GALGOTIAS UNIVERSITY
School of Computing Science and Engineering

Program: B.C.A.

Course Code: BCAS3003

Course Name: Computer Graphics

Vision

To be known globally as a premier department of Computer Science and Engineering for value-based education, multidisciplinary research and innovation.

Mission

- **M1:** Developing a strong foundation in fundamentals of computing science with responsiveness towards emerging technologies.
- **M2:** Establishing state-of-the-art facilities and adopt education 4.0 practices to analyze, develop, test and deploy sustainable ethical IT solutions by involving multiple stakeholders.
- **M3:** Establishing Centers of Excellence for multidisciplinary collaborative research in association with industry and academia.

Course Outcomes (COs)

CO Number	Title
CO1	Describe the fundamental concepts of Computer Graphics. (K1)
CO2	To demonstrate with the relevant mathematics of computer graphics, ex. line, circle and ellipse drawing algorithms. (K3)
CO3	To understand the attributes of output primitives of Graphics. (K2).
CO4	Apply simple and composite transformation on graphic objects/elements in two dimensions. (K3).
CO5	Analyze two dimensions modeling and clipping techniques. (K4).
CO6	List out the various contemporary research areas and tool in graphics domain. (K2).

Course Prerequisites

- **Knowledge of Mathematics**
- **Fundamental knowledge of Computer**

Syllabus

Unit 2 – Output Primitives (8 hours)

- Line Drawing Algorithms
- Circle Generation Algorithms
- Ellipse Generating Algorithm
- Pixel Addressing
- Filled-Area Primitives
- Fill Area Function,
- Cell Array, Character Generation

Recommended Books

Text books

- D. Hearn, P. Baker, "Computer Graphics - C Version", 2nd Edition, Pearson Education, 1997

Reference Book

- Heam Donald, Pauline Baker M: "Computer Graphics", PHI 2nd Edn. 1995.
- Harrington S: "Computer Graphics - A Programming Approach", 2nd Edn. Mc GrawHill.
- Shalini Govil-Pai, Principles of Computer Graphics, Springer, 2004

Additional online materials

- Coursera - https://www.coursera.org/learn/fundamentals-of-graphic-design
- https://www.youtube.com/watch?v=fwzYuhduME4&list=PLE4D97E3B8DB8A590
- NPTEL - https://nptel.ac.in/courses/106/106/106106090/
- https://www.coursera.org/learn/research-methods
- https://www.coursera.org/browse/physical-science-and-engineering/research-methods

Ellipse Generation Algorithms

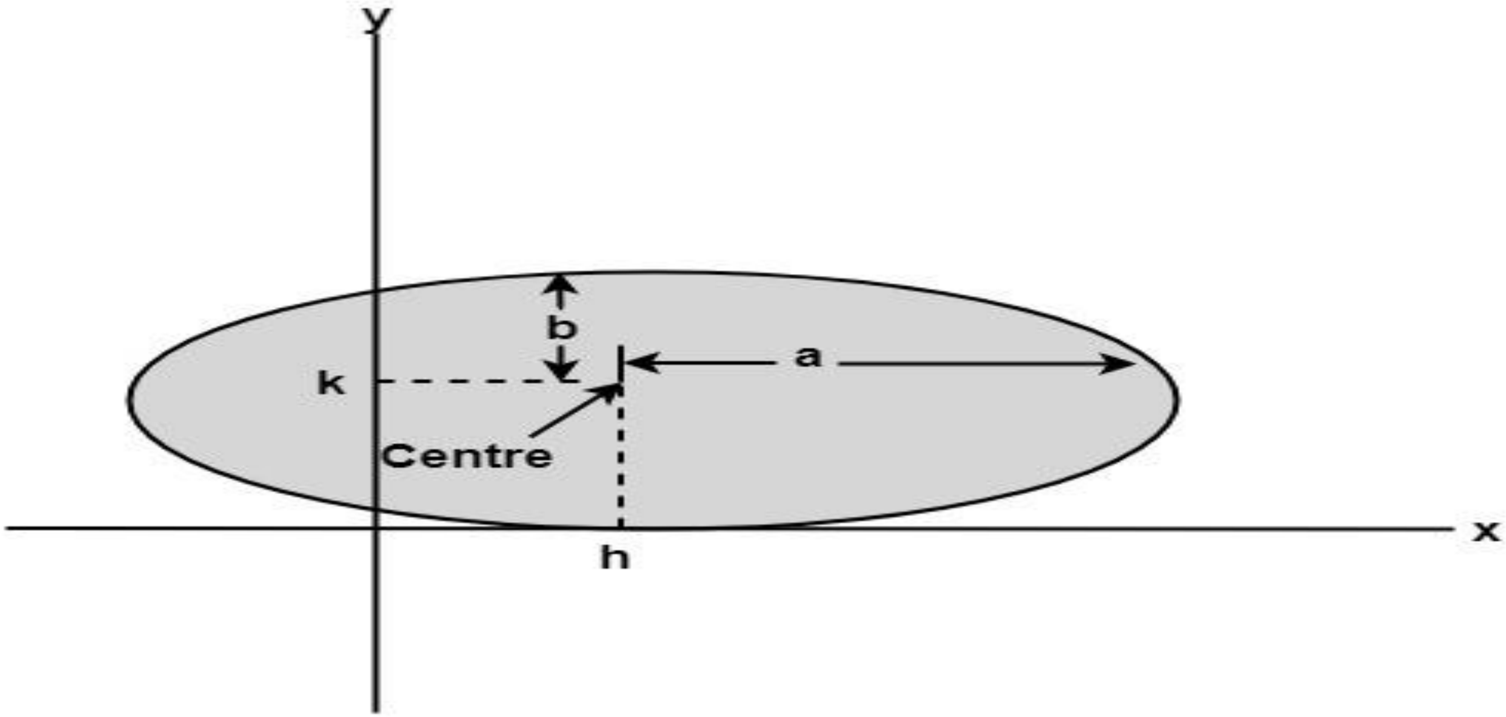
- Scan Converting a Ellipse
- Polynomial Method
- Trigonometric Method
- Midpoint Ellipse Algorithm

Scan Converting a Ellipse

- The ellipse is also a symmetric figure like a circle but is four-way symmetry rather than eight-way.
- Ellipse is defined as the locus of a point in a plane which moves in a plane in such a manner that the ratio of its distance from a fixed point called focus in the same plane to its distance from a fixed straight line called directrix is always constant, which should always be less than unity.
- If the distance to the two foci from any point **P=(x,y)** on the ellipse are labeled $\mathbf{d_1}$ and $\mathbf{d_2}$ then the general equation of the ellipse can be stated as- $\mathbf{d_1+d_2=constant}$.
- **The midpoint ellipse method** is applied throughout the first quadrant in two parts. Now let us take the start position at $\mathbf{(0,r_y)}$ and step along the ellipse path in clockwise order throughout the first quadrant.
- There two methods of defining an Ellipse: Polynomial Method of defining an Ellipse, Trigonometric method of defining an Ellipse

Scan Converting a Ellipse

- The ellipse is also a symmetric figure like a circle but is four-way symmetry rather than eight-way.
- There two methods of defining an Ellipse: Polynomial Method of defining an Ellipse, Trigonometric method of defining an Ellipse



Polynomial Method

- The ellipse has a major and minor axis. If a_1 and $b_1$are major and minor axis respectively. The centre of ellipse is (i, j). The value of x will be incremented from i to a_1 and value of y will be calculated using the following formula

$$y = b_1 \sqrt{1 - \frac{x - i}{a_1^2}} + j$$

Drawback of Polynomial Method

- It requires squaring of values. So floating point calculation is required.
- Routines developed for such calculations are very complex and slow.

Polynomial Method

Algorithm

- Step 1: Set the initial variables: a = length of major axis; b = length of minor axis; (h, k) = coordinates of ellipse center; x = 0; i = step; x_{end} = a.
- Step 2. Test to determine whether the entire ellipse has been scan-converted. If $x > x_{end}$, stop.
- Step 3. Compute the value of the y coordinate:

$$y = b\sqrt{1 - \frac{x^2}{a^2}}$$

- Step 4: Plot the four points, found by symmetry, at the current (x, y) coordinates:

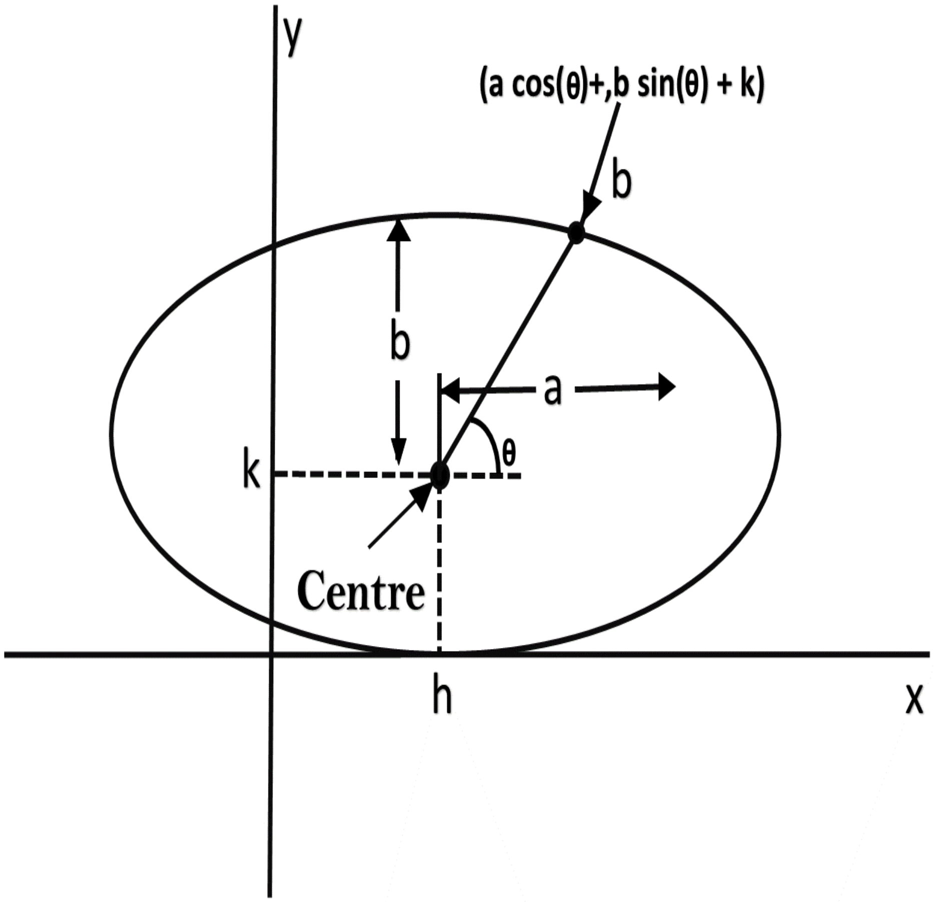
  Plot (x + h, y + k) Plot (-x + h, -y + k)

  Plot (-y - h, x + k) Plot (y + h, -x + k)

- Step 5. Increment x; x = x + i. Step 6. Go to step 2.

Trigonometric Method

- The following equation defines an ellipse trigonometrically as shown in fig:
- x = a * cos (θ) +h and
  y = b * sin (θ)+k
  where (x, y) = the current coordinates
  a = length of major axis
  b = length of minor axis
  θ= current angle
  (h, k) = ellipse center
- In this method, the value of θ is varied from 0 to PI/2 radians. The remaining points are found by symmetry.



Trigonometric Method

Drawback

- This is an inefficient method.
- It is not an interactive method for generating ellipse.
- The table is required to see the trigonometric value.
- Memory is required to store the value of θ.

Algorithm

- **Step1:** Start Algorithm
- **Step2:** Declare variable x_1,y_1,aa_1,bb_1,aa_2,bb_2,fx,fy,p1,a1,b1
- **Step3:** Initialize x_1=0 and y_1=b/* values of starting point of circle */
- **Step4:** Calculate $aa_1=a_1*a_1$
  Calculate $bb_1=b_1* b_1$
  Calculate $aa_2=aa_1*2$
  Calculate $bb_2=bb_1*2$

Trigonometric Method

- **Step5:** Initialize fx = 0
- **Step6:** Initialize fy = aa_2* b_1
- **Step7:** Calculate the value of $p_1$and round if it is integer
  p_1=bb_1-aa_1* b_1+0.25* aa_1/
- **Step 8:**

While (fx < fy) {

Set pixel (x1,y1); Increment x i.e., x = x + 1;

Calculate fx = fx + bb2;

If (p1 < 0)

Calculate p1 = p1 + fx + bb1/

else {

Decrement y i.e., y = y-1

Calculate fy = fy - 992;

p1=p1 + fx + bb1-fy } }

Trigonometric Method

- **Step9:** Setpixel (x_1,y_1)
- **Step10:** Calculate p1=bb_1 (x+.5)(x+.5)+aa(y-1)(y-1)-aa_1*bb_1
- **Step 11:**

```
While (y1>0){
        Decrement y i.e., y = y-1
        fy=fx-aa2/
                if (p1>=0)
                        p1=p1 - fx +  aa1/
                else  {
                        Increment x i.e., x = x + 1
                        fx= fx+bb_2; p1=p1+fx-fy-aa1
                    }       }
Set pixel (x1,y1)
```

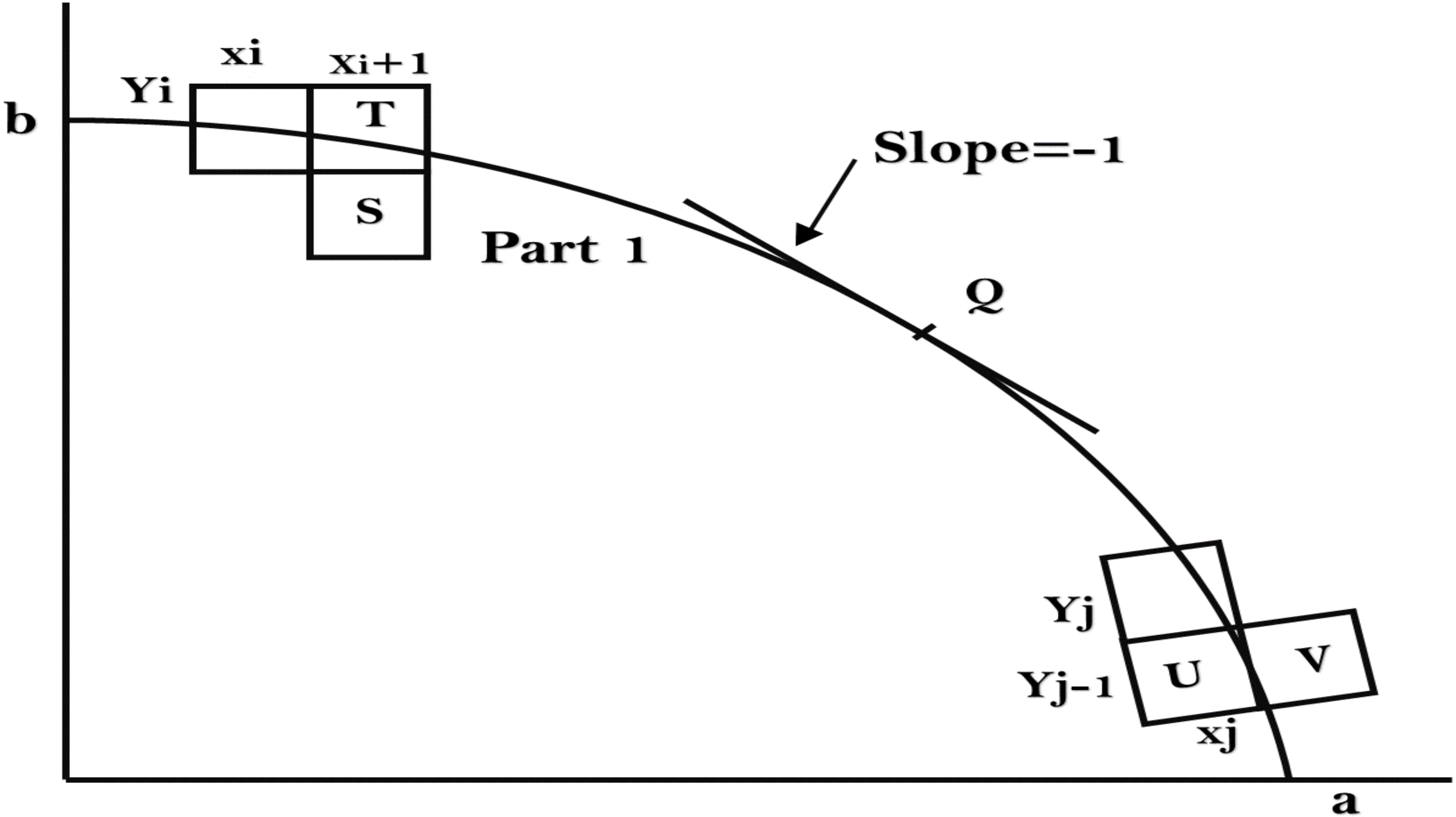
Step 12: Stop

Midpoint Ellipse Algorithm

- This is an incremental method for scan converting an ellipse that is centered at the origin in standard position i.e., with the major and minor axis parallel to coordinate system axis.
- It is very similar to the midpoint circle algorithm. Because of the four-way symmetry property we need to consider the entire elliptical curve in the first quadrant.
- Let's first rewrite the ellipse equation and define the function 'f' that can be used to decide if the midpoint between two candidate pixels is inside or outside the ellipse:

$$f(x, y) = b^2x^2 + a^2y^2 - a^2b^2 = \begin{bmatrix} < 0 (x,y) \text{inside} \\ o\ (x,y) \text{on} \\ > 0\ (x,y) \text{outside} \end{bmatrix}$$

Midpoint Ellipse Algorithm



Midpoint Ellipse Algorithm

- Now divide the elliptical curve from (0, b) to (a, 0) into two parts at point Q where the slope of the curve is -1.
- Slope of the curve is defined by the f(x, y) = 0 is $\frac{dy}{dx} = -\frac{fx}{fy}$

  where fx & fy are partial derivatives of f(x, y) with respect to x & y.
- We have fx = $2b^2$ x, fy=$2a^2$ y & $\frac{dy}{dx} = -\frac{2b^2x}{2a^2y}$
- Hence we can monitor the slope value during the scan conversion process to detect point Q. Our starting point is (0, b)
- Suppose that the coordinates of the last scan converted pixel upon entering step i are (x_i,y_i). We are to select either T $(x_{i+1}),y_i)$ or S (x_{i+1},y_{i-1}) to be the next pixel. The midpoint of T & S is used to define the following decision parameter.

Midpoint Ellipse Algorithm

$pi = f(x_{i+1}), y_i - \frac{1}{2})$

$pi = b^2 (x_{i+1})^2 + a^2 (y_i - \frac{1}{2})^2 - a^2 b^2$

If $pi < 0$, the midpoint is inside the curve and we choose pixel T.

If $pi > 0$, the midpoint is outside or on the curve and we choose pixel S.

Decision parameter for the next step is:

$p_{i+1} = f(x_{i+1} + 1, y_{i+1} - \frac{1}{2})$

$= b^2 (x_{i+1} + 1)^2 + a^2 (y_{i+1} - \frac{1}{2})^2 - a^2 b^2$

Since $x_{i+1} = xi + 1$, we have

$p_{i+1} - pi = b^2[((x_{i+1} + 1)^2 + a^2 (y_{i+1} - \frac{1}{2})^2 - (y_i - \frac{1}{2})^2]$

$p_{i+1} = pi + 2b^2 x_{i+1} + b^2 + a^2 [(y_{i+1} - \frac{1}{2})^2 - (y_i - \frac{1}{2})^2]$

Midpoint Ellipse Algorithm

If T is chosen pixel ($pi<0$), we have $y_{i+1}=yi$.

If S is chosen pixel ($pi>0$) we have $y_{i+1}=yi-1$. Thus we can express

p_{i+1}in terms of pi and (x_{i+1},y_{i+1}): $p_{i+1}= pi+2b^2 x_{i+1}+b^2$ if $pi<0$ $= pi+2b^2 x_{i+1}+b^2-2a^2 y_{i+1}$ if $pi>0$

The initial value for the recursive expression can be obtained by the evaluating the original definition of pi with (0, b):

$p1 = (b^2+a^2 (b-\frac{1}{2})^2-a^2 b^2$

$= b^2-a^2 b+a^2/4$

Suppose the pixel (x_j y_j) has just been scan converted upon entering step j. The next pixel is either U (x_j ,y_j-1) or V (x_j+1,y_j-1).

The midpoint of the horizontal line connecting U & V is used to define the decision parameter:

Midpoint Ellipse Algorithm

Suppose the pixel $(x_j\ y_j)$ has just been scan converted upon entering step j. The next pixel is either U (x_j, y_j-1) or V (x_j+1, y_j-1).

The midpoint of the horizontal line connecting U & V is used to define the decision parameter:

$$q_j=f(x_j+\frac{1}{2}, y_j-1)$$

$$q_j=b^2(x_j+\frac{1}{2})^2+a^2(y_j-1)^2-a^2b^2$$

If $q_j<0$, the midpoint is inside the curve and we choose pixel V.

If $q_j \geq 0$, the midpoint is outside the curve and we choose pixel U.Decision parameter for the next step is:

$$q_{j+1}=f(x_{j+1}+\frac{1}{2}, y_{j+1}-1)$$

$$= b^2(x_{j+1}+\frac{1}{2})^2+a^2(y_{j+1}-1)^2-a^2b^2$$

Midpoint Ellipse Algorithm

Since $y_{j+1}=y_j-1$,we have

$q_{j+1}-q_j=b^2\ [(x_{j+1}+\frac{1}{2})^2-(x_j+\frac{1}{2})^2]+a^2\ (y_{j+1}-1)^2-(y_{j+1})^2]$

$q_{j+1}=q_j+b^2\ [(x_{j+1}+\frac{1}{2})^2-(x_j+\frac{1}{2})^2]-2a^2\ y_{j+1}+a^2$

If V is chosen pixel (qj<0), we have x_{j+1}=xj.

If U is chosen pixel (pi>0) we have x_{j+1}=xj. Thus we can express

q_{j+1}in terms of q_j and (x_{j+1},y_{j+1}):

$q_{j+1}=q_j+2b^2\ x_{j+1}-2a^2\ y_{j+1}+a^2$ if qj < 0

$=q_j-2a^2\ y_{j+1}+a^2$ if qj>0

The initial value for the recursive expression is computed using the original definition of qj. And the coordinates of $(x_k\ y_k)$ of the last pixel choosen for the part 1 of the curve:

$q1 = f(x_k+\frac{1}{2}, y_k-1)=b^2\ (x_k+\frac{1}{2})^2-a^2\ (y_k-1)^2-\ a^2\ b^2$

Midpoint Ellipse Algorithm

Algorithm

- Take input radius along x axis (r_x) and y axis (r_y) and obtain center of ellipse.
- Initially, we assume ellipse to be centered at origin and the first point as $(x, y_0) = (0, r_y)$.
- Obtain the initial decision parameter for region 1 as:
  - $p1_0 = r_y^2 + 1/4 r_x^2 - r_x^2 r_y$
- For every x_k position in region 1:
  - If $p1_k < 0$ then the next point along the is (x_{k+1}, y_k) and $p1_{k+1} = p1_k + 2r_y^2 x_{k+1} + r_y^2$
  - Else, the next point is (x_{k+1}, y_{k-1}) and $p1_{k+1} = p1_k + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_y^2$

Midpoint Ellipse Algorithm

Algorithm

- Obtain the initial value in region 2 using the last point (x_0, y_0) of region 1 as
  - $p2_0=r_y^2(x_0+1/2)^2+r_x^2 (y_0-1)^2-r_x^2r_y^2$
- At each y_k in region 2 starting at k =0 perform the following task.
  - If $p2_k<0$ the next point is (x_k, y_{k+1}) and $p2_{k+1}=p2_k-2r_x^2y_{k+1}+r_x^2$
  - Else, the next point is (x_{k+1},y_{k-1}) and $p2_{k+1}=p2_k+2r_y^2x_{k+1} -2r_x^2y_{k+1}+r_x^2$
- Now obtain the symmetric points in the three quadrants and plot the coordinate value as: x=x+xc, y=y+yc
- Repeat the steps for region 1 until $2r_y^2x$&gt=$2r_x^2y$

Midpoint Ellipse Algorithm

Algorithm

Algorithm: -

1. Read radii r_x and r_y .

2. Initialize starting point as

$x = 0 \quad y = r_y$

3. Calculate the initial value or decision parameter in region 1 as

$d_1 = r^2y - r^2_x r^2y + \frac{1}{4} r^2_x$

4. Initialize dx and dy as

$dx = 2r^2_y x$

$dy = 2r^2_x y$

5. do

$\{$ plot(x , y)

if($d_1 < 0$)

$\{x = x + 1$

$y = y$

$dx = dx + 2r^2y$

$d_1 = d_1 + dx + r^2y$

$[d_1 = d_1 + 2r^2_y x + 2r^2y + r^2y]$

$\}$

else

$\{ x = x + 1$

$y = y - 1$

$dx = dx + 2r^2y$

$dy = dy - 2r^2x$

$d_1 = d_1 + dx - dy + r^2y$

$[d_1 = d_1 + 2r^2_y x + 2r^2y - (2r^2_x y - 2r^2x) + r^2y]$

$\}$ while ($dx < dy$)

Midpoint Ellipse Algorithm

Algorithm

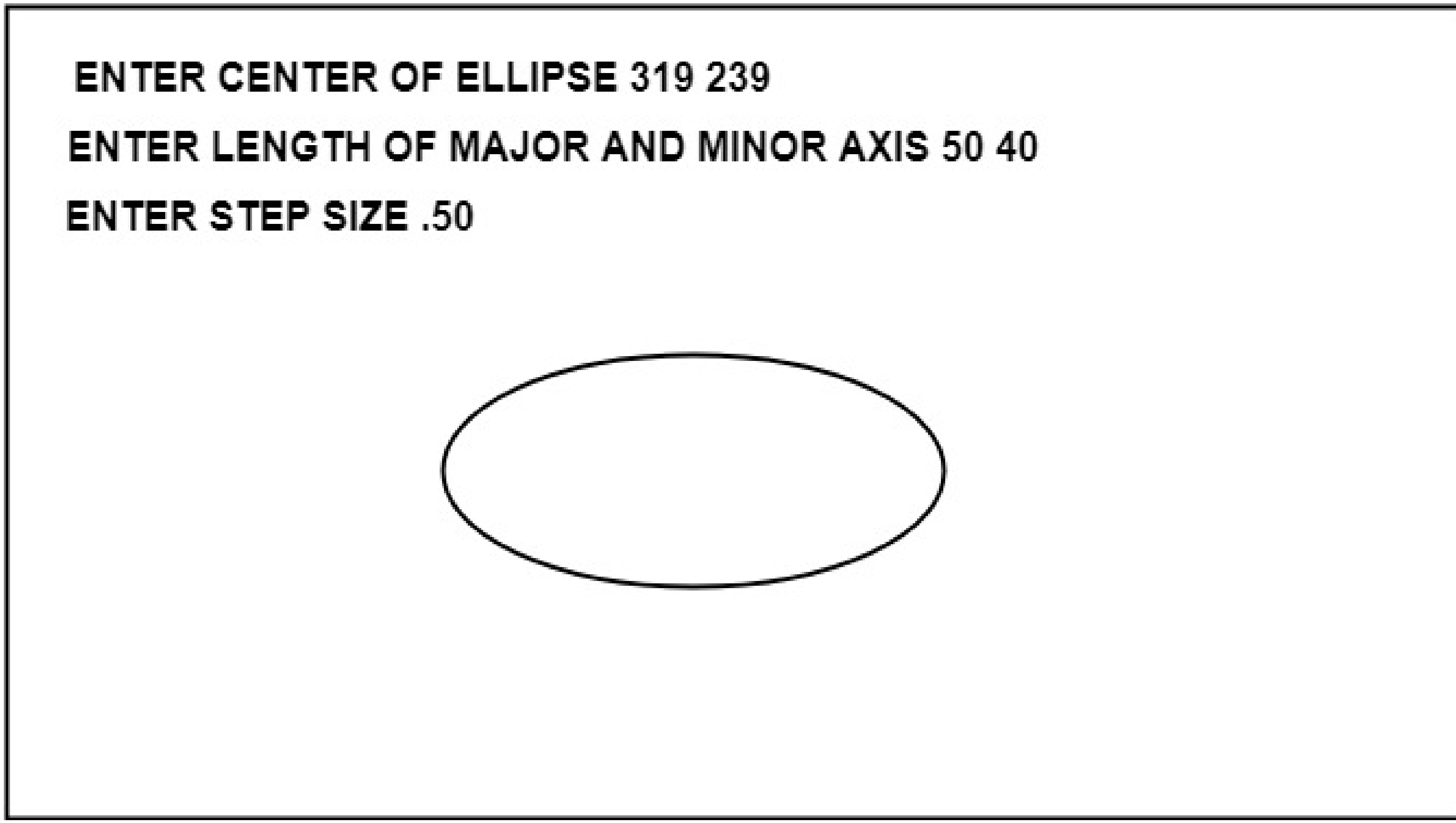
6. Calculate the initial value or decision parameter in region 2 as

$$d_2 = r^2y\left(x + \frac{1}{2}\right)^2 + r^2{}_x(y - 1)^2 - r^2{}_x\, r^2y$$

7. do
```
{   Plot(x, y)
    if( d2 > 0 )
    { x = x
       y = y – 1
     dy = dy - 2r²x
     d2 = d2 - dy + r²x
     [d2 = d2 – (2r²x y - 2r²x ) + r²x]
    }
    else
   { x = x + 1
      y = y – 1
    dy = dy - 2r²x
    dx = dx +2r²y
     d2 = d2 + dx - dy + r²x
    [d2 = d2 + 2r²y x + 2r²y – (2r²x y  - 2r²x ) + r²y ]
    } while(y >0)
```

8. Determine symmetrical points in other three quadrants.
9. Stop.

Midpoint Ellipse Algorithm Example



Questions

- Explain midpoint Ellipse algorithm.
- Write a short note on Midpoint Ellipse Algorithm.

Thank You