資料篇:2 補章(植村)

資料篇：２

補章（植村）

資料篇２－１．各国が市場国で直面する「競争者価格」作成プログラム（EViews）

本来 EViews のプログラムは行番号に依存しないが、ここでは解説のために行番号をつけてある。行番号のない行がコメント行である。また、行番号がある行でも、最初に出てくる（空白以外の）記号が「’」である行はプログラム内に書かれたコメント行である。これらの「コメント」を取り外せば（行番号も不要）、そのまま作動するプログラムとなっている。（資料篇２－２、３、４も同様）

```
00010  ’ ****************************************
00020  ’ ************ 競争者価格作成 *************
00030  ’ ****************************************
00040
00050  include MySub_CntryLabel

       年数を入れたベクトルを用意しておく必要がある。
00060
00070  ’ 用意しておく必要あり
00080  ’ vector(!rrange) <-  _vec_year

       データ格納および作業用パスの設定
00090  ’ IDE
00100  %parentpath = "R:¥Personal¥201312 Com3Link¥
00110  %workpatheviews = "R:¥Personal¥201312 Com3Link¥EViews¥
00120
00130  %workpath = %parentpath ’ ---------------------------- init path
00140
00150  %workdate = "20131229
00160
00170  workfile {%workdate}_00_com3link a 1970 2009
                ’ -------------------------- parent workfile
00180
00190        if @isobject("range_total") then
00200
00210        endif
00220
00230
```

```
00240  ' ------------ number of countries ------------
00250  !noofcntry = 15
00260  call Cntrylabel( '!noofcntry )
00270
00280  '-----------------------------------
00290  !startcntry =         1
00300  !endcntry =           !noofcntry
00310
00320  '--- sample period ------------
00330  '--- range ---
00340  !begrange = 1970
00350  !endrange = 2009
00360  !rrange = !endrange - !begrange + 1
00370          smpl !begrange !endrange
00380          sample range_total !begrange !endrange
00390  '-----------------------------------------------
00400
00410  workfile {%workdate}_00_com3link a range_total
                 ' ------------------------- parent workfile
00420  ' --- information vector/matrix ---
00430  matrix(!noofcntry,3)  _mat_exr2005
00440  matrix(!noofcntry,!noofcntry)  _mat_importshare1
00450  matrix(!noofcntry,!noofcntry)  _mat_importshare3
00460
```

財別輸入シェア行列を読み込む。もとの行列を転置したものを輸入シェア行列とする。

```
00470  ' --- read import share matrix ---
00480  %subpath = "Data_Basic¥"
00490  %workpath = %parentpath + %subpath
00500  %workfile = %workpath + "___Share_Import1.txt"
00510  _mat_importshare1.read(t=txt) %workfile
00520  _mat_importshare1 = @transpose(_mat_importshare1)
00530  %workfile = %workpath + "___Share_Import3.txt"
00540  _mat_importshare3.read(t=txt) %workfile
```

```
00550   _mat_importshare3 = @transpose(_mat_importshare3)
00560   %workpath = %parentpath ' ---------------------------- init path
00570
```

リンク参加国からの輸入合計を分母とする。

```
00580   ' --- import share total (from link partners) ---
00590   vector _vec_impsharetotal1 = @csum(_mat_importshare1)
00600   vector _vec_impsharetotal3 = @csum(_mat_importshare3)
00610
00620   workfile {%workdate}_00_com3link a range_total
                ' ------------------------- parent workfile
00630
```

ファイルとして格納してある各国の財別輸出価格を読み出すブロック。

```
00640   ' --- 価格データ読み込み　(各国輸出価格)　---
00650           matrix(!rrange,2)  _mat_z
00660           vector(!rrange)    _vec_z
00670
00680           for !zi = !startcntry to !endcntry
00690                   %rcntlist = _list_cntry(!zi)
00700                   %rcntlabel = _label_cntry(!zi)
00710                   %subpath = "Data_PX_Original¥"
00720                   %workpath = %parentpath + %subpath
00730
00740                   matrix(!rrange,!noofcntry)  _mat_px1
00750                   matrix(!rrange,!noofcntry)  _mat_px3
00760
00770                   '    ---------- 輸出価格読み込み ----------
00780                   if @isobject(  %rcntlabel+"px1"  ) then
00790                           delete {%rcntlabel}px1
00800                   endif
00810                   if @isobject(  %rcntlabel+"px3"  ) then
00820                           delete {%rcntlabel}px3
00830                   endif
00840
```

```
00850                 %workfile = %workpath +  %rcntlist+ "_px1_original.txt"
00860                 _mat_z.read(t=txt) %workfile
00870                 _vec_z = @columnextract(  _mat_z,  2  )
00880                 mtos(  _vec_z  , {%rcntlabel}px1  )
00890
```

欠損値には０を割り当てておく。
関数 colplace(・) にて、各国輸出価格の入った行列を作成する。

```
00900                 series _ser_z = @isna({%rcntlabel}px1)-1  ' NA -> 0
00910                 _ser_z = @nan( {%rcntlabel}px1, _ser_z )  'overwrite
00920                 stomna(  _ser_z,  _vec_z  )
00930                 colplace(  _mat_px1,   _vec_z,  !zi  )
00940
00950                 %workfile = %workpath +  %rcntlist+ "_px3_original.txt"
00960                 _mat_z.read(t=txt) %workfile
00970                 _vec_z = @columnextract(_mat_z,2)
00980                 mtos(  _vec_z  , {%rcntlabel}px3  )
00990
01000                 series _ser_z = @isna({%rcntlabel}px3)-1  ' NA -> 0
01010                 _ser_z = @nan( {%rcntlabel}px3, _ser_z )  'overwrite
01020                 stomna(  _ser_z,  _vec_z  )
01030                 colplace(  _mat_px3,   _vec_z,  !zi  )
01040
01050         next ' !zi
01060         delete _mat_z
01070         delete _vec_z
01080
```

競争者価格算出ブロック。リンク参加国全体に対する輸入シェアによる
ウェイト付けを行う。

```
01090         '    ---------- 競争者価格算出 ----------
01100         %subpath = "Data_PC_Original¥"
01110         %workpath = %parentpath + %subpath
01120
```

```
01130        matrix(!rrange, 2) _mat_z
01140        for !zi = !startcntry to !endcntry
01150               %rcntlist = _list_cntry(!zi)
01160               %rcntlabel = _label_cntry(!zi)
01170
01180               vector(!noofcntry)    _vec_z_share11
01190               vector(!noofcntry)    _vec_z_share12
01200               vector(!noofcntry)    _vec_z_share31
01210               vector(!noofcntry)    _vec_z_share32
01220               vector(!noofcntry)    _vec_z_1
01230               vector(!noofcntry)    _vec_z_3
01240
01250               matrix(!rrange,!noofcntry) _mat_{%rcntlabel}pc1
01260               matrix(!rrange,!noofcntry) _mat_{%rcntlabel}pc3
01270
01280               _vec_z_share11 = @columnextract(_mat_importshare1,!zi)
                           ’自国の輸入シェア
01290               _vec_z_share31 = @columnextract(_mat_importshare1,!zi)
                           ’自国の輸入シェア
01300
01310               for !zj = !startcntry to !endcntry
01320                              %pcntlabel = _label_cntry(!zj)
01330
01340                              _vec_z_share12  = _vec_z_share11
01350                              _vec_z_share32  = _vec_z_share31
01360
```

自国と市場国からの輸入シェアを０とする。もとのベクトルの当該位置に０を設定する関数を利用。

```
01370                              _vec_z_share12.fill(o=!zj) 0
01380                              _vec_z_share32.fill(o=!zj) 0
01390
```

自国と市場国のウェイトを０としたベクトルの要素和を分母として加重平均する。

```
01400                   vector(!rrange) _vec_z_1  = _mat_px1 * _vec_z_share12
                                                      / @sum(_vec_z_share12)
01410                   vector(!rrange) _vec_z_3  = _mat_px3 * _vec_z_share32
                                                     / @sum(_vec_z_share32)
01420
```

ユーロ圏（eur）以外について、自国について計算されているベクトルには強制的に０ばかりの列を代入する。

作成された変数の性格は以下の通り。
（１）「報告国（輸出者としての）が」
（２）「相手国市場の第ｎ財市場で」
直面する、競争相手（他の国々）の加重オファー価格。

ラベルのつけ方の規則は

（報告国）_pc_n_（相手国）

であり、書き出すファイル名（1430 行、1520 行）もその形式で書かれている。これは、報告国が相手国で直面する競争者の価格、という意味合いである。ただし、これらデータは実際には報告国モデルで使われることはなく、相手国市場での競争を通じた相手国の輸入関数で用いられる。

（第１財）

```
01430  %workfile = %workpath + %rcntlist+ "_pc1_"+%pcntlabel + "_original.txt"
01440          if ( !zj = !zi ) * ( !zi <> 15 ) then
01450                  _vec_z_1 = 0
01460          endif
01470           colplace(    _mat_z,    _vec_year,   1  )
01480           colplace(    _mat_z,    _vec_z_1,   2  )
01490           _mat_z.write(t=txt) %workfile
01500           mtos(  _vec_z_1  , {%rcntlabel}pc1{%pcntlabel}  )
01510
```

（第３財）

```
01520  %workfile = %workpath + %rcntlist+ "_pc3_"+%pcntlabel + "_original.txt"
```

```
01530          if ( !zj = !zi ) * ( !zi <> 15 ) then
01540                 _vec_z_3 = 0
01550          endif
01560          colplace(    _mat_z,    _vec_year,   1  )
01570          colplace(    _mat_z,    _vec_z_3,   2  )
01580          _mat_z.write(t=txt) %workfile
01590          mtos(  _vec_z_3  , {%rcntlabel}pc3{%pcntlabel}  )
01600
01610          colplace(    _mat_{%rcntlabel}pc1,    _vec_z_1,   !zj  )
01620          colplace(    _mat_{%rcntlabel}pc3,    _vec_z_3,   !zj  )
01630
01660                 next ' !zj
01670          next ' !zi
01680          %workpath = %parentpath ' ---------------------------- init path
```

一時変数群の消去。

```
01700          delete    _mat_z
01710          delete    _vec_z_share11
01720          delete    _vec_z_share12
01730          delete    _vec_z_share31
01740          delete    _vec_z_share32
01750          delete    _vec_z_1
01760          delete    _vec_z_3
01770          delete    _ser_z
01780  '  === End of File ===
```

（以上）

資料篇２－２．リンク参加国の輸出価格を反映した輸入価格の作成（EViews）

このスクリプトは多くの部分が「競争者価格」作成のものと重複しているため詳しい説明は省略する。スクリプト内のコメント文でほぼ理解できよう。

```
00010  ’*****************************************
00020  ’******* リンク参加国からの輸入価格作成 ********
00030  ’*****************************************
00040  ’  輸入シェア行列×各国輸出価格により作成
00050  ’  PM_LINK フォルダにテキストファイルを出力
00060  ’
00070  include MySub_CntryLabel
00080
00090  ’ vector(!rrange) <-  _vec_year
00100  ’ 用意しておく必要あり
00110
00120  ’ IDE
00130  %parentpath = ″R:¥Personal¥201312 Com3Link¥
00140
00150  include MySub_CntryLabel
00160
00170  %workpath = %parentpath ’ ---------------------------- init path
00180
00190  %workdate = ″20131229
00200  workfile {%workdate}_00_com3link a 1970 2009
                    ’ ------------------------- parent workfile
00210
00220          if @isobject(″range_total″) then
00230                    delete range_total
00240          endif
00250
00260  ’ ------------ number of countries ------------
00270  !noofcntry = 15
00280  call Cntrylabel( !noofcntry )
00290
```

```
00300  '--------- sample period ----------------------------
00310  '--- range ---
00320  !begrange = 1970
00330  !endrange = 2009
00340  !rrange = !endrange - !begrange + 1
00350          smpl !begrange !endrange
00360          sample range_total !begrange !endrange
00370  '----------------------------------------------
00380
00390  workfile {%workdate}_00_com3link a range_tota
                          ' ------------------------- parent workfile
00400  ' --- information vector/matrix ---
00410  matrix(!noofcntry,!noofcntry)  _mat_z
00420  vector(!rrange)   _vec_z_sumweight1
00430  vector(!rrange)   _vec_z_sumweight3
00440
00450
00460  ' --- read import share matrix ---
00470  %subpath = "Data_Basic¥
00480  %workpath = %parentpath + %subpath
00490  %workfile = %workpath + "___Share_Import1.txt
00500  _mat_z.read(t=txt) %workfile
00510  _mat_z = @transpose(_mat_z)
00520  ' --- import share total (from link partners) ---
00530  _vec_z_sumweight1 = @csum(_mat_z)
00540
00550  %workfile = %workpath + "___Share_Import3.txt
00560  _mat_z.read(t=txt) %workfile
00570  _mat_z = @transpose(_mat_z)
00580  ' --- import share total (from link partners) ---
00590  _vec_z_sumweight3 = @csum(_mat_z)
00600
00610  delete _mat_z
00620  %workpath = %parentpath ' ---------------------------- init path
00630
00640
```

```
00650
00660  ' --- 価格データ読み込み （各国輸出価格） ---
00670  matrix(!rrange,2)  _mat_z
00680  vector(!rrange)        _vec_zpx
00690  vector(!rrange)        _vec_zpx1
00700  vector(!rrange)        _vec_zpx3
00710  vector(!noofcntry)    _vec_zshare
00720
00730  for !zi = 1  to !noofcntry
00740          %rcntlist = _list_cntry(!zi)
00750          %rcntlabel = _label_cntry(!zi)
00760          %subpath = "Data_PX_Original¥"
00770          %workpath = %parentpath + %subpath
00780
00790          matrix(!rrange,!noofcntry)  _mat_px1
00800          matrix(!rrange,!noofcntry)  _mat_px3
00810
00820          '    ---------- 輸出価格読み込み ----------
00830          if @isobject(  %rcntlabel+"px1"  ) then
00840             delete {%rcntlabel}px1
00850          endif
00860          if @isobject(  %rcntlabel+"px3"  ) then
00870             delete {%rcntlabel}px3
00880          endif
00890          if @isobject(  %rcntlabel+"pm1wd_lnk"  ) then
00900             delete {%rcntlabel}pm1wd_lnk
00910          endif
00920          if @isobject(  %rcntlabel+"pm3wd_lnk"  ) then
00930             delete {%rcntlabel}pm3wd_lnk
00940          endif
00950
```

ファイルより各国輸出価格を読み出し、親 WF 内のデータシリーズとして格納しておく。

（第１財）

```
00960        %workfile = %workpath +  %rcntlist+ "_px1_original.txt"
00970        _mat_z.read(t=txt) %workfile
00980        _vec_zpx = @columnextract(  _mat_z,  2  )
00990        mtos(  _vec_zpx  ,  _ser_z1  )
01000
```

関数 colplace(・) にて、各国輸出価格の入った行列を作成する。

```
01010        series _ser_z2 = @isna( _ser_z1 )-1  ' NA -> 0
01020        _ser_z2 = @nan( _ser_z1, _ser_z2 )  'overwrite
01030        stomna(  _ser_z2,  _vec_z  )
01040        colplace(  _mat_px1,   _vec_zpx,  !zi  )
01050
01060        _vec_zshare = @columnextract(_mat_importshare1,!zi)
                    ' 自国の輸入シェア
01070        _vec_zpx1 = ( _mat_px1 * _vec_zshare ) / _vec_z_sumweight1(!zi)
01080     ' mtos(  _vec_zpx1  ,  {%rcntlabel}pm1wd_lnk  )
01090 '
```

（第３財）

```
01100        %workfile = %workpath +  %rcntlist+ "_px3_original.txt"
01110        _mat_z.read(t=txt) %workfile
01120        _vec_zpx = @columnextract(  _mat_z,  2  )
01130        mtos(  _vec_zpx  ,  _ser_z1  )
01140
01150        series _ser_z2 = @isna(_ser_z1)-1  ' NA -> 0
01160        _ser_z2 = @nan( _ser_z1, _ser_z2 )  'overwrite
01170        stomna(  _ser_z2,  _vec_z  )
01180        colplace(  _mat_px3,   _vec_zpx,  !zi  )
01190
01200        _vec_zshare = @columnextract(_mat_importshare3,!zi)
                    ' 自国の輸入シェア
```

```
01210          _vec_zpx3 = ( _mat_px3 * _vec_zshare ) / _vec_z_sumweight3(!zi)
01220        ’ mtos(  _vec_zpx3  ,  {%rcntlabel}pm3wd_lnk  )
01230
```

ファイルへの書き出し。「リンク参加国（lnk）を一時的に『対世界（w)』と見た米ドル建て（d）輸入価格」

```
01240          ’ 書き出し
01250          %subpath = ″Data_PM_Lnk_Original¥″
01260          %workpath = %parentpath + %subpath
01270
01280          colplace(    _mat_z,    _vec_year,   1  )
01290          colplace(    _mat_z,    _vec_zpx1,   2  )
01300          %workfile = %workpath +  %rcntlist+ ″_pm1wd_lnk_original.txt″
01310          _mat_z.write(t=txt) %workfile
01320
01330          colplace(    _mat_z,    _vec_zpx3,   2  )
01340          %workfile = %workpath +  %rcntlist+ ″_pm3wd_lnk_original.txt″
01350          _mat_z.write(t=txt) %workfile
01360
01370  next ’ !zi
01380
```

一時変数の消去。

```
01390  delete _mat_z
01400  delete _vec_zpx
01410  delete _vec_zpx1
01420  delete _vec_zpx3
01430
01440  delete _vec_zshare
01450  delete _vec_z_sumweight1
01460  delete _vec_z_sumweight3
01470
01480  delete _ser_z1
01490  delete _ser_z2
01500  ’  === End of File ===
```

（以上）

資料篇２－３．輸入関数の推定・選別（EViews）

```
00010  ’**************************************************
00020  ’***********      輸入関数の推定自動化      ***********
00030  ’**************************************************
00040
```

国ラベルを作成する部分をサブルーチン化。国の数を引数として渡し、
国ラベル、国リストを受け取る（130 行）。

```
00050
00060  include MySub_CntryLabel
00070
00080  ’ IDE
00090  %datapath = "R:¥Personal¥201312 Com3Link¥Data_Basic¥"
00100  %workpath = "R:¥Personal¥201312 Com3Link¥EViews¥"
00110  %outputpath = "R:¥Personal¥201312 Com3Link¥Outputs¥"
00120
00130  call Cntrylabel( 15 )
00140
00150  %workdate = "20131229"
00160
00170  workfile {%workdate}_00_com3link a 1970 2009
              ’------------------------- parent workfile
00180
00190  ’*** 対象国の設定（1-15）***
00200
```

報告国の設定。いずれかの国を単体で行う場合は 220 行と 3480 行をともに
コメントアウト、240 行で国番号を指定する。
一方、連続して行う場合には 240 行をコメントアウトし、220 行と 3480 行
にてループを形成する。

```
00210
00220  ’  for !reportcntry = 1 to 15
00230
00240  !reportcntry = 1
00250
00260  workfile {%workdate}_00_com3link a range_total
              ’------------------------- parent workfile
```

```
00270  ' --- setting cntry mdl ---
00280  %rcntlist = _list_cntry(!reportcntry)
00290  %rcntlabel = _label_cntry(!reportcntry)
00300
00310  %zworkfile = %workdate + %rcntlist ' cntry workfile name
00320
```

メモリー節約のため、各国 WF は使うたびに開き、作業が終わったら閉じる。
なお、wfuse コマンドはファイルが既に開いている場合はそのまま用い、
閉じている場合には開く。(wfopen では既に開いている WF に適用すると
エラーとなる)

```
00330          wfuse %zworkfile
00340
00350          workfile %zworkfile a range_total '----------------------- cntry
model
00360          smpl  1970 2009
00370          call Cntrylabel( 15 ,  %datapath )
00380
00390
```

推定された関数を判定するための情報格納用変数群の宣言。
変数名 t_*** は回帰係数の t 値、h_*** は自己回帰変数がある場合の
h 統計量に関する情報、dw_stat は Durbin-Watson 統計量である。
output_detector は、その定式化を採用するか否かの標識である。

```
00400
00410  scalar   t_gdp
00420  scalar   t_p1
00430  scalar   t_p2
00440  scalar   t_lag
00450  scalar   h_dev
00460  scalar   h_stat
00470  scalar   dw_stat
00480  scalar   output_detector
00490
```

ここでは、各回帰係数の統計的有意性を判定するための t 値の
限界値を設定している (critical value=0.5)。この値を大きくすれば
判定の条件は厳しいものとなり、小さくすれば緩やかなものとなる。

```
00500
00510  ’ 定数の設定
00520  !t_critical = 0.5
00530  !maxnoofspecs  = 1200
00540
```

ベクトル _svec_time は各国の一連の計算にかかる時間を測るための
ものである。

```
00550
00560  vector(4) tstats
00570  svector(2) _svec_time
00580
00590  svector(2) com_sitc
00600  com_sitc(1) = ″1″
00610  com_sitc(2) = ″3″
00620
00630  _svec_time(1) = @date + ″ + @time
00640
```

660 行から「相手国」のループに入る。国番号を 1～15 の順に変化させて
いくが、ユーロ圏を除き自国からの輸入はないため、680 行～700 行で
自国をスキップしている。スキップの方法として「次の国ラベル」に
強制的に移動させるため、ユーロ圏が「最後の国ラベル」である必要がある。

```
00650
00660  for !pcntry =  1    to  15
       ’  ------------------------------------- partner country
00670  ’ ****************************    自国をスキップ（EUR を除く）
00680          if (!pcntry = !reportcntry) * (!reportcntry <> 15) then
00690                  !pcntry = !pcntry + 1
00700          endif
00710
00720      %pcnt = _label_cntry(!pcntry)
00730          %pcntlist = _list_cntry(!pcntry)
00740
00750  for !commodity = 1 to  2
         ’  ------------------------------------- commodity
00760          %com = com_sitc(!commodity)
```

```
00770
00780         %tmp = "_" + %pcntlist + "_m" + %com + "_func"
00790         if @isobject(%tmp) then
00800                   delete %tmp
00810         endif
00820         %tmp = "_" + %pcntlist + "_m" + %com + "_tstats"
00830         if @isobject(%tmp) then
00840                   delete %tmp
00850         endif
00860
00870         svector(!maxnoofspecs) _{%pcntlist}_m{%com}_func
00880         matrix(!maxnoofspecs, 7) _{%pcntlist}_m{%com}_tstats
00890
00900 !counter=0
00910 !OKcounter = 0
00920
```

950 行で設定している「ダミーベクトル」は

_00_now_「相手国」「財番号」

という名前の空ベクトルである。このベクトルは各国 WF 内で先頭に作成され（他の変数名との兼ね合いで 場所は変わりうるが、先頭になるような名前をつけておく)、WF を表示しておくことによって現在の計算作業がどのあたりまで進んでいるのかを確認することができる。

```
00930
00940 ’ダミーベクトル（現時点確認用）
00950 vector(1)  _00__now_{%pcnt}{%com}
00960
```

被説明変数の変数変換（0=無変換、1=対数、2=階差、3=対数階差）ループ。なお、被説明変数と、説明変数のうち所得変数（GDP）は同じ変数変換をすると仮定している。

変数変換を EViews で使う書式に書き直すため、変数名の先頭と末尾に付与すべき変換名や括弧を文字変数で準備する。

```
00970
00980 for !ConvDep = 0 to  3   ’0:（none）  1:（log）  2:（dif）  3:（diflog）
```

```
00990         if !ConvDep = 1 then
01000                 %BefConvDep = "log("
01010                 %AftConvDep = ")"
01020         else
01030         if !ConvDep = 2 then
01040                 %BefConvDep = "d("
01050                 %AftConvDep = ")"
01060         else
01070         if !ConvDep = 3 then
01080                 %BefConvDep = "dlog("
01090                 %AftConvDep = ")"
01100         else
01110                 %BefConvDep = "
01120                 %AftConvDep = "
01130         endif
01140         endif
01150         endif
01160
01170         %depvarorg = "m" + %com + %pcnt
01180         %depvar = %BefConvDep + %depvarorg + %AftConvDep
01190         %indepvar1 = %BefConvDep + "gdp"+ %AftConvDep
01200
```

2 番目の説明変数（輸入価格全体もしくは輸入価格と国内価格の相対価格）に関する変数変換。上と同様に変数名の先頭と末尾につけるべき記号を準備している。ただし、ラグ構造の有無（1610 行～1650 行）と、絶対価格・相対価格のどちらを使うか（1890 行～1930 行）を後で決めるため、EViews 書式での変数名の準備はここでは行っていない。

```
01210
01220 for !ConvIndep2 = 0 to 3   '  0: (none)  1: (log)  2: (dif)  3: (diflog)
01230         if !ConvIndep2 = 1 then
01240                 %BefConvIndep2 = "log("
01250                 %AftConvIndep2 = ")"
01260         else
```

```
01270           if !ConvIndep2 = 2 then
01280                   %BefConvIndep2 = "d("
01290                   %AftConvIndep2 = ")"
01300           else
01310           if !ConvIndep2 = 3 then
01320                   %BefConvIndep2 = "dlog("
01330                   %AftConvIndep2 = ")"
01340           else
01350                   %BefConvIndep2 = "
01360                   %AftConvIndep2 = "
01370           endif
01380           endif
01390           endif
01400
```

3番目の説明変数（輸出側の国のオファー価格と、輸出国が報告国市場で直面する競争者価格の相対価格）に関する変数変換。上と同様に変数名の先頭と末尾につけるべき記号を準備している。ただし、ラグ構造の有無（1670行～1710行）を後で決めるため、EViews書式での変数名の準備はここでは行っていない。

```
01410
01420   for !ConvIndep3 = 0 to 3   ’  0:(none)  1:(log)  2:(dif)  3:(diflog)
01430           if !ConvIndep3 = 1 then
01440                   %BefConvIndep3 = "log("
01450                   %AftConvIndep3 = ")"
01460           else
01470           if !ConvIndep3 = 2 then
01480                   %BefConvIndep3 = "d("
01490                   %AftConvIndep3 = ")"
01500           else
01510           if !ConvIndep3 = 3 then
01520                   %BefConvIndep3 = "dlog("
01530                   %AftConvIndep3 = ")"
01540           else
01550                   %BefConvIndep3 = "
```

```
01560                   %AftConvIndep3 = ″
01570           endif
01580           endif
01590           endif
01600
01610   for !LagIndep2 = 0 to 1  ’ 0: (none)    1: (lag)
01620           %lag2 = ″
01630           if !LagIndep2 = 1 then
01640                   %lag2 = ″(-1)″
01650           endif
01660
01670   for !LagIndep3 = 0 to  1  ’ 0: (none)    1: (lag)
01680           %lag3 = ″
01690           if !LagIndep3 = 1 then
01700                   %lag3 = ″(-1)″
01710           endif
01720
01730           %indepvar3 = %BefConvIndep3 + %pcnt + ″px″ + %com + %lag3
                 + ″ + %pcnt + ″pc″ + %com + %lag3 + %AftConvIndep3 + ″
01740
```

被説明変数の１期ラグ項を説明変数群に含めるかどうかの決定。

```
01750
01760           For !WithWithoutLag = 0 To  1
01770                   %indepvar4 = ″
01780                   if !WithWithoutLag = 1 then
01790                           %indepvar4 =  %BefConvDep + %depvarorg + ″(-1)″
                                                                + %AftConvDep
01800                   endif
01810
01820                   %indepvar4 = ″ + %indepvar4
```

```
01830
```

輸入価格を絶対価格にするか相対価格にするかの決定。

```
01840                 For !RelativePM = 0 To 1
01850
01860                 !counter=!counter+1
01870
01880  %indepvar2 =  %BefConvIndep2 + "pm" + %com + "lc" + %lag2 + %AftConvIndep2

01890  if !RelativePM = 1 then
01900          %indepvar2 =  %BefConvIndep2 + "pm" + %com + "lc" + %lag2
                              + "/pgdp" + %lag2+ %AftConvIndep2
01910  endif

01920  %indepvar2 = " + %indepvar2 + "

01930
01940  %specs = %depvar + " c "
        +  %indepvar1+  %indepvar2+  %indepvar3+  %indepvar4
01950  equation eq999.ls {%specs}  ' EViews による回帰計算
01960
```

説明変数の回帰係数の t 値を EViews の出力より抽出する。 それぞれ
準備した変数（名前が属性をそのまま表す）に格納する。(計算上は特に
不要だが、プログラムの視認性向上のためである)

```
01970          t_gdp = @tstats(2)
01980          t_p1 = @tstats(3)
01990          t_p2 = @tstats(4)
02000          dw_stat = @dw
```

被説明変数のラグ項が入っている場合、h 統計量を計算するが、その定義上、
h 統計量が虚数になることがある。そのような定式化は採用にふさわしく
ないため排除する。h 統計量が虚数になった場合には、その値に 9999 と
いう magic number を入れて区別する。

また、被説明変数のラグ項が入っていない場合、h 統計量はもとより対応する回帰係数が存在しないが、そこにはすべて-9999 を入れておく。h 統計量は漸近的に標準正規分布に従うので、2290 行で h_stat < 1.645（片側５％有意水準で帰無仮説 H_0: ρ=0 を棄却しない）という判定条件を満たす定式化を選ぶが、被説明変数のラグ項がない場合には他の条件だけで判定するため。

```
02010         if !WithWithoutLag = 1 then
02020                         coef_lag = @coefs(5)
02030                         t_lag = @tstats(5)
02040                         h_dev =  1 - @regobs * @stderrs(5)^2
02050                         if h_dev >0 then
02060                            h_stat =
                                 @abs( (1-dw_stat/2)*(@regobs/h_dev)^(1/2) )
02070                         else
02080                            h_stat =-9999
02090                         endif
02100                 else
02110                         coef_lag = 0
02120                         t_lag = 9999
02130                         h_dev = 9999
02140                         h_stat =-9999
02150         endif
02160
02170         output_detector = 1
02180
```

「判定」ブロック。output_detector という変数に 1（true）を準備しておき、各説明要因の有意性や Durbin-Watson 統計量の範囲、h 統計量の条件など、すべて満たしている間は

1（true）×1（true）=1（true）

となり、途中で一つでも条件を満たさないものが出るとそこで

1（true）×0（false）=0（false）

となり、結果的にその定式化は採用されない。

```
02190
02200         ’以下の条件を全部満たせば「採用」
```

```
02210        output_detector = output_detector * ( coef_lag <1 )
02220        output_detector = output_detector * ( t_gdp > !t_critical )
02230        output_detector = output_detector * ( t_p1   < -!t_critical )
02240        output_detector = output_detector * ( t_p2   < -!t_critical )
02250        output_detector = output_detector * ( t_lag  > !t_critical )
02260        output_detector = output_detector * (1 < dw_stat )
02270        output_detector = output_detector * ( dw_stat < 3 )
02280        output_detector = output_detector * ( h_dev > 0 )
02290        output_detector = output_detector * (h_stat < 1.645)
```

なお、被説明変数のラグ項を説明要因に含む場合、2210 行にて係数の値が 1 よりも小さいかどうか判定しているが、同時に 2250 行でその係数の t 値が critical value （正値）を超えるかどうか判定しているため、当該係数は必然的に 0 < 係数 < 1 を満たすことになる。

判定の結果「採用」となった定式化の場合、各 t 値やその最小値、相手国や財番号などの情報を行列に格納する。

```
02300        if output_detector = 1 then
02310                !OKcounter = !OKcounter+1
02320                _{%pcntlist}_m{%com}_tstats( !OKcounter , 1 ) = !counter
02330                tstats(1) = @abs(t_gdp)
02340                tstats(2) = @abs(t_p1)
02350                tstats(3) = @abs(t_p2)
02360                tstats(4) = @abs(t_lag)
02370        _{%pcntlist}_m{%com}_tstats(!OKcounter, 2) = @min(  tstats  )
02380        _{%pcntlist}_m{%com}_tstats(!OKcounter, 3) =  !reportcntry
02390        _{%pcntlist}_m{%com}_tstats(!OKcounter, 4) =  !pcntry
02400        _{%pcntlist}_m{%com}_tstats(!OKcounter, 5) =  !commodity
02410        _{%pcntlist}_m{%com}_tstats(!OKcounter, 6) = dw_stat
02420        _{%pcntlist}_m{%com}_tstats(!OKcounter, 7) = h_stat
02430
02440
```

同様に、「採用」となった定式化の EViews の書式（関数型）も文字列ベクトルに格納しておく。

```
02450
02460                   _{%pcntlist}_m{%com}_func( !OKcounter ) = %specs
02470           endif
02480
02490                   next ' !RelativePM
02500           next ' !WithWithoutLag
02510
02520   next ' !LagIndep3
02530   next ' !LagIndep2
02540   next ' !ConvIndep3
02550   next ' !ConvIndep2
02560   next '  !ConvDep
02570
02580   ' 一次変数の消去
02590   delete  _00__now_{%pcnt}{%com}
02600
02610   next ' !Commodity
02620   next '!pcntry
02630
02640   _svec_time(2) = @date + " + @time
02650   ' ============ 計算作業ブロック終了 ============
```

判定の結果「採用」となった定式化と、それに関する諸情報は各国 WF の変数として WF 内に　保存されている。後半はこれら情報をテキストファイルに書き出すブロックである。

```
02660
02670   ' *** ファイルへの書き出し ***
```

親 WF から各国 WF を呼び出し、それぞれの情報を親 WF の管理下でファイルへ書き出す構造としている。

```
02680
02690   workfile {%workdate}_00_com3link a range_total
                ' ------------------------- parent workfile
02700   ' --- setting cntry mdl ---
```

```
02710  %rcntlist = _list_cntry(!reportcntry)
02720  %rcntlabel = _label_cntry(!reportcntry)
02730
02740  %zworkfile = %workdate + %rcntlist ' cntry workfile name
02750
02760          workfile %zworkfile a range_total ' --------------- cntry model
02770          smpl  1970 2009
02780          call Cntrylabel( 15 ,  %datapath )
02790
02800  ' ******************************************
02810          svector(2) ComVec
02820          ComVec(1) = "1"
02830          ComVec(2) = "3"
02840
```

輸入関数の推定時（670 行～700 行）と同様、「自国からの輸入」がない国（ユーロ圏以外）は、自国の番号をスキップする必要がある。

```
02850
02860  for !pcntry = 1 to 15
         '  ---------------------------------------------- partner country
02870          if (!pcntry = !reportcntry) * (!reportcntry <> 15) then
02880                  !pcntry = !pcntry + 1
02890          endif
02900
02910      %pcnt = _label_cntry(!pcntry)
02920          %pcntlist = _list_cntry(!pcntry)
02930
02940  for !commodity = 1 to 2
           '  ------------------------------------------------ commodity
02950          vector(1200) _vec1
02960
02970          %com = ComVec(!commodity)
02980          _vec1 = @columnextract(_{%pcntlist}_m{%com}_tstats, 3)
02990
```

```
03000          !dumcount = 0
03010          !temp = _vec1(1)
03020          while !temp <> 0
03030                  !dumcount=!dumcount+1
03040                  !temp = _vec1(!dumcount+1)
03050          wend
03060          if _vec1(1)=0 then
03070                  !dumcount=1
03080          endif
03090
03100          svector(!dumcount)   _{%pcntlist}_m{%com}_func
03110          matrix(!dumcount, 7) _{%pcntlist}_m{%com}_tstats
03120
03130          if @isobject("_specvec") then
03140                    delete _specvec
03150          endif
03160
```

EViews の仕様のために、文字列ファイルと数字ファイルの出力の仕方が異なる。数字の場合は行列をそのまま書き出せるが、文字列の場合は一旦文字列ベクトルを EViews 内で固定化し、それを出力する方式を取っている。（単にプログラム開発者の知識・能力不足かもしれない）

```
03170
03180          ' 採用定式化（文字列）のファイル出力
03190          freeze(_specvec)  _{%pcntlist}_m{%com}_func
03200          %workfile = %outputpath + %rcntlist + "_" + %pcntlist + "_m"
                                                    + %com + "_func.txt"
03210          _specvec.save(t=text) %workfile
03220
03230          ' 諸統計量（数字）のファイル出力
03240          %workfile = %outputpath +  %rcntlist + "_" + %pcntlist + "_m"
                                                    + %com + "_tstat.txt"
03250          _{%pcntlist}_m{%com}_tstats.write(t=txt) %workfile
03260
03270  next ' !commodity
```

```
03280   next '!pcntry
03290
03300   ' 一時的変数の消去
03310   delete   t_gdp
03320   delete   t_p1
03330   delete   t_p2
03340   delete   t_lag
03350
03360   delete tstats
03370   delete _list_cntry
03380   delete _label_cntry
03390   delete com_sitc
03400
03410   ' 一時的関数格納先の消去
03420   delete eq999
03430
03440   ' ワークスペースの保存と終了
03450   wfsave %zworkfile
03460   wfclose
03470
03480
```

複数国を連続して行う場合には 220 行と合わせてコメント行を外し、
ループを形成する。

```
03490
03500   '    next ' !reportcntry
03510   '  === End of File ===
```

（以上）

資料篇２－４．国ラベル・国リスト作成のためのサブルーチン（EViews）

```
00010  ’***** Sub-Routine ************************************
00020  ’*************     国ラベルの作成      ********************
00030  ’*******************************************************
00040  ’--- File Name
00050  ’MySub_CntryLabel    (include MySub_CntryLabel)
00060  ’--- Routine Name
00070  ’CntryLabel             (call CntryLabel)
00080  ’
00090  ’--- 機能
00100  ’国ラベルの作成
00110  ’
00120  ’***** 入力 *****
00130  ’--- 前もって存在している必要がある変数：
00140  ’なし
00150  ’--- 引数
00160  ’１：国数（scalar）
00170  ’
00180  ’***** 出力 *****
00190  １：国リストベクトル（国番号＋国ラベル）
00200  ２：国ラベルベクトル
00210  ’
00220  ’2013.12.29.
00230  ’
00240  subroutine CntryLabel(   scalar !noofcntry   )
00250  svector(!noofcntry) _label_cntry
00260  svector(!noofcntry) _list_cntry
00270
       現時点では国番号06：カンボジア（khm）、08：ラオス（lao）を
       抜いた15か国・地域
00280
00290  %allcntrynbr ="010203040507091011121314151617"
00300  %allcntrylbl ="auschnhkgidnjpnkormysnzlphlsgpthatwnusavnmeur"
00310
00320  for !i = 1 to !noofcntry
```

```
00330          !zi=(!i-1)*2+1
00340          %nbr=@mid(%allcntrynbr,!zi,2)
00350          !zi=(!i-1)*3+1
00360          %lbl=@mid(%allcntrylbl,!zi,3)
00370
00380          _label_cntry(!i)=%lbl
00390          _list_cntry(!i)= "_"+%nbr+"_"+%lbl
00400   next ' !i
00410
00420   endsub
00430   '  === End of File ===
```

（以上）

資料篇２－５．選別プログラムの出力を管理するプログラム（Excel VBA）

本来 VBA のプログラムは行番号に依存しないが、ここでは解説のために行番号をつけてある。行番号のない行がコメント行である。また、行番号がある行でも、最初に出てくる（空白以外の）記号が「’」である行はプログラム内に書かれたコメント行である。これらの「コメント」を取り外せば（行番号も不要）、そのまま作動するプログラムとなっている。

```
00010   Sub Sub1_ReadTstats()
00020   ’
00030   ’ ******* Select Reporting Country Number *******
        この例はオーストラリア（国番号 1)。報告国ごとにファイルを準備する
        必要があるので、ファイルごとコピーし、報告国番号を変えておく。
00040
00050               rcntry = 1     ’ input country code
00060
00070   ’ ****************************************
00080
00090   ’ 変数定義
00100   Dim InputfileTstat As String
00110   Dim InputfileFunc As String
00120   Dim Inputname As String
00130   Dim CntCode(15) As String
00140   Dim CntLabel(15) As String
00150   Dim ComCode(2) As String
00160   Dim TempName As String
00170
00180   Dim strSheetName As String
00190   Dim NoofSpecs As Integer
00200   Dim BestSpecNo As Integer
00210   Dim MaxTvalue As Double
00220   Dim strBestSpec As String
00230
00240   Dim Counter As Integer
00250   Dim Header As String
```

```
00260
00270  ’国ラベルの設定
00280  CntCode(1) = "_01_aus": CntCode(2) = "_02_chn": CntCode(3) = "_03_hkg"
00290  CntCode(4) = "_04_idn": CntCode(5) = "_05_jpn": CntCode(6) = "_07_kor"
00300  CntCode(7) = "_09_mys": CntCode(8) = "_10_nzl": CntCode(9) = "_11_phl"
00310  CntCode(10) ="_12_sgp": CntCode(11) = "_13_tha": CntCode(12) = "_14_twn"
00320  CntCode(13) ="_15_usa": CntCode(14) = "_16_vnm": CntCode(15) = "_17_eur"
00330
00340  CntLabel(1) = "aus": CntLabel(2) = "chn": CntLabel(3) = "hkg"
00350  CntLabel(4) = "idn": CntLabel(5) = "jpn": CntLabel(6) = "kor"
00360  CntLabel(7) = "mys": CntLabel(8) = "nzl": CntLabel(9) = "phl"
00370  CntLabel(10) = "sgp": CntLabel(11) = "tha": CntLabel(12) = "twn"
00380  CntLabel(13) = "usa": CntLabel(14) = "vnm": CntLabel(15) = "eur"
00390
00400  ’財種別のラベル設定
00410  ComCode(1) = "1": ComCode(2) = "3"
00420
00430
00440  ’=========== 第１部 ===========
00450
```

このプログラムでは、第１部では各国あたり 28 もしくは 30 ある「財別・相手国別輸入関数」の類型それぞれについて「統計量を格納したファイル」および「採用可能な定式化を格納したファイル」の２つずつ、すなわち一国あたり 56 個から 60 個のファイルから情報収集し、t 値の大きさに従って並べ替える。第２部はそれらのサマリーを作成するブロックである。

なお、これらはもともと別々の独立したプログラムであるため、変数宣言を適切に　行えば分割しても問題なく動作する。

```
00460  ’---------- 出力ファイルからの情報収集 -----------
00470  ’相手国（1:Aus～15:Eur）
00480  For pcntry = 1 To 15
00490
00500  ’ユーロ圏以外は「報告国≠相手国」である必要がある
00510  If (rcntry <> pcntry) + (rcntry = 15) Then
```

```
00520
00530       For zi = 1 To 2
00540
00550           TempName = CntLabel(pcntry) & ComCode(zi)
00560
00570   ’相手国・財別に格納用シートを作成
00580           Sheets.Add After:=Sheets(Sheets.Count)
00590           Sheets(Sheets.Count).Select
00600           Sheets(Sheets.Count).Name = TempName
00610
00620   ’オープンするファイル名の設定
00630   Inputname = CntCode(rcntry) & "_" & CntCode(pcntry) & "_m" & ComCode(zi)
00640   InputfileTstat = "TEXT;R:¥Personal¥201312 Com3Link¥Outputs¥"
         & Inputname & "_tstat.txt"
00650   InputfileFunc = "R:¥Personal¥201312 Com3Link¥Outputs¥" & Inputname
         & "_func.txt"
00660 ’
00670       With ActiveSheet.QueryTables.Add(Connection:= _
00680           InputfileTstat, _
00690           Destination:=Range("$A$1"))
00700           .Name = Inputname & "_tstat"
00710           .Refresh BackgroundQuery:=False
00720       End With
00730
00740       Open InputfileFunc For Input As #1
00750
```

定式化の入ったファイルは、最初の３行に不要のヘッダ部分があるため、
そこをスキップしている。

```
00760       For zj = 1 To 3
00770           Line Input #1, ix
00780       Next zj
00790
```

改めてファイル内容をシートに配置する。

```
00800       Range("n1").Select
00810       Counter = 0
00820       Do While Not EOF(1)
00830           Line Input #1, ix
00840           Counter = Counter + 1
00850           Header = "R" & Counter
00860           ix = Right(ix, Len(ix) - Len(Header))
00870           Selection.Value = ix
00880           ActiveCell.Offset(1, 0).Select
00890       Loop
00900       Close #1
00910
00920       Range("n1").Select
00930
00940           Range(Selection, Selection.End(xlDown)).Select
00950           Range(Selection, Selection.End(xlDown)).Select
00960           Range(Selection, Selection.End(xlToRight)).Select
00970           Range(Selection, Selection.End(xlToRight)).Select
00980           ActiveWorkbook.Worksheets(TempName).Sort.SortFields.Clear
00990
01000   '---------- t 値の降順に並べ替え -----------
01010   ActiveWorkbook.Worksheets(TempName).Sort.SortFields.Add
         Key:=Range("B:B"), _
01020    SortOn:=xlSortOnValues, Order:=xlDescending,
         DataOption:=xlSortNormal
01030           With ActiveWorkbook.Worksheets(TempName).Sort
01040               .SetRange Range("A:N")
01050               .Header = xlGuess
01060               .MatchCase = False
01070               .Orientation = xlTopToBottom
01080               .SortMethod = xlPinYin
01090               .Apply
01100           End With
01110           Range("A1").Select
01120
01130       Next zi
```

```
01140
01150   End If ’ 「報告国≠相手国」の条件終了
01160   Next pcntry
01170 ’
01180 ’
01190 ’
```

第２部では、上で収集され、シート上に配置された「最小ｔ値の最大値」と、それを与える定式化のうち、さらに一番上位に配置されたもの（係数の有意性で見て最良のもの）を収集し、第１シートにそのサマリーを配置するものである。

なお、「候補が一つも存在しない」場合にそれを排除するルーチンを設けていないため、そのような場合には「候補数１」「内容=0」のようなものが出力される。

```
01200   ’ =========== 第２部 ===========
01210   ’ ----------  各シートからの情報収集 -----------
01220       For i = 2 To Worksheets.Count
01230           Worksheets(i).Select
01240           strSheetName = ActiveSheet.Name
01250           Counter = 0
01260           Range("a1").Select
01270           BestSpecNo = Selection.Value
01280           MaxTvalue = ActiveCell.Offset(0, 1).Value
01290           strBestSpec = ActiveCell.Offset(0, 13).Value
01300           Do While Selection.Value <> ""
01310               Counter = Counter + 1
01320               ActiveCell.Offset(1, 0).Select
01330           Loop
01340           Range("a1").Select
01350
01360           Worksheets(1).Select
01370           Range("a1").Select
01380           ActiveCell.Offset(i, 0).Select
01390           Selection.Value = strSheetName
01400           ActiveCell.Offset(0, 1).Select
```

```
01410            Selection.Value = Counter
01420            ActiveCell.Offset(0, 1).Select
01430            Selection.Value = MaxTvalue
01440            ActiveCell.Offset(0, 1).Select
01450            Selection.Value = BestSpecNo
01460            ActiveCell.Offset(0, 1).Select
01470            Selection.Value = strBestSpec
01480
01490        Next i
01500
```

最後に、第１シート（サマリー表）にカーソルを移して終了する。

```
01510        Worksheets(1).Select
01520        Range("A1").Select
01530
01540    End Sub
```

（以上）

資料篇２－６．選別された「最良」定式化一覧
（行頭の括弧内は「相手国コード＋財コード」を示す）

（１）オーストラリア

```
[aus1]  (none)
[chn1]  log(m1chn) = f[ log(gdp), d(pm11c(-1)/pgdp(-1)),
                        d(chnpx1(-1)/chnpc1(-1)), log(m1chn(-1)) ]
[hkg1]  dlog(m1hkg) = f[ dlog(gdp), log(pm11c), log(hkgpx1(-1)/hkgpc1(-1)) ]
[idn1]  log(m1idn) = f[ log(gdp), log(pm11c(-1)), d(idnpx1/idnpc1) ]
[jpn1]  m1jpn = f[ gdp, pm11c(-1), log(jpnpx1/jpnpc1), m1jpn(-1) ]
[kor1]  m1kor = f[ gdp, d(pm11c(-1)), dlog(korpx1/korpc1), m1kor(-1) ]
[mys1]  log(m1mys) = f[ log(gdp), pm11c(-1)/pgdp(-1),
                        dlog(myspx1(-1)/myspc1(-1)) ]
[nzl1]  m1nzl = f[ gdp, d(pm11c(-1)), nzlpx1(-1)/nzlpc1(-1), m1nzl(-1) ]
[phl1]  m1phl = f[ gdp, pm11c(-1)/pgdp(-1), d(phlpx1/phlpc1) ]
[sgp1]  dlog(m1sgp) = f[ dlog(gdp), dlog(pm11c), d(sgppx1(-1)/sgppc1(-1)) ]
[tha1]  log(m1tha) = f[ log(gdp), log(pm11c), log(thapx1(-1)/thapc1(-1)),
                        log(m1tha(-1)) ]
[twn1]  (none)
[usa1]  m1usa = f[ gdp, log(pm11c), log(usapx1(-1)/usapc1(-1)), m1usa(-1) ]
[vnm1]  m1vnm = f[ gdp, log(pm11c/pgdp), log(vnmpx1/vnmpc1) ]
[eur1]  dlog(m1eur) = f[ dlog(gdp), dlog(pm11c), eurpx1(-1)/eurpc1(-1) ]

[aus3]  (none)
[chn3]  log(m3chn) = f[ log(gdp), dlog(pm3lc(-1)/pgdp(-1)),
                        log(chnpx3(-1)/chnpc3(-1)) ]
[hkg3]  m3hkg = f[ gdp, log(pm3lc(-1)), log(hkgpx3(-1)/hkgpc3(-1)) ]
[idn3]  m3idn = f[ gdp, pm3lc(-1)/pgdp(-1), log(idnpx3/idnpc3) ]
[jpn3]  dlog(m3jpn) = f[ dlog(gdp), d(pm3lc(-1)), jpnpx3(-1)/jpnpc3(-1) ]
[kor3]  log(m3kor) = f[ log(gdp), d(pm3lc(-1)/pgdp(-1)),
                        korpx3(-1)/korpc3(-1) ]
[mys3]  d(m3mys) = f[ d(gdp), d(pm3lc(-1)), log(myspx3(-1)/myspc3(-1)) ]
[nzl3]  m3nzl = f[ gdp, dlog(pm3lc), d(nzlpx3/nzlpc3), m3nzl(-1) ]
[phl3]  (none)
[sgp3]  d(m3sgp) = f[ d(gdp), pm3lc, sgppx3(-1)/sgppc3(-1) ]
[tha3]  dlog(m3tha) = f[ dlog(gdp), d(pm3lc(-1)), log(thapx3/thapc3) ]
```

```
[twn3]  m3twn = f[ gdp, pm3lc(-1)/pgdp(-1), twnpx3(-1)/twnpc3(-1) ]
[usa3]  d(m3usa) = f[ d(gdp), pm3lc, usapx3(-1)/usapc3(-1) ]
[vnm3]  m3vnm = f[ gdp, log(pm3lc(-1)/pgdp(-1)), d(vnmpx3/vnmpc3), m3vnm(-1) ]
[eur3]  dlog(m3eur) = f[ dlog(gdp), dlog(pm3lc/pgdp), dlog(eurpx3/eurpc3) ]
```

（2）中国

```
[aus1]  d(m1aus) = f[ d(gdp), dlog(pm1lc/pgdp), log(auspx1/auspc1) ]
[chn1]  (none)
[hkg1]  log(m1hkg) = f[ log(gdp), pm1lc, hkgpx1(-1)/hkgpc1(-1), log(m1hkg(-1)) ]
[idn1]  d(m1idn) = f[ d(gdp), log(pm1lc(-1)/pgdp(-1)), dlog(idnpx1/idnpc1) ]
[jpn1]  m1jpn = f[ gdp, pm1lc(-1), log(jpnpx1/jpnpc1) ]
[kor1]  m1kor = f[ gdp, log(pm1lc/pgdp), log(korpx1/korpc1) ]
[mys1]  d(m1mys) = f[ d(gdp), pm1lc, dlog(myspx1(-1)/myspc1(-1)) ]
[nzl1]  m1nzl = f[ gdp, log(pm1lc/pgdp), d(nzlpx1/nzlpc1) ]
[phl1]  d(m1phl) = f[ d(gdp), pm1lc(-1), log(phlpx1/phlpc1) ]
[sgp1]  d(m1sgp) = f[ d(gdp), d(pm1lc), sgppx1/sgppc1 ]
[tha1]  m1tha = f[ gdp, pm1lc, d(thapx1/thapc1) ]
[twn1]  m1twn = f[ gdp, dlog(pm1lc/pgdp), d(twnpx1/twnpc1), m1twn(-1) ]
[usa1]  m1usa = f[ gdp, log(pm1lc/pgdp), log(usapx1(-1)/usapc1(-1)) ]
[vnm1]  m1vnm = f[ gdp, log(pm1lc/pgdp), log(vnmpx1/vnmpc1) ]
[eur1]  m1eur = f[ gdp, log(pm1lc(-1)), log(eurpx1/eurpc1) ]

[aus3]  d(m3aus) = f[ d(gdp), d(pm3lc(-1)), log(auspx3/auspc3) ]
[chn3]  (none)
[hkg3]  (none)
[idn3]  m3idn = f[ gdp, pm3lc(-1), dlog(idnpx3/idnpc3) ]
[jpn3]  log(m3jpn) = f[ log(gdp), dlog(pm3lc/pgdp), d(jpnpx3/jpnpc3) ]
[kor3]  dlog(m3kor) = f[ dlog(gdp), log(pm3lc(-1)), log(korpx3/korpc3) ]
[mys3]  m3mys = f[ gdp, log(pm3lc(-1)/pgdp(-1)), myspx3(-1)/myspc3(-1) ]
[nzl3]  d(m3nzl) = f[ d(gdp), pm3lc, dlog(nzlpx3/nzlpc3) ]
[phl3]  log(m3phl) = f[ log(gdp), dlog(pm3lc(-1)), phlpx3(-1)/phlpc3(-1) ]
[sgp3]  dlog(m3sgp) = f[ dlog(gdp), log(pm3lc), log(sgppx3(-1)/sgppc3(-1)) ]
[tha3]  m3tha = f[ gdp, pm3lc, d(thapx3/thapc3) ]
[twn3]  dlog(m3twn) = f[ dlog(gdp), pm3lc/pgdp, log(twnpx3(-1)/twnpc3(-1)) ]
```

```
[usa3]  log(m3usa) = f[ log(gdp), dlog(pm3lc(-1)/pgdp(-1)),
                        log(usapx3(-1)/usapc3(-1)), log(m3usa(-1)) ]
[vnm3]  log(m3vnm) = f[ log(gdp), pm3lc(-1), vnmpx3(-1)/vnmpc3(-1) ]
[eur3]  log(m3eur) = f[ log(gdp), log(pm3lc(-1)), d(eurpx3/eurpc3),
                        log(m3eur(-1)) ]
```

（3）香港

```
[aus1]  log(m1aus) = f[ log(gdp), log(pm1lc/pgdp), d(auspx1(-1)/auspc1(-1)),
                        log(m1aus(-1) ]
[chn1]  m1chn = f[ gdp, d(pm1lc/pgdp), d(chnpx1/chnpc1), m1chn(-1) ]
[hkg1]  (none)
[idn1]  m1idn = f[ gdp, pm1lc, log(idnpx1/idnpc1), m1idn(-1) ]
[jpn1]  m1jpn = f[ gdp, pm1lc, log(jpnpx1/jpnpc1), m1jpn(-1) ]
[kor1]  log(m1kor) = f[ log(gdp), pm1lc(-1)/pgdp(-1), korpx1(-1)/korpc1(-1) ]
[mys1]  log(m1mys) = f[ log(gdp), d(pm1lc(-1)), dlog(myspx1/myspc1),
                        log(m1mys(-1)) ]
[nzl1]  log(m1nzl) = f[ log(gdp), log(pm1lc/pgdp), log(nzlpx1/nzlpc1),
                        log(m1nzl(-1)) ]
[phl1]  log(m1phl) = f[ log(gdp), log(pm1lc(-1)/pgdp(-1)), dlog(phlpx1/phlpc1),
                        log(m1phl(-1)) ]
[sgp1]  log(m1sgp) = f[ log(gdp), dlog(pm1lc(-1)/pgdp(-1)), d(sgppx1/sgppc1),
                        log(m1sgp(-1)) ]
[tha1]  log(m1tha) = f[ log(gdp), pm1lc, d(thapx1/thapc1), log(m1tha(-1)) ]
[twn1]  m1twn = f[ gdp, dlog(pm1lc(-1)/pgdp(-1)), dlog(twnpx1(-1)/twnpc1(-1)),
                        m1twn(-1) ]
[usa1]  log(m1usa) = f[ log(gdp), log(pm1lc/pgdp), log(usapx1/usapc1) ]
[vnm1]  log(m1vnm) = f[ log(gdp), log(pm1lc/pgdp), vnmpx1/vnmpc1 ]
[eur1]  log(m1eur) = f[ log(gdp), dlog(pm1lc(-1)/pgdp(-1)),
                        log(eurpx1(-1)/eurpc1(-1)), log(m1eur(-1)) ]

[aus3]  log(m3aus) = f[ log(gdp), log(pm3lc/pgdp), dlog(auspx3/auspc3),
                        log(m3aus(-1)) ]
[chn3]  log(m3chn) = f[ log(gdp), dlog(pm3lc(-1)/pgdp(-1)),
                        dlog(chnpx3(-1)/chnpc3(-1)), log(m3chn(-1)) ]
[hkg3]  (none)
```

```
[idn3]  log(m3idn) = f[ log(gdp), log(pm3lc/pgdp), log(idnpx3(-1)/idnpc3(-1)),
                        log(m3idn(-1)) ]
[jpn3]  log(m3jpn) = f[ log(gdp), log(pm3lc(-1)/pgdp(-1)),
                        log(jpnpx3(-1)/jpnpc3(-1)) ]
[kor3]  log(m3kor) = f[ log(gdp), pm3lc/pgdp, log(korpx3/korpc3) ]
[mys3]  m3mys = f[ gdp, dlog(pm3lc(-1)/pgdp(-1)), dlog(myspx3(-1)/myspc3(-1)),
                        m3mys(-1) ]
[nzl3]  m3nzl = f[ gdp, log(pm3lc/pgdp), log(nzlpx3/nzlpc3), m3nzl(-1) ]
[phl3]  m3phl = f[ gdp, log(pm3lc), phlpx3/phlpc3 ]
[sgp3]  log(m3sgp) = f[ log(gdp), pm3lc, log(sgppx3/sgppc3) ]
[tha3]  log(m3tha) = f[ log(gdp), pm3lc/pgdp, log(thapx3/thapc3) ]
[twn3]  log(m3twn) = f[ log(gdp), log(pm3lc/pgdp), twnpx3/twnpc3 ]
[usa3]  log(m3usa) = f[ log(gdp), log(pm3lc(-1)/pgdp(-1)), d(usapx3/usapc3) ]
[vnm3]  log(m3vnm) = f[ log(gdp), log(pm3lc), vnmpx3(-1)/vnmpc3(-1),
                        log(m3vnm(-1)) ]
[eur3]  log(m3eur) = f[ log(gdp), log(pm3lc/pgdp), eurpx3(-1)/eurpc3(-1),
                        log(m3eur(-1)) ]
```

（4）インドネシア

```
[aus1]  log(m1aus) = f[ log(gdp), pm1lc(-1)/pgdp(-1), auspx1/auspc1 ]
[chn1]  log(m1chn) = f[ log(gdp), d(pm1lc), log(chnpx1(-1)/chnpc1(-1)) ]
[hkg1]  m1hkg = f[ gdp, log(pm1lc), dlog(hkgpx1(-1)/hkgpc1(-1)) ]
[idn1]  (none)
[jpn1]  dlog(m1jpn) = f[ dlog(gdp), dlog(pm1lc/pgdp),
                        d(jpnpx1(-1)/jpnpc1(-1)) ]
[kor1]  log(m1kor) = f[ log(gdp), d(pm1lc/pgdp), d(korpx1/korpc1),
                        log(m1kor(-1)) ]
[mys1]  log(m1mys) = f[ log(gdp), pm1lc/pgdp, myspx1(-1)/myspc1(-1) ]
[nzl1]  dlog(m1nzl) = f[ dlog(gdp), log(pm1lc), nzlpx1(-1)/nzlpc1(-1) ]
[phl1]  m1phl = f[ gdp, d(pm1lc/pgdp), phlpx1/phlpc1, m1phl(-1) ]
[sgp1]  m1sgp = f[ gdp, log(pm1lc), log(sgppx1(-1)/sgppc1(-1)), m1sgp(-1) ]
[tha1]  log(m1tha) = f[ log(gdp), d(pm1lc/pgdp), thapx1(-1)/thapc1(-1),
                        log(m1tha(-1)) ]
[twn1]  log(m1twn) = f[ log(gdp), log(pm1lc), dlog(twnpx1(-1)/twnpc1(-1)),
                        log(m1twn(-1)) ]
```

```
[usa1]  log(m1usa) = f[ log(gdp), pm1lc/pgdp, d(usapx1(-1)/usapc1(-1)) ]
[vnm1]  log(m1vnm) = f[ log(gdp), pm1lc(-1), d(vnmpx1/vnmpc1) ]
[eur1]  d(m1eur) = f[ d(gdp), dlog(pm1lc/pgdp), d(eurpx1/eurpc1) ]

[aus3]  log(m3aus) = f[ log(gdp), pm3lc, dlog(auspx3/auspc3) ]
[chn3]  dlog(m3chn) = f[ dlog(gdp), log(pm3lc/pgdp), chnpx3(-1)/chnpc3(-1) ]
[hkg3]  log(m3hkg) = f[ log(gdp), d(pm3lc/pgdp), log(hkgpx3(-1)/hkgpc3(-1)),
                        log(m3hkg(-1)) ]
[idn3]  (none)
[jpn3]  log(m3jpn) = f[ log(gdp), log(pm3lc(-1)/pgdp(-1)),
                        jpnpx3(-1)/jpnpc3(-1) ]
[kor3]  d(m3kor) = f[ d(gdp), dlog(pm3lc/pgdp), dlog(korpx3/korpc3),
                        d(m3kor(-1)) ]
[mys3]  log(m3mys) = f[ log(gdp), dlog(pm3lc(-1)), myspx3(-1)/myspc3(-1),
                        log(m3mys(-1)) ]
[nzl3]  m3nzl = f[ gdp, log(pm3lc), log(nzlpx3(-1)/nzlpc3(-1)), m3nzl(-1) ]
[phl3]  m3phl = f[ gdp, log(pm3lc), phlpx3(-1)/phlpc3(-1) ]
[sgp3]  log(m3sgp) = f[ log(gdp), d(pm3lc/pgdp), d(sgppx3(-1)/sgppc3(-1)),
                        log(m3sgp(-1)) ]
[tha3]  log(m3tha) = f[ log(gdp), d(pm3lc/pgdp), thapx3(-1)/thapc3(-1),
                        log(m3tha(-1)) ]
[twn3]  log(m3twn) = f[ log(gdp), log(pm3lc), twnpx3/twnpc3 ]
[usa3]  dlog(m3usa) = f[ dlog(gdp), log(pm3lc/pgdp),
                        log(usapx3(-1)/usapc3(-1)) ]
[vnm3]  log(m3vnm) = f[ log(gdp), dlog(pm3lc/pgdp), d(vnmpx3/vnmpc3) ]
[eur3]  log(m3eur) = f[ log(gdp), log(pm3lc), log(eurpx3(-1)/eurpc3(-1)) ]
```

（5）日本

```
[aus1]  log(m1aus) = f[ log(gdp), d(pm1lc(-1)/pgdp(-1)), d(auspx1/auspc1),
                        log(m1aus(-1) ]
[chn1]  m1chn = f[ gdp, log(pm1lc), d(chnpx1/chnpc1) ]
[hkg1]  dlog(m1hkg) = f[ dlog(gdp), d(pm1lc/pgdp), log(hkgpx1(-1)/hkgpc1(-1)) ]
[idn1]  log(m1idn) = f[ log(gdp), d(pm1lc(-1)/pgdp(-1)),
                        d(idnpx1(-1)/idnpc1(-1)), log(m1idn(-1)) ]
[jpn1]  (none)
```

```
[kor1]  dlog(m1kor) = f[ dlog(gdp), dlog(pm1lc(-1)/pgdp(-1)),
                         d(korpx1(-1)/korpc1(-1)) ]
[mys1]  d(m1mys) = f[ d(gdp), dlog(pm1lc/pgdp), log(myspx1(-1)/myspc1(-1)) ]
[nzl1]  (none)
[phl1]  (none)
[sgp1]  dlog(m1sgp) = f[ dlog(gdp), dlog(pm1lc(-1)/pgdp(-1)),
                         log(sgppx1(-1)/sgppc1(-1)), dlog(m1sgp(-1)) ]
[tha1]  log(m1tha) = f[ log(gdp), dlog(pm1lc/pgdp), d(thapx1/thapc1),
                         log(m1tha(-1)) ]
[twn1]  dlog(m1twn) = f[ dlog(gdp), dlog(pm1lc), log(twnpx1/twnpc1) ]
[usa1]  d(m1usa) = f[ d(gdp), d(pm1lc/pgdp), log(usapx1(-1)/usapc1(-1)) ]
[vnm1]  log(m1vnm) = f[ log(gdp), dlog(pm1lc(-1)/pgdp(-1)), d(vnmpx1/vnmpc1),
                         log(m1vnm(-1)) ]
[eur1]  log(m1eur) = f[ log(gdp), d(pm1lc), eurpx1/eurpc1, log(m1eur(-1)) ]

[aus3]  log(m3aus) = f[ log(gdp), pm3lc/pgdp, dlog(auspx3/auspc3) ]
[chn3]  log(m3chn) = f[ log(gdp), dlog(pm3lc(-1)), dlog(chnpx3(-1)/chnpc3(-1)),
                         log(m3chn(-1)) ]
[hkg3]  d(m3hkg) = f[ d(gdp), d(pm3lc(-1)), d(hkgpx3(-1)/hkgpc3(-1)),
                         d(m3hkg(-1)) ]
[idn3]  d(m3idn) = f[ d(gdp), dlog(pm3lc(-1)/pgdp(-1)), idnpx3(-1)/idnpc3(-1) ]
[jpn3]  (none)
[kor3]  log(m3kor) = f[ log(gdp), d(pm3lc(-1)/pgdp(-1)), korpx3(-1)/korpc3(-1),
                         log(m3kor(-1)) ]
[mys3]  m3mys = f[ gdp, dlog(pm3lc(-1)/pgdp(-1)), d(myspx3(-1)/myspc3(-1)),
                         m3mys(-1) ]
[nzl3]  log(m3nzl) = f[ log(gdp), log(pm3lc(-1)/pgdp(-1)),
                         d(nzlpx3(-1)/nzlpc3(-1)) ]
[phl3]  log(m3phl) = f[ log(gdp), log(pm3lc(-1)), dlog(phlpx3(-1)/phlpc3(-1)),
                         log(m3phl(-1)) ]
[sgp3]  log(m3sgp) = f[ log(gdp), log(pm3lc(-1)), log(sgppx3(-1)/sgppc3(-1)),
                         log(m3sgp(-1)) ]
[tha3]  d(m3tha) = f[ d(gdp), dlog(pm3lc(-1)), thapx3(-1)/thapc3(-1),
                         d(m3tha(-1)) ]
[twn3]  log(m3twn) = f[ log(gdp), dlog(pm3lc(-1)/pgdp(-1)),
                         dlog(twnpx3(-1)/twnpc3(-1)), log(m3twn(-1)) ]
```

```
[usa3]  d(m3usa) = f[ d(gdp), log(pm3lc), log(usapx3(-1)/usapc3(-1)) ]
[vnm3]  log(m3vnm) = f[ log(gdp), d(pm3lc(-1)/pgdp(-1)),
                        d(vnmpx3(-1)/vnmpc3(-1)), log(m3vnm(-1)) ]
[eur3]  d(m3eur) = f[ d(gdp), dlog(pm3lc(-1)), log(eurpx3(-1)/eurpc3(-1)) ]
```

（６）韓国

```
[aus1]  log(m1aus) = f[ log(gdp), d(pm1lc), d(auspx1(-1)/auspc1(-1)),
                        log(m1aus(-1) ]
[chn1]  m1chn = f[ gdp, pm1lc, chnpx1(-1)/chnpc1(-1) ]
[hkg1]  m1hkg = f[ gdp, d(pm1lc(-1)), dlog(hkgpx1(-1)/hkgpc1(-1)), m1hkg(-1) ]
[idn1]  dlog(m1idn) = f[ dlog(gdp), dlog(pm1lc), dlog(idnpx1/idnpc1),
                        dlog(m1idn(-1)) ]
[jpn1]  d(m1jpn) = f[ d(gdp), dlog(pm1lc/pgdp), dlog(jpnpx1/jpnpc1) ]
[kor1]  (none)
[mys1]  dlog(m1mys) = f[ dlog(gdp), log(pm1lc), dlog(myspx1(-1)/myspc1(-1)) ]
[nzl1]  log(m1nzl) = f[ log(gdp), log(pm1lc/pgdp), dlog(nzlpx1/nzlpc1) ]
[phl1]  log(m1phl) = f[ log(gdp), dlog(pm1lc(-1)/pgdp(-1)), d(phlpx1/phlpc1),
                        log(m1phl(-1)) ]
[sgp1]  log(m1sgp) = f[ log(gdp), pm1lc(-1)/pgdp(-1), sgppx1/sgppc1 ]
[tha1]  m1tha = f[ gdp, dlog(pm1lc/pgdp), dlog(thapx1(-1)/thapc1(-1)),
                        m1tha(-1) ]
[twn1]  log(m1twn) = f[ log(gdp), pm1lc/pgdp, twnpx1(-1)/twnpc1(-1) ]
[usa1]  log(m1usa)  =  f[  log(gdp),  d(pm1lc/pgdp),  usapx1(-1)/usapc1(-1),
log(m1usa(-1)) ]
[vnm1]  log(m1vnm) = f[ log(gdp), dlog(pm1lc), log(vnmpx1/vnmpc1) ]
[eur1]  d(m1eur) = f[ d(gdp), dlog(pm1lc/pgdp), dlog(eurpx1/eurpc1) ]

[aus3]  m3aus = f[ gdp, pm3lc, auspx3/auspc3 ]
[chn3]  m3chn = f[ gdp, log(pm3lc(-1)), d(chnpx3(-1)/chnpc3(-1)), m3chn(-1) ]
[hkg3]  m3hkg = f[ gdp, pm3lc, log(hkgpx3/hkgpc3) ]
[idn3]  log(m3idn) = f[ log(gdp), dlog(pm3lc/pgdp), idnpx3/idnpc3,
                        log(m3idn(-1)) ]
[jpn3]  dlog(m3jpn) = f[ dlog(gdp), d(pm3lc(-1)/pgdp(-1)),
                        d(jpnpx3(-1)/jpnpc3(-1)) ]
[kor3]  (none)
```

```
[mys3]  d(m3mys) = f[ d(gdp), pm3lc/pgdp, dlog(myspx3(-1)/myspc3(-1)) ]
[nzl3]  log(m3nzl) = f[ log(gdp), pm3lc/pgdp, nzlpx3/nzlpc3 ]
[phl3]  log(m3phl) = f[ log(gdp), pm3lc(-1)/pgdp(-1), dlog(phlpx3/phlpc3),
                        log(m3phl(-1)) ]
[sgp3]  log(m3sgp) = f[ log(gdp), log(pm3lc/pgdp), d(sgppx3(-1)/sgppc3(-1)) ]
[tha3]  d(m3tha) = f[ d(gdp), d(pm3lc), dlog(thapx3/thapc3) ]
[twn3]  log(m3twn) = f[ log(gdp), pm3lc/pgdp, dlog(twnpx3/twnpc3) ]
[usa3]  dlog(m3usa) = f[ dlog(gdp), d(pm3lc), log(usapx3(-1)/usapc3(-1)) ]
[vnm3]  log(m3vnm) = f[ log(gdp), log(pm3lc(-1)/pgdp(-1)),
                        vnmpx3(-1)/vnmpc3(-1) ]
[eur3]  (none)
```

（7）マレーシア

```
[aus1]  log(m1aus) = f[ log(gdp), log(pm1lc(-1)), d(auspx1/auspc1) ]
[chn1]  log(m1chn) = f[ log(gdp), log(pm1lc(-1)/pgdp(-1)),
                        chnpx1(-1)/chnpc1(-1) ]
[hkg1]  log(m1hkg) = f[ log(gdp), log(pm1lc), d(hkgpx1(-1)/hkgpc1(-1)) ]
[idn1]  log(m1idn) = f[ log(gdp), d(pm1lc), dlog(idnpx1/idnpc1),
                        log(m1idn(-1)) ]
[jpn1]  m1jpn = f[ gdp, log(pm1lc), log(jpnpx1/jpnpc1) ]
[kor1]  (none)
[mys1]  (none)
[nzl1]  log(m1nzl) = f[ log(gdp), pm1lc, log(nzlpx1/nzlpc1) ]
[phl1]  m1phl = f[ gdp, dlog(pm1lc/pgdp), dlog(phlpx1/phlpc1), m1phl(-1) ]
[sgp1]  d(m1sgp) = f[ d(gdp), dlog(pm1lc(-1)/pgdp(-1)),
                        log(sgppx1(-1)/sgppc1(-1)) ]
[tha1]  log(m1tha) = f[ log(gdp), d(pm1lc(-1)), d(thapx1/thapc1),
                        log(m1tha(-1)) ]
[twn1]  log(m1twn) = f[ log(gdp), log(pm1lc), dlog(twnpx1(-1)/twnpc1(-1)) ]
[usa1]  log(m1usa) = f[ log(gdp), log(pm1lc/pgdp), usapx1(-1)/usapc1(-1),
                        log(m1usa(-1)) ]
[vnm1]  log(m1vnm) = f[ log(gdp), log(pm1lc), log(vnmpx1/vnmpc1),
                        log(m1vnm(-1)) ]
[eur1]  m1eur = f[ gdp, dlog(pm1lc), log(eurpx1/eurpc1), m1eur(-1) ]
```

```
[aus3]  d(m3aus) = f[ d(gdp), dlog(pm3lc(-1)/pgdp(-1)),
                      log(auspx3(-1)/auspc3(-1)) ]
[chn3]  m3chn = f[ gdp, d(pm3lc(-1)), dlog(chnpx3/chnpc3), m3chn(-1) ]
[hkg3]  dlog(m3hkg) = f[ dlog(gdp), d(pm3lc(-1)/pgdp(-1)),
                      d(hkgpx3(-1)/hkgpc3(-1)) ]
[idn3]  log(m3idn) = f[ log(gdp), pm3lc, dlog(idnpx3/idnpc3), log(m3idn(-1)) ]
[jpn3]  dlog(m3jpn) = f[ dlog(gdp), d(pm3lc/pgdp), dlog(jpnpx3/jpnpc3) ]
[kor3]  log(m3kor) = f[ log(gdp), pm3lc(-1)/pgdp(-1),
                      log(korpx3(-1)/korpc3(-1)) ]
[mys3]  (none)
[nzl3]  m3nzl = f[ gdp, pm3lc(-1), nzlpx3(-1)/nzlpc3(-1) ]
[phl3]  (none)
[sgp3]  log(m3sgp) = f[ log(gdp), pm3lc(-1), log(sgppx3(-1)/sgppc3(-1)),
                      log(m3sgp(-1)) ]
[tha3]  log(m3tha) = f[ log(gdp), pm3lc(-1), dlog(thapx3(-1)/thapc3(-1)) ]
[twn3]  m3twn = f[ gdp, d(pm3lc/pgdp), twnpx3/twnpc3 ]
[usa3]  m3usa = f[ gdp, pm3lc(-1), log(usapx3(-1)/usapc3(-1)), m3usa(-1) ]
[vnm3]  log(m3vnm) = f[ log(gdp), d(pm3lc/pgdp), log(vnmpx3(-1)/vnmpc3(-1)) ]
[eur3]  log(m3eur) = f[ log(gdp), log(pm3lc), eurpx3/eurpc3 ]
```

（8）ニュージーランド

```
[aus1]  log(m1aus) = f[ log(gdp), log(pm1lc(-1)/pgdp(-1)), log(auspx1/auspc1) ]
[chn1]  log(m1chn) = f[ log(gdp), log(pm1lc/pgdp), log(chnpx1(-1)/chnpc1(-1)),
                      log(m1chn(-1)) ]
[hkg1]  log(m1hkg) = f[ log(gdp), pm1lc/pgdp, hkgpx1(-1)/hkgpc1(-1) ]
[idn1]  m1idn = f[ gdp, pm1lc, dlog(idnpx1/idnpc1) ]
[jpn1]  dlog(m1jpn) = f[ dlog(gdp), pm1lc(-1), jpnpx1(-1)/jpnpc1(-1) ]
[kor1]  dlog(m1kor) = f[ dlog(gdp), log(pm1lc(-1)),
                      log(korpx1(-1)/korpc1(-1)) ]
[mys1]  m1mys = f[ gdp, pm1lc(-1), log(myspx1/myspc1), m1mys(-1) ]
[nzl1]  (none)
[phl1]  d(m1phl) = f[ d(gdp), log(pm1lc(-1)/pgdp(-1)), dlog(phlpx1/phlpc1) ]
[sgp1]  dlog(m1sgp) = f[ dlog(gdp), log(pm1lc(-1)), sgppx1/sgppc1 ]
[tha1]  log(m1tha) = f[ log(gdp), log(pm1lc/pgdp), thapx1(-1)/thapc1(-1) ]
```

```
[twn1]  log(m1twn) = f[ log(gdp), log(pm1lc/pgdp),
                        dlog(twnpx1(-1)/twnpc1(-1)) ]
[usa1]  m1usa = f[ gdp, pm1lc, usapx1/usapc1, m1usa(-1) ]
[vnm1]  log(m1vnm) = f[ log(gdp), log(pm1lc), dlog(vnmpx1(-1)/vnmpc1(-1)) ]
[eur1]  dlog(m1eur) = f[ dlog(gdp), pm1lc, log(eurpx1(-1)/eurpc1(-1)) ]

[aus3]  d(m3aus) = f[ d(gdp), dlog(pm3lc(-1)/pgdp(-1)), auspx3/auspc3 ]
[chn3]  d(m3chn) = f[ d(gdp), log(pm3lc(-1)/pgdp(-1)),
                        log(chnpx3(-1)/chnpc3(-1)) ]
[hkg3]  m3hkg = f[ gdp, d(pm3lc(-1)), log(hkgpx3/hkgpc3) ]
[idn3]  log(m3idn) = f[ log(gdp), pm3lc(-1)/pgdp(-1), idnpx3(-1)/idnpc3(-1),
                        log(m3idn(-1)) ]
[jpn3]  m3jpn = f[ gdp, log(pm3lc/pgdp), log(jpnpx3/jpnpc3) ]
[kor3]  log(m3kor) = f[ log(gdp), pm3lc(-1)/pgdp(-1),
                        log(korpx3(-1)/korpc3(-1)) ]
[mys3]  m3mys = f[ gdp, pm3lc, log(myspx3/myspc3), m3mys(-1) ]
[nzl3]  (none)
[phl3]  d(m3phl) = f[ d(gdp), log(pm3lc/pgdp), log(phlpx3/phlpc3) ]
[sgp3]  m3sgp = f[ gdp, pm3lc(-1), d(sgppx3/sgppc3) ]
[tha3]  d(m3tha) = f[ d(gdp), log(pm3lc(-1)/pgdp(-1)),
                        log(thapx3(-1)/thapc3(-1)) ]
[twn3]  log(m3twn) = f[ log(gdp), pm3lc/pgdp, twnpx3/twnpc3 ]
[usa3]  m3usa = f[ gdp, log(pm3lc(-1)/pgdp(-1)), usapx3/usapc3, m3usa(-1) ]
[vnm3]  log(m3vnm) = f[ log(gdp), log(pm3lc(-1)), vnmpx3(-1)/vnmpc3(-1),
                        log(m3vnm(-1)) ]
[eur3]  d(m3eur) = f[ d(gdp), pm3lc, eurpx3(-1)/eurpc3(-1) ]
```

（9）フィリピン

```
[aus1]  log(m1aus) = f[ log(gdp), d(pm1lc/pgdp), auspx1(-1)/auspc1(-1),
                        log(m1aus(-1) ]
[chn1]  log(m1chn) = f[ log(gdp), pm1lc(-1), d(chnpx1(-1)/chnpc1(-1)),
                        log(m1chn(-1)) ]
[hkg1]  log(m1hkg) = f[ log(gdp), pm1lc(-1), log(hkgpx1/hkgpc1) ]
[idn1]  log(m1idn) = f[ log(gdp), log(pm1lc(-1)/pgdp(-1)), idnpx1/idnpc1 ]
[jpn1]  log(m1jpn) = f[ log(gdp), pm1lc, log(jpnpx1(-1)/jpnpc1(-1)) ]
```

```
[kor1]  log(m1kor) = f[ log(gdp), pm1lc/pgdp, korpx1/korpc1 ]
[mys1]  log(m1mys) = f[ log(gdp), log(pm1lc(-1)/pgdp(-1)), d(myspx1/myspc1) ]
[nzl1]  log(m1nzl) = f[ log(gdp), dlog(pm1lc), nzlpx1/nzlpc1, log(m1nzl(-1)) ]
[phl1]  (none)
[sgp1]  log(m1sgp) = f[ log(gdp), pm1lc(-1)/pgdp(-1), sgppx1(-1)/sgppc1(-1) ]
[tha1]  m1tha = f[ gdp, dlog(pm1lc/pgdp), d(thapx1(-1)/thapc1(-1)) ]
[twn1]  m1twn = f[ gdp, dlog(pm1lc/pgdp), d(twnpx1/twnpc1), m1twn(-1) ]
[usa1]  m1usa = f[ gdp, pm1lc, log(usapx1(-1)/usapc1(-1)), m1usa(-1) ]
[vnm1]  m1vnm = f[ gdp, log(pm1lc), dlog(vnmpx1/vnmpc1) ]
[eur1]  m1eur = f[ gdp, log(pm1lc/pgdp), log(eurpx1/eurpc1) ]

[aus3]  m3aus = f[ gdp, d(pm3lc(-1)/pgdp(-1)), auspx3(-1)/auspc3(-1),
                        m3aus(-1) ]
[chn3]  m3chn = f[ gdp, log(pm3lc), log(chnpx3/chnpc3) ]
[hkg3]  log(m3hkg) = f[ log(gdp), pm3lc(-1)/pgdp(-1),
                        log(hkgpx3(-1)/hkgpc3(-1)) ]
[idn3]  log(m3idn) = f[ log(gdp), log(pm3lc/pgdp), log(idnpx3/idnpc3) ]
[jpn3]  dlog(m3jpn) = f[ dlog(gdp), pm3lc, log(jpnpx3/jpnpc3) ]
[kor3]  log(m3kor) = f[ log(gdp), pm3lc(-1)/pgdp(-1),
                        log(korpx3(-1)/korpc3(-1)) ]
[mys3]  (none)
[nzl3]  log(m3nzl) = f[ log(gdp), pm3lc/pgdp, nzlpx3/nzlpc3 ]
[phl3]  (none)
[sgp3]  d(m3sgp) = f[ d(gdp), log(pm3lc(-1)/pgdp(-1)), dlog(sgppx3/sgppc3) ]
[tha3]  dlog(m3tha) = f[ dlog(gdp), pm3lc, thapx3/thapc3 ]
[twn3]  log(m3twn) = f[ log(gdp), d(pm3lc/pgdp), twnpx3/twnpc3, log(m3twn(-1)) ]
[usa3]  dlog(m3usa) = f[ dlog(gdp), pm3lc(-1), d(usapx3(-1)/usapc3(-1)) ]
[vnm3]  log(m3vnm) = f[ log(gdp), pm3lc(-1)/pgdp(-1), vnmpx3/vnmpc3 ]
[eur3]  d(m3eur) = f[ d(gdp), pm3lc, log(eurpx3/eurpc3) ]
```

（１０）シンガポール

```
[aus1]  log(m1aus) = f[ log(gdp), pm1lc(-1)/pgdp(-1), dlog(auspx1/auspc1) ]
[chn1]  log(m1chn) = f[ log(gdp), dlog(pm1lc(-1)/pgdp(-1)),
                        log(chnpx1(-1)/chnpc1(-1)) ]
[hkg1]  dlog(m1hkg) = f[ dlog(gdp), d(pm1lc), dlog(hkgpx1/hkgpc1) ]
```

```
[idn1]  d(m1idn) = f[ d(gdp), dlog(pm1lc/pgdp), d(idnpx1/idnpc1) ]
[jpn1]  m1jpn = f[ gdp, pm1lc, log(jpnpx1(-1)/jpnpc1(-1)) ]
[kor1]  log(m1kor) = f[ log(gdp), pm1lc(-1)/pgdp(-1), korpx1(-1)/korpc1(-1) ]
[mys1]  (none)
[nzl1]  log(m1nzl) = f[ log(gdp), dlog(pm1lc/pgdp), d(nzlpx1(-1)/nzlpc1(-1)),
                        log(m1nzl(-1)) ]
[phl1]  m1phl = f[ gdp, pm1lc/pgdp, phlpx1(-1)/phlpc1(-1) ]
[sgp1]  (none)
[tha1]  dlog(m1tha) = f[ dlog(gdp), dlog(pm1lc), d(thapx1/thapc1) ]
[twn1]  dlog(m1twn) = f[ dlog(gdp), d(pm1lc), twnpx1/twnpc1 ]
[usa1]  log(m1usa) = f[ log(gdp), pm1lc, log(usapx1(-1)/usapc1(-1)) ]
[vnm1]  log(m1vnm) = f[ log(gdp), pm1lc, log(vnmpx1/vnmpc1) ]
[eur1]  m1eur = f[ gdp, dlog(pm1lc), log(eurpx1/eurpc1), m1eur(-1) ]

[aus3]  m3aus = f[ gdp, dlog(pm3lc/pgdp), d(auspx3(-1)/auspc3(-1)), m3aus(-1) ]
[chn3]  d(m3chn) = f[ d(gdp), d(pm3lc), dlog(chnpx3/chnpc3), d(m3chn(-1)) ]
[hkg3]  log(m3hkg) = f[ log(gdp), pm3lc, log(hkgpx3/hkgpc3) ]
[idn3]  dlog(m3idn) = f[ dlog(gdp), d(pm3lc(-1)), dlog(idnpx3/idnpc3) ]
[jpn3]  dlog(m3jpn) = f[ dlog(gdp), dlog(pm3lc/pgdp), d(jpnpx3/jpnpc3) ]
[kor3]  log(m3kor) = f[ log(gdp), pm3lc/pgdp, log(korpx3/korpc3) ]
[mys3]  m3mys = f[ gdp, log(pm3lc(-1)), log(myspx3(-1)/myspc3(-1)), m3mys(-1) ]
[nzl3]  m3nzl = f[ gdp, dlog(pm3lc), log(nzlpx3/nzlpc3) ]
[phl3]  m3phl = f[ gdp, d(pm3lc(-1)/pgdp(-1)), phlpx3(-1)/phlpc3(-1) ]
[sgp3]  (none)
[tha3]  (none)
[twn3]  m3twn = f[ gdp, dlog(pm3lc(-1)), log(twnpx3/twnpc3) ]
[usa3]  dlog(m3usa) = f[ dlog(gdp), dlog(pm3lc), log(usapx3(-1)/usapc3(-1)) ]
[vnm3]  m3vnm = f[ gdp, log(pm3lc), d(vnmpx3/vnmpc3) ]
[eur3]  log(m3eur) = f[ log(gdp), pm3lc, log(eurpx3/eurpc3) ]
```

（１１）タイ

```
[aus1]  log(m1aus) = f[ log(gdp), d(pm1lc(-1)/pgdp(-1)), log(auspx1/auspc1),
                        log(m1aus(-1) ]
[chn1]  (none)
[hkg1]  m1hkg = f[ gdp, pm1lc, hkgpx1(-1)/hkgpc1(-1) ]
```

```
[idn1]  log(m1idn) = f[ log(gdp), pm1lc/pgdp, idnpx1/idnpc1, log(m1idn(-1)) ]
[jpn1]  log(m1jpn) = f[ log(gdp), pm1lc/pgdp, log(jpnpx1/jpnpc1) ]
[kor1]  m1kor = f[ gdp, log(pm1lc), korpx1/korpc1 ]
[mys1]  log(m1mys) = f[ log(gdp), log(pm1lc(-1)/pgdp(-1)), myspx1/myspc1 ]
[nzl1]  m1nzl = f[ gdp, log(pm1lc), log(nzlpx1/nzlpc1) ]
[phl1]  log(m1phl) = f[ log(gdp), dlog(pm1lc(-1)), dlog(phlpx1/phlpc1) ]
[sgp1]  log(m1sgp) = f[ log(gdp), log(pm1lc(-1)/pgdp(-1)), sgppx1/sgppc1 ]
[tha1]  (none)
[twn1]  m1twn = f[ gdp, pm1lc, d(twnpx1/twnpc1) ]
[usa1]  log(m1usa) = f[ log(gdp), pm1lc/pgdp, dlog(usapx1(-1)/usapc1(-1)) ]
[vnm1]  m1vnm = f[ gdp, dlog(pm1lc), dlog(vnmpx1/vnmpc1), m1vnm(-1) ]
[eur1]  m1eur = f[ gdp, log(pm1lc(-1)), eurpx1(-1)/eurpc1(-1) ]

[aus3]  log(m3aus) = f[ log(gdp), log(pm3lc(-1)/pgdp(-1)), log(auspx3/auspc3) ]
[chn3]  log(m3chn) = f[ log(gdp), dlog(pm3lc(-1)/pgdp(-1)),
                        dlog(chnpx3(-1)/chnpc3(-1)), log(m3chn(-1)) ]
[hkg3]  log(m3hkg) = f[ log(gdp), log(pm3lc(-1)/pgdp(-1)),
                        d(hkgpx3(-1)/hkgpc3(-1)), log(m3hkg(-1)) ]
[idn3]  log(m3idn) = f[ log(gdp), dlog(pm3lc(-1)/pgdp(-1)), idnpx3/idnpc3,
                        log(m3idn(-1)) ]
[jpn3]  log(m3jpn) = f[ log(gdp), log(pm3lc(-1)/pgdp(-1)),
                        log(jpnpx3(-1)/jpnpc3(-1)) ]
[kor3]  log(m3kor) = f[ log(gdp), pm3lc/pgdp, log(korpx3/korpc3) ]
[mys3]  log(m3mys) = f[ log(gdp), d(pm3lc(-1)/pgdp(-1)),
                        dlog(myspx3(-1)/myspc3(-1)), log(m3mys(-1)) ]
[nzl3]  m3nzl = f[ gdp, pm3lc(-1), nzlpx3(-1)/nzlpc3(-1) ]
[phl3]  log(m3phl) = f[ log(gdp), dlog(pm3lc), d(phlpx3/phlpc3),
                        log(m3phl(-1)) ]
[sgp3]  log(m3sgp) = f[ log(gdp), pm3lc(-1)/pgdp(-1), dlog(sgppx3/sgppc3),
                        log(m3sgp(-1)) ]
[tha3]  (none)
[twn3]  log(m3twn) = f[ log(gdp), log(pm3lc/pgdp), log(twnpx3/twnpc3) ]
[usa3]  log(m3usa) = f[ log(gdp), log(pm3lc(-1)/pgdp(-1)), dlog(usapx3/usapc3),
                        log(m3usa(-1)) ]
[vnm3]  (none)
```

```
[eur3]  log(m3eur) = f[ log(gdp), log(pm3lc(-1)), eurpx3(-1)/eurpc3(-1),
                        log(m3eur(-1)) ]
```

（１２）台湾

```
[aus1]  m1aus = f[ gdp, dlog(pm1lc), log(auspx1(-1)/auspc1(-1)) ]
[chn1]  dlog(m1chn) = f[ dlog(gdp), dlog(pm1lc/pgdp), chnpx1(-1)/chnpc1(-1) ]
[hkg1]  d(m1hkg) = f[ d(gdp), pm1lc(-1), dlog(hkgpx1(-1)/hkgpc1(-1)) ]
[idn1]  m1idn = f[ gdp, d(pm1lc(-1)/pgdp(-1)), d(idnpx1/idnpc1), m1idn(-1) ]
[jpn1]  d(m1jpn) = f[ d(gdp), d(pm1lc/pgdp), log(jpnpx1/jpnpc1), d(m1jpn(-1)) ]
[kor1]  dlog(m1kor) = f[ dlog(gdp), d(pm1lc), log(korpx1/korpc1),
                        dlog(m1kor(-1)) ]
[mys1]  d(m1mys) = f[ d(gdp), log(pm1lc), log(myspx1(-1)/myspc1(-1)) ]
[nzl1]  (none)
[phl1]  log(m1phl) = f[ log(gdp), pm1lc(-1), log(phlpx1(-1)/phlpc1(-1)),
                        log(m1phl(-1)) ]
[sgp1]  log(m1sgp) = f[ log(gdp), dlog(pm1lc(-1)/pgdp(-1)),
                        log(sgppx1(-1)/sgppc1(-1)), log(m1sgp(-1)) ]
[tha1]  log(m1tha) = f[ log(gdp), dlog(pm1lc(-1)/pgdp(-1)), thapx1/thapc1 ]
[twn1]  (none)
[usa1]  log(m1usa) = f[ log(gdp), log(pm1lc/pgdp), log(usapx1/usapc1) ]
[vnm1]  log(m1vnm) = f[ log(gdp), dlog(pm1lc/pgdp), vnmpx1(-1)/vnmpc1(-1) ]
[eur1]  d(m1eur) = f[ d(gdp), pm1lc, dlog(eurpx1/eurpc1) ]

[aus3]  d(m3aus) = f[ d(gdp), pm3lc(-1)/pgdp(-1), dlog(auspx3(-1)/auspc3(-1)) ]
[chn3]  dlog(m3chn) = f[ dlog(gdp), log(pm3lc), dlog(chnpx3/chnpc3) ]
[hkg3]  log(m3hkg) = f[ log(gdp), pm3lc(-1)/pgdp(-1), hkgpx3(-1)/hkgpc3(-1) ]
[idn3]  d(m3idn) = f[ d(gdp), d(pm3lc/pgdp), dlog(idnpx3/idnpc3) ]
[jpn3]  dlog(m3jpn) = f[ dlog(gdp), d(pm3lc(-1)), log(jpnpx3(-1)/jpnpc3(-1)) ]
[kor3]  dlog(m3kor) = f[ dlog(gdp), dlog(pm3lc(-1)), korpx3(-1)/korpc3(-1) ]
[mys3]  m3mys = f[ gdp, log(pm3lc(-1)/pgdp(-1)), myspx3(-1)/myspc3(-1),
                        m3mys(-1) ]
[nzl3]  m3nzl = f[ gdp, log(pm3lc), nzlpx3(-1)/nzlpc3(-1) ]
[phl3]  dlog(m3phl) = f[ dlog(gdp), d(pm3lc(-1)/pgdp(-1)),
                        dlog(phlpx3(-1)/phlpc3(-1)), dlog(m3phl(-1)) ]
[sgp3]  d(m3sgp) = f[ d(gdp), d(pm3lc/pgdp), log(sgppx3(-1)/sgppc3(-1)) ]
```

```
[tha3]  m3tha = f[ gdp, log(pm3lc(-1)/pgdp(-1)), d(thapx3/thapc3), m3tha(-1) ]
[twn3]  (none)
[usa3]  log(m3usa) = f[ log(gdp), d(pm3lc(-1)/pgdp(-1)), log(usapx3/usapc3) ]
[vnm3]  log(m3vnm) = f[ log(gdp), log(pm3lc/pgdp), vnmpx3(-1)/vnmpc3(-1) ]
[eur3]  m3eur = f[ gdp, pm3lc, d(eurpx3(-1)/eurpc3(-1)) ]
```

（１３）米国

```
[aus1]  dlog(m1aus) = f[ dlog(gdp), d(pm1lc), d(auspx1(-1)/auspc1(-1)) ]
[chn1]  log(m1chn) = f[ log(gdp), dlog(pm1lc/pgdp), log(chnpx1/chnpc1),
                        log(m1chn(-1)) ]
[hkg1]  dlog(m1hkg) = f[ dlog(gdp), dlog(pm1lc/pgdp), d(hkgpx1(-1)/hkgpc1(-1)),
                        dlog(m1hkg(-1)) ]
[idn1]  d(m1idn) = f[ d(gdp), log(pm1lc(-1)/pgdp(-1)), dlog(idnpx1/idnpc1) ]
[jpn1]  log(m1jpn) = f[ log(gdp), log(pm1lc), dlog(jpnpx1(-1)/jpnpc1(-1)),
                        log(m1jpn(-1)) ]
[kor1]  d(m1kor) = f[ d(gdp), d(pm1lc/pgdp), korpx1(-1)/korpc1(-1) ]
[mys1]  dlog(m1mys) = f[ dlog(gdp), d(pm1lc/pgdp), dlog(myspx1/myspc1) ]
[nzl1]  log(m1nzl) = f[ log(gdp), pm1lc(-1)/pgdp(-1), dlog(nzlpx1/nzlpc1),
                        log(m1nzl(-1)) ]
[phl1]  (none)
[sgp1]  dlog(m1sgp) = f[ dlog(gdp), d(pm1lc/pgdp), dlog(sgppx1/sgppc1) ]
[tha1]  log(m1tha) = f[ log(gdp), d(pm1lc/pgdp), dlog(thapx1/thapc1),
                        log(m1tha(-1)) ]
[twn1]  m1twn = f[ gdp, dlog(pm1lc), d(twnpx1(-1)/twnpc1(-1)), m1twn(-1) ]
[usa1]  (none)
[vnm1]  m1vnm = f[ gdp, dlog(pm1lc), log(vnmpx1/vnmpc1) ]
[eur1]  d(m1eur) = f[ d(gdp), log(pm1lc), eurpx1(-1)/eurpc1(-1) ]

[aus3]  dlog(m3aus) = f[ dlog(gdp), d(pm3lc), dlog(auspx3/auspc3) ]
[chn3]  log(m3chn) = f[ log(gdp), dlog(pm3lc), chnpx3/chnpc3, log(m3chn(-1)) ]
[hkg3]  dlog(m3hkg) = f[ dlog(gdp), pm3lc, dlog(hkgpx3(-1)/hkgpc3(-1)) ]
[idn3]  d(m3idn) = f[ d(gdp), d(pm3lc), idnpx3/idnpc3 ]
[jpn3]  dlog(m3jpn) = f[ dlog(gdp), log(pm3lc), d(jpnpx3(-1)/jpnpc3(-1)) ]
[kor3]  dlog(m3kor) = f[ dlog(gdp), log(pm3lc(-1)), d(korpx3(-1)/korpc3(-1)) ]
[mys3]  dlog(m3mys) = f[ dlog(gdp), pm3lc, myspx3/myspc3 ]
```

```
[nzl3]  m3nzl = f[ gdp, log(pm3lc), dlog(nzlpx3(-1)/nzlpc3(-1)), m3nzl(-1) ]
[phl3]  dlog(m3phl) = f[ dlog(gdp), d(pm3lc), log(phlpx3/phlpc3) ]
[sgp3]  (none)
[tha3]  d(m3tha) = f[ d(gdp), d(pm3lc), dlog(thapx3/thapc3) ]
[twn3]  dlog(m3twn) = f[ dlog(gdp), log(pm3lc), log(twnpx3/twnpc3) ]
[usa3]  (none)
[vnm3]  (none)
[eur3]  d(m3eur) = f[ d(gdp), log(pm3lc/pgdp), dlog(eurpx3(-1)/eurpc3(-1)) ]
```

（１４）ベトナム

```
[aus1]  m1aus = f[ gdp, dlog(pm1lc/pgdp), log(auspx1/auspc1) ]
[chn1]  m1chn = f[ gdp, log(pm1lc(-1)), chnpx1/chnpc1 ]
[hkg1]  m1hkg = f[ gdp, pm1lc, dlog(hkgpx1(-1)/hkgpc1(-1)) ]
[idn1]  (none)
[jpn1]  d(m1jpn) = f[ d(gdp), pm1lc(-1), dlog(jpnpx1(-1)/jpnpc1(-1)) ]
[kor1]  m1kor = f[ gdp, pm1lc/pgdp, d(korpx1/korpc1) ]
[mys1]  (none)
[nzl1]  d(m1nzl) = f[ d(gdp), pm1lc, nzlpx1(-1)/nzlpc1(-1) ]
[phl1]  d(m1phl) = f[ d(gdp), log(pm1lc(-1)), phlpx1/phlpc1 ]
[sgp1]  (none)
[tha1]  dlog(m1tha) = f[ dlog(gdp), dlog(pm1lc(-1)/pgdp(-1)),
                         thapx1(-1)/thapc1(-1) ]
[twn1]  log(m1twn) = f[ log(gdp), d(pm1lc(-1)/pgdp(-1)),
                         twnpx1(-1)/twnpc1(-1) ]
[usa1]  m1usa = f[ gdp, log(pm1lc/pgdp), dlog(usapx1/usapc1) ]
[vnm1]  (none)
[eur1]  log(m1eur) = f[ log(gdp), d(pm1lc(-1)/pgdp(-1)),
                         eurpx1(-1)/eurpc1(-1) ]

[aus3]  d(m3aus) = f[ d(gdp), pm3lc, auspx3(-1)/auspc3(-1) ]
[chn3]  m3chn = f[ gdp, log(pm3lc(-1)), log(chnpx3(-1)/chnpc3(-1)) ]
[hkg3]  m3hkg = f[ gdp, dlog(pm3lc(-1)), dlog(hkgpx3(-1)/hkgpc3(-1)) ]
[idn3]  dlog(m3idn) = f[ dlog(gdp), d(pm3lc(-1)), idnpx3(-1)/idnpc3(-1) ]
[jpn3]  (none)
[kor3]  m3kor = f[ gdp, d(pm3lc(-1)), log(korpx3(-1)/korpc3(-1)) ]
```

```
[mys3]  d(m3mys) = f[ d(gdp), pm3lc(-1), dlog(myspx3(-1)/myspc3(-1)) ]
[nzl3]  d(m3nzl) = f[ d(gdp), log(pm3lc), dlog(nzlpx3/nzlpc3) ]
[phl3]  dlog(m3phl) = f[ dlog(gdp), d(pm3lc(-1)), d(phlpx3(-1)/phlpc3(-1)) ]
[sgp3]  m3sgp = f[ gdp, d(pm3lc(-1)/pgdp(-1)), d(sgppx3/sgppc3) ]
[tha3]  m3tha = f[ gdp, pm3lc/pgdp, log(thapx3/thapc3) ]
[twn3]  d(m3twn) = f[ d(gdp), dlog(pm3lc(-1)/pgdp(-1)),
                      log(twnpx3(-1)/twnpc3(-1)) ]
[usa3]  (none)
[vnm3]  (none)
[eur3]  d(m3eur) = f[ d(gdp), log(pm3lc(-1)), d(eurpx3(-1)/eurpc3(-1)) ]
```

（１５）ユーロ地域

```
[aus1]  (none)
[chn1]  log(m1chn) = f[ log(gdp), dlog(pm1lc/pgdp),
                      dlog(chnpx1(-1)/chnpc1(-1)), log(m1chn(-1)) ]
[hkg1]  dlog(m1hkg) = f[ dlog(gdp), d(pm1lc(-1)/pgdp(-1)), log(hkgpx1/hkgpc1) ]
[idn1]  m1idn = f[ gdp, d(pm1lc(-1)/pgdp(-1)), dlog(idnpx1/idnpc1) ]
[jpn1]  dlog(m1jpn) = f[ dlog(gdp), d(pm1lc/pgdp), jpnpx1(-1)/jpnpc1(-1) ]
[kor1]  dlog(m1kor) = f[ dlog(gdp), d(pm1lc/pgdp), korpx1/korpc1 ]
[mys1]  (none)
[nzl1]  m1nzl = f[ gdp, d(pm1lc/pgdp), log(nzlpx1/nzlpc1) ]
[phl1]  (none)
[sgp1]  d(m1sgp) = f[ d(gdp), dlog(pm1lc), sgppx1/sgppc1 ]
[tha1]  m1tha = f[ gdp, d(pm1lc/pgdp), dlog(thapx1(-1)/thapc1(-1)), m1tha(-1) ]
[twn1]  m1twn = f[ gdp, dlog(pm1lc(-1)/pgdp(-1)), dlog(twnpx1(-1)/twnpc1(-1)) ]
[usa1]  m1usa = f[ gdp, dlog(pm1lc(-1)/pgdp(-1)), dlog(usapx1/usapc1),
                      m1usa(-1) ]
[vnm1]  log(m1vnm) = f[ log(gdp), log(pm1lc/pgdp), vnmpx1(-1)/vnmpc1(-1) ]
[eur1]  m1eur = f[ gdp, dlog(pm1lc/pgdp), eurpx1(-1)/eurpc1(-1) ]

[aus3]  log(m3aus) = f[ log(gdp), dlog(pm3lc), log(auspx3(-1)/auspc3(-1)) ]
[chn3]  log(m3chn) = f[ log(gdp), dlog(pm3lc/pgdp), d(chnpx3/chnpc3) ]
[hkg3]  (none)
[idn3]  m3idn = f[ gdp, d(pm3lc), idnpx3/idnpc3 ]
[jpn3]  d(m3jpn) = f[ d(gdp), d(pm3lc), jpnpx3(-1)/jpnpc3(-1) ]
```

```
[kor3]  (none)
[mys3]  m3mys = f[ gdp, dlog(pm3lc), log(myspx3/myspc3), m3mys(-1) ]
[nzl3]  m3nzl = f[ gdp, d(pm3lc/pgdp), d(nzlpx3(-1)/nzlpc3(-1)), m3nzl(-1) ]
[phl3]  d(m3phl) = f[ d(gdp), d(pm3lc/pgdp), d(phlpx3/phlpc3) ]
[sgp3]  (none)
[tha3]  log(m3tha) = f[ log(gdp), dlog(pm3lc), log(thapx3/thapc3) ]
[twn3]  log(m3twn) = f[ log(gdp), dlog(pm3lc), log(twnpx3/twnpc3) ]
[usa3]  log(m3usa) = f[ log(gdp), dlog(pm3lc(-1)), d(usapx3/usapc3) ]
[vnm3]  log(m3vnm) = f[ log(gdp), dlog(pm3lc/pgdp), dlog(vnmpx3/vnmpc3) ]
[eur3]  m3eur = f[ gdp, dlog(pm3lc/pgdp), eurpx3/eurpc3 ]
```

(以上)