

SEKVENSIELL OPTIMERING AV OLJEPRODUKSJON UNDER USIKKERHET

av

NIKITA MORATCHEVSKI

MASTEROPPGAVE

for graden

Master i Modellering og dataanalyse

(Master Of Science)



*Det matematisk-naturvitenskapelige fakultet
Universitetet i Oslo*

November 2011

*Faculty of Mathematics and Natural Sciences
University of Oslo*

1 Innledning

Oljeindustrien utviklet seg til å bli den mest inntektsgivende næring for den norske staten. Statens inntekter fra denne virksomheten kommer hovedsakelig inn på to måter; gjennom skatter og avgifter, samt ved salg av den oljen og naturgassen som staten selv eier. Dessuten har næringen ringvirkninger for andre næringer som yter tjenester til oljeindustrien, som f.eks. forskning, teknologiutvikling og logistikk. Petroleumsvirksomheten i Norge sammenliknet med annen næringsvirksomhet utgjør omtrent $1/5$ av bruttonasjonalproduktet.

Totalt utvinnbare oljereserver på norsk sokkel er regnet som ca 4250 millioner standardkubikkmeter - tilsvarende 26,7 milliarder fat. Dette tallet endrer seg etter hvert som nye felt blir funnet og erklært kommersielle, og når utvinningsgraden i eksisterende felt økes som følge av teknologi- og feltutvikling. Totalt er det om lag 50 store og små oljefelt i produksjon eller under utbygging.

I tillegg til å være et betydelig inntektskilde, er petroleumsindustrien et område som det er knyttet stor usikkerhet til. Blant annet grunnet svinginger i oljemarkedet, men ikke minst usikkerheten som er knyttet til det enkelte reservoaret/feltet.

Det er vanskelig å avgjøre nøyaktig hvor mye olje det er å hente ut fra et potensielt oljefelt som ligger i havbunnen langt under overflaten. Oljen som blir pumpet opp må bearbeides før den går videre som sluttprodukt (som for eksempel går til drivstoff eller oppvarming). Siden utbyggingen av et eventuelt full prosessanlegg som både pumper opp og bearbeider produktet kan koste milliarder, kan man konkludere med at det er betydelig økonomisk og teknologisk usikkerhet knyttet til hvert enkelt reservoar.

Optimeringene i forbindelse med olje- og gassutvinning er et spennende og omfattende forskningsfelt. Noen av de siste studiene på dette feltet er blant annet Horne (2002), Merabet & Bellah (2002) og Neiro & Pinto (2004). I Huseby & Haavardsson (2009) ble det sett nærmere på en optimeringssituasjon der flere reservoarer deler samme prosessanlegg. Der ble såkalte begrensningsfaktorene introdusert, slik at for et gitt tidspunkt vil produksjonen fra hver av reservoarene bli nedjustert av passende begrensningsfaktorer som sørger for at produksjonen ikke overstiger kapasiteten ved prosessanlegget. En *Produksjonsstrategi* blir da en funksjon definert for alle tidspunktene $t \geq 0$ som representerer begrensningsfaktorene som anvendes på reservoarene i tid t . Basert på produksjonsprofilmodellene som ble presentert i Arps (1945), har Haavardsson & Huseby (2007) utviklet en generell modell for olje- og gassproduksjonen ved å bruke differensiallikninger. Huseby & Haavardsson (2009) benyttet denne til å utvikle en fremgangsmåte for å finne en produksjonsstrategi som blir optimal gitt visse betingelser. I Huseby

& Haavardsson (2010) ble også usikkerheten i forbindelse med produksjonen introdusert.

2 Min oppgave

I denne oppgaven antar vi at gjennom seismiske undersøkelser ble det oppdaget et oljefelt. En av utfordringene i dette tilfellet er knyttet til vanskeligheten av å avgjøre nøyaktig hvor mye olje det er å hente ut.

For at råoljen skal omdannes til ulike typer av petroleum, må den gå gjennom et industrielt prosessanlegg - et oljeraffineri. Raffineringen foregår ved partiell destillasjon der de ulike fraksjonene tas ut ved bestemte temperaturer.

Videre i oppgaven antar vi at vi må leie prosesskapasitet ved et slikt anlegg som skal bearbeide oljen som blir utvunnet i påfølgende tidsperiode, siden en typisk situasjon vil være at ett prosessanlegg betjener flere oljereservoarer. For hver produksjonsperiode gjelder følgende: produksjonen er begrenset av to faktorer: det potensielle produksjonsvolumet *eller* det antallet fat som kan bli bearbeidet ved prosessanlegget. Hvis vi bestiller større kvota enn det vi har behov for så ender vi opp med å betale for mye, siden vi ikke får utnyttet hele den destilte kvoten ved prosessanlegget. På en annen side, hvis kvotaen er *mindre* enn det potensielle produksjonsvolumet for en gitt tidsperiode, så blir inntektet for denne perioden redusert, siden det antallet fat som blir til overs må da produseres på et senere tidspunkt. Fra et økonomisk perspektiv, vil det redusere nåverdien av oljeproduksjonen.

Siden det naturlige målet i forbindelse med dette feltet er å minimere utgifterne/kostnadene og maksimere inntekten, blir min oppgave å forsøke å estimere de optimale produksjonskvotene for en sekvens av tidsperioder som vil maksimere nettoinntekten knyttet til leie av prosessanlegget. Med andre ord er vi interessert i modellen for hvordan usikkerheten utvikler seg etterhvert som vi produserer.

Først vil jeg introdusere noen variabler/notasjoner som blir brukt i oppgaven.

3 Størrelser/notasjoner

Vi skal se på det totale utvinnbare volumet i reservoaret, V , som er en stokastisk variabel. Videre har vi decline raten, D , som kan beskrives som andelen av gjenværende volum som kan produseres i en gitt periode. Etter hvert som reser-

voaret tømmes vil evnen til å opprettholde produksjonen svekkes og utvinningen begynne å falle - uttømmingen fører til ulike grader av fall ('decline') i den årlige utvinningen. Også denne er en stokastisk variabel, som naturlig nok ligger i intervallet mellom 0 og 1. I praksis ville den ikke vært over 0.5 og i de aller fleste tilfeller ville den også ligget godt under, for eksempel mellom 0.1 og 0.3. Merk at jo høyere decline raten er desto mer økonomisk gunstig er det for oljeselskapet siden større andel (av gjenværende volum?) blir hentet ut i hver tidsperiode, noe som reduserer produksjonstiden og dermed blir de totale kostnadene i forbindelse med produksjonen lavere).

Videre har vi produksjonen fra en enkelt periode:

$$q_j = \text{produksjonen i } j\text{te periode,} \quad j = 1, 2, \dots$$

Vi innfører også kumulative produksjonsvolumer:

$$Q_j = \sum_{i=1}^j q_i = \text{Den totale produksjonen til og med } j\text{te periode,} \quad j = 1, 2, \dots$$

Innfører også $Q_0 = 0$

Etterhvert som man utvinner olje fra reservoaret, vil trykket synke. Dermed går også produksjonsraten ned. Dersom man produserer *uten* noen kvotebegrensninger (dvs. *urestriktert* produksjon), så antar vi at vi har følgende sammenheng mellom produksjonen i j te periode, q_j , og det produserte volumet før perioden starter, Q_{j-1} :

$$q_j = D(V - Q_{j-1}), \quad j = 1, 2, \dots$$

Merk at siden $Q_0 = 0$, så blir $q_1 = DV$.

Tilsvarende sammenheng *med* kvoteberegningen (dvs. *restriktert* produksjon) blir:

$$q_j = \min(K_j, D(V - Q_{j-1})), \quad j = 1, 2, \dots$$

der K_j er prosesskvoten for j te periode. Siden $Q_0 = 0$, får vi at $q_1 = \min(K_1, DV)$.

Videre innfører vi to nye variabler - *inntektsraten* og *utgiftsraten*, som skal henholdsvis betegnes δ og κ , der vil anta at $\kappa < \delta$.

Utgiften i j te periode blir da gitt ved:

$$\kappa K_j = \text{Utgift i } j\text{te periode,} \quad j = 1, 2, \dots$$

Inntekten i j te periode er gitt ved:

$$\delta q_j = \text{Inntekt i } j\text{te periode,} \quad j = 1, 2, \dots$$

Resultatet for en gitt periode er differansen mellom inntekten og utgiften i den aktuelle perioden. Uttrykket for resultatet for den j te perioden blir da på formen:

$$\delta \min(K_j, D(V - Q_{j-1})) - \kappa K_j = R_j = \text{Resultat i } j\text{te periode,} \quad j = 1, 2, \dots \quad (1)$$

Vi observerer at formelen for resultatet kan forenkles til følgende, hvis $K_j \leq D(V - Q_{j-1})$:

$$(\delta - \kappa)K_j$$

I det motsatte tilfellet blir resultatet på formen:

$$\delta D(V - Q_{j-1}) - \kappa K_j$$

Merk at resultatet blir påvirket av avgjørelsen hvor stor kvoten K_j er, så man kan si at resultatet (j -te periode) R_j er en funksjon av K_j , og kan eventuelt skrives på formen $R_j(K_j)$.

Det samlede (totale) resultatet R blir da

$$R = \sum_{j=1}^n R_j \quad (2)$$

hvor n er antall perioder.

4 Fremgangsmåte

I denne oppgaven vil vi anta at *apriorifordelingene* til de to ukjente stokastiske variablene D (decline raten) og V (totale utvinnbare volumet) er kjent. Utgifts- og inntektsratene antas å være konstante og kjente.

Selve produksjonen betraktes i *diskret* tid, noe som betyr at produksjonsraten $D(V - Q_{j-1})$ er ett tall som gjelder for hele den j -te tidsperioden, $j = 1, 2, \dots$

Jeg velger å modellere volumet, V , ved å bruke lognormalfordelingen, en fordeling som er mye brukt i risiko- og pålitelighetsanalyse. En stokastisk variabel Y sies å være lognormalfordelt med parametrene μ and σ hvis $Y = \ln(X)$ er normalfordelt med forventning μ og standardavvik σ . Tetthetsfunksjonen for lognormalfordelingen er på formen:

$$f(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \exp \left\{ -\frac{(\ln x - \mu)^2}{2\sigma^2} \right\}$$

Lognormalfordelingen er en skjev fordeling med en litt tyngre høyrehale (som er godt egnet i dette tilfellet siden det er grunn til å tro at volumet kan være høyere enn forventningen, men ikke så veldig mye mindre siden volumet må være positivt).

For å modellere decline rate-variabelen D er det et naturlig valg å bruke uniformfordelingen. Uniformfordelingen er en kontinuerlig fordeling definert på et intervall, beskrevet ved to parametre, vanligvis kalt for a og b som er henholdsvis den nedre og den øvre grensene for det aktuelle intervallet.. Tetthetsfunksjonen for uniformfordelingen ser slik ut:

$$f(x; a, b) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & \text{ellers} \end{cases}$$

Parametrene til fordelingene til variablene D og V kommer til å varieres gjennom oppgaven, dog jeg vil se på en såkalt basic-case. Når det gjelder decline raten D så valgte vi å la denne være uniformt fordelt på intervallet 0.1 til 0.5. . Når det gjelder parametrene for volumet V , blir disse spesifisert nærmere i kapittelet *Beregninger*.

Alle numeriske beregninger og statistiske simuleringer gjort i forbindelse med denne oppgaven er gjort ved hjelp av programmeringsspråket Java.

4.1 Hvordan optimere produksjonskvotene for de ulike tidsintervallene?

Det dukker opp flere utfordringer når man skal forsøke å optimere den optimale K -verdien, altså den verdien for prosesskvoten som gir oss best økonomisk gevinst. Den største utfordringen er naturlig nok usikkerheten i det totale utvinnbare volumet og decline raten som påvirker den utvunnede mengden for hver tidsperiode.

Når vi skal modellere så er det to mulige situasjoner som kan dukke opp:

1) Vi kan få en situasjon hvor det er *kvoten* som er den begrensende faktoren, det vil si at det totale gjenværende volumet minker med K_j , og ikke produktet $D(V - Q_{j-1})$, med andre ord at vi observerer at:

$$q_j = K_j$$

Det vil si, at siden $q_j = \min(K_j, D(V - Q_{j-1}))$ så må D og V tilfredsstillе ulikheten:

$$q_j \leq D(V - Q_{j-1}) \quad j = 1, 2, \dots \quad (3)$$

Når vi har denne situasjonen så får vi ikke så mye informasjon om produktet DV .

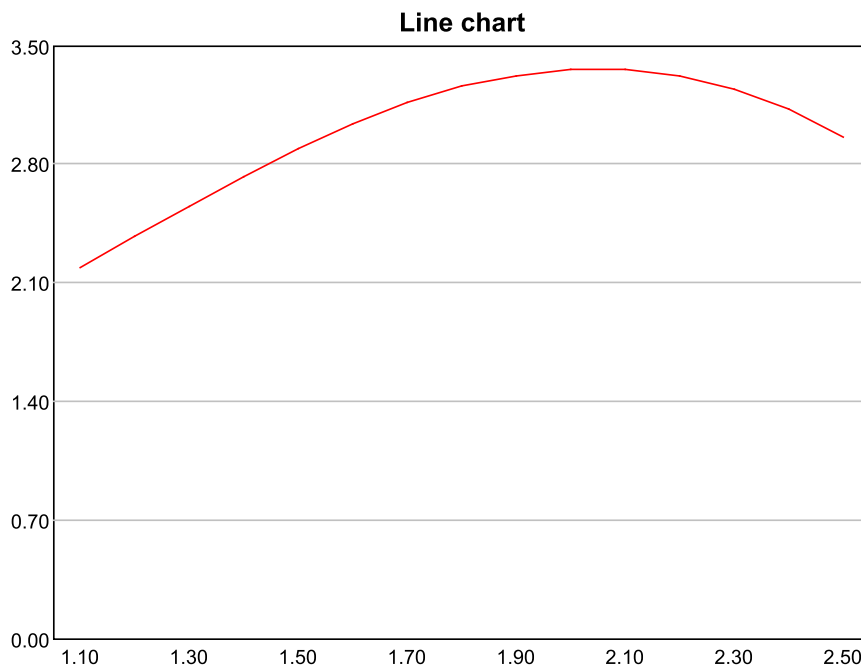
2) Den andre situasjonen som kan forekomme er at vi ikke klarer å oppfylle den bestilte kvoten, slik at kvoten ikke er den begrensende faktoren lenger, det vil si at D og V må tilfredsstillе likheten:

$$q_j = D(V - Q_{j-1}) \quad (4)$$

I denne situasjonen får vi mye info om D og V , mens i det første tilfellet får vi ikke så mye informasjon om disse.

Når vi da skal utføre en sekvensiell optimering så må vi ha simuleringsmetoder som skal kunne håndtere begge situasjonene, ikke minst med tanke på at informasjonen stadig oppdateres underveis.

Etter hver tidsperiode så sitter vi igjen med mer og mer informasjon om produksjonsparametrene. Derfor må det stadig utføres oppdateringer når det gjelder fordelingene. Hovedutfordringen er at denne oppdaterte informasjonen vil komme



Figur 1: Forventet resultat i 1-te periode som funksjon av kvoten K_1

i form av likninger og/eller ulikheter. Hvordan vi skal håndtere dette skal vi se på nærmere i neste seksjon - Simuleringsmetoder.

4.2 Simuleringsmetoder

Som forklart over, så er vi nødt til å håndtere to mulige typer informasjon svarende til om det er *produksjonskvoten* (K_j) eller *produksjonskapasiteten* $D(V - Q_{j-1})$ som er den begrensende faktoren.

La oss se nærmere på hvordan dette kan løses ved å beskrive simuleringene i de ulike periodene.

For å finne den optimale kvoten for den første perioden, K_1 , starter vi med å simulere variablene D og V (decline-rate og volum). Siden vi ikke har noe annet enn apriori informasjon om disse på dette tidspunktet, kan dette gjøres enkelt ved å trekke *ubetinget* fra respektive apriorifordelingene. Ved å bruke formlene for inntekt, kostnader og resultat (gitt i seksjon 'Størrelser/notasjoner') kan man lage en graf som viser hvordan det forventede resultatet blir påvirket av forskjellige verdiene for K_1 , og deretter finne en verdi for K_1 som vil gi høyest mulig resultat. (Se figur 1)

Vi antar nå at i første periode er det kvoten som er den begrensende faktoren, altså $q_1 = K_1$. Vi får dermed at D og V må tilfredsstillе ulikheten $D(V - Q_0) \geq q_1$, derfor må vi trekke variablene D og V for neste periode *gitt* denne informasjonen, det vil si at vi filtrerer ut de verdiene av D og V som *ikke* tilfredstiller denne betingelsen. Med andre ord, for den andre tidsperioden vil vi kun beholde de kombinasjonene av D og V der $D(V - Q_0) = DV \geq q_1$.

For de etterfølgende periodene går man frem på tilsvarende vis. Så lenge vi kun har observert perioder der *produksjonskvotene* har vært de begrensende faktorene, så vil simuleringen av D og V for $j + 1$ -periode kunne gjøres ved å trekke fra de ubetingede fordelingene for D og V , og deretter filtrere bort de verdiene som ikke tilfredstiller betingelsene:

$$D(V - Q_{j-1}) \geq q_j \quad j = 1, 2, \dots$$

der

$$Q_i = \sum_{r=1}^i K_r \quad i = 1, 2, \dots$$

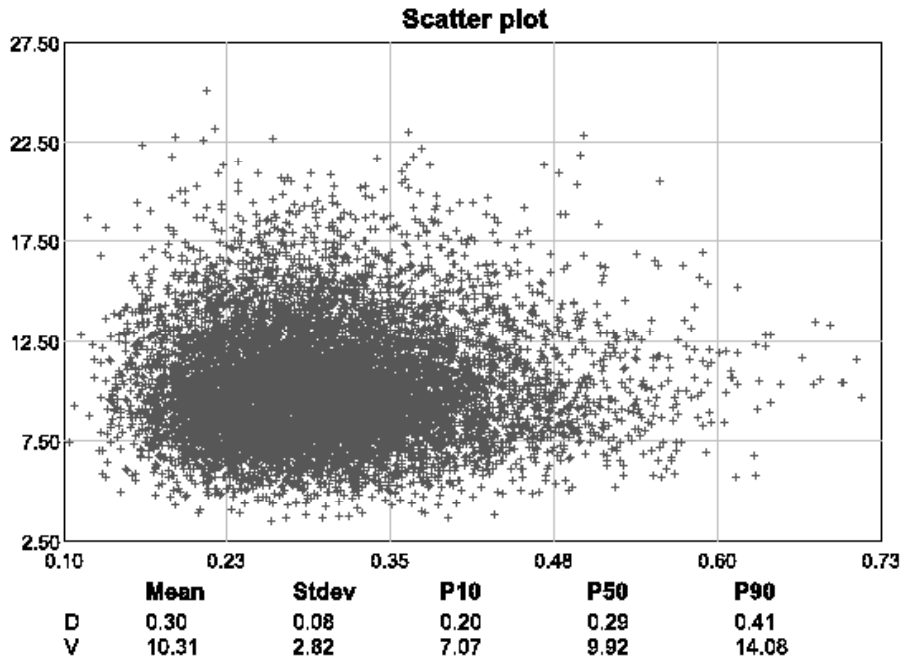
Dette er en såkalt *filtreringsmetode*.

Simuleringene av D og V brukes hele tiden til å estimere forventet resultat som funksjon av kvoten, og basert på dette kan vi så velge den optimale kvoten for neste periode.

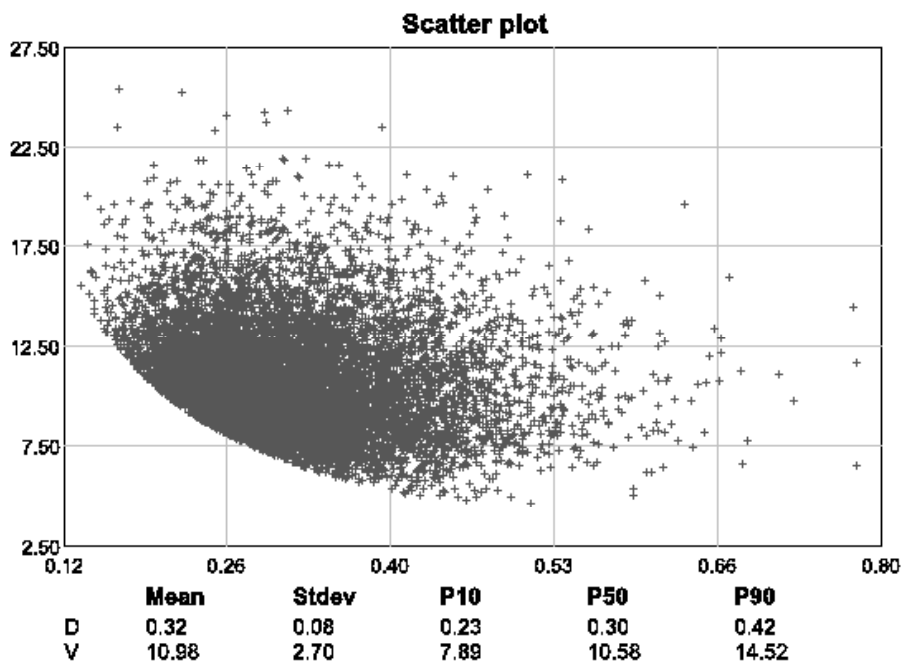
Merk at filtreringsmetoden sikrer oss at vi får med oss *all* informasjon fra de tidligere tidsperiodene.

Dette kan illustreres grafisk (se figur 2 og 3). Figur 2 viser oss 5000 simulerte variabler før filtreringen er foretatt, det vil si at ingenting er blitt hentet ut fra reservoaret ennå. Figur 3 viser variablene etter at den første filtreringsbetingelsen er tilfredstilt.

Filtreringsmetoden som er beskrevet over er relativt enkel å implementere men den har flere ulemper. Blant annet for hver tidsperiode så sitter vi igjen med stadig færre observasjoner (som illustrert i figur 2). Dessuten kan ikke denne metoden brukes i den andre situasjonen som vil dukke opp, nemlig når produksjonskapasiteten er den begrensende faktoren - i dette tilfellet går filtreringsbetingelsen fra å være en *ulikhet* til å være en *likning* og da risikerer vi enten å få veldig få observasjoner eller å ikke sitte igjen med noen observasjoner i det hele tatt (i figur 2 ville det vært observasjoner som befinner seg akkurat på filtreringsgrensen).



Figur 2: Ufiltrerte variabler



Figur 3: Filtrerte variabler

Et alternativ til filtreringsmetoden er å bruke en form for *forkastningsmetode*, der vil for hver ny tidsperiode vil kun beholde de genererte verdiene som tilfredstiller betingelsene. Fordelen med denne metoden er at vi alltid sitter igjen med samme antall observasjoner. Ulempen ved forkastningsmetoden er økt simuleringstid.

Hvis vi derimot skulle få en likning så er løsningen å bruke mer kompliserte simuleringsmetoder, som for eksempel Markov Chain Monte Carlo-metoder (MCMC). MCMC-metoder er en klasse av algoritmer som har et formål å simulere tilfeldige variabler fra sannsynlighetsfordelinger, ved å konstruere en Markov kjede som har fordelingen vi er interessert i som ekvilibriumfordeling.

Metoden vi velger å bruke til å generere en mengde ved et såkalt likhetstilfelle, altså et tilfelle der kvoten K_j ikke er den begrensende faktoren er Metropolis-Hastings algoritmen.

Metropolis-Hastings (MH) algoritmen tillater oss å simulere fra en vilkårlig sannsynlighetsfordeling som vi må kjenne i form av en unormalisert tetthet. MH gir oss ikke en stokastisk variabel som har den rette fordelingen umiddelbart men derimot produserer en sekvens av stokastiske variabler som *konvergerer (i fordeling)* mot den 'riktige' fordelingen.

For å beskrive hvordan algoritmen fungerer antar vi at vi er interessert i X som er en tilfeldig variabel (vektor) fra en fordeling med unormalisert tetthet $f(x)$. Videre trenger vi noe som kalles forslagsfordeling, som i sin tur er en betinget sannsynlighetsfordeling, og kan skrives som $q(y|x)$. Utfordringen kan være å finne en bra forslagsfordeling. Et krav til denne er at den bør kunne evalueres nokså lett for et hvilket som helst par x, y , og at vi har mulighet til å generere effektivt fra denne fordelingen. Forslagsfordelingen q skal enten ha samme utfallsrom som X eller i hvertfall generere verdier i hele utfallsrommet med høy sannsynlighet. Konvergenssegenskaper til MH avhenger typisk nettopp av valget av forslagsfordelingen.

Videre må vi ha en startverdi, som kan være en hvilken som helst verdi fra mengden av alle mulige verdier for X . Vi kaller denne startverdien for x_0 . Vi skal generere en sekvens x_1, x_2, \dots av stokastiske variabler som konvergerer i fordelingen mot den ønskede. Dette gjøres på en følgende måte: anta at vi har generert x_1, x_2, \dots, x_s . Forutsatt at vi har valgt en passende forslagsfordeling $q(y|x)$, genereres x_{s+1} som følger:

- Generer $y_{s+1} \sim q(\cdot|x_s)$
- Evaluer *Hasting ratio*:

$$r = \frac{f(y_{s+1})q(x_s|y_{s+1})}{f(x_s)q(y_{s+1}|x_s)}$$

- Sett x_{s+1} lik y_{s+1} med sannsynlighet $\min(1,r)$ eller behold den 'gamle', med andre ord sett x_{s+1} lik x_s .

Hvis fordelingen q er symmetrisk, det vil si $q(y|x) = q(x|y)$, blir Hastings ratio lik

$$\frac{f(y)}{f(x)}$$

som også i noen ganger blir betegnet som *Metropolis ratio*.

Hvordan kan MH-algoritmen brukes i vår oppgave når vi har et tilfelle med likhet? Vi antar at vi på grunnlag av observert produksjonsrate i j -te periode har funnet at D og V tilfredsstillers:

$$q_j = D(V - Q_{j-1})$$

Som nevnt over så antar vi at apriorifordelingene til parametrene D og V er kjent. Vi kan betegne disse fordelingene som for eksempel $f_1(d)$ og $f_2(v)$. Før vi observerer hvor mye vi faktisk har utvunnet i periode 1 (q_1), så kan vi trekke variablene fra direkte. Men denne metoden fungerer kun *før* vi har observert en verdi for q . Når vi vet verdien til q så må vi ta hensyn til det og dermed blir målet vårt å trekke D og V gitt denne betingelsen. For å kunne utføre det i praksis, løser vi (1) med hensyn på V :

$$V = Q_{j-1} + \frac{q_j}{D} = g(D, q)$$

Nå kan man simulere på følgende måte:

Først, trekke variabelen D fra en betinget sannsynlighetsfordeling til D gitt q . Deretter kan vi sette inn q og den nye verdien til D i $g(D, q)$ for å beregne V .

Den betingede sannsynlighetstettheten til D gitt q er proporsjonal med den simultane tettheten til D og q :

$$f(d|q) \propto f(d, q)$$

der $f(d, q)$ er gitt på formelen:

$$f(d, q) = f_1(d)f_2\left(\frac{q}{d} + Q\right)\left|\frac{1}{d}\right|$$

En litt annen fremgangsmåte må til når vi har et tilfelle med *ulikhet*. Her vil det være litt enklere å simulere enn i likhetstilfellet ved å bruke en såkalt forkastningsmetode. For hver ny periode vil vi bare beholde de genererte verdiene som tilfredsstillende betingelsene.

Vi legger merke til at hvis vi sitter igjen med to likheter når vi optimerer en sekvens, så har vi faktisk et tilfelle med to likninger med to ukjente, og dette fører til at usikkerheten blir eliminert. Dette er et ganske så forenklet tilfelle siden vi kun har to stokastiske variabler, D og V . I praksis ville det vært betydelig flere usikkerhetsparametere som beskriver diverse egenskaper ved reservoaret og for å kunne eliminere usikkerheten vil vi trenge like mange likheter.

Til slutt kan vi se på situasjonen hvor vi *både* har én likhet og et sett med ulikheter. I dette tilfelle simuleres D og V ved at vi først bruker Metropolis-Hastings og deretter sjekkes det om D og V tilfredsstillende ulikheten.

4.3 Mer om simuleringsmetoder

Som nevnt i seksjonen 'Størrelser/notasjoner', kan resultatet for j te periode betegnes som funksjon av kvoten og skrives på formen $R_j(K_j)$. Ved å utføre Monte Carlo simuleringer der V og D blir generert fra sin simultanfordeling, kan vi estimere den forventede resultatene, $E[R_j(K_j)]$ og deretter velge en verdi for K_j som vil maksimere denne.

Alternativt kan man estimere den *deriverte* av den forventede resultatet som en funksjon av kvoten og så velge en verdi for K_j slik at den deriverte blir lik null. Det viser seg at denne metoden er lettere å gjennomføre og at den gir mer nøyaktige resultater.

For å kunne estimere den deriverte av $E[R_j(K_j)]$ med hensyn på K_j kan vi bruke at:

$$\frac{\partial}{\partial K_j} E[R_j(K_j)] = E\left[\frac{\partial}{\partial K_j} R_j(K_j)\right]$$

Dette er et resultat som kan bevises ved bruk av Lebesgue's dominerte konvergensteorem:

Videre får vi at:

$$\frac{\partial}{\partial K_j} R_j(K_j) = \begin{cases} (\delta - \kappa) & K_j < D(V - Q_{j-1}) \\ -\kappa & K_j > D(V - Q_{j-1}) \end{cases}$$

I det 'optimale' tilfellet der $K_j = D(V - Q_{j-1})$ er den deriverte udefinert, men sannsynligheten for at dette tilfellet oppstår er lik null så dette bidrar ikke til forventningen av den deriverte.

Siden vi nå har uttrykket for den deriverte av resultatet med hensyn på kvoten K_j kan vi nå finne et uttrykk for forventningsverdien av dette:

$$E\left[\frac{\partial}{\partial K_j} R_j(K_j)\right] = (\delta - \kappa)P[D(V - Q_{j-1}) > K_j] - \kappa P[D(V - Q_{j-1}) < K_j]$$

Vi observerer at siden D og V er kontinuerlig fordelte så vil:

$$P(D(V - Q_{j-1}) < K_j) = 1 - P(D(V - Q_{j-1}) > K_j)$$

Hvis den deriverte settes lik null, får vi at:

$$P(D(V - Q_{j-1}) < K_j) = \frac{\delta - \kappa}{\delta}$$

Siden parametrene κ og δ antas å være kjente på forhånd, kan den optimale kvoten K_j dermed beregnes ved å estimere den kumulative sannsynlighetsfordelingen til $D(V - Q_{j-1})$. Når vi har denne fordelingen, velges den optimale K_j -en som $(\frac{\delta - \kappa}{\delta})$ -persentilen i denne fordelingen.

5 Hoveddel/beregninger

I denne seksjonen kommer vi til å se nærmere på eksempler med konkrete parametre/tall. I samtlige tilfellene antas D og V å være uavhengige apriori. Som nevnt tidligere, vil det totale utvinnbare volumet V antas til å være lognormalfordelt med en viss forventning, $E[V]$, og standardavvik $sd[V]$, mens decline raten D er uniformfordelt på intervallet 0.1 til 0.3.

Verdiene for utgifts- og kostnadsratene, κ og δ er deterministiske, og vil bli satt til henholdsvis 30 og 50. (Disse tall kan typisk representere oljepris, for eksempel i USD per barrel).

Vi velger å se på 8 produksjonsperioder og se på det kumulative resultatet etter den siste utvinningsperioden er avsluttet. Vi vil i tillegg se på det diskonterte resultatet, der vi bruker diskonteringsrente på 5 prosent. Vi vil i første omgang se hva resultatet blir i en såkalt base-case, og hvordan forandring i diverse parametere vil påvirke produksjonsresultatet.

I en base-case nevnt i forrige avsnittet antas det at forventningsverdien til volumet V er 12 (mln. barrell), mens standardavviket antas å være lik 1. Decline-raten er uniform på intervallet $[0.1, 0.3]$.

5.1 Beregninger med sanne verdier

I denne seksjonen vil vi anta at *sanne* verdiene for V og D er henholdsvis 12 og 2, det vil si at sanne verdiene er lik forventningene i de respektive apriorifordelingene (noe som ikke alltid vil være tilfelle, dette vil vi komme tilbake til).

I samtlige eksemplene under er det brukt Monte Carlo simuleringsteknikker til å estimere den kumulative fordelingen til $D(V - Q_{j-1})$. Disse i sin tur brukes til å finne kvotene K_j .

La oss nå se på konkrete resultater fra simuleringene. Vi begynner med en base-case, der $E[V] = 12.0$, $sd[V] = 1.0$ og de *sanne* verdiene til V og D antas til å være lik forventningene, det vil si hhv. 12 og 0.2. Tabell 1 viser oss resultatene av kjøringen.

i	Q_{i-1}	K_i	q_i	$D(V - Q_{i-1})$	R_i
1	0,00	2,14	2,14	2,40	42,86
2	2,14	2,21	1,97	1,97	32,29
3	4,11	1,56	1,56	1,58	31,22
4	5,68	1,28	1,26	1,26	24,83
5	6,94	1,01	1,01	1,01	20,24
6	7,95	0,81	0,81	0,81	16,19
7	8,76	0,65	0,65	0,65	12,95
8	9,41	0,52	0,52	0,52	10,36

Tabell 1: Base-case: de sanne verdiene for D og V er henholdsvis 0.2 og 12

Når vi summerer resultatene for de 8 produksjonsperiodene (observerer disse i siste kolonne i tabellen) ser vi at det kumulative resultatet blir 190.95 millioner

dollar. Dersom vi også tar hensyn til diskonteringen med en diskonteringsrate på 5 prosent så blir det totale resultatet på 169.75 millioner dollar. (Denne diskonteringsraten vil også bli brukt i senere eksempler).

La oss se nærmere på tabellen:

I første periode, $i = 1$, observerer vi at $q_1 = K_1 = 2.14$. Det vil si at den sanne verdien av $D(V - Q_0)$ er *større eller lik* den observerte verdien for q_1 , som er 2.14 i dette tilfellet. I første periode er $D(V - Q_{i-1}) = D(V - Q_0) = DV$ siden $Q_0 = 0$. Så den kumulative fordelingen til $f(Q_0)$ estimeres ved å trekke fra apriorifordelingen til (V,D)

Ser vi videre på periode 2, $i = 2$, så er den observerte verdien for q_2 lik 1.97, som er mindre enn $K_2 = 2.21$. Det innebærer at vi har $D(V - Q_1) = q_2$. I andre perioden er $D(V - Q_{i-1}) = D(V - Q_1)$. Merk at Q_1 er lik $q_1 = 2.14$. For å estimere den kumulative fordelingen til $D(V - Q_1)$ så må vi trekke fra den *betingede* simultanfordelingen for (V,D), altså gitt at $dv \geq 2.14$.

Neste tilfelle vi skal se på er følgende: vi vil anta at de *sanne* verdiene for volumet og decline-raten er henholdsvis 12 og 0.1, mens parametere i apriorifordelingene er samme som før. Det vil si nå at det sanne volumet er lik dens forventede verdien, mens decline-raten er nå bare *halvparten* av forventningen.

Resultatene fra simuleringene ser vi i tabell 2.

i	Q_{i-1}	K_i	q_i	$D(V - Q_{i-1})$	R_i
1	0.00	2.14	1.20	1.20	-4.10
2	1.20	1.07	1.07	1.08	21.41
3	2.27	0.96	0.96	0.97	19.25
4	3.23	0.87	0.87	0.88	17.35
5	4.10	0.78	0.78	0.79	15.66
6	4.88	0.70	0.70	0.71	14.10
7	5.59	0.63	0.63	0.64	12.69
8	6.22	0.57	0.57	0.58	11.47

Tabell 2: Sanne verdiene for D og V er henholdsvis 0.1 og 12

Ved å summere R_i -ene for samtlige 8 perioder ser vi at det kumulative resultatet er lik 107.83 millioner dollar, mens det totale resultatet etter diskonteringen

er redusert til 90.29 millioner dollar. Forklaringen på dette er at siden vi har en lavere decline-rate så vil det bli produsert færre mengder per produksjonsperiode.

Her ser vi at kvoten, $K_1 = 2.14$ er altfor høy ($2.14 > 1.20$). Det skyldes nok det altfor optimistiske valget for apriorifordelingen til D når den sanne verdien er såpass lav (0.1). Vi observerer at den altfor høye kvoten i første periode gir oss faktisk et tap - et negativt resultat på -4.10 millioner dollar. I de neste periodene derimot henter vi mer informasjon, og det påvirker valget av K_i -ene (som blir en del lavere), og resultatene snur seg og blir positive fra og med periode 2.

Neste naturlige steg er å sette den *sanne* verdien for D til 0.3, altså høyere enn forventningen, og se på resultatene. Intuitivt vil man 'gjette' at det vil påvirke resultatet positivt. Man kan se resultatene av kjøringen i tabell nr 3.

i	Q_{i-1}	K_i	q_i	$D(V - Q_{i-1})$	R_i
1	0,00	2,14	2,14	3,60	42,72
2	2,14	2,23	2,23	2,96	44,55
3	4,36	1,94	1,94	2,29	38,89
4	6,31	1,60	1,60	1,71	31,94
5	7,91	1,30	1,23	1,23	22,56
6	9,13	0,89	0,86	0,86	16,29
7	9,99	0,60	0,60	0,60	12,04
8	10,60	0,42	0,42	0,42	8,43

Tabell 3: Sanne verdiene for D og V er henholdsvis 0.3 og 12

Nå ser vi økning i det samlede resultatet til 217.42 millioner dollar, noe som gir oss 194.31 millioner dollar når diskonteringen er tatt i betraktning. Vi ser en økning fra det forrige tilfellet der forventningsveriden til decline-raten var 0.1, og det skyldes at siden vi har en høyere decline-rate så vil vi produsere mer per tidsperiode.

I dette tilfellet er de valgte kvotene lave i begynnelsen ($i = 1,2,3,4$) og at $q_i = K_i$ for $i = 1, \dots, 4$, mens i periodene 5 og 6 er kvotene for høye ($K_i > q_i$ for $i = 5,6$). Merk at i denne fasen kan vi faktisk oppnå/beregne de sanne verdiene for D og V eksplisitt.

La oss nå se på hva vil skje når den sanne verdien til D er lik 0.2, det vil si lik forventningsverdien, mens den sanne verdien for V blir satt ned til 10. Resulta-

tene kan man se i tabell 4.

i	Q_{i-1}	K_i	q_i	$D(V - Q_{i-1})$	R_i
1	0.00	2.16	2.00	2.00	35.31
2	2.00	1.66	1.60	1.60	30.33
3	3.60	1.28	1.28	1.28	25.60
4	4.88	1.02	1.02	1.02	20.48
5	5.90	0.82	0.82	0.82	16.38
6	6.72	0.66	0.66	0.66	13.11
7	7.38	0.52	0.52	0.52	10.49
8	7.90	0.42	0.42	0.42	8.39

Tabell 4: Sanne verdiene for D og V er henholdsvis 0.2 og 10

Vi får naturlig nok betydelig nedgang i resultatene: det samlede resultatet blir på 160.09 millioner dollar og det diskonterte resultatet blir på 142.64 millioner dollar.

Vi prøver nå med den sanne verdien for volumet på 14 (tabell 5):

i	Q_{i-1}	K_i	q_i	$D(V - Q_{i-1})$	R_i
1	0.00	2.13	2.13	2.80	42.52
2	2.13	2.22	2.22	2.37	44.46
3	4.35	1.94	1.93	1.93	38.21
4	6.28	1.43	1.43	1.54	28.57
5	7.71	1.12	1.12	1.26	22.34
6	8.82	0.90	0.90	1.04	17.99
7	9.72	0.74	0.74	0.86	14.84
8	10.47	0.62	0.62	0.71	12.45

Tabell 5: Sanne verdiene for D og V er henholdsvis 0.2 og 14

Ingen store overraskelser i dette tilfellet heller: vi får et samlet resultat på 221.37 millioner dollar og et diskontert resultat på 196.59 millioner dollar.

5.1.1 Sensitivitet med hensyn på tap av informasjon

Felles for alle de numeriske eksemplene vist over er at simuleringene ble utført på en slik måte at *informasjonen fra samtlige tidsperioder* er tatt med, det vil si at hele produksjonshistorien til og med periode $i - 1$ er tatt med i beregningen for å kunne optimere kvoten for periode i . Nå kan vi se hvordan situasjonen blir når kun data *fra den siste perioden* er utslagsgivende. Det første eksempelet vi vil se på er base-casen, der $E[V] = 12.0$, $SD[V] = 1.0$, og de sanne verdiene for V og D antas å være henholdsvis 12 og 0.2. I tabellen nedenfor ser vi resultatene:

i	Q_{i-1}	K_i	q_i	$D(V - Q_{i-1})$	R_i
1	0,00	2,14	2,14	2,40	42,89
2	2,14	2,23	1,97	1,97	31,67
3	4,12	1,56	1,56	1,58	31,21
4	5,68	1,50	1,26	1,26	18,37
5	6,94	0,99	0,99	1,01	19,88
6	7,94	0,97	0,81	0,81	11,60
7	8,75	0,64	0,64	0,65	12,71
8	9,38	0,65	0,52	0,52	6,57

Tabell 6: Base-case, kun data fra siste periode utslagsgivende

Her ser vi at det totale resultatet faller fra 190.95 millioner dollar i situasjonen hvor all foregående data var med i beregningen til 174.92 millioner dollar, mens det totale diskonterte resultatet reduseres fra 169.75 millioner dollar til 156.84 millioner dollar.

Hvis vi ser nærmere på tabellen ser vi at resultatene for de 3 første produksjonsperiodene er så å si de samme i begge tilfellene. Mens resultatet for den fjerde perioden er redusert med over 6 millioner dollar. Det som foregår nå er at i optimeringen av kvoten for den fjerde perioden er det nå kun informasjonen fra periode 3 som er tatt med. Dermed mister vi all informasjon fra de 2 første tidsperiodene. Som et resultat av dette blir den valgte kvoten for den fjerde periode, K_4 , for høy (1.50) sammenlignet med den potensielle produksjonsraten $f(Q_3) = D(V - Q_3)$, som er 1.26. Merk at samme situasjon oppstår også i tidsperiodene 6 (0.97 mot 0.81) og 8 (0.65 mot 0.52). Tap av informasjon blir grunnen til ikke-optimale verdiene for kvoten blir valgt, og dermed påvirker dette det akkumulerte resultatet negativt. Vi legger også merke til at differansen mellom de totale diskonterte resultatene er mindre enn differansen mellom de totale resultatene, dette skyldes at resultatene for periodene i begynnelsen var ganske like

og det er periodene i begynnelsen som påvirker det diskonterte resultatet mest. Dette viser tydelig at det lønner seg å ta resultatene fra samtlige tidligere periodene med i betraktningen.

La oss nå nedjustere den sanne verdien for decline-raten D til 0.1 (det vil si lavere enn forventningsverdien), og se hva resultatet blir (se Tabell 7).

i	Q_{i-1}	K_i	q_i	$D(V - Q_{i-1})$	R_i
1	0.00	2.13	1.20	1.20	-3.98
2	1.20	1.07	1.07	1.08	21.41
3	2.27	1.74	0.97	0.97	-3.64
4	3.24	0.87	0.87	0.88	17.31
5	4.11	1.42	0.79	0.79	-3.18
6	4.90	0.70	0.70	0.71	14.00
7	5.60	1.15	0.64	0.64	-2.50
8	6.24	0.56	0.56	0.58	11.29

Tabell 7: Sanne verdiene for D og V er henholdsvis 0.1 og 12; kun data fra siste periode utslagsgivende

Det gir oss et akkumulert resultat på kun 50.71 millioner dollar og tilsvarende diskontert resultat på 42.58 millioner dollar. La oss studere tabellen nærmere.

Vi observerer at resultatet for første periode blir negativ, og at fortegnet skifter etter hver tidsperiode som går slik at vi ender opp med negativ resultat i periodene 1,3,5 og 7, men har resultatverdiene som likner på de vi ellers har fått før i påfølgende periodene 2,4,6 og 8.

Vi legger merke til at det ikke er første gang vi ender opp med et negativt resultat - vi fikk også et resultat på -4.1 millioner kroner da vi tok all data med i betraktning og den sanne verdien for D var lik 0.1. Men da snudde trenden seg og resultatene ble positive ettersom mer informasjon ble hentet i de neste periodene. Her derimot er det kun data fra siste periode som gjelder, så selv om etter den første perioden blir kvoten nedjustert betydelig og vi får et positivt resultat på 21.41 millioner kroner i andre periode, så ender vi opp igjen med altfor høy verdi for K_3 i forhold til q_3 (henholdsvis 1.74 og 0.97) og igjen blir resultatet snudd til negativ.

La oss nå se hva vil skje når den sanne verdien for decline-raten overstiger

forventningen og er lik 0.3. Resultatene av simuleringene kan vi se i Tabell 8.

i	Q_{i-1}	K_i	q_i	$D(V - Q_{i-1})$	R_i
1	0.00	2.13	2.13	3.60	42.70
2	2.13	2.23	2.23	2.96	44.63
3	4.37	1.95	1.95	2.29	39.03
4	6.32	1.60	1.60	1.70	32.10
5	7.92	1.30	1.22	1.22	22.06
6	9.15	0.89	0.86	0.86	15.99
7	10.00	0.63	0.60	0.60	10.98
8	10.60	0.45	0.42	0.42	7.49

Tabell 8: De sanne verdiene for D og V er henholdsvis 0.3 og 12; kun data fra siste periode utslagsgivende

Her får vi det akkumulerte resultatet og det diskonterte resultatet på henholdsvis 214.97 og 192.52 millioner dollar. Ingen av de 8 periodene gir oss negativt resultat, og det skyldes mer produsjon per tidsperiode når den sanne verdien for D er såpass høy. Hvis vi skal sammenligne med de resultatene vi fikk da all historikken var med så ser vi at det er veldig liten forskjell i det akkumulerte resultatet (217.42 mot 214.97 millioner dollar).

Nå går vil tilbake til den sanne verdien for decline-raten lik forventningen i den uniforme apriorifordelingen, det vil si 0.2. Den sanne verdien for volumet V derimot reduseres fra 12 til 10. Se Tabell 9.

i	Q_{i-1}	K_i	q_i	$D(V - Q_{i-1})$	R_i
1	0.00	2.14	2.00	2.00	35.94
2	2.00	1.66	1.60	1.60	30.33
3	3.60	1.33	1.28	1.28	23.98
4	4.88	1.07	1.02	1.02	18.95
5	5.90	0.87	0.82	0.82	14.96
6	6.72	0.70	0.66	0.66	11.76
7	7.38	0.56	0.52	0.52	9.29
8	7.90	0.45	0.42	0.42	7.38

Tabell 9: Sanne verdiene for D og V er henholdsvis 0.2 og 10; kun data fra siste periode utslagsgivende

Her er: akkumulert resultat på 152.29 millioner USD, diskontert resultat - 136.64 millioner USD. Det er en nedgang fra da den sanne verdien var lik forventningen, noe som er ganske åpenbart. Det som er interessant er faktisk at vi faktisk fikk høyere nettoresultat i periodene 4 og 8.

Det vil nok være naturlig å forvente et høyere resultat hvis den sanne verdien for V ble satt høyere enn forventningen 12, la oss allikevel foreta en kjøring der vi øker den til 14 (se Tabell 10).

i	Q_{i-1}	K_i	q_i	$D(V - Q_{i-1})$	R_i
1	0.00	2.14	2.14	2.80	42.70
2	2.14	2.22	2.22	2.37	44.42
3	4.36	1.95	1.93	1.93	38.03
4	6.28	1.43	1.43	1.54	28.59
5	7.71	1.21	1.21	1.26	24.29
6	8.93	1.02	1.01	1.01	20.23
7	9.94	0.74	0.74	0.81	14.80
8	10.68	0.64	0.64	0.66	12.83

Tabell 10: Sanne verdiene for D og V er henholdsvis 0.2 og 14; kun data fra siste periode utslagsgivende

Som forventet får vi ganske høy akkumulert resultat på 225.89 millioner dollar, diskontert resultat i dette tilfellet blir på 200.20 millioner dollar. Her fikk vi faktisk lavere resultat i den første perioden sammenlignet med da den sanne verdien for V var lik forventningen, men betydelig høyere resultat i de påfølgende 7 tidsperioder.

5.1.2 Sensitivitet med hensyn på kvoter

I dette avsnittet skal vi se på hvorvidt manipuleringen av kvotene vil påvirke resultatene våre. Man kan tenke seg at det kan lønne seg å sette prosesskvoten litt *høyere* i starten. Dette vil naturlig nok påføre oss en viss økonomisk tap i begynnelsen av produksjonen siden vi ikke får oppfylt hele den bestilte kvoten i de aktuelle produksjonsperiodene. Så det er ingen optimal metode på kort sikt. Men det kan tenkes at en mulig fordel med denne fremgangsmåten er at vi får

mer informasjon om produktet DV , og følgelig kan resultatet bli bedre.

For å teste det kan vi manipulere K_i -verdiene kunstig ved å *skalere* de optimale verdiene for K_i ved å multiplisere disse med en viss skalar. For eksempel, vil vi i tilfellet beskrevet i forrige avsnitte kunne multiplisert K_1 med 1.1 og dermed har vi satt kvoten for den første produksjonsperiode litt høyere. Tilsvarende kan man velge en lavere kvote for en eller flere perioder av interesse ved å multiplisere denne eller disse med et tall mindre enn 1.

Denne fremgangsmåten er først og fremst interessant når vi skal operere med stokastiske verdier for variablene D og V (se neste seksjon). Dette fordi når vi antar sanne verdier for variablene D og V så kan vi faktisk beregne skaleringsmåten som gir oss best resultat nøyaktig. Derfor nøyer vi oss nå med å se på et par eksempler på mulig skalering.

La oss anta at vi velger følgende skaleringsvektor: (1.1, 0.8, 0.95, 0.96, 0.97, 0.98, 0.99, 1). Det vil si at kvoten i første perioden er satt litt høyere, og så en del lavere i andre periode, kvotene for periodene 3 til 7 er så vidt under de optimale verdiene, og kvoten for den siste perioden gjøres det ingen forandringer på. Vi kan observere resultatene i tabell 11.

i	Q_{i-1}	K_i	q_i	$D(V - Q_{i-1})$	R_i
1	0.00	2.35	2.35	2.40	47.07
2	2.35	1.82	1.82	1.93	36.44
3	4.18	1.75	1.56	1.56	25.72
4	5.74	1.17	1.17	1.25	23.45
5	6.91	0.94	0.94	1.02	18.88
6	7.86	0.78	0.78	0.83	15.56
7	8.63	0.65	0.65	0.67	13.03
8	9.29	0.55	0.54	0.54	10.65

Tabell 11: Eksempel på mulig skalering av kvoten. Skaleringsvektoren i dette eksempelet er (1.1, 0.8, 0.95, 0.96, 0.97, 0.98, 0.99, 1)

Denne kunstige manipuleringen gir oss tilnærmet lik resultat som uten skalering: 190.80 millioner USD (170.38 diskontert). Legg merke til en liten detalj: det akkumulerte resultatet er ca 0.15 mindre nå enn da vi ikke har skalering, mens det diskonterte totale resultatet er ca 0.63 høyere med skalering.

La oss eksperimentere litt med skaleringen og justere skaleringsvektoren til å bli (1.0, 0.9, 0.95, 0.95, 0.95, 0.95, 0.95). Resultatet blir da følgende (se tabell 12)

i	Q_{i-1}	K_i	q_i	$D(V - Q_{i-1})$	R_i
1	0.00	2.15	2.15	2.40	43.09
2	2.15	2.01	1.97	1.97	38.03
3	4.12	1.48	1.48	1.58	29.63
4	5.61	1.19	1.19	1.28	23.81
5	6.80	0.97	0.97	1.04	19.47
6	7.77	0.81	0.81	0.85	16.17
7	8.58	0.69	0.68	0.68	13.67
8	9.26	0.55	0.55	0.55	10.95

Tabell 12: Et annet eksempel på mulig skalering av kvoten. Her er skaleringsvektoren (1.0, 0.9, 0.95, 0.95, 0.95, 0.95, 0.95, 1.0)

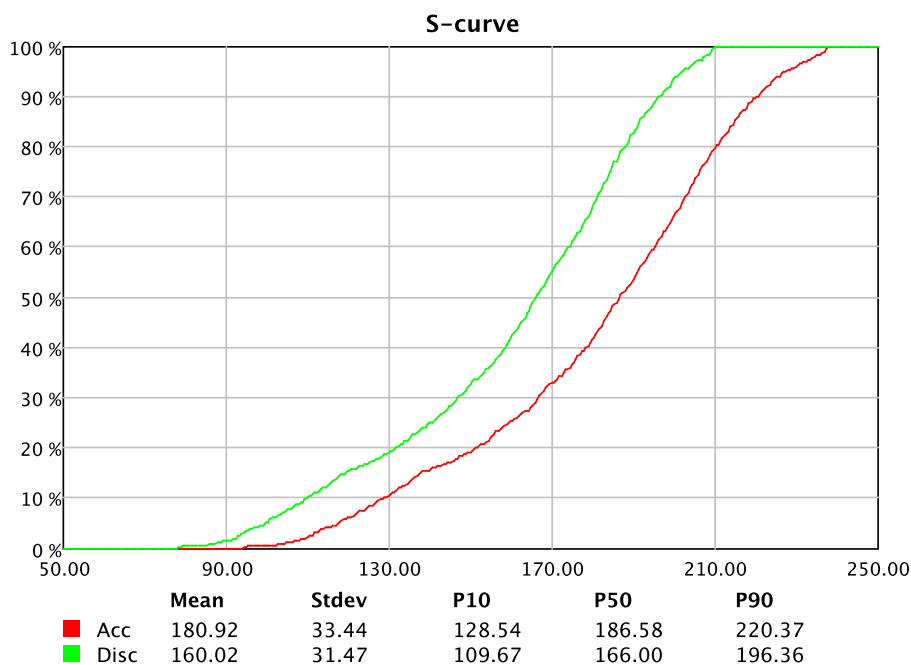
Etter denne kunstige manipuleringen blir det totale resultatet på 194.83 mill.dollar (diskontert beløp: 173.43 mill. dollar), og det gir faktisk en forbedring av resultatet.

Det finnes selvfølgelig veldig mange forskjellige skaleringsscenarier og det er begrenset hvor mange forskjellige muligheter vi får testet. I neste seksjon vil vi eksperimentere med å sette kvoten høyere/lavere i startfasen og undersøke hvorvidt det gir oss et bedre samlet resultat.

5.2 Resultater av simuleringer når D og V er stokastiske

Felles for samtlige tilfellene vi har sett på før er antagelsen at de sanne verdiene for variablene D og V er kjent, og vi så nærmere på hvordan de kumulative resultatene forandrer seg når de sanne verdiene varierer.

I denne seksjonen skal vi se på hva resultatene blir når D og V er stokastiske, som er en mer realistisk situasjon. I et virkelig tilfelle ville det alltid vært kun én sann V og én sann D. Imidlertid, selvom V og D er faste tall, så vil de være ukjente for beslutningstakeren som styrer valg av produksjonskvoter. Når vi gjorde beregninger med faste parameterverdier, så fikk vi demonstrert hvordan metodikken vår virket for ulike parameterverdier. Disse simuleringene viser dog ikke hvor stor usikkerheten i totalproduksjonen blir med våre metoder. Dette as-



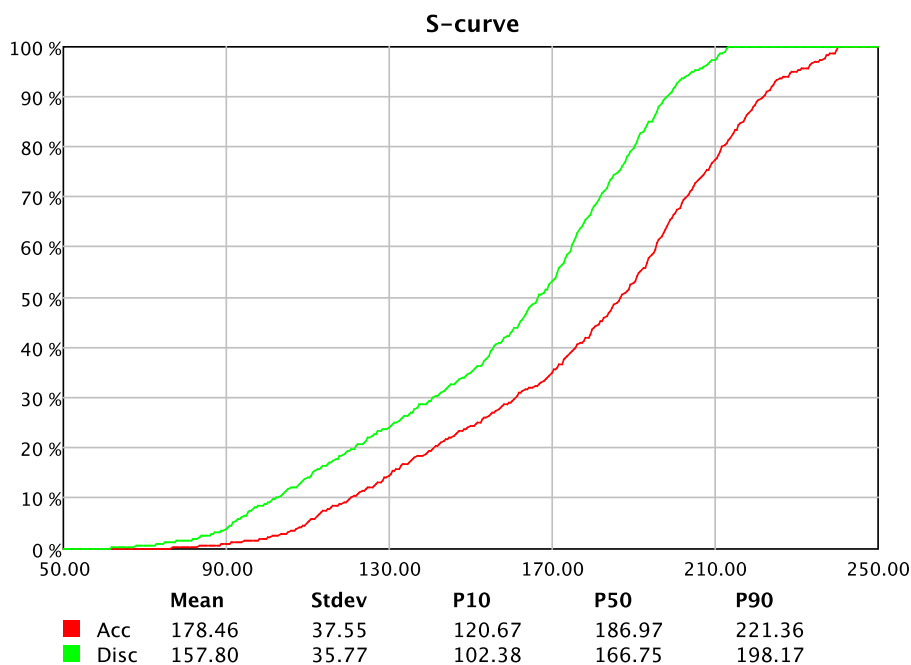
Figur 4: S-kurve for akkumulert og diskontert resultat for stokastiske D og V

pektet får vi først med når vi inkluderer usikkerheten.

Vi vil også se hvordan manipulering av kvotene (som vist i forrige seksjon) vil påvirke sluttresultatet. I hvert tilfelle skal vi kjøre 1000 simuleringer og studere utfallet. Vi bruker fremdeles de samme apriorifordelingene til variablene D og V, det vil si henholdsvis uniformfordelingen på intervallet 0.1 til 0.3 og lognormalfordelingen med forventning 12.0 og standardavvik 0.1.

Vi starter med å se på hvordan det akkumulerte og diskonterte resultatet blir (se figur 4), og det er av interesse å sammenligne med resultatene vi fikk da det ble brukt faste verdier for variablene D og V som var lik respektive forventningene. Vi observerer at gjennomsnittsverdien for det akkumulerte resultatet er 180.92 millioner dollar, som er lavere enn da de faste verdiene ble brukt (190.95 mill.dollar), mens medianen ligger på 186.58. Hvis vi tar med diskonteringen på 5 prosent, så får vi et gjennomsnittresultat på 160.02 mill.dollar (169.75 mill.dollar i et tilfelle med faste verdier). Medianen ligger på 166.00.

Vi noterer oss også standardavvik for det akkumulerte og diskonterte resultatene, som er henholdsvis 33.44 og 31.47.



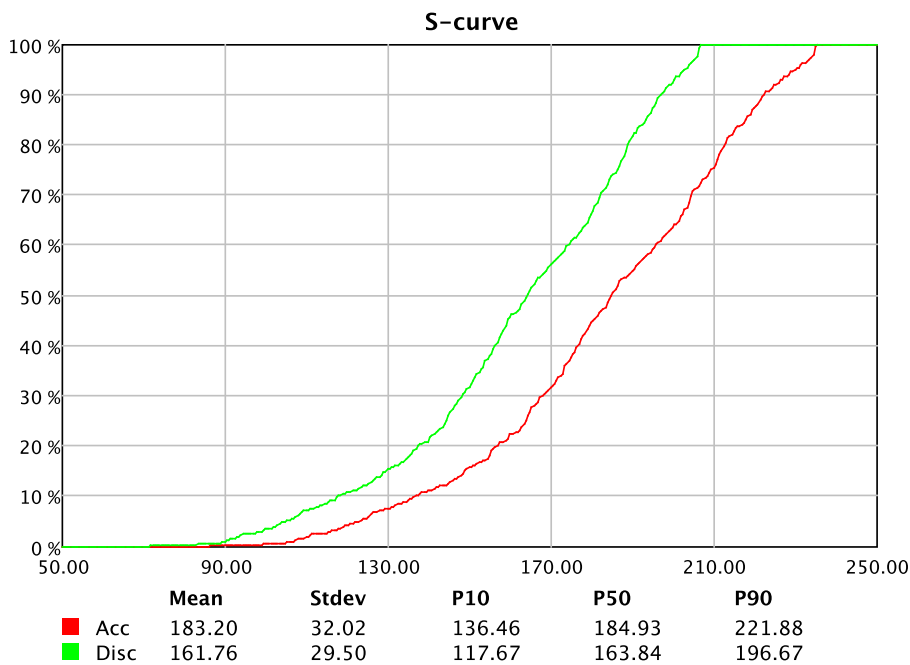
Figur 5: Økning av kvoten i første periode

5.2.1 Skalering av kvoten

Det neste steget vil være å foreta manipuleringen av kvoten som forklart i seksjon 5.1.2. Vi varierer skaleringsverdiene i begynnelsen av produksjonen og ser hvordan det vil påvirke den totale inntekten. Ved å skalere kvoten kunstig i begynnelsen kan man se om det for eksempel vil lønne seg å være litt dristig i starten (det vil si sette kvoten litt høyere, men samtidig få mer informasjon om produktet *DV* underveis) eller om det å være litt mer forsiktig i starten og nedjustere kvoten litt kan faktisk gi oss høyere inntjening totalt.

La oss se hvordan utviklingen blir når vi setter kvoten litt høyere helt i starten, det vil si at kvoten i produksjonsperiode 1 justeres litt opp (multipliseres med 1.1). Figur 5 viser resultatet, der den røde kurven er det akkumulerte resultatet og den grønne er det diskonterte resultatet:

Vi ser her at strategien å sette kvoten litt høyere i første periode ikke ble lønnsom hvis man ser på gjennomsnittsverdiene for det akkumulerte og det diskonterte resultatet - de ligger litt lavere enn i tilfellet over (henholdsvis 178.46 og 157.80 mill.dollar). Hvis vi ser på medianverdiene så er det omtrent like, mens standardavvikene er betydelig høyere nå: 37.55 for det akkumulerte resultatet og 35.77 for det diskonterte resultatet.



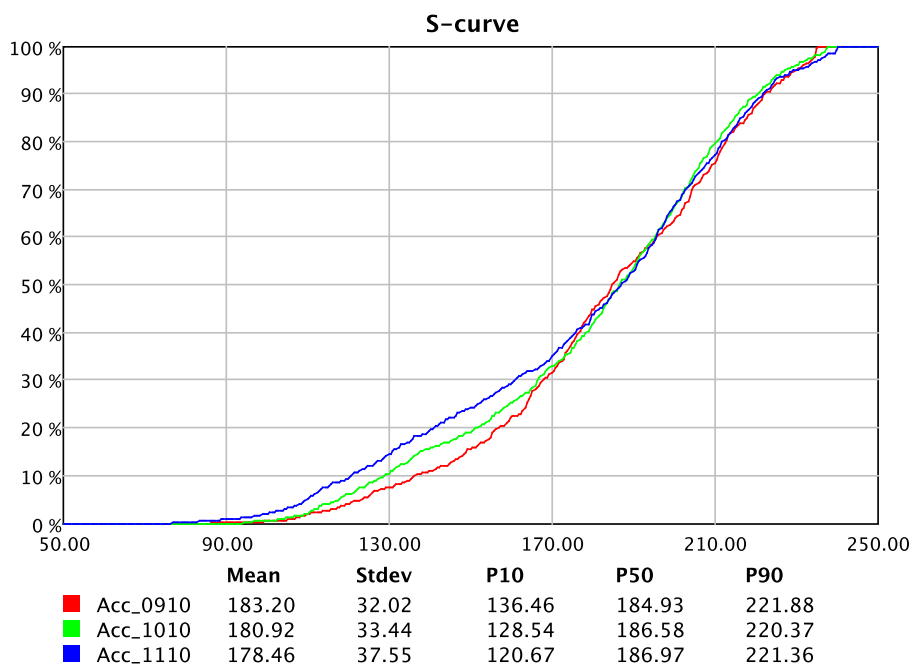
Figur 6: Nedskalering av kvoten i første periode

Så gjør vi et forsøk hvor kvoten i første perioden nedjusteres, det vil si at vi multipliserer den optimale verdien for K_1 med 0.9. Se figur 6:

Sammenlignet med tilfellet hvor vi ikke hadde noen form for skalering, så får vi her litt høyere gjennomsnittresultat (183.20 mill. dollar for det akkumulerte og 161.76 - for det diskonterte resultatet), mens medianverdiene er faktisk litt lavere. Samtidig reduseres usikkerheten litt - standardavvikene i dette tilfellet er henholdsvis 32.02 og 29.50.

Vi kan oppsummere de resultatene vi har kommet frem til nå grafisk: figur 7 viser oss 3 kurver for det akkumulerte resultatet, der den grønne kurven representerer resultatet *uten* skalering, den blå - der kvoten i første periode settes litt høyere, og til slutt den røde kurven, som viser oss resultatet når nedjusteringen av kvoten i periode 1 finner sted. Legg merke til at variasjonen i resultatene for disse tre strategiene blir størst i starten, men blir redusert betraktelig ettersom produksjonen fortsetter.

På en tilsvarende måte kan vi sammenligne de diskonterte resultatene. Igjen,



Figur 7: Sammenligning av de akkumulerte resultatene

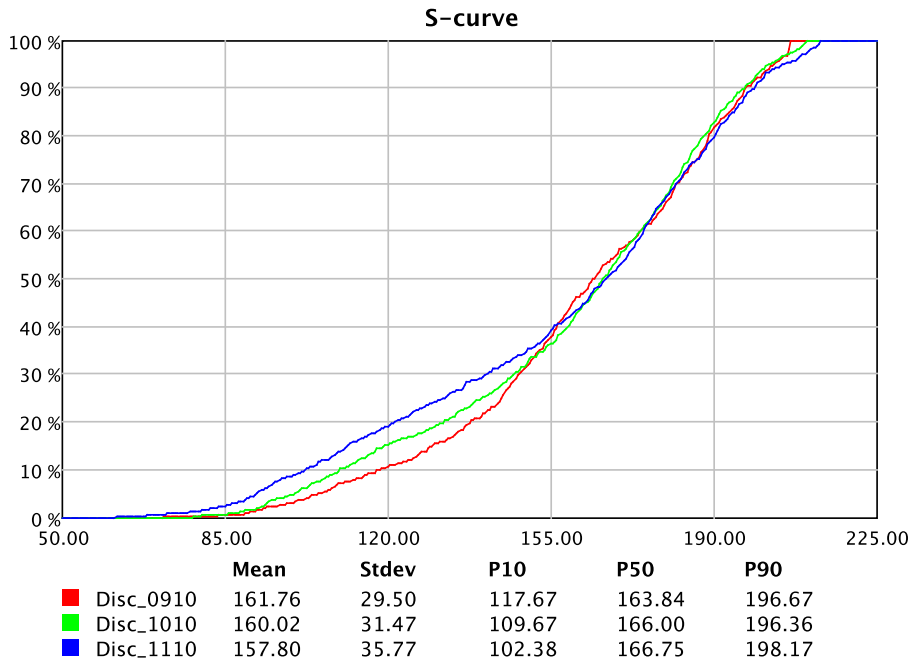
viser den grønne linjen det kumulative resultatet uten skalering, den blå - der kvoten justeres opp, og den røde viser resultatet der vi foretar en nedjustering. (Se figur 8)

La oss nå se hvordan utviklingen blir nå vi skalerer kvoten i de *to* første produksjonsperiodene, og om det blir vesentlig forskjell i de totale inntektene og ikke minst i usikkerheten rundt disse.

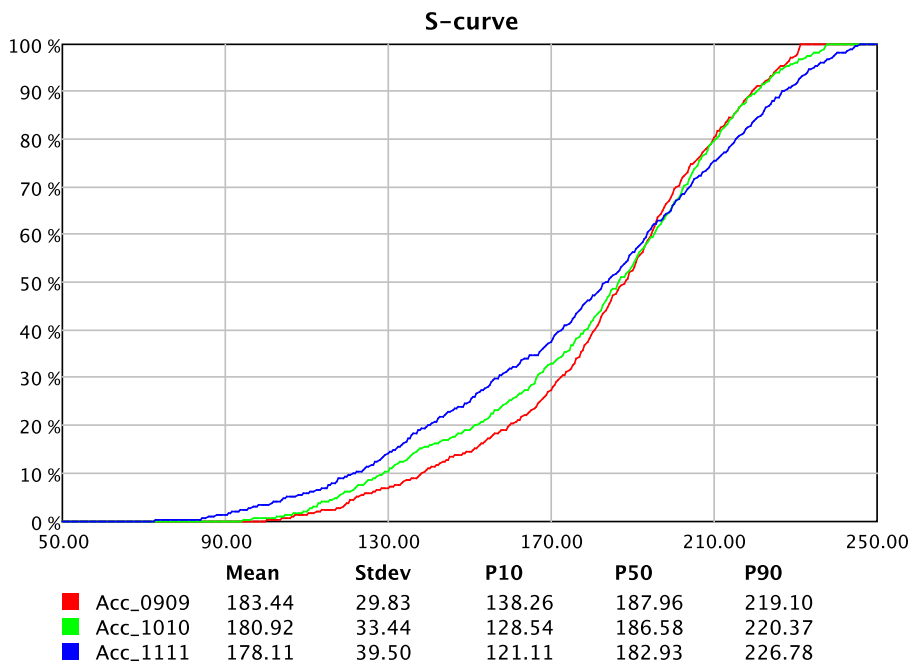
Vi skal først ta en nærmere titt på de 3 følgende scenarier: 1) der ingen skalering er tilstede 2) der kvotene for periodene 1 og 2 justeres *opp* og 3) der kvotene for periodene 1 og 2 justeres *ned*.

Igjen, vil vi først se på de totale resultatene uten å ta hensyn til diskontering, og så de diskonterte resultatene. Figur 9 viser oss 3 kurver for de akkumulerte resultatene.

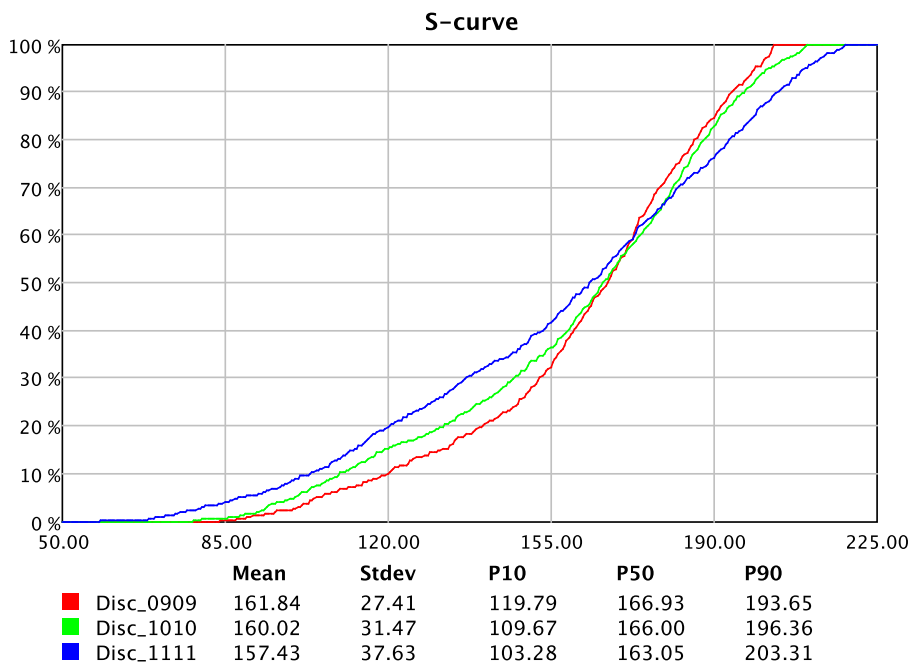
Også her ser vi at den mest effektive strategien er å være litt forsiktig i begynnelsen. Det er ikke store forskjeller i gjennomsnittsresultatene her sammenlignet



Figur 8: Sammenligning av de diskonterte resultatene



Figur 9: Sammenligning av de akkumulerte resultatene, forts.



Figur 10: Sammenligning av de diskonterte resultatene, forts.

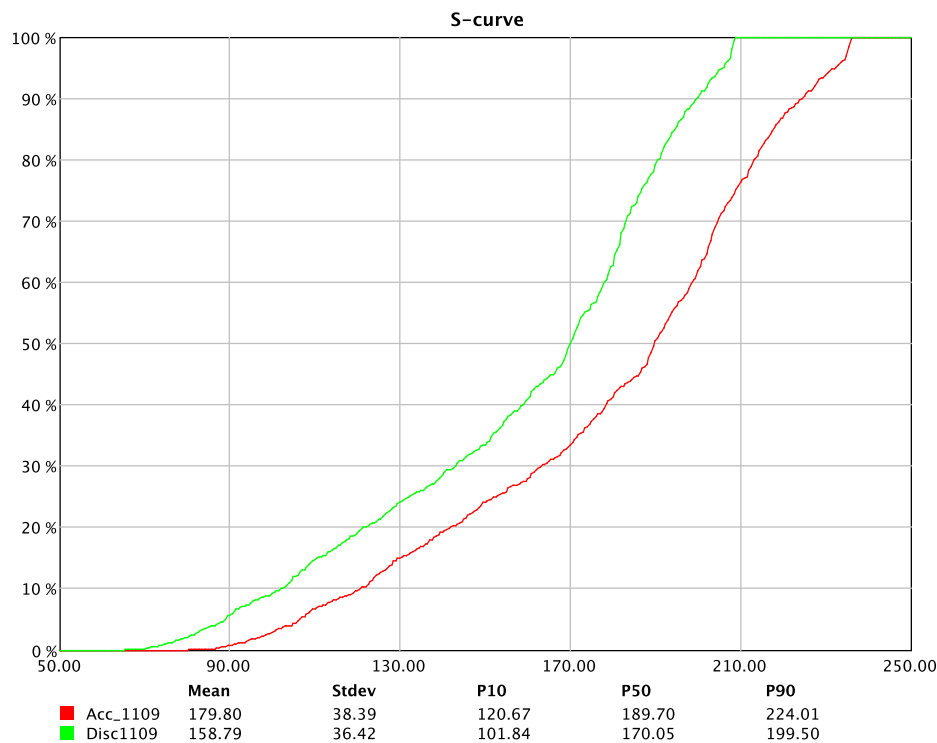
med situasjonen der vi kun skalerte den første kvoten. Men hvis vi ser på standardavvikene, så er den faktisk enda mer redusert nå ved bruk av den forsiktige strategien (29.83) mens den økte til 39.50 ved den 'dristige' strategien.

Vi observerer også en vesentlig forskjell fra det forrige tilfellet hvor kvoten i kun den første perioden ble justert: ved skaleringen av kvoten i første periode var det en viss variasjon i resultatene i begynnelsen men som jevnet seg ut etterhvert slik at det nesten ikke var signifikant forskjell mellom de tre strategiene. Mens når vi justerer kvotene i *to* første perioder, så er det faktisk den dristige strategien som ser ut til å gi bedre resultater i slutfasen av produksjonen (figur 9).

Figur 10 viser de diskonterte resultatene ved de tre scenarier.

Vi observerer den samme trenden, nemlig ingen betydelige forskjeller i det gjennomsnittlige sluttresultatet, men økende standardavvik ved å skalere kvoten litt høyere i begge perioder enn kun i den første (37.63) og redusert standardavvik ved å skalere kvote litt ned i begge periodene enn kun i den første (27.41).

Det finnes mange forskjellige muligheter når det gjelder skalering av kvoten. Vi velger å konsentrere oss om situasjoner hvor kun kvotene helt i starten av produksjonen blir justert. Enda et tilfelle som vi ennå ikke har sett på er å justere kvoten litt opp i første perioden og deretter justere den ned i den andre, og



Figur 11: Akkumulerte resultater

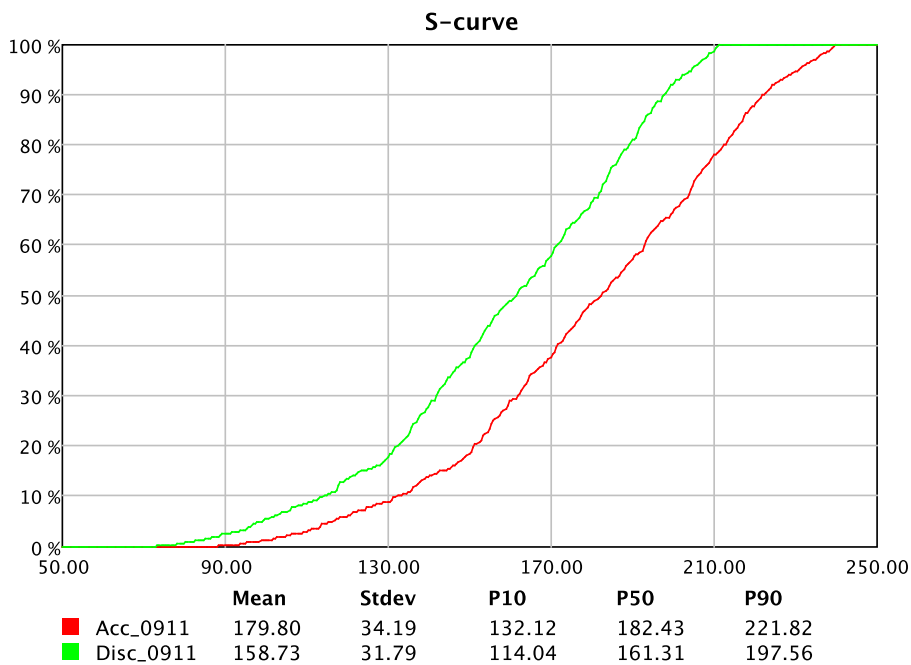
motsatt.

Vi kan se på det akkumulerte og det diskonterte resultatet når kvoten settes litt opp i periode 1 og skaleres ned i periode 2 (se figur 11)

Sammenlignet med situasjonen hvor vi ikke foretar noen form for skalering gir dette scenariet signifikant høyere usikkerhet og litt lavere gjennomsnittsverdier. Medianverdiene derimot er litt høyere i dette tilfellet.

Nå bytter vi om og velger i stedet å sette kvoten litt lavere i første perioden og litt høyere i den andre (se figur 12). Også her er den ingen veldig store forandringer i selve resultatverdiene sammenlignet med tilfellet uten skalering, men medianverdiene er et stykke lavere enn i situasjonen beskrevet over, og det er også standardavviket.

La oss nå foreta en oppsummering av resultatstudier for de forskjellige skaleringsscenarioer. Tabellen nedenfor viser det akkumulerte resultatet i de ni situasjonene vi har sett på (der de to første kolonnene angir type skalering):



Figur 12: Diskonterte resultater

Periode 1	Periode 2	Snitt	Std.avvik	P10	P50	P90
Ned	Ned	183.44	29.83	138.26	187.96	219.10
Ned	Ingen	183.20	32.02	136.46	184.93	221.88
Ned	Opp	179.80	34.19	132.12	182.43	221.82
Ingen	Ned	181.71	34.51	127.05	189.92	221.15
Ingen	Ingen	180.92	33.44	128.54	186.58	220.37
Ingen	Opp	178.60	37.15	122.89	181.77	223.77
Opp	Ned	179.80	38.39	120.67	189.70	224.01
Opp	Ingen	178.46	37.55	120.67	186.97	221.36
Opp	Opp	178.11	39.50	121.11	182.93	226.78

Og en tilsvarende oppsummering av de diskonterte resultatene:

Periode 1	Periode 2	Snitt	Std.avvik	P10	P50	P90
Ned	Ned	161.84	27.41	119.79	166.93	193.65
Ned	Ingen	161.76	29.50	117.67	163.84	196.67
Ned	Opp	158.73	31.79	114.04	161.31	197.56
Ingen	Ned	160.44	32.31	108.23	168.99	196.16
Ingen	Ingen	160.02	31.47	109.67	166.00	196.36
Ingen	Opp	157.79	35.03	105.39	161.05	200.05
Opp	Ned	158.79	36.42	101.84	170.05	199.50
Opp	Ingen	157.80	35.77	102.38	166.75	198.17
Opp	Opp	157.43	37.63	103.28	163.05	203.31

Videre kan det være interessant å se hvordan utviklingen i variablene K_i (kvote i periode i), q_i (den utvunnede mengden i periode i) og R_i (resultat før diskontering i periode i) blir ved bruk av de forskjellige skaleringsstrategiene over tid.

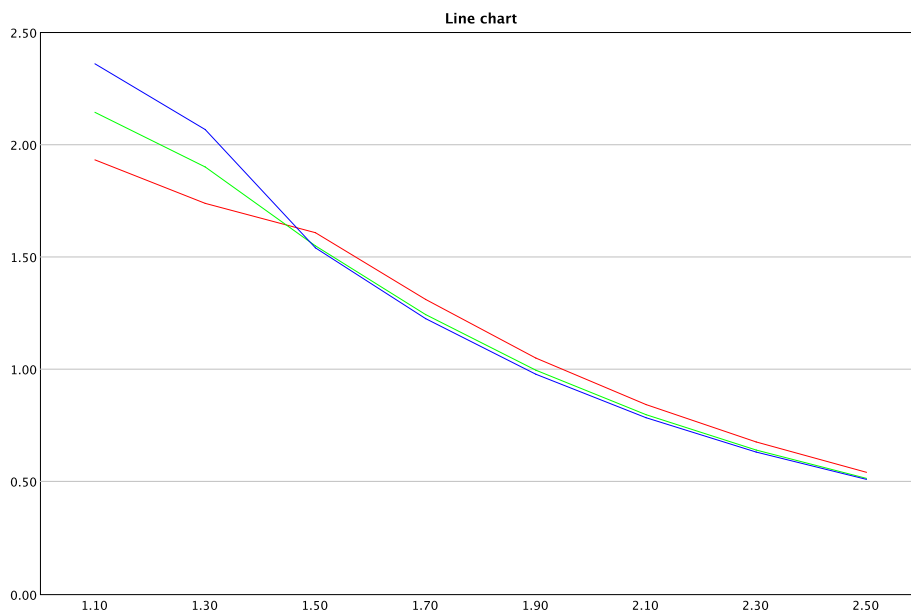
Vi skiller mellom den *beskjedne* (der vi justerer kvoten *ned* i de første 2 produksjonsperiodene), den *normale* (der vi ikke foretar noe form forskalering) og den *dristige* (der vi justerer kvoten *opp* i de 2 første produksjonsperiodene) strategier.

Figur 13 viser oss utviklingen i kvotene, der den røde linjen representerer den forsiktige strategien, den grønne - den normale, og den blå - den dristige. Ingen overraskelse når det gjelder starten - naturlig nok høyeste kvoter ved den *dristige* strategien, laveste ved den *forsiktige* - men etterhvert ser man at den forsiktige strategien gir oss litt høyere kvoter mens det er så å si ingen forskjell på kvotene ved den dristige og den uskalerte strategien.

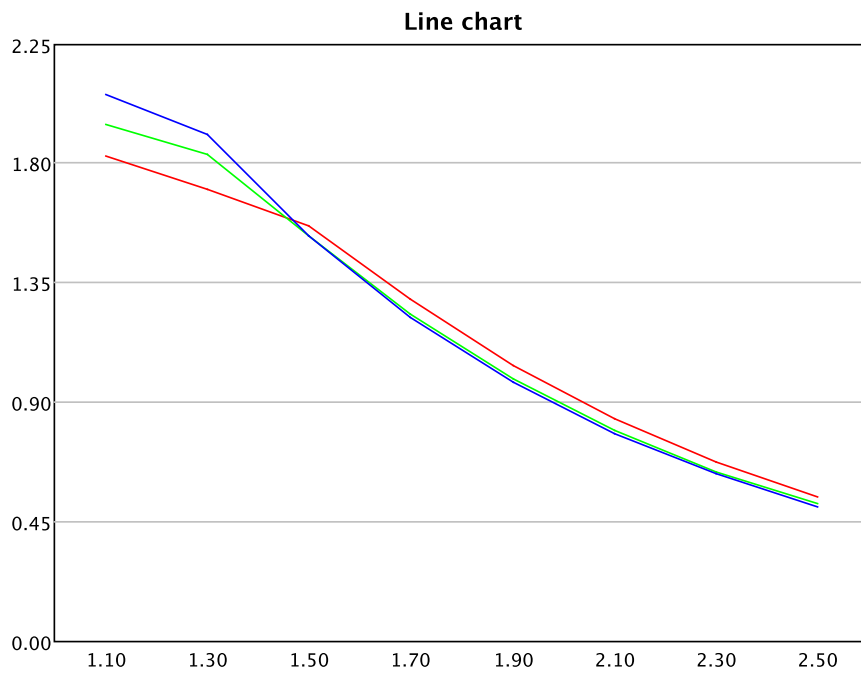
I figur 14 ser vi hvordan den reelle produksjonsraten q varierer over tid hvor, igjen, den røde linjen representerer den forsiktige strategien, den grønne - den normale, og den blå står for den dristige. Vi ser samme mønster som i figur 13.

Figur 15 viser utviklingen i resultatene som er mest interessante for oss som er beslutningstakere.. Legg særlig merke til den røde linjen som representerer den forsiktige strategien (som ga oss de beste resultatene) - i det lange løp får vi kompensasjon for det tapte i begynnelsen. Igjen er det tilsynelatende lite forskjell mellom den normale og den dristige strategien i mellom- og sluttfasene av produksjonen, mens den normale kommer bedre helt i begynnelsen.

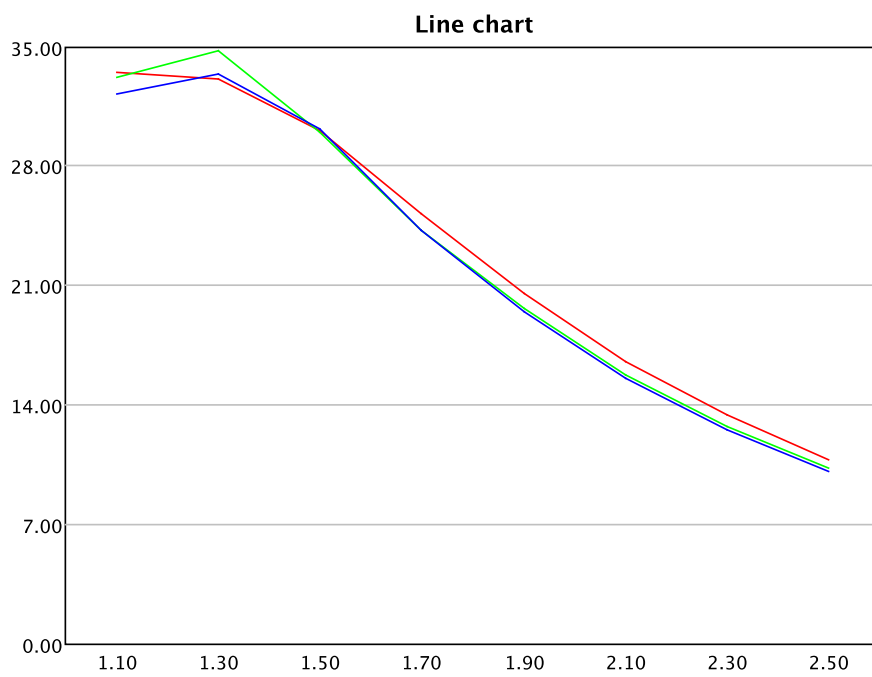
Til slutt er kan det være interessant å studere korrelasjonen mellom det akkumulerte/diskonterte resultatet og nøkkelvariablene våre D og V , og se hvordan



Figur 13: Utviklingen i gjennomsnittlige K -verdiene over tid ved bruk av diverse skaleringsmåter. Rød linje - *forsiktig* strategi, grønn linje - *normal* strategi, blå linje - *dristig* strategi



Figur 14: Utviklingen i gjennomsnittlige q -verdier over tid ved bruk av diverse skaleringsmåter. Rød linje - *forsiktig* strategi, grønn linje - *normal* strategi, blå linje - *dristig* strategi



Figur 15: Utviklingen i gjennomsnittresultatet over tid ved bruk av diverse skaleringsmåter. Rød linje - *forsiktig* strategi, grønn linje - *normal* strategi, blå linje - *dristig* strategi

den eventuelt forandrer seg når vi tar i bruk de 3 skaleringsstrategiene nevnt over (se tabellene 13-15).

	Akkumulert	Diskontert
D	0.793	0.809
V	0.458	0.422

Tabell 13: Korrelasjonstabell, *forsiktig* strategi

	Akkumulert	Diskontert
D	0.831	0.847
V	0.427	0.392

Tabell 14: Korrelasjonstabell, *ingen* skalering

	Akkumulert	Diskontert
D	0.853	0.869
V	0.470	0.439

Tabell 15: Korrelasjonstabell, *dristig* strategi

Vi ser først og fremst at i samtlige situasjoner har vi høyere korrelasjon mellom det totale resultatet og decline-raten D enn mellom det totale resultatet og volumet V . Dette skyldes mer usikkerhet i D enn i V , og sånn sett er det ikke sammenlignbart på likt grunnlag. Vi observerer også at korrelasjonen mellom resultatet og decline-raten er høyest der vi skalerte kvoten *opp* i de to første produksjonsperiodene og lavest ved forsiktig strategi, der vi skalerte kvoten *ned* i de to første periodene. Dette skyldes økt usikkerhet ved bruk av den dristige strategien.

Vi legger merke til at i samtlige 3 tilfeller er det større korrelasjon mellom D og det diskonterte resultatet enn mellom D og det akkumulerte resultatet, mens det omvendte gjelder når det gjelder korrelasjonskoeffisientene mellom resultatene og volumet V . En mulig forklaring på dette kan være at hvis vi ser på den potensielle produksjonsraten vår, $D(V - Q)$, så betyr den relative usikkerheten i D mest når $(V - Q)$ er stor (det vil si tidlig fase i produksjonen), mens den

relative usikkerheten i V betyr mest når den kumulative produksjonsvolumet Q blir stor (og Q vokser ettersom produksjonen pågår).

6 Konklusjon

I denne oppgaven presenterte vi en metode for hvordan forskjellige simuleringsalgoritmer kan implementeres for å håndtere diverse situasjoner i forbindelse med sekvensiell optimering av oljeproduksjon under usikkerhet i diskret tid. Eksempelene i oppgaven viser oss hvordan samlingen av informasjonen underveis påvirker valget av de optimale kvotaene og hvordan variasjonen i usikkerhetsparametre påvirker det totale sluttresultatet.

I seksjonen hvor vi antok at de sanne verdiene er kjent så vi hvordan de totale resultatene blir når de sanne verdiene for decline-raten og volumet er lik respektive forventningsverdiene. Vi har også sett på situasjoner der de sanne verdiene økes/redueres i forhold til forventningverdiene, og i hvor stor grad disse økningene/reduksjonene påvirker valget av kvotene og det samlede resultatet.

Vi så også på tilfeller hvor vi både samler all informasjon fra alle tidligere produksjonsperioder til å optimere kvotene, og på tilfeller der kun data fra siste periode er utslagsgivende. Vi fikk bedre resultater når all informasjon fra tidligere periodene ble brukt i simuleringene.

Videre har vi foretatt kjøringene uten faste parameterverdier (det vil si der kvotene ble optimert simultant) der vi blant annet kunne studere effekten av usikkerheten på den totale produksjonen og sammenligne med resultatene vi fikk da vi opererte med faste verdier.

Vi så også på tilfeller hvor vi studerte utviklingen i kvoten i det lange løp der vi eksperimenterte med ulike måter å justere kvotene i starten av produksjonen på. Man skulle tro at ved å velge en litt dristigere strategi i begynnelsen vil skaffe oss mer informasjon om nøkkelvariablene D og V , og på denne måten redusere usikkerheten i det lange løp og gi oss bedre sluttresultat (selv om det kan påføre et viss økonomisk tap i begynnelsen). Det viste seg derimot at det er den forsiktede strategien der kvotene i de første periodene ble *nedjustert* som faktisk ga oss de beste akkumulerte resultater etter 8 produksjonsperioder - det som ble tapt i starten ble kompensert senere i produksjonen.

Skaleringsmetoden vi brukte i denne oppgaven gikk ut på å bestemme en viss justeringsvektor i forveien av produksjonen. En alternativ fremgangsmåte som ikke ble sett nærmere på i denne oppgaven kunne vært å bruke dynamisk pro-

grammering ved skalering av kvotene, slik at oppdateringen skjer underveis. Det er en interessant fremgangsmåte som man eventuelt kunne komme tilbake til i senere arbeid.

Referanser

- Arps, J. J. 1945. Analysis of Decline Curves. *Trans. AIME*, 160, 228–247.
- Haavardsson, N. F. & Huseby, A. B. 2007. Multisegment production profile models, a tool for enhanced total value chain analysis. *Journal of Petroleum Science and Engineering*, 58, 325–338.
- Huseby, A. B. & Haavardsson, N. F. 2009. Multi-reservoir production optimization. *European Journal of Operational Research*, 199, 236–251.
- Hastings, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57, 97–109.
- Huseby, A. B. & Haavardsson, N. F. 2010. Multi-reservoir production optimization under uncertainty. In *Reliability, Risk and Safety. Theory and Applications*, CRC Press, 407–413.
- Horne, R. N. 2002. *Optimization applications in oil and gas recovery*. Handbook of Applied Optimization, Oxford University Press.
- Merabet, N. & Bellah, S. 2002. Optimization techniques in mature oilfield development. Presented at The Sixth Annual U.A.E. University Research Conference. United Arab Emirates.
- Neiro, S. M. S. & Pinto, J. M. 2004 A general modelling framework for the operational planning of petroleum supply chains. *Computers & Chemical Engineering*, 28, 871–896.

7 Vedlegg: javakode

/*

MCMCMain

@author nikitam
@version \$Revision: 1.1 \$
@package
@project nmmcmc

Description: Some description of the class

Copyright (C) 2010 by University of Oslo. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER
IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF
SUCH DAMAGE.

```
import java.io.*;
import java.text.DecimalFormat;
import java.text.NumberFormat;

import no.uio.mcmc.*;
import no.uio.mcmc.impl.*;
import no.uio.stochutils.*;
import no.uio.stochutils.distrib.*;

import com.riscue.relutils.*;

public class MCMCMain
{

    public static final String TAB = " & ";
    //public static final String TAB = "\t";
    public static final String RET = " \\\\";
    //public static final String RET = "";
    public static final String HLINE = "\\hline";
    //public static final String HLINE =
        "_____";

    public static final boolean LaTeX_Table = true;

    public static final boolean
        INCLUDE_RESTRICTIONS_IN_OUTPUT = false;
    public static final boolean
        INCLUDE_ACCUM_RESULTS_IN_OUTPUT = false;
    public static final boolean
```



```

INCLUDE_DISC_RESULTS_IN_OUTPUT = false;

public static final int USE_ALL_DATA = 1;
public static final int USE_LAST_DATA = 2;

public static int DECISION_MODE = USE_ALL_DATA;
//public static int DECISION_MODE = USE_LAST_DATA;

public static int ACC_MODE = ProductionConstraints.
    USE_ALL_CONSTRAINTS;
//public static int ACC_MODE = ProductionConstraints.
    USE_LAST_CONSTRAINT;

public static final boolean USE_FIXED_VALUES = true;
public static final double TRUE_D = 0.2;
public static final double TRUE_V = 14.0; // million
    barrels

public static final double MIN_D = 0.1;
public static final double MAX_D = 0.3;
public static final double E_V = 12.0;
public static final double SD_V = 1.0;
public static final double DISC_RATE = 0.05;

public static final double d = 50.0; //
    Income rate usd/barrel
public static final double c = 30.0;
    // Cost rate usd/barrel
// public static final double q = 0.0;
    // Previous production rate
// public static final double Q = 2.0;
    // Previous cumulative production
public static final double K_max = 10.0;

public static final double[] K_scale = {0.9, 0.9, 0.9,
    1.0, 1.0, 1.0, 1.0, 1.0};

public static final int Scale_periods = 0;

public static final int NUM_OUTER_SIMULATIONS = 1;

```

```

public static final int NUM_ITER = 1000;
public static final int NUM_SIM = 5000;
public static final int CURVE_SIZE = 1001;
public static final int NUM_PHASES = 8; // antall
    perioder

public static final String ZXT_FILE_NAME = "
    RejectionSampling";
public static final String RXF_FILE_NAME = "
    optimize_K2";
public static PrintWriter PW = null;
public static NumberFormat DEC_FORMAT = new
    DecimalFormat("#0.00");

/**
 * @param args
 */
public static void main(String [] args)
{
    RGen.INIT_RANDOM_SEED();
    ProductionConstraints.SET_ACCEPT_MODE(ACC_MODE
        );

    runOptimizationSequence();
}

public static void runOptimizationSequence()
{
    Formula dFormula = new UNIFORMFormula(MIN_D,
        MAX_D);
    Formula vFormula = new LOGNORMALFormula(E_V,
        SD_V);

    if (!LaTeX_Table)
    {
        createOutputTxtFile();

        System.out.print("D\tV\t");
        PW.print("D\tV\t");

        for (int i = 0; i < NUM_PHASES; i++)
        {
            int j = i+1;

            System.out.print("K" + j + "\t" + "q" + j + "\t"

```

```

        t" + "f" + j + "\t" + "R" + j + "\t");
    PW.print("K" + j + "\t" + "q" + j + "\t" + "f"
        + j + "\t" + "R" + j + "\t");
}

System.out.println("Acc\tDisc");
PW.println("Acc\tDisc");
}

```

```

for(int sim = 0; sim < NUM_OUTER_SIMULATIONS;
    sim++)
{
    double trueD, trueV;

    int numGT = 0;
    int numEQ = 0;

    double K_opt, curr_q, max_q, curr_Q, result,
        accres;

    double disc_r = 0.0;
    double disc_f = 1.0;

    // Initialize
    if(USE_FIXED_VALUES)
    {
        trueD = TRUE_D;
        trueV = TRUE_V;
    }
    else
    {
        trueD = dFormula.getFormulaValue();
        trueV = vFormula.getFormulaValue();
    }

    curr_q = 0.0;
    curr_Q = 0.0;
    accres = 0.0;

    ProductionEquality pEQ1 = null;
    ProductionEquality pEQ2 = null;
}

```

```

ProductionConstraints pGT = new
    ProductionConstraints(NUM_PHASES);

if (LaTeX_Table)
{
    System.out.println("trueD\t" + trueD
        );
    System.out.println("trueV\t" + trueV
        );

    System.out.println(HLINE);
    System.out.print("$i$");
    System.out.print(TAB + "$Q_{i-1}$");
    System.out.print(TAB + "$K_i$");
    System.out.print(TAB + "$q_i$");
    System.out.print(TAB + "$f(Q_{i-1})$
        ");

    if (INCLUDE_RESTRICTIONS_IN_OUTPUT)
    {
        System.out.print(TAB + "
            Constr.");
    }

    System.out.print(TAB + "$R_i$");

    if (INCLUDE_ACCUM_RESULTS_IN_OUTPUT)
    {
        System.out.print(TAB + "Acc.
            r");
    }

    if (INCLUDE_DISC_RESULTS_IN_OUTPUT)
    {
        System.out.print(TAB + "Disc
            .r");
    }

    System.out.println(RET);
    System.out.println(HLINE);
}

```

```

else
{
    System.out.print(" " + DEC_FORMAT.
        format(trueD) + "\t" + DEC_FORMAT
            .format(trueV) + "\t");
PW.print(" " + DEC_FORMAT.format(trueD) + "\t" +
    DEC_FORMAT.format(trueV) + "\t");
}

for (int i = 0; i < NUM_PHASES; i++)
{
    K_opt = optimizeK(pGT, pEQ1, pEQ2,
        curr_q, curr_Q);

    if (i < Scale_periods)
    {
        K_opt *= K_scale[i];
    }

    curr_Q += curr_q;

    max_q = trueD * (trueV - curr_Q);
    curr_q = Math.min(K_opt, max_q);
    result = d * curr_q - c * K_opt;
    accres += result;
    disc_r += result / disc_f;
    disc_f *= (1 + DISC_RATE);

    if (curr_q < max_q)
    {
        pGT.addConstraint(curr_q,
            curr_Q);
        numGT = pGT.size();

        if (DECISION_MODE ==
            USE_LAST_DATA)
        {
            pEQ1 = null;
            pEQ2 = null;
            numEQ = 0;
            numGT = 1;
        }
    }
    else if (pEQ1 == null)
    {

```

```

pEQ1 = new
    ProductionEquality(curr_q
        , curr_Q);
numEQ = 1;

if (DECISION_MODE ==
    USE_LAST_DATA)
{
    pGT.
        removeAllConstraints
            ();
    numGT = 0;
}
}
else if (pEQ2 == null)
{
    pEQ2 = new
        ProductionEquality(curr_q
            , curr_Q);
    numEQ = 2;

    if (DECISION_MODE ==
        USE_LAST_DATA)
    {
        pGT.
            removeAllConstraints
                ();
        pEQ1 = pEQ2;
        pEQ2 = null;
        numEQ = 1;
        numGT = 0;
    }
}

if (LaTeX_Table)
{
    System.out.print(" " +(i+1));
    System.out.print(TAB +
        DEC_FORMAT.format(curr_Q)
        );
    System.out.print(TAB +
        DEC_FORMAT.format(K_opt))
        ;
    System.out.print(TAB +
        DEC_FORMAT.format(curr_q)

```

```

    );
System.out.print(TAB +
    DEC_FORMAT.format(max_q))
    ;

if (
    INCLUDE_RESTRICTIONS_IN_OUTPUT
    )
{
    System.out.print(TAB
        + "[" + numGT +
        ", " + numEQ +
        "]"");
}

System.out.print(TAB +
    DEC_FORMAT.format(result)
    );

if (
    INCLUDE_ACCUM_RESULTS_IN_OUTPUT
    )
{
    System.out.print(TAB
        + DEC_FORMAT.
        format(accum));
}

if (
    INCLUDE_DISC_RESULTS_IN_OUTPUT
    )
{
    System.out.print(TAB
        + DEC_FORMAT.
        format(disc_r));
}

System.out.println(RET);
}
else
{
    System.out.print(DEC_FORMAT.
        format(K_opt) + "\t");
System.out.print(DEC_FORMAT.format(curr_q) +
    "\t");
}

```

```

        System.out.print(DEC_FORMAT.format(max_q) +
            "\t");
        System.out.print(DEC_FORMAT.format(result) +
            "\t");

        PW.print(DEC_FORMAT.format(K_opt) + "\t");
        PW.print(DEC_FORMAT.format(curr_q) + "\t");
        PW.print(DEC_FORMAT.format(max_q) + "\t");
        PW.print(DEC_FORMAT.format(result) + "\t");

    }

}

if (LaTeX_Table)
{
    System.out.println(HLINE);

    System.out.print("Accum. result: " +
        DEC_FORMAT.format(accres) + " ");
    System.out.print("Disc. result: " +
        DEC_FORMAT.format(disc_r));
    System.out.println();

}
else
{
    System.out.print(DEC_FORMAT.format(
        accres) + "\t");
    System.out.println(DEC_FORMAT.format(disc_r));

    PW.print(DEC_FORMAT.format(accres) + "\t");
    PW.println(DEC_FORMAT.format(disc_r));
}

}

if (!LaTeX_Table)
{
    closeOutputTxtFile();
}
}

public static double optimizeK(ProductionConstraints
    pGT, ProductionEquality pEQ1, ProductionEquality

```



```

    pEQ2, double q, double Q)
{
    if (pEQ1 == null)
    {
        return optimizeK_GT(pGT, q, Q);
    }
    else if (pEQ2 == null)
    {
        return optimizeK_EQ(pEQ1, pGT, q, Q);
    }
    else
    {
        double q1 = pEQ1.get_q();
        double q2 = pEQ2.get_q();
        double Q1 = pEQ1.get_Q();
        double Q2 = pEQ2.get_Q();

        /**
         * Solve D and V using the two
         * equations:
         *
         * (1)  $D(V - Q1) = q1$ 
         * (2)  $D(V - Q2) = q2$ 
         *
         * This implies that:
         *
         * (3)  $D = (q1 - q2) / (Q2 - Q1)$ 
         * (4)  $V = Q1 + q1/D$ 
         *
         * Given the true values of D and V,
         * the optimal K is:
         *
         * (5)  $D * (V - Q - q)$ 
         */

        double D = (q1 - q2) / (Q2 - Q1);
        double V = Q1 + q1/D;

        return D * (V - Q - q);
    }
}

```

```

public static double optimizeK_EQ(ProductionEquality

```

```

    pEQ, ProductionConstraints pGT, double q, double Q)
{
    Density distD = new UNIFORMDensity(MIN_D,
        MAX_D);
    Density distV = new LOGNORMALDensity(E_V, SD_V
        );

    ConditionalDensity cd = new ConditionalDensity
        (distD, distV, pEQ);
    Proposal pr = new LOGNORMALProposal(0.3);

    MetropolisHastingsGenerator mhg = new
        MetropolisHastingsGenerator(cd, pr);
    TimeCurve netResultCurve = new TimeCurve("Net
        result", K_max, CURVE_SIZE);
    TimeCurve netDerivativeCurve = new TimeCurve("
        Net derivative", K_max, CURVE_SIZE);

    double D = 0.0;
    double V = 0.0;
    boolean accept = false;

    double D_Sum = 0.0;
    double V_Sum = 0.0;
    double M_Sum = 0.0;
    double A_Sum = 0.0;

    for (int i = 0; i < NUM_SIM; i++)
    {
        accept = false;
        int kk = 0;

        while (!accept && kk < NUM_ITER)
        {
            D = mhg.generateValue(0.25);
            V = pEQ.condition(D);

            accept = (mhg.getUpdates() >
                0) && pGT.accept(D, V);
            kk++;
        }

        double maxProdRate = D * (V - Q - q);

        A_Sum += (accept ? 1.0 : 0.0);
    }
}

```

```

if (accept)
{
    D_Sum += D;
    V_Sum += V;

    M_Sum += maxProdRate;

    for (int k = 0; k < CURVE_SIZE
        ; k++)
    {
        double K = K_max * k /
            CURVE_SIZE;

        double result = d *
            Math.min(K,
                maxProdRate) - c *
            K;
        netResultCurve.
            addValue(result , k)
            ;

        double deriv = (K <=
            maxProdRate ? (d -
            c) : -c);
        netDerivativeCurve.
            addValue(deriv , k);
    }
}

if (A_Sum > 0.0)
{
    // System.out.println(”” +
    netResultCurve.getMaxValTime());

    try
    {
        return netDerivativeCurve.
            getFirstRootTime();
    }
    catch (RelutilsException rex)
    {
        return -99.99; // ERROR
    }
}

```

```

    }
    else
    {
        return -99.99; // ERROR
    }
}

public static double optimizeK_GT(
    ProductionConstraints pGT, double q, double Q)
{
    Formula distD = new UNIFORMFormula(MIN_D,
        MAX_D);
    Formula distV = new LOGNORMALFormula(E_V, SD_V
    );
    TimeCurve netResultCurve = new TimeCurve("Net
        result", K_max, CURVE_SIZE);
    TimeCurve netDerivativeCurve = new TimeCurve("
        Net derivative", K_max, CURVE_SIZE);

    double D = 0.0;
    double V = 0.0;
    boolean accept = false;

    double D_Sum = 0.0;
    double V_Sum = 0.0;
    double M_Sum = 0.0;
    double A_Sum = 0.0;

    for (int i = 0; i < NUM_SIM; i++)
    {
        accept = false;
        int kk = 0;

        while (!accept && kk < NUM_ITER)
        {
            D = distD.getFormulaValue();
            V = distV.getFormulaValue();

            accept = pGT.accept(D, V);
            kk++;
        }

        double maxProdRate = D * (V - Q - q);

        A_Sum += (accept ? 1.0 : 0.0);
    }
}

```

```

if (accept)
{
    D_Sum += D;
    V_Sum += V;

    M_Sum += maxProdRate;

    for (int k = 0; k < CURVE_SIZE
        ; k++)
    {
        double K = K_max * k /
            CURVE_SIZE;

        double result = d *
            Math.min(K,
                maxProdRate) - c *
            K;
        netResultCurve.
            addValue(result , k)
            ;

        double deriv = (K <=
            maxProdRate ? (d -
            c) : -c);
        netDerivativeCurve.
            addValue(deriv , k);
    }
}

if (A_Sum > 0.0)
{
    // System.out.println( "" +
        netResultCurve.getMaxValTime());

    try
    {
        return netDerivativeCurve.
            getFirstRootTime();
    }
    catch (RelutilsException rex)
    {
        return -99.99; // ERROR
    }
}

```

```

        }
        else
        {
            return -99.99; // ERROR
        }
    }

    public static void createOutputTxtFile ()
    {
        File resultFile = new File("ztxt/" +
            ZXT_FILE_NAME + ".zxt");
        if (resultFile == null)
        {
            PW = null;
            return;
        }

        try
        {
            PW = new PrintWriter(new
                FileOutputStream(resultFile));
        }
        catch (Exception ex)
        {
            PW = null;
        }
    }

    public static void closeOutputTxtFile()
    {
        if (PW != null)
        {
            PW.close();
        }
    }
}

```