Computer Graphics

Lab 3

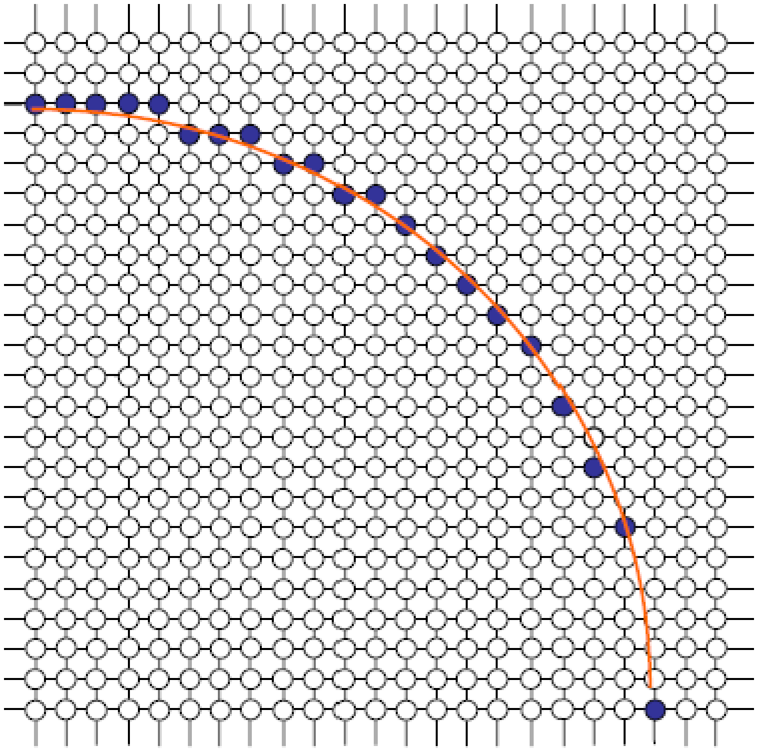
Midpoint Circle Algorithm

Eng.Takwa

A Simple Circle Drawing Algorithm

- The equation for a circle is:
- where *r* is the radius of the circle
- So, we can write a simple circle drawing algorithm by solving the equation for *y* at unit *x* intervals using:
- $(x)^2 + (y)^2 = r^2$

A Simple Circle Drawing Algorithm



$$y_0 = \sqrt{20^2 - 0^2} \approx 20$$

$$y_1 = \sqrt{20^2 - 1^2} \approx 20$$

$$y_2 = \sqrt{20^2 - 2^2} \approx 20$$

$$\vdots$$

$$y_{19} = \sqrt{20^2 - 19^2} \approx 6$$

$$y_{20} = \sqrt{20^2 - 20^2} \approx 0$$

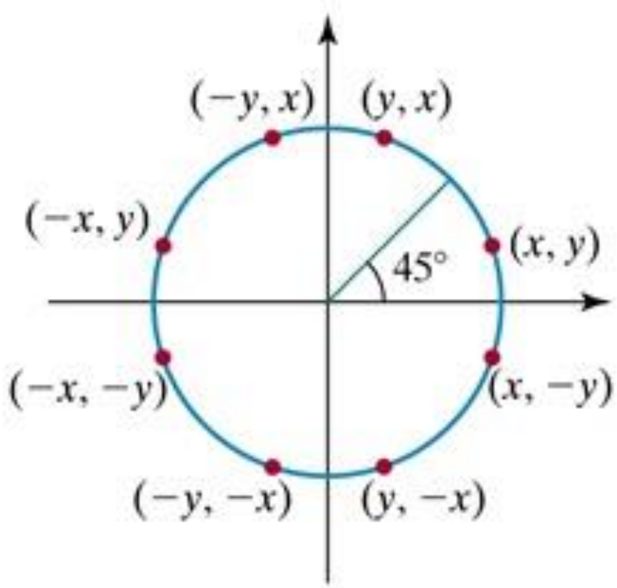
A Simple Circle Drawing Algorithm

- However, unsurprisingly this is not a brilliant solution!
- Firstly, the resulting circle has large gaps where the slope approaches the vertical
- Secondly, the calculations are not very efficient
- –The square (multiply) operations
- –The square root operation –try really hard to avoid these!

- We need a more efficient, more accurate solution

Eight-Way Symmetry

- The first thing we can notice to make our circle drawing algorithm more efficient is that circles centred at (*0, 0*) have *eight-way symmetry*



Eight-Way Symmetry

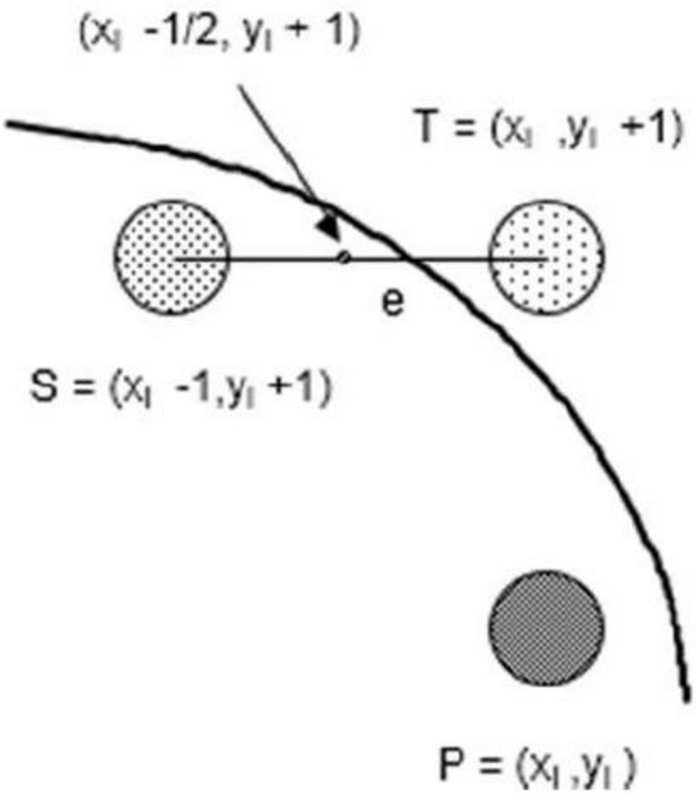
- Consider **symmetry of circles**
- Shape of the circle is similar in each quadrant
- i.e. if we determine the curve positions in the 1stquadrant, we can generate the circle section in the 2ndquadrant of the xy plane (the 2 circle sections are symmetric with respect to the y axis)
- The circle section in the 3rdand 4thquadrant can be obtained by considering symmetry about the x axis
- One step further ⍰symmetry between octants
- Calculation of a circle point (x, y) in 1 octant yields the circle points for the other 7 octants

Bresenham to Midpoint

- Bresenham's line algorithm for raster displays is adapted to circle generation by setting up decision parameters for finding the closest pixel to the circumference at each sampling step.
- Bresenham's circle algorithm avoids square-root calculations by comparing the squares of the pixel separation distances.

Midpoint Circle Algorithm

For any given pixel (x, y), the next pixel to be plotted is either **(x, y+1)** or **(x-1, y+1)**. This can be decided by following the steps below.



Midpoint Circle Algorithm

- Find the mid-point **p** of the two possible pixels i.e (x-0.5, y+1)
- If **p** lies inside or on the circle perimeter, we plot the pixel (x, y+1), otherwise if it's outside we plot the pixel (x-1, y+1)
- **Boundary Condition :** Whether the mid-point lies inside or outside the circle can be decided by using the formula:-
- *Given a circle centered at (0,0) and radius r and a point p(x,y)*
  $F(p) = x^2 + y^2 - r^2$

if $F(p)<0$, the point is inside the circle

$F(p)=0$, the point is on the perimeter

$F(p)>0$, the point is outside the circle

Midpoint Circle Algorithm

- F(p) with P. The value of P is calculated at the mid-point of the two contending pixels i.e. (x-0.5, y+1). Each pixel is described with a subscript k.
- $\boldsymbol{P_k = (X_k - 0.5)^2 + (y_k + 1)^2 - r^2}$
- *Now,*

  $x_{k+1} = x_k$ *or* x_{k-1} , $y_{k+1} = y_k + 1$

$\therefore P_{k+1} = (x_{k+1} - 0.5)^2 + (y_{k+1} + 1)^2 - r^2$

$= (x_{k+1} - 0.5)^2 + [(y_k + 1) + 1]^2 - r^2$

$= (x_{k+1} - 0.5)^2 + (y_k + 1)^2 + 2(y_k + 1) + 1 - r^2$

$= (x_{k+1} - 0.5)^2 + [P_k - (X_k - 0.5)^2 + r^2] + 2(y_k + 1) - r^2 + 1$

$= P_k + (x_{k+1} - 0.5)^2 - (x_k - 0.5)^2 + 2(y_k + 1) + 1$

$= P_k + (x^2_{k+1} - x_k^2) + (x_k - x_{k+1}) + 2(y_k + 1) + 1$

= $P_k + 2(y_k + 1) + 1$, when $P_k <= 0$ i.e the midpoint is inside the circle ($x_{k+1} = x_k$)

$P_k + 2(y_k + 1) - 2(x_k - 1) + 1$, when $P_k > 0$ I.e the mid point is outside the circle($x_{k+1} = x_k - 1$)

Eng.Takwa 17/11/2021

Midpoint Circle Algorithm

- The first point to be plotted is (r, 0) on the x-axis. The initial value of P is calculated as follows:-
- $P1 = (r – 0.5)^2 + (0+1)^2 – r^2$
  $= 1.25 – r$
  = 1 -r (When rounded off)

Example:

- Draw using Midpoint algorithm circle with given radious=10
- **Step 1**: Obtain the first point on the circumference of the circle centered on the origin as **(X_0, Y_0)=(r,0)=(10,0)**

- **Step 2:** Calculate the starting value for the decision

parameter as P_0

- $P_0 = 1 - r = -9$

- **Step 3: set K=0**

If $P_k <= 0$

$P_{k+1} = P_k + 2(y_k + 1) + 1$

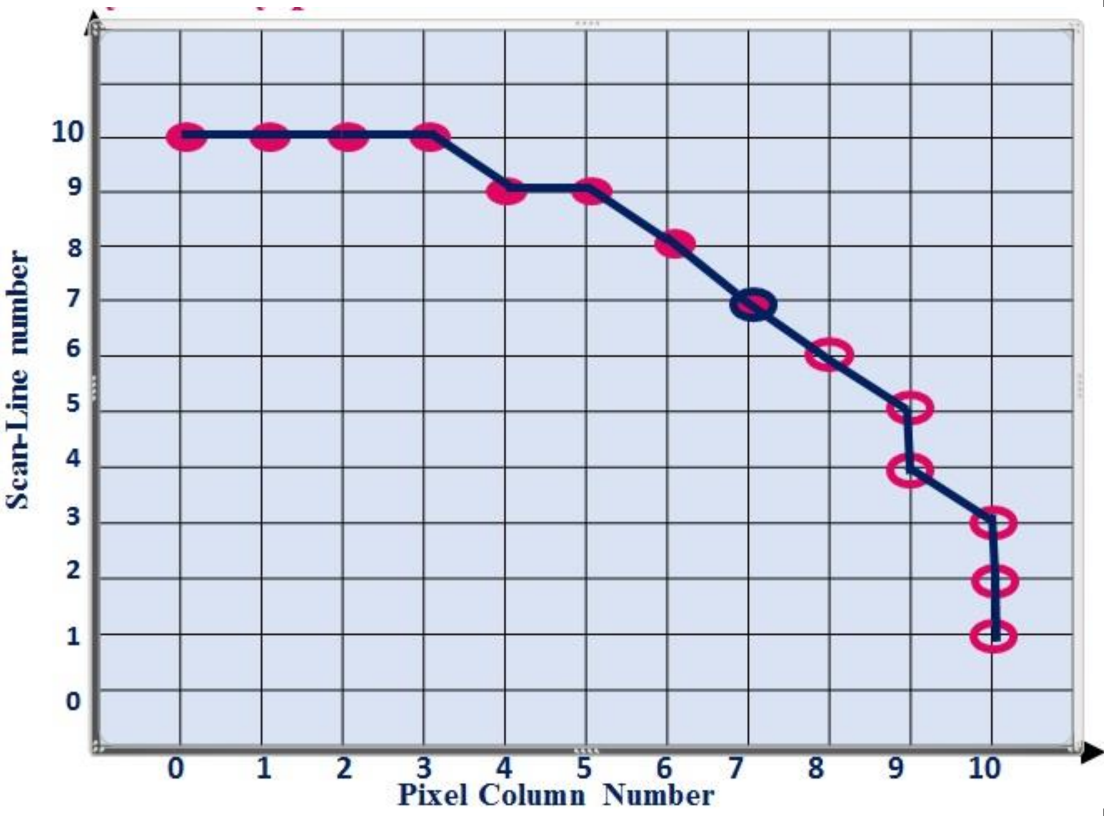
$(x_k, y_k + 1)$

If $P_k > 0$

$P_{k+1} = P_k + 2(y_k + 1) - 2(x_k - 1) + 1$

$(x_k - 1, y_k + 1)$

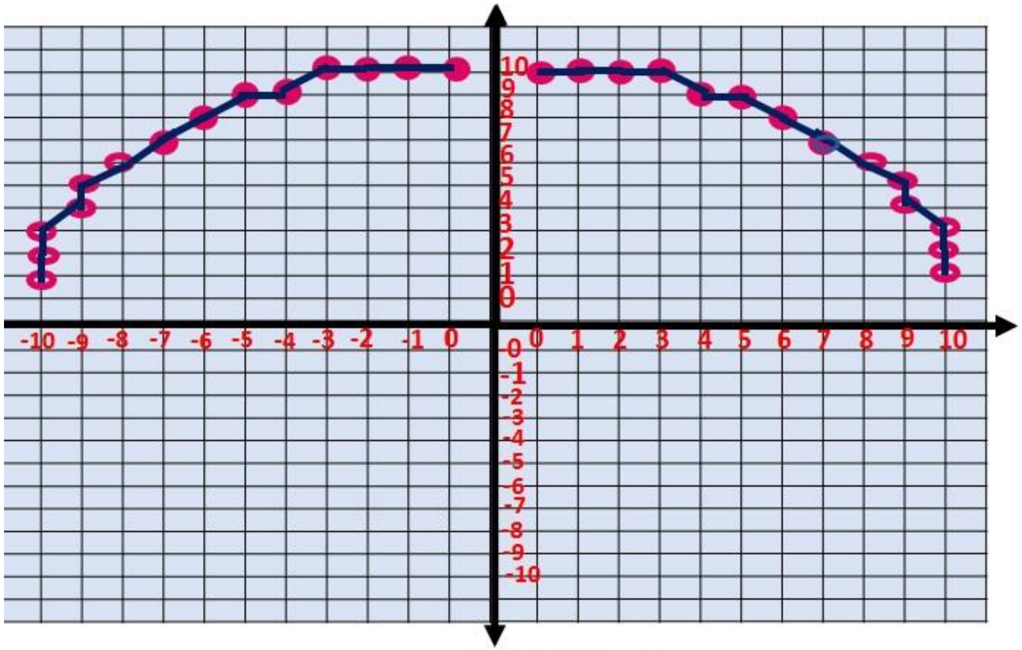
K	X_{k+1} Y_{new}	P_k		Points
0	-9	10	0+1 =1	(10,1)
1	-9+2+1=-6	10	1+1 =2	(10,2)
2	-6+4+1=-1	10	2+1 =3	(10,3)
3	-1+6+1=6	9	3+1 =4	(9,4)
4	6+8+1-18=-3	9	4+1 =5	(9,5)
5	-3+10+1=8	8	5+1 =6	(8,6)
6	8+12+1-16=5	7	6+1 =7	(7,7)



Pleating Points
(10,0)
(10,1)
(10,2)
(10,3)
(9,4)
(9,5)
(8,6)
(7,7)

Pleating Points
(0,10)
(1,10)
(2,10)
(3,10)
(4,9)
(5,9)
(6,8)
(7,7)

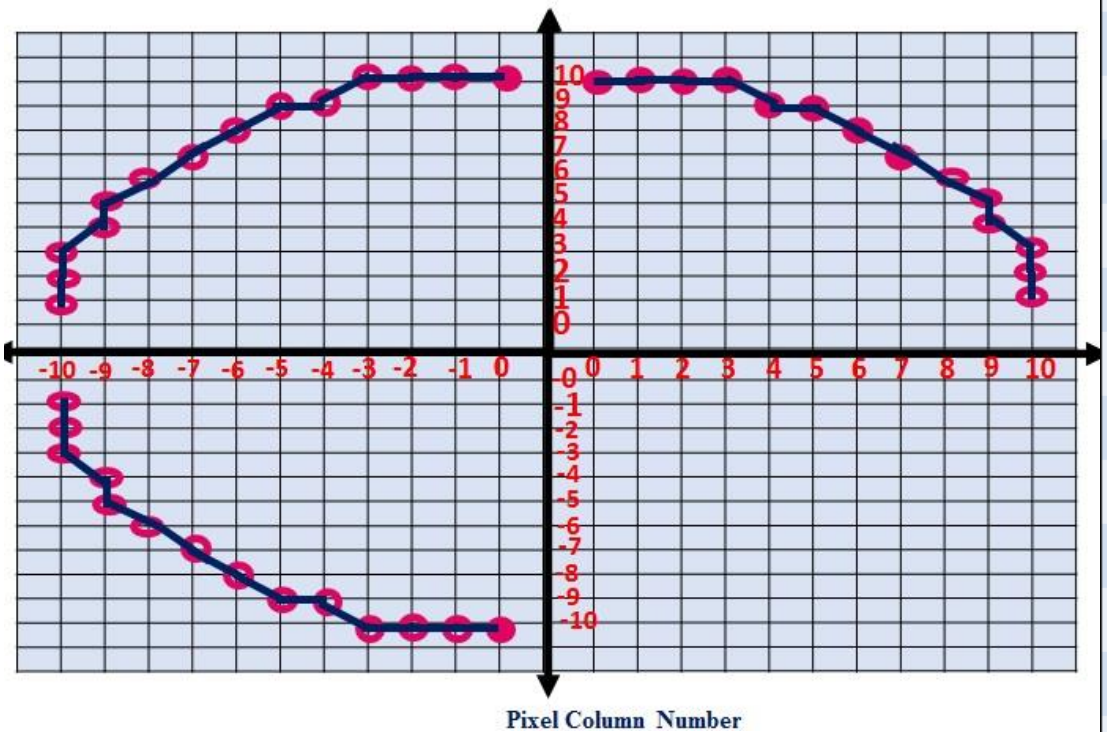
Swap the values of axis coordinates



Pixel Column Number

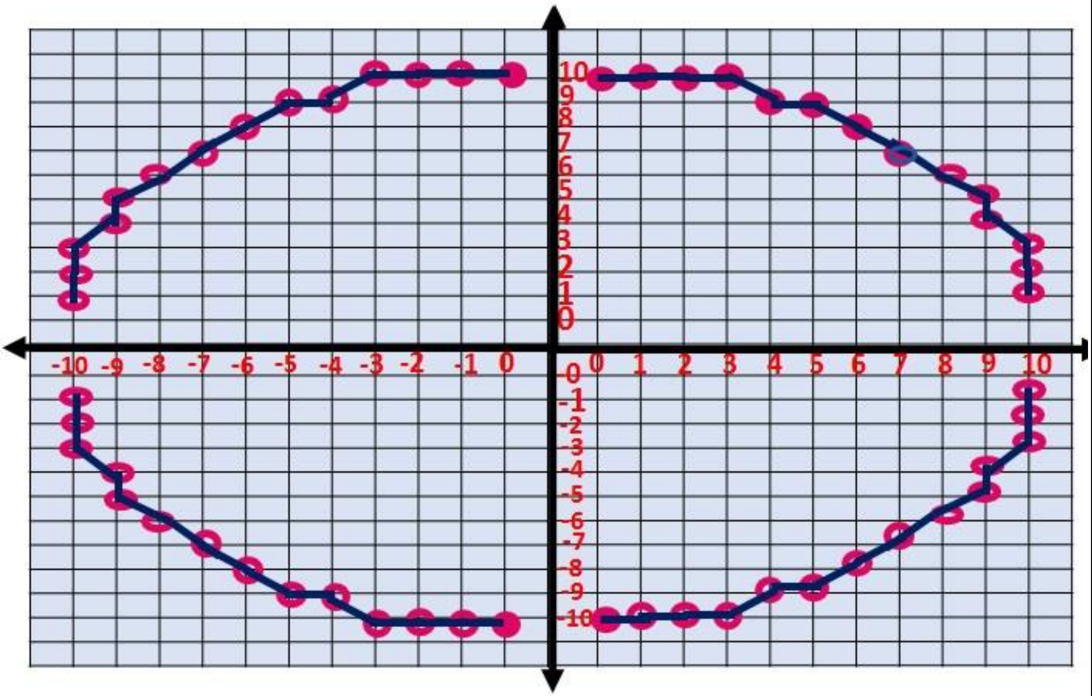
Pleating Points	Pleating Points
(0,10)	(0,10)
(1,10)	(-1,10)
(2,10)	(-2,10)
(3,10)	(-3,10)
(4,9)	(-4,9)
(5,9)	(-5,9)
(6,8)	(-6,8)
(7,7)	(-7,7)
(10,0)	(-10,0)
(10,1)	(-10,1)
(10,2)	(-10,2)
(10,3)	(-10,3)
(9,4)	(-9,4)
(9,5)	(-9,5)
(8,6)	(-8,6)
(7,7)	(-7,7)

multiply values of X coordinate by -1



Multiply values of Y coordinates by -1 & Swap the values of axis coordinates

Pleating Points	Pleating Points
(0,10)	(-10,0)
(-1,10)	(-10,-1)
(-2,10)	(-10,-2)
(-3,10)	(-10,-3)
(-4,9)	(-9,-4)
(-5,9)	(-9, -5)
(-6,8)	(-8,-6)
(-7,7)	(-7,-7)
(-10,0)	(0,-10)
(-10,1)	(-1,-10)
(-10,2)	(-2,-10)
(-10,3)	(-3,-10)
(-9,4)	(-4,-9)
(-9,5)	(-5,-9)
(-8,6)	(-6,-8)
(-7,7)	(-7,-7)



Pleating Points	Pleating Points
(-10,0)	(0,-10)
(-10,-1)	(1,-10)
(-10,-2)	(2,-10)
(-10,-3)	(3,-10)
(-9,-4)	(4,-9)
(-9, -5)	(5,-9)
(-8,-6)	(8,-6)
(-7,-7)	(7,-7)
(0,-10)	(10,0)
(-1,-10)	(10,-1)
(-2,-10)	(10,-2)
(-3,-10)	(10,-3)
(-4,-9)	(9,-4)
(-5,-9)	(9,-5)
(-6,-8)	(8,-6)
(-7,-7)	(7,-7)

Multiply values of Y coordinates by -1 &Swap the values of axis coordinates

- **Step 7:** Move each calculated pixel position(X,Y) onto the circular path centered at (X_c, Y_c) and plot the coordinate values X=X+X_c, Y=Y+Y_c ?
-

Example 2:

- Centre -> (4, 4), Radius -> 2
- Radius (r) = 10, Centre = (3, 4)
- ---

(2,0) (-2,0)

P0=1-r=1-2= -1<0 (2,1) (-2,1

P1=2 >0 (1,2) x (0,2) (0,2)

(1,2) (-1,2)