

Wärmeübertragernetzwerke

complex_heatexchanger_networks.ipynb

März 2024

```
[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, dh. Netzwerke welche nicht üblichen Stromführungen wie z.B. in Rohrbündelwärmeübertragern entsprechen, angewendet werden können. Weitere Details zur Methode lassen sich aus der Erklärung zur Zellenmethode ([Zellenmethode Theorie](#)) entnehmen.

1.1 Implementierung der Zellenmethode

Die folgenden Beispiele zeigen die Verwendung der Klasse `ExchangerNetwork`. Dazu sind auch Objekte der Klassen `Fluid` und `Flow` notwendig, die bereits in [standard_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 Fluidstromeintritte 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 Austrittstemperaturen aus den Apparaten 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, bzw. in der Erklärung zur Zellenmethode auch Schaltungscharakteristik Φ^S genannt, ist durch die Eigenschaft `network_characteristics` gegeben. Damit kann der relative Einfluss der Eintrittstemperaturen auf die Austrittstemperaturen des gesamten Netzwerks direkt abgelesen werden.

Es ist zu beachten, dass in der aktuellen Implementierung die Information über die Strömungsführung nur in der Strukturmatrix **S** enthalten ist. Die objektorientierte Programmierung hilft nur bei der Initialisierung, durch den Aufbau der Funktionsmatrix Φ , welche nur noch die skalaren Werte der Betriebscharakteristik enthält, geht die anfänglich hinterlegte Information zu den einzelnen Fluiden (Temperatur, Dichte etc.) und Apparaten (Typ) verloren. Die Anpassung der Parameter nach der inertialen Berechnung kann daher nicht automatisch erfolgen, sondern wäre in der gegebenen Implementierung nur durch eine neue Initialisierung des gesamten Netzwerks möglich.

1.2 Wärmeübertragerschaltung mit Stromteilung

Folgendes Beispiel der Wärmeübertragerschaltung sowie die Parameter sind entnommen aus Kapitel 4.3 [1].

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

- $T_1^I = 373K$
- $T_2^I = 405K$
- $T_3^I = 293K$
- $\Phi_{11} = 0.8, \Phi_{21} = 0.6$
- $\Phi_{12} = 0.6, \Phi_{22} = 0.6$
- $\Phi_{13} = 0.76, \Phi_{23} = 0.76$
- $\Phi_{14} = 0.64, \Phi_{24} = 0.16$

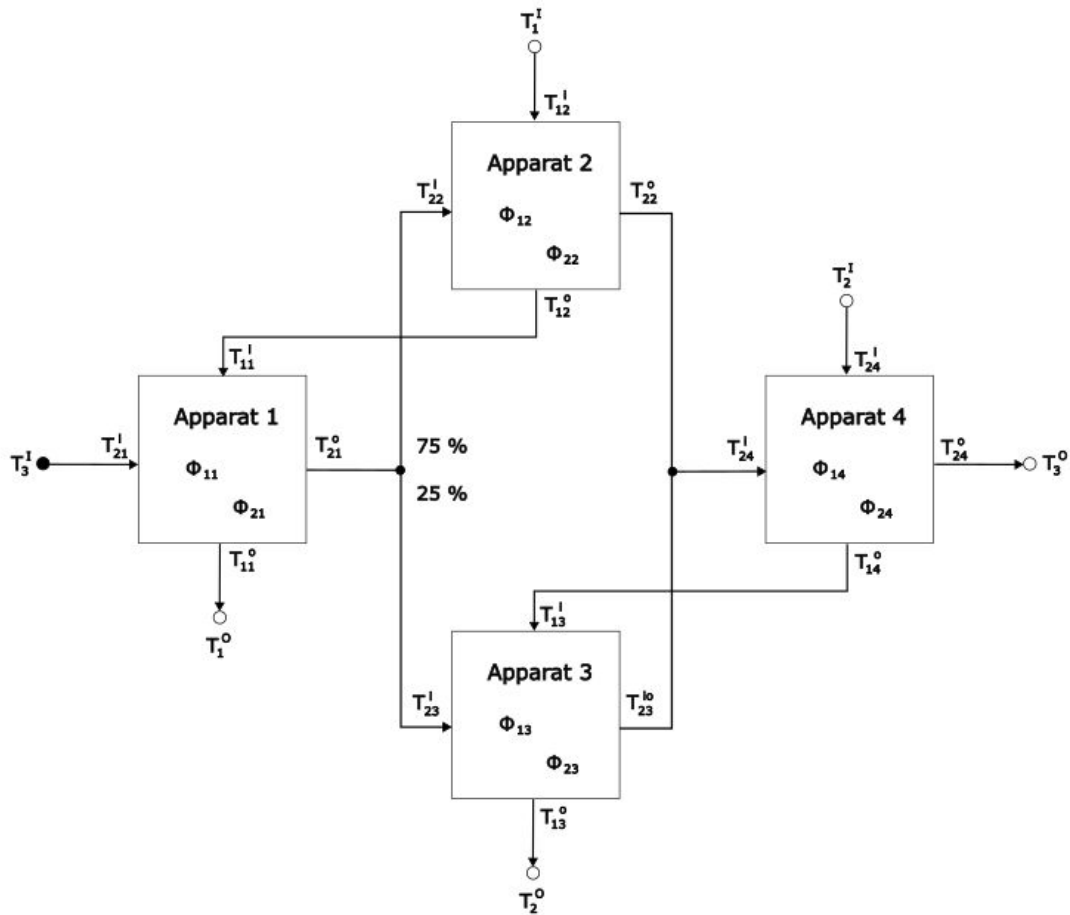


Abbildung 1: Wärmeübertragerschaltung mit Stromteilung

Wie zuvor beschrieben werden zunächst die Ströme initialisiert und anschließend die das Netzwerk beschreibende Matrizen. Dazu wird die aufsteigende Nummerierung der Ströme (vgl. `flows`) bzw. der Apparate gewählt.

```
[2]: # defining Flows (mass flow irrelevant for following calculations)
flow_1 = Flow(Fluid("Water", temperature=373), 1)
flow_2 = Flow(Fluid("Water", temperature=405), 1)
flow_3 = Flow(Fluid("Water", temperature=293), 1)
flows = [flow_1, flow_2, flow_3]
```

```
[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., 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 Apparate können mithilfe der Netzwerkeigenschaft `temperature_matrix` abgerufen werden. Diese Eigenschaft ein Tupel zurück, wobei der erste Eintrag eine Matrix der dimensionslosen Temperaturen und der zweite Eintrag `temperature_matrix[1]` eine Matrix der Temperaturen in Kelvin liefert. Die Matrix hat die doppelte Anzahl an Zeilen im Vergleich zur Anzahl der Apparate. Die erste Hälfte der Zeilen enthält die Austrittstemperaturen des ersten Stroms jedes Apparats, während die zweite Hälfte die des zweiten Stroms enthält. Die Reihenfolge der Apparate entspricht derjenigen, die bei der Definition der Netzwerkmatrizen verwendet wurde.

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

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

Die Austrittstemperaturen des gesamten Netzwerks in Kelvin ergibt sich, ähnlich zu jenen der einzelnen Apparate, durch die Netzwerkeigenschaft `temperature_outputs[1]`. Die Dimension und Sortierung ergibt sich aus der Initialisierung der Netzwerkströme.

Beispielsweise ist die Austrittstemperatur des Stroms 1 aus dem gesamten Netzwerk $T_1^O = 303K$.

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

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

Die Schaltungscharakteristik wird durch folgende Matrix beschrieben. Damit kann der relative Einfluss der Eintrittstemperaturen auf die Austrittstemperaturen des gesamten Netzwerks direkt abgelesen werden.

Beispielsweise hat die Änderung der Eintrittstemperatur des Fluidstroms 2 keinen Einfluss auf die Austrittstemperatur des Fluidstroms 1, bewirkt aber eine Änderung der Austrittstemperatur von 9,8 % der Eintrittstemperaturänderung, des selbigen Fluidstroms und eine Änderung von 22,5 % der Eintrittstemperaturänderung des 3. Fluidstroms.

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

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

1.3 Doppelrohr-Wärmeübertrager mit drei Strängen

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

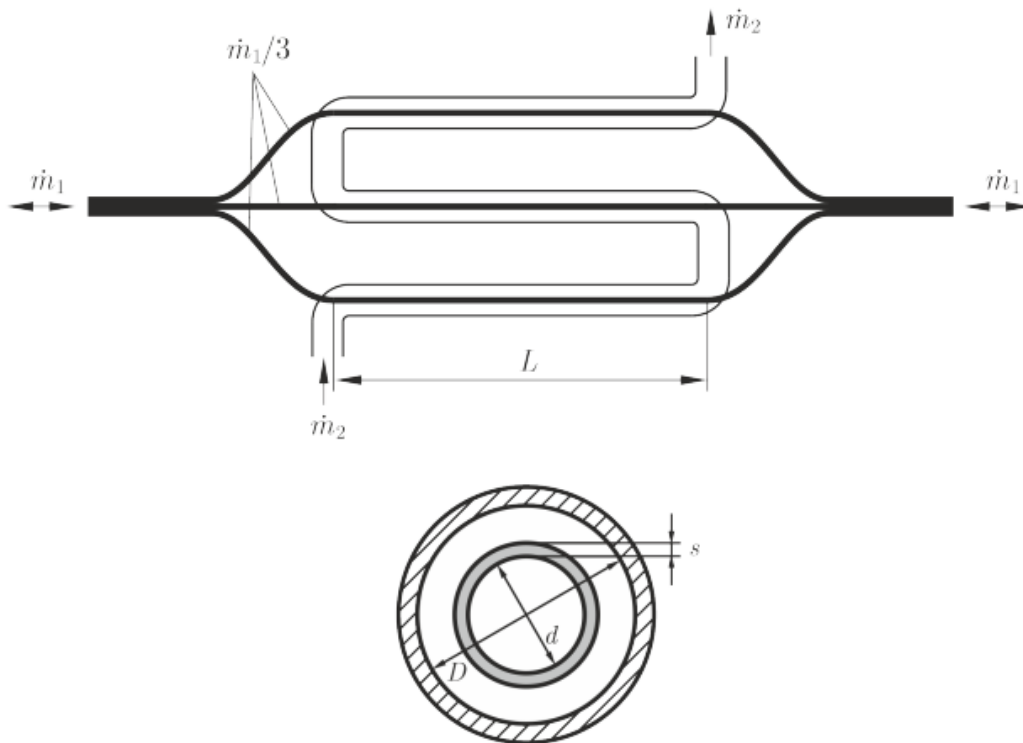


Abbildung 2: Doppelrohr-Wärmeübertrager mit drei Strängen

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

Für die Betriebscharakteristik der 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 Austrittstemperaturen in Grad Celsius sind somit:

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

array([[56.16      ],
       [60.32388   ]],
```

```
[62.3665626 ],  
[29.31069024]])
```

Wie zuvor bereits beschrieben ergibt sich die Sortierung der Temperaturen durch die Reihenfolge wie die Ströme im Netzwerk z. B. durch `flows` initialisiert werden.

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

Alternativ könnte auch die Outputmatrix für das gesamte Netzwerk definiert 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]])
```

Literatur

- [1] Olaf Strelow. „A general calculation method for heat exchanger networks/Eine allgemeine Berechnungsmethode für Wärmeübertragerschaltungen“. In: *Forschung im Ingenieurwesen* 63.9 (1997), S. 255 (siehe S. 2).