

# Beregningsprogram for varmerør

**Bjarte Ross Idsø**

Master i produktutvikling og produksjon  
Oppgaven levert: Juni 2009  
Hovedveileder: Erling Næss, EPT



# Oppgavetekst

Målet for oppgaven er å etablere et beregningsprogram for den termisk-hydrauliske oppførselen til varmerør. Beregningsprogrammet skal spesielt kunne håndtere anvendelser med alkalimetaller som arbeidsmedium.

Oppgaven gitt: 21. januar 2009  
Hovedveileder: Erling Næss, EPT





## HOVEDOPPGAVE

for

Stud. techn. Bjarte Ross Idsø

Våren 2009

### **Beregningsprogram for varmerør**

*Calculation program for heat pipes*

#### **Bakgrunn**

Bruk av varmerør (engelsk: "heat pipes") har vist seg å være en meget effektiv metode for varmetransport, og finner stadig nye anvendelsesområder. Institutt for Energi- og Prosesseteknikk er involvert i forskning og utvikling av varmerør for høytemperatur prosesskjøling og varmegjenvinning. I den forbindelse er det ønskelig å få laget et beregningsprogram for varmerør. Beregningsprogrammet skal gi en helhetlig beskrivelse av varme- og massetransportmekanismene i varmerør, samt de ytelsesbegrensningene som gjelder for dette konseptet. Bruk av veker for sikring av effektiv distribusjon av arbeidsmediet i fordampersesksjonen er av spesiell interesse, og skal undersøkes spesielt. Programmet vil i en slik form bli et viktig verktøy i den forsknings- og utviklingsaktiviteten som foregår ved instituttet.

#### **Mål**

Målet for oppgaven er å etablere et beregningsprogram for den termisk-hydrauliske oppførselen til varmerør. Beregningsprogrammet skal spesielt kunne håndtere anvendelser med alkalimetaller som arbeidsmedium.

#### **Oppgaven bearbeides ut fra følgende punkter:**

1. Konseptet for varmerør ("heat pipe") skal gjennomgås og presenteres. Hovedkomponentene skal beskrives, og prosessene som påvirker varmetransporten skal dokumenteres. Videre skal begrensninger i varmerørs varmetransportkapasitet vurderes og presenteres.
2. Det skal gjennomføres et litteraturstudium innen termisk-hydrauliske forhold knyttet til bruk av veker i varmerør. Det skal spesielt fokuseres på anvendelser med alkalimetaller som arbeidsmedia. Virkemåte skal presenteres, og metoder for beregning av vekers ytelse skal sammenliknes og diskuteres. Videre skal anbefalte vekegeometrier vurderes og presenteres i den grad slik informasjon er tilgjengelig.
3. Med basis i oppgavens pkt. 1 og 2 skal det utarbeides et beregningsprogram for termisk ytelse av varmerør basert på alkalimetall som arbeidsmedium. Programmet skal beregne ytelse og ytelsesbegrensninger ved ulike stasjonære driftssituasjoner. Behovet for å kunne beregne transiente forhold skal vurderes, og eventuelt implementeres dersom tiden tillater dette.

4. Det skal gjennomføres beregninger med programmet for varmerør utsatt for utvalgte driftssituasjoner, som velges i samarbeid med Instituttet. Beregningene skal presenteres, og resultatene skal presenteres og diskuteres.
5. Det skal utarbeides forslag til videre arbeid.

- ” -

Senest 14 dager etter utlevering av oppgaven skal kandidaten levere/sendte instituttet en detaljert fremdrift- og evt. forsøksplan for oppgaven til evaluering og evt. diskusjon med faglig ansvarlig/veiledere. Detaljer ved evt. utførelse av dataprogrammer skal avtales nærmere i samråd med faglig ansvarlig.

Besvarelsen redigeres mest mulig som en forskningsrapport med et sammendrag både på norsk og engelsk, konklusjon, litteraturliste, innholdsfortegnelse etc. Ved utarbeidelsen av teksten skal kandidaten legge vekt på å gjøre teksten oversiktlig og velskrevet. Med henblikk på lesning av besvarelsen er det viktig at de nødvendige henvisninger for korresponderende steder i tekst, tabeller og figurer anføres på begge steder. Ved bedømmelsen legges det stor vekt på at resultatene er grundig bearbeidet, at de oppstilles tabellarisk og/eller grafisk på en oversiktlig måte, og at de er diskutert utførlig.

Alle benyttede kilder, også muntlige opplysninger, skal oppgis på fullstendig måte. (For tidsskrifter og bøker oppgis forfatter, tittel, årgang, sidetall og evt. figurnummer.)

Det forutsettes at kandidaten tar initiativ til og holder nødvendig kontakt med faglærer og veileder(e). Kandidaten skal rette seg etter de reglementer og retningslinjer som gjelder ved alle (andre) fagmiljøer som kandidaten har kontakt med gjennom sin utførelse av oppgaven, samt etter eventuelle pålegg fra Institutt for energi- og prosesssteknikk.

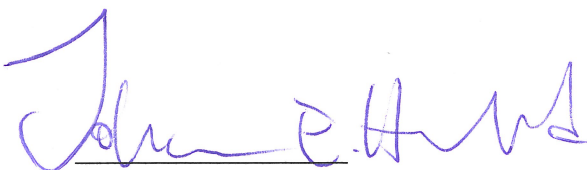
I henhold til ”Utfyllende regler til studieforskriften for teknologistudiet/sivilingeniørstudiet” ved NTNU § 20, forbeholder instituttet seg retten til å benytte alle resultater i undervisnings- og forskningsformål, samt til publikasjoner.

Ett -1 komplett eksemplar av originalbesvarelsen av oppgaven skal innleveres til samme adressat som den ble utlevert fra. (Det skal medfølge et konsentrert sammendrag på maks. en maskinskrevet side med dobbel linjeavstand med forfatternavn og oppgavetittel for evt. referering i tidsskrifter).

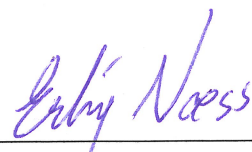
Til Instituttet innleveres to - 2 komplette, kopier av besvarelsen. Ytterligere kopier til evt. medveiledere/oppgavegivere skal avtales med, og evt. leveres direkte til, de respektive.

Til instituttet innleveres også en komplett kopi (inkl. konsentrerte sammendrag) på CD-ROM i Word-format eller tilsvarende.

Institutt for energi og prosesssteknikk, 12. januar 2009



Johan E. Hustad  
Instituttleder



Erling Næss  
Faglig ansvarlig/veileder

Kontaktperson/medveileder:  
Stipendiat Geir Hansen, NTNU

## Forord

Denne rapporten er skrevet i forbindelse med min hovedoppgave ved Institutt for energi- og prosesseteknikk, NTNU, i perioden 21.januar til 17.juni 2009.

Denne rapporten er tenkt lest uavhengig uten noen form for tilleggslitteratur. Det er likevel å anbefale at leseren har noe kjennskap til varme og massetransport samt enkel programmering for å lettere skjønne teori og programlisting.

Jeg vil gjerne takke mine veiledere Professor Erling Næss og stipendiat Geir Hansen ved NTNU. Begge har vært til stor hjelp.

Trondheim, 14.juni 2009



Bjarte Ross Idsø  
Stud.techn





## Sammendrag

Denne rapporten beskriver utviklingen av et beregningsprogram for den termisk-hydrauliske oppførselen til et varmerør. Programmet har spesielt blitt utviklet for å kunne håndtere alkalimetaller som arbeidsmedium.

Konseptet for varmerør er blitt gjennomgått. Komponenter sin virkemåte, prosesser og begrensninger som påvirker varmetransporten er blitt belyst. Av begrensninger er kapillærbegrensning, viskøs begrensning, sonisk begrensning og kokebegrensning identifisert som de mest aktuelle. Tre ulike modeller for damptransport i varmerøret er blitt presentert. Videre er et litteraturstudium med fokus på termisk-hydrauliske forhold i veker blitt gjennomført. Behovet for å kunne beregne transiente forhold er blitt vurdert.

Aktuelle forhold med hensyn på alkalimetaller er blitt tatt i betraktning. To typer veker, veker av metallnetting og veker av sintret metallpulver er blitt vurdert og presentert.

Med basis i dette er det blitt utviklet et beregningsprogram i *Java* for beregning av ytelse og begrensninger for et varmerør ved ulike stasjonære driftssituasjoner. Data-listing for beregningsdelen av programmet er lagt ved og algoritmer er blitt presentert og gjennomgått. Programmet benytter seg av en dampmodell som ikke tar hensyn til kompressibilitet. Implementering av dampmodell for kompressibel strømning anbefales i eventuelt videre arbeid.

Ved hjelp av programmet er det blitt utført to beregninger. Den første beregningen forsøker ved hjelp av programmet å reprodusere resultater for tre ulike stasjonære driftssituasjoner presentert av Faghri [7]. I den andre beregningen er den minste mulige radius i varmerør blitt beregnet ved ulike temperaturer og varmekullukser. Resultatene fra programmet stemmer godt overens med teori og beregninger presentert av Faghri.

Hoveddelen av programlistingen og egenskaper for arbeidsmedium brukt i beregninger er lagt ved. Til slutt i rapporten er et forslag til videre arbeid.

## Summary

This report describes the development of a computation program for the calculation of thermic-hydraulic behavior of a heat pipe. The program has been developed especially with alkali metals in mind.

The concept for heat pipes has been examined. The mode of operation for components, processes and limitations has been studied. The capillary limit, the viscos limit, the sonic limit and the boiling limit have been identified as the most relevant limitations. Three models for vapor flow have been presented. A literature study with focus on thermic-hydraulic conditions in wicks has been done. The need for transient calculations has been evaluated.

Relevant conditions concerning alkali metals have been gone through. Two wick types, metal mesh, and sintered metal powder wicks have been evaluated and presented.

Based on this foundation a computation program for calculation of performance and limitations in a heat pipe at different steady state conditions has been written in *Java*. Data listing for the calculation part of the program has been appended to the report and algorithms have been presented and gone through. The program uses a model for vapor flow that does not take compressibility in account. Implementation of a model that does is recommended as future work.

With the aid of the program two computations have been carried out. The first computation tries to reproduce results from three different steady state conditions presented by Faghri [7]. In the second calculation the minimum radius in a heat pipe has been calculated at different temperatures and heat fluxes. The results are evaluated and presented in the report. The results from the program agrees well with teori and calculations presented by Faghri.

# Innhold

<b>1</b>	<b>Innledning</b>	<b>1</b>
1.1	Bakgrunn . . . . .	1
1.2	Mål . . . . .	1
1.3	Oppbygging av rapport . . . . .	1
<b>2</b>	<b>Virkemåte for varmerør</b>	<b>2</b>
2.1	Konsept for varmerør . . . . .	2
2.1.1	Dampstrømning fra fordamper til kondensator . . . . .	3
2.1.2	Kondensasjon i kondensator . . . . .	4
2.1.3	Væskestrømning fra kondensator til fordamper . . . . .	4
2.1.4	Fordampning i fordamper . . . . .	4
2.2	Kapillært pumpetrykk $\Delta P_{kapp}$ . . . . .	4
2.3	Trykktap for væskestrøm $\Delta P_l$ . . . . .	6
2.4	Trykktap for dampstrøm $\Delta P_v$ . . . . .	6
2.4.1	Første orden lukket form tilnærming . . . . .	8
2.4.2	Andre orden tilnærming . . . . .	9
2.4.3	Kompressibel endimensjonal analyse . . . . .	10
2.4.4	Sammenligning av modeller for dampstrøm . . . . .	12
2.5	Varmegjennomgang radielt gjennom beholdervegg . . . . .	12
2.5.1	Fordampning i fordamperdel . . . . .	12
2.5.2	Kondensering i kondesator del . . . . .	13
2.5.3	Termisk motstand i damp-væske grensesnitt . . . . .	13
2.5.4	Forandring i mettet damptrykk over en krum væskefilm . . . . .	13
2.6	Diskusjon . . . . .	13
<b>3</b>	<b>Varmetransportbegrensninger for varmerør</b>	<b>15</b>
3.1	Introduksjon . . . . .	15
3.2	Kokebegrensning . . . . .	16
3.3	Kapillærbegrensning . . . . .	18
3.3.1	Lokalisering av våtpunkt . . . . .	18
3.4	Kondensatorbegrensning . . . . .	19
3.5	Medrivningsbegrensning . . . . .	19
3.6	Sonisk begrensning . . . . .	20

3.7	Viskøs begrensning . . . . .	20
3.8	Frossen start begrensning . . . . .	21
3.9	Sammenhengende strømnings begrensning . . . . .	21
3.10	Behov for å beregne transiente forhold . . . . .	21
3.11	Diskusjon . . . . .	21
<b>4</b>	<b>Virkemåte for veker i varmerør</b>	<b>22</b>
4.1	Ulike vekestrukturer og design . . . . .	22
4.1.1	Homogene veker . . . . .	22
4.1.2	Kompositivveker . . . . .	23
4.2	Anbefalte vekegeometrier med hensyn på alkalimetaller . . . . .	23
4.3	Permeabilitet $K$ for veker . . . . .	24
4.3.1	Permeabilitet for veker av metallnetting . . . . .	24
4.3.2	Permeabilitet for sintrede veker med sfæriske partikler . . . . .	25
4.4	Effektiv poreradius $r_{eff}$ for veker . . . . .	25
4.4.1	Effektiv poreradius for veker av metallnetting . . . . .	25
4.4.2	Effektiv poreradius for sintrede veker av sfæriske partikler . . . . .	26
4.5	Effektiv termisk konduktivitet $k_{eff}$ . . . . .	26
4.5.1	Veker av metallnetting . . . . .	26
4.5.2	Sintrede veker av metallpulver . . . . .	27
<b>5</b>	<b>Beregningsprogram for varmerør</b>	<b>28</b>
5.1	Introduksjon . . . . .	28
5.2	Struktur for programkode . . . . .	28
5.3	Kjøre program . . . . .	30
5.4	Foreta beregninger . . . . .	30
5.5	Egenskaper for arbeidsmedium . . . . .	30
5.6	Beregning av ytelse for varmerør . . . . .	30
5.7	Generere dampkurve ved første orden lukket form tilnærming . . . . .	32
5.8	Generere væskekurve . . . . .	33
5.9	Beregning av begrensninger . . . . .	34
5.9.1	Kapilærbegrensning og viskøs begrensning . . . . .	34
5.9.2	Sonisk begrensning . . . . .	34
5.9.3	Kokebegrensning . . . . .	34
5.10	Beregning av radielt Reynoldstall . . . . .	34
5.11	Begrensninger og feilkilder i modell . . . . .	34
<b>6</b>	<b>Utførte beregninger</b>	<b>36</b>
6.1	Introduksjon . . . . .	36
6.2	Beregninger med numerisk dampmodell av første orden . . . . .	36
6.3	Beregninger av minste diameter for varmerør . . . . .	40
6.3.1	Framgangsmåte . . . . .	40
6.3.2	Resultat . . . . .	41
6.3.3	Diskusjon av resultat . . . . .	41

<i>INNHold</i>	ix
<b>7 Konklusjon</b>	<b>44</b>
<b>8 Forslag til videre arbeid</b>	<b>46</b>
<b>A Programlisting</b>	<b>49</b>
<b>B Egenskaper for arbeidsmedium</b>	<b>69</b>

# Figurer

2.1	Skisse av varmerør med hovedkomponenter og prinsipp for virkemåte. . .	2
2.2	T-s diagram for et varmerør. 1-2: Fordamper, 2-3: Dampstrøm fra fordamper til kondensator, 3-4: Kondensering, 4-1: Væskestrøm fra kondensator til fordamper. (Basert på figur 1.6 fra Faghri [7]) . . . . .	3
2.3	Damp- og væsketrykk aksielt gjennom varmerør (våtpunkt lokalisert ved begynnelse av kondensator). Basert på figurer av Faghri [7] . . . . .	5
2.4	Forenklet oversiktsskisse over fysiske fenomener i et varmerør . . . . .	14
3.1	Maksimal varmeovergang $Q_{maks}$ som funksjon av operasjonstemperatur. .	15
3.2	Ulike typer varmetransport og dampgenerering i en veke (Basert på figur fra Faghri [7]). . . . .	17
4.1	Framstilling av nettingveke med geometriske mål. . . . .	24
4.2	Framstilling av sintret veke med geometriske mål og kubisk partikkelpakking (til høyre). . . . .	25
5.1	Grafisk framstilling av programstruktur. . . . .	28
5.2	Flytskjema for ytelseberegning. . . . .	30
5.3	Snitt gjennom fordamperdel av et varmerør . . . . .	31
5.4	Flytskjema for dampkurvealgoritme . . . . .	32
5.5	Flytskjema for veskekurvealgoritme . . . . .	33
6.1	Sammenligning av dampstrømningsmodeller fra Faghri [7] for $Re_r = 0,6$ . .	37
6.2	Resultat av dampstrømningsberegning med Cotters endimensjonale modell fra beregningsprogram for $Re_r = 0,6$ . . . . .	37
6.3	Sammenligning av dampstrømningsmodeller fra Faghri [7] for $Re_r = 2,4$ . .	38
6.4	Resultat av dampstrømningsberegning med Cotters endimensjonale modell fra beregningsprogram for $Re_r = 2,4$ . . . . .	38
6.5	Sammenligning av dampstrømningsmodeller fra Faghri [7] for $Re_r = 1000$	39
6.6	Resultat av dampstrømningsberegning med Cotters endimensjonale modell fra beregningsprogram for $Re_r = 1000$ . . . . .	39
6.7	Damptrykk gjennom varmerør ved dampstrømsradius $R_v = 0,0065 m$ . . .	41
6.8	Begrensningskurve ved dampstrømningsradius $R_v = 0,0011 m$ . . . . .	42
6.9	Begrensningskurve ved dampstrømningsradius $R_v = 0,0044 m$ . . . . .	43

B.1	Egenskaper for kalium hentet fra <i>Faghri</i> [7]. . . . .	70
B.2	Egenskaper for vann hentet fra <i>Faghri</i> [7]. . . . .	71

# Tabeller

5.1	Tabell over klasser med beskrivelse . . . . .	29
6.1	Beregninger av minimum dampradius $R_v$ . . . . .	41



# Symbolliste

$\alpha$	Koeffisient i termisk grensesnitt mellom væske og damp (normalt satt til 1), side 13
$\bar{w}$	Midlere aksiell damphastighet, side 7
$\Delta P_g$	Differanse i væsketrykk i veke på grunnlag av gravitasjon, side 3
$\Delta P_{kapp}$	Differanse i kapillært pumpetrykk i varmerør, side 3
$\Delta P_l$	Differanse i væsketrykk i veke, side 3
$\Delta P_v$	Differanse i damptrykk i varmerør, side 3
$\delta_1$	Tykkelse av et lag (for veker av metallnetting), side 27
$\delta_n$	Tykkelse av samtlige lag (for veker av metallnetting), side 27
$\dot{m}_l$	Massestrøm væske basert på gjennomsnittlig hastighet over vekens tverrsnitt-areal, side 6
$\dot{m}_v$	Massestrøm damp, side 8
$\mu_l$	Dynamisk viskositet for væske, side 6
$\mu_v$	Dynamisk viskositet for damp, side 8
$\nu$	Kinematisk viskositet, side 7
$\nu_l$	Kinematsik viskositet for væske, side 25
$\phi$	Helling på varmerør ( $90^\circ$ betyr fordamper i toppen), side 18
$\rho$	Tetthet, side 7
$\rho_0$	Referansetetthet, side 20
$\rho_l$	Tetthet for væske, side 6
$\rho_m$	Midlere tetthet for damp, side 11

$\rho_R$	Tetthet for damp i umiddelbar nærhet av veke, side 11
$\rho_v$	Tetthet for damp, side 8
$\sigma$	Overflatespenningen mellom fluid, side 5
$\tau$	Skjærkraft, side 6
$\theta$	Fluidets kontaktvinkel på flate, side 5
$\varphi$	Porøsitet, side 24
$A$	Busses korreksjonsfaktor, side 10
$A_{lv}$	Areal av individuelle overflateporer, side 19
$A_{veke}$	Tverrsnittareal for veke, side 6
$c$	Lydhastighet, side 7
$c_0$	Lydhastighet, side 20
$D$	Diameter for sintrede partikler i veke, side 25
$d$	Diameter for tråder i nettingveke, side 24
$D_h$	Hydraulisk diameter, side 24
$D_v$	Diameter av dampstrømningsområde, side 20
$h_\delta$	Varmetransportkoeffisient i damp-væske grensesnitt, side 13
$h_{fg}$	Fordampningsentalpi, side 8
$h_m$	Midlere entalpi for damp, side 11
$h_R$	Entalpi for damp i umiddelbar nærhet av veke, side 11
$K$	Permabilitet for veke, side 6
$k$	Varmetransportkoeffisient, side 12
$k'$	Forholdet $c_p/c_v$ , side 20
$k_s$	Varmetransportkoeffisient for vekemateriale, side 26
$k_l$	Termisk konduktivitet for væske, side 18
$L$	Tykkelse av beholdervegg, side 12
$L_a$	Lengde av adiabatisk del, side 8
$L_{eff}$	Effektiv lengde av varmerør, side 20

$L_e$	Lengde av fordamperdel, side 8
$L_t$	Total lengde av varmerør, side 8
$Ma$	Machtall for damp, side 7
$N$	Antall åpninger per enhet i nettingveke, side 24
$n$	Antall nettinglag (for veker av metallnetting), side 27
$P$	Fuktet omkrets, side 19
$p$	Trykk, side 7
$P_d$	Adskillelseskraft (eng: “disjoining pressure”), side 6
$p_{sat}$	Metningstrykk, side 13
$p_{v,\delta}$	Damptrykk over væskefilm, side 13
$q_\delta$	Varmefluks i termisk grensesnitt mellom væske og damp, side 13
$Q_e$	Varme tilført i fordamper, side 8
$Q_{kritisk}$	Kritisk varmekraft, side 18
$Q_s$	Varme rate ved sonisk grense, side 20
$r$	Koordinat radielt på varmerør, side 7
$R_b$	Radius for dampboble i væske, side 16
$r_{eff}$	Effektiv poreradius, side 5
$R_g$	Universell gasskonstant, side 11
$R_{h,w}$	Hydraulisk radius for porer i verkeoverflate, side 19
$R_i$	Indre radius av varmerørbeholder, side 18
$R_{men}$	Radius for væske damp flate, side 16
$R_v$	Radius av dampstrømningsområde i varmerør, side 8
$R_y$	Ytre radius på varmerørbeholder., side 31
$Re_{l,h}$	Aksielt Reynoldstall i væskefase, side 24
$Re_r$	Radielt Reynoldstall, side 8
$S$	Krimpefaktor (for veker av metallnetting), side 27
$T_0$	Referansetemperatur, side 20

$T_\delta$	Temperatur like over væskefilm, side 13
$T_e$	Gjennomsnittlig metningstemperatur i fordamper, side 4
$T_{kritisk}$	Kritisk overoppheting, side 16
$T_k$	Gjennomsnittlig metningstemperatur i kondensator, side 4
$T_l$	Temperatur i væskefase, side 13
$T_m$	Midlere temperatur for damp, side 11
$T_v$	Temperatur i dampfase, side 13
$u_R$	Radiell hastighetskomponent., side 11
$v$	Radiell hastighetskomponent, side 7
$v_\delta$	Hastighetskomponent for strømming mellom damp og veke, side 8
$v_{fg}$	Differanse for spesifikt volum mellom damp og væskefase, side 13
$v_l$	Spesifikt volum for væske, side 18
$v_m$	Midlere aksiell hastighet for damp, side 11
$v_v$	Spesifikt volum for damp, side 18
$W$	Åpning mellom tråder i nettingveke, side 24
$w$	Aksiell hastighetskomponent, side 7
$w_l$	Gjennomsnittlig aksiell hastighet for væske (volumbasert), side 25
$We$	Webernummer, side 19
$z$	Koordinat aksielt gjennom varmerør, side 7
$q_{vis}$	Varmefluks ved viskøs begrensning, side 20

# Kapittel 1

## Innledning

### 1.1 Bakgrunn

Bruk av varmerør (engelsk: “heat pipes”) er en meget effektiv metode for transport av varme ved høge flukser og med ekstremt små temperaturdifferanser. Konstruksjonen er enkel og robust med ingen bevegelige deler eller med ekstern pumpekraft. Størrelsen kan variere fra ørsmå varmerør til bruk i kjøling av elektroniske komponenter eller til lengder på over hundre meter. Den geometriske utformingen er meget fleksibel og kan tilpasses en rekke behov.

Varmerør har blitt brukt i fra alt til kjøling av satellitter til varming av vann i solfangere.

Institutt for Energi- og Prosessteknikk er involvert i forskning og utvikling av varmerør for høytemperatur prosesskjøling og varmegjenvinning. I den forbindelse har det vært ønskelig å lage et beregningsprogram for varmerør.

### 1.2 Mål

Målet for denne oppgaven er å etablere et beregningsprogram for den termisk-hydrauliske oppførselen til varmerør. Beregningsprogrammet skal gi en helhetlig beskrivelse av varme- og massetransportmekanismene i varmerør, samt de ytelsesbegrensninger som gjelder. Programmet skal kunne håndtere anvendelser med alkalimetaller som arbeidsmedium.

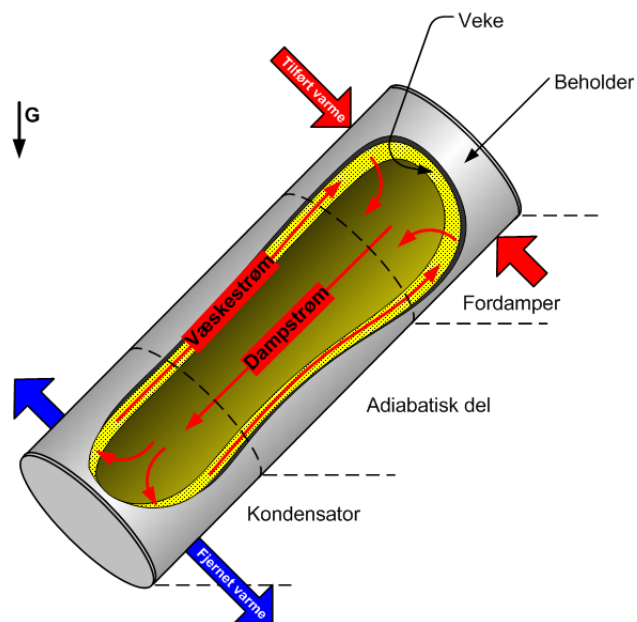
### 1.3 Oppbygging av rapport

Den første delen av rapporten inneholder et resultat av et litteraturstudium der der konseptet for varmerør, begrensninger og virkemåte for veker er nærmere presentert. Den neste delen av rapporten tar for seg utviklingen av beregningsprogrammet og beskriver modell og algoritmer i nærmere detalj. Etter dette er to beregninger utført med programmet presentert. Til slutt er en konklusjon og et forslag til videre arbeid. Som vedlegg ligger en datalisting av beregningsdelen av programmet samt data for arbeidsmedium.

## Kapittel 2

# Virkemåte for varmerør

### 2.1 Konsept for varmerør



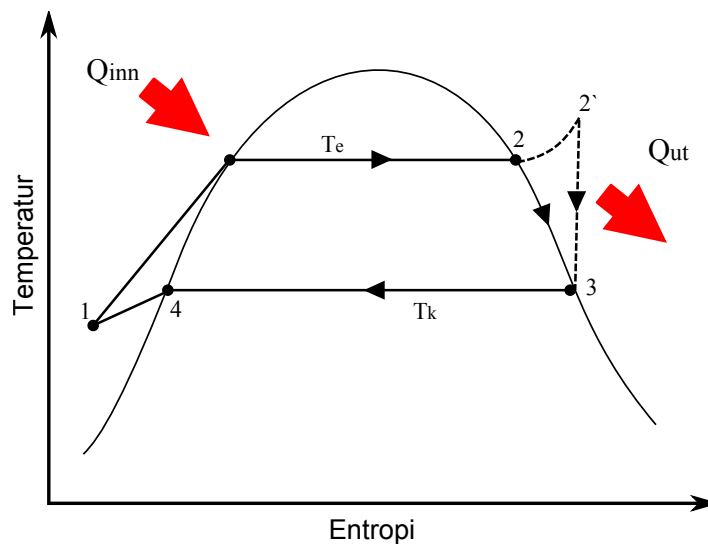
Figur 2.1: Skisse av varmerør med hovedkomponenter og prinsipp for virkemåte.

Et varmerør består i de fleste tilfeller av en hermetisk lukket beholder (dette er som regel et rør som er lukket i begge ender), en veke og et arbeidsmedium. Varme blir tilført i den ene enden av røret (fordamperdel) og blir ledet radielt gjennom rørvegg og vekestruktur til arbeidsmediumet som deretter fordamper (se figur 2.1). Fordampningen fører til en lokal trykkøkning som leder dampen gjennom den adiabatisk delen av røret og fram til kondensatordelen. Utvendig kjøling av kondensatordelen medfører at dampen frigir latent varme og kondenserer. Ved hjelp av kapillærkrefter blir kondensatet pumpet

gjennom vekstrukturen tilbake til fordamperen der prosessen gjentas.

Veken sin funksjon er å sørge for en mekanisme for å transportere arbeidsmediumet tilbake til fordamperen og å sørge for en jevn fordeling av arbeidsmedium over hele fordamperoverflaten. Prinsippet med veke er det som skiller et varmerør fra en “thermosyphone”. I en “thermosyphone” returnerer kondensatet til fordamperen kun ved hjelp av tyngdekraften. Fordamperen må derfor alltid være lokalisert i det laveste punktet. Denne begrensningen er ikke gjeldende for et varmerør som kan operere i mot tyngdekraften eller i miljøer uten gravitasjon (satellitter og andre utenomjordiske applikasjoner). For at prosessen ikke skal stoppe opp må veken kontinuerlig supplere fordamperen med væske. Dette kravet oppfylles så lenge det kapillære pumpetrykket  $\Delta P_{kapp}$  er større enn det totale trykktapet i røret. Trykktapet i røret består av følgende tre komponenter:

1.  $\Delta P_l$  som er differanse i væsketrykk mellom fordamper og kondensator.
2.  $\Delta P_v$  som er trykkdifferansen i damptrykket mellom fordamperen og kondensatoren.
3.  $\Delta P_g$  er tyngdekraftens innvirkning på væsketrykket.



Figur 2.2: T-s diagram for et varmerør. 1-2: Fordamper, 2-3: Dampstrøm fra fordamper til kondensator, 3-4: Kondensering, 4-1: Væskestrøm fra kondensator til fordamper. (Basert på figur 1.6 fra Faghri [7])

### 2.1.1 Dampstrømning fra fordamper til kondensator

Funksjonsmåte for et varmerør er skissert i T-s diagram 2.2. En grafisk framstilling av trykktapet er skissert i figur 2.3. Av figur 2.3 kan en se at laveste damptrykk er lokalisert ved inngangen til kondensator. Trykkgjenvinningen for dampfasen i kondensatoren kommer av at aksial impuls vil bli tapt til trykk ved at dampen bremses ned. Dette

er ofte tilfelle ved høye varmefflukser i fordamper. Ved lavere varmefflukser vil laveste damptrykk befinne seg ved enden av kondensatoren. Dette punktet blir kalt våtpunkt (eng: “wetpoint”). Av T-s diagrammet (punkt 2 eller 2') kan en se at dampen vil være enten mettet eller overopphetet når den forlater fordamperen. Når dampen kommer inn i kondensatoren vil den være mettet.

### 2.1.2 Kondensasjon i kondensator

Dampen vil kondensere ved gjennomsnittlig metningstemperatur  $T_k$  diktert av midlere damptrykk i kondensatoren. Under stasjonære forhold vil samme varmeffluks bli avgitt i kondensator som ble gitt i fordamper.

### 2.1.3 Væskestrømning fra kondensator til fordamper

Kondensatet vil deretter ved hjelp av kapillærkrefter bli pumpet fram til fordamperen igjen. På grunn av friksjon og hydrostatisk trykk vil væsketrykket synke fra kondensator til fordamper. Av T-s diagrammet (2.2) kan en se at væsken bli underkjølt fra punkt 4 til 1. Dette er til tross for at den er ved mettet tilstand når den kommer ut av kondensatoren. På grunn av at damptrykket på utsiden av veke stiger, vil trykkdifferansen mellom væske og damp stige etter hvert som væsketrykket faller. Væsken vil derfor ikke koke i veke selv om den har et lavere trykk enn i kondensatoren. Væsken følger en isoterm fra 3 til 4. Væsketemperatur ved inngang til fordamper er derfor den samme som i kondensator  $T_k$ .

### 2.1.4 Fordampning i fordamper

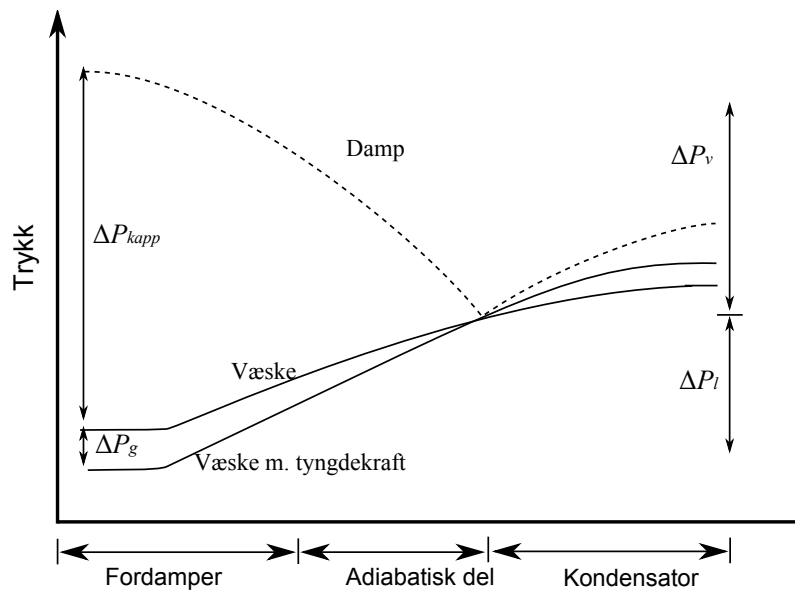
Væsken vil først bli varmet opp fra kondensatortemperatur  $T_k$  til gjennomsnittlig fordampertemperatur  $T_e$  før den fordamper.

## 2.2 Kapillært pumpetrykk $\Delta P_{kapp}$

I en væske vil molekylene være tiltrukket av hverandre. For et vilkårlig molekyl vil denne kraften normalt sett være lik i alle retninger fordi molekylet er omringet av andre molekyler. Et molekyl som er i eller ved en væskeoverflate vil ha et ulikt antall molekyler på ulike sider av seg selv. Dette fører til en resultantkraft innover fra væskeoverflaten. For å kompensere for denne resultantkraften vil væsken prøve å minimere væskeoverflaten. I tillegg vil væskemolekyler i kontakt med en fast overflate oppleve molekylære krefter fra molekylene i den faste overflaten. Disse kreftene vil enten være tiltrekkende eller frastøtende.

I grensen mellom en væske, en fast overflate og en væskeoverflate vil de to overnevnte kreftene føre til en resultantkraft. Ut fra dette vil væsken få en kontaktvinkel med overflaten. En konsekvens av denne kontaktvinkelen er at væskeoverflater i kontakt med en solid overflate i mange tilfeller vil få en krumming. Kapillærkrefter dannes på grunnlag av denne krumningen da trykket under en konkav overflate er lavere enn under en flat





Figur 2.3: Damp- og væsketrykk aksielt gjennom varmerør (våtpunkt lokalisert ved begynnelse av kondensator). Basert på figurer av Faghri [7]

eller konveks overflate. Veken i varmerøret pumper arbeidsmedium tilbake til fordamper ved hjelp av kapillærkrefter. I figur 2.4 kan en se en illustrasjon av væskeoverflaten sin krumming gjennom veken. Av figuren ser en att krumningen er størst i fordamper.

Maksimalt kapillær trykkdifferanse mellom to fluidier på grunnlag av overflatespenninger kan uttrykkes ved hjelp av **Young-Laplace-likningen**

$$\Delta P_{kapp,maks} = \frac{2\sigma \cos \theta}{r_{eff}} \quad (2.1)$$

der

$$\begin{aligned} \sigma &= \text{Overflatespenningen mellom fluidene} \\ r_{eff} &= \text{Effektiv poreradius} \\ \theta &= \text{Fluidets kontaktvinkel på flate} \end{aligned}$$

Den effektive poreradiusen  $r_{eff}$  kan beregnes for enkle geometrier, men for mer komplekse geometrier må verdiene beregnes eksperimentelt. En rekke forsøk for ulike vekestrukturer er blitt foretatt og blant annet listet opp i Faghri [7]. For disse strukturene er ofte fluidets kontaktvinkel bakt inn i verdien for effektiv poreradius, en må derfor kompensere for dette dersom en bruker et annen fluid enn det  $r_{eff}$  er beregnet på. Hvis dette er tilfelle sløyfes  $\cos \theta$  leddet i likning 2.1.

I tillegg til kapillærkreftene virker det en adskillelseskraft  $P_d$  (“disjoining pressure”) mellom væske og overflate. Denne kraften er et resultat av intermolekylære krefter dannet av molekulære og elektrostatiske interaksjoner. Denne kraften vil føre til at en væskefilm vil bli dratt oppover veggene i f.eks et rør. Det vil være en frastøtende van der Waals kraft tvers over væskeoverflaten. Denne kraften vil prøve å lage filmen tykkere for å minske energien. På grunn av gravitasjonskrefter vil væskefilmen bli tynnere med høyden [9]. Denne kraften vil være med på å danne et høyere pumpetrykk i tillegg til kapillærkreftene sin innvirkning. Adskillelseskraften  $P_d$  har størst betydning for mikrovarmerør og varmerør med renner (“grooved heat pipes”) [7]. Den blir derfor sett bort fra i denne rapporten.

### 2.3 Trykktap for væskestrøm $\Delta P_l$

I varmerør er det vanlig å behandle væskestrømmen i veken som stasjonær, inkompressibel og laminær. Vekestrukturen er i de fleste tilfeller relativt tynn og væskestrømmen kan derfor også betraktes som endimensjonal. Fra **Navier-Stokes-likning** kan en utligne **Darcy’s lov** for strømning i porøse medier

$$\frac{dp_l}{dz} = -\frac{\mu_l \dot{m}_l}{\rho_l A_{veke} K} \quad (2.2)$$

der

- $\mu_l$  = Dynamisk viskositet for væsken
- $\dot{m}_l$  = Massestrøm væske basert på gjennomsnittlig hastighet over vekens tverrsnittareal
- $\rho_l$  = Væskens tetthet
- $A_{veke}$  = Tverrsnittareal for veke
- $K$  = Permabilitet for veke

For strukturer med årer eller kanaler kan en bruke **Hagen-Poiseuilles-likning**

$$\frac{dp_l}{dz} = -\frac{2\tau}{r} \quad (2.3)$$

der  $r$  er indre radius av åre og  $\tau$  er skjærkraft.

Detaljer vedrørende permeabiliteten  $K$  er nærmere beskrevet i kapittel 4.3.

### 2.4 Trykktap for dampstrøm $\Delta P_v$

Det er blitt publisert flere ulike modeller for dampstrømning i varmerør. Flere av disse modellene neglisjerer kompressibilitet. Faghri[7] har oppsummert tre av disse modellene

som er *Første orden lukket form tilnærming* publisert av Cotter [2], *Andre orden tilnærming* publisert av Busse [1] og *Generalisert lukket form tilnærming* av Faghri [6]. Tower og Hainley [14] har utviklet en endimensjonal kompressibel modell basert på Busse[1] sitt tidligere arbeid. Denne modellen tar hensyn til kompressibilitet og monomer-dimer likevekt kan også beregnes.

Neglisjering av kompressibilitet er akseptabelt når den aksielle damphastigheten er lav i forhold til lydshastigheten. Kriteriet kan uttrykkes som

$$Ma = \frac{\bar{w}}{c} < 0,3 \quad (2.4)$$

der  $Ma$  er dampens machttall,  $\bar{w}$  er midlere aksial hastighet og  $c$  er lydshastigheten ved de gitte forhold.

For machttall større enn 0,3 bør en modell som tar hensyn til kompressibilitet benyttes. Hensyn til kompressibilitet kan også være viktig i høytemperatur- alkaliemetall-varmerør da trykket ofte er lavt, noe som resulterer i en høy damphastighet [3].

Modellene baserer seg på likningene for konservering av impuls og kontinuitet. Likningen for konservering impuls i radiell retning kan beskrives som [7]

$$v \frac{\partial v}{\partial r} + w \frac{\partial v}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial r} + \nu \left[ \frac{\partial}{\partial r} \left( \frac{1}{r} \frac{\partial}{\partial r} (rv) \right) + \frac{\partial^2 v}{\partial z^2} \right] \quad (2.5)$$

og i aksial retning

$$v \frac{\partial w}{\partial r} + w \frac{\partial w}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial r} + \nu \left[ \frac{\partial}{\partial r} \left( \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial w}{\partial r} \right) \right) + \frac{\partial^2 w}{\partial z^2} \right] \quad (2.6)$$

Likning for kontinuitet

$$\frac{1}{r} \frac{\partial}{\partial r} (rv) + \frac{\partial w}{\partial z} = 0 \quad (2.7)$$

Grensebetingelsene kan settes opp som følger

$$r = 0 : \frac{\partial w}{\partial r} = 0, v = 0 \quad (2.8)$$

og

$$r = R_v : w = 0, v = \begin{cases} v_\delta & 0 < z < L_e \\ 0 & L_e < z < L_e + L_a \\ -v_\delta & L_e + L_a \leq z < L_t \end{cases} \quad (2.9)$$

der

- $w$  = Hastighetskomponent i aksial retning
- $v$  = Hastighetskomponent i radiell retning
- $\nu$  = Kinematisk viskositet
- $R_v$  = Indre radius av varmerør
- $v_\delta$  = Hastighetskomponent for strømning mellom damp og veke
- $L_e$  = Lengde av fordamperdel
- $L_a$  = Lengde av adiabatisk del
- $L_t$  = Total lengde av varmerør

$v_\delta$  kan defineres med hensyn på Reynoldstallet

$$Re_r = \frac{\rho_v v_\delta R_v}{\mu_v} \quad (2.10)$$

I kondensator og fordamper vil dampen strømme i radiell retning.  $v_\delta$  vil derfor være  $> 0$  i fordamper og  $< 0$  i kondensator. Det kan være gunstig å definere et radielt Reynoldstall med hensyn på massestrøm

$$Re_r = \frac{1}{2\pi\mu_v} \frac{d\dot{m}_v}{dz} \quad (2.11)$$

der  $\mu_v$  er dynamisk viskositet for damp og  $\frac{d\dot{m}_v}{dz}$  er massestrømmen damp som blir generert i fordamper eller forsvinner (kondenserer) i kondensator.

Massestrømmen av damp kan uttrykkes som en funksjon  $\dot{m}_v(z)$  gitt ved

$$\dot{m}_v(z) = \frac{Q(z)}{h_{fg}} = \begin{cases} \frac{z}{L_e} \frac{Q_e}{h_{fg}} & 0 \leq z \leq L_e \\ \frac{Q_e}{h_{fg}} & L_e \leq z \leq L_e + L_a \\ \frac{L_t - z}{L_t - (L_e + L_a)} \frac{Q_e}{h_{fg}} & L_e + L_a \leq z \leq L_t \end{cases} \quad (2.12)$$

### 2.4.1 Første orden lukket form tilnærming

Denne modellen baserer seg i hovedsak på Cotter[2] sitt arbeid.

For radielle Reynoldstall  $|Re_r| \ll 1$  antar Cotter at de viskøse kreftene dominerer strømningen og hastighetsprofilen i aksial retning vil ha en parabolisk form som kan beskrives med **Hagen-Poiseuilles likning**

$$w(r) = 2\bar{w} \left( 1 - \frac{r^2}{R_v^2} \right) \quad (2.13)$$

For Reynoldstall  $|Re_r| \gg 1$  tar hastighetsprofilen en annen form og er proporsjonalt med

$$\cos \left[ \frac{\pi}{2} \left( \frac{r}{R_v} \right)^2 \right] \quad (2.14)$$

Beskrevet med hensyn på  $r$  får vi følgende likning

$$w(r) = 2\bar{w} \cos \left[ \frac{\pi}{2} \left( \frac{r}{R_v} \right)^2 \right] \quad (2.15)$$

Cotter forutsetter at strømmingen er fullt utviklet i den adiabatisk delen av varmerøret og at varmetilførsel og varmefjerning langsmed fordampere og kondensator er konstant. Videre er  $\rho_v$  antatt til å være konstant. Dette medfører at strømningshastigheten  $v$  er liten i forhold til sonisk hastighet. Strømmingen er også antatt til å være laminær.

I fordampere øker den aksielle hastigheten  $v$  på bekostning av trykk. Dette trykktapet vil delvis bli gjenvunnet i kondensatoren. Ved  $|Re_r| \gg 1$  er trykkgjenvinningen 40% i følge Cotter sin modell.

På grunnlag av dette har Cotter utledet to likninger for trykkfall[7]

$$\frac{dp_v}{dz} = -\frac{8\mu_v \dot{m}_v}{\pi \rho_v R_v^4} \left( 1 + \frac{3}{4} Re_r - \frac{11}{270} Re_r^2 + \dots \right) \quad (2.16)$$

som gjelder for  $|Re_r| \ll 1$ . Og

$$\frac{dp_v}{dz} = -\frac{s \dot{m}_v}{4 \rho_v R_v^4} \frac{d\dot{m}_v}{dz} \quad (2.17)$$

som gjelder for  $|Re_r| \gg 1$  og der  $s = 1$  for fordampere og  $s = 4/\pi^2$  for kondensator.

### 2.4.2 Andre orden tilnærming

Busse [1] utviklet en modell der strømmingen fortsatt var antatt til å være laminær og inkompressibel. Videre antok Busse at den aksiale trykkgradienten ikke varierte over tverrsnittet på varmerøret. Denne tilnærmingen tar utgangspunkt i likning 2.5 til 2.7 og gir følgende trykktap i fordampere del

$$\Delta p_e = -\frac{8\mu_v Q_e}{\pi \rho_v R_v^4 h_{fg}} \frac{L_e}{2} \left[ 1 + Re_r \left( \frac{7}{9} - \frac{8A}{27} + \frac{23A^2}{405} \right) \right] \quad (2.18)$$

i adiabatisk del

$$\Delta p_a = -\frac{8\mu_v Q_e}{\pi \rho_v R_v^4 h_{fg}} L_e \left[ 1 + \frac{Re R_v}{8 L_a} \left( \frac{8(A-a)}{27} - \frac{23(A^2 - a^2)}{405} \right) \right] \quad (2.19)$$

i kondensatordel

$$\Delta p_a = -\frac{8\mu_v Q_e}{\pi \rho_v R_v^4 h_{fg}} \frac{L_c}{2} \left[ 1 - Re \left( \frac{7}{9} - \frac{8a}{27} + \frac{23a^2}{405} \right) \right] \quad (2.20)$$

og der  $Re_z$  er aksielt Reynoldstall definnert ved

$$Re_z = \frac{2R_v \bar{w}_a \rho_v}{\mu_v} \quad (2.21)$$

og korreksjonsfaktoren  $A$  er gitt ved

$$A = \frac{15}{22} \left[ 5 + \frac{18}{Re_r} - \left[ \left( 5 + \frac{18}{Re_r} \right)^2 - \frac{44}{5} \right]^{1/2} \right] \quad (2.22)$$

### 2.4.3 Kompressibel endimensjonal analyse

Tower og Hainley [14] utviklet en algoritme for modellering av dampstrømning i varmerør. Modellen er en videreutvikling av Busse [1] sin modell. I denne modellen er strømmingen antatt til å være laminær, den ideelle gassloven gjelder. Faselikevekt er bare ved vegg, det er dermed ingen væskedråper i dampstrømmen. Radiell trykkgradient blir neglisjert og hastighetskomponentene ved endene av varmerøret er null.

Modellen er bare tiltenkt for bruk i fordampere og adiabatisk del da strømningsmønstret i kondensatoren kompliseres på grunn av turbulens og reversert strømning.

Alkalimetaller i dampfase har ofte en signifikant mengde dimerisering [5]. Med dimerisering menes her en kjemisk union mellom to identiske dampmolekyl (monomer). Likevekten mellom dimer og monomer endres ved endring i trykk og temperatur. Mengden påvirker tilstanden for gassen. Denne modellen er utviklet for å ta hensyn til dette. Modellen består i hovedsak av fire likninger

$$\begin{aligned} & \left[ 1 - \frac{4\rho_m v_m^2}{3p} (GB)(GRP) \right] \frac{d \ln p}{dz} + \left[ \frac{4\rho_m v_m^2}{3p} (GB)(GRT) \right] \frac{d \ln T_m}{dz} \\ & \quad - \frac{4\rho_m v_m^2}{3p} (HB) \frac{dA}{dz} \\ & = -\frac{8\mu v_m}{R_v^2 p} \left( 1 + \frac{2A}{3} \right) - \frac{8\rho_m v_m^2}{3p} (GB) \left( \frac{d \ln m}{dz} - \frac{d \ln A_v}{dz} \right) \end{aligned} \quad (2.23)$$

Kjemisk likevekt i dampen blir vurdert og gir  $h_m$  som funksjon av  $T_m$  og  $p$

$$\begin{aligned} & \left[ \left( \frac{\partial h_m}{\partial \ln p} \right)_{T_m} - F_3 v_m^2 (GRP) \right] \frac{d \ln p}{dz} + \\ & \left[ \left( \frac{\partial h_m}{\partial \ln T_m} \right)_p + F_3 v_m^2 (GRT) \right] \frac{d \ln T_m}{dz} + (HAF) \frac{dA}{dz} \\ & = 8\mu \left[ \frac{(1 + 2A^2/9)v_m}{\rho_m R_v^2} \right] - (HVE) \left( \frac{d \ln m}{dz} - \frac{d \ln A_v}{dz} \right) \end{aligned} \quad (2.24)$$

$$\begin{aligned} & \left[ 1 - \frac{4\rho_m v_m^2}{p} (1 - A/3)^2 (GRP) \right] \frac{d \ln p}{dz} + \left[ \frac{4\rho_m v_m^2}{p} (1 - A/3)^2 (GRT) \right] \frac{d \ln T_m}{dz} \\ & \quad - \left[ \frac{4\rho_m v_m^2}{3p} (1 - A/3) \right] \frac{dA}{dz} \\ & = \frac{-8\mu v_m}{p R_v^2} (1 - 4A/3) - \left[ \frac{4\rho_m v_m^2}{p} (1 - A/3)^2 \right] \left( \frac{d \ln m}{dz} - \frac{d \ln A_v}{dz} \right) \end{aligned} \quad (2.25)$$

Massefluks kan relateres til varmekraft ut fra følgende likning

$$\frac{dq_R}{dz} = \frac{1}{2\pi R_v} \frac{dm}{dz} \left[ h_{fg} + \left( \frac{1}{2\pi R \rho_R} \frac{dm}{dz} \right)^2 \right] \quad (2.26)$$

der

$$\begin{aligned} F_3 &= 8 \left( \frac{1}{4} - \frac{A}{10} + \frac{A^2}{30} - \frac{2A^3}{945} \right) \\ HAF &= -\frac{2}{5} \left( 1 - \frac{2A}{3} + \frac{4A^2}{63} \right) \\ HVE &= h_m - H_R + \left( \frac{3F_3 v_m^2}{2} \right) \\ GRP &= 1 - \frac{\partial \ln R_g}{\partial \ln p} \\ GRT &= 1 + \frac{\partial \ln T_m}{\partial \ln p} \\ GB &= \left( 1 - \frac{A}{6} + \frac{2A^2}{45} \right) \\ HB &= -\left( \frac{1}{6} - \frac{4A}{45} \right) \\ H_R &= h_R + \frac{u_R}{2} \end{aligned}$$

og

$$\begin{aligned} h_m &= \text{Entalpi} \\ \rho_m &= \text{Midlere tetthet} \\ v_m &= \text{Midlere aksiell hastighet} \\ T_m &= \text{Midlere temperatur} \\ p &= \text{Trykk} \\ A &= \text{Busses korreksjonsfaktor (gitt ved likning 2.22)} \\ h_{fg} &= \text{Fordampningsentalpi (heat of evaporation)} \\ R_g &= \text{Gasskonstant} \\ \rho_R &= \text{Tetthet for damp i umiddelbar nærhet av veke} \\ h_R &= \text{Entalpi for damp i umiddelbar nærhet av veke} \\ u_R &= \text{Radiell hastighetskomponent} \\ \mu &= \text{Dynamisk viskositet} \end{aligned}$$

Likning 2.23, 2.24, 2.25 og 2.26 kan med passende grensebetingelser løses med hensyn på  $p$ ,  $A$  og  $T_m$ .

#### 2.4.4 Sammenligning av modeller for dampstrøm

Faghri [7] har foretatt en sammenligning av Cotters *Første orden lukket form tilnærming*, Busses *Andre orden tilnærming* og Faghris *Generalisert lukket form tilnærming* ved tre Reynolds nummer. Alle modellene stemte godt overens ved det laveste Reynoldstallet  $Re_r = 0,6$  (figur 6.1). Ved moderate Reynoldstall  $Re_r = 2,4$  (figur 6.3) gir modellene litt ulike resultat. Faghri hevder Cotter sin teori gir en motsatt kurving på trykkgradientkurven i kondensatoredelen. Ved Reynoldstall på  $Re_r = 1000$  (figur 6.5) gir modellene ganske ulikt resultat. Faghri og Cotter sin modell er ganske like mens Busse sin modell fraviker en del. Alle modellene gir det samme totale damptrykketapet (hvis  $s = 1$  i adiabatisk del i Cotters modell for høye Reynoldstall) og har den samme trykkgradienten i den adiabatisk delen. Faghri nevner ingen ting om hvilken modell som er mest rett.

Tower og Hainley [14] testet sin algoritme opp mot reelle data og en tidligere modell. Algoritmen gir omtrent de samme resultatene som de reelle dataene og den tidligere modellen.

### 2.5 Varmegjennomgang radielt gjennom beholdervegg

Varmen som blir tilført i fordamperdelen og som blir bortført i kondensatordelen må bli ledet gjennom ytterveggen og vekestrukturen i varmerøret. Hvis en ser bort i fra væsehastigheten i veke så foregår denne varmetransporten i hovedsak ved konduksjon. I fordamperen vil det også være koking/fordampning eller konveksjon og i kondensator vil det være kondensasjon. Varmeovergangstall for de ulike mekanismene og materialene må beregnes. For varmerør med flytende metaller vil den høye konduktiviteten til arbeidsmediumet føre til en relativt høy konduktivitet gjennom veke. For et arbeidsmedium med lav konduktivitet vil vekedesignet være svært avgjørende for den totale varmemotstanden. En annen konsekvens av flytende metall er at fordampning vil foregå på overflaten av vekestrukturen i motsetning til andre arbeidsmedium der en kan risikere å få koking internt i veke (se kapittel 3.2 ).

Hvis en forutsetter at fordampning skjer ved vekeoverflaten kan en benytte Fouriers lov å beregne varmetransporten gjennom veke og sylindervegg. Fouriers lov med hensyn på sylindervegg er gitt ved [4]

$$Q = -2\pi kL \frac{T_1 - T_2}{\ln(r_2/r_1)} \quad (2.27)$$

Der  $L$  er tykkelse på sylindervegg,  $r_2$  og  $T_2$  er ytterradius og ytterveggtemperatur og  $k$  er varmekonduksjonskoeffisient for materiale.

#### 2.5.1 Fordampning i fordamperdel

Kapittel 3.2 tar for seg koking i fordamperdel.



### 2.5.2 Kondensering i kondesatordele

I kondensatoren vil arbeidsmediumet kondensere på vekeoverflaten gitt at temperaturen på overflaten er lavere enn metningstemperaturen ved det aktuelle damptrykket. Siden kondenseringen skjer på en veke vil det ikke dannes en kondensasjonsfilm på samme måte som for kondensering på glatte flater. En tynn film vil dannes på vekeoverflaten og væske fra denne flaten vil bli dratt inn i porene i veke ved hjelp av kapillærkrefter. I grensen mellom film og damp vil det eksistere en termisk motstand lik den som eksisterer i fordamperdelen.

### 2.5.3 Termisk motstand i damp-væske grensesnitt

I grenseflaten mellom væske og damp vil det være en utveksling av molekyler mellom væskeoverflaten og dampen. Ved likevekt er denne utvekslingen lik i begge retninger og temperaturen i dampen og væskeoverflaten vil være den samme. Ved kondensasjon må fluksen av dampmolekyler inn i væskefasen være større enn fluksen av væskemolekyler inn i dampfasen. Det motsatte skjer ved fordampning. Det er en termisk motstand forbundet med denne faseovergangen.

Carey [11] har utviklet en likning for varmetransportkoeffisienten  $h_\delta$  ved grenseflaten

$$h_\delta = \frac{q_\delta}{(T_v - T_l)} = \left( \frac{2\alpha}{2 - \alpha} \right) \left( \frac{h_{fg}^2}{T_v v_{fg}} \right) \sqrt{\frac{M_v}{2\pi} R_u T_v} \left( 1 - \frac{p_v v_{fg}}{2h_{fg}} \right) \quad (2.28)$$

Carey har antatt  $(p_v - p_l)/p_v \ll 1$  og  $(T_v - T_l)/T_v \ll 1$ .

Koeffisienten  $\alpha$  er av betydning for resultatet av likningen ovenfor. Faghri foreslår å sette denne til ca. 1. For forurensede arbeidsmedium bør en lavere verdi velges.

### 2.5.4 Forandring i mettet damptrykk over en krum væskefilm

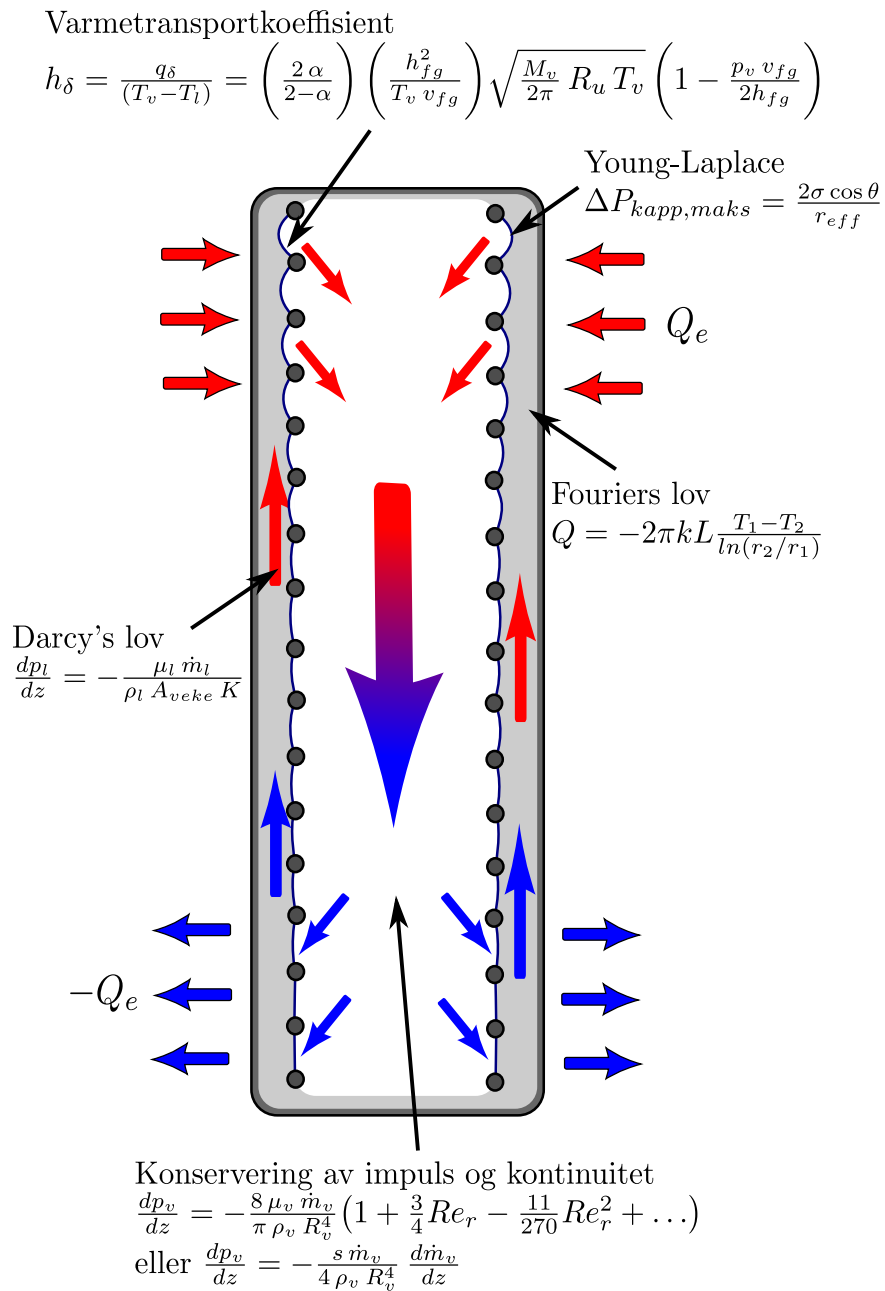
På grunn av adskillelseskraften  $p_d$  og kapillærkreftene ved væske-damp grensesnittet vil det være en forandring i det mettede damptrykket over en krum væskeflate. Denne forandringen  $p_{v,\delta}$  kan beskrives som følgende [7]

$$p_{v,\delta} = p_{sat}(T_\delta) \exp \left[ \frac{p_{v,\delta} - p_{sat}(T_\delta) - 2\sigma/r_{eff} + p_d}{\rho_l R_g T_\delta} \right] \quad (2.29)$$

Forandringen i mettet damptrykk kan i de fleste tilfeller neglisjeres.

## 2.6 Diskusjon

Ytelsen til et varmerør kan defineres som varmetransportkapasiteten fra fordamper til kondensator. Denne ytelsen er en konsekvens av en rekke fysiske fenomener som er gjennomgått tidligere i dette kapitlet. Figur 2.4 gir en enkel oversikt over noen av de viktigste fenomenene og styrende likninger i et varmerør.



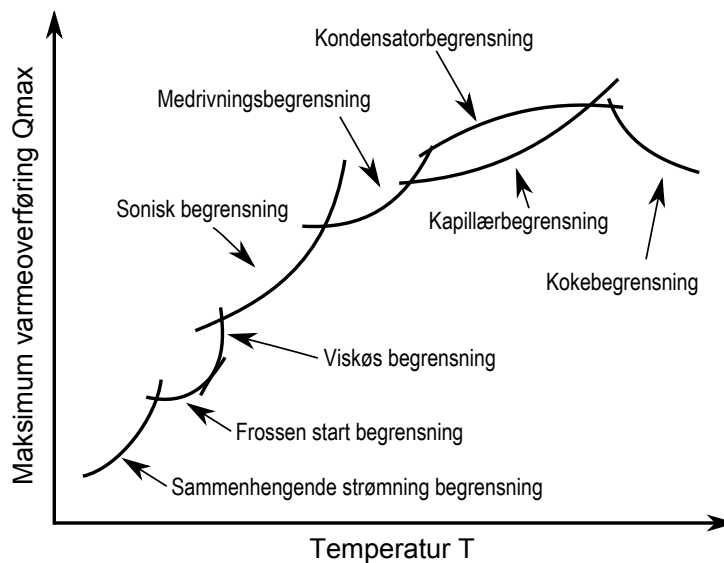
Figur 2.4: Forenklet oversiktsskisse over fysiske fenomener i et varmerør

## Kapittel 3

# Varmetransportbegrensninger for varmerør

### 3.1 Introduksjon

For å kunne lage en modell for et varmerør er det viktig å forstå de ulike begrensningene. Et varmerør leder varme fra en kilde til et sluk. I et tradisjonelt varmerør skjer dette ved aksiell varmeovergang i fordampner og kondensator. Figur 3.1 viser ulike begrensende fenomener for varmeovergangen.



Figur 3.1: Maksimal varmeovergang  $Q_{maks}$  som funksjon av operasjonstemperatur.

Varmetransportbegrensninger i et varmerør stammer i all hovedsak fra veien sin evne til å returnere kondensat til fordampneren og termodynamiske barrierer i dampstrømmen.

## 3.2 Kokebegrensning

Kokefenomenet er godt dokumentert i litteraturen og denne rapporten kommer ikke til å gå i dybden når det gjelder generell koking. Koking fra veker skiller seg derimot fra tradisjonell koking og er generelt mer komplisert. Ulike fenomener kan råde for ulike veker og betingelser. I figur 3.2 har Faghri [7] definert fire ulike former for varmtransport og dampgenerering i en veke. Faghri forklarer disse ulike typene som følger:

Type 1 For flytende metaller under en lav til moderat varmekraft vil varmtransporten gjennom veken være dominert av konduksjon. Fordampning skjer fra overflaten av arbeidsmediumet og det foregår ingen koking internt i veken.

Type 2 Ved større varmekraft vil fordampningen fra overflaten øke og veken vil ikke klare å supplere med nok væske. Dersom varmekraften ikke blir redusert kan veken tørke ut. Denne begrensningen er styrt av kapillære begrensninger i veken. Heller ikke her vil koking inntreffe internt i veke.

Type 3 I noen tilfeller der temperaturdifferansen over veken er stor over veken vil koking med bobledannelse inntreffe internt i veke. Hvis sirkulasjonen av arbeidsmediumet er drevet av kapillærkrefter kan denne kokingen føre til forstyrrelser og ødelegge for den kapillære pumpeeffekten.

Koking med bobledannelse er mest vanlig for arbeidsmedium med lavere varmeledningsevne. Dette fenomen er mindre aktuelt for flytende metaller der de to første typene for varmoverføring er gjeldene.

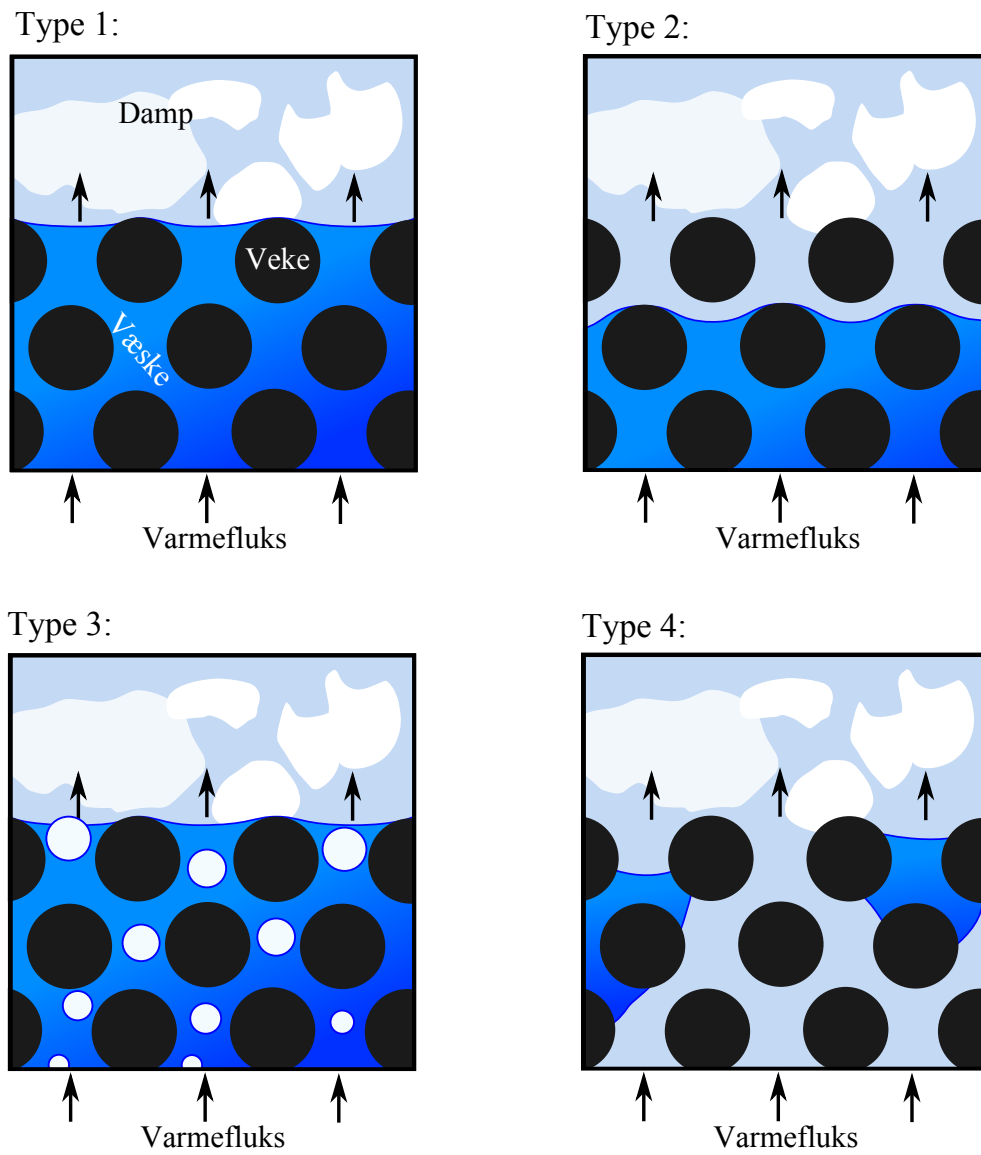
Type 4 Ved store varmekraft vil bobledannelse skje på innsiden av beholderveggen. Boblene vil vokse og forene seg sammen og skape en film av damp over veggen. Denne filmen hindrer ny væske i å komme til, og som et resultat av dette vil veggtemperaturen stige og varmerøret kan brenne ut. Dette er en øvre begrensning (kritisk varmekraft) for koking.

Den kritiske varmekraften for vekede overflater er antatt å være høyere enn for tilsvarende glatte overflater. Faghri[7] rapporterer om eksperimenter utført av Reiss i 1968 der han kokte natrium i et rillet varmerør og oppnådde varmekraft på  $2000 \text{ W/cm}^2$  ved veggtemperatur på  $800^\circ\text{C}$ . Dette er mye høyere enn det som kan oppnås ved tilsvarende temperatur og "pool boiling".

Det konkluderes med at for varmerør med flytende metaller som arbeidsmedium er det sjelden at kokebegrensningen er den begrensende faktoren. I de fleste tilfeller vil det være den kapillære pumpeeffekten som begrenser varmekraften. Den kritiske overopphetingen kan i følge Faghri[7] beregnes ut fra

$$\Delta T_{kritisk} = \frac{2\sigma T_v}{h_{fg} \rho_v} \left( \frac{1}{R_b} - \frac{1}{R_{men}} \right) \quad (3.1)$$

Der  $\sigma$  er overflatespenning i  $\text{N/m}$  for væsken,  $T_v$  er damptemperatur,  $R_b$  er bobleradius ved vegg-veke grensesnittet og  $R_{men}$  er radius for væske-damp overflate.  $R_{men}$  er



Figur 3.2: Ulike typer varmetransport og dampgenerering i en veke (Basert på figur fra Faghri [7]).

avhengig av vekeegenskapene og vanskelig å beregne. Faghri hevder at  $R_{men}$  i de fleste tilfeller kan antas å være omtrent lik effektiv poreradius  $r_{eff}$  for veke. Basert på den kritiske temperaturen gitt i likning 3.1 kan den kritiske varmetransportbegrensningen beregnes ut fra

$$\Delta Q_{kritisk} = \frac{2 \pi L_e k_{eff} \Delta T_{kritisk}}{\ln(R_i/R_v)} \quad (3.2)$$

her er  $R_i$  indre radius av varmerørbeholder.

Også  $R_b$  er svært vanskelig å estimere nøyaktig, men likning 3.3 kan gi et estimat. Eventuelt anbefaler Faghri å sette  $R_b$  til  $10^{-7} m$  for konvensjonelle varmerør.

$$R_b = \sqrt{\frac{2 \sigma T_{sat} k_l (v_v - v_l)}{h_{fg} q_r}} \quad (3.3)$$

der  $q_r$  er radiell varmefluks og  $v_l$  og  $v_v$  er spesifikt volum for mettet damp og væske.

Varmeovergangskoeffisienten gjennom veken kan med tilstrekkelig nøyaktighet beregnes ut fra en konveksjonsmodell.

### 3.3 Kapillærbegrensning

Hvis en ser bort fra trykktap på grunnlag av fordampning og kondensasjon så kan det generelle uttrykket for kapillærbegrensningen uttrykkes som følger

$$\Delta P_{kapp,maks} \geq \Delta P_l + \Delta P_v + \Delta P_g \quad (3.4)$$

Der et uttrykk for  $\Delta P_{kapp,maks}$  er gitt av likning 2.1 og  $\Delta P_g$  kan uttrykkes som følger

$$\Delta P_g = \rho_l g L_t \sin \phi \quad (3.5)$$

#### 3.3.1 Lokalisering av våtpunkt

I likning 3.5 er våtpunktet antatt til å være ved enden av kondensatoren (derfor  $L_t$ ). Hvis våtpunktet er lokalisert et annet sted må dette tas hensyn til. I våtpunktet vil væsketrykk og damptrykk være likt. Ved lave flukser kan våtpunkt antas til å være ved enden av kondensatoren. For høyere flukser og lave trykk der damptrykkjennvinningen i kondensatoren er større enn væsketrykktapet i kondensatoren, er våtpunktet lokalisert ved begynnelsen av kondensatoren (se figur 2.3). Hvis dette hensynet ikke blir tatt og våtpunktet blir feilaktig satt til å være ved enden av kondensator, vil væsketrykket være høyere enn damptrykket ved inngang til kondensator.

Uttrykk for  $\Delta P_l$  og  $\Delta P_v$  er gitt i kapittel 2. Også for disse trykkdifferansene spiller våtpunktet inn.

### 3.4 Kondensatorbegrensning

Ved stasjonære forhold vil varmen som er tilført i fordampere være lik varmen som blir ledet bort i kondensator. Hvis kondensatoren ikke er i stand til å kvitte seg med like mye varme som er blitt tilført vil dette bli en begrensende faktor for varmerøret. Kondensatoren kvitter seg med varme enten gjennom stråling eller konveksjon (tvungen eller naturlig). Varmeovergangskoeffisienter kan forbedres med geometriske modifikasjoner som for eksempel finner.

Siden kondensatorbegrensningen er relatert til ytre forhold vil dette ikke bli vektlagt i denne rapporten.

### 3.5 Medrivningsbegrensning

Damp og væske i et varmerør strømmer i motsatt retning av hverandre. Væsken er i hovedsak beskyttet av vekestrukturen, men i grenseflaten mellom væske og damp vil skjærspenninger oppstå. Ved høy nok damp hastighet kan disse spenningene få bølger til å oppstå på væskeoverflaten og væskedråper vil kunne løsrive seg og bli ført tilbake til kondensator med dampstrømmen. Denne kortslutningen i væskestrømmen kan føre til at fordamperdelen tørker ut på grunn av væskemangel.

Dampstrømmen i et varmerør vil ha en mye større hastighet enn væskestrømmen. På grunnlag av dette kan en anta at skjærkreftene som virker i væske-damp grensesjiktet er proporsjonale med det dynamiske trykket til dampstrømmen og væskearealet som er i kontakt med dampen.

Cotter har utviklet et kriterium for medrivningsbegrensningen i varmerør basert på et Webernummer. Medrivningsbegrensningen er nådd når Webernummeret er lik 1 og kan uttrykkes som følger

$$We = \frac{2 R_{h,w} \rho_v \bar{w}^2}{\sigma} = 1 \quad (3.6)$$

der  $R_{h,w}$  er hydraulisk radius for porer i vekeoverflate og kan uttrykkes som

$$R_{h,w} = \frac{2A_{lv}}{P}$$

der  $A_{lv}$  er areal av de individuelle overflateporene på veke og  $\sigma$  er koeffisient for overflatespenning og  $P$  er fuktet omkrets.

For "screen" veke gir Faghri følgende tall  $R_{h,w} = 0,5W$  og for aksielle spor  $R_{h,w} = W$  der  $W$  er distanse mellom vaiere i en screen veke og sporvidde for sporveke.

Væskemedrivning for kapillært drevne varmerør er ikke blitt observert eksperimentelt og det er derfor usikkerheter om dette fenomenet har noen innvirkning for konvensjonelle varmerør. Faghri hevder at kapillærstrukturen vil beskytte mot dannelse av overflatebølger og medrivning. En artikkel av Peterson og Bage [12] støtter også denne teorien.

### 3.6 Sonisk begrensning

Gjennom fordamperdelen på et varmerør vil massestrømmen med damp øke på grunn av fordampning av væske. Det omvendte skjer i kondensatordelen. Dette er analogt til en konvergerende-divergerende dyse. Ulikheten er at det i dysen er tverrsnittarealet som endrer seg i stedet for massestrømmen.

Hvis trykket i kondensatordelen er tilstrekkelig lavt vil hastigheten bli supersonisk ( $Ma = 1$ ) ved enden av fordampere. Ytterligere trykkreduksjon i kondensator vil ikke føre til økt massestrøm, kun økt strømningshastighet etter fordampedel. Dette vil være en øvre begrensning for massestrøm og aksial varmeffluks. Faghri [7] gir følgende korrelasjon for sonisk begrensning

$$Q_s = \frac{\rho_0 c_0 h_{fg} A_v}{\sqrt{2(K' + 1)}} = A_v \rho_0 h_{fg} \left[ \frac{K' R_g T_0}{2(K' + 1)} \right]^{1/2} \quad (3.7)$$

der  $\rho_0$ ,  $c_0$  og  $T_0$  er tetthet, lydhastighet og temperatur ved fordampere.

Korrelasjonen gitt ovenfor tar ikke hensyn til friksjon. Dette fører til at den kalkulerede begrensningen blir noe høyere enn den reelle. Dette gjelder særlig for varmerør som opererer ved lavere arbeidstemperaturer og trykk. F.eks refererer Faghri til eksperimenter der natrium varmerør har vist avvik på 20 % for arbeidstemperaturer rundt  $500^\circ\text{C}$ .

En mer nøyaktig løsning kan oppnås ved å løse to dimensjonale likninger for kompressibel strømning, impuls, energibalanse og tilstandslikningen i dampkjernen.

Varmerøret vil som regel ikke stoppe å operere når den soniske begrensningen nåes. Denne begrensningen er derfor ikke like kritisk som noen av de andre. For å øke varme-transporten ytterligere kan en øke temperaturen i fordampere ytterligere. Den soniske begrensningen vil ofte nåes i oppstartfasen av et varmerør, men vil forsvinne når arbeidstemperaturen blir høy nok.

### 3.7 Viskøs begrensning

Den viskøse begrensningen er ofte referert til som nedre damptrykkbegrensning. Denne begrensningen nåes når væsketrykket i fordampere går mot null. I praksis vil deler av fordampere slutte å fungere når denne begrensningen nærmer seg. Matematisk kan dette uttrykkes som

$$\bar{q}_{vis} = \frac{D_v^2 h_{fg} \rho_0 p_0}{64 \mu L_{eff}} \quad (3.8)$$

der

$$\begin{aligned} D_v &= \text{Diameter av dampstrømningsområde} \\ L_{eff} &= \text{Effektiv lengde av varmerør} \end{aligned}$$



Den viskøse begrensningen kan betraktes som det motsatte av den soniske begrensningen. Når den soniske begrensningen nåes er det treghetskrefter som dominerer og viskøse kan neglisjeres.

Viskøs begrensning kan være aktuelt for varmerør med alkalimetaller da damptrykket ofte er relativt lavt. Av denne grunn kan det også være svært aktuelt ved oppstart av varmerør siden temperaturen da normalt vil være lav [7].

### 3.8 Frossen start begrensning

Alkalimetaller er i fast form ved normal romtemperatur. Et varmerør med alkalimetall som arbeidsmedium vil derfor ha alkalimetall i fast form ved oppstart. Når varme tilføres i fordamperdel vil alkalimetallet først bli flytende og til slutt fordampe. Dampen vil deretter kondensere i kondensatoren og normalt sett bli transportert tilbake til fordamperdel ved hjelp av vekestrukturen. Men problemer kan oppstå hvis utfrysinger forekommer i veke eller kondensatordel. Dette kan skje fordi deler av varmerøret ikke har oppnådd høy nok temperatur. Arbeidsmediumet som fryser i kondensatoren kan føre til at for lite væske føres tilbake til fordamper og uttørking kan forekomme. En vil da ikke klare å starte opp varmerøret.

Dette er i hovedsak et transient problem og er utelukket fra denne rapporten.

### 3.9 Sammenhengende strømning begrensning

For veldig små varmerør kan dampstrømmen bli usammenhengende. Dette fører til en drastisk reduksjon i varmerørets ytelse. På grunnlag av at denne begrensningen bare gjelder for svært små varmerør vil denne rapporten ikke gå nærmere i dybden på dette emnet.

### 3.10 Behov for å beregne transiente forhold

Ved oppstart av varmerør med alkalimetaller som arbeidsmedium kan begrensninger oppstå som bremses eller stopper oppstartsprosessen. Dette kan være frossen start- eller viskøs begrensning. Behovet for å kunne modellere transiente forhold er derfor til stede. En modell som tar hensyn til transiente forhold vil fort bli meget avansert og det blir derfor ikke fokusert på i denne rapporten.

### 3.11 Diskusjon

De viktigste begrensningene for ytelsen til et varmerør er kapillærbegrensning, kokebegrensning, sonisk begrensning, samt viskøs begrensning. Frossen start begrensning er en viktig begrensning for alkalimetallvarmerør men kommer utenom denne rapporten da transiente forhold ikke blir vurdert. Andre begrensninger er også nevnt men disse er mindre aktuelle for varmerør som benytter alkalimetaller som arbeidsmedium.

## Kapittel 4

# Virkemåte for veker i varmerør

### 4.1 Ulike vekestrukturer og design

I kapittel 3.2 påpekes det at den kapillære pumpeeffekten er en av de viktigste begrensningene for et varmerør. Av likning 2.1 kan vi lese at det maksimale kapillærtrykket øker med minkende effektiv poreradius  $r_{eff}$ . Permabiliteten  $K$  i likning 2.2 øker med økende poreradius. Forsøk på å øke kapillærtrykket  $\Delta P_{kapp}$  ved å minke effektiv poreradius  $r_{eff}$  vil derfor føre til reduksjon av permeabilitet og dermed økning av trykktap i væskestrøm  $\Delta P_l$ . Optimalisering av ytelsen i et varmerør krever en vekestruktur som kan gi et høyt kapillært trykk samtidig som den gir en lav motstand mot væskestrømmen.

Ulike vekestrukturer prøver på ulik måte å redusere trykktapet i væskestrømmen uten at dette går på bekostning av kapillærtrykket. En kan dele vekestrukturer i homogene og kompositt veker. Homogene veker er som regel enkle å designe og installere. Komposittveker er ofte mer avanserte og dyre.

#### 4.1.1 Homogene veker

Homogene veker er konstruert med en type materiale eller ved hjelp av en type bearbeiding. En av de vanligste og enkleste vekene av denne typen er en omviklet netting på innerveggen av varmerøret. Nettingen kan være av stoff eller metall. Størrelsen på de rektangulære porene mellom trådene i nettingen bestemmer kapillærtrykket. Denne typen veke genererer et relativt høyt kapillært pumpetrykk men har en relativt lav permeabilitet. Radiell termisk konduktivitet er også relativt lav i forhold til andre homogene veker.

Sintrede veker er en type veke der metallpartikler blir presset/smeltet sammen på innsiden av varmerørveggen slik at de klebrer seg sammen. Fordi et metallpulver blir bruk har denne formen for veke relativt høy termisk konduktivitet. Veken genererer også et relativt høyt kapillært pumpetrykk fordi kanalene blir relativt små. Permeabiliteten er dermed også lav.

Aksielle spor er også blitt brukt som veker. Størrelsen på kanalene er relativt stor i forhold til sintrede veker eller nettingveker. Konsekvensen er lav permeabilitet og lavt

kapillærtrykk. Den radielle konduktiviteten er relativt høy. Denne typen veke er pålitelig og enkel å designe. Den er også vanlig i bruk i applikasjoner der høyt kapillært pumpetrykk ikke er et krav. For eksempel i miljø med lav tyngdekraft.

### 4.1.2 Komposittveker

Komposittveker prøver å lage et høyt kapillærtrykk uten at dette skal gå på bekostning av permeabilitet. Dette kan gjøres ved å kombinere ulike vekedesign. Den enkleste typen er på samme måte som for homogene veker, en netting som er viklet rundt innerveggen av varmerøret. Men i dette tilfellet er det brukt to ulike nettinger med ulik porestørrelse. Ytterst mot rørveggen er flere lag med nettinger med stor poreradius brukt for å sørge for god permeabilitet. Innerst er det et nettinglag med liten porestørrelse installert for å øke kapillærtrykket.

Andre komposittveker benytter seg av arterier for å øke transporten av væske. Disse arteriene må være avstengt fra dampstrømmen for å hindre dampbobler i å trenge inn og forstyrre væskestrømmen. Aksielle spor i kombinasjon med en netting kan også benyttes. Nettingen er da lokalisert mellom sporene og dampstrømmen.

Et annet konsept innenfor komposittveker er bipolare veker. Her er ulike vekestrukturer benyttet i fordamper og resten av varmerøret. I fordamper benyttes gjerne en fin veke som genererer et høyt kapillærtrykk. I resten av varmerøret benyttes en grovere vekestruktur som tillater høy permeabilitet. Mwaba med flere [10] har foretatt en simulering av denne typen veker der et grovt kobbernett ble brukt i kondensator og adiabatisk del, og en sintret kobberveke ble brukt i fordamper. Mwaba sammenlignet denne veken med to veker som var bestod av kun kobbernett og kun sintret vekestruktur. Resultatet var at den bi-polare veken hadde en maksimal varmetransportkapasitet på  $37 W$  mot kobbernettveken med  $30 W$  og sintret vekestruktur med  $8 W$ . Den maksimale varmetransportkapasiteten var her basert på antagelsen om at kapillærtrykket var den øvre begrensende faktor. Temperaturdifferansen mellom fordamper og kondensator var også mye lavere for den bipolare vekestrukturen.

## 4.2 Anbefalte vekegeometrier med hensyn på alkalimetaller

Når alkalimetaller anvendes som arbeidsmedium er det viktig å bruke en vekestruktur som tåler den høye temperaturen samtidig som den klarer å opprettholde et tilstrekkelig kapillært pumpetrykk.

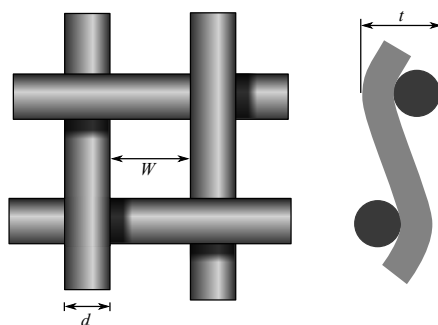
Dersom varmerøret skal operere i et miljø med tyngdekraft stiller dette store krav til det kapillære pumpetrykket. Dette er også viktig selv om fordamper og kondensator er i samme høyde da veken vil hjelpe til med å distribuere arbeidsmedium ut til alle sider av fordamperen. Sintrede veker av metallpulver har en relativt lav effektiv poreradius samtidig som den har en relativ god permeabilitet. Veker av denne typen egner seg derfor til bruk med alkalimetaller der en ønsker å transportere væske mot tyngdekraften. Veker av metallnetting genererer ikke like stort kapillært trykk, men har en større permeabilitet.

Det er valgt å fokusere på disse to vekestrukturene da disse er relativt enkle, velprøvde og egner seg godt sammen med alkalimetaller.

### 4.3 Permeabilitet $K$ for veker

Av Darcys lov 2.2, introdusert i kapittel 2.3, ser en at permeabiliteten  $K$  har stor betydning for trykkdifferansen i væskestrømmen  $\Delta P_l$ . For å kunne beregne en veks ytelse er det derfor viktig å kunne fastslå dette tallet på en nøyaktig måte. En vanskeliggjørende faktor er at diversiteten i vekedesign ofte fører til at det kreves en spesifikk modell for å kunne kalkulere permeabiliteten for en bestemt vekestruktur. Ofte må en ty til eksperimentelle data der nøyaktige analytiske modeller ikke eksisterer. Dette gjelder for veketyper som ikke består av årer eller kanaler. Eksempler er sintrede veker og nettingveker. For å finne en verdi for permeabilitet eksperimentelt vil en løse Darcys lov med hensyn på  $K$ . Mange slike eksperimenter er foretatt og verdier og semi-empiriske korrelasjoner er tilgjengelig for spesifikke veketyper.

#### 4.3.1 Permeabilitet for veker av metallnetting



Figur 4.1: Framstilling av nettingveke med geometriske mål.

For veker av metallnetting kan den effektive permeabiliteten estimeres ut fra følgende likning [7]

$$K = \frac{d^2 \varphi^3}{122(1 - \varphi)^2} \quad (4.1)$$

Porøsiteten  $\varphi$  kan her beregnes ut fra

$$\varphi = 1 - \frac{1.05\pi Nd}{4} \quad (4.2)$$

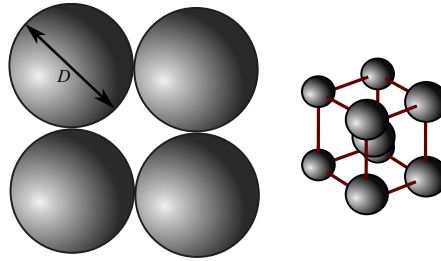
der  $N$  er antall åpninger per enhet lengde og er gitt ved  $1/(d + W)$  der  $d$  og  $W$  er geometriske mål for veken (se figur 4.1). Faghri [7] presiserer at eksperimentelle data for porøsitet er å foretrekke.

Likning 4.1 er gyldig for tettpakkede veker med tre eller flere lag. Likning 4.1 og 4.2 er gyldige for aksielle Reynoldstall  $Re_{l,h} < 10$ . Det aksielle Reynoldstallet i væskefasen er gitt ved

$$Re_{l,h} = D_h w_l / \nu_l \quad (4.3)$$

og hydraulisk diameter er gitt ved  $D_h = d\varphi/(1 - \varphi)$ .

### 4.3.2 Permeabilitet for sintrede veker med sfærisk partikler



Figur 4.2: Framstilling av sintret veke med geometriske mål og kubisk partikkelpakking (til høyre).

Permeabiliteten for løst pakkede sintrede veker av sfæriske partikler kan estimeres ut fra følgende likning[7]

$$K = \frac{D^2 \varphi^3}{150(1 - \varphi)^2} \quad (4.4)$$

Der  $D$  er diameter på partikler (se figur 4.2).

Likning 4.4 er etablert med hensyn på sintrede veker av metallpulver med diameter mellom  $50 \times 10^{-6} < D < 3 \times 10^{-4}$  og porøsitet mellom  $0,27 < \varphi < 0,66$

Porøsiteten  $\varphi$  kan settes til 0,48 dersom en antar kubisk pakkede partikler.

Likning 4.4 er gyldig for aksielle Reynoldstall  $Re_{l,h} < 10$  gitt av likning 4.3. Hydraulisk diameter er her  $D_h = 2D\varphi/(3(1 - \varphi))$ .

## 4.4 Effektiv poreradius $r_{eff}$ for veker

### 4.4.1 Effektiv poreradius for veker av metallnetting

For veker av metallnetting kan den effektive poreradiusen bestemmes ut i fra likning

$$r_{eff} = \frac{W + d}{2} \quad (4.5)$$

eller

$$r_{eff} = \frac{1}{2N} \quad (4.6)$$

#### 4.4.2 Effektiv poreradius for sintrede veker av sfæriske partikler

For sintrede veker av sfæriske partikler kan effektiv poreradius  $r_{eff}$  estimeres til å være  $0,21D$  der diameteren er  $40 \times 10^{-6} < D < 10^{-3}$  og porøsiteten er  $0,27 < \varphi < 0,66$ .

#### 4.5 Effektiv termisk konduktivitet $k_{eff}$

Varme som tilføres i fordampere og tas ut i kondensatoren blir transportert gjennom vekestrukturen. Dette er forutsett at arbeidsmediumet har en høy konduktivitet og fordampning av type 1 forekommer (se kapittel 3.2). På grunnlag av dette er det nødvendig å kunne fastslå konduktiviteten i veken. Den effektive konduktiviteten er avhengig av både konduktiviteten til vekematerialet og arbeidsmediumet.

Faghri [7] utleder en likning (likning 4.7) for konduktivitet gjennom et medium bestående av et fast materiale og et flytende materiale ved å anta at varmeovergangen skjer i parallell gjennom mediumene (det er ingen varmeutveksling mellom solid og flytende materiale) og at mediumene er isotropiske og at effekten av stråling, viskøs dissipasjon og trykkarbeid er neglisjerbare.

$$k_{eff} = (1 - \varphi)k_s + \varphi k_l \quad (4.7)$$

For et medium der varmeovergangen skjer i serie vil den effektive varmeovergangskoeffisienten bli

$$k_{eff} = \frac{k_l k_s}{\varphi k_s + k_l(1 - \varphi)} \quad (4.8)$$

der  $\varphi$  er porøsiteten til veken,  $k_s$  og  $k_l$  er henholdsvis konduktivitet for solid materiale og arbeidsmedium.

Likning 4.7 for parallell varmeovergang gir den høyeste varmeovergangskoeffisienten og 4.8 den minste. I reelle veker vil den effektive varmeovergangskoeffisienten  $k_{eff}$  ofte være en kombinasjon av disse to ytterkantene og dermed ligge imellom verdiene som disse likningene gir.

##### 4.5.1 Veker av metallnetting

Faghri gir en likning for den effektive termiske konduktiviteten  $k_{eff}$  gitt ved

$$k_{eff} = \frac{k_l[(K_l + k_s) - (1 - \varphi)(K_l - k_s)]}{[(K_l + k_s) + (1 - \varphi)(K_l - k_s)]} \quad (4.9)$$

For å kunne løse denne likningen må porøsiteten  $\varphi$  være kjent. Likning 4.10 gir et estimat for porøsiteten.

$$\varphi = 1 - \pi S N d / 4 \quad (4.10)$$

Her er  $S$  krimping faktoren. Denne faktoren tar hensyn til at fibre blir bøyd når de blir vevd og ikke bare krysser hverandre. For veker bestående av flere lag med metallnetting gir Faghri følgende likning for porøsitet

$$\varphi = 1 - \frac{n\delta_1(1 - \varphi_1)}{\delta_n} \quad (4.11)$$

der  $\delta_1$  er tykkelsen av et lag med metallnetting,  $\delta_n$  er tykkelsen av samtlige lag,  $n$  er antall lag og  $\varphi_1$  er porøsiteten av et lag.

Faghri [7] viser til eksperimenter der den kalkulererte termiske konduktiveten  $k_{eff}$  blir sammenlignet med den målte verdien. I disse eksperimentene gir likning 4.9 og 4.11 verdier som er  $\pm 10\%$  av den målte verdien.

#### 4.5.2 Sintrede veker av metallpulver

For sintredete veker av metallpulver er den effektive konduktiviteten gitt ved følgende likning[7]

$$k_{eff} = \frac{2 + k_l/k_s - 2\varphi(1 - k_l/k_s)}{2 + k_l/k_s - \varphi(1 - k_l/k_s)} \quad (4.12)$$

## Kapittel 5

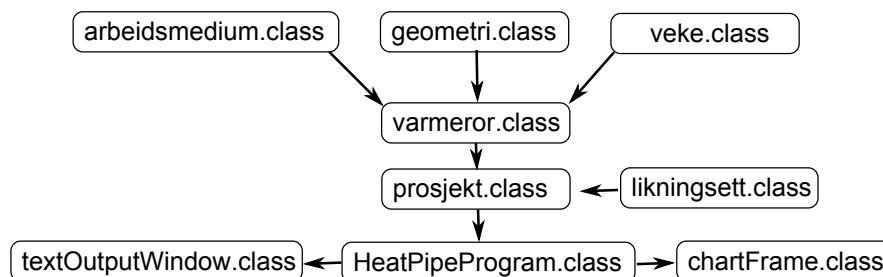
# Beregningsprogram for varmerør

### 5.1 Introduksjon

Med basis i de foregående kapitlene er det blitt utviklet et beregningsprogram for termisk ytelse av varmerør. Programmet skal kunne beregne ytelse og begrensende faktorer for et sirkulært varmerør med en fordamperdel, en kondensatordel og en adiabatisk del. Ytelsen for et varmerør kan betegnes som termisk varmeledningsevne ved en gitt varmeffluks og temperatur. I tillegg kan programmet blant annet beregne og vise damp- og væske-trykkskurver ved gitte betingelser.

Alle beregninger foretas ved stasjonære driftssituasjoner.

### 5.2 Struktur for programkode



Figur 5.1: Grafisk framstilling av programstruktur.

Programmet er skrevet i det objekt orienterte programmeringsspråket (OOP) Java (kompilatorversjon 1.6.0.13). Programmkoden er derfor delt inn i ulike abstrakte klasser. Når en klasse blir brukt i en annen klasse blir det ofte referert til som et objekt. Klassen *prosjekt.class* har to objekter, et *likningsett* og et *varmeror* (varmerør) objekt. Objektet *likningsett* inneholder alle likninger som blir brukt i programmet. Objektet *varmeror* består av et *veke* objekt, et *arbeidsmedium* objekt og et *geometri* objekt. Alle data for det aktuelle varmerøret blir lagret i disse klassene. Klassen *prosjekt* inneholder også



alle beregningsmetodene for beregninger i programmet. Ved å hente data fra *varmeror* objektet og resultat fra *likninger* objekter kan beregninger utføres og vises i som grafer eller tekst. På toppen av det hele ligger det en *HeatPipeProgram* klasse som håndterer brukergrensesnittet og dialogen med brukeren. Figur 5.1 viser en grafisk framstilling av noe av programstrukturen og hvordan noen av klassene forholder seg til hverandre. Tabell 5.1 oppsummerer alle klasser og beskriver hva de inneholder. I tillegg til disse

Klasse	Beskrivelse
<i>arbeidsmedium.class</i>	Inneholder alle egenskaper for arbeidsmediumet
<i>arbeidsmediumFane.class</i>	Bruker grensesnitt for arbeidsmedium egenskaper dialogboks
<i>begrensningsDialog.class</i>	Bruker grensesnitt for begrensning dialog ved beregning av begrensninger
<i>chartFrame.class</i>	Bruker grensesnitt for visning av grafer
<i>geometri.class</i>	Inneholder alle egenskaper for varmerørbeholder
<i>geometriFane.class</i>	Bruker grensesnitt for beholdergeometri dialogboks
<i>HeatPipeProgram.class</i>	Hovedklasse. Kobler brukergrensesnitt mot beregninger
<i>likningsett.class</i>	Inneholder alle likninger
<i>limitChartFrame.class</i>	Bruker grensesnitt for visning av begrensningsgraf
<i>propertiesFrame.class</i>	Bruker grensesnitt for egenskaper dialogboks
<i>prosjekt.class</i>	Alle beregningsmetoder og algoritmer
<i>punktberegningDialog.class</i>	Bruker grensesnitt for dialog ved enkeltberegning
<i>textOutputWindow.class</i>	Bruker grensesnitt for visning av tekstvindu
<i>varmeror.class</i>	Subklasse for arbeidsmedium, geometri og veke
<i>veke.class</i>	Inneholder alle egenskaper for veke
<i>vekeFane.class</i>	Bruker grensesnitt for veke egenskaper dialogboks

Tabell 5.1: Tabell over klasser med beskrivelse

klassene består programmet av en tredjeparts klassen JFreeChart for å vise grafer. Dette er *jfreechart.jar* og *jcommon.jar*.

### 5.3 Kjøre program

For å kunne kjøre programmet må maskinen ha Java Runtime Environment (JRE eller JVM) installert. Dette kan lastes ned fra [www.java.com](http://www.java.com). For å starte programmet i et Windowsmiljø kan filen *heatpipe.bat* kjøres. Eventuelt kan man kjøre programmet ved å skrive `java -classpath jcommon.jar;jfreechart.jar;; HeatPipeProgram` i kommandodialogen (gitt at en står i riktig katalog).

### 5.4 Foreta beregninger

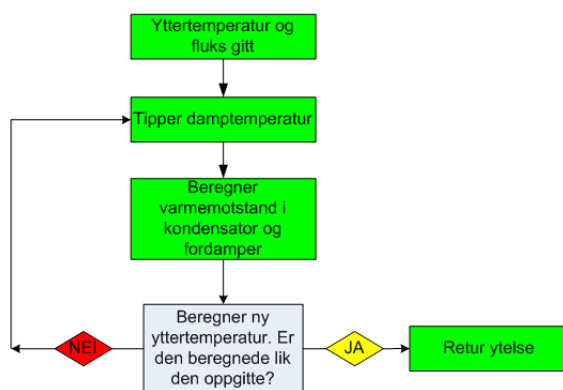
Når programmet starter blir det automatisk lastet inn geometrier for veke og beholder for et gitt varmerør (se appendiks A datalisting A.1 linje 37). Vann blir lastet inn som arbeidsmedium. For å endre på dette kan brukeren åpne *Rediger* → *Egenskaper* på menyen eller trykke ALT+E. Tabulerte data for kalium kan således hentes inn i program. Alle tabulerte data er hentet fra Faghri[7].

Ulike beregninger kan foretas ved å velge menyvalg *Beregn* eller trykke ALT+B. Bruker kan også lagre prosjektet sitt ved å velge *Fil* → *Lagre*. Prosjektet kan senere hentes fram igjen.

### 5.5 Egenskaper for arbeidsmedium

Egenskaper for arbeidsmedium er lagret i en tekstfil som kan importeres inn i program. Data for vann og kalium er hentet fra Faghri [7]. Programmet bruker linær interpolasjon og ekstrapolasjon for å finne verdien mellom tabulerte verdier eller utenfor tabulert område (se appendiks A datalisting A.6 linje 99).

### 5.6 Beregning av ytelse for varmerør



Figur 5.2: Flytskjema for ytelseberegning.

I programmet kan en beregne ytelsen til et varmerør dersom geometri, veke, arbeidsmedium og overflatetemperatur og varmefluks på fordampner er gitt.

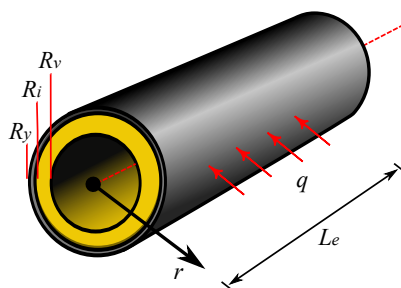
Varmen som blir tilført i fordampner blir ledet gjennom fordampnervegg og veke ved hjelp av konduksjon. *Fouriers lov* (likning 5.1) kan benyttes til å beregne varmetransporten gjennom sylinderveggen. Nettverksanalogien nærmere beskrevet i Cengel [4] benyttes for å estimere varmemotstanden  $R_{total}$ .  $\dot{Q}$  er da gitt ved

$$\dot{Q} = \frac{T_v - T_y}{R_{total}} \quad (5.1)$$

der  $T_v$  er temperatur ved veke damp grensesnitt,  $T_y$  er temperatur på yttervegg og  $R_{total}$  er den totale varmemotstanden. I fordampner og kondensator vil  $R_{total}$  være gitt ved

$$R_{total} = \frac{1}{h_\delta 2\pi R_v L} + \frac{\ln(R_i/R_v)}{2\pi L k_{eff}} + \frac{\ln(R_y/R_i)}{2\pi L k_{vegg}} \quad (5.2)$$

der  $R_y$ ,  $R_i$  og  $R_v$  er radiuser målsatt på figur 5.3.  $k_{eff}$  er her effektiv konduktivitet



Figur 5.3: Snitt gjennom fordampnerdel av et varmerør

for veke som er beskrevet i kapittel 4.3.

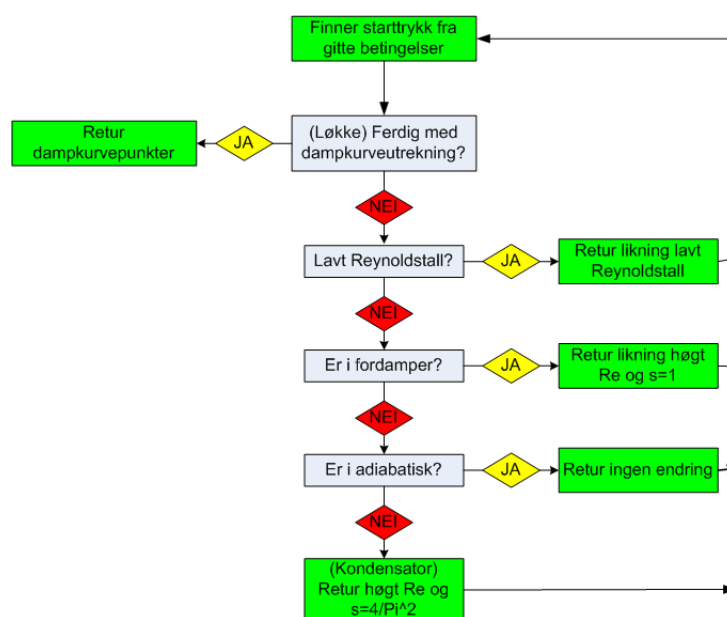
Varmeovergangstall i damp-væske grensesnitt  $h_\delta$  i fordampner og kondensator blir beregnet på grunnlag av likning 2.28 (se appendiks A datalisting A.2 linje 239). For å kunne beregne denne må en først beregne dampkurven ved den aktuelle varmefluksen. Deretter kan en beregne motstanden i fordampner og kondensator. Alle fluidegenskaper blir evaluert ved middeltemperatur. I fordampnerdel blir væskeegenskapene evaluert ved metningstemperaturen for væsken. Metningstemperatur er antatt til å være gitt ved gjennomsnittlig damptrykk i kondensator.

Programmet beregner det totale temperaturfallet i varmerøret ved å tippe en mettet dampstemperatur ved begynnelsen av fordampneren som er lavere enn den gitte ytterveggstemperaturen (se datalisting A.1 linje 591). Deretter genererer programmet  $h_\delta$  verdier for fordampner (linje 602). Den totale varmemotstanden  $R_{total}$  blir beregnet og en ytterveggstemperatur på fordampneren blir beregnet ut fra dette (linje 615). Dersom den beregnede verdien ikke samsvarer med den oppgitte, vil programmet tippe en ny verdi basert på differansen (linje 618).

For å beregne varmegjennomgangen i veke benyttes likninger oppgitt i kapittel 4.5 (se datalisting A.2 linje 246). Programmet kan derfor foreløpig bare håndtere veker av metallnetting og sintrede veker av metallpulver.

Programmet sjekker om kravet til inkompressibel strømning er oppfylt (likning 2.4) og advarer dersom det brukes en inkompressibel modell uten at kravet er oppfylt (datalisting A.1 linje 550). Programmet sjekker ikke om den oppgitte varmeffluksen og temperaturen i fordampere er mulig med henhold til termiske og hydrauliske begrensninger.

## 5.7 Generere dampkurve ved første orden lukket form tilnærming



Figur 5.4: Flytskjema for dampkurvealgoritme

Beregningsprogrammet implementerer en enkel modell for dampstrømning basert på *Første orden lukket form tilnærming* av Cotter som er beskrevet i kapittel 2.4.1. Likning 2.11 er brukt for å beregne det radielle Reynoldstallet som benyttes for å velge mellom likning 2.16 der  $|Re| < 1$  og 2.17 der  $|Re| > 1$  (se appendiks A datalisting A.2 linje 16 og linje 88).

For høye Reynoldstall varierer likningen om en er i fordampere, kondensator eller adiabatisk del fordi  $s$  er ulik. Programmet må derfor finne ut hvor i varmerøret det er kommet.

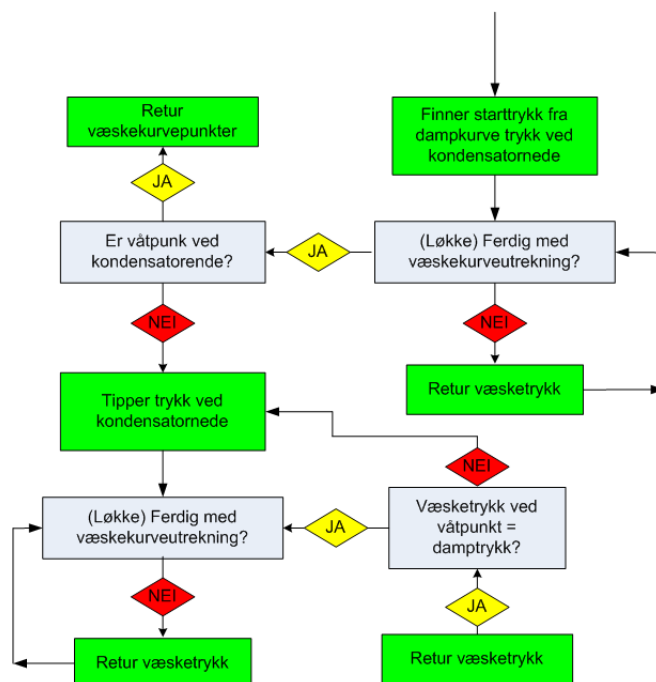
Det er verdt å merke seg at det for denne beregningsmodellen antas at strømmingen er laminær både i fordampere og kondensator og at varmeffluks i fordampere og kondensator er konstant og med motsatt fortegn. Det radielle Reynoldstallet vil derfor være konstant i både fordampere og kondensator.

Massestrømmen damp er gitt av likning 2.12 (se tillegg A datalisting A.2 linje 43).

Programmet beregner dampkurve ved en gitt fluks og metningstemperatur spesifisert

ved starten av fordamer. En løkkefunksjon bruker forlengs Euler for å løse likningen for damptrykk (se appendiks A datalisting A.1 linje 101). Det er også mulig å oppgi antall trinn som blir brukt i løsningen. Figur 5.4 viser et flytskjema for prosessen.

## 5.8 Generere væskekurve



Figur 5.5: Flytskjema for væskekurvealgoritme

For beregning av væsketrykk gjennom veke blir likning 2.2 brukt. Likningen blir løst med forlengs Euler, men med snudd koordinatsystem. Programmet regner seg fra kondensator mot fordamer.

Væsketrykket blir regnet ut i forhold til damptrykket. Det blir antatt at væske- og damptrykk samsvarer ved et punkt i kondensator (våtpunkt). Programmet antar først at våtpunktet er lokalisert ved enden av kondensatoren og regner ut væsketrykkurve for dette tilfellet (tillegg A datalisting A.1 linje 169). Deretter sjekker programmet om væsketrykket i kondensatoren er lavere enn damptrykkgjennvinningen (tillegg A datalisting A.2 linje 303). Dersom dette er tilfelle er antagelsen om våtpunkt ved kondensatormede feil. Programmet regner da ut en ny væsketrykkskurve ved å tippe starttrykk ved kondensatormede. Hvis trykket ved våtpunktet ikke samsvarer tipper programmet en ny startverdi på grunnlag av feilen fra forrige iterasjon. Iterasjonene fortsetter til programmet finner en løsning. Figur 5.5 viser et flytskjema for algoritmen.

## 5.9 Beregning av begrensninger

Ved beregning av begrensninger må bruker oppgi nedre og øvre varmekraft og temperatur i fordampere. Programmet vil deretter begynne med å teste for begrensninger mellom nedre og øvre varmekraft gjennom hele temperaturintervallet (se tillegg A datalisting A.1 linje 287). Resultatet vil bli vist som en graf. Programmet tester for totalt fire ulike begrensninger. Dette er kapillærbegrensning, viskøs begrensning, sonisk begrensning og kokebegrensning.

### 5.9.1 Kapillærbegrensning og viskøs begrensning

Kapillærbegrensninger blir beregnet ut fra likning 3.4 (tillegg A datalisting A.2 linje 141). Lokalisering av våtpunkt blir gjort på samme måte som ved generering av væskekurve. For å finne kapillærgrensen må programmet regne ut kapillærtrykket for alle flukser for en gitt temperatur fram til kapillærgrensen er nådd. Dette blir gjort i en løkkefunksjon (se tillegg A datalisting A.1 linje 287).

Før programmet tester for kapillærgrense sjekker programmet om væsketrykket ved begynnelse av fordampere er mindre enn null. Viskøs begrensning er nådd dersom væsketrykket er mindre enn null og ingen beregning av kapillærbegrensning blir da foretatt.

### 5.9.2 Sonisk begrensning

Varmekraft ved sonisk begrensning blir beregnet ut fra likning 3.7 (tillegg A datalisting A.2 linje 188). Forholdet  $k'$  beregnes ut fra  $k' = C_p / (C_p - R)$  (der  $R$  er gasskonstant for medium).

### 5.9.3 Kokebegrensning

Varmekraft ved kokebegrensning blir beregnet ut fra likning 3.1 og likning 3.2 (tillegg A datalisting A.2 linje 155 og linje 162). Programmet inneholder likning for å beregne  $R_b$  (likning 3.3), men programmet setter  $R_b$  til  $10^{-7} m$  som standard.

## 5.10 Beregning av radielt Reynoldstall

Programmet har en funksjon for å beregne radielt Reynoldstall i dampstrømning. Dette blir gjort ved å først beregne en damptrykkkurve for en gitt varmebelastning  $Q_e$  og fordampertemperatur. Deretter brukes likning 2.11 for å beregne Reynoldstallet for alle punkt i dampkurve.

## 5.11 Begrensninger og feilkilder i modell

Modellen beregner varmeovergangskoeffisienter ut fra aritmetisk middeltemperatur i fordampere og kondensator. Dette kan være en feilkilde.

Varmetransport aksielt gjennom veke og beholdervegg på grunnlag av konduksjon er ikke tatt hensyn til. I de fleste tilfeller vil denne varmeovergangen være neglisjerbar.

Programmet har ingen modell som tar hensyn til kompressibel dampstrømning. Dette er en stor feilkilde ved forhold som fører til høy hastighet på dampstrøm ( $Ma > 0.3$ ). Ved bruk av alkalimetaller ved lave trykk (relativt lave temperaturer) vil strømningshastigheten på dampen i mange tilfeller overstige kravet til inkompressibel strømning. Ved beregning av ytelse for en stasjonær driftssituasjon vil programmet advare når dampmodellen er ugyldig.

Programmet kan feile i beregninger når uventede verdier blir brukt. Programmet gir ingen advarsel hvis en verdi er urealistisk eller i feil format. Resultatene som framkommer kan derfor i noen situasjoner være feil.

De tabulerte dataene for arbeidsmediumene har store temperatursprang og inneholder få verdier. De fleste verdier som blir benyttet i beregningene er resultat av interpolasjoner mellom tabulerte data. Dette stiller store krav til nøyaktigheten av disse dataene og interpolasjonsrutinene. Figur 6.9 viser tydelige knekker ved verdier for  $900^\circ K$  og  $1000^\circ K$  for sonisk begrensning (grønn kurve). Disse hjørnene framkommer av unøyaktigheter i tabulerte data.

Ved beregning av begrensingskurve og minste radius bruker programmet veldig lang tid (mange minutter). Dette kommer av trege algoritmer blant annet for interpolasjon. Programmet vil ikke respondere under beregning men i de fleste tilfeller vil det vises et timeglassikon.

# Kapittel 6

## Utførte beregninger

### 6.1 Introduksjon

To beregninger er blitt utført ved hjelp av Beregningsprogrammet.

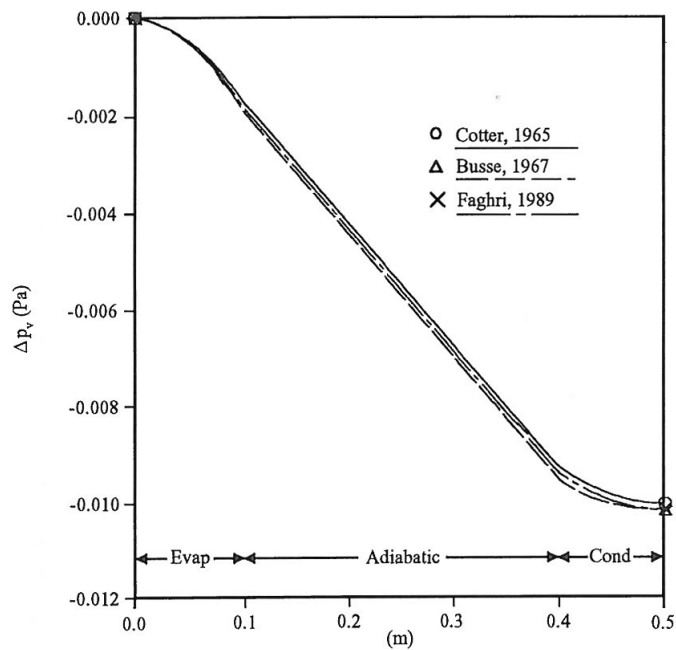
### 6.2 Beregninger med numerisk dampmodell av første orden

Som et ledd i oppgaven skal beregninger utføres i programmet. Det er blitt foretatt beregninger med dampmodell basert på *Første orden lukket form tilnærming* opp i mot beregninger foretatt av Faghri [7]. Dette for å verifisere at beregningsprogrammet vil produsere de samme resultater som Faghri har kommet fram til i sine utregninger.

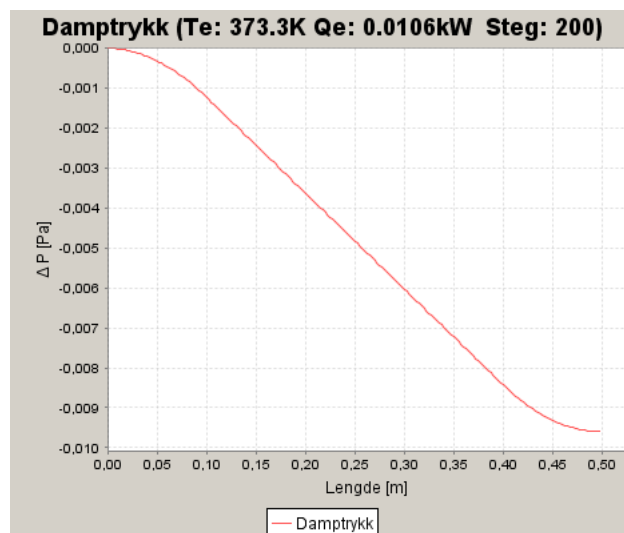
Faghri baserer sine beregninger på et varmerør med indre radius  $Ri = 0,01\text{ m}$ , fordampertilengde  $Le = 0,1\text{ m}$ , kondensatorlengde  $Lk = 0,1\text{ m}$  og adiabatisk lengde  $La = 0,3\text{ m}$ . Arbeidsmediumet i varmerøret er vann som opererer ved en metningstemperatur på  $100^\circ\text{C}$ . Varmebelastningen ble variert slik at resultatet var tre ulike radielle Reynoldstallet på 0,6, 2,4 og 1000. I beregningsprogrammet er det en funksjon for å plote en kurve med Reynoldstallet for dampen igjennom varmerøret (*Beregn  $\rightarrow$  Radielt Reynoldsnummer*). Denne funksjonen ble brukt for å finne varmebelastning som korresponderer til Reynoldstall. Figur 6.1 viser Faghri sitt resultat for ulike modeller for et Radielt Reynoldstall på  $Re_r = 0,6$ . Tilsvarende figur fra beregningsprogram er framstilt i figur 6.2. For å oppnå det overnevnte Reynoldstallet måtte en fluks på  $0,0106\text{ kW}$  tilføres i fordamperdelen av varmerøret. Figur 6.3 og figur 6.4 sammenligner resultat ved et radielt Reynoldstall  $Re_r = 2,4$  som tilsvarer en fluks på  $0,042\text{ kW}$ . Figur 6.5 viser sammenligning av Faghri sine resultat ved et Reynoldstall på  $Re_r = 1000$ . Dette tilsvarer en fluks i fordampere på  $18\text{ kW}$ . Figur 6.6 viser resultatet fra beregningsprogrammet. Dette er som Faghri påpeker en så høy fluks at det i et virkelig tilfelle ville oppstå tørrkoking i fordampere.

I tilfellet for et radielt Reynoldstall  $Re_r = 2,4$  (figur 6.3) viser Faghri sin graf en konveks kurve for Cotter sin modell i kondensatordelen av varmerøret. Dette er ikke i

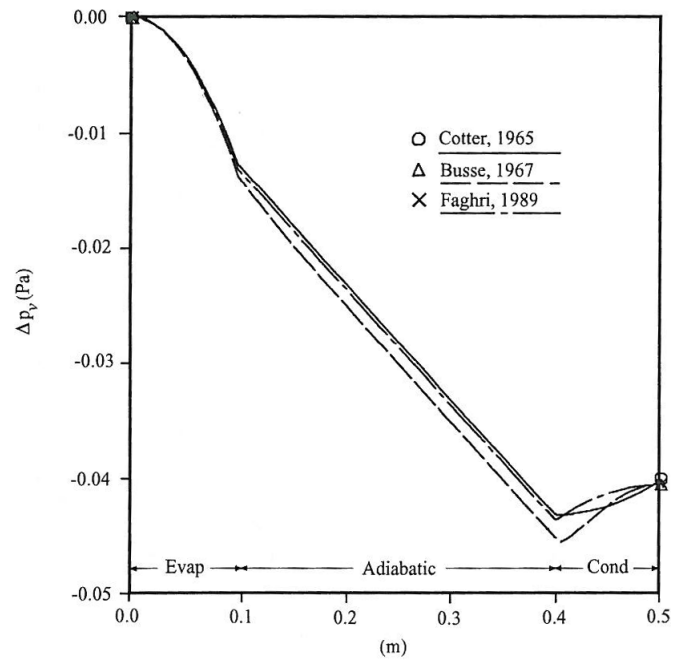




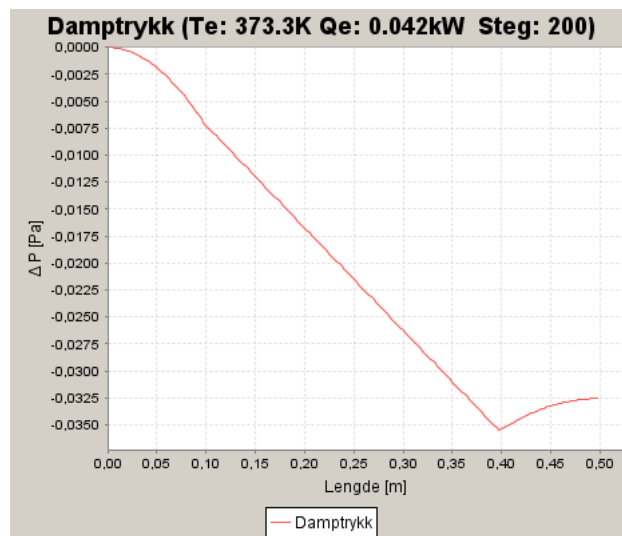
Figur 6.1: Sammenligning av dampstrømningsmodeller fra Faghri [7] for  $Re_r = 0,6$ .



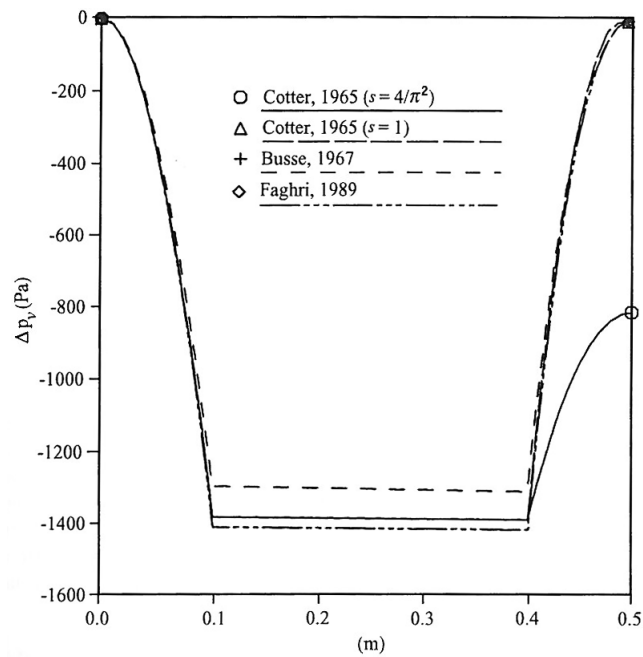
Figur 6.2: Resultat av dampstrømningsberegning med Cotters endimensjonale modell fra beregningsprogram for  $Re_r = 0,6$ .



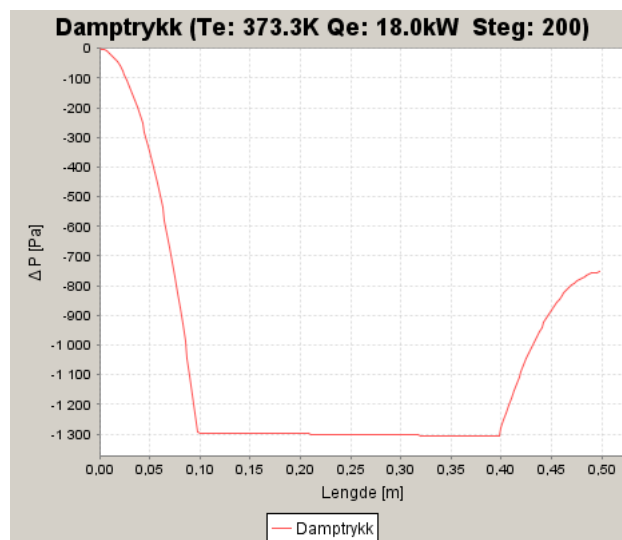
Figur 6.3: Sammenligning av dampstrømningsmodeller fra Faghri [7] for  $Re_r = 2, 4$ .



Figur 6.4: Resultat av dampstrømningsberegning med Cotters endimensjonale modell fra beregningsprogram for  $Re_r = 2, 4$ .



Figur 6.5: Sammenligning av dampstrømningsmodeller fra Faghri [7] for  $Re_r = 1000$



Figur 6.6: Resultat av dampstrømningsberegning med Cotters endimensjonale modell fra beregningsprogram for  $Re_r = 1000$

overensstemmelse med resultatet som beregningsprogrammet gir (figur 6.4). Beregningsprogrammet gir en konkav kurve i kondensator. Ut fra teorien er det også dette resultatet som forventes. Det er derfor uvisst hvorfor Faghri ikke får dette resultatet. Videre gir resultatene fra beregningsprogrammet et lavere trykkfall enn det Faghri beregner i alle tilfeller. En forklaring på dette kan være at beregningsprogrammet beregner fysikalske data for alle damptrykk i motsetning til Faghri som trolig benytter seg av konstante verdier. Ut over dette stemmer resultatet fra beregningsprogrammet godt med Faghri sine utregninger.

### 6.3 Beregninger av minste diameter for varmerør

I samarbeid med instituttet er det blitt framlagt et problem som skal løses ved hjelp av programmet. Det skal beregnes en minste diameter for et varmerør med spesifikk fordampfluks  $q_e$  mellom  $20\,000\text{ W/m}^2$  og  $300\,000\text{ W/m}^2$ , arbeidstemperatur fra  $400^\circ\text{C}$  til  $600^\circ\text{C}$ , kalium som arbeidsmedium,  $30\text{ cm}$  lang adiabatisk del og  $10\text{ cm}$  lang fordamp og kondensator. Som veke skal en *Inco 225 nickel* veke benyttes. Verdier for permabilitet, effektiv poreradius og porøsitet har blitt beregnet til å være  $K = 3,4\mu\text{m}^2$ ,  $r_{eff} = 11,9\mu\text{m}$  og  $\varphi = 0,9$  [8].

Metanol ble brukt som væske under målingene og kontaktvinkelen for hydrokarboner er derfor bakt inn i uttrykket  $r_{eff}$ . Kontaktvinkel for hydrokarboner er omtrent  $30^\circ$  og for kalium omtrent  $26^\circ$  [13]. For å kompensere for ulik kontaktvinkel blir  $r_{eff}$  med hensyn på kalium gitt ved  $r_{eff} \cos(26)/\cos(30)$ . Effektiv poreradius blir da  $1,03r_{eff}$ .

Varmerøret skal operere mot tyngdekraften. Fordamp er dermed øverst i varmerøret og arbeidsmedium må bli pumpet opp til fordamp ved hjelp av kapillærkrefter.

#### 6.3.1 Framgangsmåte

For å forenkle prosessen ble en metode for å finne laveste dampstrømningsradius  $R_v$  skrevet inn i programmet (se appendiks A datalisting A.1 linje 668). Metoden er i stor grad en resirkulering av koden for å beregne begrensninger.

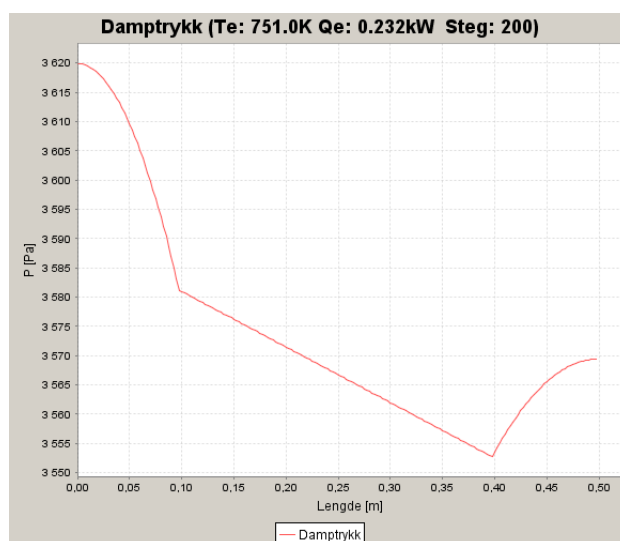
Programmet viser først en dialogboks der brukeren skriver inn varmeffluks  $q_e$  og dampstemperatur i start av fordamp (dialogboksen er den samme som er brukt for andre beregninger og teksten i dialogboksen er derfor feil men spesifikk fluks går inn i første, dampstemperatur ved start av fordamp i andre og antall steg i Euler metode i tredje tekstboks). Programmet reduserer dampradiusen  $R_v$  til kapillærbegrensning, viskøs begrensning, sonisk begrensning eller kokebegrensning blir møtt. Deretter skriver programmet ut minste radius  $R_v$  ved dette punktet.

Programmet bruker radiusen  $R_v$  på *Egenskaper* menyen som startverdi for beregning av minimum radius. Programmet feiler derfor i å beregne minste radius dersom  $R_v$  allerede før beregning er lavere enn minimumsradiusen. Det lønner seg derfor å sette inn relativt høye verdier for  $R_v$  og  $R_i$  i *Egenskaper* -menyen i forkant av beregning.

I beregningen er differansen mellom dampradius  $R_v$  og ytre radius av veke  $R_i$  (dette blir veketykkelse) konstant med  $0,01\text{ m}$ .

### 6.3.2 Resultat

Ved  $q_e = 300\,000\text{ kW/m}^2$  og  $T_v = 673^\circ\text{K}$  vil damptrykket være så lavt at væsketrykk vil bli negativt. Dette er ikke fysisk mulig. Ved å prøve forskjellige verdier i *Lag kombiner væske/damp kurve* vil det vise seg at damptemperatur i fordampner må være minimum  $T_v = 751^\circ\text{K}$  (Trykk er i stor grad avhengig av temperatur og svært lite avhengig av fluks). Selv ved denne temperaturen er trykket i varmerøret ekstremt lavt. Damptrykk i fordampner vil begynne på under  $3\,700\text{ Pa}$  (se figur 6.7). I en reel applikasjon ville det vært fornuftig å velge et annet arbeidsmedium med høyere damptrykk ved denne temperaturen. En del beregninger er blitt foretatt og resultatet er oppsummert i tabell 6.1



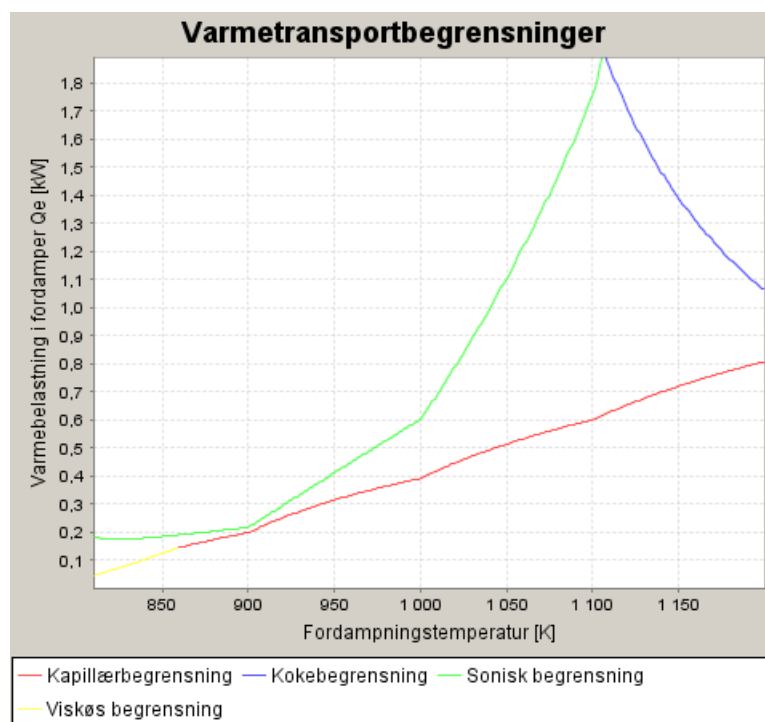
Figur 6.7: Damptrykk gjennom varmerør ved dampstrømsradius  $R_v = 0,0065\text{ m}$ .

Fluks $q_e$	$Q_e$	Damptemp. $T_e$	Min. radius $R_v$	Begrensning
$300\text{ kW/m}^2$	$8,297\text{ kW}$	$751^\circ\text{K}$	$0,032\text{ m}$	Viskøs
$20\text{ kW/m}^2$	$0,232\text{ kW}$	$751^\circ\text{K}$	$0,0065\text{ m}$	Viskøs
$300\text{ kW/m}^2$	$3,1\text{ kW}$	$873^\circ\text{K}$	$0,0044\text{ m}$	Sonisk (figur 6.9)
$20\text{ kW/m}^2$	$0,164\text{ kW}$	$873^\circ\text{K}$	$0,0011\text{ m}$	Kapillær (figur 6.8)

Tabell 6.1: Beregninger av minimum dampradius  $R_v$

### 6.3.3 Diskusjon av resultat

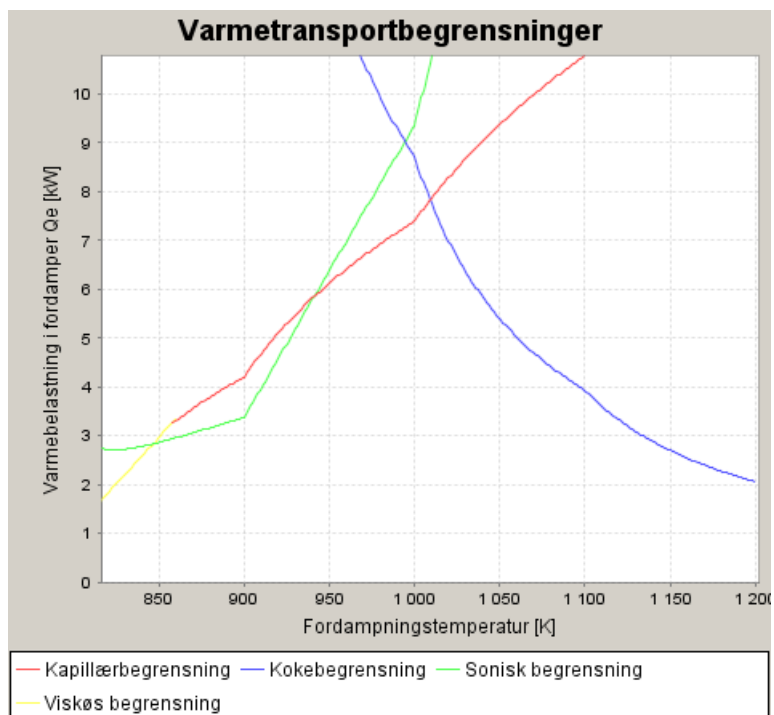
Det er fysisk umulig å benytte det spesifiserte varmerøret ved  $673^\circ\text{K}$ . Dette fordi væsketrykket i fordampneren i dette tilfellet må være lavere enn null. Fra  $751^\circ\text{K}$  og oppover er det mulig å nytte varmerøret ved de gitte betingelsene. Ikke uventet viser resultatet at en større diameter tillater en større total varmetransport  $Q_e$  i fordampner. Resultatene



Figur 6.8: Begrensningskurve ved dampstrømningsradius  $R_v = 0,0011 \text{ m}$ .

viser også at ved lavere radius  $R_v$  er det kapillærtrykket som begrenser varmetransporten (se figur 6.8). Ved større radius  $R_v$  kan andre begrensninger spille en større rolle (se figur 6.9)

Begrensningskurvene figur 6.8 og 6.9 viser tydelige likheter med figur 3.1. Selv om sistnevnte er en illustrasjon avtegnet fra Faghri [7] gir dette en indikasjon på at programmet sine beregninger er riktige. Begrensningskurvene er også i henhold til teori framlagt i kapittel 3. Blant annet tydeliggjør begge figurer at viskøse begrensninger (gul kurve) er framtrædende ved lave temperaturer og varmeflukser. Kokebegrensningen er en begrensende faktor for høye temperaturer i figur 6.9 men som teorien tilsier er det den kapillære pumpeeffekten som er den største begrensningen i figur 6.8.



Figur 6.9: Begrensningskurve ved dampstrømningsradius  $R_v = 0,0044 \text{ m}$ .

## Kapittel 7

# Konklusjon

Ytelsen til et varmerør kan defineres som varmetransportkapasiteten fra fordamper til kondensator. Denne ytelsen er en konsekvens av en rekke fysiske fenomen. Damp blir generert i fordamper og strømmer til kondensator der den latente varmen frigis under kondensasjon. Deretter strømmer væske tilbake til fordamper gjennom en vekestruktur.

Fysiske fenomener legger en øvre begrensning på ytelsen til et varmerør. Der alkalimetaller benyttes som arbeidsmedium under stasjonære forhold er de viktigste begrensningene kapillærbegrensning, viskøs begrensning, sonisk begrensning og kokebegrensning. Ved transiente forhold vil andre begrensninger også påvirke ytelsen.

Veker av metallnetting og sintrede veker av metallpulver er anbefalt brukt i applikasjoner med alkalimetaller som arbeidsmedium som arbeider mot tyngdekraften. Dette på grunn av robustheten, permeabiliteten og de kapillære pumpeegenskapene.

Modellen som er utviklet innehar en rekke begrensninger og forenklinger. Blant annet er varmetransport aksielt gjennom veke og beholdervegg på grunnlag av konduksjon neglisjert. Dette er i de fleste tilfeller akseptabelt.

Modellen bruker *Første orden lukket form tilnærming* bassert på Cotter [2] for beregning av trykk i dampstrømm. Denne modellen tar ikke hensyn til kompressibilitet. Dette er en stor feilkilde ved forhold som fører til høy hastighet på dampstrøm ( $Ma > 0.3$ ). Ved bruk av alkalimetaller ved lave trykk (relativt lave temperaturer) vil strømningshastigheten på dampen i mange tilfeller overstige kravet til inkompressibel strømming. Ved beregning av ytelse for en stasjonær driftssituasjon vil programmet advare når dampmodellen er ugyldig. Teorien for en kompressibel dampstrømningsmodell er presentert i kapittel 2.4.3. Denne tar også hensyn til likevekten mellom dimer og monomer i gassen. Denne modellen bør implementeres i eventuelle videre arbeid.

Modellen benytter seg av tabulerte data for egenskaper til arbeidsmediumene. Tabulerte data for vann og kalium er vedlagt modell. Nøyaktigheten til tabulerte data spiller en rolle for resultatene til programmet. For tabulerte data for kalium er det bare ti målepunkt i temperaturintervallet fra  $600^{\circ}K$  til  $1500^{\circ}K$ ). Denne spredningen i data fører til unøyaktigheter. Som et forslag til videre arbeid er det derfor anbefalt å anskaffe mer detaljerte data.

Programmet ble brukt til å utføre to beregninger. Den første beregningen forsøker



ved hjelp av programmet å reprodusere resultater for tre ulike stasjonære driftssituasjoner presentert av Faghri [7]. Resultatene fra beregningsprogrammet gir et noe lavere trykkfall enn det Faghri beregner i alle tilfeller. Ut over dette stemmer resultatet fra beregningsprogrammet godt med Faghri sine utregninger.

I den andre beregningen ble minste radius for et varmerør med spesifikk fordampfluks  $q_e$  mellom  $20\,000\text{ W/m}^2$  og  $300\,000\text{ W/m}^2$  og arbeidstemperatur fra  $673^\circ\text{K}$  til  $873^\circ\text{K}$  beregnet.

Resultatet viste at det er fysisk umulig å benytte det spesifiserte varmerøret ved  $673^\circ\text{K}$ . Dette fordi væsketrykket i fordamperen i dette tilfellet må være lavere enn null. Fra  $751^\circ\text{K}$  og oppover er det mulig å nytte varmerøret ved de gitte betingelsene.

Videre viser resultatet att en større diameter tillater en større total varmetransport  $Q_e$  i fordampere. Resultatene viser også at ved lavere radius  $R_v$  er det kapillærtrykket som begrenser varmetransporten og ved større radius  $R_v$  kan andre fenomener spille en større rolle. Resultatene stemmer godt overens med teoriene og dette indikerer at modellen er korrekt.

## Kapittel 8

# Forslag til videre arbeid

Dampmodellen som er brukt i programmet tar ikke hensyn til kompressibilitet. Tower og Hainley[14] sin modell (kapittel 2.4.3) bør implementeres i program.

Programkoden bærer preg av å være skrevet for å virke, ikke for å være rask. Optimalisering av programkode, og da særlig den som brukes ved beregning av begrensninger, bør prioriteres.

Sammenligninger av resultat fra program opp mot reelle data bør foretas for å verifisere at modellen er feilfri. Eventuelt bør feil lokaliseres og utbedres.

Mer detaljerte tabeller for kaliumegenskaper bør anskaffes og implementeres.

# Bibliografi

- [1] C. A. Busse. Pressure drop in the vapor phase of long heat pipes. *IEEE Conf. of Thermionic Conversion Specialists*, 1967.
- [2] T. P. Cotter. Theory of heat pipes. Technical report, Los Alamos Scientific Lab, Univ. of California, 1965.
- [3] P. D. Dunn and D. A. Reay. *Heat Pipes*. Pergamon, 1994.
- [4] Yunus A. Çengel. *Heat and mass transfer : a practical approach*. McGraw-Hill, 2007.
- [5] C. T. Ewing, J. P. Stone, J. R. Spann, and R. R. Miller. Molecular association in sodium, potassium, and cesium. *The Journal of Physical Chemistry*, 3, 1967.
- [6] A. Faghri and M. M Chen. Numerical analysis of the effects of conjugate heat transfer, vapor compressibility and viscous dissipation in heat pipes. *Numerical heat Transfer*, 16, 1989.
- [7] Amir Faghri. *Heat pipe Science and Technology*. Taylor & Francis, 1995.
- [8] Brian Holley and Amir Faghri. Permeability and effective pore radius measurements for heat pipe and fuel cell applications. *Applied Thermal Engineering*, 26, 2005.
- [9] Jacob N. Israelachvili. *Intermolecular and Surface Forces*. Academic Press Inc., 1991.
- [10] Xiao Huang Misheck G Mwaba and Junjie Gu. Influence of wick characteristics on heat pipe performance. *International Journal of Energy Research*, 30, 2006.
- [11] Carey V. P. *Liquid Vapor Phase Change Phenomena An Introduction to the Thermophysics of Vaporization and Condensation Processes in Heat Transfer Equipment*. Washington: Hemisphere Publ, 1992.
- [12] G. P. Peterson and B. K. Bage. Entrainment Limitations in Thermosyphons and Heat Pipes. *Journal of Energy Resources and Technology*, 113:7, September 1991.
- [13] Lloyd A. Quarterman and William L. Primak. The capillary rise, contact angle and surface tension of potassium. *J. Am. Chem. Soc*, 7, 1950.

- [14] L. K. Tower and D. C. Hainley. An improved algorithm for the modeling of vapor flow in heat pipes. *NASA STI/Recon Technical Report N*, 90:13748–+, December 1989.

# Tillegg A

## Programlisting

Følgende er et utdrag av de viktigste programlistingene i beregningsprogrammet.

Listing A.1: prosjekt.java

```
1 import java.io.File;
2 import java.io.FileInputStream;
3 import java.io.IOException;
4 import java.io.ObjectInputStream;
5 import java.math.BigDecimal;
6
7 import javax.swing.JFileChooser;
8 import javax.swing.JOptionPane;
9
10 public class prosjekt {
11     varmeror varmeror;
12     likningsett likningsett;
13     chartFrame testframe;
14
15     double surfacetension_water = 0.007564; // sigma
16
17     // Begrensninger ved simulering av varmerørsbegrensninger
18     double Qmin, Qmax, Tmin, Tmax;
19     double capLimitKurve[], boilingLimitKurve[], sonicLimitKurve[],
20         viciousLimitKurve[];
21
22     double h_kondensasjonfilm = 0.0;
23     double hDelta_kondensator = 0.0;
24     double hDelta_fordamper = 0.0;
25
26     public prosjekt() {
27         varmeror = new varmeror();
28         likningsett = new likningsett();
29         Qmin = 0.0;
30         Qmax = 0.0;
31         Tmin = 0.0;
32         Tmax = 0.0;
33         // Dette er for testformål
34     }
35
36     //
37     public void initDummyVariables() {
38
39         File file = new File("beregning1_vann.hpf");
40         FileInputStream fis = null;
41         ObjectInputStream in = null;
42         try {
43             fis = new FileInputStream(file);
44             in = new ObjectInputStream(fis);
45             varmeror = (varmeror) in.readObject();
46             in.close();
47             JOptionPane.showMessageDialog(null,
48                 "beregning1_vann.hpf er lastet inn i program!");
49         } catch (IOException ex) {
50             JOptionPane.showMessageDialog(null,
51                 "Feil ved åpning av fil: beregning1_vann.hpf", "Feil",
52                 JOptionPane.ERROR_MESSAGE);
53         }
```

```

54         ex.printStackTrace();
55     }
56     } catch (ClassNotFoundException ex) {
57         JOptionPane.showMessageDialog(null,
58             "Feil ved åpning av fil: beregning1_vann.hpf", "Feil",
59             JOptionPane.ERROR_MESSAGE);
60         ex.printStackTrace();
61     }
62 }
63 }
64
65 public double[] ReLagkurve(double Qe, int steps, double temperatur_fordamper) {
66     double[] dampkurve = lagDampkurve(Qe, steps, temperatur_fordamper);
67     double[] ReKurve = new double[steps];
68
69     double zincrease = (varmeror.geometri.lengdeFordamper
70         + varmeror.geometri.lengdeAdiabatisk + varmeror.geometri.lengdeKondensator)
71         / steps;
72     double zvalue = zincrease;
73
74     for (int z = 1; z < steps; z++) {
75         // System.out.println("u_v "
76         // +varmeror.arbeidsmedium.getProperty(punkter[z-1],
77         // arbeidsmedium.PRESSURE, arbeidsmedium.VISKOSITET_VAPOR));
78         ReKurve[z] = likningssett.Re_radiell(zvalue, varmeror.arbeidsmedium
79             .getProperty(dampkurve[z], arbeidsmedium.PRESSURE,
80                 arbeidsmedium.VISKOSITET_VAPOR),
81                 varmeror.geometri.lengdeFordamper,
82                 varmeror.geometri.lengdeAdiabatisk,
83                 varmeror.geometri.lengdeKondensator, Qe,
84                 varmeror.arbeidsmedium.getProperty(dampkurve[z],
85                     arbeidsmedium.PRESSURE, arbeidsmedium.HFG));
86
87         zvalue = zvalue + zincrease;
88     }
89     return ReKurve;
90 }
91
92 public chartFrame getRekurve(double punkter[], double stepsize,
93     double temp, double fluks) {
94     chartFrame dampkurve = new chartFrame(
95         "Radial Reynoldstall gjennom varmerør", "Lengde [m]",
96         "Reynoldstall", punkter, "Reynoldstall", stepsize, // u0394=delta
97         temp, fluks);
98     return dampkurve;
99 }
100
101 public double[] lagDampkurve(double Qe, int steps, //
102     double temperatur_fordamper) {
103
104     double[] punkter = new double[steps];
105     double[] punkter_korrigert_differanse = new double[steps];
106
107     double zincrease = (varmeror.geometri.lengdeFordamper
108         + varmeror.geometri.lengdeAdiabatisk + varmeror.geometri.lengdeKondensator)
109         / steps;
110     double zvalue = zincrease;
111
112     punkter[0] = varmeror.arbeidsmedium.getProperty(temperatur_fordamper,
113         arbeidsmedium.TEMPERATURE, arbeidsmedium.PRESSURE); // Initialverdi
114
115     for (int z = 1; z < steps; z++) {
116         // System.out.println("u_v "
117         // +varmeror.arbeidsmedium.getProperty(punkter[z-1],
118         // arbeidsmedium.PRESSURE, arbeidsmedium.VISKOSITET_VAPOR));
119         punkter[z] = likningssett
120             .Pv_by_simplified_closed_form_approximation_euler(
121                 punkter[z - 1], zincrease, zvalue,
122                 varmeror.arbeidsmedium.getProperty(punkter[z - 1],
123                     arbeidsmedium.PRESSURE,
124                     arbeidsmedium.VISKOSITET_VAPOR),
125                 varmeror.geometri.lengdeFordamper,
126                 varmeror.geometri.lengdeAdiabatisk,
127                 varmeror.geometri.lengdeKondensator, Qe,
128                 varmeror.arbeidsmedium.getProperty(punkter[z - 1],
129                     arbeidsmedium.PRESSURE, arbeidsmedium.HFG),
130                 varmeror.arbeidsmedium.getProperty(punkter[z - 1],
131                     arbeidsmedium.PRESSURE,
132                     arbeidsmedium.TETHET_VAPOR),
133                 varmeror.geometri.indreRadius);
134         // System.out.println("Punkt " + z + " = " + punkter[z]);
135         zvalue = zvalue + zincrease;
136     }

```

```

137     punkter_korrigert_differanse[z] = punkter[z] - punkter[0];
138 }
139 return punkter;
140 }
141
142 public chartFrame getDampkurve(double punkter[], double stepsize,
143     double temp, double fluks) {
144     chartFrame dampkurve = new chartFrame("Damptrykk gjennom varmerør",
145         "Lengde [m]", "P [Pa]", punkter, "Damptrykk", stepsize, // u0394=delta
146         temp, fluks);
147     return dampkurve;
148 }
149
150 public double[] lagVeskekurve(double Qe, int steps,
151     double temperatur_fordamper) {
152
153     double[] punkter = new double[steps];
154     double[] dampkurve = lagDampkurve(Qe, steps, temperatur_fordamper);
155
156     double[] hydrostatiskTrykk = new double[steps];
157
158     double[] reeltTrykk = new double[steps];
159
160     double zincrease = (varmeror.geometri.lengdeFordamper
161         + varmeror.geometri.lengdeAdiabatisk + varmeror.geometri.lengdeKondensator)
162         / steps;
163     double zvalue = zincrease;
164
165     // lager først en vanlig dampkurve med wetpoint i enden og sjekker om
166     // dette er rett
167     punkter[steps - 1] = dampkurve[steps - 1];
168
169     for (int z = steps - 2; z >= 0; z--) { //
170
171         punkter[z] = likningssett.Pl(punkter[z + 1], zincrease, zvalue,
172             varmeror.arbeidsmedium.getProperty(punkter[z + 1],
173                 arbeidsmedium.PRESSURE,
174                 arbeidsmedium.VISKOSITET_LIQUID),
175             varmeror.arbeidsmedium.getProperty(punkter[z + 1],
176                 arbeidsmedium.PRESSURE,
177                 arbeidsmedium.TETHET_LIQUID), Math.PI
178             * (Math.pow(varmeror.geometri.ytreRadius, 2) - Math
179                 .pow(varmeror.geometri.indreRadius, 2)),
180             varmeror.veke.permeabilitet,
181             varmeror.geometri.lengdeFordamper,
182             varmeror.geometri.lengdeAdiabatisk,
183             varmeror.geometri.lengdeKondensator, Qe,
184             varmeror.arbeidsmedium.getProperty(punkter[z + 1],
185                 arbeidsmedium.PRESSURE, arbeidsmedium.HFG));
186
187         hydrostatiskTrykk[z] = varmeror.arbeidsmedium.getProperty(
188             punkter[z + 1], arbeidsmedium.PRESSURE,
189             arbeidsmedium.TETHET_LIQUID)
190             * Math.sin(Math.toRadians(varmeror.geometri.tilt))
191             * -9.81
192             * zvalue;
193         zvalue = zvalue + zincrease;
194
195         reeltTrykk[z] = punkter[z] + hydrostatiskTrykk[z];
196     }
197
198     reeltTrykk[steps - 1] = punkter[steps - 1]
199         + hydrostatiskTrykk[steps - 1];
200
201     int wetPointPosition = likningssett.getWetpoint(dampkurve, reeltTrykk,
202         varmeror.geometri.lengdeKondensator,
203         varmeror.geometri.lengdeAdiabatisk,
204         varmeror.geometri.lengdeFordamper);
205
206     if (wetPointPosition < (dampkurve.length - 1)) {
207         punkter[wetPointPosition] = (dampkurve[dampkurve.length - 1] - 1000.0);
208         double guess_starttrykk = dampkurve[dampkurve.length - 1];
209
210         int endlessloopbrak = 0;
211
212         while (Math.abs(reeltTrykk[wetPointPosition]
213             - dampkurve[wetPointPosition]) > 0.0001) {
214             zvalue = zincrease;
215
216             guess_starttrykk += dampkurve[wetPointPosition]
217                 - reeltTrykk[wetPointPosition];
218
219             punkter[steps - 1] = guess_starttrykk;

```

```

220
221     for (int z = steps - 2; z >= 0; z--) {
222
223         punkter[z] = likningssett.Pl(punkter[z + 1], zincrease,
224             zvalue, varmeror.arbeidsmedium.getProperty(
225                 punkter[z + 1], arbeidsmedium.PRESSURE,
226                 arbeidsmedium.VISKOSITET_LIQUID),
227             varmeror.arbeidsmedium.getProperty(punkter[z + 1],
228                 arbeidsmedium.PRESSURE,
229                 arbeidsmedium.TETHET_LIQUID), Math.PI
230                 * (Math
231                     .pow(varmeror.geometri.ytreRadius,
232                         2) - Math.pow(
233                             varmeror.geometri.indreRadius, 2)),
234             varmeror.veke.permeabilitet,
235             varmeror.geometri.lengdeFordamper,
236             varmeror.geometri.lengdeAdiabatisk,
237             varmeror.geometri.lengdeKondensator, Qe,
238             varmeror.arbeidsmedium.getProperty(punkter[z + 1],
239                 arbeidsmedium.PRESSURE, arbeidsmedium.HFG));
240
241         hydrostatiskTrykk[z] = varmeror.arbeidsmedium.getProperty(
242             punkter[z + 1], arbeidsmedium.PRESSURE,
243             arbeidsmedium.TETHET_LIQUID)
244             * Math.sin(Math.toRadians(varmeror.geometri.tilt))
245             * -9.81 * zvalue;
246         zvalue = zvalue + zincrease;
247
248         reeltTrykk[z] = punkter[z] + hydrostatiskTrykk[z];
249     }
250
251     endlessloopbrak++;
252     if (endlessloopbrak > 10000) {
253         System.out.println("Ikke gyldig løsning av væsketrykk");
254         break;
255     }
256 }
257
258 }
259 reeltTrykk[steps - 1] = punkter[steps - 1]
260     + hydrostatiskTrykk[steps - 1];
261 return reeltTrykk;
262 }
263
264 public chartFrame getVeskekurve(double punkter[], double stepsize,
265     double temp, double fluks) {
266     chartFrame veskekurve = new chartFrame("Væsketrykk gjennom varmerør",
267         "Lengde [m]", "Trykk [Pa]", punkter, "Væsketrykk", stepsize,
268         temp, fluks);
269     return veskekurve;
270 }
271
272 public limitChartFrame getBegrensninger(double qmin, double qmax,
273     double tmin, double tmax, double qStepSize, int EulerSteps) {
274     capLimitKurve = new double[(int) (tmax - tmin)];
275     bolingLimitKurve = new double[(int) (tmax - tmin)];
276     sonicLimitKurve = new double[(int) (tmax - tmin)];
277     vicousLimitKurve = new double[(int) (tmax - tmin)];
278
279     double temp;
280     boolean CapLimitFound;
281
282     int t;
283     double q;
284     double veskeKurveIpunkt[];
285     double dampkurveIpunkt[];
286
287     for (t = 0; t < (int) (tmax - tmin); t++) { //
288         temp = tmin + t; // temp er temperaturen som skal brukes i
289         // beregninger
290         CapLimitFound = false;
291
292         for (q = qmin; q <= qmax; q += qStepSize) {
293
294             veskeKurveIpunkt = lagVeskekurve(q, EulerSteps, temp);
295             dampkurveIpunkt = lagDampkurve(q, EulerSteps, temp);
296
297             if (veskeKurveIpunkt[0] < 0
298                 || dampkurveIpunkt[likningssett
299                     .getLowestValuePosition(dampkurveIpunkt)] < 0) //
300             {
301                 vicousLimitKurve[t] = q;
302                 CapLimitFound = true;

```



```

303     q = qmax;
304 }
305
306 else if (likningssett.capillary_limit_reached(
307     varmeror.arbeidsmedium.getProperty(temp,
308         arbeidsmedium.TEMPERATURE,
309         arbeidsmedium.SURFACETENSION_LIQUID),
310     varmeror.veke.effektivPoreradius, lagDampkurve(q,
311         EulerSteps, temp), veskeKurveIpunkt,
312     varmeror.arbeidsmedium.getProperty(temp,
313         arbeidsmedium.TEMPERATURE,
314         arbeidsmedium.TETHET_LIQUID),
315     varmeror.geometri.tilt,
316     varmeror.geometri.lengdeFordamper,
317     varmeror.geometri.lengdeAdiabatisk,
318     varmeror.geometri.lengdeKondensator)
319     && !CapLimitFound) {
320     CapLimitFound = true;
321     capLimitKurve[t] = q;
322     q = qmax;
323 }
324
325 }
326
327 // gir egentlig ut i Watt så må derfor gange med 0.001 for å få
328 // Kilowatt (hfg er gitt i KJ/KG)
329 boilingLimitKurve[t] = 0.001 * likningssett.boilingLimit(
330     varmeror.geometri.lengdeFordamper,
331     varmeror.veke.konduktivitet,
332     varmeror.geometri.ytreRadius, // radius av innervegg
333     varmeror.geometri.indreRadius, // vapor space radius
334     varmeror.arbeidsmedium.getProperty(temp,
335         arbeidsmedium.TEMPERATURE,
336         arbeidsmedium.SURFACETENSION_LIQUID),
337     temp,
338     varmeror.arbeidsmedium.getProperty(temp,
339         arbeidsmedium.TEMPERATURE, arbeidsmedium.HFG),
340     varmeror.arbeidsmedium.getProperty(temp,
341         arbeidsmedium.TEMPERATURE,
342         arbeidsmedium.TETHET_VAPOR),
343     varmeror.veke.effektivPoreradius, // Rmem kan settes til
344     // effektiv poreradius
345     // ifølge faghri
346     temp, varmeror.arbeidsmedium.getProperty(temp,
347         arbeidsmedium.TEMPERATURE, arbeidsmedium.K_LIQUID),
348     varmeror.arbeidsmedium.getProperty(temp,
349         arbeidsmedium.TEMPERATURE,
350         arbeidsmedium.VISKOSITET_VAPOR),
351     varmeror.arbeidsmedium.getProperty(temp,
352         arbeidsmedium.TEMPERATURE,
353         arbeidsmedium.VISKOSITET_LIQUID), q);
354
355 sonicLimitKurve[t] = 0.001 * likningssett
356     .sonicLimit(
357         varmeror.geometri.indreRadius,
358         varmeror.arbeidsmedium.getProperty(temp,
359             arbeidsmedium.TEMPERATURE,
360             arbeidsmedium.HFG),
361         varmeror.arbeidsmedium.getProperty(temp,
362             arbeidsmedium.TEMPERATURE,
363             arbeidsmedium.TETHET_VAPOR),
364
365         varmeror.arbeidsmedium.getProperty(temp,
366             arbeidsmedium.TEMPERATURE,
367             arbeidsmedium.CP_VAPOR)
368         / (varmeror.arbeidsmedium.getProperty(temp,
369             arbeidsmedium.TEMPERATURE,
370             arbeidsmedium.CP_VAPOR) - 8.314 / varmeror.arbeidsmedium.molarMasse),
371         temp);
372
373 }
374 limitChartFrame graf = new limitChartFrame(capLimitKurve,
375     boilingLimitKurve, sonicLimitKurve, viscousLimitKurve, tmin);
376 return graf;
377 }
378
379 public String getCapLimit(double Qe, int steps, double temperatur_fordamper) {
380     String contentstring = "Kalkulerer maksimum kappilærtrykk for varmerør.<BR>Fluks i fordamper Qe:"
381         + Double.toString(Qe)
382         + " kW<BR> Damp temperatur i fordamper Te:"
383         + Double.toString(temperatur_fordamper) + "&#176;K<BR><BR>";
384     double[] dampkurve = lagDampkurve(Qe, steps, temperatur_fordamper);
385     double[] veskekurve = lagVeskekurve(Qe, steps, temperatur_fordamper);

```

```

386
387 double middeltrykk = (veskekurve[steps - 1] + veskekurve[0]) * 0.5;
388
389 contentstring += "Høgeste væsketrykk i kondensator: "
390 + Double.toString(veskekurve[steps - 1]) + " Pa<BR>";
391 contentstring += "Laveste væsketrykk i fordampner: "
392 + Double.toString(veskekurve[0]) + " Pa<BR>";
393 contentstring += "Væskeegenskaper vurderes ved middeltrykk: "
394 + Double.toString(middeltrykk) + " Pa<BR><BR>";
395
396 double sigma = varmeror.arbeidsmedium.getProperty(middeltrykk,
397 arbeidsmedium.PRESSURE, arbeidsmedium.SURFACETENSION_LIQUID);
398 double rho = varmeror.arbeidsmedium.getProperty(middeltrykk,
399 arbeidsmedium.PRESSURE, arbeidsmedium.TETTHET_LIQUID);
400 contentstring += "Overfaltetespenning &#963; = " + Double.toString(sigma)
401 + " N/m<BR>";
402 contentstring += "Tetthet &#961; = " + Double.toString(rho)
403 + " kg/m3<BR><BR>";
404
405 double pkappmax = likningssett.Pkapp_max(sigma,
406 varmeror.veke.effektivPoreradius);
407 double totaltTrykktap = Math
408 .abs((veskekurve[0] - veskekurve[veskekurve.length - 1])
409 + Math.abs(dampkurve[0]
410 - dampkurve[dampkurve.length - 1]))
411 + rho
412 * 9.81
413 * Math.sin(Math.toRadians(varmeror.geometri.tilt))
414 * (varmeror.geometri.lengdeFordampner
415 + varmeror.geometri.lengdeAdiabatisk + varmeror.geometri.lengdeKondensator);
416 if (likningssett.capillary_limit_reached(sigma,
417 varmeror.veke.effektivPoreradius, dampkurve, veskekurve, rho,
418 varmeror.geometri.tilt, varmeror.geometri.lengdeFordampner,
419 varmeror.geometri.lengdeAdiabatisk,
420 varmeror.geometri.lengdeKondensator)) {
421 contentstring += "<FONT COLOR=red><B>ADVARSEL:</B> Kappilærbegrensning nådd<BR>"
422 + "Maksimum kappilæretrykk er "
423 + Double.toString(pkappmax)
424 + " Pa<BR>"
425 + "Totalt trykkfall er "
426 + Double.toString(totaltTrykktap) + " Pa</FONT><BR> ";
427 }
428
429 else {
430 contentstring += "<FONT COLOR=blue>Kappilærbegrensning er ikke nådd<BR>"
431 + "Maksimum kappilæretrykk er "
432 + Double.toString(pkappmax)
433 + " Pa<BR>"
434 + "Totalt trykkfall er "
435 + Double.toString(totaltTrykktap) + " Pa</FONT><BR> ";
436 }
437 return contentstring;
438 }
439
440 public String gethDelta_kondensator(double Qe, int steps,
441 double TsatFordampner) // termisk motstand i damp-væske grensesnitt
442 {
443 double[] dampkurve = lagDampkurve(Qe, steps, TsatFordampner);
444 double[] veskekurve = lagVeskekurve(Qe, steps, TsatFordampner);
445
446 double alpha = 1.0; // koeffisient foreslått av Faghri til å sette til 1
447 // . annet for forurensede medier
448
449 int kondensatorInngangPosisjon = (int) (steps - varmeror.geometri.lengdeKondensator
450 / ((varmeror.geometri.lengdeAdiabatisk
451 + varmeror.geometri.lengdeFordampner + varmeror.geometri.lengdeKondensator) / steps));
452
453 double kondensatorInngangTemp = varmeror.arbeidsmedium.getProperty(
454 dampkurve[kondensatorInngangPosisjon], arbeidsmedium.PRESSURE,
455 arbeidsmedium.TEMPERATURE);
456 double kondensatorUtgangTemp = varmeror.arbeidsmedium.getProperty(
457 dampkurve[dampkurve.length - 1], arbeidsmedium.PRESSURE,
458 arbeidsmedium.TEMPERATURE);
459 double Tv_middel = likningssett.aritmetiskMiddelTemperatur(
460 kondensatorInngangTemp, kondensatorUtgangTemp);
461
462 double kondensatorInngangVeskeTemp = varmeror.arbeidsmedium
463 .getProperty(veskekurve[kondensatorInngangPosisjon],
464 arbeidsmedium.PRESSURE, arbeidsmedium.TEMPERATURE);
465 double kondensatorUtgangVeskeTemp = varmeror.arbeidsmedium.getProperty(
466 veskekurve[dampkurve.length - 1], arbeidsmedium.PRESSURE,
467 arbeidsmedium.TEMPERATURE);
468 double Tl_middel = likningssett.aritmetiskMiddelTemperatur(

```

```

469         kondensatorInngangVeskeTemp, kondensatorUtgangVeskeTemp);
470
471     double TsatKondenser = Tv_middel;
472
473     double hfg = varmeror.arbeidsmedium.getProperty(TsatKondenser,
474         arbeidsmedium.TEMPERATURE, arbeidsmedium.HFG);
475     double v_damp = 1 / varmeror.arbeidsmedium.getProperty(TsatKondenser,
476         arbeidsmedium.TEMPERATURE, arbeidsmedium.TETHET_VAPOR);
477     double v-veske = 1 / varmeror.arbeidsmedium.getProperty(TsatKondenser,
478         arbeidsmedium.TEMPERATURE, arbeidsmedium.VISKOSITET_LIQUID);
479     double vfg = v-veske - v_damp;
480     double rho_v = 1 / v_damp;
481
482     double Mv = varmeror.arbeidsmedium.molarMasse;
483
484     double h = likningssett.h_delta(alpha, hfg, Tv_middel, vfg, rho_v, Mv);
485     String contentstring = "<TR><TD>Damp<TD>temperatur ved begynnelse av kondensator:</TD><TD>"
486         + Double.toString(round(kondensatorInngangTemp, 2))
487         + "</TD><TD> &#176;K</TD></TR>"
488         + "<TR><TD>Damp<TD>temperatur ved ende av kondensator:</TD><TD>"
489         + Double.toString(round(kondensatorUtgangTemp, 2))
490         + "</TD><TD> &#176;K</TD></TR>"
491         + "<TR><TD>Aritmetisk damp temperatur i kondensator (Tv): </TD><TD>"
492         + Double.toString(round(Tv_middel, 2))
493         + "</TD><TD> &#176;K</TD></TR>"
494         + "<TR><TD>Aritmetisk vesketemperatur i kondensator (Tl):</TD><TD>"
495         + Double.toString(round(Tl_middel, 2))
496         + "</TD><TD> &#176;K</TD></TR>"
497         + "<TR><TD>Varmetransportkoeffisient damp-v&#228;skegrensesnitt i kondensator h&#948;</TD><TD><FONT
498             COLOR=green>"
499         + Double.toString(round(h, 2))
500         + "</FONT><TD><TD>W/m^2K</TD></TR>";
501     hDelta_kondensator = h;
502     return contentstring;
503 }
504
505 public String gethDelta_fordamper(double Qe, int steps, double TsatFordamper) {
506     double[] dampkurve = lagDampkurve(Qe, steps, TsatFordamper);
507     double[] veskekurve = lagVeskekurve(Qe, steps, TsatFordamper);
508
509     double alpha = 1.0; // koeffisient foresl&#228;tt av Faghri til &#228; sette til 1
510     // . annet for forurenkede medier
511
512     int fordamperUtgangPosisjon = (int) (varmeror.geometri.lengdeFordamper / ((varmeror.geometri.
513         lengdeAdiabatisk
514         + varmeror.geometri.lengdeFordamper + varmeror.geometri.lengdeKondensator) / steps));
515
516     double fordamperDampInngangTemp = varmeror.arbeidsmedium
517         .getProperty(dampkurve[0], arbeidsmedium.PRESSURE,
518         arbeidsmedium.TEMPERATURE);
519     double fordamperDampUtgangTemp = varmeror.arbeidsmedium.getProperty(
520         dampkurve[fordamperUtgangPosisjon], arbeidsmedium.PRESSURE,
521         arbeidsmedium.TEMPERATURE);
522
523     double Tv_middel = likningssett.aritmetiskMiddelTemperatur(
524         fordamperDampInngangTemp, fordamperDampUtgangTemp);
525
526     double fordamperDampInngangVeskeTemp = varmeror.arbeidsmedium
527         .getProperty(veskekurve[0], arbeidsmedium.PRESSURE,
528         arbeidsmedium.TEMPERATURE);
529     double fordamperDampUtgangVeskeTemp = varmeror.arbeidsmedium
530         .getProperty(veskekurve[fordamperUtgangPosisjon],
531         arbeidsmedium.PRESSURE, arbeidsmedium.TEMPERATURE);
532
533     double Tl_middel = likningssett.aritmetiskMiddelTemperatur(
534         fordamperDampInngangVeskeTemp, fordamperDampUtgangVeskeTemp);
535
536     double hfg = varmeror.arbeidsmedium.getProperty(Tv_middel,
537         arbeidsmedium.TEMPERATURE, arbeidsmedium.HFG);
538     double v_damp = 1 / varmeror.arbeidsmedium.getProperty(Tv_middel,
539         arbeidsmedium.TEMPERATURE, arbeidsmedium.TETHET_VAPOR);
540     double v-veske = 1 / varmeror.arbeidsmedium.getProperty(Tv_middel,
541         arbeidsmedium.TEMPERATURE, arbeidsmedium.VISKOSITET_LIQUID);
542     double vfg = v-veske - v_damp;
543     double rho_v = 1 / v_damp;
544
545     double Mv = varmeror.arbeidsmedium.molarMasse;
546
547     // Tester om det er ok &#228; negligjere kompressibilitet (antar mettet
548     // damp ut av fordamper)
549     String contentstring = "";
550     double Pv = dampkurve[fordamperUtgangPosisjon];
551     if (!likningssett.okToNeglectCompressibleFlow(

```

```

550 //
551 varmeror.geometri.lengdeFordamper ,
552 varmeror.geometri.lengdeFordamper ,
553 varmeror.geometri.lengdeAdiabatisk ,
554 varmeror.geometri.lengdeKondensator, Qe, varmeror.arbeidsmedium
555 .getProperty(Pv, arbeidsmedium.PRESSURE,
556 arbeidsmedium.HFG), varmeror.arbeidsmedium
557 .getProperty(Pv, arbeidsmedium.PRESSURE,
558 arbeidsmedium.TETHET_VAPOR), Pv,
559 varmeror.arbeidsmedium.getProperty(Pv, arbeidsmedium.PRESSURE,
560 arbeidsmedium.CP_VAPOR), varmeror.arbeidsmedium
561 .getProperty(Pv, arbeidsmedium.PRESSURE,
562 arbeidsmedium.CP_VAPOR) / 1.6, Math.PI
563 * Math.pow(varmeror.geometri.indreRadius, 2)))
564 contentstring += "<TR><TD colspan=3><FONT color=red><B>Advarsel:</B> Neglisjerer kompressibilitet
    til tross for Ma > 0.3</font></TD></TR>";
565
566 double h = likningsett.h_delta(alpha, hfg, Tv_middel, vfg, rho_v, Mv);
567
568 contentstring += "<TR><TD>Dampetemperatur ved begynnelse av fordamper:</TD><TD>"
569 + Double.toString(round(fordamperDampInngangTemp, 2))
570 + "</TD><TD> &#176;K</TD></TR>"
571 + "<TR><TD>Dampetemperatur ved fordamperutgang:</TD><TD>"
572 + Double.toString(round(fordamperDampUtgangTemp, 2))
573 + "</TD><TD> &#176;K</TD></TR>"
574 + "<TR><TD>Aritmetisk damp temperatur i fordamper (Tv):</TD><TD>"
575 + Double.toString(round(Tv_middel, 2))
576 + "</TD><TD> &#176;K</TD></TR>"
577 + "<TR><TD>Aritmetisk vesketemperatur i fordamper (Tl):</TD><TD> "
578 + Double.toString(round(Tl_middel, 2))
579 + "</TD><TD> &#176;K</TD></TR>"
580
581 + "<TR><TD>Varmetransportkoeffisient damp-v skegrensesnitt i fordamper h #948;:</TD><TD> <FONT
    COLOR=green>"
582 + Double.toString(round(h, 2))
583 + "</FONT></TD><TD> W/m^2*K</TD></TR>";
584 hDelta_fordamper = h;
585
586 return contentstring;
587 }
588
589 public String getOutsideTempeKondensator(double Qe, int steps,
590 double TfordamperUtside) {
591 double Tv_guess = TfordamperUtside - 1; //
592 double tolerance = 0.00000001;
593 double TfordamperUtside_Beregnet = TfordamperUtside + 1;
594 double keff = 0.0;
595 double Tkondensator_Beregnet;
596 String contentstring2 = "";
597 String contentstring = "<TABLE BORDER=0 CELLPADDING=3><TR><TD Width=100%><B>Variabel</B></TD><TD><B>
    Verdi</B></TD><TD><B>Enhhet</B></TD></TR>";
598 int endlessloopwarning = 0;
599 double R_fordamper = 0;
600
601 while (Math.abs(TfordamperUtside - TfordamperUtside_Beregnet) > tolerance) {
602 contentstring2 = gethDelta_fordamper(Qe, steps, Tv_guess);//
603
604 double kl = varmeror.arbeidsmedium.getProperty(Tv_guess,
605 arbeidsmedium.TEMPERATURE, arbeidsmedium.K_LIQUID);
606 keff = likningsett.keff(varmeror.veke.vekeType, kl,
607 varmeror.veke.konduktivit t, varmeror.veke.porositet);
608
609 R_fordamper = likningsett.R_fordamper(hDelta_fordamper, keff,
610 varmeror.geometri.beholderKonduktivit t,
611 varmeror.geometri.indreRadius,
612 varmeror.geometri.ytreRadius, varmeror.geometri.ytreRadius
613 + varmeror.geometri.beholderTykkelse,
614 varmeror.geometri.lengdeFordamper);
615 TfordamperUtside_Beregnet = Tv_guess + Qe * R_fordamper;//
616
617 if (Math.abs(TfordamperUtside - TfordamperUtside_Beregnet) > tolerance)
618 Tv_guess += TfordamperUtside - TfordamperUtside_Beregnet;//
619 endlessloopwarning++;
620 if (endlessloopwarning > 1000)
621 break;
622 }
623 contentstring2 += contentstring2;
624 contentstring += "<TR><TD>Fant l sning etter " + endlessloopwarning
625 + " itterasjoner</TD></TR>";
626 contentstring += "<TR><TD>Ty ytterside fordamper beregnet</TD><TD>"
627 + Double.toString(round(TfordamperUtside_Beregnet, 2))
628 + "</TD><TD> &#176;K</TD></TR>";
629

```

```

630 contentstring += gethDelta_kondensator(Qe, steps, Tv_guess);
631 double R_kondensator = likningssett.R_kondensator(hDelta_kondensator,
632 1.0, keff, varmeror.geometri.beholderKonduktivitet,
633 varmeror.geometri.indreRadius, varmeror.geometri.ytreRadius,
634 varmeror.geometri.ytreRadius
635 + varmeror.geometri.beholderTykkelse,
636 varmeror.geometri.lengdeKondensator);
637 Tkondensator_Beregnet = Tv_guess - Qe * R_kondensator;
638 contentstring += "<TR><TD>Ty ytterside kondensator beregnet til</TD><TD><FONT Color=green>"
639 + Double.toString(round(Tkondensator_Beregnet, 2))
640 + "</FONT></TD><TD> &#176;K</TD></TR>"
641 + "<TR><TD>Varmemotstand i fordampner (R_fordampner):</TD><TD><FONT Color=green>"
642 + Double.toString(round(R_fordampner, 2))
643 + "</FONT></TD><TD> &#176;K/W</TD></TR>"
644 + "<TR><TD>Varmemotstand i kondensator (R_kondensator):</TD><TD><FONT Color=green>"
645 + Double.toString(round(R_kondensator, 2))
646 + "</FONT></TD><TD> &#176;K/W</TD></TR>"
647
648 + "<TR><TD>Temperaturdifferanse &Delta;T over varmerør</TD><TD><FONT Color=green>"
649 + Double.toString(round(TfordampnerUtside
650 - Tkondensator_Beregnet, 2))
651 + "</FONT></TD><TD> &#176;K</TD></TR>";
652
653 double varmeledningsevne = (Qe * 1000
654 * varmeror.geometri.lengdeAdiabatisk
655 + varmeror.geometri.lengdeFordampner + varmeror.geometri.lengdeKondensator)
656 / (Math.PI
657 * Math.pow(varmeror.geometri.ytreRadius
658 + varmeror.geometri.beholderTykkelse, 2) * (TfordampnerUtside - Tkondensator_Beregnet
659 ));
659 contentstring += "<TR><TD>Termisk varmeledningsevne</TD><TD><FONT Color=green>"
660 + Double.toString(round(varmeledningsevne, 2))
661 + "</FONT></TD><TD> W/m&#176;K</TD></TR>";
662
663 contentstring += "</TABLE>";
664
665 return contentstring;
666 }
667
668 public String minimumDiameter(double qe, int steps, double temp)//
669 {
670 String contentstring = "<TABLE BORDER=0 CELLSPACING=3><TR><TD Width=100%><B>Variabel</B></TD><TD><B>
671 Verdi</B></TD><TD><B>Enhet</B></TD></TR>";
672 double Rv = varmeror.geometri.indreRadius;
673 double Ry = varmeror.geometri.ytreRadius;
674 double L = varmeror.geometri.beholderTykkelse;
675
676 double step = 0.00001;
677
678 double[] veskeKurveIpunkt;
679 double[] dampkurveIpunkt;
680 Double Qe = 0.0;
681 boolean LimitNotReached = true;
682
683 while (LimitNotReached) {
684 Qe = qe * (varmeror.geometri.ytreRadius + L) * 2 * Math.PI
685 * varmeror.geometri.lengdeFordampner;
686
687 veskeKurveIpunkt = lagVeskekurve(Qe, steps, temp);
688 dampkurveIpunkt = lagDampkurve(Qe, steps, temp);
689 if (veskeKurveIpunkt[0] < 0
690 || dampkurveIpunkt[likningssett
691 .getLowestValuePosition(dampkurveIpunkt)] < 0) {
692 contentstring += "<TR><TD>Veskestrykket i fordampner er mindre enn 0, Rv</TD><TD><FONT Color=green>"
693 + Double.toString(varmeror.geometri.indreRadius)
694 + "</TD></TR>";
695 LimitNotReached = false;
696 }
697
698 else if (likningssett.capillary_limit_reached(varmeror.arbeidsmedium
699 .getProperty(temp, arbeidsmedium.TEMPERATURE,
700 arbeidsmedium.SURFACETENSION_LIQUID),
701 varmeror.veke.effektivPoreradius, lagDampkurve(Qe, steps,
702 temp), veskeKurveIpunkt, varmeror.arbeidsmedium
703 .getProperty(temp, arbeidsmedium.TEMPERATURE,
704 arbeidsmedium.TETHET_LIQUID),
705 varmeror.geometri.tilt, varmeror.geometri.lengdeFordampner,
706 varmeror.geometri.lengdeAdiabatisk,
707 varmeror.geometri.lengdeKondensator)) {
708 LimitNotReached = false;
709 contentstring += "<TR><TD>Kapillærbegrensning funnet ved Rv</TD><TD><FONT Color=green>"
710 + Double.toString(varmeror.geometri.indreRadius)

```

```

710         + "</TD></TR>";
711     }
712
713     double boiling = 0.001 * likningsett.boilingLimit(
714         varmeror.geometri.lengdeFordamper,
715         varmeror.veke.konduktivitet,
716         varmeror.geometri.ytreRadius, // radius av innervegg
717         varmeror.geometri.indreRadius, // vapor space radius
718         varmeror.arbeidsmedium.getProperty(temp,
719             arbeidsmedium.TEMPERATURE,
720             arbeidsmedium.SURFACETENSION_LIQUID),
721         temp,
722         varmeror.arbeidsmedium.getProperty(temp,
723             arbeidsmedium.TEMPERATURE, arbeidsmedium.HFG),
724         varmeror.arbeidsmedium.getProperty(temp,
725             arbeidsmedium.TEMPERATURE,
726             arbeidsmedium.TETHET_VAPOR),
727         varmeror.veke.aktivPoreradius, // Rmem kan settes til
728         // effektiv poreradius
729         // ifølge faghri
730         temp, varmeror.arbeidsmedium.getProperty(temp,
731             arbeidsmedium.TEMPERATURE, arbeidsmedium.K_LIQUID),
732         varmeror.arbeidsmedium.getProperty(temp,
733             arbeidsmedium.TEMPERATURE,
734             arbeidsmedium.VISKOSITET_VAPOR),
735         varmeror.arbeidsmedium.getProperty(temp,
736             arbeidsmedium.TEMPERATURE,
737             arbeidsmedium.VISKOSITET_LIQUID), Qe);
738
739     if (boiling < Qe) {
740         LimitNotReached = false;
741         contentstring += "<TR><TD>kokebegrensning ved Rv</TD><TD><FONT Color=green>"
742             + Double.toString(varmeror.geometri.indreRadius)
743             + "</TD><TD>m</TD></TR>";
744     }
745
746     double sonic = 0.001 * likningsett
747         .sonicLimit(
748         varmeror.geometri.indreRadius,
749         varmeror.arbeidsmedium.getProperty(temp,
750             arbeidsmedium.TEMPERATURE,
751             arbeidsmedium.HFG),
752         varmeror.arbeidsmedium.getProperty(temp,
753             arbeidsmedium.TEMPERATURE,
754             arbeidsmedium.TETHET_VAPOR),
755         varmeror.arbeidsmedium.getProperty(temp,
756             arbeidsmedium.TEMPERATURE,
757             arbeidsmedium.CP_VAPOR)
758         / (varmeror.arbeidsmedium.getProperty(temp,
759             arbeidsmedium.TEMPERATURE,
760             arbeidsmedium.CP_VAPOR) - 8.314 / varmeror.arbeidsmedium.molarMasse),
761         temp);
762     if (sonic < Qe) {
763         LimitNotReached = false;
764         contentstring += "<TR><TD>Sonisk begrensning ved Rv</TD><TD><FONT Color=green>"
765             + Double.toString(varmeror.geometri.indreRadius)
766             + "</TD><TD>m</TD></TR>";
767     }
768
769     varmeror.geometri.indreRadius -= step;
770     varmeror.geometri.ytreRadius -= step;
771     if (varmeror.geometri.indreRadius < 0) {
772         contentstring += "<TR><TD>Rv mindre enn null</TD><TD><FONT Color=green>+"
773             + Double.toString(varmeror.geometri.indreRadius)
774             + "</TD></TR>";
775         break;
776     }
777 }
778
779 varmeror.geometri.indreRadius = Rv;
780 varmeror.geometri.ytreRadius = Ry;
781 contentstring += "<TR><TD>Varmebelastning Qe</TD><TD><FONT Color=green>"
782     + Double.toString(Qe) + " </TD><TD>kW</TD></TR>";
783 contentstring += "<TR><TD>Spesifikk varmefluks</TD><TD><FONT Color=green>"
784     + Double.toString(qe) + " </TD><TD>kW/m^2</TD></TR>";
785 contentstring += "<TR><TD>Dampetemperatur ved fordampstart</TD><TD><FONT Color=green>"
786     + Double.toString(temp) + " </TD><TD>K</TD></TR>";
787 contentstring += "</TABLE>";
788 return contentstring;
789 }
790
791 public static double round(double d, int decimalPlace) {
792     BigDecimal bd = new BigDecimal(Double.toString(d));

```

```

793     bd = bd.setScale(decimalPlace, BigDecimal.ROUND_HALF_UP);
794     return bd.doubleValue();
795 }
796
797 }// EOC

```

---

### Listing A.2: likningsett.java

---

```

1
2 public class likningsett {
3
4 //Gir teoretisk maks kappilært trykk
5 public double youngLaplace(double overflatespenning,double poreradius,double kontaktvinkel)
6 {
7     if (!Double.isNaN(overflatespenning) & !Double.isNaN(poreradius) & !Double.isNaN(kontaktvinkel))
8     {
9         return (2*overflatespenning*Math.cos(kontaktvinkel))/poreradius;
10    }
11
12    System.out.println("Young-Laplace ikke kalkulert");
13    return 0.0;
14 }
15
16 //
17 public double Re_radiell(double z, double u_v,double Le, double La,double Lk,double Qe,double hfg) //radielt
18     reynoldstall
19 {
20     return (1/(2*Math.PI*u_v))*m_damp_derivert(z,Le,La,Lk,Qe,hfg);
21 }
22 public double m_damp_derivert(double z, double Le, double La, double Lk,double Qe,double hfg) //egentlig
23     massestrøm damp generert i fordamer eller kondensert i kondensator
24 {
25     if (z <= Le)
26     {
27         return (1/Le)*(Qe/hfg);
28     }
29     else if (z > Le && z<= (Le +La))
30     {
31         return 0.0;
32     }
33     else if (z > (Le+La) && z<= (Le +La+Lk))
34     {
35         return (-1/Lk)*(Qe/hfg);
36     }
37     else {
38         System.out.println("Feil: utenfor område i funksjon m_damp");
39         return 0.0;}
40 }
41 }
42
43 //
44 public double m_damp(double z, double Le, double La, double Lk,double Qe,double hfg) //egentlig massestrøm
45     damp generert i fordamer eller kondensert i kondensator
46 {
47     if (z <= Le)
48     {
49         return (z/Le)*(Qe/hfg);
50     }
51     else if (z > Le && z<= (Le +La))
52     {
53         return (Qe/hfg);
54     }
55     else if (z > (Le+La) && z<= (Le +La+Lk))
56     {
57         return ((Le+La+Lk-z)/Lk)*(Qe/hfg);
58     }
59     else {
60         System.out.println("Feil: utenfor område i funksjon m_damp");
61         return 0.0;}
62 }
63 }
64
65
66 public double m_veske(double z, double Le, double La, double Lk,double Qe,double hfg){
67 if (z <= Le)//Fordamper
68 {
69     return ((z/Le))*(Qe/hfg);
70 }
71 else if (z > Le && z<= (Le +La))

```

```

72     {
73     return (Qe/hfg);
74     }
75 else if (z > (Le+La) && z <= (Le +La+Lk))
76     {
77     return (((Le+La+Lk-z)/Lk))*(Qe/hfg);
78     }
79     }
80 else {
81     System.out.println("Feil: utenfor område i funksjon m_veske z=" + Double.toString(z));
82     return 0.0;}
83 }
84 }
85
86
87 //----- LIKNINGER FOR DAMPTRYKK I HEAT PIPE
88 //
89 public double Pv_by_simplified_closed_form_approximation_euler(double p1,double stepSize,double z, double u_v
, double Le, double La, double Lk, double Qe, double hfg, double rho_v, double Rv)
90     {
91     //System.out.println("Re_radiell " +Re_radiell(z,u_v,Le,La,Lk,Qe,hfg));
92     //System.out.println("z: "+z + " u_v: "+u_v + " Le "+Le + " Qe " + Qe + " hfg " + hfg);
93     if (Math.abs(Re_radiell(z,u_v,Le,La,Lk,Qe,hfg)) < 1) //egentlig |Re| << 1 (bassert på Faghri likning 3.61)
94     {
95
96         //System.out.println(" u_v: "+u_v + " Qe " + Qe +"z^2" + Math.pow(z,2) + " PI "+Math.PI + " Rho_v "
+rho_v + " Rv^4 " + Math.pow(Rv, 4) + " hfg " + hfg + " le " + Le);
97
98         return p1 + ((-8*u_v*m_damp(z,Le,La,Lk,Qe,hfg))/(Math.PI*rho_v*Math.pow(Rv, 4))*(1+(3/4)*Re_radiell(
z,u_v,Le,La,Lk,Qe,hfg)))*stepSize;
99     }
100 }
101 else if (Math.abs(Re_radiell(z,u_v,Le,La,Lk,Qe,hfg)) > 1) //|Re| >> 1
102     {
103     if (z >= 0 && z <= Le)//Fordamper
104     {
105     return p1 + (((-m_damp(z,Le,La,Lk,Qe,hfg))/(4*rho_v*Math.pow(Rv, 4)))*(Qe/(Le*hfg)))*stepSize;
106     }
107     else if (z > Le && z <= (Le+La))
108     {
109     return p1 + 0.0*stepSize;
110     }
111     else if (z > (Le+La) && z <= (Le+La+Lk))
112     {
113     return p1 + (((-(4/Math.pow(Math.PI, 2))*m_damp(z,Le,La,Lk,Qe,hfg))/(4*rho_v*Math.pow(Rv, 4)))*(-Qe
/(Lk*hfg)))*stepSize;
114     }
115     }
116     else //Re = 1...what to do
117     {
118     return 0.0;
119     }
120     return 0.0;
121     }
122
123
124
125 //----- LIKNINGER FOR VÆSKETRYKK I VEKE
126 //-----
127 public double Pl(double p1, double stepSize, double z, double u_l, double rho_l, double A_veke, double K,
double Le, double La, double Lk, double Qe, double hfg)
128     {
129     return p1 - ((u_l*m_veske(z,Le,La,Lk,Qe,hfg))/(rho_l*A_veke*K))*stepSize;
130     }
131
132 public double Pkapp_max(double sigma, double reff)
133     {
134     //return 2*sigma*Math.cos(fluidKontaktVinkel)/reff;
135     return 2*sigma/reff;
136     }
137
138
139 //----- LIKNINGER FOR BEGRENSNINGER
140 //-----
141 public boolean capillary_limit_reached(double sigma,double reff,double pv[],double pl[],double rho_l,double
tilt,double Le,double La,double Lk) //
142     {
143     int wetpoint = getWetpoint(pv, pl, Lk,La,Le);
144
145     if (wetpoint >= pv.length-2) La+=Lk; //kompanserer for Wetpoint ved endcap

```



```

146     if((2*sigma)/reff < Math.abs((pl[0]-pl[wetpoint]) + Math.abs(pv[0]-pv[wetpoint])) + rho_l*9.81*Math.sin
147         (Math.toRadians(tilt))*(La+Le))
148     {
149         return true;
150     }
151     else return false;
152 }
153
154 //
155 //
156 public double boilingLimit(double Le,double keff,double Ri,double Rv,double sigma,double T_v, double hfg,
157     double rho_v,double Rmen, double tsat, double kl, double v_v,double v_l, double qr)
158 {
159     //Denne returnerer i enhet WATT dvs qr må bli gitt i watt
160     return ( (2*Math.PI*Le*keff*Tkritisk( sigma, T_v, hfg, rho_v, Rmen, tsat, kl, v_v, v_l, qr)) / (
161         Math.log(Ri/Rv) );
162 }
163 //
164 //
165 public double Tkritisk(double sigma,double T_v, double hfg, double rho_v,double Rmen, double tsat, double kl,
166     double v_v,double v_l, double qr)
167 {
168     return ((2*sigma*T_v)/(1000*hfg*rho_v))*((1/0.000010)-(1/Rmen)); //OPSOPS T_v er kanskje ikke rett her
169     !!!
170 }
171
172 public double rb(double sigma, double tsat, double kl, double v_v,double v_l, double hfg, double qr)
173 {
174     if (sigma == Double.NaN || tsat == Double.NaN || kl == Double.NaN || v_v == Double.NaN || v_l == Double.
175         NaN || hfg == Double.NaN || qr == Double.NaN)
176     {return 0.000010;} //anbefalt verdi av faghri
177     else
178     {
179         return Math.sqrt((2*sigma*tsat*kl*(v_v-v_l))/(hfg*qr));
180     }
181 }
182
183 public double speedOfSound(double cp,double R, double Pv,double rho_v)
184 {
185     return Math.sqrt((cp/(cp-R))*(Pv/rho_v));
186 }
187
188 public double sonicLimit(double Ri,double hfg, double rho_v,double K, double T_v) //
189 {
190     return (Math.PI*Math.pow(Ri, 2))*rho_v*hfg*1000*Math.sqrt( (K*T_v*8.314)/(2*(K+1)) );
191 }
192
193 public boolean okToNeglectCompressibleFlow(double z, double Le, double La, double Lk,double Qe,double hfg,
194     double rho_v,double Pv,double cp,double cv,double crossSectionArea)
195 {
196     double massFlow = m_damp(z,Le,La,Lk, Qe,hfg);
197     double w = speedOfFlow(massFlow,rho_v,crossSectionArea);
198     double c = speedOfSound(cp,cv,Pv,rho_v);
199     if (w/c <= 0.3) return true;
200     else return false;
201 }
202 public double speedOfFlow(double massFlow,double rho,double crossSectionArea)
203 {
204     return massFlow/(rho*crossSectionArea);
205 }
206 //----- LIKNINGER FOR FILMKONDENSASJON
207
208 public double Prandtl(double Cp,double u, double k)
209 {
210     return (Cp*u)/k;
211 }
212
213 public double kinematisk_viskositet(double u_l,double rho_l) //u_l [pa*sec] og Rho [kg/m^3] returnerer m^2/sec
214 {
215     return u_l/rho_l;
216 }
217
218
219
220

```

```

221 //----- ANDRE RUTINER
222 public int getLowestValuePosition(double[] array) //Finner laveste verdi i denne arrayen og returnerer
    posisjon (Treig algoritme)
223 {
224     int position = 0;
225     for (int i=0;i < array.length;i++)
226     {
227         if (array[i] < array[position]) position = i;
228     }
229 }
230 return position;
231 }
232
233 public double aritmetiskMiddelTemperatur(double Tinn,double Tut)
234 {
235     return (Tinn + Tut)/2;
236 }
237
238 //----- TERMISK GRENESKIKT
239 public double h_delta(double alpha,double hfg,double Tv, double vfg, double rho_v, double Mv)//
240 {
241     //System.out.println("Her: " + Double.toString( vfg ) );
242     return ((2*alpha)/(2-alpha)) *(Math.pow(hfg,2)/(Tv*vfg)) * Math.sqrt((Mv/(2*Math.PI))*8.314*Tv) *
        (1-((rho_v*vfg)/(2*hfg))) ;
243 }
244
245 public double keff(int veketype,double kl,double ks,double porositet) //
246 {
247     if (veketype==veke.SINTREDE_VEKER_AV_METALLPULVER) //sintrede veker av metallpulver
248     {
249         return (2+kl/ks-2*porositet*(1-kl/ks))/(2+kl/ks-porositet*(1-kl/ks));
250     }
251     else if (veketype==veke.METALLNETT) //sintrede veker av metallpulver
252     {
253         return (kl*((kl+ks)-(1-porositet)*(kl-ks)))/ ((kl+ks)+(1-porositet)*(kl-ks));
254     }
255 }
256
257 else return 0.0;
258 }
259
260 public double R_kondensator(double h_veskeDampGrensesnitt,double h_kondensfilm,double keff,double kvegg,
    double Rv,double Ri,double Ry,double L)
261 {
262     double veskeDampGrensesnitt = 1/(h_veskeDampGrensesnitt*2*Math.PI*Rv*L);
263
264     //double kondensfilm = 1/(h_kondensfilm*2*Math.PI*Rv*L);
265     double kondensfilm = 0.0;
266
267     double veke = Math.log(Ri/Rv)/(2*Math.PI*L*keff);
268
269     double vegg = Math.log(Ry/Ri)/(2*Math.PI*L*kvegg);
270
271     return veskeDampGrensesnitt+kondensfilm+veke+vegg;
272 }
273
274 public double R_fordamper(double h_veskeDampGrensesnitt,double keff,double kvegg,double Rv,double Ri,double
    Ry,double L)
275 {
276     double veskeDampGrensesnitt = 1/(h_veskeDampGrensesnitt*2*Math.PI*Rv*L);
277
278     double veke = Math.log(Ri/Rv)/(2*Math.PI*L*keff);
279
280     double vegg = Math.log(Ry/Ri)/(2*Math.PI*L*kvegg);
281
282     return veskeDampGrensesnitt+veke+vegg;
283 }
284
285
286 public double Pv_delta(double psatForTdeltaLiquid,double TdeltaLiquid,double sigma,double reff,double rho_l,
    double Rg)
287 {
288     double Pv_delta = psatForTdeltaLiquid+10;
289     double x = (Pv_delta-psatForTdeltaLiquid-2*sigma / reff) / (rho_l*Rg*TdeltaLiquid);
290
291     int endlessloopstopper = 0;
292
293     while(Math.abs(Pv_delta - psatForTdeltaLiquid*Math.exp(x)) > 0.000001)
294     {
295         Pv_delta -= Pv_delta - psatForTdeltaLiquid*Math.exp(x);
296         x = (Pv_delta-psatForTdeltaLiquid-2*sigma / reff) / (rho_l*Rg*TdeltaLiquid);

```

```

297     if (endlessloopstopper > 1000) {System.out.println("Pv_delta kunne ikke bli funnet");break;}
298     endlessloopstopper++;
299     }
300     return Pv_delta;
301     }
302
303     public int getWetpoint(double[] Pv, double[] Pl, double Lk, double La, double Le) //
304     {
305     int kondensatorinngang = (int) (Pv.length - (Pv.length / (Lk + La + Le))*Lk);
306     double dampTrykkdifferanse = Pv[Pv.length-1] - Pv[kondensatorinngang];
307     double v skekTrykkdifferanse = Pl[Pl.length-1] - Pl[kondensatorinngang];
308
309     if (dampTrykkdifferanse > v skekTrykkdifferanse) return kondensatorinngang;
310     else return Pl.length-1;
311     }
312 }
313 //----- KOMPRESIBEL ENDIMENSJONAL ANALYSE
314
315     public double A(double Re_r)
316     {
317     double c = Math.pow(Math.pow(5+ 18/Re_r,2)-8.8, 0.5);
318     return (15/22)*(5+18/Re_r-c);
319     }
320
321     public double F3(double A)
322     {
323     return 8*(0.25-(A/10)+(Math.pow(A,2)/30) - 2*(Math.pow(A,3)/945));
324     }
325     public double HAF(double A)
326     {
327     return -0.4 * (1 - 2*A/3 + 4*(Math.pow(A,2)/63) );
328     }
329     public double HVE(double A, double hm, double Hr, double vm)
330     {
331     return hm - Hr + 1.5*(F3(A)*Math.pow(vm,2));
332     }
333     public double GRP(double Rg, double P)
334     {
335     return 1- Math.log(Rg)/Math.log(P); //(egentlig partiellderivert)
336     }
337     public double GRT(double Tm, double P)
338     {
339     return 1+ Math.log(Tm)/Math.log(P); //(egentlig partiellderivert)
340     }
341     public double GB(double A)
342     {
343     return (1- A/6 + (2/45)*Math.pow(A, 2));
344     }
345     public double HB(double A)
346     {
347     return -((1/6)-(4*A/45) );
348     }
349
350
351
352     public double dqR_dz(double R_v, double dm_dz, double rho_R, double hfg)
353     {
354     return (1/(2*Math.PI*R_v))*dm_dz*(hfg+Math.pow((1/(2*Math.PI*R_v*rho_R))*dm_dz,2));
355     }
356
357
358
359     public double lewis1(double rho_m, double v_m, double p, double A, double Rg, double Tm, double u, double R_v, double
dlnp_dz, double dlnAv_dz)
360     {
361     double a_dlnp_dz = 1- (4*rho_m*Math.pow(v_m,2))/(3*p) * GB(A)*GRP(Rg,p);
362     double b_dlnTm_dz = (4*rho_m*Math.pow(v_m,2))/(3*p) * GB(A)*GRT(Tm,p);
363     double c_dA_dz = (4*rho_m*Math.pow(v_m,2))/(3*p)*HB(A);
364
365     double d = -((8*u*v_m)/(Math.pow(R_v, 2)*p))*(1+(2/3)*A)-((8*rho_m*Math.pow(v_m,2))/(3*p))*GB(A)*(dlnp_dz-
dlnAv_dz);
366
367     return a_dlnp_dz+b_dlnTm_dz+c_dA_dz-d; //skal bli null
368     }
369
370     public double lewis2(double dhm_dlnp_Tm, double A, double v_m, double Rg, double P)
371     {
372     double a_dlnp_dz = dhm_dlnp_Tm - F3(A)*Math.pow(v_m, 2)*GRP(Rg, P);
373     return 0.0;
374     }
375 }//eND OF class

```

Listing A.3: varmeror.java

---

```

1 import java.io.Serializable;
2
3 import javax.swing.JDesktopPane;
4
5 /*
6  * Denne klassen er ein subklasse av veke, geometri og arbeidsmedium
7  */
8 public class varmeror implements Serializable {
9
10 /**
11  *
12  */
13 private static final long serialVersionUID = 1L;
14
15 public arbeidsmedium arbeidsmedium;
16 public veke veke;
17 public geometri geometri;
18
19
20 public varmeror()
21 {
22 arbeidsmedium = new arbeidsmedium();
23 veke = new veke();
24 geometri = new geometri();
25 }
26
27 }

```

---

Listing A.4: veke.java

---

```

1 import java.io.Serializable;
2
3 public class veke implements Serializable {
4
5 public double permeabilitet; // Permeabilitet K for arbeidsmediumet
6 public double effektivPoreradius; // Effektiv poreradius reff
7 public int vekeType; // Type refererer til de ulike typene nedenfor.
8 public String vekeNavn; // Navn på veke
9 public String vekeBeskrivelse; // kort beskrivelse av vekestruktur
10
11 public double konduktivitet; // Konduktivitet for vekemateriale ks
12
13 public double porositet;
14
15 // uLIKE TYPER VEKESTRUKTURER
16 static final int CIRCULAR_ARTERY = 0;
17 static final int ANNULAR_CHANNEL = 1;
18 static final int RECTANGULAR_CHANNEL = 2;
19 static final int RECTANGULAR_GROOVES = 3;
20 static final int TRIANGULAR_GROOVES = 4;
21 static final int TRAPEZODIAL_GROOVES = 5;
22 static final int CIRCULAR_GROOVES = 6;
23 static final int METALLNETT = 7; // PLAIN OR SINTERED
24 static final int SINTREDE_VEKER_AV_METALLPULVER = 8; // PLAIN OR SINTERED
25 static final int SINTERED_FELTED_METAL_FIBERS = 9;
26 static final int SCREEN_COVERED_RECTANGULAR_GROOVE = 10;
27 static final int SCREEN_COVERED_TRIANGULAR_GROOVE = 11;
28 static final int FOAM = 12;
29
30 static final String[] vekeTypeNavn = { "IKKE TILGJENGELIG Circual artery",
31 "IKKE TILGJENGELIG Annular channel",
32 "IKKE TILGJENGELIG Rectangular channel",
33 "IKKE TILGJENGELIG Rectangular grooves",
34 "IKKE TILGJENGELIG Triangular grooves",
35 "IKKE TILGJENGELIG Trapezodial grooves",
36 "IKKE TILGJENGELIG Circular grooves", "Metallnetting (flere lag)",
37 "Sintrede veker av metallpulver",
38 "IKKE TILGJENGELIG Sintered feltet metal fibers",
39 "IKKE TILGJENGELIG Screen covered Rectangular groove",
40 "IKKE TILGJENGELIG Screen covered triangular groove",
41 "IKKE TILGJENGELIG Foam" };
42
43 }

```

---

Listing A.5: geometri.java

---

```

1 import java.io.Serializable;
2

```

```

3 public class geometri implements Serializable {
4
5     public double indreRadius; // indre radius for vekestruktur..
6     public double ytreRadius; // Yttre radius for vekestruktur
7     public double lengdeFordamper;
8     public double lengdeAdiabatisk;
9     public double lengdeKondensator;
10    public double tilt; // Vinkel på varmerøret. 0 = kondensator i bunnen.
11    public double beholderTykkelse;
12    public double beholderKonduktivitet;
13
14 }

```

Listing A.6: arbeidsmedium.java

```

1 import java.io.BufferedWriter;
2 import java.io.File;
3 import java.io.FileNotFoundException;
4 import java.io.FileWriter;
5 import java.io.IOException;
6 import java.io.Serializable;
7 import java.util.Scanner;
8
9 public class arbeidsmedium implements Serializable {
10
11     static final int TEMPERATURE = 0;
12     static final int PRESSURE = 1;
13     static final int HFG = 2;
14     static final int TETTHET_LIQUID = 3;
15     static final int TETTHET_VAPOR = 4;
16     static final int VISKOSITET_LIQUID = 5; // dynamisk viskositet
17     static final int VISKOSITET_VAPOR = 6; // dynamisk viskositet
18     static final int K_LIQUID = 7;
19     static final int K_VAPOR = 8;
20     static final int SURFACETENSION_LIQUID = 9;
21     static final int CP_LIQUID = 10;
22     static final int CP_VAPOR = 11;
23
24     public String arbeidsmediumNavn;
25     public double molarMasse; // Mv
26
27     int i;
28     boolean solution_reached;
29
30     public String[] arbeidsmediumKolonneOverskrifter = { "Temp [K]",
31         "Metningstrykk [Kpa]", "Latent varme (hfg) [kJ/kg]",
32         "Tetthet væske [kg/m^3]", "Tetthet damp [kg/m^3]",
33         "Dynamisk Viskositet væske [N*s/m^2]",
34         "Dynamisk Viskositet damp [N*s/m^2]",
35         "Termisk konduktivitet væske [W/m*K]",
36         "Termisk konduktivitet damp [W/m*K]",
37         "Overflatespenning væske [N/m]", "Cp væske [kJ/kg*K]",
38         "Cp damp [kJ/kg*K]" };
39
40     public double[][] arbeidsmediumData = null;
41
42     public double[][] lesFraFil(File arbeidsmediumfil) {
43
44         int rad = 0;
45         double[][] data = new double[10][12]; // datafil for arbeidsmedium må
46         // ha max dette antall data
47         // fordelt i [rader][kolonner]
48         Scanner scanner = null;
49         try {
50             // Create a scanner to read the file
51             scanner = new Scanner(arbeidsmediumfil);
52         } catch (FileNotFoundException e) {
53             System.out.println("Finner ikke arbeidsmediumfil "
54                 + arbeidsmediumfil);
55             // Stop program if no file found
56             return null;
57         }
58
59         try {
60             // first use a Scanner to get each line
61             while (scanner.hasNextLine()) {
62                 // processLine( scanner.nextLine() );
63                 // use a second Scanner to parse the content of each line
64                 Scanner linjescanner = new Scanner(scanner.nextLine());
65                 linjescanner.useDelimiter(",");
66                 if (linjescanner.hasNext()) {

```

```

68         for (int kolonne = 0; kolonne < 12; kolonne++) {
69             try {
70                 data[rad][kolonne] = Double
71                     .parseDouble(linjescanner.next());
72             } catch (NumberFormatException e) {
73                 System.out
74                     .println("Feil ved konvertering fra String til Double (arbeidsmedium)");
75             }
76         }
77     }
78 }
79
80 // (no need for finally here, since String is source)
81 linjescanner.close();
82 } else {
83     System.out
84         .println("Empty or invalid line. Unable to process.");
85 }
86     rad++;
87 }
88 } finally {
89     // ensure the underlying stream is always closed
90     scanner.close();
91 }
92
93 return data;
94 }
95
96 // Read a value from a table of data corresponding to a input value and a
97 // desired output value. Forutsetter at tabell er sortert i stigende eller
98 // minkende verdier
99 public double getProperty(double inputvalue, int input_type, int output_type) {
100
101     // er større enn noe tall i tabell
102     if (inputvalue > arbeidsmediumData[(arbeidsmediumData.length - 1)][input_type]
103         && inputvalue > arbeidsmediumData[0][input_type]) // er større
104         // enn noe
105         // tall i
106         // tabell
107     {
108         if (arbeidsmediumData[(arbeidsmediumData.length - 1)][input_type] >
109             arbeidsmediumData[(arbeidsmediumData.length - 2)][input_type])
110         {
111             return extrapolate(
112                 arbeidsmediumData[(arbeidsmediumData.length - 1)][input_type],
113                 arbeidsmediumData[(arbeidsmediumData.length - 2)][input_type],
114                 arbeidsmediumData[(arbeidsmediumData.length - 1)][output_type],
115                 arbeidsmediumData[(arbeidsmediumData.length - 2)][output_type],
116                 inputvalue);
117         } else {
118             return extrapolate(arbeidsmediumData[1][input_type],
119                 arbeidsmediumData[0][input_type],
120                 arbeidsmediumData[1][output_type],
121                 arbeidsmediumData[0][output_type], inputvalue);
122         }
123     }
124 }
125
126 // er mindre enn noe tall i tabell
127 else if (inputvalue < arbeidsmediumData[(arbeidsmediumData.length - 1)][input_type]
128         && inputvalue < arbeidsmediumData[0][input_type])
129 {
130     if (arbeidsmediumData[(arbeidsmediumData.length - 1)][input_type] >
131         arbeidsmediumData[(arbeidsmediumData.length - 2)][input_type])
132     {
133         return extrapolate(arbeidsmediumData[1][input_type],
134             arbeidsmediumData[0][input_type],
135             arbeidsmediumData[1][output_type],
136             arbeidsmediumData[0][output_type], inputvalue);
137     } else {
138         return extrapolate(
139             arbeidsmediumData[(arbeidsmediumData.length - 1)][input_type],
140             arbeidsmediumData[(arbeidsmediumData.length - 2)][input_type],
141             arbeidsmediumData[(arbeidsmediumData.length - 1)][output_type],
142             arbeidsmediumData[(arbeidsmediumData.length - 2)][output_type],
143             inputvalue);
144     }
145 }
146 }
147 }
148
149 i = arbeidsmediumData.length / 2; // Start to look in the middle
150 // of the array. (More effective

```

```

151 // than to start in one end)
152 solution_reached = false;
153
154 while (!solution_reached) {
155     if (inputvalue == arbeidsmediumData[i][input_type])
156     {
157
158         solution_reached = true;
159         return arbeidsmediumData[i][output_type];
160     } else if (inputvalue > arbeidsmediumData[i][input_type])
161     {
162
163         if (inputvalue < arbeidsmediumData[(i + 1)][input_type])
164         {
165
166             return interpolate(arbeidsmediumData[i][input_type],
167                               arbeidsmediumData[i + 1][input_type],
168                               arbeidsmediumData[i][output_type],
169                               arbeidsmediumData[i + 1][output_type], inputvalue);
170         } else if (inputvalue < arbeidsmediumData[(i - 1)][input_type])
171         {
172
173             return interpolate(arbeidsmediumData[i - 1][input_type],
174                               arbeidsmediumData[i][input_type],
175                               arbeidsmediumData[i - 1][output_type],
176                               arbeidsmediumData[i][output_type], inputvalue);
177         } else if (arbeidsmediumData[i][input_type] > arbeidsmediumData[i + 1][input_type])
178         {
179             i--;
180         } else if (arbeidsmediumData[i][input_type] < arbeidsmediumData[i + 1][input_type])
181         {
182             i++;
183         }
184     }
185 }
186
187 else if (inputvalue < arbeidsmediumData[i][input_type]) // Verdien
188 {
189     if (inputvalue > arbeidsmediumData[(i + 1)][input_type]) // ligger
190     {
191
192         return interpolate(arbeidsmediumData[i][input_type],
193                           arbeidsmediumData[i + 1][input_type],
194                           arbeidsmediumData[i][output_type],
195                           arbeidsmediumData[i + 1][output_type], inputvalue);
196     } else if (inputvalue > arbeidsmediumData[(i - 1)][input_type]) // ligger
197     {
198
199         return interpolate(arbeidsmediumData[i - 1][input_type],
200                           arbeidsmediumData[i][input_type],
201                           arbeidsmediumData[i - 1][output_type],
202                           arbeidsmediumData[i][output_type], inputvalue);
203     } else if (arbeidsmediumData[i][input_type] > arbeidsmediumData[i + 1][input_type])
204     {
205         i++;
206     } else if (arbeidsmediumData[i][input_type] < arbeidsmediumData[i + 1][input_type])
207     {
208         i--;
209     }
210 } else {
211     solution_reached = true;
212 }
213
214 }
215
216 return 0.0;
217 }
218
219 private double interpolate(double lowerValueX, double upperValueX,
220 double lowerValueY, double upperValueY, double middleValueX) {
221     return upperValueY
222         - ((upperValueX - middleValueX) * (upperValueY - lowerValueY) / (upperValueX - lowerValueX));
223 }

```

```
234     }
235
236     private double extrapolate(double lowerValueY, double middleValueY,
237         double lowerValueX, double middleValueX, double upperValueY) {
238         return ((middleValueX - lowerValueX) / (middleValueY - lowerValueY))
239             * (upperValueY - lowerValueY) + lowerValueX;
240     }
241
242     public void skrivTilFil(File filnavn) {
243         BufferedWriter bufferedWriter = null;
244
245         try {
246
247             bufferedWriter = new BufferedWriter(new FileWriter(filnavn));
248
249             String linje = new String();
250
251             for (int y = 0; y < arbeidsmediumData.length; y++) {
252                 for (int x = 0; x < 12; x++) {
253                     linje += Double.toString(arbeidsmediumData[y][x]);
254                     if (x < 11) {
255                         linje += ",";
256                     }
257                 }
258                 bufferedWriter.write(linje);
259                 linje = "";
260                 if (y < (arbeidsmediumData.length - 1)) {
261                     bufferedWriter.newLine();
262                 }
263             }
264
265             } catch (FileNotFoundException ex) {
266                 System.out.println("Kan ikke skrive til fil. Fant ikke fil");
267             } catch (IOException ex) {
268                 System.out.println("IO feil");
269             } finally {
270
271                 try {
272                     if (bufferedWriter != null) {
273                         bufferedWriter.flush();
274                         bufferedWriter.close();
275                     }
276                 } catch (IOException ex) {
277                     ex.printStackTrace();
278                 }
279             }
280
281         }
282     }
```

---



## Tillegg B

# Egenskaper for arbeidsmedium

Følgende arbeidsmedium er brukt som grunnlag for beregninger i program.

Table A.33: Thermophysical Properties of Saturated Potassium

$T$ Temp. K	Potassium, K, Molecular Weight: 39.1, ( $T_b = 1032.2$ K; $T_m = 336.4$ ) [1]										
	$p_v$ Saturation Pressure ( $10^5$ Pa)	$h_{fg}$ Latent Heat (kJ/kg)	$\rho_\ell$ Liquid Density ( $\text{kg}/\text{m}^3$ )	$\rho_v$ Vapor Density ( $10^{-3}$ $\text{kg}/\text{m}^3$ )	$\mu_\ell$ Liquid Viscosity ( $10^{-4}$ $\text{N}\cdot\text{s}/\text{m}^2$ )	$\mu_v$ Vapor Viscosity ( $10^{-7}$ $\text{N}\cdot\text{s}/\text{m}^2$ )	$k_\ell$ Liquid Thermal Conductivity (W/m.K)	$k_v$ Vapor Thermal Conductivity (W/m.K)	$\sigma$ Liquid Surface Tension ( $10^{-3}$ $\text{N}/\text{m}$ )	$c_{p,\ell}$ Liquid Specific Heat (kJ/kg.K)	$c_{p,v}$ Vapor Specific Heat ( $10^{-1}$ kJ/kg.K)
600	0.0009258	2143	766.9	0.69	2.380		43.85	98.2	0.771	0.8194	
700	0.01022	2108	743.3	6.68	1.981		40.72	92.2	0.762	0.9646	
800	0.06116	2068	719.6	36.44	1.707		37.58	86.2	0.761	1.066	
900	0.2441	2023	695.7	134.80	1.507	134	34.45	80.2	0.769	1.116	
1000	0.7322	1970	671.6	380.20	1.354	148	31.32	74.2	0.792	1.121	
1100	1.864	1924	647.3	871.90	1.233	163	28.19	68.2	0.819	1.100	
1200	3.913	1872	622.9	1703.00	1.135	178	25.05	62.2	0.846	1.064	
1300	7.304	1820	598.4	2969.10	1.053	196	22.00	56.2	0.873	1.022	
1400	12.44	1765	573.6	4768.70	0.984	212	19.00	52.0	0.899	0.9796	
1500	20.0	1711	548.8	7062.10	0.925	228	16.00	47.0	0.924		

Figur B.1: Egenskaper for kalium hentet fra *Faghri*[7].

Table A.37: Thermophysical Properties of Saturated Water

T Temp. °C	Water, H <sub>2</sub> O, Molecular Weight: 18.0, (T <sub>b</sub> = 100°C; T <sub>m</sub> = 0.0°C)										
	P <sub>v</sub> Saturation Pressure (10 <sup>5</sup> Pa)	h <sub>fg</sub> Latent Heat (kJ/kg)	ρ <sub>l</sub> Liquid Density (kg/m <sup>3</sup> )	ρ <sub>v</sub> Vapor Density (kg/m <sup>3</sup> )	μ <sub>l</sub> Liquid Viscosity (10 <sup>-4</sup> N·s/m <sup>2</sup> )	μ <sub>v</sub> Vapor Viscosity (10 <sup>-4</sup> N·s/m <sup>2</sup> )	k <sub>l</sub> Liquid Thermal Conductivity (W/m·K)	k <sub>v</sub> Vapor Thermal Conductivity (W/m·K)	σ Liquid Surface Tension (10 <sup>-3</sup> N/m)	C <sub>p,l</sub> Liquid Specific Heat (kJ/kg·K)	C <sub>p,v</sub> Vapor Specific Heat (kJ/kg·K)
20	0.023368	2453.8	999.0	0.01729	10015	88.5	0.602	0.0188	4.182	1.874	
40	0.073749	2406.5	993.05	0.05110	6513	96.6	0.630	0.0201	4.179	1.894	
60	0.199190	2358.4	983.28	0.13020	4630	105.0	0.653	0.0216	4.185	1.924	
80	0.473590	2308.9	971.82	0.29320	3510	113.0	0.669	0.0231	4.197	1.969	
100	1.013250	2251.2	958.37	0.59740	2790	121.0	0.680	0.0248	4.216	2.034	
120	1.985400	2202.9	943.39	1.1100	2300	128.0	0.685	0.0267	4.245	2.124	
140	3.613600	2144.9	925.33	1.96560	1950	135.0	0.687	0.0288	4.285	2.245	
160	6.180400	2082.2	907.44	3.25890	1690	142.0	0.684	0.0313	4.339	2.406	
180	10.02700	2014.0	887.31	5.15970	1493	149.0	0.676	0.0341	4.408	2.615	
200	15.55100	1939.0	865.05	7.86530	1338	156.0	0.664	0.0375	4.497	2.869	

Figur B.2: Egenskaper for vann hentet fra Faghri[7].