In [1]:

```
from sympy import Matrix, solve, Symbol, re, sqrt, Eq, MatrixSymbol as MSymbol
from IPython.display import display
R = Symbol("R", positive=True, real=True)
```

In [2]:

```
Yt = Matrix([
    [2, 1],
    [1, 2]
]).inv()/R*2
display(Eq(MSymbol("Y_t", 2, 2), Yt))
```

$$Y_t = \begin{bmatrix} \frac{4}{3R} & -\frac{2}{3R} \\ -\frac{2}{3R} & \frac{4}{3R} \end{bmatrix}$$

In [3]:

```
gamma = Matrix([
    [Yt[1,1],  1],
    [Yt.det(), Yt[0,0]]
])*-1/(Yt[1,0])
display(Eq(MSymbol("gamma", 2, 2), gamma))
```

$$\gamma = \begin{bmatrix} 2 & \frac{3R}{2} \\ \frac{2}{R} & 2 \end{bmatrix}$$

In [4]:

```
A, B, C, D = gamma
display((A, B, C, D))
```

(2, 3*R/2, 2/R, 2)

In [5]:

```
Zi2 = Symbol("Z_i2")
Zi2 = solve(C*Zi2**2+(A-D)*Zi2-B, Zi2)
display(Eq(Symbol("Z_i2"), *filter(lambda x: re(x) > 0, Zi2)))
```

$$Z_{i2} = \frac{\sqrt{3}R}{2}$$

In [6]:

```
Z01 = sqrt((A*B)/(C*D))
Z02 = sqrt((B*D)/(A*C))
display(Eq(Symbol("Z_01") ,Z01), Eq(Symbol("Z_02") ,Z02))
```

$$Z_{01} = \frac{\sqrt{3}R}{2}$$

$$Z_{02} = \frac{\sqrt{3}R}{2}$$