
FRML 方程式名 変数名 = 算術式 ;
FRML 方程式名 算術式 ;

機能 :

FRML は TSP 方程式を定義します。これらの方程式は、後で推定やシミュレーションに対するプログラムで使えますし、また後で行う計算のために保存だけをすることもできます。ANALYZ, DIFFER, LSQ, FIML, SIML, ML, EQSUB, そして SOLVE プロシジャーでは、すべて FRML で定められた方程式がインプットとして必要です。GENR でも FRML 名を引数として使うことが出来ます。

使用法:

FRML の記述方法は2通りあります。先ず第1の方法では、方程式名を与え、続いて正規化された方程式、つまり従属変数名を等号の左辺に与え、右辺には当該変数の算術式を与えます。式の表現は、このマニュアルの基本ルールのセクションにある TSP の規則に従わなくてはなりません。この規則は TSP のプログラム中で方程式が使われる場合にはいつも守られなければなりません。つまり IF, GENR, SET, FRML, IDENT, SMPLIF, SELECT そして GOTO などです。DOT 付の変数は、FRML と IDENT では使えません。しかし他の記述法では利用できます (DOT コマンドの記述の項を参照して下さい)。

第2の方法は、陰関数形式の FRML 表現にします。この方法では等号はつかわれず単に代数式だけがおかれます。この方法は完全な連立方程式モデルで利用されますが、この場合は方程式を正規化することが多分出来ないでしょう。ANALYZ, FIML, そして SIML プロシジャーでは、陰関数形式の方程式表現を用いることが出来ます。もしいつもゼロ値を採る変数を作り、その変数を左辺においた FRML 表現を作れば、LSQ も陰関数の方程式を処理することが出来ます。

FRML で定義される方程式は IDENT で定義される方程式と同様ですが、推定プロシジャーでは FRML による方程式はその最後に誤差項がついていると考え、他方 IDENT による式には誤差項はないと仮定します。もし FRML で定義された式が正規化されていないければ、それは誤差項に等しいと考えられます。IDENT と FRML によって決められた式の違いは、FIML においてのみ意味があります。FIML では (平方行列にするために) 定義式がヤコビアン行列の計算に必要です。

FRML ステートメントで定義された方程式には数値、パラメータ、定数項、そして系列を含むことが出来ます。方程式の値は GENR によって任意の時点で計算することが出来ます (ステートメントの形については GENR を参照して下さい)。計算されるときは、パラメータと定数項にはその時点での値が与えられます。そして現在の標本期間内で系列のすべての値に対して方程式が計算されます。

例:

次の方程式は、例に使われたモデルを LSQ の三段階最小二乗法で推定するのに用いるプログラムです。

```
FRML CONSEQ CONS=A+B*GNP;
FRML INVEQ I=LAMBDA*I(-1)+ALPHA*GNP/(DELTA+R);
FRML INTRASTEQ R=D+F*(LOG(GNP)+LP-LM);
FRML PRICEQ LP=LP(-1)+PSI*(LP(-1)-LP(-2))+PHI*LOG(GNP)+TREND*TIME+PQ;
```

FRML

このモデルでは従属変数は CONS, I, R, と LP です。他の変数は, GNP, LM, と TIME です。方程式にもラグ付系列があります。その他 A, B, LAMBDA, ALPHA, DELTA, D, F, PSI, PHI, TREND, PO がありますが, これらはパラメーターと定数です。最初の FRML 式をその残差として正規化しない形で書くと

```
FRML CONSEQ CONS-(A+B*GNP);
```

となります。もし従属変数が式表現になっていると, 正規化しない方法を採用ねばなりません。次の例は誤りになります。

```
FRML EQNL LOG(Y)=A+B*X;
```

(FIML あるいは SIML に対しては) 陰関数型の FRML に書き直さなければなりません。

```
FRML EQNL LOG(Y)-(A+B*X);
```

もうすこし例をあげましょう。正規密度関数や累積正規分布関数を使う例に対しては, DIFFER のセクションも参照して下さい。

```
FRML ZERO A72-A73;  
FRML TRIGEQ COSX=COS(X);  
FRML TRIGEQ2 COSXY=X*Y/(X*X+Y*Y)**0.5;  
FRML RCONSTR RHO=(2/PI)*ATAN(PARAM);
```

DOT ループでいくつかの似たような FRML を定義する例については, DOT コマンドを参照して下さい。