

```

=====
=====;
***** ANIMAL BREEDING NOTES *****
***** CHAPTER 20 SAGM *****
***** ACCUMULATED ADDITIVE GENETIC GROUP MODEL (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_20_AAGM_Example_February-19-2010_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 20 AAGM';
print 'ACCUMULATED ADDITIVE GENETIC GROUP MODEL (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 = Weaning weight dataset used in Chapters 16 and 17';
print 'Nonparents only (animals 11 to 15)';
*=====
=====
print 'Enter nt = Number of traits';
nt=1; print nt;
print 'Enter nrec = Number of records';
nrec=5; print nrec;
print 'Enter nanim = Number of animals (4 sires and maternal
grandfathers)';
nanim=4; print nanim;
print 'Enter nf = Number of fixed effects in the MME';
nf=2; print nf;
print 'Enter nga = Number of random additive direct and maternal genetic
effects in the MME';
nga=nanim; print nga;
print 'Enter number of fixed genetic group effects in the MME';
ngg=2; print ngg;
print 'Compute neq = f+nga+ngg = total number of MME';
neq=nf+nga+ngg; print neq;

=====
=====
print 'Enter X = matrix of fixed sex effects';
X={1 0,
   1 0,
   0 1,
   1 0,
   0 1};
print X;

print 'Enter Z = matrix of direct additive genetic effects';
Z={0 1.5 0  0,
   0 0.5 1  0,
   0 0   1.5 0,

```

```

0 0    0    1.5,
0 0    0    1.5};
print Z;

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

print 'Enter Q0 = matrix of accumulated genetic group effects';
Q0={1 0,
     0 1,
     0 1,
     0 1};
print Q0;

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

=====
=====
print 'Enter va = additive gene variance';
va=22; print va;

print 'Enter ve = residual variance';
ve=88; print ve;

print 'Compute g0 = va/ve';
g0=va/ve; print g0;

print 'Compute rg0 = 0.25*g0 = ratio for the SMGS group model (4 sires
and mgs: 1,3, 4,5)';
rg0=0.25*g0; print rg0;

print 'Compute rg0inv = inv(rg0)';
rg0inv=inv(rg0); print rg0inv;

=====
=====
print 'Enter dvecsmgs = vector with elements of D for SMGS AAGM';
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 TinvT = t(Tinv) = transpose of Tinv';
TinvT=t(Tinv); print TinvT;

print 'Compute Ainv    = transpose(Tinv)*Dsmgsinv*Tinv';
Ainv=(TinvT*Dsmgsinv)*Tinv; print Ainv;

print 'Compute A = inv(Ainv) for checking purposes only';
A=inv(Ainv); print A;

print 'Compute Ginv = Ainv*rg0inv';
Ginv=Ainv*rg0inv; print Ginv;

=====
=====
print 'Compute negTDQrg0inv = (-TinvT*Dsmgsinv)*Q0*rg0inv';
negTDQrg0inv=(-TinvT*Dsmgsinv)*Q0*rg0inv; print negTDQrg0inv;

print 'Compute QTDQrg0inv = (t(Q0)*Dsmgsinv)*Q0*rg0inv';
QTDQrg0inv=(t(Q0)*Dsmgsinv)*Q0*rg0inv; print QTDQrg0inv;

=====
=====
print 'Construct matrix of random additive and group additive
contributions to the MME';

print 'Concatenate Ginv and negTDQrg0inv horizontally = ggtdq';
ggtdq=Ginv || negTDQrg0inv; print ggtdq;

print 'Concatenate transpose(negTDQrg0inv) and QTDQrg0inv horizontally =
qtdqq';
qtdqq=t(negTDQrg0inv) || QTDQrg0inv; print qtdqq;

print 'Concatenate ggtdq and qtdqq vertically = lhsrandom';
lhsrandom=ggtdq // qtdqq; print lhsrandom;

=====
=====
print 'Enter dnvec = vector of diagonal elements of matrix DN for
nonparents';
print 'From Chapter 17: animals with records (11 to 15)';
dnvec={.609375,.671875,.6875,.6875,.6875}; print dnvec;

print 'Compute matrix R2 = (DN*g0 + I) for nonparents (i.e., animals with
records 11 to 15)';
R2=diag(dnvec)*g0+i(nrec); print R2;

print 'Compute R2inv = inv(R2)';
R2inv=inv(R2); print R2inv;

=====
=====;

```

```

print 'Construct xf = [X Z 0]';
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];
    else if j > nf+nga then xf[i,j]=0;
  end;
end;
print xf;

print 'Compute lhsfixed = fixed components of the lhs of the MME';
xftr2invxf=(t(xf)*R2inv)*xf; print xftr2invxf;

print 'Compute lhs of the MME';
print 'Add lhsrandom to the appropriate submatrices of xftinvrx';
lhs=xftr2invxf;
do i=nf+1 to neq;
  do j=nf+1 to neq;
    lhs[i,j]=lhs[i,j]+lhsrandom[i-nf,j-nf]; *Add lhsrandom elements
to lhs;
  end;
end;
print lhs;
print lhs [format=6.3];

=====
=====
print 'Compute rhs = (t(xf)*R2inv)*y';
rhs=(t(xf)*R2inv)*y; print rhs;
print rhs;
print rhs [format=6.2];

=====
=====
=====
if restronsol > 0 then do;
  print 'Impose restrictions on solutions';
  print 'Set solution for mean to zero';
  do i=1 to neq;
    if i=1 then do; *Set solutions for mean 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';
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 within groups sum to zero';

print 'Extract siresol from sol: siresol=sol(3,4,5,6)';
print 'The AAGM yields usol = TQ0g + siresol';
print '=> Need to compute Q0TDsmgsinvQ0*gsol - Q0TDsmgsinvTinv*usol =0 to
test that siresol = 0 within genetic groups';
print 'Extract usol from solution vector';
usol=j(nanim,1,0);
do i=1 to nanim;
  usol[i]=sol[nf+i];
end;
print sol usol;

```

```

print 'Extract group solutions from solution vector';
gsol=j(ngg,1,0);
do i=1 to ngg;
  gsol[i]=sol[nf+nga+i];
end;
print sol gsol;

print 'Compute sumsiresol = (t(Q0)*Dsmgsinv)*Q0*gsol -
(t(Q0)*Dsmgsinv)*Tinv*usol';
sumsiresol=(t(Q0)*Dsmgsinv)*Q0*gsol - (t(Q0)*Dsmgsinv)*Tinv*usol;
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;

```