3. Numeriske formuleringer og `sor`

Problemstilling

a) Modifiser programmet til å omfatte opsjon for implisitt, eksplisitt og Crank-Nicolson formulering. Les inn trigger `icrn` som er 1 for implisitt, 2 for C-N, og 3 for eksplisitt formulering.

Løs ligningssettet med `sor, gsit, tridia` ved å bruke `isol` lik 1 for `tridia`, 2 for `sor` eller `gsit` dersom $\omega = 1.0$, og 3 dersom ingen løsningsrutine er nødvendig.

b) Kjør med data som i Øving 2 med følgende unntak: `stmax` = 12, `eps` = 0.001, og sammenlign transienttrykkene i blokk 1, dvs. plott `po(1)` mot $\log t$ fra følgende kjøringer: (1) implisitt, `tridia`, (2) C-N, `tridia`, (3) eksplisitt, (4) analytisk løsning `po`(5 ft, t), fra utdelt program.

c) 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.

Godkjenning. Følgende leveres til godkjenning: Programkode, utskrift av resultatene fra kjøringene under b), et plott med de fire kurvene fra b), et plott av total-$\sum$ `kkkcum` mot ω som viser optimal ω.

Kommentarer

La k angi iterasjonsnivå. Da har vi følgende iterasjonsskjema for `gsit`:

$$p_i^{k+1} = a_i p_{i-1}^{k+1} + c_i p_{i+1}^{k} + d_i,$$

og for `sor:`

$$p_i^{k+1} = p_i^{k} + \omega \left[a_i p_{i-1}^{k+1} + c_i p_{i+1}^{k} + d_i - p_i^{k} \right]$$

Forkortelsen `sor` betyr Suksessiv OverRelaksasjon og ω kalles relaksasjonsfaktor. Dersom $\omega \in [1,2]$ så har vi overrelaksasjon, om $\omega \in [0,1]$ så er det underrelaksasjon, og dersom $\omega = 1.0$ så er `sor = gsit`. Dersom $\omega \geq 2$ så vil `sor`-metoden divergere.

I koden skiftes `gsit` ut med `sor:`

```
subroutine sor(............., om)
      .
      .    innholdet her er som for gsit,
      .    men skift ligning til å inkludere ω
      .
return
end
```

Overrelaksasjonsfaktoren `om` leses inn fra datafil og skrives ut i hovedprogrammet. I hovedprogrammet, etter `call flocon...` settes inn

```
select case(isol)
case(1)
    call tridia
case(2)
         po = to*po - pold
         comp = po
    call sor(..... , om)
case(3)
    im = 1
    do i = 1, mx                    For eksplisitt
         if (- - -) im =            formulering.
         if (- - -) ip =            Beregner trykket
                                    i hver blokk uten
                                    løsningsrutine.
              po = a · pold + c · pold + d
    end do
end select
```

Fra før har vi for en generell blokk i:

$$ox_i^+ \Delta p_i^+ + ox_i^- \Delta p_i^- + a9_i = a2(p_i(t+\Delta t) - p_i(t))/\Delta t.$$

Dette ligningssettet ønsker vi på formen,

$$-a_i p_{i-1} + p_i - c_i p_{i+1} = d_i,$$

uansett numerisk formulering.

Eksplisitt formulering.

$$ox_i^+ (p_{ip} - p_i) + ox_i^- (p_{im} - p_i) + a9_i = \frac{a2}{\Delta t}(p_i(t+\Delta t) - p_i),$$

hvor alle trykkene, bortsett fra $p_i(t+\Delta t)$ er ved "gammel" tid.
Ved å bruke $b = \Delta t/a2$
$a_i = b * ox_i^-$
$c_i = b * ox_i^+$
$d_i = b * (a9_i - (ox_i^+ + ox_i^-) * pold_i) + pold_i,$
blir ligningen

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

som settes inn i hovedprogrammet under `isol` $= 3$ opsjonen.

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

```
subroutine flocon (----, icrn)
.....
.....
      if (icrn .eq. 1) then
           Implisitt formulering -- a_i, c_i, d_i, se neste side
      else if (icrn .eq. 2) then
           C-N formulering -- a_i, c_i, d_i, se neste side
      else if (icrn .eq. 3) then
           Eksplisitt formulering -- a_i, c_i, d_i, som over
      endif
return
end
```

Implisitt formulering (som før)

$$\begin{aligned}
\mathtt{b} &= \mathtt{ox}^- + \mathtt{ox}^+ + \mathtt{a2}/\Delta\mathtt{t} \\
\mathtt{a} &= \mathtt{ox}^-/\mathtt{b} \\
\mathtt{c} &= \mathtt{ox}^+/\mathtt{b} \\
\mathtt{d} &= (\mathtt{a2}/\Delta\mathtt{t} * \mathtt{pold} + \mathtt{a9_i})/\mathtt{b}
\end{aligned}$$

C–N:

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

Med det foreslåtte valg av `icrn` og `isol`, må `isol = 3` når `icrn =3.`

icrn = 1, $\begin{cases} \mathtt{isol} = 1 \\ \mathtt{isol} = 2 \end{cases}$

icrn = 2, $\begin{cases} \mathtt{isol} = 1 \\ \mathtt{isol} = 2 \end{cases}$

icrn = 3, `isol` = 3.