

```
In[1]:= Quit[]
```

```
In[1]:= ClearAll["Global`*"]
```

We can define the constraints of the plane the DC is in

```
In[2]:= right = ContourPlot[x2 == Cot[29.5 °] y + .09156,  
  {y, -1, 1}, {x2, .2, 1.8}, Frame → {True, True, False, False},  
  PlotLabel → "Right side limit of DC as a function of X and Y",  
  FrameLabel → {"y (meters)", "x (meters)"},  
  ContourStyle → Black, PlotLegends → Automatic];
```

```
In[3]:= left = ContourPlot[x2 == -Cot[29.5 °] y + .09156,  
  {y, -1, 1}, {x2, .2, 1.8}, Frame → {True, True, False, False},  
  PlotLabel → "Right side limit of DC as a function of X and Y",  
  FrameLabel → {"y (meters)", "x (meters)"},  
  ContourStyle → Black, PlotLegends → Automatic];
```

Similarly, the bottom limit of the DC can be modeled

```
In[4]:= bottom = ContourPlot[x0 == .22070,  
  {y, -1, 1}, {x0, .2, 1.8}, Frame → {True, True, False, False},  
  PlotLabel → "Lower limit on DC as a function of X and Y",  
  FrameLabel → {"y(meters)", "x(meters)"},  
  ContourStyle → Gray, PlotLegends → Automatic];
```

We can define the x coordinate of the wires as they cross the midpoint plane as shown earlier.

```
In[5]:= x0forWires[number_] := .23168 + .01337 * (number);
```

We can define the point midway between two parallel lines as the point where one wire is recorded versus its next highest neighbor

```
In[6]:= x0forWireMiddles[number_] :=  
  ((.23168 + .01337 * (number)) + (.23168 + .01337 * (number + 1))) / 2;
```

All of the conditions dependent on θ and ϕ

In[7]:=

```

Δa := FullSimplify[(R Sin[θ °]) / 2 (Csc[65 ° - θ °] - Csc[115 ° - θ °]), θ > 0];
e := Sin[25 °] / Cos[θ °];
a := FullSimplify[(R Sin[θ °]) / 2 (Csc[65 ° - θ °] + Csc[115 ° - θ °]), θ > 0];
rD1 := Simplify[(a e - Δa) Tan[65 °] Cos[θ °], θ > 0];
rD2 := Simplify[(a e + Δa) Tan[65 °] Cos[θ °], θ > 0];
xD1 := Simplify[rD1 Cos[φ °]];
yD1 := Simplify[rD1 Sin[φ °]];
zD1 := Simplify[rD1 Cot[θ °], θ > 0];
xD2 := Simplify[rD2 Cos[φ °], θ > 0];
yD2 := Simplify[rD2 Sin[φ °], θ > 0];
zD2 := Simplify[rD2 Cot[θ °], θ > 0];
xP := Simplify[(R Cos[φ °]) / (Cot[θ °] + Cos[φ °] Cot[65 °]), θ > 0];
yP := Simplify[(R Sin[φ °]) / (Cot[θ °] + Cos[φ °] Cot[65 °]), θ > 0];
zP := Simplify[(R Cot[θ °]) / (Cot[θ °] + Cos[φ °] Cot[65 °]), θ > 0];
x1 := Simplify[(rD22 - rD12 + Cot[θ °]2 (rD22 - rD12) -
  2 xP (xD2 - xD1) - 2 yP (yD2 - yD1) - 2 zP (zD2 - zD1)] / (4 a e) - a e, θ > 0];
x := Simplify[x1 - Δa + a e, θ > 0];
xCenter := x + Δa;
n := -957.412 / (Tan[θ °] + 2.14437) + 430.626;
D2P := Simplify[(x2 - xP2)2 + (y2 - yP2)2 + (z2 - zP2)2].5, θ > 0] // N
D1P := Simplify[(x2 - xD12)2 + (y2 - yD12)2 + (z2 - zD12)2].5, θ > 0] // N;
y := Simplify[(D1P2 - x12).5, θ > 0] // N;
b := Simplify[a (1 - e), θ > 0] // N;
R = 2.52934271645;

```

We can define the xy position on the DC plane as a function of ϕ for and limit the angles with the wall on the right and left hand sides. By symmetry these angles are equal, but opposite in sign.

```

In[30]:= Limits = Table[φ /. NSolve[ $\sqrt{a^2 \left(1 - \frac{y^2}{b^2}\right) - \Delta a} == \text{Cot}[29.5^\circ] y + .09156$ , φ], {θ, 5, 40}];

```

```

Lim = Table[0, {rows, 1, 36}];
For[rows = 1, rows < 37, rows++,
  For[i = 1, i < 5, i++,
    If[Limits[[rows, i]] > 0 && Limits[[rows, i]] < 30,
      Lim[[rows]] = Limits[[rows, i]];
    ];
  ];
];

```

The angle must have 4 subtracted to equal the line element numbering

```

In[33]:= Lim[[7 - 4]]

```

```

Out[33]= 20.4531

```

This limiting condition will record the angle ϕ in degrees within an array of wire number(1-112) horizontally and angle θ (5° - 40°) vertically

```
In[34]:= RightSolutions =
Table[{ $\phi$  /. Solve[ $\sqrt{a^2 \left(1 - \frac{y^2}{b^2}\right) - \Delta a} = \text{Tan}[6^\circ] y + x0$ forWireMiddles[number] ,  $\phi$ ]},
{ $\theta$ , 5, 40}, {number, 1, 112}];
LineRight = Table[{ $\phi$ , columns}, {rows, 1, 36}, {columns, 1, 112}];
For[rows = 1, rows < 37, rows++,
For[columns = 1, columns < 113, columns++,
For[i = 1, i < 5, i++,
If[RightSolutions[[rows, columns, 1, i]] > 0 &&
RightSolutions[[rows, columns, 1, i]] < Lim[[rows]],
LineRight[[rows, columns]] = {RightSolutions[[rows,
columns, 1, i]], columns + .5};
];
];
];
];
```

At $\phi=0$, the condition of $n = \frac{-959.637}{\tan \theta^\circ + 2.14437} + 430.189$ should be met

```
In[37]:= f[ $\theta$ for $\phi$ at0_] :=  $\frac{-959.637}{\text{Tan}[\theta\text{for}\phi\text{at}0^\circ] + 2.14437} + 430.189$ 
```

```
In[38]:= f[40]
```

```
Out[38]:= 108.538
```

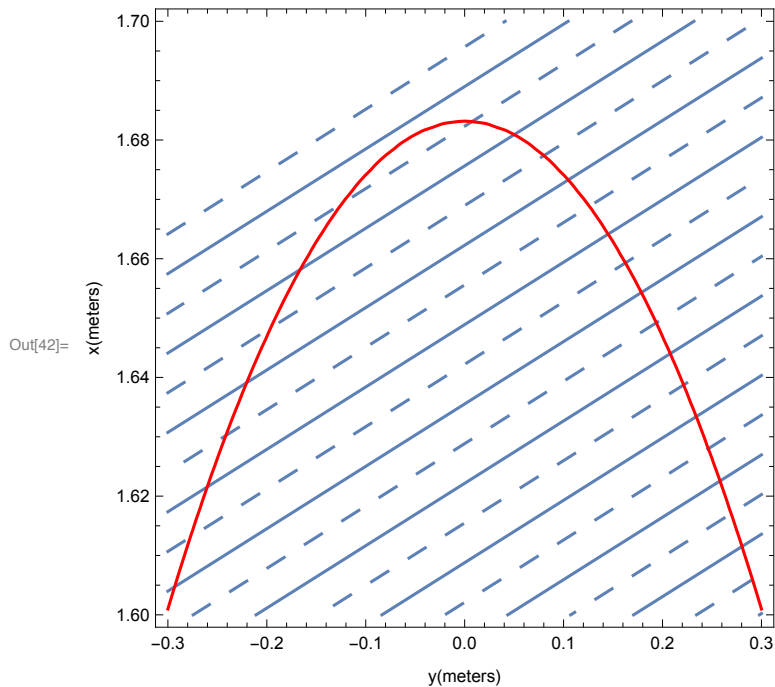
This implies that for $\theta=40$, we know that the wire number to be 108.538. This corresponds to the position being in between 108.5 and 109.5, hence it falls into the 109 "bin".

Testing the geometry

```

In[39]:= ClearAll[θ];
θ = 40;
ellipse40 = ContourPlot[ $\frac{(x + \Delta a)^2}{a^2} + \frac{y^2}{b^2} == 1$ ,
  {y, -.3, .3}, {x, 1.6, 1.7}, Frame → {True, True, False, False},
  PlotLabel → "XY position on DC as a function of φ for θ=40°",
  FrameLabel → {"y (meters)", "x (meters)"},
  ContourStyle → Red, PlotLegends → Automatic];
Show[Table[ContourPlot[xWire == Tan[6 °] yWire + x0forWires[number],
  {yWire, -0.3, 0.3}, {xWire, 1.6, 1.7},
  FrameLabel → {"y(meters)", "x(meters)"}], {number, 99, 109}],
  Table[ContourPlot[xWire == Tan[6 °] yWire + x0forWireMiddles[number2],
  {yWire, -0.3, 0.3}, {xWire, 1.6, 1.7},
  ContourStyle → {Dashing[Large]}], {number2, 99, 109}],
  bottom, right, left, ellipse40]

```



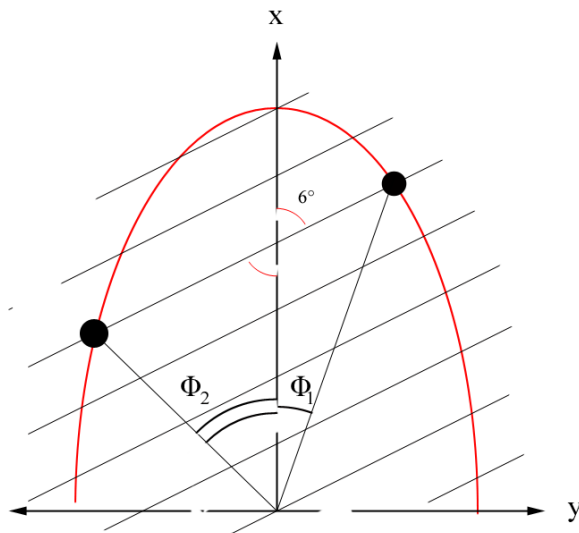
Clearing wire numbers greater than the “wire number” at $\phi=0$

```

In[43]:= For[rows = 1, rows < 37, rows++,
  For[columns = 1, columns < 113, columns++,
    If[LineRight[[rows, columns, 2]] > f[rows + 4],
      LineRight[[rows, columns, 1]] = {ϕ}
    ]
  ]
];

```

The fact the equation for the y component in the plane of the detector uses a square root function, we know functions within Mathematica, only the positive values of y will be used. We can find the angle ϕ where the elliptical path hits the right and left wires. This corresponds to hits at a constant θ occurring on different wires. The elliptical path hits the right and left wires in the 1st quadrant. On the left hand side, the wire number limit is lower since the slope of this plane is positive.



By symmetry, we know that the positions within the 2nd quadrant can be reflected into the 1st quadrant by taking the opposite slope of the wire function

```
ClearAll[ $\theta$ ];
```

```
In[88]:= LeftSolutions =
```

```
Table[{ $\phi$  /. Solve[ $\sqrt{a^2 \left(1 - \frac{y^2}{b^2}\right) - \Delta a} == -\text{Tan}[6^\circ] y + x_0$ forWireMiddles[number] ,  $\phi$ ],
  number}, { $\theta$ , 5, 40}, {number, 1, 112}];
LineLeft = Table[{ $\phi$ , columns}, {rows, 1, 36}, {columns, 1, 112}];
For[rows = 1, rows < 37, rows++,
  For[columns = 1, columns < 113, columns++,
    For[i = 1, i < 7, i++,
      If[LeftSolutions[[rows, columns, 1, i]] > 0 &&
        LeftSolutions[[rows, columns, 1, i]] < 25,
        LineLeft[[rows, columns]] = {LeftSolutions[[rows,
          columns, 1, i]], columns + .5};
    ];
  ];
  For[i = 1, i < 6, i++,
    If[LeftSolutions[[rows, columns, 1, i]] > 0 &&
      LeftSolutions[[rows, columns, 1, i]] < 25,
      LineLeft[[rows, columns + 1]] = {LeftSolutions[[rows,
        columns, 1, i]], columns + .5};
    ];
  ];
];
```

We can combine the angles recorded in both quadrants

Collecting the ϕ angles as an x variable with the wire as the y variable for a constant value of θ first for the right hand side

```
In[91]:= constant $\theta$ right = Table[0, {rows, 1, 36}, {columns, 1, 112}];
constant $\theta$  = Table[0, {rows, 1, 36}, {columns, 1, 112}];
```

```
In[93]:= For[rows = 1, rows < 37, rows++,
  For[columns = 1, columns < 113, columns++,
    constant $\theta$ right[[rows, columns]] = LineRight[[rows, columns]]
  ]
];
```

Flattening the list elements to numerical values

```
In[94]:= For[rows = 1, rows < 37, rows++,
  constant $\theta$ right[[rows]] = Flatten[constant $\theta$ right[[rows]], 2]];
```

Partitioning the data into $\{\phi, \text{wire number}\}$ combinations that increase with respect to the ϕ component

```
In[95]:= For[rows = 1, rows < 37, rows++,
  constant $\theta$ [[rows]] = Partition[constant $\theta$ [[rows]], 2]
];
```

Sorting the x values from smallest to largest

```
In[96]:= For[i = 1, i < 37, i++,
  constant $\theta$ [[i]] = Sort[constant $\theta$ [[i]]]
];
```

Now doing the same for the left hand side

```
In[97]:= constant $\theta$ left = Table[0, {rows, 1, 36}, {columns, 1, 112}];
constant $\theta$ l = Table[0, {rows, 1, 36}, {columns, 1, 112}];
```

```
In[99]:= For[rows = 1, rows < 37, rows++,
  For[columns = 1, columns < 113, columns++,
    constant $\theta$ left[[rows, columns]] = LineLeft[[rows, columns]]
  ]
];
```

Changing the sign to reflect positions within the 2nd quadrant

```
In[100]:= For[rows = 1, rows < 37, rows++,
  For[columns = 1, columns < 113, columns++,
    constant $\theta$ left[[rows, columns, 1]] = -1 constant $\theta$ left[[rows, columns, 1]]
  ]
];
```

Get rid of negative ϕ terms, since this has a dimension of 2, a condition that be utilized for a different object condition

```
In[101]:= For[rows = 1, rows < 37, rows++,
  For[columns = 1, columns < 113, columns++,
    If[constant $\theta$ left[[rows, columns, 1]] == - $\phi$ ,
      constant $\theta$ left[[rows, columns, 1]] =  $\phi$ 
    ]
  ]
];
```

There could be two possible solutions on the left hand side for the ellipse as it crosses it's maximum, this separates the solutions and puts on value in the next largest bin with the same wire value.

```
In[102]:= For[rows = 1, rows < 37, rows++,
  For[columns = 1, columns < 113, columns++,
    If[Dimensions[constantøleft[[rows, columns, 1]]] == {2},
      constantøleft[[rows, columns + 1]] =
        {constantøleft[[rows, columns, 1, 2]], constantøleft[[rows, columns, 2]]};
      constantøleft[[rows, columns]] = {constantøleft[[rows, columns, 1, 1]],
        constantøleft[[rows, columns, 2]]}
    ]
  ]
]
```

Sorting the x values from smallest to largest

```
In[103]:= For[i = 1, i < 37, i++,
  constantøleft[[i]] = Sort[constantøleft[[i]]]
];
```

Get rid of extra braces and make everything uniformly sized, i.e. {x,y}

```
In[104]:= For[rows = 1, rows < 37, rows++,
  constantøl[[rows]] = Partition[Flatten[constantøleft[[rows]]], 2]
];
```

Make sure everything is sorted correctly

```
In[105]:= For[i = 1, i < 37, i++,
  constantøl[[i]] = Sort[constantøl[[i]]]
];
```

Merging the left and right side data,

```
In[106]:= constantø = Table[0, {rows, 1, 36}, {columns, 1, 224}];
```

```
In[107]:= For[row = 1, row < 37, row++,
  constantø[[row]] = Union[constantør[[row]], constantøl[[row]]]
];
```

Fill elements with null characters if numerical data is not present

```
In[108]:= For[rows = 1, rows < 37, rows++,
  For[columns = 1, columns < 226, columns++,
    If[constantø[[rows, columns, 1]] == ϕ, constantø[[rows, columns]] = {Nothing}]
  ]
];
```

Change the dimension of the list to only have numerical elements

```
In[109]:= For[rows = 1, rows < 37, rows++,
  constantø[[rows]] = Partition[Flatten[constantø[[rows]]], 2]
];
```

An ellipse can be parameterized in the form

$$\begin{aligned}x &= a \cos(t) \\ y &= b \sin(t)\end{aligned}$$

Where a and b are the semi-major and semi-minor axes found earlier. For the case of an off center ellipse at $(-\Delta a, 0)$ in the plane of the sensor, this becomes

$$\begin{aligned}x &= -\Delta a + a \cos(t) \\ y &= b \sin(t)\end{aligned}$$

The parameterization t , is measured from the beamline position

```
In[110]:= xyPoints = constantθ;
constantθxy = constantθ;
Parameterized = constantθ;
RowLengths = Table[{Nothing}, {i, 1, 36}];
For[rows = 1, rows < 37, rows++,
  RowLengths[[rows]] = Length[constantθ[[rows]]];
  For[columns = 1, columns < RowLengths[[rows]] + 1, columns++,
    θ = rows + 4;
    φ = constantθ[[rows, columns, 1]];

    constantθxy[[rows, columns]] = {y,  $\sqrt{a^2 \left(1 - \frac{y^2}{b^2}\right) - \Delta a}$ };

    xyPoints[[rows, columns]] = {y,  $\sqrt{a^2 \left(1 - \frac{y^2}{b^2}\right) - \Delta a}$ ,
      constantθ[[rows, columns, 1]], constantθ[[rows, columns, 2]]};
    Parameterized[[rows, columns]] = {ArcSin[ $\frac{xyPoints[[rows, columns, 1]]}{b}$ ],
      ArcCos[ $\frac{xyPoints[[rows, columns, 2]] + \Delta a}{a}$ ]}];
    If[constantθ[[rows, columns, 1]] < 0,
      xyPoints[[rows, columns, 1]] = -xyPoints[[rows, columns, 1]];
      constantθxy[[rows, columns, 1]] = -constantθxy[[rows, columns, 1]];
      Parameterized[[rows, columns, 1]] = -Parameterized[[rows, columns, 1]];
      Parameterized[[rows, columns, 2]] = -Parameterized[[rows, columns, 2]];
    ]
  ]
];
ClearAll[θ, φ]
```

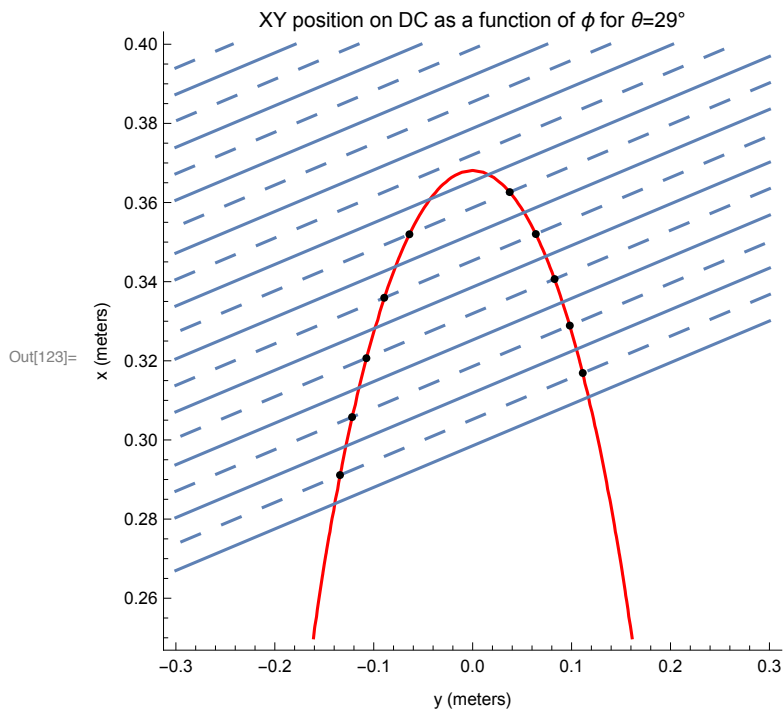
Check to see that both parameterized components are equal

```
In[116]:= RowLengths = Table[{Nothing}, {i, 1, 36}];
Para = Parameterized;
For[rows = 1, rows < 37, rows++,
  RowLengths[[rows]] = Length[constant $\theta$ [[rows]]];
  For[columns = 1, columns < RowLengths[[rows]] + 1, columns++,
    If[Abs[Parameterized[[rows, columns, 1]] -
      Parameterized[[rows, columns, 2]]] > .0001,
      Print[rows, columns];
    ];
    If[Abs[Parameterized[[rows, columns, 1]] -
      Parameterized[[rows, columns, 2]]] < .0001,
      Para[[rows, columns]] = {Parameterized[[rows, columns, 1]],
        xyPoints[[rows, columns, 4]]};
    ];
  ];
];
```

```

In[119]:= DataXY = Table[ListPlot[{constant $\theta$ xy[[i]]}], PlotStyle  $\rightarrow$  Black, AxesLabel  $\rightarrow$  {"y", "x"},
          PlotLabel  $\rightarrow$  "DC Wire for Constant  $\theta$  as a Function of  $\phi$ ", {i, 1, 36}];
ClearAll[ $\theta$ ];
 $\theta = 8$ ;
ellipse8 = ContourPlot[ $\frac{(x + \Delta a)^2}{a^2} + \frac{y^2}{b^2} = 1$ ,
          {y, -.3, .3}, {x, .25, .4}, Frame  $\rightarrow$  {True, True, False, False},
          PlotLabel  $\rightarrow$  "XY position on DC as a function of  $\phi$  for  $\theta=29^\circ$ ",
          FrameLabel  $\rightarrow$  {"y (meters)", "x (meters)"},
          ContourStyle  $\rightarrow$  Red, PlotLegends  $\rightarrow$  Automatic];
Show[ellipse8, Table[ContourPlot[xWire == Tan[6  $^\circ$ ] yWire + x $\theta$ forWires[number],
          {yWire, -0.3, 0.3}, {xWire, .25, .4},
          FrameLabel  $\rightarrow$  {"y(meters)", "x(meters)"}], {number, 5, 15}],
          Table[ContourPlot[xWire == Tan[6  $^\circ$ ] yWire + x $\theta$ forWireMiddles[number2],
          {yWire, -0.3, 0.3}, {xWire, .25, .4},
          ContourStyle  $\rightarrow$  {Dashing[Large]}], {number2, 5, 15}], DataXY[[4]]]
ClearAll[ $\theta$ ];

```



```
Names["Global`*"]
```

```
{a, b, bottom, columns, constant $\theta$ , constant $\theta$ l, constant $\theta$ left, constant $\theta$ r,  
constant $\theta$ right, constant $\theta$ xy, D1P, D2P, DataXY, e, ellipse40, ellipse8, f, i, left,  
LeftSolutions, Lim, Limits, LineLeft, LineRight, n, number, number2, number$,  
Para, Parameterized, R, rD1, rD2, right, RightSolutions, row, RowLengths, rows,  
x, x0, x0forWireMiddles, x0forWires, x1, x2, xCenter, xD1, xD2, xP, xWire,  
xyPoints, y, yD1, yD2, yP, yWire, zD1, zD2, zP,  $\Delta$ a,  $\theta$ ,  $\theta$ for $\phi$ at0,  $\theta$ for $\phi$ at0$,  $\phi$ }
```

```
In[125]:= DumpSave["Part1.mx", "Global`"]
```

```
Out[125]= {Global`}
```