

Standard Wärmeübertrager

standard_heatexchangers.ipynb

März 2024

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

Über das Konfigurationsfile `pyfluids.json` kann das Einheitensystem (für die Fluiddaten) festgelegt werden. Standardmäßig werden alle Daten in SI-Einheiten definiert und verarbeitet.

1 Rohrbündelwärmeübertrager mit bekannter Charakteristik

1.1 Problemstellung

Im Folgenden soll die Anwendung des Codes für einen Wärmeübertrager mit gegebener Charakteristik bzw. Stromführung gezeigt werden. Basierend auf dem Beispiel 1, Kapitel 3.5.1 aus dem Skriptum zur kombinierten LV Technische Thermofluidodynamik (WS 2022/2023) von Michael Krieger, soll ein Rohrbündelwärmeübertrager mit einem äußeren und drei inneren Durchgängen (davon zwei im Gegenstrom) nachgerechnet werden.

1.1.1 Fluidströme

Folgende Daten zu den Fluidströmen sind bekannt:

Außenstrom:

- Fluid: Luft
- Massenstrom: 0,4 kg/s
- Eintrittstemperatur: 20 °C
- Druck: 1 atm

Innenstrom:

- Fluid: Wasser
- Massenstrom: 0,15 kg/s
- Eintrittstemperatur: 105 °C
- Druck: 5 bar

1.1.2 Geometriedaten

Vom Aufbau des Wärmeübertragers sind folgende Daten bekannt.

Der Innenstrom ist auf $n = 20$ Rohrleitungen (Innendurchmesser $D_i = 8mm$, Wandstärke $s = 2mm$) aufgeteilt, welche in zwei Schichten versetzt angeordnet sind. Die Rohre haben außen kleine Strukturen, um bei der Umströmung eine turbulente Grenzschicht zu gewährleisten. Die folgenden weiteren Daten zur Geometrie sind bekannt (vgl. dazu die Skizzen in Abbildung):

- $B = 0,35m$
- $H = 0,18m$
- $L_M = 1,0m$
- $L_E = 41mm$
- $d = 48mm$

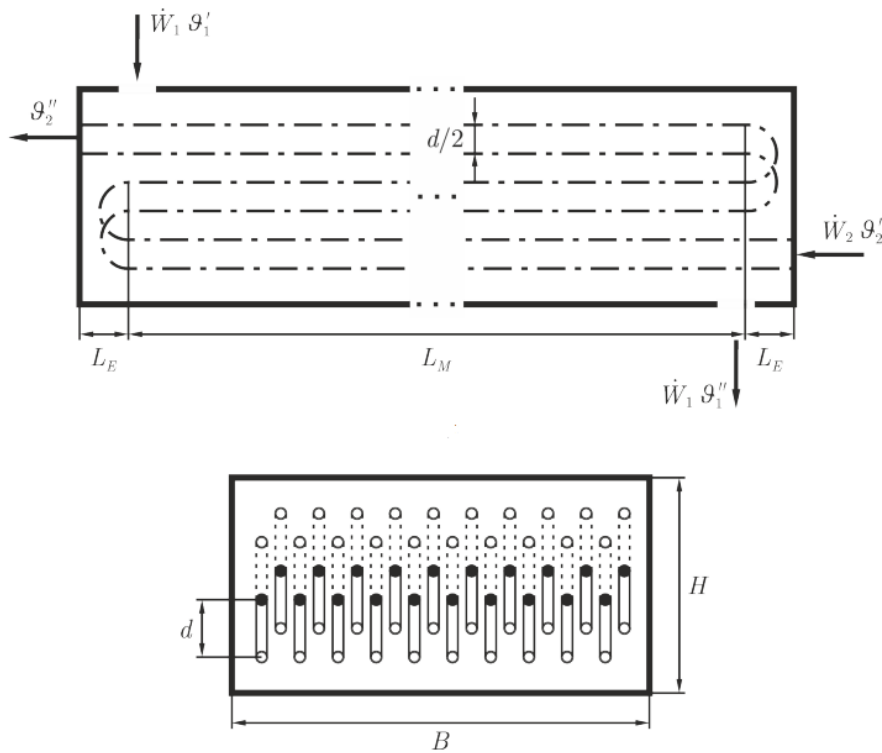


Abbildung 1: Prinzipskizzen des Wärmeübertragers
 (oben) Längsschnitt: nur Rohrmittellinien dargestellt
 (unten) Querschnitt: Rohrdurchströmung im Gegen-/Gleichstrom durch helle und dunkle Rohrquerschnitte angedeutet
 (entnommen aus [1], Seite 164)

Dadurch ist der verfügbare Bauraum zur Gänze ausgenutzt.

Der Wärmedurchgangskoeffizienten wird mit $k = 35W/(m/K)$ abgeschätzt.

1.2 Implementierung

1.2.1 Initialisierung der Fluidparameter

Ein Fluidstrom der Klasse `Flow` wird über ein Fluid und einen Durchsatz definiert. Der Durchsatz kann entweder über den Massenstrom `mass_flow` oder Volumenstrom `volume_flow` festgelegt werden. Die Fluidinstanz wird durch die Klasse `Fluid` initialisiert. Dazu sind einerseits der Fluidname - dieser muss in der Fluidliste des Moduls `pyfluids` hinterlegt sein - sowie der Zustand erforderlich. Standardmäßig wird der Zustand mit der Methode `ntp_state` auf NTP (Normal Temperature and Pressure) festgelegt. Über die Eigenschaften `temperature` und `pressure` kann der Zustand angepasst werden.

Die Implementierung eines Fluidstroms erfolgt durch Vorgabe eines Volumenstroms - dabei wird angenommen, dass der Strömungsquerschnitt konstant ist - sowie eines Fluids am Ein- und Auslass. Die Initialisierung über den Konstruktor legt das Eintrittsfluid fest. Weiterhin wird der Volumenstrom festgelegt oder, sofern der Massenstrom initialisiert wird, basierend auf den Daten des Eintrittsfluids berechnet. Das Austrittsfluid wird zunächst als Klon des Eintrittsfluids definiert. Die Fluidstrom-Kennzahlen wie Wärmekapazitätsstrom, Massenstrom und Änderung des Wärmestroms werden mit den Daten des Fluidobjekts `mean_fluid` berechnet und über gemittelte Zustandsdaten definiert.

Zunächst werden die Temperaturen der Fluidströme am Austritt gleich zu jener am Eintritt festgelegt. Dadurch ergeben sich zunächst die mittleren Fluidparameter basierend auf den Eintrittsparametern. Da in späteren Schritten die Austrittstemperaturen angepasst werden und mehrere Iterationen mit den angepassten mittleren Temperaturen durchgeführt werden, sind anfängliche Abweichungen der Materialparameter nicht problematisch.

Da zunächst die Fluidparameter am Ein- und Auslass sowie die gemittelten Parameter gleich sind, ist auch die Änderung des Wärmestroms (`heat_flow`) gleich 0. Wenn die Austrittstemperatur (`out_temperature`) geändert oder ein Druckverlust (`pressure_loss`) definiert wird, werden auch die entsprechenden Parameter angepasst. Die Variable `phase_change` wird verwendet, um zu überprüfen, ob sich die Phase zwischen Ein- und Auslass ändert. Ist dies der Fall, sollte die Änderung des Wärmestroms in der aktuellen Implementierung nicht mehr für weitere Berechnungen verwendet werden.

```
[2]: # defining outside flow
flow_outside = Flow(Fluid("Air",temperature = 20+273.15),mass_flow=0.4)
flow_outside
```

```
[2]: Flow: id = 2458521360304
      mass flow = 0.40000 kg/s
      heat capacity flow: W_dot = 402.45761 W/K
      heat flow: Q_dot = 0.00000 kW
```

```
Input Fluid:
  Fluid: title = Air, id = 2458543041664
  p = 101325 Pa
  t = 20.0 °C
```

```
Output Fluid:
  Fluid: title = Air, id = 2458521367696
```

```
p = 101325 Pa
t = 20.0 °C
```

```
[3]: # defining inside flow
flow_inside = Flow(Fluid("Water",pressure=5e5,temperature=105+273.
↪15),mass_flow=0.15)
flow_inside
```

```
[3]: Flow: id = 2458543041616
      mass flow = 0.15000 kg/s
      heat capacity flow: W_dot = 633.12137 W/K
      heat flow: Q_dot = 0.00000 kW
```

```
Input Fluid:
      Fluid: title = Water, id = 2458543044592
      p = 500000.0 Pa
      t = 105.0 °C
```

```
Output Fluid:
      Fluid: title = Water, id = 2458543042432
      p = 500000.0 Pa
      t = 105.0 °C
```

1.2.2 Initialisierung der Geometrie und Wärmeübertragungsfähigkeit

Für die Berechnung eines Wärmeübertrags ist in späterer Folge die Wärmeübertragungsfähigkeit relevant. Diese muss über die Klasse bzw. Subklasse **Part** übergeben werden. Die Instanz **Part** kann entweder direkt über die Parameter Wärmeübertragungsfähigkeit **heat_transferability** oder über die wärmeübertragende Fläche **heat_transfer_area** und den Wärmedurchgangskoeffizienten **heat_transfer_coefficient** definiert werden.

Die aktuelle Implementierung beruht darauf, dass der Wärmedurchgangskoeffizient abgeschätzt und vorgegeben wird. Zur Berechnung sind einerseits Informationen über Strömungen der einzelnen Fluidströme sowie die Geometrie des Wärmeübertragers notwendig. Aufgrund der objektorientierten Programmierung können ohne Änderung der Implementierung neue Funktionen oder (Sub-)Klassen implementiert werden, um Wärmedurchgangskoeffizienten auch zu berechnen und nicht nur den abgeschätzten vorzugeben.

In diesem Beispiel wird die Länge eines Rohrs manuell kalkuliert. Damit kann zunächst ein einzelnes glattes Rohr **StraightPipe** über die Eigenschaften **diameter_in**, **diameter_out** und **length** definiert werden. Ein Rohrbündel wird über **PipeLayout** initialisiert und erfordert ein Objekt der Klasse **Pipe** sowie die Anzahl der Rohre. Damit kann die wärmeübertragende Fläche aus der Oberfläche eines einzelnen Rohrs und deren Anzahl berechnet werden. Der Wärmedurchgangskoeffizient wird über **heat_transfer_coefficient** gesetzt. Damit ist die Wärmeübertragungsfähigkeit des Rohrbündels vollständig definiert. Da wie bereits beschrieben der Wärmedurchgangskoeffizienten vorgegeben wurde, muss die äußere Geometrie des Wärmeübertragers im vorliegenden Beispiel nicht mehr definiert werden.

```
[4]: l_m,l_e,d = 1,41e-3,48e-3
length = 3*l_m+2*l_e +d*pi

pipe = StraightPipe(diameter_in=11.9e-3,diameter_out=12e-3,length=length)
pipe_layout = PipeLayout(pipe,20)
pipe_layout.heat_transfer_coefficient = 35
pipe_layout
```

```
[4]: part:
      id = 2458543045888
      typ: PipeLayout

geometric properties:
  pipe type: StraightPipe
  length = 3.23280 m
  diameter: in = 0.01190 m, out = 0.01200 m
  number of pipes: 20
  pipe pattern: square

hydraulic properties:
  flow area = 0.002224 m^2

thermic properties:
  heat transferability: 84.956 W/K
  heat transfer area: 2.4273 m^2
  heat transfer coefficient: 35.00 W/(m^2 K)
```

1.2.3 Berechnung des Wärmeübertrages

Wenn die Austrittstemperatur eines Wärmeübertragers berechnet werden soll, für den bereits implementierte Formeln zur Verfügung stehen, kann dies über die Klasse **HeatExchanger** und ihre Subklassen erfolgen. Dazu sind die beiden Fluidströme der Klasse **Flow** sowie ein Bauteil der Klasse **Part** erforderlich.

Der Wärmeübertragertyp wird also über die entsprechende Subklasse definiert. Zum Beispiel für einen reinen Gegenstromwärmetauscher **CounterCurrentFlow** oder für einen Rohrbündelwärmeübertrager mit einem äußeren und drei inneren Durchgängen (zwei im Gegenstrom) **OneOuterThreeInnerTwoCounterFlow**.

Die Berechnung des Wärmeübertrages mit der Klasse **HeatExchanger** erfolgt nun gemäß folgenden Schritten. Für eine genauere Beschreibung sei auf die Dokumentation im Code verwiesen. Durch die **Flow**-Instanzen sind auch die Wärmekapazitätsströme der Fluide **heat_capacity_flow** definiert. Die Wärmeübertragungsfähigkeit **heat_transferability** ergibt sich aus der Vorgabe der **Part**-Instanz. Damit können zunächst die dimensionslosen Parameter, Anzahl der Transfer Units (NTU) **ntu** und dimensionslose Übertragungsfähigkeiten **r** berechnet werden. Die dimensionslosen Temperaturänderungen **p** werden durch den jeweiligen Typ entsprechenden Formeln - die z.B. aus dem VDI-Wärmeatlas entnommen wurden - kalkuliert.

Mit den dimensionslosen Temperaturänderungen und den Temperaturen der Eintrittsfluide

`flow_1.in_fluid.temperature` bzw. `flow_2.in_fluid.temperature` des entsprechenden Fluidstroms können die Austrittstemperaturen mithilfe der Methode `_calc_output` in 5 Iterationen berechnet werden. Nach jeder Iteration werden die Austrittstemperaturen der Fluidströme und damit auch die anderen Strom- und Fluidparameter angepasst. Danach wird auch der übertragene Wärmestrom (sofern es zu keiner Phasenänderung kommt) entsprechend neu berechnet.

Standardmäßig ist das Attribut `auto_adjust` der Klasse `HeatExchanger` auf `True` gesetzt. Dadurch wird beim Ausführen der Methoden `str()` bzw. `print()` die Methode `_calc_output` automatisch aufgerufen. Um die Initialisierung noch ohne berechneten Ausgangstemperaturen auszugeben - um beispielsweise eine Abschätzung der dimensionslosen Temperaturänderungen zu erhalten - oder die Anzahl der Iterationen zu beeinflussen, muss das Attribut `auto_adjust` auf `False` gesetzt und die Methode `_calc_output` manuell ausgeführt werden.

```
[5]: ex = OneOuterThreeInnerTwoCounterFlow(flow_outside,flow_inside,pipe_layout)
ex.auto_adjust = False
ex
```

```
[5]:
heat exchanger:
    id = 2458543044352
    type: OneOuterThreeInnerTwoCounterFlow
Flows:
Flow 1:
Flow: id = 2458521360304
    mass flow = 0.40000 kg/s
    heat capacity flow: W_dot = 402.45761 W/K
    heat flow: Q_dot = 0.00000 kW

Input Fluid:
    Fluid: title = Air, id = 2458543041664
    p = 101325 Pa
    t = 20.0 °C
Output Fluid:
    Fluid: title = Air, id = 2458521367696
    p = 101325 Pa
    t = 20.0 °C

Flow 2:
Flow: id = 2458543041616
    mass flow = 0.15000 kg/s
    heat capacity flow: W_dot = 633.12137 W/K
    heat flow: Q_dot = 0.00000 kW

Input Fluid:
    Fluid: title = Water, id = 2458543044592
    p = 500000.0 Pa
    t = 105.0 °C
Output Fluid:
```

```
Fluid: title = Water, id = 2458543042432
p = 500000.0 Pa
t = 105.0 °C
```

Parameters:

```
dimensionless parameters:
number of transfer units:
    NTU_1 = 0.211
    NTU_2 = 0.134
heat capacity flow ratios:
    R_1 = 0.636
    R_2 = 1.573
dimensionless temperature change:
    P_1 = 0.179
    P_2 = 0.114
```

```
[6]: ex._calc_output(5)
      ex
```

```
[6]: heat exchanger:
      id = 2458543044352
      type: OneOuterThreeInnerTwoCounterFlow
Flows:
Flow 1:
Flow: id = 2458521360304
      mass flow = 0.38960 kg/s
      heat capacity flow: W_dot = 392.09788 W/K
      heat flow: Q_dot = -6.11901 kW

Input Fluid:
      Fluid: title = Air, id = 2458543041664
      p = 101325 Pa
      t = 20.0 °C
Output Fluid:
      Fluid: title = Air, id = 2458521367696
      p = 101325 Pa
      t = 35.605706717043915 °C

Flow 2:
Flow: id = 2458543041616
      mass flow = 0.15055 kg/s
      heat capacity flow: W_dot = 634.57660 W/K
      heat flow: Q_dot = 6.11909 kW

Input Fluid:
```

```

Fluid: title = Water, id = 2458543044592
p = 500000.0 Pa
t = 105.0 °C
Output Fluid:
Fluid: title = Water, id = 2458543042432
p = 500000.0 Pa
t = 95.35740581452308 °C

```

Parameters:

```

dimensionless parameters:
number of transfer units:
    NTU_1 = 0.217
    NTU_2 = 0.134
heat capacity flow ratios:
    R_1 = 0.618
    R_2 = 1.618
dimensionless temperature change:
    P_1 = 0.184
    P_2 = 0.113

```

1.2.4 Alternativer Wärmeübertrager Typ

Ein Wärmeübertrager mit reinem Gegenstrom und gleicher Wärmeübertragungsfähigkeit würde im gegebenen Fall zu einer nahezu gleichen Austrittstemperatur führen.

Beim Erhöhen der Übertragungsfähigkeit würde der Gegenstrom Wärmeübertrager einen besseren Wirkungsgrad ergeben.

```

[7]: ex = CounterCurrentFlow(flow_outside,flow_inside,pipe_layout)
ex

```

```

[7]:
heat exchanger:
    id = 2458543042192
    type: CounterCurrentFlow
Flows:
Flow 1:
Flow: id = 2458521360304
    mass flow = 0.38956 kg/s
    heat capacity flow: W_dot = 392.06020 W/K
    heat flow: Q_dot = -6.14130 kW

Input Fluid:
    Fluid: title = Air, id = 2458543041664
    p = 101325 Pa
    t = 20.0 °C
Output Fluid:

```


Fluid: title = Air, id = 2458521367696
p = 101325 Pa
t = 35.66405423629652 °C

Flow 2:

Flow: id = 2458543041616
mass flow = 0.15056 kg/s
heat capacity flow: \dot{W} = 634.58189 W/K
heat flow: \dot{Q} = 6.14138 kW

Input Fluid:

Fluid: title = Water, id = 2458543044592
p = 500000.0 Pa
t = 105.0 °C

Output Fluid:

Fluid: title = Water, id = 2458543042432
p = 500000.0 Pa
t = 95.32236458886882 °C

Parameters:

dimensionless parameters:

number of transfer units:

NTU₁ = 0.217

NTU₂ = 0.134

heat capacity flow ratios:

R₁ = 0.618

R₂ = 1.619

dimensionless temperature change:

P₁ = 0.184

P₂ = 0.114

Literatur

- [1] Michael Krieger. *Skriptum zur kombinierten LV Technische Thermo-fluiddynamik*. 2023 (siehe S. 2).