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

```
Loading KappaLib v1.2
```

- **Verify that the Fresnel polynomial for the QDCM medium class in**

  Lindell, Bergamin, Favaro : Decomposable medium conditions in four - dimensional representation

  factorises into the product of two quadratic forms.

- **Define most general QDCM medium**

In[3]:=
```
QQ = emMatrix["q", 4, Structure → "General"];
CC = emMatrix["a", 4, Structure → "AntiSymmetric"];
DD = emMatrix["b", 4, Structure → "AntiSymmetric"];
(* rho1 = scalar density of weight 1 *)
(* rho2 = scalar density of weight 1 *)
(* C1 = constant *)
```

In[6]:=
```
kappa = C1 emIdentityKappa[] + emQMedium[rho1, QQ] + emBiProduct[rho2, CC, DD];
```

- **Compute Fresnel polynomial**

In[7]:=
```
vars = {x0, x1, x2, x3};
fresnel = emKappaToFresnel[kappa, vars];
```

- **Check that the Fresnel polynomial factorises into a product if two quadratic forms**

In[9]:=
```
(* define adjugate of Q *)
adjQ = Table[
            (-1)^(i + j) Det[Drop[QQ, {i, i}, {j, j}]],
            {j, 1, 4}, {i, 1, 4}
            ];

(* check *)
Simplify[adjQ / Det[QQ] - Inverse[QQ]]
```

Out[10]=
```
{{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}}
```

In[11]:=
```
factor1 = vars.(rho1 Det[QQ] QQ + 2 rho2 CC.adjQ.Transpose[DD]).vars;
factor2 = vars.QQ.vars;
fresnelExp = rho1^2 factor1 factor2;

Simplify[fresnel - fresnelExp]
```

Out[14]=
```
0
```

- **Note 1: The above decomposition is valid also when Q is singular.**

  Note 2: The factor 2 in front of rho2 does not appear in the original paper.