

MCA101 : COMPUTER GRAPHICS

2D GEOMETRY REPRESENTATION

Raghav B. Venkataramaiyer

CSED TIET Patiala India.

August 31, 2024

1 2D GEOMETRY — INTRODUCTION

2 MID-POINT ALGORITHM

1 2D GEOMETRY — INTRODUCTION

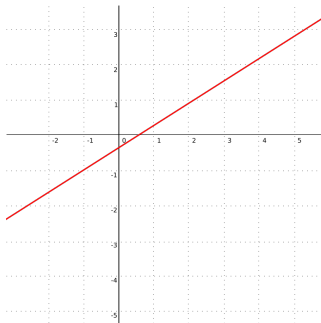
- Straight Lines

- Conics

2 MID-POINT ALGORITHM

$$y = mx + c$$

$$y = f(x) = mx + c$$



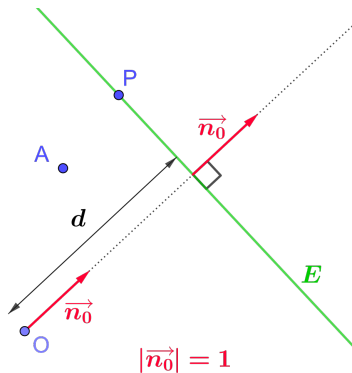
For any two vectors $\mathbf{u}, \mathbf{v} \in V$, a point on the line segment joining them is given parameterised by $t \in [0, 1]$, as

$$\mathbf{p} = f(t) = (1 - t)\mathbf{u} + t\mathbf{v}$$

Any point on a line in the direction of unit vector $\mathbf{u} : \|\mathbf{u}\|_2^2 = 1$, and an incident point \mathbf{p}_0 may be given parameterised by $t \in \mathbb{R}$ as,

$$\mathbf{p} = f(t) = \mathbf{p}_0 + t\mathbf{u}$$

HESSE NORMAL FORM



Given,

Normal to the line $\mathbf{n}_0 : \|\mathbf{n}_0\|_2^2 = 1$, and
its distance from origin, d ;

The point on the line is given implicitly as the locus
of all points \mathbf{p} that satisfy,

$$\mathbf{n}_0 \cdot \mathbf{p} - d = 0$$

Distance from the origin O to the line E
calculated with the Hesse normal form.
Normal vector in red, line in green, point O
shown in blue.

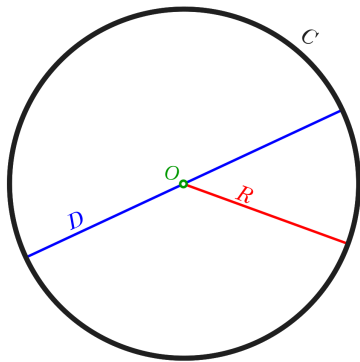
1 2D GEOMETRY — INTRODUCTION

- Straight Lines

- Conics

2 MID-POINT ALGORITHM

CIRCLE



Implicit Form:

$$f\begin{pmatrix} x \\ y \end{pmatrix} = x^2 + y^2 - r^2 = 0$$

Parametric Form:

$$f(r, t) = \begin{bmatrix} r \cos t \\ r \sin t \end{bmatrix}$$

FIGURE: Image Courtesy: [Wikipedia](#)

ELLIPSE

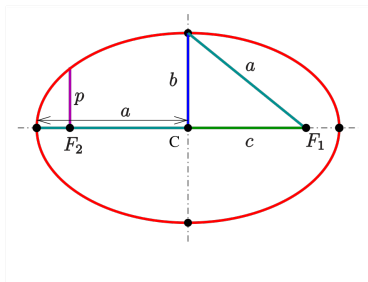


FIGURE: Image Courtesy: [Wikipedia](#)

Standard form

$$f\begin{pmatrix} x \\ y \end{pmatrix} = \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0$$

Parametric Form

$$f(t; a, b) = \begin{bmatrix} a \cos t \\ b \sin t \end{bmatrix}$$

OUTLINE

1 2D GEOMETRY — INTRODUCTION

2 MID-POINT ALGORITHM

1 2D GEOMETRY — INTRODUCTION

2 MID-POINT ALGORITHM

- Fundamentals
- Straight Line
- Circle

PROBLEM

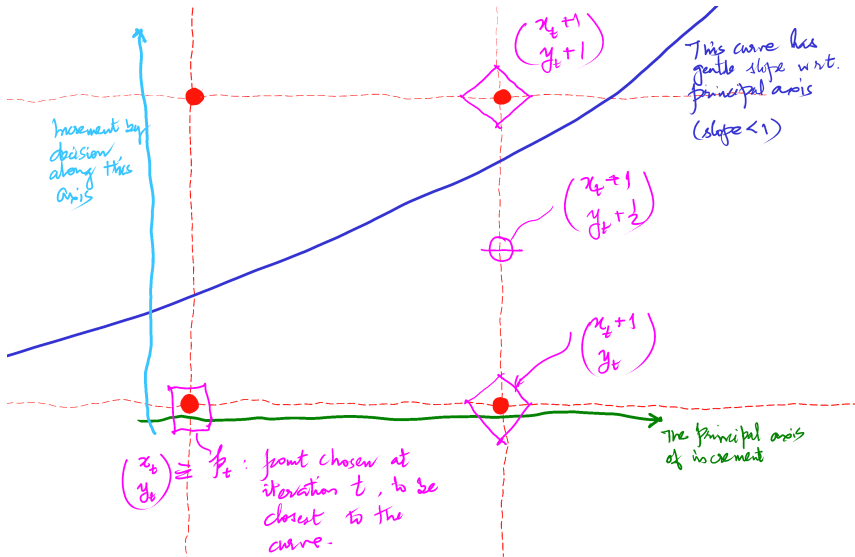
In a quantised (pixelated or discrete) 2d plane, find the set of points that visually approximate a given curve, say a straight line or a conic.

Iteratively, increment along one axes,
with respect to which, the slope of the curve
is gentle.

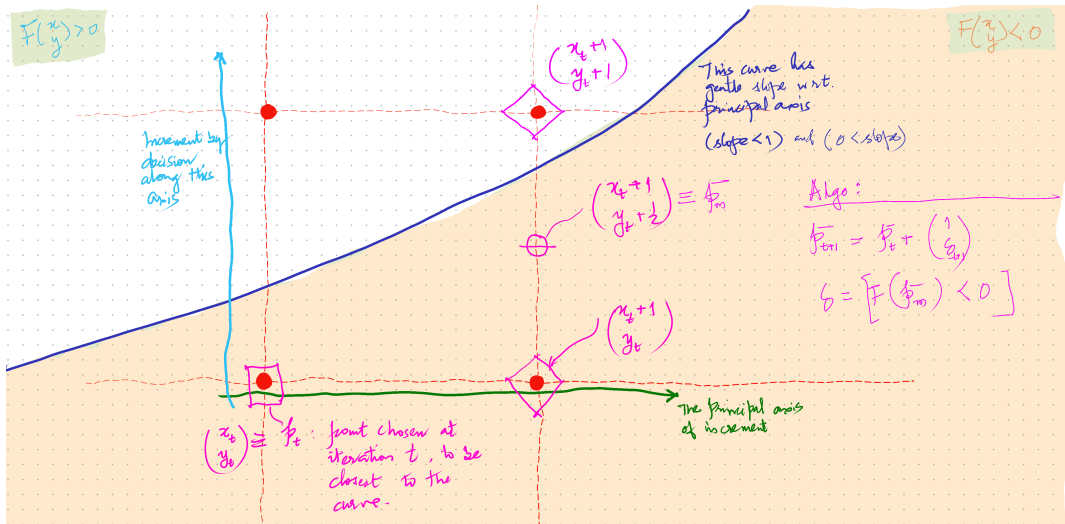
Decide whether it is required to increment
along the perpendicular axis or not.

Increment if required.

EXAMPLE



EXAMPLE



1 2D GEOMETRY — INTRODUCTION

2 MID-POINT ALGORITHM

- Fundamentals
- Straight Line
- Circle

Algorithm 1: Generic Mid-point Algorithm

Input: $x_0, x_{\max} \in \mathbb{Z}$

Start and end x-coordinates.

Input: $F: \mathbb{R}^2 \rightarrow \mathbb{R}$

Signed Distance Function from the curve.

Output: $C \equiv \{\mathbf{p}_0, \dots, \mathbf{p}_{\max}\} \subset \mathbb{Z}^2$

Curve in discrete 2D space.

1 $\mathbf{p}_0 \leftarrow \begin{bmatrix} x_0 \\ \lceil y_0 \rceil \end{bmatrix} \vdash F\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = 0$

2 **for** $t \in \{1, \dots, \max\}$ **do**

3 $\mathbf{p}_{\text{mid}} \leftarrow \mathbf{p}_{t-1} + \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}$

4 $\delta_t \leftarrow I[F(\mathbf{p}_{\text{mid}}) < 0]$

5 $\mathbf{p}_t \leftarrow \mathbf{p}_{t-1} + \begin{pmatrix} 1 \\ \delta_t \end{pmatrix}$

Algorithm 2: Mid-Point Algorithm for Straight Line (intermediate attempt)

Input: $x_0, x_{\max} \in \mathbb{Z}$

Start and end x-coordinates.

Input: $a, b, c \in \mathbb{Z} : 0 \leq a < b; b \text{ even}$

Coefficients: $F(x, y) = ax - by + c$. **Rem:** $-by$.

Output: $C \equiv \{(x_0, y_0), \dots, (x_{\max}, y_{\max})\} \subset \mathbb{Z}^2$

An ordered sequence; a curve in discrete 2D space.

```

1   $y_0 \leftarrow \lceil \frac{ax_0 + c}{b} \rceil$ 
2   $\delta_0 \leftarrow ax_0 - by_0 - \frac{b}{2} + c$ 
3  for  $t \in \{1, \dots, \max\}$  do
4       $x_t \leftarrow x_{t-1} + 1$ 
5       $\delta_t \leftarrow \delta_{t-1} + a - b \cdot I[\delta_{t-1} < 0]$ 
6       $y_t \leftarrow y_{t-1} + I[\delta_t < 0]$ 
    
```

Rem: $-by_0 - \frac{b}{2}; \frac{b}{2} \in \mathbb{Z}$ because b even.

MID-POINT ALGO FOR ST. LINE

Algorithm 3: Mid-Point Algorithm for Straight Line

Input: $x_0, x_{\max} \in \mathbb{Z}$

Start and end x-coordinates.

Input: $a, b, c \in \mathbb{Z} : 0 \leq a < b; b \text{ even}$

Coefficients: $F(x, y) = ax - by + c$. **Rem:** $-by$.

Output: $C \equiv \{(x_0, y_0), \dots, (x_{\max}, y_{\max})\} \subset \mathbb{Z}^2$

An ordered sequence; a curve in discrete 2D space.

1 **Init:** C as a new Array.

2 **Init:** x, y, δ as integers.

3 $x \leftarrow x_0$

4 $y \leftarrow \lceil \frac{ax+c}{b} \rceil$

5 $\delta \leftarrow ax - by - \frac{b}{2} + c$

Rem: $-by - \frac{b}{2}; \frac{b}{2} \in \mathbb{Z}$ because b even.

6 $C \cdot \text{Push}((x, y))$

7 **for** $t \in \{1, \dots, \max\}$ **do**

8 $x \leftarrow x + 1$

9 $\delta \leftarrow \delta + a - b \cdot I[\delta < 0]$

10 $y \leftarrow y + I[\delta < 0]$

11 $C \cdot \text{Push}((x, y))$

1 2D GEOMETRY — INTRODUCTION

2 MID-POINT ALGORITHM

- Fundamentals
- Straight Line
- Circle

Algorithm 4: Generic Mid-point Algorithm

Input: $x_0, x_{\max} \in \mathbb{Z}$

Start and end x-coordinates.

Input: $F: \mathbb{R}^2 \rightarrow \mathbb{R}$

Signed Distance Function from the curve.

Output: $C \equiv \{\mathbf{p}_0, \dots, \mathbf{p}_{\max}\} \subset \mathbb{Z}^2$

Curve in discrete 2D space.

1 $\mathbf{p}_0 \leftarrow \begin{bmatrix} x_0 \\ \lceil y_0 \rceil \end{bmatrix} \vdash F\left(\begin{smallmatrix} x_0 \\ y_0 \end{smallmatrix}\right) = 0$

2 **for** $t \in \{1, \dots, \max\}$ **do**

3 $\mathbf{p}_{\text{mid}} \leftarrow \mathbf{p}_{t-1} + \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}$

4 $\delta_t \leftarrow I[F(\mathbf{p}_{\text{mid}}) < 0]$

5 $\mathbf{p}_t \leftarrow \mathbf{p}_{t-1} + \begin{pmatrix} 1 \\ \delta_t \end{pmatrix}$
