

CALL EXECUTEを用いたマクロの再帰呼び出しと 統計計算への応用

伊藤 要二

アストラゼネカ株式会社
臨床統計・プログラミング部

SUGI-J 2003

はじめに

- いくつかのプログラミング言語
 - 「サブルーチンの再帰呼び出し」の機能が利用可
 - 自分自身を呼び出すサブルーチンプログラムが可
 - 反復処理が非常に容易になることがある
- SASシステム
 - そのような機能は提供されていない
- 本発表では
 - 新しいテクニック「マクロの再帰呼び出し」を提案

発表内容

- CALL EXECUTEとは
- SASマクロの一般的な問題点
改善策: CALL EXECUTEによるマクロの呼び出し
- CALL EXECUTEによるマクロの再帰呼び出し
- 統計処理への応用事例:
MIXEDプロシジャを用いたPower-of-the-mean model

CALL EXECUTEとは

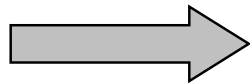
- CALL EXECUTEの使い方

```
data ...;  
    ...  
    call execute(' SASステートメント ');  
    ...  
run;
```

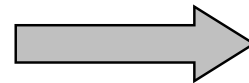
CALL EXECUTEのつまらない使用例

```
data a;  
  input x @@;  
  if _n_=1 then  
    call execute('proc sort; by x; proc print; run;');  
  cards;  
7 5 3  
;  
run;
```

実行を要求されたステートメント



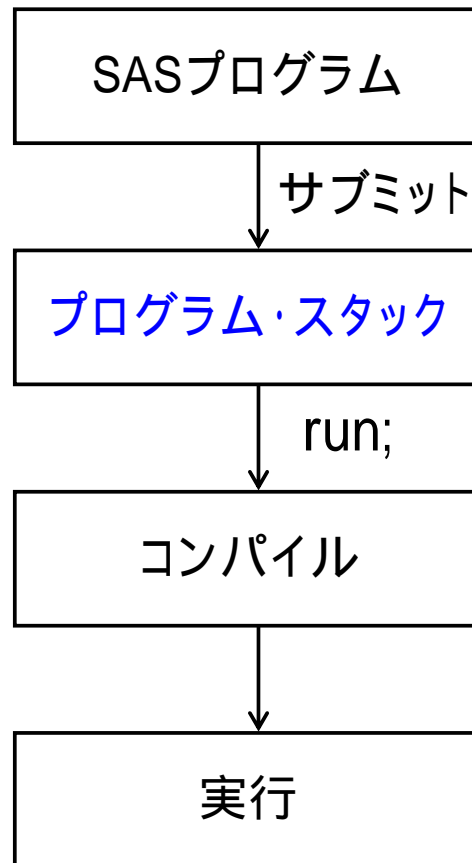
```
proc sort;  
  by x;  
proc print;  
run;
```



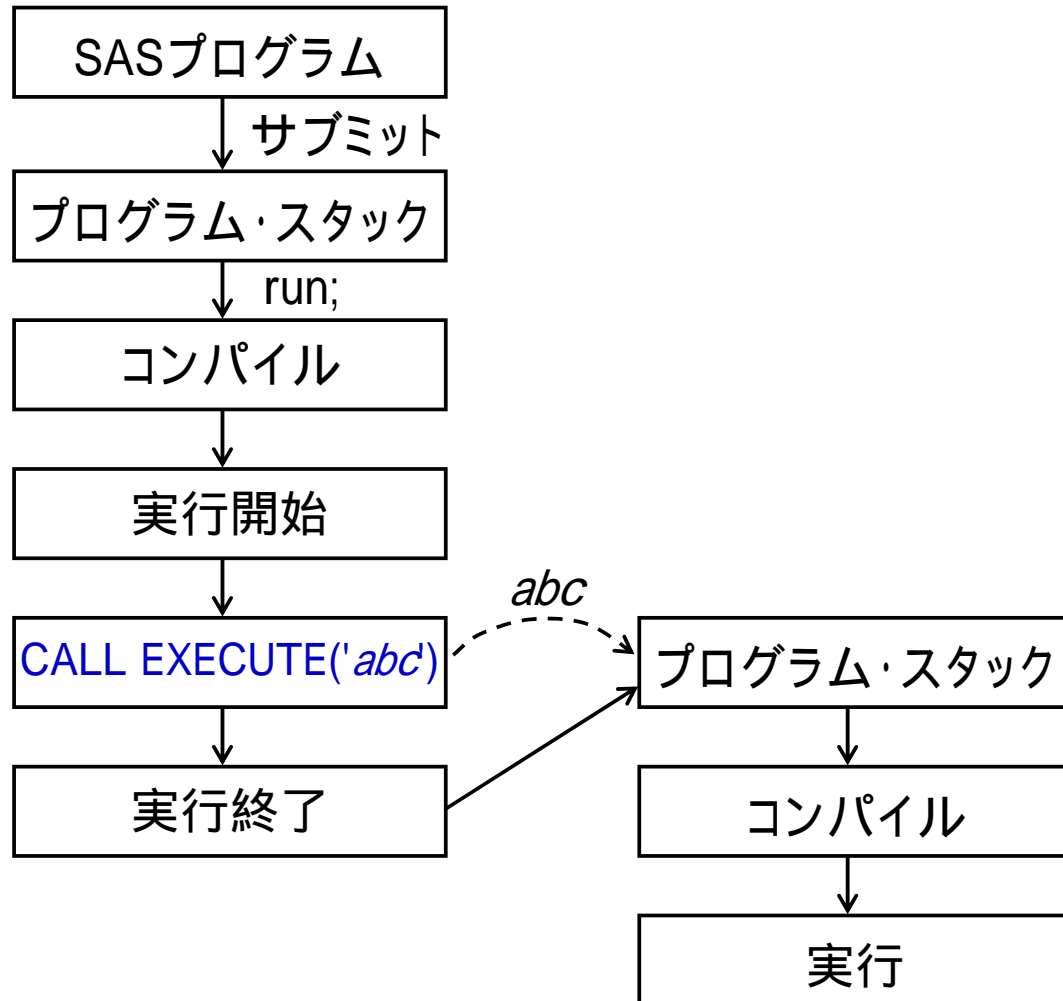
OUTPUT

Obs	x
1	3
2	5
3	7

通常のSASプログラムの処理の流れ



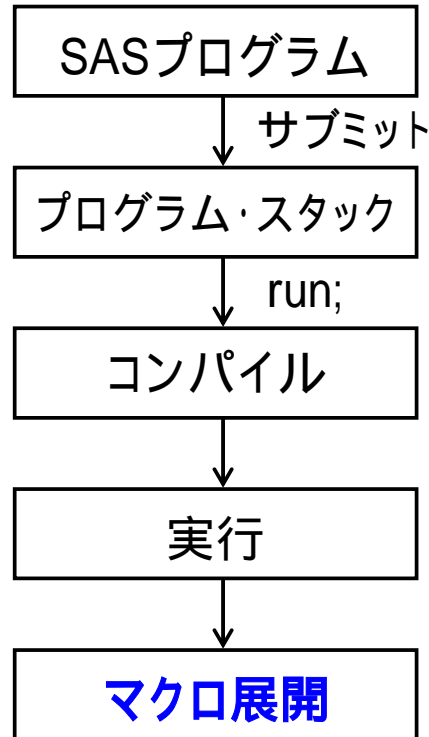
CALL EXECUTEを含むプログラムの処理の流れ



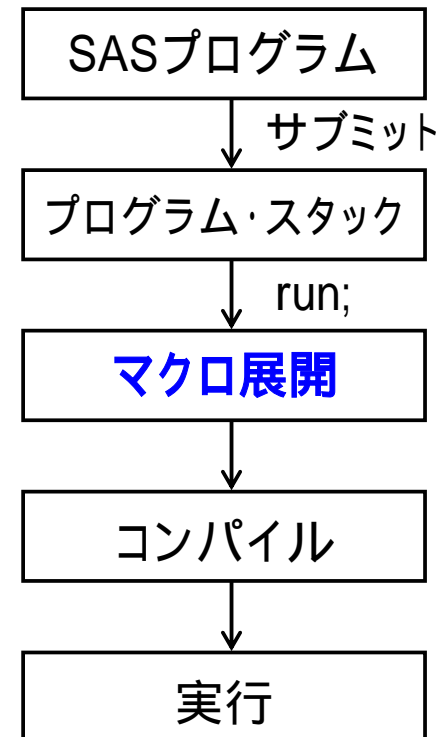
プログラムの実行結果によって、 マクロの展開をコントロールできるか？

```
data ...;  
...  
if (条件) then %macrox(a);  
    else %macrox(b);  
...  
run;
```

結論：プログラムの
実行結果によ
ってマクロ展開を
コントロールする
ことはできない



× 誤った解釈



正しい解釈

CALL EXECUTEによってマクロを呼び出す

- プログラムの実行結果によって、マクロの展開をコントロールすることが可能

```
data ...;  
...  
if (条件) then call execute( '%macrox(a);' );  
                else call execute( '%macrox(b);' );  
...  
run;
```

いずれかがプログラム・スタックに蓄えられ、データステップが終了するまで展開されない

CALL EXECUTEによるマクロの再帰呼び出し

```
%macro mcrx;
```

```
...
```

```
data ...;
```

```
...
```

```
if (条件) call execute('%mcrx;');
```

```
...
```

```
run;
```

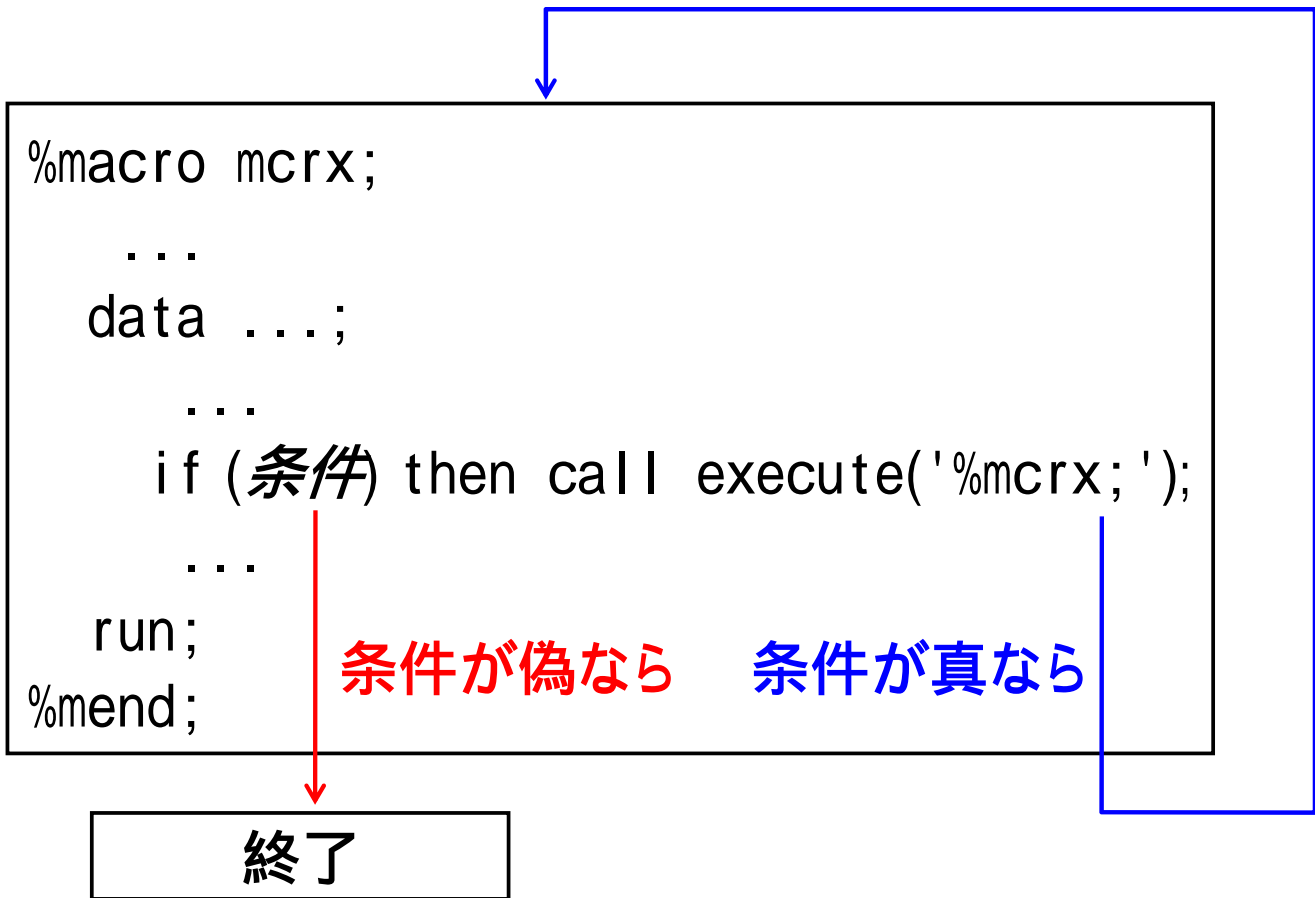
```
...
```

```
%mend;
```

自分自身を呼び出す



マクロの再帰呼び出しによる反復



Power-of-the-mean model

- **Power-of-the-mean model**

各観測値の誤差分散はその期待値のべき乗に比例。
線形モデルの場合、 i 番目の個体の誤差分散：

$$\sigma_{e_i}^2 = \sigma^2 | \mathbf{x}_i' \boldsymbol{\beta} |^\theta$$

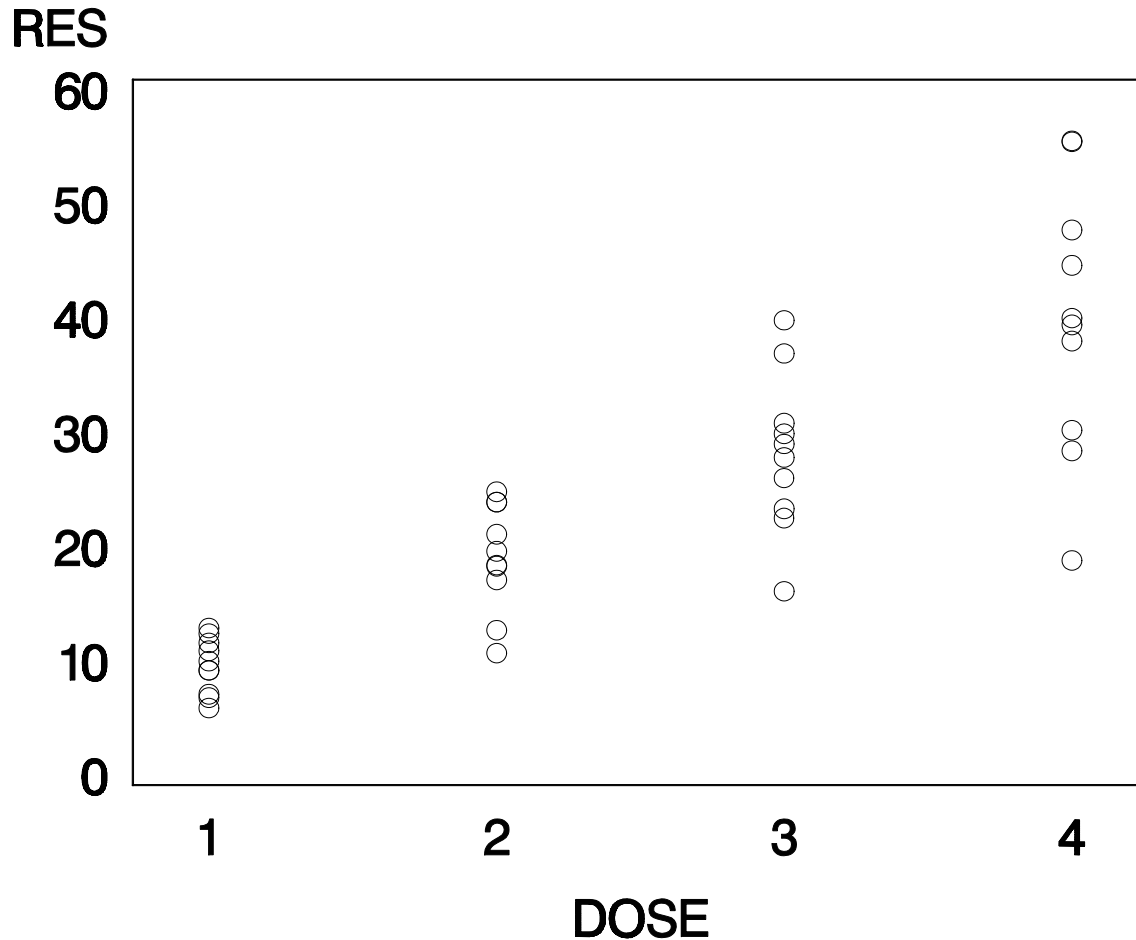
ただし、 σ^2 は未知の分散パラメタ、
 \mathbf{x}_i' はデザインマトリックス \mathbf{X} の i 番目の行、
 $\boldsymbol{\beta}$ は未知の固定効果のベクトル、
 θ は未知のべきパラメタ

- **SAS**
MIXEDプロシジャのREPEATEDステートメントにおいて：
REPEATED /LOCAL=POM(SASデータセット);


Power-of-the-mean model 数值例

```
data doseres;  
  input dose @;  
  do i=1 to 10;  
    input res @;  
    output;  
  end;  
  keep dose res;  
cards;  
  1   9.2  6.8 10.0 12.4  9.2 11.6 10.9  7.1 12.9  5.9  
  2  23.9 24.8 23.9 19.6 18.3 12.7 10.7 18.4 17.1 21.1  
  3  26.0 22.5 36.9 27.8 29.0 30.8 23.3 39.8 29.9 16.1  
  4  44.6 47.7 30.2 55.4 18.8 40.0 39.4 55.5 28.4 38.0  
;  
run;
```


Power-of-the-mean model 数値例の散布図



Power-of-the-mean model 解析プログラムの 基本的考え方

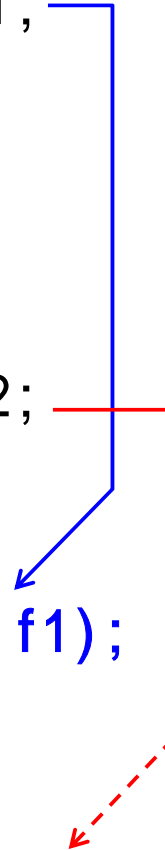


```
ods output solutionf=solf1;  
proc mixed data=doseres;  
  model res=dose / s;  
run;
```



```
ods output solutionf=solf2;  
proc mixed data=doseres;  
  model res=dose / s;  
  repeated / local=pom(solf1);  
run;
```

...



Power-of-the-mean model 解析プログラム

%let reg=1; 回帰係数初期値の設定(マクロ変数)

```
%macro pom(first);  
  ods output solutionf=solf2;  
  proc mixed data=doseres;  
    model res=dose / s;  
    %if &first^=1 %then repeated/ local=pom(solf1);;  
run;
```

2回目以降
repeatedを指定

```
  data solf1;  
    set solf2;  
    if effect='dose'  
      and abs(estimate - &reg)>1e-8 then do;  
        call symput('reg',left(put(estimate,e17.10)));  
        call execute('%pom()');  
      end;  
  run;  
%mend;
```

収束判定
マクロ変数更新
マクロの
再帰呼び出し

%pom(1); マクロの最初の呼び出し

Power-of-the-mean model 数値例の計算結果

反復	回帰係数	変化	べきパラメタ(θ)
1	9.9760000000	-	-
2	9.6919601675	-0.2840398325	2.0816386164
3	9.6898237032	-0.0021364643	2.1587574892
4	9.6898171411	-0.0000065621	2.1590014405
5	9.6898171209	-0.0000000202	2.1590021894
6	9.6898171209	-0.0000000000	2.1590021917

CALL EXECUTEを用いた反復の 一般的アルゴリズム

%let マクロ変数 = ...; マクロ変数に対する初期値の設定

%macro マクロ名;

...
data ...;

...
if (さらに反復すべきか?) then do;
 call symput('マクロ変数', 変数);
 call execute('マクロ名');

end;

run;

%mend;

% マクロ名;

マクロの最初の呼び出し

収束判定

マクロ変数を更新
マクロの再帰呼び出し

最後に (1)

- マクロの再帰呼び出しにより反復計算が可能
- 反復計算は
 - ・DATAステップDOループ
 - ・SAS/IMLのDOループによっても可能。
ただし、計算アルゴリズムの全てのプログラムコードを書く必要あり
- マクロの再帰呼び出しでは強力なSASの種々のプロシジャを反復して利用することが可能

最後に (2)

- 統計では種々の問題に対して反復計算がなされる。
例えば、
 - ・ 非線形モデルに対する推定、
 - ・ 不完全データに対するEMアルゴリズム、などなど
- 既存のSASのプロシジャを反復実行することにより計算が遂行可能な場合、本テクニックが非常に有用