Computer Graphics (L05)

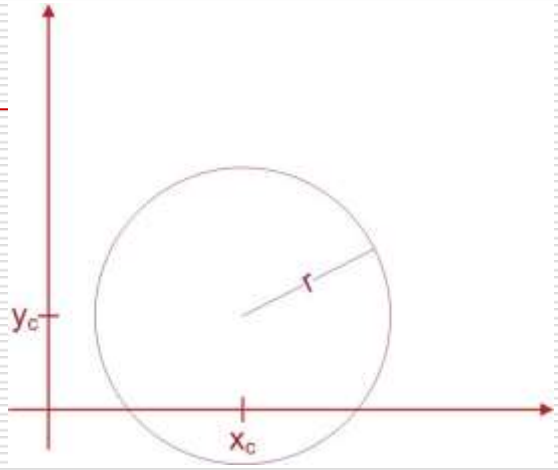
EG678EX

2-D Algorithms

Circle-Generating Algorithms (Basic Foundations)

- Circle Equation:

$$(x-x_c)^2+(y-y_c)^2=r^2$$



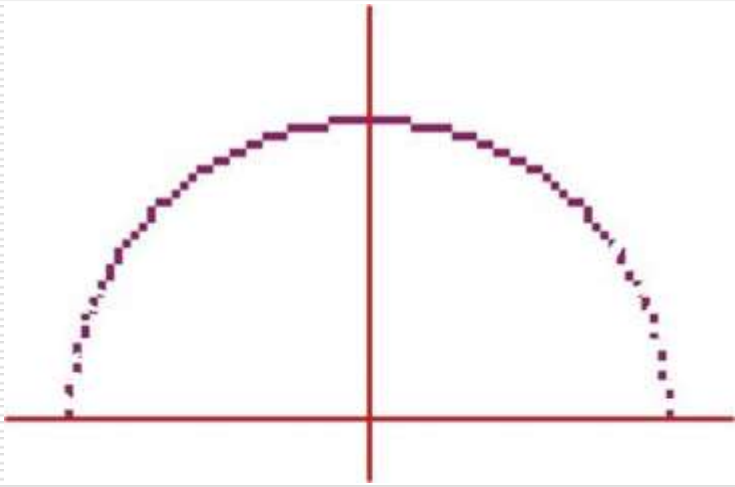
- Points along circumference could be calculated by stepping along x-axis:

$$y=y_c\pm\sqrt{r^2-(x_c-x)^2}$$

- Problem ????

Problem (in above method)

- Computational complexity
- Spacing:
  - Non-uniform spacing of plotted pixels



Adjustments (To fix problems)

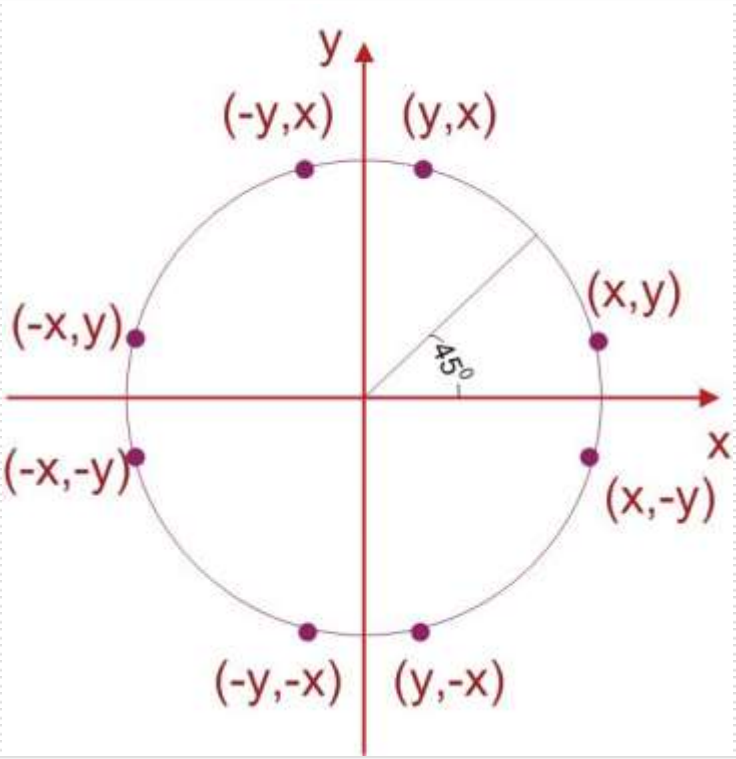
- Spacing problem (2 ways):
  - Interchange the role of x and y whenever the absolute value of the slope of the circle tangent > 1
  - Use polar co-ordinate:

$$x = x_c + r\cos\theta$$

$$y = y_c + r\sin\theta$$

    - Equally spaced points are plotted along the circumference with fixed angular step size.
    - step size chosen for θ depends on the application and display device.
      - For more continuous boundary on a raster display, set step size at 1/r; i.e pixel positions are approximately 1 unit apart.
- Computation Problem:
  - Use symmetry of circle; i.e calculate for one octant and use symmetry for others.

Circle Symmetry



Bresenham's Algorithm Could Be Adapted ??

- Yes
- How ?
  - Setting decision parameter for finding the closest pixel to the circumference
- And what to do For Non-linear equation of circle ?
  - Comparison of squares of the pixel separation distance avoids square root calculations