Innføring
i
RESERVOARSIMULERING

Svein M. Skjæveland
og
Jann-Rune Ursin

Høgskolen i Stavanger

August 2000

Innhold

I Teoretisk grunnlag **1**

1 Opplegg **2**
1.1 Undervisningsopplegg 2
1.2 Faglig omfang 2
1.3 Litteratur 3
1.4 Fontkonvensjon 3
1.5 Fortranversjon 4

2 Diffusivitetsligningen **5**
2.1 Darcy's lov og massebevarelse 5
2.2 Generell diffusivitetsligning 8
2.3 Diffusivitetsligning for tre faser 9

3 Numerisk formulering **10**
3.1 Tilnærming av deriverte 10
3.2 Avstandsderiverte i diffusivitetsligningen 11
3.3 Diskretisering i tid og rom 13
3.4 Valg av numerisk formulering 14
3.5 Eksplisitt formulering 15
3.6 Implisitt formulering 15
3.7 Iterativ løsningsmetode 17
3.8 Direkte løsningsmetode 18
3.9 Skisse til flytskjema for utlevert program 20

II Fortrankode for utlevert program **22**
3.10 Module 32
3.11 Module 32
3.12 Datafil 32
3.13 Resultatfil 33

III Øvingsoppgaver **36**

1 Kjøring av utlevert program **37**
1.1 Utprøving av løsningsrutiner 37

2 Automatisk tidsstegsregulering **38**
2.1 Først et par justeringer 38
2.2 Ny tidsstegsregulering 38
2.3 Kjør programmet for testing 39
2.4 Praktiske råd 40

3 Sammenligning med analytisk løsning **41**
3.1 Problemstilling 41
3.2 Analytisk løsning 44

4 Numeriske formuleringer og `SOR` **48**
4.1 Problemstilling 48
4.2 Kommentarer 50

5 Variabel k, ϕ og Δx **53**
5.1 Oppgave 53
5.2 Kommentarer 53

6 Hellende reservoar **56**
6.1 Oppgave 56
6.2 Kommentarer 56

7 Radielt system **59**
7.1 Oppgave 59
7.2 Teori 60
7.3 Kommentarer 62

8 Tabellbruk **66**
8.1 Oppgave 66
8.2 Kommentarer 66

9 En fase i to dimensjoner **71**
9.1 Kommentarer til Øving 9 73
9.2 Løsning med naturlig nummerering og subrutinen SOLVE 80
9.3 Løsning med LSOR 85
9.4 Noen påminnelser 88

10 To faser, olje og vann, i en dimensjon **89**
10.1 Kommentarer til Øving 10 92
10.2 IMPES-metoden 93
10.3 Oppstrøms relative permeabiliteter 94
10.4 Fullstendig implisitt formulering 95
10.5 Kordemetoden – kvasi-Newton 96
10.6 Vannligningen i detalj — oppsummering 99
10.7 Ligningssystem 102
10.8 Rateformulering 102
10.9 Noen hint til programmering 104
10.10 Andre kommentarer 107

11 Måling av relative permeabiliteter **109**
11.1 Innledning 109
11.2 Framgangsmåte 110
11.3 Data 111
11.4 Forventede resultater 111

IV Eksamensoppgaver **113**

Del I

Teoretisk grunnlag

Kapittel 1

Opplegg

1.1 Undervisningsopplegg

Dette kompendiet dekker pensum i faget *TE 6192 Reservoarsimulering, innføringskurs* ved Institutt for Petroleumsteknologi. Undervisningen er praktisk lagt opp med øvinger på datamaskin.

Etter en kortfattet gjennomgåelse av det teoretiske grunnlaget blir det utdelt et `fortran` program som simulerer enfasestrøm i en dimensjon. Programkoden blir gjennomgått systematisk. På dette grunnlag skal hver student utvikle sin egen simuleringsmodell ved å utvide den utdelte modellen gjennom øvingsoppgavene. Noen av oppgavene vil være obligatoriske og skal leveres inn for godkjenning.

I de første ukene blir det forelesninger i 4 timer per uke, inntil det teoretiske grunnlaget er gjennomgått. Deretter vil det bli delt ut en ny øving hver uke. Denne blir gjennomgått med forslag og hint til løsning i løpet av 2 forelesningstimer. De resterende to undervisningstimene benyttes til veiledning på terminalrommet. Det reserveres tid på terminalrom for hver student tilsvarende en formiddag per uke, men ellers er det åpent for den enkelte å benytte terminalene etter behov.

1.2 Faglig omfang

Det tas utgangspunkt i en simuleringsmodell (et `fortran`-program) som løser numerisk den differensialligningen som styrer horisontal strøm av en fase i en dimensjon. Reservoaret har uniforme egenskaper, og den numeriske formuleringen som løser differensialligningen er av enkleste slag. *Del I* av kompendiet omhandler det teoretiske grunnlag for denne ligningen. I *Del II* inngår `fortran`-koden med inngangsfil og resultatfil og i *Del III* inngår øvingsoppgavene som brukes til å utvide modellen. Følgende emner vil bli behandlet på denne måten, så langt tiden rekker:

1. Utprøving av utlevert program.

2. Automatisk tidsstegsregulering

3. Sammenligning med analytisk løsning.
4. Ulike numeriske formuleringer og løsningsmetoder.
5. Varierende blokklengde, permeabilitet og porøsistet.
6. Hellende reservoar.
7. Radiell strøm.
8. Tabellbruk.
9. Strøm av en fase i to dimensjoner.
10. Strøm av to faser i en dimensjon.
11. Måling av relative permeabiliteter.

I *Del IV* er tatt med eksamenssett med utkast til løsninger for noen av de siste årene.

1.3 Litteratur

Som støttebok anbefales boken til Aziz og Settari [1]. Det kan ellers svare seg å ha tilgang på en håndbok i `FORTRAN` og eventuelt lærebøker i numeriske metoder, selv om dette neppe er nødvendig for faget som sådant.

På biblioteket er det konferanserartikler fra de siste symposiene i numerisk reservoarsimulering. Der vil en få inntrykk av hvilke problemstillinger som er aktuelle innen denne grenen av petroleumsteknologi. Dessuten er det publisert mange simuleringsstudier i den ordinære petroleumslitteraturen.

1.4 Fontkonvensjon

Et kompendium som dette bør i hovedsak holde seg til tre typer fonter. Vanlig tekst skrives som dette. Latinske bokstaver som brukes som matematiske symboler skrives med kursiv: a, b, c. I tillegg vil vi få bruk for å angi `FORTRAN` kode. Variable og utsagn i `FORTRAN` skrives med skrivemaskintyper som følger: `if (oxmin .lt. 0.d0) oxmin = 0.d0`.

Det er forsøkt å velge `FORTRAN` variabelnavn som er lette å huske, som ligner på det matematiske symbolet for samme variable. For eksempel brukes `po` for oljetrykket p_o.

Eksemplene på kode er noen ganger skrevet med store bokstaver, `FORTRAN`, og andre ganger med små, `fortran`, uten særlig systematikk. Moderne `FORTRAN`-kompilatorer skiller ikke mellom store og små bokstaver.

1.5 Fortranversjon

Fra høsten 1998 vil kurset i hovedsak bli gjennomført med Fortran-90 til erstatning for Fortran-77. Programvaren som blir brukt er Digital Visual Fortran for Win95 som kan kjøre både F90 og F77 siden F77 nære på er en komplett undermengde av F90. Programmet har en god hjelpefunksjon både for F90 generelt og for selve programbruken. Andre innføringsopplegg i F90 vil ligge utlagt på http://www.ux.his.no/~s-skj/. En bør være oppmerksom på at flere av kodeforslagene kan innholde rester etter F77 og det er ikke sikkert at alle vil virke uten videre.

Kapittel 2

Diffusivitetsligningen

Vi skal utlede diffusivitetsligningen på to litt ulike måter, først ved å starte med Darcy's lov og massebevarelse, og deretter mer direkte fra den generelle diffusivitetsligning.

Vi begrenser oss til såkalt *beta*-formulering. Det vil si at fluidene kan beskrives ved hjelp av de tradisjonelle volumfaktorene B_o, R_{so}, B_g, R_{sg}, B_w, som alle kun er funksjoner av trykket. Dette medfører at modellen er lite egnet til å simulere prosesser i reservoaret dersom det er store endringer i sammensetningen av fasene. Uttrykket *beta*-formulering kommer av at volumfaktorene tidligere ble betegnet med gresk bokstav β.

Vi skal starte med lineær, horisontal strøm av en fase i en dimensjon i et reservoar med konstant porøsitet og permeabilitet.

2.1 Darcy's lov og massebevarelse

Darcy's lov kan uttrykkes på følgende måte:

$$u_o = -\frac{Ck_x}{\mu_o}\frac{\partial p_o}{\partial x}, \qquad (2.1)$$

hvor

C : Omregningskonstant med verdi 0.00632827 for de gitte enhetene.

u_o : volumhastigheten, ft/d.

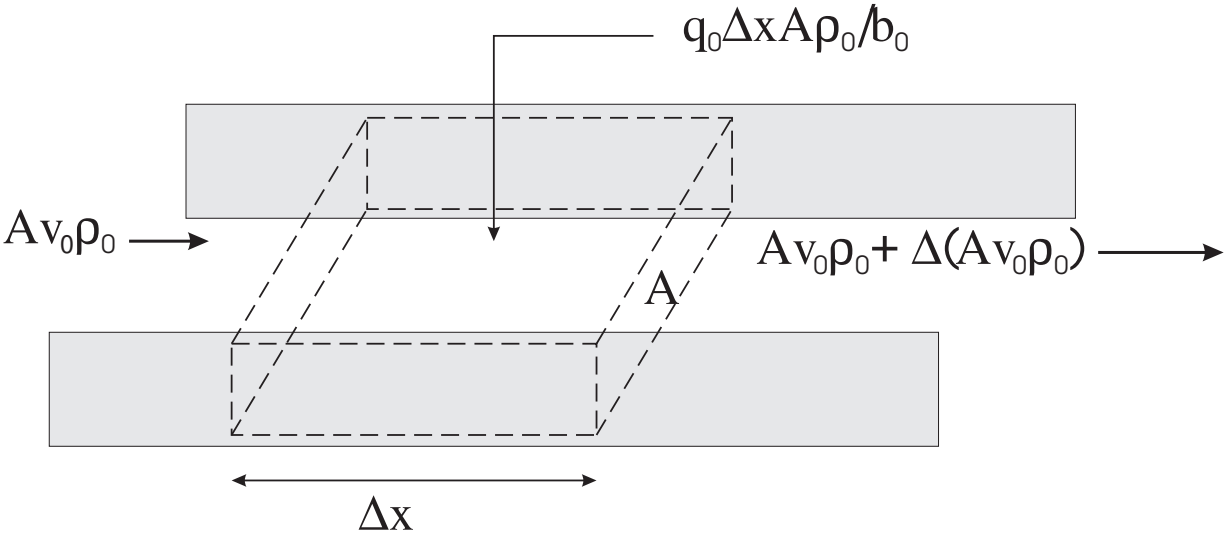
k_x : absolutt permeabilitet i x-retning, md.

μ_o : viskositet, cp.

p_o : trykk, psi.

x : avstand langs strømningsretningen, ft.

Indeks o står for olje og er egentlig overflødig før vi tar med strøm av flere faser. Kontinuitetsligningen kan utledes ved å betrakte massestrøm inn og ut av volumelementet $\Delta V = A\Delta x$ i Figur 2.1.



Figur 2.1: Masserate inn og ut av volumelement $A\Delta x$.

I Figur 2.1 er det brukt endel nye størrelser:

A : tverrsnitt av det en-dimensjonale reservoaret, ft^2.

q_o : oljerate per bulkvolum, stb/d/bbl. Raten er positiv for injeksjon og negativ for produksjon. Dette er kildeleddet i kontinuitetsligningen [2]. [1]

b_o : volumfaktor for olje, stb/rb.[2]

ρ_o : tetthet av olje ved reservoarbetingelser, lb/rft^3.

Endring av masse per tidsenhet i volumelementet ΔV på grunn av strøm inn eller ut, samt injeksjon eller produksjon er gitt ved

$$A(\rho_o u_o + \frac{q_o \Delta x \rho_o}{b_o} - (u_o \rho_o + \Delta(u_o \rho_o))). \qquad (2.2)$$

Masse tilstede i volumelementet er $A\Delta x \phi \rho_o$ og endring av denne massen per tidsenhet blir

$$\frac{\partial}{\partial t}(A\Delta x \phi \rho_o). \qquad (2.3)$$

[1]Merk at bbl her betegner volumenheten 'fat' av bergartsvolum.

[2]Merk at denne definisjonen er den inverse av den vanlige volumfaktoren B_o og at rb betegner "reservoir barrel" av fluid.

Settes uttrykket 2.2 lik uttrykket 2.3, og antas porøsiteten ϕ konstant, får en etter litt forenkling

$$\Delta x \phi \frac{\partial \rho_o}{\partial t} = -\Delta(u_o \rho_o) + \frac{q_o \Delta x \rho_o}{b_o}.$$

Vi deler nå ligningen med Δx og lar $\Delta x \to 0$, og får

$$\phi \frac{\partial \rho_o}{\partial t} = -\frac{\partial(\rho_o u_o)}{\partial x} + \frac{q_o \rho_o}{b_o}.$$

Denne ligningen multipliserer vi med b_o/ρ_o som har dimensjonen stb/rb $\times$ rft^3/lb $\to$ 5.61 $\cdot$ stb/lb. Dette er en konstant og kan tas innenfor derivasjonstegnet:

$$\phi \frac{\partial \left(\rho_o \frac{b_o}{\rho_o} \right)}{\partial t} = -\frac{\partial \left(\rho_o \frac{b_o}{\rho_o} u_o \right)}{\partial x} + \frac{q_o \rho_o}{b_o} \frac{b_o}{\rho_o},$$

som ordnet gir

$$\frac{\partial(b_o u_o)}{\partial x} - q_o = -\phi \frac{\partial b_o}{\partial t}. \qquad (2.4)$$

Ligning 2.4 er kontinuitetsligningen skrevet på en noe spesiell form med volumfaktor istedenfor tetthet. Det er imidlertid den vanlige framstillingsmåten for β–modeller. Ligningen uttrykker bevarelse av overflatevolum av olje. Dette er det samme som massebevarelse når sammensetningen av oljen er konstant.

I ligning 2.4 setter vi nå inn for volumhastigheten u_o fra ligning 2.1 og får

$$\frac{\partial}{\partial x} \left(\frac{C k_x b_o}{\mu_o} \frac{\partial p_o}{\partial x} \right) + q_o = \phi \frac{\partial b_o}{\partial t}.$$

Siden vi har antatt at volumfaktoren b_o kun avhenger av trykket, kan vi skrive

$$\frac{\partial b_o}{\partial t} = \frac{\partial b_o}{\partial p_o} \frac{\partial p_o}{\partial t},$$

og dermed

$$\boxed{\frac{\partial}{\partial x} \left(\frac{C k_x b_o}{\mu_o} \frac{\partial p_o}{\partial x} \right) + q_o = \phi \frac{\partial b_o}{\partial p_o} \frac{\partial p_o}{\partial t}.} \qquad (2.5)$$

Dette er diffusivitetsligningen vi skal løse numerisk. Den gjelder under de forutsetninger som er gitt innledningsvis i dette kapitlet.

Ligning 2.5 uttrykker at strøm i x-retning pluss produksjon eller injeksjon er lik ekspansjon eller kompresjon.

I den utleverte `FORTRAN`-koden, som løser ligning 2.5, er det brukt symbolene `A2` = $\phi \partial b_o / \partial p_o$ og `A9` = q_o.

Enheten til leddet på høyre siden av ligningen, $\phi(\partial b_o/\partial t)$, er stb/rb/d. Dette er i samsvar med enheten til q_o. Dersom en praktisk rate q_o' med enhet stb/d blir produsert fra et volumelement ΔV, så blir kildeleddet i ligning 2.5 gitt ved $q_o = q_o'/(\Delta V/\texttt{FPB})$, hvor `FPB` = 5.6146 [ft^3/bbl] og ΔV er i ft^3.

2.2 Generell diffusivitetsligning

Den generelle kontinuitetsligning er beskrevet i kompendiet til Papatzacos [2] og i standard tekster i fluidmekanikk,

$$\frac{\partial}{\partial x}(\rho_o u_o) = -\frac{\partial}{\partial t}(\phi \rho_o). \qquad (2.6)$$

I tillegg kan det være et kildeledd. Tettheten er gitt ved

$$\rho_o = b_o(\rho_o^{st} + \rho_g^{std} R_{so}/\texttt{FPB}), \qquad (2.7)$$

hvor ρ_o^{st} er oljetettheten ved *stock tank* forhold i lb/ft^3, ρ_g^{std} er gasstettheten ved *standard* forhold i lb/ft^3, og R_{so} er oppløst gass-olje forhold i ft^3/stb.

Ligning 2.7 kan utledes ved å betrakte hvordan en gitt masse m fordeler seg mellom gass- og væskefasen ved reservoarforhold og ved overflateforhold. Ved reservoarforhold er all masse i oljefasen, $m = \rho_{or} V_{or}$, hvor r betegner reservoarbetingelser og V volumet.

> **Merk:** Modellen er for enfase strøm. For olje betyr det at trykket i reservoaret må være over kokepunktstrykket for oljen, ellers vil det også være gass tilstede, og da må en bruke tofase-formulering: en konserveringsligning for gassen og en for oljen.

Samme masse m kan ved overflaten (indeks s for "surface") uttrykkes ved $m = V_{os}\rho_{os} + V_{gs}\rho_{gs}$, hvor V_{os} er tatt ved *stock tank* forhold og V_{gs} er tatt ved *standard* forhold på overflaten. Videre er $V_{gs} = V_{os}R_{so}/5.61$ og $b_o = V_{os}/V_{or}$. Satt sammen så gir dette ligning 2.7.

Vi antar nå igjen at ϕ er konstant og kan settes utenfor derivasjonstegnet. Setter en inn for ρ_o i ligning 2.6 og for u_o fra ligning 2.1, så får en

$$\frac{\partial}{\partial x}\left(b_o(\rho_o^{st} + \rho_g^{std} R_{so}/\texttt{FPB})\frac{Ck_x}{\mu_o}\frac{\partial p_o}{\partial x}\right) = \phi\frac{\partial}{\partial t}\left(b_o(\rho_o^{st} + \rho_g^{std} R_{so}/\texttt{FPB})\right).$$

Siden ρ_o^{st}, ρ_g^{std} og R_{so} er konstanter, reduseres dette uttrykket direkte til ligning 2.5. At R_{so} er en konstant følger av at vi betrakter bare en fase, i form av olje. Siden det ikke er fri gass tilstede i reservoaret, må trykket være over kokepunktstrykket, og da er R_{so} konstant.

> **Merk:** Oppgave 1 i eksamenssettene fra desember 1995 og desember 1996 gir to eksempler på utledning av diffusivitetsligningen dersom det antas hvordan tettheten avhenger av trykket (tilstandsligning).

2.3 Diffusivitetsligning for tre faser

Uten utledning tar vi her med utvidelsen av ligning 2.5 til strøm av olje, vann og gass i en dimensjon. Vi har da tre diffusivitetsligninger,

$$\begin{aligned}
\frac{\partial}{\partial x}\left(Ck_x\frac{k_{ro}b_o}{\mu_o}\frac{\partial p_o}{\partial x}\right)+q_o &= \phi\frac{\partial}{\partial t}(b_oS_o),\\
\frac{\partial}{\partial x}\left(Ck_x\frac{k_{rw}b_w}{\mu_w}\frac{\partial p_w}{\partial x}\right)+q_w &= \phi\frac{\partial}{\partial t}(b_wS_w),\\
\frac{\partial}{\partial x}\left(Ck_x\frac{k_{rg}b_g}{\mu_g}\frac{\partial p_g}{\partial x}\right)+\frac{\partial}{\partial x}\left(Ck_x\frac{k_{ro}b_o}{\mu_o}R_{so}\frac{\partial p_o}{\partial x}\right)+q_g &= \phi\frac{\partial}{\partial t}(b_gS_g+b_oR_{so}S_o),
\end{aligned}$$

og de tre føringsbetingelsene

$$\begin{aligned}
S_o+S_w+S_g &= 1,\\
p_g &= p_o+P_{cgo},\\
p_w &= p_o-P_{cow}.
\end{aligned}$$

Her står S for metning, P_{cgo} er kapillartrykket mellom gass og olje og P_{cow} mellom olje og vann. Det er også enkelt å ta med ledd som behandler olje oppløst i gassen, slik at en kan simulere et gasskondensat felt med denne β-formuleringen.

Funksjonssammenhengene i ligningene er slik at b,μ,R_{so} kun avhenger av trykket, og de metningsavhengige parametrene avhenger av

$$\begin{aligned}
&P_{cow}:P_{cow}(S_w), && k_{ro}:k_{ro}(S_w,S_g),\\
&P_{cgo}:P_{cgo}(S_g), && k_{rw}:k_{rw}(S_w),\\
& && k_{rg}:k_{rg}(S_g).
\end{aligned}$$

Vi skal i første omgang holde oss til ligning 2.5 med begrensningene at b_o varierer lineært med trykket gitt ved $b_o = b_{orig} + \texttt{dbdp}(p_o - p_{orig})$, hvor `dbdp` er konstant og betegner `FORTRAN`-symbolet for $\partial b_o/\partial p_o$, og indeks *orig* betegner startverdi. I tillegg skal vi anta at viskositeten μ_o er en konstant.

Kapittel 3

Numerisk formulering

Diffusivitetsligningen kan løses analytisk kun for enkle geometrier og grensevilkår, og med visse forenklinger. Et eksempel på dette er gitt i en av øvingene. I det alminnelige tilfellet må en begrense seg til numeriske løsninger, og for denne typen partielle differensialligninger finnes det to hovedklasser av metoder:

1. Differansemetoden
2. Elementmetoden

Elementmetoden brukes ofte til simulering av statiske belastninger og sjelden til simulering av strøm i et reservoar. Vi skal holde oss til differansemetoden hvor hovedpoenget er at alle deriverte tilnærmes med et forhold mellom differanser.

3.1 Tilnærming av deriverte

Ligning 2.5 inneholder deriverte med hensyn på både tid og avstand. I differansemetoden tilnærmes begge som vist under, for en vilkårlig funksjon $y(x)$ framstilt i Figur 3.1.

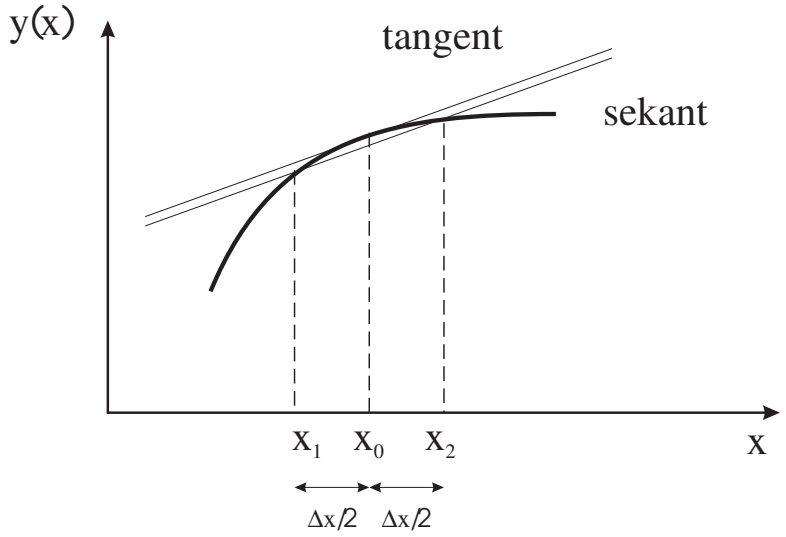
Den deriverte i punktet $(y(x_o), x_o)$ er gitt ved

$$\left.\frac{dy}{dx}\right|_{x=x_o} = \lim_{\Delta x \to 0} \frac{y(x_o + \Delta x/2) - y(x_o)}{\Delta x/2},$$

og den tilnærmes numerisk ved hjelp av differanser. Det vil si at vi tilnærmer tangentens stigningsforhold med sekantens på følgende måte

$$\left.\frac{dy}{dx}\right|_{x=x_o} \approx \frac{y(x_o + \Delta x/2) - y(x_o - \Delta x/2)}{\Delta x} = \frac{y(x_2) - y(x_1)}{\Delta x}.$$

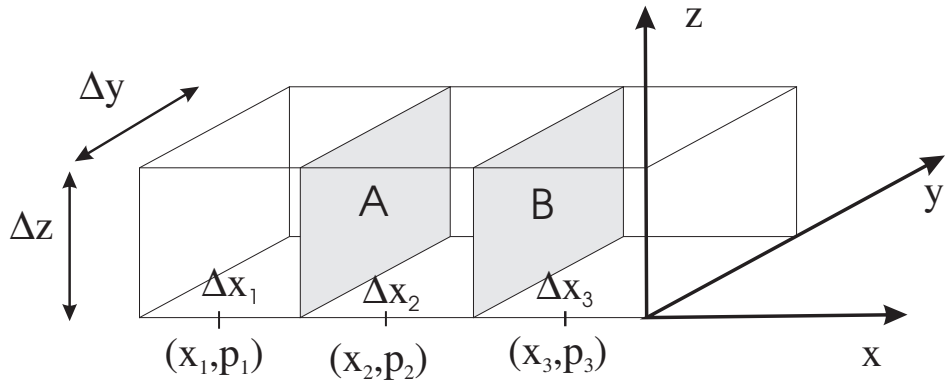
Dette svarer til å ta med kun første ordens ledd i en Taylorutvikling av $y(x)$ omkring punktet x_0.



Figur 3.1: Tilnærming av tangent ved sekant.

3.2 Avstandsderiverte i diffusivitetsligningen

Vi skal først se hvordan en kan tilnærme deriverte med hensyn på avstanden x i ligning 2.5. La oss dele det en-dimensjonale reservoaret inn i tre numeriske blokker, slik som vist i Figur 3.2.



Figur 3.2: Endimensjonalt reservoar med tre numeriske blokker.

I denne figuren betegner Δx_i lengden av numerisk blokk nr. i, x_i avstanden til midtpunktet av blokk i og p_i trykket ved x_i.

> **Merk:** Fra og med dette punkt i framstillingen skifter vi notasjon slik at indeks o for olje sløyfes. Vær imidlertid oppmerksom på at variabelen `PO` brukes til å betegne trykket i den utleverte `FORTRAN`-kode.

La A betegne grensen mellom blokk 1 og 2, og B grensen mellom 2 og 3. Da kan

vi tilnærme de deriverte ved de to grensene på følgende måte

$$\left.\frac{dp}{dx}\right|_A \approx \frac{p_1 - p_2}{x_1 - x_2} = -\frac{p_1 - p_2}{x_2 - x_1},$$

og

$$\left.\frac{dp}{dx}\right|_B \approx \frac{p_3 - p_2}{x_3 - x_2}.$$

La oss definere en del hjelpestørrelser før vi går videre i diskretiseringen av ligningen. Vi setter

$$\Delta p_2^+ = p_3 - p_2,$$

$$\Delta p_2^- = p_1 - p_2,$$

$$\Delta x_2^+ = x_3 - x_2 = \frac{1}{2}(\Delta x_2 + \Delta x_3),$$

$$\Delta x_2^- = x_2 - x_1 = \frac{1}{2}(\Delta x_1 + \Delta x_2) = \Delta x_1^+,$$

$$N_{x2}^+ = \frac{C\left[\frac{k_x b}{\mu}\right]_2^+}{\Delta x_2^+},$$

$$N_{x2}^- = \frac{C\left[\frac{k_x b}{\mu}\right]_2^-}{\Delta x_2^-}.$$

Notasjonen er her at for en vilkårlig variabel Y betegner $[Y]_2^+$ en verdi tatt ved grenseflaten mellom blokkene 2 og 3, eller en middelverdi mellom blokkene.

Størrelsen N_{x2}^+ representerer midlere strømningskoeffisient eller transmissibilitet mellom blokk 2 og 3. Volumfaktoren b og viskositeten μ kan avhenge av trykket $p(x,t)$, som altså er en funksjon av både avstand x og tid t. Når simuleringsprogrammet flytter seg et tidssteg Δt fra t til $t + \Delta t$, kan en vise at det oppnås god stabilitet og mest korrekte verdier for trykkløsningen dersom det brukes tidsmidlete verdier for b og μ i uttrykket for N_x, se avsnitt 3.4. Det vil si at en for volumfaktoren beregnet ved grenseflaten B bruker

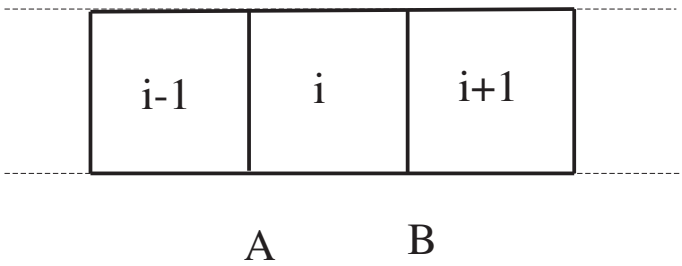
$$[b]_2^+ = \frac{1}{4}(b(p(x_2, t+\Delta t)) + b(p(x_3, t+\Delta t)) + b(p(x_2, t)) + b(p(x_3, t))).$$

Således blir den b-verdien som inngår i NX-ledd både tids- og avstandsmidlet.

> **Merk:** I avsnitt 3.4 blir omtalt hva som menes med implisitt formulering: Verdien av de ukjente trykkene velges ved slutten av tidssteget. For å få en konsistent, fullstendig implisitt formulering må b og μ i N_x også beregnes ved slutten av tidssteget. Dette har imidlertid liten praktisk betydning siden b og μ er svakt trykkavhengige for olje.

3.3 Diskretisering i tid og rom

Vi er nå klar til å diskretisere diffusivitetsligningen 2.5 slik at den kan løses numerisk. I Figur 3.3 har den generelle blokken (i) grenseflaten A mot naboen $(i-1)$ i negativ x-retning og grenseflaten B mot naboen $(i+1)$ i positiv x-retning.



Figur 3.3: Generell blokknummerering.

For å forenkle framstillingen setter vi hjelpestørrelsen Y lik

$$Y = \frac{Ck_x b}{\mu}\frac{\partial p}{\partial x},$$

og Ligning 2.5 skrives som

$$\frac{\partial Y}{\partial x} + \mathtt{A9} = \mathtt{A2}\frac{\partial p}{\partial t}. \qquad (3.1)$$

Vi bruker differansemetoden til å diskretisere Ligning 3.1, høyre siden i romlig x-retning, og venstre siden i tid, og får[1]

$$\frac{Y_i\big|_B - Y_i\big|_A}{\Delta x_i} + \mathtt{A9}_i = \mathtt{A2}_i\frac{p_i(t+\Delta t) - p_i(t)}{\Delta t}. \qquad (3.2)$$

Videre tilnærmes

$$\begin{aligned} Y_i\big|_B &= \left[\frac{Ck_x b}{\mu}\frac{\partial p}{\partial x}\right]_B, \\ &\approx \left[\frac{Ck_x b}{\mu}\right]_B \left[\frac{\partial p}{\partial x}\right]_B, \\ &\approx \left[\frac{Ck_x b}{\mu}\right]_i^+ \frac{\Delta p_i^+}{\Delta x_i^+}, \\ &\approx N_{xi}^+ \cdot \Delta p_i^+, \end{aligned}$$

[1]For enkelthets skyld bruker vi fortsatt standard likhetstegn mellom venstre og høyre siden, selv om disse nå er diskretiserte tilnærminger til differensialligningens høyre og venstre side.

og tilsvarende for det andre leddet,

$$Y_i\big|_A \approx -N_{xi}^- \cdot \Delta p_i^- .$$

Vi definerer nå

$$\begin{aligned} O_{xi}^+ &= \frac{N_{xi}^+}{\Delta x_i}, \\ O_{xi}^- &= \frac{N_{xi}^-}{\Delta x_i}, \end{aligned}$$

som innsatt i Ligning 3.2 gir diffusivitetsligningen på følgende numerisk form

$$\boxed{O_{xi}^+ \cdot \Delta p_i^+ + O_{xi}^- \cdot \Delta p_i^- + \mathtt{A9}_i = \mathtt{A2}_i(p_i(t+\Delta t) - p_i(t))/\Delta t.} \qquad \ldots\ldots (3.3)$$

Det er løsningen av denne simuleringsligningen som er programmert i den utdelte FORTRAN-koden. Programmet er laget med en initialiseringsdel som gir verdi til alle variable ved $t = 0$, og med en tidsstegssløyfe som beregner trykkene i de numeriske blokkene etter et fastsatt nytt tidssteg Δt, når alle størrelsene i ligningen, inklusive trykkene i hver blokk, er kjent på tidsnivå t.

3.4 Valg av numerisk formulering

Alle størrelser som inngår i Ligning 3.3 er kjente på tidsnivå t og simuleringsprogrammet oppdaterer dem til nivå $t + \Delta t$. På høyre side av ligningen inngår trykket på begge nivåene som følge av tilnærmingen av den tidsderiverte. Vi har imidlertid hittil ikke angitt på hvilket tidsnivå leddene $\Delta p_i^{\pm}$ skal evalueres. De tre mest vanlige valgene er angitt i Tabell 3.1.

Numerisk formulering	TidsnivÑ for $\Delta p_i^{\pm}$
eksplisitt	t
implisitt	$t + \Delta t$
Crank-Nicolson	$t + \Delta t/2$

Tabell 3.1: De tre vanligste numeriske formuleringer.

Før vi går videre trenger vi noen begreper fra numerisk analyse. De vil bare bli omtalt her og interesserte kan se nærmere i boken til Carnahan *et al* [3] eller andre standardverker i numeriske metoder for utfyllende kommentarer.

La oss anta at u er den korrekte, eventuelt analytiske, løsning til en differensialligning, og at v er den tilsvarende numeriske løsning. *Diskretiseringsfeilen* er gitt ved $w = u - v$. Dersom $w \to 0$ når $(\Delta x, \Delta t) \to 0$, sier vi at den numeriske formuleringen

konvergerer. Dersom differensialligningen i numerisk formulering går mot den opprinnelige differensialligning når $(\Delta x, \Delta t) \to 0$, sier vi at den numeriske formuleringen er *konsistent* med differensialligningen. Dersom den numeriske løsningen v er endelig under hele simuleringen, sier vi at løsningen er *stabil*, selv om den kan fluktuere eller oscillere mellom tidsstegene. Dersom løsningsrutinen som brukes er iterativ, se avsnitt 3.7, brukes også begrepet *konvergens* om den trinnvise tilnærming til en endelig verdi for v innen et gitt tidssteg.

For de tre formuleringene i Tabell 3.1 kan det vises at følgende egenskaper gjelder

eksplisitt er enklest å programmere; trenger ingen løsningsrutine; $w \sim \quad (\Delta t) + \quad (\Delta x^2)$; blir ustabil når $\Delta t/\Delta x^2 > 1/2$ og er derfor betinget stabil

implisitt er ubetinget stabil og blir derfor ofte foretrukket; brukes i simuleringsmodellen *ECLIPSE*; må ha en løsningsrutine; $w \sim \quad (\Delta t) + \quad (\Delta x^2)$

Crank-Nicolson er også ubetinget stabil, men krever noe mer programmeringsarbeid enn for implisitt formulering; har en tendens til å oscillere rundt den riktige løsning; $w \sim \quad (\Delta t^2) + \quad (\Delta x^2)$

Merk: Disse uttrykkene for diskretiseringsfeil forutsetter at Δt og Δx er dimensjonsløse, det vil si at diffusivitetsligningen er gjort dimensjonsløs før diskretiseringen.

3.5 Eksplisitt formulering

Nå velges $\Delta p_i^{\pm}$ i Ligning 3.3 ved tid t, og er derfor kjente tall. Den eneste ukjente i lingningen er $p_i(t+\Delta t)$ som vi kan finne direkte:

$$p_i(t+\Delta t) = \frac{\Delta t}{\mathtt{A2}_i}\left(O_{xi}^{+} \cdot \Delta p_i^{+}(t) + O_{xi}^{-} \cdot \Delta p_i^{-}(t) + \mathtt{A9}_i\right) + p_i(t). \qquad \ldots\ldots \quad (3.4)$$

Vi får altså ikke noe lineært ligningssystem med denne formuleringen. Nytt trykk i blokk nummer i finnes nesten uavhengig av trykkene i naboblokkene $i \pm 1$. Det er en svak kopling via midling til naboblokkene av den trykkavhengige koeffisienten b/μ som inngår i $N_x^{\pm}$, se avsnitt 3.2.

3.6 Implisitt formulering

Her velges $\Delta p_i^{\pm}$ i Ligning 3.3 ved tid $t + \Delta t$.

Merk: Vi forenkler nå notasjonen slik at trykk $p_i(t)$ er ved tid t, mens trykk $p_i(t+\Delta t)$ betegnes forenklet bare som p_i.

Vi bruker definisjonene på $\Delta p_i^{\pm}$ fra avsnitt 3.2, og setter inn i Ligning 3.3 og får

$$O_{xi}^{+}(p_{i+1}-p_i)+O_{xi}^{-}(p_{i-1}-p_i)+\mathtt{A9}_i=\frac{\mathtt{A2}_i}{\Delta t}\,(p_i-p_i(t)),$$

som rearrangert gir

$$O_{xi}^{-}\,p_{i-1}+(-O_{xi}^{-}-O_{xi}^{+}-\frac{\mathtt{A2}_i}{\Delta t})\,p_i+O_{xi}^{+}\,p_{i+1}=-\frac{\mathtt{A2}_i}{\Delta t}\,p_i(t)-\mathtt{A9}_i.$$

Vi definerer følgende hjelpestørrelser

$$\begin{aligned}
B_i &= O_{xi}^{-}+O_{xi}^{+}+\frac{\mathtt{A2}_i}{\Delta t},\\
A_i &= O_{xi}^{-}/B_i,\\
C_i &= O_{xi}^{+}/B_i,\\
D_i &= (\frac{\mathtt{A2}_i}{\Delta t}p_i(t)+\mathtt{A9}_i)/B_i,
\end{aligned}$$

og simuleringsligningen blir

$$\boxed{-A_ip_{i-1}+p_i-C_ip_{i+1}=D_i.} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.5)$$

Ligningen gir altså et lineært ligningssett med like mange ukjente som numeriske blokker, og hvor ligningen for en blokk avhenger av de ukjente i de to naboblokkene.

For å eksemplifisere velger vi et system med fire blokker, slik at $i \in \{1,2,3,4\}$. Da gir Ligning 3.5 følgende sett av ligninger:

$$\begin{aligned}
0+p_1-C_1p_2&=D_1,\\
-A_2p_1+p_2-C_2p_3&=D_2,\\
-A_3p_2+p_3-C_3p_4&=D_3,\\
-A_4p_3+p_4+0&=D_4,
\end{aligned} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.6)$$

som kan skrives som en matriseligning

$$\begin{pmatrix} 1 & -C_1 & 0 & 0\\ -A_2 & 1 & -C_2 & 0\\ 0 & -A_3 & 1 & -C_3\\ 0 & 0 & -A_4 & 1 \end{pmatrix}\cdot\begin{pmatrix} p_1\\ p_2\\ p_3\\ p_4 \end{pmatrix}=\begin{pmatrix} D_1\\ D_2\\ D_3\\ D_4 \end{pmatrix}, \qquad \ldots\ldots\ldots \quad (3.7)$$

eller

$$\overset{\Rightarrow}{K}\cdot\vec{p}^t=\vec{D}^t,$$

hvor koeffisientmatrisen $\overset{\Rightarrow}{K}$,

$$\overset{\Rightarrow}{K}=\begin{pmatrix} 1 & -C_1 & 0 & 0\\ -A_2 & 1 & -C_2 & 0\\ 0 & -A_3 & 1 & -C_3\\ 0 & 0 & -A_4 & 1 \end{pmatrix},$$

er en *tridiagonal* matrise, hvor kun hoveddiagonalen og de to subdiagonalene har ledd forskjellig fra 0.

3.7 Iterativ løsningsmetode

En ligning av typen 3.7 kan løses iterativt eller direkte, ved eliminasjon. Som vi skal demonstrere senere, i forbindelse med en øving, er direkte metoder å foretrekke for en-dimensjonale system. Det samme er tilfelle for to-dimensjonale problemer opp til en viss størrelse. I tre dimensjoner derimot blir det selv for 'små' systemer nødvendig å bruke iterative metoder. En moderne, industriell modell som *ECLIPSE*, for eksempel, har ikke direkte løsningsmetode innlagt.

La oss beskrive prinsippet for iterative metoder med et enkelt eksempel, to ligninger med to ukjente:

$$\begin{aligned} 1: &\quad x_1 + 2x_2 = 5, \\ 2: &\quad x_1 + 3x_2 = 7, \end{aligned}$$

eller

$$\begin{aligned} 1: &\quad x_1 = 5 - 2x_2, \\ 2: &\quad x_2 = (7 - x_1)/3. \end{aligned}$$

La k betegne iterasjonstelleren. Prosedyren for *Gauss-Jordan's* metode er: Start med antatte verdier på både x_1 og x_2. For iterasjonsrunde $k+1$, beregn samtidig $x_1^{k+1} = 5 - 2x_2^k$ og $x_2^{k+1} = (7 - x_1^k)/3$, og forsett inntil konvergens.

Prosedyren for *Gauss-Seidel's* metode er litt anderledes: Beregn først $x_1^{k+1} = 5 - 2x_2^k$ og deretter $x_2^{k+1} = (7 - x_1^{k+1})/3$. En bruker altså oppdaterte verdier straks de er generert. Denne metoden er lagt inn som en subrutine i det utleverte programmet og reprodusert nedenfor. Iterasjonssløyfen kalles `itera`, sløyfen for løsing av trykkene i hver blokk kalles `pressures`. Ligning 3.5 er brukt inne i denne sløyfen. Triggeren `isw` brukes til å sjekke for konvergens.

Gauss-Seidel subrutine i utlevert program

```
   subroutine gsit(mx,eps)
    use felles_matriser
    implicit none

    integer :: kkk, i, isw, im, ip, mx, kkkcum
    real(dp) :: eps

itera:     do kkk = 1,300
             kkkcum = kkk
             isw = 0
             im = 1

pressures:     do i = 1,mx
```

```
                if (i .gt. 1) im = i-1
                if (i .lt. mx) ip = i+1

                po(i) = a(i)*po(im)+c(i)*po(ip)+d(i)

                if (isw == 0) then
                    if (dabs(comp(i)-po(i)) > eps) isw = 1
                end if

                comp(i) = po(i)

            end do pressures

            if (isw .eq. 0) exit

        end do itera

        write (6,"('  kkkcum .........', i4/)") kkkcum

    return
end
```

3.8 Direkte løsningsmetode

La oss anskueliggjøre hvordan det lineære ligningssystemet 3.6 kan løses ved eliminasjon. Ligningene tenkes nummerert fra 1 til 4 fra toppen og ned. Framgangsmåten er å eliminere p_1 fra andre ligning ved å bruke første ligning; deretter p_2 fra tredje ligning ved å bruke den oppdaterte andre ligning —hvor p_1 er fjernet; og til sist p_3 fra fra fjerde ligning.

Sammenligner en med ligningen på formen 3.7, så har vi nå endret matriseelementer i $\vec{\vec{K}}$ slik at det kun er nuller under diagonalen. Lar vi $\otimes$ betegne et matriseelement som har fått endret verdi, har Ligning 3.7 nå følgende utseende:

$$\begin{pmatrix} 1 & -C_1 & 0 & 0 \\ 0 & \otimes & -C_2 & 0 \\ 0 & 0 & \otimes & -C_3 \\ 0 & 0 & 0 & \otimes \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} = \begin{pmatrix} D_1 \\ \otimes \\ \otimes \\ \otimes \end{pmatrix}.$$

Fjerde ligning, tilsvarende linje nummer 4 i $\vec{\vec{K}}$, innholder nå bare p_4 som ukjent. Denne bestemmes og settes inn i tredje ligning, som da bare har p_3 som ukjent, etc.

Prosedyren består altså i to sveip, først nedover for å skaffe nuller under diagonalen, så oppover igjen med suksessiv innsetting av funne trykk. Denne algoritmen er lagt inn som en subrutine i det utleverte program, og er reprodusert under. I line 04 betegner `al` diagonalleddet; videre er `s` den endrete høyreside av ligningene.

En kan merke seg at det kun opereres på ledd i de tre diagonalene i $\vec{\vec{K}}$, eller innenfor båndet til denne båndmatrisen. Løsningsrutinen `tridia` er derfor meget effektiv og trenger lite lagerplass.

Subrutine for løsing av lineært ligningssett med tridiagonal koeffisientmatrise

```
subroutine tridia(mx)
  use felles_matriser
        implicit none

        integer :: i, mx, k

        al(1) = 1
        s(1) = d(1)


! Lager øvre triangulær matrise med
! kun nuller under hoveddiagonalen.
upper_trian:    do i = 2,mx
                    al(i) =  1. - a(i)/al(i-1)*c(i-1)
                    s(i) = d(i) + a(i)/al(i-1)*s(i-1)
                end do upper_trian


! Løser for trykket i blokk mx
                po(mx) = s(mx)/al(mx)    !last pressure


! "Spaserer" oppover diagonalen igjen og løser for
! resten av trykkene.
back_subst:     do k = 2,mx
                    i = mx - k + 1
                    po(i) = (s(i)+c(i)*po(i+1)) / al(i)
                end do back_subst

      return
end subroutine tridia
```

3.9 Skisse til flytskjema for utlevert program

INITIALISERING
Les data
Skriv data
Beregn konstante størrelser
Initialiser variable
Beregn `OIP`
α **START TIDSSTEGSSLØYFE**
Oppdater trykk, `POLD = PO`
β **START YTRE ITERASJON**, på ulineære ledd, dvs. b, μ
`PEST = PO(1)`
`PO = (PO + POLD)/2`
Fluidegenskaper —subrutine `FLPROP`
Strømningskoeffisienter —subrutine `FLOCON`
Løs ligningssettet med en av subrutinene
`TRIDIA` –dersom direkte løsning
`GSIT` –dersom iterativ løsning

Skriv beregnet trykk dersom `IWRITE=1`
Beregn $= \|$`PO(1) - PEST`$\|$
β' **SLUTT** Dersom $\geq \varepsilon$: Gå til β ! ε er innlest toleransekriterium
Gjør materialbalanse
Skriv ut trykk og tidsstegsammendrag
α' **SLUTT** Dersom flere tidssteg: Gå til α
STOP

Bibliografi

[1] Aziz, K. and Settari, A.(1979). *Petroleum Reservoir Simulation*, Applied Science Publishers, London.

[2] Paul Papatzacos, *Matematisk Modellering*, kompendium i fag av samme navn, Avsnitt 2.2, Høgskolesenteret i Rogaland.

[3] Carnahan, B., Luther, H. A., and Wilkes J. O.: *Applied Numerical Methods*, John Wiley & Sons, New York (1969).

[4] Johnson, E.F., Bossler, D.P., and Naumann, V.O.: "Calculation of Relative Permeability from Displacement Experiments," *Trans.* AIME, **1959**, 371.

[5] Welge, H.J.: "A Simplified Method for Computing Oil Recovery by Gas or Water Drive," *Trans.* AIME, **1952**, 91.

[6] Kyte, J.R., and Rapoport, L.A.:"Linear Waterflood Behavior and End Effects in Water-Wet Porous Media,"*Trans.* AIME, **1958** 423.

[7] Dake, L.P.: *Fundamentals of Reservoir Engineering*, Elsevier, New York **1978**

Del II

Fortrankode for utlevert program

```
  program utdelt_sim
   use felles_matriser
   implicit none

!list of real variables in double precision:
!    kx      permeabilitet i x-retning, md
!    nt      oljevolum tilstede ved et gitt tidspunkt, stb
!    ni      oljevolum injisert, stb
!    np      oljevolum produsert, stb
!    npc     oljevolum produsert i %
!    visorg  opprinnelig viskositet, cp
!    dbdp    derivert av bo mhp trykk, stb/rb/psi
!    delt1   de 12 første tidssteg er delt1/(1.5**(12-l)), hvor
!            l er antall tidssteg, days
!    eps     trykktoleranse, iterasjon på ulineære ledd _og_ i
!            iterativ løsningsrutine, psi
!    dex     delta x, ft
!    dey     delta y, ft
!    dez     delta z, ft
!    fpb     cubic feet per barrel, 5.61
!    pi      3.14
!    vol     dex*dey*dez/fpb, rb
!    volphi  vol*phi, rb
!    ckbak   konstantleddet i strømningskoeffisientene
!    a2      phi*dbdp
!    ooip    original oil in place, stb
!    pest    estimated value of po(1), psi
!    qotot   total oil rate, stb/day
!    ctim    kumulativ simulert tid, days
!    eot     total oil error
!    borig   original value of bo, stb/rb
!    phi     porosity
!    porig   original oil pressure, psi
!    delt    time step size, days
!    convf   conversion factor from Darcy units to oil-field
!            units
!    en      1.d0
!    to      2.d0
!

    real(dp) :: kx,nt,ni,np,npc,visorg,dbdp,delt1,eps,dez,&
                dey,fpb,pi,vol,volphi,ckbak,a2,ooip,&
                pest,qotot,ctim,eot,borig,phi,dex,porig,&
                delt,convf,en,to
```

```
!list of integers
!    stmax   number of time steps
!    mx      number of numerical blocks
!    isol    specify solution method
!    ipe     trigger, not in use
!    iskip   trigger, not in use
!    iwrite  trigger, if equal to 1, intermediat pressures are
!            written
!    i       loop counter
!    im      i-1
!    ip      i+1
!    lcysw   number of iterations per time step on non-linear terms
!    kcy     sum of lcysw
!    l       time step counter
!    k       loop counter
!

    integer  :: stmax,mx,isol,ipe,iskip,iwrite,i,im,ip,&
                kcy,l,k,lcysw

        open (unit=5,file='res90dat.dat', status='old')
        open (unit=6,file='res90out.dat', status='replace')

!     ** read and write input data

        write (6,3000)
        read  (5,*)       mx, stmax, isol, ipe, iskip, iwrite

allocate (&
            a(mx),c(mx),d(mx),al(mx),s(mx),comp(mx),po(mx),&
            pold(mx),oxplus(mx),oxmin(mx),a9(mx),qo(mx),bo(mx),x(mx)&
         )

      write (6,3010)    mx, stmax, isol, ipe, iskip, iwrite
      read  (5,*)        kx, phi, dex, porig, borig
      read  (5,*)        visorg, dbdp, delt1, eps, dez
      read  (5,*)        dey
      write (6,3020)    kx, phi, dex, porig, borig, visorg, &
                        dbdp, delt1, eps, dez, dey
      read (5,*)        (qo(i),i = 1,mx)

      write (6,3030)    (i,qo(i), i = 1,mx)
      write (6,3080)
!
!     ** begin calculations; initialize variables
```

```
!
      fpb    =    5.6146d0
      pi    =    3.1415926d0
      convf    =    0.00632827

      en    =    1.d0
      to    =    2.d0

      oxplus(mx)    = 0
      oxmin(1)        = 0
      delt          = 0
      ni            = 0
      np            = 0
      ctim            = 0
      kcy            = 0

      vol    =    dez*dey*dex/fpb
      volphi=    vol*phi
      ckbak =    convf*kx/dex/visorg/dex
      a2    =    phi*dbdp

      ooip    =    0.0
      x(1)    =    dex/to
pressure_points:&
    do i    = 2,mx
        im = i -1
        x(i) = dex + x(im)
    end do pressure_points

      write (6,3040) (i,x(i),i=1,mx)
      write (6,3080)

      po    = porig
      qo    = qo /vol
      a9    = qo
      pold    = porig

ooip_in_place:&
    do i = 1,mx
        ooip = ooip + volphi * borig
    end do ooip_in_place

time_step_loop:&
    do l = 1, stmax
```

```
        pold = po

        lcysw    =    0
        if (l <= 12) delt = delt1/1.5**(12-l)

        write (6,3070)
!
!   cc start iteration loop on unlinear variables i.e. bo
!

7744    continue

        pest = po(1)
        kcy = kcy + 1

        po = (po + pold)/to

        call flprop(mx,porig,dbdp,borig)
        call flocon(mx,delt,a2,ckbak)


        select case(isol)

            case(1)
                call tridia(mx)

            case(2)
                po = to*po - pold
                comp = po
                call gsit(mx,eps)
        end select

        if (iwrite == 1) then
            write(6,3050) (i,po(i), i = 1,mx)
            write(6,3080)
        end if

        lcysw = lcysw + 1

check_convergence_of_nonlinear_term:&
        if (&
                dabs(po(1)-pest) > eps &
            .and. &
                lcysw < 4) then
```

```
                    goto 7744

        end if check_convergence_of_nonlinear_term

!
!       start material balance
!
        ctim = ctim + delt

        nt = 0
        qotot = 0.

        call flprop(mx,porig,dbdp,borig)


matbal:&
        do i = 1,mx
           nt = nt + volphi * bo(i)
           if (qo(i) .gt. 0.) ni = ni+qo(i)*vol*delt
           if (qo(i) .lt. 0.) np = np-qo(i)*vol*delt
           if (qo(i) .lt. 0.) qotot = qotot-qo(i)*vol
        end do matbal

        eot = (-ni+np+nt-ooip)/(ooip+ni)*100
        npc = np/ooip*100

        if (iwrite /=  1) then
           write (6,3050) (i,po(i),i=1,mx)
        end if

        write (6,3060)l,ctim,np,delt,kcy,ni,nt,qotot,eot,npc

    end do time_step_loop

!
!     ** close files
!
      close (5)
      close (6)
!
!     ** format statements
!
 3000 format (&
     '     ***** one dimensional, one phase simulator *****    '/&
     '           ------------------------          '//&
```

```
     '     basic assumptions of this model include:            '//&
     '      - horizontal flow of a compressible fluid          '/&
     '      - pressure dependent volume factor                 '/&
     '      - pressure independent viscosity                   '/&
     '      - homogeneous rock properties, that is the same    '/&
     '        porosity and permeability in all blocks.         '/&
     '      - flowing fluid is undersaturated oil              '//)
!
 3010 format (&
     '  number of blocks in x-direction, mx .......... :', i3  /&
     '  maximum number of time steps, stmax .......... :', i3  /&
     '  solution method used, isol ................... :', i3  /&
     '  ipe ........................................... :', i3  /&
     '  iskip ......................................... :', i3  /&
     '  write option, iwrite ......................... :', i3  //)
!
 3020 format (&
     '  horizontal permeability, kx, md  ............ :', f12.4/&
     '  porosity, phi, fraction ...................... :', f8.4
/&
     '  delta length (x-dir), dex, ft ................ :', f10.2/&
     '  original oil pressure, porig, psia ........... :', f10.1/&
     '  original oil fvf, borig, std vol / res vol ... :', f10.4/&
     '  original oil viscosity, visorg, cp ........... :', f10.4/&
     '  oil fvf pressure derivative, dbdp,              ',      /&
     '    std vol/res vol/psi ........................ :', f10.8/&
     '  time increment, delt1, days .................. :', f10.6/&
     '  pressure tolerence, eps ...................... :', f8.5
/&
     '  block width, dez, ft ......................... :', f8.1
/&
     '  block thickness, dey, ft ..................... :', f8.1
//)
!
 3030 format (&
     '  block no. :', i3, '     oil rate, qo, stb/d .... ', f10.1)
!
 3040 format (&
     '  block no. :', i3, '     pressure point, x(i), ft ', f8.1)
!
 3050 format (&
     '  block no. :', i3, '     oil pressure, psia ..... ', f12.3)
!
 3060 format (2x,'steps=',i4,t22,'time=',d12.4,t43,'np=',d12.4/2x,&
    'delt=',d12.4,t22,'cum cycle=',i4,t43,'ni=',d12.4/2x,&
```

```
    'nt=',d12.4,t22,'qotot=',d12.4,t43,'cum oil error=',d12.4/2x,&
    'npc=',d12.4)
!
 3070 format (25('*****')/)
!
 3080 format (/)
!      stop
      end
!
!     ****************************************************************
!     *                   end of main program
!     ****************************************************************
!
!
!     ****************************************************************
!     *                   subroutine tridia
!     ****************************************************************
!
subroutine tridia(mx)
  use felles_matriser
        implicit none

        integer :: i, mx, k

        al(1) = 1
        s(1) = d(1)

upper_trian:    do i = 2,mx
                    al(i) =  1.   - a(i)/al(i-1)*c(i-1)
                    s(i) = d(i) + a(i)/al(i-1)*s(i-1)
                end do upper_trian

                po(mx) = s(mx)/al(mx)    !last pressure

back_subst:     do k = 2,mx
                    i = mx - k + 1
                    po(i) = (s(i)+c(i)*po(i+1)) / al(i)
                end do back_subst

      return
end subroutine tridia
!
!     ****************************************************************
!     *                 end of subroutine tridia
!     ****************************************************************
```

```
!
!
!     ****************************************************************
!     *                      subroutine gsit
!     ****************************************************************
!
subroutine gsit(mx,eps)
    use felles_matriser
    implicit none

    integer :: kkk, i, isw, im, ip, mx, kkkcum
    real(dp) :: eps

itera:    do kkk = 1,300
            kkkcum = kkk
            isw = 0
            im = 1

pressures:    do i = 1,mx
                if (i .gt. 1) im = i-1
                if (i .lt. mx) ip = i+1

                po(i) = a(i)*po(im)+c(i)*po(ip)+d(i)

                if (isw == 0) then
                    if (dabs(comp(i)-po(i)) > eps) isw = 1
                end if

                comp(i) = po(i)

            end do pressures

            if (isw .eq. 0) exit

        end do itera

        write (6,"('  kkkcum .........', i4/)") kkkcum

    return
end
!
!     ****************************************************************
!     *                      end subroutine gsit
!     ****************************************************************
!
```

```
!
!     ****************************************************************
!     *                      subroutine flprop
!     ****************************************************************
!
subroutine flprop(mx,porig,dbdp,borig)
    use felles_matriser
    implicit none
        integer :: mx, i
        real(dp) :: porig,dbdp,borig

        bo = borig + (po - porig)*dbdp

    return
end subroutine flprop
!
!     ****************************************************************
!     *                      end subroutine flprop
!     ****************************************************************
!
!     ****************************************************************
!     *                      subroutine flocon
!     ****************************************************************
!
subroutine flocon(mx,delt,a2,ckbak)
    use felles_matriser
    implicit none
        integer :: mx,i,im
        real(dp) :: delt,a2,ckbak,b
oxp:    do i = 2, mx
            im = i-1
            oxmin(i) = ckbak*(bo(i)+bo(im))/2
            oxplus(im) = oxmin(i)
        end do oxp
acd:    do i = 1, mx
            b    = oxmin(i)+oxplus(i)+a2/delt
            a(i) = oxmin(i)/b
            c(i) = oxplus(i)/b
            d(i) =(a2/delt*pold(i)+a9(i))/b
        end do acd
return
end subroutine flocon
!
!     ****************************************************************
!     *                      end subroutine flocon
```

```
!     ****************************************************************
!
```

3.10 Module

```
Module felles_matriser
    use nrtype
!    integer, parameter :: dp = kind(1.d0)

  real(dp), dimension(:), allocatable ::&
       a,c,d,al,s,comp,po,&              !common/blk1/
       pold,oxplus,oxmin,a9,&            !common/blk2/
       qo,bo,x                           !dimensioned matrices
end module felles_matriser
```

3.11 Module

```
Module nrtype
integer, parameter :: dp = kind(1.d0)
integer, parameter :: sp = kind(1.0)
end module nrtype
```

3.12 Datafil

```
6,4,1,1,1,0                          MX,STMAX,ISOL,IPE,ISKIP,IWRITE
100,.05,10,2000,.7889                 KX,PHI,DEX,PORIG,BORIG
1.358,8.03D-06,.002,0.001,1               VISORG,DBDP,DELT1,EPS,DEZ
10                                    DEY
-1,4*0,1                            OIL FLOW RATES (MX VALUES)
```

3.13 Resultatfil

```
***** one dimensional, one phase simulator *****
      ------------------------

basic assumptions of this model include:

 - horizontal flow of a compressible fluid
 - pressure dependent volume factor
 - pressure independent viscosity
 - homogeneous rock properties, that is the same
   porosity and permeability in all blocks.
 - flowing fluid is undersaturated oil


number of blocks in x-direction, mx .......... :  6
maximum number of time steps, stmax .......... :  4
solution method used, isol ................... :  1
ipe .......................................... :  1
iskip ........................................ :  1
write option, iwrite ......................... :  0


horizontal permeability, kx, md  ............ :    100.0000
porosity, phi, fraction ...................... :  0.0500
delta length (x-dir), dex, ft ................ :     10.00
original oil pressure, porig, psia ........... :    2000.0
original oil fvf, borig, std vol / res vol ... :    0.7889
original oil viscosity, visorg, cp ........... :    1.3580
oil fvf pressure derivative, dbdp,
  std vol/res vol/psi ........................ :0.00000803
time increment, delt1, days .................. :  0.002000
pressure tolerence, eps ...................... : 0.00100
block width, dez, ft ......................... :     1.0
block thickness, dey, ft ..................... :    10.0


block no. :  1     oil rate, qo, stb/d ....       -1.0
block no. :  2     oil rate, qo, stb/d ....        0.0
block no. :  3     oil rate, qo, stb/d ....        0.0
block no. :  4     oil rate, qo, stb/d ....        0.0
block no. :  5     oil rate, qo, stb/d ....        0.0
block no. :  6     oil rate, qo, stb/d ....        1.0
```

```
  block no. :  1     pressure point, x(i), ft      5.0
  block no. :  2     pressure point, x(i), ft     15.0
  block no. :  3     pressure point, x(i), ft     25.0
  block no. :  4     pressure point, x(i), ft     35.0
  block no. :  5     pressure point, x(i), ft     45.0
  block no. :  6     pressure point, x(i), ft     55.0


**************************************************

  block no. :  1     oil pressure, psia .....     1997.259
  block no. :  2     oil pressure, psia .....     1999.584
  block no. :  3     oil pressure, psia .....     1999.946
  block no. :  4     oil pressure, psia .....     2000.054
  block no. :  5     oil pressure, psia .....     2000.416
  block no. :  6     oil pressure, psia .....     2002.741
  steps=   1         time=  0.2312D-04    np=  0.2312D-04
  delt=  0.2312D-04  cum cycle=   2       ni=  0.2312D-04
  nt=  0.4215D+01    qotot=  0.1000D+01   cum oil error=  0.0000D+00
  npc=  0.5485D-03
**************************************************

  block no. :  1     oil pressure, psia .....     1993.878
  block no. :  2     oil pressure, psia .....     1998.504
  block no. :  3     oil pressure, psia .....     1999.729
  block no. :  4     oil pressure, psia .....     2000.271
  block no. :  5     oil pressure, psia .....     2001.496
  block no. :  6     oil pressure, psia .....     2006.122
  steps=   2         time=  0.5781D-04    np=  0.5781D-04
  delt=  0.3468D-04  cum cycle=   4       ni=  0.5781D-04
  nt=  0.4215D+01    qotot=  0.1000D+01   cum oil error=  0.2107D-
13
  npc=  0.1371D-02
**************************************************

  block no. :  1     oil pressure, psia .....     1989.815
  block no. :  2     oil pressure, psia .....     1996.558
  block no. :  3     oil pressure, psia .....     1999.213
  block no. :  4     oil pressure, psia .....     2000.787
  block no. :  5     oil pressure, psia .....     2003.443
  block no. :  6     oil pressure, psia .....     2010.185
  steps=   3         time=  0.1098D-03    np=  0.1098D-03
  delt=  0.5202D-04  cum cycle=   6       ni=  0.1098D-03
  nt=  0.4215D+01    qotot=  0.1000D+01   cum oil error=  0.2107D-
13
```

```
  npc=  0.2606D-02
**************************************************

  block no. :  1     oil pressure, psia .....     1985.067
  block no. :  2     oil pressure, psia .....     1993.695
  block no. :  3     oil pressure, psia .....     1998.317
  block no. :  4     oil pressure, psia .....     2001.684
  block no. :  5     oil pressure, psia .....     2006.305
  block no. :  6     oil pressure, psia .....     2014.933
  steps=   4         time=  0.1879D-03    np=  0.1879D-03
  delt=  0.7804D-04  cum cycle=   8       ni=  0.1879D-03
  nt=  0.4215D+01    qotot=  0.1000D+01   cum oil error=  0.2107D-
13
  npc=  0.4457D-02
```

Del III

Øvingsoppgaver

Øving 1

Kjøring av utlevert program

1.1 Utprøving av løsningsrutiner

Sett antall blokker, `mx`, i datafilen til 6. Utfør deretter følgende 6 simuleringer:

EPS = 0.001	EPS = 1	EPS = 10
TRIDIA, ISOL = 1	TRIDIA, ISOL = 1	TRIDIA, ISOL = 1
GSIT, ISOL = 2	GSIT, ISOL = 2	GSIT, ISOL = 2

Noter følgende størrelser etter 4 tidssteg:

1. Trykkløsningen i blokk 1,
2. Materialbalansen,
3. Antall ytre iterasjoner (`CUM CYCLE`),
4. Antall indre iterasjoner (`KKKCUM`) når `ISOL` = 2,
5. Plott trykkprofilen, det vil si trykkene som funksjon av posisjonen x for tidsstegene nr. 1, 5 og 10 i samme graf.

Kommenter resultatet og angi årsaker til variasjonene.

Øving 2

Automatisk tidsstegsregulering

2.1 Først et par justeringer

1. Før tidsstegsløyfen starter skal *OIP* og *initial oil pressure distribution* skrives ut.
2. Innfor konvergenssjekk på alle trykkene i ytre iterasjonssløyfe. Nå sjekkes bare `po(1)` mot `pest`.

2.2 Ny tidsstegsregulering

Les inn og skriv ut

`delmin:` minimum tidsstegslengde

`delmax:` maksimum tidsstegslengde

`dpmax:` maksimum trykkendring pr. tidssteg

`dtmult:` multiplikator på tidsstegslengden

Sett `dpmx = 0.d0` for start av tidsstegsløyfe.
Etter beregning av trykk:

```
      dpmx = dabs(po(1) - pold(1))
      ii = 1
!
!     største trykkendring og tilhørende blokk
!
      do i = 2, mx
         dpp = dabs(pold(i) - po(i))
         if (dpp.gt.dpmx) then
```

```
            dpmx = dpp
            ii = i
         endif
      end do
!
!     sjekk om største verdi er over maks. tillatte.
!     tidsstegslengden reduseres og
!     melding skrives og iterasjoner i ytre
!     sløyfe startes på ny.
!
      if(dpmx.gt.dpmax) then
         delt = delt * 0.8 * dpmax/dpmx
         write(6,yy) dpmx, ii
         lcysw = 0
         goto 7744
      endif
yy    format -- time step reduction caused by pressure change,
      dpmx, in block, ii --
```

Erstatt tidsstegsberegningen med:

```
delt = delt * dtmult
delp = dpmx * dtmult
if (delp.gt.dpmax) then
   delt = delt * 0.8 * dpmax/delp
endif
if (delt.lt.delmin) delt = delmin
if (delt.gt.delmax) delt = delmax
```

For tidsstegsløyfen må en sette `delt = delmin`.

2.3 Kjør programmet for testing

Reproduser resultatet i øving 1.

Bruk

```
delmin = 0.002/1.5 * * 12 = 0.000015415
delmax = 1.0
dpmax = 100.
dtmult = 1.5
```

Automatisk tidsstegsregulering

Sjekk tidsstegsreguleringen ved å gjøre en ny kjøring med `dpmax = 0.5.`

Hvordan endres tidsutviklingen (cumulativ tid) og hvordan endres programkjøringen mht. konvergens?

Merk: Det må brukes `mx = 6, dex = 10, qo(1) = -1, qo(2)..qo(5) = 0, qo(6) = 1` for at tidsstegsreguleringen inntre som planlagt.

2.4 Praktiske råd

> **NB:** Behold alltid en kopi av den siste koden som fungerer, og etter endring av kode, test først med et tidligere eksempel.

Øving 3

Sammenligning med analytisk løsning

I denne øvingen skal en sammenligne den numeriske løsning med en analytisk løsning. Den analytiske løsningen får en også bruk for i øving 4.

3.1 Problemstilling

Gitt simuleringsligningen

$$\frac{\partial}{\partial x}\left(\frac{Ckb}{\mu}\frac{\partial P}{\partial x}\right) = \phi\frac{\partial b}{\partial P}\frac{\partial P}{\partial t}, \qquad (3.1)$$

hvor $\mu, C, k, \phi, \partial b/\partial P$ er konstanter. Anta at b i strømningsleddet med god tilnærmelse kan betraktes som konstant, kfr. oppgave 1 og 2 i eksamenssettet fra desember 1995. Da får en

$$\frac{\partial^2 P}{\partial x^2} = \frac{\phi\mu}{Ckb}\frac{db}{dP}\frac{\partial P}{\partial t}. \qquad (3.2)$$

Definer

$$\frac{1}{c_1^2} = \frac{\phi\mu}{Ckb}\frac{db}{dP},$$

og med $C = 0.00632827$ har en samme enheter som i simuleringsprogrammet. Ligningen blir da

$$\frac{\partial^2 P}{\partial x^2} = \frac{1}{c_1^2}\frac{\partial P}{\partial t}. \qquad (3.3)$$

Gitt følgende problem som svarer til det i øving 2:
I et en-dimensjonalt reservoar med lengde $2L$ og konstant tverrsnitt A, startes det ved tiden $t = 0$ samtidig injeksjon og produksjon i hver sin ende. Injeksjon og produksjon har like stor og konstant rate Q.
a) Utled den stasjonære løsningen p_s, av Ligning 3.3. Bruk Darcy's lov som grensevilkår.
Hint:

- Sett $\dfrac{\partial P}{\partial t} = 0$.
- Generell løsning: $p_s(x) = a_1 x + a_2$
- a_1 bestemmes fra Darcy's lov
- a_2 bestemmes ved at $p_s(L) = P_i$ hele tiden på grunn av antisymmetri om dette midtpunktet, P_i er initielt trykk.

Svar:

$$p_s(x) = P_i - q(x - L),$$

$$q = \frac{Q\mu}{0.0011271kAb}, \; q \text{ er et negativt tall.}$$

b) Innfør $p(x,t) = P(x,t) - p_s(x)$ og vis at ligning 3.3 blir

$$\frac{\partial^2 p}{\partial x^2} = \frac{1}{c_1^2}\frac{\partial p}{\partial t}. \qquad \ldots \qquad (3.4)$$

c) Vis at grensevilkårene nå er

i) $\dfrac{\partial p(0,t)}{\partial x} = 0$,

ii) $p(L,t) = 0$,

ii) $p(x,0) = q(x - L)$.

d) Løs nå ligning 3.4 ved å bruke metoden med separasjon av variable ved å gjennomføre følgende trinn:

- Sett $p(x,t) = f(x)g(t)$
- Vis at generell løsning blir

$$p(x,t) = (a\cos(\lambda x) + b\sin(\lambda x))\exp(-\lambda^2 c_1^2 t),$$

  hvor a, b og λ er konstanter.

- Vis at grensevilkår i) og ii) gir løsningen

$$p_n = a_n \cos(\frac{(2n+1)\pi}{2L}x)\exp(-(\frac{(2n+1)\pi}{2L})^2 c_1^2 t).$$

  Denne løsningen tilfredstiller ikke grensevilkår iii).

- Prøv å tilfredstille grensevilkår iii) ved å sette

$$p(x,t) = \sum_{n=0}^{\infty} p_n,$$

og vis at kravet iii) medfører:

$$a_n = -\frac{8Lq}{((2n+1)\pi)^2}.$$

- Vis til slutt at den fullstendige løsningen da blir:

$$P(x,t) = p(x,t) + p_s(x,t),$$

$$P(x,t) = P_i - q(x-L) - 8Lq \sum_{n=0}^{\infty} \frac{\cos(\frac{(2n+1)\pi}{2L}x)}{((2n+1)\pi)^2} \exp(-(\frac{(2n+1)\pi}{2L})^2 c_1^2 t). \qquad (3.5)$$

e) Lag et FORTRAN-program som regner ut trykket $P(x,t)$ for et vilkårlig trykkpunkt x og tid t. Bruk ellers data som i øving 2, med $2L = 60$ ft, $Q = -1$STB/D etc.

f) Sammenlign trykket i blokk 1 etter 4 tidssteg fra øving 2, med trykket $P(5\text{ft}, t_0)$

fra Ligning 3.5, med $t_0 = 0.1879 \cdot 10^{-3}$ dager.

g) Vis at numerisk løsning, trykket i blokk 1, nærmer seg analytisk løsning dersom

i) Antall blokker økes til 20 beregn $P(1.5\text{ft}, 0.1879 \cdot 10^{-3})$

ii) Antall tidssteg økes fram til omtrent samme tidspunkt ved å redusere `DELMIN` og `DTMULT`. Når `DELMIN` settes lik 0.1×10^{-4} og `DTMULT` settes lik 1.2 er det tilstrekkelig å inkludere 10 tidssteg (dvs. `STMAX=10`).

3.2 Analytisk løsning

a) Stasjonær løsning:

$$\lim_{t\to\infty} P(x,t) = p_s(x), \quad \frac{\partial P}{\partial t} = 0.$$

$$\frac{\partial^2 p_s}{\partial x^2} = 0,$$

$$p_s = a_1 x + a_2.$$

Fra Darcy's lov:

$$\frac{\partial p_s(0)}{\partial x} = -\frac{Q\mu}{0.0011271kAb} = -q = a_1$$

Dette gir

$$p_s(x) = -qx + a_2$$

$$p_s(L) = P_i \quad \Rightarrow \quad a_2 = P_i + qL$$

$$\underline{p_s(x) = P_i - q(x-L).}$$

b) Ligningen er

$$\frac{\partial^2 P}{\partial x^2} = \frac{1}{c_1^2}\frac{\partial P}{\partial t}$$

Sett

$$p = P - p_s(x)$$

$$\frac{\partial^2}{\partial x^2}(p + p_s(x)) = \frac{1}{c_1^2}\frac{\partial}{\partial t}(p + p_s(x))$$

Siden

$$\frac{\partial^2 p_s}{\partial x^2} = 0$$

fra a) og

$$\frac{\partial p_s(x)}{\partial t} = 0$$

får en direkte

$$\underline{\frac{\partial^2 p}{\partial x^2} = \frac{1}{c_1^2}\frac{\partial p}{\partial t}} \qquad \text{(3.6)}$$

c)

i)

$$\frac{\partial p(0,t)}{\partial x} = \frac{\partial P(0,t)}{\partial x} + q = -q + q = 0,$$

siden Darcy's lov gjelder ved utløpet $x = 0$ for alle t og

$$\frac{\partial P(0,t)}{\partial x} = -q.$$

ii)

$$p(L,t) = P_i - P_i + q(L-L) = 0.$$

iii)

$$p(x,0) = P_i - P_i + q(x-L) = q(x-L).$$

d) I ligning 3.6 prøver vi med $p(x,t) = f(x)g(t)$ og får

$$f''(x) \cdot g(t) = \frac{1}{c_1^2} f(x)\dot{g}(t)$$

$$\frac{f''(x)}{f(x)} = \frac{1}{c_1^2}\frac{\dot{g}(t)}{g(t)} = -\lambda^2$$

hvor λ er en konstant uavhengig av x og t. f'' og $\dot{g}$ er hhv. den andre deriverte av f mhp. x og den tidsderiverte av g. Dermed har en først

$$f''(x) + \lambda^2 f(x) = 0$$

Generell løsning av denne ligningen er

$$f(x) = a\cos\lambda x + b\sin\lambda x$$

Tilsvarende får en for den tidsavhengige funksjonen:

$$\dot{g}(t) + \lambda^2 c_1^2 g(t) = 0$$

Med generell løsning

$$g(t) = \exp(-\lambda^2 c_1^2 t)$$

Setter en sammen f og g får en den generelle trykkløsningen

$$\underline{p(x,t) = (a\cos\lambda x + b\sin\lambda x)\exp(-\lambda^2 c_1^2 t)}$$

På denne generelle løsning må en nå anvende grensevilkårene i), ii) og iii) for å bestemme konstantene a, b, λ.

<u>Grensevilkår i)</u>

$$\frac{\partial p(x,t)}{\partial x} = (-a\lambda\sin\lambda x + b\lambda\cos\lambda x)\exp(-\lambda^2 c_1^2 t)$$

$$\frac{\partial p(0,t)}{\partial x} = b\lambda \exp(-\lambda^2 c_1^2 t) = 0$$

som gir $\underline{b = 0.}$

<u>Grensevilkår ii)</u>

$$p(L,t) = a\cos\lambda L \exp(-\lambda^2 c_1^2 t) = 0$$

$$\lambda L = \frac{\pi}{2} + n\pi, \quad n = 0,1,2,\ldots$$

$$\underline{\lambda = \frac{(2n+1)\pi}{2L}}$$

<u>Grensevilkår iii)</u>

$$p(x,0) = a\cos\frac{(2n+1)\pi}{2L}x = q(x-L),$$

som er initialbetingelsene er ikke oppfylt for noen konstant a. Vi prøver derfor med

$$p(x,t) = \sum_{n=0}^{\infty} p_n(x,t),$$

$$p_n(x,t) = a_n\cos(\frac{(2n+1)\pi}{2L}x)\exp(-(\frac{(2n+1)\pi}{2L})^2 c_1^2 t).$$

Siden hver p_n tilfredstiller ligningen og de homogene grensevilkårene i) og ii) vil også p gjore det. Fra grensevilkår iii) får vi

$$\sum_{n=0}^{\infty} a_n\cos(\frac{(2n+1)\pi}{2L}x) = q(x-L) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.7)$$

Vi multipliserer nå denne ligningen med

$$\cos(\frac{(2k+1)\pi}{2L}x)$$

og integrerer over x fra 0 til $2L$ ved først å innføre nye variable

$$u = \frac{(2k+1)\pi}{2L}x, \quad dx = \frac{2L}{(2k+1)\pi}du$$

$$v = \frac{(2n+1)\pi}{2L}x.$$

Ligning 3.7 blir da

$$\sum_{n=0}^{\infty} a_n \int_0^{(2k+1)\pi} \cos v \cos u \, du = q\int_0^{(2k+1)\pi} (\frac{2L}{(2k+1)\pi}u - L)\cos u \, du \qquad . \quad (3.8)$$

Vi ser først på venstre side av ligningen:

$$\begin{aligned}\int_0^{(2k+1)\pi} \cos v \cos u \, du &= \frac{1}{2}\int_0^{(2k+1)\pi} (\cos(u-v)+\cos(u+v))du \\ &= \begin{cases} \dfrac{(2k+1)\pi}{2}, & \text{for } u=v \text{ eller } k=n \\ 0, & \text{for } u \neq v \end{cases}\end{aligned}$$

og dermed blir venstre siden

$$\sum_{n=0}^{\infty} a_n \frac{2k+1}{2}\pi \cdot \delta_{n,k} = a_k \frac{(2k+1)\pi}{2}$$

Høyre side av ligning 3.8 blir

$$q\int_0^{(2k+1)\pi} \frac{2L}{(2k+1)\pi} u \cos u \, du - q \underbrace{\int_0^{(2k+1)\pi} L \cos u \, du}_{=0}$$

$$= \frac{2Lq}{(2k+1)\pi}\Big[\cos u + u \sin u\Big]_0^{(2k+1)\pi} = -\frac{4Lq}{(2k+1)\pi}$$

Setter en så venstre side lik høyre side får en:

$$a_k \frac{(2k+1)\pi}{2} = -\frac{4Lq}{(2k+1)\pi},$$

$$a_k = -\frac{8Lq}{[(2k+1)\pi]^2}$$

Oppsummert gir dette

$$P(x,t) = p_s(x) + p(x,t) = p_s(x) + \sum_{k=0}^{\infty} p_k$$

$$= P_i - q(x-L) - 8Lq \sum_{k=0}^{\infty} \frac{\cos(\frac{(2k+1)\pi}{2L}x)}{((2k+1)\pi)^2} \exp(-(\frac{(2k+1)\pi}{2L})^2 c_1^2 t).$$

Øving 4

Numeriske formuleringer og `SOR`

4.1 Problemstilling

a) Modifisering

Modifiser programmet til å omfatte følgende opsjoner:

1. Implisitt, eksplisitt og Crank-Nicolson formulering

   Les inn trigger `ICRN:` $\begin{cases} 1 & \text{Implisitt} \\ 2 & \text{C-N} \\ 3 & \text{Eksplisitt} \end{cases}$

2. Løs ligningssettet med `SOR, GSIT, TRIDIA`

   Trigger `ISOL:` $\begin{cases} 1 & \texttt{TRIDIA} \\ 2 & \texttt{SOR} \ (= \texttt{GSIT} \text{ dersom } \omega = 1.0) \\ 3 & \text{Ingen losningsrutine, eksplisitt formulering } (\texttt{ICRN} = 3) \end{cases}$

b) Datakjøring

Bruk data som i Øving 2 med folgende unntak: `STMAX` = 12, `EPS` = 0.001, og sammenlign transienttrykkene i blokk 1, dvs. plott `PO(1)` mot $\ln t$ fra følgende kjoringer:

1. Implisitt, Tridia
2. C-N, Tridia
3. Eksplisitt
4. Analytisk løsning `PO`(5 ft, t)

c) Optimaliser ω

Bruk implisitt formulering og optimaliser ω i løsningsrutinen `SOR`, dvs. bestem den ω som gir det minste totale antall iterasjoner i `SOR` etter 12 tidssteg.

Merk: Øving 4 er obligatorisk og folgende skal leveres til godkjenning:

Programkode
Utskrift av resultatene fra kjøringene under b).
Et plott med de fire kurvene fra b).
Et plott av $\sum$ `KKKCUM` mot ω som viser optimal ω.

4.2 Kommentarer

La k angi iterasjonsnivå. Da har vi for `GSIT`:

$$P_i^{k+1} = A_i P_{i-1}^{k+1} + C_i P_{i+1}^k + D_i,$$

og for `SOR`:

$$P_i^{k+1} = P_i^k + \omega \left[A_i P_{i-1}^{k+1} + C_i P_{i+1}^k + D_i - P_i^k \right]$$

`SOR` betyr Suksessiv OverRelaksasjon,

og ω kalles relaksasjonsfaktor: $\begin{cases} \omega \in [1,2]: & \text{Overrelaksasjon} \\ \omega \in [0,1]: & \text{Underrelaksasjon} \\ \omega = 1.0: & \texttt{SOR = GSIT} \end{cases}$

I koden skiftes `GSIT` ut med `SOR`:

```
SUBROUTINE SOR(............., OM)
     .
     .
     .    som for GSIT, men skift ligning til Ę inkludere ω
     .
     .
     RETURN
     END
```

`OM` leses inn fra datafil og skrives ut i hovedprogrammet.
I hovedprogrammet, etter `CALL FLOCON`:

```
select case(isol)
case(1)
    CALL TRIDIA
case(2)
         po = to*po - pold
         comp = po
    CALL SOR(...... , OM)
case(3)
    IM = 1
    DO I = 1, MX                    For eksplisitt for-
         IF (- - -) IM =            mulering. Beregn-
         IF (- - -) IP =            er trykket i hver
                                    blokk uten løs-
                                    ningsrutine.
              PO = A · POLD + C · POLD + D
    end do
end select
```

Fra for har en for en generell blokk i:

$$OX_i^+ \Delta P_i^+ + OX_i^- \Delta P_i^- + A9_i = a2(P_i(t+\Delta t) - P_i(t))/\Delta t.$$

Dette ligningssettet onsker en pĘ formen

$$-A_i P_{i-1} + P_i - C_i P_{i+1} = D_i.$$

Eksplisitt formulering

$$OX_i^+ (P_{ip} - P_i) + OX_i^- (P_{im} - P_i) + A9_i = \frac{a2}{\Delta t}(P_i(t+\Delta t) - P_i),$$

hvor alle trykkene, bortsett fra $P_i(t+\Delta t)$ er ved "gammel" tid.
Ved Ę bruke $B = \Delta t / a2$
$A_i = B * OX_i^-$
$C_i = B * OX_i^+$
$D_i = B * (A9_i - (OX_i^+ + OX_i^-) * POLD_i) + POLD_i,$
blir ligningen

```
PO(I) = A(I)*POLD(IM) + C(I)*POLD(IP) + D(I),
```

som settes inn i hovedprogrammet under $ISOL = 3$ opsjonen.

Skisse til kode
Variabelen `ICRN` leses inn fra datafilen og skrives ut til resultatfilen.

```
SUBROUTINE FLOCON (----, ICRN)
.....
.....
     IF (ICRN .EQ. 1) THEN
          Implisitt formulering -- A_i,C_i,D_i, se neste side
     ELSE IF (ICRN .EQ. 2) THEN
          C-N formulering -- A_i,C_i,D_i, se neste side
     ELSE IF (ICRN .EQ. 3) THEN
          Eksplisitt formulering -- A_i,C_i,D_i, som over
     ENDIF
RETURN
END
```

Implisitt formulering (som for)

$b = ox^- + ox^+ + a2/\Delta t$
$a = ox^-/b$
$c = ox^+/b$
$d = (a2/\Delta t * pold + a9_i)/b$

C–N:

$$
\begin{aligned}
& im \\
& ip \\
& b = ox^- + ox^+ + 2 * a2/\Delta t \\
& a = ox^-/b \\
& c = ox^+/b \\
& d \;=\; (ox^+ * (pold_{ip} - pold_i) + ox^- * (pold_{im} - pold_i) + \\
& \qquad 2 * a9_i + 2 * a2/delt * pold_i)/b
\end{aligned}
$$

Med det foreslĘtte valg av `ICRN` og `ISOL`, mĘ `ISOL = 3` nĘr `ICRN =3`.

ICRN = 1, $\begin{cases} ISOL = 1 \\ ISOL = 2 \end{cases}$

ICRN = 2, $\begin{cases} ISOL = 1 \\ ISOL = 2 \end{cases}$

ICRN = 3, ISOL = 3.

Øving 5

Variabel k, ϕ og Δx

5.1 Oppgave

Modifiser programmet til å inkludere varierende horisontal *permeabilitet, porøsitet og blokklengde*. Bruk implisitt formulering og `TRIDIA`.

Programmet kontrolleres med samme datasett som i forrige øving med følgende unntak:

- $\phi = 0.01$ for alle blokker.
- `MX = 8`.
- Simuler fram til stasjonær tilstand, `STMAX` ≈ 20.

Simuler to tilfeller:

1. Alle blokkene er 10 ft lange med alternerende 75 md og 150 md permeabilitet.
2. Uniform permeabilitet lik 100 md og blokklengder på: 10, 5, 5, 20, 20, 5, 5, 10 ft.

Dersom programmeringen er utfort riktig, skal begge tilfellene gi samme trykkfall, Δ`p = po(8) - po(1)`, ved stasjonær tilstand.

5.2 Kommentarer

Det skal legges inn variabel permeabilitet, porøsitet og blokklengde. Dette gjøres for bedre å kunne tilpasse modellen til reservoardata.

F. eks.— Anta at en skal simulere trykkendringene rundt en produksjonsbronn. (Dette krever egentlig radielt koordinatsystem som vil bli gjennomgått i Øving 7.) Da kan en bruke modellen slik:

- Blokk 1 utgjør brønnen og må ha et porevolum lik brønnvolumet. Dette kan gjøres kunstig ved at $\phi(1) \gg 1$ og samtidig at $k(1)$ er stor.
- Blokk 2 kan utgjøre skadet sone med en lavere permeabilitet enn ute i formasjonen.
- Blokk 1 og 2 må generelt kunne ha ulik lengde. I tillegg ønskes en finere blokkinndeling i de områder hvor det skjer størst trykkendringer, altså nærmest brønnen, for å få optimal noyaktighet. En tommelfinger-regel er at trykkfallet per blokk skal være likt ved stasjonær tilstand.

La `ip = i+1` og `im = i-1`. Fra for har en

$$OX_i^- = \left(\frac{Ckb}{\mu}\right)_i^- / \Delta x_i^- / \Delta x_i = Ck_i^- \overline{\overline{\left(\frac{b}{\mu}\right)}} / \Delta x_i^- / \Delta x_i$$

$$OX_i^+ = \left(\frac{Ckb}{\mu}\right)_i^+ / \Delta x_i^+ / \Delta x_i = Ck_i^+ \overline{\overline{\left(\frac{b}{\mu}\right)}} / \Delta x_i^+ / \Delta x_i.$$

Her er $\overline{\overline{(b/\mu)}}$ tids- og avstandsmiddel, (μ er foreløpig konstant), $k_i^{\pm}$ er permeabiliteten tatt ved blokkgrensene. Når $k_i \neq k_{im}, k_{ip}$ brukes for $k_i^{\pm}$ middelverdien mellom blokkene, altså seriekobling av permeabiliteten:

$$\frac{\Delta x_i/2 + \Delta x_{ip}/2}{k_i^+} = \frac{\Delta x_i/2}{k_i} + \frac{\Delta x_{ip}/2}{k_{ip}}$$

Siden $\Delta x_i/2 + \Delta x_{ip}/2 = \Delta x_i^+$ får en

$$k_i^+ = \frac{2\Delta x_i^+}{\dfrac{\Delta x_i}{k_i} + \dfrac{\Delta x_{ip}}{k_{ip}}},$$

og

$$OX_i^+ = \frac{2C\overline{\overline{\left(\dfrac{b}{\mu}\right)}}}{\left(\dfrac{\Delta x_i}{k_i} + \dfrac{\Delta x_{ip}}{k_{ip}}\right)\Delta x_i},$$

$$OX_i^- = \frac{2C\overline{\overline{\left(\dfrac{b}{\mu}\right)}}}{\left(\dfrac{\Delta x_i}{k_i} + \dfrac{\Delta x_{im}}{k_{im}}\right)\Delta x_i}.$$

<u>Tips til koding</u>

- Dimensjoner størrelser:
  `ckx, dex, phi, volphi, a2, vol, kx.`
- `dex(i), kx(i), phi(i)` leses inn og skrives ut på samme måte som `qo(i).`
- Der hvor disse opptrer i koden må en nå enten bruke sløyfer med `do`–setninger, eller med litt omtanke bruke F90 sine regneregler for vektorer.
- Skift ut `ckbak` med `ckx(i)`,
  `ckx(i) = 2.do * c / visorg / (dex(i)/kx(i) + dex(im)/kx(im))/dex(i)`
- Tidligere i `flocon`: `oxplus(im) = oxmin(i).`
  Nå: `oxplus(im) = oxmin(i)*dex(i)/dex(im).`
- Kjør først med et datasett som en kjenner løsningen til.

Øving 6

Hellende reservoar

6.1 Oppgave

Utvid programmet til å inkludere helning. Bruk data som i øving 5, med $\alpha = +45^\circ$ og $\alpha = -45^\circ$, `RHOSTD` = 0.38 psi/ft. Kjor programmet til stasjonær tilstand og kontroller svaret ved å sammenlikne med Darcy's lov.

6.2 Kommentarer

Simuleringsligningen for et horisontalt system er

$$\frac{\partial}{\partial x}\left(\frac{Ck_o b_o}{\mu_o}\frac{\partial P_o}{\partial x}\right) + A9 = A2\frac{\partial P_o}{\partial t}.$$

I Figur 6.1 er vist en skisse av et reservoar med helning. Det angitte horisontalplanet er valgt til et datumplan. Dette innebærer at det vertikale hoydenivået settes lik null i trykkpunktet til forste blokk. I figuren regnes α negativ, slik at $\sin\alpha = dh/dx$ er negativ. Trykket i ligningen må nå erstattes med potensialet:

$$\frac{\partial P_o}{\partial x} \leftarrow \frac{\partial P_o}{\partial x} + \rho_o\frac{dh}{dx}, \quad \Phi_o = P_o + \rho_o h.$$

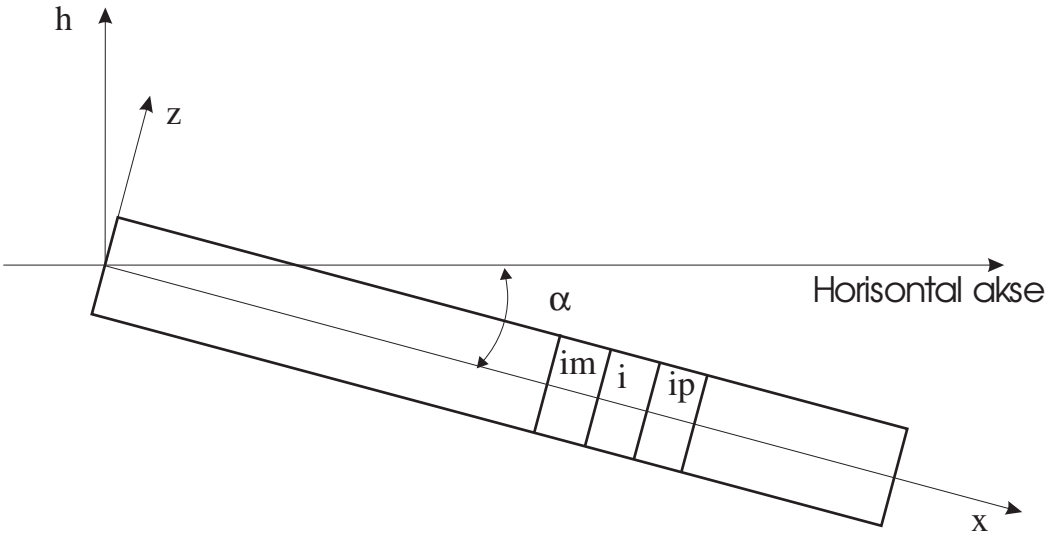
Her er ρ_o tettheten av olje ved reservoarforhold, dimensjonen er $lb_f/ft^3/144$, eller psi/ft. Simuleringsligningen på differanseform for horisontalt system er

$$\begin{cases} O_{xi}^- \cdot \Delta p_i^- + O_{xi}^+ \cdot \Delta p_i^+ + a9_i = H.S. \\ \Delta p_i^- = p_{im} - p_i \\ \Delta p_i^+ = p_{ip} - p_i. \end{cases}$$

Dette må endres til

$$\begin{cases} \Delta p_i^- = p_{im} - p_i - gho_i^- \\ \Delta p_i^+ = p_{ip} - p_i + gho_i^+. \end{cases}$$

Her er



Figur 6.1: Endimensjonalt reservoar med helning

`gho`: Gravity Head Oil

$\texttt{gho}_i^-$: $\texttt{(rho}_{im}\texttt{ * dex}_{im}\texttt{ + rho}_i\texttt{* dex}_i\texttt{) * 0.5 * dhdx}$

$\texttt{gho}_i^+$: $\texttt{gho}_{ip}^-$

`dhdx`: dh/dx

`rhostd`: $\rho_o^{st} + \rho_g^{std} \cdot R_{so}/FPB$

`rho`: ρ_o.

Dersom vi i tillegg definerer $gho_i \equiv gho_i^-$, kan alle gravitasjonsleddene samles opp i $a9_i$:

$$a9_i = qo_i + gho_{ip} * ox_i^+ - gho_i * ox_i^- .$$

Forslag til koding

- `gho(20)` deklareres
- Inne i `flprop`, etter at `bo` er regnet ut:

```
if(dabs(dhdx) .gt. 1.D-06) then
   rhoim = rhostd * bo(1)
   gho(1) = rhoim * dex(1) * 0.5D0 * dhdx
   do i = 2, mx
      im = i -1
      rhoi = rhostd * bo(i)
      gho(i) = (rhoi * dex(i) + rhoim * dex(im))* 0.5D0 * dhdx
```

```
            rhoim = rhoi
        end do
    end if
```

- I `flocon`: $a9_i$ = qo_i+ gho_{ip} * ox_i^+ - gho_i * ox_i^-
- I hovedprogrammet
  - Les inn og skriv ut `rhostd, dhdx`
  - For initiell trykkfordeling skrives ut og `OOIP` beregnes og skrives ut, utføres følgende:

```
if(dabs(dhdx) .gt. 1.D-06) then
    do k = 1, 4
        call flprop .........
        do i = 2, mx
            po(i) = po(i-1) - gho(i)
        end do
    end do
end if
```

Øving 7

Radielt system

7.1 Oppgave

1. Utvid programmet til å inkludere
   (a) Alternativt lineært eller radielt system,
   (b) Opsjon for produksjon mot konstant trykk ved utløpet av blokk nr. 1
   (c) Innlesing av nye data ved tid `tqchg`: `qo(1), qo(mx), delmin, delmax, dtmult, dpmax, pconst, tqchg`.
2. Test programmet på et lineært, kjent system
3. Simuler et radielt system med følgende egenskaper ved å bruke implisitt formulering og `tridia`:

$r_w = 0.25$	$\Delta x_5 = 40.5$	$k_x = 200md$	`qo(1)` $= -75$
$\Delta x_1 = 0.5$	$\Delta x_6 = 121.5$	$\phi = 0.01$	$\phi(8) = 10000$
$\Delta x_2 = 1.5$	$\Delta x_7 = 477.75$	`delmin` $= 0.0005$	
$\Delta x_3 = 4.5$	$\Delta x_8 = 660.$	`dtmult` $= 1.5$	
$\Delta x_4 = 13.5$	$\Delta y = 10.$	`delmax` $= 1.0$	
	$\Delta z = 1.$	`dpmax` $= 10.$	

   og ellers som før.

   Simuler en trykkfallstest til nøyaktig 0.02 dager med en påfølgende trykkoppbyggingstest.
4. Beregn k_x fra de to testene og sammenlign med innlest k_x.
5. Kjør trykkfallstesten ut til stasjonær tilstand og sammenlign simulert trykkfall med Darcy's lov.

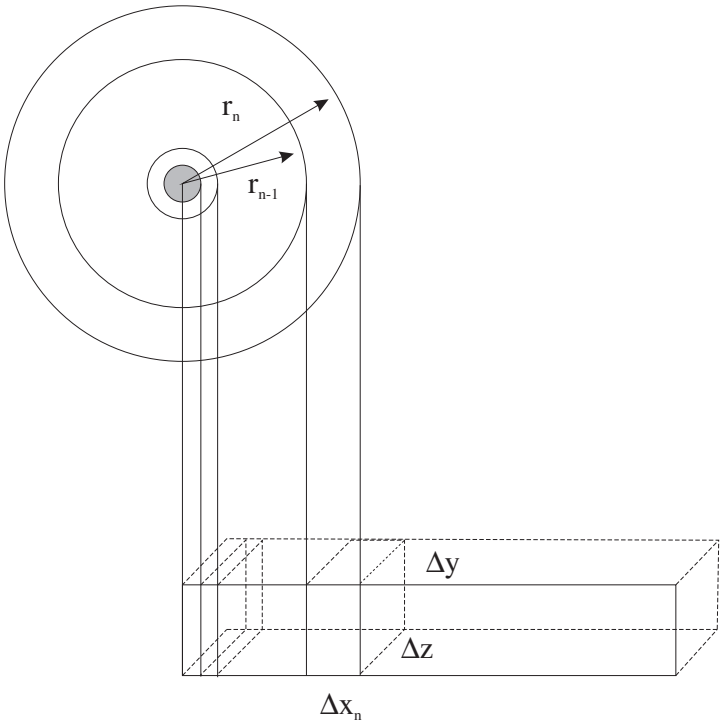
NB: Øvingen er obligatorisk, og følgende skal innleveres for godkjenning:

- En listing av programkode
- En kjøring som i spørsmål 3
- Plott av trykkfallstest og trykkoppbyggingstest med tilhørende beregninger av k_x, som i spørsmål 4
- Beregningen i spørsmål 5

7.2 Teori

Det lineære systemet som nå er programmert kan også brukes til å simulere radiell strøm i en dimensjon dersom en foretar en enkel transformasjon av porøsiteten og permeabiliteten. Denne transformasjonen er utledet i dette teoriavsnittet. (For en mer omfattende diskusjon henvises det til boken av Aziz og Settari, avsn. 3.6).

Figur 7.1 viser et radielt system med konsentriske ringer, alle med høyde h. Selve brønnen med radius r_w er skravert. For en generell ring n er ytre radius lik r_n og indre radius lik r_{n-1}. I eksemplet i figuren er $n = 4$. En ønsker nå å transførmere ring n med bredde $r_n - r_{n-1}$ øg høyde h til en blokk i det skisserte lineære system med lengde $\Delta x_n = r_n - r_{n-1}$, høyde $\Delta y = h$, og vilkårlig dybde Δz. En kan se at dybden er vilkårlig siden alle additive ledd i den transformerte ligningen vil inneholde Δz som faktor.



Figur 7.1: Transformasjon fra radielt til lineært system.

Tranformasjonen skal være slik at det lineære systemet, som vi allerede har tilgjengelig programkode for, skal kunne brukes til også å simulere det radielle systemet ved at porevolumet til ringen settes lik porevolumet til blokken, og at trykkfallet over ringen settes lik trykkfallet over blokken, for samme gjennomstrømningsrate.

Settes porevolumene like får en

$$\pi(r_n^2 - r_{n-1}^2)\Delta y \phi_r = \Delta z \Delta y \Delta x_n \phi_l,$$

hvor ϕ_r er den virkelig, innleste porøsitet i det radielle system, mens ϕ_l er den fiktive, transformerte porøsiteten i det ekvivalente lineære system. Løser en for ϕ_l fås

$$\phi_l = \frac{\pi(r_n + r_{n-1})}{\Delta z}\phi_r. \qquad (7.1)$$

Skal trykkfallene være like for de to systemene når raten er lik, så må vi ha

$$\left(\frac{q}{\Delta p}\right)_{\text{radiell}} = \left(\frac{q}{\Delta p}\right)_{\text{lineær}},$$

og dermed

$$\frac{2\pi k_{xr}\Delta y}{\mu \ln\left(\frac{r_n}{r_{n-1}}\right)} = \frac{k_{xl}\Delta z \Delta y}{\mu \Delta x},$$

og løst med hensyn på den transformerte, fiktive, ekvivalente, lineære permeabilitet k_{xl} får vi

$$k_{xl} = \frac{2\pi\Delta x}{\Delta z \ln\left(\frac{r_n}{r_{n-1}}\right)} k_{xr}. \qquad (7.2)$$

Her er på samme måte som for porøsitet i ligning 7.1 k_{xr} den reelle, radielle permeabiliteten.

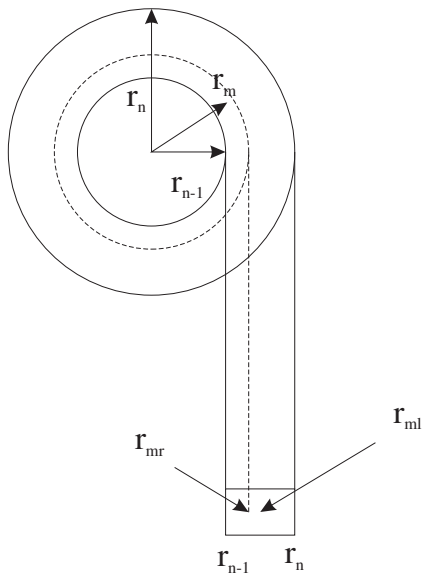
På tilsvarende måte kan en lett vise at dersom en har to-dimensjonal strøm vil en få for permeabiliteten i y-retning:

$$k_{yl} = \frac{\pi(r_n + r_{n-1})}{\Delta z} k_{yr}. \qquad (7.3)$$

Ligningene 7.1 og 7.2 er de to vi trenger. Vi må imidlertid også ta hensyn til at trykkpunktet forskyver seg fra midtpunktet når en går over til et radielt system. Dette er illustrert i Figur 7.2. Med trykkpunkt mener vi den posisjonen i ringen, r_{mr} som gir midlere trykk i ringen, i den forstand at trykkfallet fra r_n til r_{mr} skal være lik trykkfallet fra r_{mr} til r_{n-1}. Ved bruk av Darcy's lov får en da

$$\frac{r_{mr}}{r_n} = \frac{r_{n-1}}{r_{mr}}. \qquad (7.4)$$

For å få samme nøyaktighet for alle trykkene kan det vises at blokklengene bør velges slik at trykkfallet er konstant over hver blokk ved stasjonær strøm.



Figur 7.2: Posisjonen til middeltrykk i radielt system.

7.3 Kommentarer

Kommentar til spørsmål 1a

Kodeforslag

```
if (rw .gt. 0.d0) then                  ! rw leses og skrives
rp = rw                                 ! rm: r_{n-1}
do i = 1, mx                            ! rp: r_n
rm = rp
rp = rm + dex(i)
x(i) = dsqrt(rm*rp)                     ! x(i): Trykkposisjon
phi(i) = pi*(rp + rm)/dez*phi(i)
kx(i) = 2.d0*pi*dex(i)/dez/dlog(rp/rm)*kx(i)
 end do
endif
```

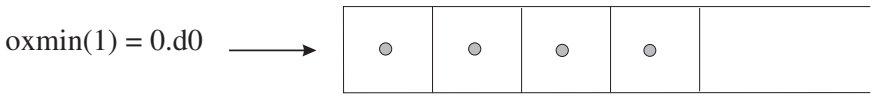
Dette, som er alt som trengs for å simulere et radielt system, kan utføres i hovedprogrammet for tidsstegsløyfen starter.

Kommentar til spørsmål 1b

Fra før har vi ligningen

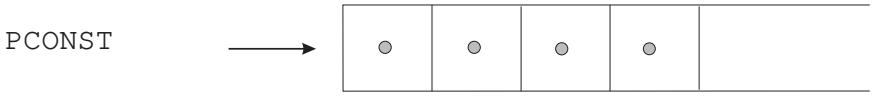
$$ox_i^+ \Delta p_i^+ + ox_i^- \Delta p_i^- + a9_i = h.s.,$$

hvor $ox_1^- = 0.d0$, altså ingen strøm ut av venstre endeflate av blokk 1, som antydet i Figur 7.3. Vi ønsker nå å inkludere mulighet for å produsere mot konstant trykk `p-`



Figur 7.3: Ingen strøm ut av venstre endeflate.

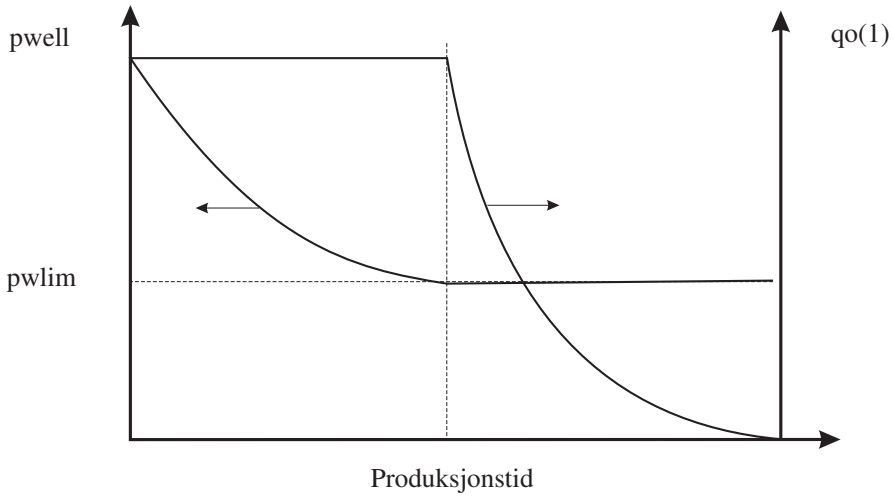
`const` ved utløpet av blokk 1, antydet i Figur 7.4. Dette gjøres for å simulere mer



Figur 7.4: Produksjon mot konstant trykk.

realistiske produksjonsvilkår. Når en brønn starter å produsere, vil det ofte være med konstant rate, som kontraktsmessig er avtalt, inntil trykket i bunn av brønnen har falt av så mye at raten ikke kan opprettholdes. Da vil trykket holdes konstant (egentlig på første separator på overflaten) mens raten faller av. For å kunne simulere dette nøyaktig trenger en å kople en brønnmodell til reservoarmodellen, men tilnærmet vil den foreslåtte metode kunne fungere.

Samspillet mellom trykket i bronnen, `pwell`, og raten er vist i Figur 7.5. I figuren



Figur 7.5: Overgang fra konstant rate til konstant trykk.

er `pwell` bunnhullstrykket slik modellen vår beregner det, og `pwlim` er en innlest verdi for minimum bunnhullstrykk. I programkoden foreslås det å bruke størrelsen *pconst* for bunnhullstrykket, slik at `pconst = 0.` når brønnen produserer med konstant rate, og `pconst = pwlim` når brønnen produserer mot konstant trykk.

Variabelen `pconst` leses inn og skrives ut. Dersom $rw > 0$ får en å skille mellom to tilfeller, avhengig av om `pconst` er satt mindre eller lik 0, eller større enn null.

`pconst` $\leq$ `0.` En produserer med konstant rate fra blokk nr. 1 og trykket ved utløpet av blokk 1, `pwell`, regnes ut for hvert tidssteg fra ligningen

$$qo(1) = ox_1^- * (pwell - po(1)), \qquad \ldots\ldots\ldots\ldots\ldots\ldots \quad (7.5)$$

og skrives ut. Denne ligningen er Darcy's lov brukt fra sentrum av blokk 1 og ut til venstre endeflate.

Det er ikke nødvendig å gjøre noen endringer i koden for å få dette til, men en bør merke seg at `pwell` regnes ut etter ligning 7.5 etter at trykkløsningen har konvergert. Merk også at ox_1^- fortsatt skal være null i `flocon`, slik at en kanskje med fordel kan bruke et annet variabelnavn for ox_1^- i ligning 7.5.

Dersom `pwell` faller under innlest verdi `pwlim` settes `pconst= pwlim` og en går over til neste tilfelle.

`pconst` $>$ `0.` For blokk 1 har vi fra før

$$ox_1^- = 0.d0, \quad ox_1^+(po_2 - po_1) + a9_1 = h.s.$$

Dette endres nå til

$$ox_1^-(pconst - po_1) + ox_1^+(po_2 - po_1) + a9_1 = h.s.,$$

hvor $ox_1^- \neq 0$, og kan regnes ut direkte fra Darcy's lov, eller ved å sette `kx(im)` lik uendelig i det generelle uttrykket for `ckx(i)`. En finner at

$$ckx(1) = \frac{2Ckx(1)}{\mu \Delta x_1^2}.$$

I `flocon` må en nå utføre følgende

```
a9(1) = oxmin(1)*pconst
a(1) = 0.d0
c(1) : som før
d(1) : som før
```

Merk at på denne måten går `oxmin(1)` automatisk inn i koeffisienten `b`, slik at alle leddene i ligning 7.5 blir tatt vare på.

I hovedprogrammet, etter at trykkløsningen har konvergert, regnes raten ut etter ligning 7.5 og raten i stb/d finner en ved å multiplisere med `vol(1)`. Denne verdien har en ikke bruk for før den skal skrives ut og en skal gjøre materialbalanse.

Dersom `pconst` endres til en verdi mindre enn null igjen, må en huske å nullstille `oxmin(1)` i `flocon`.

Denne delen av koden kan testes ved å lese inn f.eks. `pwlim=1950`, `pconst = 0.d0` og ellers simulere med data som under spørsmål 3. i denne oppgaven.

Kommentar til spørsmål 1c

I hovedprogrammet, etter start av tidstegssløyfen, `do l = 1, stmax`, men før tidsstegsberegningen setter en inn følgende kode:

```
    if (int .eq. 1) then
       int = 0
       read(5, .....) qo(1), qo(mx), delmin, delmax,
*       dtmult, dpmax, pconst, tqchg
       write(6, .....) det samme
       if (tqchg .lt. 0.d0) then
          stop
       endif
       qo(1) = qo(1)/vol(1)
       qo(mx) = qo(mx)/vol(mx)
       a9(1) = .....
       a9(mx) = .....
       delt = delmin
    endif
```

Merk at den første `tqchg` må leses inn og skrives ut sammen med det første datasettet.

I hovedprogrammet, etter at tidsstegslengden `delt` er beregnet, settes inn følgende tilleggssjekk:

```
    if(ctim + delt .gt. tqchg) then
       int = 1
       delt = tqchg - ctim
    endif
```

Merk at dersom en i løpet av simuleringen får tidsstegsreduksjon på grunn av verdien til `dpmax` må en sette `int = 0` igjen.

Kommentar til spørsmål 4

Her plottes `pwell` mot $\ln t$ for trykkfallstesten, og for trykkoppbyggingstesten bruker en `po(1)`.

Øving 8

Tabellbruk

8.1 Oppgave

1. Lag et programsegment som leser inn (p_o, b_o, μ_o) ved en rekke vilkårlige trykk og som genererer en intern, ekvidistant trykktabell over disse størrelsene.

PVT-data:

p_o	b_o	μ_o
0	0.7401	1.00
2013	0.7506	1.01
4000	0.7752	1.02

2. Modifiser programmet slik at b_o og μ_o leses fra den internt genererte tabellen for et vilkårlig trykk. Test programmet på det radielle system brukt i Øving 7.3. Sjekk permeabiliteten beregnet fra simulert trykk med den innleste permeabilitet.

8.2 Kommentarer

For en nærmere beskrivelse, se boken til Aziz og Settari, side 409.

Hittil har vi brukt

$$b_o = borig + dbdp * \Delta p,$$

hvor $dbdp$ er en konstant, og dessuten at μ_o er konstant. Generelt er volumfaktorer og viskositeter ikke-lineære funksjoner av trykket. Senere vil vi også få bruk for relative permeabiliteter og kapillartrykk som ikke-lineære funksjoner av metninger. Disse funksjonssammenhengene innføres vanligvis i et simuleringsprogram ved hjelp av tabeller. Under kjøring av modellen interpolerer programmet i tabellene. Interpolasjonen er vanligvis lineær.

Rask interpolering kan foretas dersom tabellen er ekvidistant i trykk (eller metning). Laboratoriedata er vanligvis ikke ekvidistante (i trykk) og en trenger derfor å generere en intern, ekvidistant tabell før simuleringen starter.

Følgende inngangstabell leses fra datafil og skrives ut til resultatfil:

p_1	b_1	μ_1
p_2	b_2	μ_2
p_3	b_3	μ_3
$\vdots$	$\vdots$	$\vdots$
p_{npvt}	b_{npvt}	μ_{npvt}

En leser inn (og skriver ut) `pmax` $=p_{nvpt}$, `npvt` og antar at $p_1 < p_2 < \cdots < p_{npvt}$ og at $p_1 = 0$. Fra denne innleste tabell genereres det så en intern, ekvidistant tabell i p på 81 linjer, fra p_1 til p_{npvt}.

I kodeforslaget under brukes følgende betegnelser:
`pv(1) = p, pv(2) = b, pv(3) =` μ.

En må videre huske å dimensjonere størrelsene:
`pv(3),pvld(3), dp(3), tpo(81), tbo(81), tvo(81).`

Størrelsen `itab`, som dirigerer utskrift av intern tabell, må også leses og skrives.

```
    pinc = pmax/80.d0
    dpres = 0.d0
    read(5,...) (pv(i), i=1,3)
    write(6,...) (pv(i), i=1,3)

    do i = 1,3
       pvld(i) = pv(i)
    end do

    tpo(81) = pv(1)
    tbo(81) = pv(2)
    tvo(81) = pv(3)
    dp(1) = 1.d0
    notab = 1

    do k = 2, npvt
       read(5,...) (pv(i), i = 1, 3)
       write(6,...) (pv(i), i = 1, 3)
       kinc = (pv(1) - pvld(1))/pinc + 0.5d0
       do kk = 2, 3
          dp(kk) = (pv(kk) - pvld(kk))/(pv(1) - pvld(1))
       end do

       kr = 0

       do kk = 1, kinc
          kr = kr + 1
          delp = kr*pinc - dpres
          notab = notab + 1
          no = 82 - notab
          tpo(no) = pvld(1) + dp(1)*delp
          tbo(no) = pvld(2) + dp(2)*delp
          tvo(no) = pvld(3) + dp(3)*delp
       end do

       do kk = 1, 3
          pvld(kk) = pv(kk)
       end do
       dpres = pv(1) - tpo(no)
    end do
    if(itab .eq. 1) write(6,...) (tpo(i),tbo(i),tvo(i), i=1,81)
```

Som eksempel på bruk av den internt genererte tabellen tenker vi oss at en ønsker å finne b_o ved $p_o = 3920$ psi. Den interne tabellen er:

	tpo	tbo	tvo
1	4000.	-	-
2	3950.	-	-
3	3900.	-	-
⋮	⋮	⋮	⋮
81	-	-	-

Først beregnes et heltall `in` som angir at det oppgitte trykk ligger mellom tabellinje `in` og `inm = in - 1`. Deretter foretas det lineær interpolering innen dette intervallet. Heltallet finnes slik:

```
in  =  2 + (tpo(1) - po(i))/pinc
    =        2 + (4000 - 3920)/50
    =           2 + 80/50 = 3
```

Deretter interpoleres det lineært i tabellen mellom `in` og `inm`:

$$\begin{aligned}
\texttt{zmc} &= (\texttt{po(i)} - \texttt{tpo(in)})/\texttt{pinc} \\
b_o &= \texttt{tbo(in)} + \texttt{zmc} * (\texttt{tbo(inm)} - \texttt{tbo(in)}) \\
\mu_o &= \texttt{tvo(in)} + \texttt{zmc} * (\texttt{tvo(inm)} - \texttt{tvo(in)})
\end{aligned}$$

Med denne måten å angi volumfaktoren på, vil den ikke nødvendigvis være en lineær funksjon av trykket. En må nå regne ut en variabel `dbdp`. Dette framkommer ved å utvikle høyre side av simuleringsligningen:

$$\begin{aligned}
\phi \frac{\partial b_o}{\partial t} &= \phi \frac{b_o^{t+\Delta t} - b_o^t}{\Delta t} \\
&= \phi \frac{b_o - b_{old}}{delt} \\
&= \phi \frac{b_o - b_{old}}{p_o - p_{old}} \frac{p_o - p_{old}}{delt}
\end{aligned}$$

En får da å definere

$$\texttt{dbodp(i)} = (\texttt{bo(i)} - \texttt{bold(i)})/(\texttt{po(i)} - \texttt{pold(i)}),$$

og å dimensjonere $dbodp(20), vo(20), vold(20), bold(20)$. Andre kodetips er som følger:

- I hovedprogrammet
  - For å beregne `ooip`

```
ipre = 3
call flprop(....,ipre)
```

- Ta `visorg` ut av `ckx`
- Rett etter start av tidstegsløyfen

```
bold(i) = bo(i)
vold(i) = vo(i)
```

- Foran `call flocon`

  Ta vekk midling av trykk

```
ipre = 2
call flprop(....,ipre)
```

- Foran materialbalansen settes inn

```
ipre = 1
call flprop(....,ipre)
```

- I `flocon`

```
oxmin(i)=ckx(i)*(bo(i)/vo(i)+bo(im)/vo(im))*0.5d0
```

- I `flprop`

```
do i=1,mx
    in=2+(tpo(1)-po(i))/pinc
    inm=in-1
    zmc=(po(i)-tpo(in))/pinc
    bo(i)=tbo(in)+zmc*(tbo(inm)-tbo(in))
    vo(i)=tvo(in)+zmc*(tvo(inm)-tvo(in))
    if (ipre .eq. 3) then
        dbodp(i)=(tbo(in)-tbo(inm) )/(tpo(in)-tpo(inm))
    end if
    if (ipre .eq. 2) then
        dp=po(i)-pold(i)
        if(dabs(dp) .gt. 1.d-06) then
            dbodp(i)=(bo (i)-bold(i))/dp
        end if
        bo(i)=(bo(i)+bold(i))* 0.5d0
        vo(i)=(vo(i)+vold(i))*0.5d 0
    end if
    beregn 'gravity heads'
end do
```

Øving 9

En fase i to dimensjoner

Utvid programmet til også å inkludere strøm i to dimensjoner for en fase i et horisontalt system.

- Bruk implisitt formulering
- Løs ligningssettet med LSOR eller eliminasjon og sammenlign
- Skriv programmet slik at det fremdeles kan simulere strøm i en dimensjon dersom `mx = 1` eller `my = 1`
- Dimensjoner til et 5*5 system
- Testdata er

  `mx = my =5`
  `Δx = 10, i = 1, mx`
  `Δy = 10, j = 1, mx`
  `Δz = 1`
  `kx = 100,` alle blokker
  `ky = 100,` alle blokker
  `qo(1,1) = - 1.0`
  `qo(5,5) = 1.0`
  ϕ `= 0.05,` alle blokker
  `stmax = 5`
  væskeegenskaper som i Øving 8
  `delmin, delmax, dtmult, dpmax` som i øving 5

- Dersom programmet fungerer riktig skal følgende være oppfylt:
  - Materialbalansen er iorden
  - Trykkene er symmetriske og antisymmetriske om diagonalene i rutenettet

9.1 Kommentarer til Øving 9

For strøm av en fase i en dimensjon har vi fra før

$$\frac{\partial}{\partial x}\left(\frac{ckb}{\mu}\frac{\partial p}{\partial x}\right)+q=\phi\frac{db}{dp}\frac{\partial p}{\partial t},$$

og utvidet til to dimensjoner får en etter en helt tilsvarende utledning som for en dimensjon

$$\frac{\partial}{\partial x}\left(\frac{ckb}{\mu}\frac{\partial p}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{ckb}{\mu}\frac{\partial p}{\partial y}\right)+q=\phi\frac{db}{dp}\frac{\partial p}{\partial t}.$$

Også helt analogt til en dimensjon gir nå diskretiseringen

$$ox^{+}\Delta p_x^{+}+ox^{-}\Delta p_x^{-}+oy^{+}\Delta p_y^{+}+oy^{-}\Delta p_y^{-}+a9=a2(p_{i,j}-p_{i,j}(t))/\Delta t. \tag{9.1}$$

I denne ligningen er $\Delta p_x^{+}=p_{ip,j}-p_{i,j}$, trykkdifferansen mellom to naboblokker i x -retning, ip = i +1 , trykk $p_{i,j}(t)$ er tatt ved gammel tid t, mens de andre trykkene er ved ny tid $t+\Delta t$, (implisitt formulering). Variabelen i angir blokk nummer i x -retning, og j i y -retning:

(i,jp)

(im,j) (i,j) (ip,j)

(i,jm)

Setter en inn i ligning 9.1 for trykkdifferansene får en

$$\begin{aligned} &ox_{i,j}^{+}(p_{ip,j}-p_{i,j})+ox_{i,j}^{-}(p_{im,j}-p_{i,j})+\\ &oy_{i,j}^{+}(p_{i,jp}-p_{i,j})+oy_{i,j}(p_{i,jm}-p_{i,j})+a9_{i,j} \quad = \quad a2(p_{i,j}-p_{i,j}(t))/\Delta t. \end{aligned} \tag{9.2}$$

Samler en nå ledd med like trykk får en

$$\begin{aligned} & ox_{i,j}^{-}\,p_{im,j}\\ &+oy_{i,j}^{-}\,p_{i,jm}\\ -(ox_{i,j}^{-}+ox_{i,j}^{+}+oy_{i,j}^{-}+oy_{i,j}^{+}+a2_{i,j}/\Delta t)\,&p_{i,j}\\ &+ox_{i,j}^{+}\,p_{ip,j}\\ &+oy_{i,j}^{+}\,p_{i,jp} \quad = \quad -(a2_{i,j}p_{i,j}t/\Delta t+a9_{i,j}). \end{aligned}$$

Nå settes, for hver (i,j), $bb=ox^{-}+ox^{+}+oy^{-}+oy^{+}+a2/\Delta t$, og en definerer

$$\begin{aligned} \mathtt{a(i,j)} &= ox_{i,j}^{-}/bb,\\ \mathtt{c(i,j)} &= ox_{i,j}^{+}/bb,\\ \mathtt{e(i,j)} &= oy_{i,j}^{-}/bb,\\ \mathtt{f(i,j)} &= oy_{i,j}^{+}/bb,\\ \mathtt{d(i,j)} &= (a2_{i,j}\,p_{i,j}(t))/\Delta t+a9_{i,j})/bb, \end{aligned}$$

og ligning 9.1 blir

$$\mathtt{a(i,j)p(im,j)+c(i,j)p(ip,j)-p(i,j)+e(i,j)p(i,jm)+f(i,j)p(i,jp)=-d(i,j)}. \tag{9.3}$$

For å løse ligning 9.3 ved eliminasjon, må en først velge et nummereringssystem for blokkene. Dette kan gjøres på to hovedmåter,

- Naturlig nummerering,
- Alternerende diagonal nummerering.

La oss først se på naturlig nummerering, og vi bruker som eksempel et 5*4 blokksystem, nummerert som vist under

4	8	12	16	20
3	7	11	15	19
2	6	10	14	18
1	5	9	13	17

Vi antar at alle `a(1,j)`, `c(mx,j)`, `e(i,1)`, `f(i,my)` er lik null ved at de tilsvarende $\mathtt{ox}^{\pm}, \mathtt{oy}^{\pm}$ er satt lik null. Fra ligning 9.3 får en da det lineaære ligningssystemet

$$\vec{\vec{K}} \cdot \vec{p}^t = \vec{d}^t,$$

hvor koeffisientmatrisen $\vec{\vec{K}}$ er gitt ved

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	-1	f_1			c_1															
2	e_2	-1	f_2			c_2														
3		e_3	-1	f_3			c_3													
4			e_4	-1				c_4												
5	a_5				-1	f_5			c_5											
6		a_6			e_6	-1	f_6			c_6										
7			a_7			e_7	-1	f_7			c_7									
8				a_8			e_8	-1				c_8								
9					a_9				-1	f_9			c_9							
10						a_{10}			e_{10}	-1	f_{10}			c_{10}						
11							a_{11}			e_{11}	-1	f_{11}			c_{11}					
12								a_{12}			e_{12}	-1				c_{12}				
13									a_{13}				-1	f_{13}			c_{13}			
14										a_{14}			e_{14}	-1	f_{14}			c_{14}		
15											a_{15}			e_{15}	-1	f_{15}			c_{15}	
16												a_{16}			e_{16}	-1				c_{16}
17													a_{17}				-1	f_{17}		
18														a_{18}			e_{18}	-1	f_{18}	
19															a_{19}			e_{19}	-1	f_{19}
20																a_{20}			e_{20}	-1

og $\vec{p}^t$ er søylevektoren av de ukjente trykkene og $\vec{d}^t$ er søylevektoren av leddene på høyre siden av ligning 9.3. Koeffisientmatrisen $\vec{\vec{K}}$ er pentadiagonal med nuller utenom de 5 diagonalene. Båndbredden er `2*my + 1 = 9`. En teller hurtigst i den retning som har minst blokker for å få minst båndbredde.

I prinsipp kan dette lineære ligningssettet løses på samme måte som ved *tridia*: en skaffer først nuller under diagonalen mens en arbeider seg nedover, og går så oppover igjen.

Ved *alternerende diagonal nummerering*, med samme 5*4 system, velges følgende nummereringsmønster:

13	5	17	9	20
2	14	6	18	10
11	3	15	7	19
1	12	4	16	8

Koeffisientmatrisen $\vec{\vec{K}}$ blir da

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	-1										f_1	c_1								
2		-1									e_2		f_2	c_2						
3			-1								a_3	e_3		f_3	c_3					
4				-1								a_4			f_4	c_4				
5					-1								X	X			X			
6						-1								X	X		X	X		
7							-1								X	X		X	X	
8								-1								X			X	
9									-1								X	X		X
10										-1								X	X	X
11	X	X	X								$\oplus$	$\oplus$		$\oplus$	$\oplus$					
12	X		X	X							$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$					
13		X			X						$\oplus$	$\oplus$	$\oplus$	$\oplus$			$\oplus$			
14		X	X		X	X					$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$		$\oplus$	$\oplus$		
15			X	X		X	X				$\oplus$	$\oplus$		$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	
16				X			X	X				$\oplus$			$\oplus$	$\oplus$		$\oplus$	$\oplus$	
17					X	X			X				$\oplus$	$\oplus$	$\oplus$		$\oplus$	$\oplus$		$\oplus$
18						X	X		X	X				$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
19							X	X		X					$\oplus$	$\oplus$		$\oplus$	$\oplus$	$\oplus$
20									X	X							$\oplus$	$\oplus$	$\oplus$	$\oplus$

Matrisen er ikke skrevet helt ut, og X betegner et ledd av typen `a,c,e,f`. I nedre høyre kvadrant er det før eliminering kun tallet 1 på diagonalen som er forskjellig fra null.

En skaffer seg nå nuller under diagonalen ved å eliminere leddene i nedre venstre kvadrant. Dette medfører nye ledd i nedre høyre kvadrant —her er endrete ledd angitt

med symbolet $\oplus$. Disse nye leddene kan, med litt omtanke, settes opp direkte. Deretter løser en for trykkene til blokkene 11-20 ved å skaffe nuller under diagonalen i nedre høyre kvadrant etc. Til slutt løses for trykkene 1-10 ved å bruke opprinnelig ligning. Båndbredden er `2*my +1 = 9`, som før.

9.2 Løsning med naturlig nummerering og subrutinen SOLVE

```
    subrutine solve(c,b,ibdwth,n)
    implicit double precision(a-h,o-z)
c-----------------------------
c*** n: antall ligninger, n = my*mx
c*** ibdwth: båndbredden, enten lik 2*my + 1 eller
c... lik 2*mx + 1, avhengig av hva som er minst
c... For et 5*5 system er ibdwth = 11
c*** c: inneholder høyre siden av ligningen, konstantleddet
c... For et 5*5 system: c(25)
c*** b: koeffisientmatrisen, kun båndet er lagret
c... For et 5*5 system: b(25,11)
c-------------------------------------
    dimension c(n), b(n,ibdwth)
    ihalf = (ibdwth + 1)/2
    ihafm2 = ihalf - 2
    ihafm1 = ihalf - 1
    nm1 = n - 1
    index = 0
c
c-- Her begynner eliminasjonen under diagonalen
c
```

```
    do 60 k = 1, nm1
       if(dabs(b(k,ihalf)).lt.1.d-19) go to 25
       limit = k + ihafm2
       if (limit.gt.nm1) limit = nm1
       l = ihalf + index
       jlim = ibdwth + index
       do 50 i = k, limit
          lmi = l - i
          ip1 = i + 1
          if (b(ip1,lmi).eq.0.) go to 50
          factor = - b(ip1,lmi)/b(k,ihalf)
          m = ihafm1 + k
          do 30 j = m, jlim
             if (dabs(factor).lt.1.d-30) go to 30
             jmndx = j - index
             if (dabs(b(k,jmndx)).lt.1.d-30) go to 30
             jmi = j - i
             b(ip1,jmi) = factor * b(k,jmndx) + b(ip1,jmi)
30         continue
          b(ip1,lmi) = factor
50     continue
       index = index + 1
60  continue
c
c-Ferdig med å gjøre koeffisientmatrisen øvre triangulær
c-Justerer nå høyre siden av ligningen på tilsvarende måte
c
```

```
     index = 0
     do 61 K = 1, nm1
        limit = k + ihafm2
        l = ihalf + index
        if (limit.gt.nm1)limit = nm1
           do 51 i = k, limit
           ip1 = i + 1
           c(ip1) = c(ip1) + b(ip1,l-i)*c(k)
 51         continue
        index = index + 1
 61  continue
 c
 c-Regner ut trykket i siste blokken, og går så oppover
 c-langs diagonal igjen.
 c
     c(n) = c(n)/b(n,ihalf)
 c

    do 70 i = 1, nm1
       temp = 0.d0
       isub = n-i
       do 80 j = 1, ihafm1
          if (j.gt.i) go to 70
          temp = temp + b(isub,ihalf+j)*c(isub+j)
 80     continue
       c(isub) = (c(isub) - temp)/b(isub,ihalf)
 70  continue
    go to 26
 25  write(6,27)
 27  format(1h0,10x,' **** sorry, your pivot element is too small ***')
    stop
 26  return
    end
```

Før en kan kalle opp SOLVE og løse ligningssystemet, må en sette opp matrisene `bb(n,ibdwth)` og `cc(n)`. Vi bruker her `bb` og `cc` for ikke å komme i konflikt med andre symboler i hovedprogrammet. Vi antar at `my` $\leq$ `mx`. Merk at `ih` er posisjonen til diagonalleddet i det båndet som er lagret i `bb`-matrisen. Først nullstiller vi båndmatrisen:

```
      ibdwth = 2*my + 1
      ih = (ibdwth +1)/2
      ic = 0
      do 10 i = 1, mx
         do 20 j = 1, my
            ic = ic + 1
            do 30 ii = 1, ibdwth
               bb(ic,ii) = 0.d0
 30         continue
 20      continue
 10   continue
```

```
     ic = 0
     do 40 i = 1, mx
        do 50 j = 1, my
           ic = ic + 1
           bb(ic,1) = a(i,j)
           bb(ic,ih-1) = e(i,j)
           bb(ic,ih) = -1.d0
           bb(ic,ih+1) = f(i,j)
           bb(ic,ibdwth) = cc(i,j)
           cc(ic) = - d(i,j)
50      continue
40   continue
     n = mx*my
     call solve(cc,bb,ibdwth,n)
```

Løsningen returneres fra solve i matrisen `cc` og vi må fiske ut de todimensjonale trykkene fra denne endimensjonale matrisen:

```
                        ic = 0
        do 60 i = 1, mx
           do 70 j = 1, my
              ic = ic + 1
              p(i,j) = cc(ic)
     70      continue
     60   continue
```

Subrutinen *solve* bruker samme framgangsmåte som en har i *tridia*.

I boken til Aziz og Settari er det reprodusert kode til en annen løsningsrutine som en også kan bruke. Denne er basert på endimensjonale matriser.

9.3 Løsning med LSOR

I denne metoden, *Line SOR*, brukes SOR linjevis med kall til *tridia* for hver linje. Vi kan velge mellom å sveipe i x - eller y -retning, eller en blanding.

Denne metoden er mye brukt, spesielt i tredimensjonale modeller. Dens anvendelighet avhenger blant annet av et godt estimat på ω, og at reservoaret har liten anisotropi.

Vi ser på ligningen for blokk (i,j) og sløyfer disse indeksene på koeffisientene:

$$\mathtt{a}\,\mathtt{p}_{\mathtt{im,j}} + \mathtt{c}\,\mathtt{p}_{\mathtt{ip,j}} - \mathtt{p}_{\mathtt{i,j}} + \mathtt{e}\,\mathtt{p}_{\mathtt{i,jm}} + \mathtt{f}\,\mathtt{p}_{\mathtt{i,jp}} = -\mathtt{d}.$$

Dersom vi nå sveiper i x -retning, så skriver vi denne ligningen på formen

$$\mathtt{a}\,\mathtt{p}_{\mathtt{im,j}} - \mathtt{p}_{\mathtt{i,j}} + \mathtt{c}\,\mathtt{p}_{\mathtt{ip,j}} = -\mathtt{e}\,\mathtt{p}_{\mathtt{i,jm}} - \mathtt{f}\,\mathtt{p}_{\mathtt{i,jp}} - \mathtt{d}.$$

For $\mathtt{p}_{\mathtt{i,jm}}$ og $\mathtt{p}_{\mathtt{i,jp}}$ på høyre side av ligningen bruker vi de sist itererte verdiene. Dermed kan vi uttrykke høyresiden med et tall, og bruker *tridia* til å løse for de ukjente trykkene på venstre siden.

```
c
c- start iterasjonssløyfe for LSOR
c
   do 10 kkk = 1, 20
      isw = 0
      jm = 1
      do 20, j = 1, my
         if (j.gt.1) jm = j - 1
         if (j.lt.my) jp = j + 1
         im = 1
         do 30 i = 1, mx
            if(i.gt.1) im = i - 1
            if(i.lt.1) ip = i + 1
            aa(i) = a(i,j)
            cc(i) = c(i,j)
            dd(i) = d(i,j) + e(i,j)*p(i,jm) + f(i,j)*p(i,jp)
30        continue
         call tridia(mx, ···)
c
c -- Her overføres aa, cc og dd til tridia, og løsningen returneres
c -- i pp. Må derfor endre common block i tridia
c
         do 40 i = 1, mx
            p(i,j) = p(i,j) + om*(pp(i) - p(i,j))
            if (dabs(p(i,j) - pp(i)).gt. eps) isw = 1
40        continue
20     continue
   kkkcum = kkk
   if (isw.eq.0) hopp ut og skriv kkkcum
10  continue
```

Den teoretiske bakgrunnen for *LSOR* kan en finne i boken til Aziz og Settari.

9.4 Noen påminnelser

- De fleste "gamle" størrelsene får nå doble indekser og må dimensjoneres. Dette gjelder f.eks. `kx(5,5), phi(5,5), p(5,5), dex(5), ckx(5,5), etc.`
- `kx, ky, phi, q,` etc. leses og skrives for hver blokk
- Nye størrelser deklareres, som f.eks. `oyplus(5,5), oymin(5,5), dey(5), cky(5,5), ky(5,5), bb(25,11), cc(25), aa(5), cc(5), dd(5), pp(5)`

- De fleste "gamle" sløyfer med telleren `i` blir nå doble med både `i` og `j`.

Øving 10

To faser, olje og vann, i en dimensjon

Utvid programmet til også å inkludere strøm av de to fasene olje og vann i en dimensjon for et horisontalt system.

- Bruk kvasi-implisitt formulering med kordemetoden.
- Bruk p_o og Δs_o som ukjente.
- Løs ligningssettet med SOLVE, dvs. med eliminasjon eller simultan løsning for p_o og Δs_o.
- Legg inn opsjon for produksjon mot konstant trykk, `pconst`, ved utløpet av blokk 1.
- Les inn tabell av PVT-data og bergartsdata.
- Dimensjoner til 20 blokker.
- Bruk oppstrøms relative permeabiliteter.

Testeksempelet en skal simulere tilsvarer øving 10.2 i Dake sin lærebok [7], sml. også øving 10.1 i samme bok.

I dette eksempelet er kapillartrykket p_c satt lik null. Vi skal likevel formulere ligningene i modellen med kapillartrykksledd. En får bruk for dette i øving 11, som behandler simulering av et laboratorieeksperiment for å må relative permeabilitetskurver, med og uten kapillartrykk.

Følgende data inngår:

- `mx` = 20 —bruk lik blokklengde
- Lengde 2000 ft, tverrsnitt 625×40 ft^2
- $\phi = 0.18$
- Initiell vannmetning $s_{wi} = 0.20$

- $B_{oi} = 1.3$ rb/stb, $c_o = 8.0 \times 10^{-6}$ psi^{-1}
- $B_{wi} = 1.0$ rb/stb, $c_w = 3.0 \times 10^{-6}$ psi^{-1}
- $\mu_o = 5.0$ cp, konstant
- $\mu_w = 0.5$ cp, konstant
- Relative permeabiliteter er gitt av tabell 10.1 i boken til Dake, og en setter $p_c = 0$
- En vannrate på 1000 stb/d injiseres i blokk nr. 20 og produksjonen skjer mot konstant trykk `pconst` $= p_i$ ved utløpet av blokk nr. 1
- Initielt trykk p_i = 2000 psia

Simuler vanndrivet i 25 år, og gjør følgende sammenligninger:

A. Ved tid ett år, plott s_w som funksjon av avstand fra injeksjonsenden beregnet fra:

1. Simuleringsmodellen
2. Buckley-Leverett teorien, se Dake [7], kapittel 10.

Forklar forskjellen mellom i) og ii).

B. Sammenlign

- WOR versus tid i 25 år
- NPC versus tid i 25 år

beregnet med de samme to metodene som under **A**.

10.1 Kommentarer til Øving 10

For strøm av en fase i en dimensjon har vi fra før

$$\frac{\partial}{\partial x}\left(\frac{ckb}{\mu}\frac{\partial p}{\partial x}\right)+q=\phi\frac{\partial b}{\partial t},$$

og utvidet til to faser får en etter en helt tilsvarende utledning

$$\frac{\partial}{\partial x}\left(\frac{ckk_{ro}b_o}{\mu_o}\frac{\partial p_o}{\partial x}\right)+q_o=\phi\frac{\partial}{\partial t}(s_ob_o), \qquad (10.1)$$

for oljeligningen, og for vannligningen

$$\frac{\partial}{\partial x}\left(\frac{ckk_{rw}b_w}{\mu_w}\frac{\partial p_w}{\partial x}\right)+q_w=\phi\frac{\partial}{\partial t}(s_wb_w). \qquad (10.2)$$

I tillegg har en definisjonen av kapillartrykk, $p_w = p_o - p_c$, og at summen av metningene må være lik en: $s_w + s_o = 1$. I disse ligningene er b_o, μ_o funksjoner kun av p_o, og b_w, μ_w kun av p_w, mens k_{ro}, k_{rw}, p_c kun er funksjoner av s_w.

Det fins mange måter å formulere ligningssettet på i numerisk forstand:

A Impes metoden

B Fullstendig implisitt formulering (Newton)

C Kordemetoden eller kvasi-Newton

Her skal vi kort beskrive metode A og B, og gå i detalj og bruke metode C. I det følgende lar vi superskript k betegne iterasjonsnivå, og n tidsnivå, slik at modellen går fra tidsnivå n til $n+1$ ved hjelp av flere iterasjoner.

La oss først formulere ut høyre side av ligningen:

$$\begin{aligned}\phi\frac{\partial}{\partial t}(s_ob_o) &= \frac{\phi}{\Delta t}\left((s_ob_o)^{n+1}-(s_ob_o)^n\right)\\ &= \frac{\phi}{\Delta t}\left(\overline{b_o}\cdot\Delta s_o+\overline{s_o}\cdot\Delta b_o\right)\\ &= \frac{\phi}{\Delta t}\left(\overline{b_o}\cdot\Delta s_o+\overline{s_o}\cdot\frac{\partial b_o}{\partial p_o}\Delta p_o\right),\end{aligned}$$

hvor

$$\begin{aligned}\overline{b_o} &= (b_o^{n+1}+b_o^n)/2\\ \overline{s_o} &= (s_o^{n+1}+s_o^n)/2\\ \Delta s_o &= s_o^{n+1}-s_o^n\\ \Delta p_o &= p_o^{n+1}-p_o^n.\end{aligned}$$

Tilsvarende finner en for vannligningen:

$$\phi\frac{\partial}{\partial t}(s_wb_w)=\frac{\phi}{\Delta t}\left(\overline{b_w}\cdot\Delta s_w+\overline{s_w}\cdot\frac{\partial b_w}{\partial p_w}\Delta p_w\right).$$

10.2 IMPES-metoden

Betegnelsen *IMPES* er et akronym for *Implicit Pressure Explicit Saturation*. De metningsavhengige størrelsene, k_{ro}, k_{rw}, p_c, holdes fast på tidsnivå n, mens en løser trykket implisitt for tidsnivå $n+1$. Siden $\Delta s_w = -\Delta s_o$, elimineres Δs_o på høyre side av ligningen og en står igjen med en ligning i trykk. Denne ligningen er fremdeles ulinær siden b, μ avhenger av trykk, og $\overline{s_o}$ inngår. Grovt sett løses den i følgende trinn:

1. Foreta en implisitt løsning av trykkene
2. Beregn Δs_o fra ligning 10.1 eller 10.2
3. Oppdater koeffisienter og $b, \mu, \overline{s_o}, \overline{s_w}$ etc.
4. Start på punkt 1. igjen

Denne metode fungerer godt dersom det ikke er for "store" metningsendringer per tidssteg. Den blir derfor ofte brukt i full felt modeller hvor de numeriske blokkene er store. Trykket løses ved eliminasjon eller LSOR.

10.3 Oppstrøms relative permeabiliteter

I strømningsleddet på venstre side av oljeligningen 10.1 inngår leddet $k_{ro}b_o/\mu_o$. For strøm av en fase, hvor $k_{ro} = 1$, har vi tidligere brukt middelverdi av b_o/μ_o både i tid og avstand. Dette kan ikke gjøres uten videre for k_{ro}, som eksemplifisert: Figuren

Figur 10.1: Illustrasjon av oppstrøms relative permeabiliteter

illustrerer et vanndriv fra venstre mot høyre. I blokk i er oljemetningen s_{or} og $k_{ro} = 0$. I blokk ip er $s_o > s_{or}$ og $k_{ro} > 0$. Dersom k_{ro} midles arimetisk mellom de to blokkene, så vil transmissibiliteten til olje bli større enn null. Olje vil strømme fra blokk i til blokk ip og oljemetningen i blokk i vil bli redusert til en verdi under s_{or}. Dette er ufysikalsk og det foreslåtte en rekke metoder i litteraturen for å bøte på dette:

- Harmonisk midling
- Topunkts oppstrøms relative permeabiliteter
- Oppstrøms relative permeabiliteter

Vi skal bruk oppstrøms relative permeabiliteter, som vel er den vanligste formuleringen. For hvert tidssteg, ev. etter første iterasjon, sjekkes hvilken retning oljestrømmen har, fra $i \to ip$ eller $ip \to i$. I transmissibiliteten mellom de to blokkene velges så oppstrøms verdi av k_{ro}.

10.4 Fullstendig implisitt formulering

Vi ser på strømningsleddet

$$\frac{\partial}{\partial x}\left(\frac{ckk_{ro}b_o}{\mu_o}\frac{\partial p_o}{\partial x}\right)$$

mellom blokk i og ip og skriver dette på formen $\mathtt{ckx_{ip}} \cdot F(p_{oi}, p_{oip}, s_{oi}, s_{oip})$, hvor $\mathtt{ckx_{ip}}$, som før, inneholder de konstante deler av transmissibiliteten mellom ip og i, og

$$F(p_{oi}, p_{oip}, s_{oi}, s_{oip}) = \frac{k_{ro}b_o}{\mu_o}(p_{oip} - p_{oi}).$$

Vi bruker Newton's metode på F, eller sagt på en annen måte, vi rekkeutvikler F til første orden i alle variable:

$$F^{k+1} \simeq F^k + \left(\frac{\partial F}{\partial p_{oi}}\right)^k \Delta p_{oi}^{k+1} + \left(\frac{\partial F}{\partial p_{oip}}\right)^k \Delta p_{oip}^{k+1} + \left(\frac{\partial F}{\partial s_{oi}}\right)^k \Delta s_{oi}^{k+1} + \left(\frac{\partial F}{\partial s_{oip}}\right)^k \Delta s_{oip}^{k+1}.$$

Her må en generelt ta med både Δs_{oi} og Δs_{oip} siden en ikke vet hvilken vei strømmen går. Ved oppstrøms relative permeabiliteter er enten $(\partial F/\partial s_{oi})^k$ eller $(\partial F/\partial s_{oip})^k$ lik null. Strømningsleddet i vannligningen formuleres tilsvarende. Høyre sidene av ligningen uttrykkes også i $\Delta p_{oi}^{k+1}, \Delta s_{oi}^{k+1}$. En får da to ligninger med to ukjente, Δp_o^{k+1} og Δs_o^{k+1}, som det løses for ved eliminasjon eller med (L)SOR. Merk at $\Delta p_{oi}^{k+1}, \Delta s_{oi}^{k+1}$ er endringer per Newton iterasjon —ikke endringer over tidssteget.

10.5 Kordemetoden – kvasi-Newton

Metoden kan illustreres ved å formulere uttrykket for k_{ro}:

$$k_{ro}^{n+1} \simeq k_{ro}^n + \frac{k_{ro}^{n+1,k} - k_{ro}^n}{s_o^{n+1,k} - s_o^n}\Delta s_o^{n+1,k+1}. \qquad (10.3)$$

Her er

$k_{ro}^{n+1,k}$ verdien av k_{ro} etter iterasjon k,

$s_o^{n+1,k}$ verdien av s_o^{n+1} etter iterasjon k,

$\Delta s_o^{n+1,k+1} = s_o^{n+1,k+1} - s_o^n$.

Når løsning har konvergert er $s_o^{n+1,k+1} = s_o^{n+1}$, $k_{ro}^{n+1,k} = k_{ro}^{n+1}$, $\Delta s_o^{n+1,k+1} = \Delta s_o = s_o^{n+1} - s_o$, og ligning 10.3 er eksakt oppfylt.

De trykkavhengige størrelsene, b_o/μ_o, b_w/μ_w, er kun svakt avhengige av trykket. Det er derfor tilstrekkelig å behandle dem som før, dvs. bruke tids- og avstandsmiddel.

Vi ser videre på strømningsleddene mellom blokk i og ip i oljeligningen:

$$\frac{\partial}{\partial x}\left(\frac{ckk_{ro}b_o}{\mu_o}\frac{\partial p_o}{\partial x}\right) \rightarrow \mathtt{ckx_{ip}} \cdot \overline{\overline{\left(\frac{\mathtt{b_o}}{\mu_o}\right)}}_{\mathtt{i,ip}} \cdot \mathtt{k_{ro}} \cdot (\mathtt{p_{oip}} - \mathtt{p_{oi}}),$$

hvor $(\overline{\overline{\mathtt{b_o}/\mu_o}})_{\mathtt{i,ip}}$ er tids- og avstandsmiddel. La oss definere $\mathtt{bovo_{ip}} \equiv (\overline{\overline{\mathtt{b_o}/\mu_o}})_{\mathtt{i,ip}}$. Videre kan vi sette:

$$\begin{aligned} k_{ro} &= \mathtt{fdo_i} * \mathtt{krold_i} + \mathtt{fdo_i} * \mathtt{dkods_i} * \mathtt{dso_i} \\ &\quad + (1 - \mathtt{fdo_i}) * \mathtt{krold_{ip}} + (1 - \mathtt{fdo_i}) * \mathtt{dkods_{ip}} * \mathtt{dso_{ip}}, \end{aligned}$$

hvor

$$\begin{aligned} \mathtt{fdo_i} &= 1.0 \text{ dersom oljestrømmen går fra } i \rightarrow ip \\ &= 0.0 \text{ dersom oljestrømmen går fra } ip \rightarrow i \\ \mathtt{krold_i} &= k_{roi}^n \\ \mathtt{dkods_i} &= (k_{roi}^{n+1,k} - k_{roi}^n)/(s_{oi}^{n+1,k} - s_{oi}^n) \\ \mathtt{dso_i} &= s_{oi}^{n+1,k+1} - s_{oi}^n. \end{aligned}$$

Vi minner også om at $\mathtt{ckx_{ip}} = \mathtt{ckx_i} \cdot \Delta\mathtt{x_i}/\Delta\mathtt{x_{ip}}$.

Videre definerer vi for strømningsledd mellom blokk i og ip:

$$\begin{aligned} \mathtt{oxldp_i} &= \mathtt{bovo_{ip}} * (\mathtt{fdo_i} * \mathtt{krold_i} + (1. - \mathtt{fdo_i}) * \mathtt{krold_{ip}}) \\ \mathtt{doxpp_i} &= \mathtt{bovo_{ip}} * (1. - \mathtt{fdo_i}) * \mathtt{dkods_{ip}} \\ \mathtt{doxpm_i} &= \mathtt{bovo_{ip}} * \mathtt{fdo_i} * \mathtt{dkods_i} \\ \mathtt{pop_i} &= \mathtt{po_{ip}^k} - \mathtt{po_i^k} \end{aligned}$$

For strøm mellom blokk im og i definerer vi:

$$\begin{aligned} \mathtt{oxld_i} &= \mathtt{bovo_i} * (\mathtt{fdo_{im}} * \mathtt{krold_{im}} + (1. - \mathtt{fdo_{im}}) * \mathtt{krold_i}) \\ \mathtt{doxmm_i} &= \mathtt{bovo_i} * \mathtt{fdo_{im}} * \mathtt{dkods_{im}} \\ \mathtt{doxmp_i} &= \mathtt{bovo_i} * (1. - \mathtt{fdo_{im}}) * \mathtt{dkods_i} \\ \mathtt{pom_i} &= \mathtt{po_{ip}^k} - \mathtt{po_i^k}. \end{aligned}$$

Siden

$$\begin{aligned} \mathtt{oxldp_i} &= \mathtt{oxld_{ip}} * \mathtt{dexm} \\ \mathtt{doxpp_i} &= \mathtt{doxmp_{ip}} * \mathtt{dexm} \\ \mathtt{doxpm_i} &= \mathtt{doxmm_{ip}} * \mathtt{dexm} \end{aligned}$$

trenger en bare bruke ett sett av disse størrelsene.

Vannligningen formuleres helt tilsvarende, bortsett fra et ekstra ledd som inneholder kapillartrykket p_c:

$$\frac{\partial}{\partial x}\left(\frac{ckk_{rw}b_w}{\mu_w}\frac{\partial p_w}{\partial x}\right) = \frac{\partial}{\partial x}\left(\frac{ckk_{rw}b_w}{\mu_w}\frac{\partial p_o}{\partial x}\right) + \frac{\partial}{\partial x}\left(\frac{ckk_{rw}b_w}{\mu_w}\frac{\partial p_c}{\partial x}\right).$$

Det første leddet på høyre side formuleres på samme måte som for oljeligningen. Det er ledd nummer to, kapillarleddet, som volder noe problemer.

Vi definerer størrelsene `wxldm, dwxmp, dwxmm` på samme måte som for oljeligningen. I kapillarleddet har vi et produkt av to metningsavhengige størrelser, nemlig k_{rw} og p_c. Dette produktet tilnærmer vi på følgende måte:

$$\begin{aligned} k_{rw}^{n+1} &\simeq k_{rw}^n \cdot p_c^n + k_{rw}^n \cdot \Delta p_c + p_c^n \cdot \Delta k_{rw} \\ &\simeq k_{rw}^n \cdot p_c^n + k_{rw}^n \cdot \frac{\partial p_c}{\partial s_o}\Delta s_o + p_c^n \cdot \frac{\partial k_{rw}}{\partial s_o}\Delta s_o \end{aligned}$$

Her har en altså neglisjert andre ordens ledd, også kalt kryssledd, av typen $\Delta s_o * \Delta s_o$. Vi definerer $\partial p_c/\partial s_o = \mathtt{dpcds}$, $\partial k_{rw}/\partial s_o = \mathtt{dkwds}$ og er nå istand til å skrive ut kapillarleddet i detalj:

$$\frac{\partial}{\partial x}\left(\frac{ckk_{rw}b_w}{\mu_w}\frac{\partial p_c}{\partial x}\right) \rightarrow \mathtt{wx}^+ * \Delta\mathtt{p}_\mathtt{c}^+ + \mathtt{wx}^- * \Delta\mathtt{p}_\mathtt{c}^-,$$

og

$$\begin{aligned} \mathtt{wx}^+ \cdot \Delta\mathtt{p}_\mathtt{c}^+ &= (\mathtt{wxldm}_\mathtt{ip} * \mathtt{dexm} + \mathtt{dwxmp}_\mathtt{ip} * \mathtt{dexm} * \Delta\mathtt{s}_\mathtt{oip} + \mathtt{dwxmm}_\mathtt{ip} * \mathtt{dexm} * \Delta\mathtt{s}_\mathtt{oi}) \\ &= *(\mathtt{pcld}_\mathtt{ip} + \mathtt{dpcds}_\mathtt{ip} * \Delta\mathtt{s}_\mathtt{oip} - \mathtt{pcld}_\mathtt{i} - \mathtt{dpcds}_\mathtt{i} * \Delta\mathtt{s}_\mathtt{oi}) \end{aligned}$$

Disse leddene multipliseres så ut, og en neglisjerer kryssledd. Dette gir

$$\begin{aligned} \mathtt{wx}^+ * \Delta\mathtt{p}_\mathtt{c}^+ \rightarrow\ & \mathtt{wxldm}_\mathtt{ip} * \mathtt{dexm} * \mathtt{pcld}_\mathtt{ip} + \mathtt{dwxmp}_\mathtt{ip} * \mathtt{dexm} * \mathtt{pcld}_\mathtt{ip} * \Delta\mathtt{s}_\mathtt{oip} \\ & + \mathtt{dwxmm}_\mathtt{ip} * \mathtt{dexm} * \mathtt{pcld}_\mathtt{ip} * \Delta\mathtt{s}_\mathtt{oi} + \mathtt{wxldm}_\mathtt{ip} * \mathtt{dexm} * \mathtt{dpcds}_\mathtt{ip} * \Delta\mathtt{s}_\mathtt{oip} \\ & + \text{tilsvarende ledd ved å bruke } i\text{-leddene} \end{aligned}$$

På høyre side av vannligningen har en størrelsen Δb_w som en tilnærmer med

$$\Delta b_w = \frac{\partial b_w}{\partial p_w} \cdot \Delta p_w \rightarrow \frac{\partial b_w}{\partial p_w}(\mathtt{po} - \mathtt{pold} - \mathtt{dpcds} * \Delta\mathtt{s}_\mathtt{o}).$$

10.6 Vannligningen i detalj — oppsummering

Fullstendig diskretisering av vannligningen 10.2 gir:

$$\mathtt{wx}_\mathtt{i}^- * \Delta\mathtt{p}_\mathtt{wi}^- + \mathtt{wx}_\mathtt{i}^+ * \Delta\mathtt{p}_\mathtt{wi}^+ + \mathtt{q}_\mathtt{wi} = \phi/\Delta\mathtt{t}(\overline{\mathtt{b}}_\mathtt{wi} * \Delta\mathtt{s}_\mathtt{wi} + \overline{\mathtt{s}}_\mathtt{wi} * \Delta\mathtt{b}_\mathtt{wi}),$$

med følgende definisjoner:

$$
\begin{aligned}
\mathtt{wx}_i^- &= \mathtt{wxldm}_i + \mathtt{dwxmm}_i * \Delta\mathtt{s}_{oim} + \mathtt{dwxmp}_i * \Delta\mathtt{s}_{oi} \\
\mathtt{wx}_i^+ &= (\mathtt{wxldm}_{ip} + \mathtt{dwxmm}_{ip} * \Delta\mathtt{s}_{oi} + \mathtt{dwxmp}_{ip} * \Delta\mathtt{s}_{oip}) * \mathtt{dexm} \\
\Delta\mathtt{p}_{wi}^- &= \mathtt{p}_{wim} - \mathtt{p}_{wi} = \mathtt{p}_{oim} - \mathtt{p}_{oi} - \mathtt{p}_{cim} + \mathtt{p}_{ci} \\
\Delta\mathtt{p}_{wi}^+ &= \mathtt{p}_{wip} - \mathtt{p}_{wi} = \mathtt{p}_{oip} - \mathtt{p}_{oi} - \mathtt{p}_{cip} + \mathtt{p}_{ci} \\
\mathtt{pom} &= \mathtt{p}_{oim} - \mathtt{p}_{oi} \\
\mathtt{pop} &= \mathtt{p}_{oip} - \mathtt{p}_{oi}
\end{aligned}
$$

Vi velger som sagt å bruke $\mathtt{p}_o$ og $\Delta\mathtt{s}_o = \mathtt{s}_o^{n+1} - \mathtt{s}_o^n$ som ukjente. Leddet $\mathtt{wx}_i^- \cdot \Delta\mathtt{p}_{wi}^-$ blir da

$$
\begin{array}{llll}
\mathtt{wxldm}_i * \Delta\mathtt{p}_o^- & -\mathtt{wxldm}_i * \mathtt{pcld}_{im} & +\mathtt{dwxmm}_i * \mathtt{pom} * \Delta\mathtt{s}_{oim} & +\mathtt{dwxmp}_i * \mathtt{pom} * \Delta\mathtt{s}_{oi} \\
 & +\mathtt{wxldm}_i * \mathtt{pcld}_{im} & -\mathtt{wxld}_i * \mathtt{dpcds}_{im} * \Delta\mathtt{s}_{oim} & -\mathtt{dwxmp}_i * \mathtt{pcld}_{im} * \Delta\mathtt{s}_{oi} \\
 & & -\mathtt{dwxmm}_i * \mathtt{pcld}_{im} * \Delta\mathtt{s}_{oim} & +\mathtt{wxldm}_i * \mathtt{dpcds}_i * \Delta\mathtt{s}_{oi} \\
 & & +\mathtt{wxld}_i * \mathtt{pcld}_i * \Delta\mathtt{s}_{oim} & +\mathtt{dwxmp}_i * \mathtt{pcld}_i * \Delta\mathtt{s}_{oi}
\end{array}
$$

Vi ønsker å skrive ligningen på formen

$$\mathtt{apow} * \mathtt{po}_{im} + \mathtt{asow} * \mathtt{dso}_{im} + \mathtt{bpow} * \mathtt{po}_i + \mathtt{bsow} * \mathtt{dso}_i + \mathtt{cpow} * \mathtt{po}_{ip} + \mathtt{csow} * \mathtt{dso}_{ip} + \mathtt{dw} = 0.$$

Med

$$
\begin{aligned}
\mathtt{pwm} &= \mathtt{pom} + \mathtt{pcld}_i - \mathtt{pcld}_{im} \\
\mathtt{pwp} &= \mathtt{pop} + \mathtt{pcld}_i - \mathtt{pcld}_{im}
\end{aligned}
$$

får en

$$
\begin{aligned}
\mathtt{apow} &= \mathtt{wxldm}_i \\
\mathtt{asow} &= \mathtt{dwxmm}_i * \mathtt{pwm} - \mathtt{wxldm}_i * \mathtt{dpcds}_{im} \\
\mathtt{bpow} &= -(\mathtt{wxmld}_i + \mathtt{wxldm}_{ip} * \mathtt{dexm}) - \phi/\Delta\mathtt{t} * \overline{\mathtt{s}}_{wi} * \mathtt{dbwdp}_i \\
\mathtt{bsow} &= \mathtt{dwxmp}_i * \mathtt{pwm} + \mathtt{dwxmm}_{ip} * \mathtt{dexm} * \mathtt{pwp} + (\mathtt{wxldm}_i + \mathtt{wxldm}_{ip} * \mathtt{dexm}) * \mathtt{dpcds}_i \\
&\quad + \phi/\Delta\mathtt{t} * (\overline{\mathtt{s}}_{wi} * \mathtt{dbwdp}_i * \mathtt{dpcds}_i + \overline{\mathtt{b}}_{wi}) \\
\mathtt{cpow} &= \mathtt{wxldm}_{ip} * \mathtt{dexm} \\
\mathtt{csow} &= (\mathtt{dwxmp}_{ip} * \mathtt{pwp} - \mathtt{wxldm}_{ip} * \mathtt{dpcds}_{ip}) * \mathtt{dexm} \\
\mathtt{dw} &= \mathtt{wxldm}_i * (\mathtt{pcld}_i - \mathtt{pcld}_{im}) + \mathtt{wxldm}_{ip} * \mathtt{dexm} * (\mathtt{pcld}_i - \mathtt{pcld}_{ip}) \\
&\quad \phi/\Delta\mathtt{t} * \overline{\mathtt{s}}_{wi} * \mathtt{dbwdp}_i * \mathtt{pold}_i + \mathtt{q}_{wi}.
\end{aligned}
$$

Oljeligningen blir utformet tilsvarende, og her er det ikke noe kapillartrykksledd

$$
\begin{aligned}
\mathtt{apoo} &= \mathtt{oxldm}_i \\
\mathtt{asoo} &= \mathtt{doxmm}_i * \mathtt{pom} \\
\mathtt{bpoo} &= -(\mathtt{oxmld}_i + \mathtt{oxldm}_{ip} * \mathtt{dexm}) - \phi/\Delta\mathtt{t} * \overline{\mathtt{s}}_{oi} * \mathtt{dbodp}_i \\
\mathtt{cpoo} &= \mathtt{oxldm}_{ip} * \mathtt{dexm} \\
\mathtt{csoo} &= \mathtt{doxmp}_{ip} * \mathtt{pop} * \mathtt{dexm} \\
\mathtt{do} &= \phi_i/\Delta\mathtt{t} * \overline{\mathtt{s}}_{oi} * \mathtt{dbodp}_i * \mathtt{pold}_i + \mathtt{q}_{oi}.
\end{aligned}
$$

Ukjente →		*blokk*1		*blokk*2		*blokk*3		*blokk*4		*blokk*5	
Lign. ↓		$\mathtt{po}_1$	$\Delta\mathtt{s}_{o1}$	$\mathtt{po}_2$	$\Delta\mathtt{s}_{o2}$	$\mathtt{po}_3$	$\Delta\mathtt{s}_{o3}$	$\mathtt{po}_4$	$\Delta\mathtt{s}_{o4}$	$\cdots$	$\cdots$
w	1	bpow	bsow	cpow	csow						
o	2	bpoo	bsoo	cpoo	csoo	$\ddots$					
w	3	apow	asow	bpow	bsow	cpow	csow				
o	4	apoo	asoo	bpoo	bsoo	cpoo	csoo	$\ddots$			
w	5		$\ddots$	apow	asow	bpow	bsow	cpow	csow		
o	6			apoo	asoo	bpoo	bsoo	cpoo	csoo	$\ddots$	
$\vdots$	$\vdots$				$\ddots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$

10.7 Ligningssystem

For et enfasesystem har vi sett at ligningssystemet har en tridiagonal koeffisientmatrise. For to faser har hver blokk to ukjente. Det som i enfasetilfellet var en (skalar) koeffisient blir nå en 2×2 matrise, som illustrert i følgende skjema:

Ukjente →		*blokk*1		*blokk*2		*blokk*3		*blokk*4		*blokk*5	
Lign. ↓		$\mathtt{po}_1$	$\Delta\mathtt{s}_{o1}$	$\mathtt{po}_2$	$\Delta\mathtt{s}_{o2}$	$\mathtt{po}_3$	$\Delta\mathtt{s}_{o3}$	$\mathtt{po}_4$	$\Delta\mathtt{s}_{o4}$	$\cdots$	$\cdots$
w	1	bpow	bsow	cpow	csow						
o	2	bpoo	bsoo	cpoo	csoo	$\ddots$					
w	3	apow	asow	bpow	bsow	cpow	csow				
o	4	apoo	asoo	bpoo	bsoo	cpoo	csoo	$\ddots$			
w	5		$\ddots$	apow	asow	bpow	bsow	cpow	csow		
o	6			apoo	asoo	bpoo	bsoo	cpoo	csoo	$\ddots$	
$\vdots$	$\vdots$				$\ddots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$

Koeffisientmatrisen blir altså en båndmatrise med båndbredde lik 7. Ligningssystemet kan derfor løses ved den tidligere omtalte subrutinen SOLVE, eller en annen matriseløser, [1].

10.8 Rateformulering

I tillegg til å bruke konstante rater ønsker en også å produsere mot konstant trykk, `pconst`, ved utløpet av blokk nr. 1.:

$$\mathtt{qo}_1 = \mathtt{bovo}_1 * \mathtt{kro}_1 * (\mathtt{pconst} - \mathtt{po}_1)$$
$$\mathtt{qw}_1 = \mathtt{bwbw}_1 * \mathtt{krw}_1 * (\mathtt{pconst} - \mathtt{po}_1 + \mathtt{pc}_1).$$

Vi regner her at trykket i brønnen, `pconst`, er det samme for olje som for vann.

Uttrykket for oljeraten kan skrives ut på følgende måte, når vi bruker tidligere definerte størrelse:

$$\begin{aligned}\mathtt{qo}_1 = {} & \mathtt{bovo}_1 * \mathtt{krold}_1 * \mathtt{pconst} + \mathtt{bovo}_1 * \mathtt{dkods}_1 * \mathtt{pconst} * \mathtt{dso}_1 \\ & - \mathtt{bovo}_1 * \mathtt{krold}_1 * \mathtt{po}_1 - \mathtt{bovo}_1 * \mathtt{dkods}_1 * \mathtt{po}_1 * \mathtt{dso}_1 .\end{aligned}$$

Dette kan vi videre skrive som

$$\mathtt{qo}_1 = \mathtt{qo}_1 + \mathtt{bqop} * \mathtt{po}_1 + \mathtt{bqos} * \mathtt{dso}_1, \qquad \ldots\ldots\ldots\ldots\ldots\ldots \quad (10.4)$$

hvor

$$\begin{aligned}\mathtt{qo}_1 &= \mathtt{bovo}_1 * \mathtt{krold}_1 * \mathtt{pconst} \\ \mathtt{bqop} &= -\mathtt{bovo}_1 * \mathtt{krold}_1 \\ \mathtt{bqos} &= \mathtt{bovo}_1 * \mathtt{dkods}_1 * (\mathtt{pconst} - \mathtt{po}_1),\end{aligned}$$

hvor $\mathtt{po}_1$ er den sist iterert verdien.

Tilsvarende får en for vannraten:

$$\mathtt{qw}_1 = \mathtt{qw}_1 + \mathtt{bqwp} * \mathtt{po}_1 + \mathtt{bqws} * \mathtt{dso}_1, \qquad \ldots\ldots\ldots\ldots\ldots\ldots \quad (10.5)$$

hvor

$$\begin{aligned}\mathtt{qw}_1 &= \mathtt{bwvw}_1 * \mathtt{krwld}_1 * (\mathtt{pconst} + \mathtt{pcld}_1) \\ \mathtt{bqwp} &= -\mathtt{bwvw}_1 * \mathtt{krwld}_1 \\ \mathtt{bqws} &= \mathtt{bwvw}_1 * \mathtt{dkwds}_1 * (\mathtt{pconst} - \mathtt{po}_1) + \mathtt{bwvw}_1 * (\mathtt{krwld}_1 * \mathtt{dpcds}_1 + \mathtt{dkwds}_1 * \mathtt{pcld}_1).\end{aligned}$$

Uttrykkene for $\mathtt{qo}_1$ og $\mathtt{qw}_1$ settes inn for $\mathtt{qo}$ og $\mathtt{qw}$ i ligningene og fordeles på de riktige matriseelementene på følgende måte:

$$\begin{aligned}\mathtt{bqop} &\rightarrow \mathtt{bpoo} \\ \mathtt{bqwp} &\rightarrow \mathtt{bpow} \\ \mathtt{bqos} &\rightarrow \mathtt{bsoo} \\ \mathtt{bqws} &\rightarrow \mathtt{bsow},\end{aligned}$$

hvor $\rightarrow$ betyr 'settes inn i'. Etter at løsningen har konvergert beregnes ratene for blokk 1 fra ligningene 10.4 og 10.5 før materialbalansen beregnes. Dersom $\mathtt{qo}_1$ eller $\mathtt{qw}_1$ blir positive, settes de lik 0. Vannraten $\mathtt{qw}_1$ kan fysikalsk sett godt bli naturlig negativ, på grunn av endeeffekt.

10.9 Noen hint til programmering

- Les inn PVT-tabell og omgjør til ekvidistant, intern tabell på 81 linjer

TP	TBP	TVO	TBW	TVW
-	-	-	-	-
-	-	-	-	-

- Les inn bergartstabell og omgjør til ekvidistant, intern tabell på 51 linjer:

TSO	TKRO	TKRW	TPC
-	-	-	-
-	-	-	-

- Les inn initiell metningsfordeling
- Beregn OOIP og OWIP
- Rett etter start av tidsstegssløfe uføres følgende:

$$\begin{aligned}
\texttt{pold} &= \texttt{po} \\
\texttt{bold} &= \texttt{bo} \\
\texttt{vold} &= \texttt{vo} \\
\texttt{bwld} &= \texttt{bw} \\
\texttt{vwld} &= \texttt{vw} \\
\texttt{pcld} &= \texttt{pc} \\
\texttt{krwld} &= \texttt{krw} \\
\texttt{krold} &= \texttt{kro} \\
\texttt{sold} &= \texttt{so}
\end{aligned}$$

- Tidssteget kontrolleres nå både av maksimum metningsendring og maksimum trykkendring:

$$\begin{aligned}
\texttt{dpmax} &= 5.0\text{psi} \\
\texttt{dsmax} &= 0.05
\end{aligned}$$

- Konvergenssjekk utføres på både `po` og `dso`.
- Materialbalanse utføres både for vann og olje
- `SUBROUTINE FLPROP`

$$\begin{aligned}
&\texttt{bo}, \texttt{vo}, \texttt{dbodp} \\
&\texttt{bw}, \texttt{vw}, \texttt{dbwdp}
\end{aligned}$$

- `SUBROUTINE SAT`

$$\begin{aligned}
&\texttt{kro}, \texttt{dkods} \\
&\texttt{krw}, \texttt{dkwds} \\
&\texttt{pc}, \texttt{dpcds}
\end{aligned}$$

- `SUBROUTINE FLODIR`

$$\begin{aligned}
\texttt{fdo}_{\texttt{i}} &= 1.0 \text{ dersom } \texttt{po}_{\texttt{i}} > \texttt{po}_{\texttt{ip}} \\
&= 0. \text{ dersom } \texttt{po}_{\texttt{ip}} < \texttt{po}_{\texttt{i}} \\
\texttt{fdw}_{\texttt{i}} &= 1.0 \text{ dersom } \texttt{pw}_{\texttt{i}} > \texttt{pw}_{\texttt{ip}} \\
&= 0. \text{ dersom } \texttt{pw}_{\texttt{ip}} < \texttt{pw}_{\texttt{i}}
\end{aligned}$$

Denne subrutinen kalles opp <u>en</u> gang etter 1. iterasjon i hvert tidssteg.

- `SUBROUTINE TRANS`

$$\begin{array}{c} \texttt{oxldm}, \texttt{wxldm} \\ \texttt{doxmp}, \texttt{dwxmp} \\ \texttt{doxmm}, \texttt{dwxmm} \end{array}$$

- `SUBROUTINE RAT`
  Dersom `pconst > 0.` beregnes følgende størrelser:

$$\begin{array}{c} \texttt{qo}, \texttt{qw} \\ \texttt{bqop}, \texttt{bqwp} \\ \texttt{bqos}, \texttt{bqws} \end{array}$$

- `SUBROUTINE FLOCON`

$$\begin{array}{cc} \texttt{apow} & \texttt{apoo} \\ \texttt{asow} & \texttt{asoo} \\ \vdots & \vdots \\ \texttt{bb(ic,1)} & = \texttt{apoo} \\ \texttt{bb(ic,2)} & = \texttt{asoo} \\ \vdots & \vdots \\ \texttt{cc(ic)} & = -\texttt{do} \\ \vdots & \vdots \\ \texttt{call solve} & \end{array}$$

  Løsningen returneres i `cc`-matrisen. Deretter settes `so(i) = sold(i) + dso(i).`

- Sjekk for konvergens
- Når løsningen har konvergert beregnes `qo(1)` og `qw(1)` dersom `pconst > 0.`

10.10 Andre kommentarer

- Dersom alle deriverte med hensyn på metning settes lik null, altså at `dkods, dkwds, dpcdsså` er lik null, fås en IMPES formulering av ligningene. Da kan Δs_{oi} elimineres og en får kun en ligning per numerisk blokk.
- Dersom en legger inn helning og gravitasjonsledd `gho, ghw`, må reservoarets ekvilibreres før start av simuleringen.
  - Definer `woc` der hvor $\texttt{p}_\texttt{c} = 0$.
  - Beregn p_o og p_w som funksjon av høyden

  - $p_c = p_o - p_w$
  - Med denne verdien av p_c går en inn i tabellen og finner s_o.
  - Iterer til likevekt

- Noen modeller bruker sekvensiell løsning. Da løser en først for p_o med `TRIDIA`, deretter for Δs_o med `TRIDIA` og iterer

- På grunn av blokkinndelingen vil modellen vise numerisk dispersjon. Vannfronten vil bli utsmurt. Dette kan rettes på med en frontfølgeteknikk

Øving 11

Måling av relative permeabiliteter

11.1 Innledning

Hensikten med denne oppgaven er å bruke modellen utviklet i Øving 10 til å studere effekt av kapillartrykk, endeeffekt og rate på relative permeabilitetskurver slik at disse kan bestemmes fra et fortrengingseksperiment.

Det tas utgangspunkt i de to artiklene [5, 4]. Welge's metode blir ofte brukt til å bestemme k_{ro}, k_{rw} fra et fortrengingseksperiment etter ligningene 8a, 4a, 9 i [5]. [Merk at dersom $k_{ro}(s_{iw}) < 1.0$, må ligning 7b, og dermed 8a i [5] justeres.

Teorien forutsetter at $p_c = 0$. Dersom $p_c > 0$, vil en få en endeeffekt ved utløpet av plutggen, en opphoping av vann som influerer på relative permeabilitetskurvene. Denne endeeffekten kan reduseres ved å foreta eksperimentene ved høy nok rate til at dynamiske trykkgradienter dominerer over kapillartrykksgradienter.

Kyte og Rapoport [6] har vist at dette er oppfylt dersom $L\mu q/A > 1-5$, hvor L er lengden av pluggen i cm, μ er viskositeten til fortrengende fase i cp, q/A er strømningsrate per tverrsnittsareal med benevningen cm/min.

11.2 Framgangsmåte

1. Simuler laboratorieeksperimentet som er angitt med data under.
   - Bruk 20 blokker, produksjon mot konstant trykk lik initielt trykk, og $p_c = 0$.
   - Tolk simuleringen etter Welge's metode og beregn k_{ro}, k_{rw} og sammenlign med innleste kurveverdier.
2. Gjenta pkt. 1 med en lineær p_c-kurve som går fra 10 psi ved s_{iw} til 0 psi ved s_{or}.
3. Repeter pkt. 2 med raten øket med en faktor 5.

En bør merke seg følgende to punkter:

- Med den rateformuleringen som er foreslått i Øving 10, vil $p_o = p_w =$ `pconst` ved utløpet av pluggen, slik at en får kapillar endeeffekt med $p_c > 0$.
- Numerisk dispersjon vil også ha innflytelse på resultatene.

11.3 Data

Lengde av kjerneprøve, cm	21.3
Tversnitt av kjerneprøve, cm^2	11.3
Porøsitet, fraksjon	0.47
Initiell, irredusibel vannmetning, fraksjon	0.20
Oljeviskositet, cp	6.8
Vannviskositet, cp	0.52
Initielt trykk, psia	3100
Injeksjonsrate av vann, reservoar forhold, cm^3/min	0.438
Effektiv oljepermeabilitet ved s_{iw}, md	46.9
Trykkdifferanse Δp_s ved $s_{iw} = 0.20$, psi	29.32

Dette er data hentet fra et virkelig eksperiment. I simuleringen skal en i tillegg bruke

$b_{oi} = 1.0$ rb/stb $\quad c_o = 8.0 \times 10^{-6}$ psi^{-1}
$b_{wi} = 1.0$ rb/stb $\quad c_w = 3.0 \times 10^{-6}$ psi^{-1} sammen med relative permeabilitetskurver som i øving 10.1 i boken til Dake [7].

11.4 Forventede resultater

Trinn 1. i *Framgangsmåte* på side 110 vil antagelig gi relative permeabilitetskurver som ligger nær de innleste. En vil få noe avvik på grunn av numerisk dispersjon — vannet vil bre seg for raskt mot enden av pluggen.

Trinn 2. vil gi avvik på grunn av kapillar endeeffekt, siden $L\mu q/A = 0.43$.

Trinn 3 vil gi bedre samsvar igjen, siden $L\mu q/A = 2.15$. Dessuten, på grunn av endeeffekten, vil vannet bli holdt noe igjen ved utløpet fortsatt. Dette motvirker det tidlige vanngjennombrudd fra numerisk dispersjon. Effekten av denne dispersjonstypen på beregnede relative permeabilitetskurver vil da være redusert.

Del IV

Eksamensoppgaver

HØGSKOLEN I STAVANGER

AVDELING FOR TEKNISK - NATURVITENSKAPELIGE FAG

DATO: 19. DESEMBER 1995

EKSAMEN I: TE 192 Reservoarsimulering, innføring

VARIGHET: kl 09.00 – 14.00

TILLATTE HJELPEMIDLER: Kalkulator

OPPGAVESETTET BESTÅR AV: 3 sider

MERKNADER: Ingen

Oppgave 1

a) Gitt bevaringsligningen

$$\frac{\partial}{\partial x}(\rho u) - q + \frac{\partial}{\partial t}(\phi\rho) = 0, \qquad (1)$$

hvor ρ er tettheten av fluidet som strømmer. Hva er det som bevares, hva står symbolene u og q for og hvilke enheter uttrykkes q i?

b) Kompressibiliteten c_f til et fluid er definert ved

$$c_f = -\frac{1}{V_f}\left(\frac{\partial V_f}{\partial p}\right)_T, \qquad (2)$$

hvor V_f er fluidvolum og T er temperatur. Vis at dersom c_f er konstant så er

$$\rho = \rho_r \exp[c_f(p - p_r)], \qquad (3)$$

hvor p_r er et referansetrykk, f.eks. intielt trykk, og ρ_r er den tilhørende tetthet.

c) Anta at permeabilitet, porøsitet, viskositet og kompressibilitet av fluid er konstant. Vis at (1) da kan skrives som

$$\frac{Ck}{\mu c_f}\frac{\partial^2\rho}{\partial x^2} + q = \phi\frac{\partial\rho}{\partial t}, \qquad (4)$$

hvor C er en omregningsfaktor som avhenger av hvilket sett av enheter som er valgt. [Hint: Sett inn i (1) for u fra Darcy's lov og deriver (3) m.h.p. x.]

d) I (3) vil vanligvis $c_f(p - p_r) \ll 1$ for væsker. Vis ved rekkeutvikling til 2. orden at

$$\frac{\partial^2\rho}{\partial x^2} \approx c_f\rho_r(1 + c_f(p - p_r))\frac{\partial^2 p}{\partial x^2}$$

dersom

$$\left|\frac{\partial^2 p}{\partial x^2}\right| \gg \frac{c_f\left(\frac{\partial p}{\partial x}\right)^2}{1 + c_f(p - p_r)}.$$

[Oppgitt: $\exp(x) = 1 + x + \frac{x^2}{2!} + \cdots$]

e) Vis nå at (4) kan skrives som

$$\frac{Ckb_r}{\mu}\frac{\partial^2 p}{\partial x^2} + \frac{qb_r}{\rho_r} = \phi c_f b_r \frac{\partial p}{\partial t}, \qquad \ldots \quad (5)$$

hvor b_r er volumfaktoren ved p_r, og forklar hva størrelsen qb_r/ρ_r uttrykker.

Oppgave 2

Diffusivitetsligningen (5) gjelder for strøm i en dimensjon av en væske med liten og konstant kompressibilitet. I denne oppgaven skal det formuleres en numerisk løsning av ligningen.

a) Reservoaret diskretiseres i N like blokker med lengde Δx. Tiden diskretiseres i like lange tidssteg $\Delta t = t_{n+1} - t_n$. Det brukes implisitt formulering.

Utled simuleringsligningen

$$a_i p_{i-1} - p_i + c_i p_{i+1} = -d_i \qquad \ldots \quad (6)$$

hvor

$$\begin{aligned} B &= kT + \phi c_f b_r/\Delta t \\ a_i &= T/B \\ c_i &= T/B \\ d_i &= (\phi c_f b_r p_i(t_n)/\Delta t + \frac{q_i b_r}{\rho_r})/B, \end{aligned}$$

og $T = Ckb_r/(\mu(\Delta x)^2)$; i angir blokknummer; $k = 1$ dersom $i = 1$ eller N, og 2 ellers.

b) Skisser en Fortran subrutine som kan løse (6) ved eliminasjon.

c) Skisser strukturen til et Fortran simuleringsprogram basert på (5). Forklar spesielt om det er nødvendig å iterere.

d) Hvordan kan vi kontrollere nøyaktigheten til den numeriske løsningen?

Oppgave 3

I vestre del av Trollfeltet er det en oljesone som Hydro vil utvinne med horisontale brønner. Det er en tykk gassone over oljen og vannsone under. Når produksjonen starter, vil gass- olje kontakten over en horisontal brønn deformeres ned mot brønnen i form av en langstrakt renne eller dal og olje-vann kontakten vil tilsvarende trekkes opp som en kam (koning). Det er ønskelig å forutsi hvor lang tid det vil ta før gass eller vann bryter inn i brønnen.

Tiden til gjennombrudd i et tilsvarende *uendelig- virkende* reservoar ønskes brukt som en referanse. Tykkelsen på gassonen settes derfor til ca. 3500 ft, vannsonen til ca. 3500 ft og horisontal utstrekning ca. 30 000 ft. Det regnes med at reservoaret da

kan betraktes som uendelig i alle retninger ut fra brønnen i perioden før gjennombrudd. Videre kan det ses bort fra endeeffekter i den horisontale brønnen, som har en lengde på ca. 1500 ft, og interferens fra nabobrønner kan også neglisjeres. Brønnen plasseres forsøksvis midt i oljesonen som er 80 ft tykk.

Ta hensyn til symmetrien i problemet og vurder hvor mange dimensjoner og faser som må brukes i en simuleringsmodell for å kunne estimere tiden til vann- og/eller gassgjennombrudd.

Det kan brukes inntil ca. 2500 numeriske blokker og en kan starte med 1 ft blokker inne ved brønnen. Skisser et fornuftig numerisk rutenett for dette problemet og forklar hvilke kriterier du legger til grunn.

Utkast til løsning

Oppgave 1

a) Det er massen som bevares. Symbol u står for Darcy hastighet (eller volumhastighet, eller 'superficial velocity' i f.eks. m/s; q er kildeledd som uttrykker injeksjon eller produksjon pr. (bulk-) volumenhet av reservoaret og har enheten (f.eks.) kg/(m^{3}s).

b) I reservoaret er som oftest temperaturen konstant, gitt av jordvarmen. Den kan imidlertid variere dersom det f.eks. injiseres kaldt vann. Her, med en fase, antar vi at den er konstant. Da vil V_f kun avhenge av trykket og vi kan skifte fra partiell til ordinær derivasjon.

Ved å føre inn massen m_f til volumet V_f får vi

$$\begin{aligned} c_f &= -\frac{m_f}{V_f}\frac{d\left(\frac{V_f}{m_f}\right)}{dp} \\ &= -\rho\frac{d\left(\frac{1}{\rho}\right)}{dp} \\ &= -\rho\left(-\frac{1}{\rho^2}\frac{d\rho}{dp}\right) \\ &= \frac{1}{\rho}\left(\frac{d\rho}{dp}\right) \end{aligned}$$

Dette er en ordinær differensialligning som kan løses ved

$$\int\frac{d\rho}{\rho} = c_f\int dp + C,$$

$$\ln\rho = c_f p + C,$$

og ved å kreve

$$\ln\rho_r = c_f p_r + C$$

får en bestemt C og dermed blir

$$\rho = \rho_r \exp[c_f(p-p_r)]. \qquad (7)$$

c) Setter inn for u fra Darcy's lov

$$u = -\frac{Ck}{\mu}\frac{\partial p}{\partial x}$$

i bevarelsesligningen

$$\frac{\partial}{\partial x}(\rho u) - q + \frac{\partial}{\partial t}(\phi\rho) = 0$$

og finner et uttrykk for $\frac{\partial p}{\partial x}$ ved å derivere (7) med hensyn på x:

$$\frac{\partial \rho}{\partial x} = \rho c_f \frac{\partial p}{\partial x}. \qquad \ldots \qquad (8)$$

Samlet gir dette

$$\frac{Ck}{\mu c_f}\frac{\partial^2 \rho}{\partial x^2} + q = \phi \frac{\partial \rho}{\partial t}. \qquad \ldots \qquad (9)$$

d) Rekkeutvikler til 2. orden:

$$\begin{aligned}
\frac{\partial^2 \rho}{\partial x^2} &= \rho_r \frac{\partial^2}{\partial x^2} \exp[c_f(p-p_r)], \\
&= \rho_r \frac{\partial^2}{\partial x^2}\left[1 + c_f(p-p_r) + \frac{1}{2!}c_f^2(p-p_r)^2 + \cdots\right], \\
&\approx \rho_r c_f \frac{\partial}{\partial x}\left[\frac{\partial p}{\partial x} + c_f(p-p_r)\frac{\partial p}{\partial x}\right], \\
&= \rho_r c_f \left[\frac{\partial^2 p}{\partial x^2} + c_f\left(\frac{\partial p}{\partial x}\right)^2 + c_f(p-p_r)\frac{\partial^2 p}{\partial x^2}\right], \\
&= \rho_r c_f \left[(1 + c_f(p-p_r))\frac{\partial^2 p}{\partial x^2} + c_f\left(\frac{\partial p}{\partial x}\right)^2\right], \\
&\approx \rho_r c_f (1 + c_f(p-p_r))\frac{\partial^2 p}{\partial x^2}
\end{aligned}$$

dersom

$$\left|\frac{\partial^2 p}{\partial x^2}\right| \gg \frac{c_f\left(\frac{\partial p}{\partial x}\right)^2}{1 + c_f(p-p_r)}.$$

e) Dersom $c_f(p-p_r) \ll 1$ blir

$$\frac{\partial^2 \rho}{\partial x^2} \approx \rho_r c_f \frac{\partial^2 p}{\partial x^2}.$$

Videre finner vi tilsvarende som for (8) at

$$\frac{\partial \rho}{\partial t} = \rho c_f \frac{\partial p}{\partial t} \approx \rho_f c_f \frac{\partial p}{\partial t}.$$

Setter vi dette inn i (9) og multipliserer med konstanten b_r så får vi

$$\frac{Ckb_r}{\mu}\frac{\partial^2 p}{\partial x^2} + \frac{qb_r}{\rho_r} = \phi c_f b_r \frac{\partial p}{\partial t}. \qquad \ldots \qquad (10)$$

La R betegne reservoarforhold, S overflateforhold, T tid, M masse og V volum. Volumene V_S og V_R blir da fluidvolum på overflaten og i reservoaret. La videre V_B bety bulkvolum. Da er $b_r = V_S/V_R(p_r)$, $q = M/(TV_B)$, $\rho_r = M/V_R(p_r)$, slik at

$$\frac{qb_r}{\rho_r} : \frac{MV_SV_R}{MTV_RV_B} = \frac{V_S}{TV_B},$$

altså volumrate på overflaten per bulkvolum.

Oppgave 2

a) Vi ser på blokk nr. i og lar A betegne grenseflaten mot naboblokken $i-1$ i negativ x-retning og B grenseflaten mot naboen $i+1$ i positiv x-retning. For å forenkle framstillingen setter vi hjelpestørrelsen Y lik

$$Y = \frac{Ckb_r}{\mu}\frac{\partial p}{\partial x},$$

og (10) skrives som

$$\frac{\partial Y}{\partial x} + \frac{qb_r}{\rho_r} = \phi c_f b_r \frac{\partial p}{\partial t}. \qquad \ldots \qquad (11)$$

Vi bruker differansemetoden til å diskretisere ligning 11, høyre siden i romlig x-retning, og venstre siden i tid, og får[1]

$$\frac{Y_i\big|_B - Y_i\big|_A}{\Delta x} + \frac{q_i b_r}{\rho_r} = \phi c_f b_r \frac{p_i(t_{n+1}) - p_i(t_n)}{\Delta t}. \qquad \ldots \qquad (12)$$

Videre er

$$Y_i\big|_B \approx \frac{Ckb_r}{\mu}\frac{p_{i+1}(t) - p_i(t)}{\Delta x},$$

og tilsvarende for det andre leddet,

$$Y_i\big|_A \approx \frac{Ckb_r}{\mu}\frac{p_i(t) - p_{i-1}(t)}{\Delta x}.$$

Disse to uttrykkene gir henholdsvis volumstrøm av olje inn fra høyre og venstre mot blokk i og ligning 12 setter differansen lik endring av oljevolum pr. tidsenhet i blokken ved ekspansjon eller injeksjon. I disse uttrykkene for volumstrømmen inn mot blokk i må vi angi ved hvilket tidsnivå vi skal velge å bruke trykket. Det vanligste i moderne, industrielle reservoarsimuleringsmodeller er å velge t ved ny tid t_{n+1} (implisitt formulering).

[1] For enkelthets skyld bruker vi fortsatt standard likhetstegn mellom venstre og høyre siden, selv om disse nå er diskretiserte tilnærminger til differensialligningens høyre og venstre side.

Innsatt i ligning 12 får vi da

$$\begin{aligned}
&\frac{Ckb_r}{\mu(\Delta x)^2}[p_{i+1}(t_{n+1}) - p_i(t_{n+1})] \\
&-\frac{Ckb_r}{\mu(\Delta x)^2}[p_i(t_{n+1}) - p_{i-1}(t_{n+1})] \\
&\qquad\qquad + \frac{q_i b_r}{\rho_r} \quad = \quad \phi c_f b_r (p_i(t_{n+1}) - p_i(t_n))/\Delta t.
\end{aligned}$$

La oss sette $T = Ckb_r/\mu(\Delta x)^2$ og forenkle notasjonen ved å sette $p_k \equiv p_k(t_{n+1})$ for vilkårlig k. Ligningen ordnet blir da

$$Tp_{i-1} - (2T + \phi c_f b_r/\Delta t)p_i + Tp_{i+1} = -(\phi c_f b_r p_i(t_n)/\Delta t + \frac{q_i b_r}{\rho_r}).$$

Det er nå nødvendig å ta spesielt hensyn til endeblokkene. Vi skal anta at venstre endeflate av den første blokken og høyre endeflate av den siste blokken er lukket, eller sagt på en annen måte, $Y_1\big|_A = 0$ og $Y_l\big|_B = 0$, når N betegner siste blokk. For første blokk blir ligningen da

$$-(T + \phi c_f b_r/\Delta t)p_1 + Tp_2 = -(\phi c_f b_r p_1(t_n)/\Delta t + \frac{q_1 b_r}{\rho_r}),$$

og for siste blokk N

$$Tp_{N-1} - (T + \phi c_f b_r/\Delta t)p_N = -(\phi c_f b_r p_N(t_n)/\Delta t + \frac{q_N b_r}{\rho_r}).$$

La k være lik 1 dersom blokknummeret i er lik 1 eller N og 2 ellers. Da kan vi definere

$$\begin{aligned}
B &= kT + \phi c_f b_r/\Delta t \\
a_i &= T/B \\
c_i &= T/B \\
d_i &= (\phi c_f b_r p_i(t_n)/\Delta t + (\frac{qb_r}{\rho_r})_i)/B,
\end{aligned}$$

og får ligning 12 på følgende numerisk form

$$\boxed{a_i p_{i-1} - p_i + c_i p_{i+1} = -d_i.} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (13)$$

b) Utkast til kode:

```
      subroutine tridia(imx)
c
c  solution subroutine for solving tridiagonal system of
c  linear equations; -1 on the main diagonal; matrix a
```

```
c  denotes the subdiagonal elements;
c  matrix c the superdiagonal, and matrix d the
c  right-hand-side;  imx is number of equations.
c
       double precision a, c, d, p, dia, dmod
       common /blk1/a(40), c(40), d(40), p(40)
       dimension dia(40), dmod(40)
c
c  initialization
       dia(1) = 1.d0
       dmod(1) = d(1)
c
c  dia: modified diagonal
c  dmod: modified right-hand- side
c
c  upper triangularization
       do 60 i = 2, imx
         dia(i) = 1.d0 - a(i)/dia(i-1)*c(i-1)
         dmod(i) = d(i) + a(i)/dia(i-1)*dmod(i-1)
   60           continue
c
c  solving for last pressure
       p(imx) = dmod(imx)/dia(imx)
c
c  back substitution
       do 70 k = 2, imx
               i = imx - k + 1
               p(i) = (dmod(i) + c(i)*p(i+1)) / dia(i)
   70           continue
       return
       end
c
c  end subrutine tridia
c
```

b) Strukturen til programmet går framgår av det komplette kodeforslag som er gitt under. For en fullverdig besvarelse kreves det at kandidaten kun gir hovedtrekkene i denne koden. Merk spesielt at `b` i koden tilsvarer b_r, `dbdp` tilsvarer $c_f b_r$ og `qs` tilsvarer qb_r/ρ_r.

```
          program endim
```

```
c **real variables in double precision
c a: matrix, normalized coefficient in front of p(i-1)
c aa: coefficient in front of p(i)
c c: matrix, normalized coefficient in front of p(i+1)
c b: volume factor, stb/rb
c conv: conversion factor, Darcy's law
c d: matrix, minus left- hand-side of equation
c dd: auxiliary variable to calculate d(i)
c dbdp: derivative of b with respect to p, stb/rb/psi
c dt: timestep size, days
c dx: block length, ft
c dy: block height, ft
c dz: block width, ft
c en: 1.d0
c fpb: conversion factor, ft3/bbl
c k: permeability, md
c p: matrix, (new) pressures at timelevel tn+1, psi
c phi: porosity, dimensionless
c pn: matrix, (old) pressures at timelevel tn, psi
c porig: original oil pressure, psi
c qs: matrix, surface rate, stb/d
c time: simulated time, days (equal to tn+1)
c tn: old time, days
c trans: transmissibility, Ckb/mu/dx/dx
c to: 2.d0
c vis: viscosity, cp
c vol: block volume, rb
c x: pressure point, ft
      double precision a, aa, b, c,
     z         conv, d, dd, dbdp, dt,
     z         dx, dy, dz, en, fpb, k,
     z         p, phi, pn, porig, qs,
     z         time, tn, trans, to, vis,
     z         vol, x
c
c **integers
c antall: number equal to 1 or 2
c i: block number
c im: i-1
c imx: number of blocks, maximum of i
c nmx: number of timesteps, maximum of n
      integer antall, i, im, imx, nmx
c
```

```
c **character variables for defining in and out files
c         character innfil*80
c         character utfil*80
c
c present dimensioning allows the simulation of a 40 block
c linear system.
      dimension qs(40), x(40), pn(40)
c
c common storage area is used between the main
c program and subroutines
      common /blk1/a(40), c(40), d(40), p(40)
c
c **open files for reading and writing.  input data are read
c from your own data file. output results are
c written to the result file.
c
c **interactive spesification of the files are commented out
c but included for reference. char(63) produces a question mark.
c if a horizontal tab is needed, its code is char(9).
c         write(*,*)'innfil ' , char(63)
c         read(*,*) innfil
c
c         write(*,*)'utfil ', char(63)
c         read(*,*)utfil
c
c the open-close-open sequence for the file 'utdat' is to delete
c old version. Otherwise, the last part of the largest 'utfil'
c would never be deleted.
      open (5,file='inndat',status='old')
      open (10,file='utdat')
      close (10,status='delete')
      open (10,file='utdat')
c
c     ** read and write input data.
      read  (5,*) imx, nmx
      write (10,3010) imx, nmx

      read  (5,*) k, phi, dx, porig, b
      read  (5,*) vis, dbdp, dt, dz, dy
      write (10,3020) k, phi, dx, porig, b, vis,
     z         dbdp, dt, dz, dy
      read (5,*) (qs(i),i = 1,imx)
      write (10,3030) (i,qs(i), i = 1,imx)
```

```
      write (10,3080)
c
c ***begin calculations
c
c **universal constants
      conv = 0.00632827d0
      fpb = 5.6146d0
      en = 1.d0
      to = 2.d0
c        pi  = 3.1415926d0
c
c **other constants
      vol = dz*dy*dx/fpb
      trans =  conv*k*b/vis/dx/dx
c
c **matrices a and c (note that a(1) and c(imx) are given values
c   that never are used while c(1) and a(imx) are different from
c   the other a(i) and c(i) that all are equal)
c **note that if the timestep size dt is changed between
c   timesteps, this loop has to be moved inside the timestep
c   loop
      do 31 i = 1, imx
              antall = to
              if (i.eq.1) antall = en
              if (i.eq.imx) antall = en
              aa = antall*trans + phi*dbdp/dt
              a(i) = trans/aa
              c(i) = trans/aa
   31          continue
c
c calculate pressure points
      x(1) = dx/to
      do 32 i = 2, imx
              x(i) = dx + x(i-1)
   32          continue
      write (10,3040) (i,x(i),i=1, imx)
      write (10,3080)
c
c initialize pressures and left-hand-side
      do 10 i = 1,imx
              pn(i) = porig
              p(i) = porig
   10          continue
```

```
c
c initialize time
      time = 0.d0
c
c ***start timestep loop
      do 6070 l = 1, nmx
c
c **update pressures and d(i) which depends on pn(i)
c **note that a(i) and c(i) also needs to be updated
c   if dt is changed
              do 21 i = 1, imx
                      pn(i) = p(i)
c
                      antall = to
                      if (i.eq.1) antall = en
                      if (i.eq.imx) antall = en
                      aa = antall*trans + phi*dbdp/dt
                      dd = phi*dbdp*pn(i)/dt
                      d(i) = (dd + qs(i)/vol)/aa
   21         continue
c
c call to the tridiagonal solution routine to get new pressures
              call tridia(imx)
c
c write *****'s indicate new timestep
              write (10,3070)
c write new pressures
              write (10,3050) (i, p(i), i=1, imx)
c
c calculate time simulated
              time = time + dt
c
c write timestep info
              write (10,3060)l,time
c
cc slutt tidsstegslØyfe
 6070          continue
c
c     ** close files
       close (5)
       close (10)
c
c     ** format statements
```

```
 3010           format (
     z  '  number of blocks in x-direction, imx .......... :', i3
/
     z  '  maximum number of time steps, nmx .......... :', i3  //)
c
 3020           format (
     z  '  permeability, k, md  ......................... :', f12.4/
     z  '  porosity, phi, fraction ..................... :', f8.4
/
     z  '  block length (x- dir), dx, ft ................ :', f10.2/
     z  '  original oil pressure, porig, psia ........... :', f10.1/
     z  '  volume factor, b, std vol / res vol .......... :', f10.4/
     z  '  oil viscosity, vis, cp ....................... :', f10.4/
     z  '  dbdp, std vol / res vol / psi................. :', f10.8/
     z  '  time increment, dt, days ..................... :', f10.6/
     z  '  block width, dz, ft .......................... :', f8.1/
     z  '  block thickness, dy, ft ...................... :', f8.1//)
c
 3030           format (
     z  '  block no. :', i3, '     oil rate, qs, stb/d .... ', f10.1)
c
 3040           format (
     z  '  block no. :', i3, '     pressure point, x(i), ft ', f8.1)
c
 3050           format (
     z  '  block no. :', i3, '     oil pressure, psia ..... ', f10.1)
c
 3060           format (2x,'steps=',i4,t22,'time=',d12.4//)
c
 3070           format (25('*****')/)
c
 3080           format (/)
c
       end
c
c end of main program
c
```

Det er ikke nødvendig å iterere siden alle koeffisientene i differensialligningen (10) er konstante (og det brukes direkte løsning av det lineære ligningssettet).

d) I dette forenklede tilfellet med konstante koeffisienter så går det an å finne analytisk løsning. Et eksempel på dette er gjennomgått i Øving 3 hvor det samtidig startes produksjon og injeksjon med samme rate i hver ende av reservoaret.

Oppgave 3

Det er tilstrekkelig å simulere et vertikalt tverrsnitt som står normalt på den horisontale brønnen i oljesonen siden reservoaret kan betraktes som uendelig; endeeffekter til brønnen kan neglisjeres; og deformasjonen av de to kontaktene kan beskrives i et slikt tverrsnitt.

I dette tverrsnittet vil det være symmetri om en normal på brønnen, venstre og høyre halvplan oppfører seg likt. Vi kan derfor nøye oss med å simulere det ene halvplanet.

Valg av rutenett bør være slik at trykkfallene mellom naboblokker er tilnærmet like i nær stasjonær tilstand. Da vil nøyaktigheten være tilnærmet den samme over hele rutenettet. Dette betyr finere grid inn mot brønner og over diskotinuiteter i strømningsegenskaper som f.eks. permeabiliteter.

Vi velger et koordinatsystem med x-aksen horisontal og normal på brønnen, y-aksen vertikal og z-aksen horisontal langs brønnen. Dermed vil det være bare en (vilkårlig) blokklengde i z-retning.

Trykkgradientene øker i x-retning inn mot brønnen og blokklengdene foreslås øket med en konstant faktor, la oss si 2.

I y-retning må det tas hensyn til at trykkgradientene øker når en nærmer seg brønnen. Samtidig må det være fin-gridding over de to kontaktene som utgjør diskontinuiteter i strømningsegenskaper. Imidlertid vil jo de to kontaktene bevege seg mens griddet ligger fast. For å kunne få god nøyaktighet selv om kontaktene beveger seg, må det derfor være like fint grid over hele oljesonen mellom de opprinnelige kontaktene.

Et forslag blir da som følger:

GRID

Number of gridblocks

$15 \times 155 \times 1 = 2325$

x-direction block sizes, ft

0.5, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 5000.

y-direction block sizes, ft, from top

20×165, 80, 57, 37, 12, 6, 4, 2, 10×1, (GOC), 40×1, 1 (WELL), 40×1, (WOC), 10×1, 2, 4, 6, 12, 20, 40, 80, 20×165

z-direction block size, ft

1500.

HØGSKOLEN I STAVANGER

AVDELING FOR TEKNISK - NATURVITENSKAPELIGE FAG

DATO: 2. MAI 1996

EKSAMEN I: TE 192 Reservoarsimulering, innføring

VARIGHET: kl 09.00 – 14.00

TILLATTE HJELPEMIDLER: Kalkulator

OPPGAVESETTET BESTÅR AV: 2 sider

MERKNADER: Ingen

Oppgave 1

For væske med konstant kompressibilitet c_f er tettheten gitt ved $\rho = \rho_r \exp[c_f(p - p_r)]$, hvor ρ_r er tettheten ved referansetrykk p_r, og det kan vises at diffusivitetsligningen kan tilnærmes med

$$\frac{Ckb_r}{\mu}\frac{\partial^2 p}{\partial x^2} + \frac{qb_r}{\rho_r} = \phi c_f b_r \frac{\partial p}{\partial t}, \qquad (14)$$

hvor C er en omregningsfaktor, b volumfaktor, qb_r/ρ_r volumrate ved overflateforhold pr. bulkvolum. Vi antar at permeabilitet, viskositet og porøsitet er konstante.

a) Ligning (14) diskretiseres etter differansemetoden med N like blokker av lengde Δx og med like lange tidssteg $\Delta t = t_{n+1} - t_n$. Det brukes implisitt formulering. Svaret kan skrives på formen

$$a_i p_{i-1} - p_i + c_i p_{i+1} = -d_i. \qquad (15)$$

Hva er uttrykkene for koeffisientene a_i, c_i og d_i?

b) Ligning (14) utvides til to dimensjoner. Bruk kvadratiske blokker og lik permeabilitet i x- og y-retning. Hva blir da formen på simuleringsligningen tilsvarende ligning (15) og uttrykkene for koeffisientene? Forklar spesielt hvordan koeffisientene vil se ut for kantblokkene.

c) Vis at for to dimensjoner må det løse et lineært ligningssystem med en pentadiagonal koeffisientmatrise, og skisser hvordan dette kan gjøres for et eksempel på 5×4 blokker dersom det brukes naturlig nummerering.

Oppgave 2

a) Vis hvilke endringer som må gjøres med koeffisientene i ligningen (15) for å simulere et radielt system. Hvor er trykkpunktet i en blokk?

b) La $r_1 = r_w$ og $r_{N+1} = r_e$ hvor N er antall blokker, r_w er brønnradius og r_e radius til ytre grense. Vis hvordan blokklengdene $\Delta x_i = r_{i+1} - r_i$ må velges for at trykkfallet over hver blokk skal bli like stort ved stasjonær strøm. Hvorfor er det ønskelig?

c) Forklar hvordan simulering av en trykkfallstest kan brukes til å kontrollere modellen. Gjør spesielt rede for om en kan bruke alle datapunktene fra en simulering.

d) Hvordan kan en legge inn en realistisk borhullseffekt i modellen?

Oppgave 3

a) Hvilke ligninger vil erstatte diffusivitetsligningen (14) for samtidig strøm av olje og vann i et reservoar med helning? Definer de størrelsene som inngår.

b) En simulering starter ofte med reservoaret i likevekt. Da må først trykk og metning i hver numeriske blokk bringes i likevekt i tyngdefeltet ved å ta hensyn til kapillartrykket som funksjon av høyden over det frie vannivå (olje-vann kontakten). Kapillartrykkskurven er gitt som en del av inngangsdataene.

Skisser en algoritme som vil initialisere simuleringsmodellen.

Oppgave 4

Et umettet oljereservoar har en stor underliggende vannsone. Reservoaret skal bygges ut med vertikale brønner som kompletteres et stykke oppe i oljesonen. Under produksjon vil vannet bli dradd oppover mot brønnen. Reservoaret er lagdelt med ulik permeabilitet i hvert lag. Lagdelingen varierer fra brønn til brønn. Det er ønskelig å anslå hvor stor produksjonsrate hver brønn kan tåle uten at vannet bryter inn i brønnen.

a) Hvilken type simuleringsmodell vil du bruke? Angi antall dimensjoner, antall faser, type koordinatsystem, om du vil bruke en komponentmodell eller "black oil" modell, og begrunn valgene.

b) Skisser et fornuftig numerisk rutenett for problemet og angi hvilke designkriterier du legger til grunn. Hvordan vil du kontrollere om rutenettet er fornuftig?

c) Må rutenettet kontrolleres for hver brønn? Begrunn svaret.

Utkast til løsning

Oppgave 1

a) Vi ser på blokk nr. i og lar A betegne grenseflaten mot naboblokken $i-1$ i negativ x-retning og B grenseflaten mot naboen $i+1$ i positiv x-retning. For å forenkle framstillingen setter vi hjelpestørrelsen Y lik

$$Y = \frac{Ckb_r}{\mu}\frac{\partial p}{\partial x},$$

og diffusivitetsligningen skrives som

$$\frac{\partial Y}{\partial x} + \frac{qb_r}{\rho_r} = \phi c_f b_r \frac{\partial p}{\partial t}. \qquad \ldots \qquad (16)$$

Vi bruker differansemetoden til å diskretisere ligning 16, høyre siden i romlig x-retning, og venstre siden i tid, og får[2]

$$\frac{Y_i\big|_B - Y_i\big|_A}{\Delta x} + \frac{q_i b_r}{\rho_r} = \phi c_f b_r \frac{p_i(t_{n+1}) - p_i(t_n)}{\Delta t}. \qquad \ldots \qquad (17)$$

Videre er

$$Y_i\big|_B \approx \frac{Ckb_r}{\mu}\frac{p_{i+1}(t) - p_i(t)}{\Delta x},$$

og tilsvarende for det andre leddet,

$$Y_i\big|_A \approx \frac{Ckb_r}{\mu}\frac{p_i(t) - p_{i-1}(t)}{\Delta x}.$$

Disse to uttrykkene gir henholdsvis volumstrøm av olje inn fra høyre og venstre mot blokk i og ligning 17 setter differansen lik endring av oljevolum pr. tidsenhet i blokken ved ekspansjon eller injeksjon. I disse uttrykkene for volumstrømmen inn mot blokk i må vi angi ved hvilket tidsnivå vi skal velge å bruke trykket. Det vanligste i moderne, industrielle reservoarsimuleringsmodeller er å velge t ved ny tid t_{n+1} (implisitt formulering).

Innsatt i ligning 17 får vi da

$$\begin{aligned}
&\frac{Ckb_r}{\mu(\Delta x)^2}[p_{i+1}(t_{n+1}) - p_i(t_{n+1})] \\
&-\frac{Ckb_r}{\mu(\Delta x)^2}[p_i(t_{n+1}) - p_{i-1}(t_{n+1})] \\
&\qquad\qquad + \frac{q_i b_r}{\rho_r} = \phi c_f b_r (p_i(t_{n+1}) - p_i(t_n))/\Delta t.
\end{aligned}$$

[2]For enkelthets skyld bruker vi fortsatt standard likhetstegn mellom venstre og høyre siden, selv om disse nå er diskretiserte tilnærminger til differensialligningens høyre og venstre side.

La oss sette $T = Ckb_r/\mu(\Delta x)^2$ og forenkle notasjonen ved å sette $p_k \equiv p_k(t_{n+1})$ for vilkårlig k. Ligningen ordnet blir da

$$Tp_{i-1} - (2T + \phi c_f b_r/\Delta t)p_i + Tp_{i+1} = -(\phi c_f b_r p_i(t_n)/\Delta t + \frac{q_i b_r}{\rho_r}).$$

Det er nå nødvendig å ta spesielt hensyn til endeblokkene. Vi skal anta at venstre endeflate av den første blokken og høyre endeflate av den siste blokken er lukket, eller sagt på en annen måte, $Y_1\big|_A = 0$ og $Y_l\big|_B = 0$, når N betegner siste blokk. For første blokk blir ligningen da

$$(T + \phi c_f b_r/\Delta t)p_1 + Tp_2 = -(\phi c_f b_r p_1(t_n)/\Delta t + \frac{q_1 b_r}{\rho_r}),$$

og for siste blokk N

$$Tp_{N-1} - (T + \phi c_f b_r/\Delta t)p_N = -(\phi c_f b_r p_N(t_n)/\Delta t + \frac{q_N b_r}{\rho_r}).$$

La m være lik 1 dersom blokknummeret i er lik 1 eller N og 2 ellers. Da kan vi definere

$$\begin{aligned} B &= mT + \phi c_f b_r/\Delta t \\ a_i &= T/B \\ c_i &= T/B \\ d_i &= (\phi c_f b_r p_i(t_n)/\Delta t + (\frac{qb_r}{\rho_r})_i)/B, \end{aligned}$$

og får ligning 17 på følgende numerisk form

$$\boxed{a_i p_{i-1} - p_i + c_i p_{i+1} = -d_i.} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (18)$$

b)

$$ap_{im,j} + cp_{ip,j} - p_{i,j} + ep_{i,jm} + fp_{i,jp} = -d_{i,j},$$

hvor a, c, d, e har indekser (i, j) og $im = i - 1$, $ip = i + 1$, etc.

Nå er

$$\begin{aligned} B &= mT + \phi c_f b_r/\Delta t \\ a &= T/B \\ c &= T/B \\ e &= T/B \\ f &= T/B \\ d_i &= (\phi c_f b_r p_i(t_n)/\Delta t + (\frac{qb_r}{\rho_r})_i)/B, \end{aligned}$$

med $m = 4$ for en generell blokk, $m = 3$ for en kantblokk, og $m = 2$ for en hjørneblokk.

c) Se kompendiet.

Oppgave 2

a) Se kompendiet.

b) Vi har at $(N-1)\ln(r_{i+1}/r_i) = \ln(r_e/r_w$, ved å bruke Darcy's lov og summere alle trykkfallene som skal være like over de $N-1$ blokkene. Dermed får vi

$$\frac{r_{i+1}}{r_i} = \left(\frac{r_e}{r_w}\right)^{1/(N-1)}.$$

På denne måten kan en tilnærmet få samme nøyaktighet i simuleringen over hele reservoaret.

c) Tidlige data kan ikke benyttes dersom en skal sammenligne med en rett linje i et halvlogaritmisk plott siden dette bare gjelder når dimensjonsløs tid basert på brønnradius $t_D > 25$. Data senere enn $t_{DA} = 0.1$ kan heller ikke brukes fordi da vil det simulerte brønntrykk være influert av ytre grense.

d) Ved å la blokk nr. 1 være en del av borhullet med høy permeabilitet og et porevolum (porøsitet) lik volumet av hulrom i borhullet fra reservoaret og opp til overflaten.

Oppgave 3

a) Se kompendiet for de to ligningene. Det er nødvendig å forklare metning, relative permeabiliteter, kapillartrykk.

Trykket må erstattes med potensialet (se kompedium) når reservoaret har helning, $\Phi = p + \rho g z$ for hver fase.

b) Kan for enkelthets skyld starte med midtpunktet av en blokk i vann-olje kontakten. Går en blokk opp. Regner ut trykket i hver fase fra tetthetene. Kapillartrykket er differansen mellom trykkene. Bruker kapillartrykkskurven til å finne metningen. Går videre til neste blokk og repeterer beregningen. Fortsetter inntil vannmetningen er blitt irredusibel.

Hele denne prosessen må nå gjøres omigjen siden endringen av trykkene vil endre tetthetene. Fortsetter med en slik sløyfe inntil trykkene ikke endres mer, sjekker mot et lite tall.

Oppgave 4

a) To faser, “black oil,” to dimensjoner (tre dimensjoner men to koordinatakser i sylindergeometri). Hvorfor sier seg nesten selv...

b) Radielt øker radiene med en faktor, kfr. tidligere oppgave. Vertikalt må det findeles rundt vann-olje kontakten og over laggrensene.

Det fins ingen annen måte enn å findele enda mer og se om det gir noen endring. En må også passe på at tidsstegslengden gir en dominerende feil.

c) Ja, siden lagdelingen er ulik.

HØGSKOLEN I STAVANGER

AVDELING FOR TEKNISK - NATURVITENSKAPELIGE FAG

DATO: 5. DESEMBER 1996

EKSAMEN I: TE 192 Reservoarsimulering, innføring

VARIGHET: kl 09.00 – 14.00

TILLATTE HJELPEMIDLER: Kalkulator

OPPGAVESETTET BESTÅR AV: 3 sider

MERKNADER: Ingen

Oppgave 1

a) Gitt bevaringsligningen

$$\frac{\partial}{\partial x}(\rho u) - q + \frac{\partial}{\partial t}(\phi \rho) = 0, \qquad (19)$$

hvor ρ er tettheten av fluidet som strømmer. Hvilken fysisk størrelse er det som bevares? Hva står symbolene u, q og ρ for og hvilke enheter kan de uttrykkes i?

b) Anta at permeabilitet og porøsitet kan betraktes som konstante og vis at ligningen for horisontal strøm av et fluid (diffusivitetsligningen) kan skrives på formen

$$k\frac{\partial}{\partial x}\left(\frac{\rho}{\mu}\frac{\partial p}{\partial x}\right) + q = \phi\frac{\partial \rho}{\partial t}. \qquad (20)$$

c) Tilstandsligningen for en reell gass er gitt ved

$$\rho = \frac{M}{RTZ(p)}p, \qquad (21)$$

hvor M er molvekten, R er gasskonstanten og T er temperaturen. Z-faktoren er lik 1 for ideell gass og er kun en funksjon av trykk når sammensetningen av gassen kan betraktes som konstant.

Kompressibiliteten c til et fluid er definert ved

$$c = \frac{1}{\rho}\left(\frac{\partial \rho}{\partial p}\right)_T. \qquad (22)$$

La pseudotrykket $m(p)$ være definert ved

$$m(p) = \int_0^p \frac{\rho(p')}{\mu(p')} dp'. \qquad \ldots \qquad (23)$$

Vis ved hjelp av ligningene (20)–(23) at diffusivitetsligningen for en reell gass med konstant sammensetning kan skrives som

$$k\frac{\partial^2 m}{\partial x^2} + q = \phi\mu c\frac{\partial m}{\partial t}. \qquad \ldots \qquad (24)$$

d) En numerisk løsning av ligning (24) vil gi verdien av m i hver numeriske blokk etter hvert tidssteg. Det vil derfor være nødvendig med en rask metode for å omgjøre pseudotrykk til trykk. Foreslå hvordan det kan gjøres.

e) Anta nå at kompressibiliteten c også er konstant og diskretiser (24) i tid og rom ved å bruke like lange blokker. Bruk implisitt formulering og skriv svaret på formen

$$a_i m_{i-1} - m_i + c_i m_{i+1} = -d_i. \qquad \ldots \qquad (25)$$

f) Hva er fordeler og ulemper med implisitt formulering i forhold til andre diskretiseringsskjemaer?

g) Skriv (25) på matriseform dersom $\max(i) = 4$ og forklar med ord hvordan dette ligningsettet kan løses.

h) Hvordan vil du kontrollere hvor god den numeriske løsningen er?

Oppgave 2

I forelesningene ble det innledningsvis utledet en diffusivitetsligning for horisontal strøm av en fase i en dimensjon (x) gjennom et konstant tverrsnitt (A). I denne oppgaven danner x-aksen en vinkel α med horisontalplanet og tverrsnittet av det endimensjonale reservoaret er en funksjon av x, $A := A(x)$.

a) Utled følgende diffusivitetsligning for dette problemet:

$$\frac{\partial}{\partial x}\left[\frac{A(x)k\rho}{\mu}\left(\frac{\partial p}{\partial x} + \rho g \sin\alpha\right)\right] + A(x)q = A(x)\frac{\partial}{\partial t}(\phi\rho). \qquad \ldots \qquad (26)$$

b) Bruk ligning (26) til å skrive ned diffusivitetsligningen for et radielt system. Gjør spesielt rede for om gravitasjonsleddet kan beholdes i ligningen.

c) La $r_1 = r_w$ og $r_{N+1} = r_e$ hvor N er antall blokker, r_w er brønnradius og r_e radius til ytre grense. Vis hvordan blokklengdene $\Delta r_i = r_{i+1} - r_i$ må velges for at trykkfallet over hver blokk skal bli like stort ved stasjonær strøm. Hvorfor er det ønskelig?

d) Forklar hvordan simulering av en trykkfallstest kan brukes til å kontrollere den numeriske modellen. Gjør spesielt rede for om en kan bruke alle datapunktene fra en simulering.

Oppgave 3

Ligningsystemet

$$\begin{aligned} 2x_1 + x_2 &= 1, \\ x_1 + 2x_2 &= 5, \end{aligned} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (27)$$

skal brukes til å sammenligne tre iterative løsningsmetoder: (1) Gauss-Jordan, (2) Gauss-Seidel og (3) SOR.

a) Skriv ned iterasjonskjemaet for de tre metodene med k som iterasjonsteller.

b) La startverdiene være $x_1 = 1$ og $x_2 = 0$. Hvor mange iterasjoner trengs det for at feilen i estimatet av x_1 skal bli mindre enn 10% for de tre metodene? (For SOR brukes $\omega = 1.15$.)

HØGSKOLEN I STAVANGER

AVDELING FOR TEKNISK - NATURVITENSKAPELIGE FAG

DATO: 11. DESEMBER 1997

EKSAMEN I: TE 6192 Reservoarsimulering, innføring

VARIGHET: kl 09.00 – 14.00

TILLATTE HJELPEMIDLER: Kalkulator

OPPGAVESETTET BESTÅR AV: 3 sider

MERKNADER: Ingen

Oppgave 1

a) Utled diffusivitetsligningen for horisontal strøm av et fluid i to dimensjoner ved å bruke massebalanse på et volumelement $\Delta V = \Delta x_i \Delta y_j \Delta z$. Tettheten ved reservoarbetingelser er gitt ved $\rho = b\rho_0$, hvor ρ_0 er tettheten ved standard forhold og b er invers volumfaktor. Svaret skal bli

$$\frac{\partial}{\partial x}\left(\frac{Ck_x b}{\mu}\frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{Ck_y b}{\mu}\frac{\partial p}{\partial y}\right) + q = \phi \frac{db}{dp}\frac{\partial p}{\partial t}, \qquad \ldots\ldots\ldots\ldots \quad (28)$$

hvor C er en omregningsfaktor.

b) Diskretiser ligning 28 med implisitt formulering og vis at svaret kan skrives på formen

$$a_{i,j}\, p_{im,j} + c_{i,j}\, p_{ip,j} - p_{i,j} + e_{i,j}\, p_{i,jm} + f_{i,j}\, p_{i,jp} = d_{i,j}. \qquad \ldots\ldots\ldots\ldots \quad (29)$$

c) Nevn to andre numeriske formulering som kunne ha vært brukt til å diskretisere ligning 28. Skisser kort hva som adskiller disse formuleringen fra den implisitte og angi fordeler og ulemper med alle tre formuleringene.

d) Vis med et eksempel, (4×3) rutenett, hvordan ligning 29 kan skrives på matriseform som et linært ligningsett og forklar med ord hvordan dette ligningsettet kan løses.

e) Vis hvordan det samme ligningssystemet kan løses med Linjevis Suksessive Over-Relaksasjoner (LSOR).

Oppgave 2

Anta at invers volumfaktoren b er gitt som en funksjon av trykket p i form av en tabell fra laboratoriet. Forklar med ord og ved hjelp av skiss(er) hvordan en slik inngangstabell kan omformes til en intern, ekvidistant tabell i trykk.

Vis hvordan denne interne tabellen kan brukes til raskt å finne b-verdien for et gitt trykk. Hvordan måtte en ha gått fram uten en ekvidistant, intern tabell og hva ville ulempen ha vært?

Oppgave 3

Den følgende formel gir trykket p som en funksjon av posisjon x og tid t i et endimensjonalt reservoar hvor det samtidig startes med injeksjon og produksjon med like stor rate i hver ende ved tid $t = 0$,

$$p(x,t) = p_i - q(x-L) - 8Lq \sum_{n=0}^{\infty} \frac{\cos(\frac{(2n+1)\pi}{2L}x)}{((2n+1)\pi)^2} \exp(-(\frac{(2n+1)\pi}{2L})^2 c_1^2 t). \qquad (30)$$

Skisser en subrutine i Fortran som beregner trykket med en viss nøyaktighet, for en gitt verdi av x og t. Anta at data som p_i, q, L, c_1, blir overført fra hovedprogrammet via en 'common block.'

Oppgave 4

Lag et blokkskjema (flytdiagram) for et simuleringsprogram som løser ligning 29.

I blokkskjemaet kan et kodesegment være angitt som en boks med en kort forklaring, eventuelt med forklaringen i egen liste under skjemaet. Dette gjelder for eksempel dersom kodesegmentet betegner en subrutine.

Følgende programelementer skal være inkludert, angitt i vilkårlig rekkefølge:

- materialbalanse
- trykkløsing ved LSOR
- innlesing av data
- utskrift for hvert tidssteg
- generering av intern og ekvidistant PVT-tabell
- opsjon for radielt system
- tidsstegsløyfe

- utskrift av inngangsdata
- oppdatering av gamle trykk
- trykkløsing ved eliminasjon
- nummereringsskjema av blokker for løsningsrutinen
- fluidegenskaper
- maksimal tillatt trykkendring per tidsteg
- ny tidsstegslengde
- iterasjonssløyfe på ulineære ledd
- innlesing av nye rater ved gitt tidspunkt
- initiell trykklikevekt
- koeffisienter til lineært ligningssett

HØGSKOLEN I STAVANGER

AVDELING FOR TEKNISK - NATURVITENSKAPELIGE FAG

DATO: 23. FEBRUAR 1998

EKSAMEN I: TE 6192 Reservoarsimulering, innføring

VARIGHET: kl 09.00 – 14.00

TILLATTE HJELPEMIDLER: Kalkulator

OPPGAVESETTET BESTÅR AV: 2 sider

MERKNADER: Ingen

Oppgave 1

Diffusivitetsligningen er en kombinasjon av Darcy's lov og kontinuitetsligningen. Den blir diskretisert i tid og rom dersom det skal lages numeriske løsninger.

a) Utled kontinuitetsligningen for **tre-dimensjonal** strøm av olje over kokepunkt-strykket. Angi hvilke antagelser som eventuelt må gjøres.

b) Utled den tilhørende diffusivitetsligning.

c) Diskretiser diffusivitetsligningen med implisitt formulering.

d) Nevn to andre numeriske formulering som kan brukes til å diskretisere ligningen. Skisser kort hva som adskiller disse formuleringen fra den implisitte og angi fordeler og ulemper med alle tre formuleringene.

e) Velg et numerisk rutenett og vis at den diskretiserte ligningen blir et lineært ligningssett. Skisser hvordan dette ligningssettet kan løses ved eliminasjon.

f) Forklar hvordan det lineære ligningssettet kan løses iterativt med SOR, (i) planvis, (ii) linjevis og (iii) punktvis.

g) Vis hvordan gravitasjonskraften kan inkluderes i modellen på en korrekt måte.

h) Strøm inn mot en vertikal brønn skal simuleres. Forklar hvilke forenklinger som kan gjøres med den tre-dimensjonale modellen, avhengig av symmetrien i problemet som skal simuleres.

Oppgave 2

Anta at invers volumfaktoren b er målt ved en rekke trykk p i laboratoriet. Skisser et Fortran programsegment som omformer måledataene til en intern, ekvidistant tabell i trykk.

Vis hvordan denne interne tabellen kan brukes til raskt å finne b-verdien for et gitt trykk. Hvordan måtte en ha gått fram uten en ekvidistant, intern tabell og hva ville ulempen ha vært?

HØGSKOLEN I STAVANGER

AVDELING FOR TEKNISK - NATURVITENSKAPELIGE FAG

DATO: 26. NOVEMBER 1999

EKSAMEN I: TE 6192 Reservoarsimulering, innføring

VARIGHET: kl 09.00 – 14.00

TILLATTE HJELPEMIDLER: Kalkulator

OPPGAVESETTET BESTÅR AV: 3 sider

MERKNADER: Ingen

Oppgave 1

Diffusivitetsligningen for strøm av en svakt kompressibel væske (vann eller olje) gjennom et porøst medium kan skrives på formen

$$\frac{\partial}{\partial X}\left(\frac{Ckb}{\mu}\frac{\partial P}{\partial X}\right) = \phi\frac{\mathrm{d}b}{\mathrm{d}P}\frac{\partial P}{\partial T}, \qquad (31)$$

hvor P er trykket i psi, X er avstand i ft, k er permeabiliteten i md, b er invers volumfaktor i stb/rb, μ er viskositeten i cp og T er tiden i døgn. Anta at b i strømningsleddet, $\mathrm{d}b/\mathrm{d}P$, μ og k er konstanter. Omregningskonstanten C er lik 0.00632827.

a) Innfør dimensjonsløse størrelser (p,x,t) gitt ved $p = (P-P_i)/P_i$, $x = X/L$, $t = Tc_1^2/L^2$. Her er L lengden av reservoaret i ft, P_i initielt trykk og $1/c_1^2 = (\phi\mu\cdot\mathrm{d}b/\mathrm{d}P)/(Ckb)$. Vis at ligning (36) nå kan skrives som

$$\frac{\partial^2 p}{\partial x^2} = \frac{\partial p}{\partial t}. \qquad (32)$$

b) Vis at eksplisitt numerisk formulering med differansemetoden gir ligningen

$$\frac{p_{i-1,n} - 2p_{i,n} + p_{i+1,n}}{(\Delta x)^2} = \frac{p_{i,n+1} - p_{i,n}}{\Delta t}, \qquad (33)$$

dersom alle blokkene har samme lengde Δx. Her er p_i dimensjonsløst trykk i midtpunktet av blokk i og $\Delta t = t_{n+1} - t_n$ er tidsstegslengden.

c) Forklar kort at metoden med separasjon av variable kan egne seg til å finne en analytisk løsning av ligning (32).

d) La oss nå anta at løsningen $p(x,t)$ av ligning (32) er funnet som et produkt av en funksjon av tiden, la oss si $\psi(t)$, og en annen funksjon som kun er avhengig av x. Denne siste funksjonen rekkeutvikles (i en kompleks Fourierrekke). Bortsett fra en konstant er et typisk ledd i rekkeutviklingen gitt ved $e^{j\beta x}$, hvor β er et reelt tall og $j = \sqrt{-1}$. Et ledd i rekkeutviklingen av løsningen $p(x,t)$ kan da representeres ved $\psi(t)e^{j\beta x}$. Sett dette inn i ligning (33) og vis at forsterkningsfaktoren $\xi = \psi(t+\Delta t)/\psi(t)$ for eksplisitt formulering er gitt ved

$$\xi = 1 - 4\lambda \sin^2 \frac{\beta \Delta x}{2}, \quad \text{hvor } \lambda = \Delta t/(\Delta x)^2. \qquad \ldots\ldots\ldots\ldots\ldots\ldots \quad (34)$$

Oppgitt: $e^{j\beta x} = \cos\beta x + j\sin\beta x$,
$1 - \cos\beta x = 2\sin^2 \beta x/2$.

e) For at den numeriske løsning skal være stabil og ikke vokse over alle grenser når Δt øker eller Δx minker, så må $|\xi| \leq 1$. Vis at dette krever at $\lambda \leq \frac{1}{2}$.

f) I øvingsoppgave 4 ble følgende data brukt til å undersøke stabilitet og diskretiseringsfeil av tre numeriske formuleringer. Enhetene er som angitt under spørsmål a). Fra hvilket tidssteg av kan en forvente ustabilitet fra den eksplisitte formuleringen?

```
mx = 6
kx = 100
phi = 0.05
dex = 10
borig = 0.7889
visorg = 1.358
dbdp = 8.03d-06
delmin = 0.000015415
dtmult = 1.5
```

g) Vis at implisitt numerisk formulering av ligning (32) gir en ubetinget stabil løsning.

h) Hvordan kan en i praksis kontrollere at diskretiseringsfeilen er liten nok?

Oppgave 2

Implisitt numerisk formulering av diffusivitetsligningen for enfasestrøm gir et lineært ligningssett på formen $\mathbf{Kp} = \mathbf{d}$ hvor $\mathbf{p}$ og $\mathbf{d}$ er to søylevektorer med N elementer og $\mathbf{K} = \{k_{ij}\}$ er en $N \times N$ koeffisientmatrise.

a) Hvilken struktur har **K** for endimensjonal strøm, dvs. hvilke elementer k_{ij} er forskjellige fra null? Hva kalles en slik matrise?

b) Skisser en Fortran-rutine som kan løse et slikt lineært problem ved eliminasjon.

c) Hva er strukturen til **K** for todimensjonal strøm og hvordan kan nå ligningssystemet løses ved eliminasjon?

Oppgave 3

a) Lag en grov skisse av strukturen til et Fortran-program som skal simulere strøm av en fase i en dimensjon. Skissen skal gi rekkefølge og mulige forgreninger mellom følgende kodeelementer, her angitt i vilkårlig rekkefølge:

```
tidsstegsløyfe
materialbalanse
innlesing og utskrift
oppdatering av "gamle" størrelser
initiell trykklikevekt
deklarering av variabeltyper
kall til løsningsrutiner
kall til subrutine som setter opp koeffisientmatrisen
kall til subrutine som gir fluidegenskaper
sløyfe for iterasjoner på ulinære ledd
```

b) Vis hvor i programstrukturen under spørsmål a) du vil sette inn følgende kodeelementer:

```
omforming av k og φ for radielt system
sjekk av maksimal trykkendring over tidssteget
logikk for rateendring
beregning av brønntrykk for radielt system
beregning av gravitasjonsledd for hellende reservoar
generering av intern tabell for fluidegenskaper
"use felles_matriser"
kall til iterativ løsningsrutine SOR
```

c) Forklar hvordan en i Fortran 90 kan få til variabel dimensjonering av matriser ved hjelp av inngangsdata. Forklar også hvilke to måter som brukes til å utveksle verdier av variable mellom hovedprogram og subrutiner.

Skisse til løsning

Eksamen i Reservoarsimulering, innføring
26. november 1999.

1a) Setter inn $P = pP_i + P_i$, $X = xL$, $T = tL^2/c_1^2$ og får

$$\frac{\partial}{L\partial x}\left(\frac{Ckb}{\mu}\frac{\partial(pP_i+P_i)}{L\partial x}\right) = \phi\frac{\mathrm{d}b}{\mathrm{d}P}\frac{\partial(pP_i+P_i)}{\frac{L^2}{c_1^2}\partial t},$$

som forenklet gir oppgitt uttrykk.

1b) Vi har

$$\frac{\left(\frac{\partial p}{\partial x}\right)_{i+} - \left(\frac{\partial p}{\partial x}\right)_{i-}}{\Delta x} = \frac{\partial p}{\partial t},$$

hvor index $i+$ betyr 'tatt ved høyre blokkgrense' og indeks $i-$ betyr 'tatt ved venstre blokkgrense' av blokk `i`. Videre har vi at

$$\left(\frac{\partial p}{\partial x}\right)_{i+} = \frac{p_{i+1,n}-p_{i,n}}{\Delta x}; \quad \left(\frac{\partial p}{\partial x}\right)_{i-} = \frac{p_{i,n}-p_{i-1,n}}{\Delta x}; \quad \frac{\partial p}{\partial t} = \frac{p_{i,n+1}-p_{i,n}}{\Delta t},$$

hvor n angir 'gammelt' tidsnivå, $n+1$ 'nytt' tidsnivå. Indeks i angir trykket i midtpunktet av blokk nummer i og tilsvarende for $i+1$ og $i-1$. Merk at begge avstandsderiverte av p er tatt ved tidsnivå n, slik som det eksplisitte skjemaet krever. Trukket sammen så fås uttrykket i oppgaven.

1c) Vi setter inn i ligningen som prøveløsning at $p(x,t) = g(t)f(x)$ og finner at metoden med separasjon av variable kommer til anvendelse, se løsning til øving 3.

1d) Innsatt får vi

$$\frac{\psi(t+\Delta t)\mathrm{e}^{j\beta x} - \psi(t)\mathrm{e}^{j\beta x}}{\Delta t} = \frac{\psi(t)}{(\Delta x)^2}\left[\mathrm{e}^{j\beta(x-\Delta x)} - 2\mathrm{e}^{j\beta x} + \mathrm{e}^{j\beta(x+\Delta x)}\right],$$

og deretter

$$\begin{aligned}&(\frac{\psi(t+\Delta t)}{\psi(t)} - 1)\mathrm{e}^{j\beta x} = \\ &\qquad \frac{\Delta t}{(\Delta x)^2}\mathrm{e}^{j\beta x}\left[\cos\beta\Delta x - j\sin\beta\Delta x - 2 + \cos\beta\Delta x + j\sin\beta\Delta x\right],\end{aligned}$$

som gir

$$\xi = \frac{\psi(t+\Delta t)}{\psi(t)} = 1 + 2\lambda(\cos\beta\Delta x - 1) = 1 - 4\lambda\sin^2\frac{\beta\Delta x}{2}.$$

1e) Siden $\max(\sin^2 \frac{\beta\Delta x}{2}) = 1$, så må $\lambda \leq \frac{1}{2}$ forat $\xi \leq 1$.

1f) Ved innsetting av oppgitte data finner en at $c_1^2 = 9.16E+05$, og at $0.141 * 1.5^l = 0.5$ når l er tidsstegtelleren og multiplikatoren på tidsstegslengden er `dtmult = 1.5`. Av dette finner en at $k = 3.12$, altså etter ca. 4 tidssteg. Dersom verdien av $\sin^2 \frac{\beta\Delta x}{2}$ holder seg under 1, så vil antall tidssteg kunne bli noe større før ustabiliteten merkes.

1g) Differanseskjemaet blir nå

$$\frac{p_{i-1,n+1} - 2p_{i,n+1} + p_{i+1,n+1}}{(\Delta x)^2} = \frac{p_{i,n+1} - p_{i,n}}{\Delta t}, \qquad \ldots\ldots\ldots\ldots\ldots \quad (35)$$

og setter vi inn $p_{i,n} = \psi(t)\mathrm{e}^{j\beta x}$, så får vi etter litt forenkling,

$$\xi = \frac{1}{1 + 4\lambda \sin^2 \frac{\beta\Delta x}{2}},$$

slik at $\xi \leq 1$ for alle λ og implisitt formulering er dermed ubetinget stabil.

1h) En kan (1) sammenligne med analytisk løsning dersom en slik fins på en forenklet problemstilling, men med samme rutenett og tidssteg; (2) systematisk redusere blokklengde og tidsstegslengde inntil svaret ikke endres. En må i det siste tilfellet spesielt være oppmerksom på at det ikke nytter å redusere blokkstørrelsen dersom feilen henger på tidsstegslengden, eller motsatt.

2a **K** er en tridiagonal matrise, se kompendiet.

2b) Se kompendiet.

2c **K** er nå en penta-diagonal matrise og ligningssystemet løses på samme måte som under forrige spørsmål: Først nuller under hoveddiagonalen, så løses for siste trykk og deretter tilbakesubstituering.

3a) Se kompendiet.

3b) Se kompendiet.

3c) Variabel dimensjonering får en til ved å lage en module-fil hvor matrisene står oppført uten størrelse, på formen

```
Module felles_matriser
      integer, parameter :: dp = kind(1.d0)
      integer, parameter :: sp = kind(1.0)
   real(dp), dimension(:), allocatable ::&
      a,c,d,al,s,comp,po,&
      pold,oxplus,oxmin,a9,&
      qo,bo,x
end module felles_matriser
```

Etter å ha lest inn dimensjonen `mx` (for endimensjonalt system) settes inn en `allocate`-kommando i hovedprogrammet, f.eks. `allocate(a(mx), c(mx) ...)`.

For å utveksle parameterverdier mellom hovedprogram og subrutine kan (1) verdiene overføres i kalle-setningen, eller (2) ved en felles `module`-fil og kommandoen `use <module-fil-navn>` øverst i hovedprogram og subrutine.

HØGSKOLEN I STAVANGER

AVDELING FOR TEKNISK - NATURVITENSKAPELIGE FAG

DATO: 24. FEBRUAR 1999

EKSAMEN I: TE 6192 Reservoarsimulering, innføring

VARIGHET: kl 09.00 – 14.00

TILLATTE HJELPEMIDLER: Kalkulator

OPPGAVESETTET BESTÅR AV: 2 sider

MERKNADER: Ingen

Oppgave 1

Gitt diffusivitetsligningen for horisontal strøm av et fluid i en dimensjon på formen

$$\frac{\partial}{\partial x}\left(\frac{Ckb}{\mu}\frac{\partial p}{\partial x}\right)+q=\frac{\partial}{\partial t}(\phi b), \qquad (36)$$

hvor C er en omregningsfaktor, k absolutt permeabilitet, b invers volumfaktor, ϕ porøsitet, p trykk og q kildeledd.

a) Anta at ϕ, μ, db/dp, k er uavhengige av trykket og at b kan betraktes som konstant i strømningsleddet på venstre side av ligningen. Blokklengden Δx og permeabilitetet k skal være de samme for hver numeriske blokk. Ligningen kan da skrives på følgende numerisk form,

$$-a_i p_{i-1}^m + p_i^{n+1} - c_i p_{i+1}^m = d_i, \qquad (37)$$

hvor m enten er lik $n+1$ eller n, og n angir tidsnivået slik at tidssteget er gitt ved $\Delta t = t^{n+1} - t^n$. Start med ligning (36) og utled uttrykk for a_i, c_i, d_i for (i) implisitt, og (ii) eksplisitt formulering.

b) Nå fjernes antagelsene under spm. a). Det vil si at ϕ, μ, b, k generelt er trykkavhengige størrelser. Ligning (36) skal formuleres på en fullstendig implisitt numerisk form, altså med både trykkavhengige størrelser og strøm mellom blokkene valgt ved slutten av tidssteget. Vis hvordan dette kan gjøres ved rekkeutvikling (Newton's metode).

Oppgave 2

Anta nå at ligning (37) gjelder med implisitt formulering. Følgende løsningsmetoder kan brukes til å finne trykkene ved ny tid t^{n+1} dersom løsningen er kjent ved tid t^n: (a) Direkte, ved eliminasjon, (b) Gauss-Seidel iterasjon, (c) SOR.

Hvilke av disse tre løsningsmetodene kunne også ha vært brukt sammen med (i) eksplisitt formulering, og (ii) Crank-Nicolson formulering? Begrunn svaret.

Oppgave 3

a) Skriv ned hvilke ligninger som erstatter ligning (36) og ligning (37) dersom det skal simuleres et enfase, todimensjonalt problem.

b) Et enfase, todimensjonalt problem skal simuleres med 6, (2×3), blokker med L-SOR (Line Successive OverRelaxation) som løsningsmetode. Nummerer blokkene og skisser en Fortrankode som kan brukes til å løse ligningssettet.

Oppgave 4

a) Angi de fire ligningene som erstatter ligning (36) dersom simultan strøm av olje og vann skal simuleres. Hvilke størrelser vil det være fornuftig å velge som ukjente?

b) Forklar hva som menes med oppstrøms relative permeabiliteter og forklar hvorfor de brukes.

Oppgave 5

Den følgende formel gir trykket p som en funksjon av posisjon x og tid t i et endimensjonalt reservoar hvor det samtidig startes med injeksjon og produksjon med like stor rate i hver ende ved tid $t = 0$,

$$p(x,t) = p_i - q(x-L) - 8Lq \sum_{n=0}^{\infty} \frac{\cos(\frac{(2n+1)\pi}{2L}x)}{((2n+1)\pi)^2} \exp(-(\frac{(2n+1)\pi}{2L})^2 c_1^2 t). \qquad (38)$$

Skisser en subrutine i Fortran som beregner trykket med en viss nøyaktighet, for en gitt verdi av x og t. Forklar hvordan dataene p_i, q, L, c_1 på to ulike måter kan overføres fra hovedprogrammet til subrutinen.