

# Arbeidsnotater

S T A T I S T I S K S E N T R A L B Y R Å

Dronningensgt. 16, Oslo-Dep., Oslo 1. Tlf. 41 38 20

IO 75/ 21

22. mai 1975

TILLEGG NR. 1 TIL HÅNDBOK  
FOR BRUK AV DATSY

DATSY-notat nr. 4

INNHold

	Side
1. Innledning .....	1
2. Nye og endrede brukerbeskrivelser for direktiver .....	2
2.1 Liste over de direktiver som er beskrevet her .....	2
2.2 Brukerbeskrivelsene .....	2
3. Rettelser til Håndbok for bruk av DATSY .....	29
4. Oversikt over dokumentasjon av DATSY .....	31
4.1 Innledning .....	31
4.2 Utgitte publikasjoner .....	32
4.3 Planlagte publikasjoner .....	32
5. Alfabetisk liste over direktiver med sidereferanser og status ved Statens Driftssentrals maskinanlegg pr. 1. mai 1975 .....	33
Referanser .....	36

Dette er et notat i en serie arbeidsnotater om DATSY. Det er meningen at serien skal omfatte både tekniske og brukerorientert dokumentasjon av videreutvikling av systemet.

*Ikke for offentliggjøring. Dette notat er et arbeidsdokument og kan siteres eller refereres bare etter spesiell tillatelse i hvert enkelt tilfelle. Synspunkter og konklusjoner kan ikke uten videre tas som uttrykk for Statistisk Sentralbyrås oppfatning.*

## 1. Innledning

DATSY (Data Treatment System) er et brukerorientert dataspråk utviklet av Norsk Regnesentral på kontrakt med Finansdepartementet. Byrået tok i sin tid initiativet til dette prosjektet for å få utviklet et hensiktsmessig programmeringssystem for større modellprosjekter og har samarbeidet med Norsk Regnesentral om utformingen av systemet. Avsnitt 4 gir en oversikt over foreliggende dokumentasjon om DATSY.

Dette tillegg til Håndbok for bruk av DATSY [1] inneholder både nye og endrede brukerbeskrivelser for direktiver, samt en del andre ting som ikke var med i den opprinnelige håndboken.

Flere slike tillegg vil trenge etterhvert som systemet utvikler seg og korreksjoner foretas. Dette tillegget er redigert av David Walker med bidrag av Eldar Børsum og Inger Henningsen\*.

En ønskelig forbedring som ikke er tatt med denne gang er et langt og detaljert kapittel om hvordan man kan bruke direktivene for å løse visse standardproblemer som ofte gjentar seg, og som ikke løses av noe enkelt direktiv, men som kan løses på en grei måte med flere direktiver brukt sammen. Dette gjelder først og fremst problemer som ikke umiddelbart er egnet for løsning ved sammensatte direktiver eller VMAKRO-programmer [2]. Generelle retningslinjer for bruk av DATSY på en fantasifull og intelligent måte kan også tas med her.

Et slikt kapittel ville ikke være bundet til den eksisterende systemdel som DATSY-direktivene er knyttet til. Det er en logisk sak hvordan man kombinerer standardoperasjoner på standarddatastrukturer for å oppnå et bestemt resultat. Et slikt arbeid kunne vise seg å ha økt betydning for Byråets bruk av DATSY i framtiden. Det ville se ut nærmest som en liste med eksempler i første omgang, men en viss systematikk kan allerede bygges via teorien om elementære matriser (se direktiv GENOBJEKT).

Et kapittel om effektiv og pålitelig bruk av DATSY ville også være ønskelig.

To store modellprosjekter (se [3], [4]) er nå gjennomført ved bruk av DATSY, samt en del mindre prosjekter. Det er således på tide å vurdere mer utstrakt bruk av DATSY på andre arbeidsområder. Dette arbeid pågår, men er vesentlig hemmet av det faktum at DATSY krever bortimot 56 000 maskinord i hurtighukommelsen til Statens Driftsentrals Honeywell-Bull maskin for å kunne kjøre selv små oppgaver.

Det er å håpe at systemdelen til DATSY blir omprogrammert i løpet av 2-3 år, når en del erfaringer med den kan ventes å foreligge. Da vil den investering som er lagt ned i utarbeiding av direktiver komme til videre nytte. Direktivene vil ikke trenge noen forandringer for å kunne bli kjørt i en ny utgave av DATSY, selv om en ny utgave bør muliggjøre noen forenklinger i direktivene.

---

\* Forfatterne vil gjerne takke forsker Olav Bjerkholt for hjelp med tilrettelegging av dette notat.

## 2. Nye og endrede brukerbeskrivelser for direktiver

### 2.1 Liste over de direktivene som er beskrevet her

(se også avsnitt 5)

<u>Direktivnavn</u>	<u>Side</u>	<u>Direktivnavn</u>	<u>Side</u>
ADDER .....	2	LØKKE .....	17
DELHORIZONT .....	3	MAKRO .....	18
DELKOLVIS .....	4	MULTIPLISER .....	19
DIVIDER .....	4	OMSTABLER .....	20
EKSDELALT-2 .....	4	OPPLYSOBJEKT .....	20
EKSDELALT-5 .....	5	PRINTMATLIST .....	21
EKSKOL .....	5	RADADDER .....	21
EKSMAT .....	6	RADSKALER .....	22
EKSRAD .....	6	REKKEMULT .....	22
ELEMULT .....	7	SAMLE .....	23
GENOBJEKT .....	7	SKIFTREK .....	24
GJEM-LØSNINGER .....	10	SUBTRAHER .....	24
HOPP .....	12	SUMKOL .....	24
INVERS .....	13	SUMRAD .....	24
KOLADDER .....	13	TRANSFORMTABELL .....	25
KOLSKALER .....	13	TRANSPONER .....	26
KONVREKMAT .....	14	VMAKRO .....	26
KOPIOMKODING .....	14	XFORDEL .....	26
LISTEPRODUKT .....	15	YFORDEL .....	27

### 2.2 Brukerbeskrivelsene

NB. Hvis ikke annet er sagt, er henvisninger i brukerbeskrivelsene (f.eks. "Se B-3.8") til [1].

#### Direktiv ADDER

(ADDER numeriske objekter)

Form: ADDER matrise1 + matrise2 = matrise3.

Innobjekter: parameter1/vektor1/matrise1/rekordsett1, parameter2/vektor2/matrise2/rekordsett2

Utobjekt: parameter3/vektor3/matrise3/rekordsett3

To forskjellige former for addering utføres, nemlig matriseadding og addering av et enkelt tall til hvert element i en vektor/matrise.

(i) Matriseadddering

Til dette kreves like dimensjoner på begge innobjekter. Direktivet setter samme dimensjon på utobjektet. Tilsvarende elementer adderes.

(ii) Addering av et tall gjennom en vektor/matrise

Det kreves at 2.objekt er av dimensjon (1x1), men ikke at det er av klasse parameter. (Dette er analogt med direktiv MULTIPLISER.) 2.objektets verdi blir addert til alle verdiene i 1.objekt. Utobjektet får da samme dimensjon som 1.objekt.

Andre bemerkninger:

Direktivet kan også brukes med rekordsett som representerer pakkede matriser (se under direktiv PAKKINN). Da må alle tre objekter være av klasse rekordsett. Beskrivelse av disse er beskrevet under direktiv PAKKINN. Man oppnår ingen gevinst tidsmessig ved å addere pakkede matriser. Fordelene er at man slipper å pakke matriser ut og inn når pakket multiplisering benyttes, og at objektene vil oppta mindre plass hvis tettheten av ikke-null elementer er under en tredjedel.

Ellers kan innobjektene ha ulik klasse, bare dimensjonene er riktige, som med direktiv MULTIPLISER. Se også KOLADDER og RADADDER.

Teknisk notat: Aritmetiske operasjoner på hele tall er med hensikt ikke implementert, siden flytende tall alltid kan benyttes med de anvendelsene Byrået forventer.

Direktiv DELHORISONT

(DEL HORISONTalt)

Form: DELHORISONT del objekt1 etter linje nummer angitt av parameter2/liste2 og sett delene i objekt3 og objekt4.

Innobjekter: objekt1 (MATRISE/VEKTOR/LISTE/REKORDSETT), parameter2/liste2/navnord2

Utobjekter: objekt3 (samme), objekt4 (samme).

Det er ikke noe i veien for å deklare f.eks. VEKTOR for objekt3 og MATRISE for objekt4 bare kombinasjonen er rimelig. I dette eksemplet måtte objekt1 da være enten VEKTOR, eller MATRISE med bare én kolonne.

I tilfelle matrisen har en tilhørende radliste, kan det være bedre å referere til rader via denne liste og bruke direktiv EKSRAD.

Parameter2 kan være enten hel- eller flytendetallsparameter. I tilfelle liste2 er oppgitt vil denne bli tolket slik at det er antallet på elementene i liste2 som gjelder. navnord2 tilsvarer en liste med ett element.

Null eller negative tall som parameter2 vil føre til at direktivet stopper med feilmelding. Dette gjelder også når parameter2/liste2 er for stor. Imidlertid er direktivet blitt laget slik at alle linjene i 1.objekt kan tas med i 3.objekt. Da vil 4.objekt bli laget av direktivet som en linje med nuller.

Direktivet passer fint til bruk sammen med LØKKE, eller LET. Se også under direktiv LISTEPRODUKT om begrepet "listestyling". Jfr. DELKOLVIS.

Direktiv DELKOLVIS

(DEL KOLonneVIS)

Form: DELKOLVIS del matrisel etter kolonnennummer angitt av parameter2/liste2 og sett delene i matrise3 og matrise4.

Innobjekter: matrisel/vektor1/parameter1, parameter2/liste2/navnord2

Utoobjekter: matrise3/vektor3/parameter3, matrise4/vektor4/parameter4

Bortsett fra at lister og rekordsett ikke kan deles kolonnevis er dette direktivet helt analogt med DELHORIZONTAL.

Direktiv DIVIDER

(DIVIDER elementvis)

Form: DIVIDER alle elementene i matrisel divideres med de tilsvarende elementene i matrise2, og resultatet settes i matrise3.

Innobjekter: matrisel/vektor1/parameter1/rekordsett1, matrise2/vektor2/parameter2/rekordsett2

Utoobjekt: matrise3/vektor3/parameter3/rekordsett3.

Objekter av forskjellige klasser innen rammen matrise/vektor/parameter kan forekomme, både som innobjekter og utobjekt, men de to innobjektene må være identisk m.h.t. dimensjon. Det går f.eks. an å dividere elementene i vektor1 med tilsvarende elementer i matrise2 dersom matrise2 har bare én kolonne, og samme antall rader som vektor1 har elementer.

Klassen til utobjektet bestemmes av brukeren selv ved deklarasjon. Hvis den er urimelig, vil direktivet stoppe og skrive ut feilmelding om dette.

I de tilfellene målsettingen bare er å invertere elementene i innobjektet, bør direktivet INVERS benyttes direkte.

Når direktivet brukes med rekordsett må alle tre objekter være rekordsett som representerer pakkede matriser (se direktiv PAKKINN).

Direktiv EKSDELALT-2

(EKStraher en DELmatrise med ALTERNativer fra 2 ledd)

Form: EKSDELALT-2 matrisel + (matrise2 RADSKALERT/KOLSKALERT med radene fra matrise4)  
(subliste5, subliste6) = matrise7.

Innobjekter: matrisel, matrise2, navnord3, matrise4, subliste5, subliste6

Utoobjekt: matrise7

Direktivet er et spesialdirektiv som brukes i modellen MSG-3 for å beregne og ekstrahere deler av hjelpematriser. Ved å sammenfatte disse operasjoner i ett ble det mulig å unngå å først lage meget store hjelpematriser, for bare å ekstrahere deler av disse etterpå.

De såkalte leddmatrisene, nemlig matrisel og matrise2, er vanlige matriser uten noen alternativstruktur (altså de er ikke rekkematriser med sammenstablede delmatriser, som hver representerer et alternativt datasett). De er altså relativt små. Resultatmatrisen, nemlig matrise7, er også regnet å være relativt liten, selv om den har alternativstruktur som definert ovenfor. Dette er fordi mange rader og kolonner ikke er tatt med i ekstraheringen representert ved

(subliste5, subliste6),

som fungerer helt analogt med direktivet EKSMAT.

Hver rad fra matrise4 brukes i sin tur for å radskalere\* eller kolskalere (spesifisert av det 3. objekt, som er navnord med verdi lik sitt navn) matrise2. Resultatet av hver rad/kolskalering blir addert til matrisel, og resultatet av denne addering blir føyet radvis til i matrise7.

I tilfelle radskalering kan beregningen uttrykkes slik, hvis man ser bort fra alternativer og ekstrahering:

$$\text{matrise7} = \text{matrisel} + (\text{rad fra } \widehat{\text{matrise4}}) \text{ matrise2}$$

Her indikerer "^" at en vektor er lagt langs hoveddiagonalen i en nullmatrise.

Alternativene stables opp radvis i matrise7 p.g.a. at den som oftest inngår i en rekkemultiplisering, på venstre side (se direktiv REKKEMULT).

De listene som brukes for å lage subliste5 og subliste6 refererer seg til h.h.v. radene og kolonnene i matrisel (eller matrise2, siden den har rader og kolonner tilsvarende matrisel).

#### Direktiv EKSDELALT-5

(EKStraher en DELmatrise med ALternativer fra 5 ledd)

Form: EKSDELALT-5 matrisel + (matrise2 RADSKALERT/KOLSKALERT med radene fra matrise4)  
 + (matrise5 RADSKALERT/KOLSKALERT med radene fra matrise7)  
 + (matrise8 RADSKALERT/KOLSKALERT med radene fra matrisel0)  
 + (matrise11 RADSKALERT/KOLSKALERT med radene fra matrisel3)  
 (subliste14, subliste15) = matrisel6.

Innobjekter: matrisel, matrise2, navnord3, matrise4, matrise5, navnord6, matrise7, matrise8,  
navnord9, matrisel0, matrisel1, navnord12, matrisel3, subliste14, subliste15

Utobjekt: matrisel6

Se direktivet EKSDELALT-2. EKSDELALT-5 er en videreføring av dette til å omfatte fem leddmatriser istedenfor bare to.

#### Direktiv EKSKOL

(EKStraher KOLonner fra matrise)

Form: EKSKOL ekstraher de kolonnene av matrisel som er angitt ved subliste og sett resultatet i matrise2.

Innobjekter: matrisel, subliste

Utobjekt: matrise2

Se direktivet EKSMAT. EKSKOL er definert analogt med dette, med den forskjell at her gjelder det bare hele kolonner som skal ekstraheres, i en rekkefølge spesifisert av subliste.

Se også direktivene EKSRAD, SETTKOL, GENOBJEKT og SUBLISTGEN.

\* Se direktivene RADSKALER og KOLSKALER.

Direktiv EKSMAT

(EKStraher delMATrise fra matrise)

Form: EKSMAT ekstraher del av matrisel gitt ved sublistel subliste2 og sett resultatet i matrise2.

Innobjekter: matrisel, sublistel, subliste2

Utobjekt: matrise2

sublistel skal være en subliste som refererer til rader, mens subliste2 refererer til kolonner.  
(Se direktiv SUBLISTGEN.)

Eks.:

SUBLISTGEN SUBA består av RADERIA og RADUTVALG.

SUBLISTGEN SUBB består av KOLONNERIA og KOLONNEUTVALG.

EKSMAT ekstraher ut av matrisen A den delmatrisen angitt ved SUBA og SUBB og sett resultatet i B.

LISTE RADERIA KOLONNERIA

RADUTVALG KOLONNEUTVALG.

MATRISER A B.

SUBLISTE SUBA SUBB.

Listene må være entydige (ingen dobbeltforekomster). Se under direktiv ANTDOBLIST i denne sammenhengen. SUBLISTGEN vil gi melding om dobbeltforekomster.

Listeelementene i listene RADUTVALG og KOLONNEUTVALG behøver ikke å forekomme i den samme orden som i RADERIA og KOLONNERIA (som betegner radene og kolonnene i matrise A). Ved å forandre rekkefølgen av listeelementene kan brukeren foreta simultan omstokking av rader og kolonner i A, samtidig med ekstrahering.

Den refererte omstokkingen er noe forskjellig fra en fullstendig omstokking av vilkårlige elementer i matrisen.

De radene og kolonnene som brukeren kan ekstrahere ved bruk av dette direktivet behøver heller ikke være noen samlet blokk eller delmatrise i matrisel.

I eksemplet, hvis f.eks. RADUTVALG inneholder elementer som ikke fins i listen RADERIA, og som ikke kommer ved slutten av RADUTVALG, vil direktivet varsle om disse elementene, men ellers ignorere dem. Det tilsvarende gjelder KOLONNEUTVALG og KOLONNERIA. Denne oppførsel fra direktivets side er nødvendig for å kunne ta snittet av to sett, hvis ikke alle elementer i det ene inneholdes av det andre.

Varselutskriften ser f.eks. slik ut:

VARSEL: EKSMAT HAR FUNGERT I SNITT-MODE

ANTALL BRUDD PÅ HØYRE SIDE AV DEN 1. SUBLISTEN = 2

Matriseelementene må være flytendetaill.

Direktiv EKSRAD

(EKStraher RADER)

Form: EKSRAD ekstraher rader av matrisel/vektor1/rekordsett1/listel som er gitt ved subliste og sett resultatet i matrise2/vektor2/rekordsett2/liste2.

Innobjekter: matrisel/vektor1/rekordsett1/listel, subliste

Utobjekt: matrise2/vektor2/rekordsett2/liste2

Se direktivene EKSMAT og EKSKOL, EKSRAD er definert analogt med disse.  
vektor1 blir tolket som kolonnevektor. Alle fornuftige kombinasjoner av objektklasser er tillatt.

Direktiv ELEMULT

(ELEMENTVIS MULTIPLISERING)

Form: ELEMULT tilsvarende elementer i matrise1 og matrise2 multipliseres sammen og resultatet settes i matrise3.

Innobjekter: matrise1/vektor1/parameter1/rekordsett1, matrise2/vektor2/parameter2/rekordsett2

Utobjekt: matrise3/vektor3/parameter3/rekordsett3

For objektklasser innen gruppen matrise/vektor/parameter tillates alle kombinasjoner, som vanlig, bare de to første objekter (innobjektene) stemmer overens m.h.t. dimensjon. Direktivet fastsetter da dimensjonene til utobjektet lik dimensjonene til innobjektene.

Når rekordsett blir brukt må alle tre objekter være rekordsett som representerer pakkede matriser (se direktiv PAKKINN).

Direktiv GENOBJEKT

(GENERERER OBJEKT)

Form: GENOBJEKT navnord bestemmer type, parameter1 og parameter2, dimensjonene, og parameter3 er vanligvis en verdi som inngår i objekt på en eller annen måte.

Innobjekter: navnord, parameter1/liste1/navnord1, parameter2/liste2/navnord2,

parameter3/liste3/navnord3

Utobjekt: objekt

objekt er ved skrivende dato begrenset til flytendetails matrise/vektor/parameter, men det er meningen at alle typer objekter skal kunne genereres, til ulike formål.

De tre objektene med alternativ bestemmer alle et heltall etter følgende regler:

- (i) heltallsparameter : direkte
- (ii) flytendetailsparameter: etter konvertering
- (iii) liste : antall elementer i listen
- (iv) navnord : tallet blir 1

Direktivet lager mange forskjellige typer objekter, bestemt av navnord. Som navnord benyttes STANDARD-objekter, tillatte navnord er KMATRISSE/KUMULER/TESTDATA/IDENTSNUDD/STIGENDE/TILFELDIG/IDENTREKKE/DUPLISER.

Klasse til objekt blir bestemt ved deklarasjon av brukeren, og den må stemme overens med det alternativ spesifisert ved navnord.

- (i) navnord = KMATRISSE (Konstant MATRISSE, vektor eller parameter)

Det genereres en parameter1 x parameter2 matrise, med hvert element lik parameter3 i verdi. Standardverdier for parametrene er oppbevart i arkivet. Matrisen blir en flytendetailsmatrise.

Det kan være greit å bruke GENOBJEKT KMATRISSE sammen med direktiv MATRISSEDIM (se dette), slik at verdier for parameter1 og parameter2 kan fås fra et annet objekt. Resultatmatrisen (objekt) fra GENOBJEKT KMATRISSE kan brukes f.eks. til å summere rader eller kolonner eller til å addere et tall til hvert element i en matrise. Dette diskuteres videre under de neste alternativer.



(ii) navnord = KUMULER (KUMULER matrise eller vektor)

GENOBJEKT KUMULER lager en kumuleringsmatrise med følgende struktur:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

dvs. med 1-tall på og under hoveddiagonalen. Når en matrise premultipliseres med en kumuleringsmatrise blir resultatet kumulering over kolonnene i matrisen som med direktiv ADDERKUMULA (se C-2.4):

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix}$$

kumulerings-      matrise      kumulert  
matrise                              matrise

Ved å postmultiplisere med denne spesielle matrisen får man en lignende virkning på radene, sådan:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}$$

I dette siste tilfelle har man også brukt en annen spesiallaget matrise for å vende kolonne- rekkefølgen om. (Denne matrisen beskrives nedenfor.)

Med navnord = KUMULER blir dimensjonene til kumuleringsmatrisen bestemt som før av parameter1 og parameter2 (som normalt fås fra direktiv MATRISEDIM). Parameter1 og parameter2 må iallfall være like. Parameter3 skal man normalt sette lik 1.

(iii) navnord = TESTDATA (TESTDATA blir laget)

En matrise, vektor, eller parameter blir laget med følgende verdier:

```
1.001 1.002 1.003 1.004 ..... 1.555 ..... 1.999
2.001 2.002 .....
3.001 3.002 .....
.
.
.
.
.
12.001 12.002
.
.
.
```

Disse testdataene kan benyttes ved testing av direktiver for å kunne gjenkjenne data etter manipulering i direktiver som uttestes. Poenget er altså at data er valgt slik at rad og kolonne framgår av verdien av matriseelementer som genereres.

Varsel: en kvadratisk matrise laget på denne måten er singular.

Parameter3 blir ignorert men noe må man spesifisere (f.eks. STANDARD-objektet 1).

(iv) navnord = IDENTSKUDD (IDENTitetsmatrise blir laget SKUDD)

Det lages en kvadratisk matrise som er en speilvendt enhetsmatrise (se eksempel nedenfor). Den brukes for å invertere rekkefølge av rader eller kolonner, ved henholdsvis pre- og postmultiplisering. Strukturen er altså følgende:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Eksempel:

$$\begin{bmatrix} 1.1 & 1.2 & 1.3 & 1.4 \\ 2.1 & 2.2 & 2.3 & 2.4 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1.4 & 1.3 & 1.2 & 1.1 \\ 2.4 & 2.3 & 2.2 & 2.1 \end{bmatrix}$$

og

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1.1 & 1.2 \\ 2.1 & 2.2 \\ 3.1 & 3.2 \\ 4.1 & 4.2 \\ 5.1 & 5.2 \end{bmatrix} = \begin{bmatrix} 5.1 & 5.2 \\ 4.1 & 4.2 \\ 3.1 & 3.2 \\ 2.1 & 2.3 \\ 1.1 & 1.2 \end{bmatrix}$$

(v) navnord = STIGENDE (matrise/vektor lages med stigende tall) (ikke implementert)

Parameter3 spesifiserer begynnelsetallet, f.eks. 1. Tallene blir konvertert automatisk til flytende som før.

Eksempel 1:

ARKIVTAPE STANDARD.

GENOBJEKT STIGENDE 4 3 3 MAT.

MAT vil inneholde:

$$\begin{bmatrix} 3 & 4 & 5 \\ 6 & 7 & 8 \\ 9 & 10 & 11 \\ 12 & 13 & 14 \end{bmatrix}$$

Eksempel 2:

ARKIVTAPE STANDARD.

GENOBJEKT STIGENDE 3 1 10 VEK.

VEK vil inneholde:

$$\begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}$$

(vi) navnord = TILFELDIG (matrise/vektor lages med TILFELDIGE tall) .

parameter3 spesifiserer en startverdi, slik at identiske resultater produseres for identiske startverdier. Verdien bør helst være et tall i området av 10 milliarder for at resultatet skal være tilfeldig. Den såkalte "mixed congruential" metode brukes.

Ikke-null elementer genereres i tilfeldige posisjoner i matrisen, slik at direktiver som behandler pakkede matriser kan testkjøres grundig. Prosenten med null-elementer øker med økende matrisedimensjon, og gis ved formelen:

$$\text{Prosenten med nuller} = (\text{parameter1}/(\text{parameter1} + 15)) \times 100,$$

hvor parameter1 betraktes av direktivet som flytende tall uansett om det er det eller ikke.

Verdiene for de ikke-null elementer varierer tilfeldig mellom -1000 og +1000, uniformt fordelt.

Det blir føyet til ekstra elementer slik at ingen rad eller kolonne inneholder bare nuller. Resultatet blir at det er svært liten sjanse for at en kvadratisk matrise generert på denne måte blir singular. Ved skrivende dato fungerer TILFELDIG ikke (programmeringsfeil).

(vii) navnord = IDENTREKKE (IDENTitetsmatrise lages i en REKKE)

Det lages en elementær matrise som kan brukes til å ettermultiplisere en matrise for å duplisere den som helhet et visst antall ganger. Antall rader i elementærmatrisen er angitt ved parameter1, og skal være lik antall kolonner i matrisen som skal dupliseres. parameter2 brukes for å angi det antall ganger matrisen skal dupliseres. Antall kolonner i elementærmatrisen blir satt av direktivet lik parameter1 x parameter2.

parameter3 skal normalt settes lik 1. Hvis andre verdier blir brukt, vil en skalering også finne sted som resultat av multipliseringen.

For parameter1 = 2, parameter2 = 3, og parameter3 = 1, blir resultatet slik:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Hvis matrisen  $\begin{bmatrix} 6 & 10 \end{bmatrix}$  blir ettermultiplisert med matrisen ovenfor, blir sluttresultatet slik:

$$\begin{bmatrix} 6 & 10 & 6 & 10 & 6 & 10 \end{bmatrix}$$

(viii) navnord = DUPLISER (DUPLISER kolonner)

Mens IDENTREKKE er et ledd i duplisering av hele matriser (se ovenfor), er DUPLISER et ledd i duplisering av hver kolonne i en matrise et fast antall ganger. (Når parameter1 = 1 blir IDENTREKKE, DUPLISER og KMATRISSE ekvivalente.

Med denne forskjell i tolkingen har innobjektene lignende roller med DUPLISER som for IDENTREKKE.

For parameter1 = 2, parameter2 = 3, og parameter3 = 1, blir resultatet slik:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Hvis matrisen  $\begin{bmatrix} 6 & 10 \end{bmatrix}$  blir ettermultiplisert med matrisen ovenfor, blir sluttresultatet slik:

$$\begin{bmatrix} 6 & 6 & 6 & 10 & 10 & 10 \end{bmatrix}$$

Direktiv GJEM-LØSNINGER

(GJEM LØSNINGER tilsvarende forskjellige alternative datasett, mens man løser et ikke-lineært ligningssystem)

Form: GJEM-LØSNINGER med 7 innobjekter og 8 utobjekter, beskrevet nedenfor.

		<u>Objekt nr.</u>
Innobjekter:	LØSNINGSVERDIER (matrise)	1
	LØSNINGENE (liste)	2
	ENDOGEN-LAGER (matrise)	3
	AVVIK-LAGER (matrise)	4
	LØS-ALTERNATIVENE (liste)	5
	ANTALT (flyt. par.)	6
	AVVIK-GRENSE (flyt. par.)	7
Utoobjekter:	LØSNINGSVERDIER (matrise)	8
	LØSNINGENE (liste)	9
	ENDOGEN-LAGER (matrise)	10
	AVVIK-LAGER (matrise)	11
	NY-LØS-ALTERNATIVENE (liste)	12
	ANT- IKKE-LØST-ENNA (flyt. par.)	13
	DIVERGERING (flyt. par.)	14
	ENDOGEN (matrise)	15

Direktivet er av generell anvendelse ved løsning av ikke-lineære ligningssystemer, og er ikke bundet til noen spesiell løsningsmetode, som f.eks. Newton-metoden. Det ble utviklet til bruk i den økonomiske modellen MSG-3.

Direktivets rolle er å vurdere utviklingen av en iterativ løsningsprosess, slik at prosessen kan avbrytes når dette er ønskelig. Avbrytelsen skjer i så tilfelle ved å bruke direktivet HOPP i forbindelse med utobjektene ANT- IKKE-LØST-ENNA og DIVERGERING, somer forklart nedenfor. Direktivet vil også redde prosessen fra virkningene av en uheldig og feilaktig akselerering<sup>x</sup>, ved å gå tilbake til tidligere resultater.

Fire objekter opptrer både som innobjekter og utobjekter. Dette er nødvendig av den tekniske grunn at direktiver ikke kan forandre på dimensjonene til innobjekter. De må initialiseres i DATSY-programmet for at direktivet skal kunne brukes.

LØSNINGSVERDIER initialiseres med en rad med nuller.  
 LØSNINGENE initialiseres med et blankt element med bredde inntil 6 tegn.  
 ENDOGEN-LAGER initialiseres med startverdiene til de endogene variablene, lagret radvis.  
 AVVIK-LAGER initialiseres med de tilsvarende avvikene.

Direktivet forutsetter at brukeren vil løse ligningene for flere alternative og uavhengige datasett, i én kjøring. Det er likevel mulig å kjøre med bare ett alternativ.

Nullraden som LØSNINGSVERDIER initialiseres med skal ha dimensjon lik antall endogene variabler. ENDOGEN-LAGER og AVVIK-LAGER skal ha like dimensjoner. Ved initialisering skal de ha et antall rader like det antall alternativer som skal løses, og et antall kolonner lik antall endogene variabler. Listen LØS-ALTERNATIVENE skal initialiseres lik listen over de alternative datasett som ligningene skal løses parallelt for, med inntil 6 tegn i bredden.

Etter hvert som direktivet blir brukt flere ganger i en iterativ prosess, vil de løsningene som tilsvarer forskjellige alternativ bli lagt til i bunnen av LØSNINGSVERDIER, én rad til hvert alternativ. Rekkefølgen avhenger først på den rekkefølge de blir løst i, og så på rekkefølgen til alternativnavnene i listen LØS-ALTERNATIVENE (se nedenfor).

Samtidig som løsningsverdiene stables nedover vil de tilsvarende alternativnavnene fra listen LØS-ALTERNATIVENE flyttes fra listen LØS-ALTERNATIVENE til listen LØSNINGENE, i den samme rekkefølge.

Flytendetallsparameteren ANTALT brukes for å spesifisere på en lettvis måte hvor mange alternativer skulle løses totalt. I første runde settes den lik antall elementer i LØS-ALTERNATIVENE.

Betegnelser på de alternativene som fremdeles står uløst stables i listen NY-LØS-ALTERNATIVENE, samtidig som nye versjoner til ENDOGEN-LAGER og AVVIK-LAGER produseres, som gjelder bare de uløste alternativene.

<sup>x</sup> Se dokumentasjon om MSG-3.

Flytendetallsparameteren AVVIK-GRENSE skal være en liten verdi (f.eks. 0,01) som brukes av direktivet for å avgjøre om enkelte alternativer har konverget, altså om en riktig løsning er nådd for disse. Avvikene er funksjonsverdiene til de ligningene som skal løses, og suppleres direktivet i AVVIK-LAGER. Etter hvert som løsningsprosessen utvikler seg må resten av DATSY-programmet føye til nye endogene verdier og tilsvarende avvik radvis i bunnen av ENDOGEN-LAGER og AVVIK-LAGER, h.h.v. Direktivet gjør sitt beste for å finne en løsning blant alle disse, og konstruerer så utobjektet ENDOGEN som det beste aktuelle sett med endogene variabler, med alternativer lagret radvis.

Uheldige endogene verdier blir ignorert av direktivet, derfor vil ikke uheldige akselereringer påvirke resultatet. Direktivet vet når det er funnet en løsning, fordi avvikene blir da alle tilnærmet lik null, til en grense spesifisert i AVVIK-GRENSE.

ANT-IKKE-LØST-ENNÅ er en flytendetallsparameter som direktivet bruker for å registrere antall elementer som står igjen i NY-LØS-ALTERNATIVE, altså det antall alternativer som ikke er skikkelig løst ennå. Ved å teste denne parameteren i en HOPP-setning kan brukeren avbryte kjøringen når alle alternativer er blitt løst. NY-LØS-ALTERNATIVE brukes (sammen med inputlisten LØS-ALTERNATIVE) senere i DATSY-programmet, for å ekstrahere fra grunndataene til modellen nettopp de alternativene som det blir aktuelt å fortsette med.

DIVERGERING er en flytendetallsparameter som settes av direktivet lik null hvis det er fremdeles håp for løsning av minst ett av de alternativene spesifisert som input til direktivet i listen LØS-ALTERNATIVE. Hvis imidlertid alle gjenstående alternativer ser ut til å divergere i motsetning til konvergere vil parameteren DIVERGERING settes lik én. Ved å teste denne i en HOPP-setning kan brukeren avbryte kjøringen eller fortsetter til en ny løsningsmetode, når det ikke lenger lønner seg å fortsette den gitte løsningsprosess.

Se forøvrig: D. Walker: Using DATSY to Implement the Long Term Planning Model MSG-3. Working paper from the Central Bureau of Statistics. Oslo, 1975.

#### Direktiv HOPP

(HOPP ut av makroprogram)

Form: HOPP ut av makroprogram hvis matrisel/vektor1/parameter1 er NULL/IKKE-NULL til en grenseverdi representert ved parameter3.

Innobjekter: matrisel/vektor1/parameter1, navnord2, parameter3

Direktivet stopper med feilmelding hvis det blir brukt noe annet sted enn i et makroprogram. Virkningen er å hoppe forbi alle de resterende direktivsetningene i makroprogrammet, slik at DATSY-programmet fortsetter på forrige nivå, altså med det programmet som kjørte makroprogrammet.

Siden makroprogrammet kan kjøre andre makroprogrammer kan ganske kompliserte <hvis>-setninger utføres.

Første og tredje objekt skal være flytendetail. Andre objekt er et navnord med verdi lik sitt navn. Tredje objekt viser hvor nøyaktig første objekt må stemme med NULL eller IKKE-NULL. Hvis f.eks. det har verdien 1,5 vil parameter1 = 1,4 bli bedømt lik null. Normalt blir parameter3 gitt en veldig liten verdi, f.eks. 0,0001. Da vil parameter1 = 1,4 ikke bli bedømt lik NULL, og ingen hopping vil skje hvis NULL er blitt spesifisert.

Hvis f.eks. parameter1 = 0,5, navnord2 = "IKKE-NULL", og parameter3 = 0,001 vil et hopp finne sted. Forandres parameter1 til 0,00001 vil ingen hopp finne sted.

Når første objekt er matrise eller vektor må alle elementer i det oppfylle betingelsen før et hopp vil skje.

Se også direktivene MAKRO, LØKKE, VMAKRO og SAMMENLIGN.

Direktiv INVERS

(INVERSe elementer)

Form: INVERS elementene i matrise1 inverteres hver for seg og resultatet settes i matrise2.

Innobjekt: matrise1/vektor1/parameter1/rekordsett1

Utobjekt: matrise2/vektor2/parameter2/rekordsett2

Eksempel på bruk av direktivet:

INVERS A B.

ABS B B.

ELEMULT A B A.

Etter denne sekvens blir A endret slik at positive elementer settes lik +1, negative elementer settes lik -1, mens nullelementer inverteres ikke, og forblir null. (Null er definert som et tall hvis absolutte verdi er mindre enn  $1 \times 10^{-20}$ .)

Når innobjektet er matrise/vektor/parameter kan utobjektet være et hvilket som helst av disse, bare dimensjonene stemmer overens med innobjektet. Når innobjektet er rekordsett må dette representere en pakket matrise (se direktiv PAKKINN), og da må utobjektet også være et lignende rekordsett.

Direktiv KOLADDER

(KOLonner får ADDERT et element)

Form: KOLADDER adder til hver kolonne i matrise1/vektor1 det tilsvarende element i vektor2/matrise2/parameter2 og sett resultatet i matrise3/vektor3.

Innobjekter: matrise1/vektor1, vektor2/matrise2/parameter2

Utobjekt: matrise3/vektor3

Det blir kontrollert at antall kolonner i matrise1 er lik antall elementer i vektor2. matrise3 får samme dimensjon som matrise1.

1. element i vektor2 adderes til hvert element i 1. kolonne i matrise1, osv.

Direktiv KOLSKALER

(KOLonner SKALERes)

Form: KOLSKALER multipliser hver kolonne i matrise1 med det tilsvarende element i vektor2 og sett resultatet i matrise3.

Innobjekter: matrise1/vektor1/parameter1, vektor2/matrise2/parameter2

Utobjekt: matrise3

Det blir kontrollert at antall elementer i vektor2 stemmer overens med antall kolonner i matrise1. matrise3 får samme dimensjoner som matrise1.

1. element i vektor2 multipliseres med hvert element i 1. kolonne i matrise1 osv.

I tilfelle 2. objekt er matrise, må det bestå av én kolonne.

Jfr. direktiv RADSKALER.

Direktiv KONVREKMAT

(KONVerter REKordsett til MATrise)

Form: KONVREKMAT rekordsettl med listel/vektor1 og liste2/vektor2 blir konvertert til matrisel og tellingsmatrisen matrise2.

Innobjekter: rekordsett, listel/vektor1, liste2/vektor2

Utoobjekter: matrisel, matrise2

rekordsettl må ha en spesiell struktur med tre elementærfelt, hvorav de to første angir rad- og kolonneposisjoner i matrisel, mens det tredje angir en verdi som skal settes i matrisel i denne posisjon.

Denne verdien må være flytendetail. Rad- og kolonneposisjonene angis ved hjelp av alfa-numeriske opplysninger fra de to listene, listel og liste2, i de to første elementærfeltene. Radrekkefølgen i matrisel bestemmes av listel/vektor1, og kolonnerekkefølgen av liste2/vektor2.

Dimensjonene til matrisel blir bestemt av dimensjonene til radlisten listel og kolonnelisten liste2. Det samme gjelder for matrise2. Dersom rekordsettl inneholder flere rekorder som refererer seg til samme matriseposisjon blir disse aggregert, dvs. verdiene addert. Direktivet vil gi melding hvis dette skjer, for at man skal kunne sjekke at det ikke skyldes feil. Direktivet vil bruke tellingsmatrisen til dette formålet. Brukeren kan også bruke den til andre formål. matrise2 er en flytendetailmatrise som viser det totale antall elementer lest inn i hver posisjon i matrisel.

Dersom det er oppgitt en flytendetails vektor1 (vektor2) istedenfor listel (liste2) tolkes den som en vektor av nedre intervallgrenser, hvor siste element er den øverste intervallgrense. Dimensjonene til matrisel og matrise2 blir da én mindre enn dimensjonen til en slik vektor. Verdiene i vektor1 (vektor2) må være stigende. Hver nedregrenseverdi inkluderes i intervallet som den definerer. (Denne bruken av direktivet er ikke testet.)

Hvis det derimot er oppgitt en heltalls vektor1 (vektor2) istedenfor listel (liste2) tolkes den akkurat som om den var en liste. rekordsettl må da bli beskrevet med TALL i tilsvarende felt, som bør inneholde hele tall. (Denne bruken av direktivet er heller ikke testet.) Jfr. med direktiv PAKKUT.

Direktivet fungerer raskest når listel/vektor1 og radnavnene/radverdiene i rekordsettl er sortert tilsvarende. I tilfelle dette ikke gjelder virker direktivet likevel, men ikke så raskt. Usortert og med stor matrisel vil direktivet være håpløst tungvint i drift. (I verste fall vil direktivet stoppe, og gi feilmelding. Likevel kan man da forsere kjøringen ved å bruke S-opsjon. Andre eksekveringsopsjoner er innført for å utnytte en tom 'dedicated' UNIVAC 1108.)

Direktiv KOPIOMKODING

(KOPIer med OMKODING)

Form: KOPIOMKODING ta objekt1 og konverter det til objekt2 med en annen representasjon.

Innobjekt: objekt1

Utoobjekt: objekt2

Direktivet tar f.eks. sifre i objekt1 og konverterer disse til hele tall i objekt2. objekt1 og objekt2 kan være parameter, vektor, eller matrise, enten i flytende eller heltallsform, eller liste, navnord, rekord, eller rekordsett.

Alle mulige kombinasjoner er tillatte, såsant dimensjonene/deskripsjonene stemmer, bortsett fra at rekord og rekordsett ikke kan kopiomkodes til andre typer objekter enn rekord og rekordsett.

Flytende tall til hele tall medfører automatisk avrunding. Liste til vektor forutsetter at listen bare inneholder siffer, feilmelding gis ellers. Mer enn 12 siffer er ikke tillatt, da vi alltid går vegen om heltall.

Hvis objekt2 skal inneholde flytende tall må det fylles med flytende tall før direktivet kan brukes, f.eks. ved direktiv GENOBJEKT. Hvis ikke objekt2 er brukt før, og det er matrise, vektor eller parameter, vil resultatet bli hele tall.

### Direktiv LISTEPRODUKT

(LISTER brukes for å lage kartesiske PRODUKT)

Form: LISTEPRODUKT (liste1 med parameter2 tegn i bredden) x (liste3 med parameter4 tegn i bredden)  
= liste5.

Innobjekter: liste1, parameter2, liste3, parameter4

Utobjekt: liste5

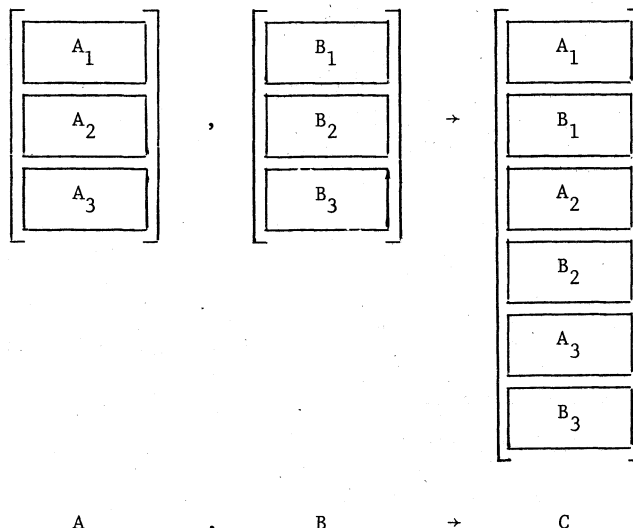
Direktivet forklares ved eksempler.

#### Eksempel 1

$$\begin{array}{ccc}
 \text{liste1} & & \text{liste3} & & \text{liste5} \\
 \begin{bmatrix} A1 \\ A2 \end{bmatrix} & \times & \begin{bmatrix} B1 \\ B2 \\ B3 \end{bmatrix} & = & \begin{bmatrix} A1B1 \\ A1B2 \\ A1B3 \\ A2B1 \\ A2B2 \\ A2B3 \end{bmatrix} \\
 \text{parameter1} = 2 & & \text{parameter4} = 2 & & 
 \end{array}$$

#### Eksempel 2

To matriser som består av delmatriser, som representerer alternative datasett i en modellkjøring, skal kombineres. Man ville gjerne brukt direktivet KOMBRAD eller en kombinasjon av KONVMATREK og KONVREKMAT, for at det hele skulle kunne styres med lister, men lister til dette formålet finnes ikke. I dette eksempel blir de generert ved hjelp av direktivet LISTEPRODUKT. Den operasjon som ønskes utført er som følger:





Det finnes 2 rader i hver  $A_i$  ( $i=1,3$ ), og 4 rader i hver  $B_i$ . Antall delmatriser i både A og B (dvs. antall alternativer) er forutsatt å være 3. De listene vi har er radlisten arad som betegner de 2 radene i hver  $A_i$ , radlisten brad som betegner de 4 radene i hver  $B_i$ , alternativlisten altliste som inneholder et element for hvert av de 3 alternativene i diagrammet ovenfor, og abrad som viser den rekkefølgen rader fra A og B skal få i resultatet.

<u>arad</u>	<u>brad</u>	<u>altliste</u>	<u>abrad</u>
<div style="border: 1px solid black; display: inline-block; padding: 2px;">a1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">a2</div>	<div style="border: 1px solid black; display: inline-block; padding: 2px;">b1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">b2</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">b3</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">b4</div>	<div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT2</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT3</div>	<div style="border: 1px solid black; display: inline-block; padding: 2px;">a1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">a2</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">b1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">b2</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">b3</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">b4</div>

Først brukes direktivet for å lage lister over alle radene i A og B henholdsvis:

LISTEPRODUKT altliste 4 arad 2 = Aradliste.

LISTEPRODUKT altliste 4 brad 2 = Bradliste.

<u>Aradliste</u>	<u>Bradliste</u>
<div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT1a1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT1a2</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT2a1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT2a2</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT3a1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT3a2</div>	<div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT1b1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT1b2</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT1b3</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT1b4</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT2b1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT2b2</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT2b3</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT2b4</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT3b1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT3b2</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT3b3</div> <div style="border: 1px solid black; display: inline-block; padding: 2px;">ALT3b4</div>

Så brukes det for å lage en liste over alle radene i resultat C:

LISTEPRODUKT altliste 4 abrad 2 = Cradliste.

Cradliste

ALT1a1  
 ALT1a2  
 ALT1b1  
 ALT1b2  
 ALT1b3  
 ALT1b4  
 ALT2a1  
 .  
 .  
 .  
 .  
 .  
 ALT3b3  
 ALT3b4

Følgende sekvens med sublistedirektiver vil da sette sammen delmatrisene i A og B:

SUBLISTGEN Asub Cradliste Aradliste.

SUBLISTGEN Bsub Cradliste Bradliste.

KOMBRAD STOKK Asub A Bsub B = C Cradliste.

(Se de relevante bruksanvisninger.)

Det samme kunne også oppnås på følgende vis, hvis kolonnenlisten heter kliste:

KONVMATREK A Aradliste kliste = mat-rek-1.

KONVMATREK B Bradliste kliste = mat-rek-2.

SETTPÅTOPP mat-rek-1 / mat-rek-2 = mat-rek-1.

SORTEK<sup>x</sup> mat-rek-1 på felt = mat-rek-1.

KONVREKMAT mat-rek-1 Cradliste kliste = C og mat-1.

Den første sekvens blir for komplisert hvis oppgaven er å sette sammen tre eller flere matriser. Den siste sekvens kan imidlertid lett utvides til å kombinere mange matriser. Den siste metoden er i tillegg mye mer effektiv når det gjelder å sette sammen store matriser hvis ikke disse er veldig tette (som store matriser sjeldent er). Den er alltid mye mer pålitelig, siden det finnes ikke den samme mulighet å bytte om listene i SUBLISTGEN-direktivet som eksisterer i den første sekvens, og det blir ikke mulig å forveksle sublistene Arad og Brad. Hvis Aradliste og Bradliste forveksles i den siste sekvens vil KONVMATREK stoppe med feilmelding. KONVREKMAT inneholder også kraftigere kontroller enn KOMBRAD. (Hvis KOMBRAD var sammensatt med SUBLISTGEN kunne den også inneholde kontroller av lignende art.)

I den siste sekvens er mat-rek-1 og mat-rek-2 forutsatt å være rekordsatt beskribert med felt, som inneholder tre elementerfelt, de to første alfanumeriske og det tredje et tallfelt (se under direktiv KONVMATREK). mat-1 er en vilkårlig matrise og blir ikke brukt videre i eksemplet.

Begge metoder oppnår fullstendig listestyring, dvs. ved å forandre i listen abrad kan hele operasjonen styres uten å forandre i DATSY-programmet. Dette er helt nødvendig hvis man skal kunne forandre i en stor modell på en pålitelig måte, og det er en av hovedmålsettingene med DATSY.

#### Direktiv LØKKE

(Kjør en LØKKE)

Form: LØKKE kjør liste som inneholder et makroprogram, parameterH ganger.

Innobjekter: liste, parameterH

Direktivet er det samme som MAKRO, bortsett fra at makroprogrammet kjøres et spesifisert antall ganger. parameterH må være en heltallsparameter.

Kjøringen av løkken kan avbrytes når som helst med direktiv HOPP (se dette). Da kan løkken kjøres et mindre antall ganger enn det spesifiserte, altså et variabelt antall ganger.

Grunnen til at LØKKE-direktivet er laget analogt med MAKRO-direktivet og ikke VMAKRO-direktivet er følgende. Et evt. VMAKRO-LØKKE-direktiv ville ta som input en liste med direktiv- og objektnavn som ville byttes inn, slik at forskjellige navn ble byttet inn i hver runde. Dette ville i og for seg være interessant, men ville både gjøre det umulig å kjøre et variabelt antall ganger uten å forandre innholdene i denne listen, og i tillegg resultere i et større antall objekter i DATSY-programmet (se diskusjonen under direktiv MAKRO).

Når man skal løse noe (f.eks. den økonomiske modellen MSG-3) ved en iterasjonsprosess, er det mye greiere om alle objekter heter det samme fra runde til runde. De kan likevel få forskjellig dimensjon og/eller innhold. Man har da heller ikke noe behov for å bytte inn nye navn som beskrevet under direktivet VMAKRO. Dette er særlig relevant når man skal bryte løkken med direktivet HOPP etter at prosessen har konvergert.

x Denne sortering er bare til nytte i tilfelle Cradliste er alfanumerisk sortert. Dette avhenger i sin tur på at altliste og abrad er sortert på denne måte. Det kan være fordelaktig å bruke alfanumerisk sorterte lister som en regel.

Direktiv MAKRO

(Kjør et MAKRO program)

Form: MAKRO kjør liste som inneholder et makroprogram.

Innobjekt: liste

Innledning

Bruken av direktivet beskrives utfyllende i DATSY-notat nr. 1, Arbeidsnotater fra Statistisk Sentralbyrå IO 74/33, "Makrosystemet" av David Walker. MAKRO ble introdusert for å løse et bestemt problem i forbindelse med kjøring av veldig store økonomiske modeller. Begrensningen på 200 setninger og 300 kort i et DATSY-program betydde at man måtte dele opp modellene inn i mange flere delprogram enn ønsket. Direktivet MAKRO tillater kjøring av et veldig stort antall direktivsetninger i et DATSY-program. Det har også ført til andre utviklinger (se direktivene VMAKRO, LØKKE og HOPP).

Bruken av MAKRO er fremdeles begrenset i noen grad av en svakhet i grunnstrukturen til den nåværende (1975) teksttolkingsdelen i DATSY, som betyr at tolkingen blir ueffektiv med et veldig stort antall dataobjekter, selv når antall setninger er lite. Videre fins det en øvre grense på hvor mange objekter med forskjellige navn man kan kjøre med i det nåværende DATSY-system. Dette antas å ligge et sted omkring 1 000.

P.g.a. disse to svakhetene er det fordelaktig å holde nede antall objekter med forskjellige navn som man bruker i et DATSY-program, hvis antallet ventes å overstige et par hundre.

Disse ulempene kan ventelig fjernes i en ny utgave til DATSY. Da ville makrosystemet også undergå vesentlige forandringer, som antydnet i DATSY-notat nr. 3, Arbeidsnotater fra Statistisk Sentralbyrå IO 75/9, "Presentasjon og drøfting av et forslag om makrodirektiver".

Bruk

liste bør inneholde direktivsetninger, som blir kjørt av setningen:

MAKRO liste.

Listen kan også inneholde andre setninger, som blir betraktet som tomme setninger. Varsler skrives ut om disse, i tilfelle de skyldes feilpunching.

Objekter som nevnes bare i et makroprogram må deklarerer i DATSY-teksten hvis de skal kunne hentes fra arkivtape. Ellers gjelder de normale regler angående deklarasjoner og beskripsjoner. Hvis et rekordsett, rekord eller felt deklarerer eksplisitt, må det også beskrives. Eksplisitt deklarasjon kan imidlertid unngås ved bruk av direktivet SAMLE.

Ved bruk av MAKRO utkobles testen på tomme innobjekter ved teksttolking av DATSY-programmer. Grunnen til dette er at et innobjekt ikke behøver å få noe innhold før eksekveringen av et makroprogram. Programmet stoppes da med feilmelding like før et tomt innobjekt forsøkes brukt.

Hvis mulig bør setningene i makroprogrammer testkjøres i en vanlig DATSY-tekst, siden en del feil da blir oppdaget ved et tidligere tidspunkt. Dette kan imidlertid være vanskelig p.g.a. begrensningen på antall setninger i et DATSY-program.

Linjer som begynner med punktum blir ignorert der de forekommer, og kommentarer kan da skrives på en slik linje. Dette er sterkt å anbefale, da man ellers fort mister oversikten i lange makroprogrammer. Venstreparentes har samme virkning, men bare innenfor makroprogrammer.

Når samme makroprogram kjøres flere ganger i en DATSY-kjøring blir innholdene i liste forandret til en mer maskinvennlig form. Dette sparer vesentlig på ressurser. liste er da ubrukbar i senere DATSY-kjøringer, men en eventuell opprinnelig utgave som ligger på arkivtape eller eksternt til DATSY blir ikke forstyrt. Hvis man ønsker å skrive ut liste i samme kjøring hvor den brukes med MAKRO, kan man kopiere den og skrive ut kopien (se direktiv KOPI).

Se også direktiv SAMLE.

Direktiv MULTIPLISER

(MULTIPLISERer numeriske objekter)

Form: MULTIPLISER parameter1/vektor1/matrise1 med parameter2/vektor2/matrise2 og sett resultatet i parameter3/vektor3/matrise3.

Innobjekter: parameter1/vektor1/matrise1, parameter2/vektor2/matrise2

Utobjekt: parameter3/vektor3/matrise3

Dette er et generelt direktiv som tillater de kombinasjoner av innobjekter som er listet nedenfor.

(i) MULTIPLISER matrise1 matrise2 matrise3.

Her brukes vanlig matrisemultiplikasjon, om dimensjonene til matrise1 og matrise2 stemmer overens, dvs. om kolonnedimensjon av matrise1 er lik raddimensjon av matrise2.

(ii) MULTIPLISER vektor1 matrise2 vektor3.

En vektor vil ordinært bli tolket som kolonnevektor. For denne multipliseringen betraktes vektor1 som en radvektor og da er multipliseringen definert og tillatt hvis dimensjonen av vektor1 er lik raddimensjon av matrise2.

(iii) MULTIPLISER matrise1 vektor2 vektor3.

Vanlig postmultiplisering av en matrise med en kolonnevektor - tillatt hvis dimensjonene stemmer.

(iv) MULTIPLISER vektor1 vektor2 parameter3.

Her beregner direktivet det indre produktet av vektor1 og vektor2, dvs. direktivet multipliserer hvert element i vektor1 med det tilsvarende elementet i vektor2 og adderer produktene. Antall elementer må stemme overens.

(v) MULTIPLISER matrise1 parameter2 matrise3.

MULTIPLISER vektor1 parameter2 vektor3.

Direktivet multipliserer hvert element i objektet med den angitte parameteren, som må bli spesifisert som 2. objekt (jfr. ADDER).

(vi) MULTIPLISER parameter1 parameter2 parameter3.

Et tall multipliseres med et annet tall.

Videre bemerkninger

(i) De forklaringer som er gitt ovenfor er ikke avhengige av objektenes klasser, men deres dimensjoner. Det gjør ikke noe hvilke objektklasser objektene har, bare de er fornuftige.

(ii) Alle objekter må inneholde flytende tall.

(iii) For å multiplisere en (n $\times$ 1) vektor/matrise med en (1 $\times$ m) tenkt linjevektor eller matrise, må det andre objektet være matrise.

(iv) Postmultiplisering av en matrise med en (kolonne-)vektor er mange ganger mer effektivt enn premultiplisering av den transponerte matrise med den tilsvarende radvektor. Dette er p.g.a. den radvise lagring av matriser.

(v) Se også direktivene PAKKINN, PAKKUT, PAKKPAKKMULT og VANPAKKMULT. Disse kan brukes når de objektene som skal multipliseres inneholder godt over 67 prosent med nuller, helst over 90 prosent. Hvis objektene likevel er små er det imidlertid ingen vits å bruke dem.

- (vi) Jfr. direktiv ELEMULT.
- (vii) Se bruken av MULTIPLISER beskrevet under direktiv GENOBJEKT, i forbindelse med matriser av 0'er og 1'er.

Direktiv OMSTABLER

(OMSTABLER en rekkematrise)

Form: OMSTABLER rekkematriksen matrise1 beskrevet av radlisten liste2 og sett resultatet i matrise3.

Innobjekter: matrise1, liste2/navnord2/parameter2

Utobjekt: matrise3

Direktivet utfører følgende operasjon:

$$\begin{bmatrix} A_1 \\ \text{-----} \\ A_2 \\ \text{-----} \\ A_3 \\ \vdots \\ \vdots \end{bmatrix} \rightarrow \begin{bmatrix} A_1 & | & A_2 & | & A_3 & \dots \end{bmatrix}$$

$A_1, A_2$  osv. er delmatriser stablet radvis, og må alle ha like dimensjoner. Antall rader i hver  $A_i$  ( $i=1, n$ ) angis av antall elementer i liste2, eller direkte av parameter2. parameter2 kan være enten hel- eller flytendetailparameter. I tilfelle navnord2 er brukt, betyr dette at hver  $A_i$  har bare én rad, siden et navnord tilsvarer en liste med ett element.

Hvis hver  $A_i$  har bare én kolonne er operasjonen identisk med transponering (jfr. direktiv TRANSPONER), men ellers ikke.

Den motsatte operasjon kan utføres ved hjelp av TRANSPONER slik:

TRANSPONER  $A = A$ .

OMSTABLER  $A$  liste2 =  $A$ .

TRANSPONER  $A = A$ .

Da tolkes liste2 som kolonnaliste til hver  $A_i$ .

Direktivet brukes ofte etter REKKEMULT-direktivet, for å stable alternative datasett kolonnevis istedenfor radvis.

Direktiv OPPLYSOBJEKT

(OPPLYS om OBJEKT)

Form: OPPLYSOBJEKT objekt.

Ved å bruke dette direktivet får man følgende opplysninger om objektet skrevet ut på linjeskriver (terminal):

- (i) Hva slags objekt det er deklart som.
- (ii) Om det er lagret på arkivtape, deklart i samme DATSY-program, eller ikke.

- (iii) I tilfelle det er lagret på arkivtape får man dato og klokkeslett da det ble lagret. I tilfelle det ikke er lagret på arkivtape (og følgelig er deklarerert i kjøringen hvor direktivet er brukt), får man opplysning om objektet på dette trinn i programmet har en bestemt struktur og verdi, eller bare en bestemt struktur, eller bare er deklarerert.
- (iv) Om objektet har en bestemt struktur, får man vite hva denne er.  
F.eks.:  
For matrise, får man dimensjonene.  
For rekord, felt og rekordsett får man navnene som ble brukt til å beskrivere objektet.
- (v) Hvilken attributter er tilknyttet objektet.
- (vi) Hvis K-opsjon (se B-6.4) er fastsatt, skrives innholdene ut av evt. tilknyttede objekter (f.eks. radliste og kolonneliste i tilfelle matrise). Se direktivene IDENTIFISER og SAMLE.

Direktiv PRINTMATLIST

(PRINT MATrise med LISTer)

Form: PRINTMATLIST matrisel/vektorl/parameterl med overskriften liste2, radnavnene liste3 og kolonnenavnene liste4, og parameterH5 siffer etter desimalpunktum.

Dette direktivet vil printe en matrise med overskrift, radnavn og kolonnenavn. Direktivet skriver 8 fulle kolonner ad gangen. Overskriften tenkes tatt fra kort lest med format (A80). Hvert kort blir én linje i overskriften. Hvert enkelt element i radnavnlisten tenkes å ha et fast format mellom (A1) og (A24), dvs. at ingen av disse navnene må overstige 24 tegn. Hvis radnavnene er lengre, vil direktivet bare skrive ut de første 24 tegn. Et varsel blir printet ut på en ny side etterpå. Kolonnenavnene kan inneholde inntil 80 tegn hver.

Radnavnene skrives ut på én linje hver, mens kolonnenavnene deles i grupper av 12 tegn som printes under hverandre.

Matrisen kan være av en hvilken som helst størrelsesorden, og direktivet vil om nødvendig skrive den ut over flere sider. Disse sidene kan eventuelt limes sammen etterpå.

Istedenfor liste3 og liste4 kan man anvende vektorer, matriser med én kolonne, rekordsett med ett elementærfelt, og enhver kombinasjon av disse. (Istedenfor liste2 kan man angi et hvilket som helst objekt med innhold, dvs. ikke NAVN, TALL, og FELT.)

Antall siffer etter desimalpunktum i tallene er gitt ved heltallsparameteren parameterH5 som må ha en verdi mellom 0 og 9. Det er totalt avsatt 13 posisjoner til hvert tall, derav går alltid 2 bort til fortegn og desimalpunktum.

Direktiv RADADDER

(RADer får ADDert et element)

Form: RADADDER adder til hver rad i matrisel det tilsvarende element i vektor/matrise2/parameter og sett resultatet i matrise3.

Innobjekter: matrisel, vektor/matrise2/parameter

Utobjekt: matrise3

Det blir kontrollert at antall rader i matrisel er lik antall elementer i vektor. matrise3 får samme dimensjoner som matrisel.

1. element i vektor adderes til hvert element i 1. rad i matrisel, osv.

RADADDER kan benyttes for å duplisere (lage mange eksemplarer av) f.eks. en VEKTOR som mangfoldige kolonner i en MATRISE.

F.eks.:

ARKIVTAPE STANDARD.  
 GENOBJEKT KMATRISE 50 50 0 A.  
 GENOBJEKT STIGENDE 50 1 1 LITEN-A.  
 MATRISE A.  
 VEKTOR LITEN-A.  
 RADADDER A med LITEN-A blir til A.

Matrisen A vil da inneholde vektoren LITEN-A som hver av sine kolonner. LITEN-A ble generert i eksemplet med stigende tall nedover kolonnen, 1,2,3,4,5, ... . Matrisen A ble opprinnelig generert av GENOBJEKT med bare nuller i. Jfr. direktiv GENOBJEKT.

#### Direktiv RADSKALER

(RADer SKALERes)

Form: RADSKALER multipliser hver rad i matrisel med det tilsvarende element i vektor2 og sett resultatet i matrise3.

Innobjekter: matrisel/vektor1/parameter1, vektor2/matrise2/parameter2

Utobjekt: matrise3

Det blir kontrollert at antall elementer i vektor2 stemmer overens med antall rader i matrisel. matrise3 får samme dimensjoner som matrisel.

1. element i vektor2 multipliseres med hvert element i 1. rad i matrisel, osv.

I tilfelle 2. objekt er matrise, må det bestå av én kolonne. Jfr. direktiv KOLSKALER.

#### Direktiv REKKEMULT

(En REKKE matriser MULTÍpliseres)

Form: REKKEMULT rekkematrisel med delradliste2 multipliseres med rekkematrise3 med delkolliste4 og gir rekkematrise5.

Innobjekter: rekkematrisel, delradliste2/navnord2/parameter2, rekkematrise3, delkolliste4/navnord4/parameter4

Utobjekt: rekkematrise5

rekkematrisel er forutsatt å være en oppstabling av matriser med like dimensjoner på toppen av hverandre, og rekkematrise3 en sammenstabling av matriser med like dimensjoner ved siden av hverandre. Tilsvarende par med matriser multipliseres. Antall kolonner i rekkematrisel og antall rader i rekkematrise3 må være like.

Direktivet er logisk sett det samme som MULTIPLISER fulgt av en ekstrahering av de delmatrisene langs hoveddiagonalen i resultatet, men er adskillig raskere i og med at urelevante delmatriser ikke beregnes.

Operasjonen kan skrives slik:

$$\begin{bmatrix} A_1 \\ \text{-----} \\ A_2 \\ \text{-----} \\ A_3 \\ \vdots \\ \vdots \end{bmatrix} \begin{bmatrix} | & | & | & \dots \\ B_1 & B_2 & B_3 & \dots \\ | & | & | & \dots \end{bmatrix} = \begin{bmatrix} A_1 B_1 \\ \text{-----} \\ A_2 B_2 \\ \text{-----} \\ A_3 B_3 \\ \vdots \\ \vdots \end{bmatrix}$$

delradliste2 og delkolliste4 må inneholde elementer tilsvarende h.h.v. radene og kolonnene i hver delmatrise  $A_i$  og  $B_i$ . Hvis det totale antall rader (kolonner) i rekkematrisel (rekkematrise3) ikke er et helt multiplum av antall elementer i delradliste2 (delkolliste4), stopper direktivet med feilmelding. Antall delmatriser i begge rekke-matriser må også være like.

Det henvises til direktiv GENOBJEKT når det gjelder tolking av rammen delradliste/navnord/parameter.

Tilsvarende direktiver for elementvis multiplisering, addering og subtrahering ville være overflødige. Tilsvarende rad- og kolonneskalering utføres ved direktivene STORADSKALER og TRANSPONER.

Se også under direktivet LISTEPRODUKT, som kan brukes på mange måter for å bearbeide objekter utgjort av delmatriser. Operasjoner kan til nød også bygges opp ved bruk av direktivene LØKKE og HOPP.

#### Direktiv SAMLE

(SAMLE objekter)

Form: SAMLE objekt under navnord.

Innobjekter: objekt, navnord

Etter at denne setning er utført vil objekt automatisk bli skrevet ut på en arkivtape, hvis navnord er skrevet ut på den (se B-3.6).

Direktivet kan brukes mange ganger med samme navnord og forskjellige objekter, slik at en hel rekke objekter samles under ett.

Hvis én eller flere av direktivene MAKRO, LØKKE og VMAKRO er brukt i teksten, vil alle de objektene som er samlet under navnord gjøres tilgjengelig, i den betydning at hvis navnord ligger på en arkivtape som er nevnt i en arkivtapesetning, og bare navnord er deklarerert, kan alle objektene samlet under navnord på tapen brukes uten videre. De behøver ikke deklarereres. Objektene kan også deklarereres og hentes hver for seg.

SAMLE kan brukeshierarkisk ved at navnord ovenfor samles under et annet navnord. Hierarkiet er ikke strengt, fordi et objekt kan samles under mange forskjellige navnord. Hierarkiet har altså ikke noen enkel trestruktur.

Det kan være fordelaktig å samle ett objekt under flere navnord, slik at man da kan velge akkurat det navnordet som tilsvarer den gruppen med objekter som man vil bruke.



Direktiv SKIFTREK

Form: SKIFTREK rekorder i rekordsett1 med rekorder i rekordsett2 hvis de stemmer overens i felt3 og plasser dem i rekordsett4.

Innobjekter: rekordsett1, rekordsett2, felt3

Utobjekt: rekordsett4

Alle rekordsettene må være identisk beskribert. felt3 må være en ekte undermengde av beskrripsjonen til rekordsettene. Det forutsettes at rekordsett1 og rekordsett2 er sortert på felt3. rekordsett4 vil være sortert på felt3 etter at direktivet er eksekvert.

Virkemåten er som følger. Det letes fortløpende gjennom rekordsett1 og rekordsett2. Hvis en rekord i rekordsett1 er lik en rekord i rekordsett2 i felt3, byttes rekorden i rekordsett1 ut med den tilsvarende i rekordsett2 og plasseres i rekordsett4. Rekorder som ikke er like i rekordsett1 og rekordsett2 flyttes direkte over i rekordsett4.

Direktiv SUBTRAHER

(SUBTRAHER numeriske objekter)

Identisk med direktivet ADDER, bortsett fra navnet og operasjonen.

Direktiv SUMKOL

(SUMmer KOLonner)

Form: SUMKOL Summer kolonnene i matrise/vektor1 og sett resultatet i vektor2/parameter.

Innobjekt: matrise/vektor1

Utobjekt: vektor2/parameter

Utobjektet får den klassen det hadde før direktivet, men hvis denne var urimelig vil direktivet stoppe og gi feilmelding.

Se alternativt direktiv GENOBJEKT.

Direktiv SUMRAD

(SUMmer RADER)

Form: SUMRAD Summer radene i matrise/vektor1 og sett resultatet i vektor2/parameter.

Innobjekt: matrise/vektor1

Utobjekt: vektor2/parameter

Det blir summert over radene slik at matrise går over til en vektor2 (kolonnevektor), mens vektor1 blir uendret. (Med SUMKOL, derimot, blir en vektor summert til en parameter.)

Hvis matrise bare har en rad, kan resultatet være parameter. Brukeren bestemmer selv klassen på utobjektet ved sin deklarasjon av objektet. Det blir testet at man ikke summerer en matrise med flere rader til en parameter.

Se alternativt direktiv GENOBJEKT.

Direktiv TRANSFORMTABELL

(TRANSFORMer TABELL)

Form: TRANSFORMTABELL parameter1/vektor1/matrise1 blir transformert ifølge tabellen matrise2 til parameter3/vektor3/matrise3.

Innobjekter: parameter1/vektor1/matrise1, matrise2

Utoobjekt: parameter3/vektor3/matrise3

matrise2 skal ha 2 kolonner og minst 2 rader. 1. kolonne skal være sortert stigende. Direktivet tar ett og ett element i 1. objekt, transformerer dem etter 2. objekt og plasserer resultatet i utobjektet.

Transformeringen skjer ved lineær interpolering. 1. kolonne i matrise2 skal gi interpoleringsintervallene og 2. kolonne de nye verdiene. Hvis et av elementene i innobjektet ligger utenfor interpoleringsskalaen, blir tilsvarende resultat satt lik største eller minste verdi, og varsel gis, men direktivet stopper ikke. Se også direktiv TRANSFORMER.

Direktiv TRANSPONER

(TRANSPONER matrise)

Form: TRANSPONER matrise1/vektor1 og sett resultatet i matrise2/vektor2.

Innobjekt: matrise1/vektor1

Utoobjekt: matrise2/vektor2

Direktivet utfører en ordinær transponering av en vilkårlig matrise. Når innobjektet er vektor må utobjektet bli deklartert som matrise, siden vektorer tolkes som kolonnevektorer.

Direktiv VMAKRO

(Kjør et Variabelt MAKROprogram)

Form: VMAKRO kjør makroprogrammet progliste med innbyttingsliste.

Innobjekter: progliste, innbyttingsliste

Direktivet er det samme som direktivet MAKRO, bortsett fra at man spesifiserer en liste med innbyttinger som vil erstatte i fast rekkefølge visse erstatningsindikatorer som brukeren har lagt inn i makroprogrammet. Erstatningsindikatorene består av 6 eller flere sammenhengende bindestreker. Hvis det f.eks. er 5 erstatningsindikatorer i progliste, må innbyttingsliste inneholde 5 elementer for å erstatte disse i fast tur og orden.

Innbyttingsliste må innleses med format (A12). Alle ord i setningene i progliste er mulige kandidater for utbytting, direktivnavn inkludert.

Direktiv XFORDEL

Form: XFORDEL eksogene anslag gitt i rekordsett1 (DXMAT) med styreskjema rekordsett2 (DSMAT) etter aggregeringsliste rekordsett3 (AGGMAT) og fordelingstall vektor1/INDEKS og forløpsliste liste1 (FLISTE) og årsliste liste2 (ALISTE) og nivåtall vektor2/INDEKS med liste3 og sett resultatet i rekordsett4 (MXMAT).

Dette direktivet er utviklet for å prosessere eksogen input til MODIS IV. Beskrivelsen nedenfor er derfor knyttet til denne anvendelsen, men gjelder også andre modeller med liknende struktur når det gjelder eksogene data, f.eks. MSG-modellen.

DXMAT, DSMAT, AGGMAT og MXMAT er rekordsett med følgende feltinndelinger:

DXMAT matrisenavn, variant, variabell, variabel2, år, alternativ, dimensjon, verdi.

DSMAT matrisenavn, forløp, år, alternativ.

AGGMAT variant-0, variant-1, variant-2, ..., variant-n.

MXMAT variabell-variabel2, år-forløp, verdi.

Brukerens eksogene forutsetninger er punchet på kort gruppevis etter en inndeling av de variable. Hver gruppe kalles en DX-matrise og utgjør et rekordsett. Direktivet behandler bare én DX-matrise av gangen. I rekordsettet som utgjør DX-matrisen, er det punchet bare én eksogen verdi i hvert kort. I beskrivelsen av DXMAT er matrisenavn navnet på DX-matrisen, f.eks. 'DX14'. Skjemaet for DX-matrisen har variabelnavn i forspalten og det fylles ut flere kolonner for ulike år eller alternativer. For de forskjellige kolonner innen en og samme matrise kan det benyttes ulike spesifikasjoner av forspalten. Disse kalles for varianter. Det vil alltid finnes en basisvariant, nullvarianten, som de øvrige er aggregeringer av. Dersom feltet variant i DXMAT er deklarerert som TALL angir feltverdien variantnr., dersom feltet er deklarerert som NAVN angir feltverdien det navn som er brukt i beskrivelsen av AGGMAT.

I hver rekord i DX-matrisen er det altså angitt navnet på matrisen og i hvilken variant av skjemaet den variable er spesifisert. Variabelkoden er gitt i to deler, variabell og variabel2. verdi vil oftest være oppgitt som endring fra et år til det neste. Sistnevnte år vil i så fall være oppgitt i rekorden. For dette året kan det være spesifisert flere alternativer av DX-matrisen. alternativ er derfor også oppgitt. Verdier som refererer seg til forskjellige forløp kan ha samme alternativ. Dimensjon angir om verdi er oppgitt som (i) nivåtall (N), (ii) absolutt endring fra foregående år (A) eller (iii) prosentvis endring fra foregående år (P).

DSMAT er styreskjemaet for DX-matrisen punchet som et rekordsett. Hver rekord inneholder navn på DS-matrisen. DSMAT skal inneholde én og bare en rekord for hver krysskombinasjon av forløp og år og for denne kombinasjon er angitt hvilket alternativ som kommer til anvendelse som endringstall eller nivåtall.

AGGMAT er en aggregeringsliste, dvs. et rekordsett som angir sammenhengen mellom ulike varianter av en variabelinndeling. Første felt vil alltid inneholde nullvarianten. De øvrige felter inneholder de øvrige varianter slik at en rekord i AGGMAT angir for en gitt variabel i nullvarianten i hvilken variabel i hver av de øvrige varianter den er inkludert.

FLISTE er en liste som inneholder de forløp som inngår i modellkjøringen. ALISTE er en liste som inneholder de beregningsår som inngår (i ordnet rekkefølge). vektor2 er en vektor med nivåtall for basisåret for de variable i DX-matrisen etter nullvarianten. liste er en liste med variabelnavn etter nullvarianten, altså til feltet variant-0 i AGGMAT. vektor1 er en vektor med tall som skal brukes som fordelingsnøkler ved bruk av andre varianter enn nullvarianten. I alminnelighet vil vektor1 være lik vektor2. INDEKS er et STANDARD-objekt som i dette direktivet angir at de variable i DX-matrisen er indekser som er lik én i basisåret. Behovet for fordelingstall og nivåtall for basisåret er da ikke til stede.

Direktivet skal bygge opp MXMAT fra de tall som er gitt i DXMAT. MXMAT må bygges opp etter stigende år fordi tallene i DXMAT oftest vil være gitt som endringstall fra år til år. I de tilfelle tallene i DXMAT er gitt etter en annen variant enn nullvarianten må tallene fordeles etter nullvarianten ved hjelp av AGGMAT dersom det dreier seg om relative anslag og også ved hjelp av vektorl dersom det gjelder absolutte anslag. DXMAT kan inneholde år og matrisialternativer som ikke er med i DSMAT, og disse skal i så fall ignoreres.

#### Direktiv YFORDEL

Form: YFORDEL rekordsett med korreksjoner av resultater fra tidligere kjøring matrisel med radliste listel og liste2, skal transformeres til absolutt endring i matrise2.

Dette direktivet er et spesialdirektiv for å prosessere inputskjemaet for virkningsmodellen til MODIS, men kan også tenkes brukt i andre modeller.

#### A. Objekter

1. rekordsett (YREK) er et rekordsett punchet fra inputskjemaet. Feltinndelingen i rekordsettet er følgende:  
VAR, ALT, DIM, VERDI.  
De tre første elementærfeltene er NAVN (1 ord), mens VERDI er TALL.
2. matrisel (YRES) er resultater (nivå tall) fra en tidligere kjøring for de variable som er med på inputskjemaet. YRES har YLISTE som radliste og YBALTLISTE som kolonneliste. Første kolonne i YRES er basisårstall.
3. listel (YLISTE) er en liste over de variable som er med på inputskjemaet, tilsvarer VAR-feltet i YREK.
4. liste2 (YBALTLISTE) er alternativliste for virkningsmodellkjøringen. Den inneholder alle år-forløps-kombinasjoner for de forløp som genereres i virkningsmodellkjøringen.
5. matrise2 (YMAT) inneholder korreksjonene i YREK transformert til absolutt endring. YMAT har YLISTE som radliste og YALTLISTE som kolonneliste.

#### B. Virkemåte

##### 1. Kontroller

Direktivet skal kontrollere

- (i) at YREK ikke inneholder andre verdier i VAR-feltet enn de som er med i YLISTE,
- (ii) at YREK ikke inneholder andre verdier i ALT-feltet enn de som er med i YBALTLISTE,
- (iii) at YRES har dimensjoner som tilsvarer YLISTE og YBALTLISTE,
- (iv) at DIM-feltet ikke inneholder andre verdier enn 'A', 'N', 'P', 'B' eller 'F'.

##### 2. Transformasjoner

For VAR-ALT kombinasjoner som ikke fins i YREK skal det tilsvarende element i YMAT være lik null. For alle andre VAR-ALT kombinasjoner skal det beregnes et element i YMAT ut fra verdien i VERDI-feltet. Beregningsmåten avhenger av verdien i DIM-feltet. De alternative verdier for DIM er forklart i punkt (i) - (v) nedenfor. NB. I punkt (ii) og (v) er det brukt symbolet ALT<sub>-1</sub>. Dette betyr følgende. Verdiene for ALT (og elementene i YBALTLISTE) består av fire karakterer. De to første vil alltid være numeriske (0-9) og betegner et årstall, f.eks. '78'. De to siste er ikke

nødvendigvis numeriske og betegner forløpet, f.eks. 'A2'. Hvis f.eks. ALT = '78A2', betyr ALT<sub>-1</sub> det samme forløp i foregående år, altså ALT<sub>-1</sub> = '77A2'. ALT og ALT<sub>-1</sub> brukes til å referere til kolonner i YRES og YMAT. Hvis ALT<sub>-1</sub> viser til basisåret, skal YRES (VAR, ALT<sub>-1</sub>) tolkes som YRES (VAR, 1) og YMAT (VAR, ALT<sub>-1</sub>) som null i punkt (ii) og (v).

(i) DIM = 'A'. Absolutt endring.

$$YMAT (VAR, ALT) = VERDI$$

(ii) DIM = 'N'. Nytt nivå-tall.

$$YMAT (VAR, ALT) = VERDI - YRES(VAR, ALT_{-1}) - YMAT(VAR, ALT_{-1})$$

(iii) DIM = 'P'. Prosentvis endring av samme års nivå-tall.

$$YMAT(VAR, ALT) = 0,001 \cdot VERDI \cdot YRES(VAR, ALT)$$

(iv) DIM = 'B'. Prosentvis endring av basisårets nivå-tall.

$$YMAT(VAR, ALT) = 0,001 \cdot VERDI \cdot YBASIS(VAR)$$

(v) DIM = 'F'. Prosentvis endring av foregående års nivå-tall.

$$YMAT(VAR, ALT) = 0,001 \cdot VERDI \cdot (YRES(VAR, ALT_{-1}) + YMAT(VAR, ALT_{-1}))$$

### 3. Sorteringsforutsetninger

YREK forutsettes sortert etter YLISTE.

### 3. Rettelser til Håndbok for bruk av DATSY

Innholdsfortegnelsen:	Side
Føy til B-1.6 Hele og flytende tall .....	22
Føy til B-3.9 Opsjonssetningen .....	29
Tittelen til B-6.4 blir simpelthen:	
Opsjoner .....	40

s. 7, midten: Hele setningen: For praktiske formål ... slettes.

s.12, linje 3: Sett \* etter ordet PRINTMAT.

Føy til en fotnote: \* Direktivet PRINTMAT er ikke implementert ved 1. mai 1975, og direktivet PRINTVERDI eller direktivet PRINTMATLIST brukes istedet.

s.13: \* P1= 1 464 rettes til \* P1 = 1464.0, 2 steder.

708 rettes til 708.0, 2 steder.

Føy til i midten: Tallene uttrykkes med .0 for å vise maskinen at de er flytende tall (se B-1.6).

Føy til like før avsnitt A-2.4: Formatet F66, som er lik (F14.8), forteller maskinen i dette tilfellet at tallene skal innleses som flytende tall (se B-1.6).

s.14, siste linje: leses ut rettes til lagres.

s.17, etter deklarasjonsetningen REKORDSETT osv.: Føy til:

Siden ordet rekordsett ligger bokstavert i systemdelen til DATSY som rekordset (med én t) må setningen ved skrivende dato være slik:

REKORDSET PERSONREGSTR KOMMUNEREGSTR.

Denne forskjell vil imidlertid bli ignorert i det følgende siden forholdet ventes rettet før eller senere.

s.21, B-1.1: Innholdene til (i) erstattes med bokstaver, siffer og bindestrek.

Føy til: Tegn av type (iii) kalles for tomme tegn.

B-1.2: 2. setning blir: Et ord avsluttes av ethvert tomt tegn, punktum, eller ny linje. I et ord blir de inntil 12 første tegn registrert.

4 linjer fra bunnen: Fjern ordet OPPLYSOBJEKT.

s.22: Føy til avsnitt B-1.6 som følger:

#### B-1.6 Hele og flytende tall

De fleste direktiver beregnes med flytende tall, dvs. tall som muligens inneholder brøkdeler. Disse skrives normalt med punktum der hvor man vanligvis ville brukt komma. Hvis tallet inneholder ingen brøkdel men skal være flytende ifølge brukerbeskrivelsen til et direktiv, skriv det med et punktum, eller punktum fulgt av null, til slutt.

Hele tall er tall uten mulighet for brøkdeler, og de skrives alltid uten punktum, for å skille dem fra flytende tall. De to typer tall representeres forskjellig i maskinen.

(Kommentar: Ifølge redaktørens egen mening bør hele tall som innobjekter og utobjekter fjernes fullstendig fra det nåværende sett med direktiver, selv om muligheten til å bruke dem kanskje bør ligge i enhver utgave av systemdelen til DATSY. Systemdelen bør i hvert fall forandres slik at tall er forutsatt flytende hvis ikke noe annet er sagt, uansett om punktum/komma er spesifisert eller ikke. Avsnitt B-1.6 bør kunne slettes fullstendig etter en slik utvikling.)

B-2.1, 2. setning blir: Hvis brukeren f.eks. skal arbeide med en matrise vil matrisen være et dataobjekt.

s.23, B-2.3: Føy til 1. setningen: , og fremmedtape (se B-2.5).

s.24, midten: sentral erstattes 3 ganger med katalog.

avsnitt d. FREMMETAPE: Føy til ved begynnelsen: Dette er ikke implementert ennå (1. juni 1975).

s.26, linje 3: komma blir andre tomme tegn.

linje 5: er blir må bli av brukeren.

B-3.3: 2. setning blir: Arkivtapasetningen har ingenting felles med en deklarasjonsetning bortsett fra skrivemåten, som er identisk. Arkivtapasetningen må gjentas i hvert eneste DATSY-program som anvender den aktuelle arkivtapen.

Føy til i bunnen av B-3.3 et nytt avsnitt:

Vitsen med deklarasjonsetninger generelt er det å registrere at en viss sammenhengende sekvens med tegn representerer et dataobjekt, og at dette er av en viss type. I tilfelle arkivtaper bruker man en NYTAPE-deklarasjon til dette formål (se ovenfor).

5 linjer fra bunnen: Setningen blir: Objekter av klassene FELT, REKORD og REKORDETT må som nevnt ovenfor alltid beskrives når de deklarerer.

s.27, linje 26: imidlertid rettes til midlertidig.

midten: Føy til:

(iv) Direktivnavnene MAKRO, VMAKRO og LØKKE må ikke substitueres (hverken inn eller ut) da dette kan forårsake vanskeligheter, generelt sett. De kan imidlertid substitueres med hverandre.

s.29, ved bunnen: Føy til:

#### B-3.9 Opsjonsetningen

I den versjon til DATSY som benyttes på Honeywell-Bull 6000 Series maskiner, må alle DATSY-programmer begynne med opsjonsetning. Denne er av formen

OPSJON. (ingen opsjon fastsettes)

eller f.eks.

OPSJON A B C. (opsjoner A, B og C fastsettes)

Se forøvrig B-6.4, hvor de forskjellige opsjoner som brukeren kan tenkes å benytte er forklart. Opsjonsetningen understrekes ikke, og gjengis ved hodet av programlistingen.

s.40, B-6.4: Tittelen blir: Opsjoner.

1. og 2. avsnitt og halvparten av 1. linje i 3. avsnitt erstattes med:

Opsjoner skrives på forskjellige måter, avhengig av hvilken maskintype man bruker, UNIVAC 1100 Series eller Honeywell-Bull 6000 Series.

I det første tilfelle må man begynne teksten med en opsjonsetning, uansett om man vil kjøre med opsjoner eller ikke. Opsjonsetningen skrives f.eks. slik:

OPSJON A B C.

I eksempelet er A, B og C betegnelser på de opsjoner som skal gjelde fra begynnelsen av kjøringen, og gjennom hele kjøringen hvis ikke de blir satt om av et direktiv. (Det finnes ennå ikke noe direktiv til dette formål.) Hvis man ikke vil sette noen opsjoner skriver man simpelthen OPSJON.

som første setning i DATSY-programmet.

På UNIVAC 1100 Series maskin legger man programmet straks etter styrekortet

"XQT DATSY

og hvis ønsket, kan opsjoner settes i dette tilfelle ved å skrive f.eks.

"XQT,ABC DATSY

W-opsjon bør alltid benyttes når man skal lese dataobjekter fra arkivtaper, da den bidrar til tidlig fjerning av tapene fra tapestasjonene og derfor tillater at andre jobber kjøres på maskinen. Alle de andre opsjoner som er beskrevet her ...osv.

s.53, midten: attribut 26 rettes til attribut 16.

#### 4. Oversikt over dokumentasjon av DATSY

##### 4.1 Innledning

Det er vanlig å skille mellom brukerdokumentasjon og teknisk dokumentasjon. Hvem som er bruker med henhold til DATSY krever forklaring, og de organisatoriske prinsipper som ligger bak definisjonen er ikke trivielle.

Det generelle organisasjonsprinsipp som Byrådet bruker når det gjelder DATSY er følgende. Systemprogrammerere ansatt i Systemkontoret mottar bestillinger for programmering av nye Fortranmoduler (direktiver), i form av brukerbeskrivelser (se avsnitt 2). Systemkontoret vurderer bestillingene før de programmeres, og det mest vanlige vil faktisk være at bestillinger kanselleres og at oppdragsgiveren henvises til eksisterende direktiver. Det viser seg nemlig at det eksisterende sett med direktiver som oftest er fullt tilstrekkelig. Hvis dette ikke er tilfelle, går prosessen videre. I vanskelige tilfeller vil Systemkontoret drøfte med oppdragsgiveren angående omformulering av direktivet, og dette krever en del innsikt i oppdragsgiverens problemstilling. I enklere tilfeller vil en slik omformulering ikke være på sin plass, og Systemkontoret programmerer direktivet og leverer det nøyaktig som spesifisert i brukerbeskrivelsen.

Med bakgrunn i denne organisasjonsformen, og i det faktum at sosialøkonomer og andre finner det mulig selv å skrive DATSY-programmer, anser vi brukeren som en som skriver DATSY-programmer, men som ikke nødvendigvis kan Fortran eller andre EDB-språk. Brukerens utgangspunkt er brukerdokumentasjonen, altså Håndbok for bruk av DATSY [b]\*, pluss de tillegg som måtte komme til denne (slik som dette heftet), og et utvalg av DATSY-notater [s,u]. En bruker er altså ikke ansatt som programmerer. Brukeren anvender DATSY for å løse de beregningsproblemer han treffer innenfor sitt felt, ved hjelp av dokumentasjonen nevnt ovenfor.

En analogi er at folk som ikke er ansatt for å regne på bordregnemaskiner likevel finner det hensiktsmessig å bla i de bruksanvisningene som vanligvis følger med slike maskiner, og bruker slike maskiner daglig i sitt arbeid.

Resten av dokumentasjonslistene gitt nedenfor er ment for folk som ikke nødvendigvis er brukere av DATSY, men som i større eller mindre grad har et visst kjennskap til EDB. Det synes naturlig å skille ut følgende kategorier av personer. (De relevante henvisningene gis i hakeparentes i hvert tilfelle).

- (i) EDB-eksperter som vil vurdere DATSY i forhold til andre systemer [b, c, l, m, n, q, r, s, t, u]. NB. [m] beskriver et delsystem som ikke er implementert foreløpig.
- (ii) Systemprogrammerere som har fått i oppdrag å lage nye Fortranmoduler til DATSY [d, p, r, t]. NB. [t] inneholder en vurdering av sammensatte vis á vis ordinære direktiver.
- (iii) Eksperter i bygging av økonomiske modeller som vil vurdere bruk av DATSY i dette feltet [u].
- (v) Interesserte som ikke kan lese norsk [u].
- (vi) Personer som vurderer forbedringer og forenklinger i DATSY [b, c, g, l, m, n, p, q, s, t, u].
- (vii) Personer som skal rette feil i systemdelen til DATSY [a, g, h, q, s].

\* Bokstaver i hakeparentes refererer til de listene over utgitte publikasjoner og planlagte publikasjoner, som gis nedenfor i dette avsnitt.



#### 4.2 Utgitte publikasjoner

- a. Draper, D.: DATSY - Program for behandling av datakortene. DATSY Rapport Nr. 9. Norsk Regnesentral. Oslo, 1971.
- b. Håndbok for bruk av DATSY. Statistisk Sentralbyrås Håndbøker 33. Oslo, 1974.
- c. Jacobsen, O.: DATSY - Versjonsgenerator. DATSY Rapport Nr. 12. Norsk Regnesentral. Oslo, 1974. (Rapport Nr. 11 ble sløyfet).
- d. Jacobsen, O.: Versjonsgenereringssystemet. DATSY-notat nr. 2. Arbeidsnotater fra Statistisk Sentralbyrå IO 75/10. Oslo, 1975.
- e. Jacobsen, O.: Vedlikehold av versjonsgenereringssystemet. Brukernotat i Systemkontoret ved Statistisk Sentralbyrå. A255/0, Oslo, februar 1975.
- f. Kristoffersen, E.: "Databanken" - Implementering Rapport Nr. 4. Norsk Regnesentral. Oslo, 1969.
- g. Kristoffersen, E.: DATSY - Program for behandling av klarteksten. DATSY Rapport Nr. 8. Norsk Regnesentral. Oslo, 1971.
- h. Kristoffersen, E., Krogdahl, S. og Totland, H.: DATSY - System- og hjelperutiner. DATSY Rapport Nr. 10. Norsk Regnesentral. Oslo, 1971.
- i. Spurkland, S.: Et brukerorientert datasprog. Norsk Regnesentral. Oslo, 1969.
- j. Spurkland, S.: "Databanken" - Implementering Rapport Nr. 1. Norsk Regnesentral. Oslo, 1969.
- k. Spurkland, S.: "Databanken" - Implementering Rapport Nr. 3. Norsk Regnesentral. Oslo, 1969.
- l. Spurkland, S.: DATSY - Generell Beskrivelse. DATSY Rapport Nr. 5. Norsk Regnesentral. Oslo, 1971.
- m. Spurkland, S.: Datakompatibilitet mellom DATSY og andre systemer. Arbeidsnotat, Norsk Regnesentral. Oslo, 1972.
- n. Spurkland, S.: Rapportgenerator i DATSY. Arbeidsnotat, Norsk Regnesentral. Oslo, 1973.
- o. Totland, H.: "Databanken" - Implementering Rapport Nr. 2. Norsk Regnesentral. Oslo, 1969.
- p. Totland, H.: DATSY - Manual for innføring av nye direktiver. DATSY Rapport Nr. 6. Norsk Regnesentral. Oslo, 1971.
- q. Totland, H.: DATSY - Implementering av et brukerorientert dataspråk. DATSY Rapport Nr. 7. Norsk Regnesentral. Oslo, 1971.
- r. Totland, H.: DATSY - Verktøy for modellformulering og databehandlingen. Foredrag på Norddata, 21.-23. juni 1971. Norsk Regnesentral. Oslo, 1971.
- s. Walker, D.: Makrosystemet. DATSY-notat nr. 1. Arbeidsnotater fra Statistisk Sentralbyrå IO 74/33. Oslo, 1974.
- t. Walker, D.: Presentasjon og drøfting av et forslag om makrodirektiver. DATSY-notat nr. 3. Arbeidsnotater fra Statistisk Sentralbyrå IO 75/9. Oslo, 1975.

#### 4.3 Planlagte publikasjoner

- u. Walker, D.: Using DATSY to Implement the Long Term Planning Model MSG-3. Working Paper from the Central Bureau of Statistics. Oslo, 1975.

5. Alfabetisk liste over direktiver, med sidereferanser og status på Honeywell-Bull 6000 Series maskiner pr. 1. mai 1975

NB. (i) Hvis et direktiv brukt i et DATSY-program ikke understrekes av DATSY, kan det bety enten at direktivnavnet er feilbokstavert, eller at direktivet ikke er med i den DATSY-versjonen som brukes. (Forskjellige DATSY-versjoner med forskjellige utvalg av direktiver brukes i forskjellige prosjekter.)

(ii) \* betegner fortsatt beheftet med visse svakheter.

(iii) H betegner Håndbok for bruk av DATSY [1]. Andre sidehenvisninger refererer til avsnitt 2 ovenfor.

NAVN PÅ DIREKTIVET	BRUKER-BESKRIVELSEN FINS VED	STATUS PR. 1. mai 1975 (H6060 VERSJON)	NAVN PÅ DIREKTIVET	BRUKER-BESKRIVELSEN FINS VED	STATUS PR. 1. mai 1975 (H6060 VERSJON)
ABS	s. 63,H	virker	EKSAGGREDREK	s. 82,H	virker
ADDER	s. 2	virker	EKSDELALT-2	s. 4	virker
ADDERKUMULA	s. 64,H	virker	EKSDELALT-5	s. 5	virker
ADDRADADDER	s. 106,H	virker	EKSKOL	s. 5	virker
ADDTILDELKOL	s. 74,H	ikke testet	EKSMAT	s. 6	virker
ADDTILDELMAT	s. 74,H	virker	EKSRAD	s. 6	virker
ADDTILDELRAD	s. 74,H	ikke testet	EKSREKORD	s. 83,H	virker*
ANTDOBLIST	s. 53,H	ikke testet	ELEMULT	s. 7	virker
BEREGN-KUMULASKATT	s. 107,H	virker	ENDRING	s. 65,H	virker
BEREGN-MARPINNTEKT	s. 107,H	virker	GENGAMMA	s. 109,H	virker
BEREGN-PAALOPENDE-SKATT	s. 108,H	virker	GENGAMMADATA	s. 111,H	ikke implementert
BIGRADSKALER	s. 64,H	ikke testet	GENMAT	s. 59,H	ikke implementert
BYTTFELT	s. 81,H	virker	GENOBJEKT	s. 7	virker*
BYTTKOL	s. 75,H	virker	GENREK	s. 59,H	ikke implementert
BYTTMAT	s. 75,H	virker	GJEM-LØSNINGER	s. 10	virker
BYTTRAD	s. 75,H	ikke testet	GRUPPELISTE	s. 87,H	virker
BYTTREK	s. 81,H	virker	HOPP	s. 12	virker
CNVADD	s. 58,H	ikke implementert	IDENTIFISER	s. 56,H	virker
DELHORISONT	s. 3	virker	IMINUSA	s. 66,H	ikke testet
DELKOLVIS	s. 4	virker	INVERS	s. 13	virker
DELLISTE	s. 87,H	virker	INVERTER	s. 67,H	virker*
DELVEKTOR	s. 93,H	ikke implementert	I-0	s. 43,H	ikke implementert
DIAGONALISER	s. 64,H	virker	KOLADDER	s. 13	virker
DIVIDER	s. 4	virker	KOLONNE-LAG	s. 67,H	virker
DOKUMENTER	s. 53,H	ikke testet	KOLSKALER	s. 13	virker
			KOMBDPELLISTE	s. 89,H	virker

NAVN PÅ DIREKTIVET	BRUKER-BESKRIVELSEN FINS VED	STATUS PR. 1. mai 1975 (H6060 VERSJON)	NAVN PÅ DIREKTIVET	BRUKER-BESKRIVELSEN FINS VED	STATUS PR. 1. mai 1975 (H6060 VERSJON)
KOMBKOL	s. 77,H	virker	RADIVKOL	s. 96,H	ikke implementert
KOMBRAD	s. 78,H	virker	RADSKALER	s. 22	virker
KONVMATREK	s. 60,H	virker	RAEKSPOL	s. 97,H	ikke implementert
KONVREKMAT	s. 14	virker	RAELOBSINF	s. 97,H	ikke implementert
KOPI	s. 89,H	virker	RAELOBSUTF	s. 97,H	ikke implementert
KOPIOMKODING	s. 14	virker	RAFORSKYV	s. 98,H	ikke implementert
LESBCDTAPE	s. 61,H	ikke testet	RAHENTFELT	s. 98,H	ikke implementert
LET	s. 44,H	ikke implementert	RAHENTKOL	s. 99,H	ikke implementert
LINK	s. 83,H	virker	RAKBRTKOL	s. 99,H	ikke implementert
LISTEPRODUKT	s. 15	virker	RAKLARGJØR	s. 99,H	ikke implementert
LØKKE	s. 17	virker	RAKOPIKOL	s. 100,H	ikke implementert
MAKRO	s. 18	virker	RAKORRMAT	s. 100,H	ikke implementert
MATRISEDIM	s. 56,H	virker	RAKUBKOL	s. 101,H	ikke implementert
MULTDELKOL	s. 79,H	ikke testet	RAKVADKOL	s. 101,H	ikke implementert
MULTDELMAT	s. 79,H	virker ikke	RAKVRTKOL	s. 101,H	ikke implementert
MULTDELRAD	s. 79,H	ikke testet	RALNKOL	s. 102,H	ikke implementert
MULTIPLISER	s. 19	virker	RAMINVERD	s. 102,H	ikke implementert
NAVNGIKOL	s. 93,H	ikke implementert	RAMULTKOL	s. 102,H	ikke implementert
NY-SIDE	s. 94,H	ikke implementert	RAPRIKORRMAT	s. 103,H	ikke implementert
OMSTABLER	s. 20	virker	RAPRIMSTAV	s. 103,H	ikke implementert
OPPLYSOBJEKT	s. 20	ikke testet	RAREGRESJON	s. 103,H	ikke implementert
PAKKINN	s. 57,H	virker	RASLUTTDATA	s. 105,H	ikke implementert
PAKKPAKKMULT	s. 69,H	virker	RASUBKOL	s. 106,H	ikke implementert
PAKKUT	s. 57,H	virker	REDREKORD	s. 85,H	virker
PRINT	s. 46,H	virker	REKKEMULT	s. 22	virker
PRINTMAT	s. 49,H	ikke implementert	SAMLE	s. 23	virker
PRINTMATLIST	s. 21	virker	SAMMENLIGN	s. 57,H	virker
PRINTNRADER	s. 50,H	ikke implementert	SENDPRINT	s. 51,H	ikke implementert
PRINTREKSET	s. 85,H	virker ikke	SETTKOL	s. 80,H	virker
PRINTVERDI	s. 51,H	virker	SETTPÅTOPP	s. 89,H	virker
RAABSKOL	s. 95,H	ikke implementert	SETTVESIDE	s. 90,H	virker
RAADDKOL	s. 95,H	ikke implementert	SKATT-ETTER-INNTEKTSTRINN	s. 112,H	virker
RABININF	s. 95,H	ikke implementert	SKIFTREK	s. 24	virker
RABINUTF	s. 96,H	ikke implementert	SKRIVBCDTAPE	s. 62,H	virker
RADADDER	s. 21	virker	SOLVE	s. 70,H	virker*

NAVN PÅ DIREKTIVET	BRUKER- BESKRIVELSEN FINS VED	STATUS PR. 1. mai 1975 (H6060 VERSJON)	NAVN PÅ DIREKTIVET	BRUKER- BESKRIVELSEN FINS VED	STATUS PR. 1. mai 1975 (H6060 VERSJON)
SORTER	s. 85,H	virker	TRANSFORMER	s. 71,H	virker
STORADSKALER	s. 113,H	virker	TRANSFORMTABELL	s. 25	virker
SUBEKSMULT	s. 80,H	ikke testet	TRANSPONER	s. 26	virker
SUBLISTGEN	s. 57,H	virker	TRAPESMETODE	s. 113,H	virker
SUBLISTKON	s. 94,H	ikke implementert	TRIANGULER	s. 72,H	ikke testet
SUBTILBAKE	s. 70,H	ikke testet	TRITILINVERS	s. 73,H	ikke testet
SUBTRAHER	s. 24	virker	VANPAKKMULT	s. 73,H	ikke testet
SUMKOL	s. 24	virker	VMAKRO	s. 26	virker
SUMRAD	s. 24	virker	XFORDEL	s. 26	virker
TAVEKKDOBB	s. 58,H	virker	YFORDEL	s. 27	virker

Referanser

- [1] Håndbok for bruk av DATSY. Statistisk Sentralbyrås Håndbøker 33. Oslo, 1974.
- [2] D. Walker: Makrosystemet. Arbeidsnotater fra Statistisk Sentralbyrå IO 74/33. Oslo, 1974.
- [3] O. Bjerkholt, A. Hustveit, P. Sand: MODIS IV Dokumentasjonsnotat nr. 1. Behandling av eksogene variabler og bruk av alternativer. Arbeidsnotater fra Statistisk Sentralbyrå IO 74/30. Oslo, 1974.
- [4] D. Walker: Using DATSY to Implement the Long Term Planning Model MSG-3. Arbeidsnotater fra Statistisk Sentralbyrå. Oslo, 1975. (Planlagt).