SINTEF A171 - Åpen

RAPPORT

Segregert implisitt algoritme for løsning av Navier-Stokes og Reynolds ligninger

Torbjørn Utnes

SINTEF IKT

Anvendt matematikk

Juni 2006

SINTEF RAPPORT

SINTEF IKT

Postadresse: 7465 Trondheim
Besøksadresse: Alfred Getz vei 1, NTNU
7491 Trondheim
Telefon: 73 59 30 48
Telefaks: 73 59 29 71

Foretaksregisteret: NO 948 007 029 MVA

TITTEL

Segregert implisitt algoritme for løsning av Navier-Stokes og Reynolds ligninger

FORFATTER(E)

Torbjørn Utnes

OPPDRAGSGIVER(E)

Avinor AS

RAPPORTNR.	GRADERING	OPPDRAGSGIVERS REF.	
SINTEF A171	Åpen	Erling Bergersen	
GRADER. DENNE SIDE	**ISBN**	**PROSJEKTNR.**	**ANTALL SIDER OG BILAG**
Åpen	82-14-02867-1	90A260	9
ELEKTRONISK ARKIVKODE		**PROSJEKTLEDER (NAVN, SIGN.)**	**VERIFISERT AV (NAVN, SIGN.)**
SINTEF A171.pdf		Karl J. Eidsvik K. J. Eidsvik	Karstein Sørli K. Sørli
ARKIVKODE	**DATO**	**GODKJENT AV (NAVN, STILLING, SIGN.)**	
90A260	2006-06-21	for Svein Nordenson, Forskningssjef Runar Holdahl	

SAMMENDRAG

Det er gjennomført innledende vurderinger og testing av alternative algoritmer med tanke på effektivisering av beregningene i Simra. Dette er interessant spesielt for fremtidig bruk i reelle tidssimuleringer på lokal skala.

De foreløpige resultatene med en ny algoritme gir betydelig effektivisering, med en stabilitet som også gjør programmet mer robust. Det gjenstår mye mer testing, og programmet må deretter parallelliseres slik at mer reelle sammenligninger kan gjennomføres.

STIKKORD	NORSK	ENGELSK
GRUPPE 1	Numerisk simulering	Numerical simulation
GRUPPE 2	CFD	CFD
EGENVALGTE	Strømningsmodellering	Flow modelling

Innhold

1 Innledning

Løsningsalgoritmen i Simra er en trykk-projeksjonsmetode, nærmere bestemt en formulering av typen 'Projeksjon 1' som beskrevet i Gresho & Sani (2000). Dette er en relativt effektiv metode, men den numeriske stabiliteten er begrenset bl.a. p.g.a. eksplisitt adveksjon.

Det er ønskelig med bedre numerisk stabilitet med tanke på reelle tidssimuleringer i varslingssammenheng. Men dette forutsetter en algoritme som er kosteffektiv, dvs. som kan bruker relativt store tidsskritt og samtidig ikke krever for mye regnetid pr. tidsskritt.

Hensikten med denne rapporten er å vurdere mulige interessante metoder, og teste en eller flere algoritmer med tanke på en mer kosteffektiv metode.

2 Bevegelsesligninger: Anelastisk Navier-Stokes

For testing av aktuelle numeriske algoritmer tar vi utgangspunkt i en anelastiske formen av bevegelsesligningene, siden dette er spesielt aktuelt for atmosfærisk strømning. Den komplette formuleringen inkluderer potensiell temperatur, tilstandsligning og ligninger for turbulensmodellering, men her betrakter vi bare Navier-Stokes-delen av systemet, siden dette utgjør kjernen i problemet med å finne en effektiv løsningsalgoritme.

Forenklet kan den anelastiske formen av Navier-Stokes ligninger skrives på følgende form (Bannon 1995, 1996):

$$\frac{D\mathbf{u}}{Dt} - \nabla \cdot (\nu \nabla \mathbf{u}) = -\nabla \left(\frac{p}{\rho_s} \right) + \mathbf{s} \tag{1}$$

$$\nabla \cdot (\rho_s \mathbf{u}) = 0 \tag{2}$$

hvor $\rho_s(\mathbf{x})$ representerer (tidsuavhengig) hydrostatisk tetthet og $\mathbf{s}$ inneholder oppdrifts/stratifikasjons- og evt. Coriolis-effekter. Disse ligningene er konsistent med en energikonservativ form av den anelastiske formuleringen, se Appendiks.

3 Algebraisk formulering

På diskretisert form kan ligningene ovenfor skrives

$$\left[\frac{M}{\Delta t} + K^{n+1} \right] u^{n+1} = \frac{M}{\Delta t} u^n + b^{n+1} - C \left(\frac{p}{\rho_s} \right)^{n+1} \tag{3}$$

$$C^T \rho_s u^{n+1} = 0 \tag{4}$$

hvor M representerer massematrisen, K er summen av diffusjons- og konveksjonsmatrise, C er gradientmatrisen og b representerer stratifikasjons- og evt. Coriolis-effekter. Videre representerer (u, p) vektorer for de variable, mens indeks n angir tidspunktet. I momentum-ligningen har vi antatt en enkel Baklengs Euler (BE) tidsintegrasjon; men denne kan enkelt erstattes med f.eks. et BDF2-skjema dersom tidsvariasjonene krever større nøyaktighet. I praksis er det nyttig å skrive momentum-ligningen på inkrementform, og (3) tar da formen

$$\left[\frac{M}{\Delta t} + K^{n+1}\right] \Delta u^{n+1} = -K^{n+1} u^n + b^{n+1} - C \left(\frac{p}{\rho_s}\right)^{n+1} \tag{5}$$

hvor $\Delta u^{n+1} = u^{n+1} - u^n$.

En konsistent trykkligning kan utledes fra (3) på formen

$$\left[C^T \tilde{M}^{-1} C\right] p^{n+1} = C^T \left(\rho_s u^n - \rho_s u^{n+1} - \tilde{M}^{-1} K \rho_s u^{n+1}\right) \tag{6}$$

hvor vi har innført forkortelsen $\tilde{M} := M/\Delta t$. Ved å innføre kontinuitetsbetingelsen (4) gir dette tilslutt

$$\left[C^T \tilde{M}^{-1} C\right] p^{n+1} = C^T \left(\rho_s u^n - \tilde{M}^{-1} K \rho_s u^{n+1}\right) \tag{7}$$

4 Segregerte implisitte algoritmer

En god løsningsalgoritme for denne typen tidsavhengige problemer bør ha egenskaper som nøyaktighet, god numerisk stabilitet og kosteffektivitet. Egenskaper som stabilitet og robusthet gjør at implisitte løsninger ofte er å foretrekke, selv om dette betyr større ligningssystemer. For å unngå altfor store systemer er det en fordel å benytte en segregert metode, dvs. splitte opp hele systemet i flere mindre ligningssystemer som deretter løses i serie, med eventuell iterasjon.

4.1 Trykk-projeksjonsalgoritme (PP)

Klassiske iterative segregerte metoder er kjent under navn som Simple, Simpler, Simplec, osv., og disse løsningsalgoritmene har vist seg å fungere utmerket. En metode av lignende type er beskrevet i Haroutunian et al. (1993), og benyttes i kommersielle programsystemer som Fluent/Fidap. Dette er en trykk-projeksjonsalgoritme som består av følgende steg, med referanse til vårt ligningssystem:

PP-algoritme:

Iterativ prosedyre innenfor hvert tidsskritt, hvor $k = 0, 1, 2, ...$ inntil ønsket konvergens, før neste tidsskritt gjennomføres.

Trykkprediksjon

Predikterer trykkfeltet fra ligning (7)

$$\left[C^T \tilde{M}_L^{-1} C\right] p_{k+1}^{n+1} = C^T \left(\rho_s u^n - \tilde{M}_L^{-1} K \rho_s u_k^{n+1}\right) \tag{8}$$

Bevegelsesligning

Beregner hastighetsfeltet fra implisitt bevegelsesligning (5)

$$\left[\tilde{M} + K\right] \Delta u_{k+1/2}^{n+1} = -K u^n + b - C \left(\frac{p}{\rho_s}\right)_{k+1}^{n+1} \tag{9}$$

Korreksjon

Korrigerer hastighetsfeltet for kontinuitetsbetingelsen vha projeksjonen

$$\left[C^T \tilde{M}_L^{-1} C\right] \delta p = C^T \rho_s u_{k+1/2}^{n+1} \tag{10}$$

$$\tilde{M}_L \left(\rho_s u_{k+1}^{n+1} - \rho_s u_{k+1/2}^{n+1}\right) = -C\delta p \tag{11}$$

Denne algoritmen er beskrevet i Haroutunian et al. (1993) med bruk av en generalisert (ad hoc) prekondisjonering for koeffisientmatrisen $(\tilde{M} + K)$. For bruk i tidssimuleringer har vi her antatt den enklere massediagonaliseringen $\tilde{M}_L$.

4.2 Prediktor-korrektorsalgoritme (PISO)

For bruk i tidssimuleringer er det ønskelig med en ikke-iterativ prosedyre. Den såkalte PISO-algoritmen er en slik ikke-iterativ metode av lignende type som foran, men hvor en prediktor-korrektor-metodikk benyttes istedenfor iterasjoner innenfor hvert tidsskritt (Issa 1986, Oliveira & Issa 2001). Denne formuleringen kan skrives på følgende form:

PISO-algoritme:

Prediktor

Predikterer hastighetsfeltet fra bevegelsesligningen (3)

$$\left[\tilde{M} + K\right] u^* = \tilde{M} u^n + b - C\left(\frac{p}{\rho_s}\right)^n \tag{12}$$

Korrektor 1

Foreta en korreksjon vha projeksjonen

$$\left[C^T \tilde{M}_L^{-1} C\right](p^* - p^n) = C^T \rho_s u^* \tag{13}$$

$$\tilde{M}_L(\rho_s u^{**} - \rho_s u^*) = -C(p^* - p^n) \tag{14}$$

Korrektor 2

Foreta en ny korreksjon vha projeksjonen

$$\left[C^T \tilde{M}_L^{-1} C\right](p^{**} - p^*) = C^T \left[\rho_s u^{**} - \tilde{M}_L^{-1} K(\rho_s u^{**} - \rho_s u^*)\right] \tag{15}$$

$$\tilde{M}_L(\rho_s u^{***} - \rho_s u^{**}) = -C(p^{**} - p^*) - K(u^{**} - u^*) \tag{16}$$

hvor $u^{n+1} = u^{***}$, $p^{n+1} = p^{**}$.

Den første korreksjonen fås ved å oppdatere bevegelsesligningen (12) til

$$\tilde{M} u^{**} = \tilde{M} u^n - K u^* - C\left(\frac{p}{\rho_s}\right)^* + b \tag{17}$$

og deretter subtrahere (12) fra denne korreksjonen. Dette gir momentum-korreksjonen i 'Korrektor 1', mens trykkligningen fås ved projeksjon og bruk av kontinuitetsligningen. 'Korrektor 2' fås fra oppdateringen

$$\tilde{M} u^{***} = \tilde{M} u^n - K u^{**} - C\left(\frac{p}{\rho_s}\right)^{**} + b \tag{18}$$

og tilsvarende prosedyre som ovenfor.

4.3 Trykk-projeksjons-korreksjonsalgoritme (PPC)

PP-algoritmen har god stabilitet, men benytter en ytre iterasjon som kan være ugunstig i vårt tilfelle hvor tidssimulering er vesentlig. Alternativet med PISO-algoritmen er derfor interessant, men stabiliteten kunne vært bedre, i det minste gjelder dette mine foreløpige tester. Derfor foreslås en modifisert ikke-iterativ type trykk-projeksjons-korreksjonsalgoritme, med trekk fra begge de tidligere nevnte algoritmene.

Istedenfor trykkligningen i PP-algoritmen predikteres trykket her vha en 'klassisk' projeksjonsalgoritme (første del av 'Projeksjon 1' hos Gresho & Sani 2000). Deretter beregnes bevegelsesligningen på implisitt form, med bruk av det predikterte trykket, og tilslutt foretas korreksjon av hastighetsfeltet og en projeksjon for å korrigere for kontinuitetsbetingelsen. Trykket oppdateres ikke, men beregnes på nytt for hvert tidssteg som i Simpler-metodikken (Comini et al. 1997).

1. Prediksjon
En klassisk projeksjonsalgoritme for ligningssystemet

$$\begin{aligned} \tilde{M}u^{**} &= \tilde{M}u^n - Ku^n + b - C\left(\frac{p}{\rho_s}\right)^{**} && (19) \\ C^T\rho_s u^{**} &= 0 && (20) \end{aligned}$$

kan utføres vha stegene

$$\begin{aligned} \tilde{M}u^{*} &= \tilde{M}u^n - Ku^n + b && (21) \\ \left[C^T\tilde{M}_L^{-1}C\right]p^{**} &= C^T\rho_s u^{*} && (22) \\ \tilde{M}_L(\rho_s u^{**} - \rho_s u^{*}) &= -Cp^{**} && (23) \end{aligned}$$

I denne omgang søker vi bare en prediksjon av trykket, og trenger derfor bare å beregne de to første ligningene i denne prosedyren.

2. Bevegelsesligning
Med gitt trykkfelt beregnes deretter bevegelsesligningen implisitt:

$$\left[\tilde{M} + K\right]u^{**} = \tilde{M}u^n + b - C\left(\frac{p}{\rho_s}\right)^{**} \qquad (24)$$

3. Korreksjon
En korreksjon av bevegelsesligningen ovenfor kan skrives eksplisitt som

$$\tilde{M}u^{***} = \tilde{M}u^n - Ku^{**} + b - C\left(\frac{p}{\rho_s}\right)^{***} \qquad (25)$$

Merk at vi her formelt antar at $K = K^*$ i stegene 2 og 3. Ved å subtrahere (24) fra korreksjonen (25) fås den ønskede korreksjonsligningen

$$\tilde{M}(\rho_s u^{***} - \rho_s u^{**}) = -C\left(p^{***} - p^{**}\right) \qquad (26)$$

som samtidig skal oppfylle kontinuitetsbetingelsen

$$C^T \rho_s u^{***} = 0 \tag{27}$$

Denne betingelsen tilfredsstilles vha en projeksjon, som gir trykkligningen

$$\left[C^T \tilde{M}_L^{-1} C\right] (p^{***} - p^{**}) = C^T \rho_s u^{**} \tag{28}$$

eller på enklere form som

$$\left[C^T \tilde{M}_L^{-1} C\right] \delta p^{n+1} = C^T \rho_s u^{**} \tag{29}$$

Den endelige algoritmen kan dermes skrives på følgende form:

PPC-algoritme:

Trykkprediksjon

Predikterer trykkfeltet (fra første del av klassisk projeksjonsalgoritme)

$$\tilde{M} \Delta u^* = -K u^n + b \tag{30}$$

$$\left(C^T \tilde{M}_L^{-1} C\right) p^{n+1} = C^T \rho_s u^* \tag{31}$$

Bevegelsesligning

Beregner hastighetsfeltet fra implisitt bevegelsesligning

$$\left[\tilde{M} + K\right] \Delta u^{**} = -K u^n + b - C \left(\frac{p}{\rho_s}\right)^{n+1} \tag{32}$$

Korreksjon

Korrigerer hastighetsfeltet for kontinuitetsbetingelsen vha projeksjonen

$$\left[C^T \tilde{M}_L^{-1} C\right] \delta p^{n+1} = C^T \rho_s u^{**} \tag{33}$$

$$M_L \left(\rho_s u^{n+1} - \rho_s u^{**}\right) = -C \delta p^{n+1} \tag{34}$$

Merk:

Trykkberegningen gjøres pånytt for hvert tidsskritt, tilsvarende som i Simpler-algoritmen. Derimot oppdateres trykket sammen med hastighetsfeltet i korreksjonssteget innenfor hvert tidsskritt.

Praksis har vist at eventuelle ligninger for turbulens, temperatur osv bør følge etter bevegelsesligningen, og før siste korreksjonssteg. Dette gir bedre numerisk stabilitet enn hvis disse ligningene tas etter korreksjonssteget, og er i samsvar med den Simpler-lignende algoritmen beskrevet i Haroutunian et al. (1993).

5 Innledende testing; seriell kode

Det er gjennomført endel testing med de tre algoritmene i seriell versjon, spesielt med den orginale PPC-algoritmen. Denne metoden unngår ytre iterasjoner (som i PP-algoritmen), og har dessuten færre operasjoner enn PISO-algoritmen.

De foreløpige beregningene gjelder simulering med nøytral og stratifisert strømning over et modell-fjell (Hunt & Snyder 1980). Disse beregningene er gjort på en PC av typen Dell Precision 670. Resultater fra denne testingen er vist i tabell 1 og 2 nedenfor. Sammenlignet med den opprinnelige Simra-koden gir PPC-algoritmen omtrent en dobling av regnetiden pr tidsskritt, men kan ta 4 - 5 ganger større tidsskritt. Altså er den nye algoritmen 2 - 3 ganger raskere for disse simuleringene.

Simuleringer med modell-fjell (Hunt & Snyder 1980):
Sammenligning av regnetid mellom PPC og Simra

Tabell 1: Nøytral strømning:

Modell	Courant-tall	tid pr tidsskritt	total regnetid
PPC	2.8	4.34 s	434 s
Simra	0.5	2.25 s	900 s

Tabell 2: Stratifisert strømning (F = 1):

Modell	Courant-tall	tid pr tidsskritt	total regnetid
PPC	2.7	4.49 s	449 s
Simra	0.5	2.95 s	1179 s

Kolonnen 'total regnetid' angir den regnetiden hver av modellene bruker for å nå samme tidspunkt i sann tid (her t = 25 s). Merk at disse regnetidene gjelder for et modell-fjell med høyde 1 m, og det er derfor bare de relative tidene som er interessante.

6 Avsluttende kommentarer

Det er gjennomført innledende vurderinger og testing av alternative algoritmer med tanke på effektivisering av beregningene i Simra. Dette er interessant spesielt for fremtidig bruk i reelle tidssimuleringer på lokal skala.

De foreløpige resultatene med en ny algoritme (PPC) gir betydelig effektivisering, med en stabilitet som også gjør programmet mer robust. Det gjenstår mye mer testing, og programmet må deretter paralelliseres slik at mer reelle sammenligninger kan gjennomføres. Dette planlegges gjennomført iløpet av høsten 2006.

A Appendiks

Anelastisk approksimasjon & energikonservering

Ligningene for momentum, kontinuitet og potensiell temperatur for adiabatisk, kompressibel strømning kan skrives i forkortet form som

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p' - g\frac{\rho'}{\rho}\mathbf{k} + \frac{\mathbf{F}}{\rho} \tag{i}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{ii}$$

$$\frac{D\theta}{Dt} = 0 \tag{iii}$$

hvor hydrostatisk likevekt er innført, med $p = p_s + p'$; $\rho = \rho_s + \rho'$, og F representerer friksjon/viskøse bidrag.

Anelastisk approksimasjon

En **energikonservativ** anelastisk formulering av dette systemet kan skrives på følgende form (Bannon 1995, 1996):

$$\frac{D\mathbf{u}}{Dt} = -\nabla\left(\frac{p'}{\rho_s}\right) + g\frac{\theta'}{\theta_s}\mathbf{k} + \frac{\mathbf{F}}{\rho_s} \tag{iv}$$

$$\nabla \cdot (\rho_s \mathbf{u}) = 0 \tag{v}$$

$$\frac{D\theta}{Dt} \approx \frac{D\theta'}{Dt} + w\frac{d\theta_s}{dz} = 0 \tag{vi}$$

For ikke-viskøs strømning (F = 0) kan det vises at dette systemet tilfredsstiller ligningen for konservering av energi:

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E+p)\mathbf{u}] = 0, \text{ hvor } E = \rho(\frac{1}{2}\mathbf{u}\mathbf{u} + gz + c_v T) \tag{vii}$$

En analog betingelse er konservering av potensiell virvling (Ertel 1942):

$$\frac{D}{Dt}\left(\frac{\boldsymbol{\omega}\cdot\nabla\theta}{\rho}\right) = 0, \text{ hvor } \boldsymbol{\omega} = 2\boldsymbol{\Omega} + \nabla \times \mathbf{u} \tag{viii}$$

Det er vist i Bannon (1995) at systemet ovenfor tilfredsstiller denne betingelsen. Merk at momentum- og kontinuitetsligningen her tilsvarer formuleringen som er benyttet tidligere i denne rapporten (ligningene 1, 2).

Referanser

[1] **Bannon P.R. (1995)**: Potential Vorticity Conservation, Hydrostatic Adjustment, and the Anelastic Approximation. J. Atmos. Sci., 52, 2302-2312.

[2] **Bannon, P.R.(1996)**: Potential vorticity conservation, hydrostatic adjustment, and the anelastic approximation. J. Atmos. Sci., 53, 3618-3624.

[3] **Haroutunian V., Engelman M.S., Hasbani I. (1993)**: Segregated finite element algorithms for the numerical solution of large-scale incompressible flow problems. Int J numer methods fluids, vol 17, 323 - 348.

[4] **Issa R.I. (1986)**: Solution of the implicitly discretized fluid flow equation by operator splitting. J. Comput. Phys., vol 62, 49-65.

[5] **Oliveira P.J., Issa R.I. (2001)**: An improved PISO algorithm for the computation of buoyancy-driven flows. Numerical Heat Transfer, Part B, 40, 473-493.

[6] **Gresho P.M., Sani R.L. (2000)**: Incompressible Flow and the Finite Element Method, J. Wiley & Sons, West Sussex, UK, 2000, 1112 pp.

[7] **Comini G., Minkowycz W.J., Shyy W. (1997)**: General algorithms for the finite element solution of incompressible flow problems using primitive variables. In: Advances in Numerical Heat Transfer, Vol 1, eds. Minkowycz W.J. and Sparrow E.M.

[8] **Durran, D.R. (1998)**: Numerical Methods for Wave Equations in Geophysical Fluid Dynamics. Springer, 465 pp.

[9] **Hunt, J.C.R., Snyder, W.H. (1980)**: Experiments on Stably and Neytrally Stratified Flow over a model three-dimensional hill. J. Fluid Mech. 96, 671-704.