In[1]:=
```
SetDirectory["~/KappaLib/"];
<< kappaLib-1.1.m
<< helper.m
```

```
KappaLib v1.1

Loading helper.m..
```

In[4]:=
```
(* top view *)
ss = {ViewPoint → {3.2913687400769907`, -0.7841113452279369`, 0.045400607141734245`},
   ViewVertical → {0.6931706303592274`, 0.3216319509377258`, -0.6450328405154142`}};
```

In[5]:=
```
(* paper view *)
ss = {ViewPoint → {1.5006100736901282`, -0.10772546054752967`, 3.030934613595207`},
   ViewVertical → {0.9850776176603016`, -0.01090749309442297`, 0.1717647046954054`}};
```

In[6]:=
```
vp = (ViewPoint //. ss);
vv = (ViewVertical //. ss);
```

Define kappa

In[8]:=
```
vars = {x0, x1, x2, x3};
sub = {a2 → a1, b2 → b1};
```

$$\text{kappa} = \text{emMatrixToKappa}\left[\begin{pmatrix} a1 & -b1 & 0 & 0 & 0 & 0 \\ b1 & a1 & 0 & 0 & 0 & 0 \\ 0 & 0 & a2 & 0 & 0 & -b2 \\ 0 & 1 & 0 & a1 & b1 & 0 \\ 1 & 0 & 0 & -b1 & a1 & 0 \\ 0 & 0 & b2 & 0 & 0 & a2 \end{pmatrix}\right];$$

```
kappa = kappa /. sub;
```

$$\text{subW} = \left\{\text{ww} \text{ -> } \sqrt{1 + 4\, \text{b1}^2}\right\};$$

```
(* Jacobian taken from Transform_II.nb *)
```

$$\text{trans} = \begin{pmatrix} \frac{b1}{ww} & 0 & 0 & \frac{-1+ww}{2\,ww} \\ -\frac{b1}{ww} & 0 & 0 & \frac{1+ww}{2\,ww} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{pmatrix};$$

```
trans//MatrixForm

FullSimplify[Det[trans]]

kappaTrans = emCoordinateChange[kappa, trans];

frTrans = FullSimplify[emKappaToFresnel[kappaTrans, vars] /. sub];
```

Out[14]//MatrixForm=

$$\begin{pmatrix} \frac{b1}{ww} & 0 & 0 & \frac{-1+ww}{2\,ww} \\ -\frac{b1}{ww} & 0 & 0 & \frac{1+ww}{2\,ww} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{pmatrix}$$

Out[15]=

$$\frac{b1}{ww}$$

- **Extract factors**

In[18]:=
```
frTrans ww^2 / b1^2 /. subW
```

Out[18]=
$$\left(b1\ (x0 - x1)\ (x0 + x1) - \sqrt{1 + 4\, b1^2}\ \left(x2^2 + x3^2\right)\right)$$
$$\left(b1\ (x0 - x1)\ \left(\left(-2 + \sqrt{1 + 4\, b1^2}\right) x0 + \left(2 + \sqrt{1 + 4\, b1^2}\right) x1\right) - \left(1 + 4\, b1^2\right) \left(x2^2 + x3^2\right)\right)$$

In[19]:=
```
g1[x0_, x1_, x2_, x3_, b1_] := (b1 (x0 - x1) (x0 + x1) - Sqrt[1 + 4 b1^2] (x2^2 + x3^2))

g2[x0_, x1_, x2_, x3_, b1_] :=
  (b1 (x0 - x1) ((-2 + Sqrt[1 + 4 b1^2]) x0 + (2 + Sqrt[1 + 4 b1^2]) x1) - (1 + 4 b1^2) (x2^2 + x3^2))
```

- **Fresnel polynomial depends on x0, x1^2+x2^2, x3**

In[21]:=
```
draw[d3_, xx_, yy_, zz_, topMesh_] := Module[
   {p1, p2, grayLevel},
   grayLevel = 0.2; (* 0 = black *)
   p1 = ContourPlot3D[
     {g1[x0, x1, 0, x3, d3] == 0},
     {x0, -xx, xx}, {x1, -yy, yy}, {x3, -zz, zz},
     Axes → False,
     Boxed → False,
     Lighting → "Neutral",
     ViewPoint → vp,
     Mesh → {5},
     PlotPoints → 40,
     MeshFunctions → { #1 &, #2 &, #3 &},

     ColorFunctionScaling → 0.1,
     MeshStyle → {Directive[GrayLevel[grayLevel], Opacity[0.5]]},
     ViewVertical → vv];
   p2 = ContourPlot3D[
     {g2[x0, x1, 0, x3, d3] == 0},
     {x0, 0, xx}, {x1, -yy, yy}, {x3, -zz, zz},
     MeshStyle → {Directive[GrayLevel[grayLevel], Opacity[0.5]]},
     Axes → False,
     Boxed → False,
     MeshFunctions → { #1 &, #3 &},
     Mesh → topMesh,
     MaxRecursion → 10,
     Lighting → "Neutral",
     PlotPoints → 40,
     ViewPoint → vp,
     ViewVertical → vv];
   Show[{p1, p2}, PlotRange → {All, All, All}]
  ];
```

In[22]:=
```
mesh1 = {2, {0, 0.144, -0.143}, 1};
mesh2 = {2, {0, -0.215, +0.215}, 1};
mesh3 = {2, {0, -0.234, -0.468}, 1};
```

In[25]:=
```
(* testing :
  plot1=draw[0.2,1,1,1,mesh1]
*)
```

In[26]:=
```
plot1 = draw[0.2, 1, 1, 1, mesh1];
plot2 = draw[0.8, 1, 1, 1, mesh2];
plot3 = draw[8, 1, 1, 1, mesh3];
grid = Show[GraphicsGrid[{{plot1, plot2, plot3}}]];
```

In[30]:=
```
Export["temp.pdf", grid, ImageResolution → 2000]
```

Out[30]=
```
temp.pdf
```

- **Note running the above notebook is relatively time consuming. Images have been removed from the above due their size.**