

Numerisk løsning av invers-problem ved 'Ensemble Kalman Smoother'

Masteroppgave i anvendt matematikk

Anh Hong Thi Vo

Matematisk institutt
Universitetet i Bergen



2. juni 2008

Forord

Jeg vil først og fremst takke min veileder, Sigurd Aanonsen, for gode råd og oppfølging gjennom arbeidet med oppgaven. Jeg er også takknemlig for din tålmodighet og forståelse under arbeidet mitt, slik at jeg ble topp motivert og hele oppgaveperioden ble en god og konstruktiv læreprosess.

Jeg vil takke Alexander Lundervold for kommentarer, diskusjoner og god faglig bistand. Tineke Røe og Ida Florvåg, jeg er utrolig takknemlig for deres gode vennskap, støtte og hjelp med språklige utfordringer.

Til sist vil jeg takke resten av mine medstudenter som har støttet meg under hele perioden jeg har arbeidet med oppgaven.

Innhold

Forord	i
Innledning	1
1 Modelligninger	5
1.1 Enfasestrømning i porøse medium	5
1.2 Svakt kompressibel væske	6
1.3 Parabolisk varmeligning	7
1.3.1 Reservoarligning for svakt kompressibel væske	7
1.3.2 Forbindelsen mellom reservoarligning og parabolisk var- meligning	8
1.4 Harmonisk middelpermeabilitet	8
1.5 Endelig differansemetode	9
1.5.1 Cellesentrert bevarelseskjema i en dimensjon	9
1.5.2 Cellesentrert bevarelseskjema i to-dimensjonal	11
1.6 Oppsummering	14
2 Statistisk teori	17
2.1 Statistiske definisjoner	17
2.1.1 Betinget sannsynlighet	17
2.1.2 Sannsynlighetstetthet	17
2.1.3 Statistiske momenter	18
2.1.4 Utvalg (Sample)	19
2.1.5 Monte Carlo-metoder	20
2.1.6 Markov-prosess	20
2.1.7 Chapman Kolmogorov-ligningen for en Markov-prosess	21
2.1.8 Bayes' teorem	21
2.2 Definisjoner og teorem for multivariat normalfordeling	21
2.3 Lemma	22

3	Sekvensiell data-assimilasjon	25
3.1	Kalmanfilteret	25
3.1.1	Prosessmodellen	25
3.1.2	Kalman-ligningene	26
3.2	Utvidet Kalman filteret for ikke-lineære modeller	30
3.3	'Kalman Smoother'	33
3.3.1	Smoother-ligningene	34
3.4	'Ensemble' Kalman filter (EnKF)	36
3.4.1	Dynamisk utvikling av feil-statistikk	37
3.4.2	Feil-kovarians-matrise	37
3.4.3	Analyse-ligningene	38
3.4.4	Initial-ensemble	39
3.4.5	Bemerkninger	39
4	Ensemble metoder	41
4.1	Probabilistiske formuleringer	41
4.1.1	Modelligninger og målinger	41
4.1.2	Bayesiansk formulering	41
4.1.3	Diskret formulering	42
4.1.4	Sekvensiell prosessering av målinger	42
4.2	Introduksjon av ensemble metoder	43
4.2.1	Ensemble-representasjon	44
4.2.2	Ensemble-representasjon av målingene	45
4.2.3	Dynamisk utvikling av feil-statistikk	45
4.2.4	Ensemble analyseligninger	46
4.2.5	Bemerkninger	46
4.3	Ensemble 'Smoother'	47
4.4	'Ensemble Kalman Smoother'(EnKS)	48
4.4.1	Bayes med sekvensiell prosessering av data	48
4.4.2	Ensemble-prediksjonsoppdatering	49
4.4.3	Første ensemble-analyseoppdatering	49
4.4.4	Generelle ensemble-analyseoppdatering	49
5	Algoritme	51
5.1	Generell algoritme for Kalman filteret	51
5.2	Grov algoritme for EnKF	53
5.3	Grov algoritme for KS	55
5.3.1	Forover rekursjon	55
5.3.2	Bakover rekursjon	56
5.3.3	Utleddning av formelen for oppdatering	59
5.3.4	Prosessens skjema	61

5.4	Grov algoritme for EnKS	62
5.4.1	Permeabilitet	62
5.4.2	Grov algoritme for EnKS	63
6	Resultater fra EnKF- og EnKS-metode	69
6.1	Resultat fra forover-modellen	69
6.2	Observasjoner for $k_{min} = k_{max} = 1$	70
6.2.1	Differansen mellom løsningene ved k_{min} og k_{max}	70
6.2.2	Differansen mellom initial- og slutt-varians til permeabiliteten for $k_{max} = 1$	71
6.3	Analyse av resultatene	73
6.3.1	Permeabiliteten	73
6.3.2	Trykk	79
6.3.3	Utvikling av filter- og smoother-løsningen	80
7	Konklusjon og videre arbeid	91
7.1	Modelligning og forover-problem	91
7.2	Invers-problem	91
7.3	Videre arbeid	93
A	Liste over symboler	95
	Appendix	95
B	Matlab-koder	97
B.1	Matlab-koder som jeg har fått fra veileder.	97
B.1.1	Koden for inndata, 'inndata1.m', som brukes i de andre kodene i oppgaven.	97
B.1.2	Koden for å beregne korrelasjonen til permeabiliteten, 'corr-sphe.m'	98
B.1.3	Koden for å beregne initialpermeabiliteten og den sanne permeabiliteten, 'cholesky.m'	98
B.1.4	Koden for å sette opp matrisen \mathcal{T} i to-dimensjoner, 'setup.m'.	100
B.1.5	Koden for å skrive ut griddata, 'write-griddata.m'.	104
B.1.6	Koden for å skrive ut sanne data, 'write-welldata.m'.	104
B.1.7	Koden for å løse varmeligningen som en forover-modell, 'forover.m'.	105
B.2	Matlab-koder som jeg har laget i samarbeid med veileder.	108
B.2.1	Koden for Ensemble Kalman filteret, 'EnKF.m'.	108
B.2.2	Koden for Ensemble Kalman Smoother, 'EnKS.m'.	111

B.3	Matlab-koder fra Anh-Hong T.V	116
B.3.1	Koden for å sette opp matrisen \mathcal{T} i en-dimensjon . . .	116
B.3.2	Koden for å beregne matrisen \mathbf{X} , 'X-beregne.m' . . .	116
B.3.3	Koden for å sammeligne verdiene til permeabilitet. . .	117
	Bibliografi	119

Innledning

Et reservoar består av forskjellige faser som berggrunn, olje og vann. Permeabiliteten er ledningsevnen til berggrunnen, det vil si et mål for hvor lett væsken kan strømme gjennom prøven, og vil vanligvis variere med bergarten. Derfor er permeabiliteten en av de viktigste parametrene til bergarten, som har påvirkning på strømmingen av væsker. For utvinning av olje er informasjonen om disse parametrene veldig viktig. Virkelige målinger av permeabiliteten er sjelden tilgjengelig, derfor er et godt matematisk estimat av permeabiliteten nyttig for oljeindustrien. I tillegg er estimering av dynamiske variable, som for eksempel trykk, også nyttig.

Ligningene for reservoarmodellering blir utledet med utgangspunkt i massekonserveringsloven og Darcys lov [1]. Disse ligningene kan behandles som paraboliske varmeligninger. I. Aavatsmark har utledet ligningene for enfase- og tofasestrømning i porøse media, og han har også vist sammenhengen mellom reservoarligningene og den paraboliske varmeligningen [2]. For å løse denne varmeligningen som et forover-problem i en- og to-dimensjoner, bruker han en endelig differansemetode [3], nemlig cellesentrert bevarelseskjema. Ved foreksempel trykktesting antar vi at permeabiliteten, u , til det porøse mediet, er konstant. Dette fører til at trykklikningen er lineær, og vi får dermed en analytisk løsning for permeabiliteten. I virkeligheten er u egentlig en funksjon av $x : u = u(x)$, så vi vil bruke en annen metode. Målet til oppgaven er at vi skal finne statistisk parameter \mathbf{u} utfra målinger av modelltilstanden, trykket, \mathbf{p} , i to-dimensjoner. Dette er et invers-problem. Det vil si at når den første gjetningen av modelltilstanden og et sett av målinger er gitte, vil vi finne ut hva som er det beste estimatet av tilstanden når *a priori* (forrige) estimat og settet av målinger er kjente. Hvordan blir løsningen så lenge vi antar at a priori-feil er Gaussiansk fordeling?

I denne oppgaven for gitt initialverdi av permeabiliteten, estimerer vi trykket og permeabiliteten samtidig fram til vi får verdien (permeabiliteten) som passer beste til den sanne verdien av permeabiliteten. Vi vil samtidig få en fornuftig løsning for trykket. Ellers blir initialmålinger og kunstige data for trykket, benyttet til å estimere både trykk og permeabilitet. Metodene

som kan benyttes for å løse dette problemet, vil vi se nærmere på i oppgaven. Vi vil også finne ut hvilken metode som gir det beste resultatet.

Det er to hovedgrupper av numeriske metoder for å løse invers-problem, sekvensiell data-assimilasjon og ensemble metoder [4]. Ved numerisk løsning diskretiseres ligningen, og antall verdier til permeabiliteten som er en statisk variabel, er et produkt av antall celler i x- og y-retningen.

Filteringsteknikker estimerer tilstanden for hver nye observasjon som er tilgjengelig, derfor er de gode teknikker for løsning av kontrollproblemer [5]. R.E. Kalman ga i 1960 ut en artikkel som beskriver en løsning til det lineære filtreringsproblemet for diskrete data [6]. Denne metoden har fått navnet Kalman filter (KF). Metoden brukes til å estimere tilstanden til prosessen slik at middelverdien til kvadratet av feilen, er minimert.

Klassisk KF egner seg ikke til ikke-lineære modeller, så Kalman utviklet filtermetoden videre slik at den kunne brukes til ikke-lineære problemer. Denne utvidede metoden kalles utvidet Kalman filter (EKF) [7].

For å få bedre løsninger av det inverse-problemet, ble metoden Kalman Smoother (KS) introdusert. Filterligningene brukes til å utlede smootherligningene [8]. Smoother-løsninger er forventet å være 'glattere' enn filterløsninger.

I følge Johns og Mandel [9] er Ensemble Kalman filter (EnKF) en metode der både data og *a priori* (førrige) fordeling blir antatt å være normalfordelte. Kombinasjonen av disse to skaper *a posteriori* (etterfølgende) fordeling, dette skjer i analysetid. Aanonsen et al [10] gir en innføring i EnKF-metoden og anvendelser av denne metoden i petroleumsindustrien.

Det finnes også en introduksjon av EnKF i [11]. Her er EnKF beskrevet som en Monte-Carlo gjennomføring av det Baysiansk oppdaterte problemet. Med Baysiansk oppdatering betyr at sannsynlighetstettheten (*pdf*) til tilstandsvektoren til modellen (*prior/prediksjon*) og sannsynligheten til data er gitt. For å oppnå *pdf* etter at data-sannsynligheten er tatt med i beregning (*posteriori/analyse*), benyttes Bayes teorem.

I rapporten til Almendral-Vazquez og Syversveen [12] summerer de ligningene for KF og EnKF. De beskriver også en algoritme for EnKF. Ved å benytte EnKF, illustrerer de estimatet av permeabiliteten utfra målinger av bunn-trykket i brønnen. Dette er en tilsvarende anvendelse av EnKF i forhold til denne oppgave.

Når det gjelder ensemble metode, introduserer også G. Evensen EnKF-metoden. I tillegg introduserer han også to andre beslektede metoder, Ensemble Smoother (ES) og Ensemble Kalman Smoother (EnKS) [4].

I denne oppgaven er disse tre introduksjonene til stor hjelp for å bygge algoritmen, som kan løse varmeligningen som et invers-problem.

EnKS er en sekvensiell Monte Carlo-metode, basert på Baysiansk stati-

stikk der et ensemble av tilstandsvariabler avanseres fremover i tid, og oppdateres kontinuerlig når nye målinger blir tilgjengelige. I motsetning til EnKF, hvor løsningen bare avhenger av tidligere målinger, tar EnKS ved hvert tidspunkt til etterretning både tidligere og senere observasjoner.

ES, EnKS og EnKF er også presentert av Zhu og Wang [13]. Her er metodene benyttet for å estimere simultant luftkvalitet tilstand og emisjon. Utfra deres eksperimenter finner de at når målinger ikke er tilgjengelige, kan EnKS estimere emisjonen godt ved hvert tidssteg, mens EnKF kan ikke. EnKF kan bare være et godt estimat ved tidssteg der målinger er tilgjengelig. De konkluderer også med at når målingene ved hvert tidssteg er tilgjengelige, vil de to metodene fungere likedan. I denne oppgaven antar vi at vi har målinger ved hvert tidssteg.

Ettersom det kreves mye av matrise algebra og statistiske kunnskaper i disse metodene, anvender vi derfor en del kunnskap om statistikk [14], [15], [4] og lineær algebra [16]. Hovedarbeid i denne oppgaven er å lage koder i Matlab for EnKF og EnKS. Med disse to metodene estimerer vi både trykket som en dynamisk tilstand, og permeabiliteten som en statisk parameter. I tillegg vil vi sammenligne filter- med smoother-løsningen, og se om det stemmer med teorien.

I kapittel 1 utleder vi reservoarmodelligninger og ser på forbindelsen til den paraboliske varmeligningen. Vi utleder også to numeriske metoder (endelig differansemetoder). Metodene kan benyttes til å løse varmeligningen som et forover-problem. Resultatet av forover-modellen kan benyttes videre som en fasit til invers-modellen.

I kapittel 2 repeterer vi noen grunnleggende statistiske definisjoner og resultater, som kan være til hjelp for å forstå senere kapitler.

Ligningene for Kalman filter (KF), utvidet Kalman filter (EKF), Kalman Smoother (KS) og Ensemble Kalman filter (EnKF) blir utledet i Kapittel 3.

Generelle Ensemble metoder er presentert i Kapittel 4. I dette kapittelet ser vi også på Ensemble Smoother (ES) og Ensemble Kalman Smoother (EnKS).

Teorien og resultatene fra disse første kapitlene er anvendt til å konstruere algoritmene i Kapittel 5. Dette kapittelet er et viktig forarbeid for å bygge Matlab kodene som kan løse varmeligningen som et invers-problem. Resultatet er en fullstendig algoritme for EnKF og EnKS. Teoretisk analyse av metodenes resultater blir også utført.

I kapittel 6 presenteres resultatene fra simuleringene vi har gjort. Dette gir en illustrasjon av teorien. Her finnes også diskusjonen av resultatene for å se om disse stemmer overens med de teoretiske analysene i Kapittel 4 og Kapittel 5.

Til slutt, i Kapittel 7, kommer en oppsummering, samt forslag til videre arbeid.

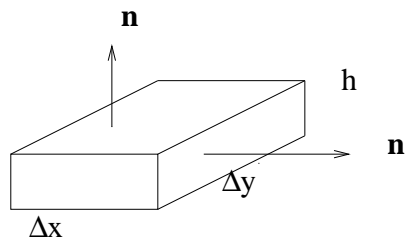
Kapittel 1

Modelligninger

Teorien i dette kapitlet er basert på kompendiet [2].

1.1 Enfasestrømning i porøse medium

I dette avsnittet skal vi bruke en konserveringslov av massen til en volumenet av faststoff og Darcys lov til å utlede reservoarmodelligninger for enfasestrømning i porøse stoffer. Disse ligningene danner et grunnlag for behandlingen av ulike diskretiseringsteknikker, for eksempel *Endelig Differansemetode* som vi skal benytte i denne oppgaven. Ligningene for strømning i porøse



Figur 1.1: En volumenet, Ω , med rand, $\partial\Omega$, og ytre enhetsnormal, \mathbf{n} .

media bygger på prinsippet om at det som strømmer blir bevart. Vi formulerer dette prinsippet ved å betrakte en volumenet, Ω , og krever at for denne volumenet må følgende ligning gjelde:

$$\text{\textit{økning av massen i volumenet } } \Omega = \text{\textit{fluksen inn gjennom sideflatene}} \\ + \text{\textit{innvendig kilde}} .$$

Her antar vi at væsken befinner seg i én fase. Videre skal vi anta at fasen er homogen, slik at vi kan betrakte fasen som én komponent. Ovenstående

bevarelsesligning gjelder da for denne ene komponenten, men siden fasen bare inneholder denne komponenten, gjelder den også for fasen. Bevarelsesligningen får dermed formen

$$\frac{\partial}{\partial t} \int_{\Omega} (\rho\phi) dV = - \int_{\partial\Omega} \rho \mathbf{v} \cdot \mathbf{n} ds + \int_{\Omega} Q dV, \quad (1.1)$$

der porøsiteten, ϕ , er en egenskap til det faste stoffet hvor strømmingen finner sted, Q er massekildetettheten, Ω er randen til volumenheten, ρ er tettheten og \mathbf{n} er ytre enhetsnormal til randen. Det er minusfortegn foran massefluksen fordi det er innstrømming, og enhetsnormalvektoren, \mathbf{n} , peker utover.

Fra Darcys lov har vi

$$\mathbf{v} = -\frac{\mathbf{u}}{\mu} \cdot \nabla p. \quad (1.2)$$

Her er \mathbf{u} permeabiliteten til det faste stoffet, \mathbf{v} volumstrømtetthet, μ viskositet og p er trykk.

Ved å innsette Ligning (1.2) i Ligning (1.1), derivere under integraltegnet og bruke Gauss-sats får vi

$$\int_{\Omega} \frac{\partial}{\partial t} (\rho\phi) dV = \int_{\Omega} \nabla \cdot \left(\rho \frac{\mathbf{u}}{\mu} \cdot \nabla p \right) dV + \int_{\Omega} Q dV. \quad (1.3)$$

1.2 Svakt kompressibel væske

Definisjon av kompressibiliteten er $c = \frac{1}{\rho} \frac{d\rho}{dp}$.

Vi skal igjen anta at det faste stoffet er inkompressibelt. For væsken skal vi nå anta at kompressibiliteten er en liten konstant. Væsken kalles da en *svakt kompressibel* væske eller en *ideell* væske. I dette tilfellet har vi

$$\frac{1}{\rho} \frac{d\rho}{dp} = c (= \text{konstant}). \quad (1.4)$$

Av Ligning (1.4) følger at $\rho = \rho_0 e^{c(p-p_0)}$. For en svakt kompressibel væske er $c \ll 1$, slik at vi kan linearisere. Det vil si at tetthetsfunksjonen kan utvides i en Taylorrekke. Alle ledd av første orden og høyere i c neglisjeres og vi får

$$\rho \approx \rho_0 (1 + c(p - p_0)), \quad (1.5)$$

hvor ρ_0 og p_0 er kjente tilstander for henholdsvis tetthet og trykk.

Av Ligning (1.5) får vi

$$d\rho = \rho_0 c dp \Rightarrow \nabla \rho = \rho_0 c \nabla p. \quad (1.6)$$

1.3 Parabolisk varmeligning

Vi skal vise en forbindelse mellom reservoarligninger og parabolisk varmeligning.

1.3.1 Reservoarligning for svakt kompressibel væske

Med utgangspunktet i Ligning (1.3), utleder vi en reservoarligning for en svakt kompressibel væske.

Ved innsetting av Ligning (1.6) i fluksledd ($= \nabla \cdot (\rho \frac{\mathbf{u}}{\mu} \cdot \nabla p)$) får vi

$$\nabla \cdot (\rho \frac{\mathbf{u}}{\mu} \cdot \nabla p) = \rho \nabla \left(\frac{\mathbf{u}}{\mu} \cdot \nabla p \right) + \rho_0 \frac{c}{\mu} \mathbf{u} \cdot (\nabla p) \cdot (\nabla p). \quad (1.7)$$

For en liten c er vanligvis ledd av typen $c(\nabla p) \cdot (\nabla p) = c(\nabla p)^2$ mye mindre enn ledd av typen $\nabla \cdot (\nabla p) = \nabla^2 p$. Av dette følger det at

$$\rho \frac{\mathbf{u}}{\mu} (\nabla \cdot \nabla p) \gg c \rho_0 \frac{\mathbf{u}}{\mu} (\nabla p)^2. \quad (1.8)$$

Når ulikheten (1.8) er oppfylt forenkles fluksledd til

$$\nabla \cdot (\rho \frac{\mathbf{u}}{\mu} \cdot \nabla p) \approx \rho \nabla \left(\frac{\mathbf{u}}{\mu} \cdot \nabla p \right) \approx \rho_0 \nabla \left(\frac{\mathbf{u}}{\mu} \cdot \nabla p \right). \quad (1.9)$$

Selv om vi endte opp med å sette $\rho = \rho_0$ i Ligning (1.9), er dette *ikke* det samme som å anta at væsken er inkompressibel, og kompressibiliteten i Ligning (1.7) er ivarettatt ved at $c \neq 0$ og $c \ll 1$.

Nå antar vi i tillegg at porøsiteten, ϕ , er konstant i tid, og setter igjen Ligning (1.6) inn i massens endringledd ($= \frac{\partial}{\partial t}(\rho\phi)$). Vi får da

$$\frac{\partial}{\partial t}(\rho\phi) = \phi c \rho_0 \frac{\partial p}{\partial t}, \quad (1.10)$$

og reservoar ligningen for en svakt kompressibel væske blir da

$$\int_{\Omega} \phi \rho_0 c \frac{\partial p}{\partial t} dV = \int_{\Omega} \rho_0 \nabla \left(\frac{\mathbf{u}}{\mu} \cdot \nabla p \right) dV + \int_{\Omega} Q dV, \quad (1.11)$$

eller

$$\phi \rho_0 c \frac{\partial p}{\partial t} - \rho_0 \nabla \left(\frac{\mathbf{u}}{\mu} \cdot \nabla p \right) - Q = 0. \quad (1.12)$$

1.3.2 Forbindelsen mellom reservoarligning og parabolisk varmeligning

Ligning (1.12) er på differensiell form av massens konserveringslov med permeabilitet, \mathbf{u} , som i reservoarmodellen tilsvarende den termiske diffusiviteten av varmeligningen. Med utbytting av parameter i Ligningen (1.12) får

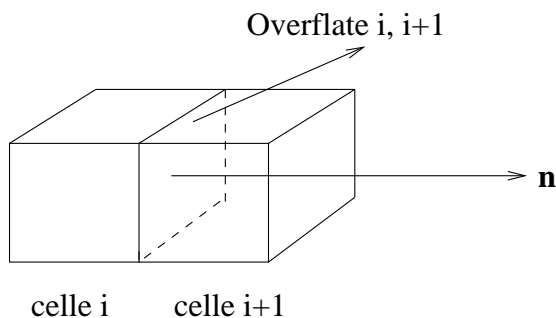
$$\frac{\partial p}{\partial \tau} - \nabla \cdot (\kappa \nabla p) = \frac{Q}{\rho_0}, \quad (1.13)$$

hvor er $\kappa = \frac{\mathbf{u}}{\mu}$ en tensor (en (3×3) -matrise), $\tau = \frac{t}{\phi c}$ og $q = \frac{Q}{\rho_0}$.

Derfor skal vi fra nå av behandle reservoarligningen for enfasestrømning som en parabolisk varmeligning når det gjelder numeriske metoder.

1.4 Harmonisk middelpermeabilitet

For å beregne harmonisk middelpermeabilitet mellom celle (i) og celle ($i+1$), $u_{i,i+1}$, bruker vi igjen massens konserveringslov ved overflaten ($i + \frac{1}{2}$). Dersom



Figur 1.2: Harmonisk middelpermeabilitet.

vi antar at hastigheten er konstant over hver flate, blir fluksen inn i celle ($i+1$) fra celle (i) beskrevet ved hjelp av permeabilitetene, u_i , $u_{i+\frac{1}{2}}$ og u_{i+1} ,

$$\frac{\rho}{\mu} u_i \left(\frac{p_{i+\frac{1}{2}} - p_i}{\frac{1}{2} \Delta x} \right) \Delta y, \quad (1.14)$$

$$\frac{\rho}{\mu} u_{i+1} \left(\frac{p_{i+1} - p_{i+\frac{1}{2}}}{\frac{1}{2} \Delta x} \right) \Delta y, \quad (1.15)$$

$$\frac{\rho}{\mu} u_{i+\frac{1}{2}} \left(\frac{p_{i+1} - p_i}{\Delta x} \right) \Delta y. \quad (1.16)$$

Konserveringslov for massen medfører at (1.14) = (1.15) = (1.16).

Etter eliminering av $p_{i+\frac{1}{2}}$ får vi harmonisk middelpermeabilitet mellom celle (i) og celle ($i + 1$) som er

$$u_{i,i+1} = u_{i+\frac{1}{2}} = 2 \frac{u_i \cdot u_{i+1}}{u_i + u_{i+1}}.$$

1.5 Endelig differansemetode

På grunn av viktigheten av varmeligningen i et bredt spekter av felt, er det mange analytiske løsninger av varmeligningen for en bred variasjon av initial- og grensevilkår. Likevel møter en ofte problemer med spesielle vilkår uten analytiske løsninger, eller hvor den analytiske løsningen er vanskeligere å realisere enn en passende eksakt numerisk løsning. For eksempel i denne oppgaven er permeabiliteten, \mathbf{u} , en funksjon av posisjon og i 2-dimensjonal. Vi vil derfor bruke en spesiell metode for løsning av varmeligningen som kalles 'Endelig Differansemetode'. Denne metoden starter med en diskretisering av rom og tid, slik at det er et heltall antall av punkter i rom, og et heltall antall tidsteg som vi beregner feltvariabelene fra. I dette tilfellet har vi bare en feltvariabel, nemlig trykk.

I dette avsnittet vil vi bruke MATLAB til å utføre en endelig differansemetode for å finne numeriske løsninger av den parabolisk varmeligningen i 2 dimensjoner.

1.5.1 Cellesentrert bevarelseskjema i en dimensjon

Implisitt-metode 1

Denne metoden baseres på 'Endelig differanse approksimasjon til den deriverte.' Den deriverte av en funksjon, f , er en annen funksjon, f' , som defineres av

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}, \quad (1.17)$$

ved alle punkter x hvor grensen eksisterer. Av dette får vi at ligningen for tangentlinjen til $f(x + \Delta x)$ i $(x, f(x))$ er $f(x + \Delta x) = f(x) + f'(x)\Delta x$. Denne formen er en del av Taylorutvikling omkring punktet x . En numerisk metode der formel $[x(t + \Delta t) = x(t) + (\Delta t) \cdot x']$ benyttes, kalles Eulersmetode. En må ta hensyn til "feil", fordi i den numeriske metoden er formelen tilnærmet.

Den lokale trunckeringsfeilen er feilen som introduseres i et steg når vi erstatter en uendelig prosess med en endelig prosess. Foreksempel bytter vi ut en uendelig Taylorrekke for $x(t + \Delta t)$, med en partiell sum i Taylorrekke metoder. Hvis vi beholder tilbake leddene opp til og med $(\Delta t)^n$ i rekken, da er den lokale trunckeringsfeilen summen av alle de gjenstående leddene som vi ikke inkluderer. Av Taylors teorem kan disse leddene presses sammen i et enkelt ledd på formen $\frac{(\Delta t)^{n+1}}{(n+1)!} \cdot x^{n+1}(\xi)$ for et nærliggende punkt ξ . Vi sier at denne metoden har lokal trunckeringsfeil som er $O((\Delta t)^{n+1})$.

Den globale trunckeringsfeilen er forskjellen mellom den eksakte verdien og den beregnende verdien. Akkumulasjonen av mange lokale trunckeringsfeil forårsaker den globale trunckeringsfeilen. Hvis lokal trunckeringsfeil er $O((\Delta t)^{n+1})$, da må global trunckeringsfeil være $O((\Delta t)^n)$, fordi antall av steg, $(T - t_0)/\Delta t$, er nødvendig for å nå et vilkårlig punkt T siden metoden startet ved t_0 . [17]

Når en funksjon $f(x)$, og dens deriverte er entydige, endelige og kontinuerlige funksjoner av x , er Taylorutviklingen omkring punktet x . [3]

$$f(x + \Delta x) = f(x) + f'(x)\Delta x + \frac{f''(x)(\Delta x)^2}{2!} + \frac{f'''(x)(\Delta x)^3}{3!} \dots \quad (1.18)$$

$$f(x - \Delta x) = f(x) - f'(x)\Delta x + \frac{f''(x)(\Delta x)^2}{2!} - \frac{f'''(x)(\Delta x)^3}{3!} \dots \quad (1.19)$$

Ved subtraksjon får vi den sentral-differanse formen av orden $O((\Delta x)^2)$, som er

$$f'(x) \approx \frac{f(x + \Delta x) - f(x - \Delta x)}{2(\Delta x)}. \quad (1.20)$$

Approksimasjonen til den andrederiverte får ved å addere Ligning (1.18) og Ligning (1.19) Altså får vi

$$f''(x) \approx \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{(\Delta x)^2}. \quad (1.21)$$

Global trunckeringsfeil på høyreside av Ligning (1.21) er av orden $O((\Delta x)^2)$, så denne metoden har derfor den lokale trunckeringsfeilen av orden $O((\Delta x)^3)$. Av dette følger en formel for å løse den homogene varmeligningen

$$\frac{p_i^{m+1} - p_i^m}{\Delta t} = \frac{p_{i+1}^{m+1} - 2p_j^{m+1} + p_{i-1}^{m+1}}{(\Delta x)^2}, \quad (1.22)$$

hvor den lokale trunckeringsfeilen for henholdsvis venstre- og høyreside er $O((\Delta t)^2)$ og $O((\Delta x)^3)$.

Ovenstående numerisk metoden benyttes ved å anta at permeabiliteten er konstant både i tid og posisjon. Med utgangspunkt i denne metoden utleder vi en formel for varmeligningen i 1-dimensjon med permeabilitet som er konstant i tid, men som varierer med posisjonen. Varmeligningen blir da

$$\frac{\partial p}{\partial t} - \nabla \cdot (u \nabla p) = \frac{\partial p}{\partial t} - u(x) \nabla^2 p - u'(x) \nabla p = 0 . \quad (1.23)$$

Ved å bruker approksimasjonen til den førstederiverte for $\frac{\partial p}{\partial t}$ og $u'(x)$, og approksimasjonen til den andrederiverte for $\nabla^2 p$, får vi

$$\frac{p_i^{m+1} - p_i^m}{\Delta t} = u_i \frac{p_{i+1}^{m+1} - 2p_j^{m+1} + p_{i-1}^{m+1}}{(\Delta x)^2} + \frac{u_{i+1} - u_{i-1}}{2\Delta x} \frac{p_{i+1}^{m+1} - p_{i-1}^{m+1}}{2\Delta x} . \quad (1.24)$$

Av dette ser vi at metoden har lokal feil av orden $O((\Delta t)^2)$ og $O((\Delta x)^3)$, og at ved $u_i = 0$, $u_{i-1} \neq 0$ og $u_{i+1} \neq 0$ eksisterer en løsning. Det vil si at det skjer strømming der permeabiliteten u_i er 0. Dette strider mot massens konserveringslov.

1.5.2 Cellesentrert bevarelseskjema i to-dimensjonal

Vi integrerer Ligning (1.13) opp igjen for å komme fram til en tilnærmet intergralformulering ved hjelp av ‘cellesentrert rektangulært gitter’

$$\int_{\Omega} \frac{\partial p}{\partial \tau} dV - \int_{\partial \Omega} \kappa \cdot \nabla p \cdot \mathbf{n} ds = \int_{\Omega} q dV , \quad (1.25)$$

hvor $q = \frac{Q}{\rho_0}$.

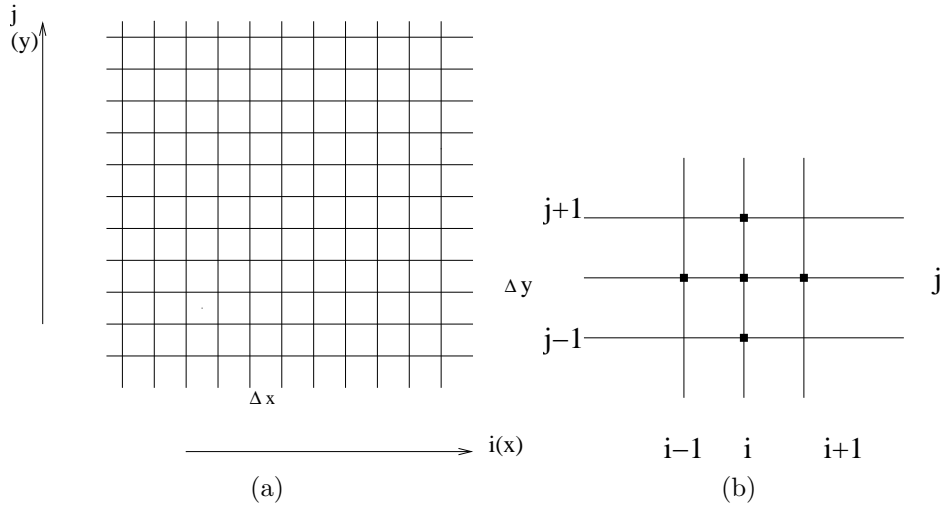
Ved å anta at p er konstant i Ω får vi

$$\int_{\Omega} \frac{\partial p}{\partial \tau} dV = \frac{\partial p}{\partial \tau} dx dy \cdot h = \frac{\partial p}{\partial \tau} \Delta x \Delta y \cdot h . \quad (1.26)$$

I forrige avsnitt nevnte vi at dersom vi antar at hastigheten er konstant, blir fluksen inn i celle k fra celle l

$$\int_{\partial \Omega_{k,l}} \kappa \cdot \nabla p \cdot \mathbf{n} ds = \kappa_{k,l} \left(\frac{p_l - p_k}{d_{k,l}} \right) \Delta s_{kl} , \quad (1.27)$$

hvor $\kappa = \frac{u_{l,k}}{\mu}$, og harmonisk middelpermeabilitet for celle (l) og celle (k) er $u_{k,l} = 2 \frac{u_k u_l}{u_k + u_l}$. Vi forutsetter nå at permeabiliteten har hovedretninger parallelt



Figur 1.3: Figur (a) er celle modellen for to-dimensjonal varmeligning. Figur (b) er cellesentrert gitter for løsningen.

med gitterets retninger, altså kan \mathbf{u} skrives på formen

$$\mathbf{u} = \begin{pmatrix} u_1 & 0 \\ 0 & u_2 \end{pmatrix},$$

i gitterets koordinatsystem. Hver celle indekseres med (i, j) , hvor i er en tellevariabel i x -retning, og j er en tellevariabel i y -retning. Vi går ut fra at \mathbf{u} er konstant i hver gittercelle. Vi søker en løsning $p_{i,j}$ i gitterpunktene. Ligning (1.25) må gjelde for alle cellene i modellen, spesielt for celle (i, j) får vi dermed en integralformulering som kan tilnæmes med varmeligningen

$$\begin{aligned} \frac{\partial p_{i,j}}{\partial \tau} \Delta x \Delta y h &= [\kappa_{i+1,j} \left(\frac{p_{i+1,j} - p_{i,j}}{\Delta x} \right) \Delta y h \\ &+ \kappa_{i-1,j} \left(\frac{p_{i,j} - p_{i-1,j}}{\Delta x} \right) \Delta y h \\ &+ \kappa_{i,j+1} \left(\frac{p_{i,j+1} - p_{i,j}}{\Delta y} \right) \Delta x h \\ &+ \kappa_{i,j-1} \left(\frac{p_{i,j} - p_{i,j-1}}{\Delta y} \right) \Delta x h] = Q_{i,j}, \end{aligned} \quad (1.28)$$

hvor $Q = \int_{\Omega_{i,j}} q dV$. Vi antar at det ikke er noen strøm gjennom rendene. Det vil si at $\kappa = \frac{u}{\mu} = 0$ langs ytterkantene.

Implisitt-metode 2

I denne metoden antar vi også at permeabiliteten varierer med posisjonen, men er konstant i tid. Og vi benytter i tillegg harmonisk middelpermeabilitet til å utlede varmeligningen i en-dimensjonal som er

$$\frac{\partial p}{\partial \tau} \Delta x \Delta y - [\kappa_{i+\frac{1}{2}} \left(\frac{p_{i+1} - p_i}{\Delta x} \right) \Delta y \cdot h + \kappa_{i-\frac{1}{2}} \left(\frac{p_i - p_{i-1}}{\Delta x} \right) \Delta y \cdot h] = Q_i, \quad (1.29)$$

hvor $\kappa_{i-\frac{1}{2}} = 2 \frac{u_{i-1} \cdot u_i}{u_{i-1} + u_i}$ og $\kappa_{i+\frac{1}{2}} = 2 \frac{u_{i+1} \cdot u_i}{u_{i+1} + u_i}$.

Ligning (1.29) får vi en implisitt formel

$$\frac{p_i^{m+1} - p_i^m}{\Delta t} = 2 \frac{\kappa_{i+\frac{1}{2}}}{\Delta x} \left(\frac{p_{i+1}^{m+1} - p_i^{m+1}}{\Delta x} \right) + 2 \frac{\kappa_{i-\frac{1}{2}}}{\Delta x} \left(\frac{p_i^{m+1} - p_{i-1}^{m+1}}{\Delta x} \right) - Q_i, \quad (1.30)$$

med lokal trunkeringsfeil for venstre- og høyreside er henholdsvis $O((\Delta t)^2)$ og $O((\Delta x)^2)$.

Ligning (1.30) kan omskrives som et tridiagonalsystem

$$a_i p_{i-1}^{m+1} + b_i p_i^{m+1} + c_i p_{i+1}^{m+1} - Q_i = p_i^m, \quad (1.31)$$

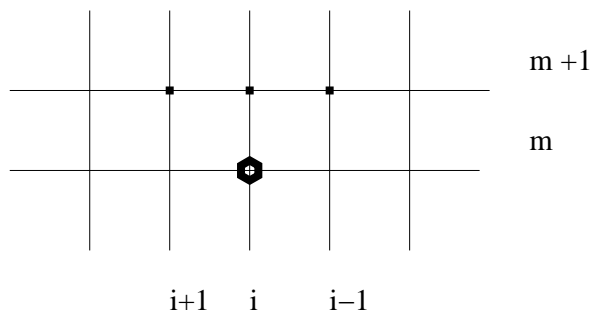
hvor de tre koeffisientene er beregnet ved hjelp av følgende formler

$$\begin{aligned} a_i &= 2 \frac{\Delta t}{(\Delta x)^2} \frac{u_{i-1} \cdot u_i}{u_{i-1} + u_i}, \\ b_i &= 1 + 2 \frac{\Delta t}{(\Delta x)^2} \frac{u_{i+1} \cdot u_i}{u_{i+1} + u_i} - 2 \frac{\Delta t}{(\Delta x)^2} \frac{u_{i-1} \cdot u_i}{u_{i-1} + u_i}, \\ c_i &= -2 \frac{\Delta t}{(\Delta x)^2} \frac{u_{i+1} \cdot u_i}{u_{i+1} + u_i}. \end{aligned} \quad (1.32)$$

Vi ser at $u_i = 0$ medfører at $a_i = 0$, $c_i = 0$ og $b_i = 1$. Dette fører til at Ligning (1.31) ikke har noen løsning. Det vil si at det skjer ingen strømming der permeabiliteten u_i er 0. Dette stemmer med konserveringsloven for massen.

Med $n - 1$ elementer av a_i , $n - 1$ elementer av c_i og n elementer av b_i , danner vi en $(n \times n)$ tridiagonal matrise, \mathcal{T}_i , hvor $a_1 = 0$ og i går fra 1 til n . Tridiagonalsystemet blir da

$$p_i^m = \mathcal{T} \cdot p_i^{m+1} - Q_i = d_i. \quad (1.33)$$



Figur 1.4: Gitteret ved implisitt metode for 1-dimensjonal varmeligningen.

Dette fører til en iterative metode som har løsningsformelen

$$\begin{aligned} p_i^{m+1} &= \mathcal{T}^{-1} \cdot (p_i^m + Q_i), \\ &= \mathcal{T}^{-1} \cdot (d_i + Q_i). \end{aligned} \quad (1.34)$$

1.6 Oppsummering

Vi har i dette kapitlet utledet to numeriske metoder som kan benyttes til å løse reservoar modelligningen i form av varmeligningen som forover-problem, med dette mener vi at vi beregner trykk p , når permeabiliteten u er kjent. Dersom permeabiliteten er konstant både i tid og posisjon, vil disse to metodene være like. Dersom u ikke er konstant i posisjon, er den numeriske løsningen fra metode-1 mer nøyaktig enn den fra metode-2, men vi velger metode-2 videre i oppgaven fordi den i forhold til fysiske egenskaper, som for eksempel massens konserveringslov, er en bedre metode.

Med utgangspunktet i den ovenstående iterative formelen, lager vi en kode i MATLAB som finner numeriske løsninger for forover-problemet i en dimensjon. Vi antar at ved tiden t_0 er både p_0 og u_0 kjente. u_0 er konstant i tid, men varierer i posisjon i mens p varierer både i tid og posisjon. Matrisen \mathcal{T}_i er den samme for alle p . I koden finner vi matrisen \mathcal{T}_i ved å lese inn verdiene til permeabiliteten som en vektor $vk0$, for så å bruke disse verdiene til å beregne a_i, b_i og c_i i matrisen \mathcal{T}_i . Ifølge den iterative formelen kan vi finne verdien av p_i ved neste tidspunkt ved å benytte den forrige verdien av p_i , og matrisen \mathcal{T}_i . Verdiene til p_i og Q_i (innvendig kilde) ved tidspunktet t_0 leses inn som henholdsvis vektorene $vp0$ og $vf0$. Med dette grunnlaget utvider vi en generell kode for den to-dimensjonale varmeligningen, og denne koden kan brukes så gunstig som mulig ved å løse forover-problem, og viderearbeidet

med inverse-problem. Det vil si at vi må finne både p og u i forskjellige tidspunkter.

Kapittel 2

Statistisk teori

2.1 Statistiske definisjoner

I dette kapitlet skal vi gå gjennom noen grunnleggende definisjoner og resultater fra statistikk som vil bli brukt i de påfølgende kapitlene. Dette kapitlet er basert på [14] og Kapittel 2 i [4].

2.1.1 Betinget sannsynlighet

Definisjon 2.1.1. Betinget sannsynlighet er sannsynligheten for at noe skal skje når vi vet at noe annet har skjedd. For eksempel, $\mathcal{P}(A|B)$, er sannsynligheten for at A skal inntreffe når B har inntruffet.

$$\mathcal{P}(A|B) = \frac{\mathcal{P}(A \cap B)}{\mathcal{P}(B)} .$$

2.1.2 Sannsynlighetstetthet

Sannsynlighetstetthet (*pdf: probability density function*)

Definisjon 2.1.2. Sannsynlighetstettheten er en statistisk funksjon som viser hvordan tettheten til de mulige observasjonene i en populasjon er fordelt.

Sannsynlighetstettheten, $f(x)$, til en variabel X tilfredsstillende følgende krav, $f(x) \geq 0$, for alle x , og totalt areal under funksjonskurven $f(x)$ er 1.

Kontinuerlige stokastiske variabler

Definisjon 2.1.3. Stokastisk variabel er en variabel som får sin numeriske verdi bestemt av utfallet av et stokastisk forsøk. Stokastiske variabler er enten diskrete eller kontinuerlige.

Definisjon 2.1.4. En kontinuerlig stokastisk variabel, også kalt måle-variabel, kan ta et uendelig antall mulige verdier, og sannsynlighetsfordelingen er uhen-siktsmessig.

Definisjon 2.1.5. For en stokastisk variabel X med fordelingsfunksjon $F(x) = \mathcal{P}(X \leq x)$, definerer vi sannsynlighetstettheten til X som den deriverte av fordelingsfunksjonen, $f(x) = F'(x)$.

En stokastisk variabel X er kontinuerlig dersom fordelingsfunksjonen $F(x)$ er en kontinuerlig funksjon.

Simultan sannsynlighetstetthet

Definisjon 2.1.6. Funksjonen $p(x, y)$ er simultan sannsynlighetstettheten til to kontinuerlige stokastiske variabler X og Y , hvis

$$\begin{aligned} p(x, y) &\geq 0 \text{ for } \forall (x, y), \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) dx dy &= 1, \\ \mathcal{P}[(XY) \in A] &= \int_A \int p(x, y) dx dy, \forall \text{ region } i \text{ } xy\text{-plan}. \end{aligned}$$

2.1.3 Statistiske momenter

Forventningsverdi

Definisjon 2.1.7. Forventningen til en stokastisk variabel er en verdi, slik at hvis man gjentar eksperimentet som ligger til grunn for variabelen, mange ganger, vil gjennomsnittet av utfallene nærme seg forventningen. Hvis en stokastisk variabel X har tetthetsfunksjon $f(x)$, er forventningsverdien gitt ved

$$\mu = E[X] = \int_{-\infty}^{\infty} x f(x) dx.$$

Forventningsverdien eksisterer bare hvis integralet $\int_{-\infty}^{\infty} |X| f(x) dx$ kon-vergerer.

Standardavvik

Definisjon 2.1.8. Standardavvik, σ , er en parameter som viser hvordan en sannsynlighetsfunksjon sentreres rundt gjennomsnittet, og er lik kvadratrotten i det øyeblikket avviket fra gjennomsnittet kvadreres.

Varians

Definisjon 2.1.9. Varians er kvadratet av standardavviket. Det er et mål på hvor mye alle verdiene i gruppen avviker fra gjennomsnittsverdien for gruppen. Hvis X er en stokastisk variable, så er variansen, $Var(X)$, gitt ved

$$\begin{aligned}\sigma^2 = E[(X - E[X])^2] &= \int_{-\infty}^{\infty} (x - E[X])^2 f(x) dx \\ &= E[X^2] - E[X]^2.\end{aligned}$$

Kovarians og korrelasjon

Korrelasjon og kovarians er to statistiske mål vi kan beregne for å finne ut om det er en lineær samvariasjon mellom to variabler, og er et styrkemål som indikerer hvordan to variabler henger sammen. ([18])

Kovarians

Definisjon 2.1.10. Gitt to stokastiske variabler X og Y , med sannsynlighetstetthet henholdsvis $f(x)$ og $f(y)$. Med disse funksjonene kan vi definere simultan sannsynlighetstetthet som $f(x, y) = f(x|y)f(y) = f(y|x)f(x)$. Deres kovarians er definert som

$$\begin{aligned}Cov(X, Y) &= E[(X - E[X])(Y - E[Y])] \\ &= \int \int_{-\infty}^{\infty} (x - E[X])(y - E[Y])f(x, y) dx dy \\ &= \int \int_{-\infty}^{\infty} xyf(x, y) dx dy - E[X]E[Y].\end{aligned}$$

Korrelasjon

Definisjon 2.1.11. Korrelasjon av to stokastiske variabler X og Y , er

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}}.$$

Korrelasjonsmatrise

Korrelasjonsmatrise er en tabell som brukes til å vise korrelasjoner mellom tre eller flere par av variabler.

2.1.4 Utvalg (Sample)

Utvalget er en liten del av populasjon.

Gjennomsnitt (Sample mean)

Definisjon 2.1.12. Gjennomsnittsverdien av variablene i utvalget er gitt ved

$$\mu = E[X] \simeq \hat{X} = \frac{1}{N} \sum_{i=1}^N X_i, \quad (2.1)$$

hvor \hat{X} er gjennomsnittsverdien.

Empirisk varians (Sample variance)

Definisjon 2.1.13. Empirisk varians kan beregnes med formelen

$$\begin{aligned} \sigma^2 &= E[(\mathbf{X} - E[\mathbf{X}])^2] \\ &= \frac{1}{N-1} \sum_{i=1}^N (X_i - \hat{X})^2. \end{aligned} \quad (2.2)$$

Empirisk kovarians (Sample covarians)

Definisjon 2.1.14. Kovariansen er beregnet ved

$$\begin{aligned} Cov(X, Y) &= E[(X - E[X])(Y - E[Y])] \\ &= \frac{1}{N-1} \sum_{i=1}^N (X_i - \hat{X})(Y_i - \hat{Y}). \end{aligned} \quad (2.3)$$

2.1.5 Monte Carlo-metoder

Metoder brukes til å finne numeriske løsninger på problemer som er så vanskelige at en analytisk løsning bli problematisk. Monte Carlo metoder baserer seg på å lage tilfeldige tall, og finne ut hvilke av disse tallene som passer med gitte egenskaper.

2.1.6 Markov-prosess

En stokastisk prosess er et matematisk objekt som anvendes for å beskrive stokastiske forandringer. En viktig type stokastiske prosesser er Markov-prosessen, der en grovt sagt ser på den tilfeldige utviklingen videre inn i fremtiden ved bare å ta utgangspunkt i hvor en står akkurat i øyeblikket, slik at mulig kjennskap til prosessen i fortiden kan ignoreres. I sannsynlighetsteori, er Markov-prosessen en stokastisk-prosess der kun sannsynlighetsfordelingen til nåværende tilstand har betydning for den fremtidige tilstanden.

Markov-prosessen er uttrykket som

$$\mathcal{P}[x(t_n) \leq x_n \mid x(t) \forall t \leq t_{n-1}] = \mathcal{P}[x(t_n) \leq x_n \mid x(t_{n-1})], \forall t_1 < t_2 \dots < t_n .$$

2.1.7 Chapman Kolmogorov-ligningen for en Markov-prosess

Chapman Kolmogorov-ligningen er en identitet som relaterer simultan sannsynlighetsfordelinger av forskjellige sett av koordinater i en stokastisk prosess, til hverandre. Vi antar at X_i er en indikert samling av stokastiske variabler, og er en stokastisk prosess. La $p_{i_1, \dots, i_n}(X_1, \dots, X_n)$ være simultan sannsynlighetstetthetene til de stokastiske variablene, X_1, \dots, X_n . Chapman Kolmogorov-ligningen blir da

$$p_{i_1, \dots, i_{n-1}}(X_1, \dots, X_{n-1}) = \int_{-\infty}^{\infty} p_{i_1, \dots, i_n}(X_1, \dots, X_n) dX_n .$$

2.1.8 Bayes' teorem

Bayes' teorem

$$\mathcal{P}(B|A) = \frac{\mathcal{P}(A|B)\mathcal{P}(B)}{\mathcal{P}(A)} \propto \mathcal{P}(A|B)\mathcal{P}(B) .$$

2.2 Definisjoner og teorem for multivariat normalfordeling

Definisjon 2.2.1 [15]. Hvis tilfeldige variabler X_1, \dots, X_n , har en multivariat normalfordeling (MVN), da er simultan sannsynlighetstettheten lik som

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}} \frac{1}{\sqrt{\det(\mathbf{C})}} e^{-(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{C}^{-1} (\mathbf{x}-\boldsymbol{\mu})/2} .$$

Her er $\boldsymbol{\mu} = [\mu_1, \mu_2, \dots, \mu_n]^T$, og \mathbf{C} inneholder kovariansene mellom de stokastiske variablene

$$C_{i,j} = Cov(X_i, X_j) .$$

Matrisen \mathbf{C} må være ikke-singulær.

Teorem 1 ([15]). La \mathbf{x} være en multivariat normalvektor med forventningsverdier som defineres ved vektoren $\boldsymbol{\mu}$ og kovarians-matrisen \mathbf{C} . I tillegg lar vi $\mathbf{Y} = \mathbf{A}\mathbf{x}$. Da er \mathbf{Y} også er en multivariat normalvektor med

$$E[\mathbf{Y}] = \mathbf{A}\boldsymbol{\mu}$$

og

$$\text{Cov}(\mathbf{Y}) = \mathbf{A} \mathbf{C} \mathbf{A}^T .$$

2.3 Lemma

Lemma 1 ([8]). La $\mathcal{G}_\psi[\boldsymbol{\mu}, \mathbf{C}]$ betegne en normal/Gaussisk sannsynlighetstetthet med forventningsverdi $\boldsymbol{\mu}$ og kovarians \mathbf{C} . Vi har følgende identiteter,

$$\mathcal{G}_d[\mathbf{A}\Psi, \mathbf{C}] = c_1(d) \mathcal{G}_\psi[(\mathbf{A}^T \mathbf{C}^{-1} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{C}^{-1} \mathbf{d}, (\mathbf{A}^T \mathbf{C}^{-1} \mathbf{A})^{-1}] . \quad (2.4)$$

$$\mathcal{G}_\psi[\mathbf{a}, \mathbf{A}] \mathcal{G}_\psi[\mathbf{h}, \mathbf{H}] = \mathcal{G}_\psi[(\mathbf{A}^{-1} + \mathbf{H}^{-1})^{-1} (\mathbf{A}^{-1} \mathbf{a} + \mathbf{H}^{-1} \mathbf{h}), (\mathbf{A}^{-1} + \mathbf{H}^{-1})^{-1}] \mathcal{G}_a[\mathbf{h}, \mathbf{A} + \mathbf{H}] . \quad (2.5)$$

Altså, hvis vi skriver $\mathbf{z} = [\boldsymbol{\psi}, \mathbf{d}]$, $\boldsymbol{\mu} = [\boldsymbol{\mu}_\psi, \boldsymbol{\mu}_z]$ og

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{\psi\psi} & \mathbf{C}_{\psi d} \\ \mathbf{C}_{d\psi} & \mathbf{C}_{dd} \end{bmatrix} ,$$

da har vi

$$\begin{aligned} \Psi &\sim \int d\mathbf{d} \mathcal{G}_z[\boldsymbol{\mu}, \mathbf{C}] = \mathcal{G}_\psi[\boldsymbol{\mu}_\psi, \mathbf{C}_{\psi\psi}] \\ \Psi | \mathbf{d} &\sim \frac{\mathcal{G}_z[\boldsymbol{\mu}, \mathbf{C}]}{\mathcal{G}_d[\boldsymbol{\mu}_d, \mathbf{C}_{dd}]} = \mathcal{G}_\psi[\boldsymbol{\mu}_\psi - \mathbf{C}_{\psi d} (\mathbf{C}_{dd}^{-1} (\boldsymbol{\mu}_d - \mathbf{d}), \mathbf{C}_{\psi\psi} - \mathbf{C}_{\psi d} (\mathbf{C}_{dd}^{-1} \mathbf{C}_{d\psi})] . \end{aligned}$$

Som et korollar legger vi merke til at $\mathbf{E}[\Psi | \mathbf{d} = \boldsymbol{\mu}_d] = \mathbf{E}[\Psi]$.

Lemma 2 ([8]). Betrakt en $(n \times n)$ -matrise $\mathbf{P} > 0$, en $(m \times m)$ -matrise $\mathbf{R} > 0$, og en $(m \times n)$ -matrise \mathbf{H} , hvor $\mathbf{P} > 0$ betyr at $\mathbf{a}^T \mathbf{P} \mathbf{a} > 0 \forall \mathbf{a}$ (alle positive egenverdi). Dette fører til likeheten

$$\begin{aligned} (\mathbf{P}^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} &= \mathbf{P} - \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T + \mathbf{R})^{-1} \mathbf{H} \mathbf{P} , \\ (\mathbf{P}^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} &= \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T + \mathbf{R})^{-1} . \end{aligned}$$

Lemma 3 ([8]). Hvis de stokastiske variablene $(\Psi_k - \widehat{\Psi}_k^\tau)$ og $\mathbf{d}^\tau = \{\mathbf{d}_1, \dots, \mathbf{d}_\tau\}$ er uavhengige av hverandre, får vi

$$\begin{aligned} \mathbf{E}[\Psi_k, \mathbf{d}^\tau] - \mathbf{E}[\widehat{\Psi}_k^\tau(\mathbf{d}^\tau) \mathbf{d}^\tau] &= \int d\Psi_k d\mathbf{d}^\tau p(\Psi_k, \mathbf{d}^\tau) \Psi_k \mathbf{d}^\tau - \int d\Psi_k d\mathbf{d}^\tau p(\mathbf{d}^\tau) p(\Psi_k | \mathbf{d}^\tau) \Psi_k \mathbf{d}^\tau \\ &= 0 , \end{aligned}$$

siden $p(\Psi_k, \mathbf{d}^\tau) = p(\mathbf{d}^\tau) p(\Psi_k | \mathbf{d}^\tau)$.

Fra dette vi avleder følgende korollar

Korollar 1.

$$\begin{aligned} \mathbf{P}_{k_1, k_2}^\tau &= \mathbf{E}[(\Psi_{k_1} - \widehat{\Psi}_{k_1}^\tau)(\Psi_{k_2} - \widehat{\Psi}_{k_2}^\tau)] \\ &= \mathbf{E}[\Psi_{k_1} \Psi_{k_2}] - \mathbf{E}[\widehat{\Psi}_{k_1}^\tau \widehat{\Psi}_{k_2}^\tau]. \end{aligned} \tag{2.6}$$

Kapittel 3

Sekvensiell data-assimilasjon

Dette kapitlet er basert på [7], [8] og [19].

3.1 Kalmanfilteret

Det klassiske Kalmanfilteret (KF) er en rekursiv løsning til et diskret filtreringsproblem, og denne metoden ble utvidet av Rudolf Kalman i 1960 [6]. Kalmanfilteret er en sekvensiell filter-metode. Det vil si at modellen er integrert framover i tid, hver gang nye målinger er tilgjengelige. Resultatene brukes til å initialisere modellen om igjen før integrasjonen forsetter.

Filteret består av et sett av matematiske ligninger som på en effektiv måte estimerer tilstanden av en prosess, og minimerer middelverdien og kvadratfeilen ved hjelp av numeriske metoder. Som en data-algoritme, kombinerer filteret de tilgjengelige opplysningene av målingene med tidligere informasjon om systemet, for å fremstille et estimat slik at feilen som oppstår er statistisk minimert. Med dette filteret kan vi estimere fortid-, nåtid- og framtidige tilstander. En slik estimering kan også utføres selv når de virkelige egenskapene av det modellerte systemet er ukjente. Filteret er derfor nyttige i flere aspekter.

3.1.1 Prosessmodellen

Ligningene i avsnittet er hentet fra [7].

Proessen har to hovedsteg: et *prediksjonsteg* og et *analysesteg*.

Prediksjonsteget er hovedsakelig kjøring av problemet framover i tid, til punktet hvor nye målinger er tilgjengelige. Det vil si at dette steget bruker tilstanden estimert fra forrige tidsteg til å produsere et estimat av tilstanden ved nåværende tidsteg.

Det neste steget er analysesteget. Her blir nye målinger tatt med i beregningen, og de sammenlignes med opplysninger fra prediksjonen. Disse opplysningene avhenger av usikkerheten til både prediksjonene og målingene. Målingsinformasjon ved nåværende tidsteg benyttes til å forbedre denne prediksjonen, for å oppnå et nytt mer nøyaktig estimat av tilstanden.

I KF-metoden antar vi at den eksakte tilstanden ved tid k utvikles fra tilstanden ved tid $(k-1)$. Tilstanden $\Psi_k \in \mathbf{R}^n$ av en prosess estimeres av en dynamisk modell

$$\Psi_k = \mathbf{A}\Psi_{k-1} + \mathbf{w}_{k-1}. \quad (3.1)$$

Relasjonen mellom tilstanden Ψ_k og målingene (eller observasjonene) $\mathbf{d}_k \in \mathbf{R}^m$ er gitt ved

$$\mathbf{d}_k = \mathbf{H}\Psi_k + \mathbf{v}_k. \quad (3.2)$$

Her antar vi at den $(n \times n)$ -matrisen \mathbf{A} er konstant i hvert tidsteg. Observasjonsmatrisen \mathbf{H} for modellen, er en $(m \times n)$ -matrise som avbilder det eksakte tilstandsrommet til det observerte rommet. Matrisen \mathbf{H} antas å være konstant i hvert tidsteg. De stokastiske variablene \mathbf{w}_k og \mathbf{v}_k representerer prosess- og målingsfeil ved tidsteg k . Disse to feilene, som antas å være uavhengige av hverandre, er normalfordelt med forventning lik 0 og kovariansene, \mathbf{Q}_k og \mathbf{R}_k

$$\begin{aligned} \mathbf{w}_k &\sim \mathcal{N}(0, \mathbf{Q}_k) = \mathcal{G}(0, \mathbf{Q}_k), \\ \mathbf{v}_k &\sim \mathcal{N}(0, \mathbf{R}_k) = \mathcal{G}(0, \mathbf{R}_k). \end{aligned} \quad (3.3)$$

3.1.2 Kalman-ligningene

Målet til filtreringen er å beregne *posteriori fordeling* av tilstand Ψ_k ved tidsteg k , gitt historien av målingene opptil tidsteg k (se [19] og [8])

$$p(\Psi_k | \mathbf{d}_1, \dots, \mathbf{d}_k) = p(\Psi_k | \mathbf{d}_{1:k}).$$

Prediksjonsteget bruker *Chapman-Kolmogorov*-ligningen for å propagere *posteriori fordelingen* av forrige- til nåværende tidsteg, og den tar hensyn til usikkerheten forårsaket av den dynamiske modellen $p(\Psi_k | \Psi_{k-1})$ som er

$$p(\Psi_k | \mathbf{d}_{1:k-1}) = \int p(\Psi_k | \Psi_{k-1}) p(\Psi_{k-1} | \mathbf{d}_{1:k-1}) d\mathbf{d}_{k-1}.$$

Analysesteget forener informasjonen i den predikerte fordelingen og målingssannsynligheten (likelihood) $p(\mathbf{d} | \Psi_k)$ ved å bruke *Bayes' teorem*:

$$p(\Psi_k | \mathbf{d}_{1:k}) = \frac{p(\mathbf{d}_k | \Psi_k) p(\Psi_k | \mathbf{d}_{1:k-1})}{\int p(\mathbf{d}_k | \Psi_k) p(\Psi_k | \mathbf{d}_{1:k-1}) d\Psi_k}.$$

Det er gitt et sett av målinger, $\mathbf{d}^\tau = \{\mathbf{d}_1, \dots, \mathbf{d}_k, \dots, \mathbf{d}_\tau\}$, og vi vil estimere modellens tilstand og dens feil-kovarians ved tidspunktet gitt ved k . Hvis τ er lik nåværende k sier vi at vi *filterer* tilstanden, og hvis τ er mindre enn k sier vi at vi *predikerer* tilstanden.

Ved $\tau = k$ har vi

$$\begin{aligned} p(\Psi_k | \mathbf{d}^\tau) &= p(\Psi_k | \mathbf{d}^k) = \frac{p(\mathbf{d}_k | \Psi_k) p(\Psi_k | \mathbf{d}^{k-1})}{p(\mathbf{d}_k | \mathbf{d}^{k-1})}, & (3.4) \\ p(\Psi_k | \mathbf{d}^{k-1}) &= \int p(\Psi_k | \Psi_{k-1}) p(\Psi_{k-1} | \mathbf{d}^{k-1}) d\Psi_{k-1}. \end{aligned}$$

Nevneren fra Ligning (3.4) er en uviktig normaliseringsfaktor. De gjenstående sannsynlighetstetthetene er gitt ved

$$\begin{aligned} p(\mathbf{d}_k | \Psi_k) &= \mathcal{G}_{\mathbf{d}_k}[\mathbf{H}\Psi_k, \mathbf{R}_k], \\ p(\Psi_k | \Psi_{k-1}) &= \mathcal{G}_{\Psi_k}[\mathbf{A}\Psi_{k-1}, \mathbf{Q}_{k-1}]. \end{aligned} \quad (3.5)$$

Ligningen for $\tau < k$ er

$$p(\Psi_k | \mathbf{d}^\tau) = \int d\Psi_{k-1} p(\Psi_k | \Psi_{k-1}) p(\Psi_{k-1} | \mathbf{d}^\tau). \quad (3.6)$$

Ved $\tau < k$ vil vi beregne middelveidien, $\widehat{\Psi}_k^{k-1}$, og feil-kovarians, \mathbf{P}_k^{k-1} , som begge er uttrykket vha. $\widehat{\Psi}_{k-1}^{k-1}$, \mathbf{P}_{k-1}^{k-1} og $p(\Psi_{k-1} | \mathbf{d}^{k-1})$.

Tilstandsprediksjon – middelveid

Ved hvert tidsteg k har vi en z -dimensjonal vektor av observasjonene \mathbf{d}_k som avhenger på tilstanden og tilleggsgøy. Vi antar at \mathbf{v}_k og \mathbf{w}_k ikke er korrelert med tilstanden Ψ_k , så

$$\mathbf{E}[\Psi_k, \mathbf{w}_k] = 0 \text{ for } k \leq z, \quad (3.7)$$

$$\mathbf{E}[\mathbf{d}_k, \mathbf{w}_z] = 0 \text{ for } k \leq z, \quad (3.8)$$

$$\begin{aligned} \mathbf{E}[\mathbf{w}_k, \mathbf{w}_z] &= 0 \text{ for } k \neq z, \\ &= \mathbf{Q}_k \text{ for } k = z. \end{aligned} \quad (3.9)$$

Benyttelse av Ligning (3.1) og Ligning (3.8) i beregningen av middelveidien

$$\begin{aligned} \widehat{\Psi}_k^{k-1} &= \mathbf{E}[\Psi_k | \mathbf{d}^{k-1}] \\ &= \mathbf{A}\mathbf{E}[\Psi_{k-1} | \mathbf{d}^{k-1}] + \mathbf{E}[\mathbf{w}_{k-1} | \mathbf{d}^{k-1}] \\ &= \mathbf{A}\widehat{\Psi}_{k-1}^{k-1}. \end{aligned} \quad (3.10)$$

Tilstandsprediksjon – feilen

$$\begin{aligned} (\Psi_k - \widehat{\Psi}_k^{k-1}) &= \mathbf{A}\Psi_{k-1} + \mathbf{w}_{k-1} - \mathbf{A}\widehat{\Psi}_{k-1}^{k-1} \\ &= \mathbf{A}(\Psi_{k-1} - \widehat{\Psi}_{k-1}^{k-1}) + \mathbf{w}_{k-1}. \end{aligned} \quad (3.11)$$

Fra Ligning (3.7) og Ligning (3.8) ser vi at \mathbf{w}_{k-1} er ikke korrelert med $(\Psi_{k-1} - \widehat{\Psi}_{k-1}^{k-1})$, som er en funksjon av Ψ_{k-1} og \mathbf{d}^{k-1} .

Tilstandsprediksjon – feil-kovariansen

$$\begin{aligned} \mathbf{P}_k^{k-1} &= \mathbf{E}[(\Psi_k - \widehat{\Psi}_k^{k-1})(\Psi_k - \widehat{\Psi}_k^{k-1})^T] \\ &= \mathbf{E}[(\mathbf{A}(\Psi_{k-1} - \widehat{\Psi}_{k-1}^{k-1}) + \mathbf{w}_{k-1})(\mathbf{A}(\Psi_{k-1} - \widehat{\Psi}_{k-1}^{k-1}) + \mathbf{w}_{k-1})^T] \\ &= \mathbf{A}\mathbf{E}[(\Psi_{k-1} - \widehat{\Psi}_{k-1}^{k-1})(\Psi_{k-1} - \widehat{\Psi}_{k-1}^{k-1})^T]\mathbf{A}^T + \mathbf{E}[\mathbf{w}_{k-1}\mathbf{w}_{k-1}^T] \\ &= \mathbf{A}\mathbf{P}_{k-1}^{k-1}\mathbf{A}^T + \mathbf{Q}_{k-1}. \end{aligned} \quad (3.12)$$

Vi skriver Ligning (3.4) som

$$p(\Psi_k|\mathbf{d}^k) = \frac{\mathcal{G}_{\mathbf{d}_k}[\mathbf{H}\Psi_k, \mathbf{R}]\mathcal{G}_{\Psi_k}[\widehat{\Psi}_k^{k-1}, \mathbf{P}_k^{k-1}]}{p(\mathbf{d}_k|\mathbf{d}^{k-1})}. \quad (3.13)$$

Hvis vi anvender Ligning (2.4) fra Lemma 1 til det andre leddet i telleren i Ligning (3.13), og så bruker Ligning (2.5) til resultatet, får vi at

$$\begin{aligned} p(\Psi_k|\mathbf{d}^k) &= c_1(\mathbf{d}^k) \mathcal{G}_{\Psi_k} [([\mathbf{P}_k^{k-1}]^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}([\mathbf{P}_k^{k-1}]^{-1}\widehat{\Psi}_k^{k-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}_k) \\ &\quad , \quad ([\mathbf{P}_k^{k-1}]^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}], \end{aligned} \quad (3.14)$$

siden produktet av to Gaussiske fordelinger er en Gaussisk fordeling. Dessuten normaliserer vi Ligning (3.14) med hensyn på Ψ_k , og lar $c_1(\mathbf{d}^k) = 1$. Til slutt, ved å benytte Lemma 2 får vi at $p(\Psi|\mathbf{d}^k)$ er en Gaussisk fordeling med forventning $\widehat{\Psi}_k^k$ og kovarians \mathbf{P}_k^k , som kan skrives på følgende måte

$$\begin{aligned} \mathbf{P}_k^k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^{k-1}, \\ \widehat{\Psi}_k^k &= \widehat{\Psi}_k^{k-1} + \mathbf{K}_k(\mathbf{d}_k - \mathbf{H}\widehat{\Psi}_k^{k-1}), \\ \mathbf{K}_k &= \mathbf{P}_k^{k-1}\mathbf{H}^T(\mathbf{R} + \mathbf{H}\mathbf{P}_k^{k-1}\mathbf{H}^T)^{-1}. \end{aligned} \quad (3.15)$$

Vi definerer prediksjonen (forutsigelsen) ved tidsteg k , $\Psi_k^f := \widehat{\Psi}_k^{k-1}$ som *a priori* (tidligere) tilstandsestimat, følgelig blir da $\Psi_k^a := \widehat{\Psi}_k^k$ *a posteriori*

(etterfølgende) tilstandsestimat etter analysen.

Den *a priori* feil-kovariansen ved tidsteg k gitt ved

$$\mathbf{P}_k^f = E \left[(\Psi_k - \Psi_k^f)(\Psi_k - \Psi_k^f)^T \right], \quad (3.16)$$

og *a posteriori* feil-kovariansen er gitt ved

$$\mathbf{P}_k^a = E \left[(\Psi_k - \Psi_k^a)(\Psi_k - \Psi_k^a)^T \right]. \quad (3.17)$$

Fra Ligning (3.15) ser vi at ligningen som beregner et *a posteriori* tilstandsestimat Ψ_k^a er en lineær kombinasjon av et *a priori* estimat og en veid differanse mellom en aktuell måling \mathbf{d}_k og en målingsprediksjon $\mathbf{H}\Psi_k^f$

$$\Psi_k^a = \Psi_k^f + \mathbf{K}_k(\mathbf{d}_k - \mathbf{H}\Psi_k^f). \quad (3.18)$$

Differansen $(\mathbf{d}_k - \mathbf{H}\Psi_k^f)$ kalles *innovasjon* eller *residual* (resterende).

A priori feilkovariansen til prediksjonen \mathbf{P}_k^f og måling-feil-kovariansen \mathbf{R}_k , bestemmer i hvilken grad målingen påvirker estimatet, gjennom \mathbf{K}_k . Denne $(n \times m)$ -matrisen \mathbf{K} er valgt til å være *gain-* eller *blending-*faktoren som minimerer *a posteriori* feil-kovariansen i Ligning (3.17). Denne matrisen kalles *Kalman Gain matrise*.

Ved å se på skalartilfellet til Ligning (3.15) og Ligning (3.18) som deles i to ekstreme tilfeller, kan vi illustrere noen av egenskapene til Kalman filteret.

1. Hvis \mathbf{R}_k i skalatilfellet går mot null, ser vi fra Ligning (3.15) at *gain* \mathbf{K} vurderer residualen i høyere grad. Spesielt er $\lim_{\mathbf{R}_k \rightarrow 0} \mathbf{K}_k = \mathbf{H}^{-1}$ og derfor er

$$\Psi_k^a = \mathbf{H}^{-1}\mathbf{d}_k. \quad (3.19)$$

Da er den aktuelle målingen \mathbf{d}_k mer pålitelig i forhold til andre tilfeller, mens i den predikerte målingen $\mathbf{H}\Psi_k^f$ er påliteligheten mindre. I dette ekstreme tilfellet forkastes den predikerte målingen fullstendig, og kun de målingene som brukes i oppdateringen benyttes.

2. Den andre ekstreme situasjonen er når *a priori* feilkovariansen \mathbf{P}_k^f nærmer seg null, da vurderer \mathbf{K} residualen i mindre grad. Spesielt er $\lim_{\mathbf{P}_k^f \rightarrow 0} \mathbf{K}_k = 0$. Her er \mathbf{d}_k mindre pålitelig sammenlignet med andre tilfeller, mens $\mathbf{H}\Psi_k^f$ er mer pålitelig. Oppdateringen blir da

$$\Psi_k^a = \Psi_k^f. \quad (3.20)$$

De nye målingene er her ikke tatt med i beregningen i analysen. Den normale situasjonen er når den oppdaterte tilstandsvektoren Ψ_k^a er en kombinasjon både av *a priori* estimat og fordelingen fra de nye målingene.

Nedenfor representerer vi de spesifikke ligningene for tidsoppdaterte og målingsoppdaterte for Kalman filteret

De tidsoppdaterte ligningene for KF

$$\begin{aligned}\Psi_k^f &= \mathbf{A}\Psi_{k-1}^a, \\ \mathbf{P}_k^f &= \mathbf{A}\mathbf{P}_{k-1}^a\mathbf{A}^T + \mathbf{Q}_{k-1}.\end{aligned}$$

Her ser vi at tilstanden og kovariansen estimeres forover fra steg $k - 1$ til steg k . \mathbf{A} og \mathbf{B} er fra Ligning (3.1), mens \mathbf{Q} er fra Ligning (3.3).

De målingsoppdaterte ligningene for KF

$$\mathbf{K}_k = \mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R}_k)^{-1}, \quad (3.21)$$

$$\Psi_k^a = \Psi_k^f + \mathbf{K}_k(\mathbf{d}_k - \mathbf{H}\Psi_k^f), \quad (3.22)$$

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^f. \quad (3.23)$$

Den første oppgaven i løpet av målingsoppdateringen er å regne ut *Kalman gain*, \mathbf{K}_k . Det neste steget er å avpasse prosessen for å oppnå \mathbf{d}_k , så genererer et *a posteriori* tilstandsestimat ved å inkorporere målingen som i Ligning (3.22). Tilslutt oppnår *a posteriori* feil-kovariansen av estimatet via Ligning (3.23). Etter hver tids- og målingsoppdatert parvis er prosessen repetert med de tidligere *a posteriori* estimatene som brukes til å prosjektere eller å predikere de nye *a priori* estimatene.

Bemerkning 3.1.1. For KF [20]

1. Løsninger er de approksimerte løsningene av invers-problem.
2. Oppdatering av modell-tilstander og usikkerhet i rekkefølge (sekvensiell).
3. Estimatenes forbedres, og usikkerheten reduseres ved hver oppdatering.

3.2 Utvidet Kalman filteret for ikke-lineære modeller

Som beskrevet i forrige avsnittet estimeres tilstanden i Kalman filteret av et sett matematiske ligninger som utledes av et *lineært* system. Filteret er

derfor begrenset til en lineære antagelse, og er ikke egnet for de fysiske modellene som ofte er *ikke-lineære*. Med hensyn til *ikke-lineære* problem, blir det klassiske Kalman filteret utvidet. I dette utvidede filteret (EKF) trenger ikke tilstandsovergangen og målingsmodellene å være *lineære* funksjoner av tilstanden. EKF er et Kalman filter som lineariserer om nåværende middelerverdi og kovarians.

La oss anta igjen at prosessen har tilstandsvektoren $\Psi \in \mathbf{R}^n$, men prosessen ledes nå av det *ikke-lineære* dynamisk systemet [7]

$$\begin{aligned}\Psi_k &= \mathbf{g}(\Psi_{k-1}, \mathbf{w}_{k-1}), \\ \mathbf{d}_k &= \mathbf{h}(\Psi_k, \mathbf{v}_k).\end{aligned}\tag{3.24}$$

Ved normal antagelse blir målingene

$$\mathbf{d}_k = \mathbf{h}(\Psi_k) + \mathbf{v}_k.\tag{3.25}$$

De stokastiske variablene \mathbf{w}_k og \mathbf{v}_k representerer prosess- og målingsfeil ved tidsteg k , slik som i Ligning (3.1) og Ligning (3.2), hvor kovariansmatrisene er gitt i Ligning (3.3) og Ligning (3.16). Den ikke-lineære funksjonen, \mathbf{g} , i Ligning (3.24) relaterer tilstanden ved det forrige tidsteget, $(k-1)$, til tilstanden ved det nåværende tidsteget, k . Og den ikke-lineære funksjonen \mathbf{h} relaterer tilstanden Ψ_k til målingen \mathbf{d}_k . Funksjonen \mathbf{g} kan derfor brukes til å beregne prediksjon-tilstanden fra det forrige estimatet, og likedan benyttes funksjonen \mathbf{h} til å regne ut prediksjon- målingen fra prediksjon-tilstanden. Likevel kan ikke \mathbf{g} og \mathbf{h} anvendes direkte til å finne kovariansen. Isteden beregnes en matrise av partielle deriverte, det vil si Jacobi-matrisen.

I praksis vet vi ikke verdiene til feilene, \mathbf{w}_k og \mathbf{v}_k , ved hvert tidsteg. Vi approksimerer derfor prediksjonen til tilstanden med null feil, dette fører til følgende ligninger for *extended* Kalman filteret som er

$$\Psi_k^f \approx \mathbf{g}(\Psi_{k-1}^a, 0),\tag{3.26}$$

$$\mathbf{d}_k^f \approx \mathbf{h}(\Psi_k^f) + \mathbf{v}_k.\tag{3.27}$$

Ligningene som lineariserer et estimat om Ligning (3.26) og Ligning (3.27), blir da

$$\Psi_k \approx \Psi_k^f + \mathbf{A}(\Psi_{k-1} - \Psi_{k-1}^a) + \mathbf{W}\mathbf{w}_{k-1},\tag{3.28}$$

$$\mathbf{d}_k \approx \mathbf{d}_k^f + \mathbf{H}(\Psi_k - \Psi_k^f) + \mathbf{V}\mathbf{v}_k.\tag{3.29}$$

Her er Ψ_k og \mathbf{d}_k eksakte tilstands- og målingsvektorer ved tidsteg k . Matrisene \mathbf{A} og \mathbf{W} er Jacobi-matriser av de deriverte av \mathbf{g} med hensyn på henholdsvis

Ψ og \mathbf{w} . De er gitt ved

$$\begin{aligned}\mathbf{A}(i, j) &= \frac{\partial \mathbf{g}(i)}{\partial \Psi(j)}(\Psi_{k-1}^a, 0), \\ \mathbf{W}(i, j) &= \frac{\partial \mathbf{g}(i)}{\partial \mathbf{w}(j)}(\Psi_{k-1}^a, 0).\end{aligned}$$

Tilsvarende er Jacobi-matrisene \mathbf{H} og \mathbf{V} de deriverte av \mathbf{h} med hensyn på Ψ og \mathbf{v} . Disse matrisene uttrykkes ved

$$\begin{aligned}\mathbf{H}(i, j) &= \frac{\partial \mathbf{h}(i)}{\partial \Psi(j)}(\Psi_k^f, 0), \\ \mathbf{V}(i, j) &= \frac{\partial \mathbf{h}(i)}{\partial \mathbf{v}(j)}(\Psi_k^f, 0).\end{aligned}$$

Ligninger (3.28) og (3.29) kan omskrives

$$\Psi_k - \Psi_k^f = \mathbf{A}(\Psi_{k-1} - \Psi_{k-1}^a) + \Upsilon_k, \quad (3.30)$$

$$\mathbf{d}_k - \mathbf{d}_k^f = \mathbf{H}(\Psi_k - \Psi_k^f) + \Phi_k. \quad (3.31)$$

Her er Υ_k og Φ_k representerer nye uavhengige, stokastiske variabler som har forventning lik 0 og kovarians henholdsvis \mathbf{WQW}^T og \mathbf{VRV}^T med \mathbf{Q} og \mathbf{R} som i Ligning (3.3). De to ovenstående Ligningene er lineære og minner om ligningene (3.1) og (3.2) fra KF.

Det *a posteriori* tilstandsestimatet for den ikke-lineære prosessen Ψ_k^a med *estimat* ϵ_k blir da

$$\Psi_k^a = \Psi_k^f + \epsilon_k. \quad (3.32)$$

Variablene i Ligning (3.30) og Ligning (3.31) har følgende sannsynlighetsfordelinger

$$\begin{aligned}(\Psi_k - \Psi_k^f) &\sim \mathcal{N}(0, E[(\Psi_k - \Psi_k^f)(\Psi_k - \Psi_k^f)^T]), \\ \Upsilon_k &\sim \mathcal{N}(0, \mathbf{WQ}_k \mathbf{W}^T), \\ \Phi_k &\sim \mathcal{N}(0, \mathbf{VR}_k \mathbf{V}^T).\end{aligned} \quad (3.33)$$

Anta at den predikerte verdien av ϵ_k er null. Hvis vi benytter Kalman-filterligningen for å estimere ϵ_k

$$\epsilon_k = \mathbf{K}_k(\Psi_k - \Psi_k^f). \quad (3.34)$$

Dette medfører at

$$\Psi_k^a = \Psi_k^f + \mathbf{K}_k(\mathbf{d}_k - \mathbf{d}_k^f). \quad (3.35)$$

Ligning (3.35) kan nå brukes for måling-oppdatering i EKF med Ψ_k^f og \mathbf{d}_k^f fra Ligning (3.26) og Ligning (3.27), og \mathbf{H}_k fra Ligning (3.21) med den passende innsetting for måling-feil-kovarians

De tidsoppdaterte ligningene for EKF

$$\begin{aligned}\Psi_k^f &= \mathbf{g}(\Psi_{k-1}^a, 0), \\ \mathbf{P}_k^f &= \mathbf{A}_k \mathbf{P}_{k-1}^a \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q}_{k-1} \mathbf{W}_k^T.\end{aligned}$$

hvor \mathbf{A}_k og \mathbf{W}_k er prosess -Jacobi ved tidsteg k , og \mathbf{Q}_k er prosess-støy kovariansen fra Ligning (3.3), ved tidsteg k .

De målingsoppdaterte ligningene for EKF

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_k^f \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T)^{-1}, \\ \Psi_k^a &= \Psi_k^f + \mathbf{K}_k (\mathbf{d}_k - \mathbf{h}(\Psi_k^f, 0)), \\ \mathbf{P}_k^a &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^f.\end{aligned}$$

Bemerkning 3.2.1. Matrisene \mathbf{A} og \mathbf{H} er ikke konstant i hvert tidsteg .

3.3 'Kalman Smoother'

Vi vet at Kalmanfilteret er en reell tidsprosess-algoritme der er den estimerte tilstanden oppdatert øyeblikkelig etter en ny observasjon (måling) er tilgjengelig. *Kalman smoother* (KS) estimerer hver tilstand Ψ_k ved å basere på alle observasjonene [8].

KS består av to deler. Den første delen er en *framover-rekursiv* der brukes observasjonene fra \mathbf{d}_1 til \mathbf{d}_τ , er kjent som Kalman filteret. Den andre delen er *bakover-rekursiv* som benytter observasjonene fra \mathbf{d}_τ til \mathbf{d}_1 . KS har som formål å beregne *posterior fordeling* av tilstand Ψ_k ved tidsteg k etter å ha tatt imot målingene opptil tidsteg τ , hvor $\tau > k$

$$p(\Psi_k | \mathbf{d}_1, \dots, \mathbf{d}_\tau) = p(\Psi_k | \mathbf{d}_{1:\tau}).$$

For å estimere en tilstand Ψ_k , gitt data $\mathbf{d}^\tau = \{\mathbf{d}_1, \dots, \mathbf{d}_k, \dots, \mathbf{d}_\tau\}$, benytter vi først Kalmanfilter-ligningene helt til vi har nådd tilstand Ψ_τ . Mens fremover-rekursive utføres, oppbevarer vi verdiene for $\hat{\Psi}_k^a$, $\hat{\Psi}_k^f$, \mathbf{P}_k^a og \mathbf{P}_k^f med $k = 1 \dots \tau$. Etterpå utfører vi bakover-rekursive ved å benytte *smoother* ligninger helt til vi har nådd tilstanden Ψ_k som vi vil estimere. Resultatet vil være mindre usikkerhet. Det vil si at estimatene forventes å være mer nøyaktige enn filter-estimatene i KF, fordi vi inkluderer flere observasjoner.

3.3.1 Smoother-ligningene

Tilstand-middelverdien

Hvis vi lar $\Psi = \Psi_{k-1}$, $\mathbf{d} = \Psi_k$, $\boldsymbol{\mu}_\Psi = \widehat{\Psi}_k^\tau$ og $\mathcal{G}_z[\boldsymbol{\mu}, \mathbf{C}] = p(\Psi_{k-1}, \Psi_k | \mathbf{d}^\tau)$ i Lemma 1, får vi

$$\widehat{\Psi}_{k-1}^\tau = \mathbf{E}[\Psi_{k-1} | \mathbf{d}^\tau] = \mathbf{E}[\Psi_{k-1} | \Psi_k = \widehat{\Psi}_k^\tau, \mathbf{d}^\tau]. \quad (3.36)$$

Vi har

$$\begin{aligned} p(\Psi_{k-1}, \Psi_k | \mathbf{d}^\tau) &= \frac{p(\Psi_{k-1}, \Psi_k, \mathbf{d}^{k-1}, \mathbf{d}_k, \dots, \mathbf{d}_\tau)}{p(\mathbf{d}^\tau)} \\ &= \frac{p(\mathbf{d}_k, \dots, \mathbf{d}_\tau | \Psi_{k-1}, \Psi_k, \mathbf{d}^{k-1}) p(\Psi_k | \Psi_{k-1}, \mathbf{d}^{k-1}) p(\Psi_{k-1} | \mathbf{d}^{k-1}) p(\mathbf{d}^{k-1})}{p(\mathbf{d}^\tau)} \\ &= \frac{p(\mathbf{d}_k, \dots, \mathbf{d}_\tau | \Psi_k) p(\Psi_k | \Psi_{k-1}) p(\Psi_{k-1} | \mathbf{d}^{k-1})}{p(\mathbf{d}_k, \dots, \mathbf{d}_\tau | \mathbf{d}^{k-1})} \\ &= c_1(\Psi_k, \mathbf{d}^\tau) p(\Psi_k | \Psi_{k-1}) p(\Psi_{k-1} | \mathbf{d}^{k-1}) \\ &= c_1(\Psi_k, \mathbf{d}^\tau) \mathcal{G}_{\Psi_k}[\mathbf{A}\Psi_{k-1}, \mathbf{Q}] \mathcal{G}_{\Psi_{k-1}}[\widehat{\Psi}_{k-1}^{k-1}, \mathbf{P}_{k-1}^{k-1}]. \end{aligned}$$

Bruker Ligning (2.4) i Lemma 1 til det andre leddet i (3.37), og så anvender vi Ligning (2.5) til dette resultatet. Vi får da

$$\begin{aligned} p(\Psi_{k-1} | \Psi_k, \mathbf{d}^\tau) &= \frac{p(\Psi_{k-1}, \Psi_k | \mathbf{d}^\tau)}{p(\Psi_k | \mathbf{d}^\tau)} \\ &= c_2(\Psi_k, \mathbf{d}^\tau) \mathcal{G}_{\Psi_{k-1}} [([\mathbf{P}_{k-1}^{k-1}]^{-1} + \mathbf{A}^T \mathbf{Q}^{-1} \mathbf{A})^{-1} ([\mathbf{P}_{k-1}^{k-1}]^{-1} \widehat{\Psi}_{k-1}^{k-1} \\ &\quad + \mathbf{A}^T \mathbf{Q}^{-1} \Psi_k), ([\mathbf{P}_{k-1}^{k-1}]^{-1} + \mathbf{A} \mathbf{Q}^{-1} \mathbf{A})^{-1}]. \end{aligned} \quad (3.37)$$

Ligning (3.37) tilsvare Ligning (3.14). Vi lar nå $c_2(\Psi_k, \mathbf{d}^\tau) = 1$, bruker Lemma 2 og Ligning (3.36), og får at

$$\widehat{\Psi}_{k-1}^\tau = \widehat{\Psi}_{k-1}^{k-1} + \mathbf{L}_{k-1} (\widehat{\Psi}_k^\tau - \widehat{\Psi}_k^{k-1}). \quad (3.38)$$

Ved bruk av (3.12) får

$$\mathbf{L}_{k-1} = \mathbf{P}_{k-1}^{k-1} \mathbf{A}^T [\mathbf{P}_k^{k-1}]^{-1}. \quad (3.39)$$

Kovarians-matrise

Fra (3.10) og (3.38), observerer vi at

$$(\Psi_{k-1} - \widehat{\Psi}_{k-1}^\tau) + \mathbf{L}_{k-1} \widehat{\Psi}_k^\tau = (\Psi_{k-1} - \widehat{\Psi}_{k-1}^{k-1}) + \mathbf{L}_{k-1} \mathbf{A} \widehat{\Psi}_{k-1}^{k-1}. \quad (3.40)$$

Hvis vi multipliserer begge sider av Ligning (3.40) med deres respektive transponerte fra høyre-side, husk at $(\Psi_{k-1} - \widehat{\Psi}_{k-1}^\tau)$ er uavhengig av $\widehat{\Psi}_k^\tau$, likedan

for $(\Psi_{k-1} - \widehat{\Psi}_{k-1}^{k-1})$ og $\widehat{\Psi}_k^{k-1}$, og tar forventningsverdien av begge sider, får vi at

$$\mathbf{P}_{k-1}^\tau + \mathbf{L}_{k-1} \mathbf{E}[\widehat{\Psi}_k^\tau \widehat{\Psi}_k^\tau] \mathbf{L}_{k-1}^T = \mathbf{P}_{k-1}^{k-1} + \mathbf{L}_{k-1} \mathbf{A} \mathbf{E}[\widehat{\Psi}_{k-1}^{k-1} \widehat{\Psi}_{k-1}^{k-1}] \mathbf{A}^T \mathbf{L}_{k-1}^T. \quad (3.41)$$

Bruk av Ligningene (3.7), (3.9) og Korollar 1 av Lemma 3 får vi

$$\begin{aligned} \mathbf{E}[\widehat{\Psi}_k^\tau \widehat{\Psi}_k^\tau] - \mathbf{P}_k^\tau &= \mathbf{E}[\widehat{\Psi}_k \widehat{\Psi}_k] \\ &= \mathbf{E}[(\mathbf{A} \Psi_{k-1} + \mathbf{w}_{k-1})(\mathbf{A} \Psi_{k-1} + \mathbf{w}_{k-1})^T] - \mathbf{P}_k^\tau \\ &= \mathbf{A} \mathbf{E}[\Psi_{k-1} \Psi_{k-1}] \mathbf{A}^T + \mathbf{Q} - \mathbf{P}_k^\tau. \end{aligned} \quad (3.42)$$

Tilsvarende har vi at

$$\mathbf{E}[\widehat{\Psi}_{k-1}^{k-1} \widehat{\Psi}_{k-1}^{k-1}] = \mathbf{E}[\Psi_{k-1} \Psi_{k-1}] - \mathbf{P}_{k-1}^{k-1}. \quad (3.43)$$

Setter vi Ligning (3.42) og Ligning (3.43) inn i Ligning (3.41), og bruker (3.12), finner vi *Kalman Smoother* ligningen for å beregne kovarians

$$\mathbf{P}_{k-1}^\tau = \mathbf{P}_{k-1}^{k-1} + \mathbf{L}_{k-1} (\mathbf{P}_k^\tau - \mathbf{P}_k^{k-1}) \mathbf{L}_{k-1}^T. \quad (3.44)$$

Bemerkning 3.3.1. Analyseverdiene fra KF-ligningene blir prediksjonsverdiene for KS-ligningene.

Vi har de spesifikke ligningene for Kalman Smoother nedenfor

De tidsoppdaterte ligningene for KS

$$\begin{aligned} \Psi_k^a &= \Psi_k^f + \mathbf{K}_k (\mathbf{d} - \mathbf{H} \Psi_k^f), \\ \mathbf{P}_k^a &= [\mathbf{I} - \mathbf{K}_k] \mathbf{P}_k^f. \end{aligned} \quad (3.45)$$

De målingsoppdaterte ligningene for KS

$$\begin{aligned} \mathbf{L}_k &= \mathbf{P}_k^a \mathbf{A}^T [\mathbf{P}_{k+1}^a]^{-1}, \\ \Psi_k^s &= \Psi_k^a + \mathbf{L}_k [\Psi_{k+1}^s - \mathbf{A} \Psi_k^a], \\ \mathbf{P}_k^s &= \mathbf{P}_k^a + \mathbf{L}_k [\mathbf{P}_{k+1}^s - \mathbf{P}_{k+1}^a] \mathbf{L}_k^T. \end{aligned}$$

3.4 'Ensemble' Kalman filter (EnKF)

EKF er ikke en optimal estimator. I tillegg, hvis initial-estimatet av tilstanden er feil eller prosessen er modellert uriktig, vil filteret divergerer raskt på grunn av dets lineærisering. Et annet problem med EKF er at den estimerte kovariansmatrisen har en tendens til å underestimere den sanne kovariansmatrisen. På grunn av dette risikerer filteret derfor å være inkonsistent i statistisk betydning hvis støyen ikke er stabilisert i tillegg [21]. EKF er ikke egnet til veldig store modeller og mislykkes hvis ikke-lineæritetene er altfor store. *Ensemble* teknikker innebærer en passelig løsning for å unngå en slik lineærisering. *Ensemble Kalman filteret* (EnKF) ble introdusert i 1994 av Geir Evensen som et alternativ til det EKF, der kovariansmatrisen byttes ut med *sample* (utvalgs-)kovariansen.

EnKF er et rekursive filter som egner seg til problemer med et stort antall av variabler, for eksempel diskretiseringen av partielle differensialligninger i geofysiske modeller. I ensemble Kalman filteret er en ensemble av modelltilstander integrert framover i tid, og brukt til å regne ut middelveidien og feil-kovariansen når det kreves. Og et antall av initial-tilstandsvektorene genereres før assimileringen av data, og de statistiske opplysningene kalkuleres av *ensemble* selv. I tillegg antar EnKF at sannsynlighetsfordelingen som involveres er *Gauss* [11]. *Kalman Gain* beregnes nå fra feil-kovariansen som skapes fra ensemble. Sammen med dette blir de oppdaterte ligningene fra det klassiske Kalman filteret brukt i EnKF.

Tilstandsvektoren etter kjøringen av prediksjons- og analysesteget blir kallet henholdsvis Ψ_k^f , og Ψ_k^a som tidligere. Med indeks i som refererer til i -te medlemmet av ensemblet ved tid k , er engangskjøringen av modellen for hvert medlem av *ensemblet* er gitt ved

$$\Psi_{k,i}^f = \mathbf{g}(\Psi_{k-1,i}^a). \quad (3.46)$$

Ved tidsteg k er de nye målingene \mathbf{d}_k tilgjengelige. Målingene er relatert til tilstandsvektoren gjennom matrisen \mathbf{H} .

$$\mathbf{d}_k = \mathbf{H}\Psi_k. \quad (3.47)$$

Ligningen (3.47) representerer en idealisert situasjon uten støy eller målingsfeil. Vi antar at disse virkningene inkluderes og kan beskrives av en *Gauss* variabel med kovariansmatrisen \mathbf{R}_k . I [22] viser Burger at dersom målingene ikke behandles som stokastiske variabler, vil det oppdaterte ensemblet ha den korrekte middelveidien, men alt for lav varians. For å unngå dette bruker vi de aktuelle målingene bare som referansesmålingene og genererer de nye

målinger for hvert medlem av ensemblet ved å addere en vektor av tilfeldig målingsusikkerhet $\mathbf{v}_{k,j}$ til den reelle målingen \mathbf{d}_k

$$\mathbf{d}_{k,j} = \mathbf{d}_k + \mathbf{v}_{k,j} .$$

med

$$\begin{aligned} \widehat{\mathbf{v}_k \mathbf{v}_k^T} &= \mathbf{R}^e , \\ \mathbf{R}^e &\simeq \mathbf{R} , \end{aligned}$$

der indeks j er j -te måling ved tidsteg k , og $\mathbf{v}_{k,j} \sim \mathcal{N}(0, \mathbf{R}_k)$.

3.4.1 Dynamisk utvikling av feil-statistikk

Ligningene i avsnittet er hentet fra Kapittel 4 i [4]. For ikke lineære-modell hvor modellen er ikke perfekt og inneholder modell-feil, vi kan beskrive modellen som en stokastisk ligning

$$d\Psi = \mathbf{g}(\Psi)dt + \mathbf{h}(\Psi)d\mathbf{q} . \quad (3.48)$$

Denne ligningen gir at en økning i tid vil føre til en økning i Ψ , og den blir påvirket av et tilfeldig bidrag av en stokastisk term $\mathbf{h}(\Psi)d\mathbf{q}$, som representerer modell-feil. Hvert medlem av ensemblet utvikles i henhold til modelldynamikk som uttrykkes ved den stokastiske differensialligningen (3.48).

Sannsynlighetstettheten $f(\Psi)$ utvikles i tid da ifølge Kolmogorov-ligning

$$\frac{\partial f}{\partial t} + \sum \frac{\partial(g_i f)}{\partial \psi_i} = \frac{1}{2} \sum_{i,j} \frac{\partial^2 f(\mathbf{hQh}^T)_{ij}}{\partial \psi_i \partial \psi_j} , \quad (3.49)$$

hvor g_i er komponent nummer i av modelloperatoren \mathbf{g} , og \mathbf{hQh}^T er kovariansmatrise for modellfeilen. Ligning (3.49) løses ved Monte Carlo metoder.

3.4.2 Feil-kovarians-matrise

Feil-kovariansmatriser for prediksjon- og analyseestimat er definert i KF, og er

$$\begin{aligned} \mathbf{P}_k^f &= E \left[\left(\Psi_k^f - \Psi_k \right) \left(\Psi_k^f - \Psi_k \right)^T \right] \\ \mathbf{P}_k^a &= E \left[\left(\Psi_k^a - \Psi_k \right) \left(\Psi_k^a - \Psi_k \right)^T \right] . \end{aligned} \quad (3.50)$$

Vi definerer $\widehat{\psi}_k$ som er forventningsverdien (middelverdien) av *a priori* ensemblet

$$\Psi_k \approx \widehat{\Psi}_k = \frac{1}{N_e} \sum_{i=1}^{N_e} \Psi_{k,i}^f, \quad (3.51)$$

hvor N_e er størrelsen til ensemblet. Tilstanden Ψ_k er ukjent, vi definerer derfor ensemble feilkovariansmatrisen omkring middelverdien $\Psi_k \approx \widehat{\Psi}_k$

$$\mathbf{P}_k^f \simeq (\mathbf{P}_k^f)^e = 1/(N_e - 1) \sum_{i=1}^{N_e} (\Psi_{k,i}^f - \widehat{\Psi}_k^f)(\Psi_{k,i}^f - \widehat{\Psi}_k^f)^T.$$

3.4.3 Analyse-ligningene

Analysesteget for EnKF består av følgende oppdateringer som utføres for hvert ensemble-medlem

$$\Psi_{k,i}^a = \Psi_{k,i}^f + (\mathbf{P}_k^f)^e \mathbf{H}^T (\mathbf{H}(\mathbf{P}_k^f)^e \mathbf{H}^T + \mathbf{R}^e)^{-1} (\mathbf{d}_{k,j} - \mathbf{H}\Psi_{k,i}^f). \quad (3.52)$$

Med en endelig størrelse av ensemblet vil Ligning (3.52) være en approksimasjon. Oppdateringen av middelverdien $\widehat{\Psi}_k^a$, og posteriori-feilkovarians $(\mathbf{P}_k^a)^e$ blir da

$$\begin{aligned} \widehat{\Psi}_k^a &= \widehat{\Psi}_k^f + (\mathbf{P}_k^f)^e \mathbf{H}^T (\mathbf{H}(\mathbf{P}_k^f)^e \mathbf{H}^T + \mathbf{R}_k)^{-1} (\widehat{\mathbf{d}}_k - \mathbf{H}\widehat{\Psi}_k^f), \\ (\mathbf{P}_k^a)^e &= (\mathbf{P}_k^f)^e - (\mathbf{P}_k^f)^e \mathbf{H}^T (\mathbf{H}(\mathbf{P}_k^f)^e \mathbf{H}^T + \mathbf{R})^{-1} \mathbf{H}(\mathbf{P}_k^f)^e. \end{aligned} \quad (3.53)$$

Her er $\widehat{\mathbf{d}}_k = \mathbf{d}_k$ den første gjettede vektoren av målinger. Derfor er relasjonen mellom analyse-tilstanden og prediksjon-tilstanden i EnKF er det samme som i KF. Uttrykket for Kalman gain matrisen $(\mathbf{K}_k)^e$ er

$$(\mathbf{K}_k)^e = (\mathbf{P}_k^f)^e \mathbf{H}^T (\mathbf{H}(\mathbf{P}_k^f)^e \mathbf{H}^T + (\mathbf{R}_k)^e)^{-1}.$$

Fra (3.52) og (3.53) har vi

$$\Psi_{k,i}^a - \widehat{\Psi}_k^a = (\mathbf{I} - (\mathbf{K}_k)^e \mathbf{H})(\Psi_{k,i}^f - \widehat{\Psi}_k^f) + (\mathbf{K}_k)^e (\mathbf{d}_j - \widehat{\mathbf{d}}).$$

Dette fører til at *a posteriori* feil kovarians-matrisen $(\mathbf{P}_k^a)^e$ blir

$$(\mathbf{P}_k^a)^e = (\mathbf{I} - (\mathbf{K}_k)^e)(\mathbf{P}_k^f)^e.$$

3.4.4 Initial-ensemble

Vi definerer et initial-ensemble for å initialisere filteret. Dette blir gjort ved å spesifisere et initial middel-ensemble $\widehat{\Psi}_0^a$, og en initial kovariansmatrise \mathbf{Q}_0 . I tillegg er de medlemmene av initial-ensemblet skapt ved å anta normalfordeling (Gaussisk-fordeling),

$$\Psi_{0,i}^a = \widehat{\Psi}_0^a + \mathbf{w}_{0,i} , \quad (3.54)$$

hvor $\mathbf{w}_{0,i} \sim \mathcal{N}(0, \mathbf{Q}_0)$. Ensemblet av modell-tilstander er integrert forover ifølge Ligning (3.48).

3.4.5 Bemerkninger

Bemerkning 3.4.1. For lineære modeller har vi følgende bemerkninger hentet Kapittel 4 fra i [4].

1. EnKF-løsninge vil konvergere akkurat til KF-løsning med økning i størrelsen av ensemblet.
2. Feil-kovarians-matrise, \mathbf{P}^e , konvergerer til \mathbf{P} for en endelig størrelse av ensemblet, og uavhengig av modellen, \mathbf{R}^e konvergerer til \mathbf{R} og \mathbf{Q}^e konvergerer til \mathbf{Q} .

Bemerkning 3.4.2. For ikke-lineare modeller bruker ikke EnKF lineariseringen, noe som fører til at EnKF kan tolkes som en Monte-Carlo-metode der ensemblet av modelltilstander utvikles i tilstandsrommet med gjennomsnittet som beste estimat og spredningen av ensemblet som som feilvarians. På målingstidspunktene er hver observasjon representert ved et annet ensemble, der gjennomsnittet er den faktiske målingen og variansen av ensemblet representerer målingsfeilen. Vi kan dermed si at vi kombinerer et stokastisk prediksjonssteg med et stokastisk analysesteg [4].

Kapittel 4

Ensemble metoder

Teorien i dette kapittelet er basert på kapittel 9 i [4].

4.1 Probabilistiske formuleringer

4.1.1 Modelligninger og målinger

Med $\Psi(\mathbf{x}, t) \in \mathbf{R}^{n_\psi}$ er en vektor som inneholder n_ψ modellvariabler hvor hver variabel er en funksjon av rom og tid, definerer vi den ikke-lineære modell som

$$\frac{\partial \Psi(\mathbf{x}, t)}{\partial t} = \mathbf{g}(\Psi(\mathbf{x}, t), \mathbf{u}(\mathbf{x})) + \mathbf{q}(\mathbf{x}, t). \quad (4.1)$$

Her er $\mathbf{g}(\Psi, \mathbf{u}) \in \mathbf{R}^{n_\psi}$ en ikke-lineære operator, \mathbf{u} er statisk parameter. For målingene er

$$\mathbf{H}(\Psi, \mathbf{u}) = \mathbf{d} + \mathbf{v}, \quad (4.2)$$

der \mathbf{H} , \mathbf{d} og \mathbf{v} er som før.

4.1.2 Bayesiansk formulering

Simultan-pdf for tilstanden som en funksjon av tid, rom og parametre, er $f(\Psi, \mathbf{u})$, og sannsynlighets(likelihood)-funksjonen for målingene er $f(\mathbf{d} | \Psi, \mathbf{u})$. Ved bruk av Bayes' teorem kan vi skrive parameter-estimering-problem som

$$\begin{aligned} f(\Psi, \mathbf{u} | \mathbf{d}) &\propto f(\Psi, \mathbf{u})f(\mathbf{d} | \Psi, \mathbf{u}), \\ f(\Psi, \mathbf{u}, \Psi_0 | \mathbf{d}) &\propto f(\Psi | \mathbf{u}, \Psi_0)f(\Psi_0)f(\mathbf{u})f(\mathbf{d} | \Psi, \mathbf{u}), \end{aligned} \quad (4.3)$$

hvor er $\Psi_0(\mathbf{x}) \in \mathbf{R}^{n_\psi}$ er initial-ensemble.

4.1.3 Diskret formulering

Modelltilstand diskretiseres i tid. Det vil si at $\Psi(\mathbf{x}, t)$ er representert ved fiksert tidsintervall som $\Psi_i(\mathbf{x}) = \Psi(\mathbf{x}, t_i)$ med $i = 0, 1, \dots, k$. Vi definerer *pdf* for modell-integrasjon fra tid t_{i-1} til t_i som $f(\Psi_i | \Psi_{i-1}, \mathbf{u})$. Vi antar at:

1. Modellen er en første ordens Markov prosess

$$f(\Psi_1, \dots, \Psi_k, \mathbf{u}, \Psi_0) \propto f(\mathbf{u})f(\Psi_0) \prod_{i=1}^k f(\Psi_i | \Psi_{i-1}, \mathbf{u}) . \quad (4.4)$$

2. Måling vektor $\mathbf{d}_j \in \mathbf{R}^{m_j}$ ved tider $t_{i(j)}$ med $j = 1, \dots, m$, er bare avhengig av $\Psi(t_{i(j)} = \Psi_{i(j)})$ eller \mathbf{u} .

3. Målingsfeilen er uavhengige av tid. Vi kan da skrive

$$f(\mathbf{d} | \Psi, \mathbf{u}) = \prod_{j=1}^m f(\mathbf{d}_j | \Psi_{i(j)}, \mathbf{u}) . \quad (4.5)$$

Bayes' teorem gir da

$$f(\Psi_1, \dots, \Psi_k, \mathbf{u}, \Psi_0 | \mathbf{d}) \propto f(\mathbf{u})f(\Psi_0) \prod_{i=1}^k f(\Psi_i | \Psi_{i-1}, \mathbf{u}) \prod_{j=1}^m f(\mathbf{d}_j | \Psi_{i(j)}, \mathbf{u}) . \quad (4.6)$$

4.1.4 Sekvensiell prosessering av målinger

Så lenge modellen er en første ordens Markov prosess, og målingene er tilgjengelige ved diskrete tider med feilen uavhengig av tid, kan vi prosessere målingene sekvensielt i tid.

Den første oppdatering

$$f(\Psi_1, \dots, \Psi_{i(1)}, \mathbf{u}, \Psi_0 | \mathbf{d}_1) \propto f(\mathbf{u})f(\Psi_0) \prod_{i=1}^{i(1)} f(\Psi_i | \Psi_{i-1}, \mathbf{u}) f(\mathbf{d}_1 | \Psi_{i(1)}, \mathbf{u}) . \quad (4.7)$$

Den andre oppdatering

$$f(\Psi_1, \dots, \Psi_{i(2)}, \mathbf{u}, \Psi_0 \mid \mathbf{d}_1, \mathbf{d}_2) \propto f(\Psi_1, \dots, \Psi_{i(1)}, \mathbf{u}, \Psi_0 \mid \mathbf{d}_1) \prod_{i=i(1)+1}^{i(2)} f(\Psi_i \mid \Psi_{i-1}, \mathbf{u}) f(\mathbf{d}_2 \mid \Psi_{i(2)}, \mathbf{u}). \quad (4.8)$$

Den k -te orden oppdatering

$$f(\Psi_1, \dots, \Psi_{i(k)}, \mathbf{u}, \Psi_0 \mid \mathbf{d}_1, \dots, \mathbf{d}_k) \propto f(\Psi_1, \dots, \Psi_{i(k-1)}, \mathbf{u}, \Psi_0 \mid \mathbf{d}_1, \dots, \mathbf{d}_{k-1}) \prod_{i=i(k-1)+1}^{i(k)} f(\Psi_i \mid \Psi_{i-1}, \mathbf{u}). \quad (4.9)$$

4.2 Introduksjon av ensemble metoder

I Kapittel 3 viste vi hvordan ensemble Kalmanfilteret konstrueres fra Kalmanfilteret. I dette kapitlet skal vi se at denne konstruksjonen kan generaliseres.

Ensemble-metoder søker å forbedre den prediktive styrken til en gitt statistisk lærende eller modelltilpassende teknikk. Det generelle prinsippet til ensemble-metoder er å konstruere en lineærkombinasjon av en modelltilpassningsmetoder, istedenfor å bruke en enkelt tilpasning [23].

Ensemble analyseoppdateringsligninger

$$\Psi^a = \Psi^f + \mathbf{H}^T [\mathbf{P}^f] (\mathbf{H}^T [\mathbf{H} [\mathbf{P}^f]] + \mathbf{R})^{-1} (\mathbf{d} - \mathbf{H} [\Psi^f]), \quad (4.10)$$

$$\mathbf{P}^a = \mathbf{P}^f - \mathbf{H}^T [\mathbf{P}^f] (\mathbf{H}^T [\mathbf{H} [\mathbf{P}^f]] + \mathbf{R})^{-1} \mathbf{H} [\mathbf{P}^f]. \quad (4.11)$$

Feil-kovariansmatrisen

Ensemble kovarians er definert omkring ensemble-forventningen $\widehat{\Psi}$ som

$$\mathbf{P} = E \left[\left(\Psi - \widehat{\Psi} \right) \left(\Psi - \widehat{\Psi} \right)^T \right]. \quad (4.12)$$

For ensemblemetoder [20]

1. Ensembleforventningen, $\widehat{\Psi}$, er den beste gjetningen.
2. Feil-variansen er definert av ensemble-spredning.
3. Glatthet av ensemble-medlemmer definerer kovariansen.

4.2.1 Ensemble-representasjon

Definisjon av ensemblematrisen

La n_ψ være antall variabler i tilstand-vektoren Ψ , n_u antall parametre i \mathbf{u} , og $n = n_\psi + n_u$. Vi definerer da matrisen $\mathbf{B}(\mathbf{x}, t_i) \in \mathbf{R}^{n \times N}$ som inneholder N ensemble-medlemmer av $\Psi(\mathbf{x}, t_i) \in \mathbf{R}^{n_\psi \times N}$ og $\mathbf{u}(\mathbf{x})$, ved tiden t_i , som

$$\mathbf{B}_i = \mathbf{B}(\mathbf{x}, t_i) = \begin{pmatrix} \Psi^1(\mathbf{x}, t_i) & \Psi^2(\mathbf{x}, t_i) & \dots & \Psi^N(\mathbf{x}, t_i) \\ \mathbf{u}^1(\mathbf{x}, t_i) & \mathbf{u}^2(\mathbf{x}, t_i) & \dots & \mathbf{u}^N(\mathbf{x}, t_i) \end{pmatrix} .$$

Ensemble-pertubasjonsmatrisen

Ensemble-forventningen er lagret i hver kolonne av ensemble-forventningsmatrisen, $\widehat{\mathbf{B}}(\mathbf{x}, t_i)$. Denne matrisen er definert som

$$\widehat{\mathbf{B}}(\mathbf{x}, t_i) = \mathbf{B}(\mathbf{x}, t_i) \cdot \mathbf{1}_N , \quad (4.13)$$

hvor $\mathbf{1}_N \in \mathbf{R}^{N \times N}$ er matrisen hvor hvert element er lik $1/N$. Ensemble-pertubasjonsmatrise kan da defineres som

$$\mathbf{B}^*(\mathbf{x}, t_i) = \mathbf{B}(\mathbf{x}, t_i) - \widehat{\mathbf{B}}(\mathbf{x}, t_i) = \mathbf{B}(\mathbf{x}, t_i)(\mathbf{I} - \mathbf{1}_N) . \quad (4.14)$$

Ensemble-kovariansmatrisen

Den feil-kovariansmatrisen som i (4.12), er da definert

$$\mathbf{P}(\mathbf{x}_1, \mathbf{x}_2, t_i) = \frac{\mathbf{B}^*(\mathbf{x}_1, t_i)(\mathbf{B}^*(\mathbf{x}_2, t_i))^T}{N - 1} . \quad (4.15)$$

Ensemblematrise for simultan tilstand

For $i' = 1, 2, \dots, i$, gitt den ensemble matrisen for forskjellige tid som er $\mathbf{B}(\mathbf{x}, t_{i'})$, kan vi definere den ensemble matrisen for simultan tilstand fra t_0 til t_i som

$$\widetilde{\mathbf{B}}_i = \begin{pmatrix} \mathbf{B}(\mathbf{x}, t_0) \\ \vdots \\ \mathbf{B}(\mathbf{x}, t_i) \end{pmatrix} . \quad (4.16)$$

Rom-tid ensemble-kovarians mellom modelltilstanden ved to vilkålige tider t_1 og t_2 blir da

$$\tilde{\mathbf{P}}^e(\mathbf{x}_1, t_1, \mathbf{x}_2, t_2) = \frac{\widetilde{\mathbf{B}}^*(\mathbf{x}_1, t_1)(\widetilde{\mathbf{B}}^*(\mathbf{x}_2, t_2))^T}{N-1}. \quad (4.17)$$

4.2.2 Ensemble-representasjon av målingene

Ved tiden $t_{i(j)}$, la en vektor av målinger $\mathbf{d}_j \in \mathbf{R}^{m_j}$, der m_j er antall målinger ved tiden t_i være gitt. Vi definerer N vektorer av pertubasjon-målinger som $\mathbf{d}_j^l = \mathbf{d}_j + \mathbf{v}_j^l$ med $l = 1, \dots, N$. Disse vektorene er lagret i kolonnene til en matrise \mathbf{D}

$$\mathbf{D}_j = (\mathbf{d}_j^1, \mathbf{d}_j^2, \dots, \mathbf{d}_j^N), \quad (4.18)$$

hvor $\mathbf{D} \in \mathbf{R}^{m_j \times N}$. Ensemblet av måling-pertubasjoner som har null-middelverdi, kan lagres i matrisen $\mathbf{V} = (\mathbf{v}_j^1, \mathbf{v}_j^2, \dots, \mathbf{v}_j^N) \in \mathbf{R}^{m_j \times N}$. Fra dette kan vi konstruere ensemble-representasjonen til måling-feil-kovarians som

$$\mathbf{R}^e(t_{i(j)}) = \frac{\mathbf{V}_j \mathbf{V}_j^T}{N-1}. \quad (4.19)$$

4.2.3 Dynamisk utvikling av feil-statistikk

Bemerkning 4.2.1. Ligning (4.1) kan skrives som en stokastisk modell

$$d\Psi = \mathbf{g}(\Psi, \mathbf{u})dt + \mathbf{h}(\Psi, \mathbf{u})d\mathbf{q}. \quad (4.20)$$

Utviklingsligning for medlemmene av ensemble

Medlemmene av ensemblet utvikles ifølge modelldynamikken

$$\begin{aligned} d\Psi &= \mathbf{g}(\Psi, \mathbf{u})dt + \mathbf{h}d\Psi, \\ d\mathbf{u} &= 0. \end{aligned} \quad (4.21)$$

Utviklingsligning for pdf (tettheten)

Sannsynlighetstettheten f nå utvikles i henhold til Kolmogorovs ligning

$$\frac{\partial f}{\partial t} + \sum \frac{\partial(g_i f)}{\partial \psi_i} = \frac{1}{2} \sum_{i,j} \frac{\partial^2 f(\mathbf{hQh}^T)_{ij}}{\partial \psi_i \partial \psi_j}, \quad (4.22)$$

som kan løses ved bruk av Monte Carlo metoder.

4.2.4 Ensemble analyseligninger

Definisjon av de stokastiske målingene

Med $M = \sum_{j=1}^m m_j$, som er antallet av målingene definerer vi nå matrise \mathbf{D} fra (4.18) som

$$\mathbf{D} = \mathbf{d} + \mathbf{V} , \quad (4.23)$$

hvor $\mathbf{D} \in \mathbf{R}^{M \times N}$.

Analyseligninger

De oppdaterte ligningene for ensemble metoder

$$\mathbf{B}^a = \mathbf{B}^f + \mathbf{P}^e \mathbf{H}^T (\mathbf{H} \mathbf{R}^e \mathbf{H}^T + \mathbf{R})^{-1} (\mathbf{D} - \mathbf{H} \mathbf{B}) . \quad (4.24)$$

Ved hjelp av ensemble pertubasjon matrise vi kan omskrive (4.24) som

$$\begin{aligned} \mathbf{B}^a &= \mathbf{B}^f + \mathbf{B}^* (\mathbf{H} \mathbf{B}^*)^T (\mathbf{H} \mathbf{B}^* (\mathbf{H} \mathbf{B}^*)^T + \mathbf{R})^{-1} (\mathbf{D} - \mathbf{H} \mathbf{B}) \\ &\vdots \\ &= \mathbf{B} \mathbf{X} . \end{aligned} \quad (4.25)$$

Formelen til matrisen \mathbf{X} er spesifisert i det neste avsnittet.

4.2.5 Bemerkninger

Bemerkning 4.2.2. For ensemble metode den ensemble-kovariansmatrisen, $(\mathbf{P})^e$, er aldri beregnet, men blir indirekte brukt til å bestemme $\mathbf{S} \mathbf{S}^T$ ved

$$\mathbf{H} (\mathbf{P})^e \mathbf{H}^T = \mathbf{S} \mathbf{S}^T , \quad (4.26)$$

hvor $\mathbf{S} = \mathbf{H} \mathbf{B}^*$.

Bemerkning 4.2.3. Nøyaktigheten av analysen er bestemt av nøyaktigheten av \mathbf{X} og egenskaper av ensemble-feil-rom. For en lineær modell, hvilket som helst valg av \mathbf{X} , vil føre til en analyse som er også en løsning til modellen [20].

Bemerkning 4.2.4. Kombinerer vi simultan pdf for modell-utvikling (4.3) med Bayesiansk oppdatering ligning (4.6), får vi

$$f(\Psi_1, \dots, \Psi_k, \mathbf{u}, \Psi_0 \mid \mathbf{d}) \propto f(\Psi_1, \dots, \Psi_k, \mathbf{u}, \Psi_0) \prod_{j=1}^m f(\mathbf{d}_j \mid \Psi_{i(j)}, \mathbf{u}) . \quad (4.27)$$

4.3 Ensemble 'Smoother'

Smoothing (glatting) er ideelt for analysering av historisk-data, slik at tilstandsestimatet ved en gitt tid er bestemt av den påfølgende observasjonen. *Ensemble smoother* (ES) var presentert ved å bruke en *Bayesiansk* form, av Leeuwen og Evensen i 1996. ES-metoden forsøker å finne den ukjente modelltilstanden Ψ som minimerer den *a posteriori* feil-kovariansen og betraktes som en realisering av en random variabel. Denne variabelen beskrives av en sannsynlighetstetthet $f(\Psi)$ som antas å være *Gaussisk*.

ES beregner en tilnærmet oppdatering av Ligning (4.27) ved å benytte den lineære oppdatering Ligning (4.10). Den lineære analyse ligning for $\widetilde{\mathbf{B}}_k$ som var definert i Ligning (4.16) blir da

$$\widetilde{\mathbf{B}}_k^a = \widetilde{\mathbf{B}}_k + \mathbf{H}^T [\widetilde{\mathbf{P}}^e] (\mathbf{H}^T [\mathbf{H}] [\widetilde{\mathbf{P}}^e] + \mathbf{R}^e)^{-1} (\mathbf{D} - \mathbf{H} [\widetilde{\mathbf{B}}_k]) . \quad (4.28)$$

Her er

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_1 \\ \vdots \\ \mathbf{D}_m \end{pmatrix}, \mathbf{H} = \begin{pmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_m \end{pmatrix},$$

og

$$\mathbf{R}^e = \begin{pmatrix} R(t_{i(1)}) & & \\ & \ddots & \\ & & R(t_{i(m)}) \end{pmatrix}.$$

\mathbf{R}^e er en diagonal matrise. Med \mathbf{D} fra (4.23) definerer vi ensemble av innovasjonvektor $\Delta \mathbf{D}$ som

$$\begin{aligned} \mathbf{D}^* &= \Delta \mathbf{D}, \\ &= \mathbf{D} - \mathbf{H} [\widetilde{\mathbf{B}}_k], \end{aligned} \quad (4.29)$$

og målingene av ensemble pertubasjoner $\mathbf{S} \in \mathbf{R}^{M \times N}$ som

$$\begin{aligned} \mathbf{S} &= \mathbf{H} [\widetilde{\mathbf{B}}_k^*], \\ &= \mathbf{H} \cdot [\widetilde{\mathbf{B}}_k (\mathbf{I} - \mathbf{1}_N)]. \end{aligned} \quad (4.30)$$

og en matrise $\mathbf{C} \in \mathbf{R}^{M \times N}$ som

$$\mathbf{C} = \mathbf{S} \mathbf{S}^T + (N - 1) \mathbf{R}^e . \quad (4.31)$$

Bruk av (4.29)–(4.31) sammen med definisjonene for $(\tilde{\mathbf{P}})^e$ og \mathbf{R}^e , kan den analyserte ligningen (4.28) skrives som

$$\begin{aligned}\tilde{\mathbf{B}}_k^a &= \tilde{\mathbf{B}}_k + \tilde{\mathbf{B}}^* \mathbf{H}^T [\tilde{\mathbf{B}}^*] (\mathbf{H} [\tilde{\mathbf{B}}^*] \mathbf{H}^T [\tilde{\mathbf{B}}^*] + (N-1)\mathbf{R}^e)^{-1} \mathbf{D}^* \\ &= \tilde{\mathbf{B}}_k (\mathbf{I} + (\mathbf{I} - \mathbf{1}_N) \mathbf{S}^T \mathbf{C}^{-1} \mathbf{D}^*) \\ &= \tilde{\mathbf{B}}_k (\mathbf{I} + \mathbf{S} \mathbf{C}^{-1} \mathbf{D}^*).\end{aligned}\quad (4.32)$$

Her vi har brukt $\mathbf{1}_N \mathbf{S}^T \equiv 0$. Oppdaterte ensemble for ES kan derfor vurderes som en kombinasjon av predikerte ensemble medlemmer

$$\begin{aligned}\tilde{\mathbf{B}}_k^{ES} &= \tilde{\mathbf{B}}_k^f \cdot \mathbf{X}^{ES}, \\ \mathbf{X}^{ES} &= (\mathbf{I} + \mathbf{S} \mathbf{C}^{-1} \mathbf{D}^*).\end{aligned}\quad (4.33)$$

4.4 'Ensemble Kalman Smoother'(EnKS)

Ensemble Kalman smoother (EnKS) utnytter Monte-Carlo estimater av prediksjon-analyse-feil-kovarians som kreves for å beregne *Kalman Smoother gain*-matrisen. Dette brukes iterativt på en tidsserie av observasjoner. Den første iterasjonen svarer til en *Ensemble Kalman filter*-analyse som bare utnyttes målinger som er fra før eller på analysetidspunktet [24]. Prediksjonsteg i EnKS er identisk med prediksjonsteg i EnKF.

EnKS løser rekursivt (4.7-4.9) ved å benytte et ensemble representasjon for feil-statistikk.

4.4.1 Bayes med sekvensiell prosessering av data

Simultan pdf for modell-prediksjon til $t_{i(1)}$ er

$$f(\Psi_1, \dots, \Psi_{i(1)}, \mathbf{u}, \Psi_0 \mid \mathbf{d}_1) \propto f(\mathbf{u}) f(\Psi_0) \prod_{i=1}^{i(1)} f(\Psi_i \mid \Psi_{i-1}, \mathbf{u}). \quad (4.34)$$

Tilsvarende prosedyren som var brukt i ES, denne simultan pdf kan evalueres til å bruke et stort ensemble av realisering for hver av prior pdf og integrasjon av disse fremover i tid ved å benytte de stokastiske modell-ligninger. Den stokastiske integrasjonen fører til en ensemble representasjon av simultan pdf modell-løsning, $(\Psi_1, \dots, \Psi_{i(1)})$. På grunn av effektiv beregning av den simultan pdf betinget på de målingene \mathbf{d}_1 , med gitt ensemble-representasjon av (4.34). Det vil si at vi trenger å løse (4.7) som er omskrevet

$$f(\Psi_1, \dots, \Psi_{i(1)}, \mathbf{u}, \Psi_0 \mid \mathbf{d}_1) \propto f(\Psi_1, \dots, \Psi_{i(1)}, \mathbf{u}, \Psi_0) f(\mathbf{d}_1 \mid \Psi_{i(1)}, \mathbf{u}). \quad (4.35)$$

Denne ligningen gir oppdatering som baserer på det første sett av målingene ved $t_{i(1)}$. EnKS er likedan som ES, unntak at EnKS prosesserer målingene sekvensielt i tid.

4.4.2 Ensemble-prediksjonsoppdatering

Vi begynner fra det initialt ensemblet som er lagret i \mathbf{B}_0 . Det blir utført en forover stokastisk integrasjon av ensemblet til det første data settet er tilgjengelig. Dette gir ensemble prediksjon

$$\tilde{\mathbf{B}}_{i(1)}^f = \begin{pmatrix} \mathbf{B}_0 \\ \mathbf{B}_1^f \\ \vdots \\ \mathbf{B}_{i(1)}^f \end{pmatrix}.$$

4.4.3 Første ensemble-analyseoppdatering

Definerer vi matrisene \mathbf{D}_j^* , \mathbf{S}_j og \mathbf{C}_j som

$$\begin{aligned} \mathbf{D}_j^* &= \mathbf{D}_j - \mathbf{H}[\tilde{\mathbf{B}}_{i(j)}^f], \\ \mathbf{S}_j &= \mathbf{H}_j[\tilde{\mathbf{B}}_{i(j)}^{f*}], \\ \mathbf{C}_j &= \mathbf{S}_j \mathbf{S}_j^T + (N - 1) \mathbf{R}(t_{i(j)}). \end{aligned}$$

Her er $\mathbf{S}_j \in \mathbf{R}^{m_j \times N}$ og $\mathbf{C}_j \in \mathbf{R}^{m_j \times m_j}$. Vi antar at prediksjon-ensemble har Gaussiansk pdf. Ved å bruke ES-oppdatering (4.32) sammen med ligningen overfor, samt benytter det første sett av målingene \mathbf{d}_1 som løser (4.35), får vi

$$\begin{aligned} \tilde{\mathbf{B}}_{i(1)}^a &= \tilde{\mathbf{B}}_{i(1)}^f + \tilde{\mathbf{B}}_{i(1)}^{f*} \mathbf{H}_1^T [\tilde{\mathbf{B}}_{i(1)}^{f*}] \\ &\times (\mathbf{H}_1 [\tilde{\mathbf{B}}_{i(1)}^{f*}] \mathbf{H}_1^T [\tilde{\mathbf{B}}_{i(1)}^{f*}] + (N - 1) \mathbf{R}^e(t_{i(1)}))^{-1} \mathbf{D}_1^* \\ &= \tilde{\mathbf{B}}_{i(1)}^f + \tilde{\mathbf{B}}_{i(1)}^f (\mathbf{I} - \mathbf{1}_N) \mathbf{S}_1^T \mathbf{C}_1^{-1} \mathbf{D}_1^* \\ &= \tilde{\mathbf{B}}_{i(1)}^f (\mathbf{I} + \mathbf{S}_1^T \mathbf{C}_1^{-1} \mathbf{D}_1^*) \\ &= \tilde{\mathbf{B}}_{i(1)}^f \mathbf{X}_1. \end{aligned} \tag{4.36}$$

Den oppdateringen (4.36) er identisk til ES i tilfellet hvor $t \in [t_0, t_{i(1)}]$, og alle data er inneholdt i \mathbf{d}_1 .

4.4.4 Generelle ensemble-analyseoppdatering

Den generelle oppdatering for målingene ved tid $t_{i(j)}$ er

$$f(\Psi_1, \dots, \Psi_{i(j)}, \mathbf{u}, \Psi_0 \mid \mathbf{d}_1, \dots, \mathbf{d}_j) \propto f(\Psi_1, \dots, \Psi_{i(j)}, \mathbf{u}, \Psi_0 \mid \mathbf{d}_1, \dots, \mathbf{d}_{j-1}) f(\mathbf{d}_j \mid \Psi_{i(j)}, \mathbf{u}). \tag{4.37}$$

Ensemble-prediksjon-matrisen er definert nå

$$\tilde{\mathbf{B}}_{i(j)}^f = \begin{pmatrix} \mathbf{B}_{i(j-1)}^a \\ \mathbf{B}_{i(j-1)+1}^f \\ \vdots \\ \mathbf{B}_{i(j)}^f \end{pmatrix}, \quad (4.38)$$

hvor ensemble-prediksjonen $\mathbf{B}_{i(j-1)+1}^f, \dots, \mathbf{B}_{i(j)}^f$ er oppnådd ved integrasjon som starter fra det siste analyse resultat i $\tilde{\mathbf{B}}_{i(j-1)}^a$. Vi kan da beregne EnKS-oppdatering som er basert på (4.37) og bruker målingene ved tid $t_{i(j)}$

$$\tilde{\mathbf{B}}_{i(j)}^a = \tilde{\mathbf{B}}_{i(j)}^f \mathbf{X}_j, \quad (4.39)$$

hvor \mathbf{X}_j er definert

$$\mathbf{X}_j = \mathbf{I} + \mathbf{S}_j^T \mathbf{C}_j^{-1} \mathbf{D}_j^*. \quad (4.40)$$

Bemerkning 4.4.1. Ensemble-metodene introdusere en approksimasjon ved å bruke bare middelveidien og kovarians av prior simultan pdf for å beregne posteriori-ensemblet i (4.37). Derfor antar at de at prior simultan pdf er Gaussiansk når en beregner oppdateringer. Det betyr at EnKF og EnKS vil ikke gi korrekt svar hvis prior simultan pdf har ikke-Gaussiansk fordeling (se Kapittel 9 i [4]).

Bemerkning 4.4.2. Etter den siste måling tid er tilstands- og parameter-estimatene i EnKF og EnKS identiske [4].

Bemerkning 4.4.3. EnKS er den mest optimale smoother-løsning for de lineære problemene med Gaussisk fordeling [4].

Bemerkning 4.4.4. EnKS er den mest optimale smoother-løsning for de lineære problemene med Gaussisk fordeling. ES og EnKS gir identiske resultater for lineære modeller. EnKS er bedre enn ES med ikke-lineære modeller [4].

Kapittel 5

Algoritme

5.1 Generell algoritme for Kalman filteret

Algoritmene for det lineære og det ikke-lineære problemet er helt like. Lineære modeller er et spesialtilfelle av ikke-lineære modeller. La indeksen k betegne tidstegene i prosessen. Når modellfeilen, \mathbf{w}_k , er lik null, har vi en tilnærming for tilstanden, som er

$$\Psi_{k+1}^f \sim \mathbf{g}(\Psi_k^a, 0) = \mathbf{g}(\Psi_k^a) .$$

Det er vanlig å anta at kovariansmatrisen, \mathbf{V} , for målingsfeilen i Ligning (3.29) er lik identitetsmatrisen. Vi har derfor en tilnærming for målingene, gitt som

$$\mathbf{d}_k^f = \mathbf{h}(\Psi_k^f + \mathbf{v}_k) \sim \mathbf{h}(\Psi_k^f) + \mathbf{v}_k .$$

Initialisering ved $k = 0$

Vi antar at den første gjetningsverdien av tilstanden er Ψ_{init} , og at $\Psi_0 = \Psi_{init}$. I tillegg antar vi at initialverdien av a posteriori feilkovariansen, \mathbf{P}_0^a , er kjent og lik \mathbf{P}_0 . Det er ingen a priori feilkovarians, \mathbf{P}_k^f , ved $k = 0$.

Benyttelse av while-løkke for $k = 1 : k_{max}$

Prediksjonsestimater, Ψ_k^f , for tilstanden, overgangsmatrisen, \mathbf{A} , målingsmatrisen, \mathbf{H} , og a priori feil-kovariansmatrisen, \mathbf{P}_k^f , beregnes i rekkefølgen

$$\begin{aligned}\Psi_k^f &= \mathbf{g}(\Psi_{k-1}^a), \\ \mathbf{A}(l, s) &= \frac{\partial \mathbf{g}(l)}{\partial \Psi(s)}(\Psi_{k-1}^a), \\ \mathbf{H}(l, s) &= \frac{\partial \mathbf{h}(l)}{\partial \Psi(s)}(\Psi_k^f), \\ \mathbf{P}_k^f &= \mathbf{A}\mathbf{P}_{k-1}^a\mathbf{A}^T + \mathbf{Q}_{k-1}.\end{aligned}$$

Her er $\mathbf{P}_k^f = 1/(N_e - 1) \sum_{i=1}^{N_e} (\Psi_{k,i}^f - \widehat{\Psi}_k)(\Psi_{k,i}^f - \widehat{\Psi}_k)^T$ for ikke-lineære modeller, hvor $\widehat{\Psi}_k = \frac{1}{N_e} \sum_{i=1}^{N_e} \Psi_{k,i}^f$.

Antagelsen $\mathbf{V} = \mathbf{I}$ medfører at vi har samme formelen for å beregne \mathbf{K}_k , både for lineære og ikke-lineære modeller. Formelen er

$$\mathbf{K}_k = \mathbf{P}_k^f \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \mathbf{R}_k)^{-1}.$$

Her er \mathbf{R}_k kovariansen til målingsfeilen, og den er kjent.

Prediksjonsestimater, \mathbf{d}_k^f , for målingene for lineære og ikke-lineære modeller er henholdsvis

$$\mathbf{d}_k^f = \mathbf{H}\Psi_k^f + \mathbf{v}_k,$$

og

$$\mathbf{d}_k^f = \mathbf{h}(\Psi_k^f) + \mathbf{v}_k.$$

Målingsfunksjonen, \mathbf{h} , er en hvilken som helst lineær eller ikke-lineær funksjon. Typisk er \mathbf{h} en lineær funksjon gitt ved formelen $\mathbf{h}(\Psi) = \mathbf{H}\Psi$.

Vi beregner \mathbf{r}_k med formelen

$$\mathbf{r}_k = \mathbf{d}_k - \mathbf{d}_k^f,$$

hvor \mathbf{d}_k , som er målingene med usikkerhet, og er oppgitt. Da kan analyseestimatet, Ψ_k^a , for tilstanden, beregnes som

$$\Psi_k^a = \Psi_k^f + \mathbf{K}_k \mathbf{r}_k.$$

A posteriori feil-kovariansmatrisen, \mathbf{P}_k^a , blir da

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^f \mathbf{P}_k^a,$$

for lineære modeller, og

$$\mathbf{P}_k^a = 1/(N_e - 1) \sum_{i=1}^{N_e} (\Psi_{k,i}^a - \widehat{\Psi}_k)(\Psi_{k,i}^a - \widehat{\Psi}_k)^T,$$

for ikke-lineære modeller.

5.2 Grov algoritme for EnKF

I dette avsnittet oppretter vi en grov algoritme for å løse varmeLigningen som et invers-problem ved EnKF. Videre kan vi bruke algoritmen for å utvikle en grov EnKS-algoritme for varmeLigningen. Den to-dimensjonale varmeLigningen er

$$\frac{\partial \mathbf{p}}{\partial t} + \nabla \cdot (\mathbf{u} \nabla \mathbf{p}) = Q .$$

Her er trykket, $\mathbf{p} = \mathbf{p}(\mathbf{x}_1, \mathbf{x}_2, t)$, en dynamisk variabel. Permeabiliteten, $\mathbf{u} = \mathbf{u}(\mathbf{x}_1, \mathbf{x}_2)$, er en statisk parameter. Ved å bruke EnKF, kan vi estimere både \mathbf{p} og \mathbf{u} samtidig. Hvis vi løser varmeLigningen som et forover-problem (Kapittel 1) varierer permeabiliteten, \mathbf{u} , med posisjonen, men er konstant i tid. I EnKF, der varmeLigningen løses som et invers-problem, vil de estimerte verdiene for \mathbf{u} oppdateres ved hvert tidssteg. For n_{x_1} og n_{x_2} henholdsvis som antall celler i x -retning og y -retning, har vi vektorene for permeabiliteten og trykket ved tidspunkt k , gitt som

$$\mathbf{u}_k = (u_{1k}, u_{2k}, \dots, u_{n_{x_1} n_{x_2} k})^T ,$$

og

$$\mathbf{p}_k = (p_{1k}, p_{2k}, \dots, p_{n_{x_1} n_{x_2} k})^T .$$

Vi definerer en modelltilstand som

$$\Psi_k = \begin{pmatrix} \mathbf{u}_k \\ \mathbf{p}_k \end{pmatrix} .$$

Initialisering ved $k = 0$

Ved $k = 0$ har vi den analyserte verdien

$$\Psi_0^a = \Psi_{init} = \Psi_0 = \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{p}_0 \end{pmatrix} ,$$

hvor \mathbf{u}_0 og \mathbf{p}_0 henholdsvis er den første gjetningsverdien for permeabiliteten og trykket, og vi har at $\mathbf{u}_0^a = \mathbf{u}_0$ og $\mathbf{p}_0^a = \mathbf{p}_0$.

Benyttelse av while-løkke $k < k_{max}$

Herfra kjøres prosessen i samme rekkefølge som algoritmen i avsnitt 5.1, for modelltilstanden $\Psi(\mathbf{u}, \mathbf{p})$. Vi må først finne systemfunksjonen, \mathbf{g}_Ψ , og

målingsfunksjonen, \mathbf{h}_Ψ , for $\Psi(\mathbf{u}, \mathbf{p})$. Vi utfører ikke målinger for permeabiliteten. Vi antar derfor at målingsfunksjonen, \mathbf{h}_u , for permeabiliteten er lik null. Målingsmatrisen, \mathbf{H} , for trykket har formelen

$$\mathbf{H} = \begin{pmatrix} 0 & 1 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 \dots & 1 & 0 & 0 \end{pmatrix},$$

hvor enerene er betegnelser for 'steder', for eksempel brønner, der vi har målinger for trykket ved forskjellige tidspunkter. Som vi nevnte i KF-algoritmen, er målingsfunksjonen for trykket, $\mathbf{h}_p = \mathbf{H}\mathbf{p}$. Dette medfører da at målingsfunksjonen \mathbf{h}_Ψ blir

$$\mathbf{h}_\Psi = \begin{pmatrix} \mathbf{0} \\ \mathbf{h}_p \end{pmatrix}.$$

Hvis vi kaller systemfunksjonene for permeabiliteten og trykket, henholdsvis som \mathbf{g}_u og \mathbf{g}_p , da blir \mathbf{g}_Ψ for $\Psi(\mathbf{u}, \mathbf{p})$

$$\mathbf{g}_\Psi = \begin{pmatrix} \mathbf{g}_u \\ \mathbf{g}_p \end{pmatrix}.$$

Ved å benytte den endelig differansemetoden i Kapittel 1, fant vi en formel for å løse varmeLigningen som et forover-problem. Formelen er

$$\begin{aligned} \mathbf{p}_k^f &= \mathcal{T}^{-1} \cdot (\mathbf{p}_{k-1}^f + Q), \\ \mathbf{p}_k^f &= \mathbf{g}_p(\mathbf{p}_{k-1}^f). \end{aligned} \quad (5.1)$$

Her er \mathcal{T} en matrise, som er satt opp etter verdiene av permeabiliteten, og varierer i hvert tidsteg på grunn av verdiene til permeabiliteten. Massekildetettheten, Q , er kjent. Dessuten har vi at

$$\begin{aligned} \mathbf{p}_k^f &= \mathbf{g}_p(\mathbf{p}_{k-1}^a), \\ \mathbf{p}_{k-1}^a &= \mathbf{p}_{k-1}^f + \mathbf{K}_{k-1}(\mathbf{d}_{k-1} - \mathbf{H}\mathbf{p}_{k-1}^f). \end{aligned} \quad (5.2)$$

Her er \mathbf{p}_k^f fra Ligning (5.1) den samme som \mathbf{p}_k^f fra Ligning, (5.2). Vi har derfor

$$\mathbf{g}_p = \mathbf{g}_p(\mathbf{p}_{k-1}^f) = \mathcal{T}^{-1}(\mathbf{p}_{k-1}^f + Q). \quad (5.3)$$

Med $k = 0 : k_{max}$ kan vi beskrive relasjonen mellom \mathbf{u}_{k+1}^f , \mathbf{p}_{k+1}^f og \mathbf{u}_k^a , \mathbf{p}_k^a , som er vist nedenfor

$$\mathbf{u}_0^a, \mathbf{p}_0^a \xrightarrow{\text{predik}} \mathbf{u}_1^f, \mathbf{p}_1^f$$

$$\begin{array}{c}
\downarrow \textit{analyse} \\
\mathbf{u}_1^a, \mathbf{p}_1^a \xrightarrow{\textit{predik}} \mathbf{u}_2^f, \mathbf{p}_2^f \\
\downarrow \textit{analyse} \\
\mathbf{u}_2^a, \mathbf{p}_2^a \xrightarrow{\textit{predik}} \mathbf{u}_3^f, \mathbf{p}_3^f \\
\vdots \\
\downarrow \textit{analyse} \xrightarrow{\textit{predik}} \mathbf{u}_{n_k}^f, \mathbf{p}_{n_k}^f \\
\downarrow \textit{analyse} . \\
\mathbf{u}_{n_k}^a, \mathbf{p}_{n_k}^a
\end{array}$$

Relasjonen mellom prediksjonsverdier og analyseverdier, kan derfor skrives som $\mathbf{u}_{k+1}^f = \mathbf{u}_k^a$. Her ser vi at $\mathbf{g}_u = \mathbf{I}$.

Bemerkning 5.2.1. Prosedyren for KF og EnKF er den samme, men for EnKF blir denne prosedyren utført N ganger, hvor N er antall medlemmer av ensemblet. På grunn av dette kan ensemble feil-kovariansmatrisen, \mathbf{P}^e , beregnes med formelen

$$\begin{aligned}
\mathbf{B}^* &= \mathbf{B}(\mathbf{I} - \mathbf{1}_N), \\
\mathbf{P}^e &= \frac{(\mathbf{B}^*)(\mathbf{B}^*)^T}{N-1}.
\end{aligned} \tag{5.4}$$

Her er \mathbf{P}^e en $(2n \times 2n)$ -diagonalmatrise, hvor variansene er diagonalelementene.

5.3 Grov algoritme for KS

Med utgangspunktet i KF-algoritmen oppretter vi en grov algoritme for KS.

5.3.1 Forover rekursjon

Denne rekursjonen er helt lik KF, så analyseligningene blir derfor de samme som for KF

$$\begin{aligned}
\Psi_k^a &= \Psi_k^f + \mathbf{K}_k(\mathbf{d}_k - \mathbf{H}\Psi_k^f), \\
\mathbf{P}_k^a &= [\mathbf{I} - \mathbf{K}_k\mathbf{H}]\mathbf{P}_k^f.
\end{aligned}$$

5.3.2 Bakover rekursjon

Initialisering ved $k = \tau$

For $k = \tau$, fikk vi verdiene, Ψ_τ^a og \mathbf{P}_τ^f fra filteret (forover-rekursjonen). Disse verdiene er brukt som initialverdiene for bakover-rekursjonen.

Bruk av while løkke for $k = (\tau - 1) : 1$

Først beregner vi matrisen, \mathbf{L}_k , som er

$$\mathbf{L}_k = \mathbf{P}_k^a \mathbf{A}^T [\mathbf{P}_{k+1}^a]^{-1} .$$

Den første verdien for \mathbf{L}_k er

$$\mathbf{L}_{\tau-1} = \mathbf{P}_{\tau-1}^a \mathbf{A}^T [\mathbf{P}_\tau^a]^{-1} ,$$

hvor både $\mathbf{P}_{\tau-1}^a$ og \mathbf{P}_τ^a er kjente (fra KF). Så beregner vi den estimerte modelltilstanden, Ψ_k^s

$$\Psi_k^s = \Psi_k^a + \mathbf{L}_k [\Psi_{k+1}^s - \mathbf{A} \Psi_k^a] .$$

Den første verdien for Ψ_k^s er

$$\Psi_{\tau-1}^s = \Psi_{\tau-1}^a + \mathbf{L}_{\tau-1} [\Psi_\tau^s - \Psi_{\tau-1}^a] , \quad (5.5)$$

hvor $\Psi_\tau^s = \Psi_\tau^a$ (den siste verdien av KF er like smoother-verdien) og $\Psi_{(\tau-1)}^a$ beregnes med formelen som er fra forover-problem.

Feil-kovariansmatrisen, \mathbf{P}_k^s , blir da

$$\mathbf{P}_k^s = \mathbf{P}_k^a + \mathbf{L}_k [\mathbf{P}_{k+1}^s - \mathbf{P}_{k+1}^a] \mathbf{L}_k^T .$$

Den første verdien for \mathbf{P}_k^s er

$$\mathbf{P}_{\tau-1}^s = \mathbf{P}_{\tau-1}^a + \mathbf{L}_{\tau-1} [\mathbf{P}_\tau^s - \mathbf{P}_\tau^a] ,$$

hvor $\mathbf{P}_\tau^s = \mathbf{P}_\tau^a$ og \mathbf{P}_τ^a er kjent fra KF.

Med antall datasett (målinger), $\tau = 0 : n_\tau$, kan relasjonen mellom KF-løsningene og KS-løsningene ifølge økende verdier for τ beskrives som

$$\begin{aligned} \tau = 1 : & \mathbf{u}_0^a, \mathbf{p}_0^a \xrightarrow{\text{predik}} \mathbf{u}_1^f, \mathbf{p}_1^f \\ & \quad \downarrow \text{analyse} \\ \tau = 2 : & \mathbf{u}_1^a = \mathbf{u}_1^s, \mathbf{p}_1^a = \mathbf{p}_1^s \xrightarrow{\text{predik}} \mathbf{u}_2^f, \mathbf{p}_2^f \\ & \quad \quad \quad \downarrow \text{analyse} \end{aligned}$$

$$\tau = 3 : \quad \mathbf{u}_1^s \mathbf{p}_1^s \longleftarrow \quad \mathbf{u}_2^a = \mathbf{u}_2^s, \mathbf{p}_2^a = \mathbf{p}_2^s \xrightarrow{\text{predik}} \mathbf{u}_3^f = \mathbf{u}_3^s, \mathbf{p}_3^f = \mathbf{p}_3^s$$

$\downarrow \text{analyse}$

\vdots

Her ser vi at de analyserte verdiene er lik smoother-verdiene ved $k = \tau$. Det vil si at

$$\mathbf{u}_k^a = \mathbf{u}_{kk}^s, \quad \mathbf{p}_k^a = \mathbf{p}_{kk}^s. \quad (5.6)$$

De tidligere analyserte verdiene fra filteret endrer seg ikke etter økende antall målinger. Med dette som utgangspunkt kan vi lagre filter-løsning for \mathbf{u} og \mathbf{p} i matrisen, $\mathbf{B} \in \mathcal{R}^{n_\tau \times n_{x_1} n_{x_2}}$, som er

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_{per} \\ \mathbf{B}_{trykk} \end{pmatrix}.$$

For $\tau = 0 : n_\tau$ blir matrisen \mathbf{B}

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_{n_\tau} \end{pmatrix},$$

hvor

$$\begin{aligned}
\mathbf{B}_0 &= \begin{pmatrix} \mathbf{u}_{0(1:n_{x_1} \cdot n_{x_2})}^a \\ \mathbf{p}_{0(1:n_{x_1} \cdot n_{x_2})}^a \end{pmatrix} = \begin{pmatrix} u_{0(1)}^a \\ \vdots \\ u_{0(n_{x_1} \cdot n_{x_2})}^a \\ p_{0(1)}^a \\ \vdots \\ p_{0(n_{x_1} \cdot n_{x_2})}^a \end{pmatrix}, \\
\mathbf{B}_1 &= (\mathbf{B}_{01} \ \mathbf{B}_{11}) = \begin{pmatrix} u_{0(1)}^a & u_{1(1)}^a \\ \vdots & \vdots \\ u_{0(n_{x_1} \cdot n_{x_2})}^a & u_{1(n_{x_1} \cdot n_{x_2})}^a \\ p_{0(1)}^a & p_{1(1)}^a \\ \vdots & \vdots \\ p_{0(n_{x_1} \cdot n_{x_2})}^a & p_{1(n_{x_1} \cdot n_{x_2})}^a \end{pmatrix}, \\
&\vdots \\
\mathbf{B}_{n_\tau} &= (\mathbf{B}_{01} \ \mathbf{B}_{11} \ \dots \ \mathbf{B}_{n_\tau n_{x_1} \cdot n_{x_2}}) \\
&= \begin{pmatrix} u_{0(1)}^a & u_{1(1)}^a & \dots & u_{n_\tau(1)}^a \\ \vdots & \vdots & \dots & \vdots \\ u_{0(n_{x_1} \cdot n_{x_2})}^a & u_{1(n_{x_1} \cdot n_{x_2})}^a & \dots & u_{n_\tau(n_{x_1} \cdot n_{x_2})}^a \\ p_{0(1)}^a & p_{1(1)}^a & \dots & p_{n_\tau(1)}^a \\ \vdots & \vdots & \dots & \vdots \\ p_{0(n_{x_1} \cdot n_{x_2})}^a & p_{1(n_{x_1} \cdot n_{x_2})}^a & \dots & p_{n_\tau(n_{x_1} \cdot n_{x_2})}^a \end{pmatrix}. \quad (5.7)
\end{aligned}$$

De forrige smoother-verdiene er tidoppdatert etter økende antall målinger. Vi har derfor mange forskjellige verdier i vektor form for smoother-løsninger ved et tidsteg. For $k = 0 : k_{max}$ og $\tau = 0 : n_\tau$, lagrer vi både filter- og smoother-løsningene i matrisen, $\mathbf{B}_{k\tau}^s \in \mathcal{R}^{n_\tau \times 2n_{x_1} n_{x_2} \times n_\tau}$, som inneholder

smoother-løsningene

$$\mathbf{B}_{k\tau}^s = \begin{pmatrix} \mathbf{u}_{00}^a & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{p}_{00}^a & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{u}_{01}^s & \mathbf{u}_{11}^a & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{p}_{01}^s & \mathbf{p}_{11}^a & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{u}_{02}^s & \mathbf{u}_{12}^s & \mathbf{u}_{22}^a & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{p}_{02}^s & \mathbf{p}_{12}^s & \mathbf{p}_{22}^a & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{u}_{03}^s & \mathbf{u}_{13}^s & \mathbf{u}_{23}^s & \mathbf{u}_{33}^a & \dots & \mathbf{0} \\ \mathbf{p}_{03}^s & \mathbf{p}_{13}^s & \mathbf{p}_{23}^s & \mathbf{p}_{33}^a & \dots & \mathbf{0} \\ \mathbf{u}_{04}^s & \mathbf{u}_{14}^s & \mathbf{u}_{24}^s & \mathbf{u}_{34}^s & \dots & \mathbf{0} \\ \mathbf{p}_{04}^s & \mathbf{p}_{14}^s & \mathbf{p}_{24}^s & \mathbf{p}_{34}^s & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ \mathbf{u}_{0n_\tau}^s & \mathbf{u}_{1n_\tau}^s & \mathbf{u}_{2n_\tau}^s & \mathbf{u}_{3n_\tau}^s & \dots & \mathbf{u}_{n_\tau n_\tau}^a \\ \mathbf{p}_{0n_\tau}^s & \mathbf{p}_{1n_\tau}^s & \mathbf{p}_{2n_\tau}^s & \mathbf{p}_{3n_\tau}^s & \dots & \mathbf{p}_{n_\tau n_\tau}^a \end{pmatrix}.$$

På grunn av Ligning (5.6) kan matrisen, $\mathbf{B}_{k\tau}^s$, skrives som

$$\mathbf{B}_{k\tau}^s = \begin{pmatrix} \mathbf{u}_{00}^s & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{p}_{00}^s & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{u}_{01}^s & \mathbf{u}_{11}^s & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{p}_{01}^s & \mathbf{p}_{11}^s & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{u}_{02}^s & \mathbf{u}_{12}^s & \mathbf{u}_{22}^s & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{p}_{02}^s & \mathbf{p}_{12}^s & \mathbf{p}_{22}^s & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{u}_{03}^s & \mathbf{u}_{13}^s & \mathbf{u}_{23}^s & \mathbf{u}_{33}^s & \dots & \mathbf{0} \\ \mathbf{p}_{03}^s & \mathbf{p}_{13}^s & \mathbf{p}_{23}^s & \mathbf{p}_{33}^s & \dots & \mathbf{0} \\ \mathbf{u}_{04}^s & \mathbf{u}_{14}^s & \mathbf{u}_{24}^s & \mathbf{u}_{34}^s & \dots & \mathbf{0} \\ \mathbf{p}_{04}^s & \mathbf{p}_{14}^s & \mathbf{p}_{24}^s & \mathbf{p}_{34}^s & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ \mathbf{u}_{0n_\tau}^s & \mathbf{u}_{1n_\tau}^s & \mathbf{u}_{2n_\tau}^s & \mathbf{u}_{3n_\tau}^s & \dots & \mathbf{u}_{n_\tau n_\tau}^s \\ \mathbf{p}_{0n_\tau}^s & \mathbf{p}_{1n_\tau}^s & \mathbf{p}_{2n_\tau}^s & \mathbf{p}_{3n_\tau}^s & \dots & \mathbf{p}_{n_\tau n_\tau}^s \end{pmatrix}.$$

Her ser vi at $\mathbf{B}_{k\tau}^s$ er en nedre triangulær matrise.

5.3.3 Utledning av formelen for oppdatering

Matrisen $\mathbf{B}_{k\tau}^s = \mathbf{B}_{(0:n_\tau)n_\tau}^s$ lagrer de beste smoother-løsningene når det n_τ -te datasettet er benyttet. Fra dette utleder vi en enkel formel for å oppdatere de beste smoother-løsningene når det $(n_\tau + 1)$ -te datasettet er tilgjengelig.

Først oppdaterer vi de beste smoother-løsningene for $k = 0 : n_\tau$. Det vil si at vi oppdaterer de verdiene i den siste rekken i $\mathbf{B}_{(0:n_\tau)n_\tau}^s$, fordi disse

løsningene er de beste smoother-løsningene etter det n_τ -te datasettet som er tilgjengelig og brukt. Matrisen, $\mathbf{B}_{(0:n_\tau)n_\tau}^s$, blir

$$\mathbf{B}_{kn_\tau}^s = \begin{pmatrix} \mathbf{u}_{0n_\tau^s} & \mathbf{u}_{1n_\tau^s} & \cdots & \mathbf{u}_{(n_\tau-1)n_\tau^s} & \mathbf{u}_{n_\tau n_\tau^s} & \mathbf{0} \\ \mathbf{p}_{0n_\tau^s} & \mathbf{p}_{1n_\tau^s} & \cdots & \mathbf{p}_{(n_\tau-1)n_\tau^s} & \mathbf{p}_{n_\tau n_\tau^s} & \mathbf{0} \end{pmatrix}.$$

Det neste steget er at vi benytter formelen for endelig differansemetode til å løse varmeLigningen, slik at vi får prediksjons-verdier ved tidsteg $k = (n_\tau + 1)$ som er $\mathbf{p}_{n_\tau(n_\tau+1)}^f$ og $\mathbf{u}_{n_\tau(n_\tau+1)}^f$. Formelen gitt ved Ligning (1.34) fra Kapittel 1 er

$$\mathbf{p}_{k+1}^f = \mathcal{T}^{-1} \cdot (\mathbf{p}_k^f + Q), \quad (5.8)$$

Her er $\mathbf{p}_{k+1}^f = \mathbf{p}_{(n_\tau+1)(n_\tau)}^f$ og $\mathbf{p}_k = \mathbf{p}_{n_\tau n_\tau}^s$. Matrisen, \mathcal{T} , er en tridiagonal matrise som er opprettet av verdiene til permeabiliteten i forover-modellen. Vi løser forover-problemet uten å bruke informasjonen av $(n_\tau + 1)$ -te datasett, men beregner prediksjonsverdiene for tilstanden akkurat ved tidsteg $k = (n_\tau + 1)$.

$$\begin{aligned} \mathbf{p}_{(n_\tau+1)(n_\tau)}^f &= \mathcal{T}^{-1}(\mathbf{p}_{n_\tau n_\tau}^s + Q_i) \\ &= \mathcal{T}^{-1}(\mathbf{p}_{n_\tau n_\tau}^a + Q_i). \end{aligned} \quad (5.9)$$

Alle smoother-verdiene for $k = 0 : n_\tau$, blir nå prediksjon-verdier. Det vil si at

$$\begin{aligned} \mathbf{p}_{(0:n_\tau)n_\tau}^s &= \mathbf{p}_{(0:n_\tau)n_\tau}^f, \\ \mathbf{u}_{(0:n_\tau)n_\tau}^s &= \mathbf{u}_{(0:n_\tau)n_\tau}^f. \end{aligned}$$

Dette fører til

$$\mathbf{B}_{kn_\tau}^s = \begin{pmatrix} \mathbf{u}_{0n_\tau^s} & \mathbf{u}_{1n_\tau^s} & \cdots & \mathbf{u}_{(n_\tau-1)n_\tau^s} & \mathbf{u}_{n_\tau n_\tau^s} & \mathbf{u}_{(n_\tau+1)(n_\tau)}^f \\ \mathbf{p}_{0n_\tau^s} & \mathbf{p}_{1n_\tau^s} & \cdots & \mathbf{p}_{(n_\tau-1)n_\tau^s} & \mathbf{p}_{n_\tau n_\tau^s} & \mathbf{p}_{(n_\tau+1)(n_\tau)}^f \end{pmatrix}.$$

Nå oppdaterer vi de beste smoother-løsningene etter at $(n_\tau + 1)$ -te datasett er brukt, ved å benytte formelen

$$\begin{aligned} \mathbf{p}_{(n_\tau+1)(n_\tau+1)}^s &= \mathbf{p}_{(n_\tau+1)(n_\tau+1)}^a \\ &= \mathbf{p}_{(n_\tau+1)(n_\tau)}^f + \mathbf{K}_{(n_\tau+1)(n_\tau+1)}(\mathbf{d}_{(n_\tau+1)(n_\tau+1)} - \mathbf{H}\mathbf{p}_{(n_\tau+1)(n_\tau)}^f). \end{aligned} \quad (5.10)$$

De neste smoother-verdiene (i bakover prosessen) er beregnet med en følgende formel

$$\begin{aligned} \mathbf{p}_{n_\tau(n_\tau+1)}^s &= \mathbf{p}_{n_\tau n_\tau}^a + \mathbf{L}_{n_\tau}[\mathbf{p}_{(n_\tau+1)(n_\tau+1)}^s - \mathbf{A}\mathbf{p}_{(n_\tau)(n_\tau)}^a] \\ &= \mathbf{p}_{n_\tau n_\tau}^s + \mathbf{L}_{n_\tau}[\mathbf{p}_{(n_\tau+1)(n_\tau+1)}^s - \mathbf{A}\mathbf{p}_{(n_\tau)(n_\tau)}^s]. \end{aligned} \quad (5.11)$$

Vi får de beste smoother-løsningene når $(n_\tau + 1)$ -te datasett er tilgjengelig og brukt som

$$\mathbf{B}_{kn_\tau}^s = \begin{pmatrix} \mathbf{u}_{0n_\tau^s} & \mathbf{u}_{1n_\tau^s} & \cdots & \mathbf{u}_{(n_\tau-1)n_\tau^s} & \mathbf{u}_{n_\tau n_\tau^s} & \mathbf{u}_{(n_\tau+1)(n_\tau+1)}^s \\ \mathbf{p}_{0n_\tau^s} & \mathbf{p}_{1n_\tau^s} & \cdots & \mathbf{p}_{(n_\tau-1)n_\tau^s} & \mathbf{p}_{n_\tau n_\tau^s} & \mathbf{p}_{(n_\tau+1)(n_\tau+1)}^s \end{pmatrix}.$$

Fra (5.9), (5.10) og (5.11) får vi

$$\begin{aligned} \mathbf{p}_{(n_\tau+1)(n_\tau)}^f &= \mathcal{L}_1(\mathbf{p}_{n_\tau n_\tau}^s) \\ \mathbf{p}_{(n_\tau+1)(n_\tau+1)}^s &= \mathcal{L}_2(\mathbf{p}_{(n_\tau+1)(n_\tau)}^f) \\ \mathbf{p}_{n_\tau(n_\tau+1)}^s &= \mathcal{L}_3(\mathbf{p}_{(n_\tau+1)(n_\tau+1)}^s, \mathbf{p}_{n_\tau n_\tau}^s), \end{aligned} \quad (5.12)$$

hvor \mathcal{L}_1 , \mathcal{L}_2 og \mathcal{L}_3 er lineære funksjoner. Vi kan beskrive Ligningsett (5.12) som

$$\begin{array}{ccccc} \mathbf{p}_{n_\tau n_\tau}^s & \xrightarrow{\mathcal{L}_1} & \mathbf{p}_{(n_\tau+1)(n_\tau)}^f & \xrightarrow{\mathcal{L}_2} & \mathbf{p}_{(n_\tau+1)(n_\tau+1)}^s \\ & \searrow \mathcal{L}_3 & & & \swarrow \mathcal{L}_3 \\ & & \mathbf{p}_{n_\tau(n_\tau+1)}^s & & \end{array}.$$

Vi kan derfor si at de nest beste smoother-løsningene som er lagret i matrisen, $\mathbf{B}_{(0:(n_\tau+1))(n_\tau+1)}^s$, er en lineær kombinasjon av de smoother-verdiene i form av kolonne-vektorene, som er

$$\mathbf{p}_{0(n_\tau+1)}^s, \mathbf{p}_{1(n_\tau+1)}^s, \dots, \mathbf{p}_{(n_\tau+1)(n_\tau+1)}^s.$$

Det er tilsvarende for \mathbf{u} . Dette medfører at vi kan oppdatere $\mathbf{B}_{(0:n_\tau)(n_\tau+1)}^s$ med en formel som er

$$\begin{aligned} \mathbf{B}_{(0:(n_\tau))(n_\tau+1)}^s &= \mathbf{B}_{(0:(n_\tau))(n_\tau+1)}^a \\ &= \mathbf{B}_{(0:n_\tau)(n_\tau+1)}^f \mathbf{X}_{n_\tau}. \end{aligned}$$

Her er \mathbf{X} definert som i Ligning (4.33).

5.3.4 Prosessens skjema

For $n_k = n_\tau = 0$, har vi initial-verdier

$$\begin{pmatrix} \mathbf{u}_{0(1:n_{x_1} \cdot n_{x_2})}^a \\ \mathbf{p}_{0(1:n_{x_1} \cdot n_{x_2})}^a \end{pmatrix} = \begin{pmatrix} u_{0(1)}^a \\ \vdots \\ u_{0(n_{x_1} \cdot n_{x_2})}^a \\ p_{0(1)}^a \\ \vdots \\ p_{0(n_{x_1} \cdot n_{x_2})}^a \end{pmatrix},$$

og

$$\begin{aligned}
 \mathbf{P}_k^a &= \mathbf{P}_0^a \\
 &= \mathbf{P}_{init} . \\
 \Psi_{init} &\xrightarrow{\text{forover}} \Psi_{(n_\tau+1)(n_\tau)} \\
 &= \Psi_{10}^f \\
 \mathbf{B}_{(0:(n_\tau+1))(n_\tau+1)}^f &= \Psi_{10}^f \\
 \mathbf{B}_{(0:(n_\tau+1))(n_\tau+1)}^s &= \mathbf{B}_{(0:(n_\tau+1))(n_\tau+1)}^f \mathbf{X} .
 \end{aligned}$$

For $n_k = n_\tau \geq 1$

$$\begin{aligned}
 \Psi_{(n_\tau+1)(n_\tau)}^f &\xrightarrow{\text{forover}} \Psi_{(n_\tau+1)(n_\tau+1)}^f , \\
 \mathbf{B}_{(0:(n_\tau+1))(n_\tau+1)}^f &= [\mathbf{B}_{(0:n_\tau)(n_\tau)}^a, \Psi_{(n_\tau+1)(n_\tau+1)}^f] \\
 \mathbf{B}_{(0:(n_\tau+1))(n_\tau+1)}^s &= \mathbf{B}_{(0:(n_\tau+1))(n_\tau+1)}^f \mathbf{X} .
 \end{aligned}$$

5.4 Grov algoritme for EnKS

Hovedarbeidet i denne oppgaven er å benytte EnKS for å estimere \mathbf{p} og \mathbf{u} samtidig, frem til vi får fornuftige løsninger for begge.

5.4.1 Permeabilitet

Cholesky-faktorisering

Hvis matrisen, \mathbf{C} , er reell, symmetrisk ($\mathbf{C} = \mathbf{C}^T$) og positiv definert ($\mathbf{y}\mathbf{C}\mathbf{y}^T > 0 \forall \mathbf{y} \neq 0$) da har \mathbf{C} en entydig faktorisering, $\mathbf{C} = \mathbf{J}\mathbf{J}^T$, der \mathbf{J} er en nedre triangulær matrise med positiv diagonal.

Beregning av permeabiliteten

Inndata som her benyttes i simuleringen er listet i Tillegg B.1.3.

I vår oppgave er de sanne og de initiale verdiene til permeabiliteten beregnet ved hjelp av en multivariat vektor, \mathbf{x} , som har normalfordeling. Vi finner \mathbf{x} ved å først beregne kovariansmatrisen, \mathbf{C} . Vi velger standardavvik, $\sigma = 0.5$. Variansmatrisen, \mathbf{U} , er en diagonalmatrise der alle diagonalelementene er varianser, og lik $\sqrt{\sigma}$. Formelen for å beregne \mathbf{C} er

$$\mathbf{C} = \mathbf{U} \cdot \mathbf{M} \cdot \mathbf{U} ,$$

hvor \mathbf{M} er en korrelasjonsmatrise.

Det neste steget er å benytte Cholesky faktorisering til å finne matrisen, \mathbf{J} . Forventningsverdien, $\boldsymbol{\mu}$, er en deterministisk (forutsigbar) variabel. Det betyr at den ikke har usikkerhet. Dette fører til at $Cov(\boldsymbol{\mu}) = 0$. Vi lar nå \mathbf{z} være en vektor av n uavhengige tilfeldige tall, som har normalfordeling med forventningsverdien lik 0 og kovarians lik 1. I tillegg er z_i og z_j uavhengige av hverandre, derfor har vi $Cov(z_i, z_j) = 0$. Dette medfører at $Cov(\mathbf{z})$ er en diagonal matrise med diagonal elementer er lik 1. Det vil si at $Cov(\mathbf{z})$ er en identitets matrise. Ved å la $\mathbf{Y} = \mathbf{Jz}$ og $\mathcal{A} = \mathbf{J}$ så benytter vi Teorem 1 fra Kapittel 2. Da får vi

$$\begin{aligned} Cov(\mathbf{x}) &= Cov(\boldsymbol{\mu} + \mathbf{Jz}) = \mathbf{J}\mathbf{J}^T = \mathbf{C} , \\ \mathbf{x} &= \boldsymbol{\mu} + \mathbf{Jz} . \end{aligned}$$

Forskjellige verdier av den stokastiske variabelen, \mathbf{z} , fører til forskjellige verdier for \mathbf{x} , så også for permeabiliteten. Med N som er antall medlemmer av ensemblet, og $n = n_i \cdot n_j$ antall celler i vår reservoar-modell, blir permeabiliteten, $perm = \mathbf{u}$, beregnet med formelen

$$perm(i, j) = 10^{x^{(i)}} , \quad (5.13)$$

hvor $i = 1 : n$, $j = 1 : N + 1$.

Resultatet for denne prosessen er at vi får ut 2 verdi-filer for permeabiliteten. Den ene filen inneholder $(n \times N)$ - matrisen eller de N settene av permeabilitetsverdien som vi skal bruke videre som initialverdier i forovermodellen. Den andre filen inneholder et sett av permeabilitetsverdier som vi har valgt for å være 'de sanne verdiene'.

I Ligning (5.13) er x normalfordelt, derfor er $\log(perm)$ normalfordelt. Dette medfører at permeabiliteten ($perm$) er logaritme-normalfordelt. Vi velger å bruke $\log(perm)$ på grunn av at vi vil ha positive verdier uansett hvor liten permeabilitetsverdien er (logaritmefunksjonen er en økende funksjon med positive verdier). I tillegg, for logaritmefunksjonen er endringer av verdiene små.

5.4.2 Grov algoritme for EnKS

Vi beskriver her en sekvens som forteller hvordan vi oppdaterer verdiene for både permeabilitet og trykk ved EnKS-metoden. Her er inndata som benyttes i simuleringene, listet i Tillegg B.1.1.

Beregning av matrisen \mathbf{X}

Som vi nevnte tidligere, danner EnKF a prior for EnKS. Det vil si at foroverrekursjonen for EnKS er helt lik EnKF. Analysesteget for EnKS er bestemt av matrisen, \mathbf{X} , og \mathbf{B} er beregnet med formlene

$$\begin{aligned}
 \mathbf{BB} &= \mathbf{B}f1(\mathbf{I} - \mathbf{1}_N) \\
 \mathbf{S} &= \mathbf{H} \cdot \mathbf{BB} , \\
 \mathbf{Y} &= \mathbf{SS}^T + (N - 1)\mathbf{R}^e , \\
 \Delta\mathbf{D} &= \mathbf{DD} - (\mathbf{HB}f1) , \\
 \mathbf{X} &= (\mathbf{I} + \mathbf{S}^T\mathbf{Y}^{-1}\Delta\mathbf{D}) , \\
 \mathbf{B} &= \mathbf{B}f1\mathbf{X} .
 \end{aligned} \tag{5.14}$$

Her er \mathbf{I} ($N \times N$)-identitetsmatrisen, $\mathbf{B}f1$ er en ($2n \times N$)-matrise som oppdaterer prediksjonsverdiene til både permeabiliteten og trykket ved nåværende tidsteg, k . Dimensjonen til matrisen \mathbf{BB} blir da ($2n \times N$). Matrisen \mathbf{H} er en ($m \times 2n$)-matrise med m er antallet posisjoner der kilden er ulike null. På grunn av \mathbf{BB} og \mathbf{H} er \mathbf{S} en ($m \times N$)-matrise. Matrisen, \mathbf{R}^e , er en ($m \times m$)-matrise. Matrisen \mathbf{C} er derfor en ($m \times m$)-matrise. Matrisen, \mathbf{DD} , er en ($m \times N$)-matrise som oppdaterer datasettene for forskjellige verdier av kk som går fra 1 til k . Matrisen, \mathbf{DD} , endrer seg på grunn av de tilfeldige verdiene til usikkerhetene i målingene, ($\sigma_d * randn$), som vi adderer til de sanne data, \mathbf{d} . Innovasjonsmatrisen, $\Delta\mathbf{D}$, blir en ($m \times N$)-matrise, som endrer seg etter kk , og \mathbf{X} blir derfor en ($N \times N$)-matrise som endrer seg etter kk . Matrisen \mathbf{B} oppdaterer analyseverdiene til både permeabiliteten og trykket ved nåværende tidsteg k . På grunn av dimensjonene til $\mathbf{B}f1$ og \mathbf{X} , har \mathbf{B} dimensjonen, ($2n \times N$), ved hvert tidsteg. Matrisen \mathbf{X} er beregnet med utgangspunktet i datasett for trykk og vi utfører ikke målinger for permeabiliteten, men bruker samme \mathbf{X} til å oppnå analyseverdiene for permeabiliteten.

Oppdatering av matrisen \mathbf{B}

Vi ønsker å oppdatere matrisen \mathbf{B} i Ligning (5.14) ved forskjellige tidsteg $k = 1 : k_{max}$.

Fra Ligning (4.38), ser vi at ensemble-prediksjonsmatrisen, $\tilde{\mathbf{B}}_k^f$, for forskjellige tidsteg ligger under hverandre. Vi antar at vi har målinger i alle

tidspunktene. Det vil si at $i(j) = j = k$. Vi kan omskrive Ligning (4.38) som

$$\begin{aligned}
\tilde{\mathbf{B}}_k^f &= \begin{pmatrix} \mathbf{B}_{(k-1)}^a \\ \mathbf{B}_k^f \end{pmatrix} \\
&= \begin{pmatrix} \tilde{\mathbf{B}}_{k-1}^f \cdot \mathbf{X}_{k-1} \\ \mathbf{B}_k^f \end{pmatrix} \\
&= \begin{pmatrix} \left(\begin{pmatrix} \tilde{\mathbf{B}}_{k-2}^a \\ \mathbf{B}_{k-1}^f \end{pmatrix} \cdot \mathbf{X}_{k-1} \right) \\ \mathbf{B}_k^f \end{pmatrix} \\
&= \begin{pmatrix} \tilde{\mathbf{B}}_{k-2}^a \cdot \mathbf{X}_{k-1} \\ \mathbf{B}_{k-1}^f \cdot \mathbf{X}_{k-1} \\ \mathbf{B}_k^f \end{pmatrix} \\
&= \begin{pmatrix} \tilde{\mathbf{B}}_{k-2}^f \cdot \mathbf{X}_{k-1} \\ \mathbf{B}_{k-1}^a \\ \mathbf{B}_k^f \end{pmatrix} \\
&\vdots \\
&= \begin{pmatrix} \mathbf{B}_1^a \\ \mathbf{B}_2^a \\ \vdots \\ \mathbf{B}_{k-1}^a \\ \mathbf{B}_k^f \end{pmatrix}. \tag{5.15}
\end{aligned}$$

Vi lar derfor \mathbf{B} være en $((2n \cdot k_{max}) \times N)$ -matrise med alle elementer lik 0 ved $k = 0$. For $j = 1 : N$ er permeabiliteten, $\mathbf{u} = perm$, og trykke, \mathbf{p} , tatt ut for en gitt k som er

$$\begin{aligned}
\mathbf{p} &= \mathbf{B}(1 : n, j) \\
perm &= \mathbf{B}((n + 1) : 2n, j) \text{ for } k = 1 \\
\mathbf{p} &= \mathbf{B}(2n + 1 : 3n, j) \\
perm &= \mathbf{B}((3n + 1) : 4n, j) \text{ for } k = 2 \\
&\vdots \\
\mathbf{p} &= \mathbf{B}(2(k - 1)n + 1 : (2(k - 1)n + n), j) \\
perm &= \mathbf{B}((2(k - 1)n + (n + 1)) : (2(k - 1)n + 2n), j) \text{ for } k = k.
\end{aligned}$$

Prosessens flytskjema

Hele prosessen blir utført i rekkefølgen

$$\begin{array}{rcl}
 \mathbf{U}, \mathbf{M} & \longrightarrow & \mathbf{C}, \\
 \text{Cholesky-faktorisering} & \longrightarrow & \mathbf{J}, \\
 \mathbf{z}, \boldsymbol{\mu}, \mathbf{J} & \longrightarrow & \mathbf{x}, \\
 \mathbf{x} & \longrightarrow & \mathbf{u}_{init} \text{ og } \mathbf{u}_{sann}, \\
 \mathbf{u}_{init} & \xrightarrow{\text{forover-modell}} & d.dat, \mathbf{p}_{sann}, \\
 d.dat & \xrightarrow{\text{EnKS}} & \begin{pmatrix} \mathbf{p}_{1:kmax}^a \\ \mathbf{u}_{1:kmax}^a \end{pmatrix}. \quad (5.16)
 \end{array}$$

Her er \mathbf{u}_{init} en $(n \times N)$ -matrise og \mathbf{u}_{sann} er en n -dimensjonal vektor. Filen *d.dat* er en $(m \times k_{max})$ -matrise som inneholder de målingene etter at forovermodellen er kjørt fram til og med k_{max} tidsteget, \mathbf{p}_{sann} er en n -dimensjonal vektor av trykkverdier. Vi bruker denne vektoren som en fasit, slik at vi kan sammenligne med gjennomsnittsverdien av de N settene analyseverdier for trykk fra EnKS-metoden ved k_{max} tidsteget.

Verdiene \mathbf{p}^a og \mathbf{u}^a i Ligning (5.16) er henholdsvis analyseverdiene til trykket og permeabiliteten vi vil få etter at vi har kjørt EnKS til og med k_{max} tidsteget. Alle disse verdiene er oppdatert i matrisen, \mathbf{B} . For hele prosessen har vi derfor oppdatert alle EnKS-løsningene i en $((2n \cdot k_{max}) \times N)$ -matrise \mathbf{B} .

Bemerkning 5.4.1. Ensemble-feil-kovariansmatrisen, \mathbf{P}^e , for EnKS har samme formelen som for EnKF, men matrisens dimensjon er nå $((2n \cdot k_{max}) \times (2n \cdot k_{max}))$, istedenfor $(2n \times 2n)$.

Teoretisk analyse av metodens resultat

Dersom EnKS-metoden vår er god, vil gjennomsnittet av de N settene av analyseverdier vi får for \mathbf{p} ved k_{max} , være omtrent lik \mathbf{p}_{sann} som vi har fått fra forover-modellen.

Når det gjelder permeabiliteten forventer vi at gjennomsnittet av de N settene av estimerte verdier, \mathbf{u}_3 , som vi får fra EnKS ved k_{max} , i forhold til den sanne verdien, \mathbf{u}_1 , har minst mulig avvik.

De analyseverdiene for permeabiliteten og kovariansmatrisen \mathbf{P}^e som vi får etter EnKS-metoden, ved k_{max} , er

$$perm = \begin{pmatrix} u(1) \\ u(2) \\ \vdots \\ u(n) \end{pmatrix},$$

og

$$\tilde{\mathbf{P}}^e = \begin{bmatrix} \text{Var}(11) & \text{Cov}(12) & \dots & \text{Cov}(1n) \\ \text{Cov}(21) & \text{Var}(22) & \dots & \text{Cov}(2n) \\ \vdots & \vdots & \vdots & \vdots \\ \text{Cov}(n1) & \text{Cov}(n2) & \dots & \text{Var}(nn) \end{bmatrix} \quad (5.17)$$

Vi får en vektor \mathbf{s} som er

$$\begin{aligned} \mathbf{s} &= \text{diag}(\tilde{\mathbf{P}}^e) \\ &= \begin{pmatrix} \text{Var}(1) \\ \text{Var}(2) \\ \vdots \\ \text{Var}(n) \end{pmatrix}. \end{aligned} \quad (5.18)$$

For å sammenligne kan vi plote ($u1$, $u3$) i rommet. Målet for metoden er at plottet til $u1$ og plottet til $u3$ har lik form. Vi kan også plote variansen til initialpermeabiliteten, så sammeligne med plottet av variansen til permeabiliteten ved k_{max} fra EnKS-metoden. En detaljert analyse av resultatene vil finnes i det neste kapitlet.

Kapittel 6

Resultater fra EnKF- og EnKS-metode

I dette kapitlet diskuterer vi resultatene for permeabilitet og trykk som vi fikk fra våre EnKF- og EnKS-metodene.

6.1 Resultat fra forover-modellen

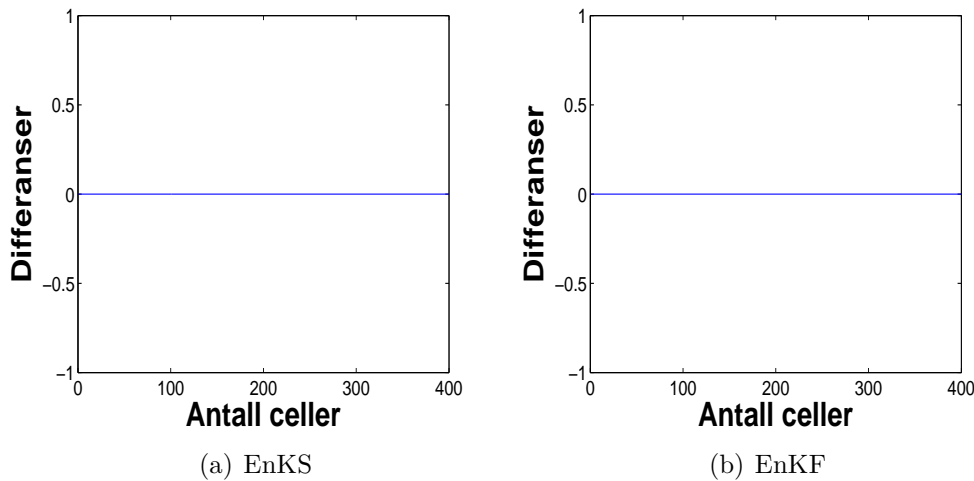
I forover-modellen er trykket beregnet ved å bruke initialverdier for permeabiliteten som vi har beregnet på forhånd. Fra starten har vi $m = 5$, det vil si at vi har 5 'steder', for eksempel brønner, der vi har målingene for trykket. Ved hvert tidssteg, k , som går fra 1 til k_{max} , i forover-metoden, får vi m nye målinger for trykket. Derfor får vi når kjører programmet til og med $k = k_{max}$, en fil som inneholder $k_{max} \times m$ målinger. Filen vi får, *d.dat*, kaller vi de 'sanne data' for trykket.

Vi lar $v = \sigma * randn$, hvor v er usikkerheten i målingen, $\sigma = 0.3$ og *randn* er betegnelsen for normalfordelt tilfeldig tall som har forventningsverdi lik 0 og standardavvik lik 1. Det vil si at vi har en usikkerhet rundt 0 som er -0.3 og 0.3 . Når vi legger denne usikkerheten til de sanne dataene, får vi det som vi betegner 'kunstige data', $\mathbf{D} \in \mathbf{R}^{m \times N \times k_{max}}$. For matrisen \mathbf{D} får vi matrisen $\mathbf{DD} \in \mathbf{R}^{m \times N}$. Dette gjøres i EnKF-metoden for hver av verdiene til k , og i EnKS-metoden for hver verdi av kk som går fra 1 til k . Dette er hovedforskjellen mellom EnKF og EnKS. Matrisen \mathbf{DD} brukes til å beregne matrisen \mathbf{X} i både EnKS- og EnKF-metodene.

6.2 Observasjoner for $k_{min} = k_{max} = 1$

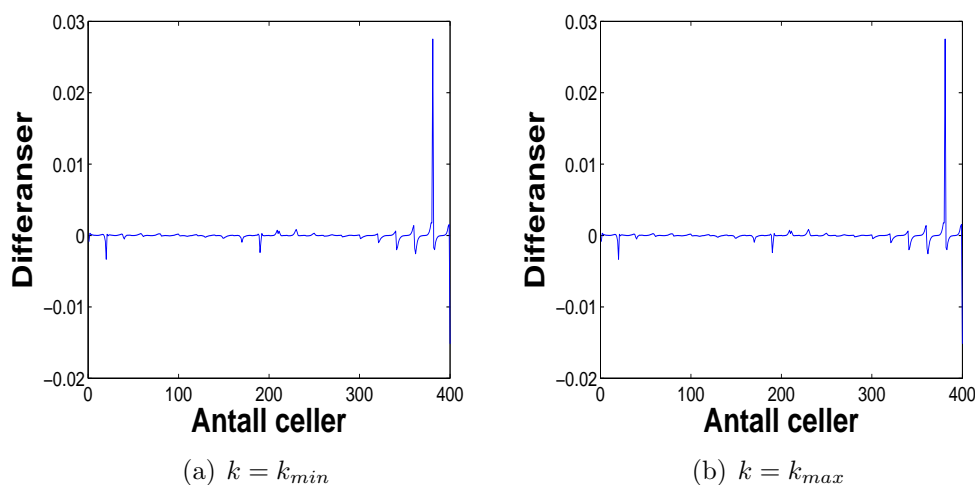
6.2.1 Differansen mellom løsningene ved k_{min} og k_{max}

La $p1s$ og $p1f$ være gjennomsnittsverdiene til de N settene av løsningene (analyseverdiene) for trykket ved k_{min} , det vil si ved det første tidssteget, for henholdsvis EnKS og EnKF. Tilsvarende er $p2s$ og $p2f$ gjennomsnittsverdiene til de N settene av analyseverdier ved k_{max} . Figur 6.1(a) viser differan-



Figur 6.1: Differansen mellom gjennomsnittsverdiene til de N settene av analyseverdier, ved k_{min} og k_{max} , for $k_{min} = k_{max} = 1$. Figur (a) for EnKS og figur (b) for EnKF.

sen mellom $p1s$ og $p2s$, og figur 6.1(b) gjør det samme for $p1f$ og $p2f$. Disse grafene antyder at gjennomsnittsverdiene ved k_{min} og ved k_{max} er identiske i dette tilfellet. Dette gjelder både for EnKS- og EnKF-metoden. Ifølge vår teori, forventet vi at $p2s$ er lik $p2f$, men 6.2(b) som er plottet av differansene mellom $p2s$ og $p2f$, viser at de to er ikke helt like. Dette gjelder også for $p1f$ og $p1s$, (se 6.2(a)) selv om $k_{min} = k_{max} = 1$, og vi i begge metodene benytter samme $d.dat$. Dette skyldes den normalfordelte usikkerheten som vi nevner ovenfor. Vi ser at grafene 6.2(a) og 6.2(b) er helt identiske, og den største verdien for differansen er 0,02753. Hvis vi sammenligner dette med intervallet for usikkerheten, $[-0.3, 0.3]$, kan vi si at disse to metodene har samme løsning ved $k_{max} = 1$. På grunn av dette kan vi ikke forvente at verdiene ved tidssteg $k = k_{max}$, for disse to metodene er helt identiske. De kan være nesten identiske i 'formen'.

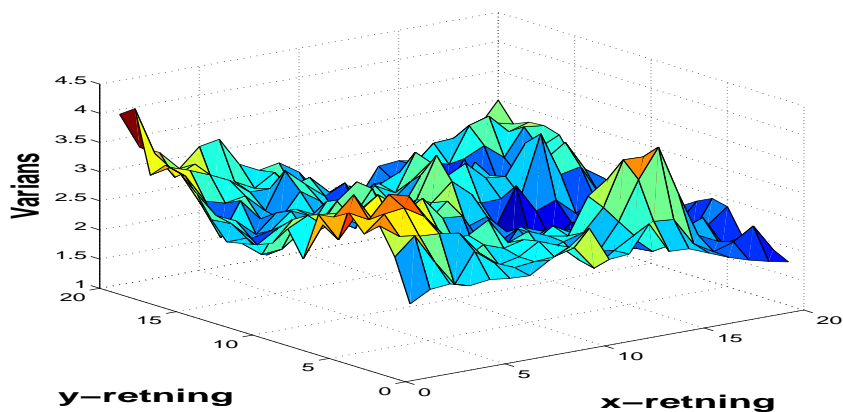


Figur 6.2: Differansene mellom smoother- og filterløsninger, (a) ved k_{min} og (b) ved k_{max} .

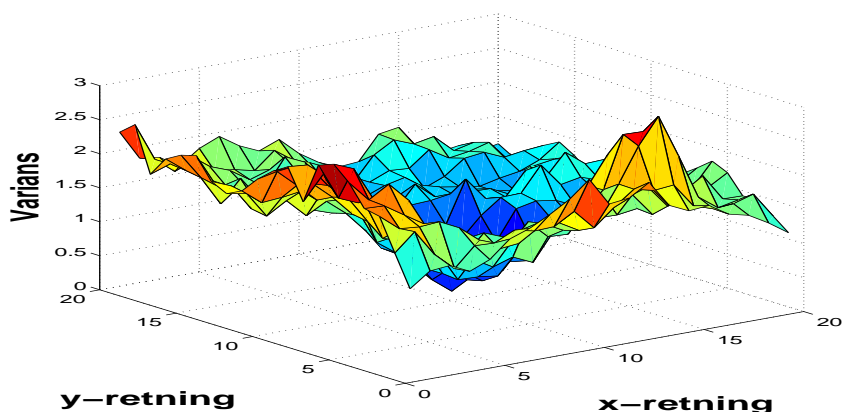
6.2.2 Differansen mellom initial- og slutt-varians til permeabiliteten for $k_{max} = 1$

Figur 6.3(a) er plottet for variansen til initialpermeabiliteten (initialvariansen) som vi har beregnet. Dette plottet vil ikke endre seg med tiden. Plottene 6.3(b) og 6.3(c) viser variansene til permeabiliteten ved $k_{max} = 1$ i henholdsvis EnKS- og EnKF-metodene. Her ser vi at de to formene ligner ikke på formen i 6.3(a), mens disse to ellers er nesten identiske. Grunnen til dette blir forklart nedenfor.

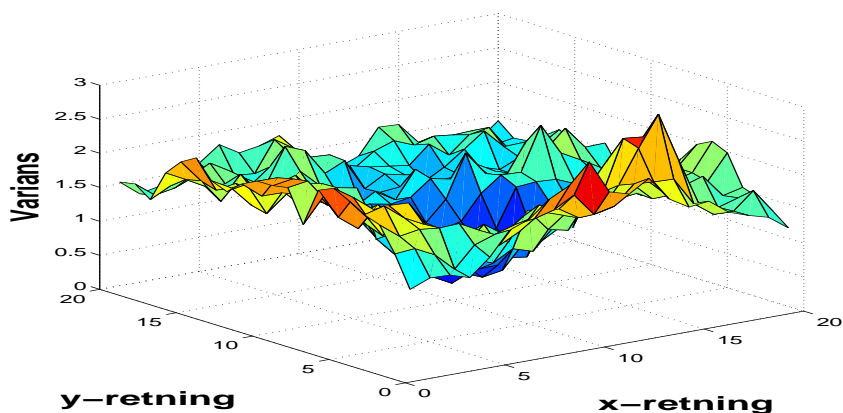
Figur 6.4(a) er plottet for sluttvariansene (variansen ved k_{max}) for både EnKS og EnKF ved $k_{max} = 1$, og viser tydeligere forholdet mellom variansverdiene for de to metodene. Her ser vi at selv om de to plottene, 6.3(b) og 6.3(c), ligner veldig mye på hverandre, så er det forskjell mellom mange parene av variansverdier. Plottet 6.4(b) viser hvor stor disse forskjellene er. Grunnen til at vi får disse forskjellene, er, som vi har nevnt i teorien (kapittel 5), at vi ikke utfører målinger for permeabiliteten. Likevel benytter vi den samme formelen for matrisen \mathbf{X} til å estimere analyseverdier for permeabiliteten, tilsvarende som for trykket. Usikkerheten fra målingene påvirker verdiene til \mathbf{X} .



(a) Initialvarians.

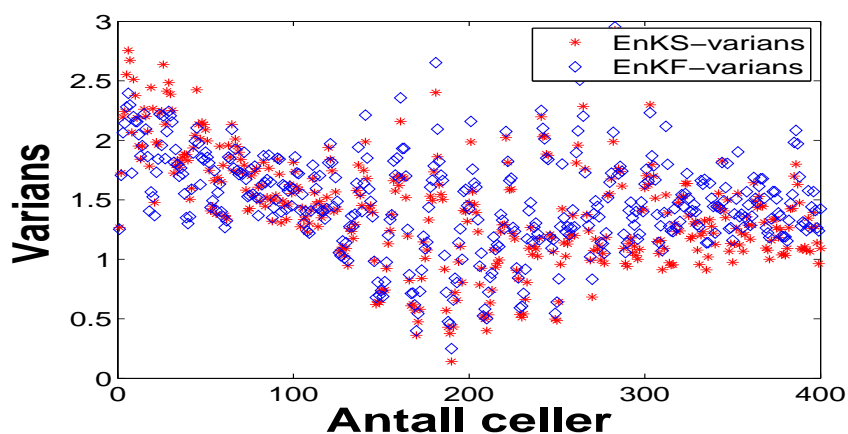


(b) EnKS

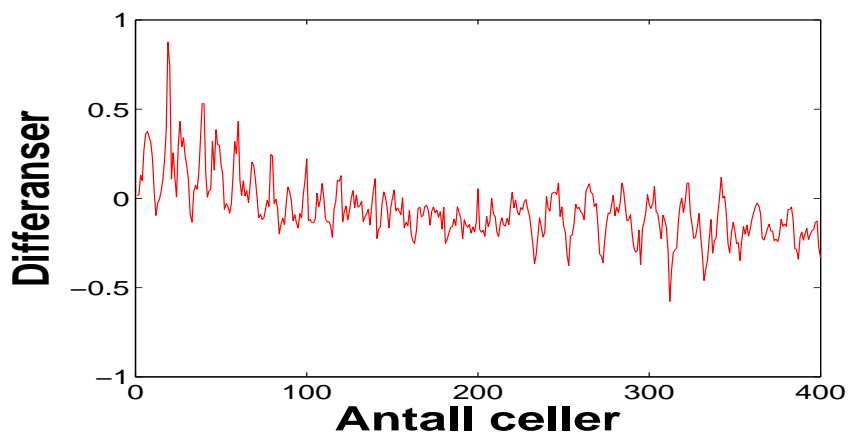


(c) EnKF

Figur 6.3: Figur (a) er variansene til initialpermeabiliteten. Figurene (b) og (c) er variansene til permeabiliteten ved $k_{max} = 1$, henholdsvis for EnKS og EnKF.



(a) EnKS- og EnKF-variansene



(b) Differansene mellom EnKS- og EnKF-variansene

Figur 6.4: Figur (a) er variansene til permeabiliteten ved $k_{max} = 1$ for både EnKS og EnKF. Figur (b) er differansene mellom EnKS- og EnKF-variansene.

6.3 Analyse av resultatene

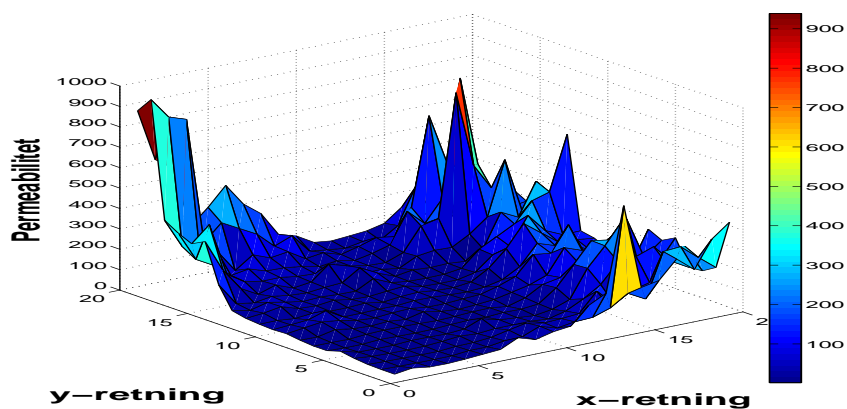
6.3.1 Permeabiliteten

Sammenligning mellom sanne verdier og analyseverdier

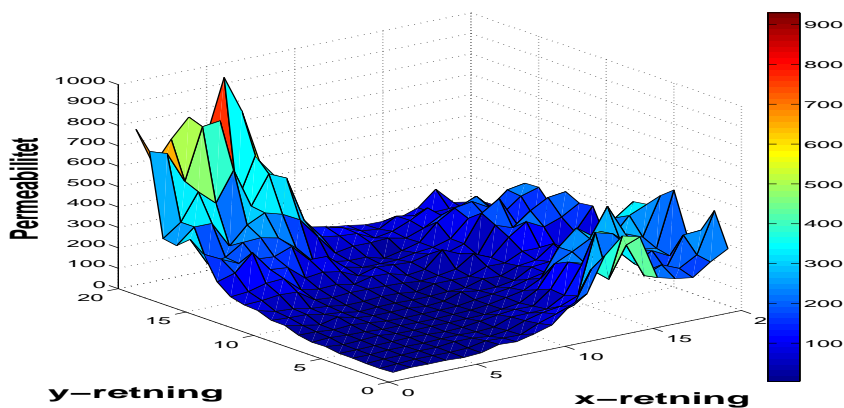
Som i kapittel 5, er \mathbf{u}_{sann} de sanne verdiene for permeabiliteten. Hos EnKS og EnKF er \mathbf{u}_{3s} og \mathbf{u}_{3f} gjennomsnittsverdiene til de N settene av analyseverdier for permeabiliteten ved k_{max} .

Figur 6.5(a) viser et tre-dimensjonalt plott for \mathbf{u}_{sann} . Figurene 6.5(b) og 6.5(c) er plottene av \mathbf{u}_{3s} og \mathbf{u}_{3f} . Den geometriske formen på flatene i disse

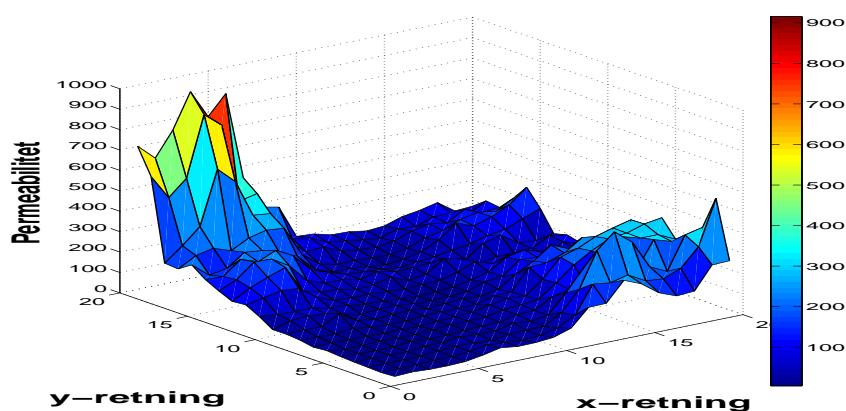
tre figurene er veldig like. Spesielt er figur 6.5(b) og figur 6.5(c) nesten identiske. Dette stemmer overens med Bemerkning 4.4.2, fordi disse to figurene er plottene til løsningene av EnKS- og EnKF-metodene ved det siste tidssteget, $k_{max} = 20$. Grunnen for at de ikke er helt identiske, følger av at vi legger til tilfeldig usikkerhet i kunstig data, \mathbf{D} , ved hvert tidssteg.



(a) Sanne permeabilitet



(b) EnKS

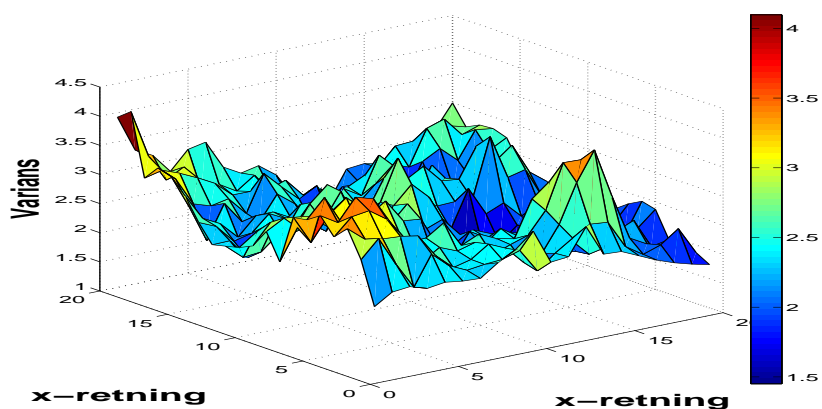


(c) EnKF

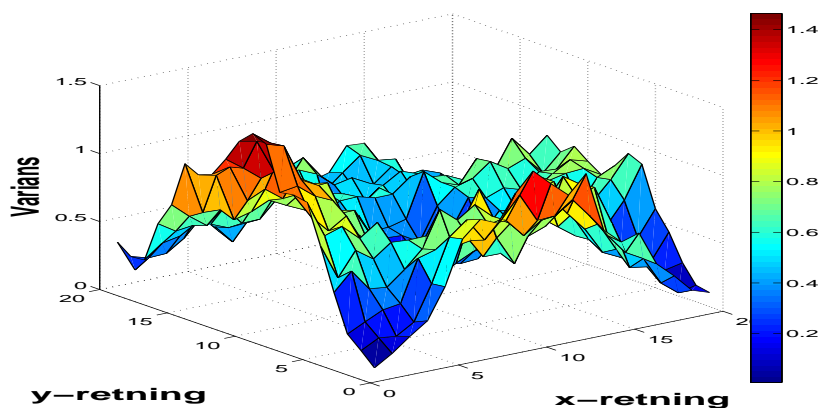
Figur 6.5: Figur (a) er den sanne permeabiliteten. Figurene (b) og (c) er gjennomsnittsverdiene til de N settene av analyseverdier for permeabilitet ved $k_{max} = 20$, henholdsvis for EnKS og EnKF.

Sammenligning av varians

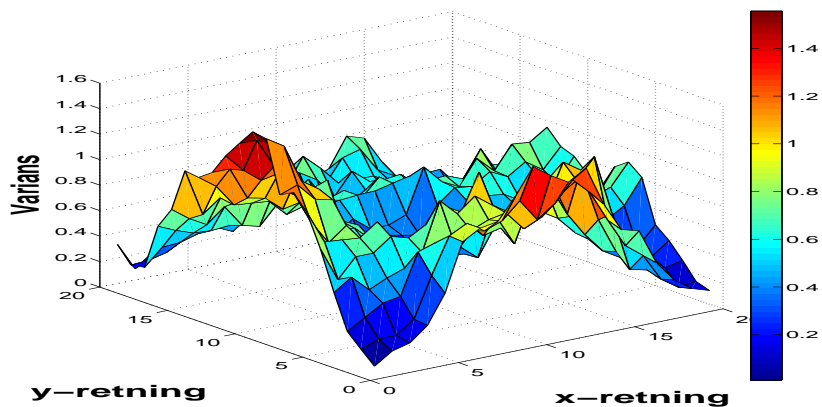
For å sammenligne initialvariansen og variansen ved $k_{max} = 20$ plottet vi variansen til logaritmen av initialpermeabiliteten (Figur 6.6(a)). Variansen til logaritmen av permeabiliteten som vi får ved $k_{max} = 20$ for de to metodene EnKS og EnKF er plottet i figurene 6.6(b) og 6.6(c). Her ser vi at disse to plottene er forskjellige i forhold til plottet for initialvariansen, men de to er nesten identiske i formen. Dette er det samme som i tilfellet der $k_{max} = 1$ som vi har observert i Kapittel 6.2.2. Her er forskjellen at variansverdiene ved $k_{max} = 20$ er mye mindre enn variansverdiene ved $k_{max}=1$, i begge metodene. Differansen mellom variansene for EnKS og EnKF ved $k_{max} = 20$, forklares nærmere i neste avsnitt. Plottene i figurene 6.6(b) og 6.6(c) viser også at vi får nesten det samme resultatet for begge metodene i det siste tidsteget, k_{max} , slik som Bemerkning 4.4.2 sier at vi skulle få.



(a) Initialvarians.

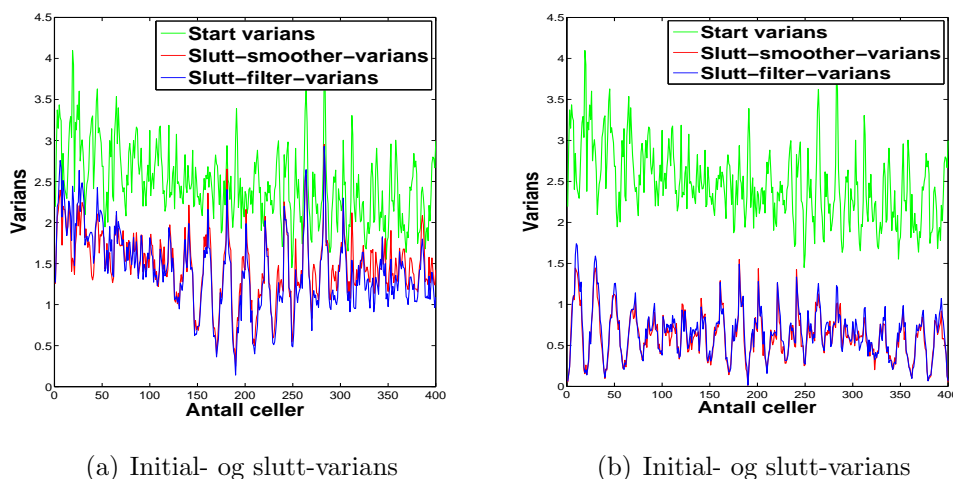


(b) EnKS



(c) EnKF

Figur 6.6: (a) Variansen til initialpermeabilitet. Figurene (b) og (c) er variansen til permeabiliteten ved $k_{max} = 20$, henholdsvis for EnKS og EnKF.

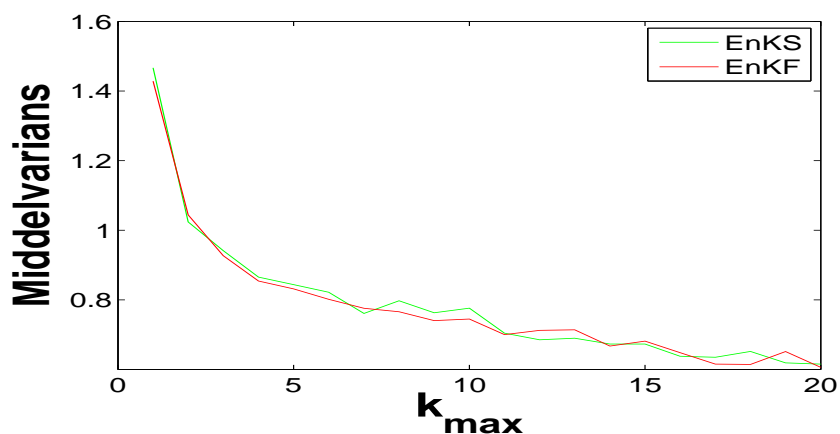


Figur 6.7: Figuren viser variansen til initialpermeabilitet og variansen for permeabilitetens analyseverdier for både EnKS og EnKF. Figur (a) ved $k_{max} = 1$ og (b) ved $k_{max} = 20$.

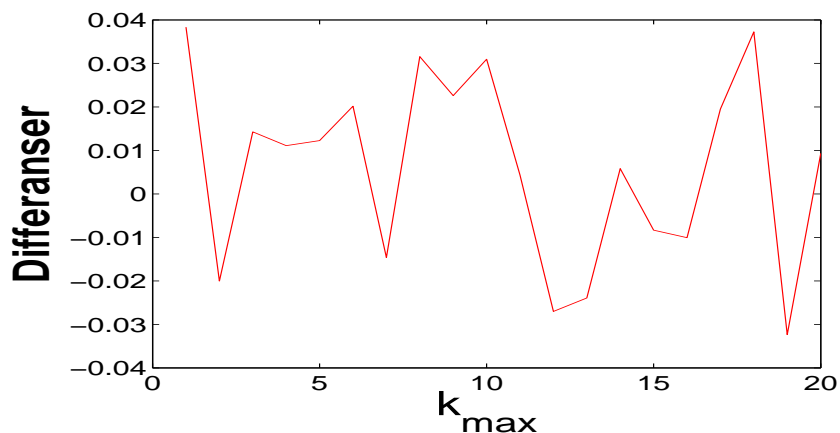
Figur 6.7(a) viser sammenhengen mellom variansen til initialpermeabiliteten og variansen til permeabiliteten ved $k_{max} = 1$ (sluttvariansen) for både EnKS og EnKF. Figur 6.7(b) viser det samme, men for $k_{max} = 20$. Her er det tydeligere å se at variansene til løsningene ved $k_{max} = 20$ i begge metodene avtar mye i forhold til når $k_{max} = 1$. Det er ikke lett å se differansen mellom variansen til EnKS-løsning og variansen til EnKF-løsning ved forskjellige verdier av k_{max} . I tillegg kan vi her ikke se hvordan de to variansen er minimerte etter økende verdier av k_{max} , og i stedet må vi kjøre begge metodene med verdier for k_{max} fra 1 til 20. Så beregner vi middelvariansen for hver verdi av k_{max} , vi beregner også ut differansene mellom middelvariansene fra EnKS og EnKF for hver verdi av k_{max} . Resultatene er vist i figurene 6.8(a) og 6.8(b).

Sammenligning mellom middelvariansen og den gitte variansen

I denne oppgaven har vi valgt variansen lik $\sqrt{(0.5)} = 0.7071$, da vi beregnet de sanne verdiene og initialverdiene for permeabiliteten. Utgangspunktet i teorien, er at EnKS-og EnKF-metoden har identiske løsninger ved tidsteg $k = k_{max}$ for hver verdi for k_{max} . De metodene har derfor identisk ensemblekovariansmatrise, \mathbf{P} , ved k_{max} . Dette medfører at variansene for disse to metodene er like, eller at differansen mellom dem er lik 0. Figur 6.8(a) viser kurvene til middelvariansene for forskjellige verdier av k_{max} for både EnKS og EnKF, og disse kurvene er ikke 'glatte'. Grafen i figur 6.8(b) beskriver differansen mellom middelvariansene for EnKS og EnKF. Her vises det at



(a) Middelvariansene



(b) Differanser

Figur 6.8: Figur (a) er middelvariansene ved forskjellige verdier av k_{max} for både EnKS og EnKF. Figur (b) er differansene mellom variansene fra EnKS og fra EnKF.

ved noen verdier for k_{max} , for eksempel $k_{max} = 1$, ligger differansen utenfor intervallet $[(-0.3, 0.3)]$. På grunn av den normalfordelte usikkerheten som vi har nevnt, betyr det at vi forventer at 67 prosent av verdiene ligger innenfor intervallet, resten kan være utenfor intervallet. Her er det bare 3 av 20 verdiene av k_{max} , eller 0.15 prosent som har differanse som ligger utenfor intervallet. Derfor kan vi antyde at differansene er lik 0. Etter kodene har blitt kjørt til og med $k_{max} = 20$, får vi at middelvariansene for EnKS og EnKF er henholdsvis $0.6157 \sim 0.6$ og $0.6064 \sim 0.6$. Differansen mellom disse to er lik 0.009, og begge er mindre enn 0.7. Med antall tidsteg lik $k_{max} = 20$,

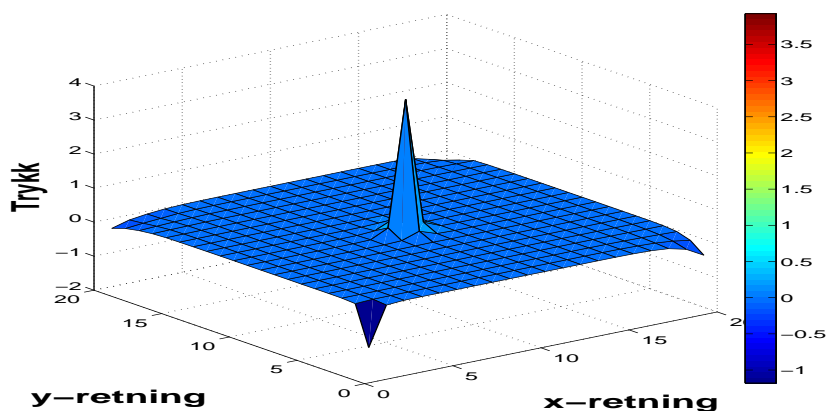
stemmer metodene våre bra når det gjelder permeabiliteten.

6.3.2 Trykk

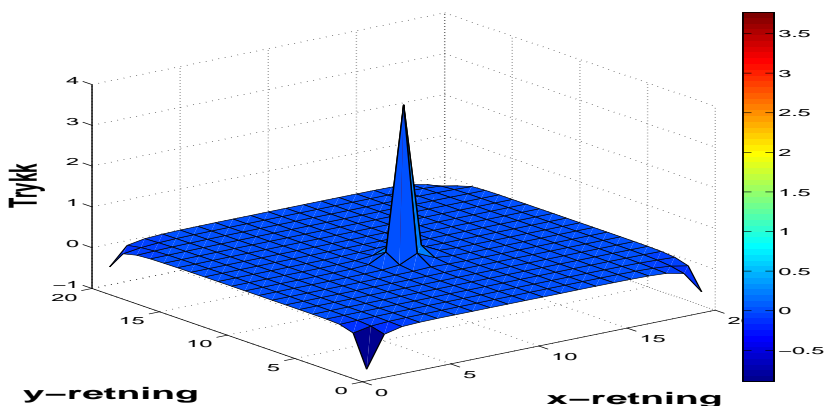
Plotting av analyseverdier ved $k_{max} = 1$

Med en verdi for hver celle er figur 6.9(a) plottet for de $n = 400$ forskjellige verdiene for trykket, ved $k_{max} = 1$, som vi får fra forover-modellen.

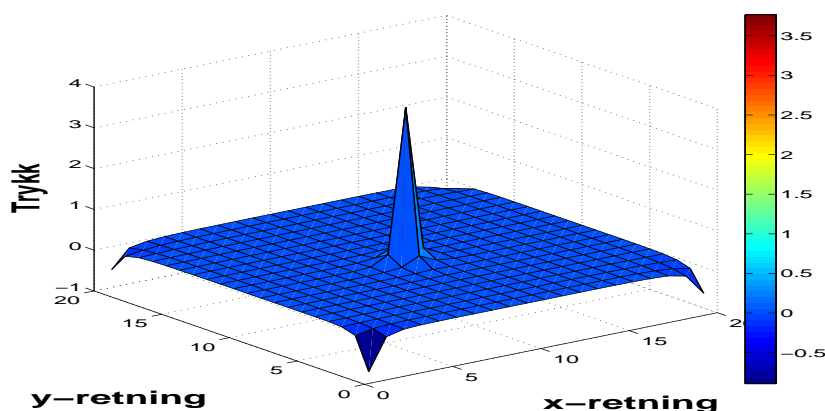
Figurene 6.9(b) og 6.9(c) er plottene for gjennomsnittsverdiene til de N settene av analyseverdier for trykket ved $k_{max} = 1$, henholdsvis for EnKS og EnKF. For $k_{max} = k_{min} = 1$ er de analyseverdiene ved k_{min} og k_{max} helt like for disse to metodene. Gjennomsnittene av løsningene for EnKS- og EnKF-metoden ved tidsteget $k_{min} = 1$ og $k_{max} = 1$, er derfor også identiske. På



(a) Forover-modell.



(b) EnKS



(c) EnKF

Figur 6.9: Resultater ved $k_{max} = 1$. Figur (a) er trykkverdier for forover-modellen. Figurene (b) og (c) er gjennomsnittet av de N settene av analyseverdier for trykk henholdsvis for EnKS og EnKF.

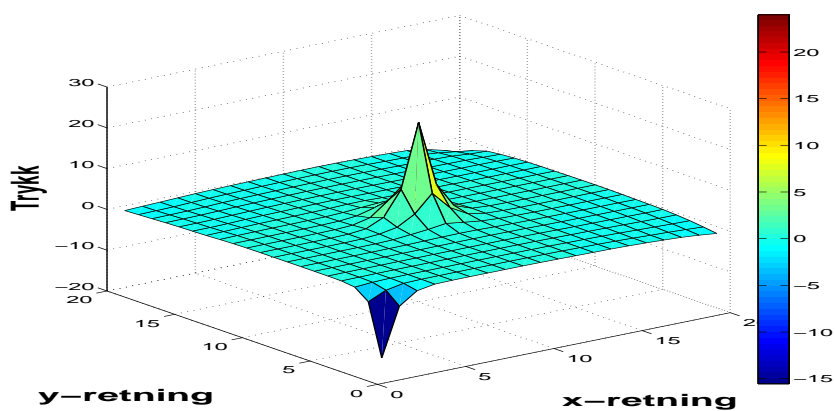
grunn av dette er plottene 6.9(b) og 6.9(c) helt identiske, og disse to plottene har lik 'form' som 6.9(a), men har forskjellige verdier.

Plotting av analyseverdier ved $k_{max} = 20$

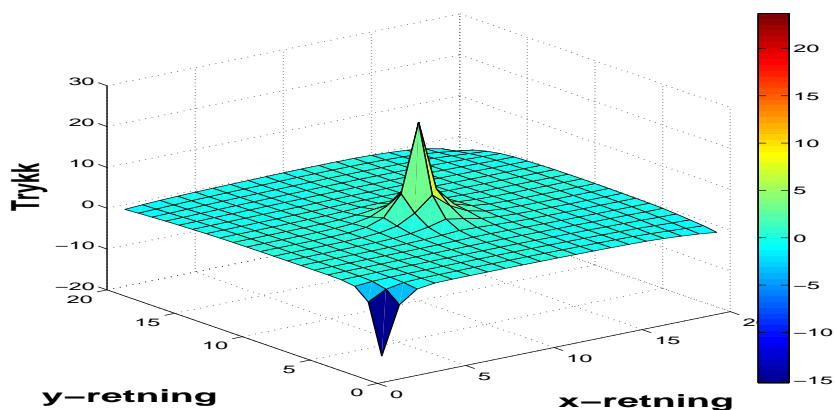
Figur 6.10(a) viser plottet vi får fra forover-modellen for $k_{max} = 20$. Plottet av gjennomsnittetsverdiene til de N settene av løsningene ved $k_{max} = 20$, for EnKS- og EnKF-metodene (se figurene 6.3.2 og 6.10(c)). Her er også plottene av gjennomsnittetsverdiene til trykk-løsningene for EnKS og EnKF ved $k_{max} = 20$, helt identisk i form, men ikke helt identisk i verdi. Dette er det samme som vi observerer i tilfellet der $k_{max} = 1$ (se Kapittel 6.2.1).

6.3.3 Utvikling av filter- og smoother-løsningen

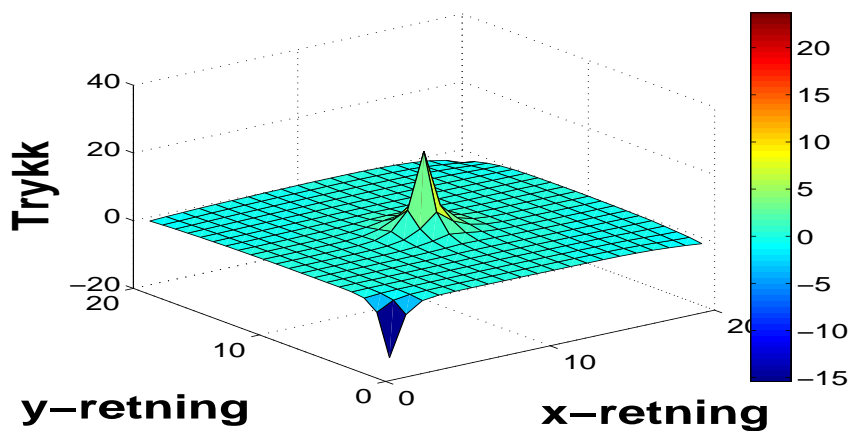
Vi kan se hvordan EnKS-løsningene for trykket utvikler seg over tid ved å sammenligne dem med EnKF-løsningene for trykket. Som vi ser ovenfor i figurene 6.9(b) og 6.9(c), er gjennomsnittene av smoother- og filter-løsningene ved $k_{min} = 1$, $p1s_1$ og $p1f_1$, like. Vi kan finne ut hva som skjer med $p1s_1$ og $p1f_1$ etter økende k_{max} -verdier. For å finne ut dette, kan vi kjøre hver kode for seg selv med forskjellige verdier for k_{max} . For hver verdi av k_{max} , når k går fra 1 til k_{max} , har vi løsningene ved $k = k_{min} = 1$ og ved $k = k_{max}$. Vi lar nå $p1s_{k_{max}}$ og $p1f_{k_{max}}$ være gjennomsnittsverdiene av løsningene ved $k = k_{max}$ til en viss verdi av k_{max} . Så regner vi ut differansen mellom $p1s_1$ og $p1s_{k_{max}}$



(a) Forover-modell



(b) EnKS



(c) EnKF

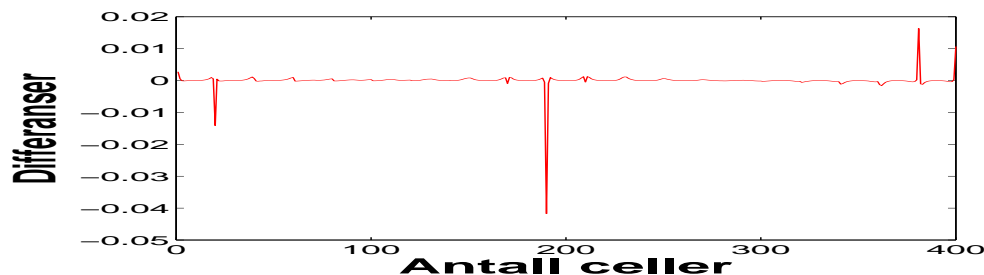
Figur 6.10: Resultater ved $k_{max} = 20$. Figur (a) viser trykkverdi fra forover-modellen. Figurene (b) og (c) er gjennomsnittet av de N settene av analyseverdiene henholdsvis for EnKS og EnKF.

i EnKS-metoden. Vi gjør det samme for EnKF-metoden. Resultatene ved $k_{max} = 2, 6, 14, 19$ for begge metodene, blir vist i figurene 6.11 og 6.12.

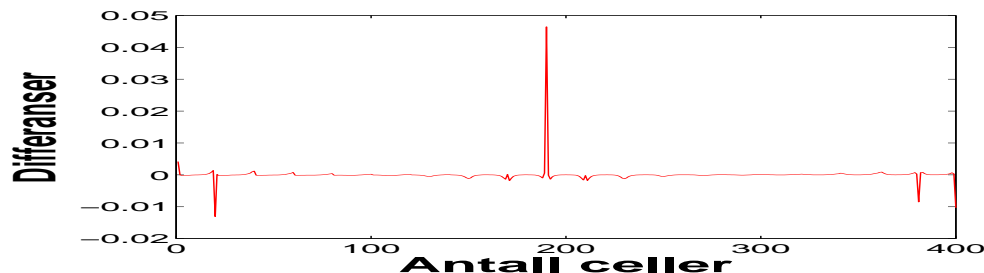
Ifølge teorien vår, er tidligere EnKF-løsninger ikke tidsoppdaterte, fordi filter-løsningene bare avhenger av tidligere målinger. Det betyr at når vi går fra k_{max} til $k_{max} + 1$, vil filter-løsningene ved $k_{max} - 1, k_{max} - 2, \dots$, ikke endre seg. I motsetning til EnKF vil tidligere EnKS-løsninger endre seg, fordi disse løsningene avhenger både av forrige og senere målinger.

Figuren 6.11 viser at grafene for differansene mellom $p1s_1$ og $p1s_{k_{max}}$ ved forskjellige verdier av k_{max} , endrer form for økende k_{max} -verdier.

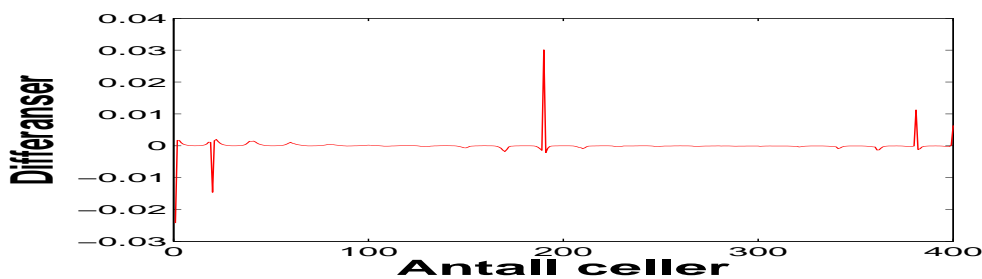
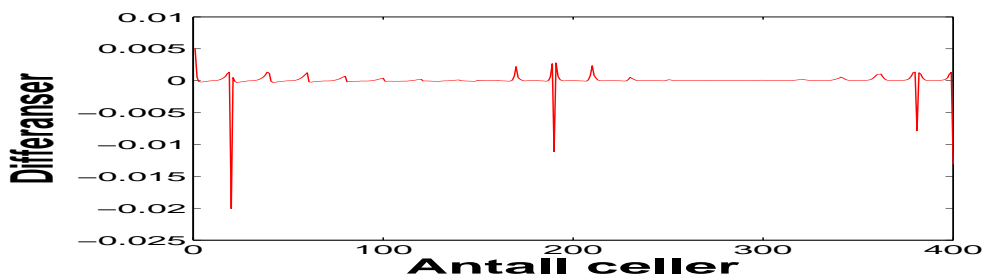
Fra figuren 6.12 ser vi at grafene for differansene mellom $p1f_1$ og $p1f_{k_{max}}$ med forskjellige verdier for k_{max} i EnKF-metoden, har de samme formene, men forskjellige størrelser. Dette skyldes også den normalfordelte usikkerheten i målingen, som endrer seg hver gang vi kjører programmet for hvert tidsteg k . Her ser vi på differansene for hele settet av verdier, det vil si verdier for 400 celler. I neste avsnittet skal vi se på bare en celle. Da blir det lettere å sammeligne resultatene.



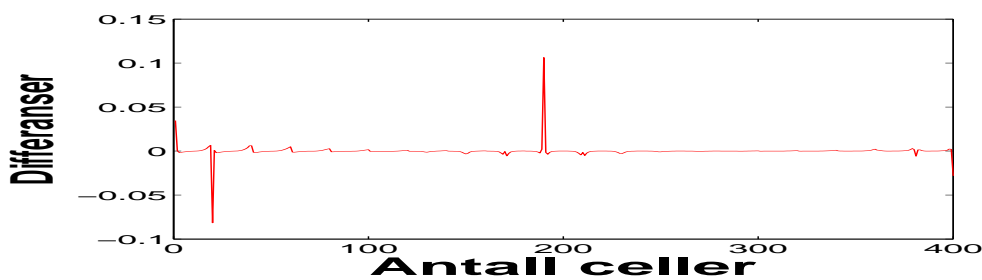
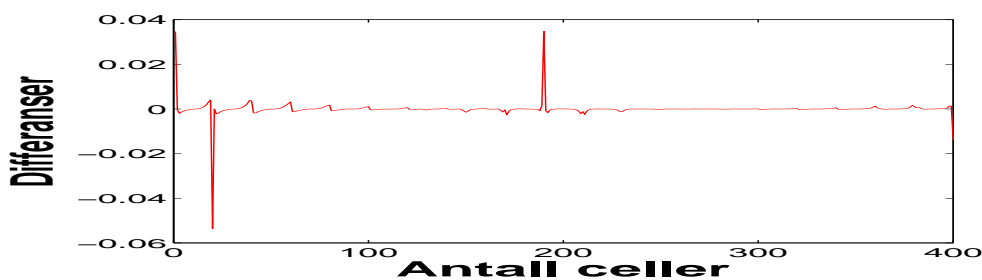
(a) For $k_{max} = 2$ ved EnKS

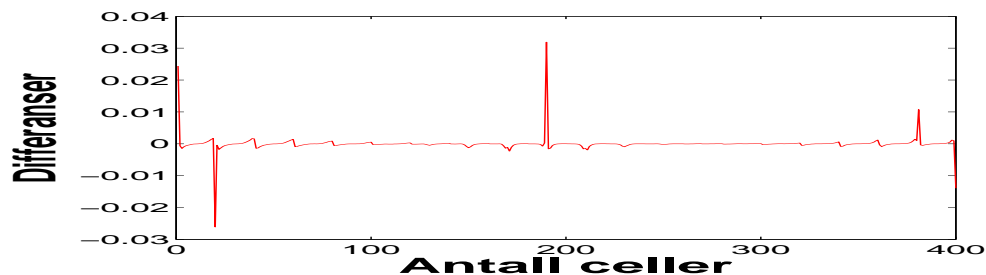
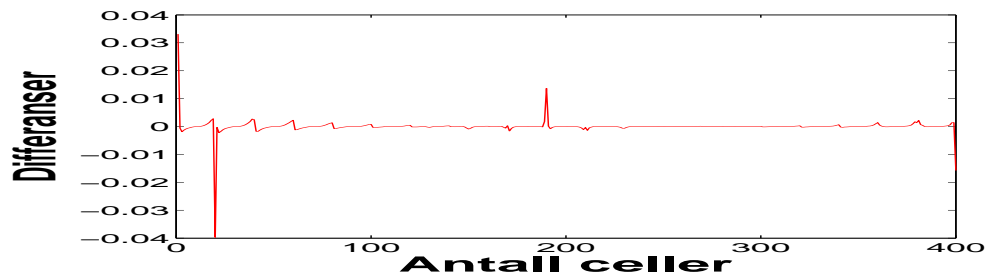


(b) For $k_{max} = 6$ ved EnKS

(c) For $k_{max} = 14$ ved EnKS(d) For $k_{max} = 19$ ved EnKS

Figur 6.11: Differansen mellom $p1s_1$ og $p1s_{k_{max}}$ for forskjellige verdier av k_{max} for EnKS-metoden.

(a) For $k_{max} = 2$ ved EnKF(b) For $k_{max} = 6$ ved EnKF

(c) For $k_{max} = 14$ ved EnKF(d) For $k_{max} = 19$ ved EnKF

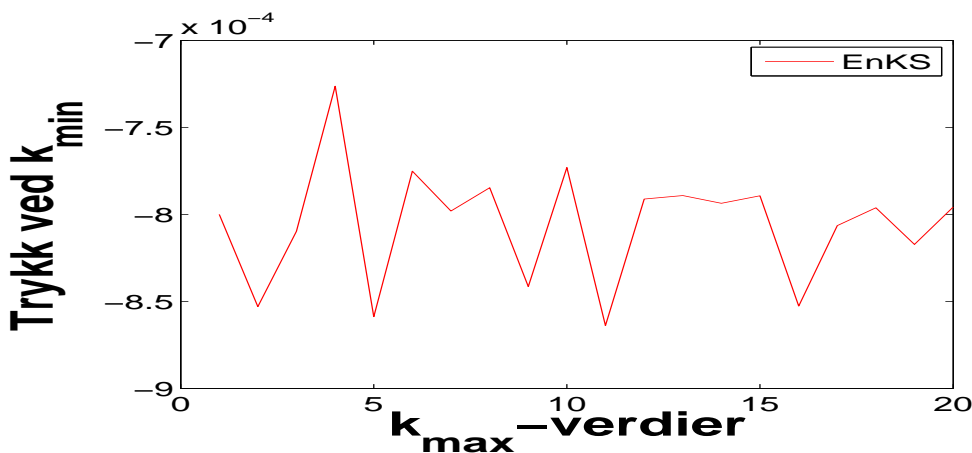
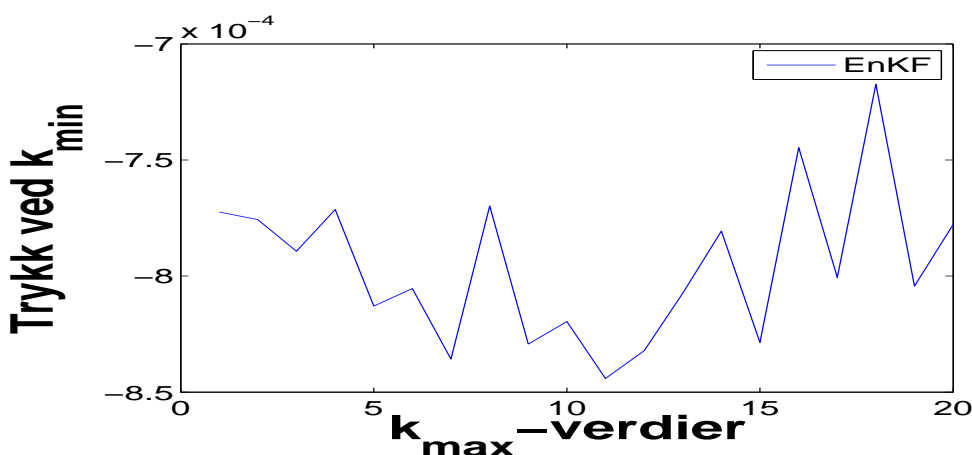
Figur 6.12: Differansen mellom $p1f_1$ og $p1f_{k_{max}}$ for forskjellige verdier av k_{max} for EnKF-metoden.

Utvikling av av filter- og smoother-løsning i celle nummer 136 for $k_{max} = 1 : 20$.

Vi vil nå finne ut hvordan filter- og smoother-løsning utvikler seg med tiden (økende k_{max} -verdier) for en celle, for eksempel cellen nummer 136.

Fra figurene 6.13(a) og 6.13(b) ser vi at kurvene for trykk-løsningene i celle nr. 136 for EnKS og EnKF, ved k_{min} , for forskjellige verdier av k_{max} , ligner ikke på hverandre. Det er ikke lett å se at de tidligere filter-øsningene i cellen 136 ikke endrer seg, utfra grafene.

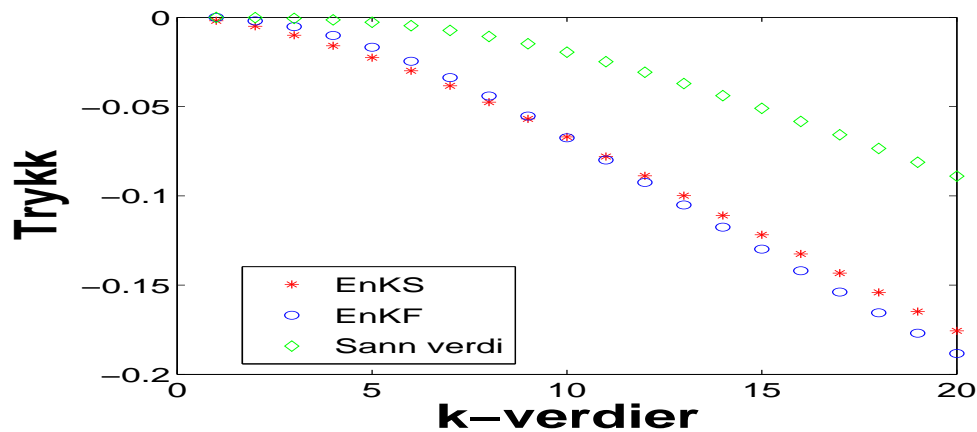
Her er trykkverdiene gjennomsnittet av de N settene av analyseverdier vi får ved k_{min} , for hver verdi av k_{max} . Middelverdiene av disse trykkverdiene er (-8.0576×10^{-4}) og (-7.9596×10^{-4}) for henholdsvis EnKS og EnKF. Utfra disse verdiene ser vi at trykkverdiene er veldig små. Dette medfører at det er vanskelig å se endringene i de tidligere smoother-løsningene. Variansen til disse trykkeverdiene som er (1.1575×10^{-9}) for EnKS, og (1.0652×10^{-9}) for EnKF. Variansverdiene viser at de tidligere EnKF-løsningene har mindre variansen enn de tidligere EnKS-løsningene, selvom variansverdiene her er veldig små. På grunn av at verdiene for trykk er veldig små i dette tilfellet, kan vi ikke få en god sammenligning her

(a) Smoother-løsningene ved k_{min} (b) Filter-løsningene ved k_{min}

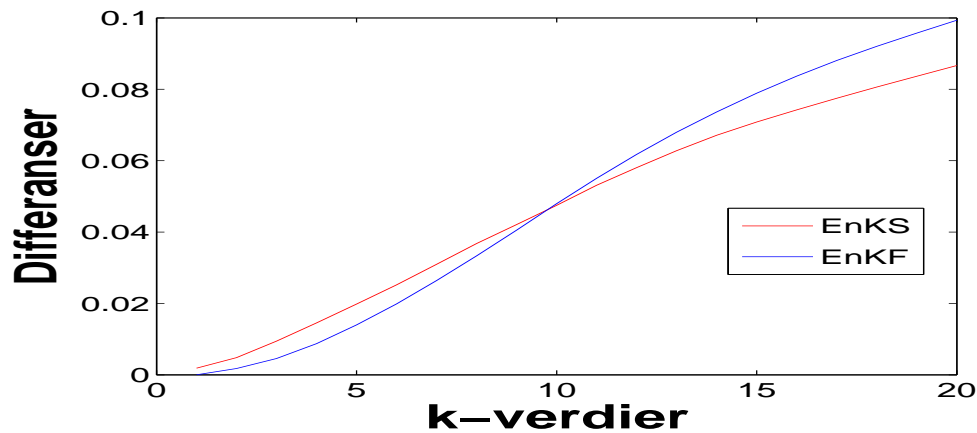
Figur 6.13: Trykkverdier (gjennomsnittet av de N settene av analyseverdier) ved k_{min} , p1f, for forskjellige verdier av k_{max} i celle nr. 136. (a) for EnKS-metoden og (b) for EnKF-metoden.

Utvikling av av filter-og smoother-løsning i celle 136 for $k = 1 : k_{max}(= 20)$

Vi kan også se hvordan filter- og smoother-løsningene i celle nummer 136, ved forskjellige tidsteg k , utvikler seg for $k_{max} = 20$. Som nevnt, utfra teorien forventer vi at smoother- og filter-løsningen ved $k = k_{max} = 20$ er like. Her ser vi i figur 6.14(a) at disse verdiene er ikke like. I figur 6.14(b), ved $k = k_{max}$ er differansen mellom smoother- og filter-løsningen i cellen 136 lik 0.0127. Denne forskjellen er mye mindre enn 0.3. Det er vanskelig å få en nøyaktig



(a) EnKS-, EnKF-løsninger og sanne verdier.



(b) Avvik fra sanne verdier.

Figur 6.14: Figur(a) er plottet for trykkverdiene i celle nummer 136 ved tidsteg k som går fra 1 til $k_{max} = 20$ for både EnKS og EnKF sammen med sanne verdier i celle 136. Figur (b) er avvik mellom sann verdier for både EnKS og EnKF.

sammelingen på grunn av den normalfordelte usikkerheten i målingene, som gjør at kunstigsdata, \mathbf{D} , ikke er den samme for disse to metodene i sammen tidsteg k .

Fra $k = 10$ framover til $k = k_{max} = 20$ er smoother-løsningene næmere de sanne verdiene enn filter-løsningene (se figur 6.14(a)). Ved $k = k_{max} = 20$ har verdien fra EnKS mindre avvik fra den sanne verdien enn verdien fra EnKF, og figur 6.14(b) viser at begge avvikene er mindre enn 0.3. Dette er sammen grunnen som overfor, at trykkverdiene er veldig små. Vi kan derfor ikke gi noen konklusjoner ut fra disse sammeligningene.

Plotting mellom analyseverdier og de sanne verdiene

Vi kan plote de N settene av analyseverdier (sluttverdier ved $k_{max} = 20$) sammen med settet av de sanne verdiene, for å se sammenhengen mellom dem. Vi får da $N + 1$ dataverdier for hvert tidsteg k . For hvilken som helst av de fem brønnene, kan vi plote verdiene på denne måten. Vi har gjort det for alle, men viser her bare resultatene for brønn nummer 1 og brønn nummer 3. Ettersom brønn nummer 2, 4 og 5, har samme målingsverdi som brønn nummer 1, fra starten. Brønn nummer 1 har positivt trykk fra begynnelsen, mens brønn nummer 3 har negativt trykk.

Plotting for brønn nummer 1

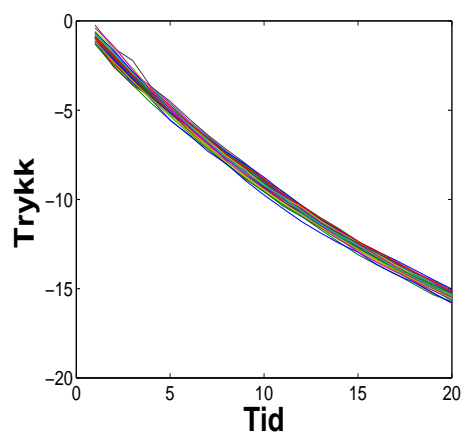
I figur 6.15(a) plotter vi analyseverdiene og de sanne verdiene for EnKS-metoden, og i figur 6.15(c) gjør vi det samme for EnKF-metoden for brønne nummer 1. Ved hjelp av 6.15(b) og 6.15(d), som er plottene for kun de sanne dataene for hver metode, ser vi at analyseverdiene ikke har så store avvik hverken i forhold til hverandre eller til de sanne dataene. De sanne verdiene er de samme for begge metodene.

Plotting for brønn nummer 3

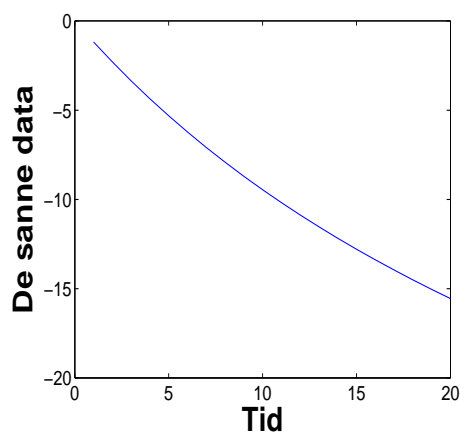
Vi gjør det samme for brønn nummer 3. Tilsvarende er plottene vist i figurene 6.16(b), 6.16(d), 6.16(a) og 6.16(c). De har den formen de har, på grunn av at brønnen har negativt trykk fra starten.

Plotting mellom kunstige og sanne data

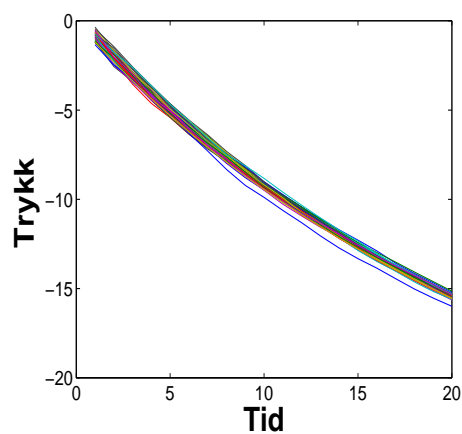
Ved $k = k_{max}$ får vi en $(m \times N \times k_{max})$ matrise som er kunstige målinger for trykket. Dette er det samme for begge metodene. Vi plotter de N settene av kunstige data og settet av de sanne dataene for å sammeligne mellom kunstige data og sanne data. Fra figurene 6.17(a) og 6.17(c) ser vi at kurvene for de kunstige dataene ikke er glatte. Dette betyr at disse endrer seg tilfeldig, noe som stemmer med det som vi har nevnt i begynnelsen av kapittelet, om kunstige data, **DD**.



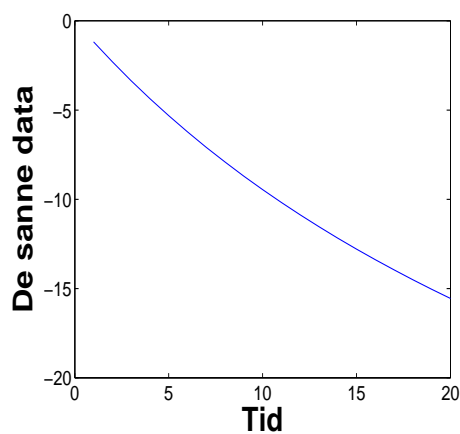
(a) Analyse- og sanne verdier ved EnKS



(b) De sanne verdier.

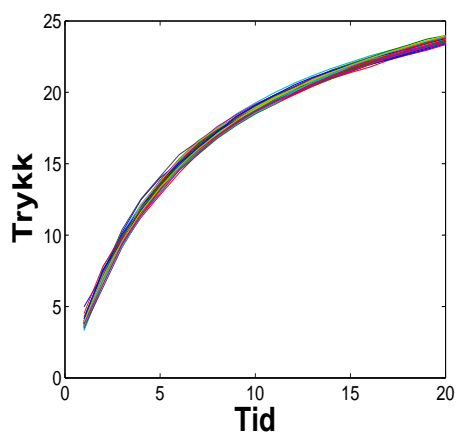


(c) Analyse- og sanne verdier ved EnKF

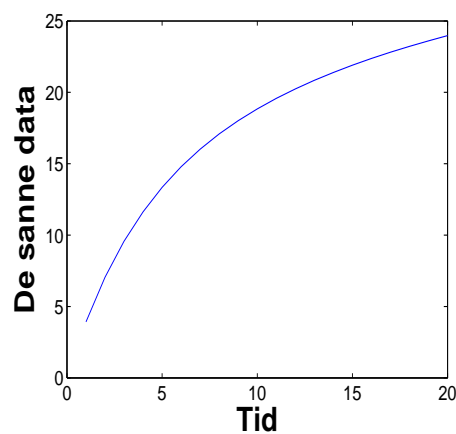


(d) De sanne verdier.

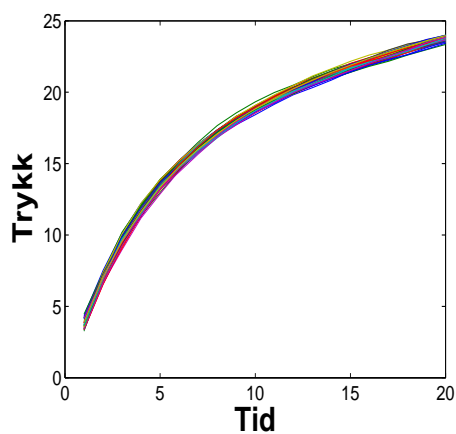
Figur 6.15: Analyse- og sannverdier for brønn nummer 1, (a) for EnKS og (c) for EnKF. Figur (b) og (d) viser sanne verdier for begge metodene.



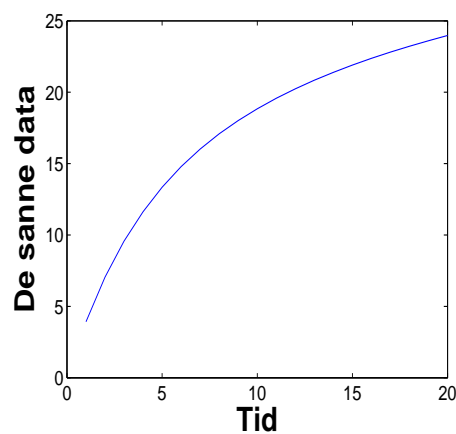
(a) Analyse- og sanneverdier ved EnKS



(b) De sanne verdiene

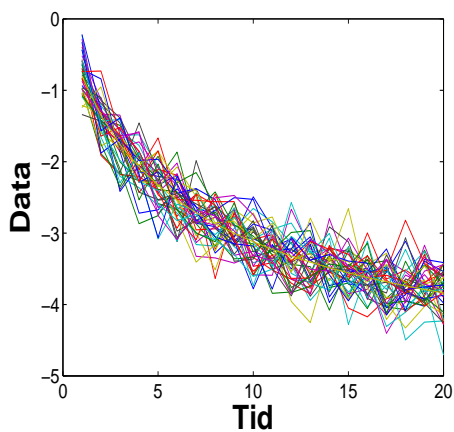


(c) Analyse- og sanne-verdiene ved EnKF

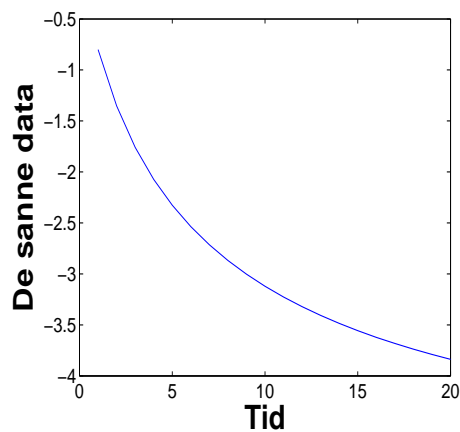


(d) De sanne verdiene

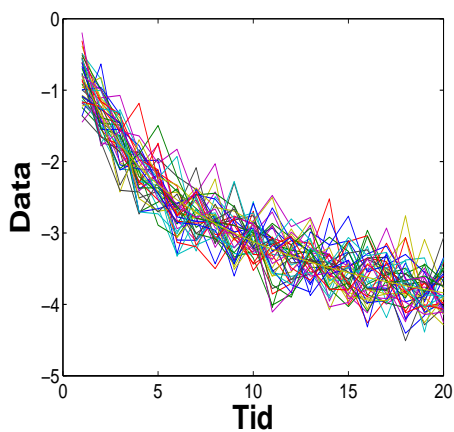
Figur 6.16: Analyse- og sannverdier for brønn nummer 3, (a) for EnKS og (c) for EnKF. Figur (b) og (d) viser sanne verdier for begge metodene.



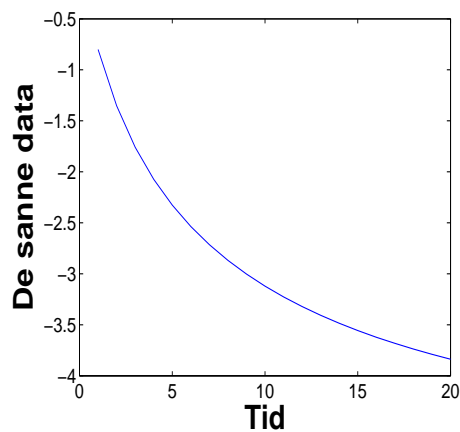
(a) Kunstige og sanne data ved EnKS



(b) De sanne data.



(c) Kunstige og sanne data ved EnKF



(d) De sanne data.

Figur 6.17: Kunstige og sanne data ved $k_{max} = 20$ (a) for EnKS, (c) for EnKF. Figurene (b) og (d) er sanne data.

Kapittel 7

Konklusjon og videre arbeid

7.1 Modelligning og forover-problem

Reservoarligningen for en svakt kompressibel væske, er utledet av bevarelseligningen for enfasestrømning i porøse stoffer, og kan behandles som varmeligningen. Varmeligningen kan løses som både forover-modell og invers-modell ved bruk av numeriske metoder. I forover-modellen kan permeabiliteten enten antas å være konstant både i tid og posisjon, eller konstant bare i tid, mens den varierer i posisjon. Cellesentrert bevarelseskjema som vi har brukt i to-dimensjoner, er en implisitt metode der vi antar at permeabilitet er konstant i tid og varierer i posisjon [2]. I tillegg har vi benyttet harmonisk permeabilitet til å utlede vår varmeligningen i to-dimensjoner. For å beregne harmonisk middelpermeabilitet har vi brukt massens konserveringslov. Resultatet fra denne implisitte metoden, er en iterativ form som vi brukte i forover-modellen for å få sanne data, *d.dat*, og en fasit for trykkløsningen til den inverse-modellen.

7.2 Invers-problem

Generelt for inverse-problemet finner vi et estimat av tilstanden gitt en dynamisk modell og målinger, der feilene i modellen og målingene er minimerte under estimeringene. I oppgaven har vi antatt at modellfeilen er lik 0.

EnKF kan utledes på mange måter, som for eksempel i denne oppgaven hvor vi har utledet EnKF fra det klassiske Kalman-filteret. Som det blir nevnt i [11], antar KF at alle *pdf* er Gaussiske, og bruker Baysiansk oppdatering til å endre gjennomsnittet og kovariansmatrisen. Et problem med KF er at det ikke er mulig å beregne kovariansmatrisen for systemer med høy dimensjon. Dette er grunnen for at EnKF har blitt utviklet. For

EnKF-metoden vi har brukt i våre simuleringer, har kovariansmatrisen dimensjon $(2n \times 2n) = (800 \times 800)$, men vi leser bare av kovariansmatrisen ved $k_{max} = 20$, da den er en $(n \times n) = (400 \times 400)$ -matrise, for permeabiliteten slik at vi kan utregne variansen dens.

I følge Johns og Mandel [9], representerer hvert medlem av ensemblet (i EnKF-metoden) en mulig løsning, og er propagert i tid ved å kjøre modellen uavhengig opptil analysetid. Her i oppgaven har vi simulert med $N = 40$ medlemmer av ensemblet, derfor får vi 40 sett av verdier ved hvert tidsteg, og i tillegg har vi analysert resultatet fra ett sett av gjennomsnittsverdier til de N settene av analyseverdier.

Det er viktig å anta at målingsfeilen og prior simultan *pdf* er normalfordelt (Gaussiske fordeling), fordi ifølge Evensen (Kapittel 4 i [4]) vil EnKS og EnKF ikke gi korrekte svar hvis prior simultan *pdf* ikke har Gaussisk fordeling. Aanonsen et al [10] nevner at for ikke Gaussiske problemer, gir EnKF ikke et korrekt estimat av postprior sannsynlighetsfordeling. I oppgaven ser vi at de estimerte verdiene av permeabiliteten blir bedre etter tid, og ved siste målingstid ligner analyseverdiene av permeabiliteten veldig mye på de sanne verdiene. Dette gir gode estimater for trykket.

Matrisen for kunstige data har det normalfordelte usikkerhetsleddet, $0.3 * randn$, som gjør at matrisen endrer seg ved hvert tidsteg for hver gang vi kjører simuleringene. Dette fører til at sammenligningene for resultatene fra EnKF og EnKS ikke gir helt nøyaktige resultater. Derfor ville sammeligningene blitt bedre hvis vi hadde brukt samme kunstige data \mathbf{D} for en viss verdi av k_{max} i begge metodene. For å få dette til, måtte man først ha kjørt den ene metoden med denne verdien av k_{max} for å få matrisen \mathbf{D} , og så bruke \mathbf{D} i den andre metoden. Vi måtte da også ha tatt hensyn til usikkerheten.

I tillegg er resultatene som har blitt analysert her, resultatene bare fra analysesteg, fordi vi ikke har tatt hensyn til prediksjonsverdien i simuleringen. Dette fører til at hvis filteret hadde utført gode estimeringer i prediksjonsteget, ville EnKS og EnKF ha like løsninger (analyseverdi) i nesten hvert tidsteg. Derfor var det i Kapittel 6 vanskelig å se forskjellen mellom de tidligere løsningene til EnKS, og til EnKF. Vi skulle ha tatt hensyn til prediksjonsverdien for å få bedre sammenligningsgrunnlag.

I denne oppgaven er trykkverdiene veldig små, og den normalfordelte usikkerheten i målingen er 0.3. Dette gjør at det blir vanskelig å konkludere noe ut fra disse sammeligningene når det gjelder trykket.

Ifølge Evensen (Kapittel 9 i [4]) er løsningene for EnKS 'glattere' enn for EnKF, og passer nærmere til den sanne løsningen. Dette stemmer når det gjelder permeabiliteten i våre eksperimenter, men for trykket er det ikke lett å gi noen kommentarer på grunn av at trykkverdiene er veldig små.

Arbeidet i denne oppgaven, viser at algoritmen for EnKF er enklere enn

algoritmen for EnKS, fordi i EnKS må vi ha en dobbelløkke (på grunn av bakover-rekursiven), ellers er algoritmene for disse to metodene, like. Grunnen til at disse ellers er like, er fordi vi har antatt at vi har målinger ved hvert tidssteg.

7.3 Videre arbeid

I denne oppgaven ser vi at når antall tidsteg k_{max} er lik 20, stemmer løsningene i metodene våre med teorien (for permeabiliteten). Det ville vært interessant å simulere videre for verdier av k_{max} større enn 20, det vil si at det ville vært flere målinger involvert. Noen interessante spørsmål, er om variansen da ville forsette å avta, og om EnKS fortsatt ville være et bedre estimat enn EnKF hvis vi har et større tidsintervall. Simuleringene i oppgaven er utført der avstanden mellom målingene er 0.01 sekunder. Hva ville skjedd hvis vi øker denne avstanden? Og hvilken metode ville vært best da?

Det ville også vært interessant å finne ut om kombinasjonen mellom størrelsene på ensemblet og antall tidsteg, har noen betydning for løsningene i disse to metodene, og hvilke kombinasjoner som ville gitt gode resultater.

En kunne se hvilken metode som har glattest kurve for variansen, ved å finne ut hvordan variansen til permeabiliteten avtar for forskjellige verdier av $k = 1 : k_{max}$ for hver metode.

Videre ville det vært spennende å se nærmere på hvordan metodene har blitt, og blir, anvendt i petroleumsindustrien. Almendral-Vazquez og Syverveen [12] nevner for eksempel i deres rapport en anvendelse av EnKF for *history matching*, der EnKF kan oppdatere både statisk (permeabilitet og porøsitet) og dynamikk (trykk og metning) modell-parametre. Utfra deres eksperimenter, konkluderer de med at når man benytter EnKF for *history matching*, kan reservoarsimuleringene kjøres parallelt. Dette vil spare tid.

Teknikkene for Kalman filter kan også anvendes som et mulig redskap for kontroll av bunnhullstrykket under boring [10].

Tillegg A

Liste over symboler

Reservoarligninger

c	Kompressibilitet
p	Trykk
Q	Massekildetetthet
\mathbf{n}	Ytre enhetsnormal vektor
u	Harmonisk permeabilitet
\mathbf{u}	Permeabilitet vektor
\mathbf{v}	Volumstrømtetthet
μ	Vikositet
κ	Tensor

Kalmanfilteret

Ψ	Tilstandsvektor
\mathbf{A}	Lineær-modellmatrise
\mathbf{w}	Modellfeil
\mathbf{Q}	Modellfeil-kovariansmatrise
\mathbf{d}	Målingsvektor
\mathbf{H}	Målingsmatrise
\mathbf{v}	Målingsfeil
\mathbf{R}	Målingsfeil kovariansmatrise
\mathbf{P}	Tilstandsfeil-kovariansmatrise
\mathbf{K}	Kalman Gain matrise
h	Målingsfunksjon
g	Systemfunksjon

Ensemble metode

B	Ensemble-matrise
B_i	Ensemble-matrise ved tid t_i
$\widehat{\mathbf{B}}$	Ensemble-forventningsmatrise
B[*]	Ensemble-pertubasjonsmatrise
$\widetilde{\mathbf{B}}_i$	Ensemble-matrise for simultan tilstand fra t_0 til t_i
D	Pertubasjonsmatrisen for måling
D_j	Pertubasjonsmatrisen for måling ved tid j
N_e	Antall medlem av ensemblet
1_N	Matrisen hvor alle element er lik $\frac{1}{N_e}$
P^e	Ensemble-feil-kovariansmatrise
Q^e	Ensemble-kovariansen til prosessfeil
R^e	Ensemble-kovariansen til målingsfeil
S	Måling av ensemble-pertubasjon
S_j	Måling av ensemble-pertubasjon ved tid j

Øvre indeks

<i>f</i>	Prediksjon (<i>a priori</i>)
<i>a</i>	Analyse (<i>a posteriori</i>)
<i>m_j</i>	Antall måling ved tid t_i
<i>s</i>	Smother-løsning

Nedre indeks

<i>k</i>	Tidsteg
<i>kk</i>	Subtidsteg
τ	Antall målinger
<i>x₁</i>	x-retning
<i>x₂</i>	y-retning
<i>init</i>	Initialverdi
<i>k_{min}</i>	Det første tidsteg
<i>k_{max}</i>	Det siste tidsteg
<i>j</i>	<i>j</i> -te måling ved tidsteg k

Permeabilitet-beregning

C	Kovariansmatrise
M	Diagonalmatrise
U	Korrelasjonsmatrise
J	Nedre triangulær matrise
x	Multivariate normalvektor
<i>perm</i>	Permeabiliteten

Tillegg B

Matlab-koder

B.1 Matlab-koder som jeg har fått fra veileder.

B.1.1 Koden for inndata, 'inndata1.m', som brukes i de andre kodene i oppgaven.

```
%S.I. Aanonsen.
%Define reservoir dimensions and various parameters
%for simple forward solver for single-phase flow in 2D.
nx = 20;           % No of gridcells in x-dir. Must be > 1
ny = 20;           % No of gridcells in y-dir. Must be > 1
nz = 1;           % No of gridcells in z-dir
n = nx*ny*nz;
dx = 10;
dy = 10;
dz = 10;
c = 1.0*10(-4); % Compressibility (1/bar)
mu = 1.0;          % Viscosity (cP)
dptarg = 2.0;      % Target maxdp over a time step
dtmin = 0.001;     % Minimum dt
dtmax = 0.001;     % Maximum dt
dt = dtmin;        % Initial dt
tmax = 0.02;       % Max simulation time
kmax = 20;         % Max no of time steps
tol = 1e-4;        % Tolerance for linear solver
maxit = 50;        % Maximum no of iterationf for linear solver
sigma_d=0.3;
```

```

N=40;           % No of ensemble member
I(1:N,1:N)=eye(N,N);
Z = (1/N)*ones(N,N);

```

B.1.2 Koden for å beregne korrelasjonen til permeabiliteten, 'corr-sphe.m'

```

%S.I. Aanonsen.
function r=corr_spher(ni,nj,lai,laj)
%Function for calculating correlation matrix for a ni*nj grid
%FOR A SPHERICAL VARIOGRAM MODEL
ninj=ni*nj;
for i=1:ninj
    for j=1:ninj
        ii=1+(i-1-mod((i-1),ni))/ni;
        jj=1+(j-1-mod((j-1),ni))/ni;
        iii=i-(ii-1)*ni;
        jjj=j-(jj-1)*ni;
        u=abs(iii-jjj);
        v=abs(ii-jj);
        h = sqrt((u/lai)^2+(v/laj)^2);
        if h < 1
            r(i,j)=1+0.5*(-3*h + h^3);
        else
            r(i,j)=0.0;
        end
    end
end
end

```

B.1.3 Koden for å beregne initialpermeabiliteten og den sanne permeabiliteten, 'cholesky.m'

```

%Code for generating stochastic realizations in 2D
%based on spherical variogram
clear all
% Number of realizations, N
inndata1
% Dimensions:
nif = 20
njf = 20

```

```
ninjf=nif*njf;
nf=nif*njf;
%Mean, Variance & Correlation lengths
meanf = 2
sigmaf = 0.5
laif = 5
lajf = 5
for i=1:nf
    mean(i) = meanf;
    for j=1:nf
        if i==j
            Sf(i,j)=sqrt(sigmaf);
        else
            Sf(i,j)=0.0;
        end
    end
end
end

%Calculate covariance matrix (Cf)
Rf=corr_spher(nif,njf,laif,lajf);
Cf=Sf*Rf*Sf;
%Cholesky decomposition
[L,p] = chol(Cf,'lower');
if p > 0
    error ('Cf not positive definite')
end

% Generate N realizations for prior ensemble and
% one "true" realization
for j = 1:N+1
    z = randn(nf,1);
    x = mean' + L*z;
    for i = 1:nf
        perm(i,j) = 10^x(i);
    end
end
hist(x)
tmp = perm(1:nf,N+1);
save perm_true.out tmp -ASCII
tmp = perm(1:nf,1:N);
save perm_init.out tmp -ASCII
```

B.1.4 Koden for å sette opp matrisen \mathcal{T} i to-dimensjoner, 'setup.m'.

```

%S.I. Aanonsen.
function [T,b] = setup(poro,perm,f,u0,dt,dx,dy,nx,ny)
% Sets up linear system of equations
% for single-phase system in 2D.
n = nx*ny;
T(1:n,1:n) = 0;
b(1:n) = 0;
% Set up matrix T
j = 1;
  i = 1;
    k = (j-1)*nx+i;
    k3 = (j-1)*nx + (i+1);
    k4 = (j)*nx + (i);
    tran1 = 0;
    tran2 = 0;
    tran3 = 2*perm(k3)*perm(k)/(perm(k3)+perm(k));
    tran4 = 2*perm(k4)*perm(k)/(perm(k4)+perm(k));
    a1 = 0;
    a2 = 0;
    a3 = dt*tran3/(poro(k3)*dx^2);
    a4 = dt*tran4/(poro(k4)*dx^2);
    T(k,k3) = -a3;
    T(k,k4) = -a4;
    T(k,k) = 1 + a1 + a2 + a3 + a4;
  for i = 2:nx-1
    k = (j-1)*nx+i;
    k2 = (j-1)*nx + (i-1);
    k3 = (j-1)*nx + (i+1);
    k4 = (j)*nx + (i);
    tran1 = 0;
    tran2 = 2*perm(k2)*perm(k)/(perm(k2)+perm(k));
    tran3 = 2*perm(k3)*perm(k)/(perm(k3)+perm(k));
    tran4 = 2*perm(k4)*perm(k)/(perm(k4)+perm(k));
    a1 = 0;
    a2 = dt*tran2/(poro(k2)*dy^2);
    a3 = dt*tran3/(poro(k3)*dx^2);
    a4 = dt*tran4/(poro(k4)*dx^2);
    T(k,k2) = -a2;
  end
end

```

```

    T(k,k3) = -a3;
    T(k,k4) = -a4;
    T(k,k) = 1 + a1 + a2 + a3 + a4;
end
i = nx;
k = (j-1)*nx+i;
k2 = (j-1)*nx + (i-1);
k4 = (j)*nx + (i);
tran1 = 0;
tran2 = 2*perm(k2)*perm(k)/(perm(k2)+perm(k));
tran3 = 0;
tran4 = 2*perm(k4)*perm(k)/(perm(k4)+perm(k));
a1 = 0;
a2 = dt*tran2/(poro(k2)*dy^2);
a3 = 0;
a4 = dt*tran4/(poro(k4)*dx^2);
T(k,k2) = -a2;
T(k,k4) = -a4;
T(k,k) = 1 + a1 + a2 + a3 + a4;
for j = 2:ny-1
    i = 1;
    k = (j-1)*nx+i;
    k1 = (j-2)*nx + (i);
    k3 = (j-1)*nx + (i+1);
    k4 = (j)*nx + (i);
    tran1 = 2*perm(k1)*perm(k)/(perm(k1)+perm(k));
    tran2 = 0;
    tran3 = 2*perm(k3)*perm(k)/(perm(k3)+perm(k));
    tran4 = 2*perm(k4)*perm(k)/(perm(k4)+perm(k));
    a1 = dt*tran1/(poro(k1)*dx^2);
    a2 = 0;
    a3 = dt*tran3/(poro(k3)*dx^2);
    a4 = dt*tran4/(poro(k4)*dx^2);
    T(k,k1) = -a1;
    T(k,k3) = -a3;
    T(k,k4) = -a4;
    T(k,k) = 1 + a1 + a2 + a3 + a4;
for i = 2:nx-1
    k = (j-1)*nx+i;
    k1 = (j-2)*nx + (i);
    k2 = (j-1)*nx + (i-1);

```

```

    k3 = (j-1)*nx + (i+1);
    k4 = (j)*nx + (i);
    tran1 = 2*perm(k1)*perm(k)/(perm(k1)+perm(k));
    tran2 = 2*perm(k2)*perm(k)/(perm(k2)+perm(k));
    tran3 = 2*perm(k3)*perm(k)/(perm(k3)+perm(k));
    tran4 = 2*perm(k4)*perm(k)/(perm(k4)+perm(k));
    a1 = dt*tran1/(poro(k1)*dx^2);
    a2 = dt*tran2/(poro(k2)*dy^2);
    a3 = dt*tran3/(poro(k3)*dx^2);
    a4 = dt*tran4/(poro(k4)*dx^2);
    T(k,k1) = -a1;
    T(k,k2) = -a2;
    T(k,k3) = -a3;
    T(k,k4) = -a4;
    T(k,k) = 1 + a1 + a2 + a3 + a4;
end
i = nx;
k = (j-1)*nx+i;
k1 = (j-2)*nx + (i);
k2 = (j-1)*nx + (i-1);
k4 = (j)*nx + (i);
tran1 = 2*perm(k1)*perm(k)/(perm(k1)+perm(k));
tran2 = 2*perm(k2)*perm(k)/(perm(k2)+perm(k));
tran3 = 0;
tran4 = 2*perm(k4)*perm(k)/(perm(k4)+perm(k));
a1 = dt*tran1/(poro(k1)*dx^2);
a2 = dt*tran2/(poro(k2)*dy^2);
a3 = 0;
a4 = dt*tran4/(poro(k4)*dx^2);
T(k,k1) = -a1;
T(k,k2) = -a2;
T(k,k4) = -a4;
T(k,k) = 1 + a1 + a2 + a3 + a4;
end
j = ny;
i = 1;
k = (j-1)*nx+i;
k1 = (j-2)*nx + (i);
k3 = (j-1)*nx + (i+1);
tran1 = 2*perm(k1)*perm(k)/(perm(k1)+perm(k));
tran2 = 0;

```

```

    tran3 = 2*perm(k3)*perm(k)/(perm(k3)+perm(k));
    tran4 = 0;
    a1 = dt*tran1/(poro(k1)*dx^2);
    a2 = 0;
    a3 = dt*tran3/(poro(k3)*dx^2);
    a4 = 0;
    T(k,k1) = -a1;
    T(k,k3) = -a3;
    T(k,k) = 1 + a1 + a2 + a3 + a4;
for i = 2:nx-1
    k = (j-1)*nx+i;
    k1 = (j-2)*nx + (i);
    k2 = (j-1)*nx + (i-1);
    k3 = (j-1)*nx + (i+1);
    tran1 = 2*perm(k1)*perm(k)/(perm(k1)+perm(k));
    tran2 = 2*perm(k2)*perm(k)/(perm(k2)+perm(k));
    tran3 = 2*perm(k3)*perm(k)/(perm(k3)+perm(k));
    tran4 = 0;
    a1 = dt*tran1/(poro(k1)*dx^2);
    a2 = dt*tran2/(poro(k2)*dy^2);
    a3 = dt*tran3/(poro(k3)*dx^2);
    a4 = 0;
    T(k,k1) = -a1;
    T(k,k2) = -a2;
    T(k,k3) = -a3;
    T(k,k) = 1 + a1 + a2 + a3 + a4;
end
i = nx;
k = (j-1)*nx+i;
k1 = (j-2)*nx + (i);
k2 = (j-1)*nx + (i-1);
tran1 = 2*perm(k1)*perm(k)/(perm(k1)+perm(k));
tran2 = 2*perm(k2)*perm(k)/(perm(k2)+perm(k));
tran3 = 0;
tran4 = 0;
a1 = dt*tran1/(poro(k1)*dx^2);
a2 = dt*tran2/(poro(k2)*dy^2);
a3 = 0;
a4 = 0;
T(k,k1) = -a1;
T(k,k2) = -a2;

```

```

    T(k,k) = 1 + a1 + a2 + a3 + a4;
% Set up right-hand side b
for k = 1:n
    b(k) = f(k)*dt + u0(k);
end
b = b';

```

B.1.5 Koden for å skrive ut griddata, 'write-griddata.m'.

```

%S.I. Aanonsen.
function error_flag = write_griddata(u,time)
% writes griddata to file
filename = ['griddata' sprintf('%g',time) '.dat'];
fid = fopen(filename,'wt');
fprintf(fid,'%e\n',u);
fclose(fid);
error_flag = 0;

```

B.1.6 Koden for å skrive ut sanne data, 'write-welldata.m'.

```

%S.I. Aanonsen.
function error_flag = write_welldata(u0,f,time,flag)
% writes welldata to file

well_indices = find(f);
n = length(well_indices);
if n == 0
    'No wells'
else
    for i = 1:n
        u(i) = u0(well_indices(i));
    end
    fid = fopen('welldata.dat',flag);
    if flag == 'wt'
        fprintf(fid,'%s\n','Time Well_pressues');
        fprintf(fid,'%s\n',sprintf('%13.4e',time,u));
    else
        fprintf(fid,'%s\n',sprintf('%13.4e',time,u));
    end
    fclose(fid);
end

```



```
error_flag = 0;
```

B.1.7 Koden for å løse varmeligningen som en forovermodell, 'forover.m'.

```
% S.I. Aanonsen, CIPR. Sept 2007
% Simple forward solver for single-phase flow in 2D
% u = Dimensionless pressure drop
% Equation units:
% pressure: bar
% length:    m
% time:      days

clear all
tic
% Define reservoir dimensions and various parameters
inndata1
print_times = [10, 100, tmax]; % Report times (days)
%print_times = load('report_times.dat'); % Report & data times (days)
%Input properties (Perm in mD)
poro = load('poro.dat');
perm = load('perm.dat');
% Normalize and multiply permeability with unit factor
perm = 0.00864*perm/(mu*c);
% Input sources/sinks (Rates in m3/d)
% File can be generated with wellrates.m
f = load('f.dat');
% Normalize sources/sinks
for i = 1:n
    f(i) = f(i)/(poro(i)*c*dx*dy*dz);
end
%Initial values
%p0 = read_props(nx,ny,'p0.dat');
for k = 1:n
    p0(k) = 0.0;
end
p0 = p0';
time = 0.0;
k = 0;
krep = 0;
```

```
reportflag = 0;
%Save values at sources/sinks
fileflag = 'wt';
write_error = write_welldata(p0,f,time,fileflag);
if write_error ~= 0
    write_error
    time
    error ('write_error welldata')
end
report_index = 1;
next_report_time = print_times(report_index)
while time < tmax;
    k = k + 1;
    if k > kmax
        error ('Max no of time steps exceeded')
    end
    if time + dt >= next_report_time
        dt = next_report_time - time;
        reportflag = 1;
        report_index = report_index + 1;
        if report_index <= length(print_times)
            next_report_time = print_times(report_index)
        end
    end
    time = time + dt;
    dt;
%Set pp linear system of equations
[A,b] = setup(poro,perm,f,p0,dt,dx,dy,nx,ny);
% Solve linear system of equations
p1 = A\b;
% [p1,flag] = pcg(T,b,tol,maxit);
% p1 = solve(poro,perm,f,p0,dt,dx,dy);
% Save values at sources/sinks
fileflag = 'at';
write_error = write_welldata(p1,f,time,fileflag);
if write_error ~= 0
    write_error
    time
    error ('write_error welldata')
end
%Save grid data at report steps
```

```
    if reportflag
        krep = krep + 1;
        report_error = write_griddata(p1,time);
        if report_error ~= 0
            report_error
            time
            error ('write error griddata')
        end
        ii = 0;
        for j = 1:nx
            for i = 1:ny
                ii = ii + 1;
                pplot(i,j) = p1(ii);
            end
        end
        figure(krep)
        surf(pplot)
        reportflag = 0;
    end
end
% Find next time step size
maxnorm = norm(p1-p0,inf);
if maxnorm == 0
    dt = dtmax;
elseif dt*dptarg/maxnorm > dtmax
    dt = dtmax;
elseif dt*dptarg/maxnorm < dtmin
    dt = dtmin;
else
    dt = dt*dptarg/maxnorm;
end
% Update solution
p0 = p1;
end
'End of simulation'
toc
```

B.2 Matlab-koder som jeg har laget i samarbeid med veileder.

B.2.1 Koden for Ensemble Kalman filteret, 'EnKF.m'.

```

%S.I. Aanonsen. og Anh-Hong.T.V
clear all
tic
inndata1
print_times = [10, 100, tmax];      %Report times (days)
poro = load('poro.dat');            %Input properties (Perm in mD)
for j=1:N
    perm(1:n,1:N) = load('perm_init.out');
end
f(1:n) = load('f.dat');
%Normalize and multiply permeability with unit factor
perm = 0.00864*perm/(mu*c);
%Normalize sources/sinks
for i = 1:n
    f(i) = f(i)/(poro(i)*c*dx*dy*dz);
end
ii=0;
for i=1:n
    if f(i) ~= 0
        ii=ii+1;
        iii(ii)=i;
    end
end
m=ii;
H(1:m,1:2*n)=0.0;
for l=1:m
    H(l,iii(l))=1;
end
H;
Cd(1:m,1:m)= 0.2 * eye(m,m);      %Cd is data error covarianse matrix
data = load('d.dat');
t= data(:,1);
d =data(:,2:m+1);
for j=1:N
    for l=1:m

```

```
        for k=1:length(t)
D(1,j,k)=d(k,1) + sigma_d *randn ;
        end
end
end
Nt=k;
for k = 1:n
    p0(k) = 0.0;
end
p0=p0';
Bf(1:2*n*Nt,1:N) = 0.0;
X(1:N,1:N) = 0.0;
krep = 0;
reportflag = 0;
%Save values at sources/sinks
fileflag = 'wt';
write_error = write_welldata(p0,f,time,fileflag);
if write_error ~= 0
    write_error
    time
    error ('write_error welldata')
end
report_index = 1;
next_report_time = print_times(report_index);
for k=1:Nt
    if k > kmax
        error ('Max no of time steps exceeded');
end
    if time + dt >= next_report_time
        dt = next_report_time - time;
        reportflag = 1;
        report_index = report_index + 1;
        if report_index <= length(print_times)
            next_report_time = print_times(report_index)
        end
    end
end
time = t(k);
for j=1:N
    if k >1
        p0(1:n)= Bf((2*(k-2)*n +1):(2*(k-2)*n+ n),j);
        perm(1:n,j) =exp(Bf((2*(k-2)*n + n+1):(2*(k-2)*n+ 2*n),j));
```

```

    end
p0;
perm(1:n,j);
[T,b] = setup(poro,perm(1:n,j),f,p0,dt,dx,dy,nx,ny);
p1 = T\b;
Bf((2*(k-1)*n+1):(2*(k-1)*n+n),j) = p1;
Bf((2*(k-1)*n+n+1):(2*(k-1)*n+2*n),j) =log(perm(1:n,j));
end
Bf1=Bf(((2*(k-1)*n+1):(2*k*n)),1:N);
DD(1:m,1:N)=D(:, :,k);
[X]= X_beregne(m,Cd,H,Bf1,Z,DD);
Bf(((2*(k-1)*n+1):(2*k*n)),1:N)= Bf1*X;
% Save values at sources/sinks
fileflag = 'at';
write_error = write_welldata(p0,f,time,fileflag);
if write_error ~= 0
    write_error
    time
    error ('write_error welldata')
end
% Save grid data at report steps
if reportflag
    krep = krep + 1;
    report_error = write_griddata(p1,time);
    if report_error ~= 0
        report_error
        time
        error ('write error griddata')
    end
    ii = 0;
for j = 1:nx
    for i = 1:ny
        ii = ii + 1;
        pplot(i,j) = p1(ii);
        pplot0(i,j) = p0(ii);
        pplotperm(ij)=perm(ii,N)*(mu*c)/0.00864;
    end
end
end
figure(krep)
surf(pplot)
figure(krep+10)

```

```

surf(pplot0)
figure(krep+20)
surf(pplotperm)
reportflag = 0;
end
end
perm = (mu*c*perm)/0.00864;
save perm_af.dat perm -ASCII
save Bf.dat Bf -ASCII
for k=1:Nt
    Bf1= Bf(((2*(k-1)*n+1):(2*k*n)),1:N);
    W = H*Bf1;
    p_ana(k,1:m,1:N) = W;
end
tmp2=[0;0];
for i= 1:m
    tmp(:,i) = p_ana(:,i,:);
    tmp2(1:Nt) = d(1:Nt,i);
    fig = 50 +i
    figure(fig)
    plot(tmp2)
    fig = 100 +i
    figure(fig)
    tmp3 = [tmp,tmp2];
    plot(tmp3)
end
tmp4(:,i)= D(m,1:N,1:Nt);
tmp5 = [tmp4',tmp2];
figure(113)
plot(tmp5)
figure(114)
plot(tmp2)
'End of simulation'

```

B.2.2 Koden for Ensemble Kalman Smoother, 'EnKS.m'.

```

%S.I. Aanonsen. og Anh-Hong.T.V
tic
% Define reservoir dimensions and various parameters
inndata1
print_times = [10, 100, tmax];           % Report times (days)

```

```
% Input properties (Perm in mD)
poro = load('poro.dat');
for j=1:N
perm(1:n,1:N) = load('perm_init.out');
end
f(1:n) = load('f.dat');
%Normalize and multiply permeability with unit factor
perm = 0.00864*perm/(mu*c);
% Input sources/sinks (Rates in m3/d)
% File can be generated with wellrates.m
% Normalize sources/sinks
for i = 1:n
    f(i) = f(i)/(poro(i)*c*dx*dy*dz);
end
ii=0;
for i=1:n
    if f(i) ~= 0
        ii=ii+1;
        iii(ii)=i;
    end
end
m=ii;
H(1:m,1:2*n)=0.0;
for l=1:m
    H(l,iii(l))=1;
end
H;
Cd(1:m,1:m)= 0.2 * eye(m,m);
data = load('d.dat');
t= data(:,1);
d =data(:,2:m+1);
for j=1:N
    for l=1:m
        for k=1:length(t)
D(l,j,k)=d(k,l) + sigma_d *randn ;
        end
    end
end
end
Nt=k;
D(1:m,1:N,1:kmax);
%Initial values
```



```

%p0 = read_props(nx,ny,'p0.dat');
for k = 1:n
    p0(k) = 0.0;
end
p0=p0';
Bs(1:2*n*Nt,1:N) = 0.0;
X(1:N,1:N) = 0.0;
time = 0.0;
%k = 0;
krep = 0;
reportflag = 0;
%Save values at sources/sinks
fileflag = 'wt';
write_error = write_welldata(p0,f,time,fileflag);
if write_error ~= 0
    write_error
    time
    error('write_error welldata')
end
report_index = 1;
next_report_time = print_times(report_index)
for k=1:Nt
    if k > kmax
        error('Max no of time steps exceeded')
    end
    if time + dt >= next_report_time
        dt = next_report_time - time;
        reportflag = 1;
        report_index = report_index + 1;
        if report_index <= length(print_times)
            next_report_time = print_times(report_index)
        end
    end
end
time = t(k);

for j=1:N
    if k > 1
        p0(1:n) = Bs((2*(k-2)*n + 1):(2*(k-2)*n + n),j);
        perm(1:n,j) = exp(Bs((2*(k-2)*n + n+1):(2*(k-2)*n + 2*n),j));
    end
end
p0;

```

```

perm(1:n,j);
[T,b] = setup(poro,perm(1:n,j),f,p0,dt,dx,dy,nx,ny);
p1 = T\b;
Bs((2*(k-1)*n+1):(2*(k-1)*n+n),j) = p1;
Bs((2*(k-1)*n+n+1):(2*(k-1)*n+2*n),j) =log(perm(1:n,j));
end
for kk=1:k
    kk
    Bf1=Bs(((2*(k-1)*n+1):(2*k*n)),1:N);
    DD(1:m,1:N)=D(:, :,kk);
    [X]= X_beregne(m,Cd,H,Bf1,Z,DD);
end
Bs(((2*(k-1)*n+1):(2*k*n)),1:N)= Bf1*X;
fileflag = 'at';
write_error = write_welldata(p0,f,time,fileflag);
if write_error ~= 0
write_error
time
error ('write_error welldata')
end
if reportflag
krep = krep + 1;
report_error = write_griddata(p1,time);
if report_error ~= 0
report_error
time
error ('write error griddata')
end
ii = 0;
for j = 1:nx
    for i = 1:ny
        ii = ii + 1;
        pplot(i,j) = p1(ii);
        pplot0(i,j) = p0(ii);
        pplotperm(ij)=perm(ii,N)*(mu*c)/0.00864;
    end
end
end
figure(krep)
surf(pplot)
figure(krep+10)
surf(pplot0)

```

```
figure(krep+20)

surf(pplotperm)
reportflag = 0;
    end
end
perm = (mu*c*perm)/0.00864;
save perm_as.dat perm -ASCII
save Bs.dat Bs -ASCII
for k=1:Nt
    Bf1= Bs(((2*(k-1)*n+1):(2*k*n)),1:N);
    W = H*Bf1;
    p_ana(k,1:m,1:N) = W;
end
tmp2=[0;0];
for i= 1:m
    tmp(:,i) = p_ana(:,i,:);
    tmp2(1:Nt) = d(1:Nt,i);
    fig = 50 +i
    figure(fig)
    plot(tmp2)
    fig = 100 +i
    figure(fig)
    tmp3 = [tmp,tmp2];
    plot(tmp3)
end
tmp4(:,i)= D(m,1:N,1:Nt);
tmp5 = [tmp4',tmp2];
figure(113)
plot(tmp5)
figure(114)
plot(tmp2)
'End of simulation'
```

B.3 Matlab-koder fra Anh-Hong T.V

B.3.1 Koden for å sette opp matrisen \mathcal{T} i en-dimensjon

```
% Anh-Hong.T.V
clear all;
vk0 =load('t0.dat');
%Innles verdier for permabilitet.
n = 6;
%Beregn b(1)med vk0(1-1)= vk0(0)=0,
b(1) = 1+0.5*((vk0(1+1)*vk0(1))/(vk0(1)+vk0(1+1)));
b(6) = 1-0.5*((vk0(6-1)*vk0(6))/(vk0(6-1)+vk0(6)));
for i = 2:n
    a(i) = 0.5*((vk0(i)*vk0(i-1))/(vk0(i)+vk0(i-1)));
end
for i = 2:n-1
b(i) = 1+0.5*((vk0(i+1)*vk0(i))/(vk0(i)+vk0(i+1)))
-0.5*((vk0(i-1)*vk0(i))/(vk0(i-1)+vk0(i)));
end
for i = 1:n-1
c(i) = -(0.5*((vk0(i+1)*vk0(i))/(vk0(i)+vk0(i+1))));
end
T = diag(b)+diag(c,+1)+diag(a(2:n),-1);      % Det er den samme matrise T for alle u
vu0 = load('t0_1.dat');                      % Innlese av initial verdien(ved t0)
vf0 = load('t0_2.dat');
%Bruke formen(implicit): T*u(m+1)= d= u(m)
m=4;
u(1:n,1) = T\vu0 + vf0;
for j = 1:m
    u(1:n,j+1) = T\u(1:n,j) + vf0;
end
```

B.3.2 Koden for å beregne matrisen X , 'X-beregne.m'

```
% Anh-Hong.T.V
function [X]=X_testen(m,Cd,H,Bf1,Z,DD,I)
inndata1
BB = (Bf1 * (I-Z));          %I=Cd=(NxN)
S = H * BB ;                %S=(mxN) og S'=(Nxm)
C =S * S' + (N-1)* Cd ;    %Cd=(mxm)
delta_D= DD -(H*Bf1) ;     %D=(1:m,1:N,1:kmax) og delda_D = (mXN)
```

```
X = I+S'*C^(-1)*(delta_D); %X=(NxN)
end
```

B.3.3 Koden for å sammeligne verdiene til permeabilitet.

```
% Anh-Hong.T.V
inndata1
perm_t = load('perm_true.out');% Sanne verdier
u1(1:n)=perm_t;
%Initialpermeabilitet som brukes i forover.m
perm_i=load('perm_init.out');
E(1:n,1:N)= perm_i;
u2(1:n)=mean(E,2); % middelsverdien av de N settene initialpermeabilitet.
%NB! u1(sanne verdier) og u2 (initialverdier) er de samme for EnKS og EnKF
save u2.dat u2 -ASCII
save perm_i.dat perm_i -ASCII
%BEREGNING AV FEIL-KOVARIANSMATRSEN
%Varians til initialpermeabilitet
B1=log(E);
L1=B1*(I-Z);
L2=L1';
P1=(L1*L2)/(N-1);
v1(1:n)=diag(P1); % v1 er den samme for EnKF og EnKS
save v1.dat v1 -ASCII
%For EnKS
perm_as=load('perm_as.dat'); %Analyseverdier ved kmax for EnKS.
Fs(1:n,1:N)=perm_as;
u3s(1:n)=mean(Fs,2);
save u3s.dat u3s -ASCII
Bs = load('Bs.dat');
B1s=Bs((2*n*kmax-n+1):(2*n*kmax),1:N);
Ls=B1s*(I-Z);
L1s=Ls';
Ps=(Ls*L1s)/(N-1);
% Ps er feil-kovariansmatrise-matrise
v2s(1:n)=diag(Ps);
v3s=mean(v2s);
save v2s.dat v2s -ASCII
save v3s.dat v3s -ASCII
```

```
%For EnKF
perm_af=load('perm_af.dat'); %Analyseverdier ved kmax for EnKF.
Ff(1:n,1:N)=perm_af;
u3f(1:n)=mean(Ff,2);
save u3f.dat u3f -ASCII
Bf = load('Bf.dat');
B1f=Bf((2*n*kmax-n+1):(2*n*kmax),1:N);
Lf=B1f*(I-Z);
L1f=Lf';
Pf=(Lf*L1f)/(N-1);
v2f(1:n)=diag(Pf);
v3f=mean(v2f);
save v2f.dat v2f -ASCII
save v3f.dat v3f -ASCII
ii = 0;
    for j = 1:nx
        for i = 1:ny
            ii = ii + 1;
            u1plot(i,j) = u1(ii);
            u2plot(i,j) = u2(ii);
            u3splot(i,j) = u3s(ii);
            u3fplot(i,j) = u3f(ii);
        end
    end
    figure(10)
    surf(u1plot)
    figure(11)
    surf(u3splot)
    figure(12)
    surf(u3fplot)
```

Bibliografi

- [1] Ø. Pettersen. *Grunnkurs i reservoarmekanikk*. Matematisk institutt, Universitetet i Bergen, 1990.
- [2] I. Aavatsmark. *Bevarelsesmetoder for elliptiske differensialligninger*. Studieretning beregningsvitenskap, UiB, 2004.
- [3] G.D. Smith. *Numerical Solution of Partial Differential Equations : Finite Difference Methods*, chapter Introduction and Finite-difference Formulae. Oxford Applied mathematics and Computing Science Series, 1985.
- [4] G. Evensen. *Data Assimilation, The Ensemble Kalman Filter*. Springer, 2007. ISBN-13-978-3-504-38300-0.
- [5] S.C. Dunne and D. Entekhabi. Reanalysis Land Data Assimilation Using Ensemble Techniques. [http : //ams.confex.com/ams/pdfpapers/72811.pdf](http://ams.confex.com/ams/pdfpapers/72811.pdf). Mass. Inst. of Tech., Dept. of Civil and Environmental Engineering, Cambridge.
- [6] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME-Journal of Basic Engineering*, 1960.
- [7] G. Welch and G Bishop. An Introduction to the Kalman Filter. [http : //www.cs.unc.edu/welch/kalman/kalmanIntro.html](http://www.cs.unc.edu/welch/kalman/kalmanIntro.html).
- [8] Max Welling. The Kalman Filter. [http : //www.cs.toronto.edu/welling/classnotes/classnotes.html](http://www.cs.toronto.edu/welling/classnotes/classnotes.html).
- [9] C.J. Johns and J. Mandel. A Two-Stage Ensemble Kalman Filter for Smooth Data Assimilation. [http : //math.cudenver.edu/jmandel/papers/cjgm-fires-revised.pdf](http://math.cudenver.edu/jmandel/papers/cjgm-fires-revised.pdf), 2005. Department of Mathematics, University of Colorado, Denver.

-
- [10] Aanonsen, S.I. and Nævdal, G. and Oliver, D.S. and Reynolds, A.C. and Valls, B. Review paper: Ensemble Kalman Filter to Petroleum Engineering. Centre for Integrated Petroleum Research, University of Bergen, International Research Institute of Stavanger, University of Oklahoma, University of Tulsa, march 2008.
- [11] Ensemble Kalman filter. *http* : [//en.wikipedia.org/wiki/Ensemble_Kalman_filter](http://en.wikipedia.org/wiki/Ensemble_Kalman_filter).
- [12] A. Almendral-Vazquez and A.R. Syversveen. The Ensemble Kalman Filter-theory and applications in oil industry. Technical Report SAND/05/06, Norwegian Computing Center, NR, 2006.
- [13] J. Zhu and P. Wang. Ensemble Kalman smoother and ensemble Kalman filter approaches to the joint air quality state and emission estimation problem. *Geophysical Research Abstracts*, 9, 2007. SRef-ID: 1607-7962/gra/EGU2007-A-01722.
- [14] E.R. Walpole, H.R. Myers, and L.S. Myers. *Probability & Statistics for Engineer & Scientist*. Pearson Education International, 2002. ISBN 0-13-098469-8.
- [15] R.C. Aster, W. Borchers, and C.H. Thurber. *Parameter Estimation and Inverse Problems*, chapter B.5 The Multivariate Normal Distribution , pages 264–265. Elsevier Academic Press, 2005. ISBN 0-12-065604-3.
- [16] D.C. Lay. *Linear Algebra and its Applications*. Pearson Education, third edition, 2003. ISBN 0-321-14992-0.
- [17] D. Kincaid and W. Cheney. *Numerical Analysis.*, chapter Numerical Solution of Ordinary Differential Equations, page 533. The University of Texas at Austin, 2001. ISBN: 0-534-38905-8.
- [18] Korrelasjon og kovarians. *http* : [//www.kunnskapssenteret.com](http://www.kunnskapssenteret.com).
- [19] S. Särkkä, A. Vehtari, and J Lampinen. Cats benchmark time series prediction by kalman smoother with cross-validated noise density. *Neurocomput.*, 70(13-15):2331–2341, 2007.
- [20] G. Evensen. EnKF-The Ensemble Kalman Filter. *http* : [//enkf.nersc.no/](http://enkf.nersc.no/), 2007. Nansen Environmental and Remote Sensing Center. Bergen, Norway.
- [21] Kalman filter-Wikipedia the free encyclopedia. *http* : [//en.wikipedia.org/wiki/Kalman_filter..](http://en.wikipedia.org/wiki/Kalman_filter..)

-
- [22] P.J Burger, G.van Leeuwen and G. Evensen. On the Analysis Scheme in the Ensemble Kalman Filter. *Mon. Wheather Rev.*126:1719-1724,1998.
- [23] An Introduction to Ensemble Methods. [http : //math.u - bourgogne.fr](http://math.u-bourgogne.fr).
- [24] J.S. Whitaker and G.P. Compo. An ensemble Kalman smoother for analysis. /*Proc. Symp. on Observations, Data Assimilation and Probabilistic Prediton*,/Orlando, FL,Amer. Meteror. Soc., 144-147, 2002.