

Vẽ đường tròn trên ASY

Trần Quân ++ ...

Bài viết sẽ mô tả một số cách vẽ đường tròn trên ngôn ngữ **ASY** để người dùng có thể vẽ được hiệu quả hơn.

1 Đường tròn trên ASY-2D

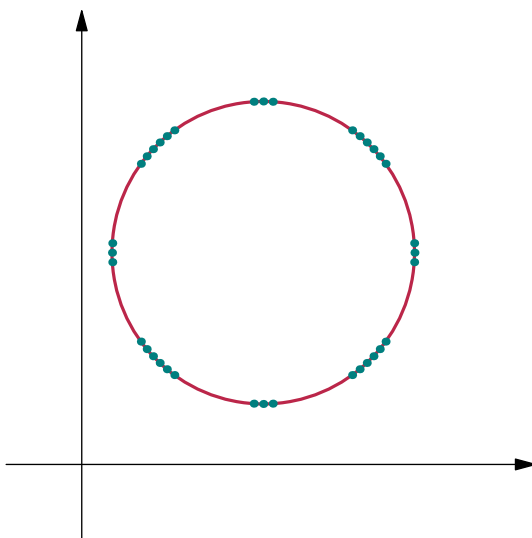
ASY-2D sử dụng các hàm sau để vẽ đường tròn:

- **circle(pair c, real r)**: kiểu **path**, hàm built-in. Hàm này dựa trên đường tròn đơn vị **unitcircle**, nó sẽ trả về đường cong **Bezier** đi qua 8 điểm.
- **Circle(pair c, real r)**: trong gói **graph**;
- **circle(point c, real r)**: trong gói **geometry**;

```
unitsize(2cm);
pen mycolor=RGB(187, 38, 73);

pair c=(1.2,1.4);
path co = circle(c, 1); draw(co, mycolor+1.2bp);

draw((-0.5, 0)--(3, 0), Arrow); draw((0, -0.5)--(0, 3), Arrow);
```



Ví dụ trên sử dụng hàm **circle(pair c, real r)** để khai báo đường tròn tâm **c**, bán kính **1** mà không cần phải khai báo thêm gói nào. Khi kiểm tra khoảng cách từ một điểm trên đường tròn đến tâm, ta chỉ thấy có một số điểm có khoảng cách đến tâm ≈ 1 !. Đường tròn này không chính xác!

Một ví dụ khác để thấy sự không chính xác của đường tròn **circle(pair c, real r)** khi vẽ là đường tròn nội tiếp của tam giác.

```
unitsize(1cm);
pen mycolor=RGB(187, 38, 73);

pair bary(pair A, pair B, real x, real y) {
    return (x*B + y*A) / (x+y);
}
```

```

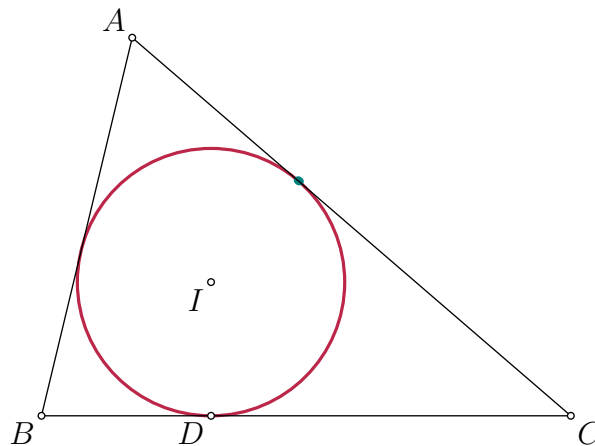
pair incenter(pair A, pair B, pair C) {
    pair E=bary(C, A, abs(C-B), abs(A-B));
    pair F=bary(A, B, abs(A-C), abs(B-C));
    return extension(B, E, C, F);
}

pair projection(pair A, pair P, pair Q) {
    return (A+reflect(P, Q)*A)/2;
}

pair A=(1.2, 5); label("$A$", A, NW);
pair B=(0, 0); label("$B$", B, SW);
pair C=(7, 0); label("$C$", C, SE);
pair I=incenter(A, B, C); label("$I$", I, SW);
pair D=projection(I, B, C); label("$D$", D, SW);
path ci=circle(I, abs(I-D)); draw(ci, mycolor);
pair [] T=intersectionpoints(ci, C--A);
for(int i=0; i<T.length; ++i) dot(T[i], deepcyan);

draw(A--B--C--cycle);
dot(A^^B^^C^^I^^D, Fill(white));

```



Trong ví dụ trên, khi xét giao của **path ci** và đoạn AC , ta thấy có **02** giao điểm. Do đó khi sử dụng hàm này để tìm giao, hãy cẩn thận!

Hàm **Circle(pair c, real r)** trong gói **graph** trả về **path** và cũng chính xác hơn, tuy nhiên trong ví dụ trên, nếu thay **path Ci=Circle(I, abs(I-D)); draw(Ci, deepblue);** và tìm giao với AC , ta sẽ thấy không có giao điểm!

```

import graph;
unitsize(1cm);

pair bary(pair A, pair B, real x, real y) {
    return (x*B + y*A) / (x+y);
}

pair incenter(pair A, pair B, pair C) {
    pair E=bary(C, A, abs(C-B), abs(A-B));
    pair F=bary(A, B, abs(A-C), abs(B-C));
    return extension(B, E, C, F);
}

pair projection(pair A, pair P, pair Q) {
    return (A+reflect(P, Q)*A)/2;
}

```

```

pair A=(1.2, 5); label("$A$", A, NW);
pair B=(0, 0); label("$B$", B, SW);
pair C=(7, 0); label("$C$", C, SE);
pair I=incenter(A, B, C); label("$I$", I, SW);
pair D=projection(I, B, C); label("$D$", D, SW);
path Ci=Circle(I, abs(I-D)); draw(Ci, deepblue);

pair [] T=intersectionpoints(Ci, A--B);
for(int i=0; i<T.length; ++i) dot(T[i], brown);

draw(A--B--C--cycle);
dot(A^^B^^C^^I^^D, Fill(white));

```

Hàm **circle(point c, real r)** trong gói **geometry** trả về kiểu **circle** là kiểu dữ liệu được định nghĩa riêng trong gói này.

```

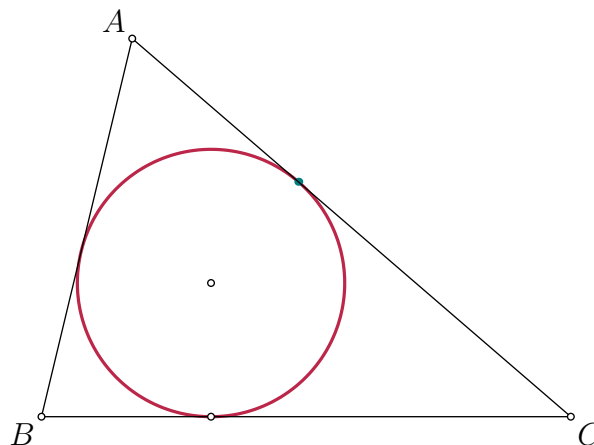
import geometry;
unitsize(1cm);
pen mycolor=RGB(187, 38, 73);

point A=(1.2, 5); label("$A$", A, NW);
point B=(0, 0); label("$B$", B, SW);
point C=(7, 0); label("$C$", C, SE);
point I=incenter(A, B, C); //label("$I$", I, SW);
point D=projection(line(B, C))*I; //label("$D$", D, SW);
circle ci=incircle(A, B, C); draw(ci, mycolor);

point [] T=intersectionpoints(ci, line(A, C)); write(T.length);
for(int i=0; i<T.length; ++i) dot(T[i], deepcyan);

draw(A--B--C--cycle);
dot(A^^B^^C^^I^^D, Fill(white));

```



Ở ví dụ này, hàm **circle ci=incircle(A, B, C);** là hàm của gói **geometry**, khi tìm giao với đường thẳng **AC** ta thấy có **01** giao điểm.

2 Đường tròn trên ASY-3D

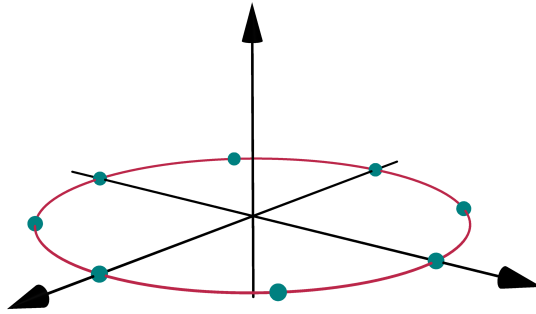
Đường tròn trên ASY-3D cần có 03 tham số: **triple C**, **real r** và **triple normal** - vector pháp tuyến và sử dụng các hàm sau để vẽ đường tròn:

- **circle(triple c, real r, triple normal)**: kiểu **path3**, hàm này trong gói **three** dựa trên đường tròn đơn vị **unitcircle3**, nó sẽ trả về đường cong **Bezier** đi qua 8 điểm gần với đường tròn.

- **Circle(triple c, real r, triple normal)**: trong gói **graph3**;

```
import three;
unitsize(1cm);

draw(circle((0,0,0), 1, normal=Z), blue);
draw(-1.2X--1.5X, Arrow3); draw(-1.2Y--1.5Y, Arrow3); draw(-0.4Z--Z, Arrow3);
```



Trong ví dụ trên sử dụng hàm kiểu **path3 circle(triple C, real r, triple normal)** trong gói **three** để định nghĩa đường tròn.

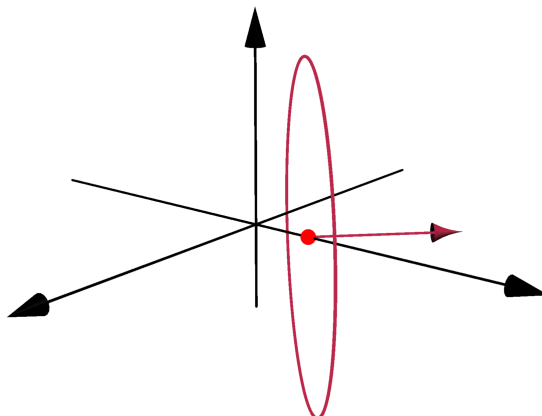
Nếu kiểm tra, ta sẽ thấy chỉ có 8 điểm trên đường tròn có khoảng cách đến tâm $\approx 1!$. Đường tròn này không chính xác!

Tương tự như 2D, gói **graph3** có định nghĩa đường tròn kiểu **path3 Circle(triple c, real r, triple normal=Z, int n=nCircle)**.

```
import three;
import graph3;
unitsize(3cm);

triple mC=(0,0.3, 0); dot(mC);
triple normal=(-0.5, 0.5, 0);
path3 Co = Circle(mC, 0.8, normal=normal, n=100); draw(Co, deepcyan);
triple nC=normal+mC-0;
draw(mC--nC, deepcyan, Arrow3(HookHead3, size=10bp));

draw(-1.2X--1.5X, Arrow3); draw(-1.2Y--1.5Y, Arrow3); draw(-0.4Z--Z, Arrow3);
```

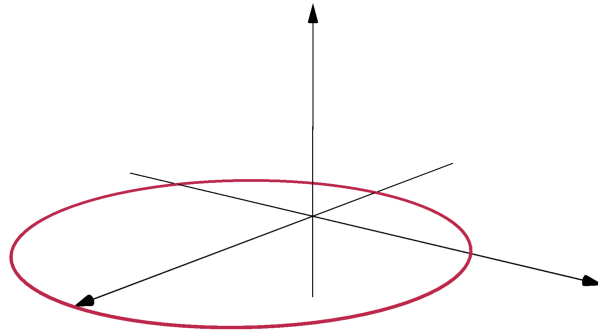


Ngoài ra ta cũng có thể sử dụng hàm **circle circle(point C, real r)** của gói **geometry 2D** để vẽ đường tròn trên 3D.

```
import three;
import geometry;
unitsize(1cm);
pen mycolor=RGB(187, 38, 73);
```

```
circle co=circle((point)(0,0), 1);
draw(path3(co), mycolor);

draw(-1.2X--1.5X, Arrow3); draw(-1.2Y--1.5Y, Arrow3); draw(-0.4Z--Z, Arrow3);
```



Trong đoạn code trên **path3(co)** chuyển từ **path** trên 2D thành **path3** trên 3D. Tuy nhiên cách này áp dụng trên mặt phẳng **OXY**, nếu muốn vẽ đường tròn trên mặt phẳng bất kỳ sẽ phức tạp hơn.

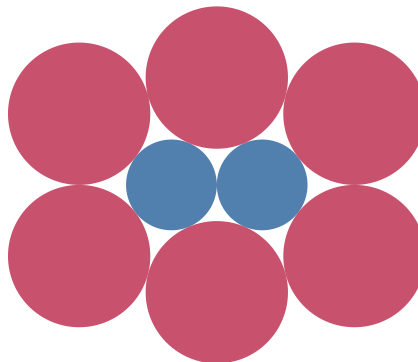
Tính chính xác của đường tròn trên ASY-3D sẽ được nói đến sau.

3 Problem

P1. Viết hàm vẽ đường tròn *A-Mixtilinear* (đường tròn tiếp xúc với 2 cạnh AB, AC và tiếp xúc trong với $\odot(ABC)$).

P2. Viết hàm tìm tâm nội tiếp của tam giác $\triangle ABC$ trên 3D.

P3. Vẽ hình dưới.



4 Tài liệu tham khảo

- [1]. Tài liệu Asymptote: the Vector Graphics Language
- [2]. Các ví dụ của Asymptote
- [3]. Tài liệu Asymptote Démarrage rapide
- [4]. Gói geometry.asy và olympiad.asy
- [5]. Tài liệu hướng dẫn gói **geometry.asy**: Euclidean geometry with asymptote
- [6]. Tài liệu hướng dẫn Asymptote 3D: Asymptote 3D, Bruno M. Colombel
- [7]. Tài liệu hướng dẫn ASY (tiếng Việt): ASY for Beginners, Trần Quân