Our problem statement

In a Cartesian plane, consider two distinct sets of points denoted as A and B. Set A exclusively consists of points located along the X-axis, serving as a reflective plane mirror. Conversely, set B comprises points positioned along an unknown, arbitrary reflective curve represented by f(x). These points represent the locations where a ray of light, originating from the origin and striking point P (which belongs to set B), undergoes successive reflections. Given this configuration, determine nature of the reflective curve f(x).

Equation Of Reflective Curve In Parametric and Cartesian Form

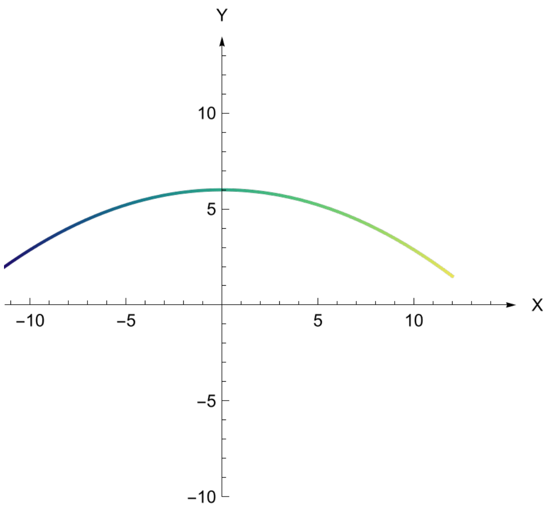
In[•]:=
```
curveparaeqn = {8t, -2t^2+6}
curvecarteqn = Last[List @@ Reduce[Eliminate[{curveparaeqn[[1]]==x,curveparaeqn[[2]]==y}, t],y]]
```

Out[•]= $\{8\,t,\ 6-2\,t^2\}$

Out[•]= $\frac{1}{32}\times\left(192-x^2\right)$

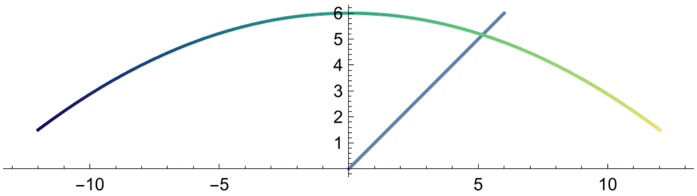
Plot of the equation

In[•]:=
```
curve = ParametricPlot[curveparaeqn, {t, -1.5, 1.5},PlotRange→{-10, 14},AxesLabel→{"X", "Y"},
```

Out[•]=


Incident ray of light at 45 degrees to x axis with source at origin

In[•]:=
```
sourceparaeqn = {t, t};
sourcecarteqn = x;
sourceline = ParametricPlot[sourceparaeqn, {t, 0, 6}, PlotRange→All];
Show[sourceline, curve]
ptlist = {};
ptlist = AppendTo[ptlist, {x /. N[Solve[sourcecarteqn==curvecarteqn, x]][[2]] , curvecarteqn /.
```

Out[•]=


Out[•]= {{5.16601, 5.16601}}

Functions for reflection angles and equations

In[•]:=
```
curverefangle[incipt_,prevrefangle_]:=
    Reduce[(prevrefangle-(-1/(D[curvecarteqn, x] /. {x→incipt[[1]]})))/(1+prevrefangle*(-1/(D
        == ((-1/(D[curvecarteqn, x] /.{x→incipt[[1]]}))-m)/(1+(-1/(D[curvecarteqn, x] /.{x→in

curverefeqn[incipt_, refangle_]:=  Reduce[y-(curvecarteqn /. {x→incipt[[1]]} )== Last[List @@ r

plincipt[refeqn_] := {x /. Solve[refeqn /. y→0][[1]][[1]], 0};

plrefeqn[incipt_, angle_] := Last[List @@ Reduce[y-0==-Last[List @@ angle](x-incipt[[1]]),y]];

curveinci[plrefeqn_] := Module[{b},
                              For[z = N[Solve[plrefeqn==curvecarteqn, x]]; i=0, i<Length[a]
                              If[(x /. z[[i]])>-10 && (x /. z[[i]])<10 , b = {x /. z[[i]], curve
                              b]
```

In[•]:=
```
eqnlist = {};
ptlist = {{0,0},{5.166010488516726`,5.166010488516724`}};
mergefunction[list_] := Module[{},
                          a = curverefangle[list[[1]], list[[2]]];
                          b = curverefeqn[list[[1]],a];
                          c = plincipt[curverefeqn[list[[1]],a]];
                          d = plrefeqn[c, a];
                          e = curveinci[d];
                         eqnlist = AppendTo[eqnlist, {Last[List @@b],d}];
                         ptlist = AppendTo[ptlist, c];
                         ptlist = AppendTo[ptlist, e];
                         z = {e,-Last[List @@ a]};
                         z]
```

12 reflection pairs after first incident

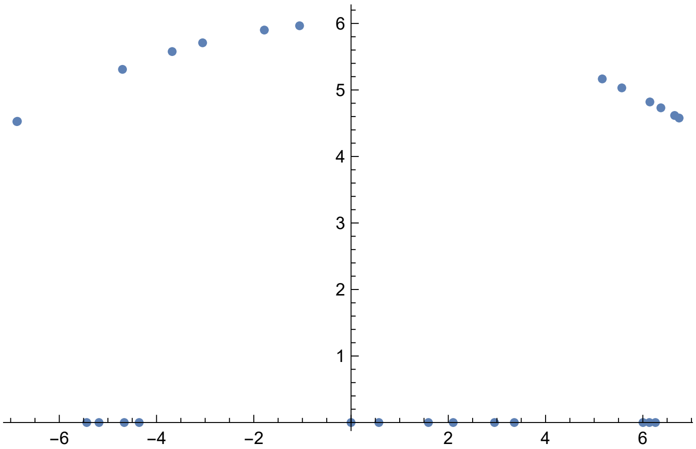
In[•]:=
```
Nest[mergefunction,{{5.166010488516726`,5.166010488516724`}, 1} , 12]
eqnlist;
ptlist;
```

Out[•]= {{-4.70059, 5.30951}, -1.00704}

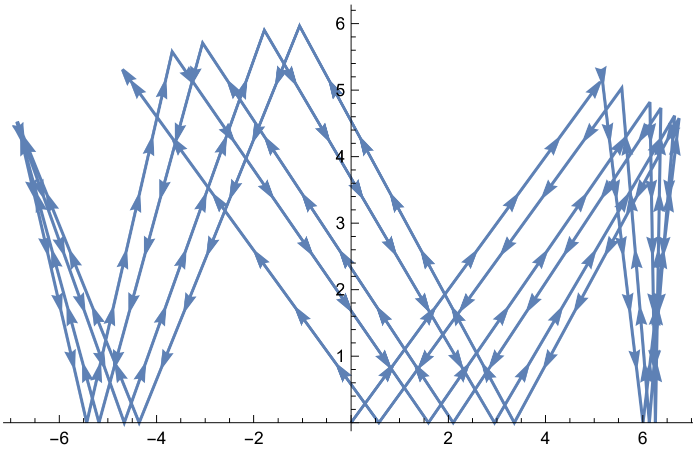
In[•]:=

```
z1 = ListPlot[ptlist]
z2 = ListLinePlot[ptlist, Axes→True, AxesOrigin→{0,0}, MeshFunctions → {#2 &}, Mesh → 6, Mes
  MeshShading → {Arrowheads[Small]}, DataRange → {0, 4 Pi}] /. Line → Arrow
```

Out[•]=



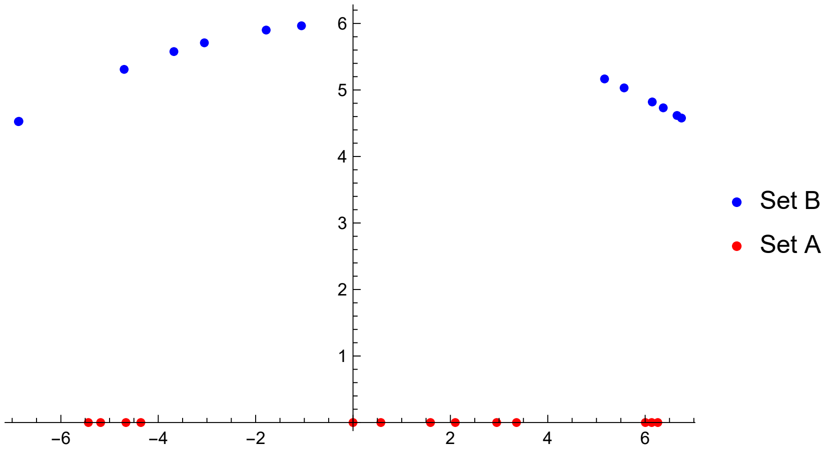
Out[•]=



Valid Path Formations

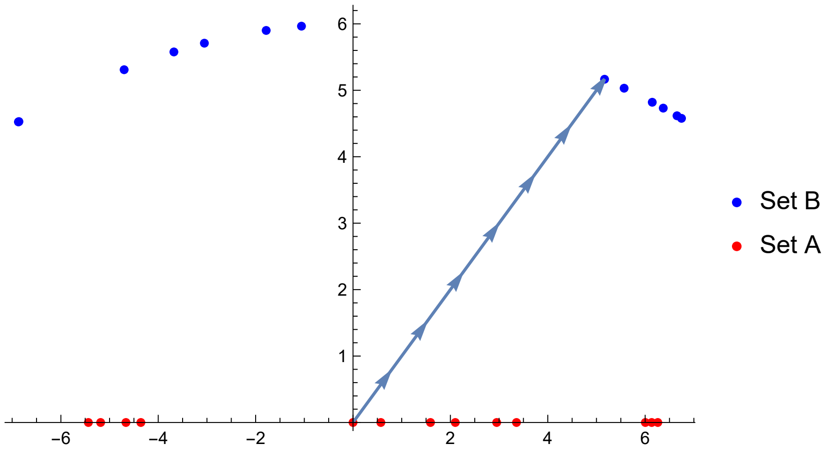
In[∘]:=
```
intersecpts = ptlist;
For[xaxispts = {}; i=0, i<Length[intersecpts], i++; If[intersecpts[[i, 2]]==0, AppendTo[xaxispts
curvepts = Sort[DeleteCases[intersecpts, Alternatives @@ xaxispts]];
xaxispts = Sort[xaxispts];
sepplot = ListPlot[{curvepts, xaxispts}, PlotStyle→{Blue, Red},PlotLegends→{"Set B", "Set A"}
```

Out[∘]=



In[∘]:=
```
z1 = ListLinePlot[{{0,0},{5.166010488516726`,5.166010488516724`}},Axes→True, AxesOrigin→{0,0}
  MeshShading → {Arrowheads[Small]}, DataRange → {0, 4 Pi}] /. Line → Arrow;
Show[sepplot,z1 ]
```

Out[∘]=

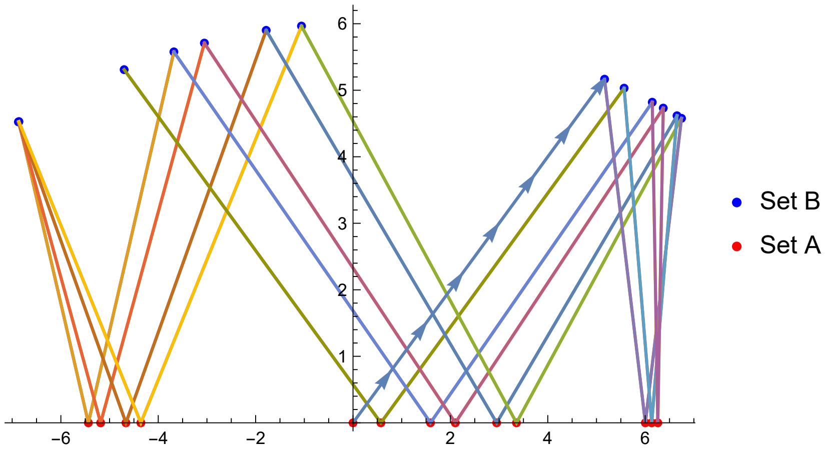


In[∘]:=
```
Clear[i, j, k]
reflecpathls = {};
For[i=0, i<Length[xaxispts], i++;
    For[j=0, j<Length[curvepts], j++;
        For[k=0, k<Length[curvepts], k++;
            If[curvepts[[j]]==curvepts[[k]], Continue[]];
            If[(curvepts[[k]][[2]]-xaxispts[[i]][[2]])/(curvepts[[k]][[1]]-xaxispts[[i]][[1]]) == -(curvepts
                reflecpathls = AppendTo[reflecpathls, {curvepts[[k]], xaxispts[[i]], curvepts[[j]]}
reflecpathls;
```

In[ ]:=
```
reversepathls = {};
For[i=0, i<Length[reflecpathls], i++;
    reversepathls = AppendTo[reversepathls, Reverse[reflecpathls[[i]]]]]
reversepathls;
```

In[ ]:=
```
legalpaths = Join[reflecpathls , reversepathls];
```

In[ ]:=
```
Show[sepplot, ListLinePlot[reflecpathls], z1]
```

Out[ ]=


20 reflection pairs after first incident

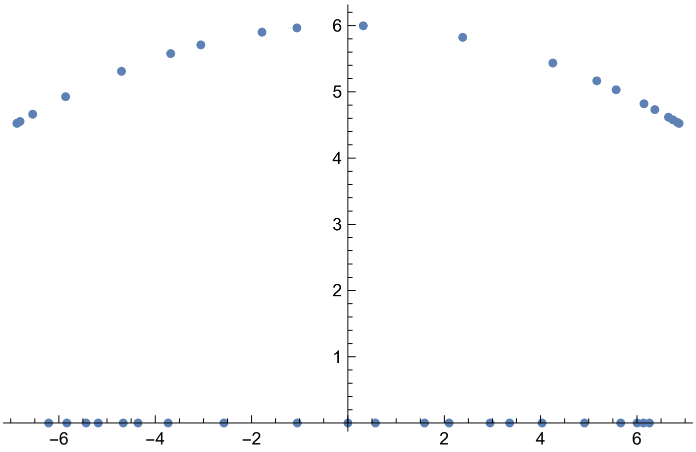
In[ ]:=
```
eqnlist = {};
ptlist = {{0,0},{5.166010488516726`,5.166010488516724`}};
Nest[mergefunction,{{5.166010488516726`,5.166010488516724`}, 1} , 20]
eqnlist;
ptlist;
```

Out[ ]= {{6.87506, 4.52292}, 2.30317}

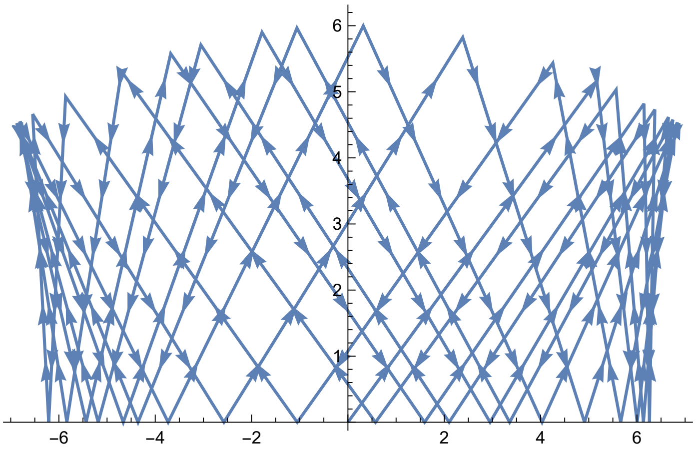
In[•]:=

```
z1 = ListPlot[ptlist]
z2 = ListLinePlot[ptlist, Axes→True, AxesOrigin→{0,0}, MeshFunctions → {#2 &}, Mesh → 6, Mesl
  MeshShading → {Arrowheads[Small]}, DataRange → {0, 4 Pi}] /. Line → Arrow
```

Out[•]=



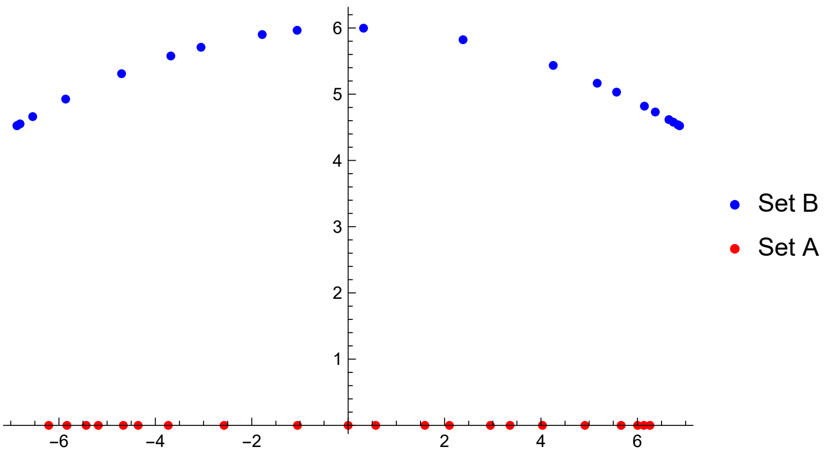
Out[•]=



In[•]:=

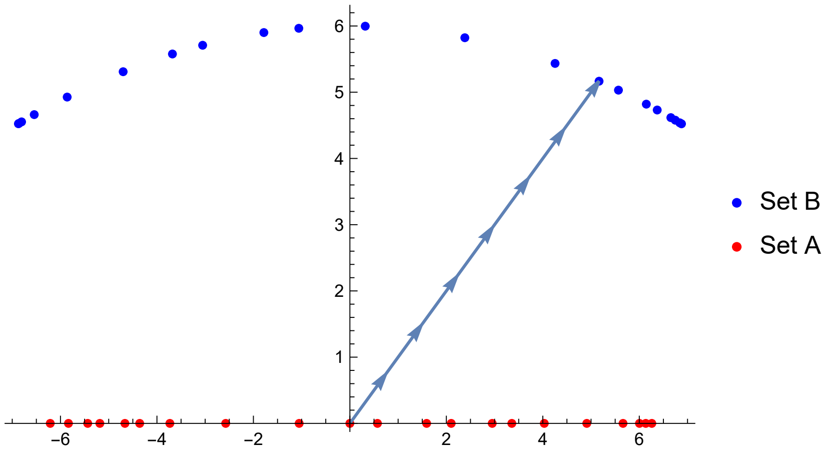
```
intersecpts = ptlist;
For[xaxispts = {}; i=0, i<Length[intersecpts], i++; If[intersecpts[[i, 2]]==0, AppendTo[xaxispts
curvepts = Sort[DeleteCases[intersecpts, Alternatives @@ xaxispts]];
xaxispts = Sort[xaxispts];
sepplot = ListPlot[{curvepts, xaxispts}, PlotStyle→{Blue, Red},PlotLegends→{"Set B", "Set A"}
```

Out[•]=



In[∘]:=
```
z1 = ListLinePlot[{{0,0},{5.166010488516726`,5.166010488516724`}},Axes→True, AxesOrigin→{0,0}
  MeshShading → {Arrowheads[Small]}, DataRange → {0, 4 Pi}] /. Line → Arrow;
Show[sepplot,z1 ]
```

Out[∘]=



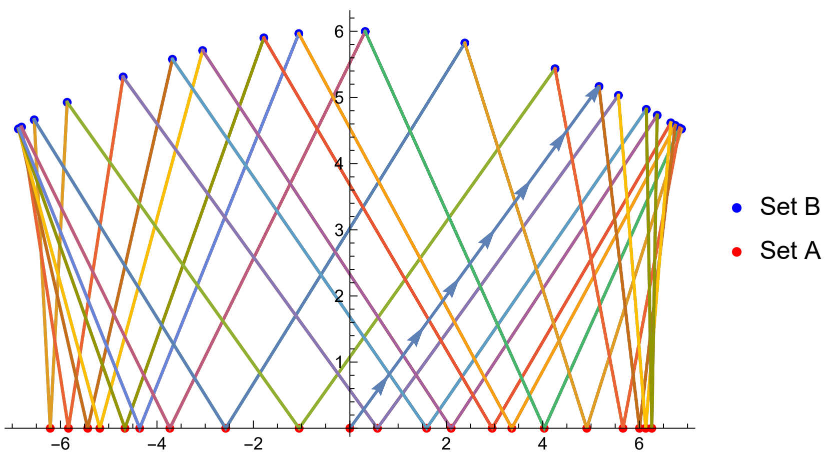
In[∘]:=
```
Clear[i, j, k]
reflecpathls = {};
For[i=0, i<Length[xaxispts], i++;
    For[j=0, j<Length[curvepts], j++;
        For[k=0, k<Length[curvepts], k++;
            If[curvepts⟦j⟧==curvepts⟦k⟧, Continue[]];
            If[(curvepts⟦k⟧⟦2⟧-xaxispts⟦i⟧⟦2⟧)/(curvepts⟦k⟧⟦1⟧-xaxispts⟦i⟧⟦1⟧) == -(curvepts
                reflecpathls = AppendTo[reflecpathls, {curvepts⟦k⟧, xaxispts⟦i⟧, curvepts⟦j⟧}
reflecpathls;
```

In[∘]:=
```
reversepathls = {};
For[i=0, i<Length[reflecpathls], i++;
    reversepathls = AppendTo[reversepathls, Reverse[reflecpathls⟦i⟧]]]
reversepathls;
```

In[∘]:=
```
legalpaths = Join[reflecpathls , reversepathls];
```

```
Show[sepplot, ListLinePlot[reflecpathls], z1]
```

Out[∘]=



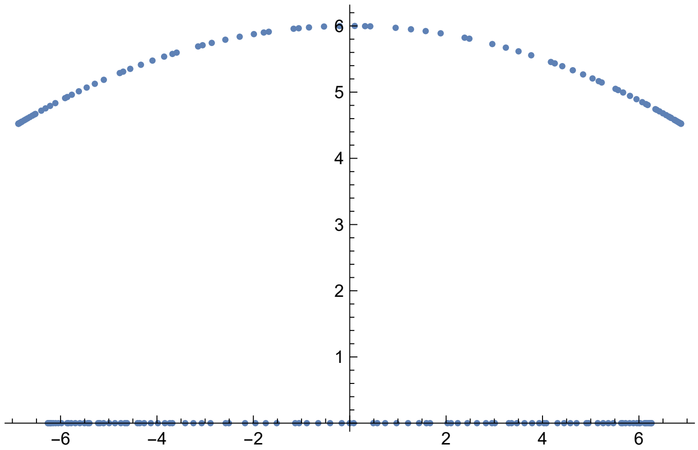
100 reflection pairs after first incident

In[∘]:=
```
eqnlist = {};
ptlist = {{0,0},{5.166010488516726`,5.166010488516724`}};
Nest[mergefunction,{{5.166010488516726`,5.166010488516724`}, 1} , 100]
eqnlist;
ptlist;
```

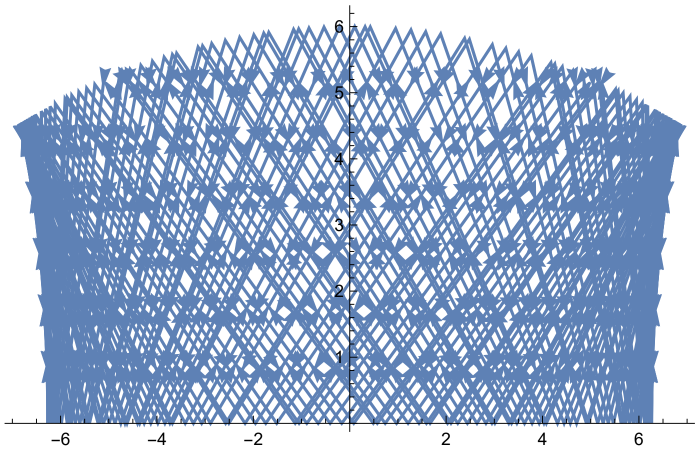
Out[∘]= {{-6.58482, 4.645}, -11.7439}

In[∘]:=
```
z1 = ListPlot[ptlist]
z2 = ListLinePlot[ptlist, Axes→True, AxesOrigin→{0,0}, MeshFunctions → {#2 &}, Mesh → 6, Mesl
  MeshShading → {Arrowheads[Small]}, DataRange → {0, 4 Pi}] /. Line → Arrow
```

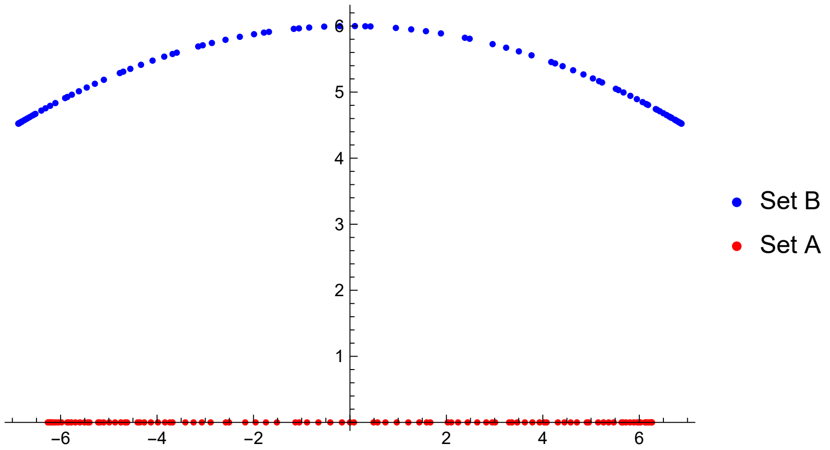
Out[∘]=



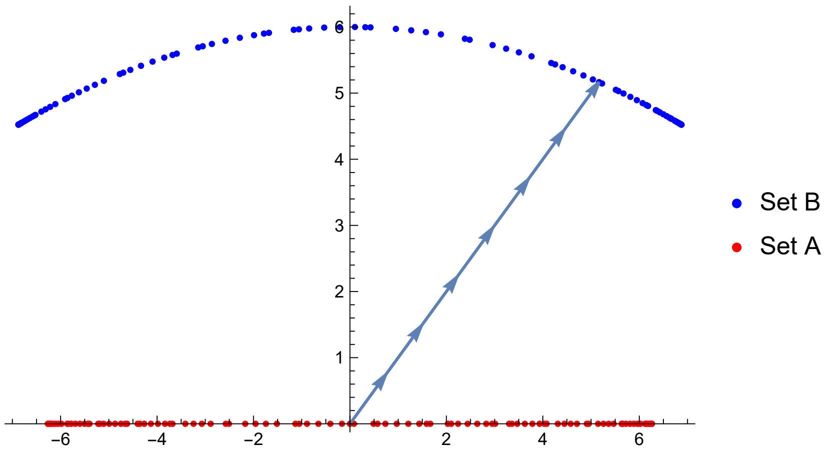
Out[∘]=



In[∘]:=
```
intersecpts = ptlist;
For[xaxispts = {}; i=0, i<Length[intersecpts], i++; If[intersecpts⟦i, 2⟧==0, AppendTo[xaxispts
curvepts = Sort[DeleteCases[intersecpts, Alternatives @@ xaxispts]];
xaxispts = Sort[xaxispts];
sepplot = ListPlot[{curvepts, xaxispts}, PlotStyle→{Blue, Red},PlotLegends→{"Set B", "Set A"}
```

Out[∘]=


In[∘]:=
```
z1 = ListLinePlot[{{0,0},{5.166010488516726`,5.166010488516724`}},Axes→True, AxesOrigin→{0,0}
  MeshShading → {Arrowheads[Small]}, DataRange → {0, 4 Pi}] /. Line → Arrow;
Show[sepplot,z1 ]
```

Out[∘]=


In[∘]:=
```
Clear[i, j, k]
reflecpathls = {};
For[i=0, i<Length[xaxispts], i++;
    For[j=0, j<Length[curvepts], j++;
        For[k=0, k<Length[curvepts], k++;
            If[curvepts⟦j⟧==curvepts⟦k⟧, Continue[]];
            If[(curvepts⟦k⟧⟦2⟧-xaxispts⟦i⟧⟦2⟧)/(curvepts⟦k⟧⟦1⟧-xaxispts⟦i⟧⟦1⟧) == -(curvepts
                reflecpathls = AppendTo[reflecpathls, {curvepts⟦k⟧, xaxispts⟦i⟧, curvepts⟦j⟧}
reflecpathls;
```

Paths with common point of beginning

In[∘]:=
```
samepts = {};
For[i=0, i<Length[reflecpathls], i++;
    For[j =0, j<Length[reflecpathls], j++;
        For[z = 0, z<Length[reflecpathls], z++;
            If[i≠j≠z && reflecpathls⟦i⟧⟦1⟧== reflecpathls⟦j⟧⟦1⟧==reflecpathls⟦z⟧⟦1⟧,samepts =
Length[reflecpathls]
Length[samepts]
```

Out[∘]= 200

Out[∘]= 0

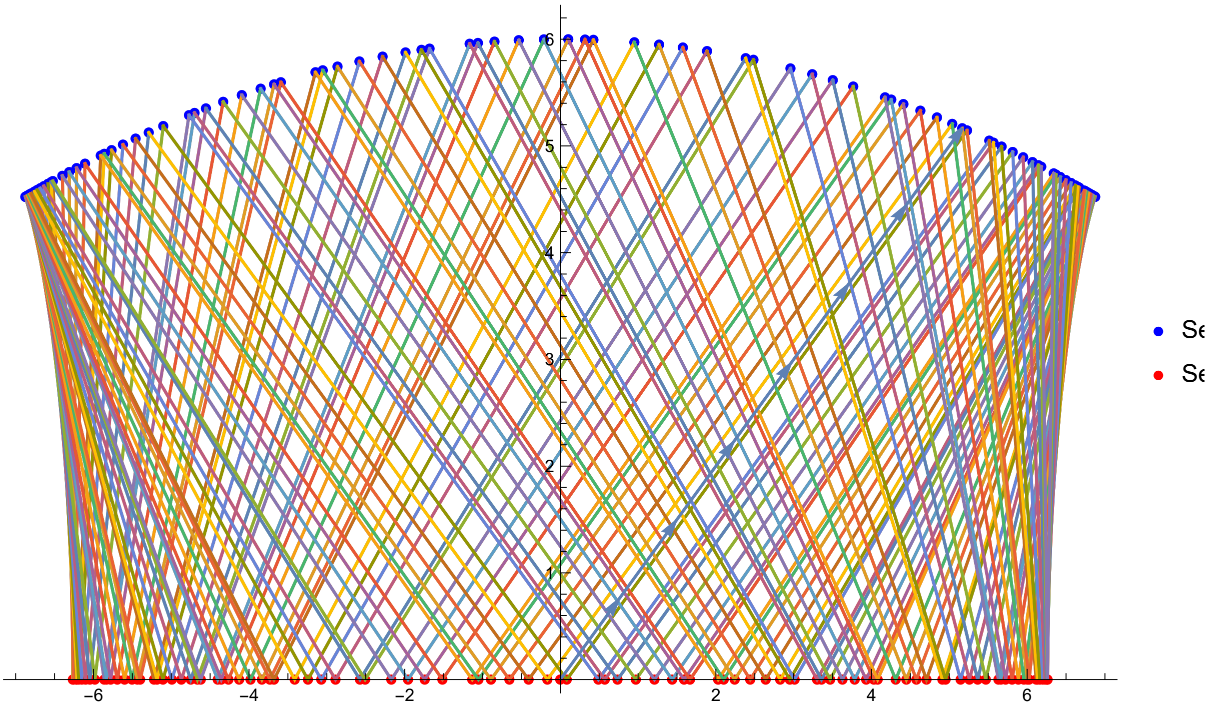
Zero common paths

In[∘]:=
```
reversepathls = {};
For[i=0, i<Length[reflecpathls], i++;
    reversepathls = AppendTo[reversepathls, Reverse[reflecpathls⟦i⟧]]]
reversepathls;
```

In[∘]:=
```
legalpaths = Join[reflecpathls , reversepathls];
```

In[∘]:=
```
Show[sepplot, ListLinePlot[reflecpathls], z1]
```

Out[∘]=



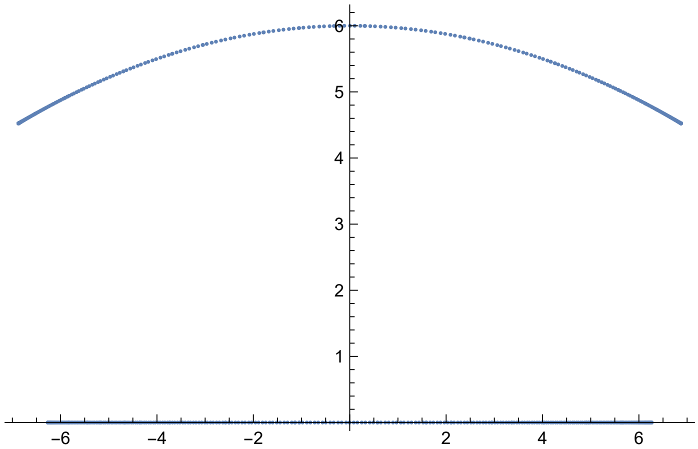
300 reflection pairs after first incident

In[∘]:=
```
eqnlist = {};
ptlist = {{0,0},{5.166010488516726`,5.166010488516724`}};
Nest[mergefunction,{{5.166010488516726`,5.166010488516724`}, 1} , 300]
eqnlist;
ptlist;
```

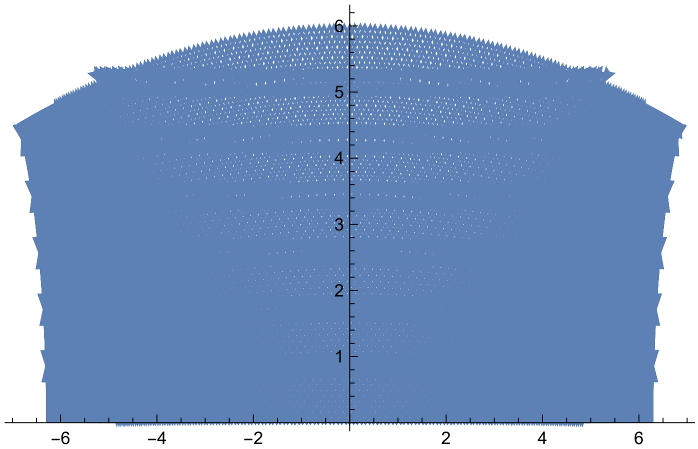
Out[∘]= {{6.84374, 4.53635}, 3.7333}

In[ ]:=
```
z1 = ListPlot[ptlist]
z2 = ListLinePlot[ptlist, Axes→True, AxesOrigin→{0,0}, MeshFunctions → {#2 &}, Mesh → 6, Mesh
  MeshShading → {Arrowheads[Small]}, DataRange → {0, 4 Pi}] /. Line → Arrow
```

Out[ ]=

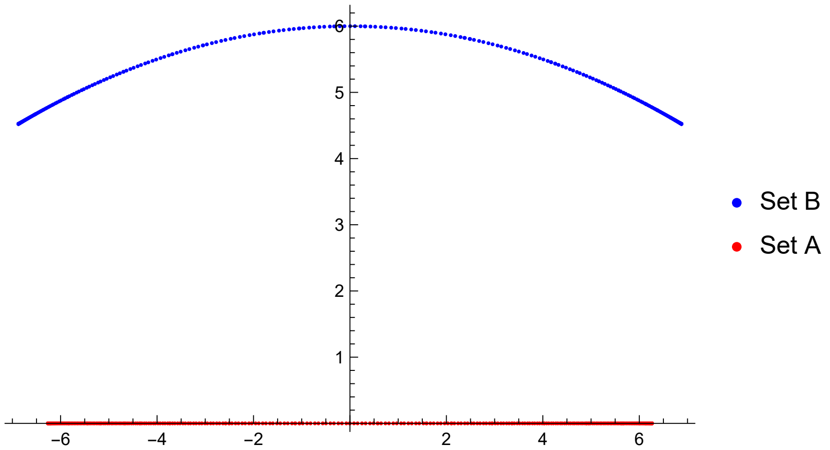


Out[ ]=



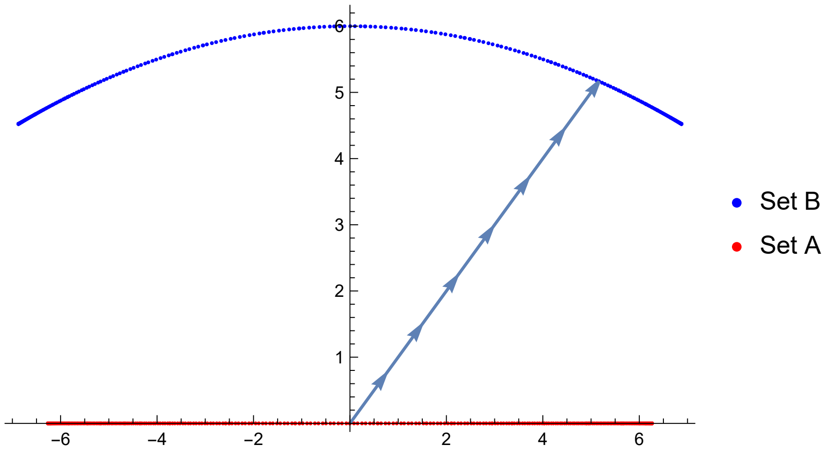
In[ ]:=
```
intersecpts = ptlist;
For[xaxispts = {}; i=0, i<Length[intersecpts], i++; If[intersecpts[[i, 2]]==0, AppendTo[xaxispts
curvepts = Sort[DeleteCases[intersecpts, Alternatives @@ xaxispts]];
xaxispts = Sort[xaxispts];
sepplot = ListPlot[{curvepts, xaxispts}, PlotStyle→{Blue, Red},PlotLegends→{"Set B", "Set A"}
```

Out[ ]=



In[]:=
```
z1 = ListLinePlot[{{0,0},{5.166010488516726`,5.166010488516724`}},Axes→True, AxesOrigin→{0,0}
  MeshShading → {Arrowheads[Small]}, DataRange → {0, 4 Pi}] /. Line → Arrow;
Show[sepplot,z1 ]
```

Out[]=



In[]:=
```
Clear[i, j, k]
reflecpathls = {};
For[i=0, i<Length[xaxispts], i++;
    For[j=0, j<Length[curvepts], j++;
        For[k=0, k<Length[curvepts], k++;
            If[curvepts⟦j⟧==curvepts⟦k⟧, Continue[]];
            If[(curvepts⟦k⟧⟦2⟧-xaxispts⟦i⟧⟦2⟧)/(curvepts⟦k⟧⟦1⟧-xaxispts⟦i⟧⟦1⟧) == -(curvepts
                reflecpathls = AppendTo[reflecpathls, {curvepts⟦k⟧, xaxispts⟦i⟧, curvepts⟦j⟧}
reflecpathls;
```

Path with common point of beginning

In[]:=
```
samepts = {};
For[i=0, i<Length[reflecpathls], i++;
    For[j =0, j<Length[reflecpathls], j++;
        For[z = 0, z<Length[reflecpathls], z++;
            If[i≠j≠z && reflecpathls⟦i⟧⟦1⟧== reflecpathls⟦j⟧⟦1⟧==reflecpathls⟦z⟧⟦1⟧,samepts =
Length[reflecpathls]
Length[samepts]
```

Out[]= $Aborted

Out[]= 596

Out[]= 0

In[]:=
```
reversepathls = {};
For[i=0, i<Length[reflecpathls], i++;
    reversepathls = AppendTo[reversepathls, Reverse[reflecpathls⟦i⟧]]]
reversepathls;
```

In[]:=
```
legalpaths = Join[reflecpathls , reversepathls];
```

In[ ]:=

```
Show[sepplot, ListLinePlot[reflecpathls], z1]
```

Out[ ]=