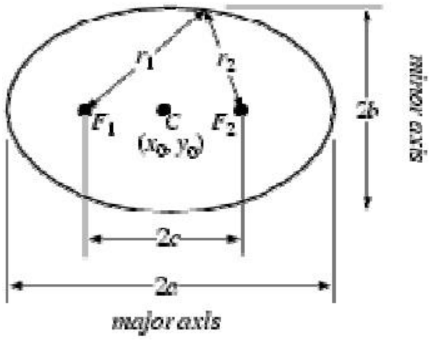
ELLIPSE, PARABOLA, HYPERBOLA

Ellipse

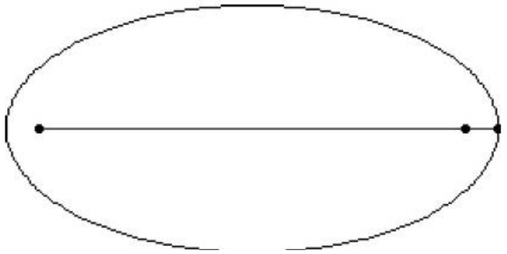
An ellipse is a curve that is the locus of all points in the plane the sum of whose distances r1 and r2 from two fixed points F1 and F2, (the foci) separated by a distance of 2c is a given positive constant 2a. This results in the two-center bipolar coordinate equation:

r1 + r2 = 2a



Where *a* is the semi-major axis and the origin of the coordinate system is at one of the foci. The corresponding parameter *b* is known as the semi-minor axis.

The ellipse was first studied by Menaechmus, investigated by Euclid, and named by Apollonius. The focus and conic section directrix of an ellipse were considered by Pappus. In 1602, Kepler believed that the orbit of Mars was oval; he later discovered that it was an ellipse with the Sun at one focus. In fact, Kepler introduced the word "focus" and published his discovery in 1609.



In 1705 Halley showed that the comet now named after him moved in an elliptical orbit around the Sun (MacTutor Archive). An ellipse rotated about its minor axis gives an oblate spheroid, while an ellipse rotated about its major axis gives a prolate spheroid. Let an ellipse lie along the *x*-axis and find the equation of the figure given above where F1 and F2 are at (-c, 0) and (c, 0). In Cartesian coordinates,

$$\sqrt{(x+c)^2+y^2}+\sqrt{(x-c)^2+y^2}=2a.$$

Bring the second term to the right side and square both sides,

$$(x+c)^2+y^2=4a^2-4a\sqrt{(x-c)^2+y^2}+(x-c)^2+y^2.$$

Now solve for the square root term and simplify

$$\begin{aligned}\sqrt{(x-c)^2+y^2} &= -\frac{1}{4a}(x^2+2xc+c^2+y^2-4a^2-x^2+2xc-c^2-y^2)\\ &= -\frac{1}{4a}(4xc-4a^2)=a-\frac{c}{a}x.\end{aligned}$$

Square one final time to clear the remaining square root,

$$x^2 - 2xc + c^2 + y^2 = a^2 - 2cx + \frac{c^2}{a^2}x^2.$$

Grouping the x terms then gives

$$x^2\frac{a^2 - c^2}{a^2} + y^2 = a^2 - c^2,$$

this can be written in the simple form

$$\frac{x^2}{a^2} + \frac{y^2}{a^2 - c^2} = 1.$$

Defining a new constant

$$b^2 \equiv a^2 - c^2$$

puts the equation in the particularly simple form

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$$

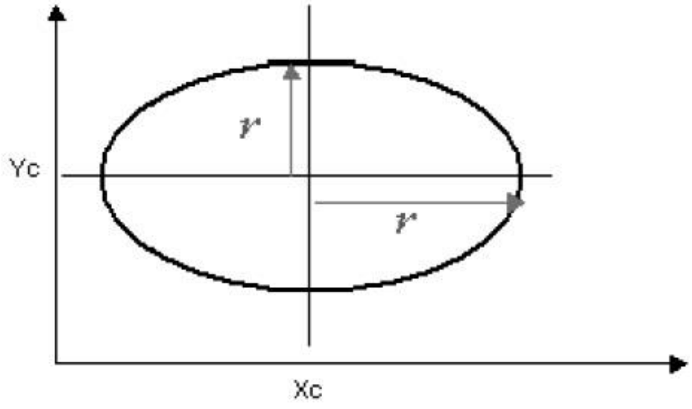
The parameter b is called the semi-minor axis by analogy with the parameter a, which is called the semi-major axis (assuming b < a). The fact that b as defined at right is actually the semi-minor axis is easily shown by letting r1 and r2 be equal. Then two right triangles are produced, each with hypotenuse a, base c, and height b = a^2 - c^2. Since the largest distance along the minor axis will be achieved at this point, b is indeed the semi-minor axis.
If, instead of being centered at (0, 0), the center of the ellipse is at (x0, y0), at right equation becomes:

Ellipse Drawing Techniques

Now we already understand circle drawing techniques. One way to draw ellipse is to use the following equation:

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1.$$

where x_0 may be replaced by x_c in case of center other than origin and same in case of y.



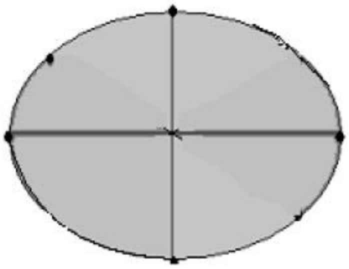
Another way is to use polar coordinates r and θ, for that we have parametric equations:

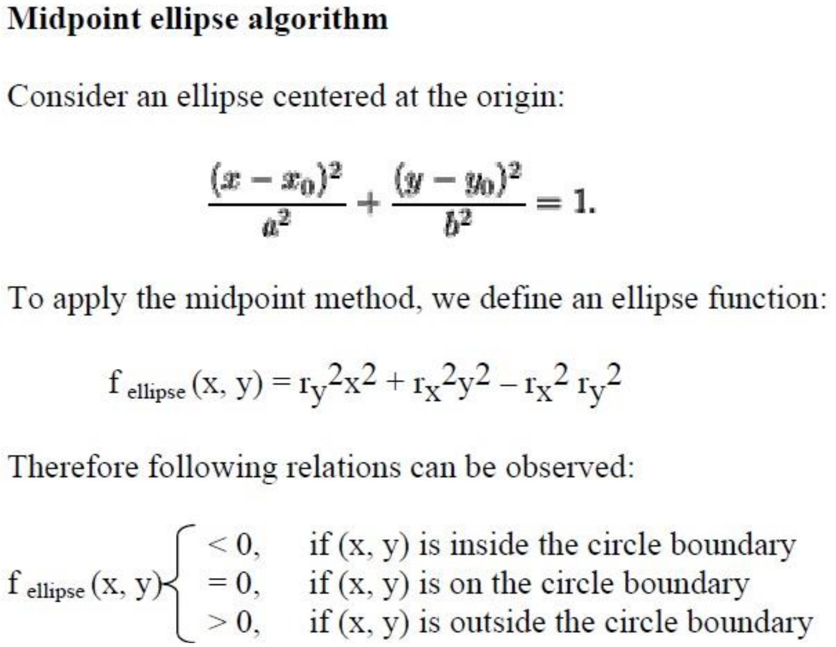
$x = x_c + rx \cos \theta$

$y = y_c + ry \sin \theta$

Four-way symmetry

Symmetric considerations can be had to further reduce computations. An ellipse in standard position is symmetric between quadrants, but unlike a circle, it is not symmetric between the two octants of a quadrant. Thus, we must calculate pixel positions along the elliptical arc throughout one quadrant, and then we obtain positions in the remaining three quadrants by symmetry as shown in at right figure.





Midpoint ellipse algorithm

Consider an ellipse centered at the origin:

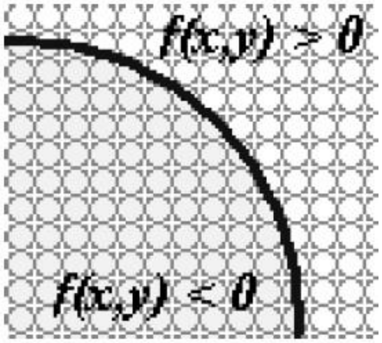
$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1.$$

To apply the midpoint method, we define an ellipse function:

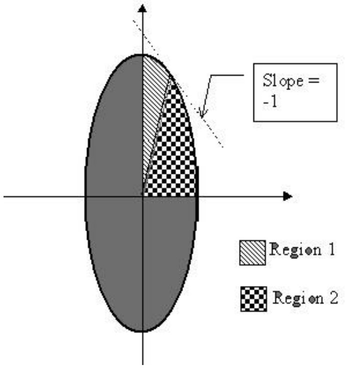
$$f_{\text{ellipse}}(x, y) = r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2$$

Therefore following relations can be observed:

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



Now as you have some idea that ellipse is different from circle. Therefore, a similar approach that is applied in circle can be applied here using some different sampling direction as shown in the figure at right. There are two regions separated in one octant.



Therefore, idea is that in region 1 sampling will be at x direction; whereas y coordinate will be related to decision parameter. In region 2 sampling will be at y direction; whereas x coordinate will be related to decision parameter.

So consider first region 1. We will start at (0, r y); we take unit steps in the x direction until we reach the boundary between region 1 and region 2. Then we switch to unit steps in the y direction over the remainder of the curve in the first quadrant. At each step, we need to test the value of the slope of the curve. The ellipse slope is calculated from following equation:

$$dy / dx = -2\, r_y{}^2 x^2 / 2\, r_x{}^2 y^2$$

At the boundary region 1 and region 2, dy/ dx = -1 and

$$2\, r_x{}^2 y^2 = 2\, r_y{}^2 x^2$$

Therefore, we move out of region 1 whenever

$$2\, r_y{}^2 x^2 >= 2\, r_x{}^2 y^2$$

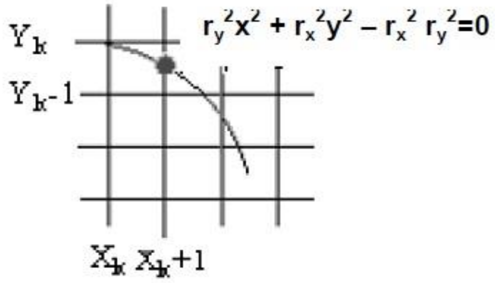


Figure at right shows the midpoint between the two candidate pixels at sampling position $x_k + 1$ in the first region. Assuming position (x_k, y_k) has been selected at the previous step; we determine the next position along the ellipse path by evaluating the decision parameter at this midpoint:

$$P1k = f_{ellipse}\,(x_k + 1, y_k - ½)$$

$$f_{ellipse}\,(x_k +1, y_k - ½) = r^2{}_y\,(x_k + 1)^2 + r_x{}^2\,(y_k - ½)^2 - r_x{}^2\, r_y{}^2 \text{ -------(1)}$$

If $p_k < 0$, this midpoint is inside the ellipse and the pixel on scan line y_k is closer to the ellipse boundary. Otherwise, the mid position is outside or on the ellipse boundary, and we select the pixel on scan-line y_k-1.

Successive decision parameters are obtained using incremental calculations. We obtain a recursive expression for the next decision parameter by evaluating the ellipse function at sampling position x_k+1=x_k+2:

$$f_{ellipse}\,(x_k+1 +1, y_{k+1} - ½) = r_y{}^2\, [(x_k + 1) + 1]^{\,2} + r_x{}^2\,(y_{k+1} - ½)^{\,2} - r_x{}^2\, r_y{}^2 \text{ ---(2)}$$

Subtracting (1) from (2), and by simplification, we get

$$P_{k+1}1 = P_k1 + 2\, r_y{}^2\,(x_k + 1) + r_x{}^2\,(y_{k+1}{}^2 - y_k{}^2) - r_x{}^2\,(y_{k+1} - y_k) + r_y{}^2$$

Where y_{k+1} is either y_k or y_k-1, depending on the sign of P_k. Therefore, if $P_k < 0$ or negative then y_{k+1} will be y_k and the formula to calculate P_{k+1} will be:

$$P1_{k+1} = P_k1 + 2\, r_y^2\, (x_k + 1) + r_x^2\, (y\, k^2 - y_k^2) - r_x^2\, (y_k - y_k) + r_y^2$$

$$P_{k+1}1 = P_k1 + 2\, r_y^2\, (x_k + 1) + r_y^2$$

Otherwise, if $P_k > 0$ or positive then y_{k+1} will be y_k-1and the formula to calculate P_{k+1} will be:

$$P_{k+1}1 = P_k1 \ + 2\, r_y^2\, (x_k + 1) + r^2{}_x\, ((y_k - 1)^2 - y_k^2) - r_x^2\, (y_k - 1 - y_k) + r_y^2$$

$$P1_{k+1} = P1_k + 2\, r_y^2 (x_k + 1) + r_x^2 \ (- 2\, y_k + 1) - r_x^2 (- 1) + r_y^2$$

$$P1_{k+1} = P1_k + 2\, r_y^2 (x_k + 1) - 2\, r_x^2\, y_k + r^2{}_x + r_x^2 + r_y^2$$

$$P1_{k+1} = P1_k + 2\, r_y^2 (x_k + 1) - 2\, r_x^2 (y_k - 1) + r_y^2$$

Now a similar case that we observe in line algorithm is from where starting P_k will evaluate. For this at the start pixel position will by (0, r_y). Therefore, putting this value is equation , we get

$$P1_0 = r_y^2\, (0 + 1)^2 + r_x^2 (r_y - ½)^2 - r_x^2\, r_y^2$$

$$P1_0 = r^2{}_y \ + r_x^2\, r_y^2 - r^2{}_x\, r_y + ¼\, r_x^2 - r_x^2\, r_y^2$$

$$P1_0 = r^2{}_y \ - r_x^2\, r_y + ¼\, r_x^2$$

Similarly same procedure will be adapted for region 2 and decision parameter will be calculated, here we are giving decision parameter and there derivation is left as an exercise for the students.

$$P_{k+1}2 = P_k2 - 2\, r_x^2\, (y_k + 1) + r_x^2 \qquad ,$$

if $p_k2 > 0$

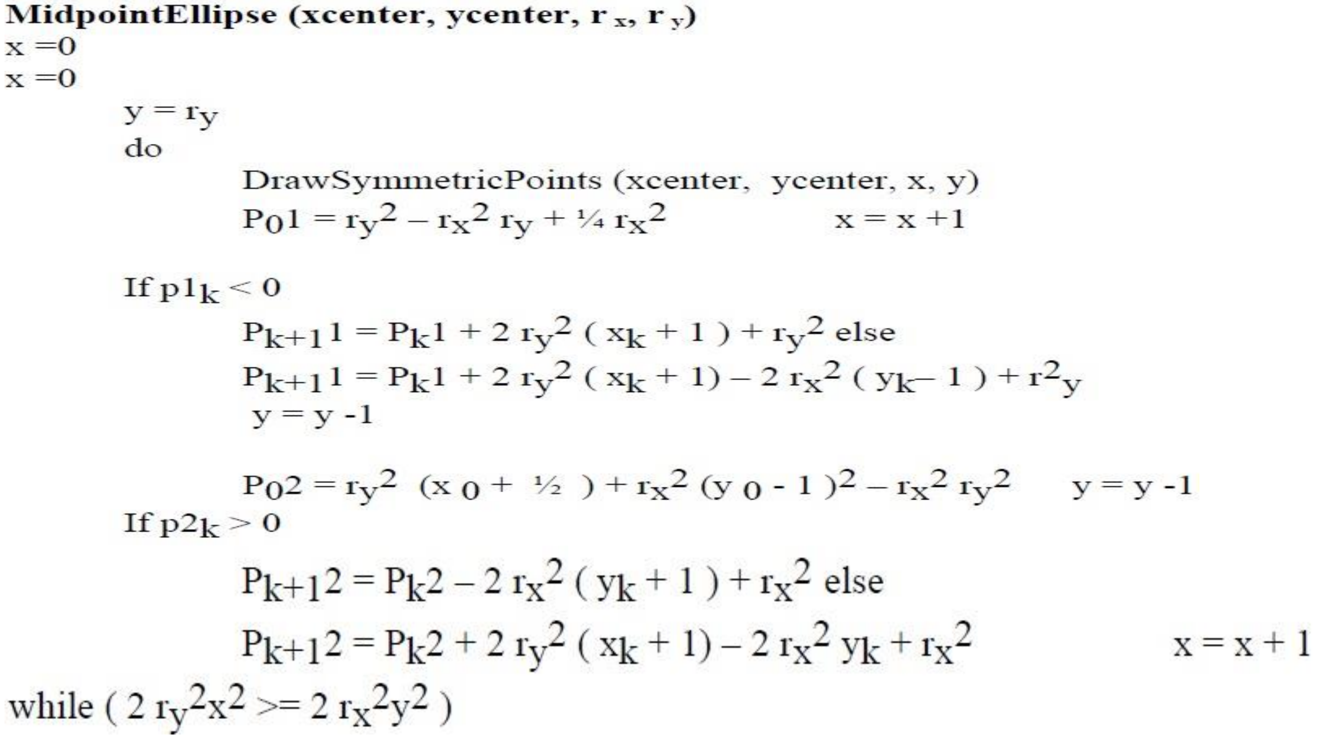
$$P_{k+1}{}^2 = P_k{}^2 + 2\, r_y^2\, (x_k + 1) - 2\, r_x^2\, y_k + r_x^2$$

otherwise

The initial parameter for region 2 will be calculated by following formula using the last point calculated in region 1 as:

$$P_02 = r_y^2 \ \ (x_0 + \ ½ \ ) + r_x^2\, (y_0 - 1)^2 - r_x^2 r_y^2$$

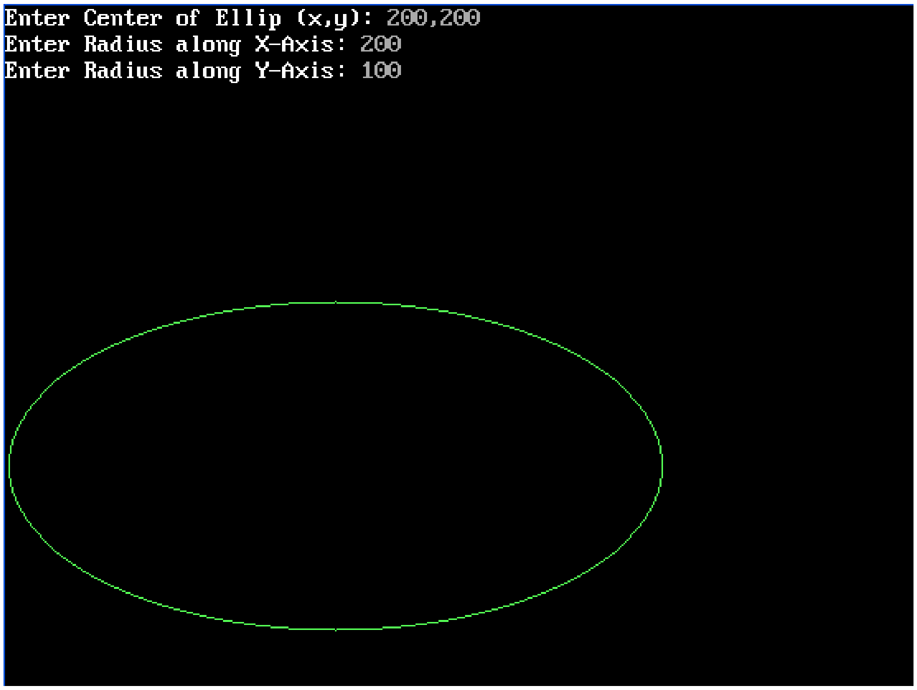
Since all increments are integer. Finally sum up all in the algorithm:



```
MidpointEllipse (xcenter, ycenter, $r_x$, $r_y$)
x =0
x =0
        $y = r_y$
        do
                DrawSymmetricPoints (xcenter,  ycenter, x, y)
                $P_0 1 = r_y^2 - r_x^2 r_y + ¼ r_x^2$                x = x +1

        If $p1_k < 0$
                $P_{k+1} 1 = P_k 1 + 2 r_y^2 ( x_k + 1 ) + r_y^2$ else
                $P_{k+1} 1 = P_k 1 + 2 r_y^2 ( x_k + 1) - 2 r_x^2 ( y_k - 1 ) + r^2_y$
                 y = y -1

                $P_0 2 = r_y^2 \ (x_0 + ½ ) + r_x^2 (y_0 - 1 )^2 - r_x^2 r_y^2$      y = y -1
        If $p2_k > 0$
                $P_{k+1} 2 = P_k 2 - 2 r_x^2 ( y_k + 1 ) + r_x^2$ else
                $P_{k+1} 2 = P_k 2 + 2 r_y^2 ( x_k + 1) - 2 r_x^2 y_k + r_x^2$                 x = x + 1
while ( $2 r_y^2 x^2 >= 2 r_x^2 y^2$ )
```

THIS PROGRAM DISPLAY THE ELLIPSE

```
#include <stdio.h>      #include <conio.h>
#include<iostream.h> #include <graphics.h>
#include <stdlib.h>     #include<math.h>
#define ROUND(x)  ((int)(x+0.5))
float xc,yc,rx,ry;
void drawpoint(int x, int y){
 putpixel(xc+x,yc+y,10);
  putpixel(xc-x,yc+y,10);
 putpixel(xc+x,yc-y,10);
 putpixel(xc-x,yc-y,10);}
void main(void){
        float x,y;
        int gdriver=DETECT, gmode, ecode;
        char comma;
        initgraph(&gdriver, &gmode, "../bgi");
/* read result of initialization */
        ecode = graphresult();
        if (ecode != grOk){
                printf("Graphics error:\n");
                printf("Press any key...");
                getch();
                exit(1);
        }
float p;
        cout<<"Enter Center of Ellip (x,y): ";
        cin>>xc>>comma>>yc;
        cout<<"Enter Radius along X-Axis: ";
        cin>>rx;
        cout<<"Enter Radius along Y-Axis: ";
        cin>>ry;
        yc = getmaxy()-yc;
        x=0;
        y=ry;
        drawpoint(x,y);
        p=ROUND(ry*ry-rx*rx*ry+0.25*(rx*rx)); //ry2-rx2*ry+(.25*rx2)
        while((ry*ry)*x<(rx*rx)*y)
        {       x=x+1;
                if(p>=0)
                {       y=y-1;
                        p=p+2*(ry*ry)*x+ry*ry-2*(rx*rx)*y;
                }
                else
                {       p=p+2*(ry*ry)*x+ry*ry;
                }
                drawpoint(x,y);
        }
        p=ROUND(ry*ry*((x+.5)*(x+.5))+rx*rx*((y-1)*(y-1))-(rx*rx)*(ry*ry));
        while(y>0)
        {       y=y-1;
                if(p<=0)
                {       x=x+1;
                        p=p-2*(rx*rx)*y+rx*rx+2*(ry*ry)*x;
                }
                else
                {
                        p=p-2*(rx*rx)*y+rx*rx;
                }
                drawpoint(x,y);
        }
        getch();
        closegraph();
}
```



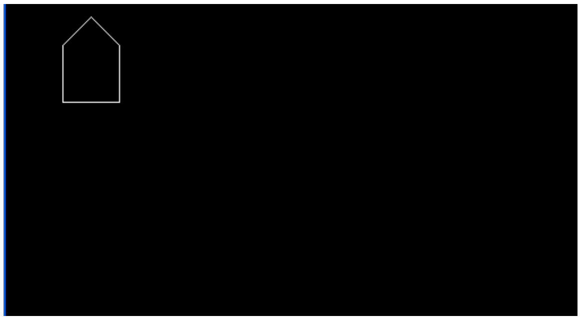
```
Enter Center of Ellip (x,y): 200,200
Enter Radius along X-Axis: 200
Enter Radius along Y-Axis: 100
```

THIS PROGRAM PERFORM DIFFERENT FUNCTION AT OBJECT (SCALING, SCALING BACK, ROTATE , ROTATE BACK, TRANSLATION)

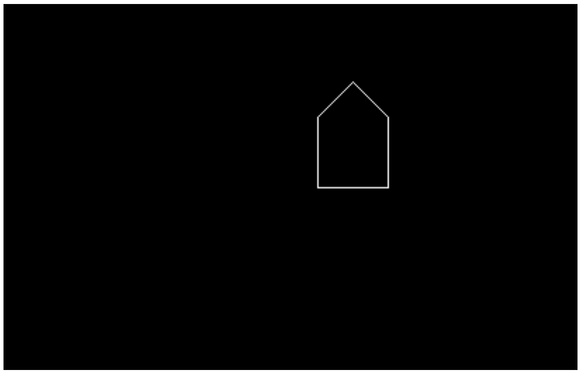
```
#include<iostream.h> #include <graphics.h>
#include <stdlib.h>     #include <stdio.h>
#include <conio.h>

int arr[10]={50,50, 50,100, 100,100, 100,50, 75,25};
int i;
        void draw(){
                for(i=0;i<=6;i+=2)
                        line(arr[i],arr[i+1],arr[i+2],arr[i+3]);
                        line(arr[i],arr[i+1],arr[0],arr[1]);  }
        void translate(int tx, int ty){
                for(i=0;i<=8;i+=2)
                        {arr[i]=arr[i]+tx;
                         arr[i+1]=arr[i+1]+ty;}
                  }
        void scaling_inc(int sx, int sy){
                for(i=0;i<=8;i+=2){
                                arr[i]=arr[i]*sx;
                                arr[i+1]=arr[i+1]*sy;}
                }
        void scaling_dec(int sx, int sy){
                for(i=0;i<=8;i+=2){
                                arr[i]=arr[i]/sx;
                                arr[i+1]=arr[i+1]/sy;}}
 void rotate(float cos, float sin){
                        for(i=0;i<=8;i+=2){
                                arr[i]=(arr[i]*cos)-(arr[i+1]*sin);
```

```
                                            arr[i+1]=(arr[i+1]*cos)+(arr[i]*sin);}
                        }
int main(void){
   int gdriver = DETECT, gmode, errorcode;
   int xmax, ymax;
   initgraph(&gdriver, &gmode, "");
   errorcode = graphresult();
if (errorcode != grOk){
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1);}
   char c;
   draw();
   do{    c=getch();
                         switch(c){
                         case '5':
                         rotate(0.9396,0.3420);
                         break;
                         case '1':
                         scaling_inc(2,2);
                         break;
                         case '3':
                         scaling_dec(2,2);
                         break;
                         case '2':
                         translate(0,10);
                         break;
                         case '6':
                         translate(10,0);
                         break;
                         case '4':
                         translate(-10,0);
                         break;
                         case '8':
                         translate(0,-10);
                         break;
                        }
             cleardevice();
             draw();
       }
       while(c!='q');
}
```

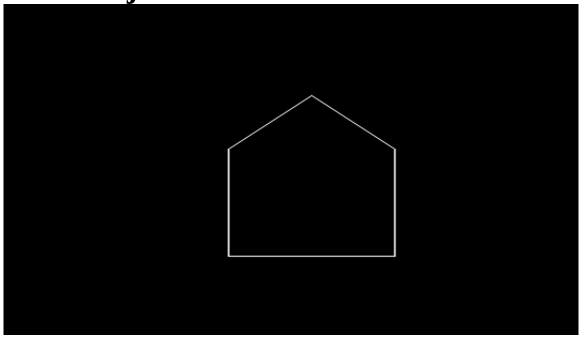
First object was at start



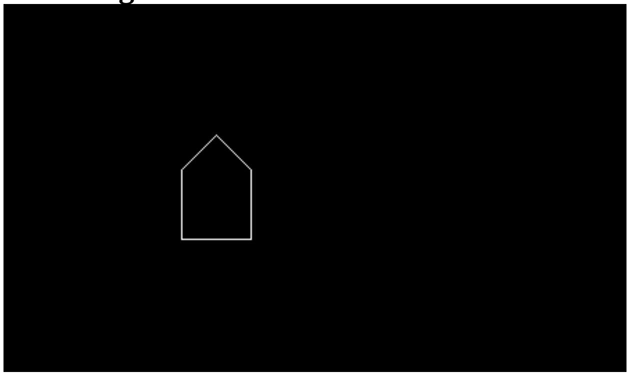
Then transformation



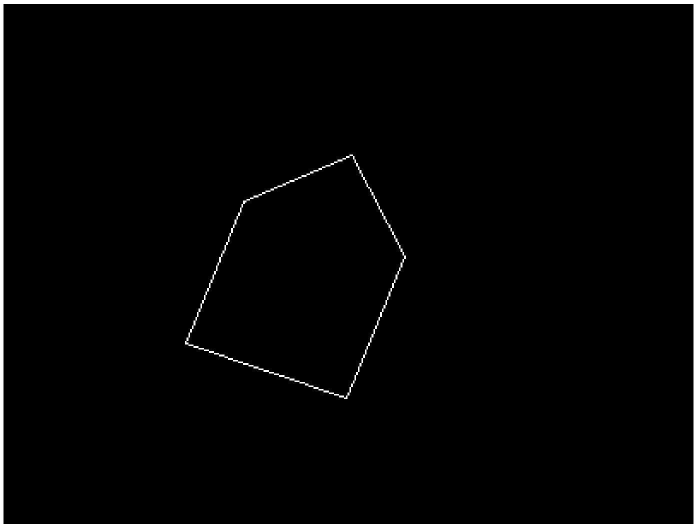
Then here perform scaling on the object



Here perform the operation scaling back

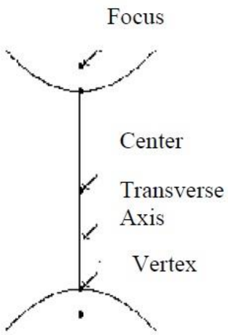


Perform clock wise rotation on the object



Hyperbola

We begin this section with the definition of a hyperbola. A **hyperbola** is the set of all points (x, y) in the plane the difference of whose distances from two fixed points is some constant. The two fixed points are called the **foci**.



Each hyperbola consists of two **branches**. The line segment; which connects the two foci intersects the hyperbola at two points, called the **vertices**. The line segment; which ends at these vertices is called the **transverse axis** and the midpoint of this line is called the **center** of the hyperbola. See figure at right for a sketch of a hyperbola with these pieces identified.

Note that, as in the case of the ellipse, a hyperbola can have a vertical or horizontal orientation. We now turn our attention to the standard equation of a hyperbola. We say that the standard equation of a hyperbola centered at the origin is given by

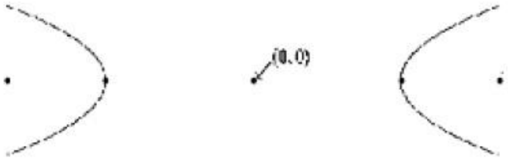
$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$$

if the transverse axis is horizontal, or

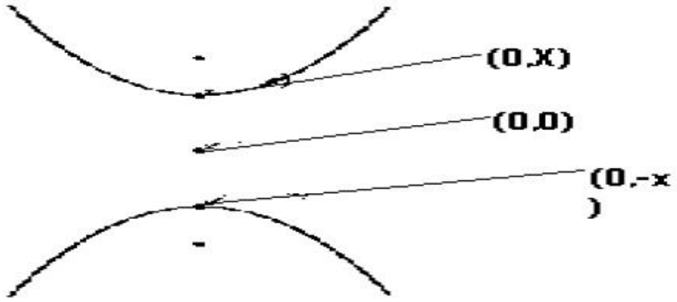
$$\frac{y^2}{a^2} - \frac{x^2}{b^2} = 1$$

if the transverse axis is vertical.

Notice a very important difference in the notation of the equation of a hyperbola compared to that of the ellipse. We see that a always corresponds to the positive term in the equation of the ellipse. The relationship of a and b does not determine the orientation of the hyperbola. (Recall that the size of a and b was used in the section on the ellipse to determine the orientation of the ellipse.) In the case of the hyperbola, the variable in the ``positive" term of the equation determines the orientation of the hyperbola. Hence, if the variable x is in the positive term of the equation, as it is in the equation



$$x^2/a^{2-}y^2/b^2=1,$$



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

then the hyperbola is oriented as follows:

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

then the hyperbola is oriented as follows:

If the variable **y** is in the positive term of the equation, as it is in the equation

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

then we see the following type of hyperbola:

Note that the vertices are always units from the center of the hyperbola, and the distance c of the foci from the center of the hyperbola can be determined using a, b, and the following equality:

$$\mathbf{b^2= c^2 - a^2}$$

We will use this relationship often, so keep it in mind.

The next question you might ask is this: ``what happens to the equation if the center of the hyperbola is not (0, 0)?" As in the case of the ellipse, if the center of the hyperbola is (h, k), then the equation of the hyperbola becomes

$$\frac{(x-h)^2}{a^2} - \frac{(y-k)^2}{b^2} = 1$$

if the transverse axis is horizontal, or

$$\frac{(y-k)^2}{a^2} - \frac{(x-h)^2}{b^2} = 1$$

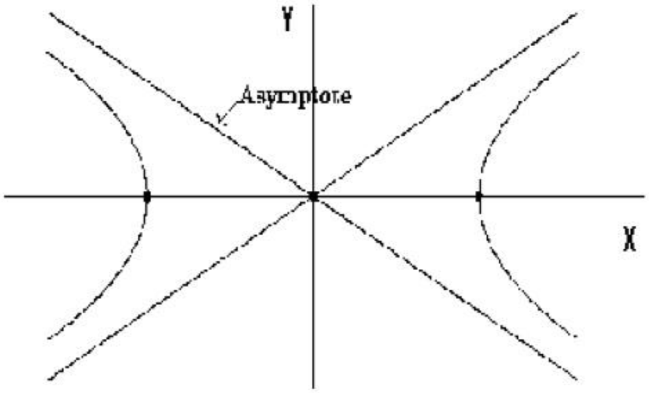
if the transverse axis is vertical.

A few more terms should be mentioned here before we move to some examples. First, as in the case of an ellipse, we say that the eccentricity of a hyperbola, denoted by e, is given by

$$e = \frac{c}{a},$$

or we say that the eccentricity of a hyperbola is given by the ratio of the distance between the foci to the distance between the vertices. Now in the case of a hyperbola, the distance between the foci is greater than the distance between the vertices. Hence, in the case of a hyperbola,

$$e > 1.$$



Recall that for the ellipse,

$$0 \leq e < 1.$$

Two final terms that we must mention are asymptotes and the conjugate axis. The two branches of a hyperbola are “bounded by” two straight lines, known as asymptotes. These asymptotes are easily drawn once one plots the vertices and the points (h, k+b) and (h, kb) and draws the rectangle which goes through these four points. The line segment joining (h, k+b) and (h, k-b) is called the conjugate axis. The asymptotes then are simply the lines which go through the corners of the rectangle.

But what are the actual equations of these asymptotes? Note that if the hyperbola is oriented horizontally, then the corners of this rectangle have the following coordinates:

$$(h+a, k+b), (h-a, k-b),$$

and

$$(h-a, k+b), (h+a, k-b).$$

Here I have paired these points in such a way that each asymptote goes through one pair of the points. Consider the first pair of points:

$$(h+a, k+b), (h-a, k-b)$$

Given two points, we can find the equation of the unique line going through the points using the point--slope form of the line. First, let us determine the slope of our line. We find this as ``change in y over change in x'' or ``rise over run''. In this case, we see that this slope is equal to

$$\frac{2b}{2a}$$

or simply

$$\frac{b}{a}.$$

Then, we also know that the line goes through the center (h, k). Hence, by the point--slope form of a line, we know that the equation of this asymptote is

$$y-k=\frac{b}{a}(x-h)$$

or

$$y=k+\frac{b}{a}(x-h).$$

The other asymptote in this case has a negative slope; which is given by

$$y-k=\frac{b}{a}(x-h)$$

or

$$y=k+\frac{b}{a}(x-h).$$

The other asymptote in this case has a negative slope; which is given by

$$-\frac{b}{a}.$$

Using the same argument, we see that this asymptote has equation

$$y=k-\frac{b}{a}(x-h).$$

What if the hyperbola is vertically oriented? Then one of the asymptote will go through the "corners" of the rectangle given by

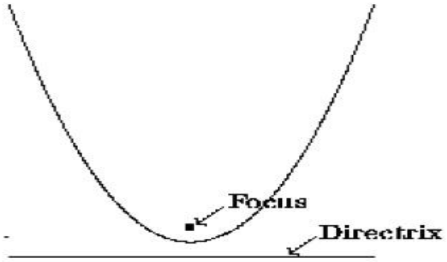
$$(h + b, k + a), (h - b, k - a).$$

Then the slope in this case will not be b/a but will be a/b. Hence, analogous to the work we just performed, we can show that the asymptotes of a vertically oriented hyperbola are determined by

$$y = k + \frac{a}{b}(x - h)$$

and

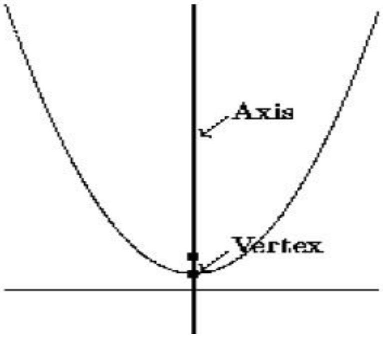
$$y = k - \frac{a}{b}(x - h).$$



Parabola

A **parabola** is the set of all points (x, y) that are the same distance from a fixed line (called the **directrix**) and a fixed point (**focus**) not on the directrix. See figure for the view of a parabola and its related focus and directrix.

Note that the graph of a parabola is similar to one branch of a hyperbola. However, you should realize that a parabola is **not** simply one branch of a hyperbola. Indeed, the branches of a hyperbola approach linear asymptotes, while a parabola does not do so. Several other terms exist which are associated with a parabola. The midpoint between the focus and directrix of the parabola is called the **vertex** and the line passing through the focus and vertex is called the **axis** of the parabola. (This is similar to the major axis of the ellipse and the transverse axis of the hyperbola.) See figure.



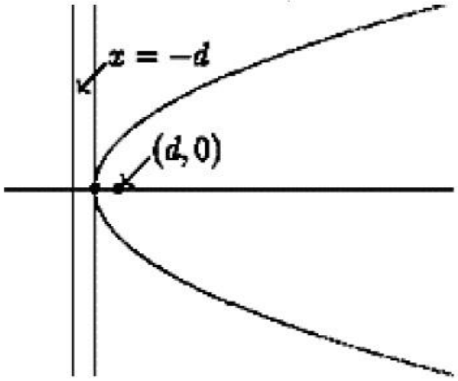
Now let's move to the standard algebraic equations for parabolas and note the four types of parabolas that exist. As we discuss the four types, you should notice the differences in the equations that are related to each of the four parabolas.

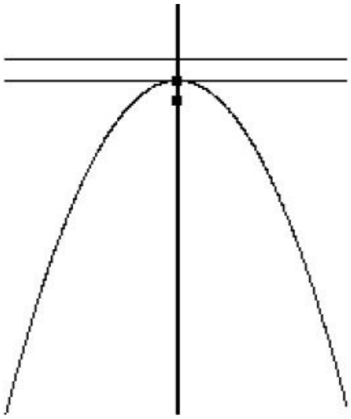
The **standard form** of the equation of the parabola with vertex at (0, 0) with the focus lying ***d*** units from the vertex is given by

if

if

—





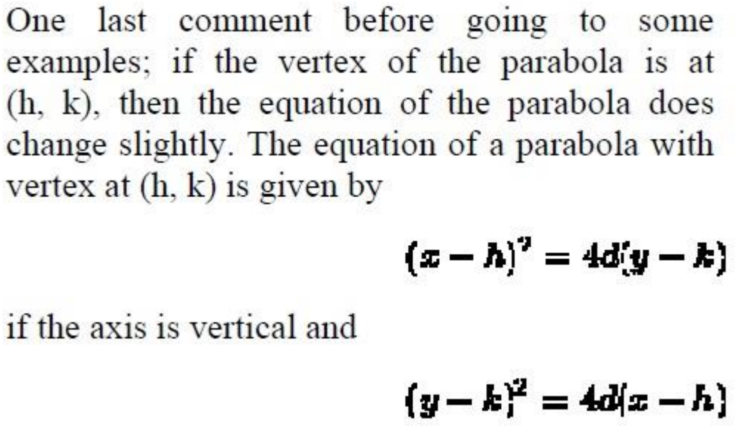
-d

5

Example with vertical axis and figure below for an example with horizontal axis.
Note here that we have assumed that:

$$\mathbf{d > 0}$$

It is also the case that ***d*** could be negative, which flips the orientation of the parabola.
Thus, we see that there are four different orientations of parabolas, which depend on a) which variable is squared (***x*** or ***y***) and b) whether ***d*** is positive or negative.



One last comment before going to some examples; if the vertex of the parabola is at (h, k), then the equation of the parabola does change slightly. The equation of a parabola with vertex at (h, k) is given by

$$(x-h)^2 = 4d(y-k)$$

if the axis is vertical and

$$(y-k)^2 = 4d(x-h)$$