

Wärmeübertrager mit zwei Fluidströmen

heatexchanger_twoflows.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 *
from exchanger.exchanger_types import *
from exchanger.network import vis_setups
from exchanger.exchanger_creator import *

import numpy as np
from numpy import exp
```

1 Wärmeübertrager mit zwei Fluidströmen

Im Folgenden wird gezeigt, wie die Implementierung angewendet werden kann, um insbesondere Rohrbündelwärmeübertrager mit zwei Fluidströmen und ohne Phasenübergang zu berechnen.

Für ausführliche Details zur verwendeten Zellenmethode sei auf die theoretische Erklärung (Zellenmethode Theorie) verwiesen. Weiterführende Informationen zur Implementierung finden sich in den Beispielen und der Klassendokumentation.

1.1 Rohrbündelwärmeübertrager (4 Zellen)

1.1.1 Problemstellung

Folgendes Beispiel der Wärmeübertragerschaltung sowie die Parameter sind entnommen von Beispiel 8.1 [1].

Die Stromführung des Wärmeübertragers besteht aus zwei inneren und zwei äußeren Durchgängen mit einer mantelseitigen Umlenkung. Die für die Wärmeübertragung maßgebende Größe kA ist für alle Zellen gleich und beträgt 4000 W/K. Der Strom im Außenraum hat den Index 1. Um die Berechnung zu vereinfachen, werden die beiden Wärmekapazitätsströme \dot{W}_1 und \dot{W}_2 mit 3500 W/K gleich groß gewählt. Die Eintrittstemperatur des Stroms 1 ist $\vartheta'_1 = 100$ °C, und die des Stroms 2 ist $\vartheta'_2 = 20$ °C.

Zu bestimmen sind die Austrittstemperaturen ϑ'_1 und ϑ'_2 .

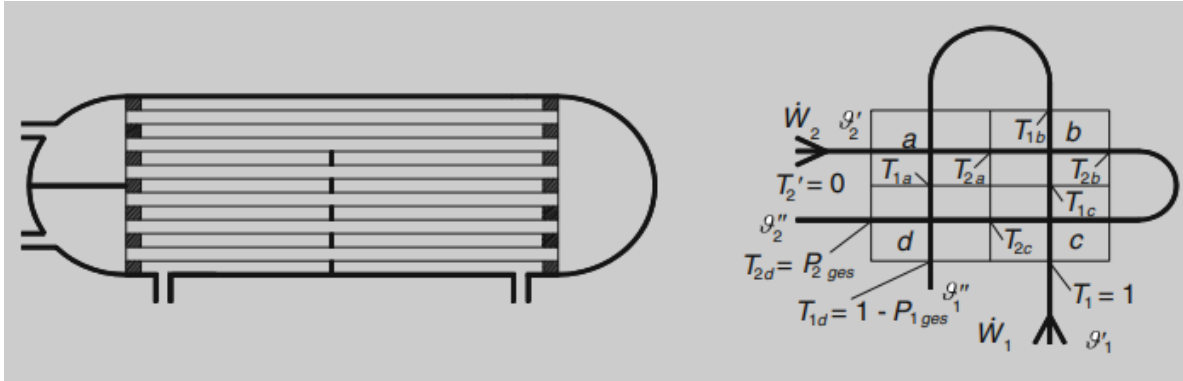


Abbildung 1: Stromführung eines Rohrbündelwärmeübertragers mit 4 Zellen (entnommen aus [1], Seite 262)

1.1.2 Initialisierung über ExchangerNetwork

Fluide und Fluidströme Zunächst werden die Fluidströme, im Folgenden beispielsweise Wasser, definiert. Der Massenstrom wird dabei so angepasst, dass die Wärmekapazitätsströme den in der Angabe vorgegebenen entsprechen. Der Druck für das Fluid wird so angepasst, dass es zu keiner Phasenänderung kommt, um eine korrekte Berechnung zu gewährleisten.

Auf die genaue Verwendung der Fluid- und Flow-Klassen sei in der Beschreibung in [standard_heatexchangers.ipynb](#) bzw. in der Klassendokumentation verwiesen.

```
[2]: W_dot_1 = 3500
fluid_1 = Fluid("Water", pressure=101420, temperature=273.15 + 100)
flow_1 = Flow(fluid_1, W_dot_1/fluid_1.specific_heat)
flow_1
```

```
[2]: Flow: id = 2079152763232
      mass flow = 0.83024 kg/s
      heat capacity flow: W_dot = 3500.00000 W/K
      heat flow: Q_dot = 0.00000 kW
```

```
Input Fluid:
  Fluid: title = Water, id = 2079152750944
  p = 101420 Pa
  t = 100.0 °C
```

```
Output Fluid:
  Fluid: title = Water, id = 2079152763328
  p = 101420 Pa
  t = 100.0 °C
```

```
[3]: W_dot_2 = 3500
fluid_2 = Fluid("Water", temperature=273.15 + 20)
flow_2 = Flow(fluid_2, W_dot_2/fluid_2.specific_heat)
flow_2
```

```
[3]: Flow: id = 2079212467392
      mass flow = 0.83651 kg/s
      heat capacity flow: W_dot = 3500.00000 W/K
      heat flow: Q_dot = 0.00000 kW
```

```
Input Fluid:
      Fluid: title = Water, id = 2079152765488
      p = 101325 Pa
      t = 20.0 °C
```

```
Output Fluid:
      Fluid: title = Water, id = 2079212466624
      p = 101325 Pa
      t = 20.0 °C
```

Wärmeübertrager-Netzwerk Die Verwendung der Klasse `ExchangerNetwork` wurde bereits im Notebook [complex_heatexchanger_networks.ipynb](#) bzw. in der Klassendokumentation beschrieben.

```
[4]: flows = [flow_1, flow_2]
      network = ExchangerNetwork(flows)
```

Um die Austrittstemperaturen mithilfe der Zellenmethode zu berechnen, muss die Charakteristik des Netzwerks bekannt sein. Dazu können beispielsweise die Matrizen, die das Netzwerk beschreiben, direkt definiert werden. Zunächst können die dimensionslosen Kennzahlen des gesamten Netzwerks berechnet werden. Für die einzelnen Zellen wird davon ausgegangen, dass sich die Wärmeübertragungsfähigkeit gleichmäßig auf die Zellen aufteilt.

```
[5]: R_1 = W_dot_1/W_dot_2
      R_2 = W_dot_2/W_dot_1
      display(f"R_1 = {R_1}, R_2 = {R_2}")
```

```
'R_1 = 1.0, R_2 = 1.0'
```

```
[6]: kA = 4000
      cells = 4
      NTU_1_ges = kA/W_dot_1
      NTU_2_ges = kA/W_dot_2
      NTU_1_cell = NTU_1_ges/cells
      NTU_2_cell = NTU_2_ges/cells
      display(f"NTU_1_cell = {NTU_1_cell}, NTU_2_cell = {NTU_2_cell}")
```

```
'NTU_1_cell = 0.2857142857142857, NTU_2_cell = 0.2857142857142857'
```

Es wird angenommen, dass eine Zelle durch einen Wärmeübertrager mit einer Rohrreihe und mantelseitigem Gegenstrom angenähert werden kann.

```
[7]: P_1_cell = 1 - exp((exp(-R_1 * NTU_1_cell) - 1) / R_1)

      P_2_cell = R_1*P_1_cell
```

```
display(f"P_1_cell = {P_1_cell}, P_2_cell = {P_2_cell}")
```

```
'P_1_cell = 0.22004784968070001, P_2_cell = 0.22004784968070001'
```

Matrixdefinition Mit diesen Parametern können nun die Matrizen für die Zellenmethode definiert werden. Wie genau die Matrizen aufgebaut werden, kann aus der theoretischen Beschreibung entnommen werden.

Für die Berechnung der Ausgangstemperaturen greift die Klasse `HeatExchangerNetwork` auf folgende Eigenschaften zurück, die entweder direkt definiert werden, wie im Folgenden zu sehen, oder automatisch generiert werden:

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

```
[8]: network.phi_matrix = np.array([[1-P_1_cell, 0., 0., 0., P_1_cell, 0., 0., 0.],
                                   [0., 1-P_1_cell, 0., 0., 0., P_1_cell, 0., 0.],
                                   [0., 0., 1-P_1_cell, 0., 0., 0., P_1_cell, 0.],
                                   [0., 0., 0., 1-P_1_cell, 0., 0., 0., P_1_cell],
                                   [P_2_cell, 0., 0., 0., 1-P_2_cell, 0., 0., 0.],
                                   [0., P_2_cell, 0., 0., 0., 1-P_2_cell, 0., 0.],
                                   [0., 0., P_2_cell, 0., 0., 0., 1-P_2_cell, 0.],
                                   [0., 0., 0., P_2_cell, 0., 0., 0., 1-P_2_cell]])
network.structure_matrix = np.array([[0, 1, 0, 0, 0, 0, 0, 0],
                                     [0, 0, 1, 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, 1, 0, 0, 0],
                                     [0, 0, 0, 0, 0, 1, 0, 0],
                                     [0, 0, 0, 0, 0, 0, 1, 0]])
network.input_matrix = np.array([[0, 0],
                                 [0, 0],
                                 [1, 0],
                                 [0, 0],
                                 [0, 1],
                                 [0, 0],
                                 [0, 0],
                                 [0, 0]])
network.output_matrix = np.asarray([[0, 0, 0, 1, 0, 0, 0, 0],
                                    [0, 0, 0, 0, 0, 0, 0, 1]])
```

Für die einzelnen Zellen ergeben sich somit die folgenden Austrittstemperaturen der einzelnen Apparate. Diese werden über die Eigenschaft `temperature_matrix` als Tupel bereitgestellt, entweder in dimensionsloser Form, also normiert auf die Eintrittstemperaturen unter `temperature_matrix[0]`,

oder in Kelvin unter `temperature_matrix[1]`.

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

```
[9]: array([[63.32779554],
          [75.5518637 ],
          [87.77593185],
          [61.86324704],
          [32.22406815],
          [44.4481363 ],
          [56.67220446],
          [58.13675296]])
```

Durch die Outputmatrix können zusätzlich die Austrittstemperaturen aus dem gesamten Netzwerk herausgefiltert bzw. berechnet werden. Für das beschriebene Netzwerk ergibt sich somit eine Austrittstemperatur des ersten Stroms von $\vartheta_1'' = 61.86$ °C und die des Stroms 2 ist $\vartheta_2'' = 58.13$ °C.

Diese Austrittstemperaturen des gesamten Netzwerks werden über die Eigenschaft `temperature_outputs`, in gleicher Form wie zuvor die Apparate-Austrittstemperaturen, bereitgestellt.

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

```
[10]: array([[61.86324704],
          [58.13675296]])
```

1.1.3 Initialisierung über `ExchangerEqualCells`

Wie sich zeigt, ist die Definition der Matrizen für die Zellenmethode nicht immer einfach. Deshalb kann die Berechnung eines Netzwerks mit zwei Strömen und gleichmäßiger Aufteilung der Wärmeübertragungsfähigkeit auf die einzelnen Zellen, also gleichen Apparateeigenschaften wie folgt durchgeführt werden:

Zunächst müssen erneut die beiden Ströme definiert werden.

```
[11]: W_dot_1 = 3500
fluid_1 = Fluid("Water", pressure=101420, temperature=273.15 + 100)
flow_1 = Flow(fluid_1, W_dot_1/fluid_1.specific_heat)

W_dot_2 = 3500
fluid_2 = Fluid("Water", temperature=273.15 + 20)
flow_2 = Flow(fluid_2, W_dot_2/fluid_2.specific_heat)
```

Netzwerkdefinition Der spezifische Wärmeüberträgertyp mit zwei Strömen, definierter Stromführung ohne Stromaufteilung und gleichmäßiger Verteilung der Wärmeübertragungsfähigkeit auf die verschiedenen Zellen kann mithilfe der Instanz `ExchangerEqualCells` definiert werden. Dazu sind folgende Angaben erforderlich:

- Die Definition der Größe des Netzwerks über ein Tupel `shape=(row,col)`. Der erste Eintrag stellt dabei die Anzahl der Zeilen des Netzwerks dar, und der zweite die Anzahl der Spalten

des Netzwerks.

- Über die Eigenschaft `exchangers_type` wird definiert, durch welchen Standard-Wärmeübertrager-Typ jede einzelne Zelle angenähert werden soll. Dabei ist erforderlich, dass der eingegebene String dem Namen einer implementierten Subklasse von `HeatExchanger` entspricht.
- Über die Eigenschaften `flow_1` und `flow_2` können die Fluidobjekte bereitgestellt werden.
- Durch `flow_order_1` und `flow_order_2` wird über einen String definiert, wie sich der jeweilige Strom durch das Netzwerk bewegt. Aufgrund der zuvor getroffenen Einschränkungen ist bereits bekannt, dass die Fluidströme nur an den "Ecken" eintreten können. Da es auch zu keiner Stromaufteilung kommt, muss nur noch definiert werden, in welche Richtung das Fluid strömt. Daher wird folgende Konvention verwendet: Die erste Stelle beschreibt die vertikale Position, entweder oben ('u') oder unten ('d'), die zweite Stelle definiert die horizontale Position ('r' für rechts und 'l' für links), danach folgt eine '2' und anschließend die Richtung, in welche sich der Strom bewegt, ebenfalls definiert durch die jeweiligen Buchstaben. Damit ist definiert, wie sich der Strom durch die rechteckig definierte Form des Netzwerks bewegt. Dabei wird angenommen, dass immer eine Zellenreihe oder -spalte für die Stromrichtungsänderung verwendet wird und die übrigen Zellen durch vertikale bzw. horizontale Konstruktionen, seien es Trennwände oder der Verlauf der Rohre, getrennt sind.

Im gegebenen Beispiel tritt der Strom rechts unten in den Mantel des Wärmetauschers ein. Die Eintrittsposition wird daher durch die Buchstaben 'dr' definiert. Da der Strom mantelseitig eintritt und durch die vertikalen Trennwände umgelenkt wird, kann sich der Strom nur in vertikaler Richtung, also 'u' oder 'd' bewegen. Beim Eintritt unten würde die Stromrichtung nach unten keinen Sinn ergeben, wodurch schlussendlich die `flow_order_1 = 'dr2u'` definiert wird.

- Über die Eigenschaften `total_transferability` oder `assembly` wird initialisiert, welche Wärmeübertragungsfähigkeit das gesamte Netzwerk besitzt und gleichmäßig auf die Apparate aufgeteilt wird.

Die Eigenschaft `total_transferability` erfordert lediglich den Float-Wert die Wärmeübertragungsfähigkeit. Ist aber beispielsweise die Konstruktion eines Wärmeübertrages bekannt, wird die Wärmeübertragungsfähigkeit über die wärmeübertragende Fläche und Wärmedurchgangskoeffizient berechnet. Diese Berechnung ist bereits in der Klasse `Assembly` implementiert. Daher ist die Eigenschaft `assembly` zu verwenden, wenn weitere Eigenschaften der Geometrie definiert werden sollen. Zur genaueren Implementierung und Verwendung der Klasse `Assembly` sei wiederum auf die Klassendokumentation verwiesen.

Mit diesen Eigenschaften kann die Klasse `ExchangerEqualCells` für das gegebene Beispiel wie folgt initialisiert werden:

```
[12]: kA = 4000
ex = ExchangerEqualCells(shape=(2, 2), exchangers_type='CrossFlowOneRow',
    ↳flow_1=flow_1, flow_order_1 = 'dr2u', flow_2=flow_2, flow_order_2 = 'ul2r',
    ↳total_transferability=kA)
```

Berechnungsmethode der Klasse `ExchangerEqualCells` Die Klasse `ExchangerEqualCells` erbt von `ExchangerNetworks`, um die Berechnung mithilfe der Zellenmethode durchzuführen. Mit

der Klasse ist es möglich, die Fluidparameter der einzelnen Zellen automatisch anzupassen oder die Betriebscharakteristik der Apparate zu berechnen. Dies erfolgt folgendermaßen:

1. Zunächst wird eine Matrix entsprechend der durch `shape` definierten Form mit `HeatExchanger`-Objekten oder den Subklassen, die durch `exchangers_type` definiert sind, erstellt. Jeder Zelle wird die entsprechende Zellenübertragungsfähigkeit zugeordnet.
2. Abhängig von den Stromführungen werden zunächst zwei Listen erzeugt, die dieselben `HeatExchanger`-Objekte enthalten. Die Reihenfolge hängt von der entsprechenden Durchlauf- richtung durch das Netzwerk oder die repräsentative Matrix ab.
3. Aus diesen Listen werden Tupellisten erzeugt, um mithilfe des Moduls `networkx` zunächst einen gerichteten Graphen mit den `HeatExchanger`-Objekten zu erstellen und daraus eine Adjazenzmatrix zu erzeugen. Diese können anschließend zur Strukturmatrix zusammengesetzt werden.
4. Aus der Liste der Heatexchanger kann die Phi-Matrix erzeugt werden, die im vorliegenden Fall nur einer Diagonalmatrix entspricht.
5. Die Input- und Outputmatrix ergeben sich ebenfalls aus der Liste der Heatexchanger und den definierten Strömen.

Damit sind alle notwendigen Matrizen erzeugt, und die Zellenmethode kann angewendet werden.

Mithilfe der Methode `_adjust_temperatures` werden die Temperaturen der einzelnen Heatexchanger-Objekte entsprechend der `temperature_matrix` angepasst, um die Berechnung mit genaueren Fluidparametern wiederholen zu können.

Die Eigenschaften des Networks können über die Methode `print` oder über `extended_info()` mit den einzelnen Zellen, ausgegeben werden.

Ergebnisse des Netzwerks und einzelnen Zellen Wie zu sehen ist, ergeben sich wieder die gleichen Austrittstemperaturen wie bei der direkten Eingabe der Matrizen. Die geringe Abweichung kommt dadurch zustande, dass die Berechnung iterativ erfolgt und die Austrittstemperaturen sowie die Fluiddaten und somit auch die dimensionslosen Kennzahlen der Zellen angepasst werden.

```
[13]: print(ex.extended_info())
```

```
Heat Exchanger Network:
    cell numbers: 4
    heat flows q_1=133.56 kW,      q_2=-133.56 kW
input flows: n=2
    flow 0: Water, temp= 100.00°C
    flow 1: Water, temp= 20.00°C
output flows: n=2
    flow 0: Water, temp= 62.18°C
    flow 1: Water, temp= 58.42°C

cell:0

heat exchanger:
    id = 2079212945712
```

```

        type: CrossFlowOneRow
Flows:
Flow 1:
Flow: id = 2079152753584
      mass flow = 0.83394 kg/s
      heat capacity flow: W_dot = 3510.12275 W/K
      heat flow: Q_dot = 42.61267 kW

Input Fluid:
      Fluid: title = Water, id = 2079152764768
      p = 101420 Pa
      t = 100.0 °C
Output Fluid:
      Fluid: title = Water, id = 2079212946432
      p = 101420 Pa
      t = 87.86043738378032 °C

Flow 2:
Flow: id = 2079212946144
      mass flow = 0.82769 kg/s
      heat capacity flow: W_dot = 3461.04439 W/K
      heat flow: Q_dot = -42.61240 kW

Input Fluid:
      Fluid: title = Water, id = 2079212470080
      p = 101325 Pa
      t = 44.616938608050646 °C
Output Fluid:
      Fluid: title = Water, id = 2079212946000
      p = 101325 Pa
      t = 56.928642847056494 °C

Parameters:

dimensionless parameters:
number of transfer units:
      NTU_1 = 0.285
      NTU_2 = 0.289
heat capacity flow ratios:
      R_1 = 1.014
      R_2 = 0.986
dimensionless temperature change:
      P_1 = 0.219
      P_2 = 0.222

cell:1

```



```

heat exchanger:
    id = 2079212468592
    type: CrossFlowOneRow
Flows:
Flow 1:
Flow: id = 2079212469504
    mass flow = 0.84090 kg/s
    heat capacity flow: W_dot = 3530.21705 W/K
    heat flow: Q_dot = 42.79428 kW

Input Fluid:
    Fluid: title = Water, id = 2079212946432
    p = 101420 Pa
    t = 87.86043738378032 °C
Output Fluid:
    Fluid: title = Water, id = 2079212468448
    p = 101420 Pa
    t = 75.73847561577139 °C

Flow 2:
Flow: id = 2079212466288
    mass flow = 0.83197 kg/s
    heat capacity flow: W_dot = 3477.07180 W/K
    heat flow: Q_dot = -42.79473 kW

Input Fluid:
    Fluid: title = Water, id = 2079212948400
    p = 101325 Pa
    t = 32.30969890322643 °C
Output Fluid:
    Fluid: title = Water, id = 2079212470080
    p = 101325 Pa
    t = 44.61693860804968 °C

Parameters:

dimensionless parameters:
number of transfer units:
    NTU_1 = 0.283
    NTU_2 = 0.288
heat capacity flow ratios:
    R_1 = 1.015
    R_2 = 0.985
dimensionless temperature change:
    P_1 = 0.218
    P_2 = 0.222

```

```

cell:2

heat exchanger:
    id = 2079212948592
    type: CrossFlowOneRow
Flows:
Flow 1:
Flow: id = 2079212947632
    mass flow = 0.84721 kg/s
    heat capacity flow: W_dot = 3549.71614 W/K
    heat flow: Q_dot = 42.98914 kW

Input Fluid:
    Fluid: title = Water, id = 2079212468448
    p = 101420 Pa
    t = 75.73847561577139 °C
Output Fluid:
    Fluid: title = Water, id = 2079212947920
    p = 101420 Pa
    t = 63.62816680007762 °C

Flow 2:
Flow: id = 2079212948112
    mass flow = 0.83528 kg/s
    heat capacity flow: W_dot = 3492.21853 W/K
    heat flow: Q_dot = -42.99117 kW

Input Fluid:
    Fluid: title = Water, id = 2079212948160
    p = 101325 Pa
    t = 20.0 °C
Output Fluid:
    Fluid: title = Water, id = 2079212948400
    p = 101325 Pa
    t = 32.30969890323888 °C

Parameters:

dimensionless parameters:
number of transfer units:
    NTU_1 = 0.282
    NTU_2 = 0.286
heat capacity flow ratios:
    R_1 = 1.016
    R_2 = 0.984
dimensionless temperature change:
    P_1 = 0.217
    P_2 = 0.221

```

```

cell:3

heat exchanger:
    id = 2079212469936
    type: CrossFlowOneRow
Flows:
Flow 1:
Flow: id = 2079152766448
    mass flow = 0.85044 kg/s
    heat capacity flow: W_dot = 3560.20351 W/K
    heat flow: Q_dot = 5.16144 kW

Input Fluid:
    Fluid: title = Water, id = 2079212947920
    p = 101420 Pa
    t = 63.62816680007762 °C
Output Fluid:
    Fluid: title = Water, id = 2079212467344
    p = 101420 Pa
    t = 62.178408317433025 °C

Flow 2:
Flow: id = 2079212469888
    mass flow = 0.82492 kg/s
    heat capacity flow: W_dot = 3451.44167 W/K
    heat flow: Q_dot = -5.16144 kW

Input Fluid:
    Fluid: title = Water, id = 2079212946000
    p = 101325 Pa
    t = 56.928642847056494 °C
Output Fluid:
    Fluid: title = Water, id = 2079212468352
    p = 101325 Pa
    t = 58.42408612612866 °C

Parameters:

dimensionless parameters:
number of transfer units:
    NTU_1 = 0.281
    NTU_2 = 0.290
heat capacity flow ratios:
    R_1 = 1.032
    R_2 = 0.969
dimensionless temperature change:

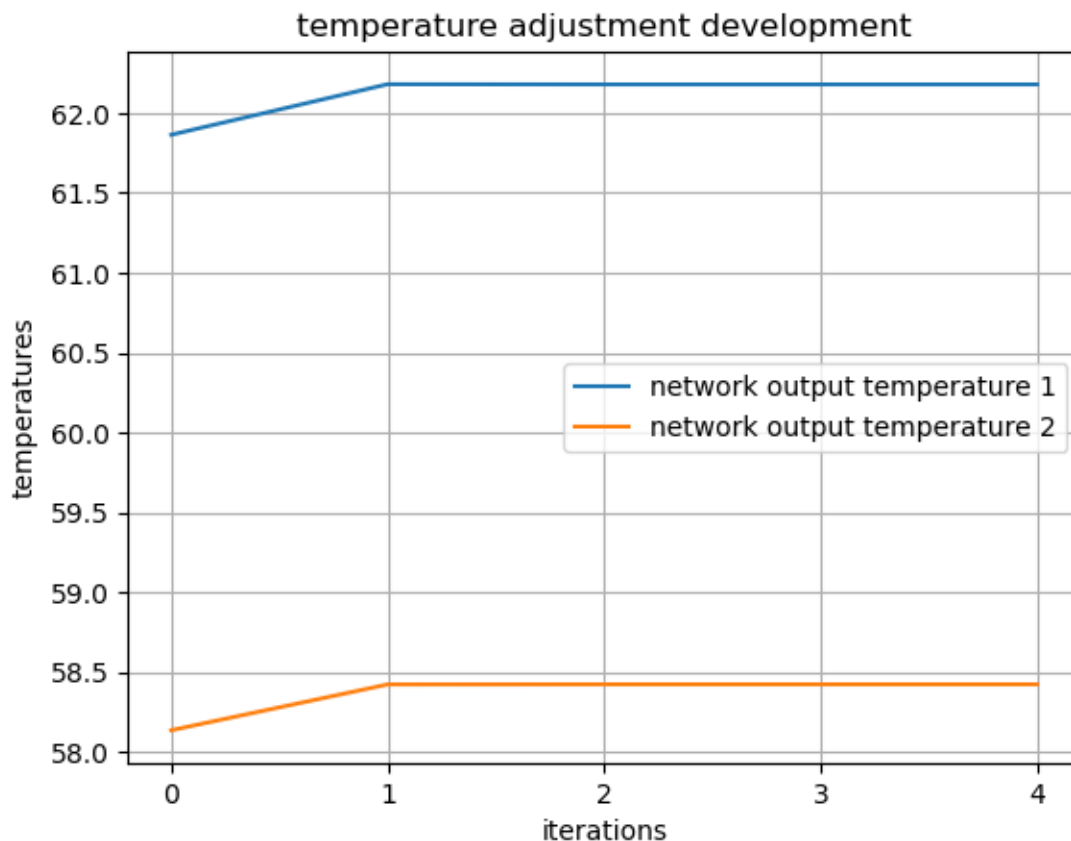
```

P_1 = 0.216
P_2 = 0.223

Anpassung der Fluidparameter und Visualisierung der Austrittstemperaturen Die Klasse `ExchangerEqualCells` ist eine Subklasse von `ExchangerTwoFlow`. Sie enthält das Attribut `auto_adjust`, welches, wie in der Klasse `Exchanger`, dazu dient, die Fluidparameter anzupassen. Dabei werden die Stromtemperaturen entsprechend der Stromführung und den Zelltemperaturen in jedem Apparat neu angepasst. Dies erfolgt standardmäßig über 5 Iterationen oder kann über die Methode `_adjust_temperatures(iterations)` angepasst werden.

Um den Einfluss der Fluidparameter auf die Ausgangstemperatur zu visualisieren, kann der folgende Plot mit der Methode `vis_temperature_adjustment_development()` herangezogen werden. Es zeigt sich, dass nach einer Iteration, also dem Anpassen der Zelltemperaturen am Ein- und Ausgang der Zelle, die Ausgangstemperatur des Netzwerks bereits ausreichend genau ist und keine signifikanten Änderungen mehr erfährt.

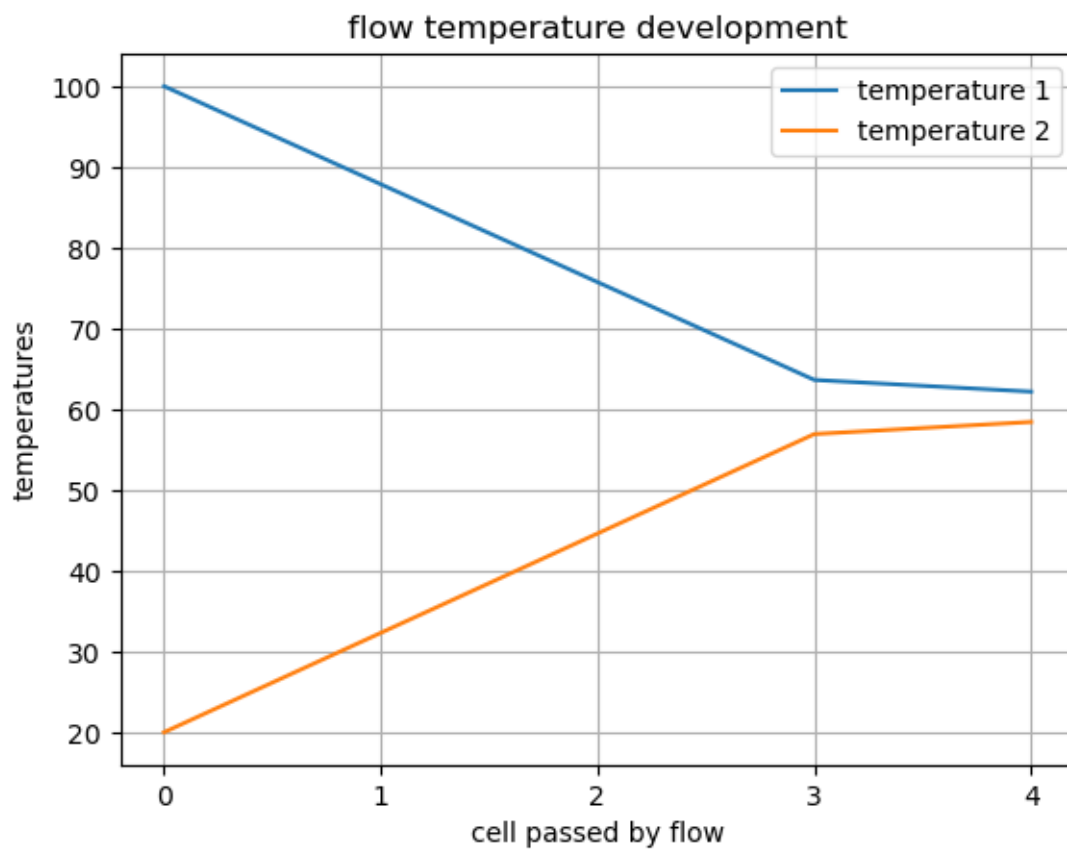
```
[14]: ex.vis_temperature_adjustment_development()
```

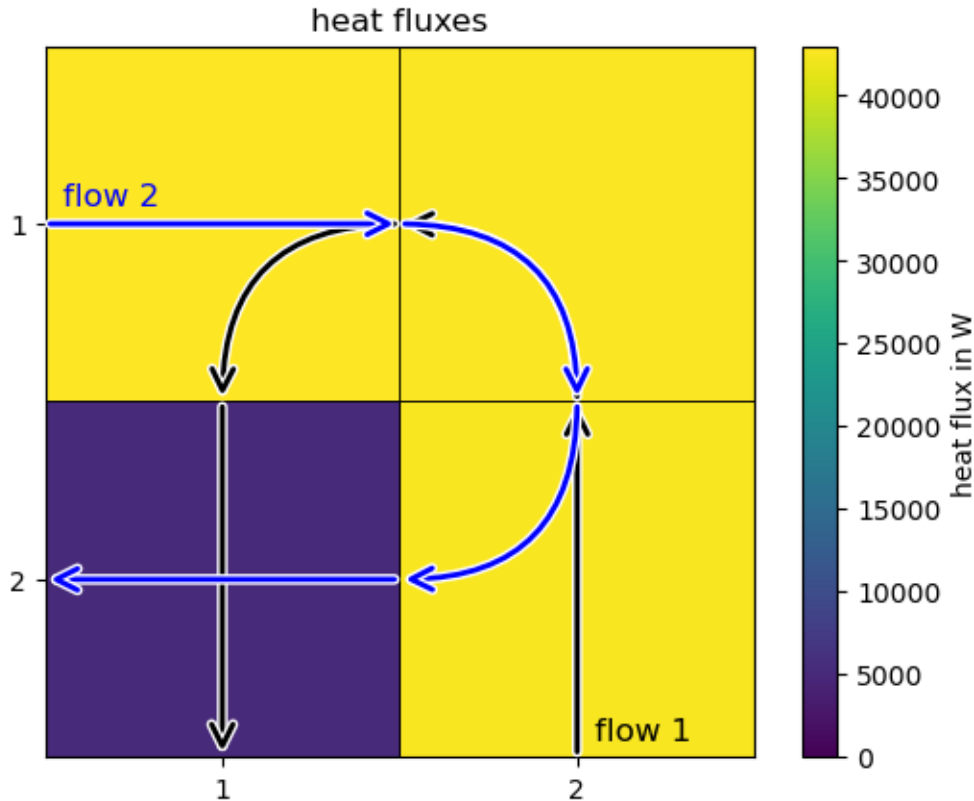


Darstellung des Temperaturverlaufs und Wärmestroms der einzelnen Apparate Zur Analyse des Netzwerks kann der Temperaturverlauf der einzelnen Ströme oder des umgesetzten Wärmestroms hilfreich sein. Im gegebenen Beispiel zeigt sich, dass in den ersten 3 Zellen, die vom jeweiligen Strom durchlaufen werden, der gleiche Wärmestrom umgesetzt wird bzw. die Temperaturänderung innerhalb einer Zelle gleich ist. In der letzten Zelle ist die Temperaturdifferenz zwischen den Strömen so gering, dass nicht mehr so viel Wärmestrom übertragen wird.

Die Visualisierung des Temperaturverlaufs durch die Methode `vis_flow_temperature_development()` zeigt nur die Ein- und Austrittstemperaturen an den Zellen. Der lineare Verlauf dazwischen entspricht nicht dem tatsächlichen Temperaturverlauf innerhalb einer Zelle. Der übertragene Wärmestrom wird durch die Methode `vis_heat_flux()` visualisiert. Die Darstellung entspricht dem zuvor über `shape` definierten Layout des Netzwerks, und die Pfeile visualisieren, in welche Richtung der jeweilige Strom durch die Zellen fließt.

```
[15]: ex.vis_flow_temperature_development()  
      ex.vis_heat_flux()
```





1.1.4 Betrachtung unterschiedlicher Stromführungen

Welche Ein- und Austrittspositionen am besten für die gegebenen Fluidraten sind, kann anhand folgender Analyse beurteilt werden. Zu beachten ist, dass im Folgenden alle (theoretisch) möglichen Eintritts- und Austrittspositionen visualisiert werden. Ob und mit welcher Konstruktion dies auch in einem Rohrbündelwärmeübertrager umgesetzt werden kann, wird außer Acht gelassen. Weiterhin ist zu beachten, dass unterschiedliche Stromführungen eventuell auch besser durch andere Wärmeübertragertypen in den einzelnen Zellen angenähert werden können und berechnete Druckverluste von der genauen Stromführung abhängig sein können.

Die folgenden Visualisierungen zeigen daher nur die Möglichkeiten, sofern alle Zellen gleich behandelt werden. Durch Anpassen der einzelnen Zellen in den Setups kann eine genauere Beurteilung durchgeführt werden.

Es zeigt sich, dass die Setups, in denen in jeder Zelle der gleiche Wärmestrom übertragen wird, den höchsten Wirkungsgrad haben, was bedeutet, dass die Austrittstemperaturen die größte Temperaturdifferenz bezogen auf die Eintrittstemperaturen aufweisen.

Implementierung Zunächst muss eine Methode `init_ex()` definiert werden, um die konstanten Daten des Wärmeübertragernetzwerks, wie oben beschrieben, zu initialisieren. Um die unterschiedlichen Stromführungen zu bewerten, wird in einer Schleife dem Wärmeübertragernetzwerk, der Reihe nach verschiedene Stromführung zugewiesen. Dies kann entweder selbst definiert werden oder es

wird auf die Methode `input_arrangements` der Klasse `ExchangerTwoFlow` zurückgegriffen, welche eine Liste aller Permutationen der implementierten Strömungseintritte erzeugt. Dabei muss darauf geachtet werden, dass nicht alle Kombinationen konstruktiv umsetzbar sind. Ist das Netzwerk vollständig initialisiert, können die Temperaturen berechnet werden. Da im Folgenden keine textuelle Ausgabe des Netzwerks erzeugt wird, muss die Methode `_adjust_temperature()` manuell ausgeführt werden. Das berechnete Netzwerk wird nun in einer Liste abgespeichert.

Die Funktion `vis_setups(network_list: list, plot_function, fig_title: str = , **ax_parameters)` ermöglicht es, eine Liste von mehreren Netzwerken zu visualisieren. Dazu muss die Liste der Wärmeübertragernetzwerke übergeben werden und der Name der gewünschten Plot-Methode, die in der jeweiligen Klasse des Netzwerks implementiert sein muss. Zusätzlich können noch Parameter für den Plot übergeben werden. Im Folgenden wird beispielsweise der minimale und maximale Wärmestrom übergeben, um die Achsenparameter aller Plots gleich anzupassen und eine bessere Vergleichbarkeit der Heatmaps zu gewährleisten.

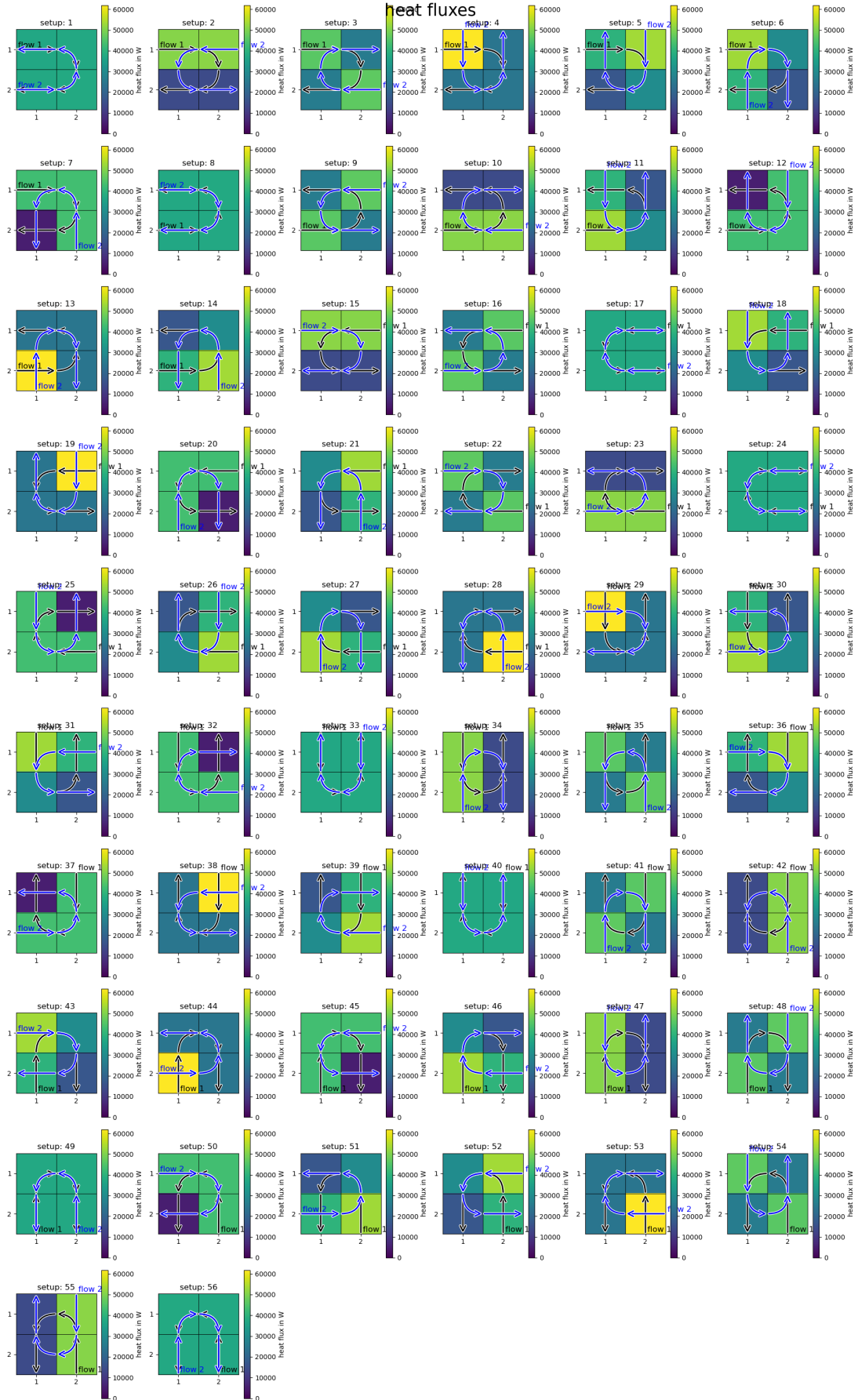
```
[16]: def init_ex():
    W_dot_1 = 3500
    fluid_1 = Fluid("Water", pressure=101420, temperature=273.15 + 100)
    flow_1 = Flow(fluid_1, W_dot_1/fluid_1.specific_heat)

    W_dot_2 = 3500
    fluid_2 = Fluid("Water", temperature=273.15 + 20)
    flow_2 = Flow(fluid_2, W_dot_2/fluid_2.specific_heat)

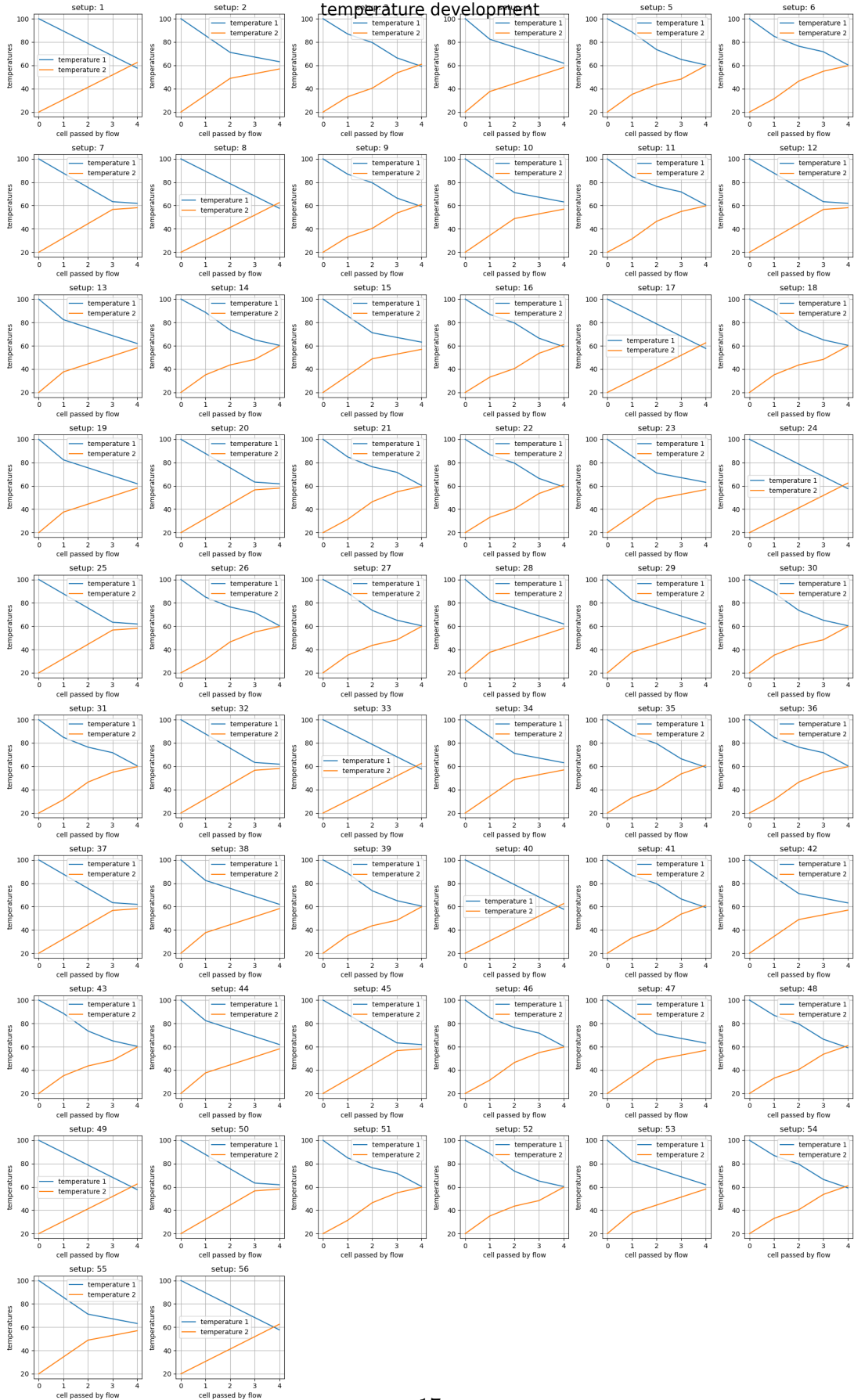
    kA = 4000
    ex = ExchangerEqualCells((2, 2), 'CrossFlowOneRow', flow_1=flow_1,
    ↪flow_2=flow_2, total_transferability=kA)
    return ex

[17]: networks = []
for inp in ExchangerTwoFlow.input_arrangements():
    netw = init_ex()
    netw.flow_order_1 = inp[0]
    netw.flow_order_2 = inp[1]
    netw._adjust_temperatures()
    networks.append(netw)

# calculate min and max heatflow of all available networks
ax_parameters_heat = {'vmin': 0, 'vmax': max([heat_flow_repr(netw.layout_matrix).
    ↪max() for netw in networks])}
vis_setups(networks, 'vis_heat_flux', fig_title='heat fluxes',
    ↪**ax_parameters_heat)
vis_setups(networks, 'vis_flow_temperature_development', fig_title='temperature
    ↪development')
```



temperature development



1.2 Rohrbündelwärmeübertrager mit mehreren Umlenkungen

Das folgende Beispiel zeigt, wie ein Rohrbündelwärmeübertrager mit mehreren Stromumlenkungen berechnet werden kann. Das Beispiel ist aus Kapitel 3.5.1, [2] entnommen.

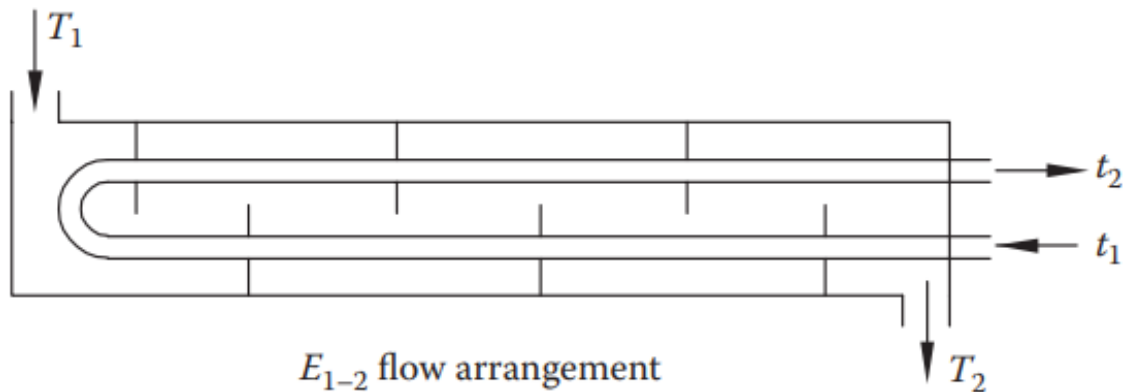


Abbildung 2: Stromführung eines Rohrbündelwärmeübertragers mit mehreren vertikalen Stromumlenkungsbleche (entnommen aus [2], Seite 330)

Zunächst müssen wiederum die konstruktiven Merkmale sowie die Fluidaten des Rohrbündelwärmeübertragers definiert werden.

Implementierung Die Definition der Fluide erfolgt, wie bereits zuvor beschrieben.

Die gesamte Konstruktion wird durch die Klasse `Assembly` definiert. Dazu sind folgende Eigenschaften zu übergeben:

- Das Rohrbündel kann, wie in `standard_heatexchangers.ipynb` beschrieben, definiert werden.
- Über die Klasse `Shell` bzw. deren Subklassen wie z.B. `SquareShellGeometry` kann die Größe des Mantels definiert werden.

Im gegebenen Fall wird für das Rohrbündel keine Rohrlänge angegeben, sondern wird über die Länge des Mantels und die Anzahl der Umlenkungen berechnet.

- Die Fluid-Eintritte können mithilfe der Klasse `Inlets` definiert werden. Die Konvention ist dieselbe wie bei der zuvor beschriebenen Stromführung, jedoch ohne die letzten zwei Stellen. Diese ergeben sich aus dem konstruktiven Aufbau und sind durch den mantelseitigen `shell_inlet` oder den Rohrbündeleinlass `tube_inlet` definiert.
- Die vertikalen Sperren bzw. Stromumlenkungsbleche können für die Klasse `Baffle` bzw. deren Subklassen definiert werden. Dazu muss auf jeden Fall die Anzahl definiert werden und eventuell noch zusätzliche Parameter.
- Die Anzahl der Rohrdurchgänge, also die Anzahl der horizontalen Stromdurchführungen, wird über `tube_passes` definiert.

Ist keine Rohrlänge angegeben, wird diese basierend auf der Mantelgeometrie und den Rohrdurchführungen berechnet.

```
[18]: W_dot_1 = 3500
fluid_1 = Fluid("Water", pressure=101420, temperature=273.15 + 100)
flow_1 = Flow(fluid_1, W_dot_1/fluid_1.specific_heat)

W_dot_2 = 3500
fluid_2 = Fluid("Water", temperature=273.15 + 20)
flow_2 = Flow(fluid_2, W_dot_2/fluid_2.specific_heat)
```

```
[19]: shell = SquareShellGeometry(length=5, width_in=2, height_in=1)
pipe = StraightPipe(10e-3, 13e-3)
pipe_layout = PipeLayout(pipe, 20)

inlets = Inlets(shell_inlet='ul', tube_inlet='dr')

baffle = SegmentalBaffle(number=6)

assembly = Assembly(shell, pipe_layout,
    ↳tube_passes=2,baffle=baffle,inlets=inlets)
assembly.heat_transfer_coefficient = 500
```

Damit können alle notwendigen Daten mithilfe der Zellenmethode berechnet werden:

Die zwei definierten Ströme und die Konstruktion können der Funktion `auto_create_exchanger()` übergeben werden. Der erste Strom `flow_1` stellt dabei den äußeren, also mantelseitigen, und der zweite `flow_2` den Strom in den Rohrbündeln dar. Die Funktion erzeugt ein `ExchangerEqualCells`-Objekt, das einem Rohrbündelwärmeübertrager, wie zuvor definiert, entspricht.

```
[20]: ex_layout=auto_create_exchanger(flow_1=flow_1,flow_2=flow_2,assembly=assembly)
print(ex_layout.extended_info())
```

Heat Exchanger Network:

cell numbers: 14

heat flows $q_1=131.09$ kW, $q_2=-131.09$ kW

input flows: n=2

flow 0: Water, temp= 100.00°C

flow 1: Water, temp= 20.00°C

output flows: n=2

flow 0: Water, temp= 62.87°C

flow 1: Water, temp= 57.71°C

cell:0

heat exchanger:

id = 2079515000992

type: CrossFlowOneRow

Flows:

Flow 1:
Flow: id = 2079468102768
 mass flow = 0.83141 kg/s
 heat capacity flow: \dot{W} = 3503.17037 W/K
 heat flow: \dot{Q} = 13.32112 kW

Input Fluid:
 Fluid: title = Water, id = 2079468101952
 p = 101420 Pa
 t = 100.0 °C

Output Fluid:
 Fluid: title = Water, id = 2079468105696
 p = 101420 Pa
 t = 96.19742199336883 °C

Flow 2:
Flow: id = 2079465283616
 mass flow = 0.82941 kg/s
 heat capacity flow: \dot{W} = 3467.23813 W/K
 heat flow: \dot{Q} = -13.32111 kW

Input Fluid:
 Fluid: title = Water, id = 2079515324144
 p = 101325 Pa
 t = 44.21568586750652 °C

Output Fluid:
 Fluid: title = Water, id = 2079515002528
 p = 101325 Pa
 t = 48.05767136882645 °C

Parameters:

dimensionless parameters:

number of transfer units:

 NTU_1 = 0.073

 NTU_2 = 0.074

heat capacity flow ratios:

 R_1 = 1.010

 R_2 = 0.990

dimensionless temperature change:

 P_1 = 0.068

 P_2 = 0.069

cell:1

heat exchanger:

 id = 2079515324672

```

        type: CrossFlowOneRow
Flows:
Flow 1:
Flow: id = 2079515326064
      mass flow = 0.83372 kg/s
      heat capacity flow: W_dot = 3509.50943 W/K
      heat flow: Q_dot = 13.33144 kW

Input Fluid:
      Fluid: title = Water, id = 2079468105696
      p = 101420 Pa
      t = 96.19742199336883 °C
Output Fluid:
      Fluid: title = Water, id = 2079515325728
      p = 101420 Pa
      t = 92.39877231079413 °C

Flow 2:
Flow: id = 2079515326304
      mass flow = 0.83074 kg/s
      heat capacity flow: W_dot = 3472.22019 W/K
      heat flow: Q_dot = -13.33144 kW

Input Fluid:
      Fluid: title = Water, id = 2079515327024
      p = 101325 Pa
      t = 40.37624132151234 °C
Output Fluid:
      Fluid: title = Water, id = 2079515324144
      p = 101325 Pa
      t = 44.21568586750959 °C

Parameters:

dimensionless parameters:
number of transfer units:
      NTU_1 = 0.073
      NTU_2 = 0.074
heat capacity flow ratios:
      R_1 = 1.011
      R_2 = 0.989
dimensionless temperature change:
      P_1 = 0.068
      P_2 = 0.069

cell:2

```

```

heat exchanger:
    id = 2079515326112
    type: CrossFlowOneRow
Flows:
Flow 1:
Flow: id = 2079515312672
    mass flow = 0.83597 kg/s
    heat capacity flow: W_dot = 3515.83303 W/K
    heat flow: Q_dot = 13.34212 kW

Input Fluid:
    Fluid: title = Water, id = 2079515325728
    p = 101420 Pa
    t = 92.39877231079413 °C
Output Fluid:
    Fluid: title = Water, id = 2079515327504
    p = 101420 Pa
    t = 88.60391496902656 °C

Flow 2:
Flow: id = 2079515314400
    mass flow = 0.83198 kg/s
    heat capacity flow: W_dot = 3477.07888 W/K
    heat flow: Q_dot = -13.34213 kW

Input Fluid:
    Fluid: title = Water, id = 2079515326160
    p = 101325 Pa
    t = 36.53908799555575 °C
Output Fluid:
    Fluid: title = Water, id = 2079515327024
    p = 101325 Pa
    t = 40.37624132151342 °C

Parameters:

dimensionless parameters:
number of transfer units:
    NTU_1 = 0.073
    NTU_2 = 0.074
heat capacity flow ratios:
    R_1 = 1.011
    R_2 = 0.989
dimensionless temperature change:
    P_1 = 0.068
    P_2 = 0.069

```

```

cell:3

heat exchanger:
    id = 2079514522240
    type: CrossFlowOneRow
Flows:
Flow 1:
Flow: id = 2079445445664
    mass flow = 0.83786 kg/s
    heat capacity flow: W_dot = 3521.26560 W/K
    heat flow: Q_dot = 9.68360 kW

Input Fluid:
    Fluid: title = Water, id = 2079515327504
    p = 101420 Pa
    t = 88.60391496902656 °C
Output Fluid:
    Fluid: title = Water, id = 2079217439872
    p = 101420 Pa
    t = 85.85388475925936 °C

Flow 2:
Flow: id = 2079215350880
    mass flow = 0.82819 kg/s
    heat capacity flow: W_dot = 3462.82445 W/K
    heat flow: Q_dot = -9.68360 kW

Input Fluid:
    Fluid: title = Water, id = 2079515002528
    p = 101325 Pa
    t = 48.05767136882645 °C
Output Fluid:
    Fluid: title = Water, id = 2079369676784
    p = 101325 Pa
    t = 50.85411308929889 °C

Parameters:

dimensionless parameters:
number of transfer units:
    NTU_1 = 0.073
    NTU_2 = 0.074
heat capacity flow ratios:
    R_1 = 1.017
    R_2 = 0.983
dimensionless temperature change:
    P_1 = 0.068
    P_2 = 0.069

```

```

cell:4

heat exchanger:
    id = 2079464478400
    type: CrossFlowOneRow
Flows:
Flow 1:
Flow: id = 2079514738016
    mass flow = 0.83931 kg/s
    heat capacity flow: W_dot = 3525.49862 W/K
    heat flow: Q_dot = 8.35899 kW

Input Fluid:
    Fluid: title = Water, id = 2079217439872
    p = 101420 Pa
    t = 85.85388475925936 °C
Output Fluid:
    Fluid: title = Water, id = 2079510221520
    p = 101420 Pa
    t = 83.48287856553708 °C

Flow 2:
Flow: id = 2079465765088
    mass flow = 0.82719 kg/s
    heat capacity flow: W_dot = 3459.28522 W/K
    heat flow: Q_dot = -8.35899 kW

Input Fluid:
    Fluid: title = Water, id = 2079369676784
    p = 101325 Pa
    t = 50.85411308929889 °C
Output Fluid:
    Fluid: title = Water, id = 2079510077904
    p = 101325 Pa
    t = 53.27050217565346 °C

Parameters:

dimensionless parameters:
number of transfer units:
    NTU_1 = 0.073
    NTU_2 = 0.074
heat capacity flow ratios:
    R_1 = 1.019
    R_2 = 0.981
dimensionless temperature change:

```


P_1 = 0.068
P_2 = 0.069

cell:5

heat exchanger:

id = 2079515313728
type: CrossFlowOneRow

Flows:

Flow 1:

Flow: id = 2079467201648
mass flow = 0.84091 kg/s
heat capacity flow: W_dot = 3530.25304 W/K
heat flow: Q_dot = 12.04085 kW

Input Fluid:

Fluid: title = Water, id = 2079510221520
p = 101420 Pa
t = 83.48287856553708 °C

Output Fluid:

Fluid: title = Water, id = 2079467204384
p = 101420 Pa
t = 80.07212433123476 °C

Flow 2:

Flow: id = 2079515319584
mass flow = 0.83307 kg/s
heat capacity flow: W_dot = 3481.60719 W/K
heat flow: Q_dot = -12.04087 kW

Input Fluid:

Fluid: title = Water, id = 2079467816032
p = 101325 Pa
t = 33.08067788702965 °C

Output Fluid:

Fluid: title = Water, id = 2079515326160
p = 101325 Pa
t = 36.53908799555637 °C

Parameters:

dimensionless parameters:

number of transfer units:

NTU_1 = 0.073
NTU_2 = 0.074

heat capacity flow ratios:

R_1 = 1.014

```

        R_2 = 0.986
dimensionless temperature change:
        P_1 = 0.068
        P_2 = 0.069

cell:6

heat exchanger:
    id = 2079467824384
    type: CrossFlowOneRow
Flows:
Flow 1:
Flow: id = 2079467583856
    mass flow = 0.84276 kg/s
    heat capacity flow: W_dot = 3535.81866 W/K
    heat flow: Q_dot = 12.05325 kW

Input Fluid:
    Fluid: title = Water, id = 2079467204384
    p = 101420 Pa
    t = 80.07212433123476 °C
Output Fluid:
    Fluid: title = Water, id = 2079467196176
    p = 101420 Pa
    t = 76.66323144847166 °C

Flow 2:
Flow: id = 2079467819680
    mass flow = 0.83402 kg/s
    heat capacity flow: W_dot = 3485.84894 W/K
    heat flow: Q_dot = -12.05328 kW

Input Fluid:
    Fluid: title = Water, id = 2079467579104
    p = 101325 Pa
    t = 29.62291844659717 °C
Output Fluid:
    Fluid: title = Water, id = 2079467816032
    p = 101325 Pa
    t = 33.080677887028344 °C

Parameters:

dimensionless parameters:
number of transfer units:
    NTU_1 = 0.073
    NTU_2 = 0.074

```

heat capacity flow ratios:

R_1 = 1.014

R_2 = 0.986

dimensionless temperature change:

P_1 = 0.068

P_2 = 0.069

cell:7

heat exchanger:

id = 2079514738304

type: CrossFlowOneRow

Flows:

Flow 1:

Flow: id = 2079465510544

mass flow = 0.84407 kg/s

heat capacity flow: W_dot = 3539.85411 W/K

heat flow: Q_dot = 5.58750 kW

Input Fluid:

Fluid: title = Water, id = 2079467196176

p = 101420 Pa

t = 76.66323144847166 °C

Output Fluid:

Fluid: title = Water, id = 2079513063584

p = 101420 Pa

t = 75.08477630509958 °C

Flow 2:

Flow: id = 2079513919440

mass flow = 0.82639 kg/s

heat capacity flow: W_dot = 3456.50238 W/K

heat flow: Q_dot = -5.58750 kW

Input Fluid:

Fluid: title = Water, id = 2079510077904

p = 101325 Pa

t = 53.27050217565346 °C

Output Fluid:

Fluid: title = Water, id = 2079512423648

p = 101325 Pa

t = 54.887020928725974 °C

Parameters:

dimensionless parameters:

number of transfer units:

```

        NTU_1 = 0.072
        NTU_2 = 0.074
heat capacity flow ratios:
        R_1 = 1.024
        R_2 = 0.976
dimensionless temperature change:
        P_1 = 0.067
        P_2 = 0.069

cell:8

heat exchanger:
        id = 2079217610672
        type: CrossFlowOneRow
Flows:
Flow 1:
Flow: id = 2079510724128
        mass flow = 0.84483 kg/s
        heat capacity flow: W_dot = 3542.21760 W/K
        heat flow: Q_dot = 4.82437 kW

Input Fluid:
        Fluid: title = Water, id = 2079513063584
        p = 101420 Pa
        t = 75.08477630509958 °C
Output Fluid:
        Fluid: title = Water, id = 2079515319488
        p = 101420 Pa
        t = 73.72281356920212 °C

Flow 2:
Flow: id = 2079515320352
        mass flow = 0.82578 kg/s
        heat capacity flow: W_dot = 3454.39793 W/K
        heat flow: Q_dot = -4.82437 kW

Input Fluid:
        Fluid: title = Water, id = 2079512423648
        p = 101325 Pa
        t = 54.887020928725974 °C
Output Fluid:
        Fluid: title = Water, id = 2079447880640
        p = 101325 Pa
        t = 56.28360825265497 °C

Parameters:

```

```

dimensionless parameters:
number of transfer units:
    NTU_1 = 0.072
    NTU_2 = 0.074
heat capacity flow ratios:
    R_1 = 1.025
    R_2 = 0.975
dimensionless temperature change:
    P_1 = 0.067
    P_2 = 0.069

cell:9

heat exchanger:
    id = 2079467581840
    type: CrossFlowOneRow
Flows:
Flow 1:
Flow: id = 2079467578192
    mass flow = 0.84599 kg/s
    heat capacity flow: W_dot = 3545.85130 W/K
    heat flow: Q_dot = 11.31228 kW

Input Fluid:
    Fluid: title = Water, id = 2079515319488
    p = 101420 Pa
    t = 73.72281356920212 °C
Output Fluid:
    Fluid: title = Water, id = 2079467578528
    p = 101420 Pa
    t = 70.53253311602703 °C

Flow 2:
Flow: id = 2079467574304
    mass flow = 0.83486 kg/s
    heat capacity flow: W_dot = 3489.94661 W/K
    heat flow: Q_dot = -11.31231 kW

Input Fluid:
    Fluid: title = Water, id = 2079468268336
    p = 101325 Pa
    t = 26.38153359537779 °C
Output Fluid:
    Fluid: title = Water, id = 2079467579104
    p = 101325 Pa
    t = 29.622918446597623 °C

```

Parameters:

dimensionless parameters:

number of transfer units:

NTU_1 = 0.072

NTU_2 = 0.074

heat capacity flow ratios:

R_1 = 1.016

R_2 = 0.984

dimensionless temperature change:

P_1 = 0.067

P_2 = 0.068

cell:10

heat exchanger:

id = 2079467318784

type: CrossFlowOneRow

Flows:

Flow 1:

Flow: id = 2079468262528

mass flow = 0.84758 kg/s

heat capacity flow: W_dot = 3550.88665 W/K

heat flow: Q_dot = 11.32552 kW

Input Fluid:

Fluid: title = Water, id = 2079467578528

p = 101420 Pa

t = 70.53253311602703 °C

Output Fluid:

Fluid: title = Water, id = 2079468267712

p = 101420 Pa

t = 67.34304843170662 °C

Flow 2:

Flow: id = 2079468271360

mass flow = 0.83559 kg/s

heat capacity flow: W_dot = 3493.94583 W/K

heat flow: Q_dot = -11.32556 kW

Input Fluid:

Fluid: title = Water, id = 2079467409216

p = 101325 Pa

t = 23.14006989861298 °C

Output Fluid:

Fluid: title = Water, id = 2079468268336

p = 101325 Pa

t = 26.381533595378528 °C

Parameters:

dimensionless parameters:

number of transfer units:

NTU_1 = 0.072

NTU_2 = 0.073

heat capacity flow ratios:

R_1 = 1.016

R_2 = 0.984

dimensionless temperature change:

P_1 = 0.067

P_2 = 0.068

cell:11

heat exchanger:

id = 2079515315312

type: CrossFlowOneRow

Flows:

Flow 1:

Flow: id = 2079515312864

mass flow = 0.84853 kg/s

heat capacity flow: W_dot = 3553.95432 W/K

heat flow: Q_dot = 2.64188 kW

Input Fluid:

Fluid: title = Water, id = 2079468267712

p = 101420 Pa

t = 67.34304843170662 °C

Output Fluid:

Fluid: title = Water, id = 2079515322560

p = 101420 Pa

t = 66.59968472573507 °C

Flow 2:

Flow: id = 2079515315456

mass flow = 0.82534 kg/s

heat capacity flow: W_dot = 3452.87494 W/K

heat flow: Q_dot = -2.64188 kW

Input Fluid:

Fluid: title = Water, id = 2079447880640

p = 101325 Pa

t = 56.28360825265497 °C

Output Fluid:

Fluid: title = Water, id = 2079515313584
p = 101325 Pa
t = 57.048733170735545 °C

Parameters:

dimensionless parameters:
number of transfer units:
NTU_1 = 0.072
NTU_2 = 0.074
heat capacity flow ratios:
R_1 = 1.029
R_2 = 0.972
dimensionless temperature change:
P_1 = 0.067
P_2 = 0.069

cell:12

heat exchanger:
id = 2079515321792
type: CrossFlowOneRow

Flows:

Flow 1:

Flow: id = 2079515324384
mass flow = 0.84886 kg/s
heat capacity flow: W_dot = 3555.02770 W/K
heat flow: Q_dot = 2.28153 kW

Input Fluid:

Fluid: title = Water, id = 2079515322560
p = 101420 Pa
t = 66.59968472573507 °C

Output Fluid:

Fluid: title = Water, id = 2079515323568
p = 101420 Pa
t = 65.95790837789167 °C

Flow 2:

Flow: id = 2079515323664
mass flow = 0.82504 kg/s
heat capacity flow: W_dot = 3451.86425 W/K
heat flow: Q_dot = -2.28153 kW

Input Fluid:

Fluid: title = Water, id = 2079515313584
p = 101325 Pa


```

        t = 57.048733170735545 °C
Output Fluid:
    Fluid: title = Water, id = 2079515313056
    p = 101325 Pa
    t = 57.70968983862889 °C

Parameters:

dimensionless parameters:
number of transfer units:
    NTU_1 = 0.072
    NTU_2 = 0.074
heat capacity flow ratios:
    R_1 = 1.030
    R_2 = 0.971
dimensionless temperature change:
    P_1 = 0.067
    P_2 = 0.069

cell:13

heat exchanger:
    id = 2079467416032
    type: CrossFlowOneRow
Flows:
Flow 1:
Flow: id = 2079467730160
    mass flow = 0.84974 kg/s
    heat capacity flow: W_dot = 3557.89828 W/K
    heat flow: Q_dot = 10.98388 kW

Input Fluid:
    Fluid: title = Water, id = 2079515323568
    p = 101420 Pa
    t = 65.95790837789167 °C
Output Fluid:
    Fluid: title = Water, id = 2079467727424
    p = 101420 Pa
    t = 62.87072989153546 °C

Flow 2:
Flow: id = 2079467722096
    mass flow = 0.83623 kg/s
    heat capacity flow: W_dot = 3497.96895 W/K
    heat flow: Q_dot = -10.98393 kW

Input Fluid:

```

```
Fluid: title = Water, id = 2079467720080
p = 101325 Pa
t = 20.0 °C
```

Output Fluid:

```
Fluid: title = Water, id = 2079467409216
p = 101325 Pa
t = 23.140069898615423 °C
```

Parameters:

dimensionless parameters:

number of transfer units:

```
NTU_1 = 0.072
```

```
NTU_2 = 0.073
```

heat capacity flow ratios:

```
R_1 = 1.017
```

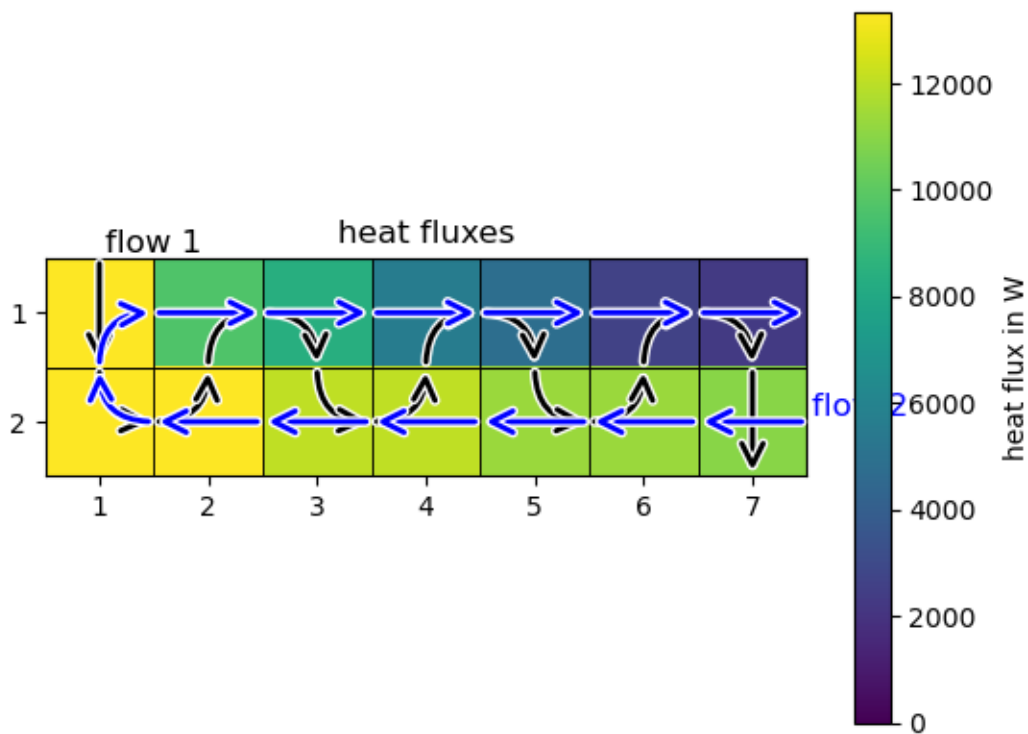
```
R_2 = 0.983
```

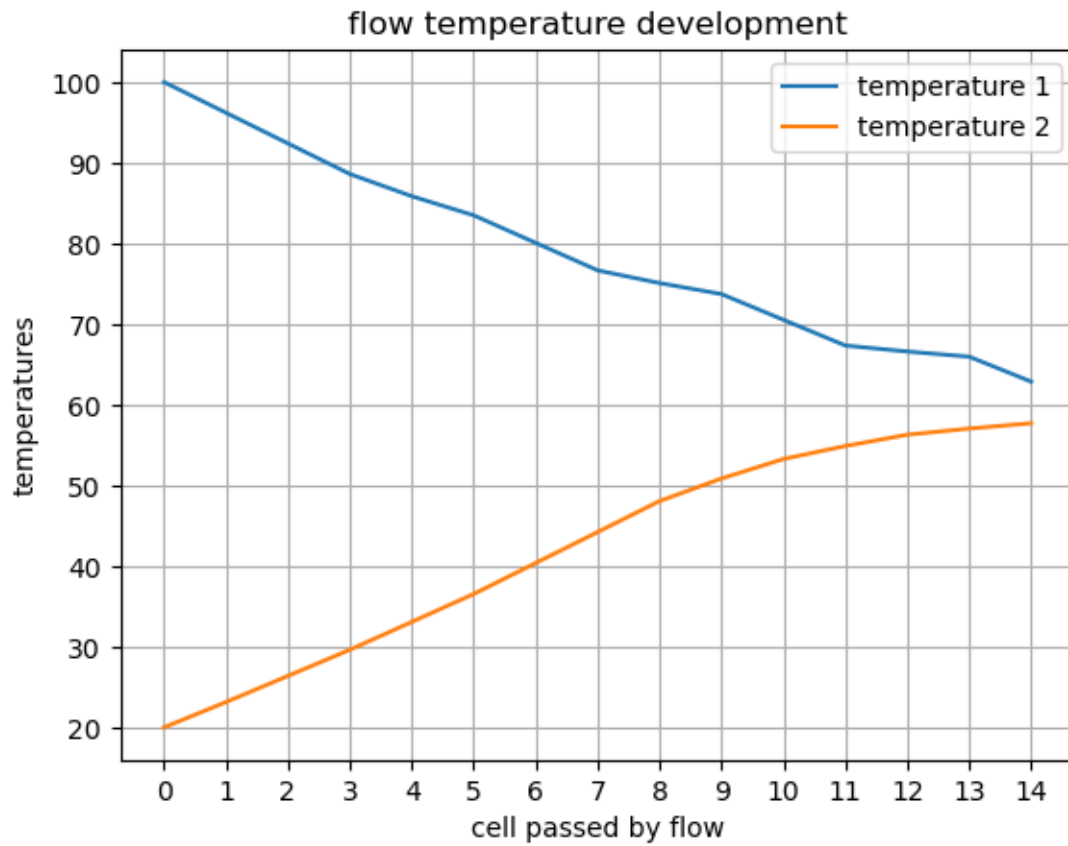
dimensionless temperature change:

```
P_1 = 0.067
```

```
P_2 = 0.068
```

```
[21]: ex_layout.vis_heat_flux()
      ex_layout.vis_flow_temperature_development()
```





Es zeigt sich, dass die Temperaturverläufe gut mit denen in der folgenden Abbildung übereinstimmen. Nur die Darstellungsart unterscheidet sich: Oben werden die Temperaturen über durchflossenen Zellen dargestellt, während unten die Darstellung über die Länge des Rohrbündelwärmeübertragers erfolgt.

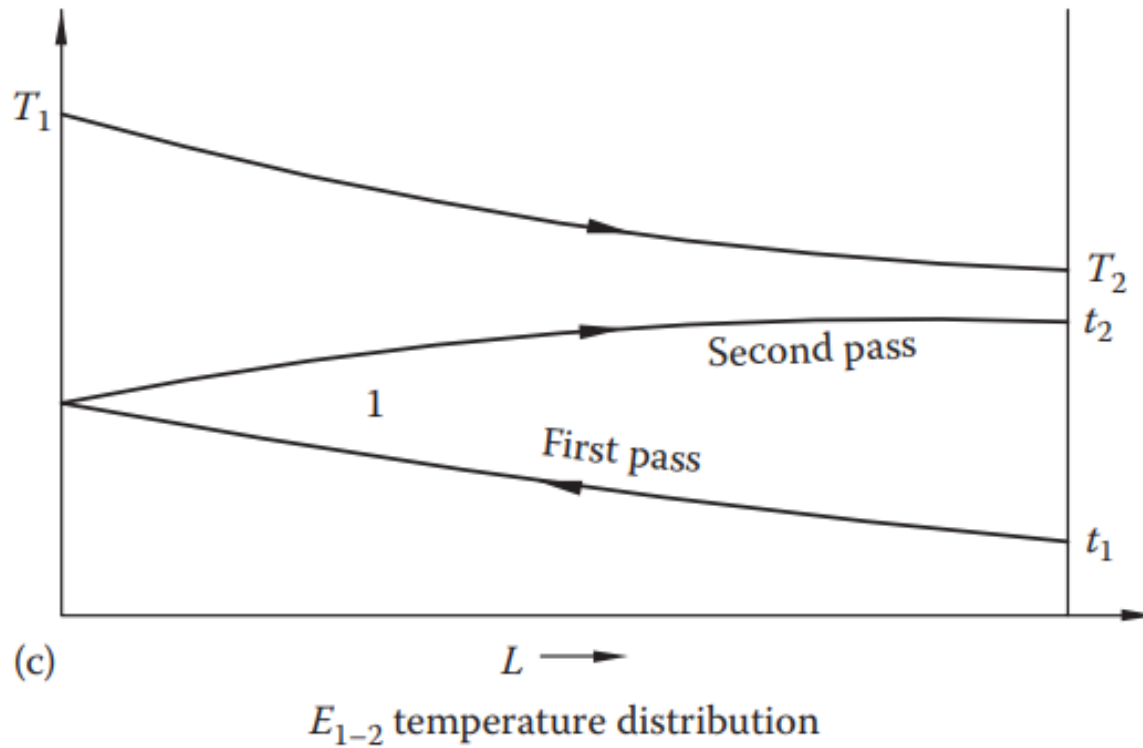


Abbildung 3: Temperaturverlauf der Fluidströme dargestellt über die Länge des Rohrbündelwärmeübertragers (entnommen aus [2], Seite 330)

Literatur

- [1] Peter Böckh. *Wärmeübertragung : Grundlagen und Praxis*. ger. 7. Aufl. 2017. Springer Berlin Heidelberg Imprint: Springer Vieweg, 2017. ISBN: 9783662554807. URL: <http://dx.doi.org/10.1007/978-3-662-55480-7> (siehe S. 1, 2).
- [2] T. Kuppan. *Heat Exchanger Design Handbook*. Second. Boca Raton, FL: CRC Press, 2013. ISBN: 978-1-4398-4213-3 (siehe S. 18, 36).