In[1]:=
```
SetDirectory["~/KappaLib/"];
<< kappaLib-1.2.m
<< helper.m
```

```
Loading KappaLib v1.2
```

```
Loading helper.m..
```

In[4]:=
```
coords = {xi0, xi1, xi2, xi3};
```

Notebook to find algebraically decomposable medium tensors with beta^2-alpha gamma = 0.

A general feature of such medium tensors seems to be that their Fresnel surfaces always have a linear factor.

- **Define general kappa and put X variables to zero to simplify the computations.**

  Change X for speed/generality

In[5]:=
```
X = 10;
kappa = emGeneralKappa["k"];
For[ii = 1, ii ≤ X, ii++,
  vKappa = Variables[kappa];
  v = Random[Integer, {1, Length[vKappa]}];
  kappa = kappa /. vKappa[[v]] → 0;
 ];
emKappaToMatrix[kappa] // MatrixForm
```

Out[8]//MatrixForm=

$$\begin{pmatrix} k11 & k12 & k13 & k14 & k15 & 0 \\ 0 & 0 & 0 & k24 & k25 & 0 \\ k31 & k32 & k33 & 0 & 0 & k36 \\ k41 & k42 & k43 & k44 & k45 & k46 \\ 0 & k52 & k53 & k54 & k55 & k56 \\ k61 & 0 & k63 & k64 & k65 & 0 \end{pmatrix}$$

- **Formulate constraints:**

  *** kappa is algebraically decomposable**
  *** beta^2 -alpha = 0**
  *** det(kappa)=1**
  *** trace(kappa)=0**

In[9]:=
```
Abivector = emMatrix["a", 4, Structure → "AntiSymmetric"];
Bbivector = emMatrix["b", 4, Structure → "AntiSymmetric"];

LHS = alpha emIdentityKappa[] +
   beta (kappa + emPoincare[kappa]) + emCompose[emPoincare[kappa], kappa];
RHS = emBiProduct[rho, Abivector, Bbivector] + emBiProduct[rho, Bbivector, Abivector];
eqs = Union[Simplify[Flatten[(LHS - RHS)]]];

eqs = Append[eqs, beta^2 - alpha];
eqs = Append[eqs, emDet[kappa] - 1];
eqs = Append[eqs, emTrace[kappa]];
```

- **Randomly assign some variables and solve the rest.**

Note: The below can take a long time.

In[167]:=
```
For[done = 0, done == 0, (* nop *),
 subs = {rho → 1};

 (* pick values for Y variables randomly from 0/1  *)
 (* Change Y for speed/generality *)
 For[ii = 1, ii ≤ 20, ii++,
  vars = Variables[eqs //. subs];
  v = Random[Integer, {1, Length[vars]}];
  v = vars[[v]];
  subs = Append[subs, v -> Random[Integer, {0, 1}]];
 ];
 vars = Variables[eqs //. subs];

 eqs0 = eqs //. subs;

 sol = TimeConstrained[Solve[toEqs[eqs0], Variables[eqs0]], 2];
 If[Length[sol] == 0,
  ,
  Print["Found " <> ToString[Length[sol]] <> " solutions."];
  done = 1;
 ];
]
subs
```

Solve::svars : Equations may not give solutions for all "solve" variables. ≫

```
Found 2 solutions.
```

Out[168]=
```
{rho → 1, k55 → 1, b24 → 0, k31 → 1, a14 → 0, k65 → 1,
 k36 → 0, k46 → 1, k54 → 1, a13 → 1, k61 → 0, k41 → 0, k15 → 0,
 k44 → 0, k42 → 0, b12 → 0, k64 → 1, k53 → 0, a24 → 0, k52 → 1, k45 → 1}
```

In[186]:=
```
index = 2;

(* Compute alpha,beta,A,B,kappa for solution *)
kSubs = Join[subs, sol[[index]]];
kSubs = Append[kSubs, kx45 → 1];
kSubs = Append[kSubs, kx64 → 1 / 2];

a0 = alpha //. kSubs;
b0 = beta //. kSubs;
A0 = Abivector //. kSubs;
B0 = Bbivector //. kSubs;
k0 = Simplify[kappa //. kSubs];

(* output *)
Variables[k0]
a0 // MatrixForm
b0 // MatrixForm
A0 // MatrixForm
B0 // MatrixForm
emKappaToMatrix[k0] // MatrixForm
```

Out[195]= {k24, k63}

Out[196]//MatrixForm=
$$1$$

Out[197]//MatrixForm=
$$1$$

Out[198]//MatrixForm=
$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & k24 \\ 0 & 0 & -k24 & 0 \end{pmatrix}$$

Out[199]//MatrixForm=
$$\begin{pmatrix} 0 & 0 & \frac{1}{2} & -k63 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & -\frac{1}{2} \\ k63 & 0 & \frac{1}{2} & 0 \end{pmatrix}$$

Out[200]//MatrixForm=
$$\begin{pmatrix} -2 & -4-\frac{1}{k24\,k63} & -2 & -2\,k24 & 0 & 0 \\ 0 & 0 & 0 & k24 & 0 & 0 \\ 1 & 2+\frac{1}{k24\,k63} & 1 & 0 & 0 & 0 \\ 0 & 0 & k63 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 2 \\ 0 & 0 & k63 & 1 & 1 & 0 \end{pmatrix}$$

In[201]:=
```
fr = FullSimplify[emKappaToFresnel[k0, coords]]
```

Out[201]=
$$-\frac{1}{k63}(xi1+2\,xi2+xi3)\Big(-xi2\,(xi2-xi3)\,(xi1+2\,xi2+xi3)+k63^2\,xi0\,(xi1-2\,xi2+3\,xi3)\,(xi0-k24\,xi3)+ k63\,\Big(xi0\,\big(-xi1^2-2\,(-2+k24)\,xi2^2+xi1\,(xi2-3\,xi3)+k24\,xi2\,xi3-2\,xi3^2\big)- k24\,(xi2-xi3)\,(xi1+2\,xi2+xi3)\,(xi1+2\,(xi2+xi3))\Big)\Big)$$

- **Does factor!**
- **Write equations that should be satisfied for D**

In[202]:=
```
(* contract kappa with a bivector from the left *)
contract[biv_, kappa_] := Table[
  1 / 2 Sum[
    biv[[i]][[j]] emReadNormal[kappa, a, b, i, j]
    ,
    {i, 1, 4}, {j, 1, 4}
   ]
  ,
  {a, 1, 4}, {b, 1, 4}
 ]
```

- **Solve D (when possible) such that**

  D (kappa + beta Id) = 1/2 trace (rho bar(D) otimes D) A + B.

In[203]:=
```
(* contract kappa with a bivector from the left *)
contract[biv_, kappa_] := Table[
  1 / 2 Sum[
    biv[[i]][[j]] emReadNormal[kappa, a, b, i, j]
    ,
    {i, 1, 4}, {j, 1, 4}
   ]
  ,
  {a, 1, 4}, {b, 1, 4}
 ]

Dbivector = emMatrix["d", 4, Structure → "AntiSymmetric"];
dLHS  =  contract[Dbivector, k0 + b0 emIdentityKappa[]];
dRHS  =  1 / 2 emTrace[emBiProduct[rho, Dbivector, Dbivector]] A0 + B0;

deqs = Union[Flatten[Simplify[dLHS - dRHS]]];
dsol = Solve[toEqs[deqs], Variables[deqs]];
```

Solve::svars : Equations may not give solutions for all "solve" variables. ≫

In[209]:=
```
Dbivector //. dsol[[1]] // MatrixForm
```

Out[209]//MatrixForm=

$$\begin{pmatrix} 0 & d12 & 2\,d12 & d12 \\ -d12 & 0 & d23 & -\frac{1}{2} \\ -2\,d12 & -d23 & 0 & -1-d23 \\ -d12 & \frac{1}{2} & 1+d23 & 0 \end{pmatrix}$$