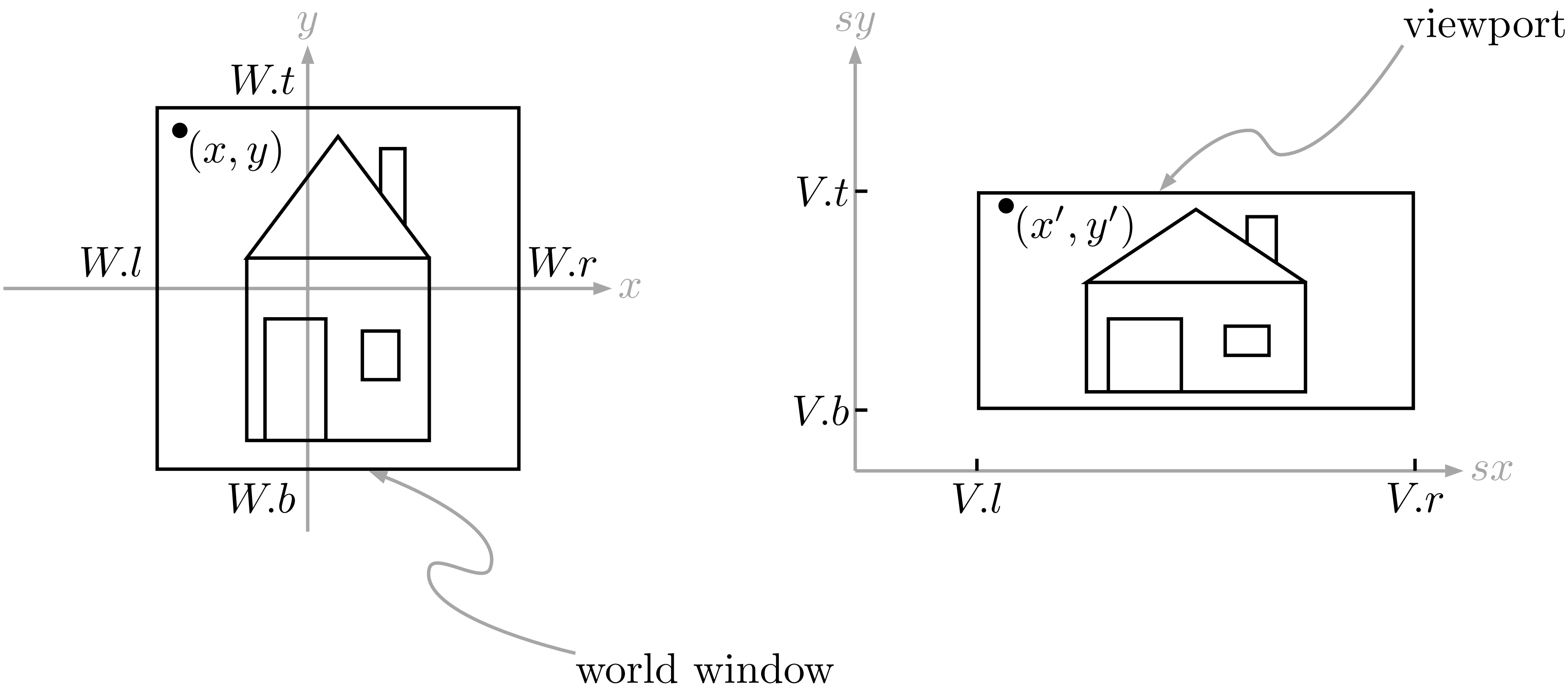
RW778 Graphics

Lecture 2: Windows & viewports

Windows & viewports

- Instead of drawing “on the screen” and using the screen coordinate system $(0 \ldots screenWidth - 1, 0 \ldots screenHeight - 1)$ we allow the user to draw more generally.
- We define the *world* as an infinite Cartesian plane on which the user can draw.
- A *world window* represented by a rectangle W defines what part of the world the user sees on the screen.
- A *viewport* represented by rectangle V defines where the user sees the world window and how it is distorted (scaled + shifted).
- Question: how do we map the contents of the world window to the viewport?

Window to viewport mapping

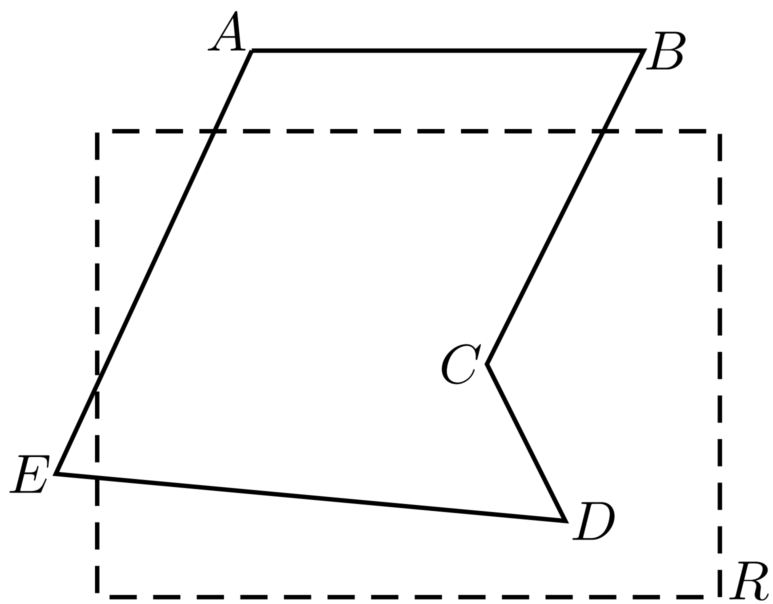


Window to viewport mapping

- $x' = Ax + C$
- $y' = Bx + D$
- $A = \frac{V.r - V.l}{W.r - W.l}$
- $C = V.l - AW.l$
- A, B constitute a scaling factor
- C, D constitute an offset

Clipping lines

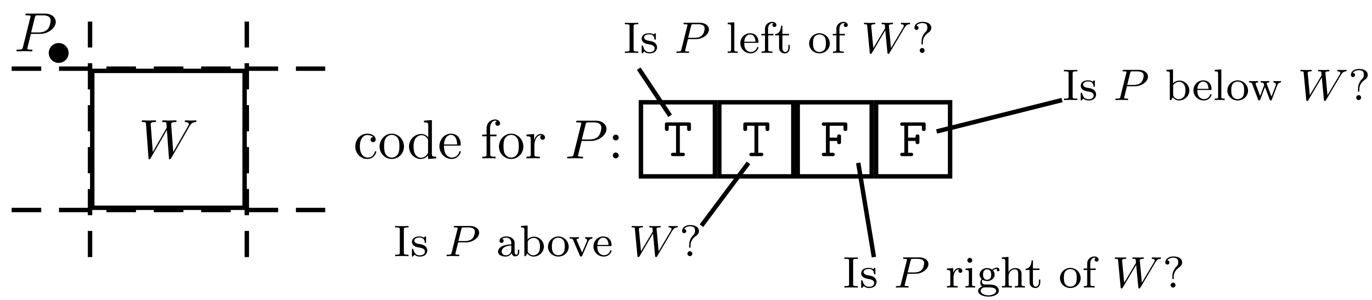
- *Clipping* is a fundamental task in graphics needed to compute which part of an object lies outside a given region and does not need to be drawn.
- Why is clipping necessary?
- OpenGL automatically clips your drawings for you, but the ideas that are involved in clipping are basic and arise in number of different situations.



function $\texttt{clip}(p_1, p_2, R)$:

1. If the entire line lies outside the window (e.g., AB), the function returns 0.
2. If the entire line lies inside the window (e.g., CD), the function returns 1.
3. If one endpoint is inside the window, and one outside (e.g., ED), the function clips the portion of the segment outside the window and returns 1.
4. If both endpoints are outside the window, but a portion of the segment passes through the window (e.g., AE), the function clips both ends and returns 1.

- The Cohen-Sutherland clipping algorithm quickly detects two common cases called “trivial reject” (case 1.) and “trivial accept” (case 2.)
- For each endpoint, an *inside-outside code word* is computed, as follows:



- There are nine different possible code words:

TTFF	FTFF	FTTF
TFFF	FFFF	FFTF
TFFT	FFFT	FFTT

- Trivial accept: both code words are `FFFF`
  Trivial reject: both code words have a `T` in the *same* position

Chopping

$A = (W.right, ?)$

$\dfrac{e}{\delta x} = \dfrac{d}{\delta y}$

$\delta x = p_1.x - p_2.x$

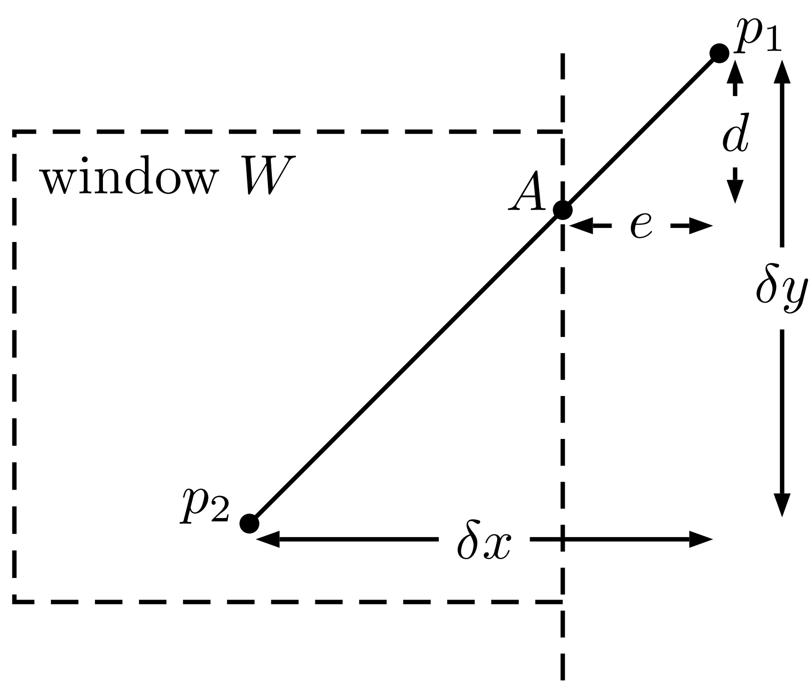
$\delta y = p_1.y - p_2.y$

$e = p_1.x - W.right$

$d\delta x = e\delta y$

$d = (p_1.x - W.right) \cdot \delta y/\delta x$

$A = (W.right, p_1.y - d)$

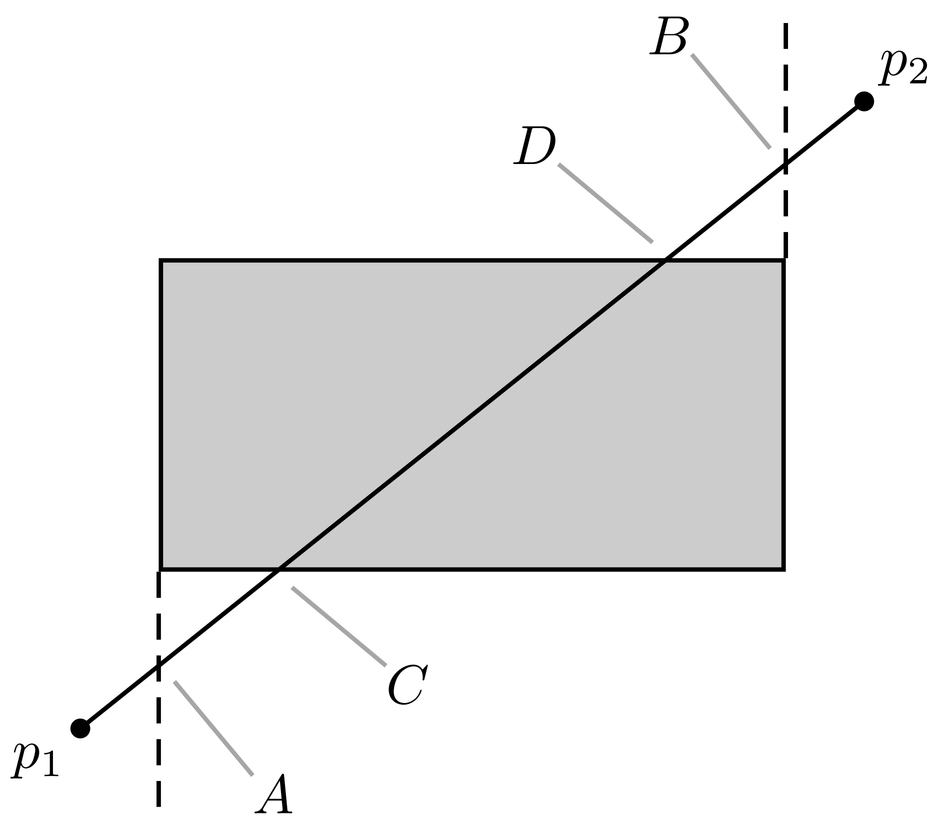


- Potential problem: $\delta x = 0$ or $\delta y = 0$
- What are the equations for the other cases?

The Cohen-Sutherland line clipper

```
int clip(p₁, p₂, R) {
  do {
    calculate the code words for p₁ and p₂
    if (trivial reject) return 0;
    if (trivial accept) return 1;
    if (p₁ is outside) {
      if (p₁ is to the left) chop against the left edge
      else if (p₁ is to the right) chop against the right edge
      else if (p₁ is below) chop against the bottom edge
      else if (p₁ is above) chop against the top edge
    }
    else { ··· } /* p₂ is outside */
  } while (1);
}
```

- In the worst case, the algorithm requires four clips:



- In what order will the clipping happen?

Curve descriptions

- There are two principle ways of describing the shape of a curved line: implicitly and parametrically.
- The *implicit form* involves a function $F(x, y)$ that provides the relationship between x and y coordinates. Point (a, b) lies on the curve if and only if $F(a, b) = 0$.
- The *parametric form* produces a curve based on the value of a parameter. The path of a particle that travels along the curve is fixed by two functions $x(\,)$ and $y(\,)$, and we speak of $(x(t), y(t))$ as the *position* of the particle at time t.

Curve descriptions: lines

Straight line with gradient m and offset c:

- Implicit form: $F(x, y) = y - mx - c$
- Parametric form: $$\begin{aligned} x(t) &= t \\ y(t) &= mt + c \end{aligned}$$

Straight line through points P and Q:

- Implicit form: $F(x, y) = (y - P_y)(Q_x - P_x) - (x - P_x)(Q_y - P_y)$
- Parametric form: $$\begin{aligned} x(t) &= P_x + (Q_x - P_x)t \\ y(t) &= P_y + (Q_y - P_y)t \end{aligned}$$

Curve descriptions: ellipses

Circle with radius R

- Implicit form: $F(x, y) = x^2 + y^2 - R^2$
- Parametric form: $$\begin{aligned} x(t) &= R\cos(t) \\ y(t) &= R\sin(t) \end{aligned}$$

Ellipse with half width W and half height H:

- Implicit form: $F(x, y) = (x/H)^2 + (y/W)^2 - 1$
- Parametric form: $$\begin{aligned} x(t) &= W\cos(t) \\ y(t) &= H\sin(t) \end{aligned}$$

An important variation of the ellipse is the *superellipse*:

$$\left(\frac{x}{W}\right)^n + \left(\frac{y}{H}\right)^n = 1.$$

where n is a parameter called the *bulge*. It's parametric form is

$$\begin{aligned} x(t) &= W\cos(t)|\cos(t)^{2/n-1}| \\ y(t) &= H\sin(t)|\sin(t)^{2/n-1}| \end{aligned}$$

Superellipses were first studied by French physicist George Lamé in 1818. More recently, the extraordinary Danish scientist and poet Piet Hein used a superellipse with $n = 2.5$ for the design of Sergels Torg, a square at the intersection of two large roads in Stockholm.

TIME

Does time exist?
I gravely doubt it.
But gosh, what should we do
without it?

THE ROAD TO WISDOM

The road to wisdom? – Well, it's plain
and simple to express:
    Err
    and err
    and err again
    but less
    and less
    and less.

Polar coordinates

- Polar coordinates can be used to represent many interesting curves.
- Each point on the curve is represented by an angle θ and a radial distance r. If θ and r are functions of t, the curve $(r(t), \theta(t))$ is swept out.
- This curve also has the Cartesian representation $(x(t), y(t))$ where

$$\begin{aligned} x(t) &= r(t)\cos(\theta(t)) \\ y(t) &= r(t)\sin(\theta(t)) \end{aligned}$$

- For a large number of appealing curves, a simplification is possible by expressing r as a function of θ. The path of the curve is then $(f(\theta), \theta)$ and the Cartesian coordinates (x, y) are

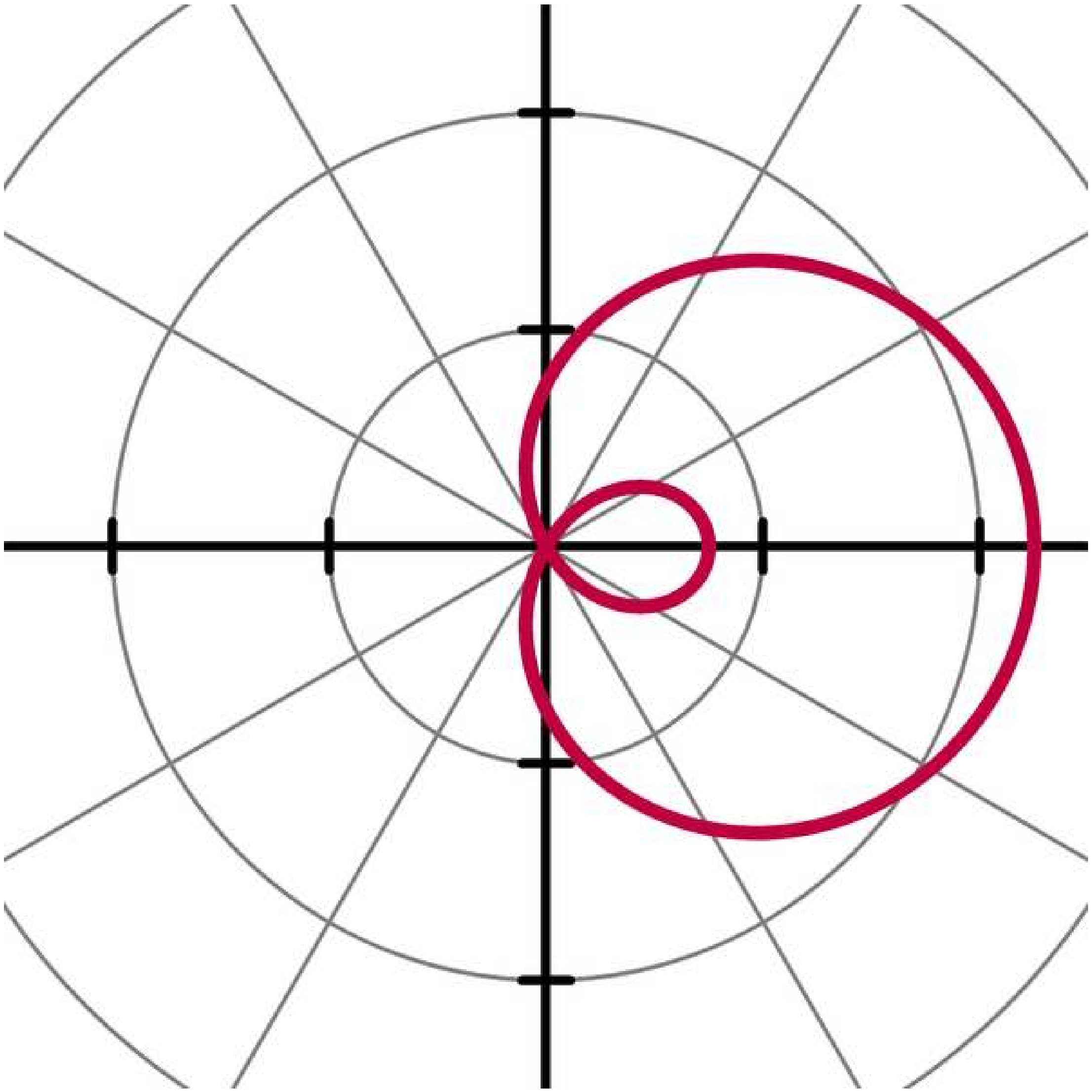
$$\begin{aligned} x &= f(\theta)\cos\theta \\ y &= f(\theta)\sin\theta \end{aligned}$$

Polar coordinates: curves

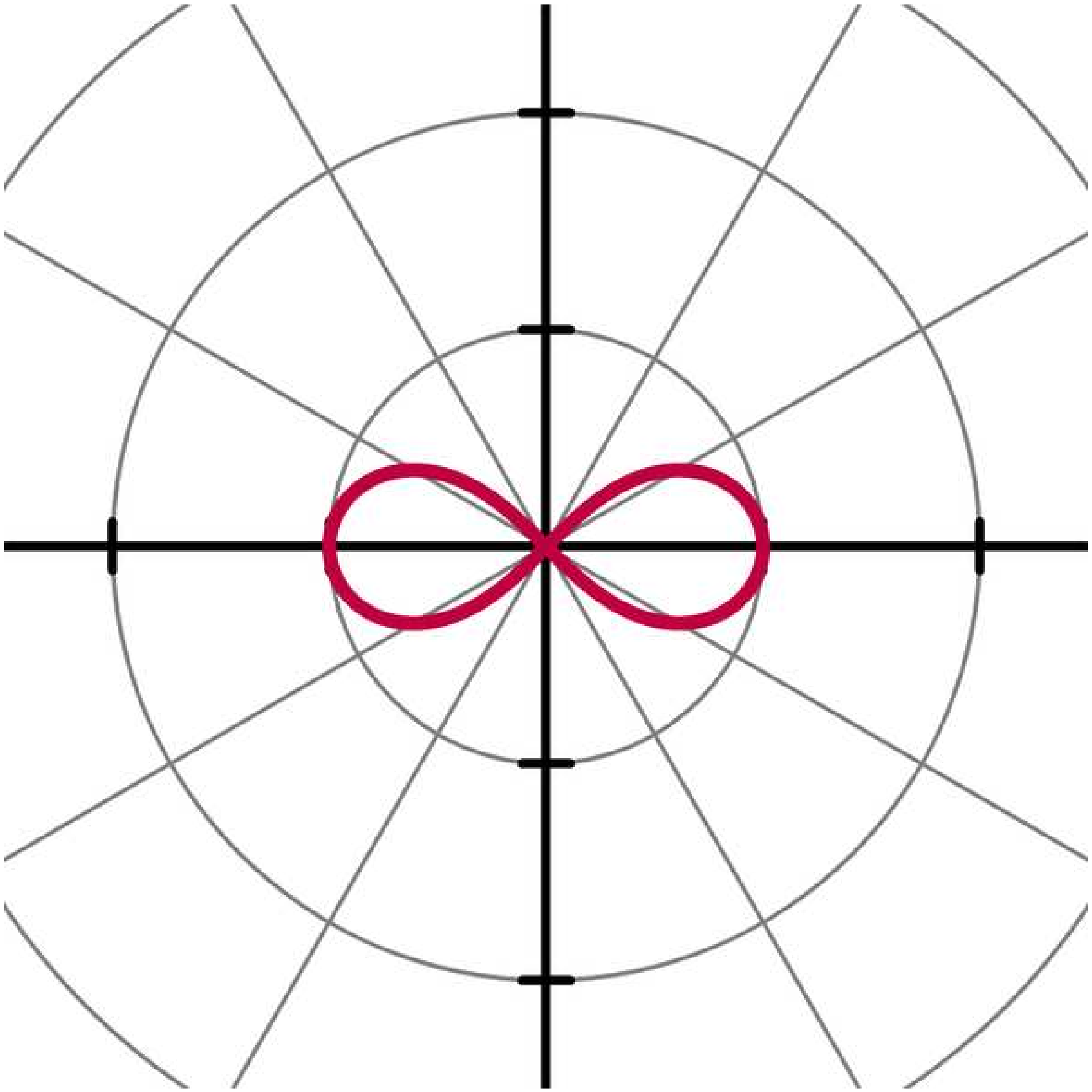
- Circle with radius R: $f(\theta) = R$
- Limaçon: $f(\theta) = a + b\cos\theta$
- Cardioid: $f(\theta) = a + a\cos\theta$
- Lemniscate: $f(\theta) = a\cos 2\theta$
- Rose curves: $f(\theta) = a\cos k\theta$
- Archimedean spiral: $f(\theta) = a + b\theta$
- Conic sections: $f(\theta) = \dfrac{1}{1 \pm e\cos\theta}$

  Parabola: $e = 1$, Ellipse: $0 \leq e < 1$, Hyperbola: $e > 1$

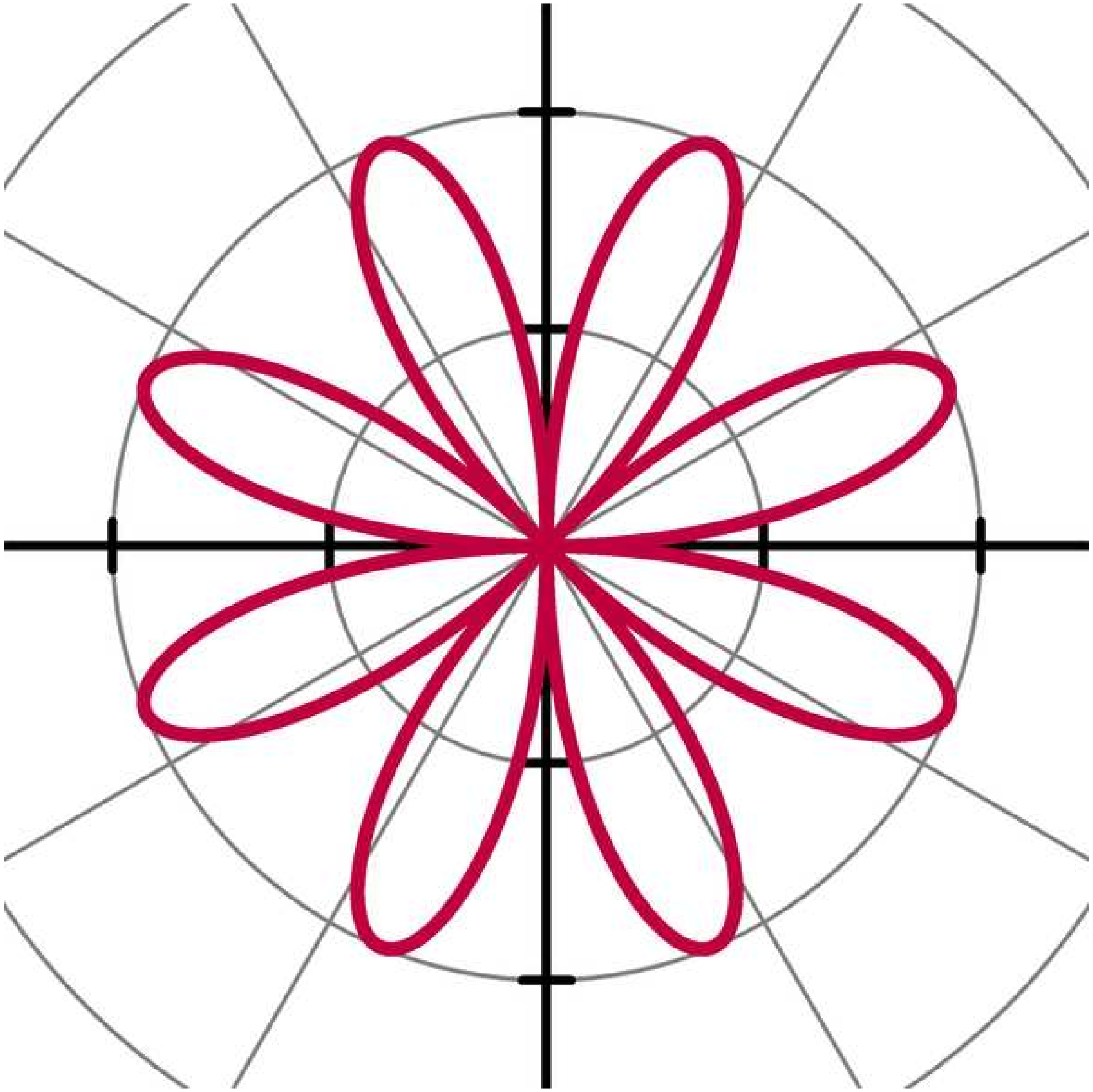
Polar coordinates: limaçon



Polar coordinates: lemniscate



Polar coordinates: polar rose



Polar coordinates: polar rose