Solver instructions, derived model quantities, and PCG convergence

Martin Lidauer, Timo Pitkänen

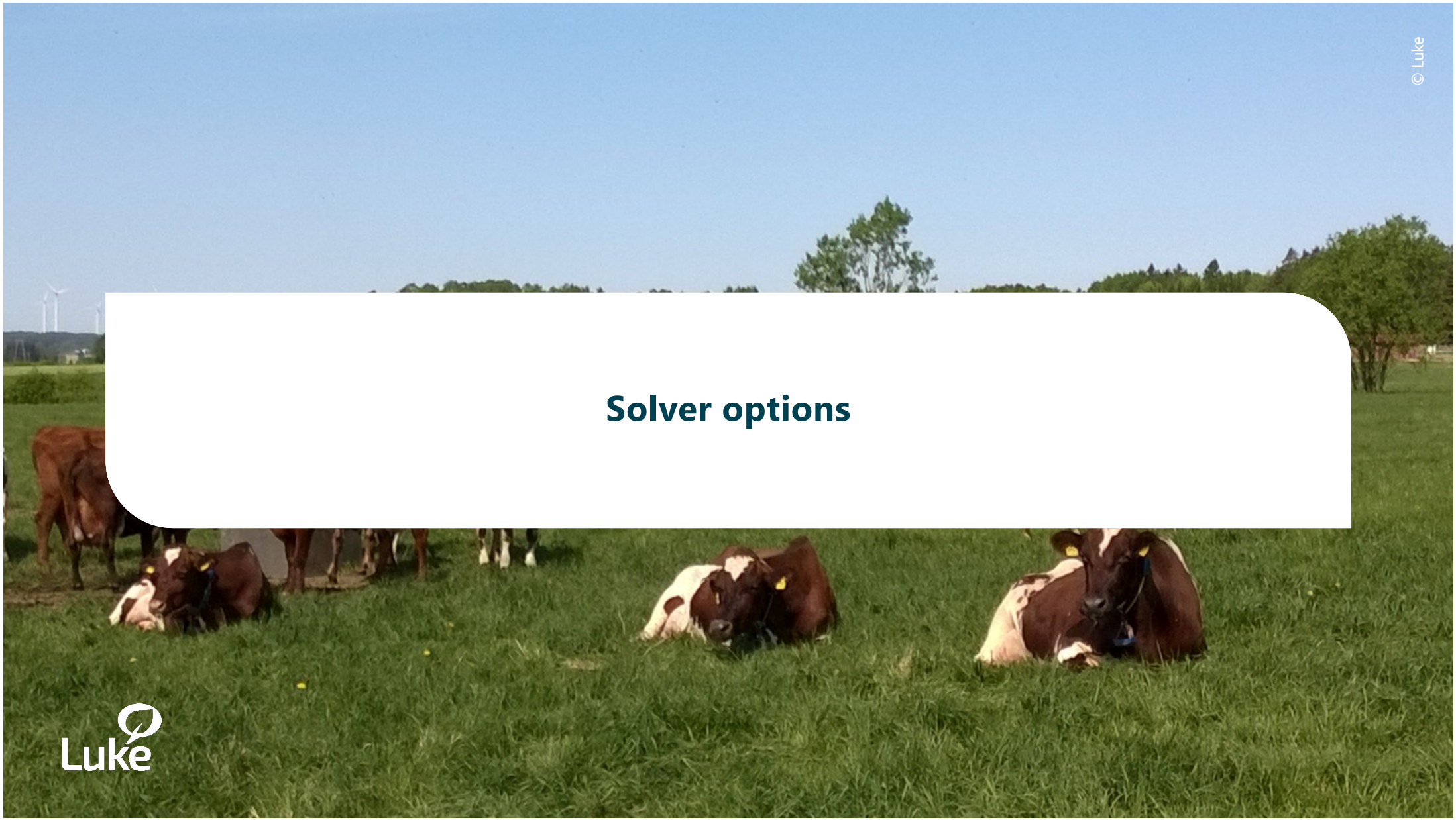
MiX99 course on genomic prediction

COURSE DAY 2, March 11th, 2026

Contents

- Solver options
  - Command line options
  - Solver option file
- Derived quantities from a fitted model
  - Residuals, predicted observations
  - Yield deviations, individual daughter deviations
  - Daughter yield deviations
  - Simulated observations
- Convergence options for PCG

Luke



Solver options

Instructing MiX99 solver programs *mix99s* and *mix99p*

Alternatives for instructing the solver programs

- Command line options: $> mix99s –s
- Solver option file given to standard input: $> mix99s < solver_options.txt
- Both: $> mix99s –i –sp 100 < solver_options.txt

Luke

Command line options

Command $> mix99s –h will show all options

```
mix99s [-s] [-i] [-nt nt] [-p|-pt|-pb] [-r|-rt|-rb] [-m]
       [-n NITER] [-ncalls NCITER] [-n0 N0ITER] [-c{a,d,r,m} TOL]
       [-peek [-]PITER] [-peek_step PSTEP]
       [-IOP|-IM|-CHM|-PAR] [-o VAL] [-t tau] [-MES|-MEM|-MEB|-MEA|-MEL]
       [-RDS|-RDM|-RDL|-RDX|-RDB <blocksize>|-RDU <mem>]
       [-noQ] [-fQP] [-sp val] [-srm n] [-nocov] [-RHS rhsfile]
       [-MS] [-PA]
       [-h|--help] [-v|--verbose] [-V|--version]
       [--bindir BINDIR] [--datadir DATADIR] [--tmpdir TMPDIR]
where
  -s: use defaults, solve and produce standard solution files.
  -i: use both input file and command line options.
  -nt nt: number of threads set to nt.
  -p or -pt: use defaults, solve and produce predictions.
  -pb: same as -p but produce binary file(s).
  -r or -rt: use defaults, solve and produce residuals.
  -rb: same as -r but produce binary file(s).
  -m: Generate MME.dat and rhs.dat of the problem.
  -n NITER: number of iterations.
  -ncalls NCITER: number of PCG solver calls (e.g. deregression).
  -n0 N0ITER: minimum number of iterations.
      Default: 20 if Cd criterion and old solutions, 1 otherwise.
  -peek [-]PITER: Write intermediate solutions at iteration PITER.
     If negative, file extension _PEEK instead of _<ITER>.
  -peek_step PSTEP: Write solutions every PSTEP iterations.
     If PITER negative, file extension _<ITER> instead of _PEEK.
  -peekcr CRTOL Write intermediate solutions at Cr==CRTOL
  -ca TOL: Ca convergence stopping criterion.
  -cd TOL: Cd convergence stopping criterion.
  -cr TOL: Cr convergence stopping criterion.
  -cm TOL: Cm convergence stopping criterion.
  -IOP: use iteration on pedigree in inv(A11) of single-step.
  -IM: use iteration on matrix in inv(A11) of single-step.
  -CHM: use Cholmod in inv(A11) of single-step.
  -PAR: use PARDISO in inv(A11) of single-step (default).
  -o: coefficient VAL multiplies inv(A22) in ssGBLUP.
  -t: coefficient TAU multiplies inv(G)/inv(H) in ssGBLUP.
  -MES: no   inv(G)/T in memory.
  -MEM: read inv(G)/T matrix to memory.
  -MEB <bSize>: read inv(G)/T matrix to memory and compute by <bSize>.
  -MEA: read inv(G)/T matrix to memory and compute by automatic block size.
  -MEL: read inv(G)/T matrix to memory and use efficient multiplication (default).
  -RDS: small block size with regression design matrices (RDM). Packed SNP matrix
        in memory.
  -RDM: medium block size with RDM. Packed SNP matrix in memory.
  -RDL: large block size with RDM. Packed SNP matrix in memory.
  -RDX: full (double precision) RDM kept in memory.
  -RDB <blocksize>: given blocksize for RDM.
        Packed SNP matrix kept in memory (if blocksize > 0).
  -RDU <memory>: given memory usage for RDM. Block size and
        packed marker matrix usage depending on <memory>,
        where <memory> is <number>[K/M/G/T/P], e.g. 12G.
  -MS: Mendelian sampling terms written to SolMS.
  -PA: Parent averages written to SolPA.
  -noQ: no groups for inv(A22) if none in inv(G).
  -fQP: include groups for inv(G) in componentwise ssGTABLUP.
  -sp val: second level preconditioner for RegMatrix.
  -srm n : RegMatrix number n is sparse and read to RAM.
  -continue_reml: Continue previous REML iteration.
  -PEV: Monte Carlo PEV.
  -nocov: No residual covariances assumed between traits.
  -RHS rhsfile: name of the file for the external right-hand side.
  -nocheck: no checking of data and release time information.
  -silent: PCG iteration screen output small.
  -h or --help    : Show usage.
  -v or --verbose: Show additional information.
  -V or --version: Show version information.
  --bindir BINDIR:
       Directory for MiX99 binaries. Default: (empty)
  --datadir DATADIR:
       Data directory. Default: (empty)
  --tmpdir TMPDIR:
       Directory for temporary files. Default: (empty)
```

Command line options

Examples

- $> mix99s –s > mix99s.log
  Default options are used for solving
- $> mix99s –s –nt 10
  Default options, number of threads set to 10
- $> mix99s –s –sp 100
  Default options, second-level preconditioner value 100
- $> mix99s –s –n 50 | tee mix99s.log
  Default options, but maximum iteration number is 50

Modifying options at run-time

- $> touch STOP
  Solver stops iterations and writes solutions
- $> echo "-500 200" > PEEK
  Stores intermediate solutions starting at iteration round 500 and in intervals of 200 iterations
- $> echo "6000 1.0e-6 R F" > ITER
  Changes convergence criteria while solving
- $> cat <<EOD > ITER
  $> # PCG
  $> 6000 1e-6 R F
  $> # REML
  $> 200 5 1e-10
  $> EOD
  Changes PCG and REML convergence criteria while solving

Luke

Solver option file

```
#-----------------------------------------------------
# RAM:    RAM options
          H  nt 10 # H  All vectors in memory
                   # X  For prallel computing only
                   # M  Solution vector on disk
                   # L  Solutions and residuals on disk
                   # nt 10  use 10 CPU threads (optional)
                   # sp <v> second level preconditioner
                   # many memory options MEL, MEA, ...
# STOP:   Max.iter, Conv.value, Criterion, Enforce
          5000      1.0e-5      d          f
                              #  a   Ca norm
                              #  r   Cr norm
                              #  m   Cm norm
                              #  d   Cd norm
# RESID:  Residuals calculation
          N
# VALID:  Model validation
          N
# VAROPT: Variance options VCE, PEV, HV
          N
# SOLTYP: Solution file options
          Y
#-----------------------------------------------------
```

Basic solving options

First character at **RAM** line defines use of random access memory

additional parameters are optional

Stopping criteria are specified at **STOP** line

- Carefully tailoring of these parameters allows to optimize solving time
- **Cr** norm (relative difference between RHS and LHS), and **Cd** norm (relative difference between solutions of the last tow iteration rounds) most useful
- Default values are only changed if “f” is specified

Specify Y at **SOLTYP** option line for writing solutions

Luke

Derived quantities from a fitted model

Derived quantities

- MiX99 allows to calculate different information for model development and validation
- The quantities are derived from the MME
- Instructions for calculating these quantities are specified in the MiX99 solver option file

Example for demonstration

Multi-trait repeatability model for Saved Feed

- 3 traits: feed intake, ECM, BW growth
- Fixed regressions and fixed herd-test-month
- Random pe and additive genetic effect

Fitted model

$$\mathbf{y} = \mathbf{X\beta} + \mathbf{Z}_p\mathbf{p} + \mathbf{Z}_a\mathbf{a} + \mathbf{e}$$

where **β**, **p**, **a**, and **e** are vectors of fixed effects, and random nonhereditary, additive genetic and residual effects

CLIM file

```
#-------------------------------------------------------------------------
DEFINE      Input  /home/ejo31/MiX99course2026/data
DATAFILE    Input/jokis.data
PEDFILE     Input/jokis.pedi
PARFILE     Input/jokis.para
TABLEFILE   Input/LegendreExp005.cov
TABLEINDEX DIM

INTEGER     ID HERD HTM DIM
REAL        dmi ecm growth age age2
MISSING     -99999.0

MODEL
 dmi     =  CalvAge(age age2)  Lac(t1 t2 t3 t4 t5|HERD)  HTM  pe(ID)  g(ID)
 ecm     =  CalvAge(age age2)  Lac(t1 t2 t3 t4 t5|HERD)  HTM  pe(ID)  g(ID)
 growth  =  CalvAge(age age2)  Lac(t1 t2 t3 t4 t5|HERD)  HTM  pe(ID)  g(ID)

RANDOM      pe g
PEDIGREE    g  am

DATASORT          BLOCK=HERD  PEDIGREECODE=ID
WITHINBLOCKORDER  g  pe  HTM
PRECON            b  b   b     b
TMPDIR            ./tmpMiX
#-------------------------------------------------------------------------
```

8 fixed effect factors

2 random factors

Residuals

Derived quantity

$$\hat{\mathbf{e}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{Z}_p\hat{\mathbf{p}} - \mathbf{Z}_a\hat{\mathbf{a}}$$

- Specify Y and t (text file) in **RESID** option line
- Residuals are written to the file(s) ***eHat.data0***
- The parallel solver *mix99p* will create for each process an own file: ***eHat.data0***, ***eHat.data1***, ***eHat.data2***, …
- Order of output lines in the file is the same as in the MiX99 input data file
- Order of output columns follow order of the traits in the **MODEL** command
- Output can be binary of text format
- Missing values are coded by -8192.0

```
sorvi-int1:~/MiX99course2026/L5mt3> cat slv.ehat
#-----------------------------------------------
# RAM:    RAM options
          H
# STOP:   Max.iter,Conv.value,Criterion,Enforce
          5000      1e-5      d          f
# RESID:  Residuals calculation
          Y  t
# VALID:  Model validation
          N
# VAROPT: Variance options VCE, PEV, HV
          N
# SOLTYP: Solution file options
          Y
#-----------------------------------------------
sorvi-int1:~/MiX99course2026/L5mt3> mix99s < slv.ehat >s.log
sorvi-int1:~/MiX99course2026/L5mt3> head eHat.data0
   2.128278      -3.027954      0.2619540
  0.4872776      -1.427297      0.2507850
   1.617288      -1.850028      0.2188042
  0.4415096      -2.777971      0.2059910
  0.8268792      -2.265003      0.1925371
  1.097215       -2.442603      0.1785123
```

Predicted observations

Derived quantity

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{Z}_p\hat{\mathbf{p}} + \mathbf{Z}_a\hat{\mathbf{a}}$$

- Specify P and t (text file) in **VALID** option line
- Predicted observations are written to the file ***yHat.data0***
- Setup of the ***yHat.data0*** file is same as for the ***eHat.data0*** file

```
sorvi-int1:~/MiX99course2026/L5mt3> cat slv.yhat
#-------------------------------------------
# RAM:    RAM options
          H
# STOP:   Max.iter,Conv.value,Criterion,Enforce
          5000      1e-5      d          f
# RESID:  Residuals calculation
          N
# VALID:  Model validation
          P  t  # predicted observations
# VAROPT: Variance options VCE, PEV, HV
          N
# SOLTYP: Solution file options
          Y
#-------------------------------------------
sorvi-int1:~/MiX99course2026/L5mt3> mix99s < slv.yhat >s.log
sorvi-int1:~/MiX99course2026/L5mt3> head yHat.data0
   18.19292       26.09965       0.3035460
   18.33972       25.95250       0.3061150
   18.98171       26.15643       0.3288957
   19.08259       26.01367       0.3322090
   19.16482       25.87820       0.3359629
```

Yield deviations

Derived quantity

$$\mathbf{YD} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{Z}_p\hat{\mathbf{p}}$$

- Specify Y and t (text file) in **VALID** option line
- Give a 1 for model factors included into the YD, and a 0 for all other model factors in **FACTOR** option line
- Order of model factors is same as specified in **MODEL** command (see CLIM file):
  - 8 fixed effect factors
  - 2 random effect factors
- YDs are written to the file ***YD.data0***
- Setup of ***YD.data0*** is same as for ***eHat.data0***

```
sorvi-int1:~/MiX99course2026/L5mt3> cat slv.YD
#-------------------------------------------
# RAM:    RAM options
          H
# STOP:   Max.iter,Conv.value,Criterion,Enforce
          5000      1e-5      d          f
# RESID:  Residuals calculation
          N
# VALID:  Model validation
          Y  t  # yield deviations
# FACTOR: age age2 t1 t2 t3 t4 t5 HTM pe g
          0   0    0  0  0  0  0  0   0  1
# VAROPT: Variance options VCE, PEV, HV
          N
# SOLTYP: Solution file options
          Y
#-------------------------------------------
sorvi-int1:~/MiX99course2026/L5mt3> mix99s < slv.YD >s.log
sorvi-int1:~/MiX99course2026/L5mt3> head YD.data0
   2.419862      -3.026446      0.2931160
  0.7788612      -1.425789      0.2819469
   1.908872      -1.848519      0.2499662
  0.7330934      -2.776463      0.2371530
   1.118463      -2.263495      0.2236990
```

Individual daughter deviations

Derived quantity

$$\mathbf{IDD} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{Z}_p\hat{\mathbf{p}} - 1/2\,\hat{\mathbf{a}}_{dam}$$

- Specify I and t (text file) in **VALID** option line
- Give a 1 for model factors included into the IDD, and a 0 for all other model factors in **FACTOR** option line
- Order of model factors is same as specified in **MODEL** command (see CLIM file)
- IDDs are written to the files ***IDD.data(0)***
- Setup of ***IDD.data(0)*** is same as for ***eHat.data(0)***

```
sorvi-int1:~/MiX99course2026/L5mt3> cat slv.IDD
#-------------------------------------------
# RAM:    RAM options
          H
# STOP:   Max.iter,Conv.value,Criterion,Enforce
          5000      1e-5      d          f
# RESID:  Residuals calculation
          N
# VALID:  Model validation
          I  t  # individual daughter deviations
# FACTOR: age age2 t1 t2 t3 t4 t5 HTM pe g
          0   0    0  0  0  0  0  0   0  1
# VAROPT: Variance options VCE, PEV, HV
          N
# SOLTYP: Solution file options
          Y
#-------------------------------------------
sorvi-int1:~/MiX99course2026/L5mt3> mix99s < slv.IDD >s.log
sorvi-int1:~/MiX99course2026/L5mt3> head IDD.data0
   2.338650      -3.017248      0.2758645
  0.6976491      -1.416591      0.2646954
   1.827660      -1.839321      0.2327147
  0.6518813      -2.767265      0.2199015
   1.037251      -2.254297      0.2064475
```

Daughter yield deviations

Implemented as described by Mrode & Swanson 2004, (Livest.Prod.Sci.86:), which is a generalization of VanRanden & Wiggan's method (1991, J. Dairy Sci.74:)

Derived quantity: a $\mathbf{DYD}_{(i)}$ vector for a sire (i) is calculated

$$\mathbf{DYD}_{(i)} = \left(\sum_j \kappa_j \, \boldsymbol{G}^{-1} \mathbf{W}_{ij} \right)^{-1} \sum_j \kappa_j \, \boldsymbol{G}^{-1} \mathbf{d}_{ij}$$

where

$$\mathbf{d}_{ij} = \left[\left(a^{ij} \boldsymbol{G}^{-1} + \boldsymbol{Q}'_j \boldsymbol{R}^{-1} \boldsymbol{Q}_j \right)^{-1} \boldsymbol{Q}'_j \boldsymbol{R}^{-1} \left(\boldsymbol{y}_j - \boldsymbol{X}_j \hat{\boldsymbol{b}} - \boldsymbol{Z}_j \hat{\boldsymbol{p}} \right) \right] - 0.5 \hat{\boldsymbol{a}}_{dam}$$

$$\mathbf{W}_{ij} = \left(a^{ij} \boldsymbol{G}^{-1} + \boldsymbol{Q}'_j \boldsymbol{R}^{-1} \boldsymbol{Q}_j \right)^{-1} \boldsymbol{Q}'_j \boldsymbol{R}^{-1} \boldsymbol{Q}_j$$

- Note, vector $\mathbf{DYD}_{(i)}$ contains regression coefficients in case of random regression models
- DYDs can be calculated for different daughter groups of a sire, e.g., birth year groups (please see manual)

Daughter yield deviations

Instruction in CLIM file

- After the **PEDFILE** command, DYD is specified before the pedigree filename
- A block diagonal preconditioner must be specified for the additive genetic animal effect

```
DATAFILE    Input/jokis.data
PEDFILE     DYD Input/jokis.pedi
PARFILE     Input/jokis.para
INTEGER     ID HERD HTM DIM
REAL        dmi ecm growth age age2
MODEL
 dmi      =  CalvAge(age age2)  Lac(t1 t2 t3 t4 t5|HERD)  HTM  pe(ID)  g(ID)
 ecm      =  CalvAge(age age2)  Lac(t1 t2 t3 t4 t5|HERD)  HTM  pe(ID)  g(ID)
 growth   =  CalvAge(age age2)  Lac(t1 t2 t3 t4 t5|HERD)  HTM  pe(ID)  g(ID)
RANDOM      pe g
PEDIGREE    g  am
WITHINBLOCKORDER  g  pe  HTM
PRECON            b  b   b     b
TABLEFILE   Input/LegendreExp005.cov
```

Instruction for *mix99s*:

- Specify D in **VALID** option line
- Give a 1 for model factors included into the DYD, and a 0 for all other model factors in **FACTOR** option line
- DYDs will be written to file ***Soldyd***

```
sorvi-int1:~/MiX99course2026/L5mt3> cat slv.DYD
#-------------------------------------------
# RAM:    RAM options
          H
# STOP:   Max.iter,Conv.value,Criterion,Enforce
          5000      1e-5      d          f
# RESID:  Residuals calculation
          N
# VALID:  Model validation
          D   # daughter yield deviations
# FACTOR: age age2 t1 t2 t3 t4 t5 HTM pe g
          0   0    0  0  0  0  0  0   0  1
# VAROPT: Variance options VCE, PEV, HV
          N
# SOLTYP: Solution file options
          Y
#-------------------------------------------
sorvi-int1:~/MiX99course2026/L5mt3> mix99s < slv.DYD >s.log
sorvi-int1:~/MiX99course2026/L5mt3> head Soldyd
1208951136       1      1      33  0.23570      0.16406E-01  0.14843E-01
1206595566       1      2      62 -0.70859      0.20047E-01 -0.79290E-01
1206828544       1      5     148 -0.39753      0.98703     -0.13861
1206813806       1      1      36 -0.19926E-01   1.4855     -0.10987
1206810743       1      5     178 -0.45491E-01  0.87411     -0.78786E-01
```

Simulated model effects and observations

Simulated quantity

$$\tilde{\mathbf{y}} = \mathbf{X}\tilde{\boldsymbol{\beta}} + \mathbf{Z}_p\tilde{\mathbf{p}} + \mathbf{Z}_a\tilde{\mathbf{a}} + \tilde{\mathbf{e}}$$

- Simulation is based on the given model, variance components, data, and pedigree
- Effects in $\tilde{\boldsymbol{\beta}}$ are assumed to be zero, unless a ***Solvec*** file with earlier estimates is available
- Specify G and t (text file) in **VALID** option line
- Specify a random seed parameter in **SEED** option line
- Generated observations are written to the file ***ySim.data0***
- Setup of ***ySim.data0*** is same as for ***eHat.data0***
- Missing values are coded by -8192.0
- Simulated effects are written to MiX99 solution files

STOP options will be ignored.
No PCG iterations.

```
sorvi-int1:~/MiX99course2026/L5mt3> cat slv.ytilde
#-------------------------------------------------
# RAM:    RAM options
          H
# STOP:   Max.iter,Conv.value,Criterion,Enforce
          5000      1e-5      d          f
# RESID:  Residuals calculation
          N
# VALID:  Model validation
          G  t  # generate observations
# VAROPT: Variance options VCE, PEV, HV
          N
# SEED:   Type of seed for random number generator
          R
# SOLTYP: Solution file options
          Y
#-------------------------------------------------
sorvi-int1:~/MiX99course2026/L5mt3> mix99s < slv.ytilde >g.log
sorvi-int1:~/MiX99course2026/L5mt3> head ySim.data0
  -1.873842      -2.533371      0.1313898
  -2.178329     -0.7883206      0.1054773
   2.283859     -7.6049313E-02 -7.0590638E-02
  -1.585310      -2.350895      1.5975308E-02
```

Convergence options for PCG

Possibilities to speed up PCG convergence

- Applying best possible preconditioner
- Optimizing relationship information
  - Pruning the pedigree
  - Cleaning pedigree from errors

    The RelaX2 program provides many options for pruning and pedigree checks
  - Optimizing genetic groups / modelling of metafounders
- Optimizing the applied model
  - Practical rule for model development: as simple as possible but as specific as needed
  - Avoiding overparameterized models (e.g. applying parameter-reduced models in test-day model evaluations)

  Experience has shown: Poor convergence is often caused by fitting an ill-conditioned model

Luke

Applying best possible preconditioner

- Preconditioner choice for within-block and across-block effects
  - Specifying block-diagonal "b" preconditioner is best choice for most models (default)

```
DATASORT            BLOCK-HERD  PEDIGREECODE-ID
WITHINBLOCKORDER    g   pe   HTM
PRECON              b   b    b      b
TMPDIR               /tmpMiX
```

After the within-block specifications one more parameter is specified for all across-block effects

  - However, for large systems of equations, it may become memory-intensive and time-intensive
  - Then, diagonal "d" preconditioner need to be applied for some or all within-block effects
- Single-step models
  - There are additional options for single-step models that depend on the applied solving strategy (please see single-step lecture slides)
- Models with REGMATRIX option, and ssSNPBLUP models
  - Applying a second-level preconditioner can enhance convergence considerably

    e.g., $> mix99s –i –sp 100 < solver_options.txt

Luke

Example of preconditioning

- Same demonstration example
- Block-diagonal *versus* diagonal preconditioner

Preconditioner	N. of iterations	Total time
d d d d	421	1.84 s
b b b b	214	1.03 s