APPENDIX TWO

THE F-G ALGORITHM

What follows is SAS code for Flury's & Gautschi's F-G algorithm, with adaptation to get leasts squares estimates, applied to Fisher’s iris data.

```
proc iml;

*C1, C2 and C3 contain the sample covariance matrices for the three species
 of iris.;

C1={26.6433 8.5184 18.2898 5.5780,
     8.5184 9.8469  8.2653 4.1204,
    18.2898 8.2653 22.0816 7.3102,
     5.5780 4.1204  7.3102 3.9106};

C2={40.4343  9.3763 30.3290 4.9094,
     9.3763 10.4004  7.1380 4.7629,
    30.3290  7.1380 30.4588 4.8824,
     4.9094  4.7629  4.8824 7.5433};

C3={12.4249  9.9216  1.6355 1.0331,
     9.9216 14.3690  1.1698 0.9298,
     1.6355  1.1698  3.0159 0.6069,
     1.0331  0.9298  0.6069 1.1106};



*The F-G algorithm consists of the macro Gstep nested within the macro
 Fstep.  In the Fstep every pair of column vectors of the current
 approximation to B is rotated such that the normal equations are satisfied.
 The Gstep determines these rotations;

*P is the number of variables, K the number of datasets or groups and L is
 the number of iterations.  B is the matrix of common principal components.

%macro Gstep;  M=j(2,2,0);
  %do i=1 %to &K;  H=B[,{&d &e}]; T&i=H`*C&i*H;
    d1&i=Q[,1]`*T&i*Q[,1];   d2&i=Q[,2]`*T&i*Q[,2];
```

```
    *The line of code immediately below is for the least squares solution.
     The line of code following it is for the maximum likelihood algorithm.;

    *M=M + (d1&i-d2&i)*T&i;
    M=M + (d1&i-d2&i)/(d1&i*d2&i)*T&i;
  %end;
%mend;

*The Fstep follows.  Bold stores the matrix B from the previous iteration.;

%macro Fstep;  %let P=4;   %let K=3;   %let L=15;  B=I(&P); Bold=I(&P);
  %do s=1 %to &L;
    %do d=1 %to &P;
      %do e=1 %to &d;    Q=I(2);
        if &e<&d then do;  c=0;
          do until(c=4);
            %Gstep
           normala=Q[,1]`*M*Q[,2];
           Q=eigvec(M);
            B[,{&d &e}]=H*Q;
            c=c+1;
          end;
        If &s=&L then do;
          normal=B[,&d]`*B[,&e];  print &d &e normal;
        end;
        end;
      %end;
    %end;
    Crit=SSQ(B-Bold); print B bold;Bold=B;
    phi=1; SSLF=0;
    %do i=1 %to &K;
      R&i=B`*C&i*B;
      phi=phi*det(diag(R&i))/det(R&i);
      SSLF=SSLF+SSQ(R&i-diag(R&i));
    %end;    printcrit phi sslf;
  %end;
%mend; %Fstep

*The following macro determines theeigenvalues;

%macro eigen;   %let P=4;   %let K=3;   %let L=15;
%do i=1 %to &K;
  eig&i=diag(B`*c&i*B)*j(&P,1,1);
  print eig&i;
%end;
%mend;
%eigen
```