của đường tròn bàng tiếp với cạnh  $AB$ .

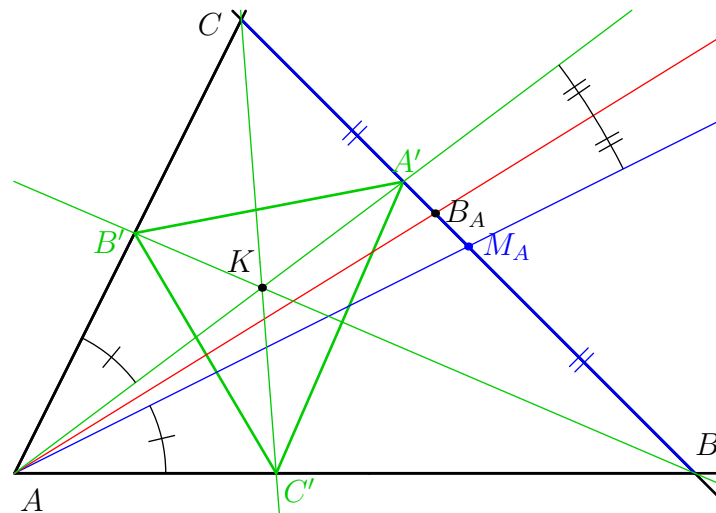
**Ví dụ 43.**



```
import geometry; unitsize(1cm);
triangle t=triangleabc(5,6,7); drawline(t, linewidth(bp));
draw(incircle(t)); // vẽ đường tròn nội tiếp tam giác t.
triangle itr=intouch(t); draw(itr,bp+0.8*green); //đn và vẽ tam giác
//nhận 3 tiếp điểm là 3 đỉnh
dot(itr,orange); // vẽ 3 đỉnh của tam giác itr
point Ge=gergonne(t); // Ge là điểm đối trung của tam giác t.
dot("$Ge$", Ge, 2*dir(-8)); // vẽ và gán nhãn cho điểm đối trung
draw(line(Ge,t.A), blue); draw(line(Ge,t.B),blue);
draw(line(Ge,t.C), blue); // vẽ 3 đường đối trung
addMargins(.5cm,.5cm);
```

- 28) `triangle st=symmedial(t)`; tam giác đối trung của  $t$ , là tam giác nhận 3 đỉnh là 3 điểm đối xứng của 3 trung điểm của 3 cạnh qua 3 chân đường phân giác trong của tam giác  $t$ .
- 29) `point P=symmedian(t)`;  $P$  là điểm đối xứng của trọng tâm qua tâm đường tròn nội tiếp tam giác  $t$ .
- 30) `point Q=symmedian(t.AB)`;  $Q$  là điểm đối trung trên cạnh  $AB$ .
- 31) `line l=symmedian(t.VA)`; đường đối trung xuất phát từ  $A$ .

Ví dụ 44.



```
import geometry; unitsize(3cm);
triangle t=triangle((-1,0), (2,0), (0,2));
drawline(t, linewidth(bp)); // vẽ tam giác t theo dạng đường thẳng
label(t,alignFactor=2, alignAngle=90); //Gắn nhãn
triangle st=symmedial(t); //Tam giác đối trung
draw(st, bp+0.8green); // Vẽ tam giác đối trung
label("$A'$", "$B'$", "$C'", st, alignAngle=45, 0.8green); // Nhãn
line mA=median(t.VA); draw(mA, blue); //đường trung bình
dot("$M_A$",midpoint(t.BC), 1.5E, blue); //Trung điểm M_A
draw(segment(t.BC), bp+blue, StickIntervalMarker(2,2,blue));
line bA=bisector(t.VA); draw(bA, grey); //Đường phân giác
dot("$B_A$", bisectorpoint(t.BC)); // Chân đường phân giác B_A
line sA=symmedian(t.VA); draw(sA, 0.8*green); // đường đối trung từ t.VA
draw(symmedian(t.VB), 0.8*green); // từ t.VB
draw(symmedian(t.VC), 0.8*green); // từ t.VC
point sP=symmedian(t); dot("$K$", sP, 2*dir(125)); // Điểm đối trung.
markangle(sA, (line) t.AC, radius=2cm, StickIntervalMarker(1,1));
markangle((line) t.AB, mA, radius=2cm, StickIntervalMarker(1,1));
markangle(mA, sA, radius=10cm, StickIntervalMarker(2,2));
```

## 13. Tô màu – lát nền

### 13.1. Lát nền

Trong phần mềm asymptote cung cấp gói lệnh `patterns` dùng để lát nền của một đường khép kín ở dạng `path`. Tổng cộng có 5 kiểu nền được cung cấp trong gói này.

- 1) `tile( )`; kiểu lưới mặc định là ô vuông cạnh 5 mm, màu lưới là màu đen hay `tile(4mm,gray)` thì có kiểu lưới cạnh 4mm và màu lưới là gray.
- 2) `checker( )`; kiểu caro mặc định ô vuông cạnh 5 mm và ô trắng đen hay `checker(4mm,red)` thì có kiểu caro cạnh 4mm và ô màu đen tô thành ô đỏ.
- 3) `brick( )`; kiểu gạch lát tường mặc định hình chữ nhật dài 5mm, rộng 2.5mm
- 4) `hatch( )`; kiểu sọc chéo mặc định là hướng đông bắc (NE) và khoảng cách giữa các đường thẳng song song là 5mm. Hay có thể thay đổi `hatch(4mm,NW)` khoảng cách giữa các sọc là 4 mm, theo hướng tây bắc.
- 5) `crosshatch( )`; kiểu lưới chéo mặc định là ô vuông cạnh 5mm. Hay có thể thay đổi `crosshatch(3mm)` thành ô vuông có cạnh 3mm.

### Cách sử dụng

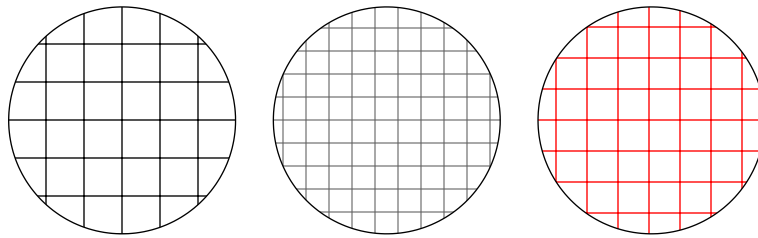
`import patterns`; khai báo gói.

`add("tên kiểu nền", kiểu nền)`; đặt tên và thêm kiểu nền (1 trong 5 kiểu).

`filldraw(đường khép kín, pattern("tên kiểu nền"))`; lát nền cho đường khép kín đã định nghĩa.

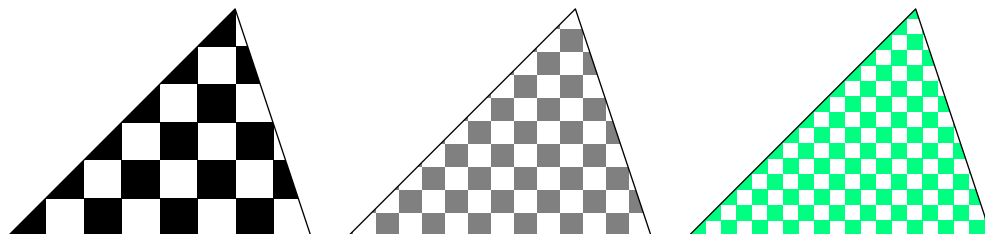
hoặc `fill(đường khép kín, pattern("tên kiểu nền"))`; lát nền nhưng không vẽ đường viền.

#### Ví dụ 45. Kiểu tile.



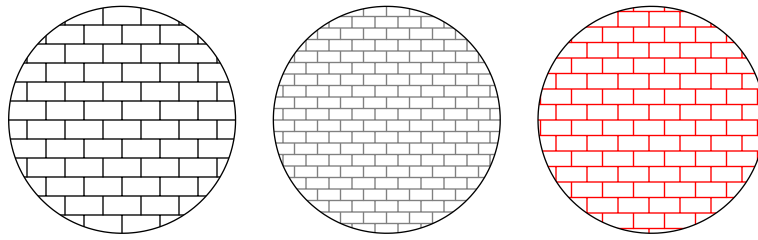
```
import geometry; import patterns; unitsize(1cm);
point O=origin; path c1=circle(O,1.5);
add("tile01",tile()); add("tile02",tile(3mm,gray));
add("tile03",tile(3mm,red)); filldraw(c1,pattern("tile01"));
filldraw(shift(3.5,0)*c1,pattern("tile02"));
filldraw(shift(7,0)*c1,pattern("tile03"));
```

#### Ví dụ 46. Kiểu checker.



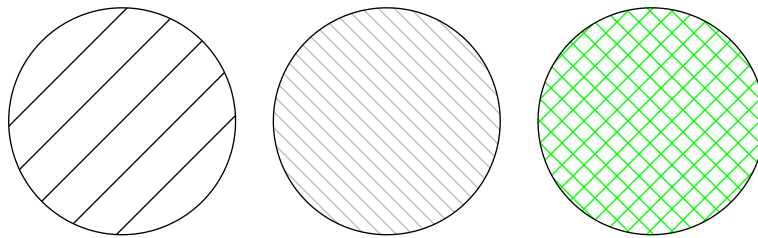
```
import geometry; import patterns; unitsize(1cm); point A=(0,0),B=(4,0),C=(3,3);
path c1=A--B--C--cycle; //đường đóng nối các đỉnh A,B,C
add("ck01",checker()); add("ck02",checker(3mm,gray));
add("ck03",checker(2mm,springgreen)); filldraw(c1,pattern("ck01"));
filldraw(shift(4.5,0)*c1,pattern("ck02")); filldraw(shift(9,0)*c1,pattern("ck03"));
```

**Ví dụ 47.** Kiểu brick.



```
import geometry; import patterns; unitsize(1cm);
point O=origin; path c1=circle(O,1.5);
add("br01",brick()); add("br02",brick(3mm,gray));
add("br03",brick(4mm,red)); filldraw(c1,pattern("br01"));
filldraw(shift(3.5,0)*c1,pattern("br02"));
filldraw(shift(7,0)*c1,pattern("br03"));
```

**Ví dụ 48.** Kiểu hatch và crosshatch.



```
import geometry; import patterns; unitsize(1cm);
point O=origin; path c1=circle(O,1.5);
add("h01",hatch()); add("h02",hatch(1.5mm,NW,mediumgray));
add("crh",crosshatch(2mm,green)); filldraw(c1,pattern("h01"));
filldraw(shift(3.5,0)*c1,pattern("h02"));
filldraw(shift(7,0)*c1,pattern("crh"));
```

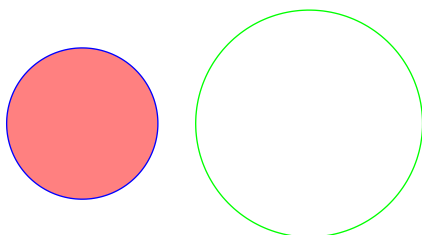
## 13.2. Tô màu

### Cú pháp.

Trong gói geometry không hỗ trợ tô màu nền cho đường kín. Do đó ta phải chuyển đường kín về dạng path đóng khi đó thì Asymptote sẽ hỗ trợ tô màu cho nền.

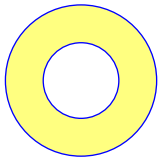
- 1) fill(đường kín,màu); tô màu cho path kín.
- 2) filldraw(đường kín, màu nền, màu viền); tô màu cho nền và viền.

**Ví dụ 49.** Tô màu cho hình tròn.



```
import geometry;
unitsize(1cm); point O=(2,0);
circle c1=circle(O,1.5);
draw(c1,green);
point I=(-1,0);
circle c2 = circle (I,1);
path c3=c2;
filldraw(c3,red+opacity(0.5),blue);
```

**Ví dụ 50.** Tô màu cho hình vành khăn.



```
unitsize(1cm);
path p1 = scale(2)*unitcircle;
path p2 = scale(1)*unitcircle;
filldraw(p1^^p2,evenodd+yellow+white,blue);
```

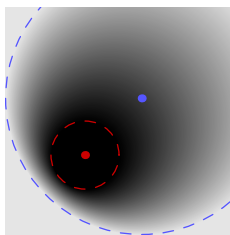
### 13.3. Phối màu

#### 13.3.1. Đồ bóng – 3D

##### Cú pháp.

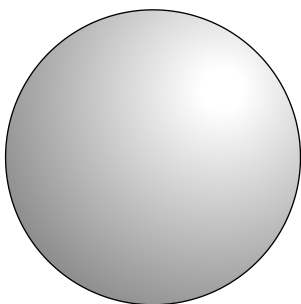
`radialshade(path đóng, màu trong, tâm 1, bán kính 1, màu ngoài, tâm 2, bán kính 2);` tạo hiệu ứng chói mặt trời pha trộn màu từ đường tròn tâm 1, bán kính 1 ra đường tròn tâm 2, bán kính 2.

**Ví dụ 51.** Ví dụ lấy trên trang piprim.fr



```
import geometry; unitsize(4cm);
point A = (0.35,0.35), B = (0.6,0.6);
real r1 = 0.15, real r2 = 0.6;
path hv = unitsquare; //path hình vuông đóng
radialshade(hv, black, A,r1, lightgrey, B, r2);
dot(A,.8red); // tâm của đường tròn trong
dot(B,blue+grey); //tâm của đường tròn ngoài
//Vẽ đường tròn trong
draw(shift(A)*scale(r1)*unitcircle,dashed+0.8red);
//vẽ đường tròn ngoài
draw(shift(B)*scale(r2)*unitcircle,dashed+blue+grey);
clip(unitsquare); //cắt khung hình vuông
```

**Ví dụ 52.** Tạo hiệu ứng 3d.

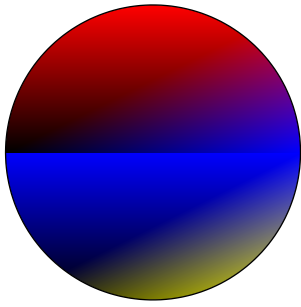


```
import geometry;
unitsize(1.3cm);
point T1 =(0.7,0.7);
point T2 = (0.6,0.6);
point T3 = (0,0);
real r1 = 0.2;
real r2 = 2.6;
real r3 = 1.5;
path c3 = circle(T3,r3);
radialshade(c3,white,T1,r1,gray,T2,r2);
draw(c3);
```

#### 13.3.2. Pha trộn nhiều màu

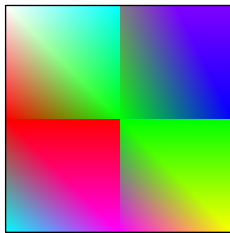
##### Cú pháp.

- 1) `pen [] p ={{màu 1}, {màu 2},...};`
- 2) `latticeshade(path đóng,p);` tô màu cho path đóng.



```
import geometry;
unitsize(1.3cm);
point O = (0,0);
real r = 1.5;
path c= circle(O,r);
pen [] p = {{red}, {blue}, {yellow}};
latticeshade(c,p);
draw(c);
```

**Ví dụ 53.** Trong tập màu chứa nhiều màu. Xem ví dụ lấy từ trang piprim.fr



```
import geometry;
unitsize(3cm);
pen[] [] p={{white,grey,black},
            {red,green,blue},
            {cyan,magenta,yellow}};
latticeshade(unitsquare,p);
draw(unitsquare);
```

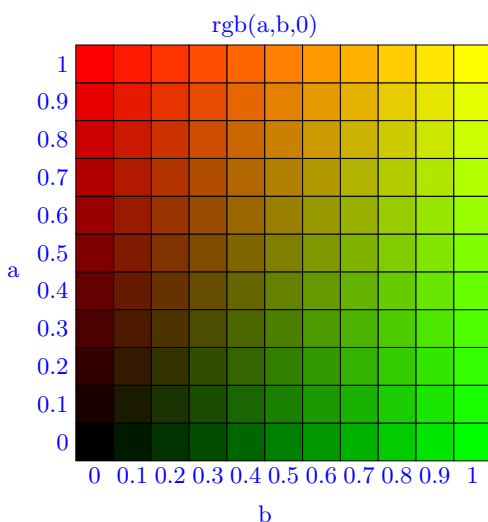
### 13.3.3. Tạo màu mới chuẩn rgb

Ngoài các màu mặc định, Asymptote còn cho phép chúng ta tạo màu mới theo chuẩn rgb (đỏ – xanh da trời – xanh biển). Tức là phối 3 màu trên để tạo thành màu mới.

#### Cú pháp

`pen p=rgb(a,b,c);` trong đó  $a, b, c$  là 3 số thuộc đoạn  $[0; 1]$ .

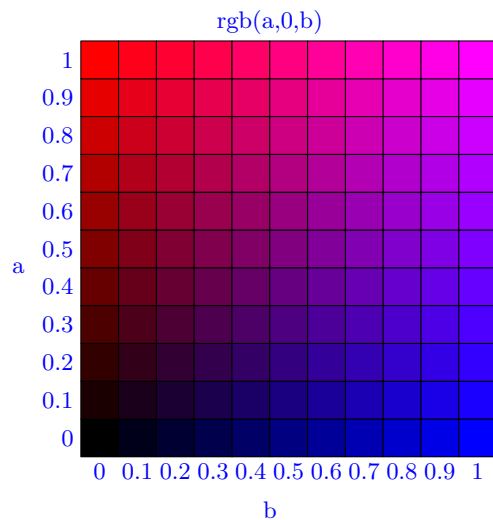
**Ví dụ 54.** Chuẩn màu dạng `rgb(a,b,0)`. Được lấy trên trang asy.marris.fr



```
unitsize(.6cm);
defaultpen(fontsize(10pt)+blue);
int n=10;
for (int i = 0; i <= n; ++i) {
    for (int j = 0; j <= n; ++j) {
        pen p =rgb(i/n,j/n,0);
        filldraw(shift(j,i)*unitsquare,p,0.2bp+black);
        if(i==0) label(string(j/n),(j+.5,0),S);
    }
    label(string(i/n),(0,i+.5),W);
}
label("rgb(a,b,0)",(n/2,n+1),N);
label("b", (n/2,-1),S);
label("a", (-1.3,n/2),W);
```

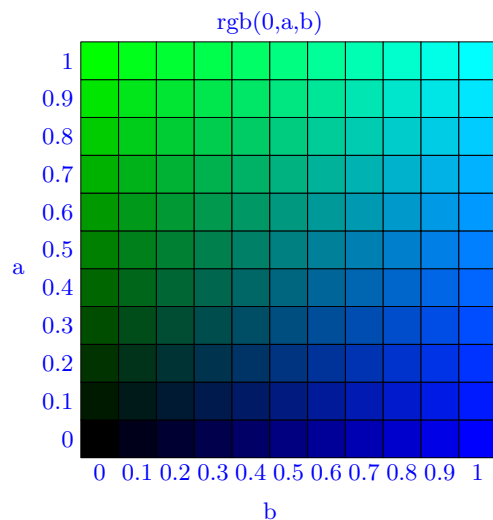
Ở ví dụ trên, nếu ta chọn `pen p=rgb(1,0.9,0);` thì ta được  $p$  là màu vàng sẫm.

**Ví dụ 55.** Chuẩn màu dạng `rgb(a,0,b)`. Được lấy trên trang asy.marris.fr



```
unitsize(.5cm);
defaultpen(fontsize(9pt)+blue);
int n=10;
for (int i = 0; i <= n; ++i) {
    for (int j = 0; j <= n; ++j) {
        pen p =rgb(i/n,0,j/n);
        filldraw(shift(j,i)*unitsquare,p,.2bp+black);
        if(i==0) label(string(j/n),(j+.5,0),S);
    }
    label(string(i/n),(0,i+.5),W);
}
label("rgb(a,0,b)",(n/2,n+1),N);
label("b", (n/2,-1),S);
label("a", (-1.3,n/2),W);
```

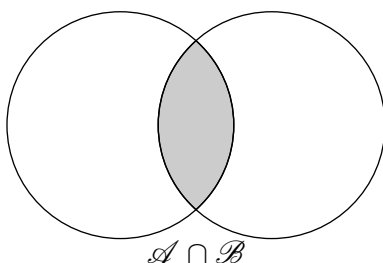
**Ví dụ 56.** Chuẩn màu dạng  $\text{rgb}(0,a,b)$ . Được lấy trên trang [asy.marris.fr](http://asy.marris.fr)



```
unitsize(.5cm);
defaultpen(fontsize(9pt)+blue);
int n=10;
for (int i = 0; i <= n; ++i) {
    for (int j = 0; j <= n; ++j) {
        pen p =rgb(0,i/n,j/n);
        filldraw(shift(j,i)*unitsquare,p,.2bp+black);
        if(i==0) label(string(j/n),(j+.5,0),S);
    }
    label(string(i/n),(0,i+.5),W);
}
label("rgb(0,a,b)",(n/2,n+1),N);
label("b", (n/2,-1),S);
label("a", (-1.3,n/2),W);
```

### 13.3.4. Biểu đồ Ven

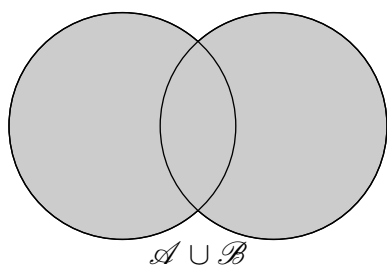
**Ví dụ 57.** Giao của hai tập hợp.



```
import geometry;
texpreable("\usepackage{mathrsfs}");// tạo chữ 'Hoa'
texpreable("\usepackage{maybemath}");// in đậm
unitsize(1cm); point I = (0,-1.4);
path c1=shift(1,0)*scale(1.5)*unitcircle;
path c2=shift(-2,0)*c1;
path c3=buildcycle(c1,c2);
filldraw(c3,gray+opacity(0.8));
draw(c1); draw(c2);
label("$\maybebm{\mathscr{A}}\cap \mathscr{B}$",I,S);
```

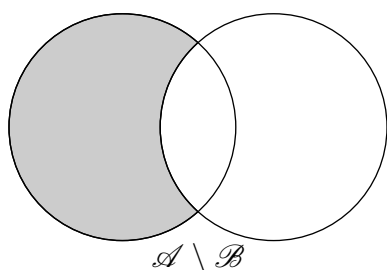
**Ví dụ 58.** Hợp của hai tập hợp.





```
import geometry;
texpreable("\usepackage{mathrsfs}");
texpreable("\usepackage{maybemath}");
unitsize(1cm); point I = (0,-1.4);
path c1=shift(-1,0)*scale(1.5)*unitcircle;
path c2=shift(2,0)*c1;
path c3=buildcycle(c1,c2);
filldraw(c3,gray+opacity(0.4));
draw(c1); draw(c2);
label("$\maybebm{\mathscr{A}}\cup \mathscr{B}$",I,S);
```

**Ví dụ 59.** Hiệu của hai tập hợp.



```
import geometry;
texpreable("\usepackage{mathrsfs}");
texpreable("\usepackage{maybemath}");
unitsize(1cm); point I = (0,-1.4);
path c1=shift(-1,0)*scale(1.5)*unitcircle;
path c2=reverse(shift(2,0)*c1);
path c4=buildcycle(c1,c2);
filldraw(c4,gray+opacity(0.4));
draw(c1); draw(c2);
label("$\maybebm{\mathscr{A}}\setminus \mathscr{B}$",I,S);
```

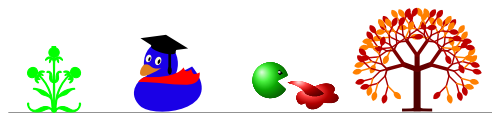
## 14. Bổ sung gói mới từ L<sup>A</sup>T<sub>E</sub>X

Ngôn ngữ lập trình Asymptote hỗ trợ gói ngôn ngữ và font cho ký hiệu toán học từ L<sup>A</sup>T<sub>E</sub>X để gắn nhãn. Nó còn hỗ trợ việc bổ sung một số gói khác để chèn ký tự, hình vẽ nhưng với lệnh đơn. Ví dụ như gói: fontawesome, pgfornament, tikzducks, tikz và một số thư viện kèm theo gói.

### Cú pháp.

`texpreable("\usepackage[option]{name package}");` Việc thêm gói gần giống như ta thêm gói trong sử dụng L<sup>A</sup>T<sub>E</sub>X.

**Ví dụ 60.** Chèn các ký hiệu, họa tiết từ một số gói trong L<sup>A</sup>T<sub>E</sub>X.

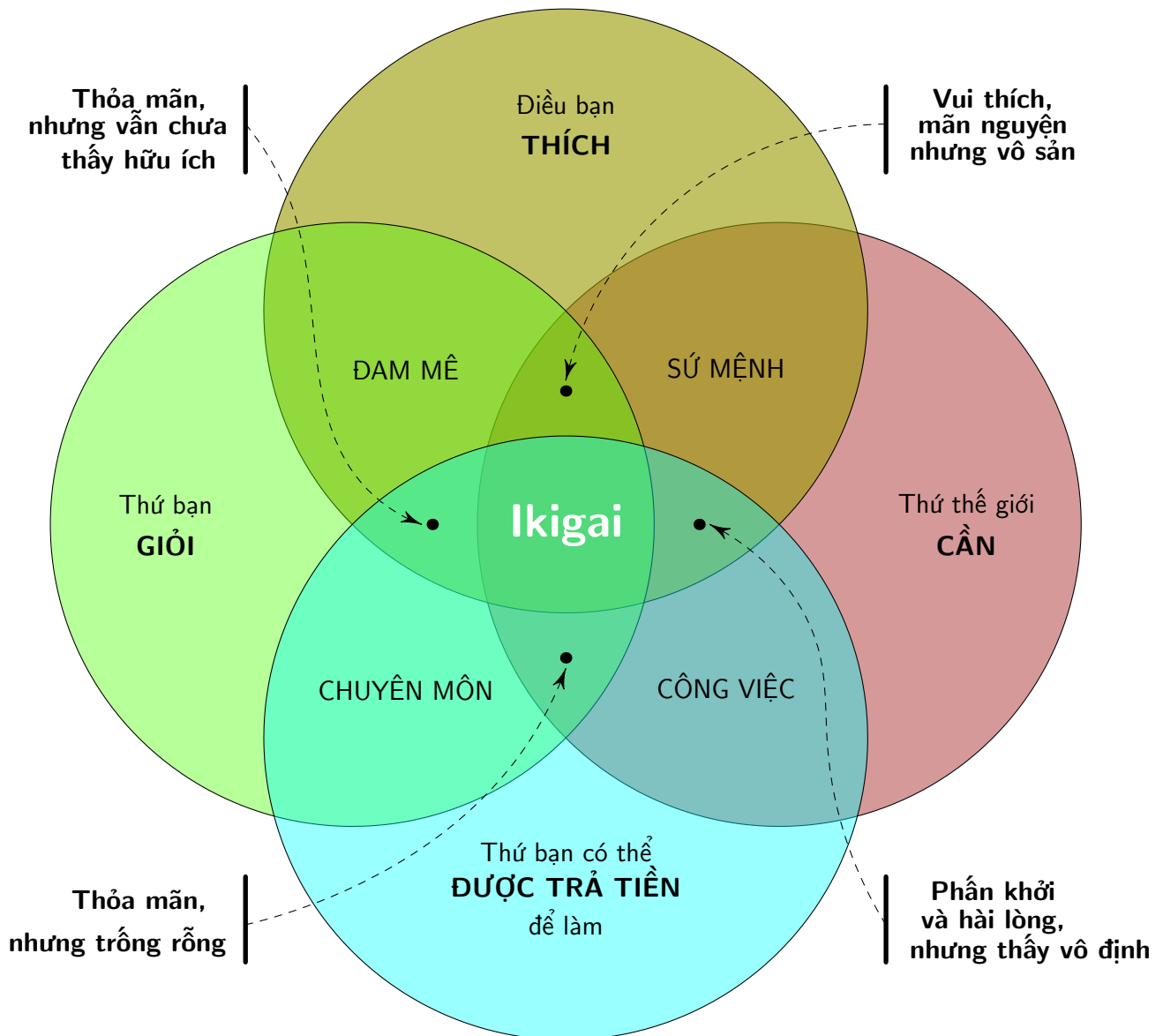


```
unitsize(1cm); //
pen mauto=rgb(0,1,0); // Màu mới
texpreable("\usepackage{fontawesome}");
texpreable("\usepackage{pgfornament}"); // bông hoa
texpreable("\usepackage{tikzducks}"); // Chú vịt
texpreable("\usepackage{tikzsymbols}"); //Hét ra lửa, cây
texpreable("\usetikzlibrary{arrows}"); //tikz
label("\pgfornament[scale=0.25]{10}",(1.5,0),0.1*N,mauto);
label(scale(0.45)*Label("\tikz \duck[body=red!10!blue,
cape=red,graduate,tassel];"),(3,0),0.2*N);
label(scale(0.45)*"\dVomey[5][green][red]",(5,0),0.2*N);
label("\Autumntree[5]",(7,0),0.2*N);
draw((0.6,0)--(1.6*5,0));
```

**Ví dụ 61.** Vòng tròn sinh mệnh.

# Ikigai

## QUAN NIỆM CỦA NGƯỜI NHẬT VỀ "LỄ SỐNG"



```

import geometry;
texpreable("\usepackage[utf8]{vietnam}"); // gói tiếng việt
texpreable("\renewcommand{\familydefault}{\sfdefault}"); // mặc định kiểu font
defaultpen(fontsize(8pt)); unitsize(1cm); dotfactor*=2;
real a=4; real r=a*sqrt(2); // bán kính của 4 đường tròn
point O=(0,0); path c[];
for(int i=1;i<=4;++i){
  if (i==1){c[i]=circle((point)(a,0),r);}
  else {c[i]=rotate((i-1)*90,0)*c[1];}
}
filldraw(c[1],0.6*red+opacity(.4));
filldraw(c[2],0.6*yellow+opacity(.6));
filldraw(c[3],green+0.4*yellow+opacity(.4));
filldraw(c[4],cyan+opacity(.4));
label("\Huge \bf{ Ikigai}",(0,11.5),N);
label("\large \bf{QUAN NIỆM CỦA NGƯỜI NHẬT VỀ 'LỄ SỐNG' }", (0,10.5),N);
label("\large Thứ thế giới", (7.5,0),1.5*N);
label("\large \bf CẦN", (7.5,-0.7),1.5*N);
label("\large Điều bạn", (0,7.5),1.5*N);
label("\large \bf THÍCH", (0,6.8),1.5*N);
label("\large Thứ bạn", (-7.5,0),1.5*N);
label("\large \bf GIỎI", (-7.5,-0.7),1.5*N);
label("\large Thứ bạn có thể", (0,-6.5),1.5*N);
label("\large \bf ĐƯỢC TRẢ TIỀN", (0,-7.2),1.5*N);
label("\large để làm", (0,-7.8),1.5*N);
label("\huge \bf Ikigai", (0,0),white);
label("\large SỰ MỆNH", (3,3)); label("\large ĐAM MÊ", (-3,3));
label("\large CHUYỀN MÔN", (-3,-3)); label("\large CÔNG VIỆC", (3,-3));
dot((2.5,0)--(0,2.5)--(-2.5,0)--(0,-2.5));
//===== NE
draw((6,6.6)--(6,8.2),linewidth(2.5bp));
label("\large \bf Vui thích", (8,8)); label("\large \bf mãn nguyện", (8,7.5));
label("\large \bf nhưng vô sản", (8,7));
//===== NW
draw((-6,6.6)--(-6,8.2),linewidth(2.5bp));
label("\large \bf Thỏa mãn", (-8,8)); label("\large \bf nhưng vẫn chưa", (-8.2,7.5));
label("\large \bf thấy hữu ích", (-8,6.85));
//=====SW
draw((-6,-6.6)--(-6,-8.2),linewidth(2.5bp)); label("\large \bf Thỏa mãn", (-8,-7));
label("\large \bf nhưng trống rỗng", (-8.4,-7.8));
//=====SN
draw((6,-6.6)--(6,-8.2),linewidth(2.5bp));
label("\large \bf Phấn khởi", (8,-6.8)); label("\large \bf và hài lòng", (8,-7.4));
label("\large \bf nhưng thấy vô định", (8.7,-7.95));
//===== NE
path p1=(0,2.7)..(2,6)..(6,7.5); draw(p1,dashed,BeginArrow(HookHead));
path p2=(-2.7,0)..(-4,1)..(-4.66,3.05)..(-5.09,5.84)..(-6,7.5); draw(p2,dashed,BeginArrow(HookHead));
//===== SW
path p3=(0,-2.7)..(-1,-4.5)..(-4,-6.8)..(-6,-7.5); draw(p3,dashed,BeginArrow(HookHead));
//===== SE
path p4=(2.7,0)..(4,-1)..(4.75,-2.4)..(4.9,-4.1)..(6,-7.5); draw(p4,dashed,BeginArrow(HookHead));
shipout(bbox(2mm, 1mm+white));

```

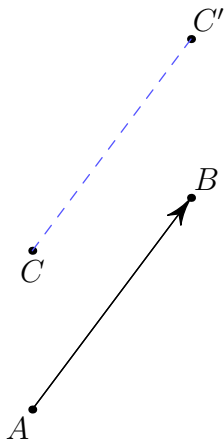
## 15. Các phép biến hình phẳng

### Phép tịnh tiến

#### Cú pháp.

`transform t=shift(v)`; phép tịnh tiến theo vectơ  $v$  đã được định nghĩa hoặc đã có tọa độ.

#### Ví dụ 62.



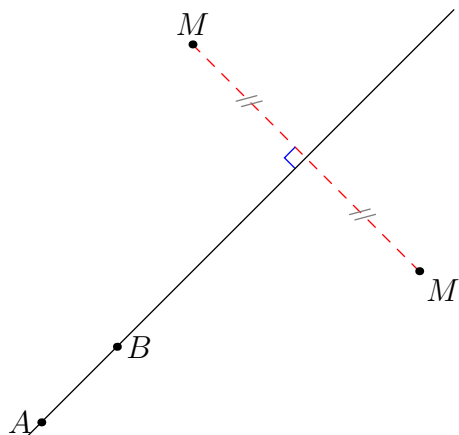
```
import geometry;
unitsize(1cm);
point A=(-1,-1), B=(2,3), C=(-1,2);
dot("$A$",A,dir(B--A));
dot("$B$",B,dir(A--B));
dot("$C$",C,S);
draw(segment(A,B),Arrow(HookHead));
vector v=B-A; // vectơ AB
transform t=shift(v);
point Cz=t*C;
dot("$C'$",Cz,dir(C--Cz));
draw(C--Cz,dashed+gray+blue);
```

### Phép đối xứng trục

#### Cú pháp.

`transform p=reflect(l)`; phép đối xứng qua đường thẳng  $l$ .

#### Ví dụ 63.



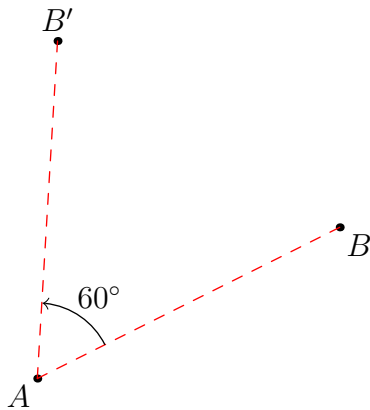
```
import geometry; unitsize(1cm);
point A=origin, B=(1,1), M=(2,5); // điểm A=(0,0) gốc tọa độ 0.
line l=line(A,B); transform re=reflect(l); point Mp=re*M;
draw(M--Mp,dashed+red,StickIntervalMarker(2,2,8,angle=-30,gray));
perpendicularmark(l,line(M,Mp),quarter=2,blue,size=2mm);
draw(l); dot("$A$",A,W); dot("$B$",B,E);
dot("$M$",M,N); dot("$M'$",Mp,dir(M--Mp));
```

## Phép quay

### Cú pháp.

`transform q=rotate(angle,0);` phép quay tâm  $O$ , góc  $angle$  (độ).

Ví dụ 64.



```
import geometry; unitsize(1cm);
DefaultHead=TeXHead; point A=(0,0),B=(4,2);
dot("$A$",A,SW);
dot("$B$",B,SE);
transform q=rotate(60,A);
point Bq=q*B; //định nghĩa ảnh của B qua phép quay.
dot("$B'$",Bq,N);
draw(A--B--A--Bq,dashed+red);//^^ vẽ cùng lúc AB,AB'
markangle("$60^\circ$",B,A,Bq,Arrow());
```

## Phép đối xứng tâm

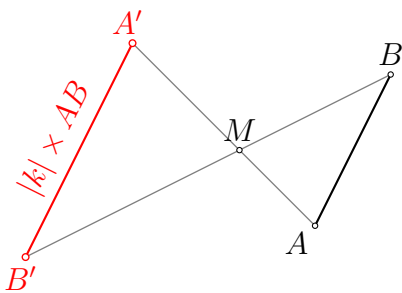
Phép đối xứng tâm  $O$  chính là phép quay tâm  $O$ , góc  $\pm 180^\circ$ .

## Phép vị tự

### Cú pháp.

`transform sc=scale(real k, point M);` phép vị tự tâm  $M$ , tỷ số  $k$  với  $k \neq 0$ .

Ví dụ 65. Xem kỹ ví dụ này.



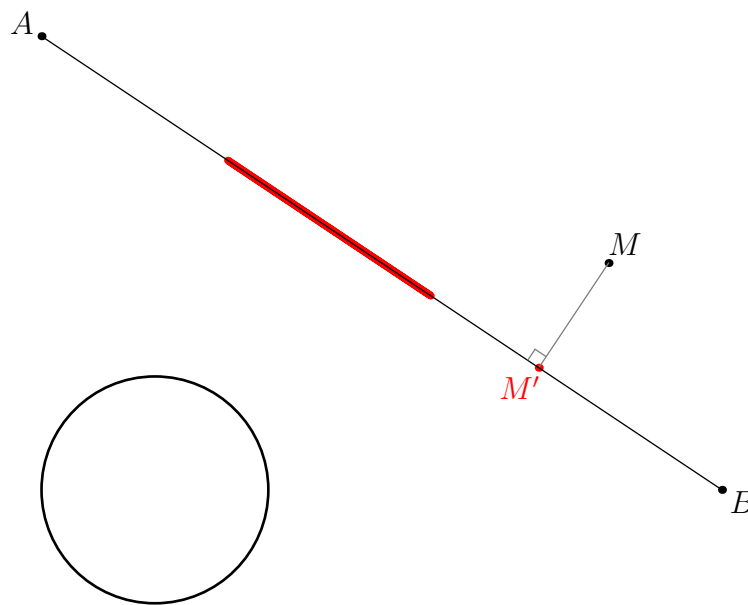
```
import geometry; unitsize(1cm);
pen p1=linewidth(.8bp),p2=linewidth(.4bp);
filltype p3=Fill(white);
real k=-sqrt(2); // khai báo số k
point A=(0,0),B=(1,2),M=(-1,1);
transform sc=scale(k, M);
point Av=sc*A, Bv=sc*B; //định nghĩa ảnh của A,B qua v
draw(A--Av, grey); // Vẽ đoạn thẳng AA'
draw(B--Bv, grey); // draw(A--B) vẽ đoạn AB mà
draw(A--B, p1); //không phải định nghĩa trước
draw(rotate(unit(Bv-Av))*"$\left|k\right|\times AB$",
    Av--Bv, p1+red); // nhãn cho đoạn A'B'
dot("$A$",A, SW,p2,p3);
dot("$B$",B,N,p2,p3);
dot("$M$",M,N,p2,p3);
dot("$A'$",Av,N,red,p3);
dot("$B'$",Bv,S,red,p3);
```

## Phép chiếu vuông góc

### Cú pháp.

`transform proj=projection(A,B);` phép chiếu vuông góc lên đường thẳng  $AB$  với  $A, B$  đã định nghĩa.

**Ví dụ 66.**



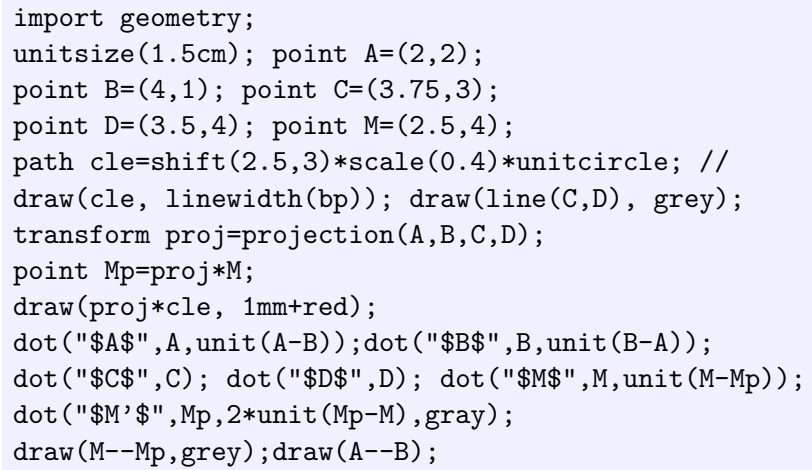
```
import geometry; unitsize(1.5cm);
point A=(-1,4); point B=(5,0); point M=(4,2);
path cle=unitcircle;// path đường tròn đơn vị
draw(cle, linewidth(bp)); // vẽ đường tròn.
transform proj=projection(A,B); //
point Mp=proj*M;
draw(proj*cle, 1mm+red); // 1 mm độ dày
dot("$A$", A, unit(A-B)); dot("$B$", B, unit(B-A));
dot("$M$", M, unit(M-Mp));
dot("$M'$", Mp, unit(Mp-M), red);
draw(M--Mp, grey); draw(A--B);
markrightangle(M,Mp,A, grey,size=5);
```

**Phép chiếu song song lên đường thẳng**

**Cú pháp.**

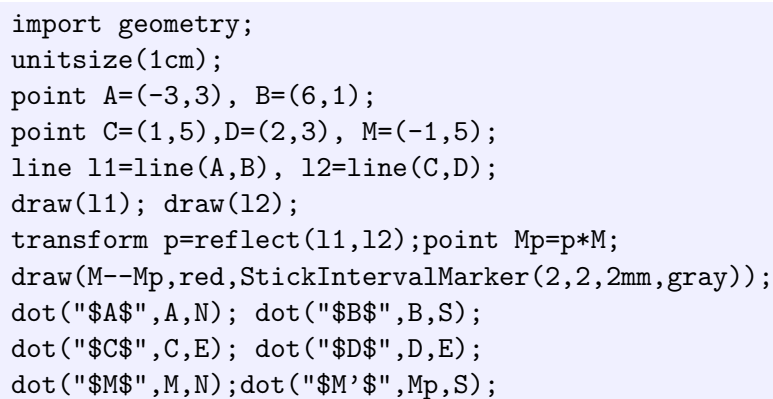
`transform proj=projection(point A, point B, point C, point D);` phép chiếu song song lên đường thẳng  $AB$  theo phương là đường thẳng  $CD$  với  $A, B, C, D$  đã được định nghĩa. Nếu  $AB, CD$  song song với nhau thì phép chiếu này là phép đồng nhất.

**Ví dụ 67.** Xem kỹ ví dụ này.



Cú pháp.

**Ví dụ 68.** Xem kỹ ví dụ này.

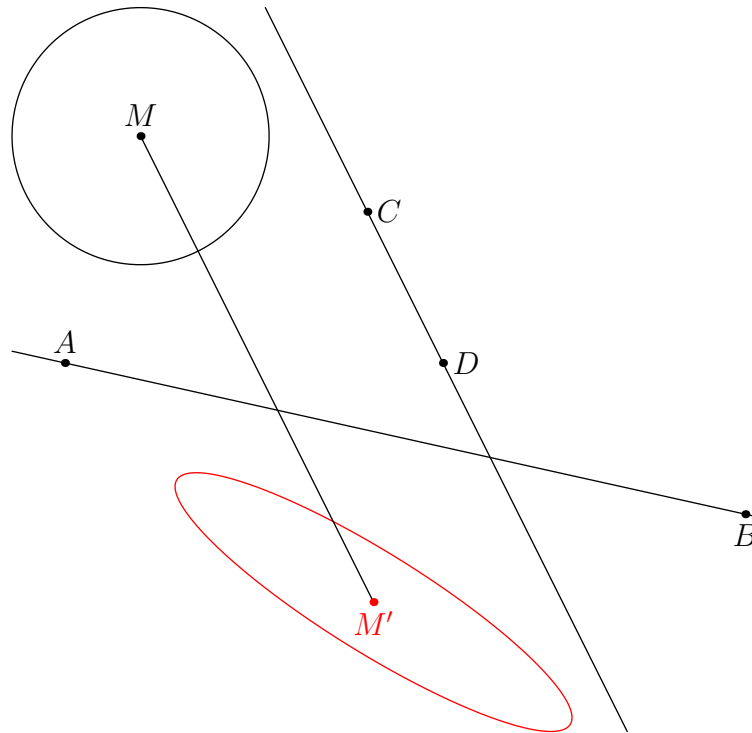


## Phép co – giãn theo phương

### Cú pháp.

`transform p=scale(số k, l1, l2)`; phép biến hình liên tục từ phép đối xứng trục  $l1$  theo phương  $l2$  và phép vị tự tỷ số  $k$ . Nếu  $l1, l2$  song song thì nó là phép đồng nhất.

**Ví dụ 69.** Xem ví dụ này.



```
import geometry; unitsize(1cm); point A=(-3,3), B=(6,1);
point C=(1,5), D=(2,3), M=(-2,6); line l1=line(A,B), l2=line(C,D);
draw(l1); draw(l2); path c=circle(M,1.7);
transform p=scale(-0.7,l1,l2); point Mp=p*M; path cp=p*c;
draw(c); draw(cp,red); draw(M--Mp);
dot("$A$",A,N); dot("$B$",B,S); dot("$C$",C,E); dot("$D$",D,E);
dot("$M$",M,N); dot("$M'$",Mp,S,red);
```

## Phép đối xứng trục theo phương $Ox, Oy$

- 1) `transform p=vprojection(l)`; phép đối xứng qua đường thẳng  $l$  theo phương là trục tọa độ  $Oy$ .
- 2) `transform p=hprojection(l)`; phép đối xứng qua đường thẳng  $l$  theo phương là trục tọa độ  $Ox$ . Nếu đường thẳng  $l$  song song với trục  $Ox$  ( $Oy$ ) thì nó là phép đồng nhất.

## 16. Đường tròn

### 16.1. Định nghĩa – vẽ đường tròn

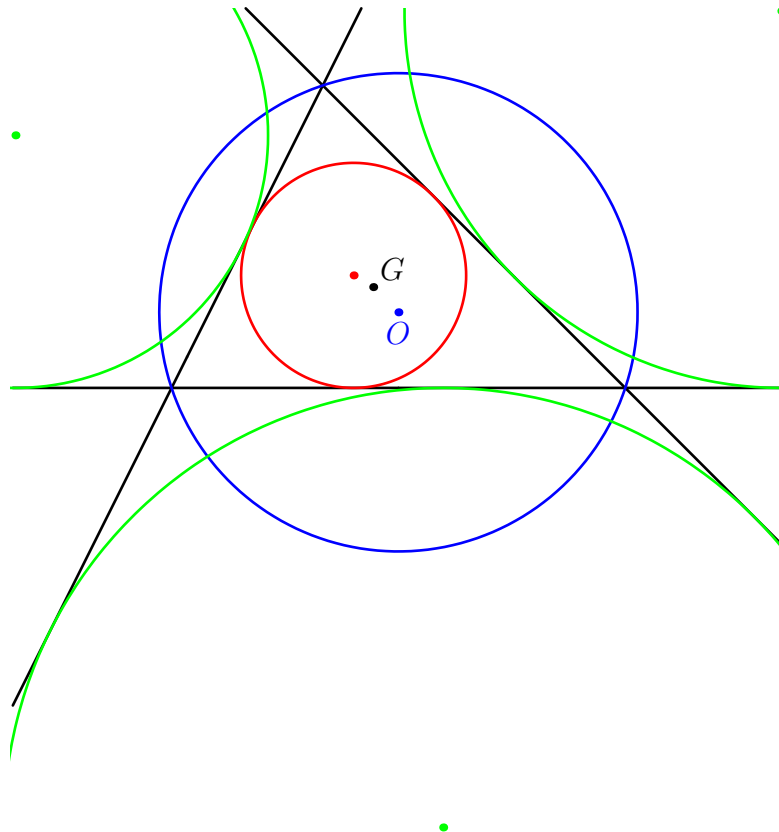
#### Cú pháp.

- 1) `circle c=circle(I,R)`; đường tròn  $c$  tâm  $I$  đã định nghĩa, bán kính  $R$ .



- 2) `circle c=circle((point)(a,b),R)`; đường tròn  $c$  tâm  $I(a,b)$ , bán kính  $R$ .
- 3) `circle c=circle(A,B)`; đường tròn  $c$  có đường kính là  $AB$ .
- 4) `circle c=circle(A,B,C)`; đường tròn  $c$  ngoại tiếp tam giác  $ABC$ .
- 5) `circle c=incircle(A,B,C)`; đường tròn  $c$  nội tiếp tam giác  $ABC$ .
- 6) `circle c=excircle(A,B,C)`; đường tròn bàng tiếp  $c$  của tam giác  $ABC$  tiếp xúc với cạnh  $AB$ .

**Ví dụ 70.**



```
import geometry; unitsize(2cm); pen bpp=linewidth(bp);
point A=(-1,0), B=(2,0), C=(0,2);
draw(line(A,B),bpp); draw(line(A,C),bpp); draw(line(B,C),bpp);
circle cc=circle(A,B,C);//đường tròn ngoại tiếp
draw(cc, bp+blue); //Vẽ đường tròn ngoại tiếp
dot("$O$",cc.C,S,blue); //vẽ tâm của cc
circle ic=incircle(A,B,C);//đường tròn nội tiếp
draw(ic,bp+red); // Vẽ đường tròn nội tiếp
dot(ic.C,red);//Vẽ tâm của đtròn nội tiếp
circle ec=excircle(A,B,C); //đường tròn bàng tiếp xúc với cạnh AB
clipdraw(ec, bp+green); dot(ec.C,green);
ec=excircle(A,C,B);//đường tròn bàng tiếp xúc với cạnh AC
clipdraw(ec, bp+green); dot(ec.C,green);
ec=excircle(C,B,A);//đường tròn bàng tiếp xúc với cạnh BC
clipdraw(ec,bp+green); dot(ec.C,green);
dot("$G$",centroid(A,B,C),NE); //Trọng tâm của tam giác ABC.
```

## 16.2. Chuyển đường tròn thành path

Vì đường tròn là một đường đóng (khép kín) nên có thể đưa về dạng path với mục đích là để tìm tô màu.

**Cú pháp.**

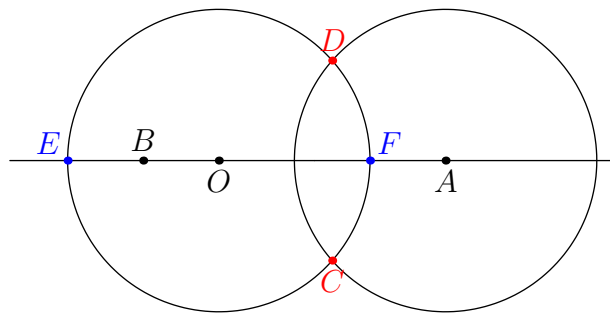
`path pc=c`; `pc` là path của đường tròn `c` đã định nghĩa.

## 16.3. Giao điểm của đường thẳng – đường tròn, đường tròn – đường tròn

**Cú pháp.**

- 1) `point [] T=intersectionpoints(c,l)`; `T[0]`, `T[1]` là hai giao điểm của đường tròn `c` và đường thẳng `l`.
- 2) `point [] T=intersectionpoints(c1,c2)`; `T[0]`, `T[1]` là hai giao điểm của đường tròn `c1` và đường tròn `c2`. Nếu trường hợp cả hai tiếp xúc thì điểm `T[1]` không tồn tại, còn nếu chúng không có điểm chung thì chương trình sẽ báo lỗi.

**Ví dụ 71.** Xem ví dụ.



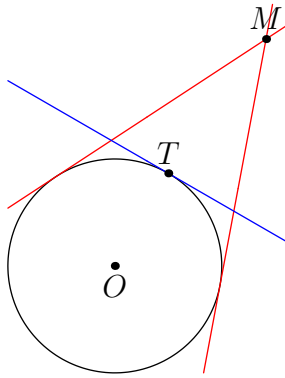
```
import geometry; unitsize(1cm);
linemargin=-0.3cm;
point O=origin, A=(3,0), B=(-1,0) ;
dot("$O$",O,S); dot("$A$",A,S); dot("$B$",B,N);
circle c1=circle(O,2), c2=circle(A,2);
draw(c1);draw(c2); line l=line(A,B); draw(l);
point [] G=intersectionpoints(c1,c2);
dot("$C$",G[0],S,red); dot("$D$",G[1],N,red);
point [] T=intersectionpoints(l,c1);
dot("$E$",T[0],NW,blue); dot("$F$",T[1],NE,blue);
```

## 16.4. Tiếp tuyến của đường tròn

**Cú pháp.**

- 1) `line [] t=tangents(c,M)`; `t[0]`, `t[1]` là hai tiếp tuyến xuất phát từ điểm `M` của đường tròn `c`.
- 2) `line t=tangent(c,x)`; `t` là tiếp tuyến của đường tròn `c` tại điểm thuộc `c` có hoành độ `x`.

**Ví dụ 72.**



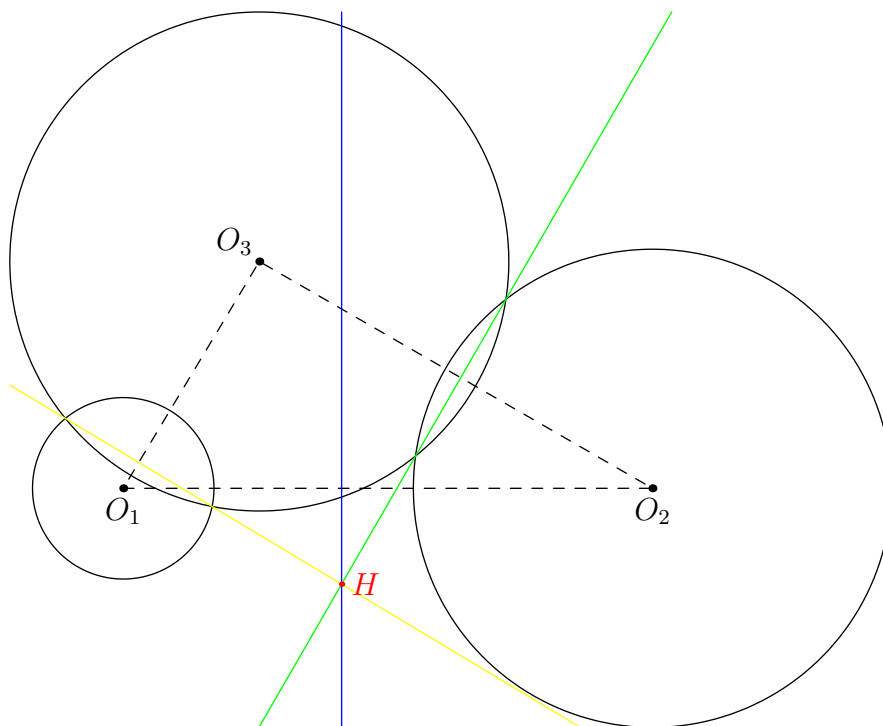
```
import geometry; unitsize(1cm);
point O=(0,0), M=(2,3);
circle c=circle(O,sqrt(2)); draw(c);
line t[] =tangents(c,M); draw(t[0]^^t[1],red);
point T=angpoint(c,60); // Điểm T thuộc c hướng 60 độ.
line tt[] =tangents(c,T); draw(tt[0],blue);
dot("$O$",O,S); dot("$M$",M,N); dot("$T$",T,N);
```

## 16.5. Một số lệnh thường dùng với đường tròn

### Cú pháp.

- 1) `point P=angpoint(c,goc)`; điểm P thuộc đường tròn c theo hướng goc (độ).
- 2) `point P=curpoint(c,số)`; điểm P thuộc đường tròn c theo hướng số (radian).
- 3) `line Tt=radicalline(c1,c2)`; Tt là trục đẳng phương của hai đường tròn c1 và c2.
- 4) `point H=radicalcenter(c1,c2)`; điểm H là giao điểm của trục đẳng phương của c1, c2 và đường thẳng nối tâm của c1 và c2.
- 5) `point H=radicalcenter(c1,c2,c3)`; điểm H là tâm đẳng phương của ba đường tròn c1,c2,c3.
- 6) `point I=c.C`; I là tâm của đường tròn c đã được định nghĩa.

**Ví dụ 73.** Xem ví dụ.



```
import geometry; unitsize(1cm);
point O1=(0,0), O2=(7,0), O3=(1.8,3);
dot("$O_1$",O1,S); dot("$O_2$",O2,S);
dot("$O_3$",O3,NW); real r1=1.2, r2=sqrt(10), r3=3.3;
circle c1=circle(O1,r1), c2=circle(O2,r2), c3=circle(O3,r3);
draw(c1^^c2^^c3); draw(O1--O2--O3--cycle,dashed);
line l1=radicalline(c1,c2), l2=radicalline(c2,c3), l3=radicalline(c3,c1);
draw(l1,blue); draw(l2,green); draw(l3,yellow);
dot("$H$",radicalcenter(c1,c2,c3),red+2bp);
```

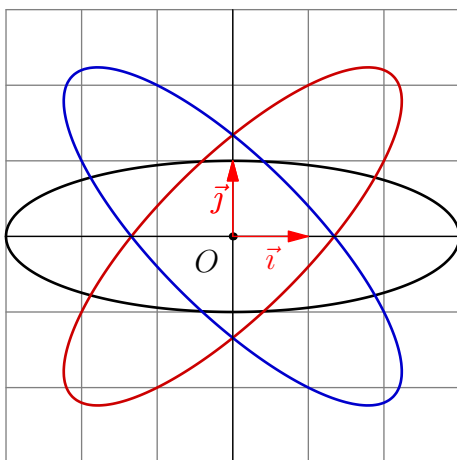
## 17. Đường ellip

### 17.1. Định nghĩa – vẽ đường ellip

Cú pháp.

- 1) ellipse e=ellipse(F1,F2,a); ellip e có 2 tiêu điểm F1,F2 và nửa độ dài của trục lớn a.
- 2) ellipse e=ellipse(F1,F2,M); ellip e có 2 tiêu điểm F1,F2 và điểm M thuộc e.
- 3) ellipse e=ellipse(C, a, b, goc); ellip e có tâm C và a,b là hai chiều của hình chữ nhật cơ sở và hướng goc (độ).

Ví dụ 74.



```
import geometry;
import math;
unitsize(1cm);
add(shift(-3,-3)*grid(6,6,gray)); //Lưới
point O=(0,0); ellipse e0=ellipse(O, 3, 1);
draw(e0, linewidth(bp));
ellipse e1=ellipse(O, 3, 1, 45);
draw(e1, bp+0.8*red);
ellipse e2=ellipse(O, 1, 3, 45);
draw(e2, bp+0.8*blue);
show(currentcoordsys); //Hiện hệ trục Oxy.
```

### 17.2. Chuyển đường ellipse thành path

Cú pháp.

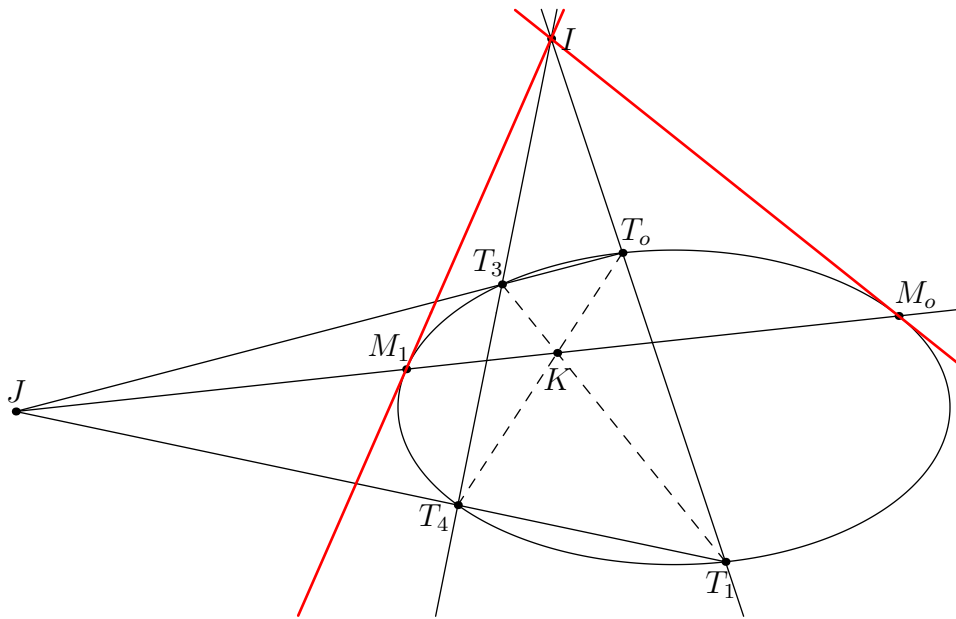
path pe=e; pe là path của đường ellipse e đã được định nghĩa.

### 17.3. Giao điểm của đường thẳng – ellipse, đường tròn – ellipse, ellipse – ellipse

Cú pháp.

- 1) point T[] =intersectionpoints(l,e); T[0],T[1] là giao điểm của đường thẳng l và đường ellipse e.
- 2) point T[] =intersectionpoints(e1,e2); T[0],T[1] là giao điểm của đường ellipse e1 và e2.
- 3) point T[] =intersectionpoints(c,e); T[0],T[1] là giao điểm của đường ellipse e và đường tròn c.

**Ví dụ 75.**



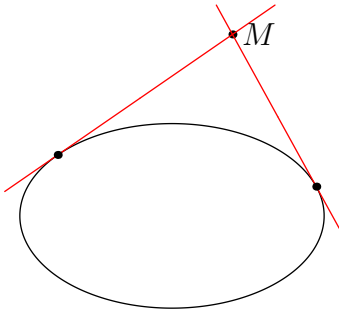
```
import geometry; import patterns; unitsize(1cm); linewidth=-0.2cm;
point F1=(0,0), F2=(6,0), M=(4,2); point A=(3,0),B=(2,3);
point C=(1,3), D=(0,-2); ellipse e=ellipse(F1,F2,M); draw(e);
line AB=line(A,B), CD=line(C,D); draw(AB^^CD);
point I=intersectionpoint(AB,CD); dot("$I$",I,E);
point T[] =intersectionpoints(AB,e);
dot("$T_o$",T[0],dir(60)); dot("$T_1$",T[1],dir(-100));
point TT[] =intersectionpoints(CD,e);
dot("$T_3$",TT[0],dir(120)); dot("$T_4$",TT[1],dir(210));
point J=intersectionpoint(line(T[0],TT[0]),line(T[1],TT[1]));
dot("$J$",J,N); point K=intersectionpoint(line(T[0],TT[1]),line(T[1],TT[0]));
dot("$K$",K,1.6S); line l1=line(K,J,false); draw(l1);
draw(T[1]--J^^T[0]--J); draw(T[0]--TT[1]^T[1]--TT[0],dashed);
point M[] =intersectionpoints(e,l1);
dot("$M_o$",M[0],dir(60)); dot("$M_1$",M[1],dir(120));
draw(line(I,M[0])^^line(I,M[1]),red+bp);
```

### 17.4. Tiếp tuyến của ellipse

Cú pháp.

line t[]=tangents(e,M); t[0],t[1] là hai tiếp tuyến của ellipse e xuất phát từ điểm M.

**Ví dụ 76.**



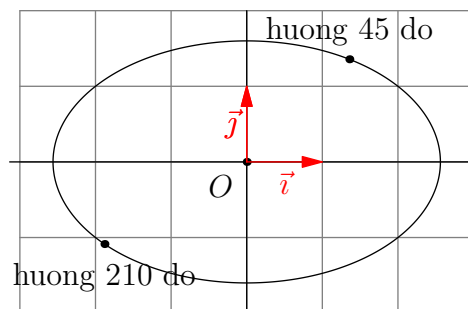
```
import geometry; import patterns;
unitsize(.8cm); linemargin=-0.2cm;
point F1=(0,0), F2=(4,0), A=(3,1.4), M=(3,3);
ellipse e=ellipse(F1,F2,A); draw(e);
dot("$M$",M,E);
line t[]=tangents(e,M);
draw(t[0]..t[1],red);
point I[] =intersectionpoints(e,line(t[0]));
point J[] =intersectionpoints(e,line(t[1]));
dot(I[0]); dot(J[0]);
```

## 17.5. Một số lệnh thường dùng với đường ellipse

### Cú pháp.

- 1) point  $E = \text{angpoint}(e, \text{goc}, \text{fromCenter})$ ; điểm  $E$  thuộc ellipse  $e$  hướng góc (độ).
- 2) point  $I = e.C$ ; điểm  $I$  là tâm của ellipse  $e$ .
- 3) point  $F1 = e.F1$ ; Điểm  $F1$  là tiêu điểm của ellipse  $e$  đã định nghĩa. tương tự cho điểm tiêu điểm  $F2$ .

### Ví dụ 77.



```
import geometry; import patterns;
import math; unitsize(1cm);
add(shift(-3,-2)*grid(6,4,gray));
point F1=(-2,0), F2=(2,0), A=(2,1);
ellipse e=ellipse(F1,F2,A); draw(e);
point E=angpoint(e,45,fromCenter);
point F=angpoint(e,210,fromCenter);
dot("huong 45 do",E,1.5N);
dot("huong 210 do",F, 2S);
show(currentcoordsys);
```

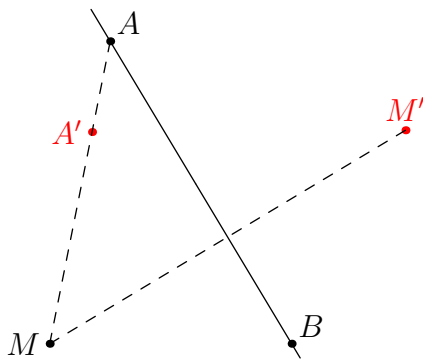
## 18. Ảnh qua phép biến hình

### 18.1. Ảnh của điểm

#### Cú pháp.

point  $Mp = p * M$ ; Điểm  $Mp$  là ảnh của điểm  $M$  qua phép biến hình  $p$  với  $p$  đã định nghĩa trước hoặc có thể định nghĩa trực tiếp vào câu lệnh. Xem các ví dụ sau.

**Ví dụ 78.** Phép vị tự và phép đối xứng trục.



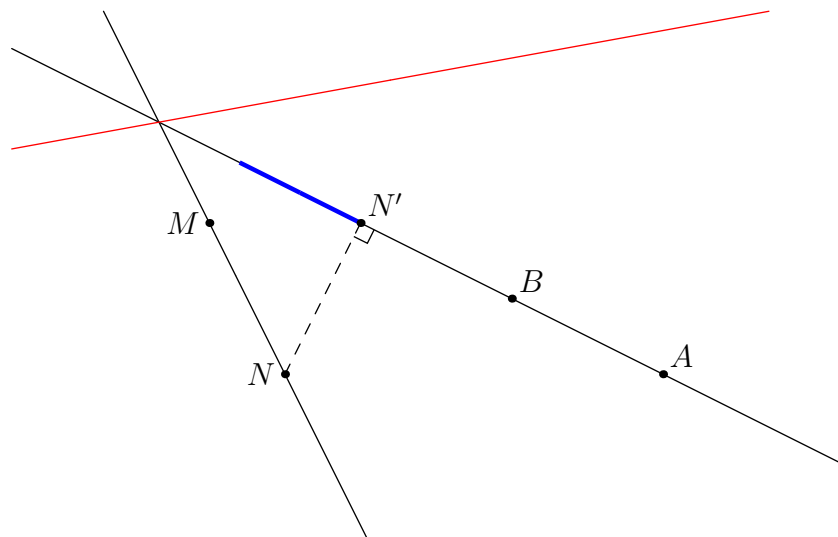
```
import geometry;
unitsize(1cm);
point A=(0,6), B=(3,1), M=(-1,1);
line l=line(A,B); draw(l);
transform p=reflect(l);
point Mp=p*M; dot("$M'$",Mp,E,red);
point Ap=scale(0.7,M)*A; dot("$A'$",Ap,W,red);
dot("$A$",A,NE);dot("$B$",B,NE);
dot("$M$",M,W);draw(M--A^^M--Mp,dashed);
```

## 18.2. Ảnh của đoạn thẳng, đường thẳng

### Cú pháp.

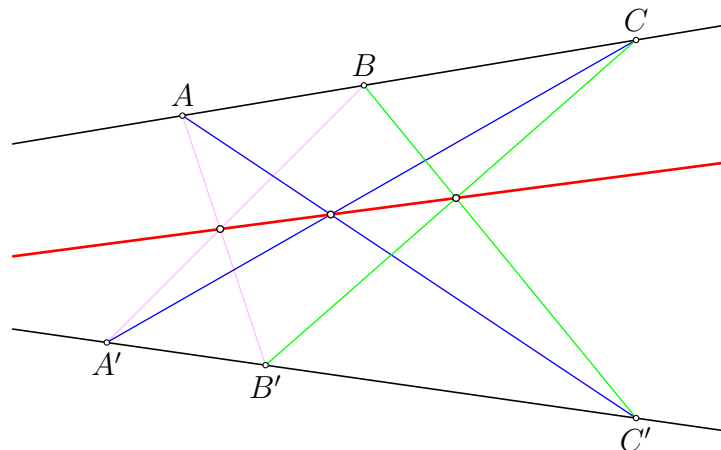
- 1) segment  $ps=operator*(p,s)$ ; định nghĩa  $sp$  là ảnh của đoạn thẳng  $s$  qua phép biến hình  $p$ .
- 2) line  $p1=operator*(p,l)$ ; định nghĩa  $p1$  là ảnh của đường thẳng  $l$  qua phép biến hình  $p$ .

**Ví dụ 79.** Xem ví dụ này.



```
import geometry;
unitsize(1cm);// độ dày điểm 3bp mặc định
linemargin=-2cm;
point A=(3,0), B=(1,1), M=(-3,2), N=(-2,0);
dot("$A$",A,NE);dot("$B$",B,NE);dot("$M$",M,W);dot("$N$",N,W);
line l=line(A,B); draw(l);// định nghĩa và vẽ đường thẳng AB
line lMN=line(M,N); draw(lMN); // định nghĩa, vẽ đường thẳng MN
transform p1=projection(l),p2=reflect(l); //đn phép chiếu, đối xứng trục
line MNp=operator*(p2,lMN); // ảnh của đường thẳng MN qua phép đối xứng
draw(MNp,red); // Vẽ ảnh của đường MN
segment sMN=segment(M,N);
segment MNp=operator*(p1,sMN); //ảnh của MN
draw(MNp,blue+1.7bp); //vẽ ảnh MN
point Np=p1*N; dot("$N'$",Np,NE); draw(N--Np,dashed);
markrightangle(N,Np,B,size=2mm);
```

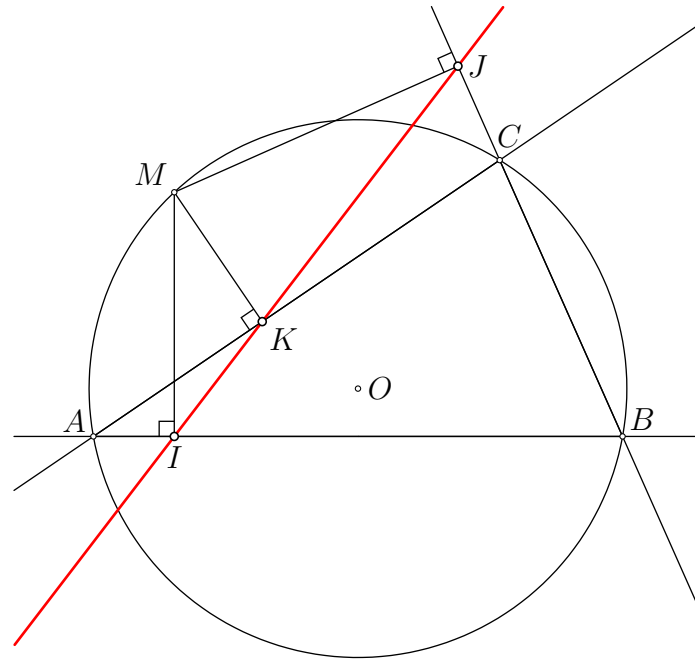
**Ví dụ 80.** Xem ví dụ này.



```
import geometry;
unitsize(1cm); linemargin=-0.6cm;
pen p1=linewidth(.4bp), p2=linewidth(.6bp);
filltype p3=Fill(white);
triangle t=triangleabc(4,6.5,7);
point A=point(t.VA), B=point(t.VB), C=point(t.VC);
drawline(t); circle c=circle(t); draw(c);
point O=circumcenter(t);
point M=curpoint(c,120); // M thuộc đ/tròn c
point I=projection(line(A,B))*M;
point J=projection(line(B,C))*M;
point K=projection(line(A,C))*M;
dot("$A$",A,dir(140),p1,p3); dot("$B$",B,NE,p1,p3);
dot("$C$",C,1.5*dir(70),p1,p3); dot("$O$",O,E,p1,p3);
draw(M--I--M--K--M--J);
perpendicularmark(line(M,I),line(A,B),quarter=3,size=2mm);
perpendicularmark(line(B,C),line(I,J),quarter=1,size=2mm);
perpendicularmark(line(A,C),line(M,K),quarter=2,size=2mm);
draw(line(I,J),red+bp);
dot("$M$",M,NW,p1,p3); dot("$J$",J,E,p2,p3);
dot("$K$",K,SE,p2,p3); dot("$I$",I,S,p2,p3);
```

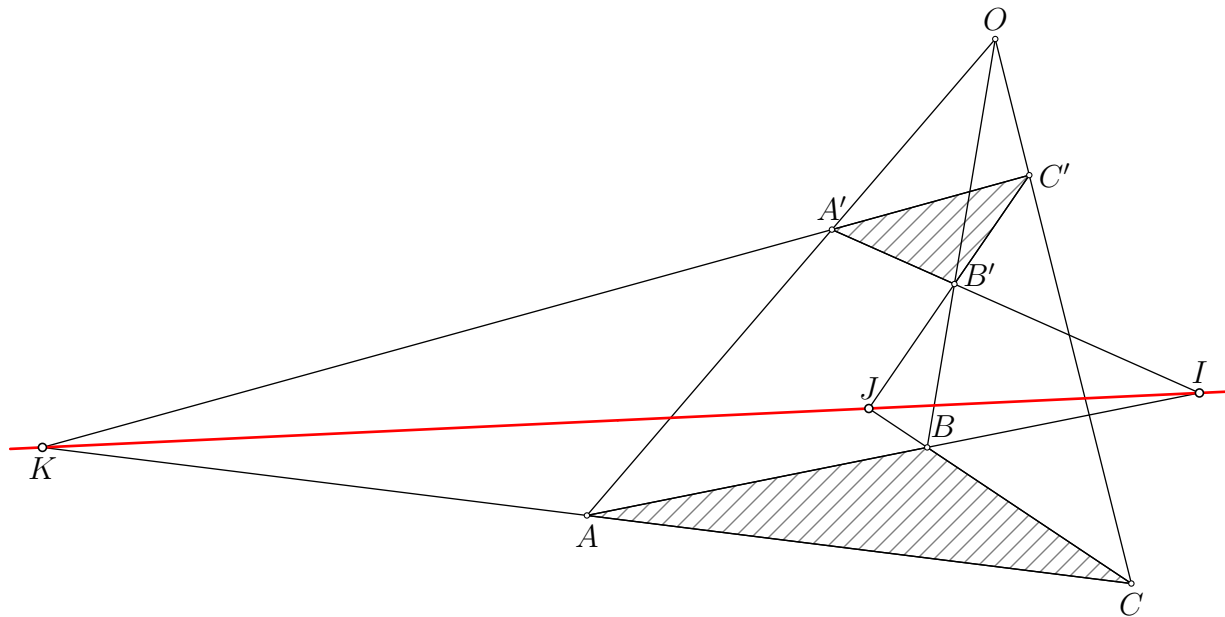
**Ví dụ 81.** Xem ví dụ này.





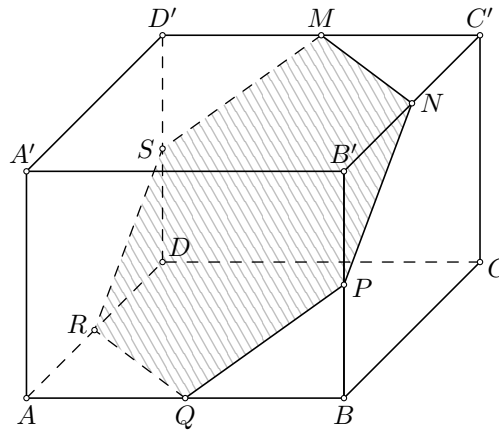
```
import geometry;
unitsize(1cm); linemargin=-0.6cm;
pen p1=linewidth(.4bp), p2=linewidth(.6bp);
filltype p3=Fill(white);
triangle t=triangleabc(4,6.5,7);
point A=point(t.VA), B=point(t.VB), C=point(t.VC);
drawline(t); circle c=circle(t); draw(c);
point O=circumcenter(t);
point M=curpoint(c,120); // M thuộc đ/tròn c
point I=projection(line(A,B))*M;
point J=projection(line(B,C))*M;
point K=projection(line(A,C))*M;
dot("$A$",A,dir(140),p1,p3); dot("$B$",B,NE,p1,p3);
dot("$C$",C,1.5*dir(70),p1,p3); dot("$O$",O,E,p1,p3);
draw(M--I--K--J);
perpendicularmark(line(M,I),line(A,B),quarter=3,size=2mm);
perpendicularmark(line(B,C),line(I,J),quarter=1,size=2mm);
perpendicularmark(line(A,C),line(M,K),quarter=2,size=2mm);
draw(line(I,J),red+bp);
dot("$M$",M,NW,p1,p3); dot("$J$",J,E,p2,p3);
dot("$K$",K,SE,p2,p3); dot("$I$",I,S,p2,p3);
```

**Ví dụ 82.** Xem ví dụ này.



```
import geometry;
import patterns; unitsize(1cm);
pen p1=linewidth(.4bp);
pen p2=linewidth(.6bp);
filltype p3=Fill(white);
linemargin=-.4cm;
point O=(10,4), A=(4,-3), B=(9,-2), C=(12,-4);
point Ap=relpoint(line(O,A),0.4), Bp=relpoint(line(O,B),0.6);
point Cp=relpoint(line(O,C),0.25);
point I=intersectionpoint(line(A,B),line(Ap,Bp));
point J=intersectionpoint(line(C,B),line(Cp,Bp));
point K=intersectionpoint(line(A,C),line(Ap,Cp));
triangle t1=triangle(A,B,C),t2=triangle(Ap,Bp,Cp);
add("hachures",hatch(H=1.5mm,dir=NE,gray));
filldraw(A--B--C--cycle,pattern("hachures"));
filldraw(Ap--Bp--Cp--cycle,pattern("hachures"));
draw(I--A--I--Ap--J--Cp--J--C--K--Cp--K--C);
draw(O--A--O--B--O--C); draw(line(I,K),red+bp);
dot("$O$",O,N,p1,p3); dot("$I$",I,N,p2,p3);
dot("$A$",A,S,p1,p3); dot("$J$",J,N,p2,p3);
dot("$B$",B,1.3*dir(55),p1,p3); dot("$K$",K,S,p2,p3);
dot("$C$",C,S,p1,p3); dot("$A'$",Ap,N,p1,p3);
dot("$B'$",Bp,dir(25),p1,p3); dot("$C'$",Cp,E,p1,p3);
```

**Ví dụ 83.** Ứng dụng phép tịnh tiến, lát nền kiểu hatch( ) và dùng lệnh fill để vẽ hình hộp.



```

import geometry; import patterns;
unitsize(1cm); defaultpen(fontsize(10pt));
pen p1=linewidth(.4bp), p2=linewidth(.6bp);
filltype p3=Fill(white);
point A=origin, B=(4.2,0), C=(6,1.8);
vector v=A-B;
transform f=shift(v);
point D=f*C;
point Ap=(0,3); vector u=Ap-A;
transform g=shift(u);
point Bp=g*B,Cp=g*C,Dp=g*D;
point Q=midpoint(segment(A,B)), P=midpoint(segment(B,Bp));
point N=midpoint(segment(Bp,Cp)), M=midpoint(segment(Cp,Dp));
point R=midpoint(segment(A,D)), S=midpoint(segment(Dp,D));
path p=Q--P--N--M--S--R--cycle;
add("lat",hatch(1mm,dir(120),mediumgray));
fill(p,pattern("lat"));
draw(A--B--Bp--Ap --cycle,p2); draw(B--C--Cp--Bp--cycle,p2);
draw(Cp--Dp--Ap,p2);
draw(A--D--C,dashed); draw(D--Dp,dashed);
draw(Q--P--N--M,p2);draw(M--S--R--Q,dashed);
dot("$A$",A,dir(-90),p1,p3); dot("$B$",B,dir(-90),p1,p3);
dot("$C$",C,dir(-20),p1,p3); dot("$D$",D,dir(45),p1,p3);
dot("$A'$",Ap,dir(100),p1,p3); dot("$B'$",Bp,dir(90),p1,p3);
dot("$C'$",Cp,dir(90),p1,p3);
dot("$D'$",Dp,dir(90),p1,p3);
dot("$Q$",Q,dir(-90),p1,p3); dot("$R$",R,dir(160),p1,p3);
dot("$S$",S,W,p1,p3);dot("$M$",M,dir(90),p1,p3);
dot("$N$",N,dir(0),p1,p3); dot("$P$",P,dir(-10),p1,p3);

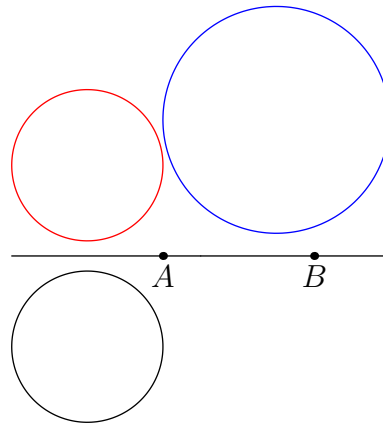
```

### 18.3. Ảnh của đường tròn

#### Cú pháp.

`circle cp=f*c`; cp là ảnh của đường tròn c qua phép biến hình f đã được định nghĩa.

**Ví dụ 84.** Ảnh của đường tròn qua phép đối xứng trục và phép vị tự.



```
import geometry; unitsize(1cm);
point A=(0,0), B=(2,0); point O=(-1,-1.2);
circle c=circle(O,1);
line l=line(A,B); draw(c); draw(l);
transform t1=reflect(l);
transform t2=scale(-1.5,A);
circle c1=t1*c; circle c2=t2*c;
draw(c1,red); draw(c2,blue);
dot("$A$",A,S); dot("$B$",B,S);
```

#### 18.4. Ảnh của đường ellipse

Cấu trúc.

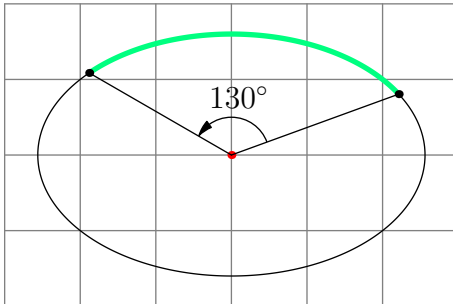
ellipse  $fe=f*e$ ; ảnh của ellipse qua phép biến hình  $f$  là  $fe$ .

### 19. Cung – quạt – vàng trắng khuyết

Cú pháp.

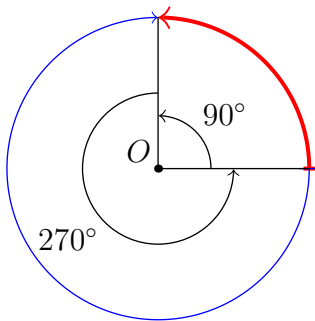
- 1) `arc ar=arc(c,goc1,goc2)`; là 1 cung của đường tròn  $c$  hướng từ  $goc1$  đến  $goc2$  (độ).
- 2) `arc ar=arc(e,goc1,goc2,fromCenter)`;  $ar$  là 1 cung của ellipse  $e$  hướng từ  $goc1$  đến  $goc2$  (độ).
- 3) `arc ar=arc(e,goc1,goc2,fromCenter,CW)`;  $ar$  là 1 cung của ellipse  $e$  hướng từ  $goc1$  đến  $goc2$  (độ) theo hướng ngược chiều kim đồng hồ.

**Ví dụ 85.** Xem ví dụ.



```
import geometry;
import math; unitsize(1cm);
real goc1=20, goc2=150;
add(shift(-3,-2)*grid(6,4,gray));
point F1=(-2,0), F2=(2,0), A=(2,1);
ellipse e=ellipse(F1,F2,A); draw(e);
arc a1=arc(e,goc1,goc2,fromCenter);
draw(a1, springgreen+2bp);
point E1=angpoint(e,goc1,fromCenter);
dot(E1);
point E2=angpoint(e,goc2,fromCenter);
dot(E2);
dot(e.C,red); draw(e.C--E1~e.C--E2);
markangle(format("$%0g^\circ$",degrees(a1)),
E1,e.C,E2,radius=.5cm,Arrow(size=2mm));
```

**Ví dụ 86.** Xem ví dụ.

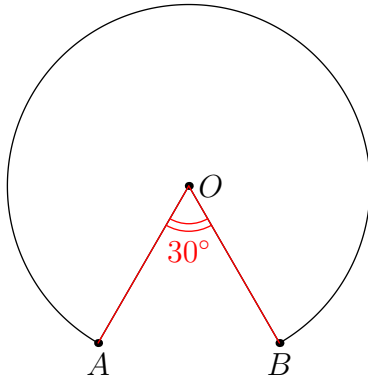


```
import geometry;
unitsize(1cm); point O=(1,0);
real r=2; circle c1=circle(O,r);
arc ar1=arc(c1,0,90);
draw(ar1,red+1.5bp,Arrow(TeXHead,size=.7mm),
BeginBar(1.5mm));
arc ar2=arc(c1,360,90,CW);
draw(ar2,blue,Arrow(TeXHead),BeginBar);
dot("$O$",c1.C,NW);
point A=angpoint(c1,0), B=angpoint(c1,90);
markangle("$90^\circ$",A,O,B, radius=7mm,
Arrow(TeXHead,0.4mm));
draw(O--A~O--B);
markangle("$270^\circ$",B,O,A,radius=10mm,
Arrow(TeXHead));
```

**Một số lệnh thường dùng với cung**

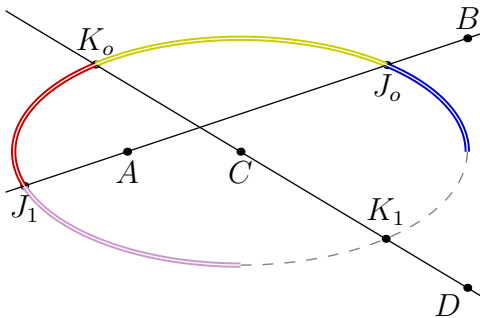
- 1) `arc ar=complementary(a)`; cung bù của cung a.
- 2) `arc ar=reverse(a)`; cung ngược của cung a.
- 3) `arc ar=arccircle(A, B, angle)`; cung tròn tâm A xuất phát từ B quét góc lượng giác anlg (độ).
- 4) `point T[]=intersectionpoints(a1,a2)`; T[0],T[1] là giao điểm của hai cung.
- 5) `arc a=arcsubtended(A,B,angle)`; cung có hai đầu mút A,B với góc angle (độ)
- 6) `point O=a.el.C`; O là tâm của đường tròn chứa cung a.
- 7) `arc a=arc(e,M,N)`; cung a của ellipse hoặc đường tròn từ điểm M tới N (trên ellipse, đường tròn)

**Ví dụ 87.** Xem ví dụ.



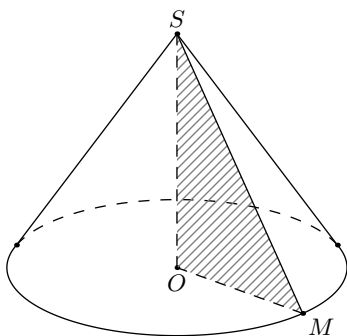
```
import geometry;
unitsize(0.8cm);
point A=(0,0), B=(3,0);
arc a=arcsubtended(A,B,30);
point O=a.el.C; dot("$O$", O);
draw(a); dot("$A$",A,S);
dot("$B$",B,S); draw(O--A^O--B);
markarc("$30^\circ\circ$",n=2, radius=5mm, a,
sectorpen=red);
```

Ví dụ 88. Xem ví dụ.



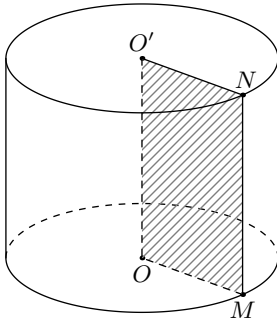
```
import geometry; unitsize(1.5cm);
point A=(-1,0), B=(2,1), C=(0,0), D=(2,-1.2);
real a=270; dot("$A$",A,S);
dot("$B$",B,N); dot("$D$",D,SW);
dot("$C$",C,S); ellipse e=ellipse(C,2,1,0);
arc c=arc(e,0,a,fromCenter);
draw(complementary(c),dashed+grey);
line ab=line(A,B), cd=line(C,D);
point J[]=intersectionpoints(ab,e);
dot("$J_0$",J[0],S); dot("$J_1$",J[1],S);
point K[]=intersectionpoints(cd,e);
dot("$K_0$",K[0],N); dot("$K_1$",K[1],
1.5*dir(90));
draw(arc(e,J[0],K[0]),2bp+.8yellow);
draw(arc(e,K[0],J[1]),2bp+.8red);
draw(arc(e,point(e,0),J[0]),2bp+.8blue);
draw(arc(e,J[1],angpoint(e,a,fromCenter)),
2bp+.8pink);
draw(ab^cd); draw(c,.5bp+white);
```

Ví dụ 89. Vẽ hình nón.



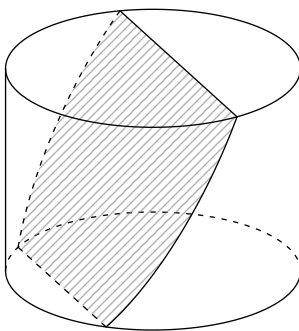
```
import geometry; unitsize(1.5cm);
import math; import patterns;
defaultpen(fontsize(9pt));
dotfactor*=.6; point O=(0,0);
real a=1.5, b=0.6; ellipse e=ellipse(O,a,b);
dot("$O$",O,S); real d=8;
point B0=angpoint(e,d,fromCenter); dot(B0);
point B1=angpoint(e,180-d,fromCenter); dot(B1);
arc a1=arc(e,B0,B1); draw(a1,dashed);
arc a2=arc(e,B1,B0); draw(a2);
point B2=angpoint(e,-20,fromCenter);
dot("$M$",B2,SE); real k=1.5;
transform q=rotate(90,0);point Sd=q*e.F2;
transform v=scale(k,0); point S=v*Sd;
dot("$S$",S,N); path g=S--e.C--B2--cycle;
add("ha",hatch(.8mm,gray));
fill(g,pattern("ha")); draw(S--B0^S--B1^S--B2);
draw(S--e.C^e.C--B2,dashed);
```

**Ví dụ 90.** Vẽ hình trụ.



```
import geometry; import patterns;
unitsize(1.2cm); defaultpen(fontsize(9pt));
pen p1=linetype("5 5"); dotfactor*=.6;
real a=1.5, b=0.6, k=1.2, g=-20; point O=(0,0);
ellipse e=ellipse(O,a,b);
arc a1=arc(e,0,180,fromCenter); draw(a1,p1);
arc a2=arc(e,180,0,fromCenter); draw(a2);
point A[]=intersectionpoints(line(e.F1,e.F2),e);
vector u=(0,2.2); transform p=shift(u);
point A00=p*A[0], A11=p*A[1];
point M=angpoint(e,g,fromCenter);
path ae=p*e; point Op=p*O, N=p*M;
dot("$O$",O,S); dot("$O'$",Op,dir(90));
dot("$M$",M,S); dot("$N$",N,dir(80));
path g=M--O--Op--N--cycle;
add("ha",hatch(.8mm,gray));
fill(g,pattern("ha")); draw(M--N);
draw(M--O--Op,p1); draw(Op--N);
draw(ae); draw(A00--A[0]^~A11--A[1]);
```

**Ví dụ 91.** Giao của khối trụ và mặt phẳng.

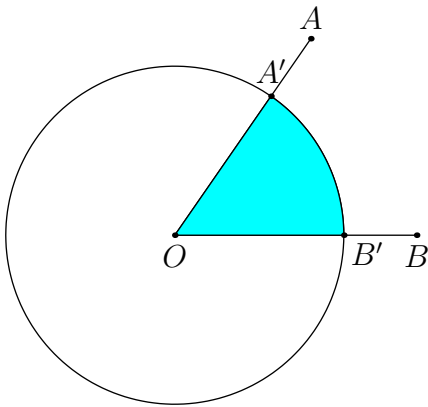


```
import geometry; import patterns;
unitsize(1cm); defaultpen(fontsize(9pt));
dotfactor*=.6; real a=1.5, b=0.6, k=1.5, goc=0;
point O=(0,0); ellipse e=ellipse(O,a,b,goc);
arc a1=arc(e,0,180), a2=arc(e,180,0);
draw(a1,linetype("4 5")); draw(a2);
point Op=scale(k,e.C)*rotate(90,e.C)*e.F2;
vector u=Op-O; transform f=shift(u);
ellipse ef=f*e;
point A[]=intersectionpoints(e,line(e.F1,e.F2));
point A0=f*A[0], A1=f*A[1];
draw(A0--A[0]^~A1--A[1]);
point A=angpoint(e,-130,fromCenter);
point B=angpoint(e,170,fromCenter);
point C=angpoint(ef,-30,fromCenter);
line l=parallel(C,line(A,B));
point D[]=intersectionpoints(ef,l);
point I=(A+B)/2, J=(C+D[0])/2; //trung điểm
real m=1.6; point E=scale(m,I)*J;
conic ec=conic(A,B,C,D[0],E); //conic đi qua 5 điểm
arc aa1=arc(ec,D[0],B), aa2=arc(ec,A,C);
add("ha",hatch(.8mm,gray+opacity(0.6)));
fill(B--A--aa2--C--D[0]--aa1--cycle,pattern("ha"));
draw(aa1,linetype("4 5")); draw(aa2);
draw(A--B,linetype("4 5")); draw(C--D[0]);
draw(ef);
```

### 💡 Thủ thuật

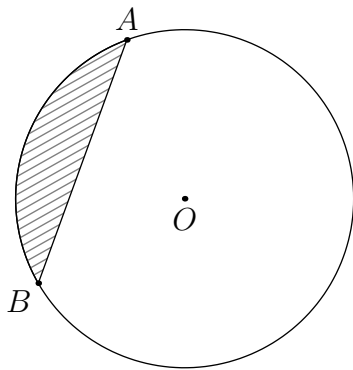
Geometry không hỗ trợ cú pháp để dựng hình quạt, hình vàng trắng khuyết. Do đó, ta dùng ngôn ngữ chính Asymptote và chức năng path để vẽ.

**Ví dụ 92.** Quạt tròn từ điểm A' đến B' trên đường tròn cho trước



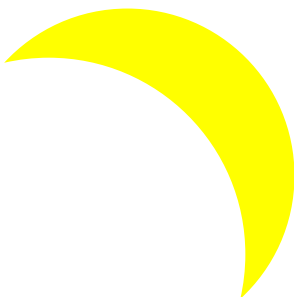
```
import geometry; unitsize(1cm);
dotfactor*=.7; point O=origin;
point A=(1.8,2.6), B=(3.2,0);
real r=sqrt(5);
circle c =circle(O,r); draw(c);
point Ap =intersectionpoint(c,O--A);
point Bp =intersectionpoint(c,O--B);
draw(O--A^^O--B);
arc a= arc(c,Bp,Ap);
path ap = O--Bp--a--cycle;
filldraw(ap,cyan);
dot("$B'$",Bp,SE);
dot("$A'$",Ap,1.4*N);
dot("$A$",A,N);
dot("$B$",B,S);
dot("$O$",O,S);
```

**Ví dụ 93.** Hình giới hạn bởi cung và dây.



```
import geometry; import patterns; unitsize(1cm);
dotfactor*=.7; point O=origin; real r=sqrt(5);
circle c =circle(O,r); draw(c);
point A=angpoint(c,110); //lấy 2 điểm trên cung
point B=angpoint(c,210);
arc a = arc(c,A,B); // cung A-->B
path a0 = A--a--cycle; // tạo vàng trắng
add("tonen",hatch(1mm,dir(30),gray));
filldraw(a0,pattern("tonen")); // tô
dot("$A$",A,N); dot("$B$",B,SW);
dot("$O$",O,S);
```

**Ví dụ 94.** Vàng trắng khuyết.



```
import geometry; unitsize(1cm);
point I=(0,0), J=(0.8,0.8);
real r1=2; real r2=1.7;
circle c1=circle(I,r1);
circle c2=circle(J,r2);
point []G=intersectionpoints(c1,c2);
// chuyển point thành pair
pair A=G[0], B=G[1], O1=I, O2=J;
path a1=arc(O1,A,B);
path a2=arc(O2,A,B);
path aa=buildcycle(a1,a2);
fill(aa,yellow); // Tô màu
```



## 20. Vòng lặp

Vòng lặp trong gói geometry được sử dụng giống như vòng lặp trong ngôn ngữ lập trình C++. Ở đây, chúng ta nói tới vòng lặp for mục đích là để thực hiện tự động một số lệnh lặp lại như vẽ các điểm trên trục số hay vẽ ảnh qua phép tịnh tiến, vị tự,... cho một số hình đặc biệt liên quan đến lập trình tự động. Có thể tham khảo thêm trên [đây](#) về vòng lặp for trong C++ và một số ví dụ sử dụng vòng lặp for trong asymptote tại [web](#)

### Cú pháp.

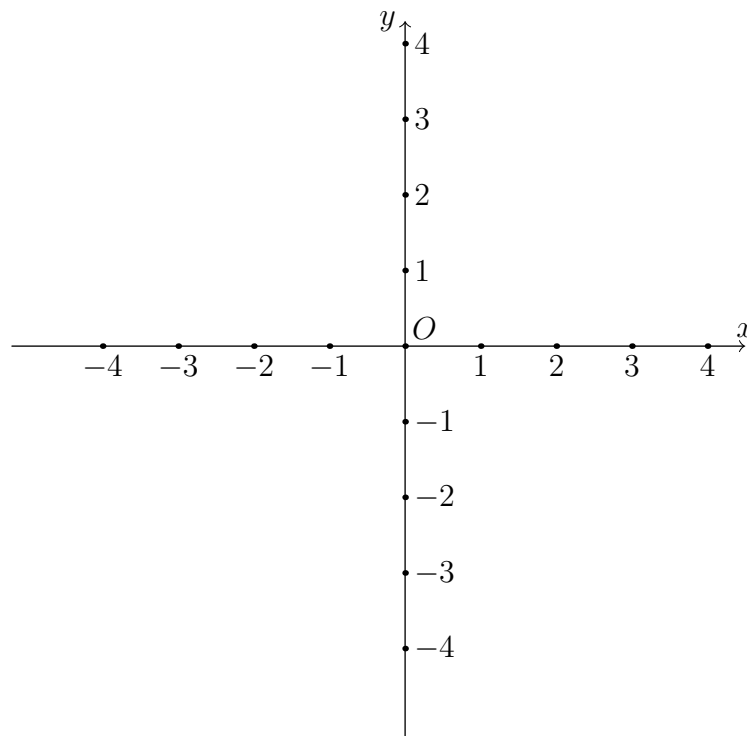
```
for (tạo biến; điều kiện biến; lặp biến){ lệnh thực thi ; ... ; ... ;}
```

1) tạo biến: khai báo `int i=...` hoặc `real i=...` khai báo biến  $i$  là số nguyên hoặc số thực.

2) điều kiện biến: khai báo `i < 1số` hoặc `<=, >, >=`.

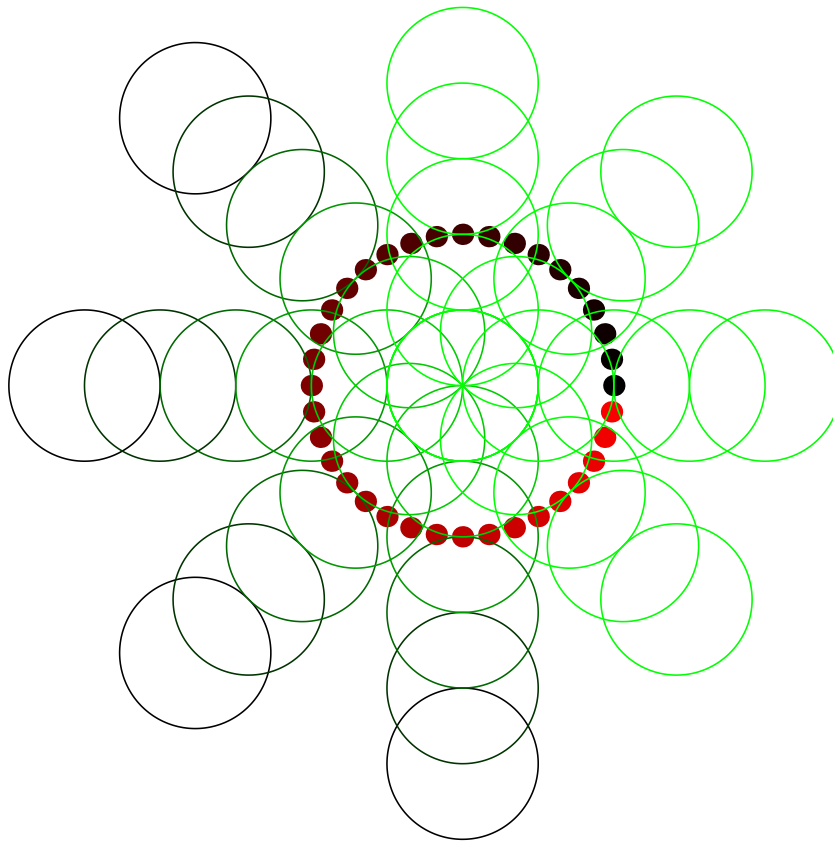
3) lặp biến: khai báo: `i+=1` hoặc `++i` mỗi lần thực hiện biến  $i$  tăng lên 1 đơn vị. Có thể `i+=0.2` mỗi lần thực hiện biến  $i$  tăng lên 0.2.

**Ví dụ 95.** Vẽ hệ trục tọa độ.



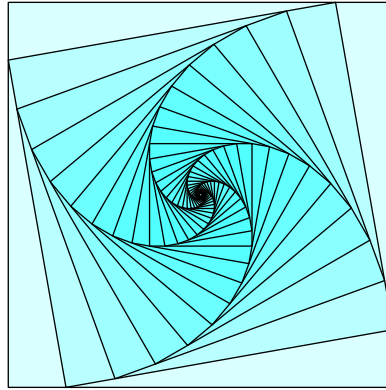
```
import geometry;
unitsize(1cm);
dotfactor*=.7;
for (int i=1; i<5; ++i){
dot(Label(format("%i",i),align=S),(i,0)); // vẽ điểm và
dot(Label(format("%i",i),align=E),(0,i)); }//gắn nhãn ở trục hoành.
for (int i=-4; i<0; ++i){
dot(Label(format("%i",i),align=S),(i,0));
dot(Label(format("%i",i),align=E),(0,i));}
draw(Label("$x$",align=N,EndPoint), (-5.2,0)--(4.5,0),Arrow(TeXHead));
draw(Label("$y$",align=W,EndPoint), (0,-5.2)--(0,4.3),Arrow(TeXHead));
dot("$0$", (0,0),NE);
```

**Ví dụ 96.** Vẽ đường tròn.



```
import geometry;
unitsize(1cm);
defaultpen(linewidth(0.6pt));
for (int i =0; i < 360; i +=10){
pen p=(i/360)*red; // Định nghĩa màu
transform q=rotate(i,(origin)); // Phép quay tâm 0, góc i độ
point A=(2,0);
point Aq=q*A;
dot(Aq,p+8bp); }
for (real i=-0; i<10; ++i){
circle c=circle((point)(-5,0),1);
vector v=(1,0);
pen col=(i/5)*green;
draw(shift(i*v)*c,col); // Vẽ các đường tròn
draw(rotate(90,origin)*shift(i*v)*c,col);
draw(rotate(45,origin)*shift(i*v)*c,col);
draw(rotate(-45,origin)*shift(i*v)*c,col); }
```

**Ví dụ 97.** Vẽ hình vuông chồng hình vuông.



```
import geometry;
unitsize(3.6cm);
point O = origin;
path p=polygon(4); // đa giác đều 4 cạnh (hình vuông)
real alpha =10; //Lấy alpha từ (0,90)
real beta = 45-alpha;
real k = sqrt(2)/(2*cos(beta));
transform T = scale(k,O)*rotate(alpha,O);
int n=35;
for (int i=0; i<n; ++i){
filldraw(T^i*p,i/n *red+opacity(0.15)+cyan); }
```

## 21. Khai báo số thực

- 1) `real k=a`; khai báo  $k$  là số thực  $a$ .
- 2) `real k=abs(a)`; trị tuyệt đối của số thực  $a$ .
- 3) `real r=abs(M)`;  $r$  là độ dài đoạn  $OM$  với  $O$  là gốc tọa độ.
- 4) `real goc=degrees(M)`;  $\text{goc}$  là góc  $(\overrightarrow{Ox}, \overrightarrow{OM})$  với đơn vị là độ.
- 5) `real goc=angle(l)`;  $\text{goc}$  là góc giữa đường thẳng  $l$  với tia  $Ox$  với đơn vị là radian.
- 6) `real goc=degrees(l)`;  $\text{goc}$  là góc giữa đường thẳng  $l$  với tia  $Ox$  với đơn vị là độ.
- 7) `real goc=degrees(l1,l2)`;  $\text{goc}$  là góc định hướng giữa hai đường thẳng  $l1, l2$ .
- 8) `real d=distance(M,l)`;  $d$  là khoảng cách từ điểm  $M$  đến đường thẳng  $l$ .
- 9) `real k=abs(B-A)`; khoảng cách giữa hai điểm  $A, B$ .

## 22. Phép nghịch đảo

### 22.1. Định nghĩa

Cấu trúc.

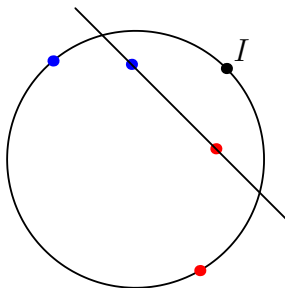
- 1) `inversion iv=inversion(k,C)`; hoặc `inversion iv=inversion(C,k)`; phép nghịch đảo tâm  $C$ , tỷ số  $k$  ( $k \neq 0$ ).
- 2) `inversion iv=circle`; phép nghịch đảo qua đường tròn `circle` đã định nghĩa.

## 22.2. Ứng dụng của phép nghịch đảo

Giả sử  $iv$  là phép nghịch đảo đã được định nghĩa.

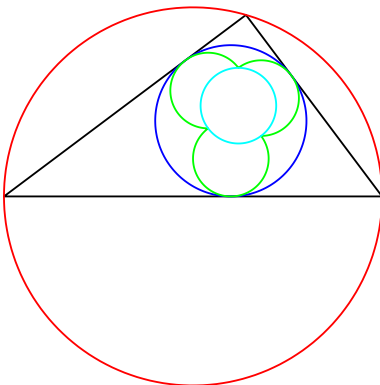
- 1) `point Ai=iv*A`; điểm  $A_i$  là ảnh của  $A$  qua  $iv$ .
- 2) `circle ci=iv*c`; đường tròn  $c_i$  là ảnh của đường tròn  $c$  qua  $iv$ .
- 3) `circle ci=iv*l`; đường tròn  $c_i$  là ảnh của đường thẳng  $l$  qua  $iv$ .
- 4) `arc si=iv*s`; cung  $s_i$  là ảnh của đoạn thẳng  $s$  qua  $iv$ .
- 5) `path ti=iv*t`; path  $t_i$  là ảnh của tam giác  $t$  qua  $iv$ .

**Ví dụ 98.** Ảnh của điểm qua phép nghịch đảo.



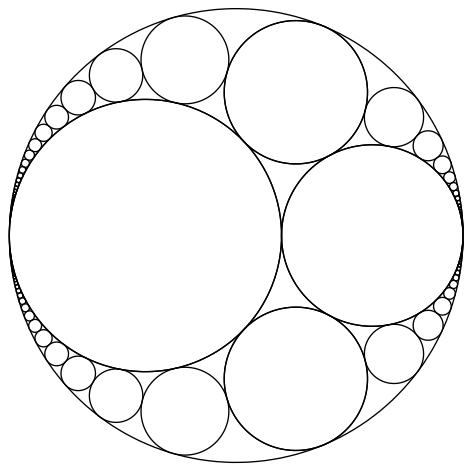
```
import geometry;
import math;
defaultpen(linewidth(.7pt));
unitsize(1.6cm);
linemargin=-0.3cm;
circle c=circle(origin,1);
draw(c);
point I=dir(45);dot("$I$",I,NE);
point A=dir(-60), B=dir(130);//điểm A hướng -60 độ
point C=2*dir(45),D=5*dir(170);
inversion iv=inversion(I,1);
dot(A,red); dot(B,blue);
dot(iv*A,red); dot(iv*B,blue);
draw(line(iv*A,iv*B));
```

**Ví dụ 99.** Ảnh của đường tròn, tam giác qua phép nghịch đảo.



```
import geometry;
unitsize(1cm);
defaultpen(linewidth(.7pt));
triangle t=triangleabc(3,4,5);
draw(t);
circle cngo=circumcircle(t); draw(cngo,red);
circle cnoi=incircle(t); draw(cnoi,blue);
inversion nd=cnoi; draw(nd*t,green);
draw(nd*cngo,cyan);
```

**Ví dụ 100.** Đường tròn tiếp xúc



```
import geometry; unitsize(1cm);
dotfactor*=0.7; point A=(0,0);
point B=(6,0); real k=0.6;
point C=scale(k,A)*B;
circle cAC=circle(A,C); draw(cAC);
circle cAB=circle(A,B); draw(cAB);
circle cBC=circle(B,C); draw(cBC);
real n=abs(C-A)*abs(B-A);
real m=abs(B-A)*abs(B-C);
inversion invn=inversion(A,n);
inversion ivm=inversion(B,m);
point I=cBC.C; //dot(I); point K=cAC.C;
point J=rotate(90,I)*B; point L=rotate(90,K)*C;
vector u=2*(J-I); vector v=2*(L-K);
int dau=-50; int cuoi=50;
for (int i=dau ; i< cuoi; ++i){
circle cc=shift(i*u)*cBC;
draw(ivn*cc);
circle ccc=shift(i*v)*cAC;
draw(ivm*ccc);}
```

## 23. Định nghĩa hàm

Hàm ở đây chính là cú pháp (lệnh) vẽ hình đã định nghĩa sẵn. Ta có thể dùng hàm đó để vẽ hình mà không cần phải định nghĩa lại mỗi khi vẽ hình mới (giống như trong ngôn ngữ lập trình, ta định nghĩa một hàm rồi sau đó sử dụng chứ không định nghĩa lại nữa). Tham khảo thêm trên trang chủ [tại đây](#) hoặc trong Asymptote: the Vector Graphics Language 2.42-1

Gói geometry chỉ hỗ trợ vẽ hình học phẳng. Như vậy, để đánh dấu góc vuông thì đã có lệnh hỗ trợ còn nếu ta vẽ hình biểu diễn của hình không gian mà hình không gian đó có góc vuông. Ta muốn đánh dấu góc vuông đó thì làm sao? Khi đó hàm sẽ giúp ta làm việc đó mà chỉ cần 1 lần định nghĩa.

### Các bước tạo hàm

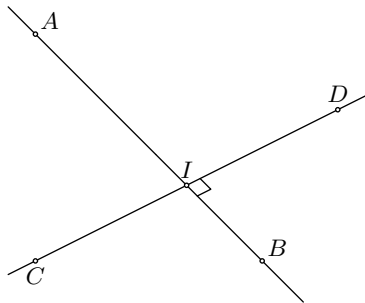
- 1) Tạo hàm trên texeditor → lưu thành file có phần mở rộng là .asy.
- 2) Lưu file vừa tạo vào thư mục cài đặt phần mềm Asymptote: đối với HĐH Windows rất dễ dàng tìm thấy, với MacOS thì thư mục cài đặt nằm trong thư mục TexLive: /usr/local/texlive/2017/texmf-dist/asymptote.
- 3) Thêm hàm đó vào tập tin chứa lệnh vẽ hình.

**Ví dụ 101.** Trong ví dụ này, ta sẽ viết hàm đánh dấu góc vuông, lưu thành tên bosung.asy và thêm lệnh `import bosung;` vào sau lệnh `import geometry;` trong tập tin chứa lệnh vẽ hình.

```
import geometry;
path gocv(pair A, pair O, pair B, real k){
return intersectionpoint(shift(O)*scale(k)*unitcircle,0--A)--
intersectionpoint(parallel(intersectionpoint(shift(O)*scale(k)*unitcircle,0--A),
line(O,B)),parallel(intersectionpoint(shift(O)*scale(k)*unitcircle,0--B),
line(O,A))) -- intersectionpoint(shift(O)*scale(k)*unitcircle,0--B);}
```

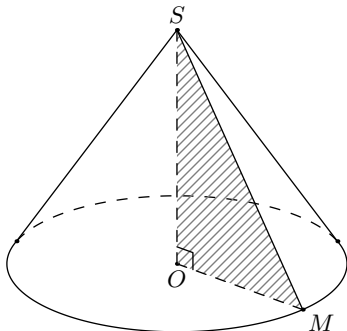
**Giải thích.** `gocv(pair A, pair O, pair B, real k)` là 2 cạnh kề của hình thoi dùng để đánh dấu góc vuông  $\angle AOB$  với độ dài cạnh  $k$  cm và để đánh dấu góc vuông đó, ta dùng lệnh `draw()`;

**Ví dụ 102.** Trong không gian, hai đường thẳng vuông góc và cắt nhau.



```
import geometry;
import bosung; // thêm hàm góc vuông
defaultpen(fontsize(9pt));
unitsize(1cm);
filltype to = Fill(white);
dotfactor*=.7;
linemargin=-.2cm; // giãn khung hình
point A=(-1,2), B=(2,-1); // để hiện đường thẳng d
point C=(-1,-1), D=(3,1) ;
draw(line(A,B));dot("$A$",A,NE,to);dot("$B$",B,NE,to);
draw(line(C,D));dot("$C$",C,S,to);dot("$D$",D,N,to);
point I = intersectionpoint(line(A,B),line(C,D));
dot("$I$",I,N,to);
draw(gocv(B,I,D,0.2)); // đánh dấu góc vuông BID, 0.2cm
```

**Ví dụ 103.** Hình nón.



```
import geometry; import math;
import patterns; import bosung;
unitsize(1.5cm); defaultpen(fontsize(9pt));
dotfactor*=.5; point O=(0,0);
real a=1.5, b=0.6; ellipse e=ellipse(O,a,b);
dot("$O$",O,S); real d=8;
point B0=angpoint(e,d,fromCenter); dot(B0);
point B1=angpoint(e,180-d,fromCenter); dot(B1);
arc a1=arc(e,B0,B1); draw(a1,dashed);
arc a2=arc(e,B1,B0); draw(a2);
point B2=angpoint(e,-20,fromCenter);
dot("$M$",B2,SE); real k=1.5;
transform q=rotate(90,0);point Sd=q*e.F2;
transform v=scale(k,0); point S=v*Sd;
dot("$S$",S,N); path g=S--e.C--B2--cycle;
add("ha",hatch(.8mm,gray));
fill(g,pattern("ha")); draw(S--B0--B1--S--B2);
draw(S--e.C--e.C--B2,dashed);
draw(gocv(B2,O,S,0.15)); // đánh dấu góc vuông
```

## 24. Hàm số

Asymptote hỗ trợ các hàm số: sin, cos, tan, asin, acos, atan, exp, log, pow10, log10, sinh, cosh, tanh, asinh, acosh, atanh, sqrt, cbrt, fabs, floor, ceil, abs, ... các hàm khác (tự định nghĩa). Đối với hàm lượng giác đã liệt kê thì asymptote mặc định là sử dụng đơn vị radian. Còn Sin, Cos, Tan, aSin, aCos và aTan sử dụng đơn vị độ. Tham khảo thêm trong tài liệu Asymptote: the Vector Graphics Language 2.42-1.pdf từ trang 68 – 69

## 25. Đồ thị hàm số – tô miền tích phân

Vẽ đồ thị hàm số bằng gói graph, patterns trong asymptote. Ở đây tôi chỉ giới thiệu các bước cơ bản để định nghĩa hàm số, vẽ đồ thị hàm số và tô miền tích phân. Các bạn có thể tham khảo thêm trên trang chủ

tại **đây** hoặc **tại đây**.

## Cú pháp.

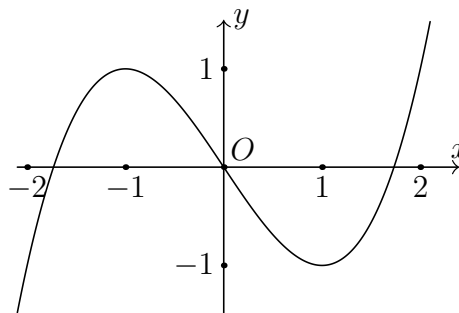
Ta thêm vào môi trường asy lệnh

```
import graph;
import patterns;
```

## Định nghĩa, vẽ đồ thị hàm số

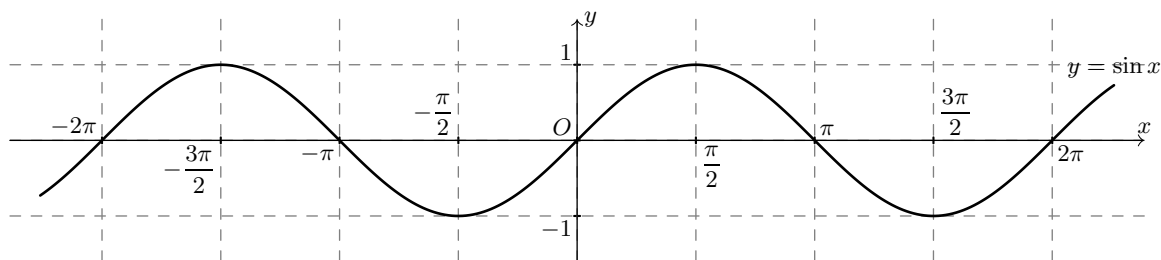
- 1) `real f(real x){ return biểu thức hàm số ;}` định nghĩa hàm số  $f$  theo biến  $x$ . Xem thêm các hàm được hỗ trợ trong asymptote trong tài liệu Asymptote: the Vector Graphics Language version 2.41.
- 2) `path pf = graph(f,a,b,n=300);` định nghĩa path  $pf$  là đồ thị hàm số  $f$  trên đoạn  $[a; b]$  và  $n=300$  là số điểm để nối thành đồ thị (độ mịn).
- 3) `draw(pf);` vẽ đồ thị hàm số.

**Ví dụ 104.** Vẽ đồ thị hàm số.



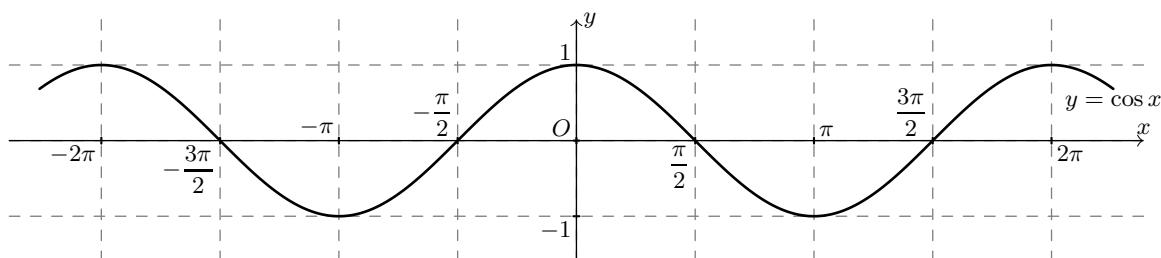
```
import geometry; import graph;
import patterns; unitsize(1.3cm);
defaultpen(linewidth(0.6pt));
dotfactor*=0.6; // giảm độ to của dot
real xmin=-2.1, xmax=2.4;
real ymin=-1.5, ymax=1.5; // dùng để vẽ trục số.
draw(Label("$x$",align=N,EndPoint),(xmin,0)--(xmax,0),Arrow(TeXHead));
draw(Label("$y$",align=E,EndPoint),(0,ymin)--(0,ymax),Arrow(TeXHead));
dot("$O$", (0,0),NE);
real f(real x){ return 0.5*x^3-1.5*x ;} //định nghĩa hàm
path pf =graph(f,-2.1,2.1,n=300); // định nghĩa đồ thị từ -2.1 đến 2.1
draw(pf); // vẽ đồ thị
for (int i=1; i <= 2; ++i){
dot(Label(format("$%i$",i),align=S),(i,0));} // vẽ điểm và nhãn
for (int i=-2; i <=-1; ++i){
dot(Label(format("$%i$",i),align=S),(i,0));}
dot("$1$", (0,1),W); dot("$-1$", (0,-1),W); // vẽ điểm và nhãn.
```

**Ví dụ 105.** Vẽ đồ thị hàm số  $y = \sin x$ .



```
import geometry; import graph; import math;
usepackage('amsmath'); unitsize(1cm); defaultpen(fontsize(9pt));
//===== số thực
real xmin = -7.5, xmax = 7.5;
real ymin = -1.6, ymax= 1.6; real d= 0.4;
//===== Định nghĩa phần nguyên để vẽ lưới và stick
real pnx_trai = floor(xmax), pnx_phai =ceil(xmin);
real pny_tren = floor(ymax), pny_duoi=ceil(ymin);
//===== Định nghĩa hàm - đồ thị
real f(real x) {return sin(x);}
path p = graph(f,xmin+d,xmax-d,n=300);
//=====Tạo stick
path gac_x = (0,0.04)--(0,-0.04);
path gac_y = (-0.04,0)--(0.04,0);
//=====Vẽ stick
for (real i=pnx_phai+3; i<=pnx_trai-3; i+=1){draw(shift(i*pi/2,0)*gac_x,linewidth(.8bp));}
for (real i=pny_duoi; i<=pny_tren; i+=1){draw(shift(0,i)*gac_y,linewidth(.8bp));}
//===== tạo lưới - vẽ lưới
path luoidoc = (0,ymin)--(0,ymax);
path luoingang = (xmin,0)--(xmax,0);
for (real i=pnx_phai+3; i<=pnx_trai-3; ++i){ draw(shift(i*pi/2,0)*luoidoc,gray+dashed);}
for (real i=pny_duoi; i<=pny_tren; ++i){draw(shift(0,i)*luoingang,gray+ dashed);}
//===== vẽ trục
draw(Label("$x$",align=N,EndPoint),(xmin,0)--(xmax,0),Arrow(TeXHead));
draw(Label("$y$",align=E,EndPoint),(0,ymin)--(0,ymax),Arrow(TeXHead));
//===== vẽ đồ thị - gắn nhãn
draw(p,linewidth(bp)); label("$0$", (0,0),NW);
label("$\dfrac{\pi}{2}$", (pi/2,0),SE); label("$\pi$", (pi,0),NE);
label("$\dfrac{3\pi}{2}$", (3*pi/2,0),NE); label("$2\pi$", (2*pi,0),SE);
label("$-\dfrac{\pi}{2}$", (-pi/2,0),NW); label("$-\pi$", (-pi,0),SW);
label("$-\dfrac{3\pi}{2}$", (-3*pi/2,0),SW); label("$-2\pi$", (-2*pi,0),NW);
label("$1$", (0,1),NW); label("$-1$", (0,-1),SW);
label("$y=\sin x$", (xmax-d,f(xmax-d)),N);
```

**Ví dụ 106.** Vẽ đồ thị hàm số  $y = \cos x$ .



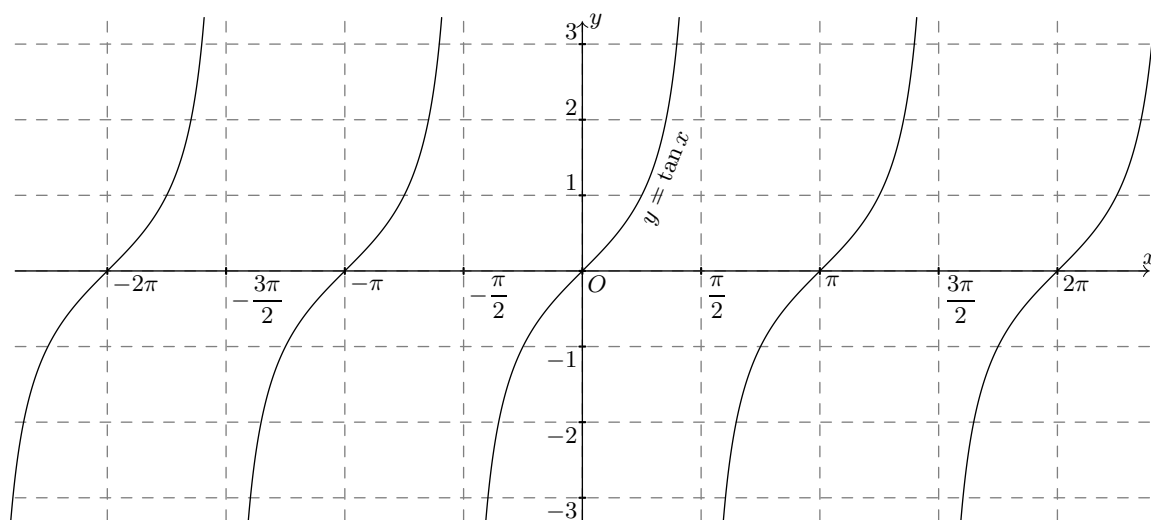


```

import geometry; import graph;
usepackage('amsmath'); unitsize(1cm);
defaultpen(fontsize(9pt));
//===== số thực
real xmin = -7.5, xmax = 7.5;
real ymin = -1.6, ymax= 1.6;
real d= 0.4;
//===== ĐỊNH NGHĨA PHẦN NGUYÊN ĐỂ VẼ LƯỚI VÀ STICK
real pnx_trai =floor(xmax), pnx_phai =ceil(xmin);
real pny_tren = floor(ymax), pny_duoi=ceil(ymin);
//===== tạo lưới
path luoidoc = (0,ymin)--(0,ymax);
path luoingang = (xmin,0)--(xmax,0);
//===== ĐỊNH NGHĨA HÀM
real f(real x) {return cos(x);}
path p = graph(f,xmin+d,xmax-d,n=300);
//=====VẼ STICK
path gac_x = (0,0.04)--(0,-0.04);
path gac_y = (-0.04,0)--(0.04,0);
for ( real i = pnx_phai+3; i<= pnx_trai-3; i+= 1){
draw(shift(i*pi/2,0)*gac_x, linewidth(.8bp));}
for (real i = pny_duoi; i<= pny_tren; i+=1){
draw(shift(0,i)*gac_y,linewidth(.8bp));}
//===== TẠO LƯỚI
for (real i=pnx_phai+3; i<=pnx_trai-3; ++i){
draw(shift(i*pi/2,0)*luoidoc,gray+dashed); }
for (real i =pny_duoi; i<=pny_tren;++i){
draw(shift(0,i)*luoingang,gray+ dashed);}
//===== VẼ TRỤC
draw(Label("$x$",align =N,EndPoint),(xmin,0)--(xmax,0),Arrow(TeXHead));
draw(Label("$y$",align =E,EndPoint),(0,ymin)--(0,ymax),Arrow(TeXHead));
//=====VẼ ĐỒ THỊ
draw(p,linewidth(bp));
label("$0$", (0,0),NW);
label("$\dfrac{\pi}{2}$", (pi/2,0),SW);
label("$\pi$", (pi,0),NE);
label("$\dfrac{3\pi}{2}$", (3*pi/2,0),NW);
label("$2\pi$", (2*pi,0),SE);
label("$-\dfrac{\pi}{2}$", (-pi/2,0),NW);
label("$-\pi$", (-pi,0),NW);
label("$-\dfrac{3\pi}{2}$", (-3*pi/2,0),SW);
label("$-2\pi$", (-2*pi,0),SW);
label("$1$", (0,1),NW);
label("$-1$", (0,-1),SW);
label("$y= \cos x$", (xmax-d,f(xmax-d)),S);

```

**Ví dụ 107.** Vẽ đồ thị hàm số  $y = \tan x$ .

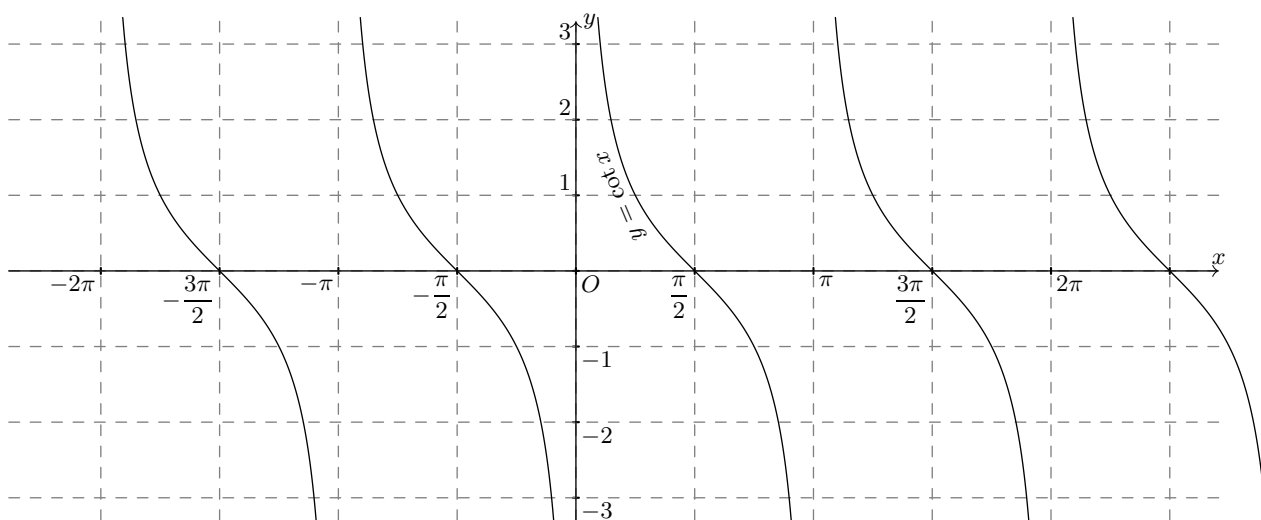


```

import geometry; import graph;
usepackage('amsmath'); unitsize(1cm);
defaultpen(fontsize(9pt));
//===== số thực=====
real xmin = -7.5, xmax = 7.5;
real ymin = -3.3, ymax= 3.3;
real d= 0.29;
//===== ĐỊNH NGHĨA PHẦN NGUYÊN ĐỂ VẼ LƯỚI VÀ STICK
real pnx_trai =floor(xmax), pnx_phai =ceil(xmin);
real pny_tren = floor(ymax), pny_duoi=ceil(ymin);
//===== ĐỊNH NGHĨA HÀM=====
real f(real x) {return tan(x);}
path p = graph(f,-pi/2+d,pi/2-d,n=300);
//=====VẼ STICK=====
path gac_x = (0,0.04)--(0,-0.04);
path gac_y = (-0.04,0)--(0.04,0);
for ( real i = pnx_phai+3; i<= pnx_trai-3; i+= 1){
draw(shift(i*pi/2,0)*gac_x, linewidth(.8bp));}
for (real i = pny_duoi; i<= pny_tren; i+=1){
draw(shift(0,i)*gac_y,linewidth(.8bp));}
//===== TẠO LƯỚI=====
path luoidoc = (0,ymin)--(0,ymax);
path luoingang = (xmin,0)--(xmax,0);
for (real i=pxn_phai+3; i<=pxn_trai-3; ++i){ draw(shift(i*pi/2,0)*luoidoc,gray+dashed); }
for (real i =pny_duoi; i<=pny_tren;++i){ draw(shift(0,i)*luoingang,gray+ dashed); }
//===== VẼ TRỤC=====
draw(Label("$x$",align =N,EndPoint),(xmin,0)--(xmax,0),Arrow(TeXHead));
draw(Label("$y$",align =E,EndPoint),(0,ymin)--(0,ymax),Arrow(TeXHead));
//=====VẼ ĐỒ THỊ =====
//draw(p,linewidth(bp));
for (real i=pxn_phai+5; i<=pxn_trai-5; ++i){draw(shift(i*pi,0)*p); }
//=====NHÃN SỐ Ở TRỤC =====
label("$0$", (0,0),SE); label("$\dfrac{\pi}{2}$", (pi/2,0),SE);
label("$\pi$", (pi,0),SE); label("$\dfrac{3\pi}{2}$", (3*pi/2,0),SE);
label("$2\pi$", (2*pi,0),SE); label("$-\dfrac{\pi}{2}$", (-pi/2,0),SE);
label("$-\pi$", (-pi,0),SE); label("$-\dfrac{3\pi}{2}$", (-3*pi/2,0),SE);
label("$-2\pi$", (-2*pi,0),SE); label("$1$", (0,1),NW);
label("$2$", (0,2),NW); label("$3$", (0,3),NW);
label("$-1$", (0,-1),SW); label("$-2$", (0,-2),SW);
label("$-3$", (0,-3),SW); label(rotate(68)*"$y=\tan x$", (0.8,f(0.8)),E);

```

**Ví dụ 108.** Vẽ đồ thị hàm số  $y = \cot x$ .



```

import geometry; import graph; import math;
usepackage('amsmath');unitsize(1cm);
defaultpen(fontsize(9pt));
//===== số thực
real xmin = -7.5, xmax = 8.5;
real ymin = -3.3, ymax= 3.3;
real d= 0.29;
//===== ĐỊNH NGHĨA PHẦN NGUYÊN ĐỂ VẼ LƯỚI VÀ STICK
real pnx_trai =floor(xmax), pnx_phai =ceil(xmin);
real pny_tren = floor(ymax), pny_duoi=ceil(ymin);

//===== ĐỊNH NGHĨA HÀM
real f(real x) {return cot(x);}
path p = graph(f,0+d,pi-d,n=300);
//=====VẼ STICK
path gac_x = (0,0.04)--(0,-0.04);
path gac_y = (-0.04,0)--(0.04,0);
for ( real i = pnx_phai+3; i<= pnx_trai-3; i+= 1){
draw(shift(i*pi/2,0)*gac_x, linewidth(.8bp));}
for (real i = pny_duoi; i<= pny_tren; i+=1){
draw(shift(0,i)*gac_y,linewidth(.8bp));}
//===== TẠO LƯỚI
path luoidoc = (0,ymin)--(0,ymax);
path luoingang = (xmin,0)--(xmax,0);
for (real i=pxn_phai+3; i<=pxn_trai-3; ++i){
draw(shift(i*pi/2,0)*luoidoc,gray+dashed); }
for (real i =pny_duoi; i<=pny_tren;++i){
draw(shift(0,i)*luoingang,gray+ dashed);}
//===== VẼ TRỤC
draw(Label("$x$",align =N,EndPoint),(xmin,0)--(xmax,0),Arrow(TeXHead));
draw(Label("$y$",align =E,EndPoint),(0,ymin)--(0,ymax),Arrow(TeXHead));
//=====VẼ ĐỒ THỊ
//draw(p,linewidth(bp));
for (real i=pxn_phai+5; i<=pxn_trai-6; ++i){
draw(shift(i*pi,0)*p); }
label("$0$", (0,0),SE); label("$\dfrac{\pi}{2}$", (pi/2,0),SW);
label("$\pi$", (pi,0),SE); label("$\dfrac{3\pi}{2}$", (3*pi/2,0),SW);
label("$2\pi$", (2*pi,0),SE); label("$-\dfrac{\pi}{2}$", (-pi/2,0),SW);
label("$-\pi$", (-pi,0),SW); label("$-\dfrac{3\pi}{2}$", (-3*pi/2,0),SW);
label("$-2\pi$", (-2*pi,0),SW); label("$1$", (0,1),NW);
label("$2$", (0,2),NW); label("$3$", (0,3),NW);
label("$-1$", (0,-1),SE); label("$-2$", (0,-2),SE);
label("$-3$", (0,-3),SE); label(rotate(113)*"$y=\cot x$", (0.9,f(0.9)),W);

```

**Xây dựng miền tô tích phân giới hạn bởi: đồ thị hàm số  $f(x)$ , trục hoành và hai đường thẳng  $x = a, x = b$**

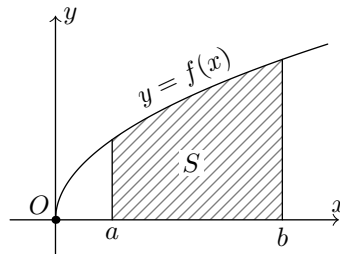
Để tô màu cho miền tích phân, ta cần dựng một path kín. Cú pháp lệnh.

- 1) path tp=buildcycle((a,f(a))--(a,0)--(b,0)--(b,f(b)),graph(f,a,b)); Dựng path kín miền cần tô.

2) `add("ha",hatch(1mm));` thêm kiểu tô trong gói patterns (đã nói ở phần trước).

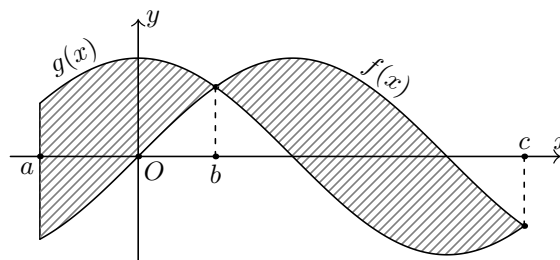
3) `fill(tp,pattern("ha"));` tô nền.

**Ví dụ 109.** Tô miền tích phân giới hạn bởi đồ thị  $f(x)$ , trục hoành và  $x = a, x = b$



```
import geometry; import graph; import patterns;
unitsize(1.5cm); defaultpen(fontsize(10pt));
real xmin=-0.4, xmax=2.5; //
real ymin=-0.3, ymax=1.8; //
real a=0.5, b=2; // 2 cận x=a, x=b.
real f(real x){return sqrt(x);} // định nghĩa hàm
path pf=graph(f,0,2.4,n=400); //định nghĩa đồ thị từ 0 đến 2.4
path tp=buildcycle((a,f(a))--(a,0)--(b,0)--(b,f(b)),graph(f,a,b));
// path để tô nền
add("ha",hatch(1mm,gray)); // Kiểu nền
fill(tp,pattern("ha")); // Tô nền
draw(pf); // Vẽ đồ thị
draw((a,0)--(a,f(a))); draw((b,0)--(b,f(b))); // hai đường thẳng x=a, x=b
draw(Label("$x$",align=N,EndPoint),(xmin,0)--(xmax,0),Arrow(TeXHead));
draw(Label("$y$",align=E,EndPoint),(0,ymin)--(0,ymax),Arrow(TeXHead));
dot("$O$",origin,NW);
label(rotate(25)*"$y=f(x)$",(1,f(1)),N); // nhãn cho đồ thị
label("$a$", (a,0),S); label("$b$", (b,0),S); //
label("$S$", (1.2,0.5),Fill(white)); // nhãn cho miền tô nền.
```

**Ví dụ 110.** Tô miền tích phân giới hạn bởi hai đồ thị hàm số  $f(x)$ ,  $g(x)$  và đường thẳng  $x = a, x = c$ .

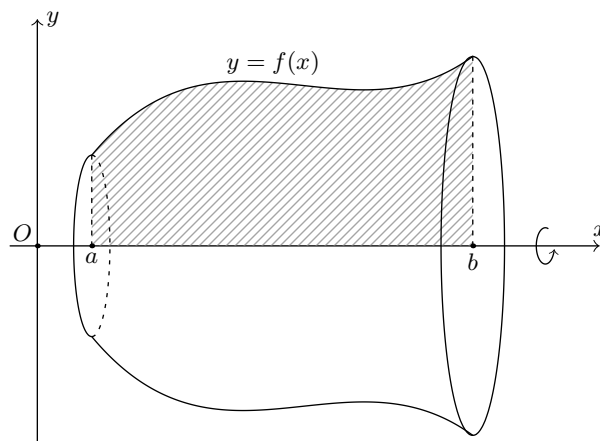


```

import geometry; import graph;
import patterns; unitsize(1.3cm);
defaultpen(fontsize(10pt)+linewidth(0.6pt));
dotfactor*=0.6;
real xmin=-1.3, xmax=4.3;
real ymin=-1.1, ymax=1.4;
real a=-1; real c=5*pi/4; // 2 cận x=a, x=c
real f(real x){return sin(x);} //định nghĩa hàm
path pf=graph(f,a,c,n=300); // đồ thị
real g(real x){return cos(x);} // định nghĩa hàm
path pg=graph(g,a,c,n=300); // đồ thị
pair I[] =intersectionpoints(pf,pg); // Dùng pair thay point
point B=I[0], C=I[1]; dot(B~C); // Chuyển pair thành point
path s=(a,f(a))--(a,g(a)); draw(s); //đường thẳng x=a.
path tphan1=buildcycle(s,pf,pg); // phần bên trái
path tphan2=buildcycle(pf,pg); // phần bên phải
add("ha",hatch(0.8mm,gray)); //kiểu nền
fill(tphan1,pattern("ha")); // Tô nền
fill(tphan2,pattern("ha")); //Tô nền
draw((B.x,0)--(B.x,B.y),linetype("3 5")); // B.x là hoành độ của B
draw((C.x,0)--(C.x,C.y),linetype("3 5")); //
draw(pf~pg); // vẽ hai đồ thị
draw(Label("$x$",align=N,EndPoint),(xmin,0)--(xmax,0),Arrow(TeXHead));
draw(Label("$y$",align=E,EndPoint),(0,ymin)--(0,ymax),Arrow(TeXHead));
dot("$a$",(a,0),SW); dot("$b$",(B.x,0),S); dot("$c$",(C.x,0),N);
dot("$O$",(0,0),SE);label(rotate(-34)*"$f(x)$",(2.6,f(2.6)),N);
label(rotate(24)*"$g(x)$",(-.7,g(-.7)),N);

```

**Ví dụ 111.** Thể tích giới hạn bởi đồ thị hàm số  $f(x)$ .



```

import geometry; import patterns;
import graph; unitsize(0.8cm);
defaultpen(fontsize(9pt)); dotfactor*=0.6;
real a=1, b=5.2;
real xmin=0.4, xmax=6.6;
real ymin=-2.2, ymax =2.5;
draw(Label("$x$",align=1.3N,EndPoint),(xmin-0.3,0)--(xmax,0),Arrow(TeXHead));
draw(Label("$y$",align=1.3E,EndPoint),(xmin,ymin)--(xmin,ymax),Arrow(TeXHead));
real f(real x){return 0.08*x^3-0.8*x^2+2.56*x-0.84;}
path pf=graph(f,1,5.2,n=300);
path pg=buildcycle((a,f(a))--(a,0)--(b,0)--(b,f(b)),graph(f,a,b));
add("ha",hatch(.8mm,gray+opacity(.7)));
fill(pg,pattern("ha")); draw(pf);
draw((a,0)--(a,f(a))^(b,0)--(b,f(b)),linetype("3 5"));
dot("$a$", (a,0),S);dot("$b$", (b,0),S);
ellipse eb=ellipse((b,0),abs(f(b)),0.35,90); draw(eb);
ellipse ea=ellipse((a,0),abs(f(a)),0.2,90);
arc a1=arc(ea,0,180,fromCenter); arc a2=arc(ea,180,0,fromCenter);
draw(a1); draw(a2,linetype("3 5")); path pgf=reflect(line((a,0),(b,0)))*pf;
draw(pgf); line la=line((xmin,0),(xmax,0)), lb=line((xmin,ymin),(xmin,ymax));
point O=intersectionpoint(la,lb); dot("$O$",O,NW);
label("$y=f(x)$", (3,f(3)),N);
arc at=arc(ellipse((b+0.8,0),0.2,0.1,90),-10,250,fromCenter);
draw(at,Arrow(TeXHead,size=0.3mm));

```

## Tài liệu tham khảo

- [1] *Asymptote: the Vector Graphics Language version 2.42*
- [2] Olivier Guibé & Philippe Ivaldi. *Euclidean geometry with asymptote*
- [3] Charles Staats III. *An Asymptote tutorial*
- [4] Trang web <http://asy.marris.fr>
- [5] Trang web <http://www.piprime.fr/asymptote/>



## Chỉ mục

a.el.C, 49  
abs, 54  
add, 28  
altitude, 22  
angle, 54  
angpoint, 39, 42  
anticomplementary, 23  
arc, 48  
    arccircle, 49  
    arcsubtended, 49  
    complementary, 49  
    reverse, 49  
arccircle, 49  
arcsubtend, 49  
Arrow, 18  
  
bisector, 12, 14, 16, 20  
bisectorpoint, 24  
brick, 27  
  
c.C, 39  
centroid, 23  
checker, 27  
circle, 21, 36, 47  
circumcenter, 24  
circumcircle, 24  
complementary, 16, 49  
crosshatch, 28  
curpoint, 39  
  
defaultcoordsys, 5  
defaultpen, 9  
degrees, 54  
distance, 19, 54  
dot, 9  
draw, 8–10, 16, 17  
drawline, 20  
  
e.C, 42  
e.F1, 42  
ellipse, 40, 48  
excenter, 25  
excircle, 21, 25, 37  
extradius, 25  
extouch, 25  
  
false, 10  
fill, 28, 29  
filldraw, 28, 29  
fontsize, 9  
  
foot, 22  
for, 52  
  
geometry, 5  
gergonnet, 25  
  
hatch, 27, 46  
HookHead, 18  
hprojection, 36  
  
import, 5  
incenter, 24  
incircle, 24, 37  
inraidus, 24  
int, 52  
intersectionpoint, 18  
intersectionpoints, 18, 38, 40  
intouch, 25  
inversion, 54  
  
label, 9  
labelmargin, 9  
latticeshade, 30  
line, 10, 11, 20, 43  
  
markangle, 14  
markrightangle, 15  
medial, 23  
median, 23  
midpoint, 16, 23  
  
operator, 20, 43  
opposite, 21  
orthic, 22  
orthocentercenter, 22  
  
p\*M, 42  
parallel, 11  
path, 30, 38, 40  
pen, 30  
perpendicular, 13  
perpendicularmark, 15  
point, 7, 20, 38  
projection, 33  
  
quater, 15  
  
radialshade, 30  
radicalcenter, 39  
radicalline, 39  
real, 25, 54

reflect, 32  
reverse, 49  
rotate, 33  
  
scale, 33  
sector, 12  
segment, 16, 21, 43  
shift, 32  
show, 5  
SimpleHead, 18  
size, 9  
StickIntervalMarker, 14, 16  
symmedial, 26  
symmedian, 26  
  
tangent, 38  
tangential, 24  
tangents, 38, 41  
TeXHead, 18  
tile, 27  
transform  
    hprojection, 36  
    projection, 33, 34  
    reflect, 32, 35  
    rotate, 33  
    scale, 33, 36  
    shift, 32  
    vprojection, 36  
triangle, 19, 22  
triangleAbc, 19  
triangleabc, 19  
  
vertex, 21  
vprojection, 36