FLAME UNIVERSITY

EVERLASTING learning

FUNDAMENTALS OF COMPUTER GRAPHICS (CSIT304)

RASTER GRAPHICS AND SCAN CONVERSION

CHIRANJOY CHATTOPADHYAY

Associate Professor,

FLAME School of Computation and Data Science

PYTHON CODE

- **Edit the Python Code that you implemented for Bresenham line drawing algorithm to make it work for slope m>1. Plot the line between the points (1,1) to (5,8).**

PYTHON CODE

- **Edit the Python Code that you implemented for Bresenham line drawing algorithm to make it work for slope m<=1. Plot the line between the points (8,5) to (1,1). [Negative slope]**

PYTHON CODE

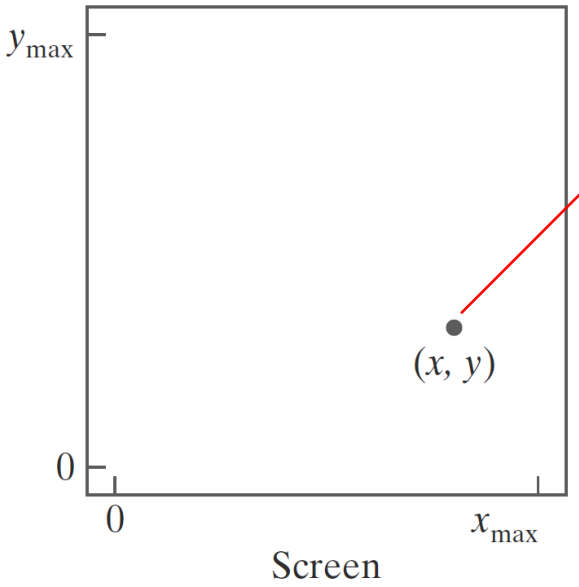
- **Write a Python Code that Plots which creates a Triangle using the line drawing algorithm you implemented. [Polyline Drawing]**

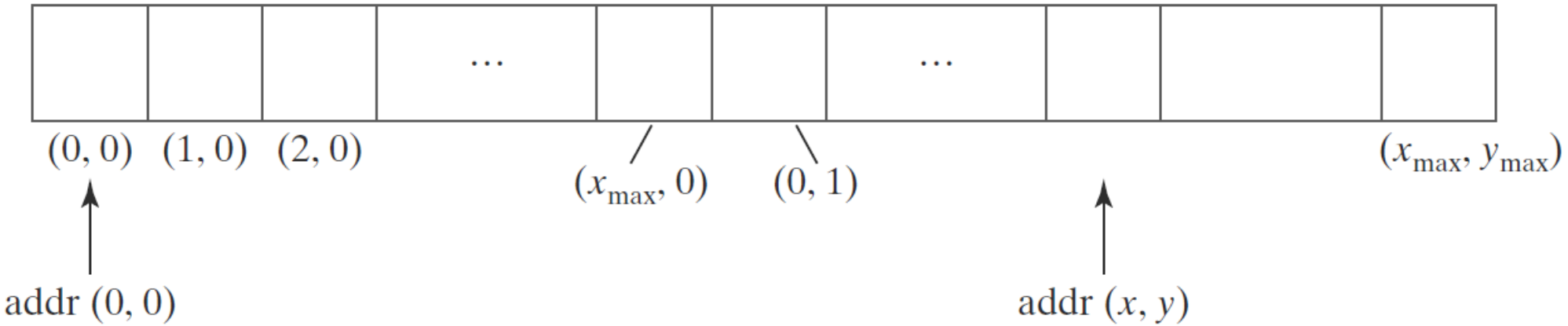
SETTING FRAME-BUFFER VALUES

$$addr(x,y) = Base + Offset$$
$$= addr(0,0) + Offset$$
$$= addr(0,0) + y(x_{max} + 1) + x$$

$$addr(x+1,y) = addr(0,0) + y(x_{max} + 1) + x + 1$$
$$= addr(x,y) + 1$$

$$addr(x+1,y+1) = addr(x,y) + x_{max} + 2$$

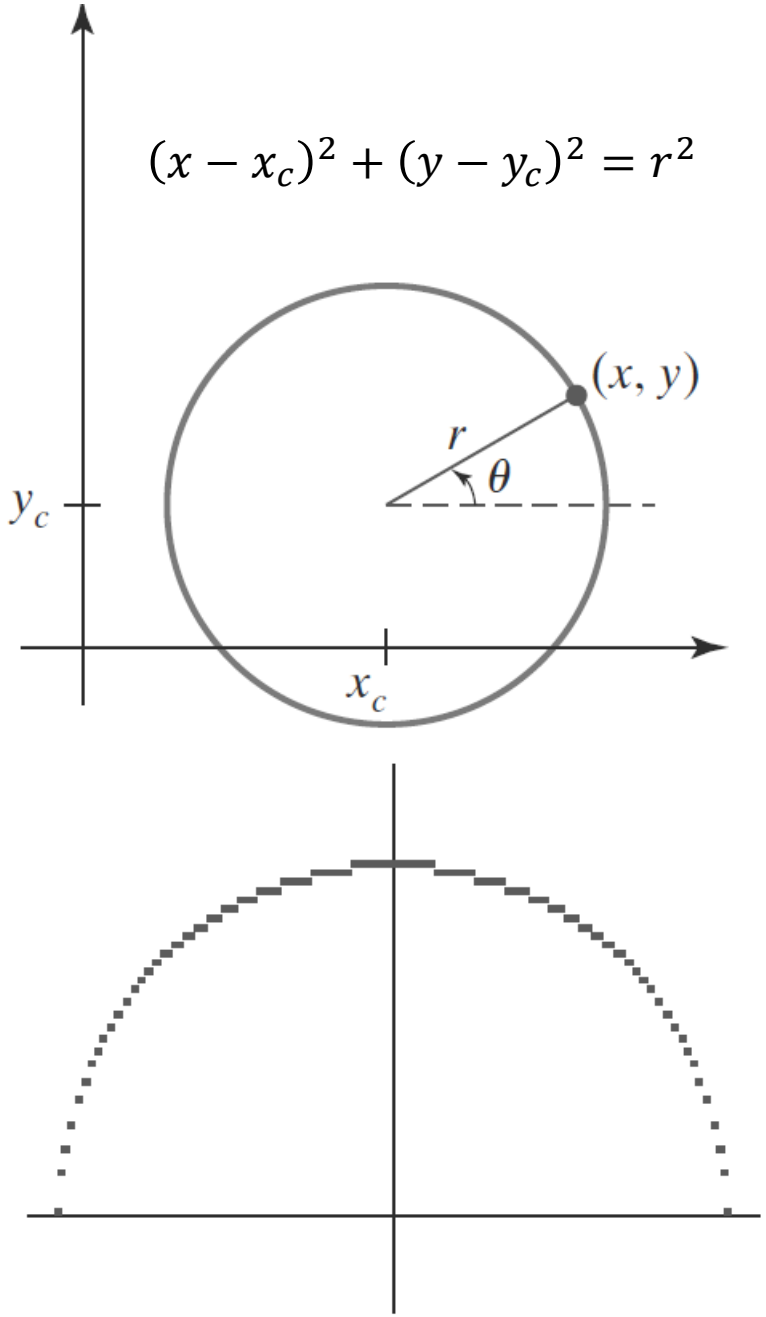




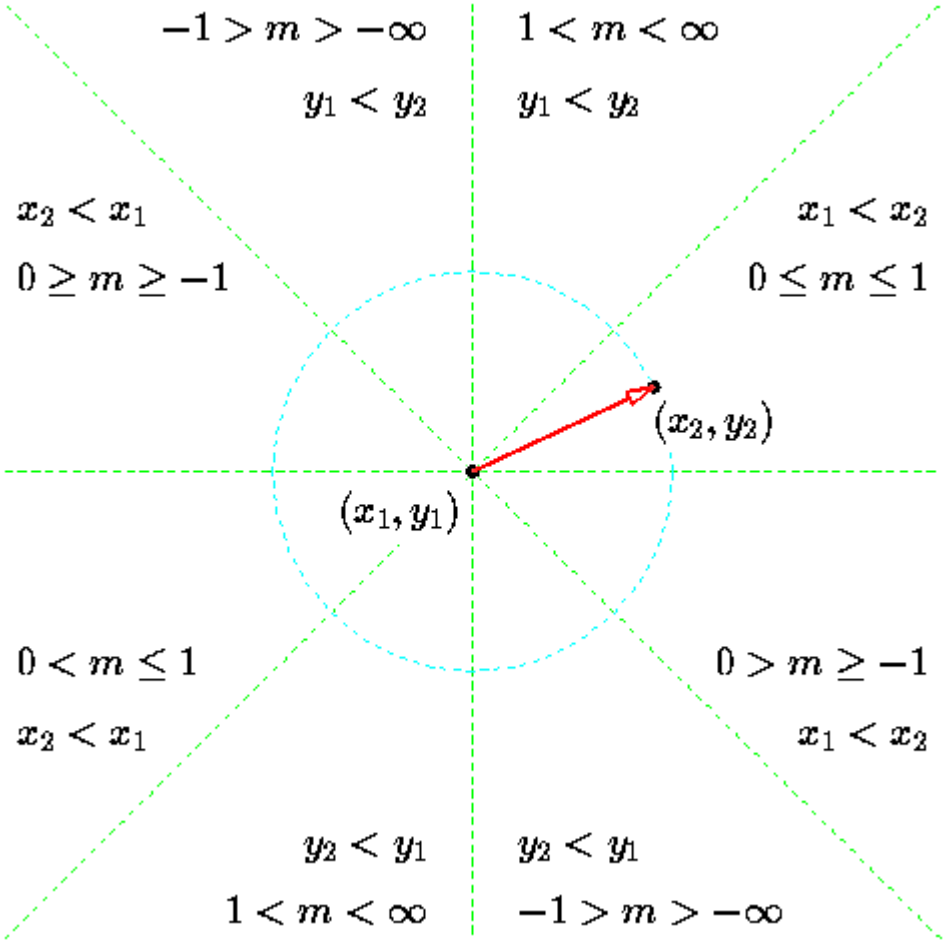
SCAN CONVERTING A CIRCLE

- Circle is a frequently used component in pictures and graphs
- Naïve Approach?
- Ineffective
  - It involves: squaring, taking roots and ROUND
  - It gives an asymmetric distribution.

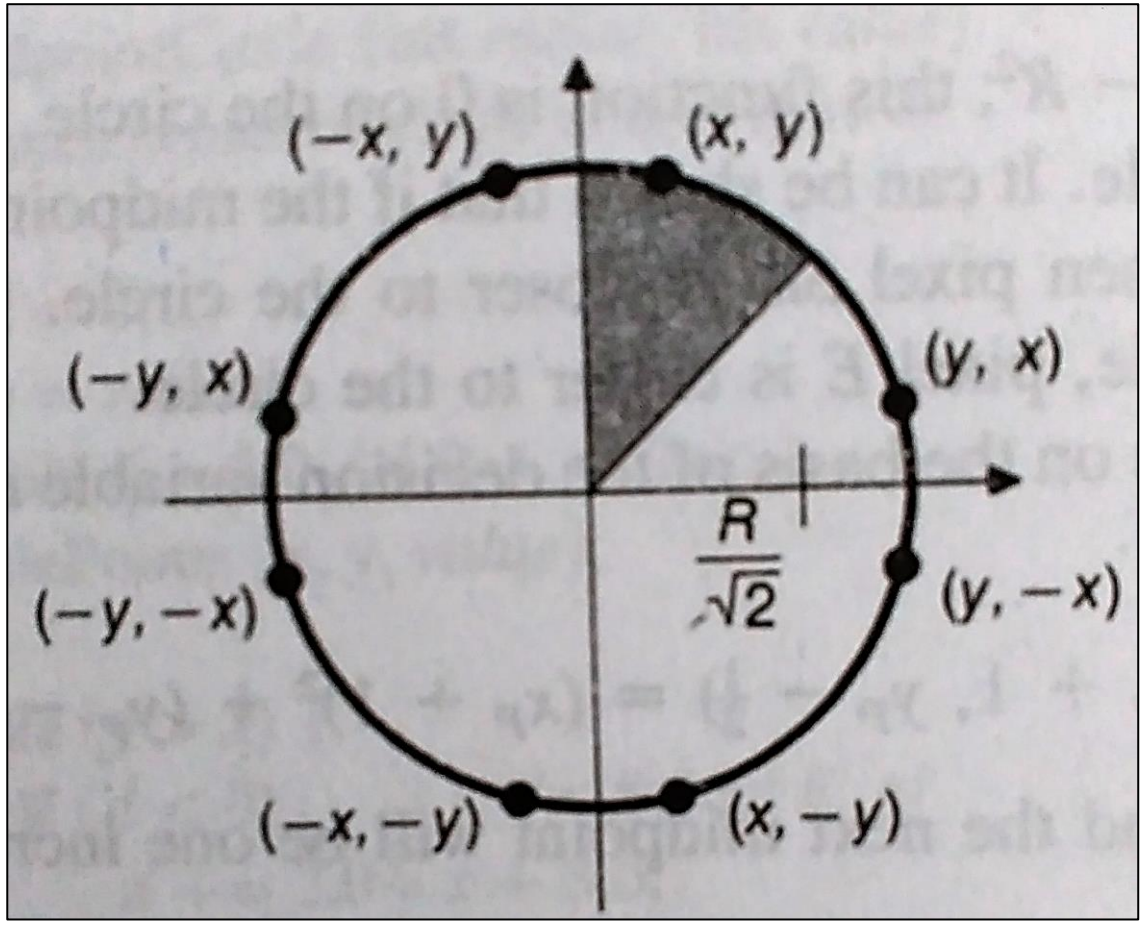
High computational requirement



GENERALIZATION FOR DIFFERENT OCTANTS



EIGHT FOLD SYMMETRY

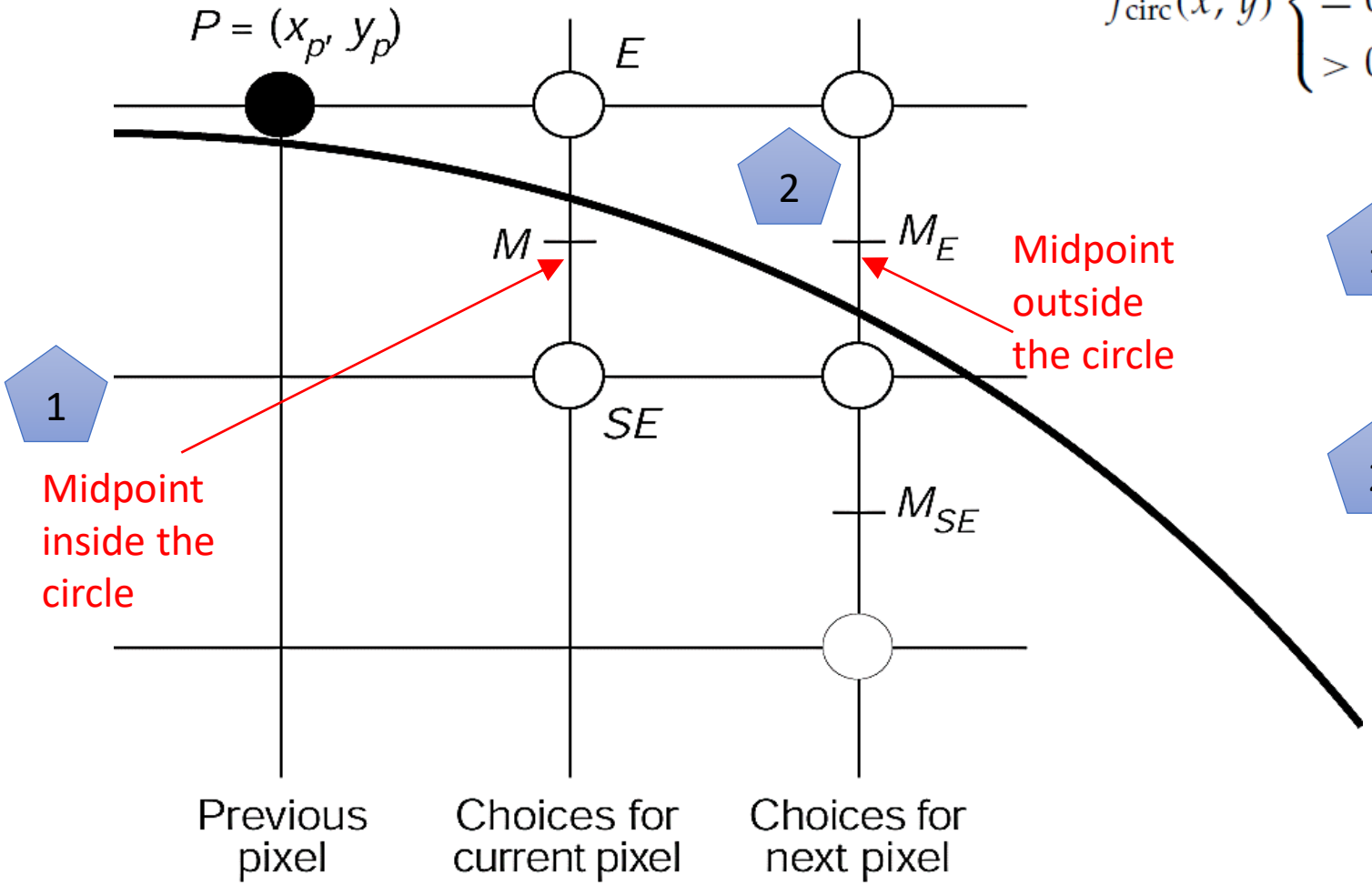


- Only have to draw 1/8 of the circle
- From x=0 to x=y
- Cartesian calculation is still expensive
- Integer operation
- Bresenham's Circle Drawing (Midpoint)

MIDPOINT CIRCLE ALGORITHM

$$f_{\text{circ}}(x, y) = x^2 + y^2 - r^2$$

$$f_{\text{circ}}(x, y) \begin{cases} < 0, & \text{if } (x, y) \text{ is inside the circle boundary} \\ = 0, & \text{if } (x, y) \text{ is on the circle boundary} \\ > 0, & \text{if } (x, y) \text{ is outside the circle boundary} \end{cases}$$



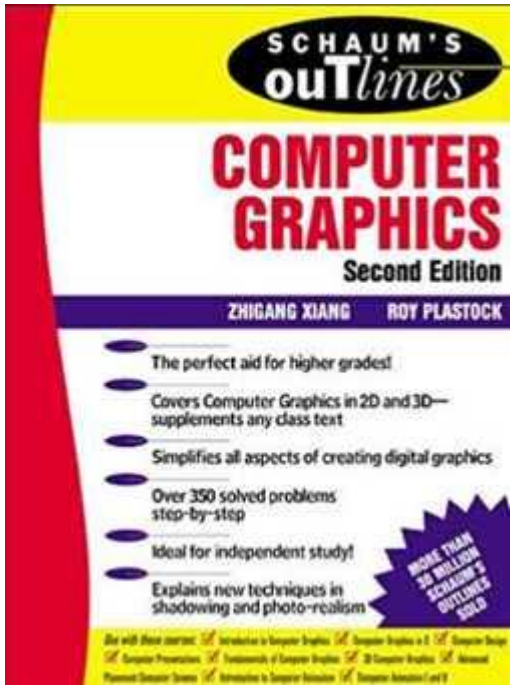
1 Position $(x_k + 1, y_k)$ is closer to the circle.

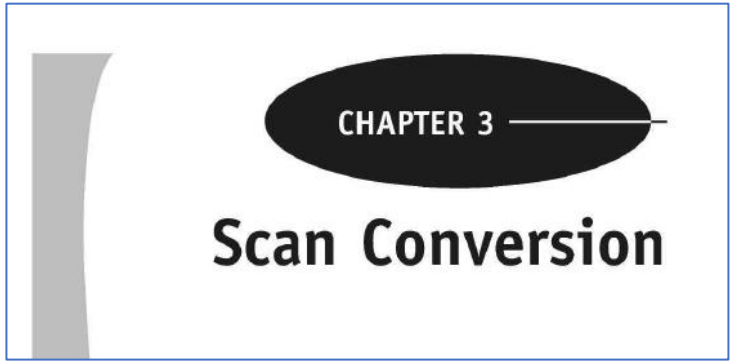
2 Position $(x_k + 1, y_k - 1)$ is closer to the circle.

CIRCLE DRAWING ALGORITHM

```
void MidpointCircle (int radius, int value)
/* Assumes center of circle is at origin */
{
    int x = 0;
    int y = radius;
    double d = 5.0 / 4.0 - radius;
    CirclePoints (x, y, value);

    while (y > x) {
        if (d < 0)              /* Select E */
            d += 2.0 * x + 3.0;
        else {                  /* Select SE */
            d += 2.0 * (x - y) + 5.0;
            y--;
        }
        x++;
        CirclePoints (x, y, value);
    }  /* while */
}  /* MidpointCircle */
```





Please refer this text for the details of Derivation.

- **Given a circle (centered at the origin) radius r = 10, write a Python Code that determines position along the circle octant in the first quadrant from x = 0 to x = y as per the midpoint circle algorithm.**

FLAME UNIVERSITY | EVERLASTING learning

THANK YOU