Exercício 3 - Componente eletrônico detalhado

Importando as bibliotecas

```
from sympy import *
import numpy as np
from scipy.optimize import fsolve
from numpy.linalg import solve

init_printing(pretty_print=true)
```

Inserindo os dados do problema

```
Tar = 21
Tp = 40
kco = 350
kce = 10
kv = 8
kaz = 0.03
ksi = 146
Lperna = 15/1000
Lpl = 30/100
Dperna = 1/1000
Q = 2
h = 80
Nperna = 8
```

Dimensões do problema

```
m1 = 6.075/100
m2 = 3.4/100
```

m3 = 0.8/100

m4 = 0.2/100

m5 = 0.8/100

m6 = 0.8/100

mchip =1.895/100

Definição das áreas

A1 = m1*mchip/2

A2 = m2*mchip/2

A3 = m3*mchip/2

A4 = m4*mchip/2

A5 = m5*mchip/2

A6 = m6*mchip/2

Aperna = (3.14*Dperna**2)/4

As = Ap = (m1 + m2)*mchip/2

Definição das distâncias

L12 = L21 = (m1/2) + (m2/2)

L34 = L43 = L56 = L65 = L78 = L87 = L12

L13 = L31 = L24 = L42 = (m3/2) + (m4/2)

L35 = L53 = L46 = L64 = (m4/2) + (m5/2)

L57 = L75 = L68 = L86 = (m5/2) + (m6/2)

L1j = L2j = m3/2

L3j = L4j = m4/2

L5j = L6j = m5/2

L7j = L8j = m6/2

L3i = L5i = m1/2

L4i = L6i = m2/2

L1s = L2s = L1j

L7p = L8p = L7j

Definição das condutâncias

G12 = G21 = kce*A3/L12

G1j = kce*A1/L1j

G3j = kv*A1/L3j

G13 = G31 = (G1j*G3j)/(G1j+G3j)

G2j = kce*A2/L2j

G4j = kaz*A2/L4j

G24 = G42 = (G2j*G4j)/(G2j+G4j)

G3i = kv*A4/L3i

G4i = kaz*A4/L4i

G34 = G43 = (G3i*G4i)/(G3i+G4i)

G5i = kv*A5/L5i

G6i = ksi*A5/L6i

G56 = G65 = (G5i*G6i)/(G5i+G6i)

G35 = G53 = kv *A1/L35

G4j = kaz*A2/L4j

G6j = ksi*A2/L6j

G46 = G64 = (G4j*G6j)/(G4j+G6j)

G5j = kv*A1/L5j

G7j = kce*A1/L7j

G57 = G75 = (G5j*G7j)/(G5j+G7j)

G78 = G87 = kce *A6/L78

G8j = kce*A2/L8j

G68 = G86 = (G6j*G8j)/(G6j+G8j)

G7p = kce *A1/L7p

G8p = kce *A2/L8p

G6p = kco *Aperna/(4*Lperna)

G1s = kce *A1/L1s

G2s = kce*A2/L2s

Gsar = h*As

Equações de energia para regime permanente (escrevendo na forma de Eq(), simbólica)

```
nos = Eq(G1s*(T1-Ts) + G2s*(T2-Ts) + Gsar*(Tar-Ts),0)

no1 = Eq(G12*(T2-T1) + G13*(T3-T1) + G1s*(Ts-T1),0)

no2 = Eq(G21*(T1-T2) + G24*(T4-T2) + G2s*(Ts-T2),0)

no3 = Eq(G31*(T1-T3) + G35*(T5-T3) + G34*(T4-T3),0)

no4 = Eq(G42*(T2-T4) + G43*(T3-T4) + G46*(T6-T4),0)

no5 = Eq(G53*(T3-T5) + G57*(T7-T5) + G56*(T6-T5),0)

no6 = Eq(G64*(T4-T6) + G65*(T5-T6) + G68*(T8-T6) + G6p*(Tp-T6) +
Q/4,0)

no7 = Eq(G75*(T5-T7) + G78*(T8-T7) + G7p*(Tp-T7),0)

no8 = Eq(G86*(T6-T8) + G87*(T7-T8) + G8p*(Tp-T8),0)
```

Verifivando como ficaram as equações

```
nos

no1

no2

no3

no4

no5

no6

no7
```

no8

Resolvendo o sistema de equações

```
def Temp1(z):
  T1 = z[0]
  T2 = z[1]
  T3 = z[2]
  T4 = z[3]
  T5 = z[4]
  T6 = z[5]
  T7 = z[6]
  T8 = z[7]
  Ts = z[8]

  F = np.empty((9))
  F[0] = G12*(T2-T1) + G13*(T3-T1) + G1s*(Ts-T1)
  F[1] = G21*(T1-T2) + G24*(T4-T2) + G2s*(Ts-T2)
  F[2] = G31*(T1-T3) + G35*(T5-T3) + G34*(T4-T3)
  F[3] = G42*(T2-T4) + G43*(T3-T4) + G46*(T6-T4)
  F[4] = G53*(T3-T5) + G57*(T7-T5) + G56*(T6-T5)
  F[5] = G64*(T4-T6) + G65*(T5-T6) + G68*(T8-T6) + G6p*(Tp-T6) + Q/4
  F[6] = G75*(T5-T7) + G78*(T8-T7) + G7p*(Tp-T7)
  F[7] = G86*(T6-T8) + G87*(T7-T8) + G8p*(Tp-T8)
  F[8] = G1s*(T1-Ts) + G2s*(T2-Ts) + Gsar*(Tar-Ts)

  return F

z0 = np.array([1,1,1,1,1,1,1,1,1])
z = fsolve(Temp1,z0)
```

Imprimindo os resultados

```
print('T1 =', z[0], 'T2=', z[1],'T3=', z[2],'T4=', z[3],'T5=', z[4],'T6=',
z[5],'T7=', z[6],'T8=', z[7],'Ts=', z[8])
```