Invertiereren einer (3x3)- Matrix mit reelen Zahlen

Du kannst dieses Code-Snippet in einem Jupyter Notebook ausführen. Stelle sicher, dass du die benötigte NumPy-Bibliothek installiert hast. Dieses Notebook erstellt eine symmetrische 3x3-Matrix A, berechnet ihre Determinante und Adj. Matrix woraus sich die inverse Matrix von A ergibt.

Autor: Hermann Völlinger, DHBW Stuttgart; Date: 5.10.2023

In [1]:
```
# Importiere die NumPy-Bibliothek
import numpy as np

# Importiere das sympy-Modul
import sympy as sp

# Definiere die reellen Variablen als Symbole
a, b, c, d, e, f = sp.symbols('a b c d e f')

# Definiere die 3x3-Matrix A mit den Symbolen
A = sp.Matrix([[a, b, c],
              [b, d, e],
              [c, e, f]])

# Berechne die Determinante von A und klammere sie aus
det_A = sp.det(A)

# Berechne die 6 Kofaktoren
C_11 = (A[1, 1] * A[2, 2] - A[2, 1] * A[1, 2])
C_12 = -(A[1, 0] * A[2, 2] - A[2, 0] * A[1, 2])
C_13 = (A[1, 0] * A[2, 1] - A[2, 0] * A[1, 1])
C_22 = (A[0, 0] * A[2, 2] - A[2, 0] * A[0, 2])
C_23 = -(A[0, 0] * A[2, 1] - A[2, 0] * A[0, 1])
C_33 = (A[0, 0] * A[1, 1] - A[1, 0] * A[0, 1])

# Ausgabe der Determinante und der Kofaktoren als einzelne Zeilen
print("Die Determinante von A ist:", det_A)
print("Die Kofaktoren sind:")
print("C_11 =", C_11)
print("C_12 =", C_12)
print("C_13 =", C_13)
print("C_22 =", C_22)
print("C_23 =", C_23)
print("C_33 =", C_33)

print("Inverse(A) = 1/det(A)*Adj(A), wobei die Adj(A) symmetrisch ist und sich aus 6 Kofaktoren zusammensetzt")
print("Adj(A) setzt sich zusammen aus C_11, -C_12, C_13, C_22, -C_23 und C_33, wobei C_12=C_21 und C_13=C_31")
```

```
Die Determinante von A ist: a*d*f - a*e**2 - b**2*f + 2*b*c*e - c**2*d
Die Kofaktoren sind:
C_11 = d*f - e**2
C_12 = -b*f + c*e
C_13 = b*e - c*d
C_22 = a*f - c**2
C_23 = -a*e + b*c
C_33 = a*d - b**2
Inverse(A) = 1/det(A)*Adj(A), wobei die Adj(A) symmetrisch ist und sich aus 6 Kofaktoren zusammensetzt
Adj(A) setzt sich zusammen aus C_11, -C_12, C_13, C_22, -C_23 und C_33, wobei C_12=C_21 und C_13=C_31
```