

Importrelaterede korrektionsfaktorer til Input-Output systemet

Resumé:

I forbindelse med inklusion og estimation af en importrelation for service og tjenesteydelser ekskl. søfart (fMs) er der i modelversionen fra juli 2013 (Jul13) inkluderet tekniske korrektionsfaktorer i IO-systemet.

jnr

Nøgleord: Input-Output, IO, korrektionsfaktorer

Modelgruppepapirer er interne arbejdsrapporter. De konklusioner, der drages i papirerne, er ikke endelige og kan vrendret inden opstillingen af nye modelversioner. Det henstilles derfor, at der kun citeres fra modelgruppepapirerne efter aftale med Danmarks Statistik.

1. Indledning

Papiret beskriver ændringer i ADAMs IOsystem ved overgangen fra modelversion Okt12 til Jul13. Der er i modelversion Jul13 inkluderet korrektionsfaktorer, som sikrer sammenhæng mellem import og indenlandsk produktion af service- og tjenesteydelser.

En tilsvarende korrektionsfaktor for importen af øvrige råstoffer, $M3q$, har altid været til stede, men er ikke blevet aktiveret før modelversion Jul13. Modposteringen ligger i leverancer fra *ng*-erhvervet. Ligningerne kan findes i bilaget.

2. Inklusion af korrektionsfaktorer

Korrektionsfaktorer vedr. den konkurrerende del af importen inkluderes i ADAMs IO-system, fordi der ligger en antagelse om, at importen erstatter indenlandsk produktion. Når eksempelvis importen af industrivarer (SITC 59 gruppen) øges, antages for uændret efterspørgsel, at den øgede import erstatter leverancerne fra *nz*-erhvervet.

I MVM11913 foreslås at estimere den konkurrerende del af serviceimporten ekskl. søfart, $fMzs$ i stil med relationen for fx industriimporten, $fMz59$. Estimationen af serviceimporten indebærer også, at der skal ske en tilsvarende korrektion i input-output systemet for serviceimporten, Ms .

Det forelås, at der i IO-ligningerne for leverancer fra Ms , inkluderes en k -faktor, $kfmzs$, som også inkluderes, med modsat fortegn i de tilsvarende ligninger for cellerne der beskriver leverance fra servicebranchen, Xqz .

Korrektionsfaktoren beskriver ændringen i forholdet mellem den konkurrerende del af importen og importens markedsandel, jf. nedenstående ligning:

$$kfmzs = (fMzs/fAms)/(fMzs_{-1}/fAms_{-1})$$

$fMzs$ Konkurrerende del af tjenesteimporten
 $fAms$ Indeks for importmarkedet

Når importen stiger ift. markedsandelen, stiger k -faktoren og leverancerne fra den indenlandske produktion erstattes af importen.

IO-Ligningerne foreslås at følge nedenstående opbygning:

$$Ms_i = Ms_{i-1} * kfmzs * \frac{f_i}{f_{i-1}} * \frac{pms}{pms_{-1}}$$

Ms_i IO celle for leverance fra Ms til i
 $kfmzs$ Korrektionsfaktor

f_i Søjlemængde/Forbrug
 pms prisindeks på serviceimport

Modpostering er valgt at komme fra leverancer fra qz-branchen, som serviceimporten må formodes at være konkurrent til, jf. nedenstående ligning¹:

$$Xqz_i = (Xqz_{i-1} + Ms_{i-1} * (1 - kfmzs)) * \frac{f_i}{f_{i-1}} * \frac{pxqz}{pxqz_{-1}}$$

Xqz_i IO celle for leverance fra Xqz til i
 $pxqz$ Sektorprisindeks

Konsekvensen af ovenstående ligninger er, at en udvidelse af tjenesteimporten vil øge leverancen fra Ms , og i tilsvarende omfang reducere leverancen fra Xqz .

Xqz er oplagt at bruge, til at modpostere ændringer i tjenesteimporten, men det kolliderer med systemet af ligninger, som gør det muligt at eksogenisere priserne på endelige anvendelser. Derfor er det nødvendigt, at inkludere ovenstående ændringer i residualberegningerne, der sikrer konsistens ved en sådan eksogenisering.

Residualerne kaldes RZ<pris> og er, jf. papiret jnr25912, tidligere beregnet som:

$$RZp_i = \left(Xqz_i - \left(Xqz_{i-1} * \frac{f_i}{f_{i-1}} * \frac{pxqz}{pxqz_{-1}} \right) \right) / f_i$$

Med inklusion af $kfmzs$ i IO-ligningen for leverancer fra qz-erhvervet, bliver de tilsvarende RZ størrelser nu opskrevet som:

$$RZp_i = \left(Xqz_i - \left((Xqz_{i-1} + Ms_{i-1} * (1 - kfmzs)) * \frac{f_i}{f_{i-1}} * \frac{pxqz}{pxqz_{-1}} \right) \right) / f_i$$

3. Egenskaber

I det følgende aftestes modelegenskaberne ved hhv. en importforøgelse og en eksogenisering af priserne på endelige anvendelser. Nedenstående tabel angiver effekterne på de relevante IO-celler.²

¹ Med udeladte dummyer og J-led.

² Eksperimenterne skrives op som

```
upd dfcs = 1 ;
upd dfcv = 1 ;
upd dfce = 1 ;
upd dpxqz = 1 ;
upd jrfmzs + 0.01 ;
upd jrfmz3q + 0.01 ;
sim ;
```

fCs , fCv samt $pxqz$ fastholdes for at isolere effekten fra $kfmzs$.

Tabel 1. Permanent forøgelse af import, fMz, 1 pct., udvalgte variable, effekter i mio. kr.

År	Ms_cs	Xqz_cs	Ms_cv	Xqz_cv	M3q_ce	Xng_ce
1	75,87	-75,67	6,95	-6,95	27,19	-27,19
2	110,25	-109,46	10,10	-10,10	55,21	-55,21
3	125,24	-124,19	11,47	-11,47	85,72	-85,72
5	138,61	-137,40	12,69	-12,69	154,48	-154,48
10	166,20	-164,87	15,22	-15,24	376,82	-376,82
20	235,57	-233,83	21,57	-21,58	1121,9	-1121,9

Når importen af en vare eller tjeneste stiger, øges leverancerne fra importen på bekostning af faldende leverancer fra den tilsvarende indenlandske produktion.

Da vi også har ændret i RZ-størrelserne, er det på sin plads at afteste om eksogeniseringen af priserne på de endelige anvendelser stadig fungerer som tiltænkt. Nedenstående tabel rapporterer effekterne af forøgelsen af en udvalgt nettopris på udvalgte ADAM-variable.³

Tabel 2. Forøgelse af nettopris vedrørende prisindeks for privatforbrug af øvrige tjenester, pncs, 1 pct., udvalgte variable

År	Pct. af grundforløb			Afvigelse i mio. kr.			
	pncs	Cs	fCs	Xqz_cs	Xqz_il	ll	Xqz_ilz
1	1,00	1,02	0,00	2749	-2722	-2828	-2707
2	1,00	1,01	0,00	2761	91	16	49
3	1,00	1,01	0,00	2814	87	32	63
5	1,00	1,01	0,00	2959	41	-208	49
10	1,00	1,01	0,00	3461	13	-186	13
20	1,00	1,01	0,00	4827	11	-256	10

Effekten af at eksogenisere priserne har ikke ændret sig i forhold til tidligere. Prisstigningen giver anledning til en forøgelse af forbruget i løbende priser. Fastprisstørrelsen er holdt uændret. Forøgelsen giver i dette tilfælde anledning til, at leverancen fra qz-erhvervet til privatforbruget øges med 2749 mio. Dette modposteres umiddelbart i erhvervets lagerinvestering, og de samlede lagerinvesteringer reduceres i tilsvarende omfang.

³ Eksperimentet skrives som

```

upd dpncs = 1 ;
upd zpncs * 1.01;
upd dfcs = 1 ;
sim;
upd dpncs = 0 ;
sim;

```

4. Konklusion

Der er i forbindelse med estimationen af den konkurrerende import af serviceydelser inkluderet en k-faktor i IO-systemet, som sikrer konsistens i leverancerne fra import og produktion af serviceydelser. K-faktoren, som er foreslået i dette papir, vil ved en importforøgelse øge leverancerne fra importen og reducere leverancerne fra den indenlandske produktionen.

Omskrivningen af IO-systemet har samtidig givet anledning til en mindre omskrivning af det eksisterende system af ligninger, som gjorde eksogenisering af priserne på endelige anvendelser mulig. Aftestningen af egenskaberne i dette papir har vist, at inklusionen af k-faktoren ikke har ændret på disse egenskaber.

Bilag

Ligninger med inklusion af *kfmz3q*:

IO-ligningerne for leverancen fra *M3q* bliver:

$$M3q_i = M3q_{i-1} * kfmz3q * \frac{f_i}{f_{i-1}} * \frac{pm3q}{pm3q_{i-1}}$$

Og ligningerne med modpostering er:

$$Xng_i = (Xqng_{i-1} + M3q_{i-1} * (1 - kfmz3q)) * \frac{f_i}{f_{i-1}} * \frac{pxng}{pxng_{i-1}}$$

Omformulerede ligninger til formelfil, g:\jul13\jul13\jul13.frm:

```

FRML _GJD Ms_xa = Ms_xa(-1)*kfmzs*fVma/fVma(-1)*pms/pms(-1) $
FRML _GJD Xqz_xa = ((Xqz_xa(-1)+Ms_xa(-1)*(1-kfmzs))*fVma/fVma(-1)*pxqz/pxqz(-1)+JDpvma*fVma)*(1-Dpvma)
FRML _GJD M3q_xa = M3q_xa(-1)*kfmz3q*fVea/fVea(-1)*pm3q/pm3q(-1) $
FRML _GJD Xng_xa = (Xng_xa(-1)+M3q_xa(-1)*(1-kfmz3q))*fVea/fVea(-1)*pxng/pxng(-1) $
FRML _GJD Ms_xng = Ms_xng(-1)*kfmzs*fVmng/fVmng(-1)*pms/pms(-1) $
FRML _GJD Xqz_xng = (Xqz_xng(-1)+Ms_xng(-1)*(1-kfmzs))*fVmng/fVmng(-1)*pxqz/pxqz(-1)+JDpvmng*fVmng $
FRML _GJD Ms_xne = Ms_xne(-1)*kfmzs*fVmne/fVmne(-1)*pms/pms(-1) $
FRML _GJD Xqz_xne = (Xqz_xne(-1)+Ms_xne(-1)*(1-kfmzs))*fVmne/fVmne(-1)*pxqz/pxqz(-1)+JDpvmne*fVmne $
FRML _GJD M3q_xne = M3q_xne(-1)*kfmz3q*fVene/fVene(-1)*pm3q/pm3q(-1) $
FRML _GJD Xng_xne = (Xng_xne(-1)+M3q_xne(-1)*(1-kfmz3q))*fVene/fVene(-1)*pxng/pxng(-1) $
FRML _GJD Ms_xnf = Ms_xnf(-1)*kfmzs*fVmnf/fVmnf(-1)*pms/pms(-1) $
FRML _GJD Xqz_xnf = ((Xqz_xnf(-1)+Ms_xnf(-1)*(1-kfmzs))*fVmnf/fVmnf(-1)*pxqz/pxqz(-1)+JDpvmnf*fVmnf)*(1-Dpvmnf)
FRML _GJD M3q_xnf = M3q_xnf(-1)*kfmz3q*fVenf/fVenf(-1)*pm3q/pm3q(-1) $
FRML _GJD Xng_xnf = (Xng_xnf(-1)+M3q_xnf(-1)*(1-kfmz3q))*fVenf/fVenf(-1)*pxng/pxng(-1) $
FRML _GJD Ms_xnz = Ms_xnz(-1)*kfmzs*fVmnz/fVmnz(-1)*pms/pms(-1) $
FRML _GJD Xqz_xnz = ((Xqz_xnz(-1)+Ms_xnz(-1)*(1-kfmzs))*fVmnz/fVmnz(-1)*pxqz/pxqz(-1)+JDpvmnz*fVmnz)*(1-Dpvmnz)
FRML _GJD M3q_xnz = M3q_xnz(-1)*kfmz3q*fVenz/fVenz(-1)*pm3q/pm3q(-1) $
FRML _GJD Xng_xnz = (Xng_xnz(-1)+M3q_xnz(-1)*(1-kfmz3q))*fVenz/fVenz(-1)*pxng/pxng(-1) $
FRML _GJD Ms_xb = Ms_xb(-1)*kfmzs*fVmb/fVmb(-1)*pms/pms(-1) $
FRML _GJD Xqz_xb = ((Xqz_xb(-1)+Ms_xb(-1)*(1-kfmzs))*fVmb/fVmb(-1)*pxqz/pxqz(-1)+JDpvmb*fVmb)*(1-Dpvmb)
FRML _GJD M3q_xb = M3q_xb(-1)*kfmz3q*fVeb/fVeb(-1)*pm3q/pm3q(-1) $
FRML _GJD Xng_xb = (Xng_xb(-1)+M3q_xb(-1)*(1-kfmz3q))*fVeb/fVeb(-1)*pxng/pxng(-1) $
FRML _GJD Ms_xqz = Ms_xqz(-1)*kfmzs*fVmzq/fVmzq(-1)*pms/pms(-1) $
FRML _GJD Xqz_xqz = ((Xqz_xqz(-1)+Ms_xqz(-1)*(1-kfmzs))*fVmzq/fVmzq(-1)*pxqz/pxqz(-1)+JDpvmzq*fVmzq)*(1-Dpvmzq)
FRML _GJD M3q_xqz = M3q_xqz(-1)*kfmz3q*fVeqz/fVeqz(-1)*pm3q/pm3q(-1) $
FRML _GJD Xng_xqz = (Xng_xqz(-1)+M3q_xqz(-1)*(1-kfmz3q))*fVeqz/fVeqz(-1)*pxng/pxng(-1) $
FRML _GJD Ms_xqf = Ms_xqf(-1)*kfmzs*fVmzf/fVmzf(-1)*pms/pms(-1) $
FRML _GJD Xqz_xqf = ((Xqz_xqf(-1)+Ms_xqf(-1)*(1-kfmzs))*fVmzf/fVmzf(-1)*pxqz/pxqz(-1)+JDpvmzf*fVmzf)*(1-Dpvmzf)
FRML _GJD M3q_xqf = M3q_xqf(-1)*kfmz3q*fVeqf/fVeqf(-1)*pm3q/pm3q(-1) $
FRML _GJD Xng_xqf = (Xng_xqf(-1)+M3q_xqf(-1)*(1-kfmz3q))*fVeqf/fVeqf(-1)*pxng/pxng(-1) $
FRML _GJD Ms_xh = Ms_xh(-1)*kfmzs*fVmh/fVmh(-1)*pms/pms(-1) $
FRML _GJD Xqz_xh = (Xqz_xh(-1)+Ms_xh(-1)*(1-kfmzs))*fVmh/fVmh(-1)*pxqz/pxqz(-1)+JDpvmh*fVmh $
FRML _GJD M3q_xh = M3q_xh(-1)*kfmz3q*fVeh/fVeh(-1)*pm3q/pm3q(-1) $

```

FRML _GJD Xng_xh = (Xng_xh(-1)+M3q_xh(-1) * (1-kfmz3q)) *fVeh/fVeh(-1) *pxng/pxng(-1) \$
 FRML _GJD M3q_cf = M3q_cf(-1) *kfmz3q*fCf/fCf(-1) *pm3q/pm3q(-1) \$
 FRML _GJD Ms_cf = Ms_cf(-1) *kfmzs*fCf/fCf(-1) *pms/pms(-1) \$
 FRML _GJD Xng_cf = (Xng_cf(-1)+M3q_cf(-1) * (1-kfmz3q)) *fCf/fCf(-1) *pxng/pxng(-1) \$
 FRML _GJD M3q_cv = M3q_cv(-1) *kfmz3q*fCv/fCv(-1) *pm3q/pm3q(-1) \$
 FRML _GJD Ms_cv = Ms_cv(-1) *kfmzs*fCv/fCv(-1) *pms/pms(-1) \$
 FRML _GJD Xng_cv = (Xng_cv(-1)+M3q_cv(-1) * (1-kfmz3q)) *fCv/fCv(-1) *pxng/pxng(-1) \$
 FRML _GJD M3q_ce = M3q_ce(-1) *kfmz3q*fCe/fCe(-1) *pm3q/pm3q(-1) \$
 FRML _GJD Ms_ce = Ms_ce(-1) *kfmzs*fCe/fCe(-1) *pms/pms(-1) \$
 FRML _GJD Xng_ce = (Xng_ce(-1)+M3q_ce(-1) * (1-kfmz3q)) *fCe/fCe(-1) *pxng/pxng(-1) \$
 FRML _GJD M3q_cg = M3q_cg(-1) *kfmz3q*fCg/fCg(-1) *pm3q/pm3q(-1) \$
 FRML _GJD Ms_cg = Ms_cg(-1) *kfmzs*fCg/fCg(-1) *pms/pms(-1) \$
 FRML _GJD Xng_cg = (Xng_cg(-1)+M3q_cg(-1) * (1-kfmz3q)) *fCg/fCg(-1) *pxng/pxng(-1) \$
 FRML _GJD M3q_cb = M3q_cb(-1) *kfmz3q*fCb/fCb(-1) *pm3q/pm3q(-1) \$
 FRML _GJD Ms_cb = Ms_cb(-1) *kfmzs*fCb/fCb(-1) *pms/pms(-1) \$
 FRML _GJD Xng_cb = (Xng_cb(-1)+M3q_cb(-1) * (1-kfmz3q)) *fCb/fCb(-1) *pxng/pxng(-1) \$
 FRML _GJD M3q_ch = M3q_ch(-1) *kfmz3q*fCh/fCh(-1) *pm3q/pm3q(-1) \$
 FRML _GJD Ms_ch = Ms_ch(-1) *kfmzs*fCh/fCh(-1) *pms/pms(-1) \$
 FRML _GJD Xng_ch = (Xng_ch(-1)+M3q_ch(-1) * (1-kfmz3q)) *fCh/fCh(-1) *pxng/pxng(-1) \$
 FRML _GJD M3q_cs = M3q_cs(-1) *kfmz3q*fCs/fCs(-1) *pm3q/pm3q(-1) \$
 FRML _GJD Ms_cs = Ms_cs(-1) *kfmzs*fCs/fCs(-1) *pms/pms(-1) \$
 FRML _GJD Xng_cs = (Xng_cs(-1)+M3q_cs(-1) * (1-kfmz3q)) *fCs/fCs(-1) *pxng/pxng(-1) \$
 FRML _GJD M3q_ct = M3q_ct(-1) *kfmz3q*fCt/fCt(-1) *pm3q/pm3q(-1) \$
 FRML _GJD Ms_ct = Ms_ct(-1) *kfmzs*fCt/fCt(-1) *pms/pms(-1) \$
 FRML _GJD M3q_co = M3q_co(-1) *kfmz3q*fCo/fCo(-1) *pm3q/pm3q(-1) \$
 FRML _GJD Ms_co = Ms_co(-1) *kfmzs*fCo/fCo(-1) *pms/pms(-1) \$
 FRML _GJD M3q_im = M3q_im(-1) *kfmz3q*fIm/fIm(-1) *pm3q/pm3q(-1) \$
 FRML _GJD Ms_im = Ms_im(-1) *kfmzs*fIm/fIm(-1) *pms/pms(-1) \$
 FRML _GJD Xng_im = (Xng_im(-1)+M3q_im(-1) * (1-kfmz3q)) *fIm/fIm(-1) *pxng/pxng(-1) \$
 FRML _GJD Xqz_cf = ((Xqz_cf(-1)+Ms_cf(-1) * (1-kfmzs)) *fCf/fCf(-1) *pxqz/pxqz(-1) +JDpncf*fCf) * (1-Dpncf) +Dpncf * (zpnfc*fCf-(Xa_cf+Xe_cf+Xng_cf+Xne_cf+Xnf_cf+Xnz_cf+Xb_cf+Xqs_cf+Xqf_cf+Xh_cf+Xo_cf+M01_cf+M2_cf+M3r_cf+M3k_cf+M3q_cf+M59_cf+M7b_cf+M7y_cf+Ms_cf+Spm_cf)) \$
 FRML _GJD Xqz_cv = ((Xqz_cv(-1)+Ms_cv(-1) * (1-kfmzs)) *fCv/fCv(-1) *pxqz/pxqz(-1) +JDpncv*fCv) * (1-Dpncv) +Dpncv * (zpnvc*fCv-(Xa_cv+Xe_cv+Xng_cv+Xne_cv+Xnf_cv+Xnz_cv+Xb_cv+Xqs_cv+Xqf_cv+Xh_cv+Xo_cv+M01_cv+M2_cv+M3r_cv+M3k_cv+M3q_cv+M59_cv+M7b_cv+M7y_cv+Ms_cv+Spm_cv)) \$
 FRML _GJD Xqz_ce = ((Xqz_ce(-1)+Ms_ce(-1) * (1-kfmzs)) *fCe/fCe(-1) *pxqz/pxqz(-1) +JDpnce*fCe) * (1-Dpnce) +Dpnce * (zpnce*fCe-(Xa_ce+Xe_ce+Xng_ce+Xne_ce+Xnf_ce+Xnz_ce+Xb_ce+Xqs_ce+Xqf_ce+Xh_ce+Xo_ce+M01_ce+M2_ce+M3r_ce+M3k_ce+M3q_ce+M59_ce+M7b_ce+M7y_ce+Ms_ce+Spm_ce)) \$
 FRML _GJD Xqz_cg = ((Xqz_cg(-1)+Ms_cg(-1) * (1-kfmzs)) *fCg/fCg(-1) *pxqz/pxqz(-1) +JDpncg*fCg) * (1-Dpncg) +Dpncg * (zpncc*fCg-(Xa_cg+Xe_cg+Xng_cg+Xne_cg+Xnf_cg+Xnz_cg+Xb_cg+Xqs_cg+Xqf_cg+Xh_cg+Xo_cg+M01_cg+M2_cg+M3r_cg+M3k_cg+M3q_cg+M59_cg+M7b_cg+M7y_cg+Ms_cg+Spm_cg)) \$
 FRML _GJD Xqz_cb = ((Xqz_cb(-1)+Ms_cb(-1) * (1-kfmzs)) *fCb/fCb(-1) *pxqz/pxqz(-1) +JDpncb*fCb) * (1-Dpncb) +Dpncb * (zpnbc*fCb-(Xa_cb+Xe_cb+Xng_cb+Xne_cb+Xnf_cb+Xnz_cb+Xb_cb+Xqs_cb+Xqf_cb+Xh_cb+Xo_cb+M01_cb+M2_cb+M3r_cb+M3k_cb+M3q_cb+M59_cb+M7b_cb+M7y_cb+Ms_cb+Spm_cb)) \$
 FRML _GJD Xqz_ch = (Xqz_ch(-1)+Ms_ch(-1) * (1-kfmzs)) *fCh/fCh(-1) *pxqz/pxqz(-1) +JDpncch*fCh \$
 FRML _GJD Xqz_cs = ((Xqz_cs(-1)+Ms_cs(-1) * (1-kfmzs)) *fCs/fCs(-1) *pxqz/pxqz(-1) - (Xo_cs-Xo_cs(-1) *fCs/fCs(-1) *pxo/pxo(-1)) +JDpncc*fCs) * (1-Dpncc) +Dpncc * (zpncc*fCs-(Xa_cs+Xe_cs+Xng_cs+Xne_cs+Xnf_cs+Xnz_cs+Xb_cs+Xqs_cs+Xqf_cs+Xh_cs+Xo_cs+M01_cs+M2_cs+M3r_cs+M3k_cs+M3q_cs+M59_cs+M7b_cs+M7y_cs+Ms_cs+Spm_cs)) \$
 FRML _GJD Xqz_ct = (Xqz_ct(-1)+Ms_ct(-1) * (1-kfmzs)) *fCt/fCt(-1) *pxqz/pxqz(-1) +JDpncct*fCt \$
 FRML _GJD Xqz_im = ((Xqz_im(-1)+Ms_im(-1) * (1-kfmzs)) *fIm/fIm(-1) *pxqz/pxqz(-1) +JDpncim*fIm) * (1-Dpncim) +Dpncim * (zpnim*fIm-(Xa_im+Xe_im+Xng_im+Xne_im+Xnf_im+Xnz_im+Xb_im+Xqs_im+Xqf_im+Xh_im+Xo_im+M01_im+M2_im+M3r_im+M3k_im

```

+M3q_im+M59_im+M7b_im+M7y_im+Ms_im+Spm_im)) $
FRML _GJD Xqz_co = (Xqz_co(-1)+Ms_co(-1)*(1-kfmzs))*fCo/fCo(-1)*pxqz/pxqz(-
1)+JDpco*fCo $
FRML _GJD M3q_ib = M3q_ib(-1)*kfmz3q*fIb/fIb(-1)*pm3q/pm3q(-1) $
FRML _GJD Ms_ib = Ms_ib(-1)*kfmzs*fIb/fIb(-1)*pms/pms(-1) $
FRML _GJD Xng_ib = (Xng_ib(-1)+M3q_ib(-1)*(1-kfmz3q))*fIb/fIb(-1)*pxng/pxng(-
1) $
FRML _GJD Xqz_ib = ((Xqz_ib(-1)+Ms_ib(-1)*(1-kfmzs))*fIb/fIb(-1)*pxqz/pxqz(-
1)+JDpnib*fIb)*(1-Dpnib)
+Dpnib*(zpnib*fIb-(Xa_ib+Xe_ib+Xng_ib+Xne_ib+Xnf_ib+Xnz_ib
+Xb_ib+Xqs_ib+Xqf_ib+Xh_ib+Xo_ib+M01_ib+M2_ib+M3r_ib+M3k_ib
+M3q_ib+M59_ib+M7b_ib+M7y_ib+Ms_ib+Spm_ib)) $
FRML _G RZpvma = (Xqz_xa -((Xqz_xa(-1)+Ms_xa(-1)*(1-kfmzs))*fVma /fVma(-1)
*pxqz/pxqz(-1)))/fVma $
FRML _G RZpvmnf = (Xqz_xnf-((Xqz_xnf(-1)+Ms_xnf(-1)*(1-kfmzs))*fVmnf/fVmnf(-
1)*pxqz/pxqz(-1)))/fVmnf $
FRML _G RZpvmnz = (Xqz_xnz-((Xqz_xnz(-1)+Ms_xnz(-1)*(1-kfmzs))*fVmnz/fVmnz(-
1)*pxqz/pxqz(-1)))/fVmnz $
FRML _G RZpvmqf = (Xqz_xqf-((Xqz_xqf(-1)+Ms_xqf(-1)*(1-kfmzs))*fVmqf/fVmqf(-
1)*pxqz/pxqz(-1)))/fVmqf $
FRML _G RZpvmb = (Xqz_xb -((Xqz_xb(-1) +Ms_xb(-1)*(1-kfmzs))*fVmb /fVmb(-1)
*pxqz/pxqz(-1)))/fVmb $
FRML _G RZpvmqz = (Xqz_xqz-((Xqz_xqz(-1)+Ms_xqz(-1)*(1-kfmzs))*fVmqz/fVmqz(-
1)*pxqz/pxqz(-1)))/fVmqz $
FRML _G RZpvme = (Xqz_xe -((Xqz_xe(-1)*fVme /fVme(-1) *pxqz/pxqz(-1)))/fVme $
FRML _G RZpvmo = (Xqz_xo -((Xqz_xo(-1)*fVmo /fVmo(-1) *pxqz/pxqz(-1)))/fVmo $
FRML _G RZpnim = (Xqz_im -((Xqz_im(-1) +Ms_im(-1)*(1-kfmzs))*fIm /fIm(-1)
*pxqz/pxqz(-1)))/fIm $
FRML _G RZpnib = (Xqz_ib -((Xqz_ib(-1) +Ms_ib(-1)*(1-kfmzs))*fIb /fIb(-1)
*pxqz/pxqz(-1)))/fIb $
FRML _G RZpncf = (Xqz_cf -((Xqz_cf(-1) +Ms_cf(-1)*(1-kfmzs))*fCf /fCf(-1)
*pxqz/pxqz(-1)))/fCf $
FRML _G RZpncv = (Xqz_cv -((Xqz_cv(-1) +Ms_cv(-1)*(1-kfmzs))*fCv /fCv(-1)
*pxqz/pxqz(-1)))/fCv $
FRML _G RZpnce = (Xqz_ce -((Xqz_ce(-1) +Ms_ce(-1)*(1-kfmzs))*fCe /fCe(-1)
*pxqz/pxqz(-1)))/fCe $
FRML _G RZpncg = (Xqz_cg -((Xqz_cg(-1) +Ms_cg(-1)*(1-kfmzs))*fCg /fCg(-1)
*pxqz/pxqz(-1)))/fCg $
FRML _G RZpnCb = (Xqz_cb -((Xqz_cb(-1) +Ms_cb(-1)*(1-kfmzs))*fCb /fCb(-1)
*pxqz/pxqz(-1)))/fCb $
FRML _G RZpnCs = ((Xqz_cs -((Xqz_cs(-1) +Ms_cs(-1)*(1-kfmzs))*fCs /fCs(-1)
*pxqz/pxqz(-1)
+(Xo_cs -Xo_cs(-1) *fCs /fCs(-1) *pxo /pxo(-1)
))/fCs) $
FRML _D kfmzs = (fMzs /fAms )/(fMzs(-1)/fAms(-1)) $

```