

Innføring
i
RESERVOARSIMULERING

Svein M. Skjæveland
og
Jann-Rune Ursin

Høgskolen i Stavanger

August 2001

Innhold

I	Teoretisk grunnlag	1
1	Opplegg	2
1.1	Undervisningsopplegg	2
1.2	Faglig omfang	2
1.3	Litteratur	3
1.4	Fontkonvensjon	3
1.5	Fortranversjon	4
2	Diffusivitetstiligningen	5
2.1	Darcy's lov og massebevarelse	5
2.2	Generell diffusivitetstiligning	8
2.3	Kommentar til utledning av diffusivitetstiligningen	9
2.4	Diffusivitetstiligning for tre faser	9
3	Numerisk formulering	11
3.1	Tilnærming av deriverte	11
3.2	Avstandsderiverte i diffusivitetstiligningen	12
3.3	Diskretisering i tid og rom	14
3.4	Valg av numerisk formulering	15
3.5	Eksplisitt formulering	16
3.6	Implisitt formulering	16
3.7	Iterativ løsningsmetode	18
3.8	Direkte løsningsmetode	19
3.9	Skisse til flytskjema for utlevert program	21
II	Fortrankode for utlevert program	23
3.10	Module felles-matriser	33
3.11	Module nrtype	33
3.12	Datafil	33
3.13	Resultatfil	34

Del I

Teoretisk grunnlag

Kapittel 1

Opplegg

1.1 Undervisningsopplegg

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

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

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

På siden <http://www.ux.his.no/~s-skj/> vil en finne en peker til faget med blant annet *Siste Nytt*, eksamensoppgaver og utkast til noen løsninger, samt annen informasjon.

1.3 Litteratur

Som støttebok anbefales boken til Aziz og Settari [1]. Det kan ellers kanskje svare seg å ha tilgang på en håndbok i `fortran90` og eventuelt lærebøker i numeriske metoder, selv om dette neppe er nødvendig. Se ellers pekere på hjemmesiden til faget.

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. *Society of Petroleum Engineers* har stått for utbeidelsen av en monografi om reservoarsimulering. Redaktørene Mattax og Dalton [2] sammen med andre bidragsytere har gitt et godt sammendrag av hva reservoarsimulering dreier seg om.

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 `FORTTRAN` 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 .

1.5 Fortranversjon

Kurset gjennomføres i Fortran90 (F90) som har F77 som en ekte delmengde. Programvaren som blir brukt er Compaque Visual Fortran. Programmet har god hjelpefunksjon for språket generelt og for selve programbruken. Andre innføringsopplegg i F90 vil ligge utlagt på <http://www.ux.his.no/~s-skj/>. En bør være oppmerksom på at flere av kodeforslagene kan inneholde rester etter F77 og det er ikke sikkert at alle vil virke uten videre.

Kapittel 2

Diffusivitetstiligningen

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

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

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

2.1 Darcy's lov og massebevarelse

Darcy's lov kan uttrykkes på følgende måte:

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

hvor

C : Omregningskonstant med verdi 0.00632827 for de gitte enhetene.

u_o : volumhastigheten, ft/d.

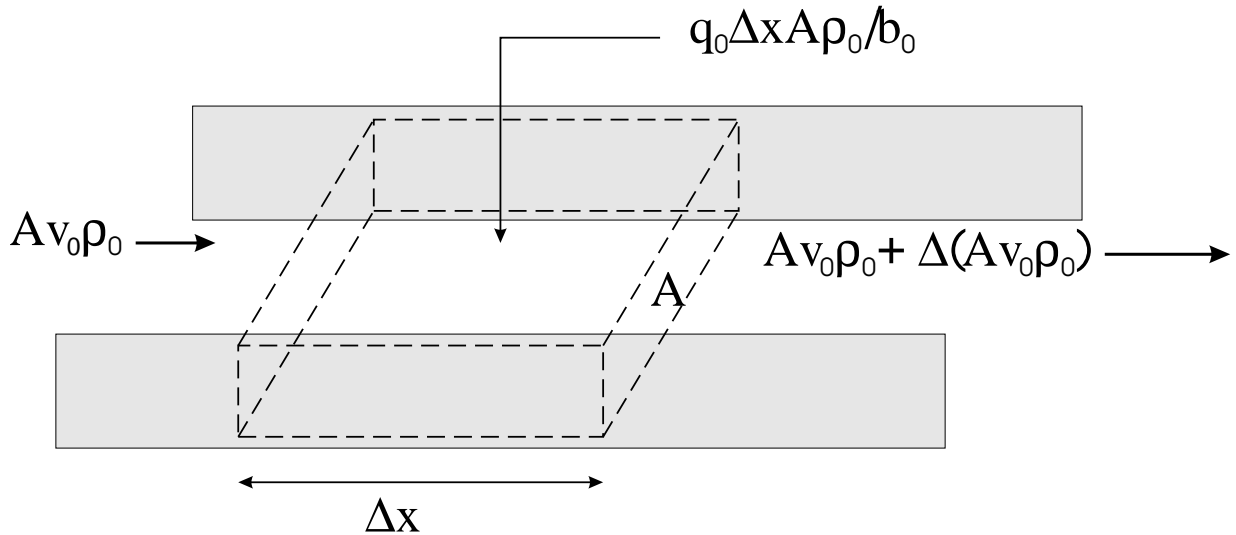
k_x : absolutt permeabilitet i x -retning, md.

μ_o : viskositet, cp.

p_o : trykk, psi.

x : avstand langs strømningsretningen, ft.

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



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

I Figur 2.1 er det brukt endel nye størrelser:

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

q_o : oljerate per bulkvolum, stb/d/bbl . Raten er positiv for injeksjon og negativ for produksjon. Dette er kildeleddet i kontinuitetsligningen [3].¹

b_o : volumfaktor for olje, stb/rb^2 .

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

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

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

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

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

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

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

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

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

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

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

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

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

som ordnet gir

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

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

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

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

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

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

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

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

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

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

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

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

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

hvor ρ_o^{st} er oljetettheten ved *stock tank* forhold i lb/ft³, ρ_g^{std} er gasstettheten ved *standard* forhold i lb/ft³, og R_{so} er oppløst gass-olje forhold i ft³/stb.

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

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

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

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

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

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

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

2.3 Kommentar til utledning av diffusivitetstiligningen

La oss betrakte volumet $\Delta V = A\Delta x$ i Figur 2.1. Masseraten ut av volumelementet ved $x + \Delta x$ er $(\rho_o u_o A)_{x+\Delta x}$ og masseraten inn i volumelementet ved x er $(\rho_o u_o A)_x$. Differansen må være lik endring av masse per tidsenhet i volumelementet,

$$(\rho_o u_o A)_{x+\Delta x} - (\rho_o u_o A)_x = -\frac{\Delta_t(\Delta V \phi \rho_o)}{\Delta t}, \quad \dots \dots \dots (2.8)$$

hvor Δ_t betyr tidsdifferanse ved konstant x -verdi. Vi deler nå ligningen på ΔV , og lar Δx og Δt gå mot null. Da får vi

$$\frac{\partial}{\partial x}(\rho_o u_o) = -\frac{\partial}{\partial t}(\phi \rho_o), \quad \dots \dots \dots (2.9)$$

som i Ligning 2.6 og kan fortsette derfra. Hvordan skal vi få formulert inn en rate i ligningen? Dette kan gjøres på to måter. Dersom vi holder oss til Ligning 2.9, så må raten formuleres som et grensevilkår for en bestemt x -verdi. For eksempel kan det bli spesifisert at raten ved utløpet av reservoaret skal ha en angitt verdi. Denne måten brukes generelt når en skal løse en differensialligning analytisk.

Den andre måten er å ta utgangspunkt i Ligning 2.8, altså i den diskrete formen. Vi bruker at Δ_x er differanseoperator i x med t konstant. Videre så multipliserer og deler vi med Δx på venstre siden og får,

$$\frac{\Delta_x(\rho_o u_o)}{\Delta x} \Delta V = -\frac{\Delta_t(\Delta V \phi \rho_o)}{\Delta t}. \quad \dots \dots \dots (2.10)$$

Hvert ledd i denne ligningen har dimensjonen masse/tid. Inn i volumet ΔV sender vi nå en rate q_i med dimensjonen masse/tid. Vi har at $q_i = q_o \Delta V (\rho_{ost} + R_{so} \rho_{gstd})$, hvor dimensjonene på q_o er overflatevolum olje per tidsenhet per bulkvolum, som definert foran. Vi innfører også $\rho_o = b_o(\rho_{ost} + R_{so} \rho_{gstd})$ i de to andre leddene i ligning 2.10. Uttrykket for q_i kan vi sette rett inn i Ligning 2.10, forkorte bort konstanten $(\rho_{ost} + R_{so} \rho_{gstd})$ og dele på ΔV . Da får vi Ligning 2.4 i grensen at Δx og Δt går mot null.

2.4 Diffusivitetstiligning 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 diffusivitetstiligninger,

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

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.

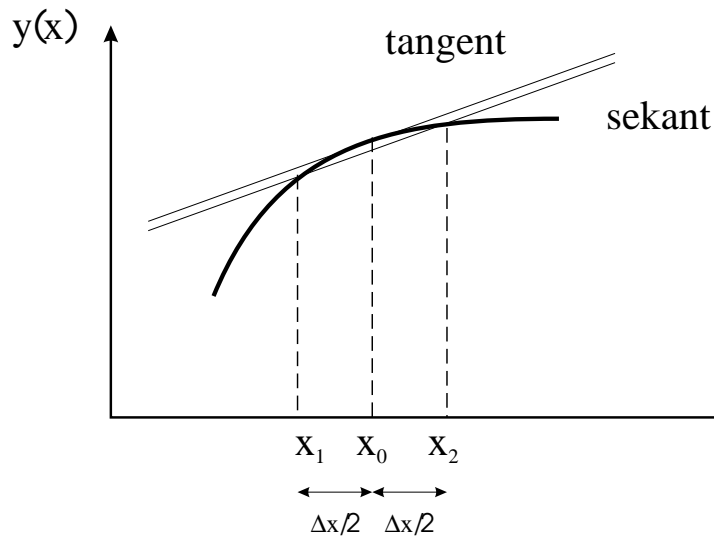
Den deriverte i punktet $(y(x_0), x_0)$ er gitt ved

$$\left. \frac{dy}{dx} \right|_{x=x_0} = \lim_{\Delta x \rightarrow 0} \frac{y(x_0 + \Delta x/2) - y(x_0)}{\Delta x/2},$$

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

$$\left. \frac{dy}{dx} \right|_{x=x_0} \approx \frac{y(x_0 + \Delta x/2) - y(x_0 - \Delta x/2)}{\Delta x} = \frac{y(x_2) - y(x_1)}{\Delta x}.$$

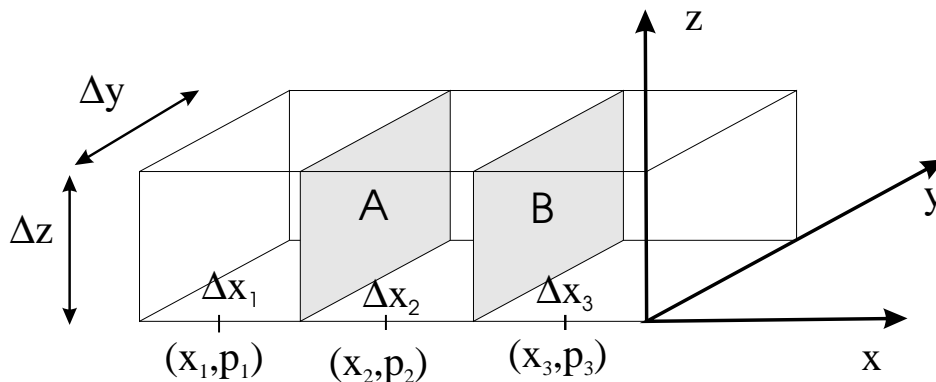
Dette svarer til å ta med kun første ordens ledd i en Taylorutvikling av $y(x)$ omkring punktet x_0 .



Figur 3.1: Tilnærming av tangent med sekant.

3.2 Avstandsderiverte i diffusiviteligningen

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

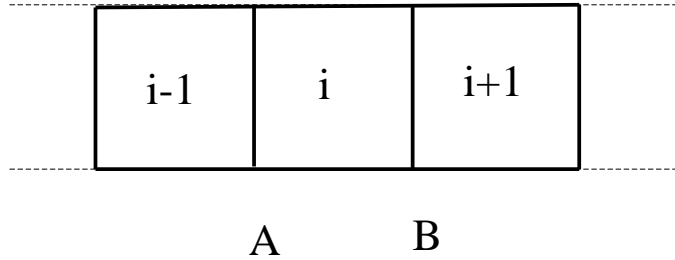
$$[b]_2^+ = \frac{1}{4}(b(p(x_2, t + \Delta t)) + b(p(x_3, t + \Delta t)) + b(p(x_2, t)) + b(p(x_3, t))).$$

Således blir den b -verdien som inngår i N_x -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

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

og Ligning 2.5 skrives som

$$\frac{\partial Y}{\partial x} + A_1 \rho = A_2 \frac{\partial p}{\partial t} \dots \dots \dots (3.1)$$

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

$$\frac{Y_i|_B - Y_i|_A}{\Delta x_i} + A_1 \rho_i = A_2 \frac{p_i(t + \Delta t) - p_i(t)}{\Delta t} \dots \dots \dots (3.2)$$

Videre tilnærmes

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

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 tilsvarende for det andre leddet,

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

Vi definerer nå

$$O_{xi}^+ = \frac{N_{xi}^+}{\Delta x_i},$$

$$O_{xi}^- = \frac{N_{xi}^-}{\Delta x_i},$$

som innsatt i Ligning 3.2 gir diffusivitetstiligningen på følgende numerisk form

$$O_{xi}^+ \cdot \Delta p_i^+ + O_{xi}^- \cdot \Delta p_i^- + A \vartheta_i = A 2_i (p_i(t + \Delta t) - p_i(t)) / \Delta t. \quad \dots \dots (3.3)$$

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 tidsstegssløyfe som beregner trykkene i de numeriske blokkene etter et fastsatt nytt tidssteg Δt , når alle størrelsene i ligningen, inklusive trykkene i hver blokk, er kjent på tidsnivå t .

3.4 Valg av numerisk formulering

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

Numerisk formulering	Tidsnivå for Δp_i^\pm
eksplisitt	t
implisitt	$t + \Delta t$
Crank-Nicolson	$t + \Delta t / 2$

Tabell 3.1: De tre vanligste numeriske formuleringer.

Før vi går videre trenger vi noen begreper fra numerisk analyse. De vil bare bli omtalt her og interesserte kan se nærmere i boken til Carnahan *et al* [4] eller andre standardverker i numeriske metoder for utfyllende kommentarer.

La oss anta at u er den korrekte, eventuelt analytiske, løsning til en differensialligning, og at v er den tilsvarende numeriske løsning. *Diskretiseringsfeilen* er gitt ved $w = u - v$. Dersom $w \rightarrow 0$ 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; $w \sim O(\Delta t) + O(\Delta x^2)$; 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; $w \sim O(\Delta t) + O(\Delta x^2)$

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

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

3.5 Eksplisitt formulering

Nå velges Δp_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:

$$p_i(t + \Delta t) = \frac{\Delta t}{\Delta 2_i} (O_{xi}^+ \cdot \Delta p_i^+(t) + O_{xi}^- \cdot \Delta p_i^-(t) + \mathbb{A}9_i) + p_i(t). \quad \dots \dots (3.4)$$

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 Δp_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 Δp_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

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

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:

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

som kan skrives som en matriseligning

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

eller

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

hvor koeffisientmatrisen \vec{K} ,

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

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

3.7 Iterativ løsningsmetode

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

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

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

eller

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

La k betegne iterasjonstallene. 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 a_1 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

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

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

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

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

    return
end subroutine tridia

```

3.9 Skisse til flytskjema for utlevert program

INITIALISERING

Les data
 Skriv data
 Beregn konstante størrelser
 Initialiser variable
 Beregn OIP

α START TIDSTEGSSLØYFE

Oppdater trykk, $POLD = PO$

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

$PEST = PO(1)$
 $PO = (PO + POLD) / 2$
 Fluidegenskaper —subrutine FLPROP
 Strømningskoeffisienter —subrutine FLOCON
 Løs ligningssettet med en av subrutinene
 TRIDIA –dersom direkte løsning
 GSIT –dersom iterativ løsning

Skriv beregnet trykk dersom $IWRITE=1$

Beregn $E = \|PO(1) - PEST\|$

β' SLUTT Dersom $E \geq \varepsilon$: Gå til β

! ε er innlest toleransekriterium

Gjør materialbalanse

Skriv ut trykk og tidsstegsammendrag

α' SLUTT Dersom flere tidssteg: Gå til α

STOP

Bibliografi

- [1] Aziz, K. and Settari, A.(1979). *Petroleum Reservoir Simulation*, Applied Science Publishers, London.
- [2] Mattax, C.C. and Dalton, R.L. *Reservoir Simulation*, SPE Monograph Volume 13, Richardson, TX (1990).
- [3] Paul Papatzacos, *Matematisk Modelling*, kompendium i fag av samme navn, Avsnitt 2.2, Høgskolesenteret i Rogaland.
- [4] Carnahan, B., Luther, H. A., and Wilkes J. O.: *Applied Numerical Methods*, John Wiley & Sons, New York (1969).
- [5] Johnson, E.F., Bossler, D.P., and Naumann, V.O.: “Calculation of Relative Permeability from Displacement Experiments,” *Trans. AIME*, **1959**, 371.
- [6] Welge, H.J.: “A Simplified Method for Computing Oil Recovery by Gas or Water Drive,” *Trans. AIME*, **1952**, 91.
- [7] Kyte, J.R., and Rapoport, L.A.: “Linear Waterflood Behavior and End Effects in Water-Wet Porous Media,” *Trans. AIME*, **1958** 423.
- [8] 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
!   deltl1  de 12 første tidssteg er deltl1/(1.5**(12-1)), hvor
!           l er antall tidssteg, days
!   eps     trykktoleranse, iterasjon på ulineære ledd _og_ i
!           iterativ løsningsrutine, psi
!   dex     delta x, ft
!   dey     delta y, ft
!   dez     delta z, ft
!   fpb     cubic feet per barrel, 5.61
!   pi      3.14
!   vol     dex*dey*dez/fpb, rb
!   volphi  vol*phi, rb
!   ckbak   konstantleddet i strømningskoeffisientene
!   a2      phi*dbdp
!   ooip    original oil in place, stb
!   pest    estimated value of po(1), psi
!   qotot   total oil rate, stb/day
!   ctim    kumulativ simulert tid, days
!   eot     total oil error
!   borig   original value of bo, stb/rb
!   phi     porosity
!   porig   original oil pressure, psi
!   delt    time step size, days
!   convf   conversion factor from Darcy units to oil-field
!           units
!   en      1.d0
!   to      2.d0
!

real(dp) :: kx,nt,ni,np,npc,visorg,dbdp,deltl1,eps,dez,&
            dey,fpb,pi,vol,volphi,ckbak,a2,ooip,&
            pest,qotot,ctim,eot,borig,phi,dex,porig,&
            delt,convf,en,to

```

```

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

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

!   ** read and write input data

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

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

write (6,3010)   mx, stmax, isol, ipe, iskip, iwrite
read  (5,*)     kx, phi, dex, porig, borig
read  (5,*)     visorg, dbdp, deltl, eps, dez
read  (5,*)     dey
write (6,3020)   kx, phi, dex, porig, borig, visorg, &
                dbdp, deltl, 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 (1 <= 12) delt = delt1/1.5**(12-1)

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, deltl, 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,&

```



```

!
!
! *****
! *
! *               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 felles-matriser

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

  real(dp), dimension(:), allocatable ::&
    a,c,d,a1,s,comp,po,&           !common/blk1/
    pold,oxplus,oxmin,a9,&        !common/blk2/
    qo,bo,x                       !dimensioned matrices
end module felles_matriser
```

3.11 Module nrtype

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

3.12 Datafil

```
6,4,1,1,1,0           mx,stmax,isol,ipe,iskip,iwrite
100,.05,10,2000,.7889  kx,phi,dex,porig,borig
1.358,8.03d-06,.002,0.001,1  visorg,dbdp,delt1,eps,dez
10                      dey
-1,4*0,1               oil flow rates (mx values)
```

3.13 Resultatfil

***** one dimensional, one phase simulator *****

basic assumptions of this model include:

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

```

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

horizontal permeability, kx, md ..... : 100.0000
porosity, phi, fraction ..... : 0.0500
delta length (x-dir), dex, ft ..... : 10.00
original oil pressure, porig, psia ..... : 2000.0
original oil fvf, borig, std vol / res vol ... : 0.7889
original oil viscosity, visorg, cp ..... : 1.3580
oil fvf pressure derivative, dbdp,
  std vol/res vol/psi ..... :0.00000803
time increment, deltt1, 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 .... 1.0

```

```

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

```

```

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

```

```

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

```

```

block no. : 1      oil pressure, psia .....      1989.815
block no. : 2      oil pressure, psia .....      1996.558
block no. : 3      oil pressure, psia .....      1999.213
block no. : 4      oil pressure, psia .....      2000.787
block no. : 5      oil pressure, psia .....      2003.443
block no. : 6      oil pressure, psia .....      2010.185
steps= 3          time= 0.1098D-03      np= 0.1098D-03
delt= 0.5202D-04  cum cycle= 6          ni= 0.1098D-03
nt= 0.4215D+01   qotot= 0.1000D+01      cum oil error= 0.2107D-13
npc= 0.2606D-02

```

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