

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STROJNÍ
ÚSTAV PROCESNÍ A ZPRACOVATELSKÉ TECHNIKY

BILANCOVÁNÍ A MODELOVÁNÍ PROCESŮ

BAKALÁŘSKÁ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kříž** Jméno: **Daniel** Osobní číslo: **459564**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav procesní a zpracovatelské techniky**
Studijní program: **Teoretický základ strojního inženýrství**
Studijní obor: **bez oboru**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Bilancování a modelování procesů

Název bakalářské práce anglicky:

Process balancing and modeling

Pokyny pro vypracování:

Provedte literární rešerši k problematice bilancování systémů se zaměřením na bilance hmotnostní a entalpické s použitím volně dostupných i komerčních systémů. Doplněte tuto literární rešerši systémy umožňující detailní modelování procesů probíhajících v aparátech, procesů přenosu hybnosti, tepla a hmoty. Demonstrujte použití vámi vybraných systémů vhodných pro bilancování a modelování na vhodných procesech/aparátech (například odparka s MVR, odparka s TVR, odparka s tepelným čerpadlem, chlazení varem, sušení, ...).

Seznam doporučené literatury:

Na základě provedené literární rešerše s využitím elektronických informačních zdrojů z <http://knihovny.cvut.cz/katalogy-a-databaze/prehled-databazi/katalog-informacnich-zdroju>.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Martin Dostál, Ph.D., ústav procesní a zpracovatelské techniky FS

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **23.04.2019**

Termín odevzdání bakalářské práce: **09.08.2019**

Platnost zadání bakalářské práce: _____

Ing. Martin Dostál, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Tomáš Jirout, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Michael Valášek, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Děkuji panu Martinu Dostálovi za velice příjemnou spolupráci, cenné rady a v neposlední řadě za vstřícné jednání během zpracovávání této bakalářské práce. Dále bych rád poděkoval své rodině a přátelům za podporu během studia.

Anotační list

Jméno autora:	Daniel Kříž
Název:	Bilancování a modelování procesů
Anglický název:	Process balancing and modeling
Akademický rok:	2018/2019
Studijní obor:	Teoretický základ strojního inženýrství
Ústav:	Ústav procesní a zpracovatelské techniky
Vedoucí BP:	Ing. Martin Dostál, Ph.D.
Bibilografické údaje:	počet stran: 71 počet obrázků: 52 počet rovnic: 30 počet tabulek: 4 počet příloh: 1 x CD
Klíčová slova:	bilancování, modelování, simulace, odparka, odpařování, vakuové chlazení, bilanční programy, simulační programy, stacionární model, dynamický model
Keywords:	balance, modeling, simulation, evaporator, evaporation, vacuum cooling, balance software, simulation software, steady-state model, dynamic model
Anotace:	Bakalářská práce se zabývá řešením komerčních i volně dostupných programů pro tvorbu stacionárních i dynamických modelů. Použití vybraných programů je podrobně ukázáno na modelu expanzní cirkulační odparky a procesu vakuového chlazení.
Annotation:	This bachelor thesis deals with research of Proprietary and Freeware/Open-Source software for steady-state and dynamic modeling. The use of selected programs is shown on model of flash evaporator with circulation and vacuum cooling process.

Obsah

Úvod	7
1 Typy modelování	8
1.1 Softwarově orientované modelování	8
1.1.1 Rovnicové	8
1.1.2 Objektové	8
1.2 Blokově orientované modelování	8
2 Software pro bilance a modelování procesů	10
2.1 Komerční software	10
2.1.1 Simulink	10
2.1.2 gPROMS	10
2.1.3 Aveva SimSci PRO/II	11
2.1.4 Dymola	12
2.2 Volně dostupný software	13
2.2.1 Xcos	14
2.2.2 OpenModelica	14
2.2.3 Cantera	15
2.2.4 COCO Simulator	15
2.3 Shrnutí	16
3 Bilance odparky	18
3.1 Princip funkce	18
3.2 Analytický výpočet expanzního separátoru	20
3.3 Komponenty modelu	22
3.3.1 Flash	22
3.3.2 Heater/Cooler	23
3.3.3 Ostatní použité komponenty	24
3.4 Tvorba modelu odparky v editoru COFE	25
3.4.1 Uživatelské prostředí editoru, definice TD systému	25
3.4.2 Sestavení PFD odparky	27
3.4.3 Nastavení vstupních a požadovaných hodnot	32
3.4.4 Spuštění simulace a zobrazení výsledků	34
3.4.5 Výsledky bilance	38
4 Simulace vakuového chlazení	41
4.1 Princip funkce	41
4.2 Matematický model vakuového chlazení kapalin	41
4.3 Matematické modely netěsností	43

4.3.1	Model trubky s konstantním průřezem, laminární proudění	43
4.3.2	Model trubky s konstantním průřezem, turbulentní proudění	45
4.3.3	Model konvergentní trysky, nevázké stlačitelné proudění	49
4.4	Tvorba simulace v programu OpenModelica	50
4.4.1	Uživatelské rozhraní a založení modelu	50
4.4.2	Zápis kódu	53
4.4.3	Kontrola a nastavení simulace	56
4.4.4	Zobrazení výsledků a jejich export	57
4.4.5	Blokový model	59
4.4.6	Výsledky simulace	59
5	Diskuse	61
5.1	Rešerše	61
5.2	Tvorba stacionárního modelu	61
5.3	Tvorba dynamického modelu	62
6	Závěr	64
	Seznam použitých symbolů	65
	Seznam použité literatury	67
	Prohlášení	71

Úvod

Při řešení úloh procesní techniky, např. návrh výrobních a zpracovatelských linek, nebo detailní řešení dílčích zařízení a systémů, je nutné provádět mnoho matematických operací. Zejména u složitějších zařízení může být jejich analytické řešení zdoluhavé, velice obtížné, nebo prozatím nemožné. Proto se při projektování procesních zařízení využívají počítačové simulace a numerická řešení. Pro různé aplikace od základních bilancí vstupních látek a produktů až po dynamické řešení energetických přeměn existuje nepřehledné množství počítačových programů a nástrojů, které tvorbu projektů usnadňují a zefektivňují.

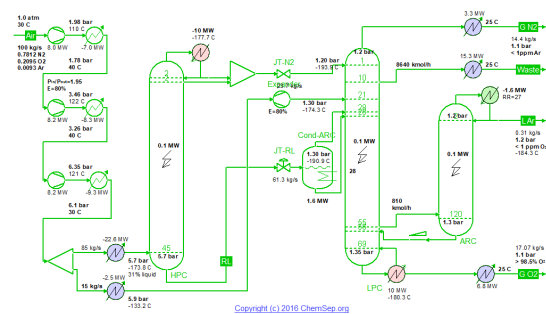
Cílem této práce je provést rešerši těchto nástrojů pro bilancování a simulování procesů, rozdělit je podle dostupnosti licence a způsobu tvorby modelu a názorně ukázat, jak se s těmito nástroji pracuje.

V nejzákladnější formě lze řešení matematického modelu zařízení vytvořit v libovolném programovacím jazyce, bez využití specializovaných nástrojů. Tímto způsobem se ale v této práci zabývat nebudeme, jelikož při převádění matematického popisu do zdrojového kódu by došlo ke ztrátě názornosti. Pro naše účely bude tedy vhodnější zaměřit se na zmíněné specializované nástroje (ale i specializované programovací jazyky), ve kterých bývá tvorba modelu relativně snadná a intuitivní.

Sestavování modelu funguje často na bázi grafického znázornění jednotlivých komponent linky, s předdefinovanými vnitřními funkcemi, které jsou navzájem propojené proudy (informačními, energetickými, látkovými, ...). Pomocí těchto procesních vývojových schémat (Process Flow Diagrams - PFDs), obrázek 1b, lze velice názorně popsat funkci daného zařízení, následně snadno měnit různé vstupní parametry a optimalizovat tak celý proces.



(a) Zařízení na separaci vzduchu (Adele, USA) (https://www.linde-engineering.com/en/images/s_20180704_119319_8928_tcm19-482244.png)



(b) PFD schéma zařízení na separaci vzduchu (COCO Simulator) (https://www.cocosimulator.org/downcs.php?dl=CScasebook_ASU.fsd)

Obrázek 1: Reálné zařízení a PFD schéma

1 Typy modelování

Abychom mohli porozumět funkci modelovacích systémů, je potřeba nejdříve definovat jednotlivé způsoby tvorby matematického modelu. Základní rozdělení na softwarově a blokově orientované modelování uvádí Ospino et al. (2017, podle Smith a Corripio, 2006). První skupina je dále rozdělena na rovnicově a objektově orientované modelování (Ospino et al., 2017, podle Bogusch a Marquardt, 1997).

1.1 Softwarově orientované modelování

1.1.1 Rovnicové

Ospino et al. (2017, podle Smith a Corripio, 2006) uvádí, že rovnicové modelování je založeno na matematickém popisu a numerickém řešení diferenciálních rovnic. Toto řešení nejprve musíme podle Bogusch a Marquardt (1997) připravit na papíře a následně převést do programovacího jazyka. Zvolený software pak provede vlastní výpočet. Takto vytvořené modely je složitější upravit, protože je celý model vytvořen pro jedno konkrétní řešení. Chceme-li tedy změnit některé parametry simulace, musíme vytvořit nový numerický model.

Na druhou stranu podle Ishii a Otto (2007) může být rovnicově orientované modelování oproti blokově orientovanému (modulárnímu) výhodnější, protože pracujeme přímo s matematickou podstatou modelu. Tento způsob může být vhodnější a z hlediska nároků na výpočetní výkon efektivnější.

1.1.2 Objektové

Phillips (2010) ve své odborné práci uvádí, že objekty jsou entity, které obsahují data a přidružené chování, v našem případě matematické operace. Tyto objekty jsou sdružovány ve třídách a komunikují s dalšími pomocí rozhraní. To znamená, že objekt vezme vstupní data a svými definovanými operacemi je změní na data výstupní, které přes rozhraní odešle do jiného objektu. Podle Bogusch a Marquardt (1997) se využitím tohoto typu modelování výrazně zjednoduší reprezentace modelu, neboť nedefinujeme přímo matematické operace, ale v důsledku můžeme psát přímo rovnice popisující zkoumaný jev. Z toho důvodu je tvorba modelu jednodušší a již existující modely lze relativně snadno upravovat. Tímto způsobem je možné vytvořit komponenty pro blokově orientované modelování, viz obrázek 2.

My tuto metodu využijeme při simulaci vakuového chlazení.

1.2 Blokově orientované modelování

Objektové modelování je sice svým principem velmi jednoduché, ale stále se jedná o programování. Od uživatele je tak bezpodmínečně vyžadována znalost daného


```

1 function Modelica.Thermal.FluidHeatFlow.Medium "Automatically generated record constructor for Modelica.Thermal.FluidHeatFlow.Medium"
2   input Real rho(quantity = "Density", unit = "kg/m^3", displayUnit = "g/cm3", min = 0.0) = 1.0;
3   input Real cp(quantity = "SpecificHeatCapacity", unit = "J/(kg.K)", min = 0.0);
4   input Real cv(quantity = "SpecificHeatCapacity", unit = "J/(kg.K)", min = 1.0);
5   input Real lambda(quantity = "ThermalConductivity", unit = "W/(m.K)", min = 1.0);
6   input Real nu(quantity = "KinematicViscosity", unit = "m^2/s", min = 0.0) = 1.0;
7   output Medium base;
8 end Modelica.Thermal.FluidHeatFlow.Medium;
9
10 class Modelica.Thermal.FluidHeatFlow.Components.IsolatedPipe "Pipe without heat exchange"
11   parameter Real medium_rho(quantity = "Density", unit = "kg/m^3", displayUnit = "g/cm3", min = 0.0) = 1.0 "Density";
12   parameter Real medium_cp(quantity = "SpecificHeatCapacity", unit = "J/(kg.K)", min = 1.0) "Specific heat capacity at constant pressure";
13   parameter Real medium_cv(quantity = "SpecificHeatCapacity", unit = "J/(kg.K)", min = 1.0) "Specific heat capacity at constant volume";
14   parameter Real medium_lambda(quantity = "ThermalConductivity", unit = "W/(m.K)", min = 1.0) "Thermal conductivity";
15   parameter Real medium_nu(quantity = "KinematicViscosity", unit = "m^2/s", min = 0.0) = 1.0 "Kinematic viscosity";
16   parameter Real rho(quantity = "Mass", unit = "kg", min = 0.0, start = 1.0) "Mass of medium";
17   parameter Real T(quantity = "ThermodynamicTemperature", unit = "K", displayUnit = "degC", min = 0.0, start = 293.15, nominal = 300.0) "Initial temperature of medium";
18   parameter Boolean offset = false "Initial temperature queue value of flow";
19   parameter Real tapLen = 0.0, max = 1.0 = 1.0 "Distance temperature of heatPort between inlet and outlet temperature";
20   Real Q(quantity = "Power", unit = "W", displayUnit = "kW") "Pressure drop a-b";
21   Real V_flow(quantity = "VolumeFlowRate", unit = "m^3/s") "Volume flow a-b";
22   Real Q_flow(quantity = "Power", unit = "W") "Heat exchange with ambient";
23
24   output Real T(quantity = "ThermodynamicTemperature", unit = "K", displayUnit = "degC", min = 0.0, start = 70, fixed = false, nominal = 300.0) "Outlet temperature of medium";
25   output Real T(quantity = "ThermodynamicTemperature", unit = "K", displayUnit = "degC", min = 0.0, start = 293.15, nominal = 300.0) "Temperature at flowPort_a";
26   output Real T(quantity = "ThermodynamicTemperature", unit = "K", displayUnit = "degC", min = 0.0, start = 293.15, nominal = 300.0) "Temperature at flowPort_b";
27   protected Real h(quantity = "SpecificEnthalpy", unit = "J/kg", start = medium_cp * T);
28   protected Real T_a(quantity = "ThermodynamicTemperature", unit = "K", displayUnit = "degC", min = 0.0, start = 293.15, nominal = 300.0) "Temperature relevant for heat exchange with ambient";
29   parameter Real flowPort_a_medium_rho(quantity = "Density", unit = "kg/m^3", displayUnit = "g/cm3", min = 0.0) = medium_rho "Density";
30   parameter Real flowPort_a_medium_cp(quantity = "SpecificHeatCapacity", unit = "J/(kg.K)", min = 1.0) = medium_cp "Specific heat capacity at constant pressure";
31   parameter Real flowPort_a_medium_cv(quantity = "SpecificHeatCapacity", unit = "J/(kg.K)", min = 1.0) = medium_cv "Specific heat capacity at constant volume";
32   parameter Real flowPort_a_medium_lambda(quantity = "ThermalConductivity", unit = "W/(m.K)", min = 1.0) = medium_lambda "Thermal conductivity";
33   parameter Real flowPort_a_medium_nu(quantity = "KinematicViscosity", unit = "m^2/s", min = 0.0) = medium_nu "Kinematic viscosity";
34   Real flowPort_a_p(quantity = "Pressure", unit = "Pa", displayUnit = "bar");
35   Real flowPort_a_h(quantity = "SpecificEnthalpy", unit = "J/kg");
36   Real flowPort_a_Q_flow(quantity = "HeatFlowRate", unit = "W");
37   Real flowPort_a_V_flow(quantity = "VolumeFlowRate", unit = "m^3/s");
38   parameter Real flowPort_b_medium_rho(quantity = "Density", unit = "kg/m^3", displayUnit = "g/cm3", min = 0.0) = medium_rho "Density";
39   parameter Real flowPort_b_medium_cp(quantity = "SpecificHeatCapacity", unit = "J/(kg.K)", min = 1.0) = medium_cp "Specific heat capacity at constant pressure";
40   parameter Real flowPort_b_medium_cv(quantity = "SpecificHeatCapacity", unit = "J/(kg.K)", min = 1.0) = medium_cv "Specific heat capacity at constant volume";
41   parameter Real flowPort_b_medium_lambda(quantity = "ThermalConductivity", unit = "W/(m.K)", min = 1.0) = medium_lambda "Thermal conductivity";
42   parameter Real flowPort_b_medium_nu(quantity = "KinematicViscosity", unit = "m^2/s", min = 0.0) = medium_nu "Kinematic viscosity";
43   Real flowPort_b_p(quantity = "Pressure", unit = "Pa", displayUnit = "bar");
44   Real flowPort_b_h(quantity = "SpecificEnthalpy", unit = "J/kg");
45   Real flowPort_b_Q_flow(quantity = "HeatFlowRate", unit = "W");
46   Real flowPort_b_V_flow(quantity = "VolumeFlowRate", unit = "m^3/s");
47   parameter Real V_flowLaminar(quantity = "VolumeFlowRate", unit = "m^3/s", min = 1e-06, start = 0.1) "Laminar volume flow";
48   parameter Real dpLaminar(quantity = "Pressure", unit = "Pa", displayUnit = "bar", start = 0.1) "Laminar pressure drop";
49   parameter Real V_flowNominal(quantity = "VolumeFlowRate", unit = "m^3/s", start = 1.0) "Nominal volume flow";
50   parameter Real dpNominal(quantity = "Pressure", unit = "Pa", displayUnit = "bar", start = 1.0) "Nominal pressure drop";
51   parameter Real frictionLosses(min = 0.0, max = 1.0) = 0.0 "Part of friction losses fed to medium";
52   Real pressureDrop(quantity = "Pressure", unit = "Pa", displayUnit = "bar");
53   Real Q_friction(quantity = "VolumeFlowRate", unit = "m^3/s");
54   Real Q_frictionLosses(quantity = "Power", unit = "W", displayUnit = "kW");
55   protected parameter Real dpNominal(quantity = "Pressure", unit = "Pa", displayUnit = "bar") = dpLaminar + V_flowLaminar / V_flowLaminar;
56   protected parameter Real k(quantity = "Friction", unit = "1/m", fixed = false);
57   parameter Real h_g(quantity = "Length", unit = "m", start = 0.0) "Geodesic height (height difference from flowPort_a to flowPort_b)";
58   initial algorithm
59     assert(V_flowNominal > V_flowLaminar, "SimpleFriction: V_flowNominal has to be > V_flowLaminar!");
60     assert(dpNominal > dpLaminar, "SimpleFriction: dpNominal has to be > dpLaminar/V_flowLaminar*V_flowNominal!");
61     k := (dpNominal - dpLaminar) / (V_flowNominal - V_flowLaminar) * 2.0;
62   end
63   volumeFlow = V_flow;
64   dp = pressureDrop + 5.306459595959596 * medium_rho * h_g;
65   Q_flow = Q_friction;
66   dp = flowPort_a_p - flowPort_b_p;
67   V_flow = flowPort_a_V_flow / medium_rho;
68   T_a = flowPort_a_T / medium_cp;
69   T_b = flowPort_b_T / medium_cp;
70   dT = if noEvent(V_flow > 0.0) then T - T_a else T_b - T;
71   T_g = T - /heat/(rho*(V_flow) * (1.0 - tapT) * dT);
72   flowPort_a_T = flowPort_b_T + h_g * dT;
73   flowPort_a_h_flow = flowPort_b_h_flow + Q_flow * m * medium_cv * dT;
74   flowPort_a_h_flow = semiLinear(flowPort_b_h_flow, flowPort_a_h_flow, flowPort_b_h_flow);
75   if volumeFlow > V_flowLaminar then
76     pressureDrop = dpLaminar + volumeFlow / V_flowLaminar * k * (volumeFlow - V_flowLaminar) ^ 2.0;
77   else
78     pressureDrop = dpLaminar + volumeFlow / V_flowLaminar * k * (volumeFlow - V_flowLaminar) ^ 2.0;
79   end if;
80   Q_friction = frictionLosses * volumeFlow * pressureDrop;
81   flowPort_a_h_flow = 0.0;
82   flowPort_a_Q_flow = 0.0;
83   flowPort_b_h_flow = 0.0;
84   flowPort_b_Q_flow = 0.0;
85 end Modelica.Thermal.FluidHeatFlow.Components.IsolatedPipe;

```

Obrázek 2: Zdrojový kód izolované trubky v jazyce Modelica

programovacího jazyka.

Toto omezení částečně odpadá u blokového modelování. Podle některých zdrojů (např. Ishii a Otto (2007)) je nazýváno též modulární a podle Bogusch a Marquardt (1997) se zaměřuje přímo na tvorbu PFD. V tomto případě uživatel pracuje s grafickou reprezentací objektů i celých tříd, tzv. komponenty. Ty mají definované matematické operace nebo celé soubory operací. Komponenty tak mohou být naprogramovány k simulaci dílčích částí (trubky, výměníky, separátory, apod.), které jsou následně spojovány do výsledného modelu zařízení. Grafické zobrazení komponenty zmíněné výše, je na obrázku 3. Jednotlivé komponenty mohou být vytvořeny buď rovnicově, nebo objektivě orientovaným způsobem.

isolatedPipe1



Obrázek 3: Komponenta Izolovaná trubka (OpenModelica)

2 Software pro bilance a modelování procesů

2.1 Komerční software

Možnosti numerických výpočtů a simulací se samozřejmě nevztahují pouze na oblast procesní techniky, ale lze je využít napříč všemi technickými i netechnickými obory. To znamená, že na vývoji těchto simulačních systémů musí spolupracovat mnoho odborníků, kteří vytvářejí jednotlivé knihovny komponent a zároveň se starají o správnost kódu. Jelikož je potřeba zmíněné služby zaplatit, jsou tyto systémy velmi nákladné.

Tato podkapitola obsahuje výběr z dostupných komerčních programů. Pro naše účely se ale zaměříme spíše na volně dostupný software uvedený dále, ve stejnojmenné podkapitole.

2.1.1 Simulink

Simulink je jedním z mnoha toolboxů, které jsou obsaženy ve velmi všestranném výpočetním prostředí Matlab. Jedná se blokově (modulárně) orientovaný software. Obsahuje mnoho knihoven s bloky, díky kterým je možné sestavit různé modely od sekvence matematických operací, přes úlohy automatického řízení, až po návrh zařízení procesní a zpracovatelské techniky.

Ukázce možností a základům ovládání prostředí Simulink se ve své odborné práci věnuje Karris (2006). V této publikaci názorně předvádí, jak z jednotlivých bloků sestavit model např. diferenciálních rovnic. Výsledky simulace jsou pak porovnány s modelem vytvořeným v objektově orientované jazyce (Matlab). Jelikož všechny modulárně orientované simulační programy pracují na podobném principu jako Simulink, je odborná práce, kterou sepsal Karris (2006) vhodná k pochopení základních principů a funkcí všech těchto programů.

Skladbou knihoven je Simulink vhodný spíše k elektrotechnickým výpočtům, nebo simulacím obvodů automatického řízení. Avšak libovolnou komponentu či knihovnu lze vytvořit v jazyce Matlab a pak do Simulink jednoduše nahrát. V důsledku toho se v tomto prostředí tvoří simulace i pro energetiku a procesní a zpracovatelskou techniku. Například Yigit a Selamet (2016) se zabývají tvorbou matematického modelu a následné simulace vysokotlakého PEM elektrolyzérů pro výrobu vodíku.

2.1.2 gPROMS

Na této platformě (general PROcess Modeling System - gPROMS) je postaveno několik nástrojů pro oblast energetiky, chemie a procesní a zpracovatelské techniky. Systém umožňuje tvorbu jak stacionárních, tak dynamických modelů. Skládá

se z balíku termodynamických a fyzikálních vlastností látek - gSAFT, sady tří modelovacích prostředí - ProcessBuilder, FormulatedProducts a ModelBuilder a dalšího doplňkového software (Process Systems Enterprise, 2019).

Modelování v gPROMS je rovnicově orientované. Avšak tyto rovnice jsou psány v objektové programovací jazyce (podobný jako Matlab nebo Modelica), který je stručný a jasný. Uživatel se tak nemusí obávat složitosti metod řešení těchto rovnic. Dále gPROMS obsahuje velké množství knihoven se základními komponenty, které mohou být volně rozšiřovány a upravovány podle požadavků uživatele. Navíc tato platforma umožňuje matematický popis jevů (například tepelné procesy) i ve více rozměrech (Ahmed et al., 2012).

Ahmed et al. (2012) uvádí i detailnější popis možností systému gPROMS. Například tvorbu operačních procedur, které řídí simulovaný model, dynamická optimalizace modelu nebo odhad parametrů. Velkou výhodou této platformy je podle autorů Ahmed et al. (2012) otevřenost architektury, která umožňuje začlenění gPROMS do aplikací třetích stran. Autor dělí komunikační rozhraní s těmito aplikacemi na 4 kategorie:

- FOI (Foreign Object Interface) - část modelu může být popsána v externím softwaru (fyzikální vlastnosti, komponenty, apod.)
- FPI (Foreign Process Interface) - umožňuje výměnu dat s externím softwarem (real-time regulace, balíky pro výcvik operátorů nebo klientské rozhraní přizpůsobené na míru méně zkušeným uživatelům)
- OCI (Output Channel Interface) - díky tomuto rozhraní lze jednoduše přenést výsledky simulace provedené v gPROMS do externího softwaru (například do Excelu)
- OSI (Open Solver Interface) - toto rozhraní umožňuje propojení externích matematických řešičů s gPROMS

gPROMS je využíván při návrhu mnoha zařízení a procesů. Například Harraz et al. (2019) využívají platformu k simulaci difuzně-adsorbčního chladicího cyklu. Teklay et al. (2014) ve své práci využívají gPROMS k simulaci bleskové dehydroxylace jílu pro ekologičtější výrobu betonu.

2.1.3 Aveva SimSci PRO/II

Blokově orientovaný simulátor pro tvorbu stacionárních modelů. Je určený k tvorbě materiálových a energetických bilancí pro širokou škálu chemických procesů. Obsahuje velké množství termodynamických modelů pro různá odvětví (Aveva, 2019). Podle datasheetu Aveva (2019) jsou u tohoto systému klíčové následující vlastnosti:

- obsáhlé databáze termodynamických a fyzikálních vlastností
- tvorba a správa vlastních komponent
- komunikace se softwarem třetích stran (např. Microsoft Excel) - analýza a řízení simulací, tvorba reportů, apod.
- přístup k systému přes cloud

Dále jsou v datasheetu Aveva (2019) uvedeny seznamy knihoven komponent, chemických sloučenin, reakcí a výpočetních metod, které PRO/II využívá k výpočtům. Základní program PRO/II Process Engineering je doplněn o přídatné moduly a rozhraní pro komunikaci se softwarem třetích stran. Tyto moduly dále rozšiřují možnosti systému. Jedná se například o:

- SIM4ME - přenos proměnných mezi různými simulacemi a různým softwarem
- Polymer Module - umožňuje simulace a analýzu polymerních procesů
- Batch Module - sada reaktorů a destilačních kolon
- Electrolyte Module - rozšiřuje schopnosti simulací elektrochemických procesů

Dalšími moduly jsou MSE, AMSIM, RATEFRAC, BATCHFRAC a Spiral Crude Suite, které jsou v datasheetu popsány detailněji.

Pipitone a Bolland (2009) využívají PRO/II v modelu aparátu, který zachytává CO₂ při výrobě elektřiny a následně tento plyn skladuje.

2.1.4 Dymola

Dymola je jedním z mnoha systémů, využívající platformu Modelica. Podle The Modelica Association (2019) je to soubor standardů pro vývoj a simulace napříč odvětvími. Do standardu Modelica lze zařadit:

- Programovací jazyk Modelica - objektově orientovaný, založený na zápisu rovnic
- Functional Mock-Up Interface (FMI) - modelovací rozhraní pro sdílení modelů kompilací do jazyka C, Java nebo Python
- System Structure and Parametrization (SSP) - definice celých systémů, skládajících se z jednoho, či více FMI s možností přenosu do jiných simulačních nástrojů
- Distributed Co-Simulation Protocol (DCP) - zajišťuje kompatibilitu FMI s cílovým simulačním softwarem

Samotná platforma Modelica je volně dostupná, avšak u programů, které ji využívají už toto pravidlem není.

Software Dymola je podle Dassault Systems (2019) nástroj pro modelování a simulaci komplexních systémů napříč různými obory. Jde o blokově orientovaný systém pro tvorbu jak stacionárních, tak i dynamických modelů s možností tvorby vlastních komponent v jazyce Modelica. V datasheetu Dassault Systems (2018) je uvedeno, že zápis rovnic (včetně diferenciálních) je v tomto jazyce téměř totožný se symbolickým zápisem. Tím je tvorba komponent v porovnání s tradičními blokově orientovanými nástroji podstatně jednodušší.

Mezi výhody softwaru Dymola podle Dassault Systems (2018) patří:

- velké množství kompatibilních knihoven pro mnoho inženýrských oborů
- vysoká podobnost modelů komplexních systémů s realitou
- intuitivní tvorba modelů
- snadná tvorba vlastních komponent v důsledku otevřenosti platformy Modelica

Je dobré zmínit, že Dymola je k dispozici i v trial verzi, která je omezena složitostí vytvářeného modelu a možností použití jedné externí funkce (Dassault Systems, 2019).

Mezi projekty vytvářené v software Dymola můžeme zařadit odbornou práci Boarin et al. (2016), zabývající se simulací výzkumného reaktoru třídy TRIGA modelovaného lehkou vodou a využívající jako palivo hydrid zirkonia a uran. Dalším zajímavým příkladem využití tohoto programu, je odborná práce Bortolin et al. (2004), kteří v něm vytvořili model mokré části linky na výrobu papíru.

2.2 Volně dostupný software

Navzdory tomu, že vývoj bilančních a simulačních systémů vyžaduje spolupráci mnoha odborníků a velké množství finančních prostředků, existují programy, které jsou volně dostupné. Způsobů podpory vývoje tohoto softwaru je více (dobrovolné příspěvky, spolupráce na vývoji nebo odlehčené verze placených programů). Nejzajímavějšími programy jsou ty, které využívají otevřené knihovny, do kterých odborníci z celého světa přidávají jimi vytvořené komponenty. Bohužel u takto otevřených a volně dostupných systémů je mnohem vyšší pravděpodobnost, že některé komponenty nemusí fungovat zcela bezchybně.

Některé z těchto programů si zde detailněji popíšeme.

2.2.1 Xcos

Tak jako komerční Matlab obsahuje rozšíření (toolbox) Simulink, i jeho volně dostupná alternativa Scilab zahrnuje podobný nástroj, jímž je Xcos (Ospino et al., 2017).

Podle Scilab (2019) je tento blokově orientovaný, ergonomický a efektivní nástroj určený pro průmyslové a akademické účely. Poskytuje komponenty pro tvorbu modelů pro aerodynamiku, automotive, hydrauliku, řídicí systémy a další oblasti.

Následkem využití FMI a integrovanému kompilátoru do jazyka Modelica lze data ze simulací nebo části modelů vytvořených v tomto programu využít i v jiných systémech.

Při základním seznamování s Xcos nám pomůže jeho podobnost se Simulink. Můžeme totiž využít například návody, které zpracoval Karris (2006).

Janík a Žáková (2011) se ve své odborné práci zabývají exportem a následnou kompilací kódu do XML (přes AJAX a JSON). Takto zkompilovaný kód je možné vložit do online editoru (např. www.w3schools.com) a simulovat tak na online platformě.

I když je Xcos využíván spíše k simulacím řídicích systémů a elektrických obvodů, lze najít i odborné práce v jiných oborech, využívajících tento software. Například Vanfretti et al. (2013) použili Xcos ve spojení s Dymolou k vytvoření modelu energetického zařízení. Cílem této práce bylo především vyzkoušet možnosti výměny dat mezi výše uvedenými programy.

2.2.2 OpenModelica

Tento volně dostupný blokově a rovnicově orientovaný simulátor s grafickým editorem (OpenModelica Connection editor) umožňuje tvorbu modelů jak stacionárních, tak dynamických. Je vytvořený na platformě Modelica (OpenModelica.org, 2019).

Díky programovacímu jazyku Modelica je tento systém vhodný pro začátek s rovnicově orientovaným modelováním. Mezi obrovské výhody tohoto programu patří také možnost kombinace bloků z různých inženýrských oblastí (např. mechaniky a elektroniky) (Fritzson, 2014).

OpenModelica byla využita například v odborné práci Nyongbela a Johannsen (2015), kteří na základě experimentu vytvořili model průběhu koroze měděných trubek. Mehl a Johannsen (2017) pak například využili tento software pro výpočet chemických vlastností vody na základě zásaditosti a kyselosti (pH, mez sytosti, potenciál srážlivosti uhličitanu vápenatého).

2.2.3 Cantera

Cantera je nástrojem pro modelování chemických, kinetických, termodynamických a přenosových procesů. Objektově orientovaný jazyk a zobecněné algoritmy umožňují snadnou úpravu modelů a změnu parametrů. Tento software může být využíván přes rozhraní dalšími aplikacemi (Python, Matlab, C/C++ nebo Fortran 90). Mimo to lze možnosti nástroje Cantera využít online přes Jupyter Notebooks (<https://mybinder.org/repo/cantera/cantera-jupyter>) (Cantera.org, 2019).

Goodwin (2002) využil tento volně dostupný software pro simulaci CVD povlakování. Dále Tao a Dongmei (2012) spojili nástroj Cantera s programem OpenFOAM a provedli simulaci spalování směsi H_2-O_2 .

2.2.4 COCO Simulator

Simulační prostředí vyhovující standardům platformy CAPE-OPEN (CAPE - Computer Aided Process Engineering) pro stacionární modelování. Celé prostředí se skládá z několika komponentů a přídatných balíčků. Podle COCOsimulator.org (2019) těmito komponenty jsou:

- COFE (CAPE-OPEN Flowsheet Environment) - grafické uživatelské prostředí pro tvorbu modelů
- TEA (Thermodynamics for Engineering Applications) - knihovna termodynamických vlastností látek založených na systému ChemSep a obsahující přes 430 běžně používaných látek
- COUSCOUS (CAPE-OPEN Unit-operations Simple package) - knihovna komponent pro tvorbu modelu (obsahuje např. výměník tepla, čerpadla, regulátory apod.)
- CORN (CAPE-OPEN Reaction Numerics package) - knihovna chemických reakcí a bilančních rovnic.

Dle COCOsimulator.org (2019) mezi přídatné moduly patří například:

- Water - samostatný balík vlastností vody a vodní páry na základě IAPWS-97
- JUIcE (Unit-operation Icon Editor) - tvorba a úpravy ikon komponent
- TERNYP - zpracování výsledků simulace do diagramů

CAPE-OPEN platforma definuje několik softwarových rozhraní, které umožňují kompatibilitu modelovacího prostředí a systémů třetích stran, jako je např. Matlab nebo Microsoft Excel (COLaN, 2019).

Model rovnovážného reaktoru v prostředí COCO vytvořili van Baten a Szczepanski (2011).

2.3 Shrnutí

Podle výše zpracované rešerše jsou volně dostupné programy ve výzkumné činnosti, ale i v průmyslu méně využívány. V důsledku toho je o uvedeném softwaru v této práci obecně méně informací.

Jelikož jsou open-source systémy lépe dostupné (nejen pro studenty, ale i pro školy), budeme se jimi zabývat i dále a dva z nich využijeme pro tvorbu stacionárního a dynamického modelu. Zjistíme, zda jsou i tyto programy schopny vygenerovat výsledky použitelné při návrhu reálných zařízení. Také se zaměříme na uživatelskou přívětivost těchto systémů.

Pro tvorbu stacionárního modelu, tedy bilance odparky, použijeme prostředí COCO, konkrétně uživatelské prostředí COFE.

Dynamický model, tedy model vakuového chlazení vody, vytvoříme v systému OpenModelica, jež umožňuje jednoduchou tvorbu rovnicově orientované simulace.

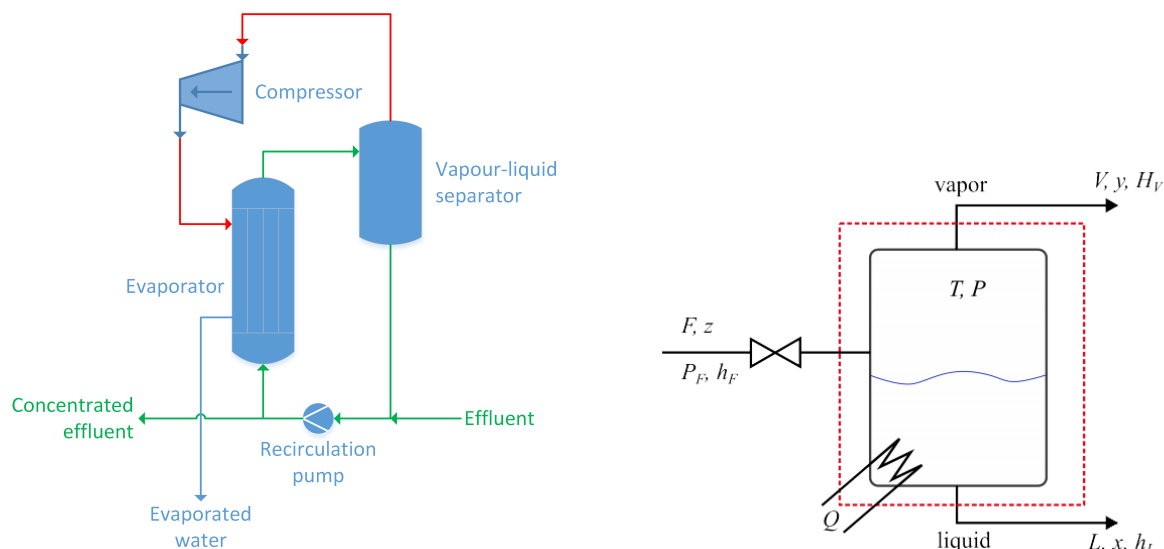
Základní informace o všech zmíněných programech a systémech jsou uvedeny v tabulce 1.

Komerční software							
	R	O	B	S	D	Hlavní vlastnosti	Příklad využití v praxi
Simulink	×	×	✓	✓	×	možnost tvorby vlastních komponent	Vysokotlaký PEM elektrolýzér (Yigit a Selamet, 2016)
gPROMS	✓	×	×	✓	✓	možnost tvorby vlastních komponent, možnost začlenění do aplikací třetích stran	Difuzně-adsorbční chladicí cyklus (Harraz et al., 2019), Blesková dehydroxolace jílu (Teklay et al., 2014)
PRO/II	×	×	✓	✓	×	obsáhlé databáze termodynamických a fyzikálních vlastností, tvorba a správa vlastních komponent, přístup přes cloud	Zachytávání a skladování CO ₂ při výrobě elektřiny (Pipitone a Bolland, 2009)
Dymola	×	×	✓	✓	✓	platforma Modelica, velké množství kompatibilních knihoven, intuitivní tvorba modelů	model reaktoru třídy TRIGA (Boarin et al., 2016), model části linky na výrobu papíru (Bortolin et al., 2004)
Volně dostupný software							
Xcos	×	×	✓	×	✓	podobnost se Simulink, kompatibilní s platformou Modelica, tvorba vlastních komponent	Model energetického zařízení (Vanfretti et al., 2013)
OpenModelica	✓	×	✓	✓	✓	platforma Modelica, kompatibilita bloků z různých inženýrských oblastí	Model koroze měděných trubek (Nyongbela a Johannsen, 2015), výpočet chemických vlastností vody (Mehl a Johannsen, 2017)
Cantera	×	✓	×	✓	✓	lze používat přes více programovacích jazyků (Matlab, Python, Fortran, C) a také online přes Jupyter Notebooks	Simulace CVD povlakování (Goodwin, 2002), simulace spalování směsi H ₂ -O ₂ (Tao a Dongmei, 2012)
COCO Simulator	×	×	✓	✓	×	platforma CAPE-OPEN, kompatibilní s Excelem, možnost tvorby vlastních komponent a úpravy stávajících	Model rovnovážného reaktoru (van Baten a Szcze-panski, 2011)

Tabulka 1: Shrnutí základních vlastností a využití programů pro simulace (Vysvětlivky: R - rovnicové modelování, O - objektové, B - blokové, S - stacionární model, D - dynamický model)

3 Bilance odparky

V této kapitole se budeme zabývat tvorbou stacionárního modelu expanzní odparky (flash) s cirkulačním oběhem koncentrátu a mechanickou kompresí brýdových par (MVR - Mechanical Vapour Recompression). Nejprve si vysvětlíme princip funkce a rozdělíme odparky podle několika hledisek. Následně si popíšeme komponenty použité v modelu a provedeme základní analytické výpočty, které nám budou sloužit k ověření výsledků bilance, vytvořené v prostředí COCO Simulator.



(a) odparka MVR (<https://www.hrs-heatexchangers.com/wp-content/uploads/2016/09/Mechanical-Vapour-Compression-MVR-or-Thermal-Vapour-Compression-TVR-Systems.png>)

(b) Expanzní separátor (flash) jako základní prvek expanzní odparky (Iggländ a Mazzoti (2015))

Obrázek 4: Cirkulační odparka s mechanickou rekompresí brýdových par a jejím základním prvkem, separátorem kapalina-pára

3.1 Princip funkce

Odparky pro svou funkci využívají proces fázové přeměny kapaliny na páru. V technické praxi se podle Zvoníček (1961) využívají z několika následujících důvodů:

- technologické - přizpůsobení dalšímu zpracování, odlehčení kvůli transportu
- konzervační - prodloužení životnosti zvýšením koncentrace roztoků, emulzí či suspenzí
- tepelně ekonomické - snížení koncentrace vody v sušené látce před vstupem do sušárny

Při odpařování je cílem získání těkavější látky (rozpuštědla) ze směsi pokud

možno v nejčistší formě (například získávání pitné vody z mořské) (Zvoníček, 1961).

Chceme-li získat pevné, nebo netěkavé látky ze směsi, hovoříme o zahušťování. Nejčastěji separovaným rozpouštědlem v odparkách je voda. Pokud je vyžadováno pouze zvýšení koncentrace pevných látek, odstraní se jen část rozpouštědla. Chceme-li získat rozpuštěné látky v pevné fázi, musíme docílit meze nasycení roztoku a tím krystalizace (Zvoníček, 1961).

Jelikož odparky fungují na principu fázové přeměny kapaliny na páru, pro pochopení funkce je vhodné ještě zmínit rozdíl mezi vypařováním a varem, který dobře popsal ve své odborné práci Zvoníček (1961).

Pokud je parciální tlak par kapaliny nad jejím povrchem nižší než celkový tlak okolního prostředí, hovoříme o **vypařování**. Páry se uvolňují pouze z povrchu kapaliny.

Vyrovná-li se parciální tlak par nad kapalinou celkovému tlaku okolního prostředí (nebo ho převýší), dojde k **varu**. Fázová přeměna je v tomto případě rychlejší a dochází k ní v celém objemu kapaliny. Proto se tento jev využívá v odparkách v technické praxi.

Zahušťování i odpařování je realizováno přívodem potřebného množství tepla do směsi. Po dosažení teploty varu se těkavější látka mění na páru, tzv. brýdy (brýdové páry).

Využití odparek je opravdu široké. Přes již zmíněné získávání pitné vody, přes těžbu solí, až po výrobu cukru, nebo zahušťování ovocných a zeleninových šťáv (Zvoníček, 1961). V odborné práci Ahmad Mansour (2019) je uvedeno, že princip odparky (konkrétně podtlakové - flash) se běžně využívá v geotermálních elektrárnách, pro produkci páry na pohon turbíny.

Zvoníček (1961) dále uvádí výstižné dělení odparek podle následujících hledisek:

- Dle pracovního tlaku:
 - atmosférický tlak
 - přetlak
 - podtlak - vakuové odparky
- Dle způsobu práce:
 - odparky s přirozenou cirkulací roztoku
 - odparky s nucenou cirkulací roztoku
 - proudové (filmové) odparky

- Dle způsobu využití přivedeného tepla:
 - jednostupňové odparky prosté
 - vícestupňové odparky prosté
 - jedno, nebo vícestupňové odparky s předhřevem
 - odparky s mechanickou kompresí brýdových par (MVR)
 - odparky s termokompresí brýdových par (TVR)

Podle Iggländ a Mazzoti (2015) podtlaková odparka přivádí kapalinu k varu náhlým - bleskovým poklesem tlaku (flash - blesk). Vlivem tohoto poklesu se část přiváděné směsi velmi rychle přemění na páru. Ta je odváděna horní částí zařízení, zatímco neodpařená (a zahuštěná) kapalina padá ve formě kapek na dno. Odvod brýdových par je podle Zvoníček (1961) zajištěn buď vývěvami (obtížné kvůli velkému objemu par) nebo jejich kondenzací. Zvoníček (1961) uvádí výhody i nevýhody tohoto typu odparek. Mezi výhody patří například:

- nízká teplota v odparce a z toho plynoucí možnost zahušťování termolabilních látek (potraviny, léčiva, apod.)
- menší topná plocha zařízení a možnost použití k vytápění odpadního tepla horkých kapalin nebo již použité páry např. z turbíny
- zahušťování látek s běžně vysokým bodem varu
- snížení ztrát sáláním do okolí

Jako nevýhody jsou pak uvedeny komplikovanější konstrukce a větší spotřeba topné páry oproti přetlakovým a atmosférickým odparkám. Dále potřeba značného množství chladicí vody v kondenzátoru (Zvoníček, 1961).

My si podtlakovou odparku doplníme ještě o cirkulační smyčku koncentrátu a kompresor brýdových par, který je bude stlačovat na vyšší tlak. Tím se zvýší jejich teplota a následně mohou být využity k vytápění odparky (Zvoníček, 1961).

3.2 Analytický výpočet expanzního separátoru

Základem analytického výpočtu expanzního separátoru z obrázku 4b je dle Iggländ a Mazzoti (2015) celková bilance hmotnosti

$$\dot{m}_F = \dot{m}_L + \dot{m}_V, \quad (1)$$

kteřou u binární směsi doplníme ještě bilancí netěkavé složky

$$\dot{m}_F \omega_0 = \dot{m}_L \omega_1, \quad (2)$$

kde ω_0 je počáteční hmotnostní zlomek netěkavé složky ve vstupním proudu a ω_1 je hmotnostní zlomek této složky v zahuštěném roztoku.

Pomocí těchto dvou rovnic můžeme získat dvě neznámé. Například při daném hmotnostní toku, složení vstupního proudu a požadované koncentraci zahuštěného roztoku je možné dopočítat hmotnostní toky brýd a zahuštěného roztoku.

Pokud zanedbáme tepelné ztráty, vyjádříme energetickou bilanci vztahem

$$\dot{m}_F h_F + \dot{Q} = \dot{m}_L h_L + \dot{m}_V h_V , \quad (3)$$

kde h_F , h_L a h_V jsou měrné entalpie jednotlivých proudů.

Dále předpokládáme, že mezi fázemi (předpokládáme plynnou a kapalnou fázi) existuje v odparce rovnováha a proto můžeme pro binární směs psát

$$y = K_1(p, T)x , \quad (4)$$

$$(1 - y) = K_2(p, T)(1 - x) , \quad (5)$$

kde x je molový zlomek kapaliny, y je molový zlomek par a $K_1(p, T)$, $K_2(p, T)$ jsou rovnovážné konstanty. V našem případě je konstanta $K_i(p, T)$ dána Raoultovým vztahem

$$K_i = \frac{p_i^{\text{sat}}(T)}{p} , \quad (6)$$

kde p_i^{sat} je tlak nasycených par v závislosti na teplotě daný Antoinovou rovnicí

$$\log p_i^{\text{sat}} = A - \frac{B}{C + T} . \quad (7)$$

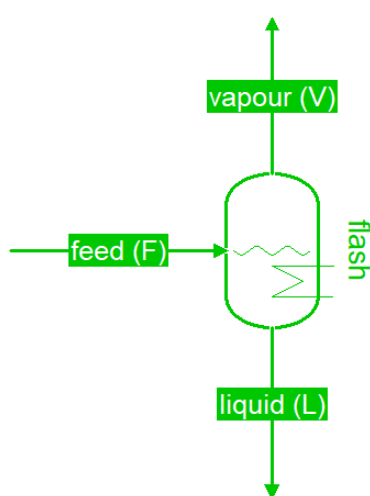
Konstanty A , B a C jsou pro dané látky uvedeny ve fyzikálních tabulkách.

3.3 Komponenty modelu

3.3.1 Flash

Začneme samotným tělesem odparky. V editoru COFE ho nalezneme pod názvem Flash. Na obrázku 5a vidíme, že zařízení má 3 materiálové porty (feed, vapour a liquid). Dále lze připojit ještě energetický, nebo informační proud pro regulaci procesu. V okně (viz obrázek 5b) můžeme nastavit, jak bude komponenta Flash řízena a následně definujeme její parametry.

Komponenta počítá na základě vstupního proudu a zadaných parametrů (např. přiváděného tepla) podle hmotnostní bilance (1) a energetické bilance (3) hmotnostní tok brýdových par \dot{m}_V a hmotnostní tok zahuštěné kapaliny \dot{m}_L .



(a) Ikona komponenty

Parameter	Value	Unit
Pressure spec.	Pressure	
Pressure drop	50000	Pa
Pressure	70000	Pa
Flash specification	Vapor fraction	
Heat duty	84837.3517946	W
Temperature	89.9315100664	°C
TP Outlet Flash	No	
Thermo Version	1.1	
Vapor fraction	0.079999933808	
3-phase flash	No	
Number of inlets	1	

(b) Okno nastavení parametrů komponenty

Obrázek 5: Komponenta Flash

Chceme-li ze separátoru s vnitřním tlakem 70 kPa odvádět například $0,04 \text{ kg s}^{-1}$ par z hmotnostního toku $0,5 \text{ kg s}^{-1}$ vody o tlaku 120 kPa a teplotě 93 °C , plyne nám z (1)

$$0,5 = \dot{m}_L + 0,04 \rightarrow \dot{m}_L = 0,46 \text{ kg s}^{-1} . \quad (8)$$

Dále z energetické bilance (3) dosazením z rovnice (8) a entalpií odečtených z COCO simulátoru získáme vztah pro výpočet potřebného tepelného toku:

$$\dot{Q} = 0,46 \cdot 376680,3 + 0,04 \cdot 2659445,2 - 0,5 \cdot 389626,77 = 84837,36 \text{ W} . \quad (9)$$

Jak můžeme vidět na obrázku 5b, výpočet v programu je téměř totožný s analytickým výpočtem.

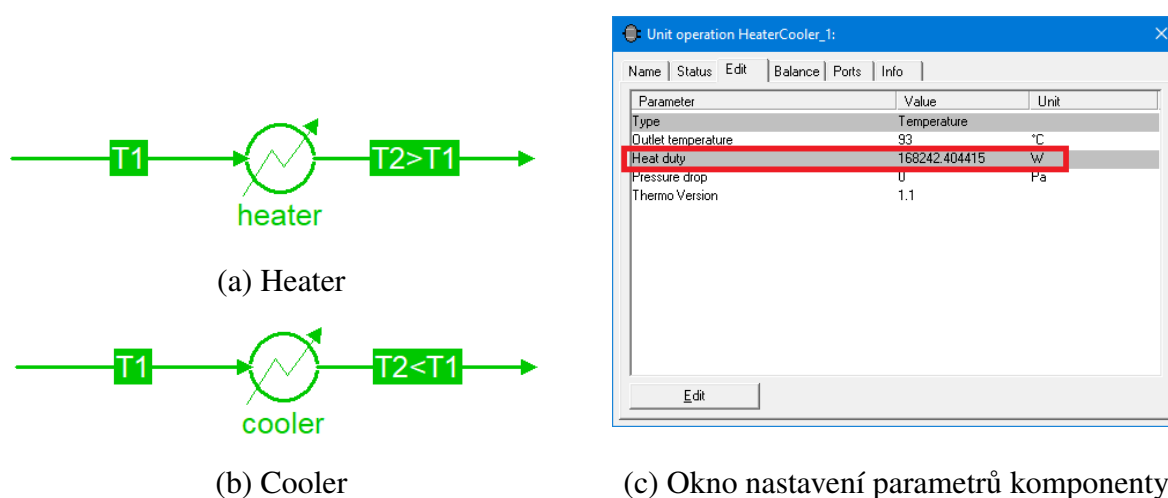
3.3.2 Heater/Cooler

Tato komponenta ohřívá, anebo naopak ochlazuje vstupní proud, buď podle zadané teploty na výstupu, nebo definováním tepelného výkonu. Zařízení tedy dodává či odebírá ze vstupního proudu určitý tepelný tok, který je daný kalorimetrickou rovnicí

$$\dot{Q} = \dot{m} \cdot c_p \cdot (T_2 - T_1) , \quad (10)$$

kde \dot{m} je hmotnostní tok, c_p měrná tepelná kapacita za konstantního tlaku protékající látky a T_1 a T_2 jsou teploty na vstupu a výstupu.

Ikona komponenty a okno nastavení parametrů je na obrázcích 6.



Obrázek 6: Komponenta Heater/Cooler

Podobně jako v případě komponenty Flash si můžeme na příkladu ukázat analytický výpočet. Mějme například proud vody s hmotnostním tokem 10 kg s^{-1} o tlaku 120 kPa a teplotě 89 °C , který budeme chtít ohřát na 93 °C . Z kalorimetrické rovnice (11) pak pro potřebný tepelný tok plyne

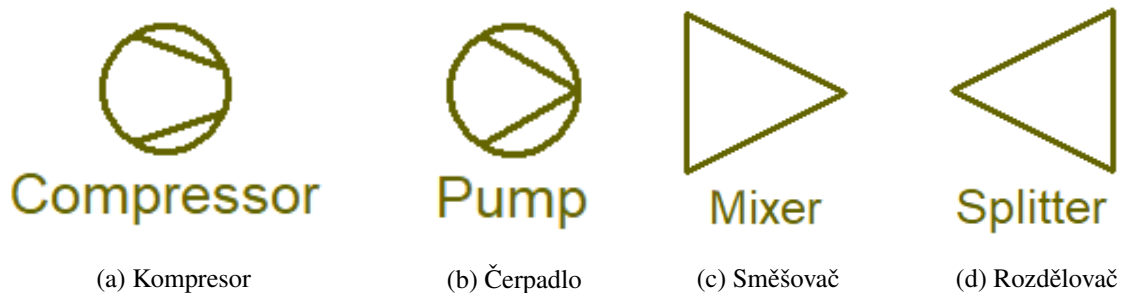
$$\dot{Q} = 10 \cdot 4203,935 \cdot (93 - 89) = 168157,4 \text{ W}. \quad (11)$$

Tato hodnota je opět téměř totožná s výsledkem bilance v programu (obrázek 6c).

3.3.3 Ostatní použité komponenty

Mezi další komponenty, které budeme při tvorbě bilančního modelu odparky potřebovat, patří:

- **Kompresor** (Compressor) - podle nastavených parametrů a účinnosti zvyšuje tlak proudu plynné fáze
- **Čerpadlo** (Pump) - podle nastavených parametrů a účinnosti zvyšuje tlak proudu kapalné fáze
- **Směšovač** (Mixer) - dva nebo více vstupních proudů sloučí do jednoho výstupního
- **Rozdělovač** (Splitter) - jeden vstupní proud rozdělí na dva nebo více proudů výstupních, podle zadaného dělicího faktoru



Obrázek 7: Ostatní komponenty

Dále budeme v modelu potřebovat vytvořit regulační obvody pro nastavení požadovaných parametrů. Tyto obvody sestavíme pomocí dvou následujících komponent:

- **Měřicí jednotka** (Measure Unit) - zařadí se do proudu, kde je nutné provést měření a odesílá informace o hlídaných parametrech do regulátoru
- **Regulátor** (Controller) - na základě informací z měřicí jednotky odesílá do akčního členu signál pro dosažení požadovaných parametrů



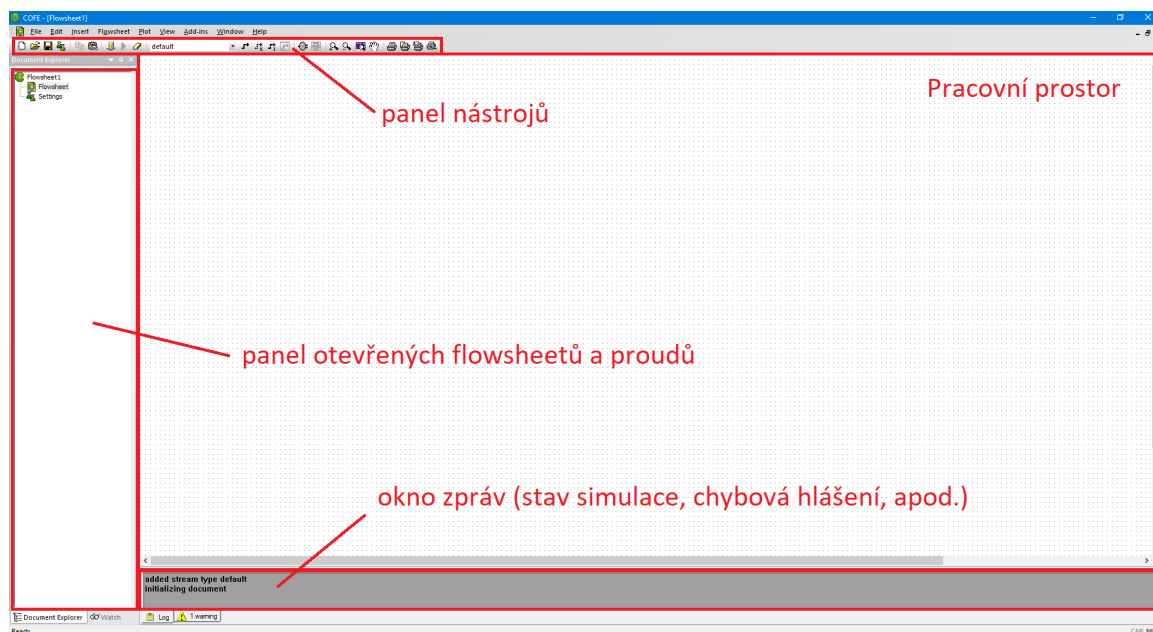
Obrázek 8: Členy regulačního obvodu

3.4 Tvorba modelu odparky v editoru COFE

Z komponent popsaných v předchozí kapitole nyní můžeme vytvořit model odparky. Nejprve si představíme uživatelské rozhraní editoru COFE a následně krok po kroku sestavíme PFD schéma odparky s cirkulačním okruhem a mechanickou kompresí brýdových par (MVR) (ideální kompresor) (varianta A), u které provedeme bilanci při odpařování čisté vody. Dále budeme bilancovat MVR odparku s reálným kompresorem (varianta B) a odparku s kondenzací brýdových par (varianta C). Výsledky těchto tří simulací si v závěru této kapitoly přehledně uspořádáme a vzájemně porovnáme. Procesní schémata simulovaných odparek jsou na obrázku 12.

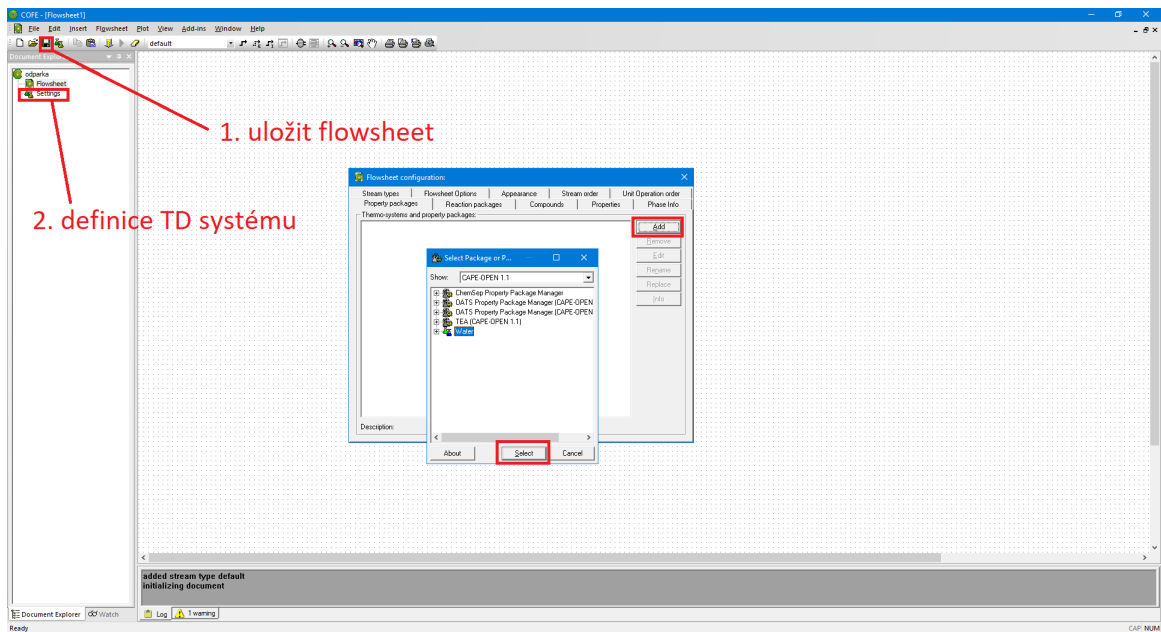
3.4.1 Uživatelské prostředí editoru, definice TD systému

Po spuštění editoru se v okně zobrazí přímo uživatelské rozhraní, které je popsáno na obrázku 9.



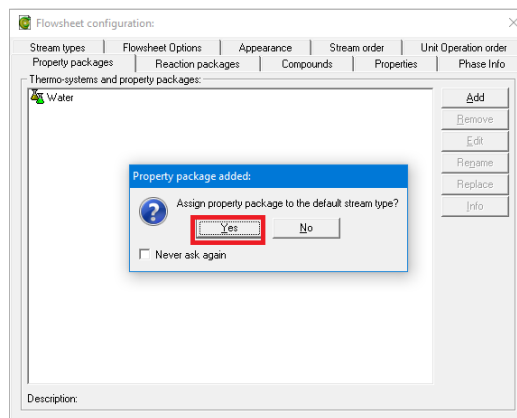
Obrázek 9: Uživatelské rozhraní COFE

Nový flowsheet se vytvořil automaticky a je třeba jej uložit pod názvem **odparka**. Potom musíme definovat ve flowsheetu termodynamické vlastnosti a látky, se kterými budeme pracovat. V našem případě vybereme z nabízených balíčků poslední, s názvem **Water** (viz obrázek 10). Tento balík čerpá z databáze IAPWS-IF97 (mezinárodní asociace pro vlastnosti vody a vodní páry).

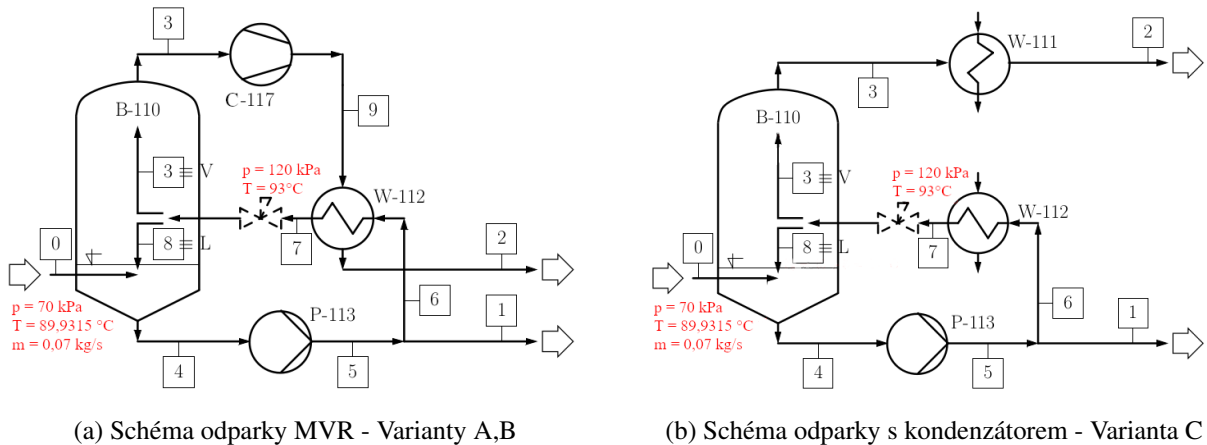


Obrázek 10: Uložení flowsheetu

Nastavíme přidany balík Water jako výchozí (obrázek 11) a můžeme začít se samotnou tvorbou modelu.



Obrázek 11: Definice TD systému



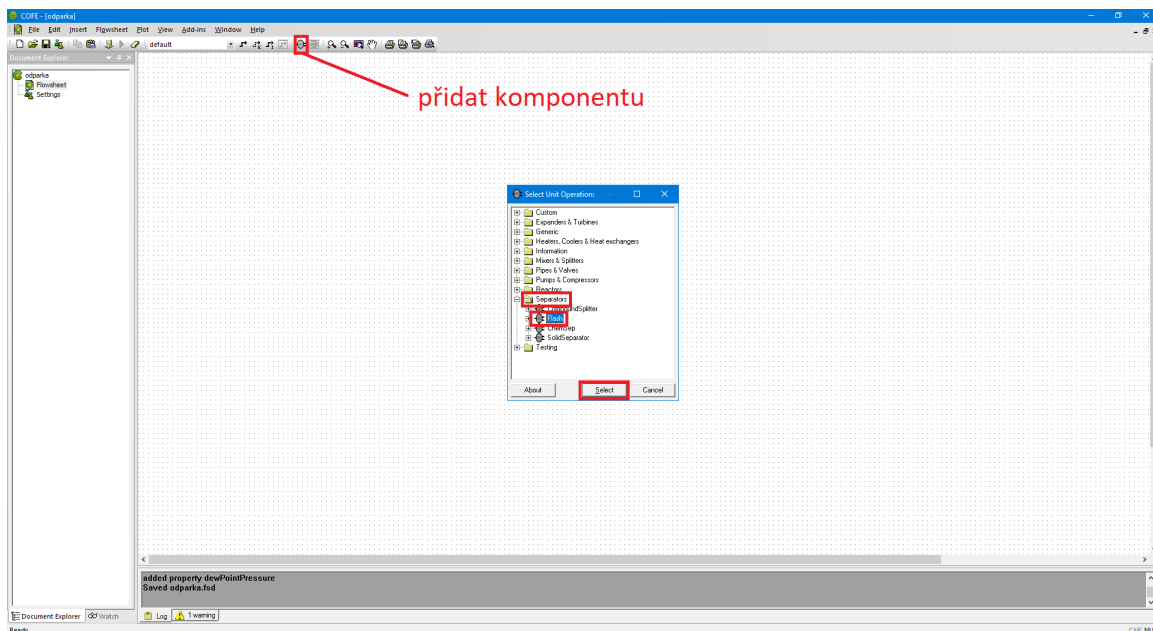
(a) Schéma odparky MVR - Varianty A,B

(b) Schéma odparky s kondenzátorem - Varianta C

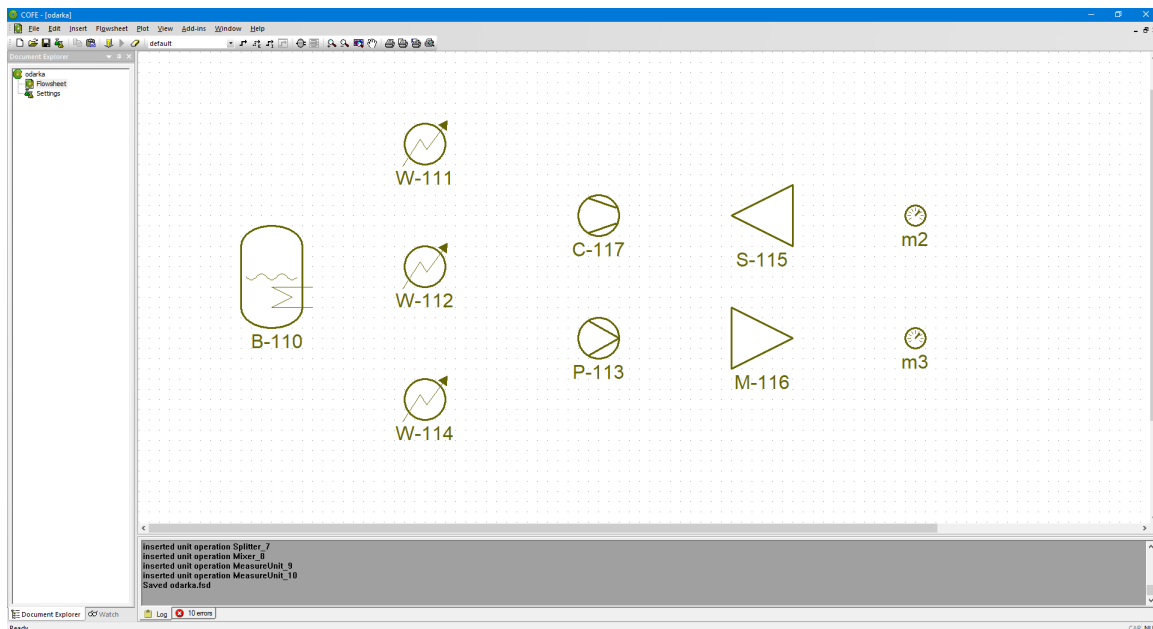
Obrázek 12: Schémata variant odparek

3.4.2 Sestavení PFD odparky

Klikneme na ikonu **insert unit operation** a vložíme komponentu Flash (obrázek 13). Následně stejným způsobem vložíme všechny ostatní komponenty (viz obrázek 14) a pro lepší přehlednost je přejmenujeme podle obrázku 12.

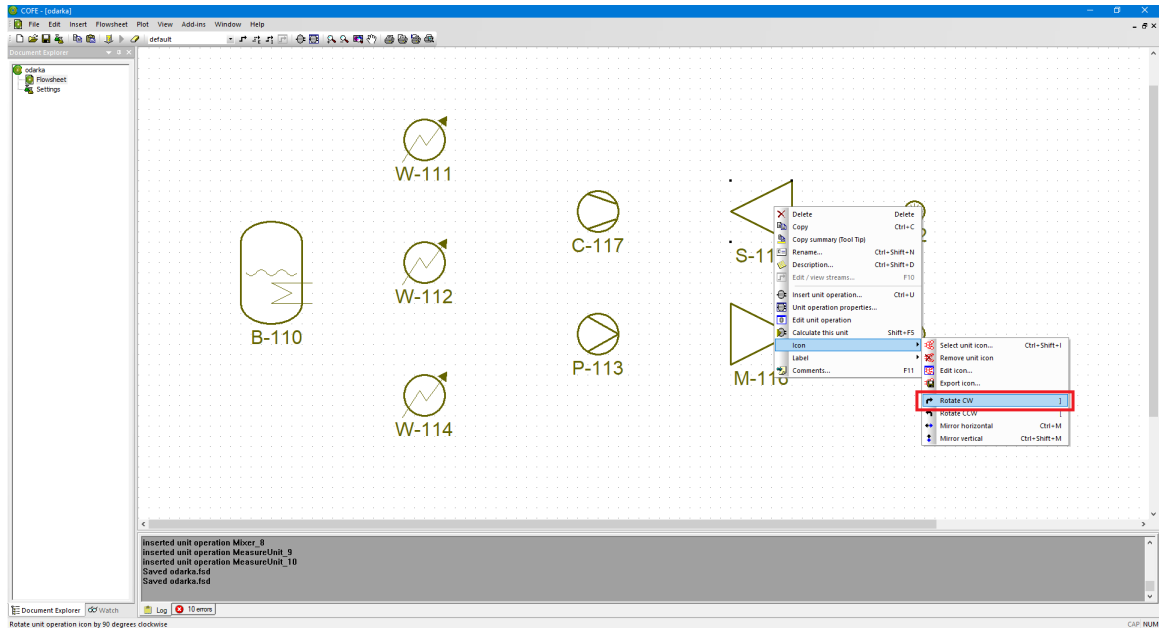


Obrázek 13: Vložení komponenty



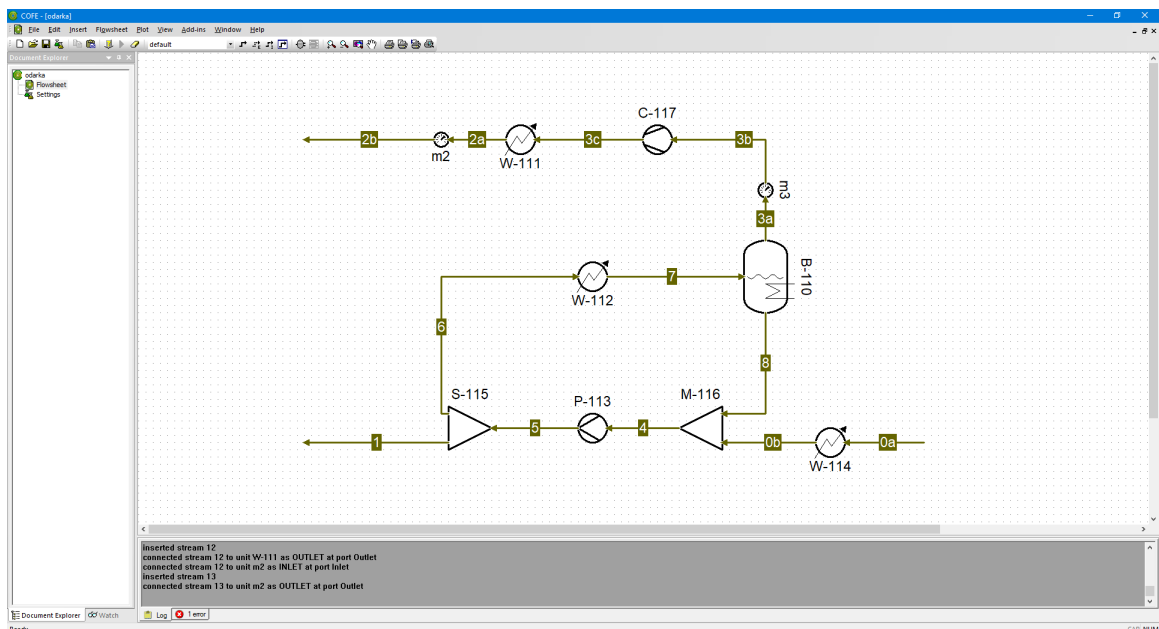
Obrázek 14: Použité komponenty

Některé komponenty bude potřeba natočit. Pravým tlačítkem myši tedy klikneme na požadovanou ikonu, v seznamu (obrázek 15) najedeme na **Icon** a podle potřeby otočíme.



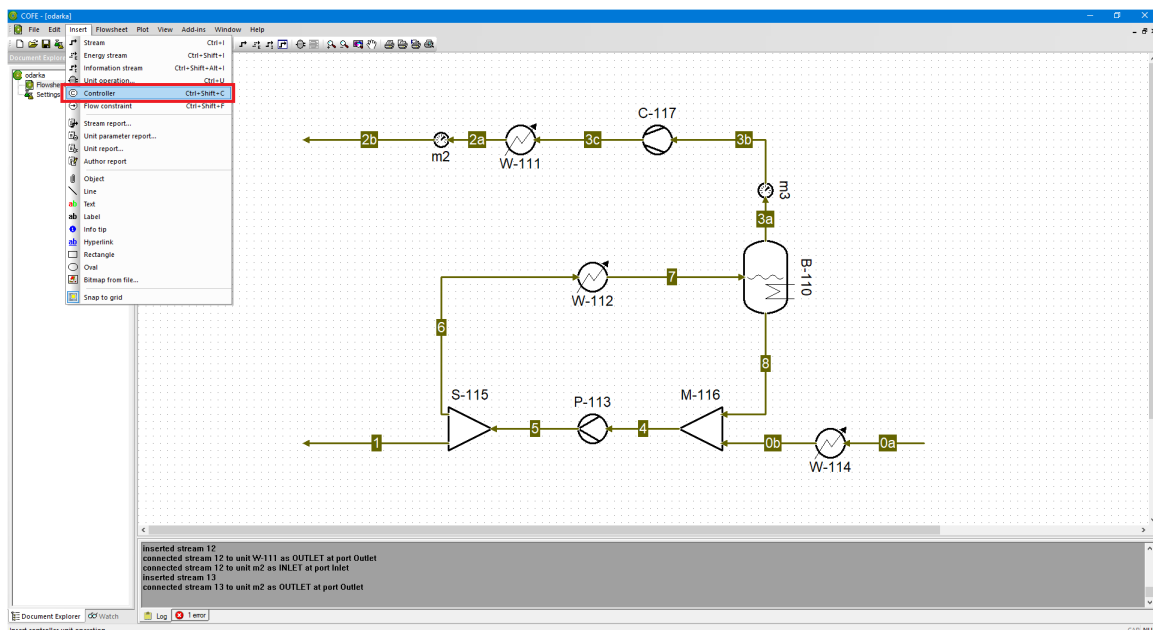
Obrázek 15: Otočení ikony

Nyní si jednotlivé ikony rozmístíme tak, jak budou zařazené v modelu podle obrázku 12 (viz obrázek 16) následně propojíme tyto komponenty proudy, které si očíslovujeme podle obrázku 12.



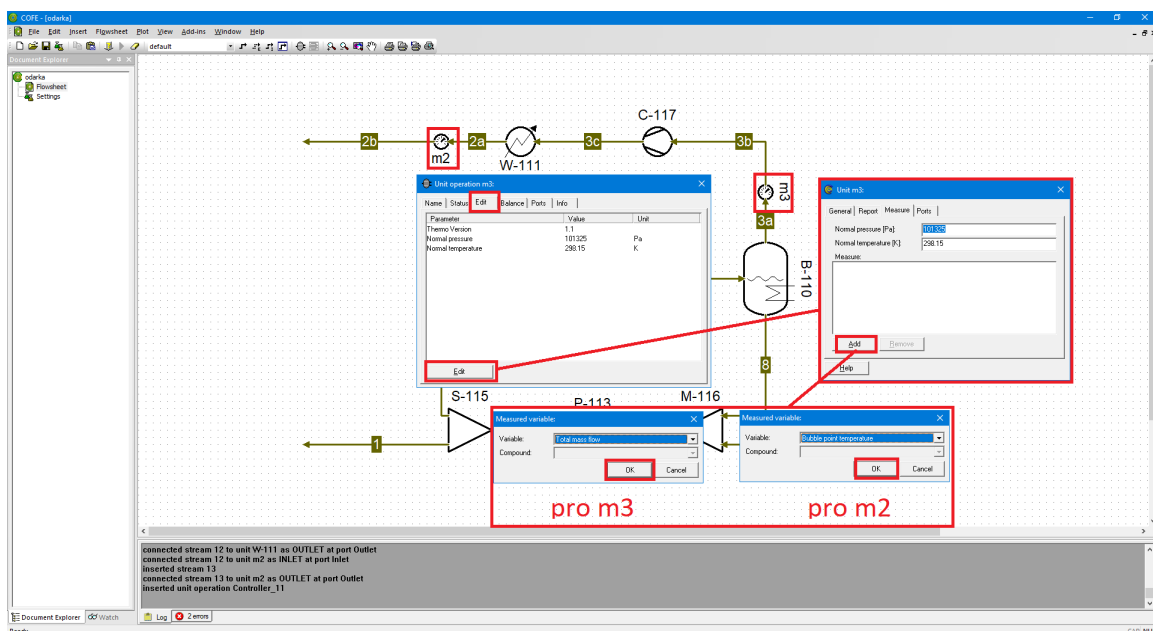
Obrázek 16: Uspořádané a propojené komponenty

Vytvořený model je třeba ještě doplnit o regulační člen, ve kterém budeme nastavovat požadované parametry. Regulátor nalezneme v záložce **Insert** (obrázek 17).



Obrázek 17: Vložení regulátoru

Regulátoru musíme nejprve přiřadit měřící zařízení, podle kterého bude hlídat předepsané parametry. Dvojklikem na měřící zařízení vyvoláme okno nastavení (viz obrázek 18).

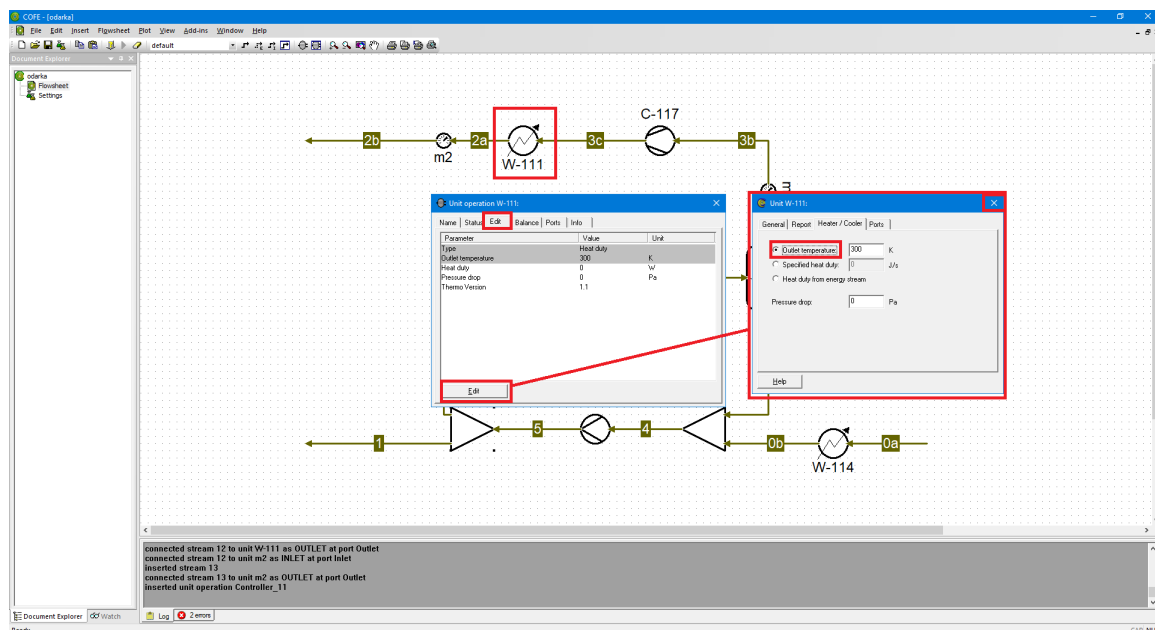


Obrázek 18: Nastavení měřících jednotek

V záložce **Edit** klikneme na **Edit** v novém okně na **Add**. Z rozbalovacího seznamu vybereme měřený parametr (u m3 celkový hmotnostní průtok). Měřící člen m2 bude přímo spojený s kondenzátorem W-111 a bude počítat teplotu bodu varu na

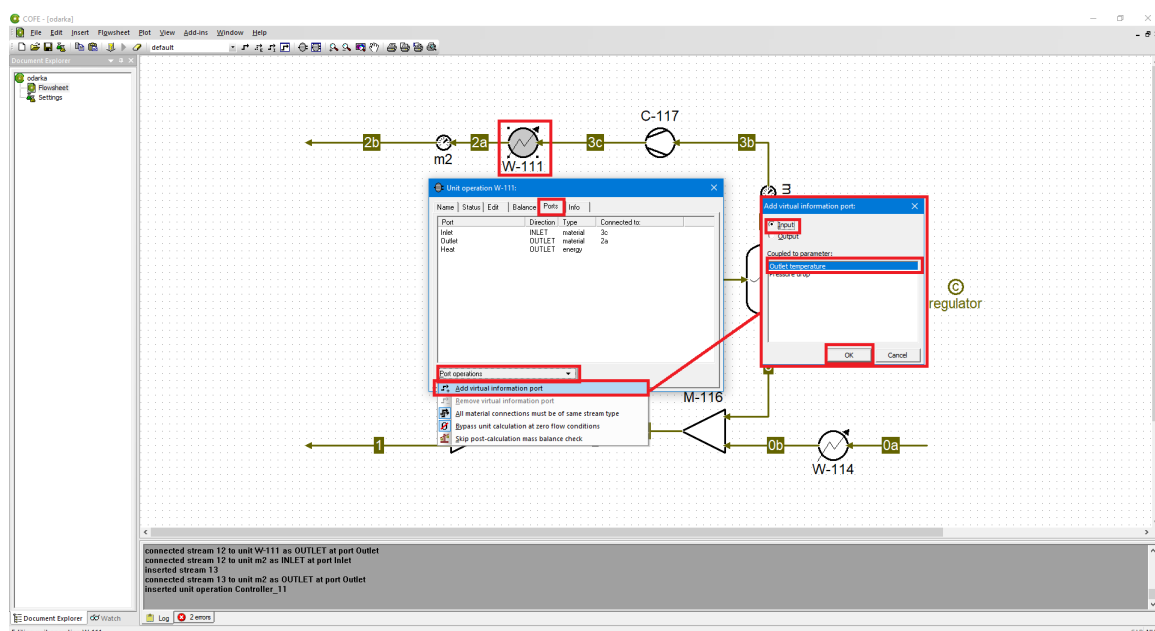
výstupu, na základě kterého se kondenzátoru nastaví jeho výkon. Brýdové páry se pak v kondenzátoru ochladí až na teplotu meze sytosti.

Dále je třeba nastavit informační vstupy u regulovaných zařízení. V našem případě půjde o rozdělovač a kondenzátor. Nejprve u kondenzátoru, opět v záložce **Edit**, klikneme na **Edit** a vybereme nastavovaný parametr (výstupní teplota) (obrázek 19).



Obrázek 19: Nastavení kondenzátoru W-111

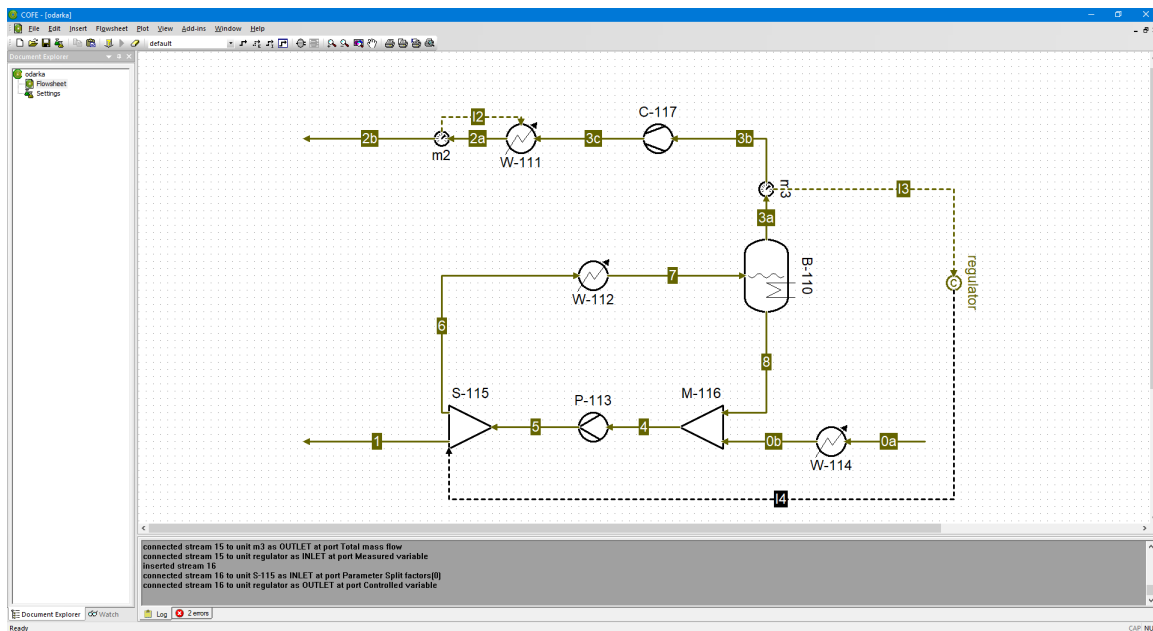
Potom přepneme na záložku **Ports** a tam otevřeme rozbalovací seznam, kde zvolíme přidání informačního proudu. V novém okně zaškrtneme vstup a vybereme výstupní teplotu (obrázek 20).



Obrázek 20: Přidání informačního proudu výměníku

U rozdělovače je regulovaný parametr již nastaven a je jím dělicí poměr. Obdobně jako u kondenzátoru přidáme informační proud a jako vstup nastavíme dělicí poměr.

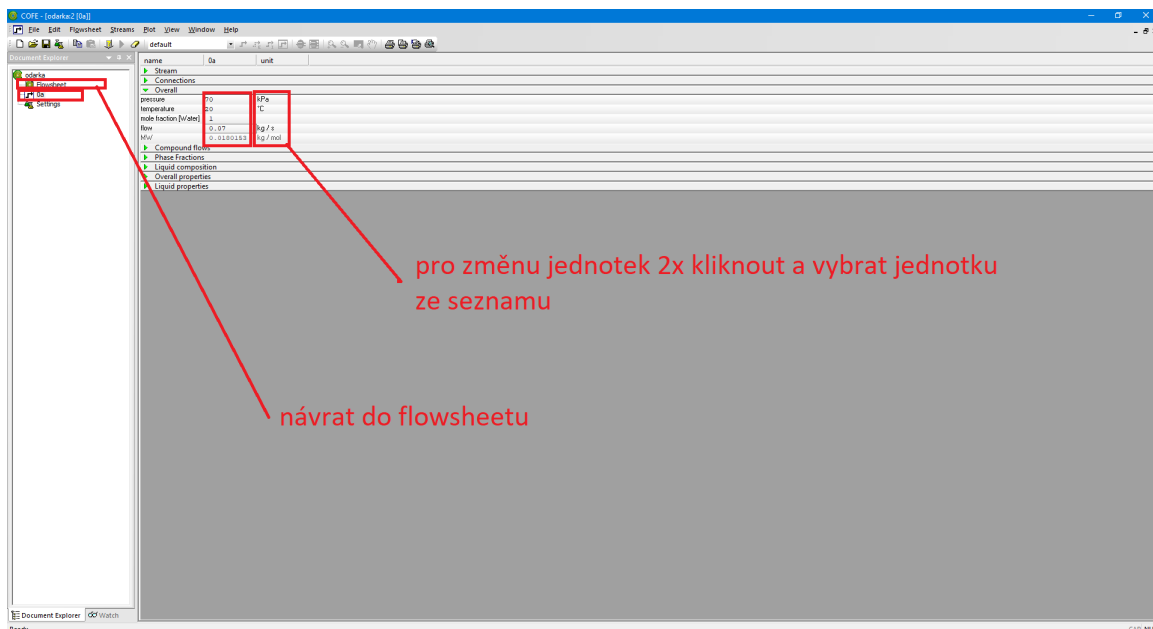
Nyní můžeme propojit regulační obvody, tedy m2 - W-111 a m3 - regulátor - S-115 podle obrázku 21.



Obrázek 21: Zapojení regulačních obvodů

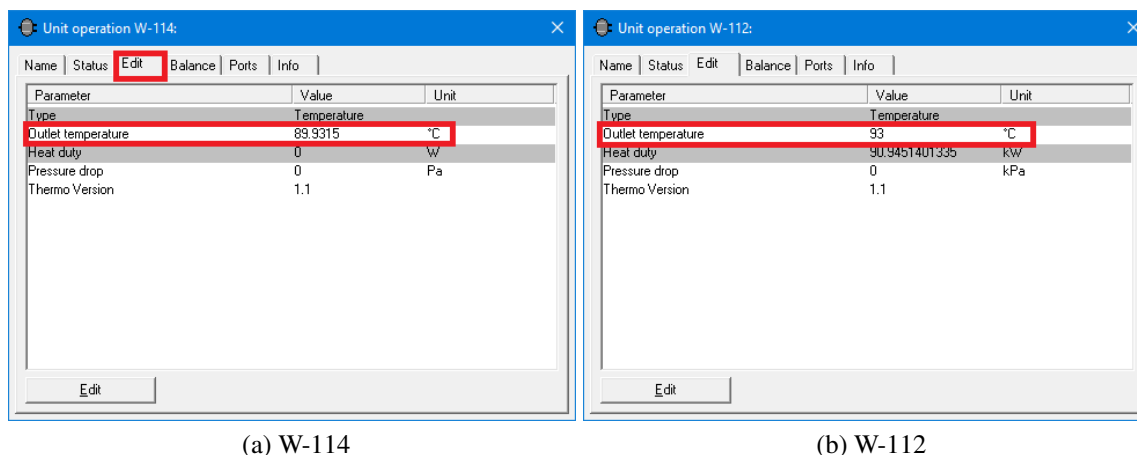
3.4.3 Nastavení vstupních a požadovaných hodnot

Tímto jsme vytvořili model, do které ještě bude potřeba doplnit vstupní a požadované hodnoty. Dvojklikem na proud 0a otevřeme detaily (obrázek 22). Zde můžeme nastavit parametry vstupního proudu. Zadané parametry jsou uvedeny v tabulce 2, uvedené na konci této kapitoly.



Obrázek 22: Nastavení proudu 0a

U výměníku W-114 na proudu 0 nastavíme výstupní teplotu na 89,9315 °C (obrázek 23a), abychom předešli vstupní proud na mez sytosti při tlaku v cirkulačním okruhu. U cirkulačního výměníku W-112 obdobně nastavíme výstupní teplotu na 93 °C (obrázek 23).



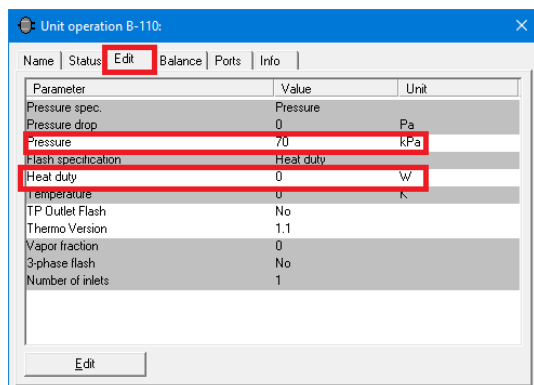
(a) W-114

(b) W-112

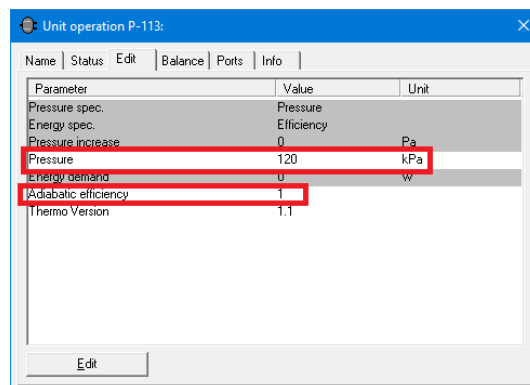
Obrázek 23: Nastavení výměníků

Použijeme adiabatický separátor a potřebné teplo dodáme pomocí výměníku tepla W-112. Ten oddělíme proto, abychom mohli definovat teplotu před vlastní expanzí. Právě tato teplota je jedním z návrhových parametrů (93 °C). Výstupní tlak

produktů bude 70 kPa (obrázek 24a). Jelikož požadujeme, aby měl proud vstupující do separátoru tlak 120 kPa, musíme na tuto hodnotu nastavit cirkulační čerpadlo s ideální účinností (obrázek 24b).



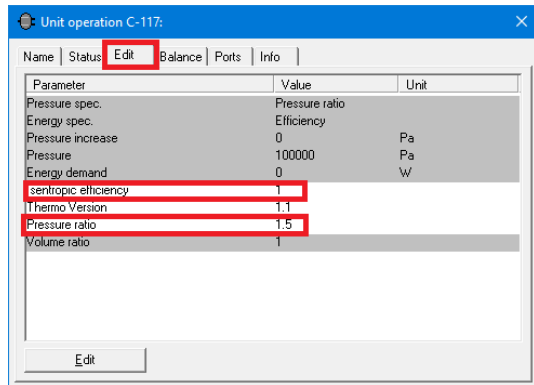
(a) B-110



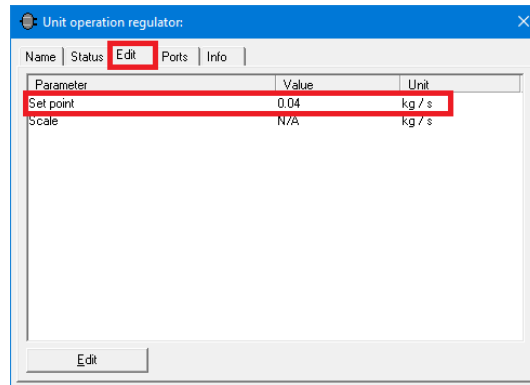
(b) P-113

Obrázek 24: Nastavení separátoru a čerpadla

Dále rozklikneme kompresor a nastavíme kompresní poměr 1,5 a pro variantu A ideální účinnost (obrázek 25a). Nakonec nastavíme regulátor dělicího poměru na požadovaný hmotnostní tok brýdových par ze separátoru B-110 ($0,04 \text{ kg s}^{-1}$) (obrázek 25b). Kondenzátor je samoregulovaný, ale v informačním proudu I2 potřebujeme nastavit počáteční hodnotu teploty (např. $90 \text{ }^\circ\text{C}$) (obrázek 26).

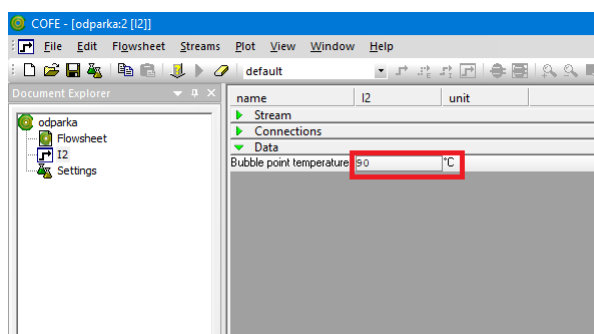


(a) C-117



(b) regulator

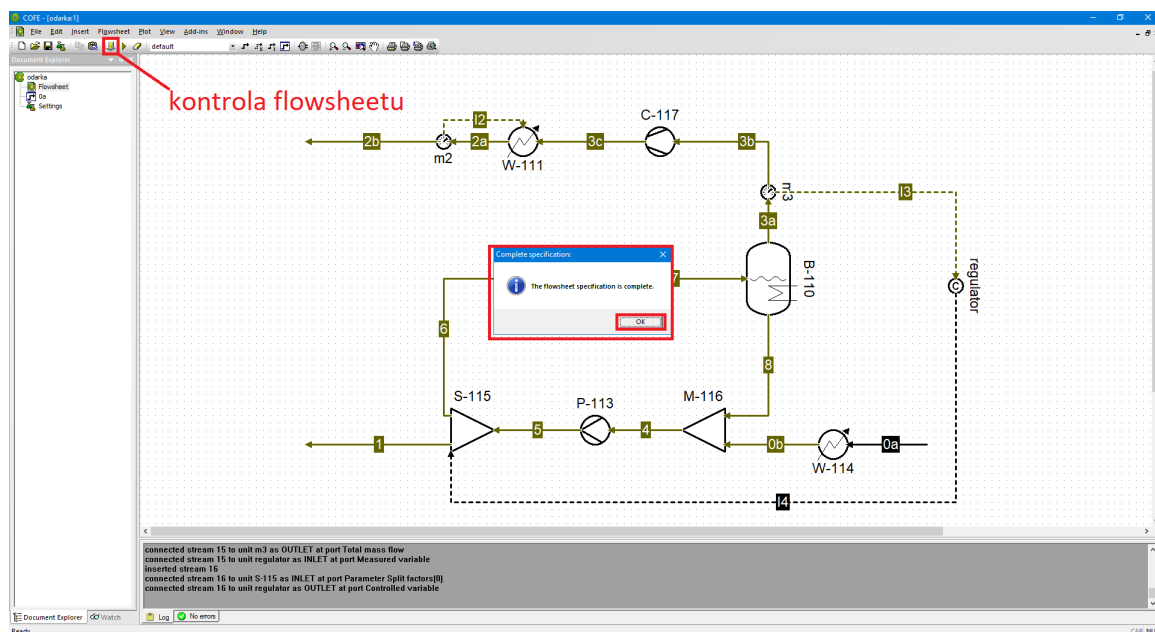
Obrázek 25: Nastavení kompresoru a regulátoru



Obrázek 26: Počáteční nastavení informačního proudu I2

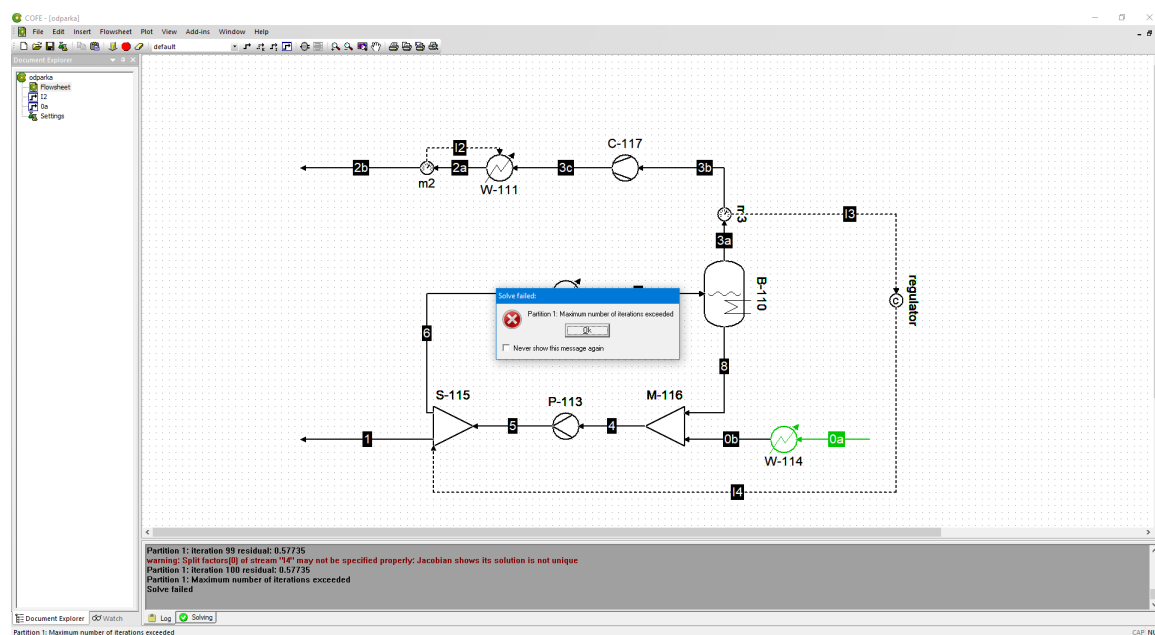
3.4.4 Spuštění simulace a zobrazení výsledků

Model je teď kompletně definovaný a můžeme přistoupit k simulaci. Nejprve v panelu nástrojů klikneme na **Validate**, čímž zkontrolujeme, zda jsou všechny komponenty ve flowsheetu propojené a mají zadané potřebné parametry. Kontrola nezaručí řešitelnost ani správnost výpočtů (obrázek 27).



Obrázek 27: Kontrola flowsheetu

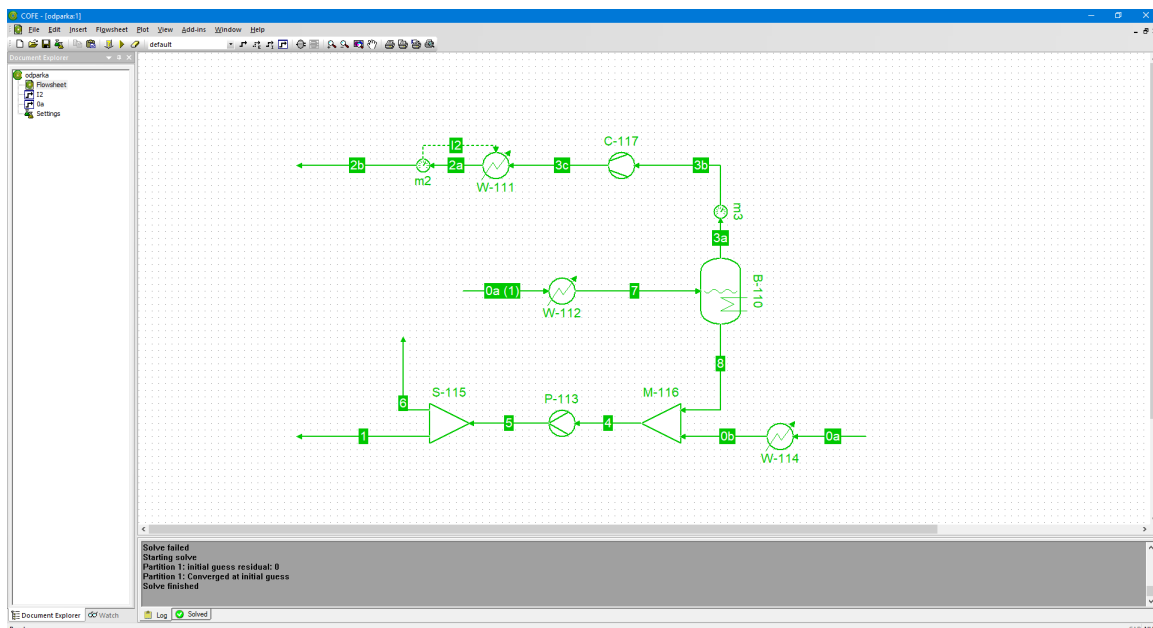
Po úspěšné kontrole můžeme kliknutím na ikonu **Solve** spustit vlastní simulaci. Na obrázku 28 vidíme, že byl výpočet přerušem pro překročení počtu iterací. V cirkulační smyčce totiž prozatím nic neprotéká.



Obrázek 28: Překročení počtu iterací

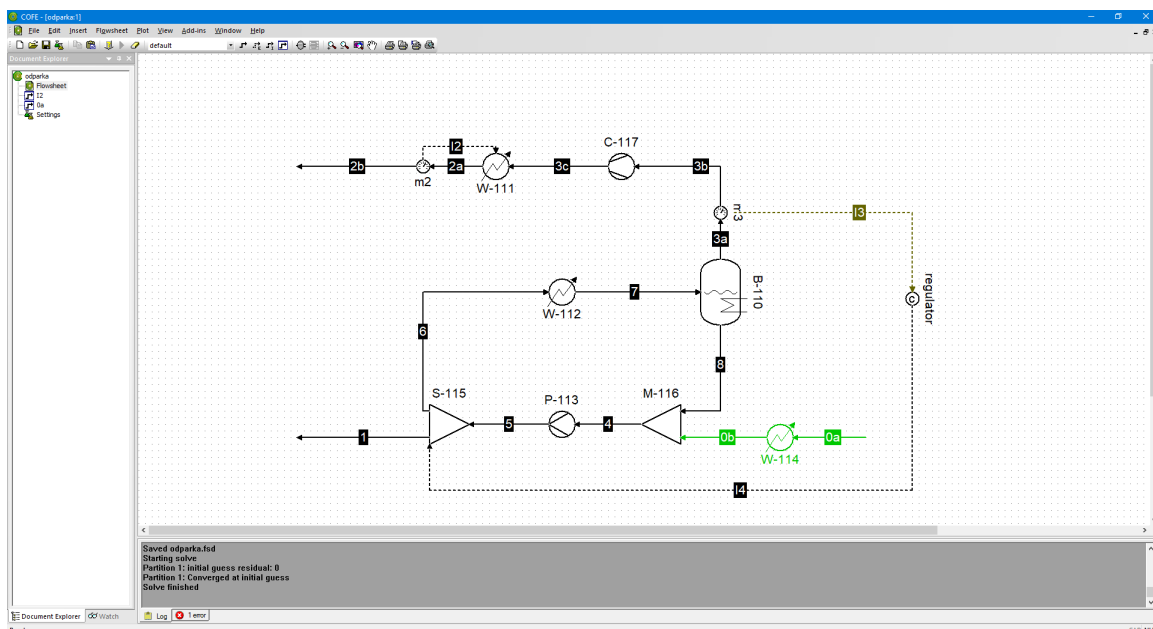
Musíme tedy odstranit regulaci děliče a rozpojit smyčku. Místo proudu 6

připojíme k výměníku W-112 například kopii proudu 0a a simulaci spustíme znovu (obrázek 29). Vidíme že je PFD schéma kompletně zelené. To znamená, že výpočet proběhl v pořádku.



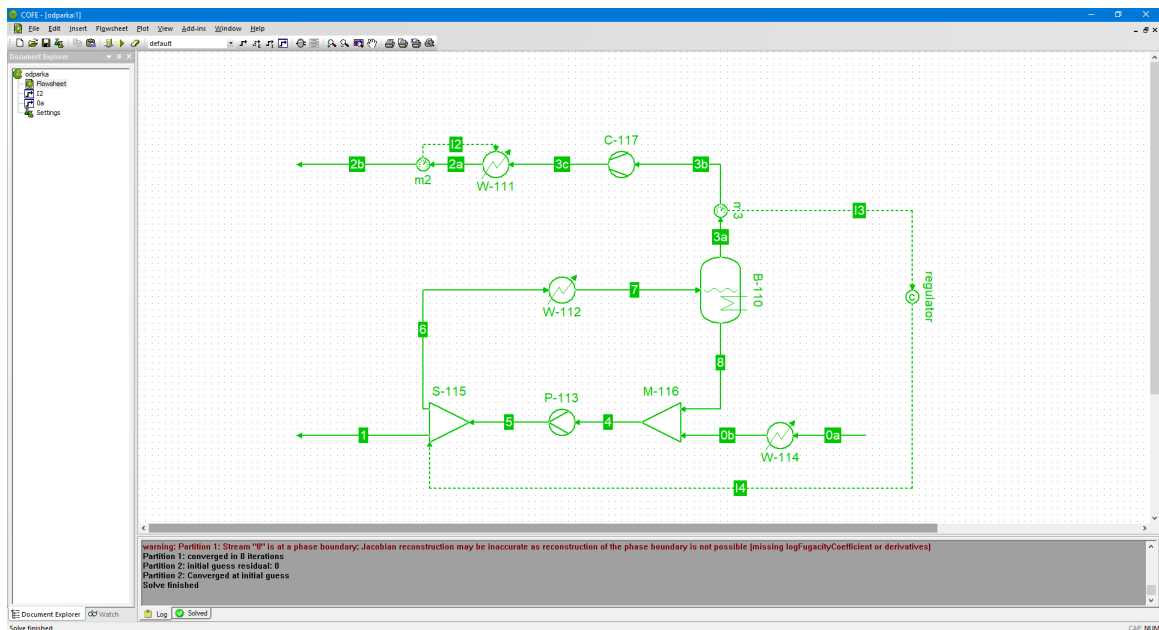
Obrázek 29: Zaběhnutí modelu bez cirkulační smyčky

Nyní již můžeme vrátit model do původního stavu (viz obrázek 30). Po znovuspouštění výpočtu se v hlášení objeví varování. Naposledy tedy zopakujeme simulaci a bilance je dokončena (obrázek 31).



Obrázek 30: Zapojení smyčky

Pokud budeme chtít měnit některé parametry modelu, musíme postupovat po menších krocích, aby nové řešení dosáhlo konvergence, při rozumném počtu iterací.



Obrázek 31: Funkční model s cirkulační smyčkou

Výsledky bilance je možné okamžitě prohlížet po njetí kurzorem na konkrétní proud, případně po rozkliknutí požadované komponenty (obrázek 32).

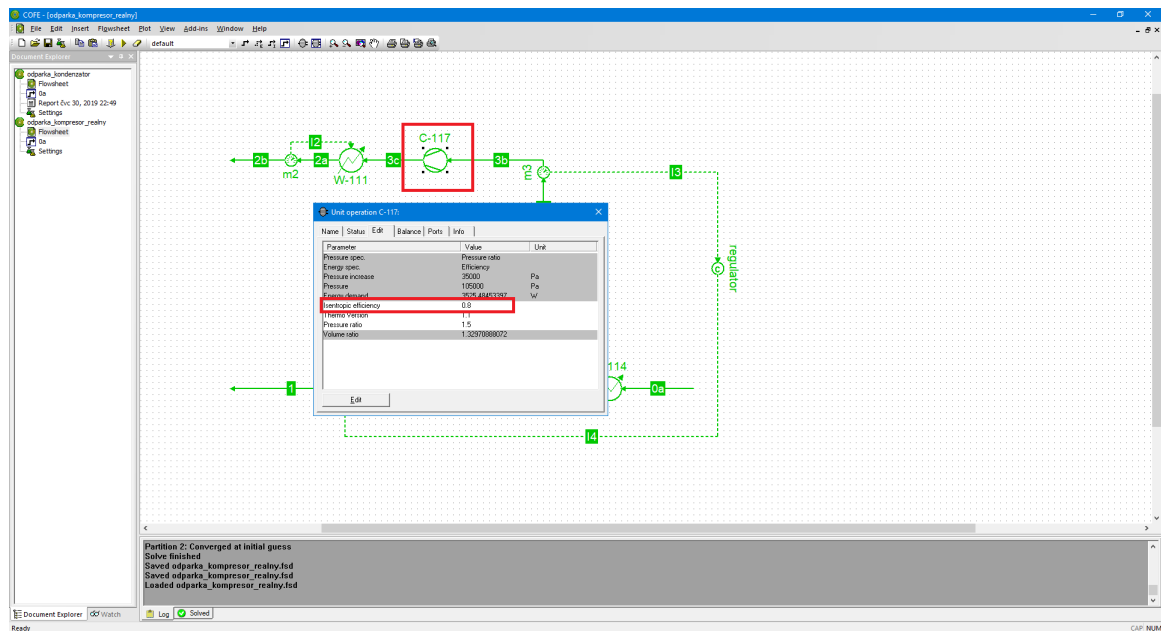
Unit operation W-111:			Unit operation W-112:		
Parameter	Value	Unit	Parameter	Value	Unit
Type	Temperature		Type	Temperature	
Outlet temperature	99.6059186113	°C	Outlet temperature	93	°C
Heat duty	32.5007181236	kW	Heat duty	90.945214062	kW
Pressure drop	5	kPa	Pressure drop	0	kPa
Thermo Version	1.1		Thermo Version	1.1	

(a) W-111

(b) W-112

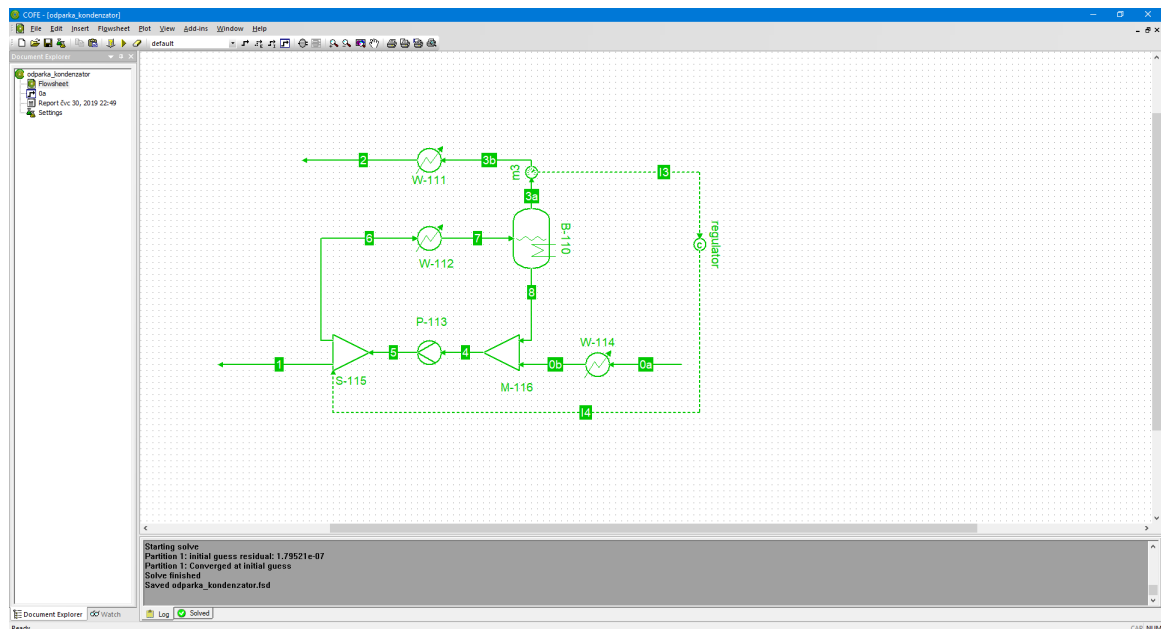
Obrázek 32: Výsledky bilance komponenty, v tomto případě výměníků W-111 a W-112

Pro variantu B, tedy kompresi brýdových par reálným kompresorem stačí upravit účinnost kompresoru C-117 na požadovanou hodnotu 0,8 (obrázek 33).



Obrázek 33: Odparka MVR - varianta B

Při tvorbě odparky s kondenzací brýdových par odstraníme kompresor C-117 a u výměníku W-111 odstraníme regulační obvod. V tomto případě bude stačit, když místo regulace nastavíme napevno výstupní teplotu zkondenzovaných par (25 °C) (obrázek 34).



Obrázek 34: Odparka s kondenzátorem - varianta C

3.4.5 Výsledky bilance

Kompletní výsledky bilance všech proudů a komponent modelu jsou uvedeny v tabulkách 2 a 3.

Pro požadovaný hmotnostní tok brýdových par (tj. množství odpařené vody) $0,04 \text{ kg s}^{-1}$ (144 kg h^{-1} , 3456 kg d^{-1}) je potřeba průtok cirkulační smyčkou $7,053 \text{ kg s}^{-1}$. Můžeme si všimnout, že na ohřev cirkulační smyčky ve výměníku W-112 je potřeba téměř 91 kW. Stlačením brýdových par ideálním kompresorem s kompresním poměrem 1,5 získáme 92,5 kW, případně reálným kompresorem se stejným kompresním poměrem a s účinností 0,8 získáme 93,2 kW. Obě varianty jsou tedy pro ohřev cirkulační smyčky dostačující. Z toho plyne, že stačí pro ohřev dodávat elektrickou energii na pohon kompresoru (varianta A 2,8 kW, varianta B 3,5 kW). I za cenu vyšších počátečních investic je proto použití MVR odparky výhodné.

Ochlazením brýdových par v kondenzátoru na teplotu $25 \text{ }^\circ\text{C}$ z nich musíme odvést chladicí vodou 102,18 kW.

Odparka ve všech třech variantách produkuje $0,03 \text{ kg s}^{-1}$ (108 kg h^{-1} , 2592 kg d^{-1}) koncentrátu při nátoke $0,07 \text{ kg s}^{-1}$.

Parametry proudů				
		Varianta A	Varianta B	Varianta C
0a	p [kPa]	70	70	70
	T [$^\circ\text{C}$]	20	20	20
	\dot{m} [kg s^{-1}]	0,07	0,07	0,07
0b	p [kPa]	70	70	70
	T [$^\circ\text{C}$]	89,9315	89,9315	89,9315
	\dot{m} [kg s^{-1}]	0,07	0,07	0,07
1	p [kPa]	120	120	120
	T [$^\circ\text{C}$]	89,9346	89,9346	89,9346
	\dot{m} [kg s^{-1}]	0,03	0,03	0,03
2	p [kPa]	100	100	70
	T [$^\circ\text{C}$]	99,6059	99,6059	25
	\dot{m} [kg s^{-1}]	0,04	0,04	0,04
3	p [kPa]	70	70	70
	T [$^\circ\text{C}$]	89,9315	89,9315	89,9315
	\dot{m} [kg s^{-1}]	0,04	0,04	0,04
3c	p [kPa]	105	105	×
	T [$^\circ\text{C}$]	126,866	135,645	×
	\dot{m} [kg s^{-1}]	0,04	0,04	×
4	p [kPa]	70	70	70
	T [$^\circ\text{C}$]	89,9315	89,9315	89,9315
	\dot{m} [kg s^{-1}]	7,083	7,083	7,083
5	p [kPa]	120	120	120
	T [$^\circ\text{C}$]	89,9346	89,9346	89,9346
	\dot{m} [kg s^{-1}]	7,083	7,083	7,083
6	p [kPa]	120	120	120
	T [$^\circ\text{C}$]	89,9346	89,9346	89,9346

	\dot{m} [kg s ⁻¹]	7,053	7,053	7,053
7	p [kPa]	120	120	120
	T [°C]	93	93	93
	\dot{m} [kg s ⁻¹]	7,053	7,053	7,053
8	p [kPa]	70	70	70
	T [°C]	89,9315	89,9315	89,9315
	\dot{m} [kg s ⁻¹]	7,013	7,013	7,013

Tabulka 2: Výsledky bilance 3 variant odparek - proudy (A - ideální kompresor, B - reálný kompresor, C - kondenzátor)

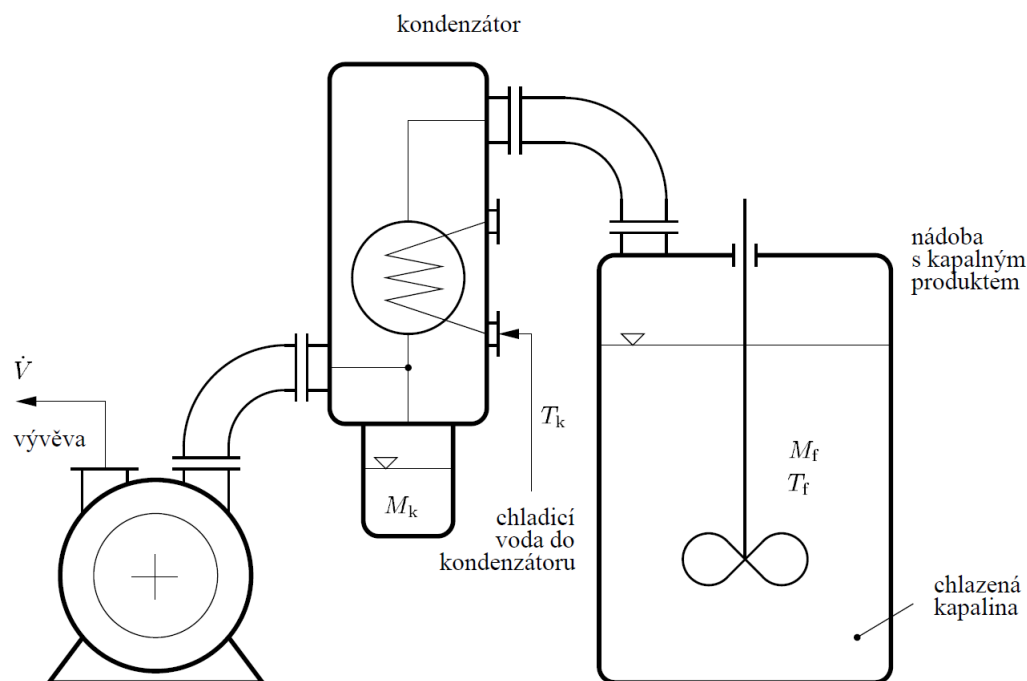
Parametry komponent				
		Varianta A	Varianta B	Varianta C
B-110	Δp [kPa]	50	50	50
	p [kPa]	70	70	70
	\dot{Q} [kW]	0	0	0
	T [°C]	89,9315	89,9315	89,9315
	ω_v [1]	0,00567	0,00567	0,00567
	\dot{m}_V [kg s ⁻¹]	0,04	0,04	0,04
	\dot{m}_L [kg s ⁻¹]	7,013	7,013	7,013
	W-111	Δp [kPa]	5	5
p [kPa]		100	100	70
\dot{m} [kg s ⁻¹]		0,04	0,04	0,04
T_1 [°C]		126,866	135,645	89,9315
T_2 [°C]		99,6059	99,6059	25
\dot{Q} [kW]		-92,5	-93,2	-102,18
W-112		Δp [kPa]	0	0
	p [kPa]	120	120	120
	\dot{m} [kg s ⁻¹]	7,053	7,053	7,053
	T_1 [°C]	89,9346	89,9346	89,9346
	T_2 [°C]	93	93	93
	\dot{Q} [kW]	90,95	90,95	90,95
	P-113	Δp [kPa]	50	50
p [kPa]		120	120	120
\dot{m} [kg s ⁻¹]		7,083	7,083	7,083
T_1 [°C]		89,9315	89,9315	89,9315
T_2 [°C]		89,9346	89,9346	89,9346
\dot{W} [W]		366,86	366,86	366,86
W-114		Δp [kPa]	0	0
	p [kPa]	70	70	70
	\dot{m} [kg s ⁻¹]	0,07	0,07	0,07
	T_1 [°C]	20	20	20
	T_2 [°C]	89,9315	89,9315	89,9315
	\dot{Q} [kW]	20,49	20,49	20,49
	S-115	Δp [kPa]	0	0
p [kPa]		120	120	120
T [°C]		89,9346	89,9346	89,9346
f [1]		0,995764	0,995764	0,995764
\dot{m}_5 [kg s ⁻¹]		7,083	7,083	7,083
\dot{m}_6 [kg s ⁻¹]		7,053	7,053	7,053
\dot{m}_1 [kg s ⁻¹]		0,03	0,03	0,03
	Δp [kPa]	0	0	0
	p [kPa]	70	70	70

M-116	T [°C]	89,9315	89,9315	89,9315
	\dot{m}_g [kg s ⁻¹]	7,013	7,013	7,013
	\dot{m}_{ob} [kg s ⁻¹]	0,07	0,07	0,07
	\dot{m}_1 [kg s ⁻¹]	7,083	7,083	7,083
C-117	p_{in}/p_{out} [1]	1,5	1,5	×
	p_1 [kPa]	70	70	×
	p_2 [kPa]	105	105	×
	\dot{m} [kg s ⁻¹]	0,04	0,04	×
	η [1]	1	0,8	×
	T_1 [°C]	89,9315	89,9315	×
	T_2 [°C]	126,866	135,645	×
	\dot{W} [kW]	2,82	3,52	×

Tabulka 3: Výsledky bilance tří variant odparek - komponenty (A - ideální kompresor, B - reálný kompresor, C - kondenzátor)

4 Simulace vakuového chlazení

Vakuové chlazení, nebo-li chlazení varem, je velmi rychlý způsob chlazení, používaný zejména v potravinářském a farmaceutickém průmyslu. Využívá přenosu velkého množství energie z kapaliny do jejích par při dosažení bodu varu (Dostal a Petera, 2004).



Obrázek 35: Schéma vakuového chlazení (zdroj: Dostál a Petera (2001))

4.1 Princip funkce

V článku, který publikovali Dostal a Petera (2004) je uvedeno, že při konkrétní teplotě a tlaku se kapalina mění na určité množství par. Parciální tlak těchto par je nižší, než tlak ve zkoumaném systému až do určité teploty, tzv. bodu varu. Při této teplotě se parciální tlak nasycených par vyrovná tlaku v systému a dojde k varu kapaliny. Při tomto ději dochází k odběru velkého množství energie (řádově $\text{MJ} \cdot \text{kg}^{-1}$) z kapaliny právě odcházejícími parami. Pokud nedodáváme do kapaliny další energii, její teplota klesne, parciální tlak par se dostane pod hodnotu tlaku v systému a var se zastaví. Jelikož je bod varu přímo úměrný tlaku, potřebujeme pro snížení teploty v systému, kontinuálně snižovat tlak. Tím klesne i hodnota bodu varu a kapalinu je možné chladit až na požadovanou teplotu.

4.2 Matematický model vakuového chlazení kapalin

Dostál a Petera (2001) provedli rešerši různých modelů vakuového chlazení, rozbor matematického popisu a naznačili numerické řešení. Dále uvedené modely

porovnali.

Jedním z modelů, který Dostál a Petera (2001) popisují je model Žitný a Šesták (1991). Tento model je rovnovážný, tzn. předpokládá, že v každém okamžiku existuje termodynamická rovnováha mezi kapalinou a jejími nasycenými parami. Dále model předpokládá, že se v nádobě s chlazenou kapalinou nacházejí pouze její nasycené páry. Posledním předpokladem je nekonečně velká teplosměnná plocha v kondenzátoru. Tím je dána konstantní teplota směsi vzduchu a nasycených par. Tato teplota se rovná teplotě chladiwa v kondenzátoru. V modelu je také zohledněn vliv netěsností v soustavě na celkový tlak uvnitř vakuové nádoby a na nejnižší dosažitelnou teplotu.

Následující diferenciální rovnice byly odvozeny v práci Dostál a Petera (2001). Jsou tři a popisují průběh tlaku inertů v kontrolním objemu, hmotnost chlazené kapaliny a hmotnost zkondenzovaných par chlazené kapaliny.

Pro průběh tlaku inertů v kontrolním objemu (viz obrázek 36) platí bilance hmotnosti inertů

$$V_k \frac{dp_i}{dt} = r_i T_k \dot{M}_{in} - \dot{V} p_i, \quad (12)$$

kde V_k je objem kondenzátoru, T_k je teplota inertů, \dot{M}_{in} je hmotnostní průtok inertů netěsnostmi a \dot{V} je charakteristika vývěvy.

Pro průběh hmotnosti chlazené kapaliny platí

$$c_f M_f \frac{dT_f}{dt} = \frac{dM_f}{dt} r_{LG}, \quad (13)$$

kde M_f je hmotnost chlazené kapaliny a T_f její teplota.

A nakonec pro průběh hmotnosti kondenzovaných par platí

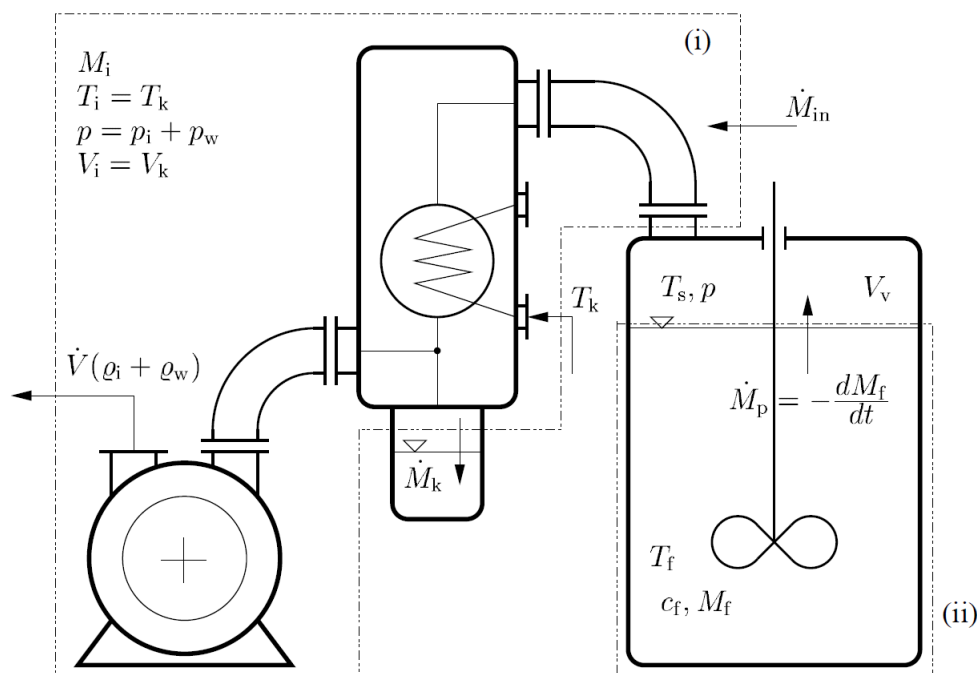
$$\frac{dM_k}{dt} = -\frac{dM_f}{dt} - \dot{V} \rho_w, \quad (14)$$

kde M_k je hmotnost zkondenzovaných par chlazené kapaliny a \dot{V} charakteristika vývěvy.

V rovnicích je několik proměnných, které lze vyjádřit jako funkci tlaku. Hmotnostní tok inertů netěsnostmi $\dot{M}_{in}(p_i)$ a objemový tok plynů odváděných z vakuového prostoru, nebo-li charakteristiky vývěvy $\dot{V}(p_i)$. Tyto závislosti budou popsány a odvozeny v následujících kapitolách.

Teplota chlazené kapaliny je jednoznačně dána velikostí celkového tlaku dle křivky nasycených par

$$T_f = T(p). \quad (15)$$



Obrázek 36: Schéma modelu Žitný a Šesták (zdroj: Dostál a Petera (2001))

kterou můžeme popsat například s pomocí Antoineovy rovnice. Celkový tlak je pak dán součtem parciálního tlaku inertů a parciálního tlaku vodních par

$$p = p_i + p_w . \quad (16)$$

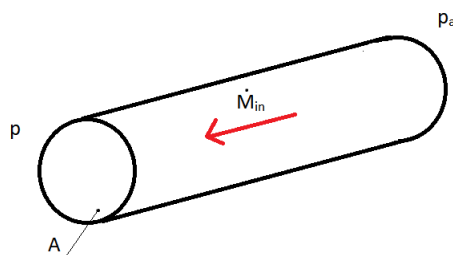
4.3 Matematické modely netěsností

Netěsnosti v zařízení, vznikající zejména v přírubových spojích a různých těsněních, mají velký vliv na udržitelnost podtlaku. Dostal a Petera (2004) uvádí, že netěsnosti mají velký vliv na rychlost poklesu teploty a taky na minimální dosažitelnou teplotu. V této kapitole se budeme věnovat různým přístupům k vytvoření matematického modelu netěsností a porovnání několika těchto modelů.

Vycházíme ze situace, kdy jsme z vakuového prostoru odsáli veškerý vzduch a celkový tlak $p = 0$. Od tohoto okamžiku budeme po určitý čas, v našem případě 5000 s (cca 80 minut) měřit nárůst celkového tlaku v nádobě. Tím získáme informace o vlivu netěsností na parciální tlak inertů (vzduchu) při procesu chlazení. Tento pomyslný experiment provedeme analytickým, ale i numerickým řešením několika matematických modelů netěsností.

4.3.1 Model trubky s konstantním průřezem, laminární proudění

Netěsnosti v celém systému si lze představit jako trubku na obrázku 37, jejíž průřez odpovídá součtu průřezů všech těchto netěsností. Při laminárním proudění



Obrázek 37: Schematické znázornění modelu netěsností

bude mít velký vliv na průtok vazkost vzduchu. Objemový tok pro tento případ je tedy dán Hagen-Poiseuillovým vztahem

$$\dot{V}_{\text{in}} = \frac{\pi D^4}{128\mu L} \Delta p. \quad (17)$$

Pro hmotnostní tok vzduchu (inertů) přitékajícího netěsnostmi pak platí

$$\dot{M}_{\text{in}} = \frac{\pi D^4}{128\mu L} \rho_i \Delta p, \quad (18)$$

kde

$$\Delta p = p_a - p, \quad (19)$$

kde p_a je atmosférický tlak.

Hmotnost inertů v kontrolním objemu vakuové nádoby je dána nestacionární hmotnostní bilancí

$$\frac{dM_i}{dt} = -\dot{M}_{\text{in}}. \quad (20)$$

Hmotnost inertů a tedy i jejich hmotnostní tok lze s použitím stavové rovnice ideálního plynu vyjádřit jako funkci parciálního tlaku inertů p_i

$$M_i = \frac{V_k}{r_i T_k} p_i. \quad (21)$$

dosazením rovnice (21) do (20) získáme diferenciální rovnici se separovatelnými proměnnými

$$\frac{V_k}{r_i T_k} \frac{dp_i}{dt} = -\dot{M}_{\text{in}} = -\frac{\pi D^4}{128\mu L} \rho_i (p_a - p_i). \quad (22)$$

Její řešení získáme časovou závislost parciálního tlaku inertů ve vakuové nádobě. Ten z počátečního nulového stavu (vzduch je na začátku procesu z nádoby odčerpán) vlivem přítoku vzduchu netěsnostmi, stoupá, dokud není dosaženo rovnováhy, tedy vyrovnání celkového tlaku v nádobě a atmosférického tlaku, za předpokladu, že je nádoba umístěna v atmosféře. Tato závislost je popsána vztahem

$$p_i = p_a \left(1 - e^{-\frac{\pi D^4 \rho_i r_i T_k}{128 V_k \mu L} t} \right). \quad (23)$$

4.3.2 Model trubky s konstantním průřezem, turbulentní proudění

Při turbulentním proudění je účinek vazkých sil zanedbatelný a tlakový spád vlivem ztrát vyjádříme Darcy-Weisbachovým vztahem

$$\Delta p = \lambda \frac{L}{D} \rho \frac{\bar{u}^2}{2}, \quad (24)$$

ze kterého si vyjádříme střední rychlost

$$\bar{u} = \sqrt{\frac{2D}{\lambda L \rho_i} \Delta p} \quad (25)$$

a hmotnostní tok vzduchu vyjádříme jako

$$\dot{M}_{\text{in}} = \sqrt{\frac{2D}{\lambda L \rho} \Delta p} A \rho_i. \quad (26)$$

Dosazením rovnice (26) do (20) a vyjádřením hmotností inertů rovnicí (21), získáme diferenciální rovnici se separovatelnými proměnnými pro turbulentní režim proudění netěsnostmi

$$\frac{V_k}{r T_k} \frac{dp_i}{dt} = -\dot{M}_{\text{in}} = -\sqrt{\frac{2D}{\lambda L \rho_i} (p_a - p_i)} A \rho_i. \quad (27)$$

Řešením této rovnice je opět průběh parciálního tlaku inertů ve vakuové nádobě, daný vztahem

$$p_i = p_a - \frac{I^2 (t + J)^2}{4N^2} \quad (28)$$

kde $I = \frac{\pi^2 D^5 \rho_i}{8 \lambda L}$, $J = \frac{16 V_k \lambda L}{r T_k \pi^2 D^5 \rho_i} \sqrt{p_a}$ a $N = \frac{V_k}{r T_k}$.

Odvozené vztahy pro laminární a turbulentní model (rovnice 23 a 28) můžeme nyní přepsat do formy zdrojového kódu a v programu OpenModelica provést výpočet.

Nejprve je potřeba zavést veličiny používané při výpočtu a těm, které známe přiřadit hodnoty

```
model model_netesnosti
//DEFINICE VELICIN
type Hmotnost = Real(unit="kg", min=0);
type Teplota = Real(unit="K", min=-60000);
type Tlak = Real(unit="Pa", min=-600000);
type Objem = Real(unit="m3", min=0);
```

```

type Plocha = Real(unit="m2", min=0);
type Hustota = Real(unit="kg/m3", min=0);
type MernaTepelnaKapacita = Real(unit="J/kg.K", min=0);
type MolarniHmotnost= Real(unit="kg/kmol", min=0);
type PoissonovaKonstanta = Real(unit="[1]", min=0);
type HmotnostniTok = Real(unit="kg/s", min=0);
type UniPlynKonst = Real(unit="J/kg.K.mol", min=0);
type PlynKonst = Real(unit="??", min=0);
type Konstanta = Real(unit="", min=-50000);
type DynVizkozita = Real(unit="Pa.s", min=0);
type Delka = Real(unit="m", min=0);
type Prumer = Real(unit="m", min=0);
type TreciZtraty = Real(unit="", min=0);

//PARAMETRY SYSTEMU
parameter Hmotnost M_f0 = 500 "pocatecni hmotnost chlazení kapaliny";
parameter Hustota rho = 1.2 "hustota vzduchu";
parameter MernaTepelnaKapacita c_f = 4180 "merna tepelna
kapacita chlazené kapaliny (voda)";
parameter Tlak p_a = 101325 "okolní atmosferický tlak";
parameter Objem V_k = 0.7 "objem kondenzátoru a okolních armatur";
parameter Plocha A = 4*10^(-7) "plocha netesnosti";
parameter Teplota T_k = 293.15 "teplota chladicí vody v kondenzátoru";
parameter MolarniHmotnost Me = 28.96 "molární hmotnost okolního vzduchu";
parameter PoissonovaKonstanta kappa = 1.4 "poissonova konstanta";
parameter UniPlynKonst R = 8314 "univerzální plynová konstanta";
parameter PlynKonst r = 287 "plynová konstanta";
parameter DynVizkozita mi = 1.71*10^(-5) "dynamická viskozita";
parameter Delka L = 0.01 "tloušťka stěny";
parameter TreciZtraty lambda = 0.03 "součinitel třech ztrat";

```

Následně definujeme které z parametrů jsou konstantami a které proměnnými.

```

//KONSTANTY
Konstanta K;
Konstanta I;
Konstanta C1;
Konstanta K1;
Konstanta I1;
constant Real pi = 2 * Modelica.Math.asin(1.0);
constant Real e = Modelica.Math.exp(1.0);

//PROMENNE
Prumer D;
Tlak p;
Tlak p_vyp_lam;
Tlak p_vyp_turb;
Tlak p_w;
Tlak pi_trubka_lam "prubeh tlaku inertu ve vakuové nadobě";
Tlak pi_trubka_turb "prubeh tlaku inertu ve vakuové nadobě";
HmotnostniTok Mtok_in_lam "tok vzduchu vnikající netesnostmi";
HmotnostniTok Mtok_in_turb "tok vzduchu vnikající netesnostmi";
HmotnostniTok Mtok_in_stl "tok vzduchu vnikající netesnostmi";

```

Nakonec napíšeme odvozené rovnice a model je kompletní.

```

//ROVNICE
equation
p_w = 0;

D = sqrt((4*A)/pi);
K = (pi*D^4*rho*r*T_k)/(128*mi*V_k*L);
I = -log(p_a - p_w)/K;

K1 = sqrt((2*D)/(lambda*L*rho))*(rho*pi*D^2/4);
C1 = V_k/(r*T_k);
I1 = -2*(C1/K1)*sqrt(p_a - p_w);

pi_trubka_lam = p_a - p_w - e^(-K*(time+I));
p_vyp_lam = pi_trubka_lam + p_w;

pi_trubka_turb = p_a - p_w - ((K1^2)*(time^2 + 2*time*I1 + I1^2))/(4*C1^2);
p_vyp_turb = pi_trubka_turb + p_w;

p = -28.95*time + p_a;
Mtok_in_lam = ((pi*D^4 * rho)/(128*mi*L))*(p_a - p);
Mtok_in_turb = sqrt(((rho * pi^2*D^5)/(lambda*L*8))*(p_a - p));
Mtok_in_stl = A*(p_a * sqrt((2*Me)/(R * T_k)) * sqrt((kappa/(kappa - 1)) *
(1 - (p/p_a)^((kappa-1)/kappa)) * (p/p_a)^(2/kappa)));

end model_netesnosti;

```

U numerického výpočtu budeme postupovat stejně, tedy nejprve definujeme parametry systému, poté přiřadíme hodnoty a určíme konstanty a proměnné. Nyní však místo odvozených rovnic budeme psát rovnou diferenciální rovnice (rovnice 22 a 27) popisující děj. Tyto rovnice musíme ještě doplnit počátečními podmínkami, tedy, že tlak v nádobě na začátku procesu je nulový. Program následně podle nastavené numerické metody tyto rovnice vyřeší.

```

//POCATECNI PODMINKY
initial equation
  p_iL = 0;
  p_iT = 0;
  p_iS = 0;

//ROVNICE
equation
  D = sqrt((4 * A) / pi);
  //parcialni tlak nasycenych par pri teplote chladici vody;
  pw = 0;

  //netesnosti - laminarni proudeni v trubce - prubeh tlaku ve vakuove;
  nadobe s netesnostmi ze stavu p_i = 0 do p_i = pa-pw
  der(p_iL) = r * Tk * pi * D ^ 4* rhoi / (128 * mi * L* Vk)
  * (pa - pw - p_iL);

  //netesnosti - turbolentni proudeni v trubce - prubeh tlaku ve vakuove
  nadobe s netesnostmi ze stavu p_i = 0 do p_i = pa-pw;
  der(p_iT) = ((r * Tk )/ Vk) * sqrt(2 * D / (lambda * L * rhoi)
  * (pa - pw - p_iT)) * A * rhoi;

  //netesnosti - stl proudeni konvergentni tryskou - prubeh tlaku

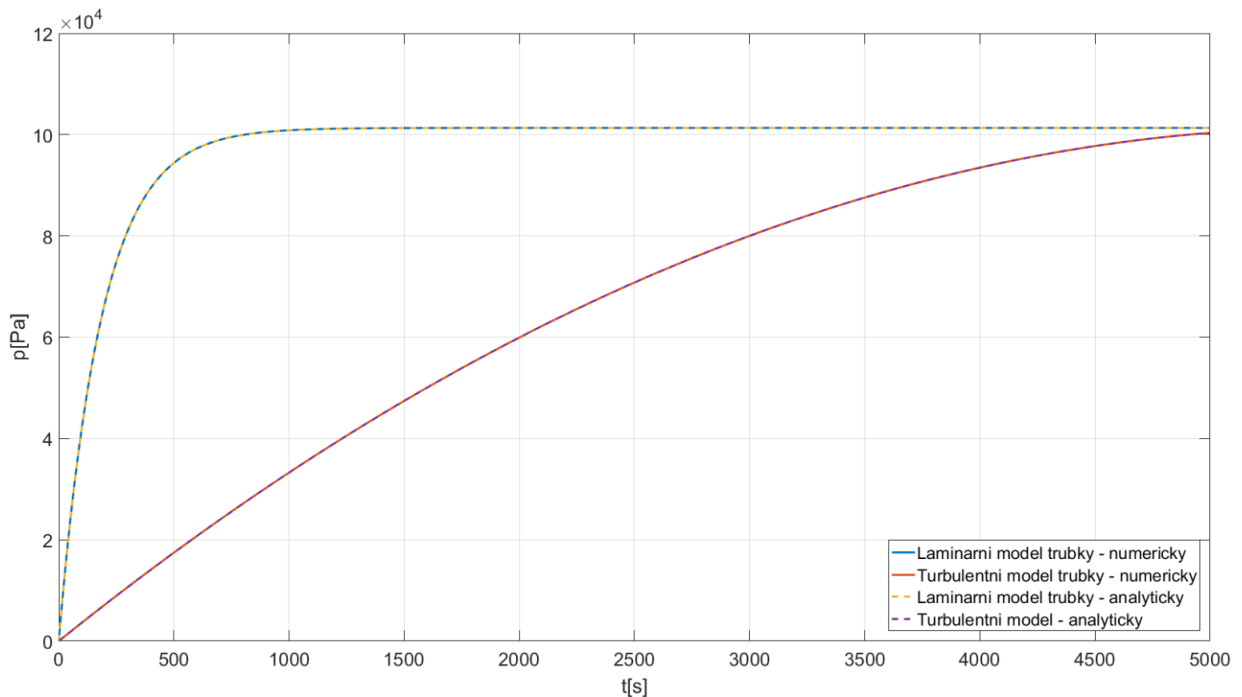
```

```

ve vakuove nadobe s netesnostmi ze stavu p_i = 0 do p_i = pa-pw;
der (p_iS) = ((r*Tk)/Vk)*A*pa*sqrt((2*Me)/(R*Tk))*sqrt((kappa/(kappa-1))
*(1-((pw+p_iS)/pa)^((kappa-1)/kappa))*((pw+p_iS)/pa)^(2/kappa));

```

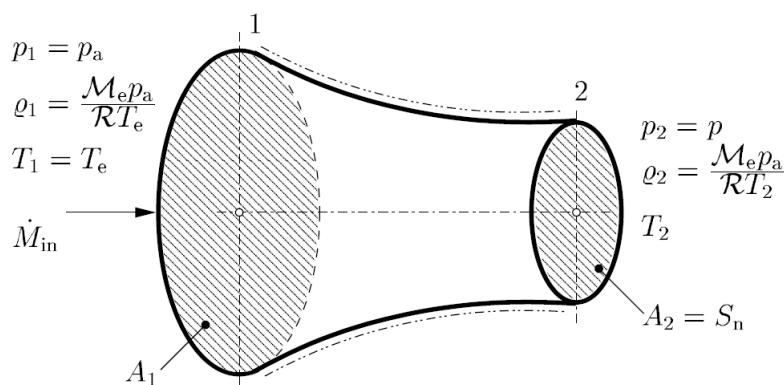
Jak můžeme vidět na obrázku 38, analytický výpočet se přesně shoduje s výsledky numerické simulace v programu OpenModelica. Zároveň je z grafu patrné, že mezi laminárním a turbulentním modelem je značný rozdíl. Ve výsledcích simulace vakuového chlazení provedených pro různé modely netěsností proto očekáváme značné odlišnosti.



Obrázek 38: Výsledky analytických výpočtů laminárního a turbulentního modelu v porovnání s numerickými

4.3.3 Model konvergentní trysky, nevazké stlačitelné proudění

Po skončení procesu vakuového chlazení, může poměr mezi atmosférickým a vnitřním tlakem dosahovat značných hodnot (až 10). Z toho důvodu může při chlazení dojít k jevu, který se nazývá aerodynamické zahlcení. Tento jev popisuje teorie toku nevazké stlačitelné tekutiny konvergentní tryskou (Dostál a Petera, 2001).



Obrázek 39: Konvergentní tryska (zdroj: Dostál a Petera (2001))

Podle Sazima et al. (1989, podle Dostál a Petera (2001)) lze vyjádřit hmotnostní průtok plynu tryskou

$$\dot{M}_{in} = S_n p_a \sqrt{\frac{2M_e}{RT_e}} \sqrt{\frac{\kappa}{\kappa - 1} \left[1 - \left(\frac{p}{p_a} \right)^{(\kappa-1)/\kappa} \right] \left(\frac{p}{p_a} \right)^{2/\kappa}}. \quad (29)$$

Nyní můžeme rovnici (29) dosadit do rovnice (20) a spolu s rovnicí (21) získáme diferenciální rovnici pro model konvergentní trysky

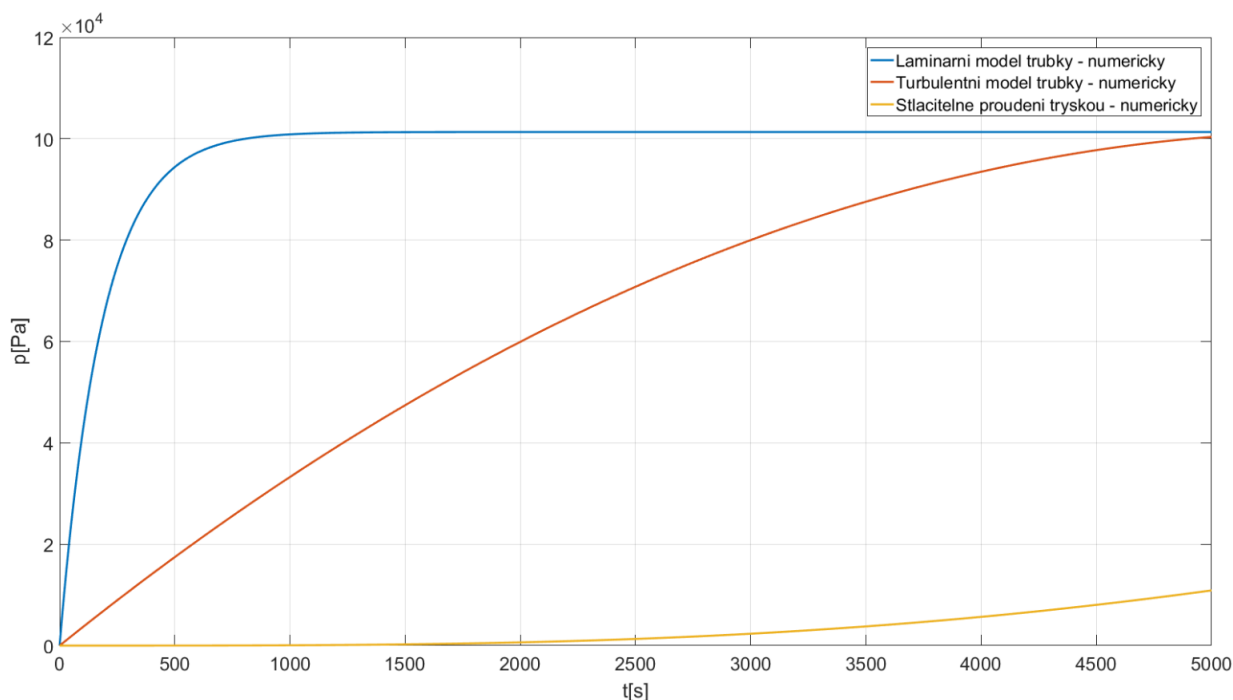
$$\frac{V_k}{rT_k} \frac{dp_i}{dt} = S_n p_a \sqrt{\frac{2M_e}{RT_k}} \sqrt{\frac{\kappa}{\kappa - 1} \left[1 - \left(\frac{p_i}{p_a} \right)^{(\kappa-1)/\kappa} \right] \left(\frac{p_i}{p_a} \right)^{2/\kappa}}. \quad (30)$$

Tuto rovnici doplníme do zdrojového kódu pro numerický výpočet a spustíme simulaci.

Na obrázku 40 jsou znázorněny výsledky simulace průběhu tlaku ve vakuové nádobě pro všechny výše uvedené modely netěsností, za předpokladu stejné plochy netěsností. V případě reálného experimentu by tyto plochy byly odlišné a byly by stanoveny na základě měření skutečného průběhu tlaku v nádobě.

Z počátečního tlaku ve vakuové nádobě, který je nulový, se vlivem netěsností tlak postupně zvyšuje. U laminárního modelu se celkový tlak v nádobě již po 1500 s vyrovnal s tlakem atmosférickým. U turbulentního modelu k vyrovnání tlaků ve

stanovený čas nedošlo, avšak tlak v nádobě je velmi blízko atmosférickému tlaku. V případě modelu stlačitelného proudění tryskou se tlak po 5000 s zvýšil zhruba na 10 kPa.



Obrázek 40: Porovnání modelů netěsností (laminární a turbulentní proudění v trubce, stlačitelné proudění tryskou)

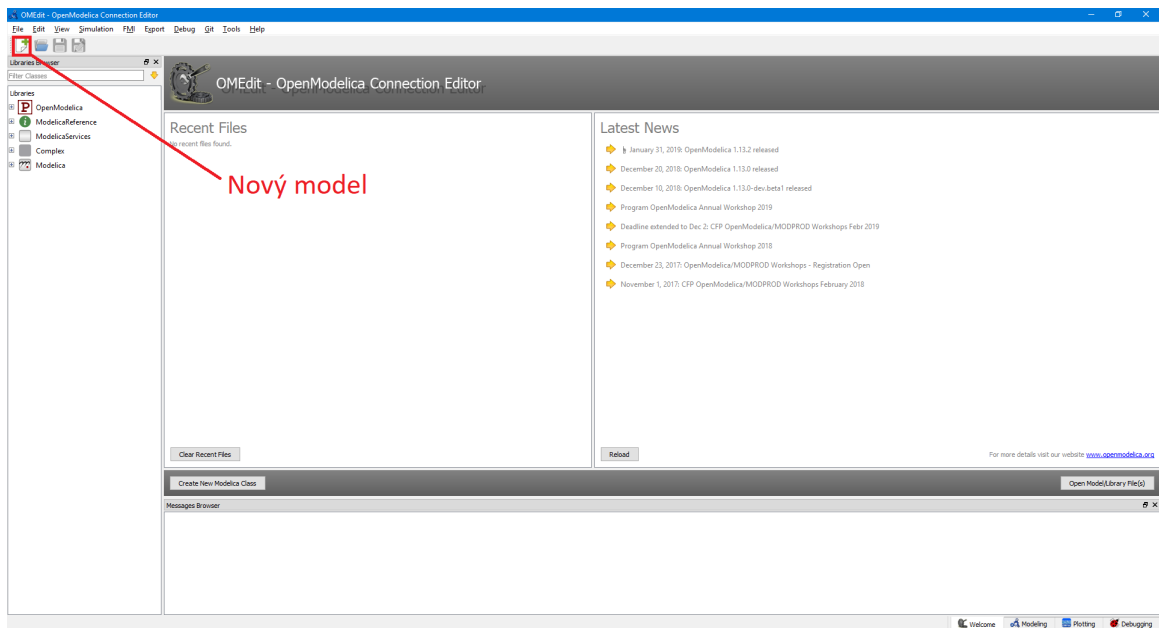
4.4 Tvorba simulace v programu OpenModelica

Již v předchozích kapitolách jsme mohli vidět ukázky zdrojového kódu v jazyce Modelica. Psaní kódu je ale jen jedna z několika možností, jak lze simulaci v této platformě tvořit. Nyní se detailněji zaměříme na uživatelské rozhraní programu OpenModelica a krok po kroku si popíšeme tvorbu modelu vakuového chlazení.

Na závěr této kapitoly si ukážeme výsledky simulace s různými modely netěsností, které jsme popsali v předchozí kapitole. Také si ukážeme, jak takový model vypadá, pokud ho vytvoříme z předdefinovaných bloků matematických operací.

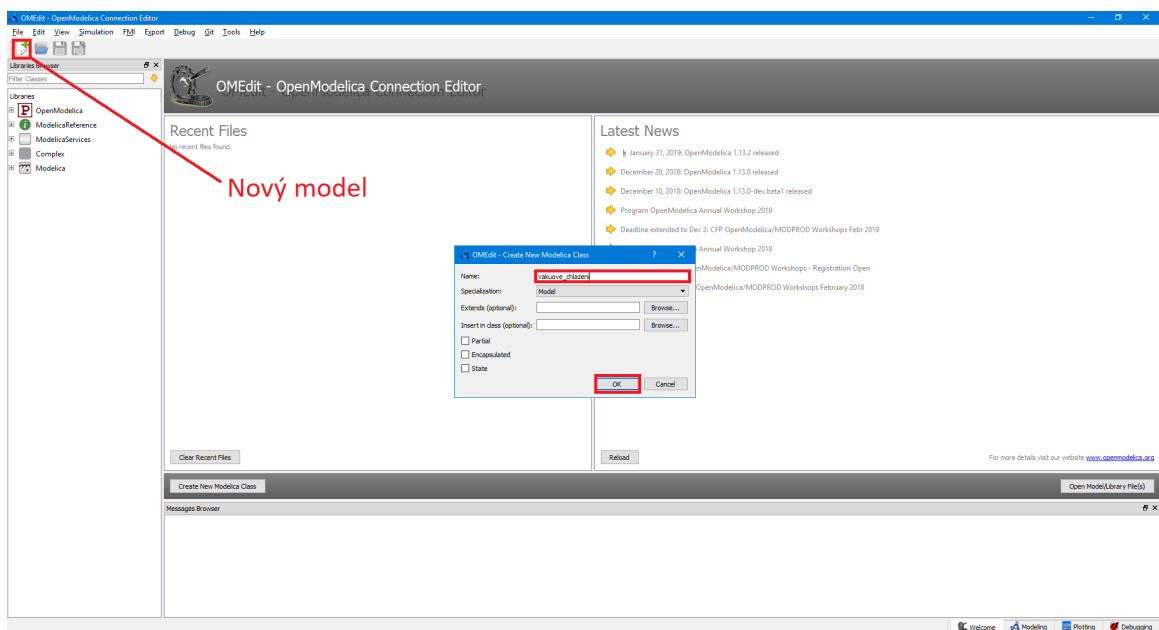
4.4.1 Uživatelské rozhraní a založení modelu

Po spuštění programu se objeví úvodní obrazovka s posledními otevřenými dokumenty a zprávami o aktualizacích. Kliknutím na ikonu označenou na obrázku 41 založíme nový model.



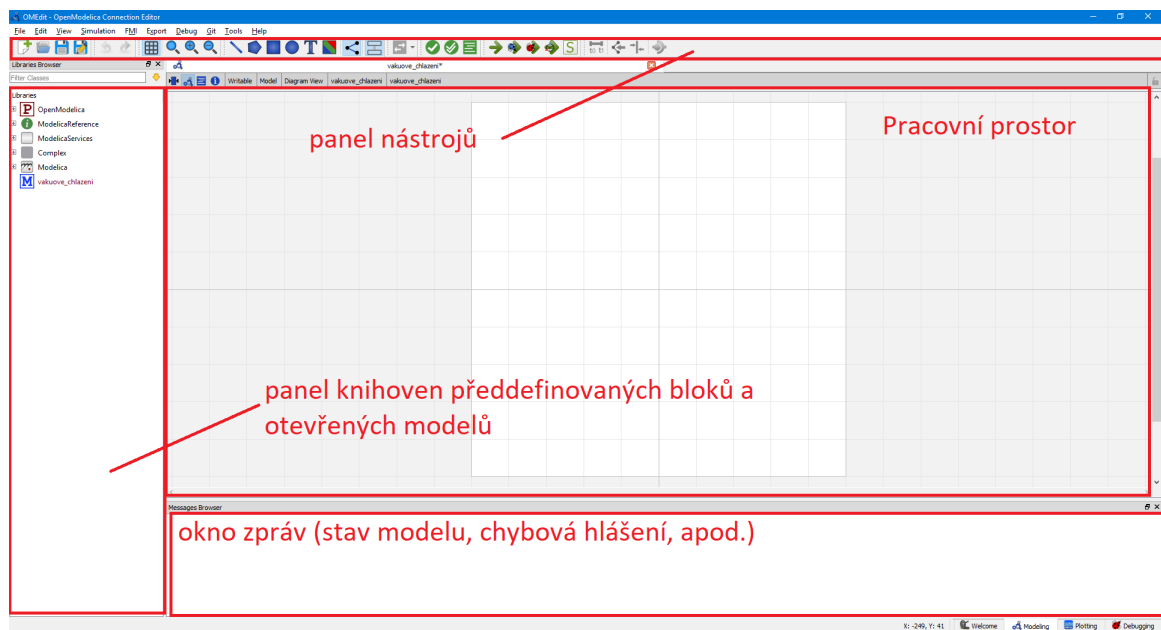
Obrázek 41: Úvodní obrazovka OpenModelica

Objeví se dialogové okno, kde vyplníme název modelu a potvrdíme (obrázek 42).



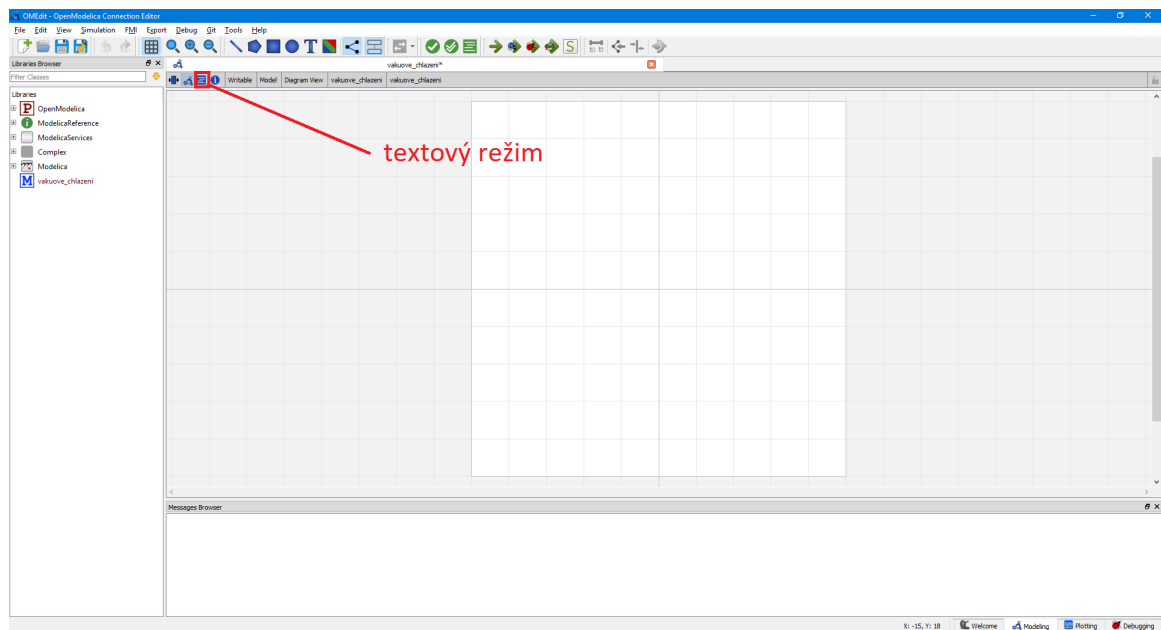
Obrázek 42: Založení modelu

Po odkliknutí se zobrazí uživatelské rozhraní tvorby modelu. Jednotlivé sekce jsou označeny na obrázku 43. V panelu vlevo jsou načteny knihovny s různými předdefinovanými komponenty a otevřené modely. S těmito komponenty lze tvořit různá zařízení, od návrhu řídicích systémů, až po simulaci funkce elektráren či zpracovatelských linek. Na závěr této kapitoly si ukážeme, jak vypadá náš model sestavený pomocí těchto komponent.



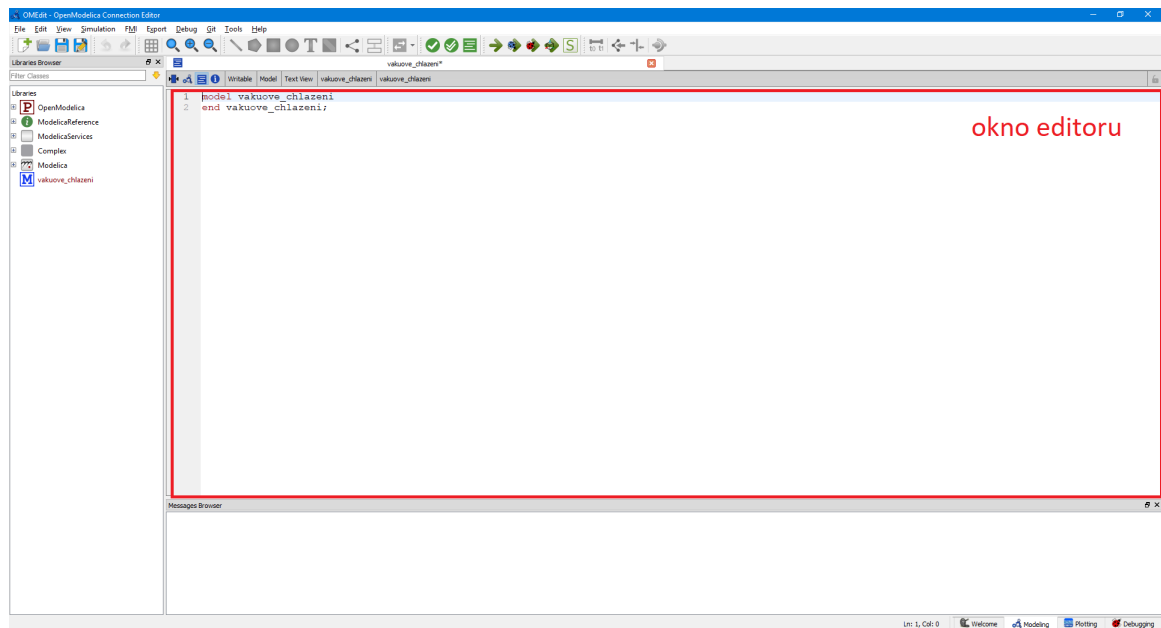
Obrázek 43: Uživatelské rozhraní

Jelikož budeme tvořit náš model pomocí zdrojového kódu, potřebujeme přepnout rozhraní do textového editoru. Přepnutí provedem kliknutím na ikonu označenou na obrázku 44.



Obrázek 44: Přepnutí na textový editor

Zobrazí se nám pole, do kterého budeme zapisovat kód v jazyce Modelica (obrázek 45).



Obrázek 45: Textové rozhraní

4.4.2 Zápis kódu

Nyní můžeme začít se samotnou tvorbou modelu. Nejprve je třeba definovat termodynamický systém, tzn. musíme ho popsat veličinami. Ty se definují příkazem *type* dále se uvede název, typ proměnné, její jednotka a případně minimální hodnota.

```
model vakuove_chl_FUNKCNI
//DEFINICE VELICIN
type hmotnost = Real(unit="kg", min=0);
type teplota = Real(unit="K", min=0);
type tlak = Real(unit="Pa");
type objem = Real(unit="m3", min=0);
type plocha = Real(unit="m2", min=0);
type hustota = Real(unit="kg/m3", min=0);
type mtepkap = Real(unit="J/kg.K", min=0);
type vyparneteplo = Real(unit="J/kg", min=0);
type molhmotnost= Real(unit="kg/kmol", min=0);
type poisson = Real(unit="[1]", min=0);
type vyveva = Real(unit="m3/s");
type tokhmot = Real(unit="kg/s");
type delka = Real(unit="m", min=0);
type prumer = Real(unit="m", min=0);
```

V těchto veličinách nyní vyjádříme parametry systému. Zadané parametry jsou uvedeny v tabulce 4.

Počáteční hmotnost chlazené kapaliny	M_{f0}	500	kg
Počáteční teplota chlazené kapaliny	T_{f0}	100	°C
Hustota chlazené kapaliny (voda)	ρ_f	1000	kg m ⁻³
Měrná tepelná kapacita chlazené kapaliny (voda)	c_f	4180	J kg ⁻¹ K ⁻¹
Výparné teplo vody (při 60 °C)	r_{LG}	2358410	J kg ⁻¹
Okolní atmosférický tlak	p_a	101325	Pa
Objem kondenzátoru a okolních armatur	V_k	0,7	m ³
Charakteristika vývěvy	\dot{V}	$0,0377[1 - e^{-0,001184(p-2252)}]$	m ³ s ⁻¹
Plocha netěsností	A	$4 \cdot 10^{-7}$	m ²
Teplota chladicí vody v kondenzátoru	T_k	20	°C
Teplota okolního vzduchu	T_e	20	°C
Molární hmotnost okolního vzduchu	M_e	28,96	kg kmol ⁻¹
Molární hmotnost vody, vodních par	M_w	18,015	kg kmol ⁻¹
Hustota inertů (vzduchu)	ρ_i	1,2	kg m ⁻³
Poissonova konstanta	κ	1,4	1
Dynamická viskozita vzduchu (do 50 °C)	μ	$1,71 \cdot 10^{-7}$	Pa s
Součinitel třecích ztrát	λ	0,03	1
Délka modelové netěsnosti	L	0,01	m

Tabulka 4: Parametry systému

```
//PARAMETRY SYSTEMU
constant Real pi = 2 * Modelica.Math.asin(1.0);
constant Real e = Modelica.Math.exp(1.0);
constant Real cf = 4180 "merna tepelna kapacita [J/kg.K]";
constant Real kappa = 1.4 "Poissonova konstanta";
constant Real rLG = 2358410 "vyparne teplo vody pri 60 st.C [J/kg]";
constant Real r = 287;
constant Real R = 8314 "molarni plynova konstanta [kJ/kmol.K]";
constant Real mi = 1.71*10^(-5) "dynamicka vizkozita vzduchu [Pa.s]";
constant Real Me = 28.96 "molarni hmotnost okolniho vzduchu [kg/kmol]";
constant Real rhof = 1000 "hustota chlazene kapaliny (voda) [kg/m^3]";
constant Real rhoi = 1.2 "hustota inertu [kg/m^3]";
constant Real lambda = 0.03 "soucinitel trecich ztrat";

parameter hmotnost Mf0 = 500 "pocatecni hmotnost chlazení kapaliny";
parameter tlak pa = 101325 "okolni atmosfericky tlak";
parameter objem Vk = 0.7 "objem kondenzatoru a okolnich armatur";
parameter plocha A = 4*10^(-7) "plocha netesnosti";
parameter teplota Tk = 293.15 "teplota chladici vody v kondenzatoru";
parameter delka L = 0.01 "teplota chladici vody v kondenzatoru";
```

Máme tedy definovaný termodynamický systém a nastavené parametry. Nyní potřebujeme definovat proměnné, které budeme chtít spočítat. Jednoduše tedy napíšeme nejprve veličinu, ke které se má nová proměnná přiřadit a následně pojmenujeme proměnnou

```
//PROMENNE
prumer D "prumer modelove netesnosti";
hustota rhow "hustota par chlazene kapaliny";
vyveva Vtok "charakteristika vyvevy";
```

```

tlak p "celkový tlak v nadobe";
tlak pw "parciální tlak par chlazené kapaliny";
tlak p_i "parciální tlak inertu";
tlak p_iL "vliv netesnoti na tlak inertu v nadobe (laminarni model)";
tlak p_iT "vliv netesnoti na tlak inertu v nadobe (turbolentni)";
tlak p_iS "vliv netesnoti na tlak inertu v nadobe (model stlacitelneho proudeni)";
tokhmot Mtok_inL "hmot. tok inertu (laminarni model)";
tokhmot Mtok_inT "hmot. tok inertu (turbolentni model)";
tokhmot Mtok_inS "hmot. tok inertu (model stlacitelneho proudeni)";
teplota Tf "teplota chlazené kapaliny";
hmotnost Mf "hmotnost chlazené kapaliny";
hmotnost Mk "hmotnost kondenzovaných par";

```

Pro vyřešení diferenciálních rovnic budeme potřebovat ještě definovat počáteční podmínky. Příkazem *initial equation* dáme programu vědět, že následující rovnice jsou počátečními podmínkami. Jelikož proces začíná při atmosférickém tlaku a počáteční hmotnost chlazené kapaliny je známa, jednoduše tyto podmínky sestavíme a zapíšeme

```

//POCATECNI PODMINKY
initial equation
  p_iL = 0;
  p_iT = 0;
  p_iS = 0.000001;
  p_i = pa - pw;
  Mf = Mf0;
  Mk = 0;

```

Tímto jsme definovali všechny potřebné parametry a proměnné. Nyní potřebujeme vyjádřit některé proměnné jako funkce zadaných parametrů nebo jiných proměnných. Jedná se o průměr modelové netěsnoti D , parciální tlak nasycených par kapaliny p_w závislý na teplotě, dále hmotnostní toky inertů netěsnotmi \dot{M}_{in} a charakteristiku vývěvy \dot{V} .

```

//ROVNICE
equation
  D = sqrt((4 * A) / pi);
  //parciální tlak nasycených par při teplotě chladič vody
  //pw = 1000*e^(16.2886 - (3816.44/(Tk-46.13)));

  //hmotnosti toky ineru netesnotmi (laminarni, turbolentni a model
  stlacitelneho proudeni);
  Mtok_inL = pi * D ^ 4 / (128 * mi * L) * rhoi * (pa - pw - p_i);
  Mtok_inT = sqrt(2 * D / (lambda * L * rhoi) * (pa - pw - p_i)) * A * rhoi;
  Mtok_inS = A*pa*sqrt((2*Me)/(R*Tk))*sqrt((kappa/(kappa-1))
  *(1-((pw+p_i)/pa)^((kappa-1)/kappa))*((pw+p_i)/pa)^(2/kappa));

  //charakteristika vyvevy
  Vtok = 0.0377 * (1 - e ^ (-0.001184 * (p_i + pw - 2252)));

```

Nyní můžeme přistoupit k zapsání samotných diferenciálních rovnic. Výraz $\frac{df}{dt}$ se v jazyce zapíše jako $der(f)$. Jelikož program má svou vnitřní časovou proměnnou $time$, je tento zápis ekvivalentní časové derivaci. Postupně tedy zapíšeme všechny tři diferenciální rovnice (12), (13) a (15). Tyto diferenciální rovnice je nutné projit dalšími pomocnými funkcemi.

```
//prubeh tlaku inertu pri odsavani vzduchu z vakuove nadoby
der(p_i) = Vk ^ (-1) * (r * Tk * Mtok_inS - Vtok * p_i);
p = p_i + pw; //celkovy tlak ve vakuove nadobe

//Antoinova rovnice pro teplotu nasycenych par v zavislosti
na jejich parcialnim tlaku
Tf = ((-4567.833118) + log(p / 1000) * 46.13) / (log(p / 1000) - 16.2886);

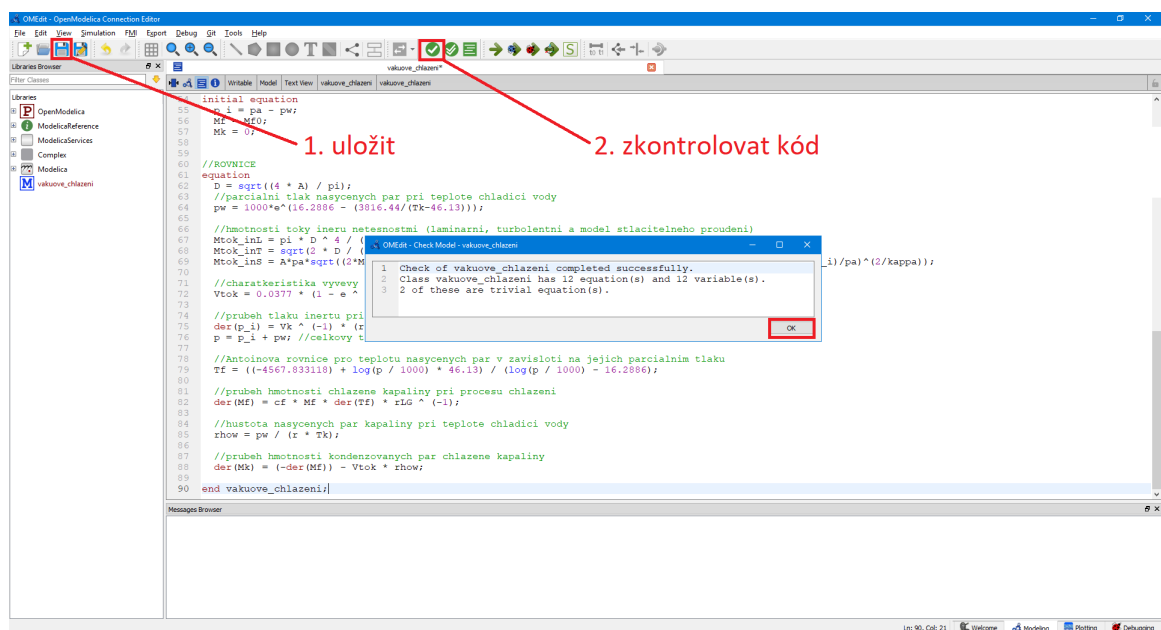
//prubeh hmotnosti chlazene kapaliny pri procesu chlazení
der(Mf) = cf * Mf * der(Tf) * rLG ^ (-1);

//hustota nasycenych par kapaliny pri teplote chladici vody
rhow = pw / (r * Tk);

//prubeh hmotnosti kondenzovanych par chlazene kapaliny
der(Mk) = (-der(Mf)) - Vtok * rhow;
end vakuove_chl_FUNKCNI;
```

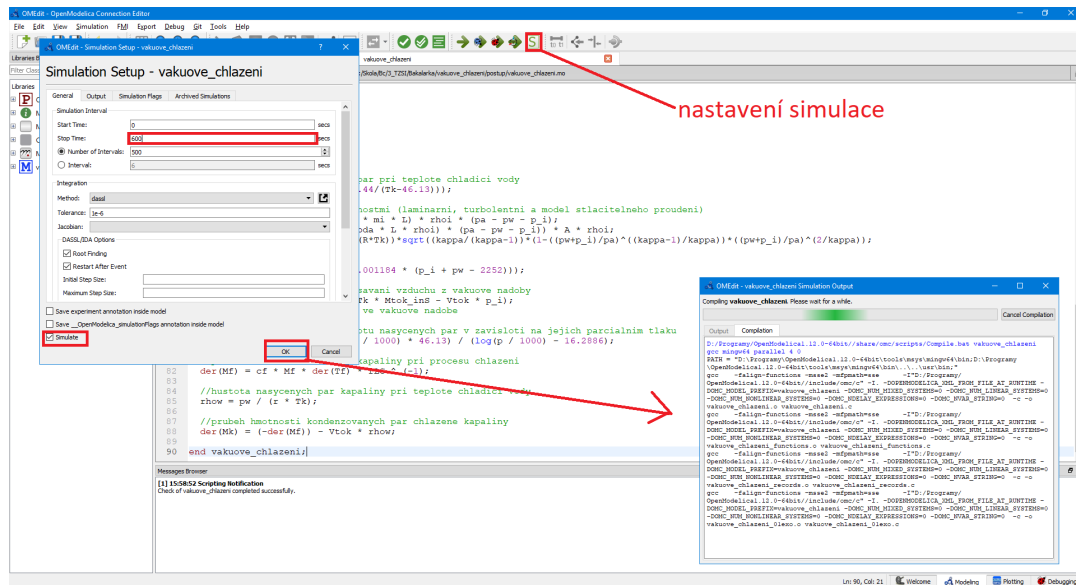
4.4.3 Kontrola a nastavení simulace

Tímto je model připravený, můžeme ho uložit a zkontrolovat kód. Tato kontrola nám nezaručuje správnost výpočtu, ale zjistíme, zda je kód správně po formální stránce. Tzn. jestli je správně zápis a hlavně jestli se počet neznámých rovná počtu rovnic (obrázek 46).



Obrázek 46: Uložení modelu a kontrola kódu

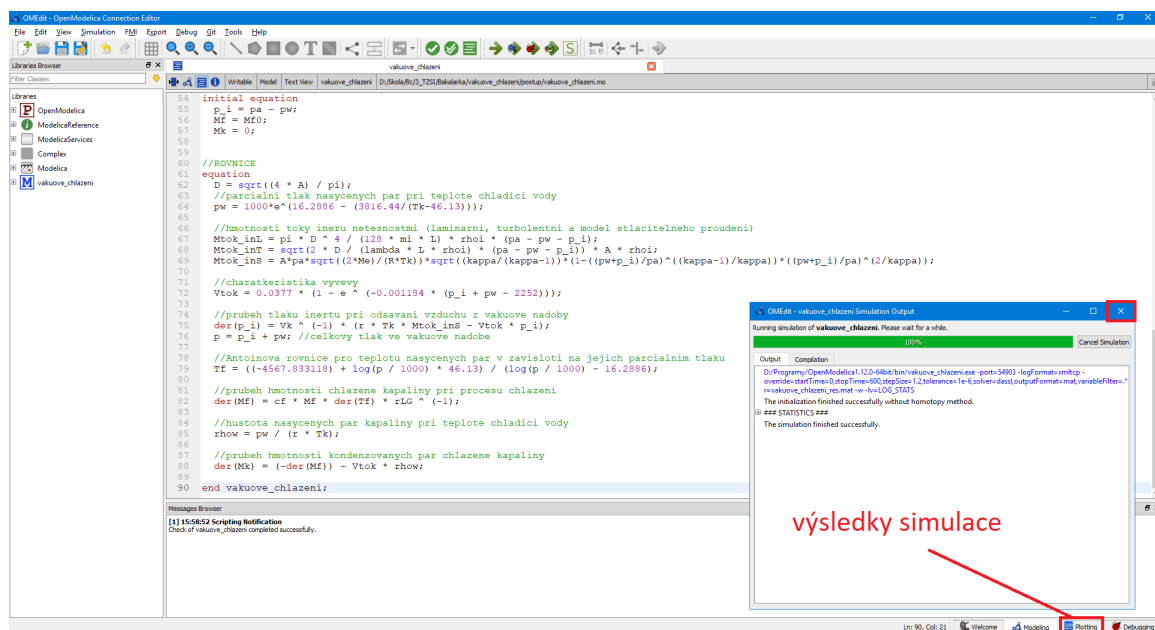
Po úspěšné kontrole můžeme přejít k vlastní simulaci. Nejprve je potřeba nastavit parametry simulace (obrázek 47). Nás bude zajímat zejména čas, který nastavíme na 600 s (10 minut). Poté můžeme simulaci spustit a objeví se dialogové okno o průběhu kompilace.



Obrázek 47: Nastavení a spuštění simulace

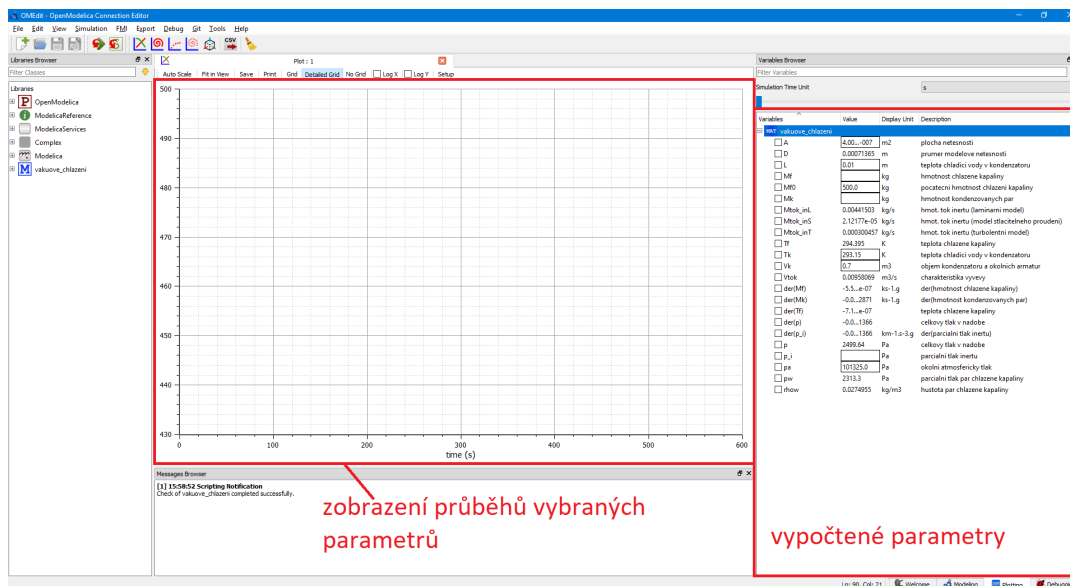
4.4.4 Zobrazení výsledků a jejich export

Pokud kompilace a výpočty proběhnou v pořádku, objeví se ve stejném dialogovém okně zpráva o úspěšném dokončení. Můžeme tedy přepnout obrazovku na zobrazení výsledků (obrázek 48).



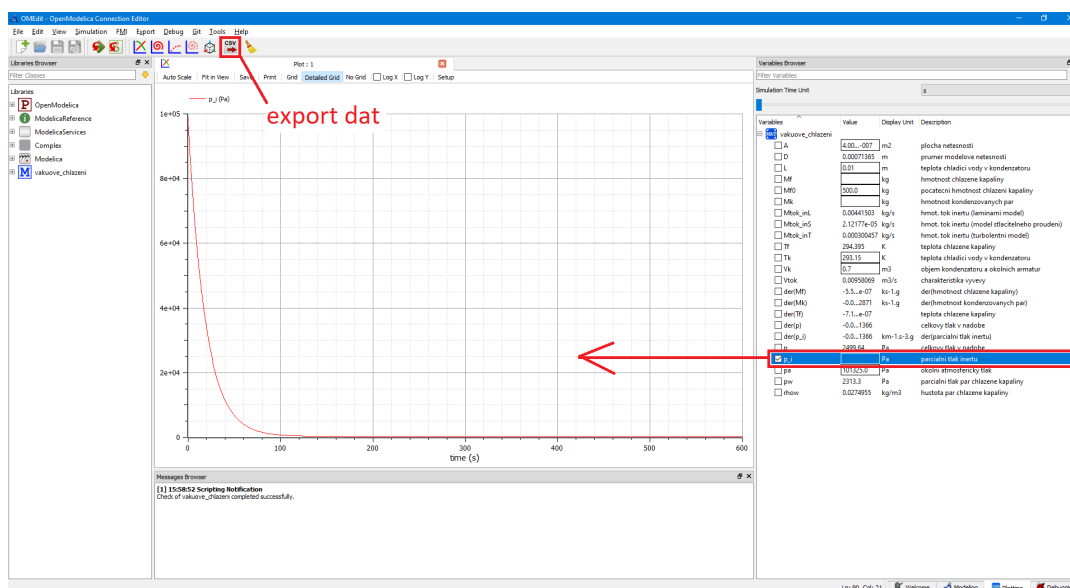
Obrázek 48: Zobrazení výsledků

Toto zobrazení se skládá z prostředního okna, kde se zobrazují průběhy vybraných proměnných a pravého panelu, kde je seznam všech počítaných parametrů (obrázek 49).



Obrázek 49: Výsledkové rozhraní

Pro zobrazení požadované proměnné ji stačí v panelu zaškrtnout (obrázek 50). Tímto způsobem lze zobrazit i několik proměnných najednou. Jelikož budeme měnit některé parametry simulace (modely netěsností) a ty vzájemně porovnávat, bude z hlediska komfortu výhodné exportovat vypočtená data, které následně vyhodnotíme v jiném programu (např. Matlab nebo Microsoft Excel). Nejprve zaškrtneme požadované proměnné pro export. Ten pak jednoduše provedeme kliknutím na ikonu z obrázku 50.



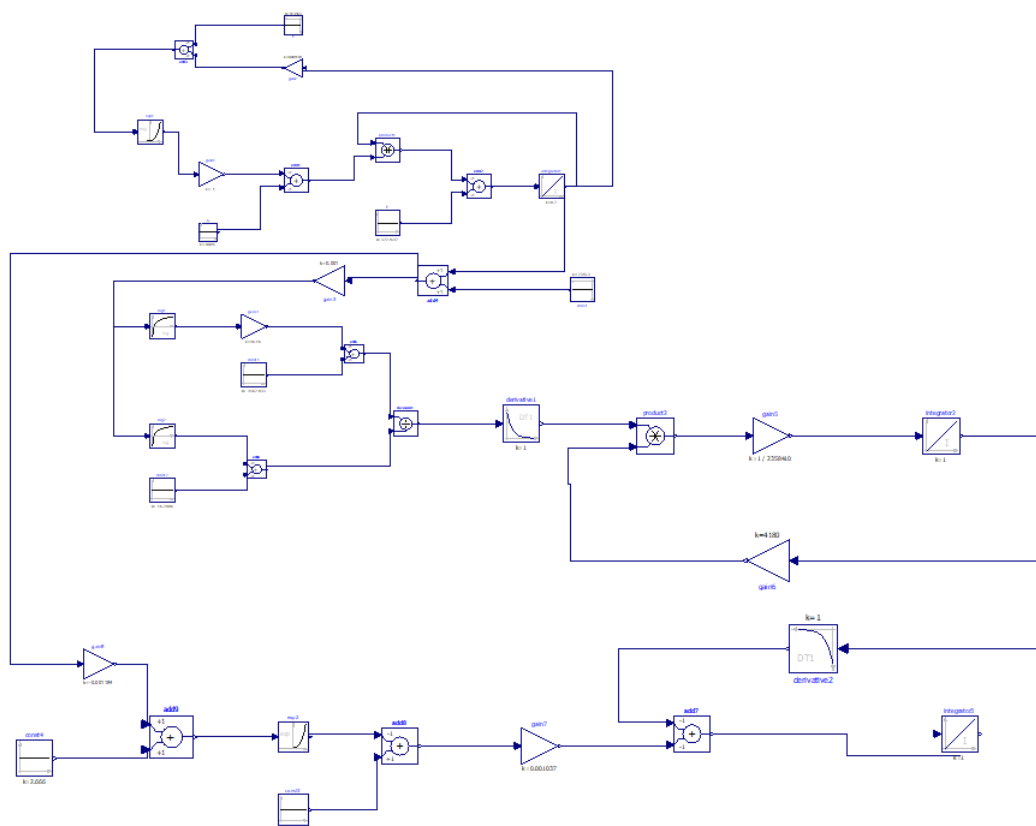
Obrázek 50: Grafické zobrazení výsledků

4.4.5 Blokový model

V programu OpenModelica je možné matematický model, místo v textovém editoru, vytvořit pomocí předdefinovaných bloků graficky (podobně jako v programu Simulink). Takto vytvořený model má výhodu v tom, že vidíme návaznost jednotlivých operací a proměnných.

Pro naše potřeby názornosti tvorby simulace je však tento způsob nevhodný, neboť některé rovnice jsou zjednodušené. Pro čtenáře pak může být toto schéma nepřehledné a matoucí.

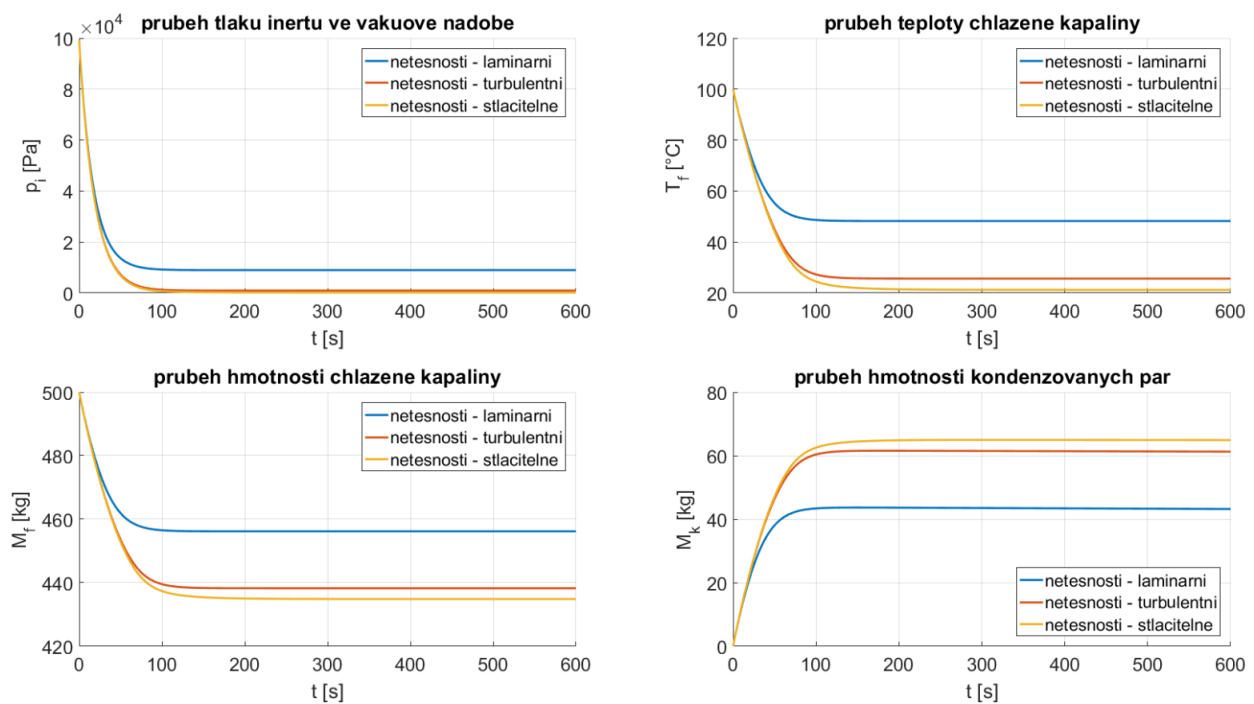
Pro zajímavost a demonstraci možností programu OpenModelica ale můžeme ukázat hotové schéma (obrázek 51) našeho modelu vakového chlazení.



Obrázek 51: Model vakového chlazení vytvořený graficky

4.4.6 Výsledky simulace

Z exportovaných dat můžeme v dalším software (v našem případě Matlab), vyhodnotit výsledky. Celkem byly provedeny tři simulace, pro každý matematický model netěsností jedna. Z diagramů na obrázku 52 je patrné, že modely netěsností měly na průběh vakového chlazení poměrně velký vliv.



Obrázek 52: Výsledky simulace - porovnání vlivu různých modelů netěsností (laminární a turbulentní proudění v trubce, stlačitelné proudění tryskou)

Zejména rozdíl mezi laminárním modelem a modelem stlačitelného proudění je markantní. Zatímco u modelu stlačitelného proudění bylo dosaženo tlaku inertů zhruba $p_i = 0,187$ kPa a teploty chlazené kapaliny $T_f = 21,3$ °C, u laminárního modelu minimální tlak inertů dosahuje $p_i = 8,9$ kPa a teplota chlazené kapaliny se sníží na $T_f = 48,2$ °C.

Tento rozdíl je dán zejména tím, že model stlačitelného proudění konvergentní tryskou zohledňuje aerodynamické zahlcení netěsností vlivem velkého rozdílu tlaků, jak je uvedeno v kapitole Matematické modely netěsností.

Na druhou stranu, chlazení při aplikovaném turbulentním modelu má velmi podobný průběh, jako chlazení s modelem konvergentní trysky. Tlak inertů dosahuje hodnoty $p_i = 0,955$ kPa a teplota chlazené kapaliny $T_f = 25,65$ °C.

Bohužel v tuto chvíli nemůžeme hodnotit jestli jsou výše uvedené výsledky podobné skutečnosti. Bylo by potřeba provést experimentální měření a výsledky simulace s tímto experimentem porovnat.

5 Diskuse

Tato bakalářská práce se zabývá nejprve řešší některých systémů pro tvorbu stacionárních i dynamických modelů nejen zařízení procesní techniky. Následně z nalezených a popsanych programů se zaměřuje na dva, jeden pro tvorbu stacionárního modelu (bilance) odparky a druhý pro tvorbu dynamického modelu vakuového chlazení. V následujících kapitolách si shrneme poznatky z jednotlivých částí.

5.1 Rešerše

Do této části bylo zařazeno celkem 8 různých programů a systémů. O všech se podařilo nalézt poměrně velké množství informací z mnoha zdrojů. Mezi nejužitečnější zdroje lze zařadit odborné práce Karris (2006), Ospino et al. (2017) a Ishii a Otto (2007), které obsahují množství zajímavých informací. Mnoho můžeme využít i v obecnějším kontextu.

Cílem rešerše bylo i zjistit, zda a za jakým účelem se tyto programy využívají v technické praxi. Ke každému software se podařilo nalézt alespoň jednu odbornou práci, ve které ho autoři využívají k numerickým simulacím od energetických až po potravinářská zařízení. Mezi nalezenými zdroji však není mnoho těch, které zabývají přímo prací s danými programy, jejich porovnáním nebo přímo vysvětlením základního ovládání. Z tohoto hlediska snad bude tato bakalářská práce přínosem.

5.2 Tvorba stacionárního modelu

Tato část nejprve popsala princip funkce modelovaného zařízení. Jelikož je odparka, ať už v jakékoliv modifikaci hojně využívána v různých aplikacích, nebyl s hledáním informací velký problém. Jako základ k přehlednému rozdělení velmi různorodých typů odparek dobře posloužila odborná práce Zvoníček (1961). I když je tato práce starší, stále jsou informace v ní poměrně aktuální.

Následně je zařízení popsáno matematickými vztahy, které jsou pak použity při vysvětlení funkce jednotlivých komponent v bilančním programu.

Pro bilanci byl zvolen program COCO simulator, který je volně dostupný a využívá standard CAPE-OPEN. Proto je i kompatibilní se software třetích stran (Microsoft Excel, Matlab, apod.). S tímto programem bylo možné se seznámit i při absolvování výuky na FS ČVUT a byl tedy vhodnou volbou při výběru.

Hlavním cílem této části bylo názorně ukázat uživatelské prostředí bilančního programu a předvést, jak se stacionární model vytváří. Práce s COCO simulator (konkrétně s editorem COFE) je vcelku intuitivní a i přes mírnou roz-

tahanost ovládacích prvků jsou všechny kroky tvorby logicky uspořádané (vytvoření dokumentu → definice termodynamického systému → vytvoření procesního vývojového diagramu → nastavení parametrů → vlastní výpočet → interpretace výsledků). Databáze sloučenin je zaměřena hlavně na uhlovodíky, ale lze dodefinovat i vlastní látky, včetně různých teplotních korelací apod.

Přímo při práci lze také vyvolat online nápovědu, ve které jsou popsány všechny základní funkce a komponenty. Zejména u cirkulačních okruhů je třeba dbát na vhodný odhad počátečních podmínek, případně opatrnou a pozvolnou změnu parametrů. Pokud se netrefíme dostatečně blízko řešení, může se stát že výpočet nedosáhne konvergence. Je tedy vhodné u okruhů zadat do každého proudu smyčky nějaké počáteční hodnoty, nebo v lepším případě smyčku rozpojit, odebrat všechny regulační obvody a první výpočet provést s náhradním vstupním proudem. V některých případech je nutné zadat počáteční parametry i do regulačních obvodů.

Výsledky bilance jsou okamžitě k dispozici přímo ve schématu, u každého proudu i komponenty. Do flowsheetu lze také přidávat různé tabulky s výsledky a nastavenými parametry. Samozřejmě lze vytvořit i report a ten vyexportovat do jiného software pro vyhodnocení.

5.3 Tvorba dynamického modelu

Pokud nás vedle množství výchozích látek a produktů nebo energetických bilancí zajímá i samotný průběh procesu, potřebujeme software, který některým způsobem dokáže řešit diferenciální rovnice. Pokud si numerické řešení připravíme ručně, lze samozřejmě využít jakýkoliv dostupný programovací jazyk. V dnešní době však existují i jazyky, nebo od nich odvozené editory, které dokáží předdefinovanou metodou vyřešit i symbolicky psané rovnice. Mezi tyto programy se řadí i software OpenModelica, využívající platformu Modelica. Model lze tvořit buď zápisem rovnic do stejnojmenného programovacího jazyka, nebo skládáním předdefinovaných komponent do PFD schémat, podobně jako v COCO simulator.

Část věnovaná dynamickým modelům se zabývá hlavně prvním případem, tedy rovnicovým modelem. Jako simulovaný proces bylo zvoleno vakuové chlazení. Tím se mnoho odborných prací nezabývá. Nejprve je v kapitole popsán princip funkce a matematické pozadí rovnovážného modelu, který je doplněn realizací v programu OpenModelica. Část kapitoly je věnována také matematickému odvození modelu netěsností v aparatuře vakuového chlazení, kde jsou porovnány tři různé přístupy.

Samotný program OpenModelica je opět volně dostupný a podle předpřipravených tutoriálů a nalezených zdrojů se jedná o velmi zajímavý nástroj se širokým

uplatněním. Avšak obsahuje několik funkčních chyb a nedokonalostí. První z nich je fakt, že nejaktuálnější verze, kterou se podařilo spustit je OpenModelica v 1.12.0 z roku 2017. V důsledku toho nemusí být všechny další popisované problémy již aktuální. Další komplikací je ukládání rozpracovaných modelů. OpenModelica v 1.12.0 vyžaduje cestu k uloženému modelu bez mezer a diakritiky, což může být zejména v českém jazyce problém. Bohužel model lze uložit bez jakéhokoliv upozornění uložit i do nevyhovující cesty a chyba se projeví až při opětovném pokusu o načtení modelu, který bude neúspěšný.

I když je pracovní postup uvedený v této práci zaměřen hlavně na rovnicovou tvorbu, v závěru kapitoly je ukázán blokový model.

Práce s textovým editorem je bezchybná, syntax je dobře barevně zvýrazněn a jednotlivé typy příkazů jsou tak dobře rozlišeny. Hlášení o chybách (a překlepech) je také přehledně zpracováno, takže lze problém rychle najít a opravit. Jazyk Modelica jako takový, je velmi intuitivně vyřešený a základní příkazy a syntax se dají rychle naučit. Zápis rovnic se prakticky neliší od symbolického zápisu.

U blokového modelování můžeme pár nedokonalostí přeci jen nalézt. Největší nevýhodou se jeví nespolehlivé propojování komponent. Jednotlivé proudy se špatně připojují a organizace celého modelu je poměrně problematická. Nastavení jednotlivých komponent jsou mírně nepřehledná a některé z nich obsahují chyby ve zdrojovém kódu. V uživatelské přívětivosti jednoznačně vítězí textový editor, tedy rovnicově tvořené modely.

Nastavení simulace (doba běhu procesu, počet iterací apod.) je přehledné a jednoduché. Výsledky jsou k dispozici ihned po dokončení simulace. Všechny proměnné a parametry vyskytující se v modelu jsou přehledně uspořádány do seznamu, ze kterého je možné jednotlivé parametry přímo zobrazit graficky. Můžeme zobrazovat jak časově závislosti, tak i vzájemné závislosti mezi různými parametry. V tomto případě lze však zobrazit pouze jednu závislost. Jelikož grafické zpracování dat v programu OpenModelica vykazuje limity, je výhodnější data exportovat ve formátu .csv a zpracovat je např. v Matlabu.

6 Závěr

Cílem této bakalářské práce bylo provést rešerši používaných bilančních a modelovacích systémů, výstižně je popsat a uvést příklady jejich využití v praxi. Následně několik z těchto programů vybrat, demonstrovat na nich průběh tvorby modelu, zhodnotit kvalitu funkcí programů a jejich uživatelskou přívětivost.

Všechny tyto cíle se podařilo splnit. Vznikl tak nejen dobrý přehled některých používaných systémů (komerčních i volně dostupných), ale i poměrně detailní popis vybraných zařízení, používaných v procesní a zpracovatelské technice, tedy odparek a vakuového chlazení. Uvedené pracovní postupy tvorby stacionárního i dynamického modelu v programech COCO Simulator a OpenModelica, pak čtenáře krok po kroku provedou celým sledem operací, vedoucích ke zdárnému dokončení simulace, včetně zobrazení, vyhodnocení a případnému exportu dat. Vše je doplněno snímkami obrazovky přímo z uživatelského rozhraní vybraných programů.

Seznam použitých symbolů

A	konstanta Antoineovy rovnice, plocha netěsností, $1, \text{m}^2$
B	konstanta Antoineovy rovnice, 1
C	konstanta Antoineovy rovnice, 1
c_f	měrná tepelná kapacita chlazené kapaliny, $\text{J kg}^{-1}\text{K}^{-1}$
c_p	měrná tepelná kapacita za konst. tlaku, $\text{J kg}^{-1}\text{K}^{-1}$
D	průměr modelové netěsnosti, m
Δp	rozdíl tlaků, Pa
f	dělicí poměr, 1
h_F	měrná entalpie směsi na nátok, J kg^{-1}
h_L	měrná entalpie kapaliné fáze, J kg^{-1}
h_V	měrná entalpie plynné fáze, J tkg^{-1}
η	účinnost, 1
I	konstanta, 1
J	konstanta, 1
K_i	konstanta, 1
κ	Poissonova konstanta, 1
L	délka modelové netěsnosti, m
λ	součinitel třecích ztrát, 1
M_e	molární hmotnost okolního vzduchu, kg kmol^{-1}
M_f	hmotnost chlazené kapaliny, kg
M_k	hmotnost kondenzovaných par, kg
M_i	hmotnost inertů, kg
M_w	molární hmotnost vody, vodních par, kg kmol^{-1}
\dot{m}	hmotnostní tok, kg s^{-1}
\dot{m}_i	hmotnostní tok v proudu i , kg s^{-1}
\dot{m}_F	hmotnostní tok směsi na nátok, kg s^{-1}
\dot{m}_L	hmotnostní tok kapalné fáze, kg s^{-1}
\dot{M}_{in}	hmotnostní tok inertů, kg s^{-1}
\dot{m}_V	hmotnostní tok plynné fáze, kg s^{-1}
μ	dynamická viskozita vzduchu, Pa s
N	konstanta, 1
p	tlak, Pa
p_1	tlak na vstupu, Pa
p_2	tlak na výstupu, Pa
p_a	atmosférický tlak, Pa
p_i	parciální tlak inertů, Pa
p_i^{sat}	tlak nasycených par daný Antoinovou rovnicí, Pa
p_w	parciální tlak vodních par, Pa
\dot{Q}	přiváděný tepelný tok, J s^{-1} , W

r_i	měrná plynová konstanta pro vzduch (inerty), $\text{J kg}^{-1}\text{K}^{-1}$
r_{LG}	výparné teplo vody, J kg^{-1}
R	univerzální plynová konstanta, $\text{J mol}^{-1}\text{K}^{-1}$
ρ_f	hustota chlazené kapaliny, kg m^{-3}
ρ_i	hustota inertů, kg m^{-3}
ρ_w	hustota nasycených vodních par, kg m^{-3}
S_n	plocha netěsností, m^2
t	čas, s, min, h, d
T	teplota, $^{\circ}\text{C}$, K
T_1	teplota na vstupu, $^{\circ}\text{C}$, K
T_2	teplota na výstupu, $^{\circ}\text{C}$, K
T_e	teplota okolí, $^{\circ}\text{C}$, K
T_f	teplota chlazené kapaliny, $^{\circ}\text{C}$, K
T_k	teplota kondenzovaných par, $^{\circ}\text{C}$, K
ω_0	hmotnostní koncentrace netěkavé složky na nátok, $1, \text{kg kg}^{-1}$
ω_1	hmotnostní koncentrace netěkavé složky v zahuštěné směsi, $1, \text{kg kg}^{-1}$
x	molový zlomek kapaliny, $1, \text{mol mol}^{-1}$
y	molový zlomek par, $1, \text{mol mol}^{-1}$
K_i	rovnovážné konstanty, 1
T_1	teplota na vstupu, $^{\circ}\text{C}$, K
T_2	teplota na výstupu, $^{\circ}\text{C}$, K
\bar{u}	střední rychlost proudění, m s^{-1}
V_k	objem kondenzátoru, m^3
\dot{V}_{in}	objemový tok inertů, $\text{m}^3 \text{s}^{-1}$
\dot{W}	technická práce, J s^{-1} , W
ω_0	počáteční hmotnostní zlomek netěkavé složky ve vstupním proudu, $1, \text{kg kg}^{-1}$
ω_1	hmotnostní zlomek netěkavé složky v zahuštěném roztoku, $1, \text{kg kg}^{-1}$
ω_v	hmotnostní zlomek par, $1, \text{kg kg}^{-1}$
x	molový zlomek kapaliny, $1, \text{mol mol}^{-1}$
y	molový zlomek par, $1, \text{mol mol}^{-1}$

Seznam použité literatury

- Ahmad Mansour, N. M. A review of flash evaporation phenomena and resulting shock waves. *Experimental Thermal and Fluid Science*, 2019, roč. 107, s. 146–168. ISSN 0894-1777.
- Ahmed, J., Shafiur, M., Rahman. *Handbook of Food Process Design*. John Wiley & Sons Inc, 2012. ISBN 978-1-4443-3011-3
- Aveva. Aveva SimSci PRO/II. [online], 2019. [28.7.2019]. Dostupné z: <https://sw.aveva.com/engineer-procure-construct/process-engineering-and-simulation/pro-ii-process-engineering>.
- Aveva. PRO/II Product Datasheet. [online], 2019. [28.7.2019]. Dostupné z: https://sw.aveva.com/hubfs/pdf/datasheet/Datasheet_SE-LIO_PROIIComprehensiveProcessSimulation_10-17.pdf?hsLang=en.
- van Baten, J., Szczepanski, R. A thermodynamic equilibrium reactor model as a CAPE-OPEN unit operation. *Computers & Chemical Engineering*, 2011, roč. 35, Č. 7, s. 1251–1256. ISSN 0098-1354.
- Boarin, S., Cammi, A., Ponciroli, R., Chiesa, D., Previtali, E., Sisti, M., Magrotti, G., Prata, M., Salvini, A. Object-Oriented Modeling and simulation of a TRIGA reactor plant with Dymola. *Energy Procedia*, 2016, roč. 101, s. 42–49. ISSN 1876-6102.
- Bogusch, R., Marquardt, W. A formal representation of process model equations. *Computers & Chemical Engineering*, 1997, roč. 21, s. 1105–1115. ISSN 0098-1354.
- Bortolin, G., Borg, S., Gutman, P. O. Modeling of the wet end part of a paper mill with Dymola. *Mathematics and Computers in Simulation*, 2004, roč. 65, s. 31–38. ISSN 0378-4754.
- Cantera.org. Cantera. [online], 2019. [28.7.2019] Dostupné z: <https://cantera.org/>.
- COCOSimulator.org. COCO. [online], 2019. [28.7.2018] Dostupné z: <https://www.cocosimulator.org/>.
- COLaN. CAPE-OPEN. [online], 2019. [28.7.2019] Dostupné z: <http://www.colan.org/general-information-on-co-lan/>.
- Dassault Systems. Dymola Referential. [online], 2018. [28.7.2019] Dostupné z: <https://www.3ds.com/fileadmin/PRODUCTS/CATIA/DYMOLA/PDF/Dymola-referential-2018.pdf>.

- Dassault Systems. Dymola. [online], 2019. [28.7.2019] Dostupné z: <https://www.3ds.com/products-services/catia/products/dymola/>.
- Dostal, M., Petera, K. Vacuum cooling of liquids: mathematical model. *Journal of Food Engineering*, 2004, roč. 61, Č. 4, s. 533–539. ISSN 0260-8774.
- Dostál, M., Petera, K. Matematické modely vakuového chlazení kapalin: přehled a diskuse. In *48. konference ChISA*. 2001, .
- Fritzson, P. A. *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3*. John Wiley & Sons Inc, 2014. ISBN 978-1-118-85912-4
- Goodwin, D. G. An Open-Source, Extensible Software Suite for CVD Process Simulation. 2002. [online] [28.7.2019] Dostupné z: https://pdfs.semanticscholar.org/91f6/df684848fe81cfec30304f80ea0df44545a5.pdf?_ga=2.124376394.1313479853.1564759327-1572918896.1564335986&_gac=1.213767526.1564759328.Cj0KCQjwvo_qBRDQARIsAE-bsH_lv5Q_7vJnvgYpNZhZ7k_OZRiAw5GIC3w9xtdRve3LdCk4arZzYBAaAqWsEALw_wcB.
- Harraz, A. A., Freeman, J., Wang, K., Dowell, N. M., Markides, C. N. Diffusion-absorption refrigeration cycle simulations in gPROMS using SAFT- γ Mie. *Energy Procedia*, 2019, roč. 158, s. 2360–2365. ISSN 1876-6102.
- Iggland, M., Mazzoti, M. Introduction to Chemical Engineering for Lecture 7: Flash Distillation. 2015. [online][24.7.2019]. Dostupné z: <https://ethz.ch/content/dam/ethz/>.
- Ishii, Y., Otto, F. Novel and fundamental strategies for equation-oriented process flowheeting Part I: A basic algorithm for inter-linked multicolumn separation processes. *Computers & Chemical Engineering*, 2007, roč. 32, s. 1842–1860. ISSN 0098-1354.
- Janík, Z., Žáková, K. Online design of Matlab/Simulink and SciLab/Xcos block schemes. *International Journal of Emerging Technologies in Learning (iJET)*, 2011, ISSN 1863-0383. [online][28.7.2019] Dostupné z: <http://online-journals.org/i-jet/article/view/1614>.
- Karris, S. T. *Introduction to Simulink with Engineering Applications*. Orchard Publications, 2006. ISBN 978-0974423975
- Mehl, V., Johannsen, K. Calculating chemical speciation, pH, saturation index and calcium carbonate precipitation potential (CCPP) based on alkalinity and acidity using OpenModelica. *Journal of Water Supply: Research and Technology - Aqua*, 2017, roč. 67, Č. 1, s. 1–11. ISSN 1606-9935.

- Nyongbela, G. N., Johannsen, K. Application of OpenModelica® in the mathematical modeling and simulation of general copper pipe corrosion. *Journal of Water Supply: Research and Technology-Aqua*, 2015, roč. 64, Č. 4, s. 462–469. ISSN 1606-9935.
- OpenModelica.org. OpenModelica. [online], 2019. [28.7.2019] Dostupné z: <https://openmodelica.org/index.php>.
- Ospino, J., Sánchez, M., Secchi, A. Implementation of a block-oriented model library for undergraduate process control courses in EMSO simulator. *Education for Chemical Engineers*, 2017, roč. 18, s. 45–57. ISSN 1749-7728.
- Phillips, D. *Python 3 Object Oriented Programming*. Packt Publishing Limited, 2010. ISBN 9781849511261
- Pipitone, G., Bolland, O. Power generation with CO₂ capture: Technology for CO₂ purification. *International Journal of Greenhouse Gas Control*, 2009, roč. 3, s. 528–534. ISSN 1750-5836.
- Process Systems Enterprise. gPROMS ProcessBuilder. [online], 2019. [26.7.2019]. Dostupné z: <https://www.psenderprise.com/products/gproms>.
- Scilab. Xcos. [online], 2019. [28.7.2019] Dostupné z: <https://www.scilab.org/software/xcos>.
- Smith, C., Corripio, R. *Principles and Practice of Automatic Process Control*. John Wiley & Sons, 2006. ISBN 978-0-471-43190-9
- Tao, L., Dongmei, Z. *Numeric simulation and analysis of H₂-O₂ premixed combustion based on OpenFOAM*. IEEE, 2012. ISBN 978-1-4673-2207-2
- Teklay, A., Bøjer, M., Adelsward, A., Yin, C., Rosendahl, L. Simlation of flash dehydroxaltion of clay particle using gPROMS: A move towards green concrete. *Energy Procedia*, 2014, roč. 61, s. 556–559. ISSN 1876-6102.
- The Modelica Association. Modelica. [online], 2019. [28.7.2019] Dostupné z: <https://www.modelica.org/>.
- Vanfretti, L., Li, W., Bogodorova, T., Panciatici, P. *Unambiguous power system dynamic modeling and simulation using modelica tools*. IEEE, 2013. ISBN 978-1-4799-1303-9
- Yigit, T., Selamet, O. F. Mathematical modeling and dynamic Simulink simulation of high-pressure PEM electrolyzer system. *International Journal of Hydrogen Energy*, 2016, roč. 41, s. 13901–13914. ISSN 0360-3199.

Zvoníček, J. *Základní operace chemického a potravinářského průmyslu - odparky.*
Vydavatelství ČVUT, 3. přepr. vyd. Praha ed., 1961

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Daniel Kříž, v. r.,

2. srpna 2019

Kuba, J.: Metodický pokyn č. 1/2009 O dodržování etických principů při přípravě vysokoškolských závěrečných prací, ČVUT v Praze, 1. července 2009, <https://www.cvut.cz/sites/default/files/content/d1dc93cd-5894-4521-b799-c7e715d3c59e/cs/20190327-metodicky-pokyn-c-12009-o-dodrzovani-etickyh-principu-pri-priprave-vysokoskolskych.pdf> [31. července 2019]