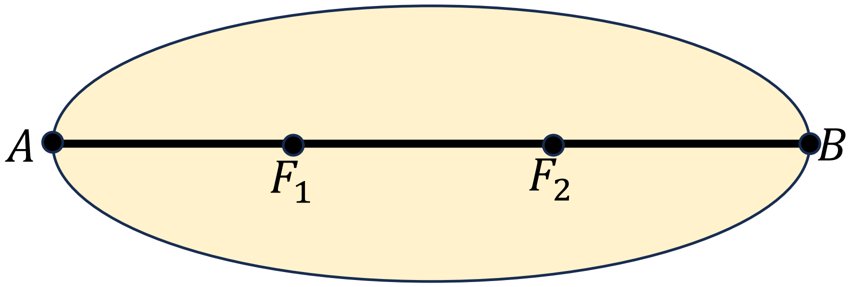
Ellipse Eclipse

An Ellipse is the set of all points where the sum of the distances to two foci is constant. You are given the coordinates of the two foci and the length of the Major Axis. What is that constant sum of distances. It turns out, it's just the length of the Major Axis! Look at this:



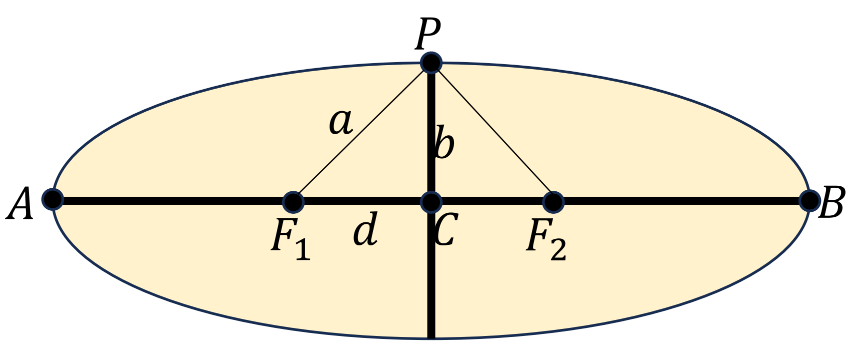
That constant is the sum of AF_1 and AF_2. But, since ellipses are symmetrical, AF_1 is the same as BF_2! Then $AF_2 + BF_2$ is just the Major Axis.

OK, now that we've got that, most programmers will probably use some combination of binary/ternary search to nail down the max/mins, but this can be done closed form, without searching, using a bit of trigonometry and calculus.

Consider an ellipse with its center at the origin and its Major Axis along the X-axis. I know that's not the case, but we'll start there. Such an ellipse can be represented parametrically:

$$x = a \sin\theta, y = b\cos\theta$$

Where a is the length of the Semimajor Axis (half of the Major Axis) and b is the length of the Semiminor Axis (which is the smallest radius, perpendicular to the Major Axis). The first, a, is just half of our Major Axis input. Now, we've got to find b.



Th Semiminor is CP. We know that F_1PF_2 is the length of the Major Axis (please excuse my poor drawing), so F_1P is the length of the Semimajor. We know that the center C is exactly halfway between F_1 and F_2. So, $C = (\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2})$. (Yeah, we could just take $\frac{F_1F_2}{2}$, but we'll need C later).

Now, let $d = F_1C$, and we can use the Pythagorean Theorem to get $b = \sqrt{a^2 - d^2}$

OK, that would be great if the ellipse were centered at the origin with its Major Axis along the X-axis, but that's not the case. Let's deal with the angle first. We can use `atan2` to get the angle of the Major Axis:

$$\alpha = \text{atan2}(y_2 - y_1, x_2 - x_1)$$

OK, everything is rotated by α. To rotate, the new coordinates are:

$$x' = x\cos\alpha - y\sin\alpha$$

$$y' = x\sin\alpha + y\cos\alpha$$

We have the parametric representation of the points of the ellipse, plug'em in!

$$x = a\cos\theta\cos\alpha - b\sin\theta\sin\alpha$$

$$y = a\cos\theta\sin\alpha + b\sin\theta\cos\alpha$$

Now, we want to find the max/min of both x and y. Those are inflection points – so all we have to do is take the derivative of each, set it to 0 and solve for θ (since that's the only variable here). We'll do x and let you do y, remembering that the derivative of $\sin\theta$ is $\cos\theta$ and the derivative of $\cos\theta$ is $-\sin\theta$.

$$\frac{d}{d\theta}a\cos\theta\cos\alpha - b\sin\theta\sin\alpha = -a\sin\theta\cos\alpha - b\cos\theta\sin\alpha = 0$$

$$a\sin\theta\cos\alpha = -b\cos\theta\sin\alpha$$

$$\frac{\sin\theta}{\cos\theta} = \frac{-b\sin\alpha}{a\cos\alpha}$$

$$\theta = \text{atan2}(-b\sin\alpha\,, a\cos\alpha)$$

Now, we can just plug that value of θ back into $x = a\cos\theta\cos\alpha - b\sin\theta\sin\alpha$ to get a max or min value of x. but wait! We don't know if that's a max or a min, and we're still assuming that the center is at the origin! That's OK. Take the absolute value, and that's a distance from the center in the X direction.

So, let $dx = |a\cos\theta\cos\alpha - b\sin\theta\sin\alpha|$, compute dy similarly, and then our bounding box goes from $(C_x - dx, C_y - dy\,)$ to $(C_x + dx, C_y + dy\,)$. I told you we'd need that center!

Here's some Java code for you:

```
a = a/2.0;
double cx = (x1+x2)/2.0;
double cy = (y1+y2)/2.0;

double dx = cx-x1;
double dy = cy-y1;
double d = Math.sqrt( dx*dx + dy*dy );
double b = Math.sqrt( a*a - d*d );

double alpha = -Math.atan2( y2-y1, x2-x1 );
double cosa = Math.cos( alpha );
double sina = Math.sin( alpha );

double theta = Math.atan2( -b*sina, a*cosa );
dx = Math.abs( a*Math.cos(theta)*cosa - b*Math.sin(theta)*sina );
double xlo = cx - dx;
double xhi = cx + dx;

theta = Math.atan2( b*cosa, a*sina );
dy = Math.abs( a*Math.cos(theta)*sina + b*Math.sin(theta)*cosa );
double ylo = cy - dy;
double yhi = cy + dy;

DecimalFormat df = new DecimalFormat( "0.000000" );
ps.println( df.format( xlo ) + " " + df.format( ylo )
    + " " + df.format( xhi ) + " " + df.format( yhi ) );
```