

```

*=====
=====;
***** ANIMAL BREEDING NOTES *****
***** CHAPTER 21 MTM *****
***** MULTIPLE TRAIT MODELS (SIRE MATERNAL GRANDSIRE MODEL) *****
***** Mauricio A. Elzo
***** University of Florida
***** Gainesville, FL 32611-0910
***** Tel: 1-352-392-7564
***** Email: maelzo@ufl.edu
*=====
=====;
dm 'clear log; clear output;';
ods output clear;
*=====
=====;
*=====
=====;
libname libiml 'C:\home\pkg\SAS\IML\ANS6386\2010';
*=====
=====;
*options nodate nocenter ls=150 ps=32767;
options date nocenter ls=150 ps=32767;

*=====
=====;
*** To print the list of GDEVICES used by PROC GPLOT ***;
*=====
=====;
/*proc gdevice catalog=sashelp.devices nofs; list; run; quit;*/
*=====
=====;
ods trace on / label;
ods graphics on;

goptions reset=all
          cback=white noborder
          colors=(black blue green red)
          ftitle=swissb ftext=swissb htitle=6 htext=3; /* ctext=red
ctitle=red;*/
          *device=gif;

*=====
=====;
*=====
=====;
*** Write date as Month day, year ***;
*=====
=====;
%macro fdate(fmt);
  %global fdate;
  data _null_;
    call symput("fdate",left(put("&sysdate9"d,&fmt)));
  run;

```

```

%mend fdate;
*=====
=====;
%fdate(worddate.); *** Get today's date ***;
*=====
=====;
*=====
=====;
%let runname=UABM_21_MTSMM_Example_February-20-2010_a &fdate; ** Change
once; *Use throughout program **;
title1 &runname;
*=====
=====;
*=====
=====;

*=====
=====;
*** Create html files in the directory for outputs ***;
*=====
=====;
ods listing close;
ods html
style=default          /** [default, d3d, minimal] name the format style of
the output **/

/** PATH FOR WORK HTML FILES **/
path="C:\home\pkg\SAS\IML\ANS6386\2010\Outputs\WORK" (url=none) /**
location of WORK html files **/
gpath="C:\home\pkg\SAS\IML\ANS6386\2010\Outputs\WORK"          /**
location of WORK graph files **/

/** NAMES OF html FILES **/
body("&runname._body.html"          /** name of body file **/
contents("&runname._contents.html" /** name of contents file **/
page("&runname._page.html"         /** name of page file **/
frame("&runname._frame.html";      /** name of frame file **/
*=====
=====;
*=====
=====;
***** NO INPUT FILE *****;
*=====
=====;
*=====
=====;
*=====
=====;
*%macro solvemme;
proc iml;

start solve;
print 'ANIMAL BREEDING NOTES';
print 'CHAPTER 21 MTM';

```

```

print 'MULTIPLE TRAIT MODELS (SIRE MATERNAL GRANDSIRE MODEL)';
print 'Mauricio A. Elzo, University of Florida, maelzo@ufl.edu';

*=====
=====;
print 'Enter Parameters for Current Run';
print 'Enter restronsol = 1 to impose restrictions on solutions to solve
the MME, else = 0 if not';
restronsol=0; print restronsol;
if restronsol > 0 then print 'Restrictions imposed on solutions to solve
the mixed model equations (MME)';
else print 'No restrictions imposed on solutions to solve MME';

*=====
=====;
print 'Dataset = Dataset used in Chapters 16 and 17 with birth weights
and weaning weights';
print 'Nonparents only (animals 11 to 15)';
*=====
=====;
print 'Enter nt = Number of traits';
nt=2; print nt;
print 'Enter ncafrec = Number of calves with records';
ncafrec=5; print ncafrec;
print 'Enter nrec = Number of records';
nrec=ncafrec*nt; print nrec;
print 'Enter nanim = Number of animals (4 sires and maternal
grandsires)';
nanim=4; print nanim;
print 'Enter nf = Number of fixed effects in the MME';
nf=2*nt; print nf;
print 'Enter nga = Number of random additive direct and maternal genetic
effects in the MME';
nga=nanim*nt; print nga;
print 'Compute neq = f+nga+ngg = total number of MME';
neq=nf+nga; print neq;

*=====
=====;
*=====
=====;

print 'Enter X = matrix of fixed sex effects (animals within traits)';
X={1 0 0 0,
    1 0 0 0,
    0 1 0 0,
    1 0 0 0,
    0 1 0 0,
    0 0 1 0,
    0 0 1 0,
    0 0 0 1,
    0 0 1 0,
    0 0 0 1};
print X;

```

```

print 'Enter Z = matrix of direct additive genetic effects (animals
within traits)';
Z={0 1.5 0 0 0 0 0 0,
  0 0.5 1 0 0 0 0 0,
  0 0 1.5 0 0 0 0 0,
  0 0 0 1.5 0 0 0 0,
  0 0 0 1.5 0 0 0 0,
  0 0 0 0 0 1.5 0 0,
  0 0 0 0 0 0.5 1 0,
  0 0 0 0 0 0 1.5 0,
  0 0 0 0 0 0 0 1.5,
  0 0 0 0 0 0 0 1.5};
print Z;

print 'Construct xf = [X Z]';
xf=j(nrec,neq,0);
do i=1 to nrec;
  do j=1 to neq;
    if j <= nf then xf[i,j]=x[i,j];
    else if j > nf & j <= nf+nga then xf[i,j]=z[i,j-nf];
  end;
end;
print xf;

print 'Enter y = vector of records (weaning weights)';
y={38,
  36,
  34,
  39,
  40,
  289,
  285,
  265,
  290,
  288};
print y;

*=====
=====;
print 'Computation of G inverse for nt traits';
print 'Computation of Ginv = Ainv@inv(g0*0.25) -> Method 1';
print 'Enter G0 = additive gene covariance matrix';
g0={2 3, 3 22}; print g0;

print 'Enter Tinv = (I - 1/2Psire -1/4Pmgs) for sires and mgs';
Tinv={ 1 0 0 0,
  -0.75 1 0 0,
  -0.5 0 1 0,
  -0.5 0 0 1};
print Tinv;

print 'Enter dvecsmgs = vector with elements of D for sires and mgs';
dvecsmgs={1, 0.6875,0.75,0.75}; print dvecsmgs;

```

```

print 'Construct Dsmgs = diag D matrix for sires and mgs';
Dsmgs=diag(dvecsmgs); print Dsmgs;

print 'Compute Dsmgsinv = inv(Dsmgs)';
Dsmgsinv=inv(Dsmgs); print Dsmgsinv;

print 'Compute Ainv = transpose(Tinv)*Dsmgsinv*Tinv';
Ainv=(t(Tinv)*Dsmgsinv)*Tinv; print Ainv;

print 'Compute Ginv = inv(g0*0.25)@Ainv for sires and mgs (@ = direct
product)';
Ginv=inv(g0*0.25)@Ainv; print Ginv;

print 'Computation of GINV2 = inv(A@g0*0.25) -> Method 2 for comparison
purposes only';
print 'Enter A = relationship matrix among sires and mgs';
A={1 0.75 0.5 0.5,
 0.75 1.25 0.375 0.375,
 0.5 0.375 1 0.25,
 0.5 0.375 0.25 1};
print A;

print 'Compute G = A@g0*0.25 for sires and mgs (@ = direct product)';
G=A@g0*0.25; print G;

print 'Compute GINV = inv(G)';
GINV2=inv(G); print GINV2;

*=====
=====;
print 'Computation of R inverse for nt traits';

print 'Enter ve = residual covariance matrix';
r0={8 8, 8 88}; print r0;

print 'Enter coefficients of diagonal elements of matrix DN for
nonparents (i.e., NOT multiplied by g0)';
dnvec={.609375,.671875,.6875,.6875,.6875};
print dnvec;

print 'Compute block-diagonal residual genetic matrix DN = g0@diag(dnvec)
for nonparents (@ = direct product)';
DN=g0@diag(dnvec); print DN;

print 'Compute block-diagonal environmental matrix EN = r0@i(ncafrec) for
nonparents (@ = direct product)';
EN=r0@i(ncafrec); print EN;

print 'Construct R = (DN + EN) = residual covariance matrix for
nonparents';
R=DN + EN; print R;

print 'Compute invr = inverse of R';
invr=inv(R); print invr;

```

```

*=====
=====;
print 'Compute lhs = left hand side of the MME';
print 'Compute xft = transpose of xf';
xft=t(xf); print xft;

print 'Compute xftinvr = xf transpose times invr';
xftinvr=xmult(xft,invr); print xftinvr;

print 'Compute xtinvrxf = xf transpose times invr times xf';
xftinvrxf=xmult(xftinvr,xf); print xftinvrxf;

print 'Add GINV to the appropriate submatrices of xftinvrxf';
lhs=xftinvrxf;
print xftinvrxf ginv ;
do i=nf+1 to neq;
  do j=nf+1 to neq;
    if i <= nf+nga & j <= nf+nga then lhs[i,j]=lhs[i,j]+ginv[i-nf,j-
nf]; *Add ginv elements to lhs;
  end;
end;
print lhs;
print lhs [format=6.3];

*=====
=====;
print 'Compute rhs = xftinvr*y';
rhs=xftinvr*y;
print rhs;
print rhs [format=6.2];

*=====
=====;
*=====
=====;
if restronsol > 0 then do;
  print 'Impose restrictions on solutions';
  print 'Set solution for mean of each trait to zero';
  do i=1 to neq;
    if i=1 then do; *Set solutions for mean of each trait to
zero;
      rhs[i]=0;
      do j=1 to neq;
        lhs[i,j]=0;
        lhs[j,i]=0;
      end;
    end;
  end;
  print 'lhs after restrictions';
  print lhs [format=6.3];
end;

```

```

*=====
=====;
*=====
=====;
print 'Compute ginvlhs = generalized inverse of the left hand side of the
MME';
ginvlhs=ginv(lhs);
print ginvlhs [format=9.6];
print ginvlhs [format=6.3];

print 'Compute gl = ginvlhs*lhs = matrix of expectations of solutions';
gl=ginvlhs*lhs;
print gl [format=6.3];

print 'Notice that lg = gl (i.e., lhs*ginvlhs = lhs*ginvlhs)';
lg=lhs*ginvlhs;
print lg [format=6.3];

print 'Verify that lgl = lhs (i.e., lhs*ginvlhs*lhs = lhs => generalized
inverse is correct)';
lgl=lhs*ginvlhs*lhs;
print lgl [format=6.3];

print 'Compute ranklhs = rank of the MME = trace of ginvlhs*lhs';
ranklhs=round(trace(gl));
print ranklhs;

print 'Compute sol = vector of solutions for the MME (Transmitting
Abilities = 1/2 BV)';
sol=ginvlhs*rhs;
print sol;
print sol [format=6.2];

print 'Compute sesol = standard error of solutions';
sesol=j(neq,1,0);
do i=1 to neq;
    if lhs[i,i] > 0 then do;
        sesol[i]=sqrt(ginvlhs[i,i]);
    end;
end;
print sesol [format=6.2];

print 'Check that sire solutions sum to zero, i.e., Ginv*siresol = 0';
print 'Extract siresol from sol: siresol=sol(5,6,7,8,9,10,11,12)';
siresol=j(nanim*nt,1,0);
do i=1 to nanim*nt;
    siresol[i]=sol[nf+i];
end;
print sol siresol;

print 'Compute sumsiresol = Ginv*siresol';
sumsiresol=sum(Ginv*siresol);
print sumsiresol;
print sumsiresol [format=6.2];

```

```
*=====
=====;
*=====
=====;

finish solve;
*=====
=====;
*=====
=====;
*=====
=====;
*=====
=====;

run solve;

*=====
=====;
*** Final statements ***;
*=====
=====;
quit; *** Must be placed BEFORE the ods statements below !!!! ***;
*%mend solvemme;

*%solvemme;
*run;

ods csv close;
ods graphics off;
ods html close;
*ods listing;
ods trace off;
```