DIFFER(DEPVARPR=新しい従属変数名, PRINT, PREFIX= 新方程式名)
方程式名 引数リスト ;

機能 :

DIFFERは, TSP 方程式を引数リストに関して解析的に微分し, それぞれの導関数を別の TSP 方程式として括弧をつけた形で保存します.

使用法:

DIFFERには, 微分しようとする方程式の名前に続けて微分に必要な 1 つ以上の引数が必要です. もし, いずれの引数も方程式にない場合は zero derivative が保存され, エラーとはなりません.

オプション:

DEPVARPR=新従属変数の前に付ける文字 デフォールトは D*lhsvar*. もし, もとの方程式に従属変数がなければ (正規化されていない), デフォールトは正規化されていない導関数を作成する.

PRINT/**NOPRINT**は, 結果としての微分方程式がプリントアウトされるかどうかを指定します.

PREFIX= は, 微分方程式に Deqname1 などとは違う名をつけるときに用います. 方程式の名前は接頭語に引数の名をつけて作ります.

例:

1. 既定オプションを用いて

```
FRML EQ Y = A + B*X + G*X**2;
DIFFER EQ A B X Q;
```

   次の FRML を作成します.

```
FRML DEQ1 DY1 = 1;            ? dY/dA
FRML DEQ2 DY2 = X;            ? dY/dB
FRML DEQ3 DY3 = B + 1*G*X;    ? dY/dX
FRML DEQ4 DY4 = 0;            ? dY/dQ
```

2. 結果に名前を付けるのに prefix オプションを用いて

```
FRML EQ Y = A + B*X + G*X**2;
DIFFER(PREFIX=GYX,DEPVAR=MPX) EQ X;
```

   次の FRML を作成します.

```
FRML GYX1 MPX1 = B + 2*G*X;   ? dY/dX
```

3. 方程式を非正規化すると (AR(1) モデルの残差)

```
FRML E Y - (A + B*X) - RHO*(Y(-1) - (A+B*X(-1));
DIFFER E B;
```

次の (非正規化された)FRML を作成します.

```
FRML DE1 -X + RHO*X(-1);
```

4. 代替の弾力性が一定の生産関数を 2 つの投入要素 K と L に関して微分すると, 導関数を用いて 2 つの限界生産系列を計算し, プリントします.

```
FRML CES Y = A*EXP(GAM*T)*(ALPHA*L**RHO+BETA*K**RHO)**(1.0/RHO);
DIFFER CES L K ;
GENR DCES1 LMP ; GENR DCES2 KMP ;
PRINT LMP KMP ;
```

5. PROBIT モデルの対数尤度関数をそのパラメータに関して微分し, その方程式を保存します.

```
FRML PROBIT LOGL = D*LOG(CNORM(A+B*X))+(1-D)*LOG(1-CNORM(A+B*X)) ;
DIFFER(PRINT,PREFIX=LOGL) PROBIT A B ;
```

その結果は, 次のようになります.

```
FRML LOGL1 DLOGL1 = [D*(NORM(A+B*X)+(1-D)*(-NORM(A+B*X))]/
      [D*CNORM(A+B*X)+(1-D)*(1-CNORM(A+B*X))] ;
FRML LOGL2 DLOGL2 = [D*(NORM(A+B*X)*X+(1-D)*(-NORM(A+B*X)*X)]/
      [D*CNORM(A+B*X)+(1-D)*(1-CNORM(A+B*X))] ;
```

この例は, 他のプログラムや Fortran のような言語で使えるように, 目的関数の解析的微分型を作成する上で DIFFER を使うことの便利さを示すものです.

アウトプット :

通常, DIFFER はプリントアウトはせず, 単に導関数を方程式としてデータ領域に保存します. もし PRINT オプションが指定されれば, DIFFER はそれぞれの方程式を, 何の微分かを示すタイトルを付けて記号の形でプリントします.