

# Fagrapport

# EIT Gullfaks 2005

## ResPunch

Av gruppe 5:

Anders Skog  
Elin Marie Kristensen  
Anders Nødland  
Ørjan Pedersen  
Åsmund Gyldenskog



## Forord

Som en sentral del av faget Eksperter i Team skal det leveres en fagrapport for å dokumentere hvordan oppgaven har blitt løst. Dette er fagrapporten til Gruppe 5 på Gullfakslandsbyen. Underveis i det faglige arbeidet er det mange som har bidratt, både andre studenter og professorer.

Herunder vil vi spesielt takke Prof. Tor Ytrehus ved Institutt for energi- og prosesseteknikk på NTNU, som har bidratt med utledninger og beregninger som gjorde vårt problem løselig.

Vi vil også rette en stor takk til Ole Torsæter ved Institutt for Petroleumsteknologi og anvendt geofysikk, som har tatt seg tid til å føre en dialog om oppgaven og hjulpet oss underveis.

---

Anders Nødland

---

Anders M. Skog

---

Elin Marie Kristensen

---

Åsmund B. Gyldenskog

---

Ørjan Pedersen

Trondheim 09.05.2005

## Innhold

1	Innledning.....	3
2	Oppgaveformulering .....	4
3	Introduksjon til Gulltopp.....	4
3.1	Olje-vann kontakt.....	5
3.2	Reserver og produksjonsprofil .....	5
3.3	Boring og komplettering .....	5
3.4	Gassløft.....	7
4	Ordinære sandskjermer .....	8
5	ResPunch.....	8
5.1	Introduksjon til ResPunch .....	8
5.2	Fordeler og ulemper med ResPunch .....	9
5.2.1	Fordeler .....	9
5.2.2	Ulemper.....	10
6	ResPunch på Gulltopp.....	10
6.1	Ness .....	10
6.2	Packers .....	10
6.3	Vann-olje kontakt.....	11
7	Trykktapsberegninger.....	12
7.1	Trykktap i formasjon.....	12
7.2	Trykktap over ResPunch .....	13
7.2.1	Profiler.....	13
7.2.2	Antagelser.....	15
7.2.3	Grunnlikninger for trykktapsberegninger.....	17
7.3	Produksjonsindeks, PI .....	22
8	Beregning og iterasjon .....	22
8.1	Registrering av variabler og parametere .....	22
8.1.1	ResPunch.....	22
8.1.2	PVT .....	23
8.1.3	Reservoar.....	23
8.1.4	Sone.....	23
8.2	Fremgangsmåte .....	23
8.2.1	Kontinuitet.....	23
8.2.2	Løsning.....	25
9	Simuleringsprogram i Java.....	29
9.1	Forstudium .....	29
9.2	Krav til programmet .....	29
9.3	Konstruksjon .....	30
9.3.1	Klassediagrammet .....	30
9.3.2	Sekvensdiagrammet .....	32
9.4	Implementering .....	32
9.5	Testing.....	32
10	Resultat av simulering.....	33
10.1	Resultater uten optimalisering ResPunch.....	33
10.1.1	Ness 1 .....	34
10.1.2	Ness 2 .....	34

10.1.3	Ness 3 .....	34
10.1.4	Total trykbalanse .....	34
10.2	Optimaliserte resultater med ResPunch .....	34
10.2.1	Ness 1 .....	34
10.2.2	Ness 2 .....	35
10.2.3	Ness 3 .....	36
10.2.4	Total trykbalanse .....	36
10.2.5	Valg av brønntrykk .....	36
10.2.6	Samlede resultater .....	37
11	Feilkilder .....	37
11.1	Formler .....	37
11.2	Regnefeil .....	38
11.3	Feil i antagelser .....	38
11.4	Feil i data .....	38
12	Konklusjon .....	39
13	Vedlegg .....	40
	.....	41
	.....	41
	Tegnforklaring .....	42
13.1	Referanser .....	43
13.1.1	Litteratur .....	43
13.1.2	Nettsteder .....	43
13.1.3	Fagpersonell .....	43

## Figurer

Figur 1: Oversikt over Gullfaksfeltet [1].....	4
Figur 2: Gulltopp med tilhørende sandkontroll enheter [3]. .....	6
Figur 3: Tverrsnitt, Gulltopp [1]. .....	7
Figur 4: Utvalg av sandskjermer (Weatherford inc [9]).....	8
Figur 5: Skisse av ResPunch [4] .....	9
Figur 6: Forflytning vannfront [11].....	11
Figur 7: Radielle strømningshastigheter i ResPunch .....	13
Figur 8: Horisontale strømningshastigheter i ResPunch .....	14
Figur 9: Trykk i ResPunch .....	15
Figur 10: Tverrsnitt av screen [2].....	15
Figur 11: Strømningsprofil gjennom punch-hull .....	16
Figur 12: Radielle avstander i ResPunch.....	17
Figur 13: Klassediagram .....	31
Figur 14: Sekvensdiagram.....	32
Figur 15: Testresultater .....	33
Figur 16: Skisse av ResPunch .....	40
Figur 17: Strømning rundt horisontal brønn .....	40
Figur 18: Skisse, gassløft [10].....	41
Figur 19: Logo, PunchSim .....	41

## 1 Innledning

Sandproduksjon er et av mange problemer knyttet til produksjon av hydrokarboner. Dette problemet oppstår ofte i dårlig konsoliderte reservoarformasjoner, der sand transporteres sammen med hydrokarboner gjennom produksjonsrør til overflaten og inn i eventuelle separasjonsanlegg. Transport av sand fører først og fremst til erosjon og slitasje på produksjonsrør, men kan også lede til fortetning av annet kritisk utstyr. Dersom sand transporteres til overflaten kan dette skape store skader på separatorene og annet prosessutstyr.

Sandskjermer er en konvensjonell metode å forhindre sandproduksjon på. Disse skjermene blir installert i etterkant av komplettering og perforering av brønnen, som regel ved hjelp av brønntraktor, wireline eller coiled tubing, som transporterer skjermen gjennom foringsrør og ned til perforering.

Denne rapporten vil se nærmere på sandkontroll i Gulltoppbrønnen fra Gullfaks-A, som består av to formasjoner, Ness og Tarbert. Ness består av en rekke lagdelte soner med variasjoner i permeabilitet. Dette vil føre til ubalansert innstrømning fra de forskjellige sonene og således et tidlig gjennombrudd av vann i brønnen, som vil hemme optimal hydrokarbonproduksjon fra de andre sonene.

Ved bruk av konvensjonelle skjermene for sandkontroll i Ness fører dette til store ekstra kostnader i følge med mye intervensjons arbeid med plassering av skjermene og perforering. Statoil har derfor bestemt å ta i bruk et nytt konsept for sandkontroll – ResPunch - utviklet av Reslink AS. Mens konvensjonelle skjermene må plasseres i brønnen etter perforering, er ResPunch integrert i casing/produksjonsrør og krever betraktelig mindre tripping ut og inn av brønnen.

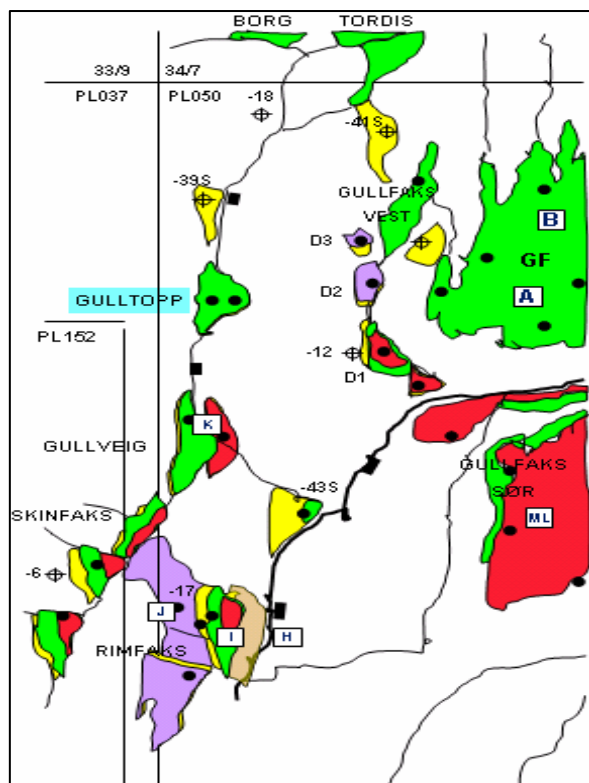
Denne rapporten vil ta for seg ”ResPunch på Gulltopp”. Herunder å se nærmere på trykktap gjennom ”ResPunch” og videre drøfte mulighetene for å variere trykk og strømningsrate fra de forskjellige soner i Ness-formasjonen for å oppnå en mest mulig effektiv og økonomisk produksjonsprofil.

## 2 Oppgaveformulering

Denne rapporten skal gi en anbefaling på komplettering av Gulltopp med ResPunch-sandkontroll enhet. Herunder:

- ✓ Beregne det spesifikke trykkfall over ResPunch modul ved ulike rater, hullstørrelser og antall hull.
- ✓ Finne produktivitet/egenskaper til hver enkelt sone i Ness
- ✓ Undersøke hvordan sonene i Ness ville produsert sammen dersom det ikke var installert ResPunch
- ✓ Optimalisere ResPunch (antall hull og størrelse) for hver enkelt sone slik at de samlet gir høyest mulig rate og utvinningsgrad.
- ✓ Gi en anbefaling på bruk av ResPunch på bakgrunn av de beregninger som er gjort

## 3 Introduksjon til Gulltopp



Figur 1: Oversikt over Gullfaksfeltet [1].

Den planlagte brønnen 34/10-A-32 C, er en oljeproducent i Brent-gruppen i N7-Gulltopp strukturen. Brønnen vil ligge ca. 8 km vest for Gullfaks A installasjonen og er planlagt som en horisontal brønn ved 2450 m TVD MSL eller grunnere i Tarbert formasjonen.

Tarbert formasjonen er forventet å kunne produsere rundt 92 % av de totale reservene, mens Ness har de resterende 8 %.[1]

### 3.1 Olje-vann kontakt

Følgende data er registrert for olje-vann kontakten på Gulltopp:

Formasjon	OWC
Ness2, Ranoch og Etive	2507 m
Tarbert og øvre Ness	2532 m

Tabell 1: Olje-vann data [1].

Svar og beregninger som gjøres i dette prosjektet, er basert på at olje-vann kontakten er ved samme nivå i samtlige deler av Ness. Dette er en forenkling som ikke påvirker prosjektets resultat, da resultatene gjelder for en tidlig produksjonsfase, og ikke i sen fase hvor vannkuttet vil øke. [1]

### 3.2 Reserver og produksjonsprofil

Følgende volumer og utvinnbare reserver er antatt av Statoil for Gulltopp segmentet [1]:

Sone	STOOIP (MSm3)	Utvinnbare res. (MSm3)	RF %
Tarbert 3	2,39	1,27	55
Tarbert 2	3,00	1,69	56
Tarbert 1	1,59	0,89	57
<b>Sum Tarbert</b>	<b>6,98</b>	<b>3,85</b>	<b>53</b>
Ness 3	1,18	0,32	27
Ness 2	0,38	0,01	3
Ness 1	0,05	0,00	0
<b>Sum Ness</b>	<b>1,61</b>	<b>0,34</b>	<b>30</b>
Etive	0,17	0,00	0
Ranoch	0,01	0,00	0
<b>SUM GULLTOPP BRENT</b>	<b>8,8</b>	<b>4,2</b>	<b>47</b>

Tabell 2: Volumer og reserver i Gulltopp segmentet [1].

### 3.3 Boring og komplettering

Det spesielle med Gulltopp brønnen, er hovedsaklig lengden. Brønnen er i følge Statoil beregnet til å ha en total lengde på 10017m MD, og et totalt dyp på 2450 m TVD i reservoar seksjonen. Dette vil bli den lengste brønnen boret fra en offshore installasjon på norsk sokkel noen sinne, og byr derfor på ekstreme utfordringer når det kommer til torque/drag under boring, ECD og installasjon av casing/liner, samt kapasitet på selve plattformen, Gullfaks A. I denne rapporten vil det ikke gås videre inn i detalj på selve boreprosedyren, da dette strekker seg utenfor selve oppgaven.

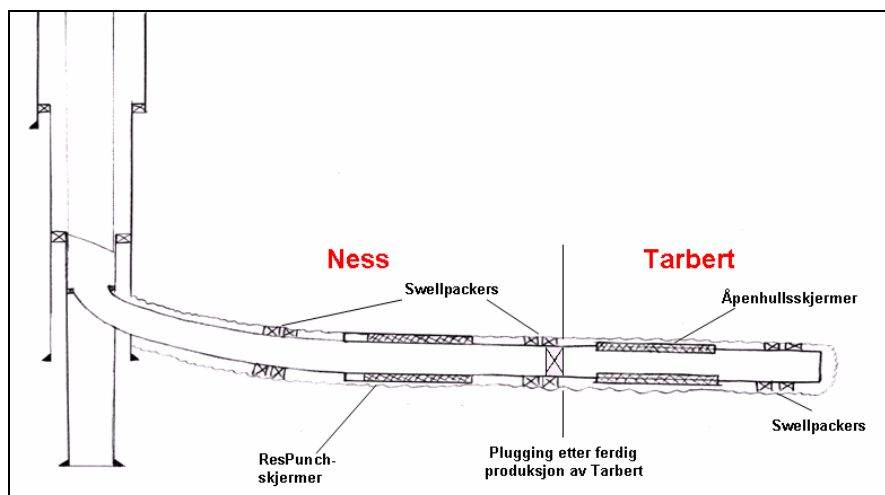
Det som er viktig å ta med, er selve kompletteringen av Gulltopp. I følge Statoil, skal selve reservoarseksjonen på Gulltopp ha en 7" monobore komplettering. Konvensjonelle 4 1/2"

sandskjermer er planlagt brukt i Tarbert, mens Ness er planlagt komplettert med ResPunch som vil bli beskrevet i detalj senere i rapporten.

Gulltopp, hvor ResPunch planlegges brukt, skal etter planen produsere fra flere ulike soner over en gitt tidsperiode. Tarbert formasjonen som antas å inneholde ca 92 % av reservene skal produseres i første omgang. Deretter skal Ness produseres gjennom bruk av ResPunch [1].

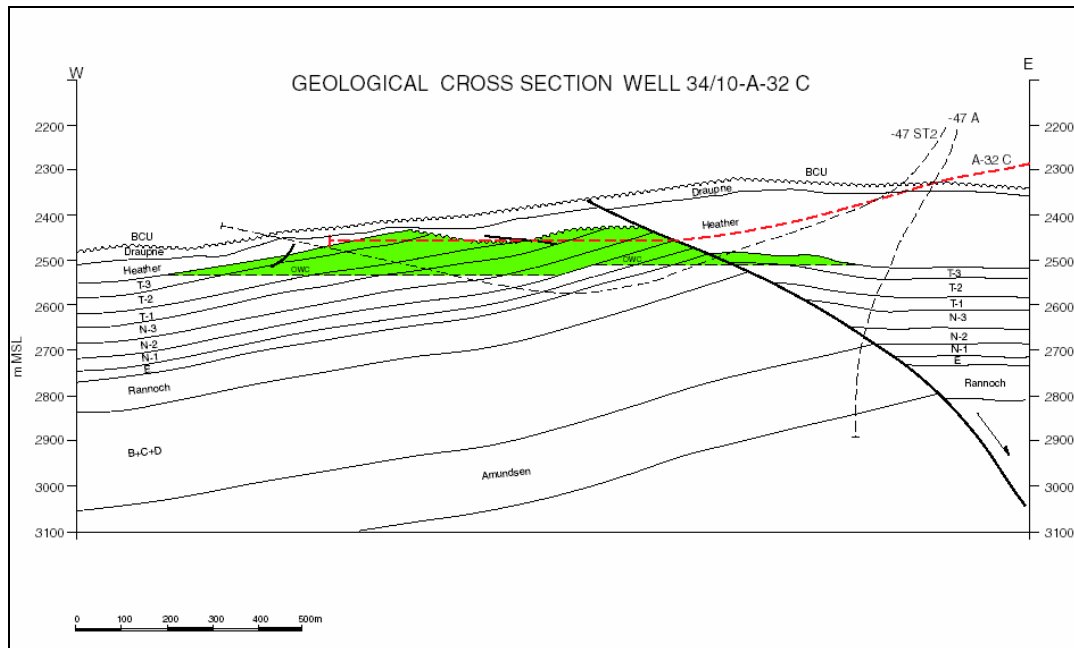
Boreprogrammet for Gulltopp kan oppsummeres slik [2]:

1. Brønnen bores til target, inn i Tarbert 10 017 mMD.
2. Komplettering
  - i. - Vanlig Casing ned til Ness.
  - ii. – Casing med ResPunch gjennom Ness
  - iii. - Konvensjonell Liner/overshot med sandskjerm gjennom Tarbert
3. Perforering av Tarbert, Ness holdes stengt
4. Tarbert produseres
5. Plugg settes mellom Tarbert og Ness
6. ResPunch modulene i Ness åpnes ved hjelp av brønntraktør og puncheverktøy
7. Ness produseres



Figur 2: Gulltopp med tilhørende sandkontroll enheter [3].





Figur 3: Tverrsnitt, Gulltopp [1].

### 3.4 Gassløft

Grunnet brønnlengde og økt vannkutt etter en gitt depletering av reservoaret, vil det være et behov for å lette oljekolonnen. Topside-trykk i kombinasjon med væskens tyngde og trykktap gjennom tubing, gjør det rett og slett for tungt å løfte væskekolonnen. På Gulltopp er dette tenkt løst ved hjelp av gassløft. Gassløftet vil føre til en lettere væskekolonne i tubing, og dermed forlenget produksjonstid.

I denne rapporten beregnes det et brønntrykk tilstrekkelig for å løfte oljen. Dette innebærer hydrostatisk tyngde av fluid, topside-trykk, og tap gjennom tubing. Dette trykket vil danne et kriterium for minimum brønntrykk uten bruk av gassløft. Da denne rapporten hovedsakelig fokuserer på Ness-formasjonen i en tidlig produksjonsfase er ikke bruk av gassløft tatt i betraktning. Gassløft kreves først når vannkuttet er nådd 30-40 %. I følge Statoil inntreffer dette først etter 3-4 år [1].

## 4 Ordinære sandskjermer

Ordinære sandskjermer er i dag brukt i de aller fleste brønner som er potensielle sandprodusenter. Skjermene er relativt enkle innretninger som er installert for å hindre uønsket innstrømning av sand til brønnen. Skjermer blir ofte benyttet i formasjoner med dårlig konsoliderte bergarter og i brønner med høye drawdown.

En ordinær sandskjerm er bygd opp av følgende deler:

- ✓ Perforert liner. Kan leveres i mange størrelser.
- ✓ Netting, lag på lag, montert utenpå liner



Figur 4: Utvalg av sandskjermer (Weatherford inc [9])

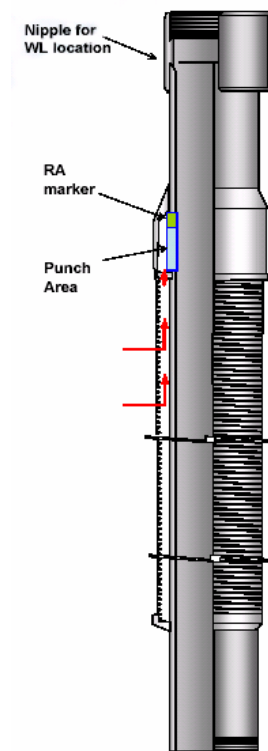
En ordinær sandskjerm er prefabrikkert med en gitt perforeringstetthet fra produsent, dvs. at den er designet for å kunne takle en antatt strømningsrate og trykkendring under produksjon. Det er ikke mulig å endre på innstrømningsarealet på en ordinær sandskjerm etter at den har blitt installert. Skulle en ønske en høyere rate og et endret trykktap etter installasjonen har funnet sted, må det perforeres gjennom skjermen, noe er svært uheldig. Skjermen vil bli skadet, og innstrømning av sand vil kunne forekomme.

## 5 ResPunch

Teknologien ”ResPunch” planlegges brukt i det som vil bli den lengste brønnen som er boret offshore noen sinne. En brønnbane på nærmere 10020m, altså ca. 1 mil, byr på ekstreme utfordringer for Statoil. Å bore lange brønner er svært avanserte operasjoner å utføre, men produksjon fra disse brønnene er også en utfordring.

### 5.1 Introduksjon til ResPunch

ResPunch er som nevnt et helt nytt konsept, utviklet av det norske selskapet ResLink AS. Produktet er ment å være et alternativ til konvensjonelle screens eller sandskjermer som er i bruk i dag. Sandskjermer hindrer sand fra reservoaret å transporteres med oljen opp til overflaten.



Installasjonen av ResPunch på Gulltopp, kan sies å ha følgende avvik i forhold til konvensjonelle sandskjermer [3]:

- ✓ Ingen perforeringer
- ✓ Installeres som en del av casingstrengen, *før* produksjonsstart
- ✓ Innstrømningsraten fra formasjonen kan reguleres ved å variere antall hull ("punches") og størrelsen på disse
- ✓ Ingen eller liten formasjonsskade

Figur 5: Skisse av ResPunch [4]

## 5.2 Fordeler og ulemper med ResPunch

Under arbeidet med ResPunch, er det registrert følgende fordeler og ulemper knyttet til bruk og installering [3],[4]:

### 5.2.1 Fordeler

- ✓ Gjør produksjon i dypere formasjoner mulig, uten å påføre reservoaret skade fra perforering og derav dårligere strømningssegenskaper.
- ✓ Alt i ett: ResPunch kan installeres ved ordinær komplettering, og kan åpnes når dette er ønsket. Dermed spares det tid ved at man slipper mange unødvendige turer inn og ut av hullet.
- ✓ Sonene i en formasjon kan produseres med likt trykktap (ved å variere antall hull samt størrelse). Slik kan man unngå innstrømning av vann fra enkelte soner unødig tidlig.
- ✓ Sandskjermene montert på ResPunch ser ut til å takle mer sand enn vanlige skjermer.
- ✓ Kostnadsbesparende
- ✓ Lite vedlikehold
- ✓ Høyere utvinningsgrad
- ✓ Lavere vannkutt

## 5.2.2 Ulemper

- ✓ Sagging. Boreslam som blir liggende i ringrommet mellom ResPunch og hullvegg, vil etter tid kunne felle ut partikler som tetter formasjonen og hindrer innstrømning. Dette er spesielt aktuelt dersom en ResPunch modul blir stående passiv lenge, og dersom boreslammet som brukes er utsatt for utfelling av partikler
- ✓ Usikkerhet. Uprøvd teknologi.

## 6 ResPunch på Gulltopp

ResPunch er tenkt plassert for første gang i Gulltopp. Det aktuelle området for plassering av modulene er i Ness formasjonen.

### 6.1 Ness

Ness formasjonen er en del av Brent gruppen og er estimert til å inneholde ca 8 % av alle reservene. Ness er lokalisert i en del av Brent med noe dårligere kvalitet enn Tarbert, med innhold av kull og skiferlag som begrenser kommunikasjonen vertikalt enkelte steder. Formasjonen har en dippvinkel på 15 -20 grader [1].

	Porøsitet	Perm.Vert [D]	Perm.Hor [D]	So
Ness 1	0,24	0,1533	1,533	0,96
Ness 2	0,24	0,1227	1,227	0,94
Ness 3	0,24	0,08	0,8	0,99

Tabell 3: Reservoardata, Ness

De tre sonene er fordelt med følgende tykkelse [1]:

Ness 1: 16 m

Ness 2: 38 m

Ness 3: 41 m

Antagelse:

Disse størrelsene er hentet fra Statoils rapport "Gulltopp Brent BoV-BoG". Det verserer her en del ulike teorier om lagtykkelsene, men ovennevnte tall anses som en god approksimasjon.

Da Ness formasjonen ligger med en dippvinkel, antas for derfor at de nevnte tykkelsene på sonene er ment som tykkelse på opprinnelig avsatte sedimenter, og ikke vertikal tykkelse per dags dato.

### 6.2 Packers

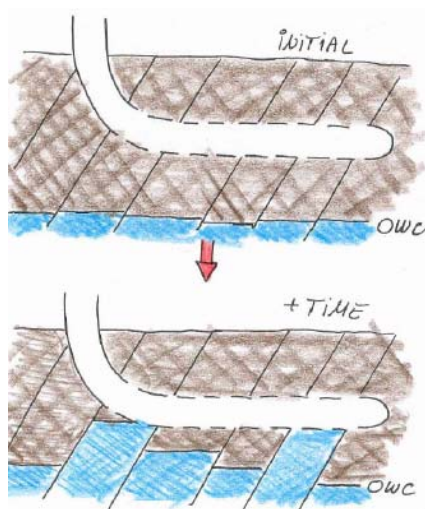
For å oppnå et optimalt resultat med installeringen av ResPunch i den aktuelle brønnen [4], er det nødvendig å isolere de ulike sonene i formasjonen fra hverandre. Ved å plassere Swell-Packers mellom sonene med ulike strømmingsegenskaper og ulik produktivitet, isoleres sonene. Swell-Packers er kraftige gummipakninger som ekspanderer i kontakt med olje.

Pakningene skaper en forsegling langs borehullet, og hindrer væsker i å strømme fritt i ringrommet mellom casing/ResPunch og hullveggen.

Det sees som en nødvendighet at Swell-Packers installeres på gulltopp mellom Ness 1, 2 og 3 for å kunne kontrollere innstrømningen og trykktapet som finner sted i formasjonen. Uten dette svekkes effekten av å installere ResPunch, da trykk og væskestrømmer ikke er isolerte fra hverandre.

### 6.3 Vann-olje kontakt

For å kunne oppnå optimal produksjon fra et reservoar med flere isolerte soner, er det ønskelig å produsere minst mulig vann, og mest mulig hydrokarboner lengst mulig. Noe vann vil produseres under hele produksjonsperioden, men dette er ikke signifikante volumer i dette tilfellet.



Figur 6: Forflytning vannfront [11].

For å unngå at olje-vann kontakten i et lag beveger seg raskere inn mot brønnen enn de andre, er det ønskelig å få denne kontakten til å forflytte seg simultant i alle de aktuelle sonene i reservoaret, i dette tilfellet Ness1 -2 og -3. Ved å gjøre dette, et oppnås et vanngjennombrudd i brønnen ved tilnærmet samme tidspunkt for alle sonene.

For at vannfronten skal forflytte seg likt i alle sonene, må alle sonene produseres med likt trykktap. Sonene kan variere i høyde, ha ulik permeabilitet og porøsitet etc. Derfor skal ikke sonene produseres med lik rate, men med likt tap av trykk, slik at vannfronten forflytter seg likt.

## 7 Trykktapsberegninger

Fluid som ligger i porøse bergarter, vil alltid ha et visst trykk. Trykket som fluidet utsettes for er et resultat av det hydrostatiske trykket fra fluidet selv, samt vekten fra overliggende bergarter og sjøvann (lithostatisk trykk) [5].

Under *produksjon* av et fluid i en porøs bergart med ResPunch installert, vil det tapes trykk i forhold til det opprinnelige reservoartrykket. *Hvor* trykktapet vil finne sted, er en funksjon av følgende:

### 7.1 Trykktap i formasjon

Når fluidene i reservoaret strømmer inn mot brønnen, vil trykk tapes i formasjonen på grunn av de trange poregangene fluidet må passere. I våre beregninger, antar vi følgende for estimering av trykktapet [5],[6]:

#### 1. Radiell strømningsmodell

Den radielle strømningsmodellen passer for de fleste normale brønner. Figur 17 viser denne modellen for en horisontal brønn.

#### 2. Darcys ligninger gjelder

For å kunne uttrykke strømmingen i formasjonen som en funksjon av trykktapet matematisk, bruker vi Darcy's ligning:

$$q = \frac{2\pi kh(p_i - p_{wf})}{\mu \ln\left(\frac{r_i}{r_w}\right)}$$

**Formel 1: Darcy likning**

#### 3. Gravitasjon neglisjeres

Effekten av gravitasjon neglisjeres da innstrømningshastigheten i brønnen er betydelig større enn hastighetseffekten fra gravitasjonen.

#### 4. Innkompressibel væske

Vi anser væsken i reservoaret og i brønnen som en innkompressibel væske. Dette er nødvendig for at Darcy's ligning skal gjelde.

#### 5. Homogen bergart

Det porøse mediet fluidet strømmer gjennom, anses som homogent. Et heterogent medium, ville ha gjort de fleste beregningene kompliserte med tanke på varierende porøsitet og permeabilitet innad i sonene.

#### 6. Begrenset strømningsperiode

I beregningene som utføres, sees det kun på en kort tidsperiode av produksjonen. Det vil derfor kunne oppleves at brønntrykket faller i likevekt med det opprinnelige reservoartrykket.

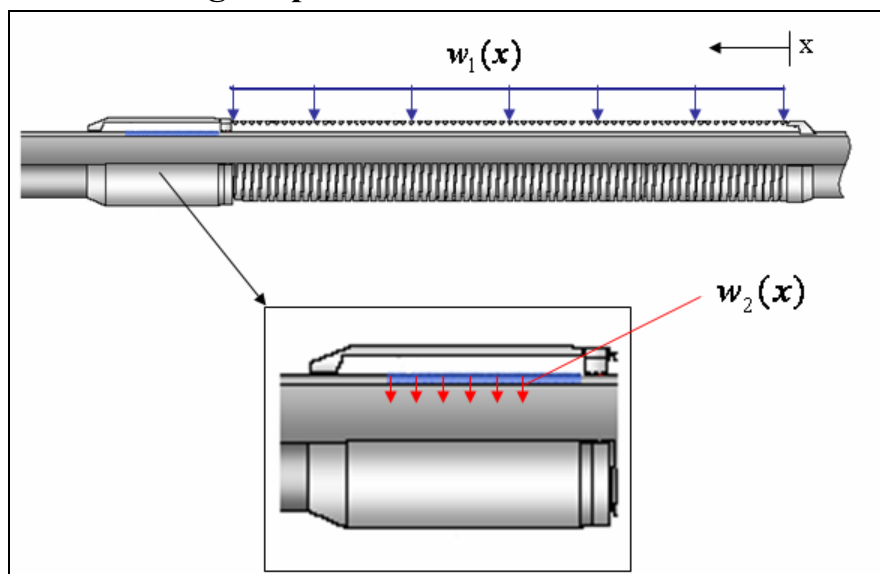
## 7.2 Trykktap over ResPunch

Et sentralt mål for denne rapporten og det ferdige produktet er at beregninger som gjøres skal basere seg på formler som beskriver forholdene mest mulig reelt. Da ResPunch er et helt nytt konsept viser det seg å være en stor utfordring å komme frem til en matematisk modell som beskriver situasjonen mest mulig nøyaktig.

### 7.2.1 Profiler

Sett fra et strømningsteknisk standpunkt vil det være en rekke ulike trykk- og hastighetsprofiler i ResPunch. Som nevnt ovenfor er det et mål i seg selv å avdekke og beskrive disse analytisk mest mulig reelt.

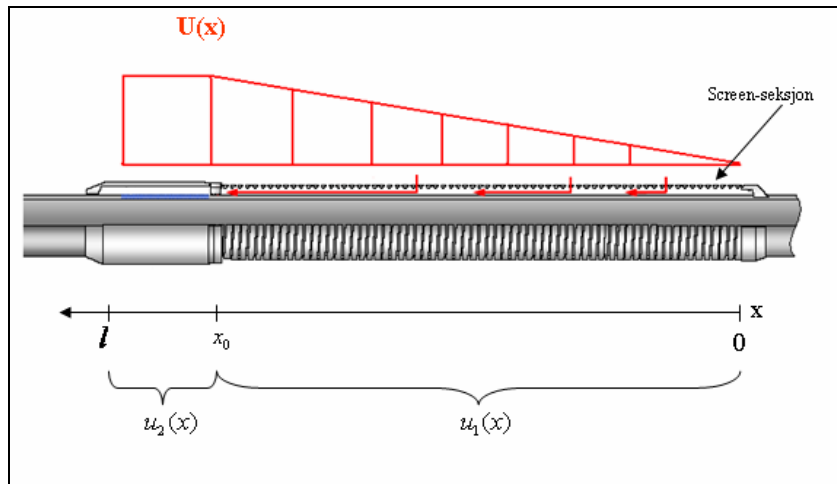
#### 7.2.1.1 Hastighetsprofiler



Figur 7: Radiale strømningshastigheter i ResPunch

Figuren over beskriver de vertikale strømningshastighetene i Respunch. Innstrømningen fra reservoar/borehull, inn gjennom screen -  $w_1(x)$  vil variere som følge av varierende trykk på innsiden av screen. Etersom trykket vil minske langs screen-lengden, i økende x-retning, er det nærliggende å anta at  $w_1(x)$  vil øke med økende x.

Strømningen gjennom perforeringsseksjonen (choke house) -  $w_2(x)$  - vil derimot avta noe i økende x-retning. Denne hastigheten vil i stor grad avhenge av størrelse og antall perforeringshull (antall punch).



Figur 8: Horisontale strømningshastigheter i ResPunch

Den horisontale hastigheten  $U(x)$  deles inn i 2 deler:  $u_1(x)$  - hastighet gjennom screenseksjon, og  $u_2(x)$  - hastighet gjennom punch-modulen.

På innsiden av screen vil hastigheten  $U(x)$  strømme gjennom ringrommet som er bestemt av den radielle avstanden mellom produksjonsrør (casing) og screen. Rent intuitivt kan det fort anslås at  $U(x)$  vil akselerere med økende  $x$  langs hele screen-lengden for deretter å avta noe i perforeringsseksjonen (choke house).  $U(x)$  vil her øke som følge av minkende trykk langs screen-lengden.

### 7.2.1.2 Trykkprofiler

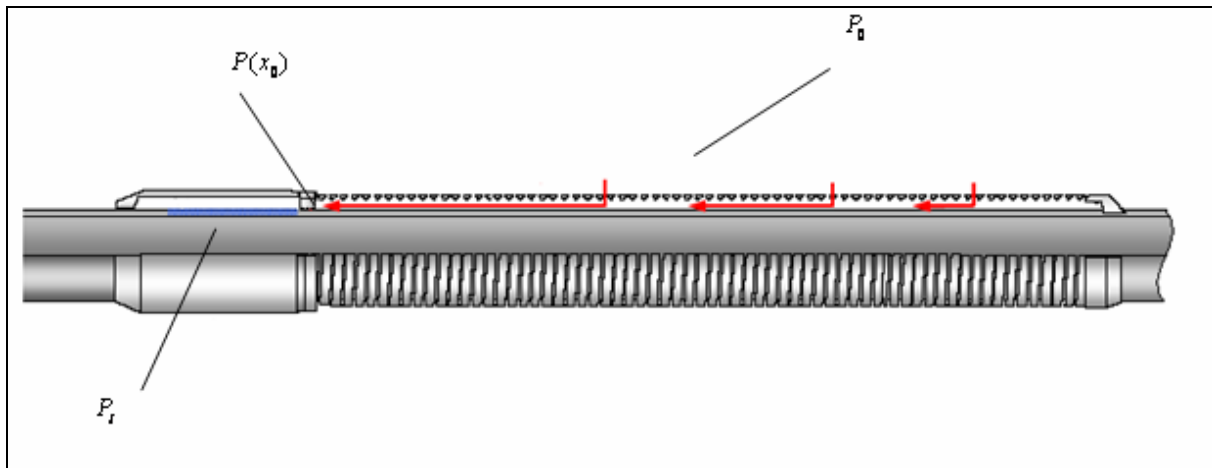
Det er generelt komplisert å forutsi konkret hvordan trykktapet i Respunch vil utspille seg. Det kan for øvrig anslås tre trykktapsseksjoner:

- i. innstrømming gjennom screen
- ii. strømming i ringrom mellom screen og produksjonsrør
- iii. utstrømming gjennom perforeringsseksjon

Innstrømmingstapet gjennom screen ansees for å være det minste bidraget. Trykktapet gjennom ringrommet mellom screen og produksjonsrør, fra  $x = 0$  til  $x = x_0$ , vil være betraktelig større. Her vil trykket minske som følge av friksjon på strømming fra screen- og rørvegg, samt varierende innstrømningshastighet  $w_1(x)$ . Den største tapskilden gjennom ResPunch forventes å finnes i perforeringsseksjonen,  $x = x_0$  til  $x = l$ . Dette trykktapet kan for øvrig reguleres i følge med det totale utstrømningsarealet (antall perforeringshull og -størrelse).

Det er et konkret mål med denne oppgaven å beskrive trykktapet i de respektive seksjonene for deretter å avdekke et eksakt trykktap mellom trykket på utsiden av screen (hullvegg),  $P_0$ , og trykk i produksjonsrør,  $P_i$ .





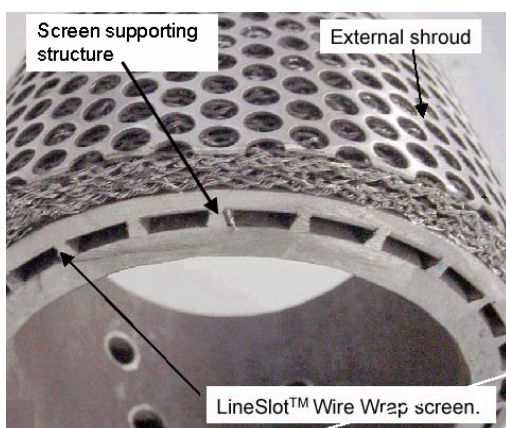
Figur 9: Trykk i ResPunch

## 7.2.2 Antagelser

Da ResPunch er en komplisert konstruksjon er det under beregningene foretatt antagelser for å gjøre problemet mer håndterlig. Antagelsene er tatt på bakgrunn av begrenset tidsramme og omfang av prosjektet. Til tross for disse anses grunnlikningene som beskriver problemet som reelt gjeldende for ResPunch sitt praktiske bruksområde.

### 1 Støttepilarer mellom screen og produksjonsrør

Ved beregning av trykktap gjennom screenseksjon er den støttende konstruksjonen mellom screen og produksjonsrør neglisjert. For å forenkle beregningene betraktes derfor den radielle avstanden som et åpent ringrom der strømmingen kun er i kontakt med innervegg av screen og yttervegg av produksjonsrør. I realiteten vil den støttende konstruksjonen påføre strømmingen større friksjon og følgelig større trykktap, men dette antas som neglisjerbart relativt til trykktapet mellom screen- og rørvegg.



Figur 10: Tverrsnitt av screen [2]

### 2 Radiell strømming i ringrom

I ringrommet mellom screen og produksjonsrør antas det at strømmingen ikke vil ha noen radiell strømmings hastighet. Da den radielle avstanden er så liten vil en eventuell radiell strømming være minimal. I tillegg vil strømmingen i definert x-retning være totalt dominerende. Altså er  $u(y) = 0$ .

### 3 Stasjonær strømning

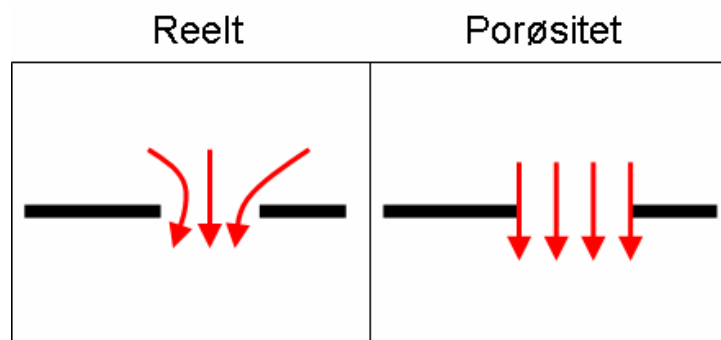
Problemet betraktes som stasjonært. Dette blir noe feil sett opp mot reservoartrykket, som vil endres over tid. Ved å betrakte ResPunch for en tidlig produksjonsfase kan for øvrig rammebetingelsene ansees for å være tilnærmet konstant.

### 4 Enfase strømning

Over tid vil det nok produseres både gass og kondensat fra reservoaret. Tidlig i produksjonen vil det hovedsakelig kun produseres olje, da trykket i reservoaret ligger over boblepunktet. Beregninger gjort gjennom ResPunch og produksjonsrør tar derfor utgangspunkt i ren enfase strømning av olje.

### 5 Porøsitet

For å beregne innstrømning gjennom skjerm og utstrømning gjennom perforeringshull benytter vi oss av en såkalt porøsitetfaktor  $\phi_{1,2}$  [7]. Bruk av porøsitet tilsier at vi har et totalt innstrømnings- eller utstrømningsareal som fylles av strømmingen. Da det spesielt i perforeringsseksjonen vil være relativt store hastigheter vil strømmingen bøye av gjennom hullene og derfor ikke fylle hele hullarealet. Alternativet til dette ville ha vært å bruke dyseteori, men beregninger med porøsitet viste seg å være atskillig mer pålitelige for dette tilfellet.



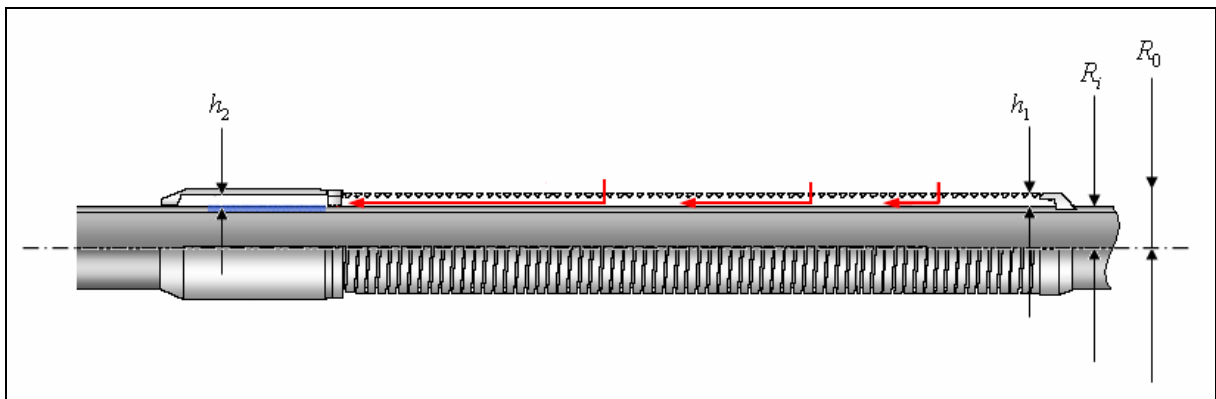
Figur 11: Strømningsprofil gjennom punch-hull

Formler som brukes i beregningsdelen tar utgangspunkt i at det strømmer gjennom hele hullstørrelsen, men disse kan enkelt korrigeres ved å bruke et forholdstall foran porøsiteten, som beskriver hvor mye av hullet som vil fylles. For å få en fullgod tilnærming på dette problemet vil det være nødvendig å kjøre tester i laboratorium, for å finne ut hvor mye av perforeringshullet som fylles.

### 7.2.3 Grunnlikninger for trykktapsberegninger

For å beskrive trykktapet over ResPunch analytisk benyttes Newtons 2.lov uttrykt som Navier-Stokes teorem, Bernoulli's likning og en global massebevarelse (kontinuitetslikning) [7],[8]. I disse ligningene brukes forskjellige radielle avstander som beskriver ulike deler av ResPunch, hvor på:

- $h_1$  – mellom skjerm og yttervegg produksjonsrør
- $h_2$  – mellom innside "choke house" og yttervegg produksjonsrør
- $R_i$  – ytre radius for produksjonsrør
- $R_0$  – ytre radius for sandskjerm



Figur 12: Radielle avstander i Respunch

Ifølge [7] og [8] har vi følgende grunnlikninger for innløp og utløp:

**Basislikning for innstrømning:**

$$\rho u \frac{\partial u}{\partial x} = -\frac{\partial p}{\partial x} + \frac{1}{2} \rho u^2 f(\text{Re}) \frac{1}{D_h} \quad , \quad 0 < x \leq x_0$$

$$\frac{1}{2} \rho u^2(x) = -p(x) + p(0) + \int_0^x \frac{1}{2} \rho u^2(x) f(\text{Re}) \frac{\partial x}{D_h}$$

Har som initialbetingelse at  $p(0) = P_0$  og følgelig:

$$\frac{1}{2} \rho u^2(x) = -p(x) + P_0 + \int_0^x \frac{1}{2} \rho u^2(x) f(\text{Re}) \frac{\partial x}{D_h}$$

**Formel 2**

Samme fremgangsmåte gir utstrømning,  $x_0 \leq x \leq l$

$$\frac{1}{2} \rho u_2^2 - \frac{1}{2} \rho u_0^2 = -P(x) + P(x_0) - \int_{x_0}^x \frac{1}{2} \rho u^2 f(\text{Re}) \frac{dx}{D_h}$$

**Formel 3**

### 7.2.3.1 Strømningshastigheter

Innstrømming gjennom skjerm og utstrømming gjennom perforering:

$$w_1 = \left[ 2 \frac{(P_0 - P(x))}{\rho} \right]^{\frac{1}{2}}, \quad w_2 = \left[ 2 \frac{(P(x) - P_i)}{\rho} \right]^{\frac{1}{2}}$$

**Kontinuitet:** innstrømming skjerm = gjennomstrømming ringrom

$$2\pi R_0 h_1 u_1(x) = \int_0^x w_1(x) 2\pi R_0 \phi_1 dx$$

$$u_1(x) = \int_0^x w_1(x) \phi_1 \frac{dx}{h_1}$$

**Formel 4**

Kontinuitet for utstrømming gir likeledes strømningshastigheten i punch modul:

$$u_2(x) = u_0 - \int_{x_0}^x w_2(x) \phi_2 \frac{dx}{h_2}$$

**Formel 5**

Videre settes  $x = x_0$  inni formel 2 og  $x = l$  i formel 3 og summerer **formel 2 + formel 3** som resulterer i den totale systemligningen:

$$P_0 - P_i = \int_x^{x_0} \frac{1}{2} \rho u_1^2 f \frac{dx}{D_h} + \int_{x_0}^l \frac{1}{2} \rho u_2^2 f \frac{dx}{D_h}$$

**Formel 6: Basal systemligning**

$$u_1(x) = \int_0^x \left[ 2 \frac{(P_0 - P(x))}{\rho} \right]^{\frac{1}{2}} \phi_1 \frac{dx}{h_1}$$

$$u_2(x) = u_0 - \int_{x_0}^l \left[ 2 \frac{(P(x) - P_i)}{\rho} \right]^{\frac{1}{2}} \phi_2 \frac{dx}{h_2}$$

$$u_0 = \int_0^{x_0} \left[ 2 \frac{(P_0 - P(x_0))}{\rho} \right]^{\frac{1}{2}} \phi_1 \frac{dx}{h_1}$$

Formel 6 er her en ikke-lineær integralligning for trykket  $P(x)$ . Løser denne med global kontinuitet som bibetingelse.

Global kontinuitet gir:

$$\int_0^{x_0} w_1(x) \phi_1 2\pi R_0 dx = \int_{x_0}^l w_2(x) \phi_2 2\pi R_i dx, \text{ Gjør tilnærmingen } R_0 \cong R_i.$$

$$\int_0^{x_0} \left[ 2 \frac{(P_0 - P(x))}{\rho} \right]^{1/2} \phi_1 dx = \int_{x_0}^l \left[ 2 \frac{(P(x) - P_i)}{\rho} \right]^{1/2} \phi_2 dx$$

**Formel 7**

### **Friksjonsfaktoren f(Re)**

Friksjonsfaktoren f(Re), som bestemmes av Reynolds tall er en langsom funksjon av u og x, eks. Blasius

$$f(\text{Re}) = \frac{0,3}{\sqrt[4]{\text{Re}}}, \quad \text{Re} = \frac{\rho D_h V}{\mu}$$

f(Re) kan derfor settes konstant og justeres iterativt etter at P(x) og u(x) er beregnet. Dette gjøre fortrinnsvis i eget regneark i MS Excel.

### 7.2.3.2 Approksimasjon av trykk P(x)

For å kunne bestemme trykket gjøres en enkel approksimasjon der det antas at trykket P(x) varierer lineært langs ResPunch:

$$P_0 - P(x) = \Delta p_1 \frac{x}{x_0}$$

**Formel 8: Lineært trykktap i innstrømningsseksjon**

hvor  $\Delta p_1 = P_0 - P(x_0)$

$$P(x) - P_i = \Delta p_2 \frac{x - x_0}{l - x_0}$$

**Formel 9: Lineært trykktap i utstrømningsseksjon**

hvor  $\Delta p_2 = P(x_0) - P_i$

Setter uttrykk for trykktap – formel 8 og formel 9- inn i strømningshastigheter – formel 3 og formel 5 - og får:

$$u_1^2(x) = \frac{2\Delta p_1}{\rho} \left( \frac{2}{3} \phi_1 \right)^2 \frac{x}{x_0} \left( \frac{x}{h_1} \right)^2$$

$$u_2^2(x) = u_0^2 - 2u_0 \left( \frac{2\Delta p_2}{\rho} \right)^{1/2} \frac{\phi_2}{h_2} F(x) + \frac{2\Delta p_2}{\rho} \left( \frac{\phi_2}{h_2} \right)^2 F^2(x) \quad , \quad \text{hvor} \quad F(x) = \frac{2}{3} \frac{(x - x_0)^{3/2}}{(l - x_0)^{1/2}}$$

Setter strømningshastighetene inn i basal systemligning – formel 6 - og integrerer ut. Dette resulterer i en ny basal systemligning på formen:

$$\boxed{P_0 - P_i = \Delta p_0 = A\Delta p_1 + B\Delta p_1 - \Delta p_1^{1/2} \Delta p_2^{1/2} + D\Delta p_2}$$

Koeffisienter A, B, C, D er dimensjonsløse og gitt ved:

$$A = f(\text{Re}) \left( \frac{2}{3} \phi_1 \right)^2 \frac{1}{4} \left( \frac{x_0}{h_1} \right)^2 \frac{x_0}{D_h}$$

$$B = f(\text{Re}) \left( \frac{2}{3} \phi_1 \right)^2 \left( \frac{x_0}{h_1} \right)^2 \frac{l - x_0}{D_h}$$

$$C = 2f(\text{Re}) \frac{2}{3} \phi_1 \frac{4}{15} \phi_2 \frac{x_0}{h_1} \frac{(l - x_0)^2}{D_h h_2}$$

$$D = \frac{1}{3} f(\text{Re}) \phi_2^2 \frac{(l - x_0)}{D_h} \frac{(l - x_0)^2}{h_2^2}$$

Alle koeffisienter avhenger her av kjente parametere. Eneste variabel for ResPunch vil være porøsitet  $\phi_2$  som vil være med å variere trykktapet gjennom perforeringsseksjonen.

Ved å benytte trykkapprosimasjonen formel 8 og formel 9 i formel 7 får vi følgende uttrykk for global massebevarelse:

$$\left( \frac{2\Delta p_1}{\rho} \right)^{1/2} \phi_1 2\pi R_0 \int_0^{x_0} \left( \frac{x}{x_0} \right)^{1/2} dx = \left( \frac{2\Delta p_2}{\rho} \right)^{1/2} \phi_2 2\pi R_i \int_{x_0}^l \left( \frac{x - x_0}{l - x_0} \right)^{1/2} dx$$

Integrerer opp og får uttrykk for strømningsrate:

$$q_{\text{innstrømning}} = \left( 2 \frac{(P_0 - P_{x_0})}{\rho} \right)^{1/2} \phi_1 \pi D_{\text{ytre}} \frac{2}{3} x_0$$

$$q_{\text{perforering}} = \left( 2 \frac{(P_{x_0} - P_i)}{\rho} \right)^{1/2} \phi_2 \pi D_{\text{indre}} \frac{2}{3} (l - x_0)$$

Utrykkene for strømningsratene brukes i samsvar med Darcy's lov. Dette vil gi et uttrykk for  $P_0$  som vil variere som funksjon av reservoartrykk  $P_{\text{res}}$ , trykk i produksjonsrør  $P_i$  og antall hull i perforeringsseksjon som bestemmer porøsitet  $\phi_2$ . Alle variable er kjente størrelser og problemet er løst.

For å finne mer eksakte trykk og hastigheter i ResPunch benyttes formel 3 og formel 5 etter å ha bestemt  $P_0$ . Disse vil gi eksakte strømningshastigheter i et vilkårlig punkt i ResPunch, som videre gjør det mulig å bestemme eksakte trykk ved hjelp av den lineære approksimasjonen.

Det er viktig å understreke at en bedre approksimasjon kan fås ved å utvikle formel 8 og formel 9 videre:

$$P_0 - P(x) = \Delta p_1 \left( a \frac{x}{x_0} + b \left( \frac{x}{x_0} \right)^2 \right)$$

$$P(x) - P_i = \Delta p_2 \left( c \frac{x - x_0}{l - x_0} + d \left( \frac{x - x_0}{l - x_0} \right)^2 \right)$$

Dette er ikke gjort i denne rapporten, men anbefales dersom man søker helt korrekte verdier av problemet. For øvrig kan det påpekes det at løste ligningssett med enkel trykkapproksimasjon, gir langt på vei tilfredsstillende resultater for trykkberegninger i ResPunch.

### 7.3 Produksjonsindeks, PI

Produksjonsindeks. PI, estimeres på følgende måte [5]:

$$PI = \frac{q}{\Delta P}, \text{ hvor } q \text{ er den totale strømningsraten fra Ness [Sm}^3/\text{d] og } \Delta P \text{ er det totale trykktap.}$$

PI er også gitt som produksjonsindeks for kort horisontal brønn:

$$PI = \frac{2\pi k_p h}{\mu\beta \left[ \ln\left(\frac{4r_e}{L_w}\right) - \frac{3}{4} + \frac{\beta h}{L_w} \left[ \frac{2h\beta}{2\pi r_w (\beta + 1)} \right] + S \right]}$$

## 8 Beregning og iterasjon

For å oppnå konkrete verdier for trykktapet gjennom ResPunch er det nødvendig å knytte en sammenheng mellom allerede kjente verdier. Denne oppnås gjennom kontinuitetsligningene i de forskjellige delene av systemet. Strategien er således å benytte disse sammenhengene i et dataprogram, fortrinnsvis Java, som effektivt utfører nødvendige beregninger og returnerer ønskede parametere.

### 8.1 Registrering av variabler og parametere

Det er nødvendig å laste inn alle data som omhandler både selve ResPunch modulen [4], reservoaret [1], PVT data [1], og spesifikasjoner for hver enkelt sone i reservoaret [1]:

#### 8.1.1 ResPunch

- ✓ Ytre diameter seksjon
- ✓ Indre diameter seksjon



- ✓ Indre diameter choke
- ✓ Lengde innstrømningsseksjon
- ✓ Lengde choke
- ✓ Total lengde
- ✓ Diameter hull i punch

### 8.1.2 PVT

- ✓ Viskositet
- ✓ Tetthet
- ✓ Boblepunkt
- ✓  $B_o$ -oljeformasjonsvolumfaktor

### 8.1.3 Reservoar

- ✓ Dippvinkel reservoar
- ✓ Trykk,  $P_{res}$
- ✓ Dreneringsradius (antatt lik 200m)
- ✓ Brønnradius
- ✓ Dybde TVD
- ✓ Brønnlengde (totalt)
- ✓ Diameter, Tubing

### 8.1.4 Sone

- ✓ Høyde
- ✓ Vertikal permeabilitet
- ✓ Horisontal permeabilitet

## 8.2 Fremgangsmåte

Ved å knytte sammen kontinuiteter oppnår vi til slutt et annengradsuttrykk for veggtrykket på utsiden av skjermen,  $P_0$  [7]. Dette trykket vil variere i samsvar med brønntrykket,  $P_i$ , som varieres mellom 0-313 bar.

### 8.2.1 Kontinuitet

Vi tar utgangspunkt i følgende kontinuiteter av massestrøm i systemet [7], [5]:

1. Massestrøm fra formasjonen

$$q_1 = \frac{2\pi kh(p_{res} - p_{wf})}{\mu \ln\left(\frac{r_i}{r_w}\right)}$$

2. Massestrøm gjennom skjerm

$$q_2 = \sqrt{2 \left( \frac{P_0 - P_{x_0}}{\rho} \right)} \phi_1 \frac{2}{3} x_0 (\pi d_{outer})$$

3. Massestrøm gjennom dyser i choke

$$q_3 = \sqrt{\frac{2}{\rho} (P_{x_0} - P_i)} \phi_2 \frac{2}{3} (l - x_0) (\pi d_{inner})$$

Av dette kan vi løse ut følgende konstanter:

$$4. \quad A = \frac{2\Pi kh}{\mu \ln\left(\frac{r_i}{r_w}\right)}$$

$$5. \quad B = \frac{2}{\rho} \left( \phi_1 \frac{2}{3} x_0 \right)^2 (\pi d_{outer})^2$$

$$6. \quad C = \frac{2}{\rho} \left( \frac{2}{3} (l - x_0) \right)^2 (\pi d_{inner})^2$$

Ved å anta at  $q_1 = q_2 = q_3$ , får vi følgende:

7. Setter  $q_1 = q_2$  og kvadrerer

$$A^2 = (P_{res}^2 - 2P_{res}P_0 + P_0^2) = B(P_0 - P_{x_0})$$

$$\boxed{P_{x_0} = P_0 - \frac{A^2}{B} (P_{res}^2 - 2P_{res}P_0 + P_0^2)}$$

**Formel 10**

8. Setter  $q_1 = q_3$  og kvadrerer

$$A^2 = (P_{res}^2 - 2P_{res}P_0 + P_0^2) = C\phi_2^2(P_{x_0} - P_i)$$

$$\boxed{P_{x_0} = P_i - \frac{A^2}{C\phi_2^2} (P_{res}^2 - 2P_{res}P_0 + P_0^2)}$$

**Formel 11**

9. Setter formel 10 = formel 11 og løser med hensyn på  $P_0$ :

$$P_0 - \frac{A^2}{B} (P_{res}^2 - 2P_{res}P_0 + P_0^2) = P_i - \frac{A^2}{C\phi_2^2} (P_{res}^2 - 2P_{res}P_0 + P_0^2)$$

$$\underbrace{-P_0^2 \left( \frac{A^2}{B} + \frac{A^2}{C\phi_2^2} \right)}_a + \underbrace{\left( 1 + \frac{2A^2P_{res}}{B} + \frac{2A^2P_{res}}{C\phi_2^2} \right) P_0}_b - \underbrace{\left( P_i + \frac{A^2P_{res}^2}{C\phi_2^2} + \frac{A^2P_{res}^2}{B} \right)}_c = 0$$

**Formel 12**

10. Løser likningen for  $P_0$

$$P_{01,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Da negativt fortegn foran rottegn returnerer verdier for  $P_0$  som er høyere en initielt reservoartrykk vil kun følgende gjelde for vårt tilfelle:

$$P_{01,2} = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

**Formel 13**

## 8.2.2 Løsning

Ved å bruke resultatet fra kontinuitetsbalansen (formel 13) kan de ulike trykktapene beregnes ved å benytte varierte verdier for  $P_i$ , altså brønntrykket. Reservoartrykket er konstant (313 bar) i det strømmingen starter (stasjonært).  $\phi_2$ , som er uttrykt ved  $A_{hull}/A_{choke}$ , kan også varieres ved å variere antall hull. Et økt antall hull, vil automatisk gi et lavere trykktap for hver enkelt sone, da innstrømningsarealet øker proporsjonalt med antall hull.

Til å begynne med, velges det å tildele hver enkelt sone en ResPunch modul med et gitt antall hull, dvs. 20 stk. Man får da en liste med ulike trykktap for ResPunch for de gitte brønntrykkene. Denne prosessen gjennomføres for alle sonene - Ness1, -2 og -3 - og plasserer resultatene i en tabell, med brønntrykkene som referanse.

SONE 1			SONE 2			SONE 3		
# hull	20	stk/seksjon	# hull	20	stk/seksjon	# hull	20	stk/seksjon
Antall skjermer	1	stk	Antall skjermer	1	stk	Antall skjermer	1	stk
Reelt # hull	20	stk/seksjon	Reelt # hull	20	stk/seksjon	Reelt # hull	20	stk/seksjon
Reelt # skjerme	1	stk	Reelt # skjermer	1	stk	Reelt # skjermer	1	stk
Lengde Punch	12,5	m	Lengde Punch	12,5	m	Lengde Punch	12,5	m
A	1,5908E-07		A	6,36631E-08		A	8,30163E-08	
B	0,000213396		B	0,000213396		B	0,000213396	
C	0,000904366		C	0,000904366		C	0,000904366	
Phi 2	0,001935238		Phi 2	0,001935238		Phi 2	0,001935238	

Det er nå en relativt stor mengde data som skal sammenlignes, men man behøver ikke alle disse. Ved å velge sonen med det minste trykktapet gjennom punch, har vi et utgangspunkt.

Vi velger denne på basis av følgende:

- ✓ Sonen med det laveste trykktapet i punch vil ha den høyeste masseraten, gitt en konstant høyde og permeabilitet for sonene. PVT data er antatt identiske for alle Ness formasjonene.
- ✓ For å oppnå maks rate fra reservoaret, settes sonen med det minste trykktapet til å være dominant.
- ✓ Gjennom å variere antall hull og ResPunch-moduler i de andre sonene, reguleres trykktapet her til et nivå som er tilnærmet likt den dominante sonen.

### 8.2.2.1 Betingelser for strømning

For at en væske skal kunne strømme gjennom et rørsystem, må det være en trykkforskjell mellom innløp og utløp for systemet. Skulle trykkforskjellen være liten eller ikke-eksisterende, vil det heller ikke kunne foregå noen strømningsprosess.

På Gulltopp har vi en lang brønn på 10 017 meter. I tillegg ligger det aktuelle strømningsområdet i Ness ca 2450 m under havoverflaten, samt at prosessanlegget på GFA krever et leveringstrykk på 60 bar. Gitt disse betingelsene, må brønntrykket i Ness mestre følgende [1]:

- ✓ Topside leveringstrykk, 60 bar.
- ✓ Trykktap i tubing fra Ness til topside
- ✓ Trykktap grunnet hydrostatisk trykkforskjell mellom reservoar og plattform

Kriterium for strømning fra Ness blir da:

$$P_{tubing}^{min} = P_{topside} + P_{hydrostatic} + \Delta P_{friction}$$

<b>P-topside</b>	60 bar
<b>Depth</b>	2450 m
<b>P<sub>head</sub></b>	225,838 bar
	23283805 Pa

Tabell 4: Trykkkrav uten friksjon

Dette vil altså si at produksjon fra Ness skal kunne forekomme, må trykket i tubing overstige minimum 226 bar, uten friksjon. Dette er en fast verdi, mens friksjonstrykktapet vil være en funksjon av masseraten fra reservoaret. Vi får altså et ulikt trykktap oppover i tubing for hvert tubingtrykk vi setter i tabellen.

Antagelser:

- i. Som tidligere nevnt, vil det bli installert gassløft for å redusere det hydrostatiske trykktapet. Under konstruksjonen av PunchSim og beregningene gjort ellers, er ikke

- gassløft lagt inn som en parameter. Dette gjør at det brønntrykket som kreves for produksjon, er noe mindre i virkeligheten enn hva våre beregninger tilsier.
- ii. Produksjonen på Gulltopp som en "Worst Case". Beregningene utføres på en slik måte at brønnen skal kunne strømme selv om gassløft funksjonen blir satt ut av funksjon. Dette medfører avvik i forhold til hva Statoil opererer med i sine beregninger.
  - iii. Det er ikke tatt hensyn til trykkendringen i væsken oppover i brønnen. Som følge av lavere trykk vil gass kunne komme ut av løsning, og medføre et redusert hydrostatisk trykk.

### 8.2.2.2 Iterasjon

Resultatet fra beregningene så langt er en tabell med trykktap fra de ulike sonene, med et gitt brønntrykk. Målet nå er å få disse trykktapene til å bli like for hver sone, ved et satt brønntrykk. Dette gjøres på følgende måte:

- ✓ Finner det minste trykktapet i punch for et gitt brønntrykk i de tre sonene
- ✓ "Låser" det minste trykktapet, kolonnen blir stående urørt. Dette er den sonen som potensielt vil kunne gi høyest masserate.
- ✓ Itererer ved å variere antall skjermer og antall hull per skjerm for de to andre sonene slik at vi får likt trykktap i alle sonene.
- ✓ Beregner strømningsraten for hver sone ut fra iterasjonsresultatet.
- ✓ Finner samlet strømningsrate for Ness 1, 2 og 3.

Vi sitter nå igjen med en modell hvor vi kan lese ut følgende fra beregningene:

- ✓ Antall ResPunch seksjoner samt antall hull per seksjon for hver enkelt sone
- ✓ Strømningsrate fra hver sone
- ✓ Samlet strømningsrate for brønnen
- ✓ Produksjonsindeks, PI for brønnen

### 8.2.2.3 Optimalisering av resultater

Fra de utførte iterasjonene, har vi nå en rekke data som ikke gir et entydig svar. Man er nødt til å sette et kriterium for å kunne vite hvilket brønntrykk man skal velge. Ut fra tabellen med brønntrykk (0-313 bar) og trykkkravet fra avsnitt 8.2.2.1, kan det beregnes et maksimalt tillatt trykktap gjennom tubing fra reservoaret til GFA.

$$\Delta P_{tubing}^{\max} = P_{tubing} - P_{hydrostatic}$$

**Formel 14: Tillatt tap i tubing**

Vi gjør denne beregningen for alle tallene i tabellen (0-313). Ut fra resultatene gitt av formel 14, kan nå en maksimal tillatt strømningsrate gjennom tubing beregnes.

Det tas utgangspunkt i likningen for trykktap i rør med tapshøyde  $h_t$ :

$$h_t = f \frac{L}{d_{tubing}} \frac{v_{flow}^2}{2g}$$

Multipliserer med  $\rho g$  og får:

$$\Delta P_{tubing} = f \frac{L}{d_{tubing}} \frac{v_{flow}^2}{2} \rho \quad \text{gitt at } v_{flow} = \frac{q}{A_{flow}}, \text{ gir:}$$

$$\Delta P_{tubing} = \frac{fL\rho}{2d_{tubing}} \frac{q_{tubing}^2}{\left(\frac{\pi}{4}d_{tubing}^2\right)^2}$$

$$\Delta P_{tubing} = \frac{8fL\rho}{d_{tubing}^5} \frac{q_{tubing}^2}{\pi^2}$$

Løser mhp på  $q$  og får:

$$q_{tubing}^{max} = \sqrt{\frac{\pi^2 \Delta P_{tubing}^{max} d_{tubing}^5}{8fL\rho}}$$

**Formel 15: Tillatt produksjonsrate**

Fra denne formelen kan nå den maksimalt tillatte strømningsraten for alle gitte brønntrykk beregnes. Man bruker denne verdien som kriterium for hvilket brønntrykk som bør velges ved å sammenlikne den maksimalt tillatte raten med den potensielle raten reservoaret kan gi. Det gjøres en sjekk hvor alle tallene (ratene) i tabellen sammenliknes, og vurderes på følgende måte:

$$q_{potensiell} < q_{tubing}^{max}$$

Dersom dette kriteriet er oppfylt, er brønntrykket anvendbart. Hele tabellen sjekkes, og raten med minst avvik fra den tillatte kan velges som strømningsrate. Det settes også en nedre grense for situasjonen, hvor brønntrykket er  $P_{res}-1$  bar. Det vil si at vi i dette tilfelle får et vindu for raten i mellom  $q_{potensiell}$  og  $q_{minimum}$  gitt av kriteriene nevnt ovenfor.

Optimaliseringen er nå utført, og man kan lese av følgende data:

- ✓ Anbefalt brønntrykk (vindu, max-min)
- ✓ Trykktap i ResPunch for alle soner
- ✓ Antall ResPunch seksjoner per sone, samt antall hull per seksjon
- ✓ Strømningsrate, samlet og fra hver sone (vindu, max-min)
- ✓ Trykktap gjennom formasjon i hver sone
- ✓ Totalt trykktap for hver sone
- ✓ Total lengde av installerte ResPunch seksjoner

Antagelser:

Den gitte friksjonsfaktoren  $f$  i likningen er funnet ved iterasjon. Beregninger med Reynoldstall ga i gjennomsnitt en friksjonsverdi på 0,003, og denne verdien er derfor brukt i alle beregningene for trykktap i tubing.

## 9 Simuleringsprogram i Java

Gruppen har valgt at oppgaven skal løses ved blant annet å lage et javaprogram, ”PunchSim”, som utfører de nødvendige beregningene i simuleringen. Brukeren av programmet legger selv inn nødvendige data for de ulike sonene, og programmet vil benytte disse dataene til å utføre simuleringen. Resultatet vil bli gitt brukeren ved hjelp av et brukergrensesnitt.

Programmet inkluderer alle variabler og faste verdier for bruk av ResPunch. Det tas utgangspunkt i Gulltopp brønnen under utviklingen av programmet, men tanken er at det skal være et generelt program som skal kunne brukes for ResPunch i alle brønner. I forbindelse med systemutvikling går utviklerne og deres samarbeidspartnere gjennom ulike faser. I dette kapitlet beskrives de ulike utviklingsfasene og hvordan de utartet seg i forbindelse med prosjektet.

### 9.1 Forstudium

Ved utvikling av programvare er det viktig å forstå hvilke oppgaver programvaren skal utføre. Når et program skal lages i tilknytning til andre fagfelt må utvikleren av dataprogrammet forstå begreper og teori knyttet til fagfeltet[13]. Innehenting av nødvendig kunnskap fører blant annet til:

- Utvikleren får forståelse for problemstillinger knyttet til oppgaven
- Utvikleren kan velge passende programvarearkitektur og mønstre ut i fra ulike problemstillinger
- Utvikleren får en bedre forståelse av hva som er essensielt i programmet og hva som kan nedprioriteres.
- Utvikleren kan ta del i diskusjoner og forklare hva som er mulig å implementere av programvare
- Utvikleren kan komme med forslag til løsninger

Forstudiet tilknyttet oppgaven har omfattet terminologi knyttet til komplettering og ResPunch moduler, forståelse av matematiske løsninger og hvordan ulike variabler påvirkes av hverandre og omgivelsene. Disse elementene er beskrevet i de foregående kapitler. Innhentning av informasjon om hvordan ønsket interaksjon mellom bruker og programvare vil foregå har også vært en del av forstudiet. Interaksjonen er dokumentert i sekvensdiagrammet i kap 9.3.

### 9.2 Krav til programmet

Så snart utvikleren av programvaren har forstått problemstillingene knyttet til oppgaven kan disse dokumenteres ved å sette krav til programmet som skal utvikles. Kravene beskriver hvordan det ferdige produktet vil være, og ikke hvordan det vil bli implementert. Kravene kan benyttes til å føre en dialog med andre som ønsker å påvirke programmets utvikling. Ved implementering er det opp til utvikleren å sørge for at kravene blir tilfredsstillt[13].

Den ferdige programvaren skal oppfylle kravene gitt i tabell 9.1. Kravene har fått tildelt prioritet og vanskelighetsgrad. Dersom det på grunn av tidspress ikke er mulig å implementere alle krav, behøver ikke utvikleren å implementere de kravene med lavest prioritet.

Krav ID	Krav	Prioritet	Vanskelighetsgrad
K1. Formler	Programmet skal kjøre formlene i riktig rekkefølge.	H	H
K2. Brukergrensesnitt	Programmet skal ha et brukergrensesnitt som gjør det mulig for brukeren å registrere variabler knyttet til soner og fluid.	H	M
K3. Resultat	Programmet skal vise resultatene av simuleringen i brukergrensesnittet.	H	M
K4. Generelle problemstillinger	Programmet skal utvikles for generelle problemstillinger. Dvs. at programmet skal simulere bruk av ResPunch i ulike brønner.	M	M
K5. Versjon	Programmet skal kjøre på J2SE versjon 5.0.	M	M
K6. Tilbakemeldinger	Programmet skal gi tilbakemeldinger til brukeren ved interaksjon mellom bruker og program.	L	L
K7. Testing	Ved utvikling skal resultater av simuleringen være mulig å teste.	H	M

Tabell 5: Krav

Etter at programvaren var ferdig utviklet ble kravene testet. Dette er nærmere beskrevet i kap 9.5.

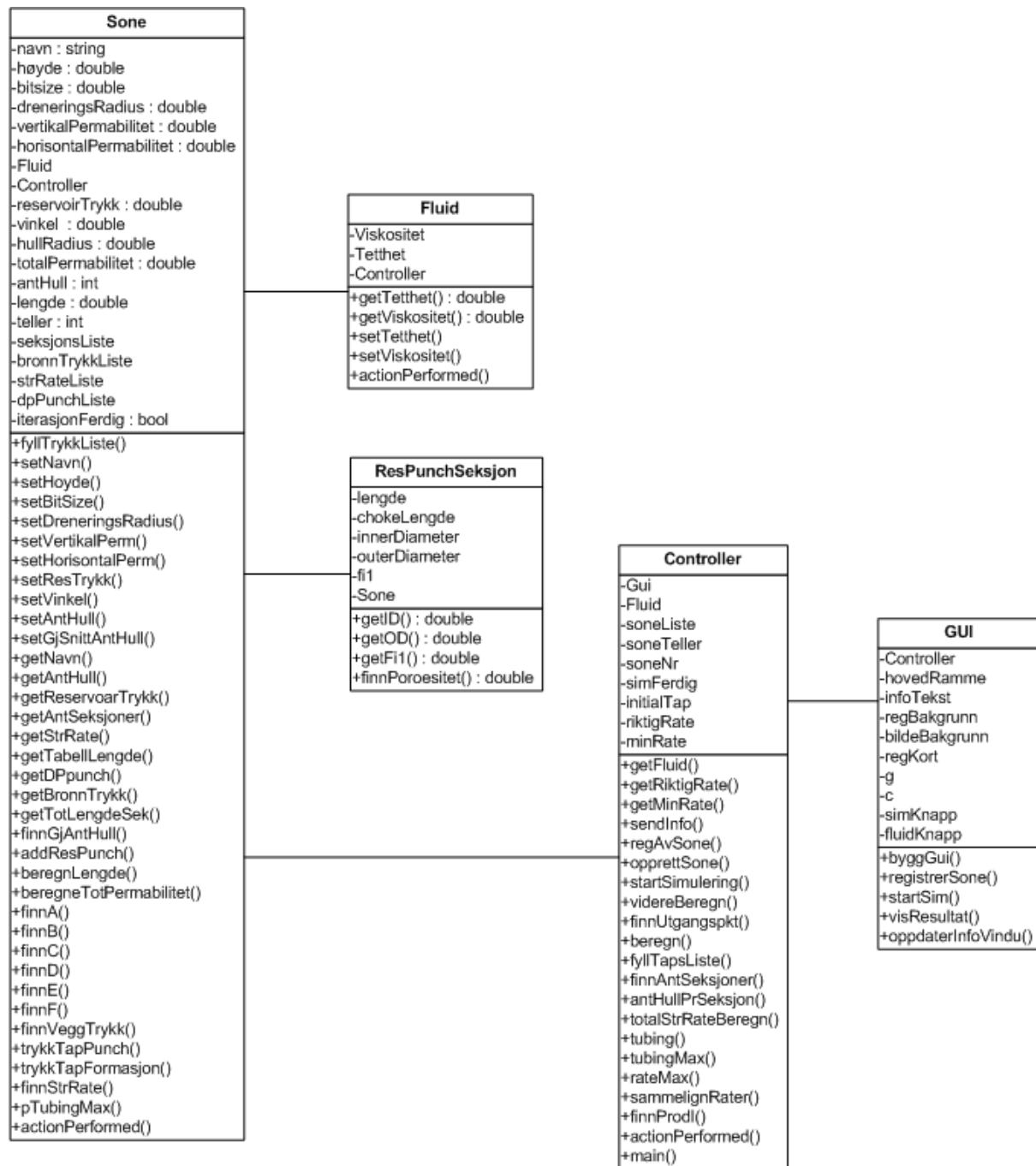
### 9.3 Konstruksjon

For å oppnå en forståelse av hvordan programmet som skal utvikles skal bygges opp, er det viktig å modellere strukturen og interaksjon mellom moduler innad i programmet. Modellering er svært viktig fordi en god modell fører til en felles forståelse av programmet og funksjonaliteten, og man unngår misforståelser. En god modell sparer derfor tid og man unngår forhåpentligvis dobbeltarbeid.

#### 9.3.1 Klassediagrammet

Grappa har modellert programmet ved hjelp av et klassediagram. . En generell beskrivelse av klassediagram er gitt i vedlegg. Klassediagrammet i figur 13 viser relasjonene mellom modulene i systemet, samt hvilken funksjonalitet de har i form av variabler og metoder i klassen. Variablene er gitt i øverste del av klasses tabellen, metodene er gitt i nederste del.



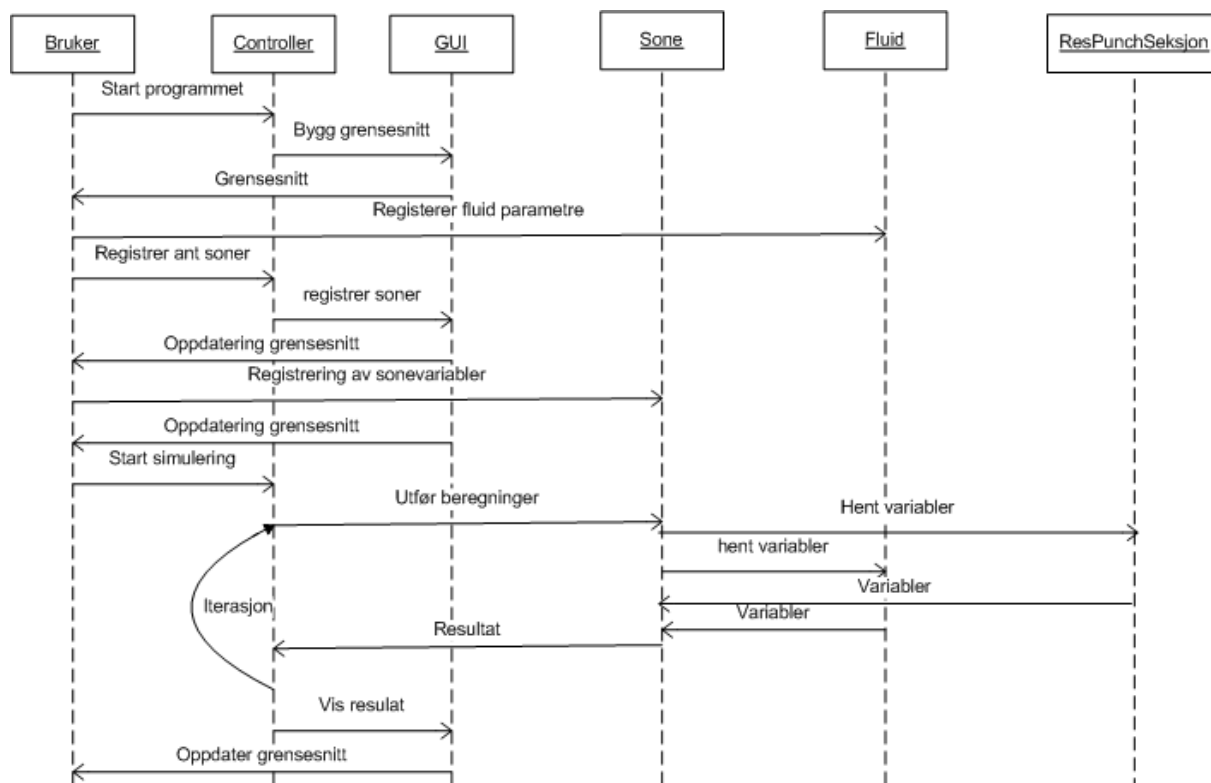


Figur 13: Klassediagram

Programmet består av klassene "Sone", "Fluid", "ResPunchSeksjon", "Controller" og "GUI". "Sone" klassen tar seg av beregningene knyttet til hver enkelt sone. I tillegg sørger klassen for registrering av de ulike sonevariablene. "Fluid" klassen ivaretar variablene som beskriver fluidegenskapene. Disse variablene benyttes av de andre klassene. "ResPunchSeksjon" klassen inneholder informasjon knyttet til hver enkelt ResPunch. Denne informasjonen benyttes også ved beregninger. "Controller" klassen styrer og kontrollerer interaksjonen mellom klassene og tar seg av de totale beregningene for hele reservoaret. "GUI" står for Graphical User Interface og bygger opp grensesnittet.

### 9.3.2 Sekvensdiagrammet

For å få forståelse av interaksjonen mellom klassene, men også mellom bruker og programmet, benyttes et sekvensdiagram. Sekvensdiagrammet viser hvordan det ferdige dataprogrammet vil kjøre i sekvenser og hvilke klasser som samarbeider. En generell beskrivelse av sekvensdiagrammer er gitt i vedlegg. Sekvensdiagrammet for PunchSim programmet er illustrert i figur 14.



Figur 14: Sekvensdiagram

### 9.4 Implementering

Etter at konstruksjonen av programmet var ferdig, var det mulig å implementere de ulike klassene med tilhørende sekvenser. I og med at det vil være mye interaksjon mellom brukeren og programmet, er mye av koden skrevet for brukergrensesnittet. Koden er skrevet i Java og er dokumentert ved bruk av javadoc. Se appendiks for fullstendig kode. Brukeren er avhengig av å ha installert Java Standard Edition, J2SE, på maskinen sin for å kunne kjøre programmet.

### 9.5 Testing

Når man utvikler et eget program som skal utføre simuleringer, er man nødt til å vite om de faktiske svarene som kommer ut, er 100 % riktige. Gruppen har løst dette ved først å kjøre hele simuleringprosessen i MS Excel. Regnearket som gjorde simuleringen var en tidkrevende prosess, men også helt nødvendig for resultatenes del.

Under programmeringen av PunchSim, gikk man steg for steg gjennom alle formler og iterasjoner som ble utført. Ved å sjekke resultatene fra PunchSim mot verdiene fra regnearket, er man sikre på at resultatene stemmer.

I tillegg har vi testet at kravene i kap 9.2 er tilfredsstilt. Resultatet av testingen er oppsummert i tabell 15.

Krav ID	Resultat av test
K1. Formler	Formlene kjører i riktig rekkefølge. Ved iterasjon oppdateres resultatet.
K2. Brukergrensesnitt	Brukeren kan registrere nødvendige variabler ved hjelp av brukergrensesnittet.
K3. Resultat	Resultatet av simuleringen vises i brukergrensesnittet
K4. Generelle problemstillinger	Programmet er utviklet slik at det kan brukes i forbindelse med bruk av Respunch i andre brønner.
K5. Versjon	Programmet bør kjøres ved hjelp av versjon J2SE.
K6. Tilbakemeldinger	Brukeren mottar tilbakemeldinger fra programmet ved hjelp av et informasjonsvindu. Grensesnittet oppdateres ut i fra brukerens handlinger.
K7. Testing	Ved bruk av MS Excel har det vært mulig å teste formlene etter hvert som de ble implementert.

Figur 15: Testresultater

## 10 Resultat av simulering

Ved hjelp av PunchSim, kan nå simuleringen utføres i et eget program. Programmet i Java vil fungere på samme måte som simuleringen gjort i MS Excel. Den merkbare forskjellen, er at brukeren av PunchSim kun trenger å legge inn de variablene programmet ber han/henne om å legge inn via brukergrensesnittet. Når dette er gjort, kjører programmet simuleringen selv, uten ekstern hjelp fra bruker. PunchSim utfører hele iterasjonsprosessen på samme måte som Solver og Visual Basic Editor i MS Excel gjør. Beregningene skjer kjapt, og bruker får umiddelbart opp de ønskede resultater på skjermen ved å følge instruksene PunchSim gir.

### 10.1 Resultater uten optimalisering ResPunch

Uten å installere ResPunch på Gulltopp vil det ikke være noen mulighet for å kunne regulere trykktapene i de ulike sonene i Ness. Dette vil føre til at det etter hvert vil bli et ujevnt trykk i de ulike sonene, og innstrømning fra en sone til en annen vil etter tid kunne forekomme.

Vann/olje kontakten vil også bevege seg ulikt i de forskjellige sonene, og vannfronten vil nå brønnen ved et tidligere tidspunkt enn ønsket.

I appendiks vises det hvordan trykkene i Ness 1, 2 og 3 vil utvikle seg ved ulike strømningsrater og trykktap uten optimalisering. Det er også vist hvordan trykktapet i ResPunch endres ved ulike brønntrykk. Det totale trykktap i hver enkelt sone er vist i korrelasjon med den totale masseraten. Man kan se på disse resultatene som en analogi til en brønn med konvensjonelle screens, uten mulighet for regulering av trykk.

### **10.1.1 Ness 1**

Ved et gitt innstrømningsareal felles for alle sonene, vil Ness 1 ha det laveste trykktapet. Tapet er estimert til å ligge mellom 0,26 – 11,99 bar, avhengig av hvilket brønntrykk som velges.

### **10.1.2 Ness 2**

Denne sonen er estimert til å ha det høyeste trykktapet. Tapet er beregnet til å ligge mellom 0,48 - 14,14 bar, avhengig av brønntrykk.

### **10.1.3 Ness 3**

Trykktap i denne sonen blir liggende mellom tapene i Ness 1 og-2 i størrelsesorden.

### **10.1.4 Total trykkbalanse**

Vi vil her få en ujevn balanse for de tre valgte sonene. Det vil tapes forskjellig i hver sone, og forskjellen mellom minste og største tap er estimert til å være ca 2,15 bar. Initielt er dette ikke noen stor differanse, men vil raskt føre til ujevn forflytning av vannfronten inn mot brønnen.

Vedlagt i appendiks er også kildekoden som er grunnlaget for disse resultatene.

## ***10.2 Optimaliserte resultater med ResPunch***

Med ResPunch installert, vil det totale trykktap i hver sone bli så å si identisk ved å justere antallet moduler og antall hull per modul.

I appendiks vises det hvordan trykkene i Ness 1, 2 og 3 vil utvikle seg ved ulike strømningsrater og trykktap med optimalisert ResPunch. Det er også vist hvordan trykktapet i ResPunch endres ved ulike brønntrykk. Også det totale trykktap i hver enkelt sone er vist i korrelasjon med den totale masseraten.

### **10.2.1 Ness 1**

Etter beregningene som er utført, viser det seg at Ness 1 har det minste trykktapet. Dette gjør denne sonen til den dominante, og egenskapene til modulene i Ness 2 og -3 må justeres ved å endre antall moduler og antall hull per modul.

Følgende er kalkulert for Ness 1:

**SONE 1**

# hull	20	stk/seksjon
Antall skjermer	1	stk

Reelt # hull	20	stk/seksjon
Reelt # skjermer	1	stk

Lengde Punch	12,5	m
--------------	------	---

Det er altså tenkt plassert 1 modul i denne sonen, med 20 hull, som er maksimalt per modul. Denne konfigurasjonen vil gi et trykktap over ResPunch i området 0,26 – 16,28 bar avhengig av valgt brønntrykk

### 10.2.2 Ness 2

Itererte resultater fra Ness 2 endte opp med følgende:

**SONE 2**

# hull	19	stk/seksjon
Antall skjermer	2	stk

Reelt # hull	19	stk/seksjon
Reelt # skjermer	2	stk

Lengde Punch	25	m
--------------	----	---

For å kompensere mot strømningsraten fra Ness 1, er det nødvendig å installere 2 moduler i Ness 2, med 19 hull i hver modul. Ved å gjøre dette, oppnås et identisk trykktap over ResPunch totalt i hver av sonene, og vannfronten beveger seg likt.

### 10.2.3 Ness 3

Itererte resultater fra Ness 3 endte opp med følgende:

<b>SONE 3</b>		
# hull	13	stk/seksjon
Antall skjermer	2	stk
Reelt # hull	13	stk/seksjon
Reelt # skjermer	2	stk
Lengde Punch	25	m

2 moduler med 13 hull i hver er nødvendig for å kompensere for Ness 1.

### 10.2.4 Total trykbalanse

Trykbalansen mellom de tre sonene er nå optimalisert og ideell. Det er kun små sprik i tapene over ResPunch, men dette er i størrelsesorden 0,04 bar og anses som neglisjerbart. Vedlagt i appendiks finnes regnearket med alle tallsvar.

### 10.2.5 Valg av brønntrykk

Hvor stort det aktuelle trykktap i hver sone vil bli, er en funksjon av brønntrykket. Ved å bruke argumentasjonen fra kapittel, kan vi beregne en maksimal tillatt rate gjennom tubing ut fra en gitt maksimums rate, gitt av det totale trykktap gjennom tubing. I dette tilfellet er det minimalt tillatte brønntrykket estimert til å være 291 bar, noe som gir et totalt tap i ResPunch i området 16,18 – 16,32 bar i de ulike sonene.

Brønntrykket er kun et estimat som vil en maksimalt tillatt strømningsrate fra reservoaret samlet, og bør nok justeres ned til rundt 300 bar, hvor det totale trykktapet er omkring 13 bar. Grunnen til dette er at strømningsraten blir i størrelsesorden 11 000 – 13 000 Sm<sup>3</sup>/dag, noe som er relativt høyt. I følge Statoil er Ness beregnet til å gi en samlet rate på rundt 3500 – 5000 Sm<sup>3</sup>/dag. Skal dette kravet etterleves, ender vi opp med et anbefalt brønntrykk på 308-310 bar og tap i samtlige ResPunch på ca 2,55 ±0,2 bar per modul.

### 10.2.6 Samlede resultater

Det konkluderes med følgende resultater ut fra beregningene:

#### Realistisk tilfelle:

$$q_{\min} = 5061-3591 \text{ Sm}^3/\text{d}$$

ved

$$P_{\text{tubing-max}} = 308-310 \text{ bar}$$

$$PI = 1012-1197 \text{ Sm}^3/\text{d}/\text{bar} \text{ for Ness samlet.}$$

#### Teoretisk tilfelle:

$$q_{\min} = 12\,499 \text{ Sm}^3/\text{d}$$

ved

$$P_{\text{tubing-min}} = 291 \text{ bar}$$

$$PI = 568 \text{ Sm}^3/\text{d}/\text{bar} \text{ for Ness samlet.}$$

Appendiks viser alle data brukt i disse beregningene.

## 11 Feilkilder

De potensielle feilkildene ved dette prosjektet er mange. Prosjektet er meget omfattende for oss som studenter, spesielt med tanke på vanskelighetsgrad og tidsperspektiv. I slutfasen av prosjektet kan derfor følgende feilkilder påpekes:

### 11.1 Formler

De analytiske metodene brukt under ved løsningen av prosjektet er en feilkilde som anses som mindre viktig. Fagpersoner med gode kunnskaper i strømningsteknikk og reservoarteknikk har bekreftet at utledningene er riktige i forhold til realiteten. Det som gjør dette til en potensiell feilkilde, er at man har svært få eksempler med liknende problematikk å ta utgangspunkt i. Utledningene som er gjort i de foregående kapitler er gjort med basis i la kontinuiteter for de aktuelle modulene, og den spesielle strømmingen som oppstår i en slik modul, gjør det vanskelig å uttrykke strømningssituasjonen matematisk. Likevel mener man bestemt at dette er løst på en korrekt og presis måte. Feilkildene knyttet til formlene anses derfor til å være mindre viktige for selve ResPunch modulen.

Når det gjelder strømningsslikningene for selve reservoaret er usikkerheten noe større. Det er brukt en enkel strømningssmodell, Darcy's likning, til å uttrykke en potensiell masserate fra et gitt reservoar. Denne likningen baserer seg på en vertikal brønn med en gitt høyde,  $h$ , og en horisontal permeabilitet gjeldende for hele reservoaret. I vårt tilfelle er brønnen horisontal, lagene i reservoaret har dippvinkel på ca 20 grader, og vi har ulike strømningsegenskaper for horisontal og vertikal retning.

Dette gjør at man ser på strømningsbildet i formasjonen som den største feilkilden med tanke på de formler som er anvendt.

### **11.2 Regnefeil**

Under utviklingen av regnearket i MS Excel, som er basisen for simuleringen og programmet, ble det til stadighet oppdaget små og store regnefeil. Programmet ble i slutfasen sjekket grundig for lignende feil, men det kan derimot ikke garanteres at samtlige utregninger 100 % feilfrie.

### **11.3 Feil i antagelser**

Relativt mange antagelser er tatt under dette prosjektet. Det kan derfor hevdes at mange av antagelsene som er gjort underveis kan utgjøre en potensiell feilkilde.

### **11.4 Feil i data**

Grunnet begrenset tilgang på geologiske data for det aktuelle området, sees det som en nødvendighet å nevne at dette er en feilkilde i det utførte arbeidet. Usikkerhet knyttet til høyden av de ulike sonene, strukturen i reservoaret, samt variabler som porøsitet og permeabilitet gjør at man er usikker på om resultatet er pålitelig nok.

På den annen side betraktes beregningene som er gjort som pålitelige



## 12 Konklusjon

Det vil være en rekke parametere som er avgjørende for kompletteringen av Gulltopp, Ness-formasjonen, herunder er produksjonsprofil med ønsket strømningsrate og brønntrykk blant de viktigste.

Det er i denne oppgaven definert et minimum brønntrykk for å kunne løfte oljesøylen til overflaten tatt i betraktning de trykktap som oppstår fra brønn til topside. Simuleringer er gjort med brønntrykk fra 200 – 313 bar, der resultatene viser at minimum anbefalt brønntrykk er 291 bar. Dette trykket vil for øvrig gi en strømningsrate på 12 499 Sm<sup>3</sup>/dag. I henhold til simuleringer gjort av Statoil nevnes det en forventet rate på 3500-5000 Sm<sup>3</sup>/dag ved produksjonsstart [1]. Et brønntrykk på 308-310 bar gjør at vi befinner oss innenfor en akseptabel strømningsrate, henholdsvis 3591-5061 Sm<sup>3</sup>/dag.

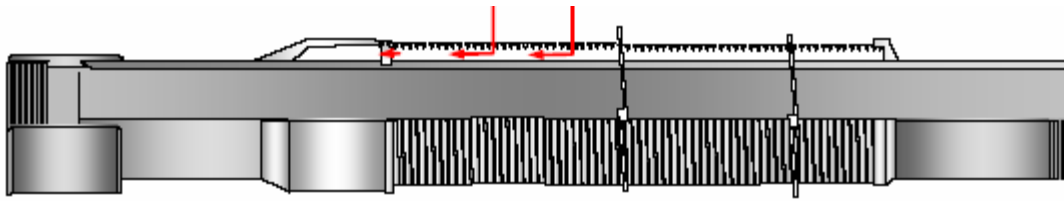
Brønntrykk i område 308–310 bar gir et trykktap gjennom ResPunch som vil variere fra 0,04-0,03 bar mellom de ulike sonene. Dette ansees som en fullt ut akseptabel differanse. Dersom det som nevnt antas at vann-olje kontakten er lik for alle sonene, vil det over tid stige likt opp mot brønnen med det resultat at optimalt mengde hydrokarboner vil kunne utvinnes fra Ness-formasjonen.

Den optimale fordeling av ResPunch moduler er således beskrevet i kapittel 10.2. Som en konsekvens av at differansen mellom det totale trykktapet for hver sone er tilnærmet uavhengig av brønntrykket, er gitte antall seksjoner og hull for hver sone konstant for Ness-formasjonen.

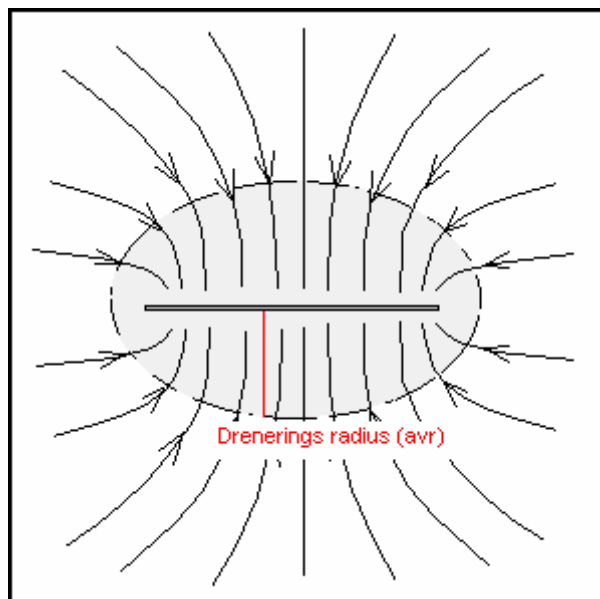
Denne rapporten har ikke tatt hensyn til den rent ”geografiske” plasseringen av ResPunch i sonene sett opp imot formasjonsstrukturer og strømmingsegenskaper i Ness. Intensjonen som nevnt i kapittel 2 har vært å bestemme hvordan man kan balansere produksjonen i Ness for å utvinne mest mulig reserver og oppnå et simultant vanngjennombrudd fra alle sonene. En nærmere plan for plassering av ResPunch i sonene bør gjøres i samsvar med geologiske studier av Ness, derunder skiferlag og kullsoner med lav permeabilitet. Fordelingen av ResPunch vil være uavhengig av plassering, så lenge hver sone har samme reservoar- og fluidparametere.

Produktet av dette prosjektet, PunchSim er ment å fungere som en generell simulator, ikke bare for Gulltopp, men for alle gitte brønner som ønskes installert med ResPunch. Dette har vist seg å fungere etter tester gjort av gruppen ansvarlige for denne rapporten

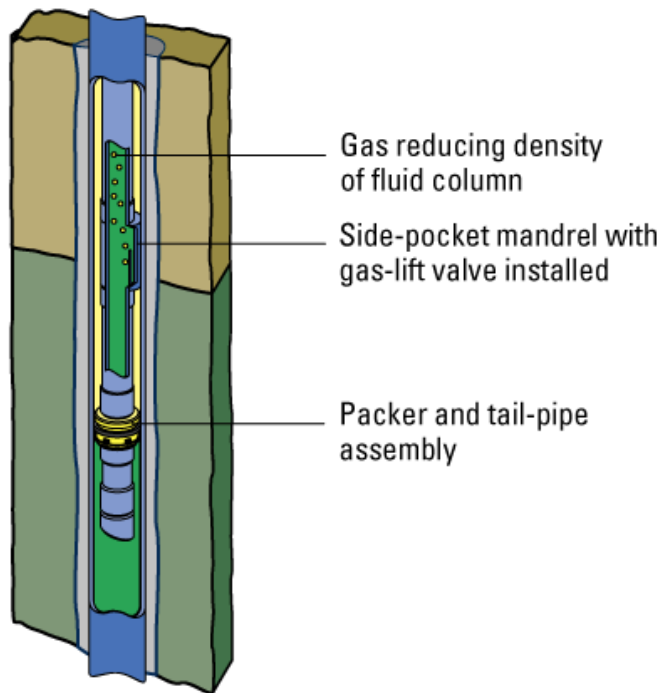
## 13 Vedlegg



Figur 16: Skisse av ResPunch



Figur 17: Strømning rundt horisontal brønn



Figur 18: Skisse, gassløft [10]



Figur 19: Logo, PunchSim

## **Tegnforklaring**

- $h_1$  – avstand mellom skjerm og yttervegg produksjonsrør  
 $h_2$  – avstand mellom innside ”choke house” og yttervegg produksjonsrør  
 $h$  – høyde av reservoar  
 $R_i$  – ytre radius for produksjonsrør  
 $R_0$  – ytre radius for sandskjerm  
 $\rho$  – tetthet  
 $\phi_1$  – porøsitet sandskjerm  
 $\phi_2$  – porøsitet i Choke House  
 $f$  – friksjonstall  
Re – Reynholdstall  
 $u$  – Strømningshastighet  
 $D_h$  – Hydraulisk diameter  
 $P_0$  – Trykk på borehullsvegg  
 $P_{res}$  – Reservoartrykk  
 $P_i$  – initielt reservoartrykk  
 $P_{hydrostatic}$  – Hydrostatisk trykk  
 $P_{topside}$  – Topside leveringstrykk.  
 $P_{tubing}$  – Bunnhullstrykk i brønn  
 $\Delta P_{friction}$  – Trykktap grunnet friksjon i produksjonsrør  
 $\mu$  – viskositet  
 $q$  – masserate  
 $r_i$  – dreneringsradius av reservoar  
 $r_w$  – brønnradius  
GFA – Gullfaks A  
 $L$  – brønnlengde målt dybde  
 $A_{flow}$  – strømningsareal

## **13.1 Referanser**

Det kan vises til følgende referanser under arbeidet med rapporten:

### **13.1.1 Litteratur**

- [1]: "Gulltopp Brent BoV-BoG", rapport av Statoil 2003.
- [2]: "Recommendation to drill, boreanb A32C", rapport Statoil 2004.
- [3]: "Gulltopp ResPunch NTNU", presentasjon Statoil 2005.
- [4]: "ResPunch qualification test report", Reslink 2004.
- [5]: "Produksjonsbrønner", Konrad Berge 1988.
- [6]: "Transport in Porous Media", Jacob Bear 2003.
- [7]: "Strømningstekniske aspekter ved produksjon gjennom screen", Tor Ytrehus, notat 2005.
- [8]: "Fluid Mechanics, Fourth Edition", Frank M. White.
- [13]: "Software Engineering, Principles and Practice, Second Edition", Hans Van Vliet 2000.

### **13.1.2 Nettsteder**

- [9]: <http://www.weatherford.no>
- [10]: <http://www.glossary.slb.oilfield.com>
- [11]: <http://www.reslink.no>
- [12]: <http://www.statoil.no>

### **13.1.3 Fagpersonell**

Ole Torsæter, Professor  
Tor Ytrehus, Professor

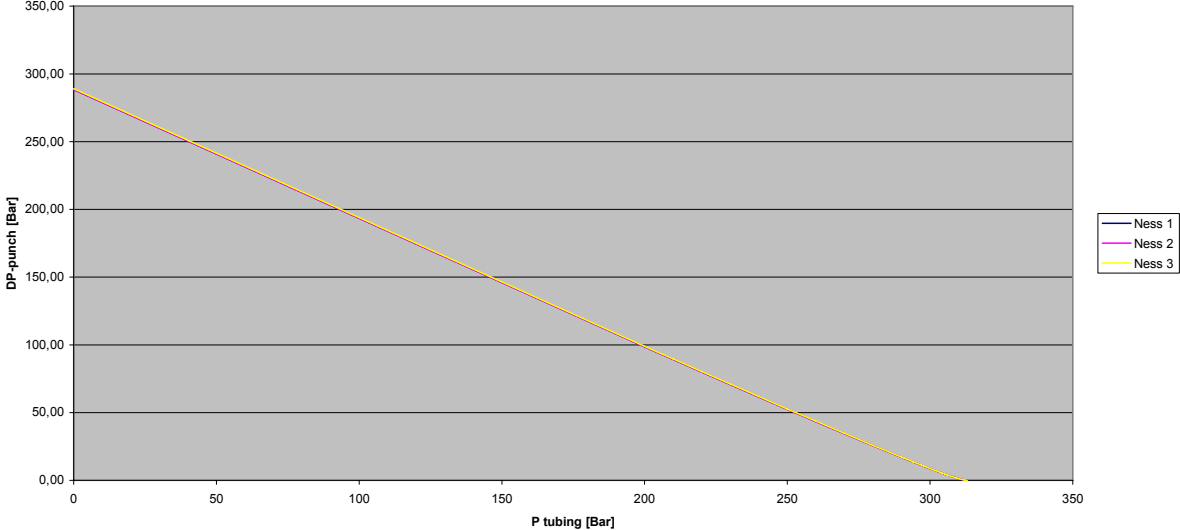
# *Appendiks*

*Data og kildekode  
for*

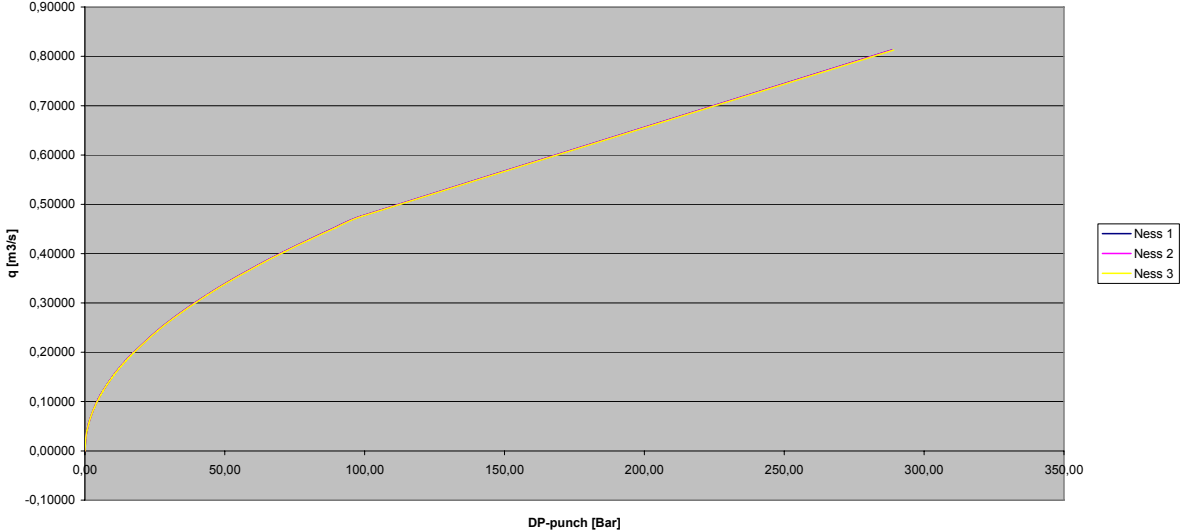


# Resultat, optimalisert

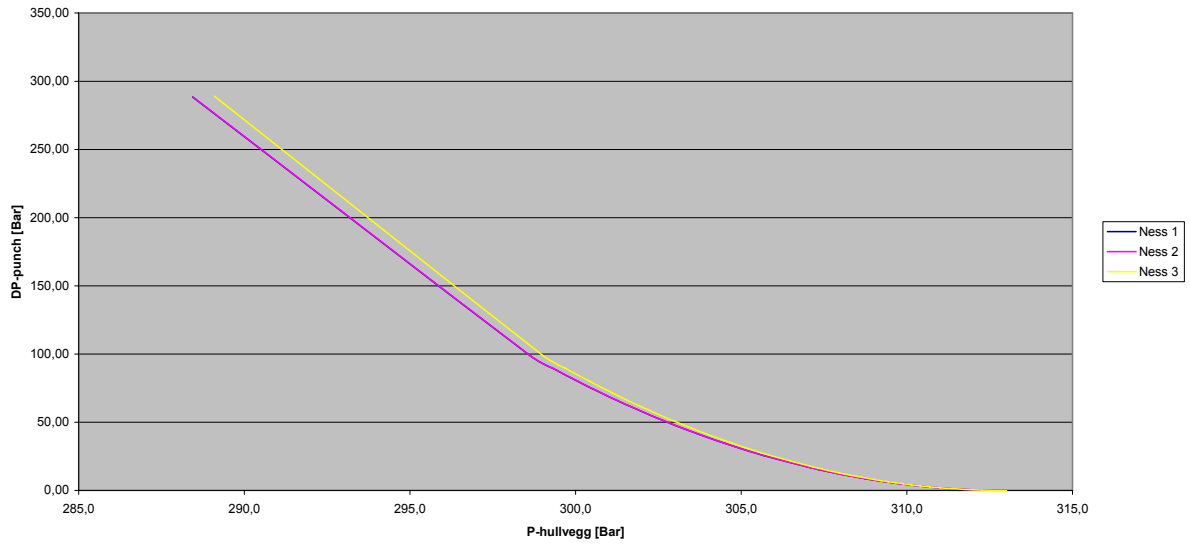
P-tubing vs DP-punch



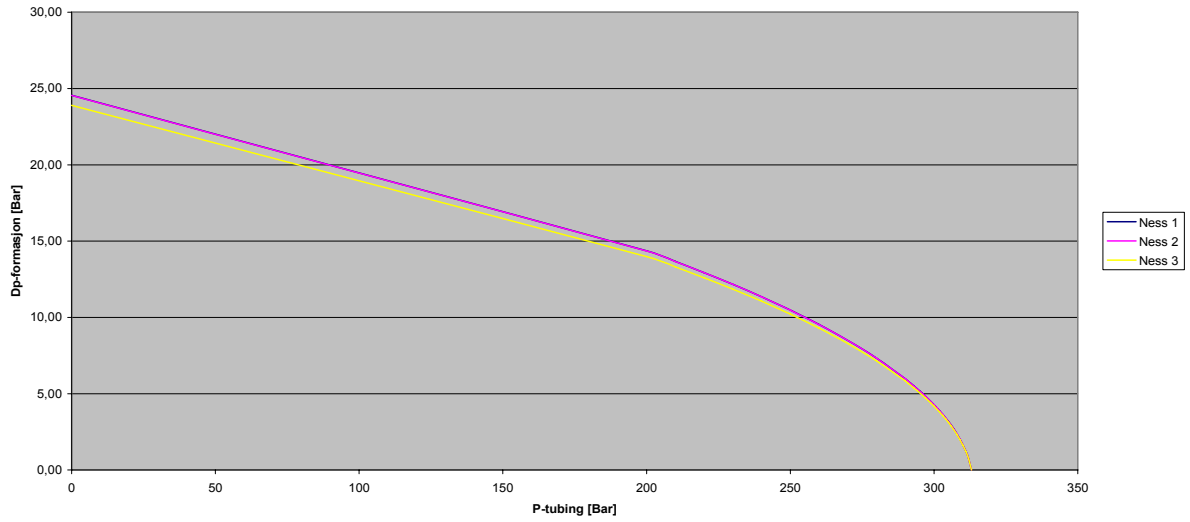
DP-punch vs Q-tot



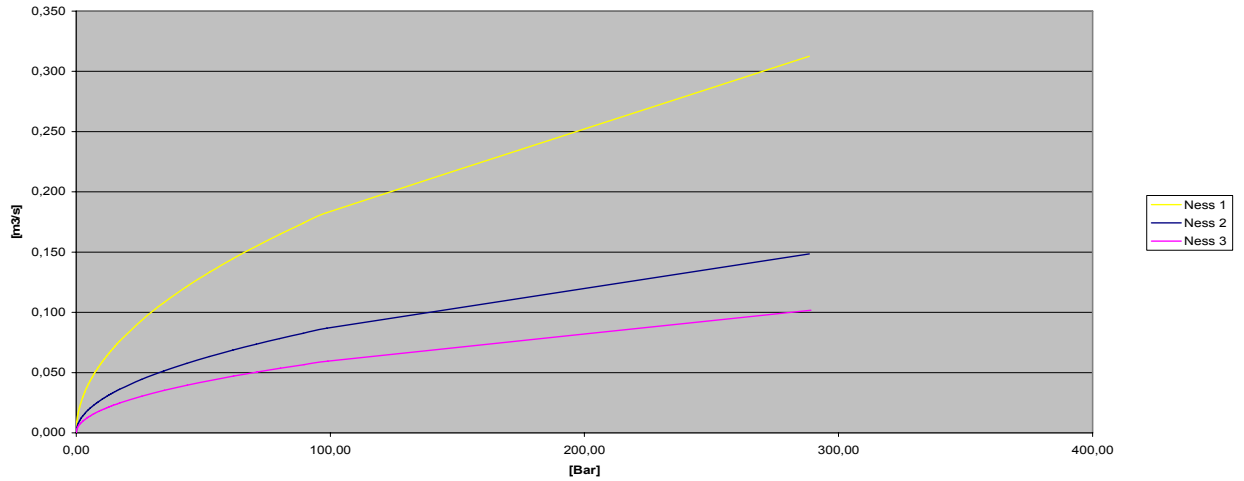
P-hullvegg vs. DP-punch



P-tubing vs DP-formasjon



DP-Punch vs q-punch





## Kildekode, optimalisert

### Faste variable:

Pres	31300000 Pa
Dy	0,196 m
Di	0,1778 m
Ah	0,005343189 m2
A-innstr	4,556565985 m2
A-screen inflow	0,407 m2
d-hull	0,01016 m
L-screen	7,4 m
Pw-min	22245360 Pa
Turb	>2300
Lam	<2300
Dh	0,0364
Phi 1	0,089321652
A-choke	0,837862761 m2
A-hull	8,10732E-05 m2
D-tubing	6,184 "
	0,1570736 m
L	10000 m
TVD	2500 m



SONE 1			SONE 2			SONE 3		
# hull	20	stk/seksjon	# hull	19	stk/seksjon	# hull	13	stk/seksjon
Antall skjerm	1	stk	Antall skjerm	2	stk	Antall skjerm	2	stk
Reelt # hull	20	stk/seksjon	Reelt # hull	19	stk/seksjon	Reelt # hull	13	stk/seksjon
Reelt # skjerm	1	stk	Reelt # skjern	2	stk	Reelt # skjerr	2	stk
Lengde Punch	12,5	m	Lengde Punc	25	m	Lengde Punc	25	m
A	1,27264E-07		A	1,2096E-07		A	8,50917E-08	
B	0,000213396		B	0,000213396		B	0,000213396	
C	0,000904366		C	0,000904366		C	0,000904366	
Phi 2	0,001935238		Phi 2	0,001838476		Phi 2	0,001257905	

## NESS 1

P-tubing		a	b	c	P-hullvegg	Dp-punch	Dp-formasjon	q	Dp-tot	PI	q-tot	
[Pa]	[Bar]				[Pa]	[Bar]	[Bar]	[m3/s]	[Bar]		[m3/s]	
0	0	-4,78194E-06	300,3497078	-4684822927	28844015,58	288,4	288,44	24,56	0,313	313,00	0,000998589	0,3125582
20000000	200	-4,78194E-06	300,3497078	-4704822927	29863784,73	298,6	98,64	14,36	0,183	113,00	0,001617508	0,18277838
21000000	210	-4,78194E-06	300,3497078	-4705822927	29933210,8	299,3	89,33	13,67	0,174	103,00	0,001688767	0,17394295
22000000	220	-4,78194E-06	300,3497078	-4706822927	30006079,22	300,1	80,06	12,94	0,165	93,00	0,001770639	0,16466943
23000000	230	-4,78194E-06	300,3497078	-4707822927	30082959,25	300,8	70,83	12,17	0,155	83,00	0,001866089	0,15488537
24000000	240	-4,78194E-06	300,3497078	-4708822927	30164596,87	301,6	61,65	11,35	0,144	73,00	0,001979395	0,14449585
25000000	250	-4,78194E-06	300,3497078	-4709822927	30252002,58	302,5	52,52	10,48	0,133	63,00	0,00211702	0,13337226
26000000	260	-4,78194E-06	300,3497078	-4710822927	30346605,07	303,5	43,47	9,53	0,121	53,00	0,002289298	0,12133278
27000000	270	-4,78194E-06	300,3497078	-4711822927	30450542,88	304,5	34,51	8,49	0,108	43,00	0,002514075	0,10810524
28000000	280	-4,78194E-06	300,3497078	-4712822927	30567285,43	305,7	25,67	7,33	0,093	33,00	0,002825701	0,09324813
29000000	290	-4,78194E-06	300,3497078	-4713822927	30703197,95	307,0	17,03	5,97	0,076	23,00	0,003302233	0,07595137
29100000	291	-4,78194E-06	300,3497078	-4713922927	30718267,98	307,2	16,18	5,82	0,074	22,00	0,003365159	0,0740335
29200000	292	-4,78194E-06	300,3497078	-4714022927	30733676,45	307,3	15,34	5,66	0,072	21,00	0,003432026	0,07207255
29300000	293	-4,78194E-06	300,3497078	-4714122927	30749447,23	307,5	14,49	5,51	0,070	20,00	0,003503275	0,07006655
29400000	294	-4,78194E-06	300,3497078	-4714222927	30765607,15	307,7	13,66	5,34	0,068	19,00	0,003579417	0,06800893
29500000	295	-4,78194E-06	300,3497078	-4714322927	30782186,52	307,8	12,82	5,18	0,066	18,00	0,003661054	0,06598987
29600000	296	-4,78194E-06	300,3497078	-4714422927	30799219,81	308,0	11,99	5,01	0,064	17,00	0,003748897	0,06373125
29700000	297	-4,78194E-06	300,3497078	-4714522927	30816746,46	308,2	11,17	4,83	0,062	16,00	0,003843796	0,06150074
29800000	298	-4,78194E-06	300,3497078	-4714622927	30834812,02	308,3	10,35	4,65	0,059	15,00	0,003946776	0,05939251
29900000	299	-4,78194E-06	300,3497078	-4714722927	30853469,48	308,5	9,53	4,47	0,057	14,00	0,004059087	0,05827222
30000000	300	-4,78194E-06	300,3497078	-4714822927	30872781,13	308,7	8,73	4,27	0,054	13,00	0,004182273	0,05436955
30100000	301	-4,78194E-06	300,3497078	-4714922927	30892821,04	308,9	7,93	4,07	0,052	12,00	0,004318266	0,05181919
30200000	302	-4,78194E-06	300,3497078	-4715022927	30913678,38	309,1	7,14	3,86	0,049	11,00	0,004469528	0,0491648
30300000	303	-4,78194E-06	300,3497078	-4715122927	30935462,17	309,4	6,35	3,65	0,046	10,00	0,004639251	0,04639251
30400000	304	-4,78194E-06	300,3497078	-4715222927	30958308,08	309,6	5,58	3,42	0,043	9,00	0,004831672	0,04348505
30500000	305	-4,78194E-06	300,3497078	-4715322927	30982388,5	309,8	4,82	3,18	0,040	8,00	0,005052561	0,04042048
30600000	306	-4,78194E-06	300,3497078	-4715422927	31007928,21	310,1	4,08	2,92	0,037	7,00	0,005310029	0,0371702
30700000	307	-4,78194E-06	300,3497078	-4715522927	31035229,82	310,4	3,35	2,65	0,034	6,00	0,005615948	0,03369569
30800000	308	-4,78194E-06	300,3497078	-4715622927	31064717,71	310,6	2,65	2,35	0,030	5,00	0,005988589	0,02994295
30900000	309	-4,78194E-06	300,3497078	-4715722927	31097020,11	311,0	1,97	2,03	0,026	4,00	0,006458004	0,02583202
31000000	310	-4,78194E-06	300,3497078	-4715822927	31133140,05	311,3	1,33	1,67	0,021	3,00	0,007078417	0,02123525
31100000	311	-4,78194E-06	300,3497078	-4715922927	31174871,54	311,7	0,75	1,25	0,016	2,00	0,007962169	0,01592434
31200000	312	-4,78194E-06	300,3497078	-4716022927	31226108,91	312,3	0,26	0,74	0,009	1,00	0,00940367	0,00940367
31300000	313	-4,78194E-06	300,3497078	-4716122927	31300000	313	0,00	0,00	0,000	0,00	0	-9,624E-14



### NESS 2

P-tubing [Pa]	[Bar]	a	b	c	P-hullvegg [Pa]	[Bar]	Dp-punch [Bar]	Dp-formasjon [Bar]	q [m3/s]	Dp-tot [Bar]	PI	q-tot [m3/s]
0	0	-4,78662E-06	300,6422215	-4689400767	28845165,71	288,5	288,45	24,55	0,148	313,00	0,00095	0,297
20000000	200	-4,78662E-06	300,6422215	-4709400767	29864438,36	298,6	98,64	14,36	0,087	113,00	0,00154	0,174
21000000	210	-4,78662E-06	300,6422215	-4710400767	29933830,7	299,3	89,34	13,66	0,083	103,00	0,00160	0,165
22000000	220	-4,78662E-06	300,6422215	-4711400767	30006663,73	300,1	80,07	12,93	0,078	93,00	0,00168	0,156
23000000	230	-4,78662E-06	300,6422215	-4712400767	30083506,45	300,8	70,84	12,16	0,074	83,00	0,00177	0,147
24000000	240	-4,78662E-06	300,6422215	-4713400767	30165104,48	301,7	61,65	11,35	0,069	73,00	0,00188	0,137
25000000	250	-4,78662E-06	300,6422215	-4714400767	30252467,84	302,5	52,52	10,48	0,063	63,00	0,00201	0,127
26000000	260	-4,78662E-06	300,6422215	-4715400767	30347024,56	303,5	43,47	9,53	0,058	53,00	0,00217	0,115
27000000	270	-4,78662E-06	300,6422215	-4716400767	30450912,17	304,5	34,51	8,49	0,051	43,00	0,00239	0,103
28000000	280	-4,78662E-06	300,6422215	-4717400767	30567598,51	305,7	25,68	7,32	0,044	33,00	0,00268	0,089
29000000	290	-4,78662E-06	300,6422215	-4718400767	30703445,9	307,0	17,03	5,97	0,036	23,00	0,00314	0,072
29100000	291	-4,78662E-06	300,6422215	-4718500767	30718508,75	307,2	16,19	5,81	0,035	22,00	0,00320	0,070
29200000	292	-4,78662E-06	300,6422215	-4718600767	30733909,87	307,3	15,34	5,66	0,034	21,00	0,00326	0,068
29300000	293	-4,78662E-06	300,6422215	-4718700767	30749673,14	307,5	14,50	5,50	0,033	20,00	0,00333	0,067
29400000	294	-4,78662E-06	300,6422215	-4718800767	30765825,38	307,7	13,66	5,34	0,032	19,00	0,00340	0,065
29500000	295	-4,78662E-06	300,6422215	-4718900767	30782396,88	307,8	12,82	5,18	0,031	18,00	0,00348	0,063
29600000	296	-4,78662E-06	300,6422215	-4719000767	30799422,09	308,0	11,99	5,01	0,030	17,00	0,00356	0,061
29700000	297	-4,78662E-06	300,6422215	-4719100767	30816940,45	308,2	11,17	4,83	0,029	16,00	0,00365	0,058
29800000	298	-4,78662E-06	300,6422215	-4719200767	30834997,48	308,3	10,35	4,65	0,028	15,00	0,00375	0,056
29900000	299	-4,78662E-06	300,6422215	-4719300767	30853646,14	308,5	9,54	4,46	0,027	14,00	0,00386	0,054
30000000	300	-4,78662E-06	300,6422215	-4719400767	30872948,72	308,7	8,73	4,27	0,026	13,00	0,00397	0,052
30100000	301	-4,78662E-06	300,6422215	-4719500767	30892979,23	308,9	7,93	4,07	0,025	12,00	0,00410	0,049
30200000	302	-4,78662E-06	300,6422215	-4719600767	30913826,83	309,1	7,14	3,86	0,023	11,00	0,00425	0,047
30300000	303	-4,78662E-06	300,6422215	-4719700767	30935600,5	309,4	6,36	3,64	0,022	10,00	0,00441	0,044
30400000	304	-4,78662E-06	300,6422215	-4719800767	30958435,83	309,6	5,58	3,42	0,021	9,00	0,00459	0,041
30500000	305	-4,78662E-06	300,6422215	-4719900767	30982505,18	309,8	4,83	3,17	0,019	8,00	0,00480	0,038
30600000	306	-4,78662E-06	300,6422215	-4720000767	31008033,23	310,1	4,08	2,92	0,018	7,00	0,00505	0,035
30700000	307	-4,78662E-06	300,6422215	-4720100767	31035322,51	310,4	3,35	2,65	0,016	6,00	0,00534	0,032
30800000	308	-4,78662E-06	300,6422215	-4720200767	31064797,25	310,6	2,65	2,35	0,014	5,00	0,00569	0,028
30900000	309	-4,78662E-06	300,6422215	-4720300767	31097085,53	311,0	1,97	2,03	0,012	4,00	0,00614	0,025
31000000	310	-4,78662E-06	300,6422215	-4720400767	31133190,14	311,3	1,33	1,67	0,010	3,00	0,00673	0,020
31100000	311	-4,78662E-06	300,6422215	-4720500767	31174904,82	311,7	0,75	1,25	0,008	2,00	0,00757	0,015
31200000	312	-4,78662E-06	300,6422215	-4720600767	31226123,85	312,3	0,26	0,74	0,004	1,00	0,00894	0,009
31300000	313	-4,78662E-06	300,6422215	-4720700767	31300000	313,0	0,00	0,00	0,000	0,00	0,00000	0,000



### NESS 3

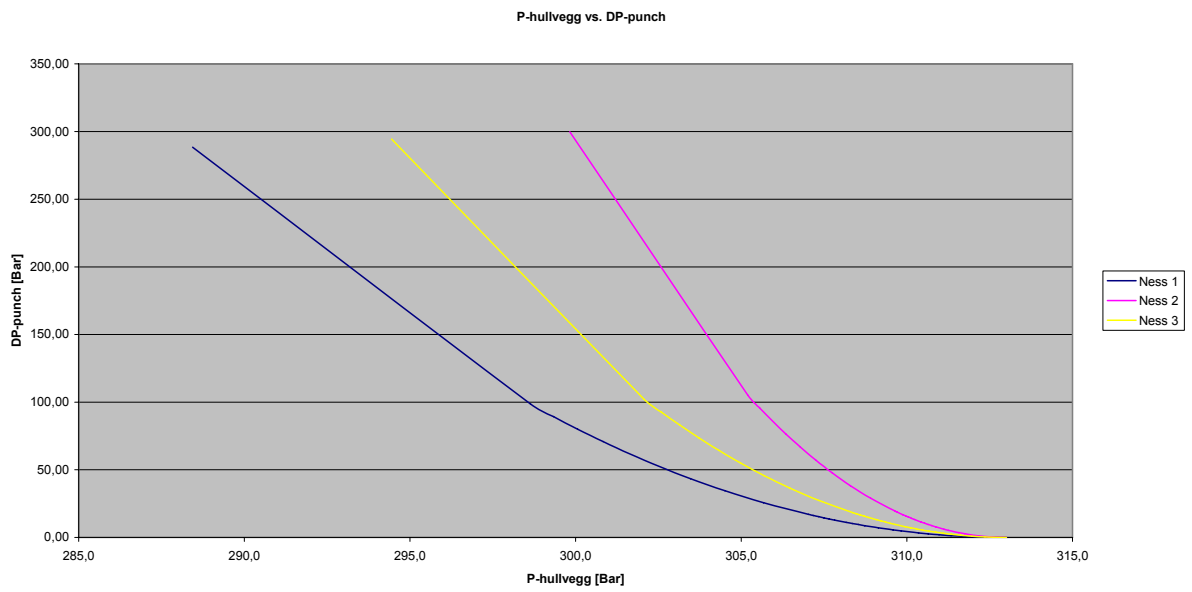
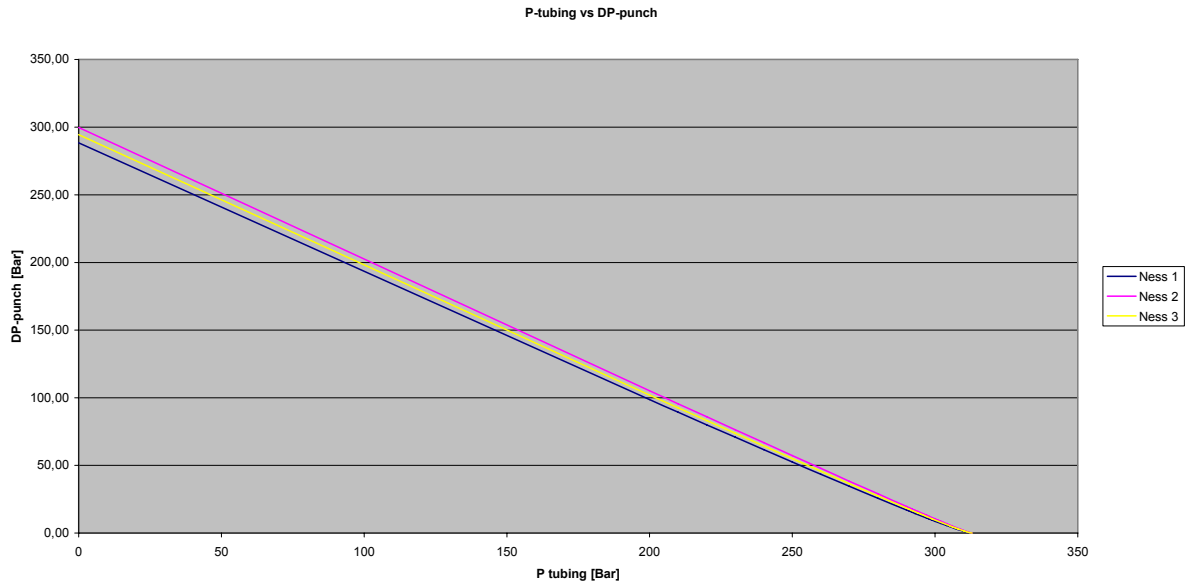
P-tubing [Pa]	[Bar]	a	b	c	P-hullvegg [Pa]	[Bar]	Dp-punch [Bar]	Dp-formasjon [Bar]	q [m3/s]	Dp-tot [Bar]	PI	q-tot [m3/s]
0	0	-5,05985E-06	317,7463111	-4957079768	28909695,96	289,1	289,10	23,90	0,102	313,00	0,000649824	0,20339494
20000000	200	-5,05985E-06	317,7463111	-4977079768	29901140,8	299,0	99,01	13,99	0,060	113,00	0,001053374	0,11903125
21000000	210	-5,05985E-06	317,7463111	-4978079768	29968642,35	299,7	89,69	13,31	0,057	103,00	0,001099878	0,11328743
22000000	220	-5,05985E-06	317,7463111	-4979079768	30039491,78	300,4	80,39	12,61	0,054	93,00	0,00115332	0,10725874
23000000	230	-5,05985E-06	317,7463111	-4980079768	30114242,91	301,1	71,14	11,86	0,050	83,00	0,001215639	0,10089804
24000000	240	-5,05985E-06	317,7463111	-4981079768	30193621,53	301,9	61,94	11,06	0,047	73,00	0,001289638	0,09414358
25000000	250	-5,05985E-06	317,7463111	-4982079768	30278610,92	302,8	52,79	10,21	0,043	63,00	0,001379551	0,08691169
26000000	260	-5,05985E-06	317,7463111	-4983079768	30370601,46	303,7	43,71	9,29	0,040	53,00	0,001492152	0,07908407
27000000	270	-5,05985E-06	317,7463111	-4984079768	30471674,75	304,7	34,72	8,28	0,035	43,00	0,001639153	0,07048357
28000000	280	-5,05985E-06	317,7463111	-4985079768	30585208,8	305,9	25,85	7,15	0,030	33,00	0,001843114	0,06082277
29000000	290	-5,05985E-06	317,7463111	-4986079768	30717403,86	307,2	17,17	5,83	0,025	23,00	0,002155394	0,04957407
29100000	291	-5,05985E-06	317,7463111	-4986179768	30732063,36	307,3	16,32	5,68	0,024	22,00	0,002196667	0,04832667
29200000	292	-5,05985E-06	317,7463111	-4986279768	30747052,49	307,5	15,47	5,53	0,024	21,00	0,002240534	0,04705122
29300000	293	-5,05985E-06	317,7463111	-4986379768	30762394,55	307,6	14,62	5,38	0,023	20,00	0,002287287	0,04574574
29400000	294	-5,05985E-06	317,7463111	-4986479768	30778115,71	307,8	13,78	5,22	0,022	19,00	0,002337263	0,0444408
29500000	295	-5,05985E-06	317,7463111	-4986579768	30794245,54	307,9	12,94	5,06	0,022	18,00	0,00239086	0,04303549
29600000	296	-5,05985E-06	317,7463111	-4986679768	30810817,67	308,1	12,11	4,89	0,021	17,00	0,002448549	0,04162534
29700000	297	-5,05985E-06	317,7463111	-4986779768	30827870,62	308,3	11,28	4,72	0,020	16,00	0,002510892	0,04017427
29800000	298	-5,05985E-06	317,7463111	-4986879768	30845448,83	308,5	10,45	4,55	0,019	15,00	0,002578568	0,03867851
29900000	299	-5,05985E-06	317,7463111	-4986979768	30863604,07	308,6	9,64	4,36	0,019	14,00	0,002652404	0,03713366
30000000	300	-5,05985E-06	317,7463111	-4987079768	30882397,17	308,8	8,82	4,18	0,018	13,00	0,002733425	0,03553452
30100000	301	-5,05985E-06	317,7463111	-4987179768	30901900,5	309,0	8,02	3,98	0,017	12,00	0,002822912	0,03387495
30200000	302	-5,05985E-06	317,7463111	-4987279768	30922201,26	309,2	7,22	3,78	0,016	11,00	0,002922502	0,03214752
30300000	303	-5,05985E-06	317,7463111	-4987379768	30943406,06	309,4	6,43	3,57	0,015	10,00	0,003034317	0,03034317
30400000	304	-5,05985E-06	317,7463111	-4987479768	30965647,65	309,7	5,66	3,34	0,014	9,00	0,003161177	0,0284506
30500000	305	-5,05985E-06	317,7463111	-4987579768	30989094,84	309,9	4,89	3,11	0,013	8,00	0,00330693	0,02645544
30600000	306	-5,05985E-06	317,7463111	-4987679768	31013967,93	310,1	4,14	2,86	0,012	7,00	0,003476992	0,02433894
30700000	307	-5,05985E-06	317,7463111	-4987779768	31040563,78	310,4	3,41	2,59	0,011	6,00	0,00367931	0,02207586
30800000	308	-5,05985E-06	317,7463111	-4987879768	31069299,27	310,7	2,69	2,31	0,010	5,00	0,003926142	0,01963071
30900000	309	-5,05985E-06	317,7463111	-4987979768	31100792,67	311,0	2,01	1,99	0,008	4,00	0,004237721	0,01695088
31000000	310	-5,05985E-06	317,7463111	-4988079768	31136033,69	311,4	1,36	1,64	0,007	3,00	0,004650722	0,01395217
31100000	311	-5,05985E-06	317,7463111	-4988179768	31176799,8	311,8	0,77	1,23	0,005	2,00	0,005241655	0,01048331
31200000	312	-5,05985E-06	317,7463111	-4988279768	31226979,25	312,3	0,27	0,73	0,003	1,00	0,006213457	0,00621346
31300000	313	-5,05985E-06	317,7463111	-4988379768	31300000	313	0,00	0,00	0,000	0,00	#DIV/0!	0



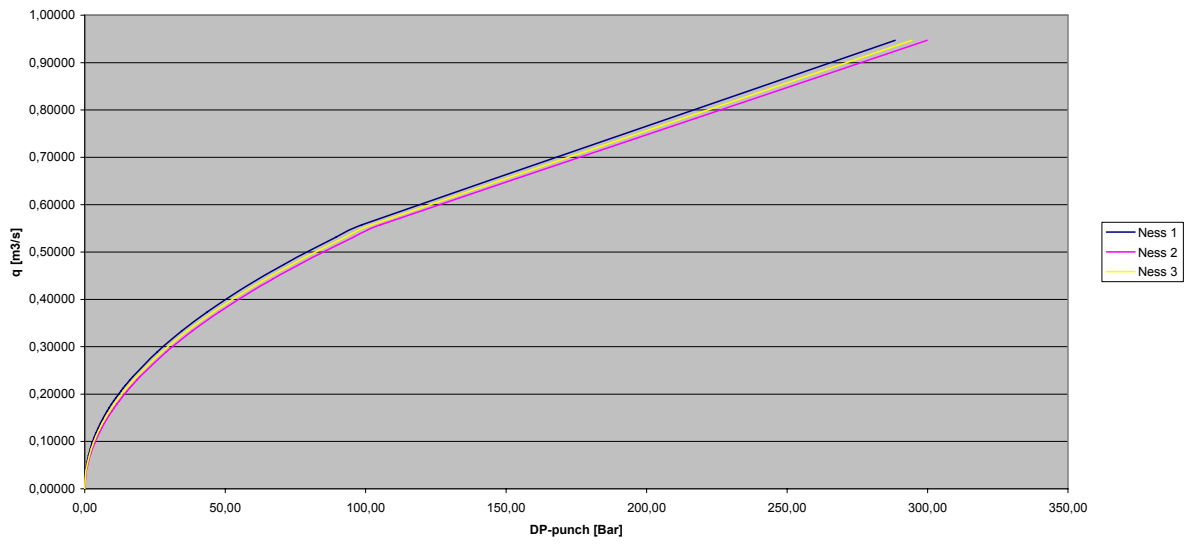
## TOTAL TRYKKBALANSE

Re	f	Dp-tubing [Bar]	Pw-min [Bar]	P-tubing [bar]	DP-tubing max [bar]	DP-Punch [bar]			DP-tot [Bar]	q [m3/s]	q-max [m3/s]	q [m3/d]	Check OK/NOT OK	PI [m3/d/bar]
						SONE 1	SONE 2	SONE 3						
9472101,312	0,003044299	1176,721912	1399,175512	0	-222,4536	288,44	288,45	289,10	313,00	0,81289	0	70233,6603	NOT OK	224,388691
5540183,927	0,003051646	403,530387	625,983987	200	-22,4536	98,64	98,64	99,01	113,00	0,47545	0	41079,311	NOT OK	363,5337258
5272494,403	0,00305254	365,5841622	588,0377622	210	-12,4536	89,33	89,34	89,69	103,00	0,45248	0	39094,4489	NOT OK	379,5577563
4991531,085	0,00305358	327,7711275	550,2247275	220	-2,4536	80,06	80,07	80,39	93,00	0,42837	0	37011,1644	NOT OK	397,9695092
4695098,656	0,003054811	290,113311	512,566911	230	7,5464	70,83	70,84	71,14	83,00	0,40293	0,065576406	34813,1796	NOT OK	419,4358985
4380320,375	0,003056299	252,639579	475,093179	240	17,5464	61,65	61,65	61,94	73,00	0,37592	0,099993487	32479,1641	NOT OK	444,9200565
4043299,138	0,003058145	215,3890365	437,8426365	250	27,5464	52,52	52,52	52,79	63,00	0,34699	0,125288187	29980,2218	NOT OK	475,8765362
3678525,01	0,003060518	178,4169731	400,8705731	260	37,5464	43,47	43,47	43,71	53,00	0,31569	0,14627222	27275,4975	NOT OK	514,6320287
3277749,899	0,003063724	141,8061552	364,2597552	270	47,5464	34,51	34,51	34,72	43,00	0,28129	0,164602535	24303,8335	NOT OK	565,2054295
2827591,748	0,003068388	105,6908699	328,1444699	280	57,5464	25,67	25,68	25,85	33,00	0,24266	0,181086794	20966,0044	NOT OK	635,3334681
2303496,205	0,00307606	70,31752517	292,7711252	290	67,5464	17,03	17,03	17,17	23,00	0,19768	0,196190875	17079,945	NOT OK	742,6063065
2245382,594	0,003077126	66,83743074	289,2910307	291	68,5464	16,18	16,19	16,32	22,00	0,19270	0,197637807	16649,0447	OK	756,7747597
2185963,474	0,003078273	63,37043731	285,8240373	292	69,5464	15,34	15,34	15,47	21,00	0,18760	0,199074222	16208,4643	OK	771,8316337
2125146,699	0,003079512	59,91746971	282,3710697	293	70,5464	14,49	14,50	14,62	20,00	0,18238	0,200500347	15757,5206	OK	787,8760288
2062828,741	0,003080855	56,479567	278,933167	294	71,5464	13,66	13,66	13,78	19,00	0,17703	0,201916399	15295,4459	OK	805,0234674
1998892,622	0,003082317	53,05790322	275,5115032	295	72,5464	12,82	12,82	12,94	18,00	0,17154	0,20332259	14821,3729	OK	823,4096037
1933205,337	0,003083918	49,65381334	272,1074133	296	73,5464	11,99	11,99	12,11	17,00	0,16591	0,204719122	14334,3153	OK	843,1950184
1865614,592	0,00308568	46,26882593	268,7224259	297	74,5464	11,17	11,17	11,28	16,00	0,16011	0,206106191	13833,144	OK	864,5715
1795944,65	0,00308763	42,90470489	265,3583049	298	75,5464	10,35	10,35	10,45	15,00	0,15413	0,207483988	13316,5559	OK	887,770392
1723990,946	0,003089804	39,56350357	262,0171036	299	76,5464	9,53	9,54	9,64	14,00	0,14795	0,208852696	12783,0341	OK	913,0738668
1649512,99	0,003092249	36,24763595	258,701236	300	77,5464	8,73	8,73	8,82	13,00	0,14156	0,210212492	12230,7956	OK	940,8304301
1572224,869	0,003095022	32,9599721	255,4135721	301	78,5464	7,93	7,93	8,02	12,00	0,13493	0,211563549	11657,7203	OK	971,476688
1491782,236	0,003098203	29,70396879	252,1575688	302	79,5464	7,14	7,14	7,22	11,00	0,12802	0,212906032	11061,2549	OK	1005,568625
1407764,089	0,003101899	26,48385253	248,9374525	303	80,5464	6,35	6,36	6,43	10,00	0,12081	0,214240103	10438,2778	OK	1043,827779
1319646,511	0,003106261	23,30488334	245,7584833	304	81,5464	5,58	5,58	5,66	9,00	0,11325	0,215565917	9784,90428	OK	1087,211587
1226763,577	0,003111509	20,17374755	242,6273475	305	82,5464	4,82	4,83	4,89	8,00	0,10528	0,216883628	9096,19666	OK	1137,024583
1128246,804	0,003117977	17,09916619	239,5527662	306	83,5464	4,08	4,08	4,14	7,00	0,09683	0,21819338	8365,71529	OK	1195,102184
1022926,702	0,003126202	14,09288459	236,5464846	307	84,5464	3,35	3,35	3,41	6,00	0,08779	0,219495318	7584,78864	OK	1264,13144
909162,6647	0,003137111	11,17138273	233,6249827	308	85,5464	2,65	2,65	2,69	5,00	0,07802	0,220789578	6741,25198	OK	1348,250396
784524,7893	0,003152471	8,359076326	230,8126763	309	86,5464	1,97	1,97	2,01	4,00	0,06733	0,222076295	5817,08807	OK	1454,272017
645130,9054	0,003176169	5,694992094	228,1485921	310	87,5464	1,33	1,33	1,36	3,00	0,05536	0,2233556	4783,51142	OK	1594,503808
484030,0541	0,003218997	3,24907007	225,7026701	311	88,5464	0,75	0,75	0,77	2,00	0,04154	0,22462762	3588,98213	OK	1794,491067
286103,036	0,003328537	1,173794558	223,6273946	312	89,5464	0,26	0,26	0,27	1,00	0,02455	0,225892476	2121,39448	OK	2121,394479
-1,12144E-06	0	0	222,4536	313	90,5464	0,00	0,00	0,00	0,00	0,00000	0,227150289	-8,315E-09	OK	0

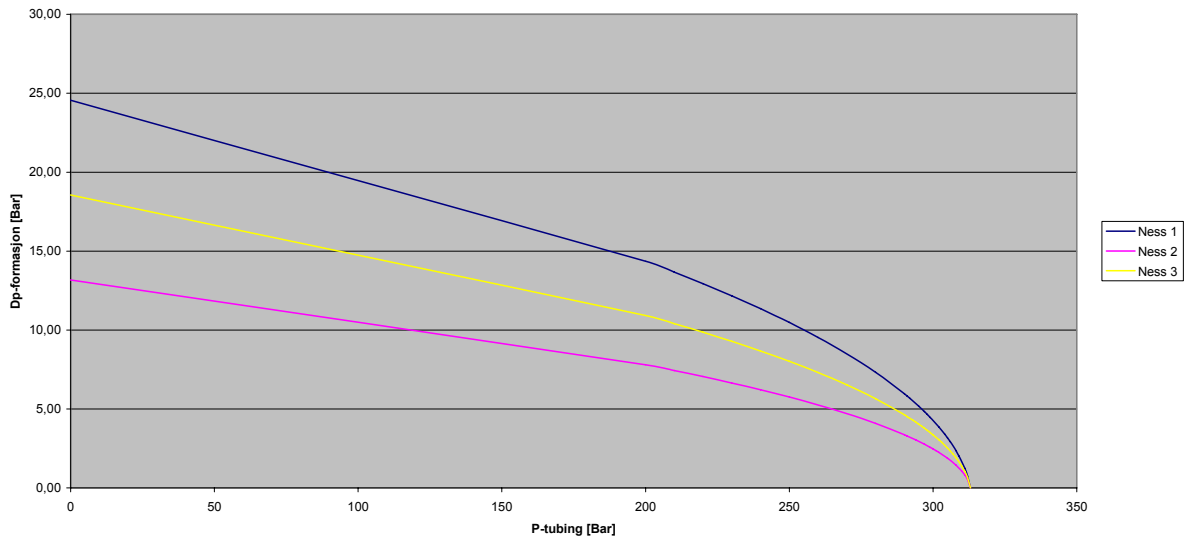
# Resultat, ikke optimalisert



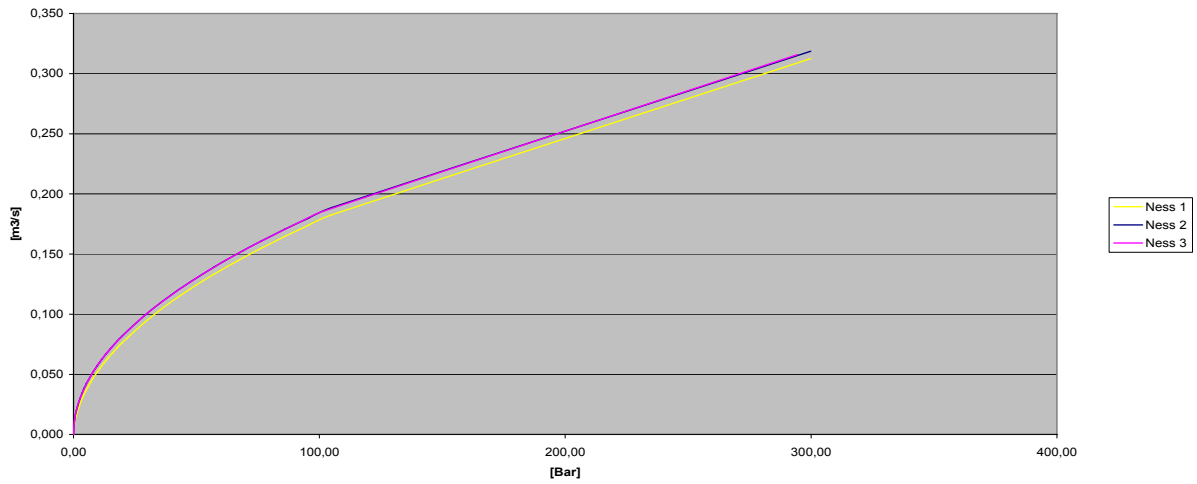
DP-punch vs Q-tot



P-tubing vs DP-formasjon



DP-Punch vs q-punch



## Kildekode, ikke optimalisert

### Faste variable:

Pres	31300000 Pa
Dy	0,196 m
Di	0,1778 m
Ah	0,005343189 m2
A-innstr	4,556565985 m2
A-screen inflow	0,407 m2
d-hull	0,01016 m
L-screen	7,4 m
Pw-min	22245360 Pa
Turb	>2300
Lam	<2300
Dh	0,0364
Phi 1	0,089321652
A-choke	0,837862761 m2
A-hull	8,10732E-05 m2
D-tubing	6,184 "
	0,1570736 m
L	10000 m
TVD	2500 m



SONE 1			SONE 2			SONE 3		
# hull	20	stk/seksjon	# hull	20	stk/seksjon	# hull	20	stk/seksjon
Antall skjerm	1	stk	Antall skjerm	1	stk	Antall skjerm	1	stk
Reelt # hull	20	stk/seksjon	Reelt # hull	20	stk/seksjon	Reelt # hull	20	stk/seksjon
Reelt # skjerm	1	stk	Reelt # skjerm	1	stk	Reelt # skjerm	1	stk
Lengde Punch	12,5	m	Lengde Punch	12,5	m	Lengde Punch	12,5	m
A	1,27264E-07		A	2,4192E-07		A	1,70183E-07	
B	0,000213396		B	0,000213396		B	0,000213396	
C	0,000904366		C	0,000904366		C	0,000904366	
Phi 2	0,001935238		Phi 2	0,001935238		Phi 2	0,001935238	

## NESS 1

P-tubing	a	b	c	P-hullvegg	Dp-punch	Dp-formasjon	q	Dp-tot	PI	q-tot	
[Pa]	[Bar]			[Pa]	[Bar]	[Bar]	[m3/s]	[Bar]		[m3/s]	
0	0	-4,78194E-06	300,3497078	-4684822927	288,4	288,44	24,56	0,313	313,00	0,000998589	0,3125582
20000000	200	-4,78194E-06	300,3497078	-4704822927	298,6	98,64	14,36	0,183	113,00	0,001617508	0,18277838
21000000	210	-4,78194E-06	300,3497078	-4705822927	299,3	89,33	13,67	0,174	103,00	0,001688767	0,17394295
22000000	220	-4,78194E-06	300,3497078	-4706822927	300,1	80,06	12,94	0,165	93,00	0,001770639	0,16466943
23000000	230	-4,78194E-06	300,3497078	-4707822927	300,8	70,83	12,17	0,155	83,00	0,001866089	0,15488537
24000000	240	-4,78194E-06	300,3497078	-4708822927	301,6	61,65	11,35	0,144	73,00	0,001979395	0,14449585
25000000	250	-4,78194E-06	300,3497078	-4709822927	302,5	52,52	10,48	0,133	63,00	0,00211702	0,13337226
26000000	260	-4,78194E-06	300,3497078	-4710822927	303,5	43,47	9,53	0,121	53,00	0,002289298	0,12133278
27000000	270	-4,78194E-06	300,3497078	-4711822927	304,5	34,51	8,49	0,108	43,00	0,002514075	0,10810524
28000000	280	-4,78194E-06	300,3497078	-4712822927	305,7	25,67	7,33	0,093	33,00	0,002825701	0,09324813
29000000	290	-4,78194E-06	300,3497078	-4713822927	307,0	17,03	5,97	0,076	23,00	0,003302233	0,07595137
29100000	291	-4,78194E-06	300,3497078	-4713922927	307,2	16,18	5,82	0,074	22,00	0,003365159	0,0740335
29200000	292	-4,78194E-06	300,3497078	-4714022927	307,3	15,34	5,66	0,072	21,00	0,003432026	0,07207255
29300000	293	-4,78194E-06	300,3497078	-4714122927	307,5	14,49	5,51	0,070	20,00	0,003503275	0,0700655
29400000	294	-4,78194E-06	300,3497078	-4714222927	307,7	13,66	5,34	0,068	19,00	0,003579417	0,06800893
29500000	295	-4,78194E-06	300,3497078	-4714322927	307,8	12,82	5,18	0,066	18,00	0,003661054	0,06598987
29600000	296	-4,78194E-06	300,3497078	-4714422927	308,0	11,99	5,01	0,064	17,00	0,003748897	0,06373125
29700000	297	-4,78194E-06	300,3497078	-4714522927	308,2	11,17	4,83	0,062	16,00	0,003843796	0,06150074
29800000	298	-4,78194E-06	300,3497078	-4714622927	308,3	10,35	4,65	0,059	15,00	0,003946776	0,05920164
29900000	299	-4,78194E-06	300,3497078	-4714722927	308,5	9,53	4,47	0,057	14,00	0,004059087	0,05682722
30000000	300	-4,78194E-06	300,3497078	-4714822927	308,7	8,73	4,27	0,054	13,00	0,004182273	0,05436955
30100000	301	-4,78194E-06	300,3497078	-4714922927	308,9	7,93	4,07	0,052	12,00	0,004318266	0,05181919
30200000	302	-4,78194E-06	300,3497078	-4715022927	309,1	7,14	3,86	0,049	11,00	0,004469528	0,0491648
30300000	303	-4,78194E-06	300,3497078	-4715122927	309,4	6,35	3,65	0,046	10,00	0,004639251	0,04639251
30400000	304	-4,78194E-06	300,3497078	-4715222927	309,6	5,58	3,42	0,043	9,00	0,004831672	0,04348505
30500000	305	-4,78194E-06	300,3497078	-4715322927	309,8	4,82	3,18	0,040	8,00	0,005052561	0,04042048
30600000	306	-4,78194E-06	300,3497078	-4715422927	310,1	4,08	2,92	0,037	7,00	0,005310029	0,0371702
30700000	307	-4,78194E-06	300,3497078	-4715522927	310,4	3,35	2,65	0,034	6,00	0,005615948	0,03369569
30800000	308	-4,78194E-06	300,3497078	-4715622927	310,6	2,65	2,35	0,030	5,00	0,005988589	0,02994295
30900000	309	-4,78194E-06	300,3497078	-4715722927	310,9	1,97	2,03	0,026	4,00	0,006458004	0,02583202
31000000	310	-4,78194E-06	300,3497078	-4715822927	311,3	1,33	1,67	0,021	3,00	0,007078417	0,02123525
31100000	311	-4,78194E-06	300,3497078	-4715922927	311,7	0,75	1,25	0,016	2,00	0,007962169	0,01592434
31200000	312	-4,78194E-06	300,3497078	-4716022927	312,3	0,26	0,74	0,009	1,00	0,00940367	0,00940367
31300000	313	-4,78194E-06	300,3497078	-4716122927	31300000	313	0,00	0,000	0,00	0	-9,624E-14



## NESS 2

P-tubing [Pa]	[Bar]	a	b	c	P-hullvegg [Pa]	[Bar]	Dp-punch [Bar]	Dp-formasjon [Bar]	q [m3/s]	Dp-tot [Bar]	PI	q-tot [m3/s]
0	0	-1,72797E-05	1082,710094	-16928762966	29982752,23	299,8	299,83	13,17	0,319	313,00	0,00102	0,319
20000000	200	-1,72797E-05	1082,710094	-16948762966	30519748,86	305,2	105,20	7,80	0,189	113,00	0,00167	0,189
21000000	210	-1,72797E-05	1082,710094	-16949762966	30556334,89	305,6	95,56	7,44	0,180	103,00	0,00175	0,180
22000000	220	-1,72797E-05	1082,710094	-16950762966	30594741,8	305,9	85,95	7,05	0,171	93,00	0,00183	0,171
23000000	230	-1,72797E-05	1082,710094	-16951762966	30635271,99	306,4	76,35	6,65	0,161	83,00	0,00194	0,161
24000000	240	-1,72797E-05	1082,710094	-16952762966	30678322,13	306,8	66,78	6,22	0,150	73,00	0,00206	0,150
25000000	250	-1,72797E-05	1082,710094	-16953762966	30724430,38	307,2	57,24	5,76	0,139	63,00	0,00221	0,139
26000000	260	-1,72797E-05	1082,710094	-16954762966	30774358,97	307,7	47,74	5,26	0,127	53,00	0,00240	0,127
27000000	270	-1,72797E-05	1082,710094	-16955762966	30829251,78	308,3	38,29	4,71	0,114	43,00	0,00265	0,114
28000000	280	-1,72797E-05	1082,710094	-16956762966	30890971,45	308,9	28,91	4,09	0,099	33,00	0,00300	0,099
29000000	290	-1,72797E-05	1082,710094	-16957762966	30962955,71	309,6	19,63	3,37	0,082	23,00	0,00355	0,082
29100000	291	-1,72797E-05	1082,710094	-16957862966	30970949,36	309,7	18,71	3,29	0,080	22,00	0,00362	0,080
29200000	292	-1,72797E-05	1082,710094	-16957962966	30979125,63	309,8	17,79	3,21	0,078	21,00	0,00370	0,078
29300000	293	-1,72797E-05	1082,710094	-16958062966	30987497,63	309,9	16,87	3,13	0,076	20,00	0,00378	0,076
29400000	294	-1,72797E-05	1082,710094	-16958162966	30996080,15	310,0	15,96	3,04	0,074	19,00	0,00387	0,074
29500000	295	-1,72797E-05	1082,710094	-16958262966	31004889,89	310,0	15,05	2,95	0,071	18,00	0,00397	0,071
29600000	296	-1,72797E-05	1082,710094	-16958362966	31013945,93	310,1	14,14	2,86	0,069	17,00	0,00407	0,069
29700000	297	-1,72797E-05	1082,710094	-16958462966	31023270,16	310,2	13,23	2,77	0,067	16,00	0,00418	0,067
29800000	298	-1,72797E-05	1082,710094	-16958562966	31032887,92	310,3	12,33	2,67	0,065	15,00	0,00431	0,065
29900000	299	-1,72797E-05	1082,710094	-16958662966	31042828,81	310,4	11,43	2,57	0,062	14,00	0,00444	0,062
30000000	300	-1,72797E-05	1082,710094	-16958762966	31053127,76	310,5	10,53	2,47	0,060	13,00	0,00459	0,060
30100000	301	-1,72797E-05	1082,710094	-16958862966	31063826,5	310,6	9,64	2,36	0,057	12,00	0,00476	0,057
30200000	302	-1,72797E-05	1082,710094	-16958962966	31074975,56	310,7	8,75	2,25	0,054	11,00	0,00495	0,054
30300000	303	-1,72797E-05	1082,710094	-16959062966	31086637,09	310,9	7,87	2,13	0,052	10,00	0,00516	0,052
30400000	304	-1,72797E-05	1082,710094	-16959162966	31098889	311,0	6,99	2,01	0,049	9,00	0,00541	0,049
30500000	305	-1,72797E-05	1082,710094	-16959262966	31111831,23	311,1	6,12	1,88	0,046	8,00	0,00569	0,046
30600000	306	-1,72797E-05	1082,710094	-16959362966	31125595,55	311,3	5,26	1,74	0,042	7,00	0,00603	0,042
30700000	307	-1,72797E-05	1082,710094	-16959462966	31140361,89	311,4	4,40	1,60	0,039	6,00	0,00644	0,039
30800000	308	-1,72797E-05	1082,710094	-16959562966	31156387,3	311,6	3,56	1,44	0,035	5,00	0,00695	0,035
30900000	309	-1,72797E-05	1082,710094	-16959662966	31174062,15	311,7	2,74	1,26	0,030	4,00	0,00762	0,030
31000000	310	-1,72797E-05	1082,710094	-16959762966	31194033,21	311,9	1,94	1,06	0,026	3,00	0,00855	0,026
31100000	311	-1,72797E-05	1082,710094	-16959862966	31217528,58	312,2	1,18	0,82	0,020	2,00	0,00998	0,020
31200000	312	-1,72797E-05	1082,710094	-16959962966	31247545,21	312,5	0,48	0,52	0,013	1,00	0,01269	0,013
31300000	313	-1,72797E-05	1082,710094	-16960062966	31300000	313,0	0,00	0,00	0,000	0,00	0,00000	0,000





### NESS 3

P-tubing [Pa]	[Bar]	a	b	c	P-hullvegg [Pa]	[Bar]	Dp-punch [Bar]	Dp-formasjon [Bar]	q [m3/s]	Dp-tot [Bar]	PI	q-tot [m3/s]
0	0	-8,55122E-06	536,3061722	-8377541596	29444387,44	294,4	294,44	18,56	0,316	313,00	0,001008928	0,31579431
20000000	200	-8,55122E-06	536,3061722	-8397541596	30207442	302,1	102,07	10,93	0,186	113,00	0,001645444	0,18593515
21000000	210	-8,55122E-06	536,3061722	-8398541596	30259414,42	302,6	92,59	10,41	0,177	103,00	0,001719323	0,17709031
22000000	220	-8,55122E-06	536,3061722	-8399541596	30313969,7	303,1	83,14	9,86	0,168	93,00	0,001804365	0,16780591
23000000	230	-8,55122E-06	536,3061722	-8400541596	30371536,07	303,7	73,72	9,28	0,158	83,00	0,001903724	0,15809098
24000000	240	-8,55122E-06	536,3061722	-8401541596	30432675,03	304,3	64,33	8,67	0,148	73,00	0,002021976	0,14760424
25000000	250	-8,55122E-06	536,3061722	-8402541596	30498147,94	305,0	54,98	8,02	0,136	63,00	0,002166061	0,13646185
26000000	260	-8,55122E-06	536,3061722	-8403541596	30569032,35	305,7	45,69	7,31	0,124	53,00	0,002347142	0,1243985
27000000	270	-8,55122E-06	536,3061722	-8404541596	30646943,77	306,5	36,47	6,53	0,111	43,00	0,002584634	0,11113928
28000000	280	-8,55122E-06	536,3061722	-8405541596	30734508,97	307,3	27,35	5,65	0,096	33,00	0,002916277	0,09623714
29000000	290	-8,55122E-06	536,3061722	-8406541596	30836564,66	308,4	18,37	4,63	0,079	23,00	0,003429086	0,07886897
29100000	291	-8,55122E-06	536,3061722	-8406641596	30847890,94	308,5	17,48	4,52	0,077	22,00	0,003497337	0,07694142
29200000	292	-8,55122E-06	536,3061722	-8406741596	30859474,25	308,6	16,59	4,41	0,075	21,00	0,003570006	0,07497014
29300000	293	-8,55122E-06	536,3061722	-8406841596	30871332,91	308,7	15,71	4,29	0,073	20,00	0,003647599	0,07295199
29400000	294	-8,55122E-06	536,3061722	-8406941596	30883487,54	308,8	14,83	4,17	0,071	19,00	0,003730709	0,07088347
29500000	295	-8,55122E-06	536,3061722	-8407041596	30895961,49	309,0	13,96	4,04	0,069	18,00	0,003820034	0,06876062
29600000	296	-8,55122E-06	536,3061722	-8407141596	30908781,32	309,1	13,09	3,91	0,067	17,00	0,003916406	0,0665789
29700000	297	-8,55122E-06	536,3061722	-8407241596	30921977,5	309,2	12,22	3,78	0,064	16,00	0,00402082	0,06433312
29800000	298	-8,55122E-06	536,3061722	-8407341596	30935585,27	309,4	11,36	3,64	0,062	15,00	0,004134487	0,06201731
29900000	299	-8,55122E-06	536,3061722	-8407441596	30949645,73	309,5	10,50	3,50	0,060	14,00	0,00425889	0,05962445
30000000	300	-8,55122E-06	536,3061722	-8407541596	30964207,3	309,6	9,64	3,36	0,057	13,00	0,004395871	0,05714632
30100000	301	-8,55122E-06	536,3061722	-8407641596	30979327,72	309,8	8,79	3,21	0,055	12,00	0,004547756	0,05457307
30200000	302	-8,55122E-06	536,3061722	-8407741596	30995076,73	310,0	7,95	3,05	0,052	11,00	0,004717532	0,05189285
30300000	303	-8,55122E-06	536,3061722	-8407841596	31011539,94	310,1	7,12	2,88	0,049	10,00	0,004909109	0,04909109
30400000	304	-8,55122E-06	536,3061722	-8407941596	31028824,34	310,3	6,29	2,71	0,046	9,00	0,005127731	0,04614958
30500000	305	-8,55122E-06	536,3061722	-8408041596	31047066,62	310,5	5,47	2,53	0,043	8,00	0,00538063	0,04304504
30600000	306	-8,55122E-06	536,3061722	-8408141596	31066446,25	310,7	4,66	2,34	0,040	7,00	0,005678136	0,03974695
30700000	307	-8,55122E-06	536,3061722	-8408241596	31087206,88	310,9	3,87	2,13	0,036	6,00	0,00603564	0,03621384
30800000	308	-8,55122E-06	536,3061722	-8408341596	31109694,01	311,1	3,10	1,90	0,032	5,00	0,006477381	0,0323869
30900000	309	-8,55122E-06	536,3061722	-8408441596	31134426,9	311,3	2,34	1,66	0,028	4,00	0,007044445	0,02817778
31000000	310	-8,55122E-06	536,3061722	-8408541596	31162252,94	311,6	1,62	1,38	0,023	3,00	0,007814084	0,02344225
31100000	311	-8,55122E-06	536,3061722	-8408641596	31194741,66	311,9	0,95	1,05	0,018	2,00	0,008956607	0,01791321
31200000	312	-8,55122E-06	536,3061722	-8408741596	31235535,76	312,4	0,36	0,64	0,011	1,00	0,010970738	0,01097074
31300000	313	-8,55122E-06	536,3061722	-8408841596	31300000	313	0,00	0,00	0,000	0,00	32,63494104	2,891E-13



## TOTAL TRYKKBALANSE

Re	f	Dp-tubing [Bar]	Pw-min [Bar]	P-tubing [bar]	DP-tubing max [bar]	DP-Punch [bar]			DP-tot [Bar]	q [m3/s]	q-max [m3/s]	q [m3/d]	Check OK/NOT OK	PI [m3/d/bar]
						SONE 1	SONE 2	SONE 3						
11035049,17	0,003042826	1596,318154	1818,771754	0	-222,4536	288,44	299,83	294,44	313,00	0,94702	0	81822,5935	NOT OK	261,4140366
6495874	0,003049049	554,2853402	776,7389402	200	-22,4536	98,64	105,20	102,07	113,00	0,55747	0	48165,5541	NOT OK	426,2438412
6186722,786	0,003049802	502,9059264	725,3595264	210	-12,4536	89,33	95,56	92,59	103,00	0,53094	0	45873,2621	NOT OK	445,3714769
5862211,952	0,003050677	451,6614763	674,1150763	220	-2,4536	80,06	85,95	83,14	93,00	0,50309	0	43467,0818	NOT OK	467,3879765
5519795,689	0,003051711	400,5743589	623,0279589	230	7,5464	70,83	76,35	73,72	83,00	0,47371	0,065576406	40928,1365	NOT OK	493,110078
5156136,031	0,003052957	349,6739035	572,1275035	240	17,5464	61,65	66,78	64,33	73,00	0,44250	0,099993487	38231,6758	NOT OK	523,7215861
4766707,722	0,0030545	298,9998703	521,4534703	250	27,5464	52,52	57,24	54,98	63,00	0,40908	0,125288187	35344,1459	NOT OK	561,0181889
4345106,539	0,003056479	248,6085279	471,0621279	260	37,5464	43,47	47,74	45,69	53,00	0,37289	0,14627222	32218,0609	NOT OK	607,8879407
3881733,039	0,003059142	198,5842152	421,0378152	270	47,5464	34,51	38,29	36,47	43,00	0,33313	0,164602535	28782,2428	NOT OK	669,354483
3360982,68	0,003062996	149,0640227	371,5176227	280	57,5464	25,67	28,91	27,35	33,00	0,28844	0,181086794	24920,9872	NOT OK	755,1814309
2754134,133	0,003069291	100,3002476	322,7538476	290	67,5464	17,03	19,63	18,37	23,00	0,23636	0,196190875	20421,3315	NOT OK	887,883979
2686792,226	0,003070161	95,48235128	317,9359513	291	68,5464	16,18	18,71	17,48	22,00	0,23058	0,197637807	19922,0053	NOT OK	905,5456949
2617923,978	0,003071096	90,67786266	313,1314627	292	69,5464	15,34	17,79	16,59	21,00	0,22467	0,199074222	19411,3616	NOT OK	924,3505504
2547420,638	0,003072105	85,88773672	308,3413367	293	70,5464	14,49	16,87	15,71	20,00	0,21862	0,200500347	18888,594	NOT OK	944,4296981
2475159,892	0,003073197	81,11304741	303,5666474	294	71,5464	13,66	15,96	14,83	19,00	0,21242	0,201916399	18352,7956	NOT OK	965,9366089
2401003,37	0,003074385	76,35500962	298,8086096	295	72,5464	12,82	15,05	13,96	18,00	0,20605	0,20332259	17802,9404	NOT OK	989,0522439
2324793,529	0,003075682	71,61500645	294,0686064	296	73,5464	11,99	14,14	13,09	17,00	0,19951	0,204719122	17237,8603	OK	1013,991781
2246349,715	0,003077108	66,89462377	289,3482238	297	74,5464	11,17	13,23	12,22	16,00	0,19278	0,206106191	16656,2157	OK	1041,013482
2165463,114	0,003078683	62,19569446	284,6492945	298	75,5464	10,35	12,33	11,36	15,00	0,18584	0,207483988	16056,4584	OK	1070,43056
2081890,201	0,003080436	57,5203559	279,9739559	299	76,5464	9,53	11,43	10,50	14,00	0,17867	0,208852696	15436,7826	OK	1102,627332
1995344,073	0,003082401	52,87112582	275,3247258	300	77,5464	8,73	10,53	9,64	13,00	0,17124	0,210212492	14795,0611	OK	1138,081624
1905482,793	0,003084626	48,25100445	270,7046044	301	78,5464	7,93	9,64	8,79	12,00	0,16353	0,211563549	14128,7584	OK	1177,396535
1811893,37	0,003087171	43,66361478	266,1172148	302	79,5464	7,14	8,75	7,95	11,00	0,15550	0,212906032	13434,8123	OK	1221,346574
1714069,183	0,003090118	39,11340021	261,5670002	303	80,5464	6,35	7,87	7,12	10,00	0,14710	0,214240103	12709,4663	OK	1270,946633
1611377,251	0,003093585	34,60591079	257,0595108	304	81,5464	5,58	6,99	6,29	9,00	0,13829	0,215565917	11948,027	OK	1327,558558
1503009,132	0,003097739	30,14823254	252,6018325	305	82,5464	4,82	6,12	5,47	8,00	0,12899	0,216883628	11144,5	OK	1393,062496
1387904,161	0,003102835	25,74965784	248,2032578	306	83,5464	4,08	5,26	4,66	7,00	0,11911	0,21819338	10291,0206	OK	1470,145794
1264623,196	0,00310928	21,42278703	243,876387	307	84,5464	3,35	4,40	3,87	6,00	0,10853	0,219495318	9376,91786	OK	1562,819643
1131127,156	0,003117773	17,18545749	239,6390575	308	85,5464	2,65	3,56	3,10	5,00	0,09707	0,220789578	8387,0725	OK	1677,414499
984354,3061	0,003129633	13,06441777	235,5180178	309	86,5464	1,97	2,74	2,34	4,00	0,08448	0,222076295	7298,78236	OK	1824,695589
819313,5942	0,003147742	9,103181283	231,5567813	310	87,5464	1,33	1,94	1,62	3,00	0,07031	0,2233556	6075,03982	OK	2025,013273
626770,274	0,003180019	5,381957066	227,8355571	311	88,5464	0,75	1,18	0,95	2,00	0,05379	0,22462762	4647,37116	OK	2323,685582
385277,4063	0,003261033	2,085431906	224,5390319	312	89,5464	0,26	0,48	0,36	1,00	0,03306	0,225892476	2856,7518	OK	2856,751799
2,24722E-06	0	0	222,4536	313	90,5464	0,00	0,00	0,00	0,00	0,00000	0,227150289	1,6663E-08	OK	0

## Sone

```
/**
 * @author Elin M. Kristensen
 * Denne klassen instansierer et Sone-objekt.
 *
 * @(#)Sone.java 1.0 02.02.05
 */

import java.util.*;
import java.math.*;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.*;

public class Sone implements ActionListener{

    //Referanse til Controllern
    private Controller controller;
    //Referanse til fluid objektet
    private Fluid fluid;

    //Parametre som skal instansieres vha brukergrensesnittet, altså via actionCommand
    //Navnet på sonen.
    private String navn;
    //Høyden på sonen i meter
    protected double hoyde;
    //diameter paa borrehull
    private double bitSize;
    //dreneringsradius
    private double dreneringsradius;
    //Gjennomstrømning av olje i sonen
    protected double vertikalPermabilitet;
    private double horisontalPermabilitet;
    //Trykket i sonen
    private double reservoirTrykk;
    //Vinkel
    private double vinkel;

    //Hullparametre
    private double hullRadius = 0.00508;
    //initverdi antall hull
    private int antHull = 20;

    //Parametere som settes vha metoder i klassen
    private double totalPermabilitet;
    //Lengden på sonen, regnes ut
    private double lengde;
    //Teller over ant registrerte variabler
    private int teller;
```

```

//Liste over seksjoner som er i sonen.
private ArrayList seksjonsListe = new ArrayList();
//Liste over brønntrykk i hver sone
private ArrayList bronnTrykkListe = new ArrayList();
//Liste strRate
private ArrayList strRateListe;
//DP punch liste, dvs trykktap punch
private ArrayList dpPunchListe = new ArrayList();
//bolsk verdi som holder styr på iterasjon
private boolean iterasjonFerdig = false;

/**
 *Konstruktoren oppretter et sone objekt.
 *Tar inn parametre for hvert objekt fra brukeren.
 *En sone har i utgangspunktet 20 hull, men simuleringen vil endre dette tallet
 * og antall seksjoner vil bli gitt ut ifra dette.
 *@param Controller
 */

    public Sone(Controller controller){
        addRespunch();
        this.controller = controller;
        fluid = controller.getFluid();
    }

/**
 *Metoden fyller opp trykklista med ulike brønntrykk i bar
 *@param double max brønntrykk
 */

    public void fyllTrykkListe(double max){
        double trykk = 200*Math.pow(10,5);
        double size = (max-(200*Math.pow(10,5)))/(1*Math.pow(10,5));
        for(int i = 0; i<size; i++){
            Double nyTrykk = new Double(trykk);
            bronnTrykkListe.add(nyTrykk);
            trykk = trykk+(1*Math.pow(10,5));
        }
    }

/*
 *Diverse set-metoder som gjør at brukeren får instansiert variabler
 *Metodene kalles fra actionPerformed metoden.
 */

    public void setNavn(String navn){
        this.navn = navn;
    }

    public void setHoyde(double hoyde){
        this.hoyde = hoyde;
    }

```

```

}

public void setBitSize(double bitSize){
    this.bitSize = (bitSize*0.0254)/2;
}

public void setDreneringsRadius(double radius){
    this.dreneringsradius = radius;
}

public void setVertikalPerm(double vertPerm){
    this.vertikalPermabilitet = vertPerm;
}

public void setHorisontalPerm(double horPerm){
    this.horisontalPermabilitet = horPerm;
}

public void setResTrykk(double resTrykk){
    this.reservoirTrykk = resTrykk * Math.pow(10,5);
    fyllTrykkListe(reservoirTrykk);
}

public void setVinkel(double vinkel){
    this.vinkel = vinkel;
}

public void setAntHull(int nyeHull){
    antHull = antHull + nyeHull;
}

public void setGjAntHull(int gjHull){
    iterasjonFerdig = true;
    antHull = gjHull;
}

/*
*Diverse metoder for å hente ut variabler
*/
public String getNavn(){
    return navn;
}

public int getAntHull(){
    return antHull;
}

public double getReservoarTrykk(){

```

```

        return reservoirTrykk;
    }

    public int getAntSeksjoner(){
        return seksjonsListe.size();
    }

    public double getStrRate(int i){
        double strRate = ((Double)strRateListe.get(i)).doubleValue();
        return strRate;
    }

    public int getTabellLengde(){
        return strRateListe.size();
    }

    public double getDpPunch(int i){
        double dpPunch = ((Double)dpPunchListe.get(i)).doubleValue();
        return dpPunch;
    }

    public double getBronnTrykk(int i){
        double bTrykk = ((Double)bronnTrykkListe.get(i)).doubleValue();
        return bTrykk;
    }

    public double getTotLengdeSek(){
        return seksjonsListe.size()*7.4;
    }

    public double finnGjAntHull(){
        if(!iterasjonFerdig){
            double antSeksjoner = antHull/20;
            double gjsnitt = antHull/antSeksjoner;
            return gjsnitt;
        }
        else
            return antHull;
    }

```

```

//*****Matteformler*****
*****

```

```

/**
 *Metoden legger til en seksjon i lista over seksjoner i sonen.
 */

```

```

public void addRespunch(){
    ResPunchSeksjon seksjon = new ResPunchSeksjon(this);
    seksjonsListe.add(seksjon);
}

/**
 *Metoden legger til x antall seksjoner i lista over seksjoner i sonen.
 *@param double ant nye seksjoner
 */

public void addRespunch(double ant){
    if((seksjonsListe.size()+ant)*7.4 < lengde){
        if(ant == 0){
            addRespunch();
        }
        else{
            for(int i = 0; i < ant; i++){
                ResPunchSeksjon seksjon = new ResPunchSeksjon(this);
                seksjonsListe.add(seksjon);
            }
        }
    }
    else{
        ant = ant-1;
        antHull = antHull-20;
        addRespunch(ant);
    }
}

/**
 *Beregner lengden til sonen vha trigonometri
 */

public void beregnLengde(){
    double delRes = Math.sin((vinkel*Math.PI)/180);
    lengde = hoyde/delRes;
}

/**
 *Beregner totalpermabilitet i kvadratmeter
 */

public void beregnTotPermabilitet(){
    totalPermabilitet = Math.sqrt(Math.pow(vertikalPermabilitet, 2)+
Math.pow(horisontalPermabilitet, 2));
    totalPermabilitet = totalPermabilitet * Math.pow(10,-15);
}

```

```
//*****Simuleringsberegninger*****  
*****
```

```
/**  
 *Ulike metoder som beregner konstanter som brukes senere til å finne riktig veggtrykk  
 */  
  
public double finnA(){  
    beregnLengde();  
    beregnTotPermabilitet();  
    double a =  
(2*Math.PI*totalPermabilitet*lengde)/(fluid.getViskositet()*Math.log(dreneringsradius/bitSize))/seksjonsListe.size();  
    return a;  
}  
  
public double finnB(){  
    ResPunchSeksjon seksjon = (ResPunchSeksjon)seksjonsListe.get(0);  
    double b =  
(2/fluid.getTetthet()*(Math.pow((seksjon.getFi1()*4.933333333),2))*(Math.pow((2*Math.PI*(seksjon.getOD()/2)),2)));  
    return b;  
}  
  
public double finnC(){  
    ResPunchSeksjon seksjon = (ResPunchSeksjon)seksjonsListe.get(0);  
    double c = (2/fluid.getTetthet()*(Math.pow(2*Math.PI*(seksjon.getID()/2),2)));  
    return c;  
}  
  
public double finnD(double a, double b, double c){  
    ResPunchSeksjon seksjon = (ResPunchSeksjon)seksjonsListe.get(0);  
    double fi2 = seksjon.finnPoroesitet(finnGjAntHull(), hullRadius);  
    double d = -(Math.pow(a,2)/b + Math.pow(a,2)/(c*Math.pow(fi2,2)));  
    return d;  
}  
  
public double finnE(double a, double b, double c){  
    ResPunchSeksjon seksjon = (ResPunchSeksjon)seksjonsListe.get(0);  
    double fi2 = seksjon.finnPoroesitet(finnGjAntHull(), hullRadius);  
    double delRes = 2*Math.pow(a,2)*reservoirTrykk;  
    double e = (1+(delRes)/b+(delRes)/(c*Math.pow(fi2,2)));  
    return e;  
}  
  
public ArrayList finnF(double a, double b, double c){  
    ArrayList fListe = new ArrayList();  
    ResPunchSeksjon seksjon = (ResPunchSeksjon)seksjonsListe.get(0);  
    double fi2 = seksjon.finnPoroesitet(finnGjAntHull(), hullRadius);
```



```

        for(int i = 0; i<bronnTrykkListe.size();i++){
            double trykk = ((Double)bronnTrykkListe.get(i)).doubleValue();
            double delRes = Math.pow(a,2)*Math.pow(reservoirTrykk, 2);
            double f = -(trykk+delRes/(c*Math.pow(fi2, 2)))+(delRes)/b);
            Double nyF = new Double(f);
            fListe.add(nyF);
        }
        return fListe;
    }

/**
 *Metoden finner veggtrykket vha de ulike konstantene beregnet ovenfor
 *@param double konstant d
 *@param double konstant e
 *@param ArrayList over f konstanter
 *@return ArrayList liste over veggtrykk
 */

public ArrayList finnVeggTrykk(double d, double e, ArrayList fListe){
    ArrayList veggTrykkListe = new ArrayList();
    for(int i = 0; i< fListe.size(); i++){
        double f = ((Double)fListe.get(i)).doubleValue();
        double trykk = ((-e+Math.sqrt(Math.pow(e,2)-4*d*f))/(2*d));
        Double nyTrykk = new Double(trykk);
        veggTrykkListe.add(nyTrykk);
    }
    return veggTrykkListe;
}

/**
 *Metoden finner trykktapet i respunchene knyttet til sonen.
 *Resultatet blir ivaretatt i en arraylist fordi det varierer med veggtrykket.
 *@param ArrayList Lista over veggtrykk
 *@return ArrayList Lista over trykktap i respunchene.
 */

public ArrayList trykkTapPunch(ArrayList veggTrykkListe){
    ArrayList trykkTapPunchListe = new ArrayList();
    if(veggTrykkListe.size()==bronnTrykkListe.size()){
        for(int i=0; i<veggTrykkListe.size(); i++){
            double bronnTrykk =
((Double)bronnTrykkListe.get(i)).doubleValue();
            double veggTrykk =
((Double)veggTrykkListe.get(i)).doubleValue();
            if(iterasjonFerdig){
                Double trykkTapPunch = new Double(((veggTrykk-
bronnTrykk)/(seksjonsListe.size())));
                trykkTapPunchListe.add(trykkTapPunch);
            }
            else{

```

```

        double antSek = antHull/20;
        Double trykkTapPunch = new Double(((veggTrykk-
bronnTrykk)/(antSek)));
        trykkTapPunchListe.add(trykkTapPunch);
    }
}
else
    return null;
dpPunchListe = trykkTapPunchListe;
return trykkTapPunchListe;
}

/**
 *Metoden finner trykktapet i formasjonen til sonen.
 *Resultatet blir ivaretatt i en arraylist fordi det varierer med veggtrykket.
 *@param ArrayList Lista over veggtrykk
 *@return ArrayList Lista over trykktap i formasjonen.
 */

public ArrayList trykkTapFormasjon(ArrayList veggTrykkListe){
    ArrayList trykkTapFormasjonListe = new ArrayList();
    for(int i = 0; i < veggTrykkListe.size(); i++){
        double veggTrykk = ((Double)veggTrykkListe.get(i)).doubleValue();
        Double trykkTapForm = new Double(reservoirTrykk-veggTrykk);
        trykkTapFormasjonListe.add(trykkTapForm);
    }
    return trykkTapFormasjonListe;
}

/**
 *Metoden finner den totale strømningsraten i en sone
 *@param double, konstanten a
 *@param ArrayList Lista over veggtrykk
 *@return ArrayList Lista over total strømningsrate.
 */

public ArrayList finnStrRate(double a, ArrayList veggTrykkListe){
    strRateListe = new ArrayList();
    for(int i = 0; i < veggTrykkListe.size(); i++){
        double veggTrykk = ((Double)veggTrykkListe.get(i)).doubleValue();
        Double strRate = new Double(a*(reservoirTrykk-veggTrykk));
        strRateListe.add(strRate);
    }
    return strRateListe;
}

/**
 *Metoden finner det maksimale trykktapet i tubingen ut ifra hydrostatisk trykk
 *@param double hydrostatisk trykk

```

```
*/@return ArrayList liste over maks trykktap
*/
```

```
public ArrayList pTubingMax(double hyd){
    ArrayList tubingMaxListe = new ArrayList();
    for(int i=0; i<bronnTrykkListe.size(); i++){
        double bronnTrykk = ((Double)bronnTrykkListe.get(i)).doubleValue();
        Double tubingMax = new Double((bronnTrykk/Math.pow(10,5)) - hyd);
        if(tubingMax.doubleValue() < 0){
            tubingMaxListe.add(new Double(0));
        }
        else
            tubingMaxListe.add(tubingMax);
    }
    return tubingMaxListe;
}
```

```
/**
 *ActionPerformed
 */
```

```
/**
 *ActionPerformed kalles når vi registrerer variabler i grensesnittet. Kaller ulike set-
metoder for
 *å ivareta variablene.
 */
```

```
public void actionPerformed(ActionEvent e){
    //Navn
    if(e.getActionCommand().equals("Navn")){
        JTextField tekstObjekt1 = (JTextField)e.getSource();
        String regNavn = tekstObjekt1.getText();
        if(regNavn != null){
            setNavn(regNavn);
            teller++;
            controller.sendInfo("Du har registrert navn på sonen");
        }
        else{
            controller.sendInfo("Registrer navn på sonen!");
        }
    }
    //Høyde
    if(e.getActionCommand().equals("Hoyde")){
        JTextField tekstObjekt2 = (JTextField)e.getSource();
        try{
            double regHoyde = Double.parseDouble(tekstObjekt2.getText());
            setHoyde(regHoyde);
            teller++;
            controller.sendInfo("Du har registrert sonens høyde");
        }
    }
}
```

```

    }
    catch(Exception nulle){
        controller.sendInfo("Du har registrert høyden på feil format");
    }
}
//BitSize
if(e.getActionCommand().equals("BitSize")){
    JTextField tekstObjekt3 = (JTextField)e.getSource();
    try{
        double regBitSize =
Double.parseDouble(tekstObjekt3.getText());
        setBitSize(regBitSize);
        teller++;
        controller.sendInfo("Du har registrert bitsize");
    }
    catch(Exception nulle){
        controller.sendInfo("Du har registrert bitsize på feil format");
    }
}
//Drenering
if(e.getActionCommand().equals("Drenering")){
    JTextField tekstObjekt4 = (JTextField)e.getSource();
    try{
        double regDrenering =
Double.parseDouble(tekstObjekt4.getText());
        setDreneringsRadius(regDrenering);
        teller++;
        controller.sendInfo("Du har registrert drenering");
    }
    catch(Exception nulle){
        controller.sendInfo("Du har registrert dreneringsradius på feil
format");
    }
}
//Vertikal permabilitet
if(e.getActionCommand().equals("VertPerm")){
    JTextField tekstObjekt5 = (JTextField)e.getSource();
    try{
        double regVertPerm =
Double.parseDouble(tekstObjekt5.getText());
        setVertikalPerm(regVertPerm);
        teller++;
        controller.sendInfo("Du har registrert vertikal permabilitet");
    }
    catch(Exception nulle){
        controller.sendInfo("Du har registrert vertikal permabilitet på
feil format");
    }
}
//Horisontal permabilitet

```

```

        if(e.getActionCommand().equals("HorPerm")){
            JTextField tekstObjekt6 = (JTextField)e.getSource();
            try{
                double regHorPerm =
Double.parseDouble(tekstObjekt6.getText());
                setHorisontalPerm(regHorPerm);
                teller++;
                controller.sendInfo("Du har registrert horisontal permabilitet");
            }
            catch(Exception nullE){
                controller.sendInfo("Du har registrert horisontal permabilitet på
feil format");
            }
        }

//ReservoirTrykk
if(e.getActionCommand().equals("ReservoirTrykk")){
    JTextField tekstObjekt8 = (JTextField)e.getSource();
    try{
        double regResTrykk =
Double.parseDouble(tekstObjekt8.getText());
        setResTrykk(regResTrykk);
        teller++;
        controller.sendInfo("Du har registrert reservoartrykk");
    }
    catch(Exception nullE){
        controller.sendInfo("Du har registrert reservoartrykket på feil
format");
    }
}

//Vinkel
if(e.getActionCommand().equals("Vinkel")){
    JTextField tekstObjekt10 = (JTextField)e.getSource();
    try{
        double regVinkel =
Double.parseDouble(tekstObjekt10.getText());
        setVinkel(regVinkel);
        teller++;
        controller.sendInfo("Du har registrert vinkel");
    }
    catch(Exception nullE){
        controller.sendInfo("Du har registrert vinkelen på feil format");
    }
}

//knappen
if(e.getActionCommand().equals("regSoneKnapp")){
    System.out.println("Er inne i actioncommand");
    if(teller < 8){

```

```
        controller.sendInfo("Du må registrere alle sone variablene");
    }
    else{
        controller.sendInfo("Lagrer sonevariabler for " +getNavn());
        teller = 0;
        controller.regAvSoner();
    }
}
}
} //Sone
```

## ResPunchSeksjon

```
/**
 * @author Elin M. Kristensen
 * Denne klassen instansierer ResPunchSeksjon-objekt.
 *
 * @(#)ResPunchSeksjon.java 1.0 02.02.05
 */

//Obs obs, sjekk lengde og screenåpning

public class ResPunchSeksjon{

    //lengden på seksjonen
    protected double lengde = 7.4;
    //chokelengde gitt i meter
    private double chokeLengde = 1.5;
    //inner radius
    private double innerDiameter = 0.1778;
    //outerRadius
    private double outerDiameter = 0.196;
    //fi1
    private double fi1 = 0.089321652;
    //Referanse til sonen hvor seksjonen er plassert
    //Antar at en Respunch ikke strekker seg over flere soner
    private Sone sone;

    /**
     *Konstruktoren oppretter et ResPunchObjekt
     */
    public ResPunchSeksjon(Sone sone){
        this.sone = sone;
    }
    /**
     *Metoden returnerer innerDiameter til seksjonen
     *@return double
     */

    public double getID(){
        return innerDiameter;
    }

    /**
     *Metoden returnerer outerDiameter til seksjonen
     *@return double
     */

    public double getOD(){
        return outerDiameter;
    }
}
```

```

    }

    /**
     *Metoden returnerer konstanten fi1
     *@return double fi1
     */

    public double getFi1(){
        return fi1;
    }

    /**
     *Metoden finner poroesiteten i en perforeringsseksjon.
     *Brukes videre til å finne trykktap.
     *@param int antHull antall hull i seksjonen
     *@param double hullRadius. Radius punch-hull
     *@param double outerRadius. Radius på casing
     *@return double poroesitet
     */

    public double finnPoroesitet(double antHull, double hullRadius){
        double poroesitet = (Math.PI*Math.pow(hullRadius,
2)*antHull)/(Math.PI*innerDiameter*chokeLengde);
        return poroesitet;
    }

} //ResPunchSeksjon

```



## Fluid

```
/**
 * @author Elin M. Kristensen
 * Denne klassen instansierer et Fluid-objekt.
 *
 * @(#)Fluid.java
 */

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.*.*;

public class Fluid implements ActionListener{

    //Seighet på olje
    private double viskositet;
    //tetthet på oljen
    private double tetthet;
    //Referanse til Kontrolleren
    private Controller controller;

    /**
     *Konstruktoren oppretter et Fluid-objekt.
     */

    public Fluid(Controller controller){
        this.controller = controller;
    }

    /**
     *Metoden gir tetthet en verdi
     * @param double ny tetthetsverdi
     */

    public void setTetthet(double tetthet){
        this.tetthet = tetthet;
    }

    /**
     *Metoden gir viskositet en verdi
     * @param double ny verdi viskositet
     */

    public void setViskositet(double viskositet){
        this.viskositet = viskositet/1000;
    }
}
```

```

/**
 *Metoden returnerer viskositeten til fluidobjektet, dvs oljeseigheten.
 *@return double viskositet
 */

public double getViskositet(){
    return viskositet;
}

/**
 *Metoden returnerer tettheten til fluidobjektet
 *@return double tetthet
 */

public double getTetthet(){
    return tetthet;
}

/**
 *Metoden mottar en hendelse naar man registrer fluidvariabler i GUI'en
 *Kaller metoder som instansierer variablene.
 */

public void actionPerformed(ActionEvent e){
    if(e.getActionCommand().equals("Viskositet")){
        JTextField tekstObjekt1 = (JTextField)e.getSource();
        try{
            double viskositet = Double.parseDouble(tekstObjekt1.getText());
            setViskositet(viskositet);
            controller.sendInfo("Du har registrert viskositeten");
        }
        catch(Exception nullE){
            controller.sendInfo("Du har registrert viskositeten på feil
format!");
        }
    }

    if(e.getActionCommand().equals("Tetthet")){
        JTextField tekstObjekt2 = (JTextField)e.getSource();
        try{
            double tetthet = Double.parseDouble(tekstObjekt2.getText());
            setTetthet(tetthet);
            controller.sendInfo("Du har registrert tettheten");
        }
        catch(Exception nullE){
            controller.sendInfo("Du har registrert tetthet på feil format!");
        }
    }
}

```

```
if(e.getActionCommand().equals("fluidKnapp")){
    if(viskositet != 0 && tetthet != 0){
        controller.sendInfo("Fluid-variablene er nå registrert");
    }
    else{
        controller.sendInfo("Du må registrere fluidvariabler!");
    }
}
}
} //Fluid
```

## Controller

```
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.*;
import java.util.*;
import java.io.*;

public class Controller implements ActionListener{

    //Referanse til brukergrensesnittet
    private GUI gui;
    //Referanse til Fluid objektet
    private Fluid fluid;
    //Liste over sonene
    private ArrayList soneListe;
    //soneteller
    private int soneTeller;
    //soneNr
    private int soneNr;
    //simulering ferdig
    private boolean simFerdig = false;
    //private boolean forste beregning tap
    private boolean initialTap = false;
    //maksimal strømningsrate
    double riktigRate = 0;
    //minimal strømningsrate
    double minRate = 0;

    /**
     *Oppretter ett nytt Controller objekt.
     *Objektet kontrollerer andre objekter
     *Disse blir instansiert i konstruktoren.
     */

    public Controller(){
        fluid = new Fluid(this);
        gui = new GUI(this);
        soneListe = new ArrayList();
    }

    /**
     *Metoden returnerer et fluid objekt
     *@return Fluid
     */

    public Fluid getFluid(){
        return fluid;
    }
}
```

```

/**
 *Metoden returnerer maks strRate for reservoaret
 *@return double maks rate
 */

public double getRiktigRate(){
    return riktigRate;
}

/**
 *Metoden returnerer min strRate for reservoaret
 *@return double min rate
 */

public double getMinRate(){
    return minRate;
}

/**
 *Metoden oppdaterer informasjonsvinduet i guien
 *@param String melding
 */
public void sendInfo(String infoMld){
    gui.oppdaterInfoVindu(infoMld);
}

/**
 *Metoden sjekker om alle soner er registrert, dersom dette ikke er tilfeller registreres
en ny sone
 */
public void regAvSoner(){
    if(soneTeller > 0){
        opprettSone();

    }
    else{
        sendInfo("Alle soner er opprettet");
        gui.startSim();
    }
}

/**
 *Metoden oppretter et nytt soneobjekt
 */

public void opprettSone(){
    Sone sone = new Sone(this);
    soneNr++;
    soneTeller--;
}

```

```

        soneListe.add(sone);
        gui.registrerSone(sone, "kort"+soneNr);
    }

    /**
    *Metoden setter igang simuleringa. For hver sone beregnes trykktapet. Metoden kalles
    rekursivt.
    *@param double. Dette trykktapet er utgangspunkt for beregninger.
    *@param ArrayList liste over tap som skal bli mest mulig lik utgangspunktet.
    */

    public void startSimulering(double maal, ArrayList tapsListe){
        ArrayList nyTapsListe = tapsListe;
        int teller = 0;
        if(simFerdig){
            videreBeregn();
        }
        else{
            for(int i = 0; i < nyTapsListe.size(); i++){
                Sone sone = (Sone)soneListe.get(i);
                double tap = ((Double)nyTapsListe.get(i)).doubleValue();
                if(tap <= maal){
                    teller ++;
                }
                else{
                    sone.setAntHull(1);
                    Double nyttTap = new Double(beregn(sone));
                    nyTapsListe.set(i, nyttTap);
                }
            }
            if(teller == tapsListe.size()){
                simFerdig =true;
            }
            startSimulering(maal, nyTapsListe);
        }
    }

    /**
    *Metoden gjør de siste beregningene etter at iterasjonen er ferdig
    */

    public void videreBeregn(){
        finnAntSeksjoner();
        antHullprSeksjon();
        ArrayList totStrRateListe = totalStrRateBeregn();
        ArrayList tubingListe = tubing(totStrRateListe);
        ArrayList tubingMaxListe = tubingMax();
        ArrayList rateMaxListe = rateMax(tubingMaxListe);
        int indexRiktigRate = sammenlignRater(totStrRateListe, rateMaxListe);
    }

```

```

        sendInfo("Simulering er ferdig");
        gui.visResultat(soneListe, indexRiktigRate);
    }

/**
 *Metoden finner utgangspunkt for ønsket trykktap i alle sonene
 *@param ArrayList de ulike trykktapene for alle sonene
 *@return double trykktapet det skal taes utgangspunkt i
 */

public double finnUtgangspkt(ArrayList fellesTapsListe){
    double utgangspkt = Math.pow(10,15);
    for(int i=0; i < fellesTapsListe.size();i++){
        double tap = ((Double)fellesTapsListe.get(i)).doubleValue();
        if(tap < utgangspkt){
            utgangspkt = tap;
        }
    }
    initialTap = true;
    return utgangspkt;
}

/**
 *Metoden gjør de nødvendige beregningene for å finne trykktap i punch
 *@param Sone, sonen det gjøres beregninger for
 *@return double trykktap, det nye trykktapet for sonen
 */

public double beregn(Sone sone){

    double a = sone.finnA();
    double b = sone.finnB();
    double c = sone.finnC();
    double d = sone.finnD(a, b, c);
    double e = sone.finnE(a,b,c);
    ArrayList f = sone.finnF(a,b,c);
    ArrayList veggTrykkListe = sone.finnVeggTrykk(d, e , f);
    ArrayList trykkTapFormListe = sone.trykkTapFormasjon(veggTrykkListe);
    ArrayList strRateSone = sone.finnStrRate(a, veggTrykkListe);
    ArrayList trykkTapPunchListe = sone.trykkTapPunch(veggTrykkListe);
    double trykkTap = ((Double)trykkTapPunchListe.get(0)).doubleValue();
    return trykkTap;
}

/**
 *Metoden fyller tapslista ved å ta vare på det første tallet i trykkTapPunchlista
 *@return ArrayList liste over hver enkeltsones tap.
 */

```

```

public ArrayList fyllTapsListe(){
    ArrayList tapsListe = new ArrayList();
    for(int i = 0; i < soneListe.size(); i++){
        Sone sone = (Sone)soneListe.get(i);
        Double tapPunch = new Double(beregn(sone));
        tapsListe.add(tapPunch);
    }
    return tapsListe;
}

/**
 *Metoden finner antall seksjoner for hver sone ved å se på antall hull
 */

public void finnAntSeksjoner(){
    for(int i = 0; i < soneListe.size(); i++){
        Sone sone = (Sone)soneListe.get(i);
        int antHull = sone.getAntHull();
        double antSeksjoner = (antHull-20)/20.0;
        if(antSeksjoner > 0){
            sone.addRespunch(antSeksjoner);
        }
    }
}

/**
 *Metoden finner gjennomsnittlig antall hull pr seksjon
 */

public void antHullprSeksjon(){
    for(int i = 0; i < soneListe.size(); i++){
        Sone sone = (Sone)soneListe.get(i);
        int antHull = sone.getAntHull()/sone.getAntSeksjoner();
        sone.setGjAntHull(antHull);
        beregn(sone);
    }
}

/**
 *Metoden beregner den totale strømningsraten for ett gitt brønntrykk
 *@return ArrayList en liste over den totale strømningsraten i reservoaret for
 * et gitt brønntrykk
 */

public ArrayList totalStrRateBeregn(){
    ArrayList totStrRate = new ArrayList();
    Sone sone = (Sone)soneListe.get(0);
    int teller = sone.getTabellLengde();
    for(int i = 0; i < teller; i++){

```



```

        double totStrRateI = 0;
        for(int j = 0; j< soneListe.size(); j++){
            Sone currSone = (Sone)soneListe.get(j);
            double strRateSone = currSone.getStrRate(i);
            totStrRateI = totStrRateI + strRateSone;
        }
        Double totRate = new Double(totStrRateI);
        totStrRate.add(totRate);
        totStrRateI = 0;
    }
    return totStrRate;
}

/**
 *Metoden finner forventet trykktap i tubing ved å bruke de totale strømningsratene
 *@param ArrayList listen over totale strømningsrater.
 *@return ArrayList listen over forventet trykktap i tubing
 */

public ArrayList tubing(ArrayList totStrRateListe){
    ArrayList tubingListe = new ArrayList();
    for(int i = 0; i< totStrRateListe.size(); i++){
        double strRate = ((Double)totStrRateListe.get(i)).doubleValue();
        double tubing =
((0.003*10000/0.1570736)*Math.pow((strRate/((Math.PI/4)*Math.pow(0.1570736,2))),2)*(fluid.getTetthet()/2))/Math.pow(10, 5);
        tubingListe.add(new Double(tubing));
    }
    return tubingListe;
}

/**
 *Metoden finner det maksimale trykktapet i tubingen
 *Benytter seg av hydrostatisk trykk
 *@return ArrayList liste over maks trykktap
 */

public ArrayList tubingMax(){
    double hyd = (fluid.getTetthet()*9.81*2400)/(Math.pow(10,5))+60;
    Sone sone = (Sone)soneListe.get(0);
    ArrayList tubingMaxListe = sone.pTubingMax(hyd);
    return tubingMaxListe;
}

/**
 *Metoden finner den maksimale strømningsraten ut fra maks trykktap i tubing.
 *@param ArrayList liste over maks tap i tubing
 *@return ArrayList liste over de maksimale strømningsratene
 */

```

```

public ArrayList rateMax(ArrayList tubingMaxListe){
    ArrayList rateMaxListe = new ArrayList();
    for(int i = 0; i < tubingMaxListe.size(); i++){
        double tubing = ((Double)tubingMaxListe.get(i)).doubleValue();
        double strRateMax =
Math.sqrt(((Math.pow(Math.PI,2))/8)*((tubing*Math.pow(10,5)*Math.pow(0.1570736,5))/(1
0000*0.003*fluid.getTetthet())));
        rateMaxListe.add(new Double(strRateMax));
    }
    return rateMaxListe;
}

```

```

/**
 *Metoden sammenligner de totale strømningsratene mot maksimal ratene
 *Dersom maks rate overstiger totalt strømningsrate har vi funnet ønsket maks
 *strømningsrate.
 *@param ArrayList de totale strømningsratene
 *@param ArrayList de maksimale strømningsratene
 *@return int tilhørende indeks til ønsket rate
 */

```

```

public int sammenlignRater(ArrayList totStrRateListe, ArrayList rateMaxListe){
    int index = 0;
    if(totStrRateListe.size()==rateMaxListe.size()){
        for(int i = 0; i < rateMaxListe.size(); i++){
            if(((Double)rateMaxListe.get(i)).doubleValue() >
((Double)totStrRateListe.get(i)).doubleValue()){
                index = i;
                riktigRate =
((Double)totStrRateListe.get(i)).doubleValue();
                minRate =
((Double)totStrRateListe.get(112)).doubleValue();
                break;
            }
        }
    }
    return index;
}

```

```

/**
 *Metoden finner produksjonsindeks
 *@param double strRate
 *@param double brønntrykk
 *@return double pi produksjonsindeks
 */

```

```

public double finnProdI(double strRate, double bTrykk){
    double prodI =
strRate/((((Sone)soneListe.get(0)).getReservoarTrykk()/Math.pow(10,5))-bTrykk);
}

```

```

        return prodI;
    }

    /**
     *Metoden fanger opp hendelsene registrering av ant soner og start simulering.
     *Ut fra dette antallet opprettes soner.
     */

    public void actionPerformed(ActionEvent e){
        if(e.getActionCommand().equals("AntSoner")){
            JTextField tekstObjekt1 = (JTextField)e.getSource();
            try{
                int antSoner = Integer.parseInt(tekstObjekt1.getText());
                soneTeller = antSoner;
            }
            catch(Exception nullE){
                sendInfo("Du må registrere antall soner");
            }
        }
        if(e.getActionCommand().equals("regAntSoneKnapp")){
            if(soneTeller != 0){
                sendInfo("Du har registrert antall soner");
                opprettSone();
            }
            else
                sendInfo("Registrer antall soner!");
        }
        if(e.getActionCommand().equals("simKnapp")){
            sendInfo("Du har startet simuleringen");
            ArrayList tapsListe = fyllTapsListe();
            double utgPkt = finnUtgangspkt(tapsListe);
            startSimulering(utgPkt, tapsListe);
        }
    }

    /**
     *Main metoden, instansierer et Controller objekt
     */

    public static void main(String [] args){
        Controller controller = new Controller();
    }
} //Controller

```

## Fluid klassen

```

/**
 * @author Elin M. Kristensen
 * Denne klassen instansierer et Fluid-objekt.

```

```

*
* @(#)Fluid.java
*/

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.*;

public class Fluid implements ActionListener{

    //Seighet på olje
    private double viskositet;
    //tetthet på oljen
    private double tetthet;
    //Referanse til Kontrolleren
    private Controller controller;

    /**
     *Konstruktoren oppretter et Fluid-objekt.
     */

    public Fluid(Controller controller){
        this.controller = controller;
    }

    /**
     *Metoden gir tetthet en verdi
     *@param double ny tetthetsverdi
     */

    public void setTetthet(double tetthet){
        this.tetthet = tetthet;
    }

    /**
     *Metoden gir viskositet en verdi
     *@param double ny verdi viskositet
     */

    public void setViskositet(double viskositet){
        this.viskositet = viskositet/1000;
    }

    /**
     *Metoden returnerer viskositeten til fluidobjektet, dvs oljeseigheten.
     *@return double viskositet
     */

```

```

    public double getViskositet(){
        return viskositet;
    }

/**
 *Metoden returnerer tettheten til fluidobjektet
 *@return double tetthet
 */

public double getTetthet(){
    return tetthet;
}

/**
 *Metoden mottar en hendelse naar man registrer fluidvariabler i GUI'en
 *Kaller metoder som instansierer variablene.
 */

public void actionPerformed(ActionEvent e){
    if(e.getActionCommand().equals("Viskositet")){
        JTextField tekstObjekt1 = (JTextField)e.getSource();
        try{
            double viskositet = Double.parseDouble(tekstObjekt1.getText());
            setViskositet(viskositet);
            controller.sendInfo("Du har registrert viskositeten");
        }
        catch(Exception nullE){
            controller.sendInfo("Du har registrert viskositeten på feil
format!");
        }
    }

    if(e.getActionCommand().equals("Tetthet")){
        JTextField tekstObjekt2 = (JTextField)e.getSource();
        try{
            double tetthet = Double.parseDouble(tekstObjekt2.getText());
            setTetthet(tetthet);
            controller.sendInfo("Du har registrert tettheten");
        }
        catch(Exception nullE){
            controller.sendInfo("Du har registrert tetthet på feil format!");
        }
    }

    if(e.getActionCommand().equals("fluidKnapp")){
        if(viskositet != 0 && tetthet != 0){
            controller.sendInfo("Fluid-variablene er nå registrert");
        }
    }
}

```

```
        else{
            controller.sendInfo("Du må registrere fluidvariabler!");
        }
    }
}
} //Fluid
```

## Brukergrensesnitt

```
/**
 * @author Elin M. Kristensen
 * Denne klassen instansierer et GUI-objekt.
 *
 * @(#)GUI.java 1.0 02.02.05
 */

import javax.swing.*;
import java.awt.*;
import javax.swing.border.TitledBorder;
import java.net.URL;
import javax.swing.ImageIcon;
import java.awt.CardLayout;
import java.util.*;
import java.awt.Color;

public class GUI{

    //Referanse til Controller objektet
    private Controller controller;
    //Hovedrammen til brukergrensesnittet
    private JFrame hovedRamme;
    //Informasjonsområdet
    private JTextArea infoTekst;
    //Registreringsbakgrunnen
    private JPanel regBakgrunn;
        //Bildebakgrunn m logo
        private JPanel bildeBakgrunn;
        //Registreringskort brukt av regBakgrunn
        private JPanel regKort;
        //Gridbaglayout brukt av flere panels
        GridBagLayout g = new GridBagLayout();
        GridBagConstraints c = new GridBagConstraints();
        //knapper som ikke kan brukes etter at variabler er lagret
        private JButton simKnapp;
        private JButton fluidKnapp;

    /*
    *Konstruktoren oppretter et GUI objekt og kaller byggGUI metoden
    */

    public GUI(Controller controller){
        this.controller = controller;
        byggGui();
    }
}
```

```

/*
*Metoden bygger grensesnittet
*/

public void byggGui(){

    //Bygger rammen rundt
    JFrame.setDefaultLookAndFeelDecorated(true);
    hovedRamme = new JFrame("PunchSimulator Gullfakslandsbyen Gruppe 5");
    hovedRamme.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    hovedRamme.getContentPane().setLayout(new
BoxLayout(hovedRamme.getContentPane(), BoxLayout.X_AXIS));
    hovedRamme.setExtendedState(hovedRamme.MAXIMIZED_BOTH);

    //de to hovedkolonnene
    regBakgrunn = new JPanel();
    regBakgrunn.setLayout(new CardLayout());
    bildeBakgrunn = new JPanel();
    bildeBakgrunn.setLayout(new BorderLayout());

    //HovedBorders
    TitledBorder regBorder = new TitledBorder("REGISTRERING AV
VARIABLER");
    regBorder.setTitleFont(new Font("", Font.ITALIC, 20));
    TitledBorder infoBorder = new TitledBorder("INFORMASJON");
    infoBorder.setTitleFont(new Font("", Font.ITALIC, 20));

    //Setter tittel på registreringsstacken
    regBakgrunn.setBorder(regBorder);

    //Registreringsstacken skal inneholde tre kort: Registrering, simulering og
resultat.

    //Legger først til Registreringskortet.

    regKort = new JPanel();
    regKort.setLayout(new CardLayout());

    //Regkortet er også en stack som skal inneholde to kort: Felles variabler og
soner

    //Legger først til fellesVariabler

    JPanel fellesBakgrunn = new JPanel();
    fellesBakgrunn.setLayout(g);
    c.gridwidth = GridBagConstraints.REMAINDER;
    regKort.add(fellesBakgrunn, "fellesVariabler");

    //Oppretter bildet og legger det på bildebakgrunnen.
    try {
        URL imageURL = GUI.class.getResource("draape.jpg");

```



```

        ImageIcon logo = new ImageIcon(imageURL);
        JLabel bildeLabel = new JLabel(logo);
        bildeBakgrunn.add(bildeLabel, BorderLayout.NORTH);
    }
    catch(Exception e){
        System.out.println("Nå finner vi ikke logoen");
    }

    JPanel infoBakgrunn = new JPanel();
    infoBakgrunn.setBorder(infoBorder);
    bildeBakgrunn.add(infoBakgrunn, BorderLayout.CENTER);

    //Informasjonsvinduet
    infoTekst = new JTextArea("Registrer de nødvendig variablene.");
    infoTekst.setEditable(false);
    infoTekst.setOpaque(false);
    infoTekst.setColumns(30);
    infoTekst.setFont(new Font("", Font.PLAIN, 18));
    infoBakgrunn.add(infoTekst);

    //Registrering av Felles variable

    //FLUID
    JPanel fluidPanel = new JPanel();
    fluidPanel.setLayout(new GridLayout(3,2));

    TitledBorder fluidBorder = new TitledBorder("FLUIDVARIABLER");
    fluidBorder.setTitleFont(new Font("", Font.ITALIC, 15));
    fluidPanel.setBorder(fluidBorder);

    JLabel viskositetLabel = new JLabel("Viskositet [cP] ");
    viskositetLabel.setFont(new Font("", Font.PLAIN, 16));
    JTextField regViskositet = new JTextField(20);
    regViskositet.addActionListener(controller.getFluid());
    regViskositet.setActionCommand("Viskositet");

    fluidPanel.add(viskositetLabel);
    fluidPanel.add(regViskositet);

    JLabel tetthetLabel = new JLabel("Oljetetthet [kg/m3] ");
    tetthetLabel.setFont(new Font("", Font.PLAIN, 16));
    JTextField regTetthet = new JTextField(20);
    regTetthet.addActionListener(controller.getFluid());
    regTetthet.setActionCommand("Tetthet");
    g.setConstraints(regTetthet, c);
    fluidPanel.add(tetthetLabel);
    fluidPanel.add(regTetthet);

```

```

JLabel tom2 = new JLabel("");
fluidPanel.add(tom2);

fluidKnapp = new JButton("Registrer fluid");
fluidKnapp.setPreferredSize(regTetthet.getPreferredSize());
fluidKnapp.addActionListener(controller.getFluid());
fluidKnapp.setActionCommand("fluidKnapp");
fluidPanel.add(fluidKnapp);

//reg ant soner
JPanel antSonePanel = new JPanel();
antSonePanel.setLayout(new GridLayout(2,2));

TitledBorder antSonerBorder = new TitledBorder("ANTALL SONER");
antSonerBorder.setTitleFont(new Font("", Font.ITALIC, 15));
antSonePanel.setBorder(antSonerBorder);

JLabel regSoner = new JLabel("Antall soner i reservoaret: ");
regSoner.setFont(new Font("", Font.PLAIN, 16));
antSonePanel.add(regSoner);

JTextField antSonerFelt = new JTextField(20);
antSonerFelt.addActionListener(controller);
antSonerFelt.setActionCommand("AntSoner");
antSonePanel.add(antSonerFelt);

//Knappen
JLabel tom = new JLabel("");
antSonePanel.add(tom);

JButton regAntSoneKnapp = new JButton("Registrer antall soner");
regAntSoneKnapp.setPreferredSize(antSonerFelt.getPreferredSize());
regAntSoneKnapp.addActionListener(controller);
regAntSoneKnapp.setActionCommand("regAntSoneKnapp");
antSonePanel.add(regAntSoneKnapp);

fellesBakgrunn.add(fluidPanel);
g.setConstraints(fluidPanel, c);
fellesBakgrunn.add(antSonePanel);

regBakgrunn.add(regKort, "RegKort");

//Legger til de to bakgrunnene.
hovedRamme.getContentPane().add(regBakgrunn);
hovedRamme.getContentPane().add(bildeBakgrunn);

hovedRamme.setVisible(true);

```

```

}

/**
 *Metoden setter opp registreringsvindu for hver sone.
 *Bruker CardLayout slik at registreringsvindue ligger på hverandre.
 *@param Sone sonen som skal registreres
 *@param String navn på sonen
 */

public void registrerSone(Sone sone, String kortNavn){

    JPanel bakgrunn = new JPanel();
    bakgrunn.setLayout(g);
    regKort.add(bakgrunn, "soner");

    JPanel soneKort1 = new JPanel();
    soneKort1.setLayout(new GridLayout(17,2));

    //Navn på sonen
    JLabel navnLabel = new JLabel("Navn på sone ");
    navnLabel.setFont(new Font("", Font.PLAIN, 16));
    JTextField regNavn = new JTextField(20);
    regNavn.addActionListener(sone);
    regNavn.setActionCommand("Navn");
    soneKort1.add(navnLabel);
    soneKort1.add(regNavn);

    //Høyde på sonen
    JLabel hoydeLabel = new JLabel("Høyde på sonen [m]");
    hoydeLabel.setFont(new Font("", Font.PLAIN, 16));
    JTextField regHoyde = new JTextField(20);
    regHoyde.addActionListener(sone);
    regHoyde.setActionCommand("Hoyde");
    soneKort1.add(hoydeLabel);
    soneKort1.add(regHoyde);

    //Borrehull, også kalt bitsize
    JLabel bitSizeLabel = new JLabel("Bitsize [in]");
    bitSizeLabel.setFont(new Font("", Font.PLAIN, 16));
    JTextField regBitSize = new JTextField(20);
    regBitSize.addActionListener(sone);
    regBitSize.setActionCommand("BitSize");
    soneKort1.add(bitSizeLabel);
    soneKort1.add(regBitSize);

    //Dreneringsradius
    JLabel dreneringsLabel = new JLabel("Dreneringsradius [m]");
    dreneringsLabel.setFont(new Font("", Font.PLAIN, 16));
    JTextField regDrenering = new JTextField(20);

```

```
regDrenering.addActionListener(sone);
regDrenering.setActionCommand("Drenering");
soneKort1.add(dreneringsLabel);
soneKort1.add(regDrenering);
```

```
//VertikalPermabilitet
JLabel vertPermLabel = new JLabel("Vertikal Permabilitet [mD]");
vertPermLabel.setFont(new Font("", Font.PLAIN, 16));
JTextField regVertPerm = new JTextField(20);
regVertPerm.addActionListener(sone);
regVertPerm.setActionCommand("VertPerm");
soneKort1.add(vertPermLabel);
soneKort1.add(regVertPerm);
```

```
//Horisontal Permabilitet
JLabel horPermLabel = new JLabel("Horisontal Permabilitet [mD]");
horPermLabel.setFont(new Font("", Font.PLAIN, 16));
JTextField regHorPerm = new JTextField(20);
regHorPerm.addActionListener(sone);
regHorPerm.setActionCommand("HorPerm");
soneKort1.add(horPermLabel);
soneKort1.add(regHorPerm);
```

```
//Reservoirtrykk
JLabel reservoirTrykkLabel = new JLabel("Reservoartrykk [bar]");
reservoirTrykkLabel.setFont(new Font("", Font.PLAIN, 16));
JTextField regReservoirTrykk = new JTextField(20);
regReservoirTrykk.addActionListener(sone);
regReservoirTrykk.setActionCommand("ReservoirTrykk");
soneKort1.add(reservoirTrykkLabel);
soneKort1.add(regReservoirTrykk);
```

```
//Vinkel
JLabel vinkelLabel = new JLabel("Vinkel [deg]");
vinkelLabel.setFont(new Font("", Font.PLAIN, 16));
JTextField regVinkel = new JTextField(20);
regVinkel.addActionListener(sone);
regVinkel.setActionCommand("Vinkel");
soneKort1.add(vinkelLabel);
soneKort1.add(regVinkel);
```

```
//Registreringsknapp
JLabel tom3 = new JLabel("");
```

```
soneKort1.add(tom3);
JButton regSoneKnapp = new JButton("Registrer sone");
regSoneKnapp.addActionListener(sone);
regSoneKnapp.setActionCommand("regSoneKnapp");
soneKort1.add(regSoneKnapp);
```

```

        bakgrunn.add(soneKort1);

        CardLayout cl = (CardLayout)(regKort.getLayout());
cl.show(regKort, "soner");
System.out.println("kortNavn "+kortNavn);

    }

    /**
    *Metoden lager vinduet som viser at simuleringa igangsettes
    */

    public void startSim(){
        TitledBorder simBorder = new TitledBorder("SIMULERING");
        simBorder.setTitleFont(new Font("", Font.ITALIC, 20));
        regBakgrunn.setBorder(simBorder);

        JPanel simulering = new JPanel();
        simulering.setLayout(g);
        regKort.add(simulering, "simulering");

        JPanel simKort = new JPanel();
        simKort.setLayout(new GridLayout(2,1));

        JLabel info = new JLabel("Alle soner er nå registrert" );
        info.setFont(new Font("", Font.PLAIN, 16));
        simKort.add(info);

        simKnapp = new JButton("Start simulering");
        simKnapp.addActionListener(controller);
        simKnapp.setActionCommand("simKnapp");

        simKort.add(simKnapp);

        simulering.add(simKort);

        regBakgrunn.add(simulering, "sim");
        CardLayout cl = (CardLayout)(regBakgrunn.getLayout());
cl.show(regBakgrunn, "sim");

    }

    /**
    *Metode som viser resultat av simuleringa
    *@param ArrayList liste over alle sonene
    *@param index. Indeks som refererer til ønsket brønntrykk
    */
    public void visResultat(ArrayList soneListe, int index){

```

```

        TitledBorder resBorder = new TitledBorder("RESULTAT AV
SIMULERING");
        resBorder.setTitleFont(new Font("", Font.ITALIC, 20));

        regBakgrunn.setBorder(resBorder);

        JPanel resKort = new JPanel();
        resKort.setLayout(new GridLayout(soneListe.size()+1, 1));

        for(int i = 0; i < soneListe.size(); i++){
            JPanel sone = new JPanel();
            sone.setLayout(new GridLayout(5,2));
            TitledBorder border = new
TitledBorder(((Sone)(soneListe.get(i))).getNavn());
            border.setTitleFont(new Font("", Font.ITALIC, 18));
            sone.setBorder(border);

            //antall ResPunch Seksjoner
            JLabel punchLabel = new JLabel("Antall ResPunch-seksjoner: ");
            punchLabel.setFont(new Font("", Font.PLAIN, 16));
            sone.add(punchLabel);

            Integer ant = new Integer(((Sone)(soneListe.get(i))).getAntSeksjoner());
            JLabel antSeksjoner = new JLabel(ant.toString());
            antSeksjoner.setFont(new Font("", Font.PLAIN, 16));
            sone.add(antSeksjoner);

            //lengde ResPunch Seksjoner
            JLabel lengdePunchLabel = new JLabel("Total lengde ResPunch-
seksjoner[m]: ");
            lengdePunchLabel.setFont(new Font("", Font.PLAIN, 16));
            sone.add(lengdePunchLabel);

            Double lengde = new
Double(((Sone)(soneListe.get(i))).getTotLengdeSek());
            JLabel lengdeSeksjoner = new JLabel(lengde.toString());
            lengdeSeksjoner.setFont(new Font("", Font.PLAIN, 16));
            sone.add(lengdeSeksjoner);

            //antall hull pr seksjon
            JLabel hullLabel = new JLabel("Antall hull per seksjon: ");
            hullLabel.setFont(new Font("", Font.PLAIN, 16));
            sone.add(hullLabel);

            Integer antGjHull = new
Integer(((Sone)(soneListe.get(i))).getAntHull());
            JLabel antHull = new JLabel(antGjHull.toString());
            antHull.setFont(new Font("", Font.PLAIN, 16));
            sone.add(antHull);

```

```

//Strømning i hver sone
JLabel rateLabel = new JLabel("Strømningsrate for sonen[m3/d]: ");
rateLabel.setFont(new Font("", Font.PLAIN, 16));
sone.add(rateLabel);

Double strRate =new
Double((((Sone)soneListe.get(i)).getStrRate(index)*3600*24)/1.332);
JLabel strRateSone = new JLabel(strRate.toString());
strRateSone.setFont(new Font("", Font.PLAIN, 16));
sone.add(strRateSone);

//Trykktap i punch i sonen
JLabel punchTapLabel = new JLabel("Trykktap
Respunchseksjoner[bar]: ");
punchTapLabel.setFont(new Font("", Font.PLAIN, 16));
sone.add(punchTapLabel);

if(i > 0){
    Double punchTap =new
Double((((Sone)soneListe.get(i)).getDpPunch(index)*2)/Math.pow(10,5));
    JLabel punchTapSone = new JLabel(punchTap.toString());
    punchTapSone.setFont(new Font("", Font.PLAIN, 16));
    sone.add(punchTapSone);
}
else{
    Double punchTap =new
Double((((Sone)soneListe.get(i)).getDpPunch(index))/Math.pow(10,5));
    JLabel punchTapSone = new JLabel(punchTap.toString());
    punchTapSone.setFont(new Font("", Font.PLAIN, 16));
    sone.add(punchTapSone);
}

resKort.add(sone);

}

JPanel reservoar = new JPanel();
reservoar.setLayout(new GridLayout(6,2));
TitledBorder reservBorder = new TitledBorder("Totalt for reservoaret");
reservBorder.setTitleFont(new Font("", Font.ITALIC, 18));
reservoar.setBorder(reservBorder);

//potensiell max, strømningsrate
JLabel totRateLabel = new JLabel("Maks strømningsrate[Sm3/d]: ");
totRateLabel.setFont(new Font("", Font.PLAIN, 16));
reservoar.add(totRateLabel);

Double totStrRate =new Double((controller.getRiktigRate()*3600*24)/1.332);

```

```

JLabel totStrRateSone = new JLabel(totStrRate.toString());
totStrRateSone.setFont(new Font("", Font.PLAIN, 16));
reservoar.add(totStrRateSone);

//Brønntrykk maks
JLabel bronnTrykkLabel = new JLabel("ved brønntrykk[bar]: ");
bronnTrykkLabel.setFont(new Font("", Font.PLAIN, 16));
reservoar.add(bronnTrykkLabel);

Double bTrykk =new
Double((((Sone)(soneListe.get(0))).getBronnTrykk(index))/Math.pow(10,5));
JLabel bronnTrykk = new JLabel(bTrykk.toString());
bronnTrykk.setFont(new Font("", Font.PLAIN, 16));
reservoar.add(bronnTrykk);

//Produksjonsindeks
JLabel prodLabel = new JLabel("Produksjonsindeks[Sm3/d/bar]: ");
prodLabel.setFont(new Font("", Font.PLAIN, 16));
reservoar.add(prodLabel);

Double prod =new Double(controller.finnProdI(totStrRate.doubleValue(),
bTrykk.doubleValue()));
JLabel prodIndeks = new JLabel(prod.toString());
prodIndeks.setFont(new Font("", Font.PLAIN, 16));
reservoar.add(prodIndeks);

//min strømningsrate
JLabel minTotRateLabel = new JLabel("Minimum strømningsrate[Sm3/d]: ");
minTotRateLabel.setFont(new Font("", Font.PLAIN, 16));
reservoar.add(minTotRateLabel);

Double minTotStrRate =new
Double((controller.getMinRate()*3600*24)/1.332);
JLabel minTotStrRateSone = new JLabel(minTotStrRate.toString());
minTotStrRateSone.setFont(new Font("", Font.PLAIN, 16));
reservoar.add(minTotStrRateSone);

//Brønntrykk min
JLabel minBronnTrykkLabel = new JLabel("ved brønntrykk[bar]: ");
minBronnTrykkLabel.setFont(new Font("", Font.PLAIN, 16));
reservoar.add(minBronnTrykkLabel);

Double minBTrykk =new
Double((((Sone)(soneListe.get(0))).getBronnTrykk(112))/Math.pow(10,5));
JLabel minBronnTrykk = new JLabel(minBTrykk.toString());
minBronnTrykk.setFont(new Font("", Font.PLAIN, 16));
reservoar.add(minBronnTrykk);

//Produksjonsindeks

```



```

        JLabel prodMinLabel = new JLabel("Produksjonsindeks[Sm3/d/bar]: ");
        prodMinLabel.setFont(new Font("", Font.PLAIN, 16));
        reservoar.add(prodMinLabel);

        Double minProd =new
Double(controller.finnProdI(minTotStrRate.doubleValue(), minBTrykk.doubleValue()));
        JLabel minProdIndeks = new JLabel(minProd.toString());
        minProdIndeks.setFont(new Font("", Font.PLAIN, 16));
        reservoar.add(minProdIndeks);

        resKort.add(reservoar);

        regBakgrunn.add(resKort, "resultat");

        CardLayout cl = (CardLayout)(regBakgrunn.getLayout());
        cl.show(regBakgrunn, "resultat");

    }

    /*
    *Metoden legger til ny info i informasjonsvinduet
    *@param String info
    */

    public void oppdaterInfoVindu(String nyInfo){
        infoTekst.setText(nyInfo);
    }

} //GUI

```