

Innføring
i
RESERVOARSIMULERING

Svein M. Skjæveland
og
Jann-Rune Ursin

Høgskolen i Stavanger
august 1999

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	3
2	Diffusivitetstiligningen	4
2.1	Darcy's lov og massebevarelse	4
2.2	Generell diffusivitetstiligning	6
2.3	Diffusivitetstiligning for tre faser	7
3	Numerisk formulering	8
3.1	Tilnærming av deriverte	8
3.2	Avstandsderiverte i diffusivitetstiligningen	9
3.3	Diskretisering i tid og rom	10
3.4	Valg av numerisk formulering	11
3.5	Eksplisitt formulering	12
3.6	Implisitt formulering	12
3.7	Iterativ løsningsmetode	14
3.8	Direkte løsningsmetode	15
3.9	Skisse til flytskjema for utlevert program	16
II	Fortrankode for utlevert program	18
3.10	Module	26
3.11	Module	26
3.12	Datafil	26
3.13	Resultatfil	26
III	Øvingsoppgaver	30
1	Kjøring av utlevert program	31
1.1	Utpøving av løsningsrutiner	31
2	Automatisk tidsstegsregulering	32
2.1	Først et par justeringer	32
2.2	Ny tidsstegsregulering	32
2.3	Kjør programmet for testing	33
2.4	Praktiske råd	33

3	Sammenligning med analytisk løsning	34
3.1	Problemstilling	34
3.2	Analytisk løsning	37
4	Numeriske formuleringer og SOR	40
4.1	Problemstilling	40
4.2	Kommentarer	41
5	Variabel k, ϕ og Δx	43
5.1	Oppgave	43
5.2	Kommentarer	43
6	Hellende reservoar	45
6.1	Oppgave	45
6.2	Kommentarer	45
7	Radielt system	47
7.1	Oppgave	47
7.2	Teori	48
7.3	Kommentarer	49
8	Tabellbruk	53
8.1	Oppgave	53
8.2	Kommentarer	53
9	En fase i to dimensjoner	57
9.1	Kommentarer til Øving 9	58
9.2	Løsning med naturlig nummerering og subrutinen SOLVE	61
9.3	Løsning med LSOR	63
9.4	Noen påminnelser	64
10	To faser, olje og vann, i to dimensjoner	65
10.1	Kommentarer til Øving 10	67
10.2	IMPES-metoden	67
10.3	Oppstrøms relative permeabiliteter	68
10.4	Fullstendig implisitt formulering	68
10.5	Kordemetoden – kvasi-Newton	69
10.6	Vannligningen i detalj — oppsummering	71
10.7	Ligningssystem	72
10.8	Rateformulering	72
10.9	Noen hint til programmering	73
10.10	Andre kommentarer	75
11	Måling av relative permeabiliteter	76
11.1	Innledning	76
11.2	Framgangsmåte	76
11.3	Data	77
11.4	Forventede resultater	77
IV	Eksamensoppgaver	78

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 dataterminal.

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 blokk lengde, permeabilitet og porøsitet.
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 .

Eksempelene 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 innføringssopplegg 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

Diffusivitetssligningen

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

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:

$$v_o = -\frac{Ck_x}{\mu_o} \frac{\partial p_o}{\partial x}, \quad \dots \dots \dots (2.1)$$

hvor

C : Omregningskonstant med verdi 0.00632827 for de gitte enhetene.

v_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.

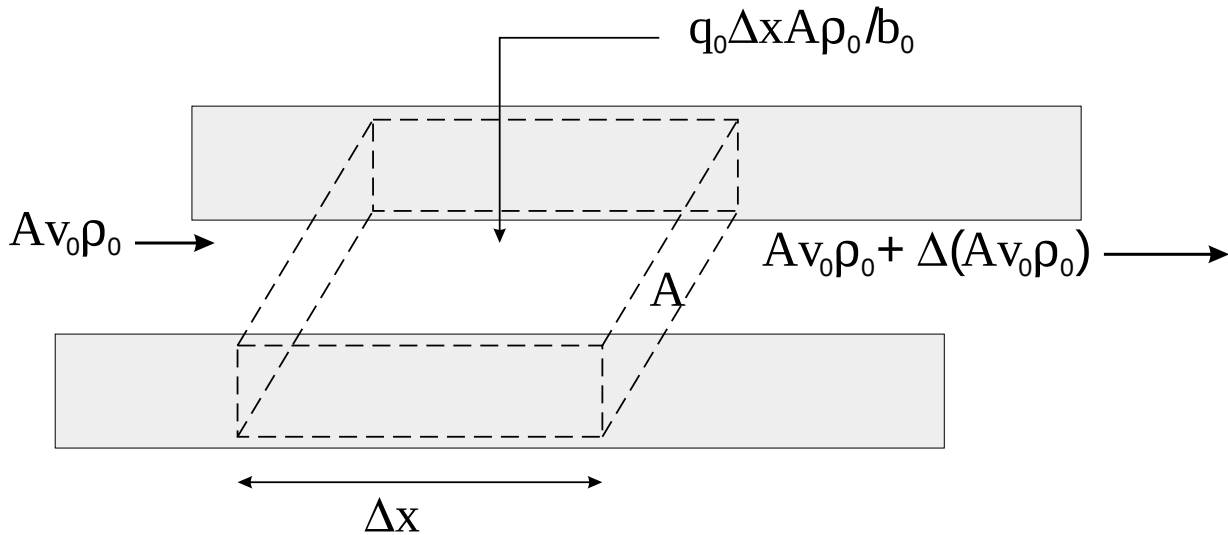
Indeks o står for olje og er egentlig overflødig før vi tar med strøm av flere faser. Kontinuitetssligningen

kan utledes ved å betrakte massestrøm inn og ut av volumelementet $\Delta V = A\Delta x$ i Figur 2.1.

I Figur 2.1 er det brukt endel nye størrelser:

A : tverrsnitt av det en-dimensjonale reservoaret, ft².

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

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

b_o : volumfaktor for olje, stb/rb.¹

ρ_o : tetthet av olje ved reservoarbetingelser, lb/rft³.

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

$$A(\rho_o v_o + \frac{q_o \Delta x \rho_o}{b_o} - (v_o \rho_o + \Delta(v_o \rho_o))). \quad \dots \quad (2.2)$$

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

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

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

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

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

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

Denne ligningen multipliserer vi med b_o/ρ_o som har dimensjonen stb/rb $\times$ rft³/lb $\rightarrow$ 5.61 · stb/lb. Dette er en konstant og kan tas innenfor derivasjonstegnet:

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

som ordnet gir

$$\frac{\partial(b_o v_o)}{\partial x} - q_o = -\phi \frac{\partial b_o}{\partial t}. \quad \dots \quad (2.4)$$

¹Merk at denne definisjonen er den inverse av den vanlige volumfaktoren B_o .

Ligning 2.4 er kontinuitetsligningen skrevet på en noe spesiell form med volumfaktor istedenfor tetthet. Det er imidlertid den vanlige framstillingsmåten for β -modeller.

I ligning 2.4 setter vi nå inn for volumhastigheten v_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

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

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/bbl/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 $q_o = q'_o / (\Delta V / \text{FPB})$, hvor $\text{FPB} = 5.6146 [\text{ft}^3/\text{bbl}]$ 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,

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

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

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

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 v_o fra ligning 2.1, så får en

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

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,

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

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 $b_o = b_{orig} + \text{dbdp}(p_o - p_{orig})$, 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

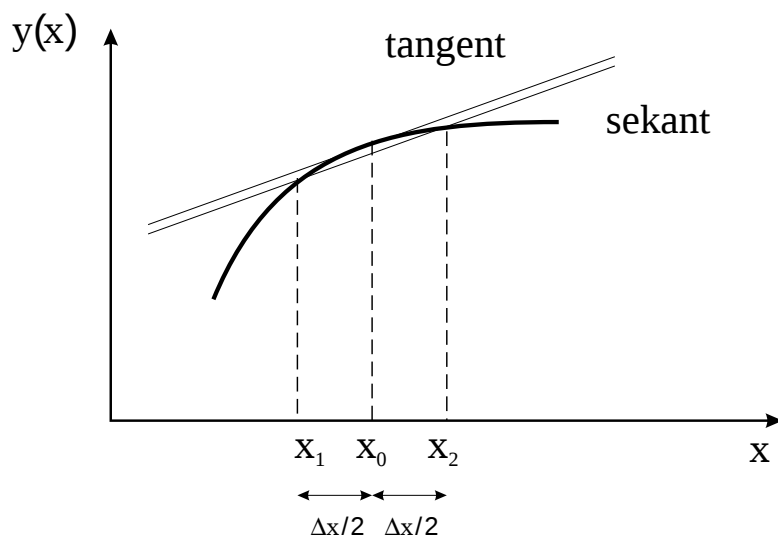
Diffusivitetstiligningen kan løses analytisk kun for enkle geometrier og grensevilkår, og med visse forenklinger. Et eksempel på dette er gitt i en av øvingene. I det alminnelige tilfellet må en begrense seg til numeriske løsninger, og for denne typen partielle differensialligninger finnes det to hovedklasser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



Figur 3.1: Tilnærming av tangent ved sekant.

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

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

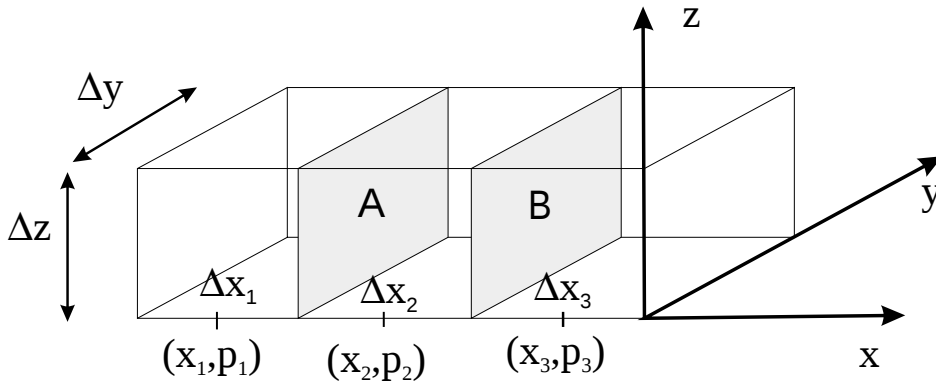
og den tilnærmes numerisk ved hjelp av differanser. Det vil si at vi tilnærmer tangentens stigningsforhold med sekantens på følgende måte

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

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

3.2 Avstandsderivate i diffusivitetstiligningen

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 trykløsningen dersom det brukes tidsmidlele 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

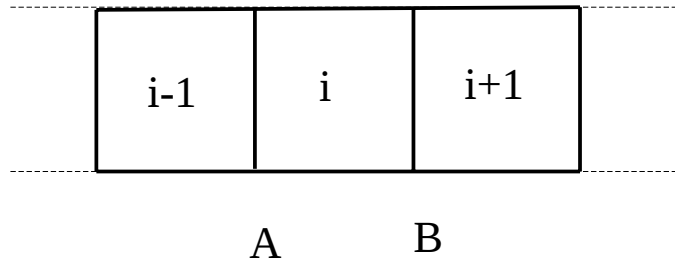
$$[b]_2^+ = \frac{1}{4}(b(p(x_2, t + \Delta t)) + b(p(x_3, t + \Delta t)) + b(p(x_2, t)) + b(p(x_3, t))).$$

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 diffusivitetstiligningen 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

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

og Ligning 2.5 skrives som

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

Vi bruker differansemetoden til å diskretisere Ligning 3.1, høyre siden i romlig x -retning, og venstre siden i tid, og får¹

$$\frac{Y_i|_B - Y_i|_A}{\Delta x_i} + A9_i = A2_i \frac{p_i(t + \Delta t) - p_i(t)}{\Delta t}. \quad \dots \dots \dots (3.2)$$

Videre tilnærmes

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

og tilsvarende for det andre leddet,

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

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

$$\boxed{O_{xi}^+ \cdot \Delta p_i^+ + O_{xi}^- \cdot \Delta p_i^- + A9_i = A2_i(p_i(t + \Delta t) - p_i(t))/\Delta t.} \quad \dots \dots (3.3)$$

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 tidsstegsløype 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 tidsderivate. 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. 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.

¹For 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.

Numerisk formulering	Tidsnivå 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.

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 $w \rightarrow 0$ når $(\Delta x, \Delta t) \rightarrow 0$, sier vi at den numeriske formuleringen *konvergerer*. Dersom differensialligningen i numerisk formulering går mot den opprinnelige differensialligning når $(\Delta x, \Delta t) \rightarrow 0$, 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; $w \sim \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x^2)$; 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; $w \sim \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x^2)$

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; $w \sim \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2)$

Merk: Disse uttrykkene for diskretiseringsfeil forutsetter at Δt og Δx er dimensjonsløse, det vil si at diffusivitetstiligningen 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 ligningen er $p_i(t + \Delta t)$ som vi kan finne direkte:

$$p_i(t + \Delta t) = \frac{\Delta t}{A2_i} (O_{xi}^+ \cdot \Delta p_i^+(t) + O_{xi}^- \cdot \Delta p_i^-(t) + A9_i) + p_i(t). \quad \dots \quad (3.4)$$

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

$$O_{xi}^+(p_{i+1} - p_i) + O_{xi}^-(p_{i-1} - p_i) + A9_i = \frac{A2_i}{\Delta t} (p_i - p_i(t)),$$

som rearrangert gir

$$O_{xi}^- p_{i-1} + (-O_{xi}^- - O_{xi}^+ - \frac{A2_i}{\Delta t}) p_i + O_{xi}^+ p_{i+1} = -\frac{A2_i}{\Delta t} p_i(t) - A9_i.$$

Vi definerer følgende hjelpestørrelser

$$\begin{aligned} B_i &= O_{xi}^- + O_{xi}^+ + \frac{A2_i}{\Delta t}, \\ A_i &= O_{xi}^- / B_i, \\ C_i &= O_{xi}^+ / B_i, \\ D_i &= (\frac{A2_i}{\Delta t} p_i(t) + A9_i) / B_i, \end{aligned}$$

og simuleringsligningen blir

$$\boxed{-A_i p_{i-1} + p_i - C_i p_{i+1} = D_i.} \quad \dots \quad (3.5)$$

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:

$$\begin{aligned} 0 + p_1 - C_1 p_2 &= D_1, \\ -A_2 p_1 + p_2 - C_2 p_3 &= D_2, \\ -A_3 p_2 + p_3 - C_3 p_4 &= D_3, \\ -A_4 p_3 + p_4 + 0 &= D_4, \end{aligned} \quad \dots \quad (3.6)$$

som kan skrives som en matriseligning

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

eller

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

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

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

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 endimensjonale 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:

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

eller

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

La k betegne iterasjonstalleren. 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 reproduisert nedenfor. Iterasjonssløyfen kalles `itera`, sløyfen for løsning 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 fjerde ligning.

Sammenligner en med ligningen på formen 3.7, så har vi nå endret matriseelementer i $\vec{K}$ 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 $\vec{K}$, inneholder 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 reproduisert under. I line 04 betegner a1 diagonalledet; videre er s den endrete høyreside av ligningene.

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

Subrutine for løsning av lineært ligningssett med tridiagonal koeffisientmatrise

```

subroutine tridia(mx)
  use felles_matriser
  implicit none

  integer :: i, mx, k

  a1(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 TIDSSTEGSSLIYFE

Oppdater trykk, POLD = P0

β START YTRE ITERASJON, på ulineære ledd, dvs. b , μ

PEST = P0(1)

P0 = (P0 + 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 $\mathcal{E} = \|P0(1) - PEST\|$

β' SLUTT Dersom $\mathcal{E} \geq \epsilon$: 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**(12-1)), hvor 1 er
!           antall tidssteg, days
!   eps     trykktoleranse, iterasjon på ulineære ledd _og_ i iter-
ativ
!           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 writ-
ten
!   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 tolerance, 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
!
! *****
! *
! *****
!
! *****
! *
! *****
!
!
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

```
18,4,1,1,1,0          MX,STMAX,ISOL,IPE,ISKIP,IWRITE
100,.05,3,2000,.7889   KX,PHI,DEX,PORIG,BORIG
1.358,8.03D-06,.002,0.001,1 VISORG,DBDP,DELT1,EPS,DEZ
10                     DEY
-1,18*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 ..... : 18
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 ..... : 3.00
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 tolerance, 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 .... 0.0
block no. : 7      oil rate, qo, stb/d .... 0.0
block no. : 8      oil rate, qo, stb/d .... 0.0
block no. : 9      oil rate, qo, stb/d .... 0.0
block no. : 10     oil rate, qo, stb/d .... 0.0
block no. : 11     oil rate, qo, stb/d .... 0.0
block no. : 12     oil rate, qo, stb/d .... 0.0
block no. : 13     oil rate, qo, stb/d .... 0.0
block no. : 14     oil rate, qo, stb/d .... 0.0
block no. : 15     oil rate, qo, stb/d .... 0.0
block no. : 16     oil rate, qo, stb/d .... 0.0
block no. : 17     oil rate, qo, stb/d .... 0.0
block no. : 18     oil rate, qo, stb/d .... 0.0

```

```

block no. : 1      pressure point, x(i), ft 1.5
block no. : 2      pressure point, x(i), ft 4.5
block no. : 3      pressure point, x(i), ft 7.5
block no. : 4      pressure point, x(i), ft 10.5
block no. : 5      pressure point, x(i), ft 13.5
block no. : 6      pressure point, x(i), ft 16.5
block no. : 7      pressure point, x(i), ft 19.5
block no. : 8      pressure point, x(i), ft 22.5
block no. : 9      pressure point, x(i), ft 25.5
block no. : 10     pressure point, x(i), ft 28.5
block no. : 11     pressure point, x(i), ft 31.5
block no. : 12     pressure point, x(i), ft 34.5
block no. : 13     pressure point, x(i), ft 37.5
block no. : 14     pressure point, x(i), ft 40.5
block no. : 15     pressure point, x(i), ft 43.5
block no. : 16     pressure point, x(i), ft 46.5

```

```

block no. : 17      pressure point, x(i), ft      49.5
block no. : 18      pressure point, x(i), ft      52.5

```

```

block no. : 1      oil pressure, psia .....      1994.900
block no. : 2      oil pressure, psia .....      1997.313
block no. : 3      oil pressure, psia .....      1998.585
block no. : 4      oil pressure, psia .....      1999.254
block no. : 5      oil pressure, psia .....      1999.607
block no. : 6      oil pressure, psia .....      1999.793
block no. : 7      oil pressure, psia .....      1999.891
block no. : 8      oil pressure, psia .....      1999.943
block no. : 9      oil pressure, psia .....      1999.970
block no. : 10     oil pressure, psia .....      1999.984
block no. : 11     oil pressure, psia .....      1999.992
block no. : 12     oil pressure, psia .....      1999.996
block no. : 13     oil pressure, psia .....      1999.998
block no. : 14     oil pressure, psia .....      1999.999
block no. : 15     oil pressure, psia .....      1999.999
block no. : 16     oil pressure, psia .....      2000.000
block no. : 17     oil pressure, psia .....      2000.000
block no. : 18     oil pressure, psia .....      2000.000
steps= 1          time= 0.2312D-04      np= 0.2312D-04
delt= 0.2312D-04  cum cycle= 2          ni= 0.0000D+00
nt= 0.3794D+01   qotot= 0.1000D+01      cum oil error= 0.1171D-13
npc= 0.6095D-03

```

```

block no. : 1      oil pressure, psia .....      1990.354
block no. : 2      oil pressure, psia .....      1993.648
block no. : 3      oil pressure, psia .....      1995.903
block no. : 4      oil pressure, psia .....      1997.398
block no. : 5      oil pressure, psia .....      1998.367
block no. : 6      oil pressure, psia .....      1998.985
block no. : 7      oil pressure, psia .....      1999.374
block no. : 8      oil pressure, psia .....      1999.616
block no. : 9      oil pressure, psia .....      1999.766
block no. : 10     oil pressure, psia .....      1999.858
block no. : 11     oil pressure, psia .....      1999.914
block no. : 12     oil pressure, psia .....      1999.948
block no. : 13     oil pressure, psia .....      1999.969
block no. : 14     oil pressure, psia .....      1999.981
block no. : 15     oil pressure, psia .....      1999.988
block no. : 16     oil pressure, psia .....      1999.993
block no. : 17     oil pressure, psia .....      1999.995
block no. : 18     oil pressure, psia .....      1999.996
steps= 2          time= 0.5781D-04      np= 0.5781D-04
delt= 0.3468D-04  cum cycle= 4          ni= 0.0000D+00
nt= 0.3794D+01   qotot= 0.1000D+01      cum oil error= 0.3512D-13
npc= 0.1524D-02

```

```

block no. : 1      oil pressure, psia ..... 1985.653
block no. : 2      oil pressure, psia ..... 1989.346
block no. : 3      oil pressure, psia ..... 1992.227
block no. : 4      oil pressure, psia ..... 1994.413
block no. : 5      oil pressure, psia ..... 1996.036
block no. : 6      oil pressure, psia ..... 1997.217
block no. : 7      oil pressure, psia ..... 1998.065
block no. : 8      oil pressure, psia ..... 1998.665
block no. : 9      oil pressure, psia ..... 1999.085
block no. : 10     oil pressure, psia ..... 1999.376
block no. : 11     oil pressure, psia ..... 1999.577
block no. : 12     oil pressure, psia ..... 1999.714
block no. : 13     oil pressure, psia ..... 1999.807
block no. : 14     oil pressure, psia ..... 1999.869
block no. : 15     oil pressure, psia ..... 1999.910
block no. : 16     oil pressure, psia ..... 1999.936
block no. : 17     oil pressure, psia ..... 1999.951
block no. : 18     oil pressure, psia ..... 1999.958
steps= 3          time= 0.1098D-03      np= 0.1098D-03
delt= 0.5202D-04  cum cycle= 6          ni= 0.0000D+00
nt= 0.3794D+01   qotot= 0.1000D+01      cum oil error= 0.2341D-13
npc= 0.2895D-02

```

```

block no. : 1      oil pressure, psia ..... 1980.404
block no. : 2      oil pressure, psia ..... 1984.325
block no. : 3      oil pressure, psia ..... 1987.614
block no. : 4      oil pressure, psia ..... 1990.321
block no. : 5      oil pressure, psia ..... 1992.513
block no. : 6      oil pressure, psia ..... 1994.260
block no. : 7      oil pressure, psia ..... 1995.636
block no. : 8      oil pressure, psia ..... 1996.705
block no. : 9      oil pressure, psia ..... 1997.528
block no. : 10     oil pressure, psia ..... 1998.155
block no. : 11     oil pressure, psia ..... 1998.628
block no. : 12     oil pressure, psia ..... 1998.981
block no. : 13     oil pressure, psia ..... 1999.242
block no. : 14     oil pressure, psia ..... 1999.432
block no. : 15     oil pressure, psia ..... 1999.567
block no. : 16     oil pressure, psia ..... 1999.659
block no. : 17     oil pressure, psia ..... 1999.716
block no. : 18     oil pressure, psia ..... 1999.743
steps= 4          time= 0.1879D-03      np= 0.1879D-03
delt= 0.7804D-04  cum cycle= 8          ni= 0.0000D+00
nt= 0.3794D+01   qotot= 0.1000D+01      cum oil error= 0.0000D+00
npc= 0.4952D-02

```

Del III

Øvingsoppgaver

Øving 1

Kjøring av utlevert program

1.1 Utprøving av løsningsrutiner

Bruk editoren til å sette MX i datafilen til 6 (blokker). 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 tidsstegene nr. 1, 5 og 10.

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 tidssteglengde

`delmax`: maksimum tidssteglengde

`dpmx`: maksimum trykkendring pr. tidssteg

`dtmult`: multiplikator på tidssteglengden

Sett $dpmx = 0 \cdot d0$ 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.
!   tidssteglengden 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,
z 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?

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

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

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

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

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

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

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 $\frac{\partial P}{\partial t} = 0$.
- 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) Innfor $p(x, t) = P(x, t) - p_s(x)$ og vis at ligning 3.3 blir

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

c) Vis at grensevilkårene nå er

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

$$\text{ii) } p(L, t) = 0,$$

$$\text{ii) } p(x, 0) = q(x - L).$$

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

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

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:

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

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

- e)** Lag et FORTRAN program som regner ut trykket $P(x, t)$ for et vilkårlig tryktpunkt 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:

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

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

$$p_s = a_1 x + a_2.$$

Fra Darcy's lov:

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

Dette gir

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

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

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

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

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

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 trykløsningen

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

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

Grensevilkår i)

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

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

som gir $b = 0$.

Grensevilkår ii)

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

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

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

Grensevilkår iii)

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

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

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

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

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

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

Vi multipliserer nå denne ligningen med

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

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

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

Vi ser først på venstre side av ligningen:

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

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

$$\begin{aligned} &q \int_0^{(2k+1)\pi} \frac{2L}{(2k+1)\pi} u \cos u \, du - q \underbrace{\int_0^{(2k+1)\pi} L \cos u \, du}_{=0} \\ &= \frac{2Lq}{(2k+1)\pi} \left[\cos u + u \sin u \right]_0^{(2k+1)\pi} = -\frac{4Lq}{(2k+1)\pi} \end{aligned}$$

Setter en så venstre side lik høyre side får en:

$$\begin{aligned} a_k \frac{(2k+1)\pi}{2} &= -\frac{4Lq}{(2k+1)\pi}, \\ a_k &= -\frac{8Lq}{[(2k+1)\pi]^2} \end{aligned}$$

Oppsummert gir dette

$$\begin{aligned} P(x, t) &= p_s(x) + p(x, t) = p_s(x) + \sum_{k=0}^{\infty} p_k \\ &= P_i - q(x-L) - 8Lq \sum_{k=0}^{\infty} \frac{\cos\left(\frac{(2k+1)\pi}{2L}x\right)}{((2k+1)\pi)^2} \exp\left(-\left(\frac{(2k+1)\pi}{2L}\right)^2 c_1^2 t\right). \end{aligned}$$

Øving 4

Numeriske formuleringer og SOR

4.1 Problemstilling

a) Modifisering

Modifiser programmet til E_4 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. Los ligningssettet med SOR, GSIT, TRIDIA

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

b) Datakjøring

Bruk data som i Øving 2 med følgende unntak: STMAX = 12, EPS = 0.001, og sammenlign transienttrykkene i blokk 1, dvs. plott $P0(1)$ mot $\ln t$ fra følgende kjøring:

1. Implisitt, Tridia
2. C-N, Tridia
3. Eksplisitt
4. Analytisk løsning $P0(5 \text{ ft}, t)$

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 følgende 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:

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

SOR betyr Suksessiv OverRelaksasjon,

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

I koden skiftes GSIT ut med SOR:

```
SUBROUTINE SOR(....., OM)
```

```
.
.
.   som for GSIT, men skift ligning til å inkludere  $\omega$ 
.
.
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
    IF (- - -) IM =
    IF (- - -) IP =
    PO = A · POLD + C · POLD + D
  end do
end select
```

For eksplisitt formulering. Beregner trykket i hver blokk uten løsningsrutine.

Fra for har en for en generell blokk i :

$$OX_i^+ \Delta P_i^+ + OX_i^- \Delta P_i^- + A9_i = A2_i (P_i(t + \Delta t) - P_i(t)) / \Delta t.$$

Dette ligningssettet ønsker en på formen

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

Eksplisitt formulering

$$OX_i^+ (P_{ip} - P_i) + OX_i^- (P_{im} - P_i) + A9_i = \frac{A2_i}{\Delta t} (P_i(t + \Delta t) - P_i),$$

hvor alle trykkene, bortsett fra $P_i(t + \Delta t)$ er ved "gammel" tid.

Ved å bruke $B = \Delta t / A2_i$

$$A_i = B * OX_i^-$$

$$C_i = B * OX_i^+$$

$$D_i = B * (A9_i - (OX_i^+ + OX_i^-) * POLD_i) + POLD_i,$$

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)

$$b = ox^- + ox^+ + a2_i / \Delta t$$

$$a = ox^- / b$$

$$c = ox^+ / b$$

$$d = (a2_i / \Delta t * pold + a9_i) / b$$

C-N:

im

ip

$$b = ox^- + ox^+ + 2 * a2 / \Delta t$$

$$a = ox^- / b$$

$$c = ox^+ / b$$

$$d = (ox^+ * (pold_{ip} - pold_i) + ox^- * (pold_{im} - pold_i) + 2 * a9_i + 2 * a2_i / \Delta t * pold_i) / b$$

Med det foreslåtte valg av ICRN og ISOL, med $ISOL = 3$ når $ICRN = 3$.

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

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

$$ICRN = 3, ISOL = 3.$$

Øving 5

Variabel k , ϕ og Δx

5.1 Oppgave

Modifiser programmet til å inkludere varierende horisontal *permeabilitet*, *porositet* og *blokk lengde*. 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, $STMAX \approx 20$.

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 blokk lengder på: 10, 5, 5, 20, 20, 5, 5, 10 ft.

Dersom programmeringen er utført riktig, skal begge tilfellene gi samme trykfall, $\Delta P = PO(8) - PO(1)$, ved stasjonær tilstand.

5.2 Kommentarer

Det skal legges inn variabel permeabilitet, porositet og blokk lengde. Dette gjøres for bedre å kunne tilpasse modellen til reservoardata.

F. eks.— Anta at en skal simulere trykkendringene rundt en produksjonsbrønn. (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 blokkindeling i de områder hvor det skjer størst trykkendringer, altså nærmest brønnen, for å få optimal nøyaktighet. En tommelfinger-regel er at trykkfallet per blokk skal være likt ved stasjonær tilstand.

La $i_p = i+1$ og $i_m = i-1$. Fra for har en

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

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

Her er $\overline{(b/\mu)}$ 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 $k_i^\pm$ middelveiden 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

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

og

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

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

Tips til koding

- 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 bruke sloyfer med do-setninger.
- 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

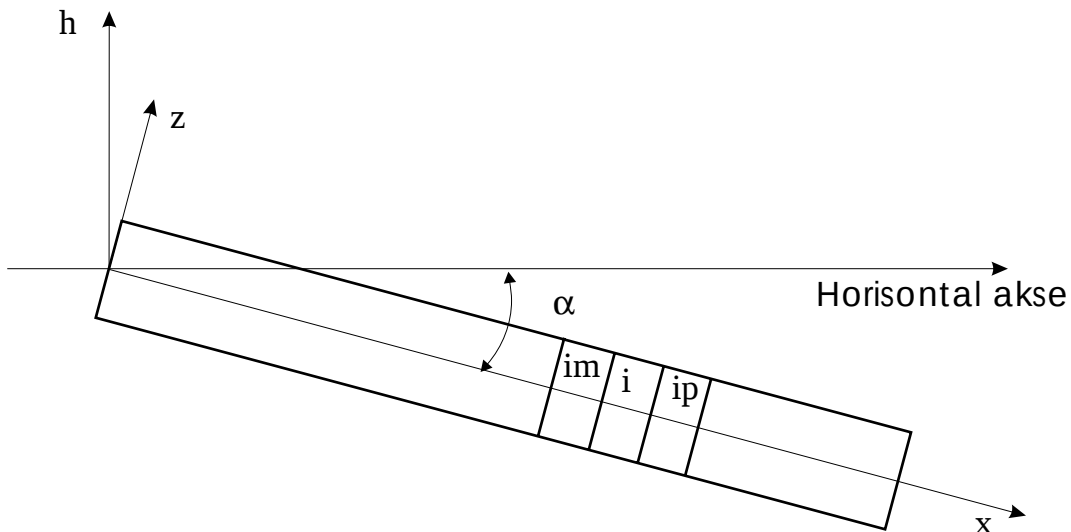
Utvid programmet til å inkludere helning. Bruk data som i øving 5, med $\alpha = +45^\circ$ og $\alpha = -45^\circ$, $RHOSTD = 0.38 \text{ psi/ft}$. Kjør programmet til stasjonær tilstand og kontroller svaret ved å sammenlikne med Darcy's lov.

6.2 Kommentarer

Simuleringsligningen for et horisontalt system er

$$\frac{\partial}{\partial x} \left(\frac{Ck_o b_o}{\mu_o} \frac{\partial P_o}{\partial x} \right) + A_9 = A_2 \frac{\partial P_o}{\partial t}.$$

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 høydenivået settes lik null i trykkpunktet til første blokk. I figuren regnes α negativ, slik at $\sin \alpha = dh/dx$ er negativ. Trykket i



Figur 6.1: Endimensjonalt reservoar med helning

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

gho: Gravity Head Oil

$gho_i^-: (\rho_{im} \cdot dex_{im} + \rho_i \cdot dex_i) \cdot 0.5 \cdot dhdx$

$gho_i^+: gho_i^-$

$dhdx: dh/dx$

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

$\rho_o: \rho_o$.

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

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

Forslag til koding

- gho(20) deklarereres

- Inne i flprop, etter at boij er regnet ut:

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

- I flocon: $a9_i = qo_i + gho_{ip} \cdot ox_i^+ - gho_i \cdot ox_i^-$

- 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 for.

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.

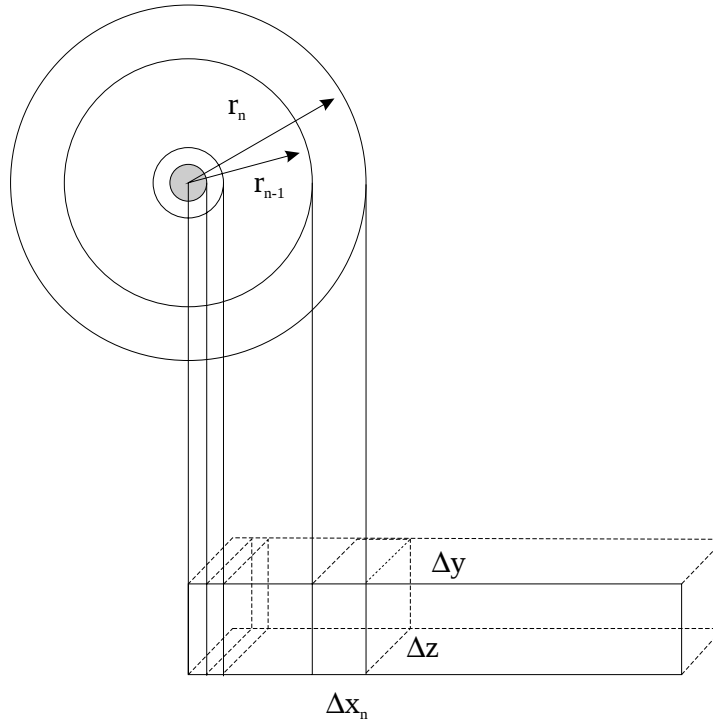
NB: Øvingen er obligatorisk, og følgende skal innleveres for godkjenning:

- 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 transformere ring n med bredde $r_n - r_{n-1}$ og 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.

Transformasjonen 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 virkelige, 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

$$\phi_l = \frac{\pi(r_n^2 - r_{n-1}^2)}{\Delta z}\phi_r. \quad \dots \dots \dots (7.1)$$

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

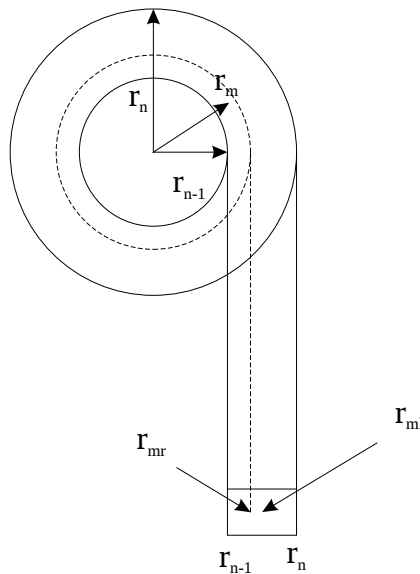
$$k_{xl} = \frac{2\pi\Delta x}{\Delta z \ln\left(\frac{r_n}{r_{n-1}}\right)} k_{xr}. \quad \dots\dots\dots (7.2)$$

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:

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

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



Figur 7.2: Posisjonen til middeltrykk i radielt system.

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

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

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

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

```

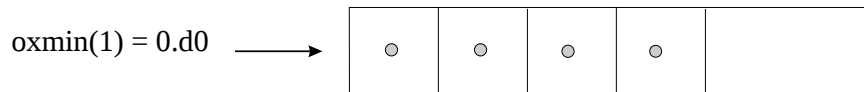
Dette, som er alt som trengs for å simulere et radielt system, kan utføres i hovedprogrammet for tidsstegsløyfen starter.

Kommentar til spørsmål 1b

Fra før har vi ligningen

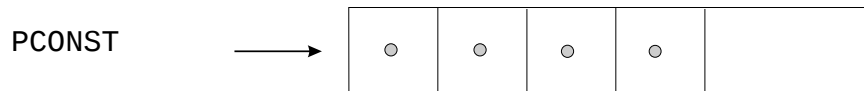
$$ox_i^+ \Delta p_i^+ + ox_i^- \Delta p_i^- + a9_i = h.s.,$$

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



Figur 7.3: Ingen strøm ut av venstre endeflate.

st ved utløpet av blokk 1, antydnet i Figur 7.4. Dette gjøres for å simulere mer realistiske



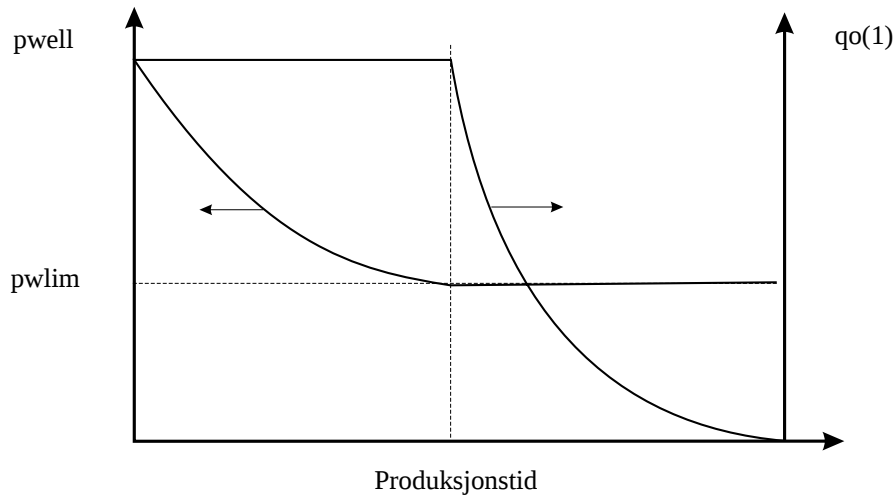
Figur 7.4: Produksjon mot konstant trykk.

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 brønnen, p_{well} , og raten er vist i Figur 7.5. I figuren er p_{well} bunnhullstrykket slik modellen vår beregner det, og p_{wlim} er en innlest verdi for minimum bunnhullstrykk. I programkoden foreslås det å bruke størrelsen p_{const} for bunnhullstrykket, slik at $p_{const} = 0$. når brønnen produserer med konstant rate, og $p_{const} = p_{wlim}$ når brønnen produserer mot konstant trykk.

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

$p_{const} \leq 0$. En produserer med konstant rate fra blokk nr. 1 og trykket ved utløpet av blokk 1, p_{well} , regnes ut for hvert tidssteg fra ligningen

$$qo(1) = ox_1^- * (p_{well} - po(1)), \quad \dots \dots \dots (7.5)$$



Figur 7.5: Overgang fra konstant rate til konstant trykk.

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 trykløsningen 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 `pwlím` settes `pconst = pwlím` 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 for
d(1) : som for
```

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 trykløsningen 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 do 6070 l = 1, stmax, men for 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 tidssteglengden 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

$$b_o = b_{orig} + dbdp * \Delta p,$$

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) $p_{\max} = p_{npvt}$, $npvt$ og antar at $p_1 < p_2 < \dots < p_{npvt}$ 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:

$pv(1) = p$, $pv(2) = b$, $pv(3) = \mu$.

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:

$$\begin{aligned}
 in &= 2 + (tpo(1) - p_o)/pinc \\
 &= 2 + (4000 - 3920)/50 \\
 &= 2 + 80/50 = 3
 \end{aligned}$$

Deretter interpoleres det lineært i tabellen mellom in og inm :

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

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

$$dbodp(i) = (boij(i) - bold(i))/(po(i) - pold(i)),$$

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) = boij(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)`
- Iflocon
 - `oxmin(i)=ckx(i)*(boij(i)/vo(i)+boij(im)/vo(im))*0.5d0`
- Iflprop

```

do i=1,mx
  in=2+(tpo(1)-po(i))/pinc
  inm=in-1
  zmc=(po(i)-tpo(in))/pinc
  boij(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)=(boij(i)-bold(i))/dp
    end if
    boij(i)=(boij(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 $m_x = 1$ eller $m_y = 1$
- 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 i orden
 - 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

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

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:

$$\begin{array}{ccccc} & & (i,jp) & & \\ (im,j) & & (i,j) & & (ip,j) \\ & & (i,jm) & & \end{array}$$

Setter en inn i ligning 9.1 for trykkdifferansene får en

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

Samler en nå ledd med like trykk får en

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

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

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

og ligning 9.1 blir

$$a(i, j)p(i_m, j) + c(i, j)p(i_p, j) - p(i, j) + e(i, j)p(i, j_m) + f(i, j)p(i, j_p) = -d(i, j). \quad (9.3)$$

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

- Naturlig nummerering,
- Alternierende diagonal nummerering.

La oss først se på naturlig nummerering, og vi bruker som eksempel et 5*4 blokkssystem, 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 $ox^\pm$, $oy^\pm$ er satt lik null. Fra ligning 9.3 får en da det lineære ligningssystemet

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

hvor koeffisientmatrisen $\vec{K}$ 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 $\vec{K}$ 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 $\vec{K}$ 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						$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$		$\oplus$			
14		X	X		X	X					$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$		$\oplus$	$\oplus$		
15			X	X		X	X				$\oplus$	$\oplus$		$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
16				X			X	X				$\oplus$		$\oplus$	$\oplus$	$\oplus$		$\oplus$	$\oplus$	$\oplus$
17					X	X			X				$\oplus$	$\oplus$	$\oplus$		$\oplus$	$\oplus$	$\oplus$	$\oplus$
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

```

      subroutine 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 reproduisert 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:

$$a p_{im,j} + c p_{ip,j} - p_{i,j} + e p_{i,jm} + f p_{i,jp} = -d.$$

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

$$a p_{im,j} - p_{i,j} + c p_{ip,j} = -e p_{i,jm} - f p_{i,jp} - d.$$

For $p_{i,jm}$ og $p_{i,jp}$ 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 deklarerer, 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 to dimensjoner

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:

- $m_x = 20$ —bruk lik blokk lengde
- Lengde 2000 ft, tverrsnitt $625 \times 40 \text{ ft}^2$
- $\phi = 0.18$
- Initiell vannmetning $s_{wi} = 0.20$
- $B_{oi} = 1.3 \text{ rb/stb}$, $c_o = 8.0 \times 10^{-6} \text{ psi}^{-1}$
- $B_{wi} = 1.0 \text{ rb/stb}$, $c_w = 3.0 \times 10^{-6} \text{ psi}^{-1}$
- $\mu_o = 5.0 \text{ cp}$, konstant
- $\mu_w = 0.5 \text{ cp}$, 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 $p_{\text{const}} = p_i$ ved utløpet av blokk nr. 1
- Initielt trykk $p_i = 2000$ psia

Simuler vanndrivnet 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

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

for oljeligningen, og for vannligningen

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

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:

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

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:

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

10.2 IMPES-metoden

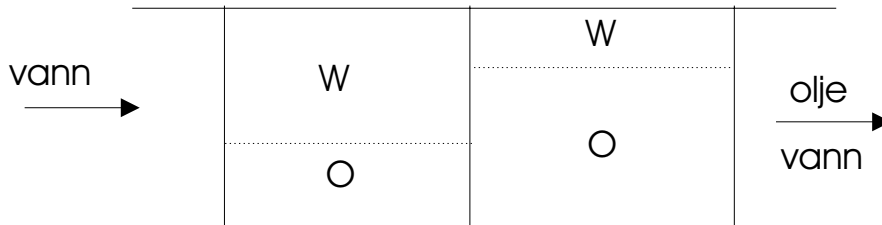
Betegnelsen *IMPES* er et akronym for *Implicit Pressure Explicit Saturation*. De metningsavhengige størrelsene, k_{ro}, k_{rw}, p_c , holdes fast på tidsnivå n , mens en løser trykket implisitt for tidsnivå $n + 1$. Siden $\Delta s_w = -\Delta s_o$, elimineres Δs_o på høyre side av ligningen og en står 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 illustrerer et vanddriv



Figur 10.1: Illustrasjon av oppstrøms relative permeabiliteter

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 aritmetisk 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 u fysisk og det foreslår 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 $i \rightarrow ip$ eller $ip \rightarrow i$. 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 $ckx_{ip} \cdot F(p_{oi}, p_{oip}, s_{oi}, s_{oip})$, hvor 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 Δp_{oi}^{k+1} , Δs_{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 Δp_{oi}^{k+1} , Δs_{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} :

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

Her er

$$\begin{aligned} k_{ro}^{n+1,k} & \text{ verdien av } k_{ro} \text{ etter iterasjon } k, \\ s_o^{n+1,k} & \text{ verdien av } s_o^{n+1} \text{ etter iterasjon } k, \\ \Delta s_o^{n+1,k+1} & = s_o^{n+1,k+1} - s_o^n. \end{aligned}$$

Når løsning har konverget 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:

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

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

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

hvor

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

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

Videre definerer vi for strømningsledd mellom blokk i og ip :

$$\begin{aligned} \text{oxldp}_i &= \text{bovo}_{ip} * (\text{fdo}_i * \text{krold}_i + (1. - \text{fdo}_i) * \text{krold}_{ip}) \\ \text{doxpp}_i &= \text{bovo}_{ip} * (1. - \text{fdo}_i) * \text{dkods}_{ip} \\ \text{doxpm}_i &= \text{bovo}_{ip} * \text{fdo}_i * \text{dkods}_i \\ \text{pop}_i &= \text{po}_{ip}^k - \text{po}_i^k \end{aligned}$$

For strøm mellom blokk im og i definerer vi:

$$\begin{aligned} \text{oxld}_i &= \text{bovo}_i * (\text{fdo}_{im} * \text{krold}_{im} + (1. - \text{fdo}_{im}) * \text{krold}_i) \\ \text{doxmm}_i &= \text{bovo}_i * \text{fdo}_{im} * \text{dkods}_{im} \\ \text{doxmp}_i &= \text{bovo}_i * (1. - \text{fdo}_{im}) * \text{dkods}_i \\ \text{pom}_i &= \text{po}_{ip}^k - \text{po}_i^k. \end{aligned}$$

Siden

$$\begin{aligned} \text{oxldp}_i &= \text{oxld}_{ip} * \text{dexm} \\ \text{doxpp}_i &= \text{doxmp}_{ip} * \text{dexm} \\ \text{doxpm}_i &= \text{doxmm}_{ip} * \text{dexm} \end{aligned}$$

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 $\partial p_c / \partial s_o = \text{dpcds}$, $\partial k_{rw} / \partial s_o = \text{dkwds}$ og er nå istand til å skrive ut kapillarleddet i detalj:

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

og

$$\begin{aligned} \text{wx}^+ * \Delta p_c^+ &= (\text{wxldm}_{ip} * \text{dexm} + \text{dwxmp}_{ip} * \text{dexm} * \Delta s_{oip} + \text{dwxmm}_{ip} * \text{dexm} * \Delta s_{oi}) \\ &= * (\text{pcld}_{ip} + \text{dpcds}_{ip} * \Delta s_{oip} - \text{pcld}_i - \text{dpcds}_i * \Delta s_{oi}) \end{aligned}$$

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

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

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

$$\Delta b_w = \frac{\partial b_w}{\partial p_w} \cdot \Delta p_w \rightarrow \frac{\partial b_w}{\partial p_w} (\text{po} - \text{pold} - \text{dpcds} * \Delta s_o).$$

10.6 Vannligningen i detalj — oppsummering

Fullstendig diskretisering av vannligningen 10.2 gir:

$$wx_i^- * \Delta p_{wi}^- + wx_i^+ * \Delta p_{wi}^+ + q_{wi} = \phi / \Delta t (\bar{b}_{wi} * \Delta s_{wi} + \bar{s}_{wi} * \Delta b_{wi}),$$

med følgende definisjoner:

$$\begin{aligned} wx_i^- &= wxldm_i + dwxmm_i * \Delta s_{oim} + dwxmp_i * \Delta s_{oi} \\ wx_i^+ &= (wxldm_{ip} + dwxmm_{ip} * \Delta s_{oi} + dwxmp_{ip} * \Delta s_{oip}) * dextr \\ \Delta p_{wi}^- &= p_{wim} - p_{wi} = p_{oim} - p_{oi} - p_{cim} + p_{ci} \\ \Delta p_{wi}^+ &= p_{wip} - p_{wi} = p_{oip} - p_{oi} - p_{cip} + p_{ci} \\ pom &= p_{oim} - p_{oi} \\ pop &= p_{oip} - p_{oi} \end{aligned}$$

Vi velger som sagt å bruke p_o og $\Delta s_o = s_o^{n+1} - s_o^n$ som ukjente. Leddet $wx_i^- * \Delta p_{wi}^-$ blir da

$$\begin{aligned} wxldm_i * \Delta p_o^- &= wxldm_i * pcld_{im} + dwxmm_i * pom * \Delta s_{oim} + dwxmp_i * pom * \Delta s_{oi} \\ &+ wxldm_i * pcld_{im} - wxld_i * dpcds_{im} * \Delta s_{oim} - dwxmp_i * pcld_{im} * \Delta s_{oi} \\ &- dwxmm_i * pcld_{im} * \Delta s_{oim} + wxldm_i * dpcds_i * \Delta s_{oi} \\ &+ wxld_i * pcld_i * \Delta s_{oim} + dwxmp_i * pcld_i * \Delta s_{oi} \end{aligned}$$

Vi ønsker å skrive ligningen på formen

$$apow * p_{oim} + asow * dso_{im} + bpow * p_{oi} + bsow * dso_i + cpow * p_{oip} + csow * dso_{ip} + dw = 0.$$

Med

$$\begin{aligned} pwm &= pom + pcld_i - pcld_{im} \\ pwp &= pop + pcld_i - pcld_{im} \end{aligned}$$

får en

$$\begin{aligned} apow &= wxldm_i \\ asow &= dwxmm_i * pwm - wxldm_i * dpcds_{im} \\ bpow &= -(wxldm_i + wxldm_{ip} * dextr) - \phi / \Delta t * \bar{s}_{wi} * dbwdp_i \\ bsow &= dwxmp_i * pwm + dwxmm_{ip} * dextr * pwp + (wxldm_i + wxldm_{ip} * dextr) * dpcds_i \\ &+ \phi / \Delta t * (\bar{s}_{wi} * dbwdp_i * dpcds_i + \bar{b}_{wi}) \\ cpow &= wxldm_{ip} * dextr \\ csow &= (dwxmp_{ip} * pwp - wxldm_{ip} * dpcds_{ip}) * dextr \\ dw &= wxldm_i * (pcld_i - pcld_{im}) + wxldm_{ip} * dextr * (pcld_i - pcld_{ip}) \\ &+ \phi / \Delta t * \bar{s}_{wi} * dbwdp_i * pold_i + q_{wi}. \end{aligned}$$

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

$$\begin{aligned} apoo &= oxldm_i \\ asoo &= doxmm_i * pom \\ bpoo &= -(oxldm_i + oxldm_{ip} * dextr) - \phi / \Delta t * \bar{s}_{oi} * dbodp_i \\ cpoo &= oxldm_{ip} * dextr \\ csow &= doxmp_{ip} * pop * dextr \\ do &= \phi_i / \Delta t * \bar{s}_{oi} * dbodp_i * pold_i + q_{oi}. \end{aligned}$$

<i>Ukjente</i> → <i>Lign.</i> ↓		<i>blokk1</i>		<i>blokk2</i>		<i>blokk3</i>		<i>blokk4</i>		<i>blokk5</i>	
		p_{o1}	Δs_{o1}	p_{o2}	Δs_{o2}	p_{o3}	Δs_{o3}	p_{o4}	Δs_{o4}	...	...
<i>w</i>	1	bpow	bsow	cpow	csow						
<i>o</i>	2	bpoo	bsoo	cpoo	csoo	...					
<i>w</i>	3	apow	asow	bpow	bsow	cpow	csow				
<i>o</i>	4	apoo	asoo	bpoo	bsoo	cpoo	csoo	...			
<i>w</i>	5		...	apow	asow	bpow	bsow	cpow	csow		
<i>o</i>	6			apoo	asoo	bpoo	bsoo	cpoo	csoo	...	
...	...				...	...	...	...	...	...	...

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:

<i>Ukjente</i> → <i>Lign.</i> ↓		<i>blokk1</i>		<i>blokk2</i>		<i>blokk3</i>		<i>blokk4</i>		<i>blokk5</i>	
		p_{o1}	Δs_{o1}	p_{o2}	Δs_{o2}	p_{o3}	Δs_{o3}	p_{o4}	Δs_{o4}	...	...
<i>w</i>	1	bpow	bsow	cpow	csow						
<i>o</i>	2	bpoo	bsoo	cpoo	csoo	...					
<i>w</i>	3	apow	asow	bpow	bsow	cpow	csow				
<i>o</i>	4	apoo	asoo	bpoo	bsoo	cpoo	csoo	...			
<i>w</i>	5		...	apow	asow	bpow	bsow	cpow	csow		
<i>o</i>	6			apoo	asoo	bpoo	bsoo	cpoo	csoo	...	
...	...				...	...	...	...	...	...	...

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, p_{const} , ved utløpet av blokk nr. 1.:

$$\begin{aligned} q_{o1} &= bovo_1 * kro_1 * (p_{const} - po_1) \\ q_{w1} &= bw bw_1 * krw_1 * (p_{const} - po_1 + pc_1). \end{aligned}$$

Vi regner her at trykket i brønnen, p_{const} , er det samme for olje som for vann.

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

$$\begin{aligned} q_{o1} &= bovo_1 * kroid_1 * p_{const} + bovo_1 * dkods_1 * p_{const} * dso_1 \\ &\quad - bovo_1 * kroid_1 * po_1 - bovo_1 * dkods_1 * po_1 * dso_1. \end{aligned}$$

Dette kan vi videre skrive som

$$q_{o1} = q_{o1} + bqop * po_1 + bqos * dso_1, \quad \dots \dots \dots (10.4)$$

hvor

$$\begin{aligned} q_{o1} &= bovo_1 * kroid_1 * p_{const} \\ bqop &= -bovo_1 * kroid_1 \\ bqos &= bovo_1 * dkods_1 * (p_{const} - po_1), \end{aligned}$$

hvor p_{o1} er den sist iterert verdien.

Tilsvarende får en for vannraten:

$$q_{w1} = q_{w1} + b_{qwp} * p_{o1} + b_{qws} * d_{so1}, \quad \dots \dots \dots (10.5)$$

hvor

$$\begin{aligned} q_{w1} &= b_{wvw1} * k_{rwld1} * (p_{const} + p_{cld1}) \\ b_{qwp} &= -b_{wvw1} * k_{rwld1} \\ b_{qws} &= b_{wvw1} * d_{kws1} * (p_{const} - p_{o1}) + b_{wvw1} * (k_{rwld1} * d_{pcds1} + d_{kws1} * p_{cld1}). \end{aligned}$$

Uttrykkene for q_{o1} og q_{w1} settes inn for q_o og q_w i ligningene og fordeles på de riktige matriseelementene på følgende måte:

$$\begin{aligned} b_{qop} &\rightarrow b_{poo} \\ b_{qwp} &\rightarrow b_{pow} \\ b_{qos} &\rightarrow b_{soo} \\ b_{qws} &\rightarrow b_{sow}, \end{aligned}$$

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 q_{o1} eller q_{w1} blir positive, settes de lik 0. Vannraten q_{w1} 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 utføres følgende:

$$\begin{aligned} p_{old} &= p_o \\ b_{old} &= b_o \\ v_{old} &= v_o \\ b_{wld} &= b_w \\ v_{wld} &= v_w \\ p_{cld} &= p_c \\ k_{rwld} &= k_{rw} \\ k_{rold} &= k_{ro} \\ s_{old} &= s_o \end{aligned}$$

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

$$\begin{aligned} dp_{max} &= 5.0 \text{ psi} \\ ds_{max} &= 0.05 \end{aligned}$$

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

```
bo, vo, dbodp
bw, vw, dbwdp
```

- SUBROUTINE SAT

```
kro, dkods
kro, dkods
pc, dpcds
```

- SUBROUTINE FLODIR

```
fdoi = 1.0 dersom poi > poip
      = 0. dersom poip < poi
fdwi = 1.0 dersom pwi > pwip
      = 0. dersom pwip < pwi
```

Denne subrutinen kalles opp en gang etter 1. iterasjon i hvert tidssteg.

- SUBROUTINE TRANS

```
oxldm, wxldm
doxmp, dwxmp
doxmm, dwxmm
```

- SUBROUTINE RAT

Dersom pconst > 0. beregnes følgende størrelser:

```
qo, qw
bqop, bqwp
bqos, bqws
```

- SUBROUTINE FLOCON

```
apow      apoo
asow      asoo
:         :
bb(ic,1)  = apoo
bb(ic,2)  = asoo
:         :
cc(ic)    = -do
:         :
call solve
```

Løsningen returneres i CC-matrisen. Deretter settes $so(i) = sold(i) + dso(i)$.

- Sjekk for konvergens
- Når løsningen har konvertert 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 $\frac{d}{dt}k_{od} = 0$, $\frac{d}{dt}k_{ow} = 0$, $\frac{d}{dt}p_c = 0$ 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 $\rho g h_o$, $\rho g h_w$, må reservoarets ekvilibres før start av simuleringen.
 - Definer WOC der hvor $p_c = 0$.
 - 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 blokkinnndelingen 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 kapillarytrykk, endeeffekt og rate på relative permeabilitetskurver slik at disse kan bestemmes fra et fortregningseksperiment.

Det tas utgangspunkt i de to artiklene [5, 4]. Welge's metode blir ofte brukt til å bestemme k_{ro}, k_{rw} fra et fortregningseksperiment 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 pluggen, 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 kapillarykksgradienter.

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 fortregende fase i cp, q/A er strømningsrate per tverrsnittsareal med benevnningen 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 $p_o = p_w = p_{const}$ 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, irreduksibel 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

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

11.4 Forventede resultater

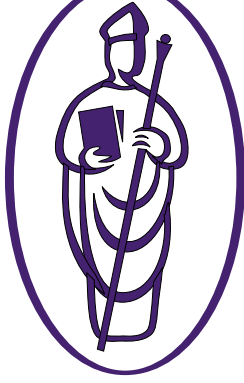
Trinn 1. i *Framgangsmåte* på side 76 vil antagelig gi relative permeabilitetskurver som ligger nær de innleste. En vil få noe avvik på grunn av numerisk dispersjon —vannet vil bre seg for raskt mot enden av pluggen.

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



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

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

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

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

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

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

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

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

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

e) Vis nå at (4) kan skrives som

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

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

Oppgave 2

Diffusivitetstiligningen (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.

Utledd simuleringsligningen

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

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

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

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/(m³s).

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

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

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 :

$$\frac{\partial \rho}{\partial x} = \rho c_f \frac{\partial p}{\partial x}. \quad \dots \dots \dots (8)$$

Samlet gir dette

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

d) Rekkeutvikler til 2. orden:

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

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

$$\frac{Ck b_r}{\mu} \frac{\partial^2 p}{\partial x^2} + \frac{q b_r}{\rho_r} = \phi c_f b_r \frac{\partial p}{\partial t}. \quad \dots \dots \dots (10)$$

La R betegne reservoarforhold, S overflateforhold, T tid, M masse og V volum. Da er $b_r = V_S/V_R(p_r)$, $q = M/T$, $\rho_r = M/V_R(p_r)$, slik at

$$\frac{q b_r}{\rho_r} : \frac{M V_S V_R}{M T V_R} = \frac{V_S}{T},$$

altså volumrate på overflaten.

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

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

og (10) skrives som

$$\frac{\partial Y}{\partial x} + \frac{q b_r}{\rho_r} = \phi c_f b_r \frac{\partial p}{\partial t}. \quad \dots \dots \dots (11)$$

Vi bruker differansemetoden til å diskretisere ligning 11, høyre siden i romlig x -retning, og venstre siden i tid, og får¹

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

Videre er

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

og tilsvarende for det andre leddet,

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

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). Innsatt i ligning 12 får vi da

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

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

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

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, $Y_1|_A = 0$ og $Y_l|_B = 0$, når N betegner siste blokk. For første blokk blir ligningen da

$$(T + \phi c_f b_r / \Delta t) p_1 + T p_2 = -(\phi c_f b_r p_1(t_n) / \Delta t + \frac{q_1 b_r}{\rho_r}),$$

og for siste blokk N

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

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

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

¹For 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 får ligning 12 på følgende numerisk form

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

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 denotes
the
c  subdiagonal elements; matrix c the superdiagonal, and matrix
d
c  the 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 subroutine 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 $q b_r / \rho_r$.

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 comment-
ed out
c      but included for reference. char(63) produces a ques-
tion 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 'ut-
fil'
c      would never be deleted.
c          open (5,file='inndat',status='old')
c          open (10,file='utdat')
c          close (10,status='delete')
c          open (10,file='utdat')
c
c      ** read and write input data.
c          read (5,*) imx, nmx
c          write (10,3010) imx, nmx
c
c          read (5,*) k, phi, dx, porig, b
c          read (5,*) vis, dbdp, dt, dz, dy
c          write (10,3020) k, phi, dx, porig, b, vis,
z          dbdp, dt, dz, dy
c          read (5,*) (qs(i),i = 1,imx)
c          write (10,3030) (i,qs(i), i = 1,imx)
c          write (10,3080)
c
c      ***begin calculations
c
c      **universal constants
c          conv = 0.00632827d0
c          fpb = 5.6146d0
c          en = 1.d0
c          to = 2.d0
c          pi = 3.1415926d0
c
c      **other constants
c          vol = dz*dy*dx/fpb
c          trans = conv*k*b/vis/dx/dx
c
c      **matrices a and c (note that a(1) and c(imx) are giv-
en values
c          that never are used while c(1) and a(imx) are differ-
ent 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 pres-
c      sures
              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 diskontinuiteter i strømningssegenskaper 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 (vilkårleg) blokkklengde i z -retning.

Trykkgradientene øker i x -retning inn mot brønnen og blokkklengdene 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 strømningssegenskaper. 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

$20 \times 165, 80, 57, 37, 12, 6, 4, 2, 10 \times 1$, (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.

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 diffusivitetstilnærmingen kan tilnærmes med

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

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

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

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 kant-blokkene.

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 blokklelgdene $\Delta x_i = r_{i+1} - r_i$ 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 diffusivitetstiligningen skrives som

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

Vi bruker differansemetoden til å diskretisere ligning 16, høyre siden i romlig x -retning, og venstre siden i tid, og får²

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

Videre er

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

og tilsvarende for det andre leddet,

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

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

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

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, $Y_1|_A = 0$ og $Y_l|_B = 0$, når N betegner siste blokk. For første blokk blir ligningen da

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

²For 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 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

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

og får ligning 17 på følgende numerisk form

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

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

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

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 irreduksibel. 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 koordinataksler 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 tidssteglengden gir en dominerende feil.

- c)** Ja, siden lagdelingen er ulik.

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

$$\frac{\partial}{\partial x}(\rho u) - q + \frac{\partial}{\partial t}(\phi \rho) = 0, \quad \dots \dots \dots (19)$$

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 (diffusivitetligningen) kan skrives på formen

$$k \frac{\partial}{\partial x} \left(\frac{\rho}{\mu} \frac{\partial p}{\partial x} \right) + q = \phi \frac{\partial \rho}{\partial t}. \quad \dots \dots \dots (20)$$

c) Tilstandsligningen for en reell gass er gitt ved

$$\rho = \frac{M}{RTZ(p)} p, \quad \dots \dots \dots (21)$$

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

$$c = \frac{1}{\rho} \left(\frac{\partial \rho}{\partial p} \right)_T. \quad \dots \dots \dots (22)$$

La pseudotrykket $m(p)$ være definert ved

$$m(p) = \int_0^p \frac{\rho(p')}{\mu(p')} dp'. \quad \dots \dots \dots (23)$$

Vis ved hjelp av ligningene (20)–(23) at diffusivitetligningen for en reell gass med konstant sammensetning kan skrives som

$$k \frac{\partial^2 m}{\partial x^2} + q = \phi \mu c \frac{\partial m}{\partial t}. \quad \dots \dots \dots (24)$$

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

$$a_i m_{i-1} - m_i + c_i m_{i+1} = -d_i. \quad \dots \dots \dots (25)$$

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 diffusivitetssligning 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 diffusivitetssligning for dette problemet:

$$\frac{\partial}{\partial x} \left[\frac{A(x)k\rho}{\mu} \left(\frac{\partial p}{\partial x} + \rho g \sin \alpha \right) \right] + A(x)q = A(x) \frac{\partial}{\partial t} (\phi \rho). \quad \dots \dots \dots (26)$$

b) Bruk ligning (26) til å skrive ned diffusivitetssligningen 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 blokkleddene $\Delta r_i = r_{i+1} - r_i$ 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

$$\begin{array}{rcl} 2x_1 & + & x_2 = 1, \\ x_1 & + & 2x_2 = 5, \end{array} \quad \dots \dots \dots (27)$$

skal brukes til å sammenligne tre iterative løsningsmetoder: (1) Gauss-Jordan, (2) Gauss-Seidel og (3) SOR.

a) Skriv ned iterasjonsskjemaet 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 $\omega = 1.15$.)

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

$$\frac{\partial}{\partial x} \left(\frac{C k_x b}{\mu} \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{C k_y b}{\mu} \frac{\partial p}{\partial y} \right) + q = \phi \frac{db}{dp} \frac{\partial p}{\partial t}, \quad \dots \dots \dots (28)$$

hvor C er en omregningsfaktor.

b) Diskretiser ligning 28 med implisitt formulering og vis at svaret kan skrives på formen

$$a_{i,j} p_{im,j} + c_{i,j} p_{ip,j} - p_{i,j} + e_{i,j} p_{i,jm} + f_{i,j} p_{i,jp} = d_{i,j}. \quad \dots \dots \dots (29)$$

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 fordel-er 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 OverRelaksasjoner (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 $t = 0$,

$$p(x, t) = p_i - q(x - L) - 8Lq \sum_{n=0}^{\infty} \frac{\cos(\frac{(2n+1)\pi}{2L}x)}{((2n+1)\pi)^2} \exp(-(\frac{(2n+1)\pi}{2L})^2 c_1^2 t). \quad (30)$$

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
- trykkløsning 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
- trykkløsning ved eliminasjon
- nummereringsskjema av blokker for løsningsrutinen
- fluidegenskaper
- maksimal tillatt trykkendring per tidssteg
- ny tidssteglengde
- iterasjonssløyfe på ulineære ledd
- innlesing av nye rater ved gitt tidspunkt
- initiell trykklikevekt
- koeffisienter til lineært ligningssett

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

Diffusivitetssligningen 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 kokepunktstrykket. Angi hvilke antagelser som eventuelt må gjøres.
- b) Utled den tilhørende diffusivitetssligning.
- c) Diskretiser diffusivitetssligningen 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) punktvís.
- 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?

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

$$\frac{\partial}{\partial x} \left(\frac{Ckb}{\mu} \frac{\partial p}{\partial x} \right) + q = \frac{\partial}{\partial t} (\phi b), \quad \dots \dots \dots (31)$$

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. Blokk lengden Δx og permeabilitet k skal være de samme for hver numeriske blokk. Ligningen kan da skrives på følgende numerisk form,

$$-a_i p_{i-1}^m + p_i^{n+1} - c_i p_{i+1}^m = d_i, \quad \dots \dots \dots (32)$$

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 (31) 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 (31) 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 (32) 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 eliminering, (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 (31) og ligning (32) dersom det skal simuleres et enfase, todimensjonalt problem.

b) Et enfase, todimensjonalt problem skal simuleres med 6, (2×3) , blokker med LSOR (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 (31) 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 $t = 0$,

$$p(x, t) = p_i - q(x - L) - 8Lq \sum_{n=0}^{\infty} \frac{\cos(\frac{(2n+1)\pi}{2L}x)}{((2n+1)\pi)^2} \exp(-(\frac{(2n+1)\pi}{2L})^2 c_1^2 t). \quad (33)$$

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.