komplex_heatexchanger_network

September 13, 2023

[1]:
```
import sys
sys.path.append('..')
from exchanger.parts import *
from exchanger.stream import *
from exchanger.exchanger import*
from exchanger.network import *
import numpy as np
```

1 Komplexe Wärmeübertragernetzwerke

Im Folgenden wird gezeigt, wie die Zellenmethode mithilfe der Implementierung für komplexe Wärmeübertragernetzwerke angewendet werden kann. Weitere Details zur Methode lassen sich aus der Erklärung zur Zellenmethode (Zellenmethode Theorie) entnehmen.

1.1 Implementierung

Die folgenden Beispiele zeigen die Verwendung der Klasse `ExchangerNetwork`. Dazu sind auch Objekte der Klassen `Fluid` und `Flow` notwendig, die bereits in `simple_heatexchangers.ipynb` bzw. in der Klassendokumentation beschrieben wurden.

Die Klasse `ExchangerNetwork` dient ausschließlich zur Berechnung der Austrittstemperaturen der Apparate bzw. des Netzwerks mithilfe der Zellenmethode. Dazu müssen folgende Eigenschaften, wie in der Theoretischen Erklärung beschrieben, definiert werden:

- Die Funktionsmatrix `phi_matrix`, die die Betriebscharakteristik der Apparate beschreibt.
- Die Strukturmatrix `structure_matrix`, die die Struktur des Netzwerks beschreibt.
- Die Eingangsmatrix `input_matrix`, die die Fluidströmeintritte in das Netzwerk beschreibt.
- Die Ausgangsmatrix `output_matrix`, um die Ausgangstemperaturen aus dem Netzwerk zu berechnen.

Die Berechnung erfolgt mithilfe der auf die minimale und maximale Eintrittsfluidtemperatur normierten Temperaturen. Dazu muss der Instanz `ExchangerNetwork` eine Liste mit allen in das Netzwerk eintretenden Fluidströmen übergeben werden. Entsprechend dieser Listenordnung müssen auch die Input- und Outputmatrizen definiert werden.

Nun kann die Berechnung mithilfe der Zellenmethode erfolgen. Die resultierenden Apparateaustrittstemperaturen sind in der Eigenschaft `temperature_matrix` gespeichert, während diejenigen des Netzwerks in der Eigenschaft `temperature_outputs` vorliegen. Dabei wird jeweils ein Tupel erstens mit den dimensionslosen und zweitens mit den dimensionsbehafteten Temperaturen ausgegeben.

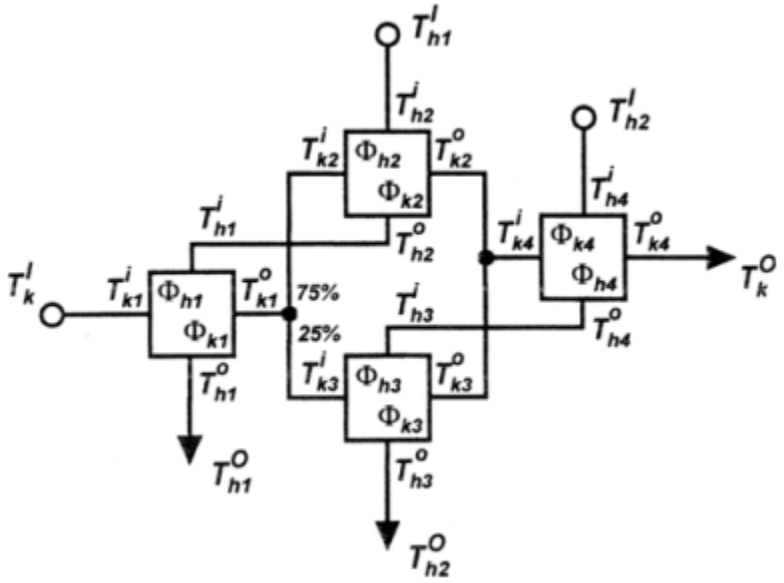
Die Netzwerkcharakteristik ist durch die Eigenschaft `network_characteristics` gegeben.

Es ist zu beachten, dass die Zellenmethode selbst keine Informationen zur Strömungsführung verwendet und die Betriebscharakteristik jedes einzelnen Apparats bekannt sein muss. Daher ist auch keine automatische Anpassung der Parameter möglich.

1.2 Wärmeübertragerschaltung mit Stromteilung

Von einem Wärmeübertragungsnetzwerk, wie in der folgenden Abbildung dargestellt, sind die Eintrittstemperaturen und die Betriebscharakteristiken der einzelnen Apparate bekannt.

- $T'_{h1} = 373K$
- $T'_{h2} = 405K$
- $T'_{k} = 293K$
- $\Phi_{h1} = 0.8$, $\Phi_{k1} = 0.6$
- $\Phi_{h2} = 0.6$, $\Phi_{k2} = 0.6$
- $\Phi_{h3} = 0.76$, $\Phi_{k3} = 0.76$
- $\Phi_{h4} = 0.64$, $\Phi_{k4} = 0.16$



[2]:
```
# defining Flows (mass flow irrelevant for following calculations)
flow_h1 = Flow(Fluid("Water", temperature=373), 1)
flow_h2 = Flow(Fluid("Water", temperature=405), 1)
flow_k = Flow(Fluid("Water", temperature=293), 1)
flows = [flow_h1, flow_h2, flow_k]
```

[3]:
```
network = ExchangerNetwork(flows)
network.phi_matrix = np.array([[0.2, 0., 0., 0., 0.8, 0., 0., 0.],
                               [0., 0.4, 0., 0., 0., 0.6, 0., 0.],
```

```
                                [0., 0., 0.24, 0., 0., 0., 0.76, 0.],
                                [0., 0., 0., 0.36, 0., 0., 0., 0.64],
                                [0.6, 0., 0., 0., 0.4, 0., 0., 0.],
                                [0., 0.6, 0., 0., 0., 0.4, 0., 0.],
                                [0., 0., 0.76, 0., 0., 0., 0.24, 0.],
                                [0., 0., 0., 0.16, 0., 0., 0., 0.84]])
network.structure_matrix = np.array([[0., 1., 0., 0., 0., 0., 0., 0.],
                                     [0., 0., 0., 0., 0., 0., 0., 0.],
                                     [0., 0., 0., 1., 0., 0., 0., 0.],
                                     [0., 0., 0., 0., 0., 0., 0., 0.],
                                     [0., 0., 0., 0., 0., 0., 0., 0.],
                                     [0., 0., 0., 0., 1, 0., 0., 0.],
                                     [0., 0., 0., 0., 1, 0., 0., 0.],
                                     [0., 0., 0., 0., 0., 0.75, 0.25, 0.]])
network.input_matrix = np.array([[0, 0, 0],
                                 [1, 0, 0],
                                 [0, 0, 0],
                                 [0, 1, 0],
                                 [0, 0, 1],
                                 [0, 0, 0],
                                 [0, 0, 0],
                                 [0, 0, 0]])
network.output_matrix = np.asarray([[1, 0, 0, 0, 0, 0, 0, 0],
                                    [0, 0, 1, 0, 0, 0, 0, 0],
                                    [0, 0, 0, 0, 0, 0, 0, 1]])
```

Die Austrittstemperaturen der einzelnen Apperate ergibt sich zu:

[4]:
```
network.temperature_matrix[1]
```

[4]:
```
array([[303.],
       [343.],
       [335.],
       [373.],
       [323.],
       [353.],
       [361.],
       [363.]])
```

und des Netzwerks:

[5]:
```
network.temperature_outputs[1]
```

[5]:
```
array([[303.],
       [335.],
       [363.]])
```

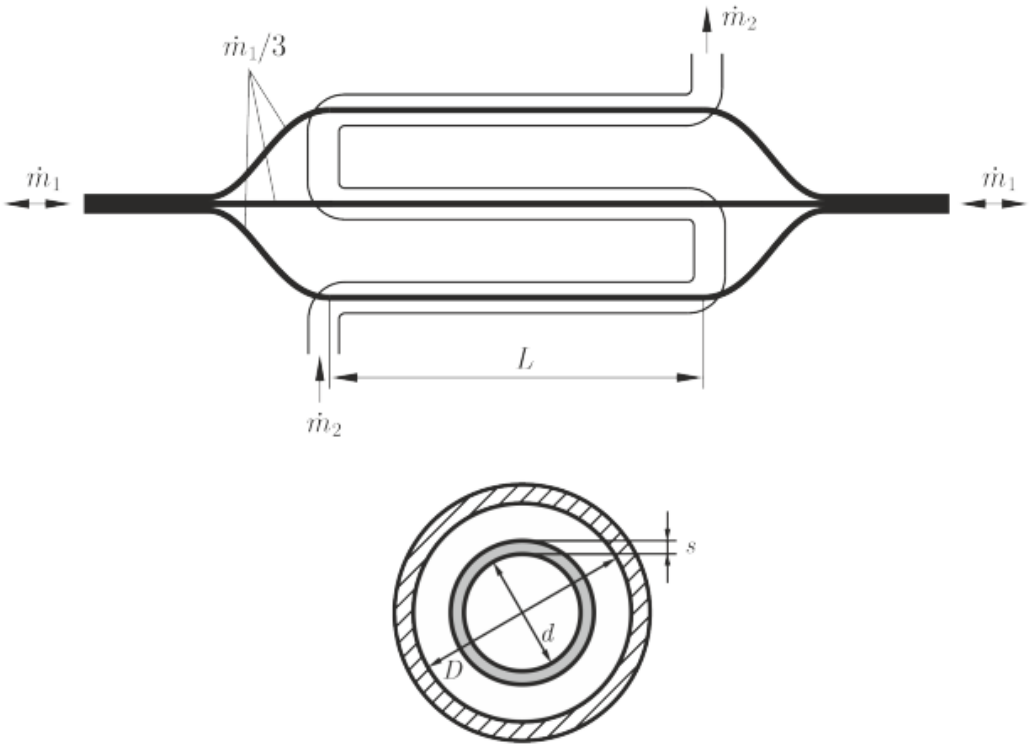
Die Schaltungscharakteristik wird ruch folgende Matrix beschrieben:

[6]:
```
network.network_characteristics
```

[6]:
```
array([[0.125     , 0.        , 0.875     ],
       [0.38729508, 0.09836066, 0.51434426],
       [0.55942623, 0.22540984, 0.21516393]])
```

1.3 Doppelrohr-Wärmeubertragers mit drei Strängen

Von einem Wärmeübertragungsnetzwerk, wie in der folgenden Abbildung dargestellt, sind die Eintrittstemperaturen und die Betriebscharakteristiken der einzelnen Apparate bekannt.



Der Fluidstrom 1 ist ein Thermofluid mit der Eintrittstemperatur von $120C$ und einem sich gleichmäßig aufteilenden Massenstrom von $m_1 = 0.18$ kg/s Der 2 Fluidstrom wird durch Wasser mit einer Eintrittstemperatur von $15C$ und einem Massenstrom von $m_2 = 0.33$ kg/s representiert.

Für die Betriebscharkateristk wder einzelnen Durchgänge werden folgende Werte angenommen

- $\Phi_{11} = 0.608$, $\Phi_{21} = 0.048$
- $\Phi_{12} = 0.597$, $\Phi_{22} = 0.047$
- $\Phi_{13} = 0.605$, $\Phi_{23} = 0.048$

[7]:
```
mass_flow_1 = 0.18
fluid_1 = Fluid("nHeptane", temperature=273.15 + 120)
flow_11 = Flow(fluid_1.clone(),mass_flow_1/3)
flow_12 = Flow(fluid_1.clone(),mass_flow_1/3)
flow_13 = Flow(fluid_1.clone(),mass_flow_1/3)
```

```
flow_2 = Flow(Fluid("Water", temperature=273.15 + 15), 0.33)
flows = [flow_11, flow_12, flow_13, flow_2]
```

[8]:
```
network = ExchangerNetwork(flows)
network.phi_matrix = np.array([[0.392, 0., 0., 0.608, 0., 0.],
                               [0., 0.403, 0., 0., 0.597, 0.],
                               [0., 0., 0.395, 0., 0., 0.605],
                               [0.048, 0., 0., 0.952, 0., 0.],
                               [0., 0.047, 0., 0., 0.953, 0.],
                               [0., 0., 0.048, 0., 0., 0.952]])
network.structure_matrix = np.array([[0., 0., 0., 0., 0., 0.],
                                     [0., 0., 0., 0., 0., 0.],
                                     [0., 0., 0., 0., 0., 0.],
                                     [0., 0., 0., 0., 0., 0.],
                                     [0., 0., 0., 1., 0., 0.],
                                     [0., 0., 0., 0., 1., 0.]])
network.input_matrix = np.array([[1, 0, 0, 0],
                                 [0, 1, 0, 0],
                                 [0, 0, 1, 0],
                                 [0, 0, 0, 1],
                                 [0, 0, 0, 0],
                                 [0, 0, 0, 0]])
network.output_matrix = np.asarray([[1, 0, 0, 0, 0, 0],
                                    [0, 1, 0, 0, 0, 0],
                                    [0, 0, 1, 0, 0, 0],
                                    [0, 0, 0, 0, 0, 1]])
```

Die einzelnen Austritstemperaturen sind somit:

[9]:
```
network.temperature_outputs[1]-273.15
```

[9]:
```
array([[56.16      ],
       [60.32388   ],
       [62.3665626 ],
       [29.31069024]])
```

Wird der Fluidstrom 1 wieder vermischt ergibt sich eine Temperatur von $59.62C$. und für den zweiten Fluidstrom $29.31C$

Alternativ könnte auch die Outputmatrix für das gesamte Netzwerk defniert werden und die Vermischung bereits im Netzwerkaustritt berücksichtigt werden.

[10]:
```
network.output_matrix = np.asarray([[1 / 3, 1 / 3, 1 / 3, 0, 0, 0],
                                    [0, 0, 0, 0, 0, 1]])
```

Somit ergeben sich wiederum die gleichen Fluidstromaustrittstemperaturen des gesamten Netzwerks

[11]:
```
network.temperature_outputs[1]-273.15
```

[11]:
```
array([[59.6168142 ],
       [29.31069024]])
```