

Programm zum Lösen von linearen Gleichungssystemen

```
def zeilen_zusammenfassen(zeile1, zeile2):
    # Das geeignete Vielfache einer Zeile wird zur 2. Zeile addiert.
    # Die an der ersten Stelle stehende Null entfällt durch [1:].
    c = - zeile2[0]/zeile1[0]
    return [c*a+b for a,b in zip(zeile1, zeile2)][1:]

def variablen_eliminieren(lgs):
    # Nach dem Gauss-Verfahren wird das LGS schrittweise jeweils um eine Variable verringert.
    # Zum Schluss entsteht die Gleichung ax=b, deren Lösung b/a zurückgeliefert wird.
    for i in range(len(lgs)):
        erste_zeile = lgs[0]
        lgs_neu = []
        for zeile in lgs[1:]:
            zeile_neu =zeilen_zusammenfassen(erste_zeile, zeile)
            lgs_neu.append(zeile_neu)
        lgs=lgs_neu
    return lgs[0][1]/lgs[0][0]

def einsetzen(lgs, wert):
    # In das Gleichungssystem wird für die ganz rechts stehende Variable die Lösung ein-
    # gesetzt. Die Zahlen werden zusammengefasst. Die rechts stehenden Zahlen entfallen
    # durch[:-1], so dass ein Gleichungssystem mit einer Variablen weniger entsteht.
    return [zeile[:len(lgs)-1] + zeile[len(lgs):-1] + [zeile[-1] - zeile[len(lgs)-1]*wert]
            for zeile in lgs][:-1]

def loese(lgs):
    loesungen = []
    for i in range(len(lgs)):
        z = variablen_eliminieren(lgs)
        loesungen = [z]+ loesungen
        lgs = einsetzen(lgs, z)
    return loesungen

lgs =[[2, 3,-1,11],
      [1,-1, 2, 3],
      [3,-2, 3, 8]]

print(loese(lgs))
```

Bei diesem Algorithmus wird jeweils nach Einsetzen einer Lösung das Gauss-Verfahren erneut angewandt, siehe loese(lgs).
Im Algorithmus auf der nächsten Seite ist das nicht mehr der Fall.

Rekursives Programm zum Lösen von linearen Gleichungssystemen

```
# Roolfs 2019

def lgs_ordnen(lgs):          # entfällt, falls die Koeffizienten ungleich null bleiben
    for zeile in lgs:
        if zeile[0] != 0:
            lgs.remove(zeile)
            lgs.insert(0, zeile)
    return lgs

def zeilen_zusammenfassen(zeile1, zeile2):
    c = - zeile2[0]/zeile1[0]
    return [c*a+b for a,b in zip(zeile1, zeile2)][1:]

def einsetzen(zeile, wert):
    return zeile[:len(zeile)-2] + zeile[len(zeile):-1] + [zeile[-1]-zeile[-2]*wert]

def variablen_eliminiieren(lgs):
    global L
    L=[]
    lgs=lgs_ordnen(lgs)
    if len(lgs)==1:
        L.append(lgs[0][1]/lgs[0][0])
    else:
        erste_zeile = lgs[0]
        lgs_neu = []
        for zeile in lgs[1:]:
            zeile_neu =zeilen_zusammenfassen(erste_zeile, zeile)
            lgs_neu.append(zeile_neu)

        variablen_eliminiieren(lgs_neu)
        for z in L:
            lgs[0] = einsetzen(lgs[0], z)
        L.append(lgs[0][1]/lgs[0][0])
    return L

lgs = [[3, 4,-5, 6, 39],
        [6, 5,-6, 5, 43],
        [9,-4, 2, 3, 6],
        [0, 2,-3, 1, 13]]

L=variablen_eliminiieren(lgs)
print(L[::-1])          # Reihenfolge umkehren
```

Erläuterung

```
[3, 4, -5, 6, 39]
[6, 5, -6, 5, 43]
[9, -4, 2, 3, 6]
[0, 2, -3, 1, 13]
```

```
[7.6.., -7.3.., 3.0, 39.0]
[5.3.., -5.6.., 5.0, 37.0]
[2.0 , -3.0 , 1.0, 13.0]
```

```
[2.3 .., 2.3.., 2.3..]
[4.16.., -0.83.., -10.83..]
```

.. bedeutet Periode

```
[2.8, 8.39..]
```

hiermit ergibt sich die 1. Lösung 2.9..

```
[1.00, 2.00, -1.9.., 2.9..]
```

Lösungen

Das Gleichungssystem wird mit jedem rekursiven Aufruf von `variablen_eliminiere(lgs)` um eine Variable reduziert. Die Koeffizientenmatrizen sind dann gespeichert, so dass die ermittelten Lösungen in eine Gleichung (hier jeweils die Erste, `lgs[0]`) eingesetzt werden können, um eine weitere Lösung zu erhalten. Das Einsetzen verläuft von hinten nach vorne (`einsetzen()` steht im Programm hinter `variablen_eliminiere(lgs)`) und beginnt, nachdem alle Koeffizientenmatrizen vorliegen, mit $ax = b$ und der 1. Lösung b/a der ganz rechts stehenden Variablen.

Ergänzung

Zur Vermeidung der Division durch Null wird das LGS geordnet.
Die 1. Zeile beginnt dann mit einer Null.

```
def lgs_ordnen(lgs):  
    for zeile in lgs:  
        if zeile[0] != 0:  
            lgs.remove(zeile)  
            lgs.insert(0, zeile)  
    return lgs
```

alternativ

```
def lgs_ordnen(lgs):  
    return sorted(lgs, key = lambda a: a[0], reverse=True)
```