プログラムのページ

担当 森口繁一

6212. 互換による順列の逐次発生

米田信夫（学習院大理学部）

順列の次数として integer N, 順列の内容として integer array P[1 : N], 補助的に mod N! の階乗進数を表示するものとして integer array F[1 : N−1] が non-local に宣言され，P には任意の初期データ順列 P_0, F には 0 がはいっているものとする．このプログラムは 1 度呼び出されるごとに P に適当な互換をほどこして P_{n-1} から P_n に変え ($n=1, 2, \cdots, N!-1$), 結局 $P_0, P_1, \cdots\cdots, P_{N!-1}$ が P_0 の内容の順列を重複なく全部つくすようにしたものである．N! 回目に呼び出されたときは，F は 0 にもどるが P は $P_{N!-1}$ のままで，label parameter Exh で指定された所に行く．

〔注〕 たとえば N=5 の場合は，次のようになる．

P_0	a b c d e				
		(12)			
P_1	b a c d e				
		(13)			
P_2	c a b d e				
		(12)	(i)		
P_3	a c b d e				
		(13)			
P_4	b c a d e				
		(12)			
P_5	c b a d e				
		(14)			
P_6	d b a c e				
·	· · ·	·	(i)	(ii)	
P_{11}	a b d c e	·			
		(24)			
P_{12}	a c d b e	·			
·	· · ·	·	(i)		
P_{17}	d c a b e	·			
		(34)			(iii)
P_{18}	d c b a e	·			
·	· · ·	·	(i)		
P_{23}	b c d a e	·			
		(15)			
P_{24}	e c d a b	·			
·	· · ·	·		(ii)	
P_{47}	c d a e b	·			
		(15)			
P_{48}	b d a e c	·			
·	· · ·	·		(ii)	
P_{71}	d a e b c	·			
		(15)			
P_{72}	c a e b d	·			
·	· · ·	·		(ii)	
P_{95}	a e b c d	·			
		(15)			
P_{96}	d e b c a	·			
·	· · ·	·		(ii)	
P_{119}	e b c d a	·			

($P_{120}=P_{119}$ で Exh に行く.)

```
procedure NEXTPERM(Exh); label Exh;
  begin integer k, l, E; Boolean keven;
    keven:=false;
    for k:=1 step 1 until N−1 do
      begin if F[k]<k then go to ŌP;
            F[k]:=0; keven:=¬keven
      end;
    go to Exh;
  ŌP: l:=F[k]:=F[k]+1; k:=k+1;
       if keven then l:=1;
       E:=P[l]; P[l]:=P[k]; P[k]:=E
  end NEXTPERM
```

6213. 曲線のあてはめ

矢島敬二（日科技研 計算センター）

変数 $x_1, x_2, \cdots\cdots, x_n$ に対する観測値 $y_1, y_2, \cdots\cdots, y_n$ が与えられたとき，x に関する m 次の多項式を最小二乗法によってあてはめる．このとき，変数 x_i の間隔は任意で，直交多項式をつくってあてはめてゆく．計算は精度の点から，原点を x の平均値のところに移し，$t_i=x_i-\bar{x}$ として，t についての多項式をあてはめる．直交多項式は次のようにつくる．一般に $\phi_r(t)$ で r 次の直交多項式をあらわすと，

$$\phi_{r+1}(t)=(t-\alpha_r)\phi_r(t)-\beta_r\phi_{r-1}(t)$$

ここで

$$\alpha_r=\frac{\sum t_i\phi_r^2(t_i)}{S_r}$$

$$\beta_r=\frac{S_r}{S_{r-1}}$$

$$S_r=\sum_{i=1}^{n}\phi_r^2(t_i)$$

である．また

$$\phi_0(t)=1$$

$$\phi_1(t)=t$$

したがって α_1, β_1 を求めて $\phi_2(t)$ をつくることから始めればよい．いま

$$f_m(t)\equiv\sum_{r=0}^{m}\lambda_r\phi_r(t)$$

として

$$\mathrm{SE}\equiv\sum_{i=1}^{n}(y_i-f_m(t_i))^2$$

を最小にする λ_r は次式で定まる．

$$\lambda_r = \frac{\sum y_i \phi_r(t_i)}{S_r}$$

これらの λ に対して SE の値 R_m は次式で求まる.

$$R_m = \sum_{i=0}^{n} y_i^2 - \sum_{j=0}^{m} \lambda_j^2 S_j$$

計算の途中では, 次のようなタイプを行なわせる.

$\bar{x}$ λ_0 R_0

λ_1 R_1

α_1 β_1 λ_2 R_2

…

α_{m-1} β_{m-1} λ_m R_m

さらに, 得られた $f_m(t)$ を

$$f_m(t) = \sum_{i=0}^{m} c_i x^i$$

として c_i を計算し, 最後に y_i, $f(t_i)$, $y_i - f(t_i)$ を印刷する.

```
CURVE FITTING:
begin integer I, K, M, N; real MEANX, MEANY,
  SSX, SSY, S1, S2, RESIDU, TEMP; real array
  X, Y, F1, F2, F3 [1 : 100], ALPHA, BETA[1:
  19], PHI 1[0 : 21], PHI 2, LAMBDA [1 : 20], F
  [1 : 21];
  READ(M); READ(N);
  for I:=1 step 1 until N do READ(X[I]);
  for I:=1 step 1 until N do READ(Y[I]);
  MEANX:=MEANY:=0;
  for I:=1 step 1 until N do
  begin MEANX:=MEANX+X[I];
        MEANY:=MEANY+Y[I]
  end;
  MEANX:=MEANX/FLOAT(N);
  MEANY:=MEANY/FLOAT(N);
  SSX:=SSY:=0;
  for I:=1 step 1 until N do
  begin X[I]:=X[I]-MEANX;
    SSX:=SSX+X[I]↑2;
    SSY:=SSY+(Y[I]-MEANY)↑2
  end;
  CRLF; CRLF; PRINT(MEANX);
    PRINT(ME-ANY);
  PRINT(SSY);
  for I:=1 step 1 until M-1 do
  begin ALPHA[I]:=0; LAMBDA[I]:=0
  end;
  LAMBDA[M]:=0;
for I:=1 step 1 until N do LAMBDA[1]:=
LAMBDA[1]+Y[I]×X[I];
LAMBDA[1]:=LAMBDA[1]/SSX; RESID:=
SSY-LAMBDA[1]↑2×SSX; CRLF; PRINT
(LAMBDA[1]); PRINT(RESID); S1:=SSX;
S2:=FLOAT(N);
for I:=1 step 1 until N do
begin F1[I]:=X[I]; F2[I]:=1.0; E3[I]:=
F1[I]↑2
end;
for K:=2 step 1 until M do
begin for I:=1 step 1 until N do ALPHA[K
  -1]:=ALPHA[K-1]+X[I]×F3[I];
ALPHA[K-1]:=ALPHA[K-1]/S1; BETA[K
  -1]:=S1/S2; S2:=S1; S1:=0;
for I:=1 step 1 until N do
  begin TEMP:=F2[I]; F2[I]:=F1[I]; F1[I]:
  =(X[I]-ALPHA[K-1])×F2[I]-BETA[K
  -1]×TEMP; F3[I]:=F1[I]↑2; S1:=S1+
  F3[I]; LAMBDA[K]:=LAMBDA[K]+Y[I]
  ×F1[I]
  end;
  LAMBDA[K]:=LAMBDA[K]/S1; RESIDU:
  =RESIDU-LAMBDA[K]↑2×S1; CRLF; PR-
  INT(A[K-1]); PRINT(B[K-1]); PRINT
  (LAMBDA[K]); PRINT(RESIDU)
end;
PHI2[1]:=1.0; PHI 2[2]:=PHI 1[0]:=PHI 1
[1]:=0; PHI1[2]:=1.0; F[1]:=MEANY;
  F[2]:=LAMBDA[1];
for K:=2 step 1 until M do
begin F[K+1]:=0;
  for I:=1 step 1 until K+1 do
  begin TEMP:=PHI2[I]; PHI2[I]:=PHI1
    [I]; PHI1[I]:=PHI1[I-1]-ALPHA[K-
    1]×PHI1[I]-BETA[K-1]×TEMP; F[I]:
    =F[I]+LAMBDA[K]×PHI1[I]
  end;
  PHI2[K+1]:=0
end;
for I:=1 step 1 until M+1 do
begin CRLF; PRINT(PHI1[I])
end;
CRLF;
```

```
  for I:=1 step 1 until N do
    begin CRLF; PRINT(Y[I]); PRINT(F 1[I]);
      TEMP:=Y[I]-F 1[I]; PRINT(TEMP)
    end
end
```

6214. 2^k 型要因実験の分散分析

矢島敬二（日科技研 計算センター）

2^k 型要因実験の データから， 自由度１に分解した効果，平方和を計算する．この際，手計算で使われているイェーツの算法を用いる．

2^k 個の実験データは 整数形にして 読み込むものとする．

```
FACTORIAL DESIGN:
begin integer I, J, K, M, N, W 1, W 2;
      real FN, W;
      integer array Y[1 : 256], SY[1 : 128];
  READ(K);
  N:=2↑K; FN:=FLOAT(N);
  M:=N div 2;
  for I:=1 step 1 until N do READ(Y[I]);
  for J:=1 step 1 until K do
    begin for I:=M step -1 until 1 do
      begin W 1:=Y[2*I]; W 2:=Y[2*I-1];
            SY[I]:=W 1+W2;
            Y[M+I]:=W 1-W 2
      end;
      for I:=1 step 1 until M do Y[I]:=SY[I]
    end;
  for I:=1 step 1 until N do
    begin W:=FLOAT(Y[I]);
      CRLF; PRINT(I);
      PRINT(W/FN);
      PRINT(W*W/FN)
    end
end
```