MiX99

Solving Large Mixed Model Equations

MTT

MiX99 overview

Martin Lidauer, Kaarina Matilainen, Esa Mäntysaari, Timo Pitkänen, Matti Taskinen, Ismo Strandén

It started on a misery winter day in 1997, somewhere on a railway platform in Helsinki …

The problem

Finnish test-day model project 1995-1999

- Need for an efficient software to solve test-day models
- Complex models: multiple trait, random regressions, …
- Large data: millions of records …
- Ambitious aim to perform continuous evaluations …
- **Commonly used solving algorithms were too slow**

Earlier routine software developments

- Very efficient repeatability animal model solver…
- Multiple -trait maternal model solver…
- **We wanted a general solver**

Preconditioned conjugate gradient (PCG)

MME to be solved: $\mathbf{Ca} = \mathbf{r}$

Set:

$\mathbf{a}^{(0)} \Leftarrow$ initial guess; $\quad \mathbf{e}^{(0)} \Leftarrow \mathbf{r} - \mathbf{C}\mathbf{a}^{(0)}$

$\mathbf{d}^{(0)} \Leftarrow \mathbf{M}^{-1}\mathbf{e}^{(0)}; \quad f_0 \Leftarrow \mathbf{e}^{(0)\prime}\mathbf{d}^{(0)}$

For $k = 1, 2, \ldots$

$\mathbf{q}^{(k)} \Leftarrow \mathbf{C}\mathbf{d}^{(k-1)}; \quad \alpha_k \Leftarrow f_{k-1} / \mathbf{d}^{(k)\prime}\mathbf{q}^{(k)}$

$\mathbf{a}^{(k)} \Leftarrow \mathbf{a}^{(k-1)} + \alpha_k \mathbf{d}^{(k-1)}$

if k is divisible by 100

$\mathbf{e}^{(k)} \Leftarrow \mathbf{r} - \mathbf{C}\mathbf{a}^{(k)}$

else

$\mathbf{e}^{(k)} \Leftarrow \mathbf{e}^{(k-1)} - \alpha_k \mathbf{q}^{(k)}$

$\mathbf{s}^{(k)} \Leftarrow \mathbf{M}^{-1}\mathbf{e}^{(k)}$

$f_k \Leftarrow \mathbf{e}^{(k)\prime}\mathbf{s}^{(k)}; \quad \beta_k \Leftarrow f_k / f_{k-1}$

$\mathbf{d}^{(k+1)} \Leftarrow \mathbf{s}^{(k)} + \beta_k \mathbf{d}^{(k)}$

if not convergence continue iteration

- PCG solves the MME in a finite number of steps (given **C** is PD)
- Search steps are conjugate (**C**-orthogonal) and have steepest descent
- PCG is suitable for an iteration-on-data implementation
- Only $\mathbf{Cd}^{(k-1)}$ and $\mathbf{M}^{-1}\mathbf{e}^{(k)}$ are computationally demanding

More speed

HP Breeding Project 1998-1999

(High-Performance Computing and Networking EU project)

- Faba co-op (initiator & responsible), Vantaa
- MTT, Jokioinen
- CSC-IT Center for Science, Espoo
- Finnish Agricultural Data Processing Center, Vantaa

- **AIM**

  Use of parallel computing in solving large mixed model equations

- **APPROACH**
  - 1. optimization of existing computer code for single processor
  - 2. developing of parallel computing code using portable MPI library

3-step approach for PCG iteration-on-data

- Matrix multiplications for each animal q to get **Cd:**

$$\mathbf{Cd} = \sum_{i=1}^{N_q} \mathbf{w}_i \mathbf{R}_i^{-1} \mathbf{w}_i' \mathbf{d} + \mathbf{V}^{-1}\mathbf{d} = \sum_{i=1}^{N_q} \mathbf{v}_i + \mathbf{v}_d$$

- **Gauss-Seidel:** in 2 steps

$$\mathbf{s}_i \Leftarrow \mathbf{R}_i^{-1}\mathbf{w}_i'\mathbf{d}; \quad \mathbf{v}_i \Leftarrow \mathbf{w}_i\mathbf{s}_i$$

- **PCG:** possible in 3 steps

$$\mathbf{s}_i \Leftarrow \mathbf{w}_i'\mathbf{d}; \ \mathbf{s}_i^r \Leftarrow \mathbf{R}_i^{-1}\mathbf{s}_i; \ \mathbf{v}_i \Leftarrow \mathbf{w}_i\mathbf{s}_i^r$$

Solving Large Mixed Linear Models Using Preconditioned Conjugate Gradient Iteration
Strandén & Lidauer, 1999, J. Dairy Sci. 82:

Floating point operations per animal record to get $\mathbf{v}_i$

	Gauss-Seidel 2 steps	**PCG** 3 steps
DEA test-day model evaluation	2 580	118
Irish beef cattle evaluation	38 991	906
Nordic RDC test-day model evaluation	81 007	674

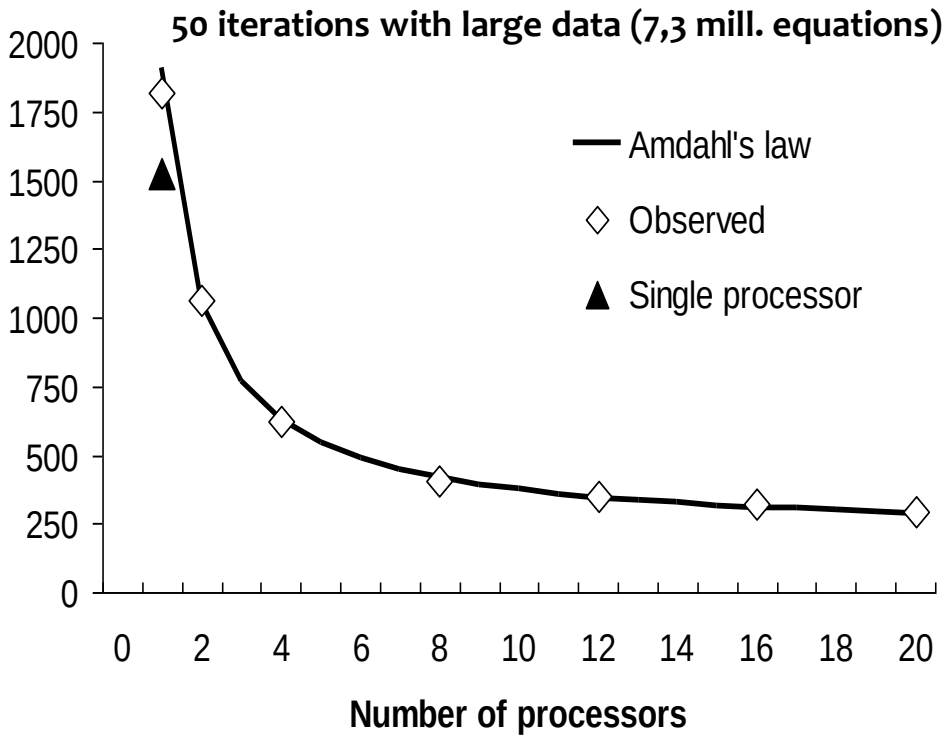
Additional speed through parallel processing

- Data and equations are organized (by the pre-processor) to create data locality
- PCG iteration-on-data solver was re-written to make optimal use of parallel computing
- We achieved very satisfying speed-up, especially for cattle data

Parallel computing applied to breeding value estimation in dairy cattle

Strandén & Lidauer, 2001, J. Dairy Sci. 84:

Speed-up tests for new solver on SGI computing platform at SCS, 1998



1999: First release of MiX99

Solving Large Mixed Model Equations

Model features

- Multiple traits
- Trait-specific model effects
- Random regression
- Reduction of model dimension (reduced rank)
- Maternal/paternal effects
- $\mathbf{A}^{-1}$: options: animal model, sire/maternal grand sire, phantom parent groups

Solving features

- PCG
- Iteration-on-data
- Data locality
- Parallel processing
- Trait grouping

2000: MiX99 in routine use

Finnish test-day model evaluation

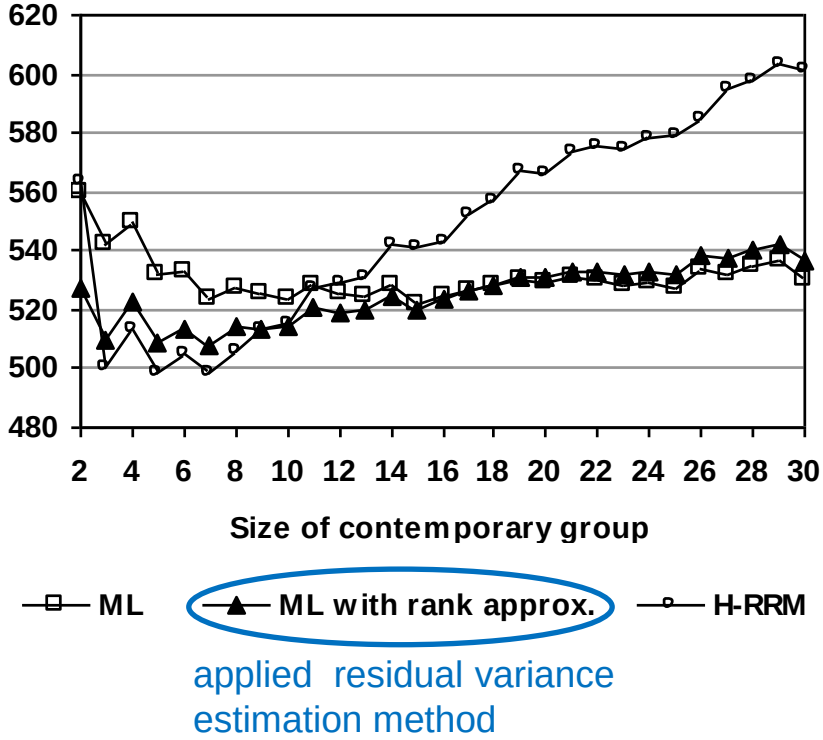
Heterogeneous variance adjustment

DEU-AUT test-day model project 2000-2002

- Need to account for heterogeneous variance (HV)
- Multiplicative mixed model approach (**Meuwissen et al.** 1996. JDS:79) implemented
- Computationally demanding approach → allows modelling of main sources of HV

Multiplicative random regression model for hetero-geneous variance adjustment in genetic evaluation for milk yield in Simmental
Lidauer et al., 2008. J.Anim.Breed.Genet. 125:

Genetic SD for milk yield in cows classified by contemporary group size



Approximation of reliabilites

Need to get reliabilities for Finnish and DEU-AUT routine evaluations

- **Interbull weighting factors** suitable for RR models (**2000**)

Calculation of Interbull weighting factors for the Finnish test day model
Strandén et al., 2002, Interbull Bulletin 26:

- **Exact reliabilities** (**2000**) limited to small size data

MiX99 for UK test-day model (2002-2004)

- **Misztal and Wiggans approach** (1988, J. Dairy Sci. 72,2:27-32)
- **Jamrozik et al. approach** (2000, Livest. Prod. Sci. 66:85-92)
- **Tier and Meyer approach** (2004, J. Anim. Breed. Genet. 121:77-89)

ApaX

Non-linear models

Kaarina Matilainen (née Vuori) joined

Pig growth curve project 2003-2004

- **Gompertz function**: $y_{ij} = \alpha \exp(-\beta \exp(-\kappa t_{ij})) e_{ij}, j = 1, \ldots, n_i$
- Implementation based on EBLUP-expansion method

Estimation of non-linear growth models by linearization: a simulation study using a Gompertz function
Vuori et al., 2006. Genet. Sel. Evol. 38:

UK beef and sheep project 2007-2009

- **Linear-threshold model:**
- Solving algorithm options:
  - Expectation Maximization
  - Newton Raphson

Linear–threshold animal model for birth weight, gestation length and calving ease in United Kingdom Limousin beef cattle data
Matilainen et al., 2009. Livest. Sci. 122:

Co-operation with Wageningen UR Livest. Res.

MiXBLUP project 2007-present

Need of a fast BLUP software for Windows platforms that allows state of the art models and is easy to use

Approach:

- Kernel from MiX99
- User-friendly interface
- Flexible data input
  - Allows also alpha numeric data
  - Derived variables
- New model options
  - Social effects
  - IBD matrices


```
TITLE EBVs for body weight 1 and body weight 2
DATAFILE datafile.txt !MISSING -999
animal    A
herd      I
sex       I
age1      R
age2      R
bw1       T
bw2       T
PEDFILE !groups 0.0
animal A
sire   A
dam    A
PARFILE para.dat
MODEL
bw1 ~ herd sex !random G(animal)
bw2 ~ herd sex !random G(animal)
SOLVING
!maxit 1000
```

Mulder et al. 2012. MiXBLUP Manual. *Anim. Breed. Genomics Centre, Wageningen UR Livest. Res.* www.mixblup.eu

MC- EM REML variance components (VC) for complex models

VC estimation for large models is computationally demanding

Idea:

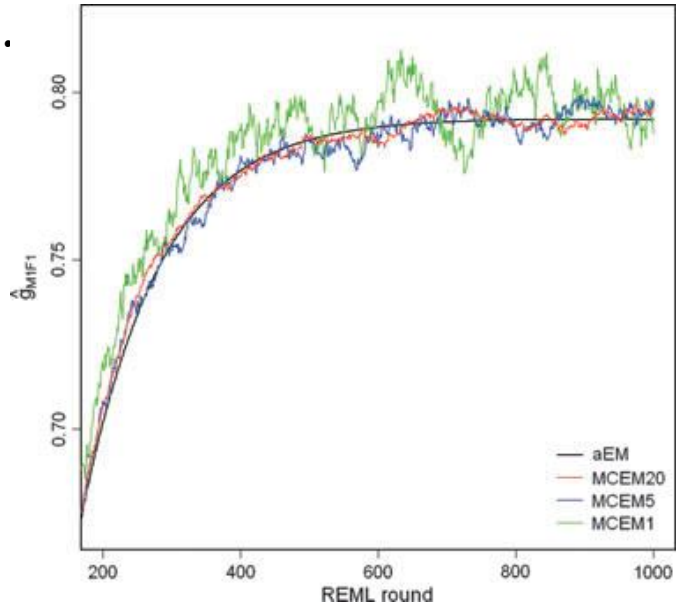
- Fast BLUP software should enhance empirical REML

  PhD project with Rothamsted Research, UK

  - Implementation **based on García-Cortés et al.** (1995, J. Anim. Breed. Genet. 112:176-182)
  - Extended for multiple trait models

    Employing a Monte Carlo algorithm in expectation maximization restricted maximum likelihood estimation of the linear mixed model
    Matilainen et al., 2012. J.Anim.Breed.Genet.129:

- Computational demand significantly smaller compared to Gibbs sampling



Command Language Interface for MiX99 CLIM

MiX99 supports 2 input syntaxes

CLIM syntax

```
TITLE        TDM: 1. lactation milk yield, 2. order Legendre pol.

DATAFILE     /koel/testi/data/mTab.dat
INTEGER      herd animal htm ym age dcc season dim
REAL         milk
MISSING      0.0
DATASORT     BLOCK=herd PEDIGREECODE=animal

PEDFILE      /koel/testi/data/blk.ped
PEDIGREE     G am+p

TABLEFILE    covarmilk.tab
TABLEINDEX   dim

PARFILE      legendrepol.par

RANDOM              htm PE G
WITHINBLOCKORDER    PE G htm herd
PRECON               d  d  b  d    m 1 1 2 2

NORANSOL            PE

MODEL
             milk = herd curve(1 t2 t3 t4 t5| season) age dcc ym &
                    htm PE(t1 t2 t3| animal) G(t1 t2 t3| animal)
```

MiX99 instruction syntax

```
# TITLE:
        TDM: 1. lactation milk yield, 2. order Legendre pol.
# INTEGER:
        herd animal htm ym age dcc season dim
# REAL:
        milk
# TRAITS:
        1
# TRAITGRP:
        1     -
# DATASORT:
        1     2
# FIXRAN:
        9     7
# MODEL: TRAITGRP, trait, wgt: herd b0 b1 b2 b3 b4 age dcc ym htm z0 z1 z2 g0 g1 g2
        1          1     -     1  7  7  7  7  7  5   6   4  3   2  2  2  2  2  2
# WITHINBLOCKORDER:
                               4  -  -  -  -  -  -  -   -  -  3   1  1  1  2  2  2
# RANDOM:                                                      htm z0 z1 z2 g0 g1 g2
                                                               1   2  2  2  3  3  3
# RELATIONSHIPS: number:
        3                                                                    1  1  1
# REGRESS: number:             herd b0 b1 b2 b3 b4 age dcc ym htm z0 z1 z2 g0 g1 g2
        16                     cl   cl t2 t3 t4 t5 cl  cl  cl cl  t1 t2 t3 t1 t2 t3
# COMBINE:
        n
# CVRFIL: name of the file with covariable table
        covarmilk.tab
# CVRNUM: number of the covariable columns in the file
        5
# CVRIND: integer column in the data, which conatins the covariable index
        8
# PEDIGREE:
        am+p
# DATAFILE:
        /koel/testi/data/mTab.dat
# VAR:
        8  1  f
# MISSVA:
        0.0
# SCALE:
        n
# PEDFILE:
        /koel/testi/data/blk.ped
# PARFILE:
        legendrepol.par
# TMPDIR:
        .
#RANSOLFILE: solution files for the random effects: htm n-ga animal
        y     n     y
# SOLUNF: unformatted solution file
        y
# PRECON:
        d  d  b d m 1 1 2 2
# PARALLEL:
        1
# COMMONBLOCKS:
        0
```

Miscellaneous useful options

Timo Pitkänen joined

- ŷ's & residuals
- Yield deviations
- Daughter yield deviations

  based on **Mrode & Swanson** (2004, *Livest. Prod. Sci.* 86: 253-260)
- Individual daughter deviations
- Simulated observations and solutions
- Multiple-trait deregression

A recipe for multiple trait deregression. Strandén & Mäntysaari, 2010. Interbull Bulletin 42:

- Inbreeding coefficients in $\mathbf{A}^{-1}$
- Random phantom parent groups
- Heterogeneous residual (co)variance matrices
- HV adjustment when observations have different measurement errors
- Restricted multiplicative model to account for HV

Comparison of multiplicative heterogeneous variance adjustment models for genetic evaluations.
Márkus et al., 2013. J. Anim. Breed. Genet. 131:

Estimation of Direct Genomic Values

SNP-BLUP

Meuwissen et al. (2001. Genetics 157:)

$y = \mu + \beta_1 g_1 + \beta_2 g_2 + \ldots + \beta_n g_n + e$

- Modeling of marker effects
  - fixed
  - or random
- For random marker effects
  - either common variance
  - or individual variance for each marker

G-BLUP

VanRaden (2008. J. Dairy Sci. 91:)

$y = \mathbf{1}_n\mu + \mathbf{Z}_u\mathbf{u} + \mathbf{e}$,

$\mathbf{u} \sim N(\mathbf{0}, \mathbf{G}\sigma^2_g)$, where $\mathbf{G} = \mathbf{Z}\mathbf{Z}^T$

- Simple G-BLUP
- G-BLUP with polygenic effect
- G inverse has to be provided
  - Different alternatives for setting up **G**

Estimation of Genomic Enhanced EBV (GEBV)

Genomic selection project 2009-2013

Single-step GBLUP

Aguilar et al. (2010. J. Dairy Sci.93:) , **Christensen and Lund** (2010. Gen.Sel.Evol.42:)

- Implemented as classical BLUP model where $\mathbf{H}^{-1}$ is accommodate by reading $\mathbf{C}_{GA}$ from an external file

$$\mathbf{C}_{\mathrm{GA}} = \mathbf{G}_{\mathrm{w}}^{-1} - \mathbf{A}_{11}^{-1}, where\ \mathbf{G}_{\mathrm{w}} = wt\mathbf{G}_{\mathrm{k}} + (1-w)\mathbf{A}_{11}$$

Single Step Genomic Evaluations for the Nordic Red Dairy Cattle Test Day Data. Koivula et al. 2012. Interbull Bulletin 46:

- Implemented without setting up $\mathbf{G}^{-1}$

  modified approach based on an idea by **Legarra and Ducrocq** (2010, J. Dairy Sci. 95:)

- Future implementations avoiding matrix inversions are under consideration

Comparison of Some Equivalent Equations to Solve Single-Step GBLUP. Strandén & Mäntysaari, 2014, WCGALP

Useful accessories

Matti Taskinen joined

- **abc_MiX**

  Software to fit BayesA, BayesB and other Bayesian genomic models
- **HGinv**

  Software to set up and invert genomic relationship matrices
- **RelaX2**

  Software for pedigree manipulation
  - Pruning
  - Inbreeding coefficients
  - Genetic contributions
  - etc.

git repository

Software version control system

- MiX99 code consists of almost 100 000 lines
- Administration of MiX99 software
  - Master branch
  - Develop branches
  - Feature branches
  - Release branch
- Simultaneous programming work by MiX99 team members

New needs … new solutions…

Currently, most effort is on genomic prediction, but not only:

- Implementation of **effective record contributions** into **ApaX** for the need of **bivariate blending GBLUP**…
- Single-step without matrix inversion…
- Genomic prediction for admixed populations …
- Improved pre-conditioner matrices …
- MC-AI-REML …

Comparison of Breeding Values from Single-Step and Bivariate Blending Methods. Taskinen et al. 2014. WCGALP

Use of random regression model as an alternative for multibreed relationship matrix. Strandén & Mäntysaari 2013. Anim. Breed. Genet. 130:

Employing a Monte Carlo algorithm in Newton-type methods for restricted maximum likelihood estimation of genetic parameters. Matilainen et al. 2013. PLoS ONE 8(12):

… and more and more MiX99 code

```
Call Csol(                                          &  ! Calculate C*sol
                  Psol, Csolut, Lsol,               &  ! and rhs
                  Presidual, Cresidual, Lresidual, &
                  0, TraitPattern, buffer4, buffer4i, buffer1, buffer2, Yc, &
                  Phelpv, Chelpv, Lhelpv)

      endif ! if (SolinFile)

!     call MiX99phdX('iodpcgHV: Going Distribute', my_id)

! Calculate first (!) residual, then distribute help-vector !
      do i=Eq_base+1,Eq_high
        Presidual(i)=Phelpv(i)-Presidual(i)
      enddo
      do i=CommonLow,neq
        Cresidual(i)=Chelpv(i)-Cresidual(i)
      enddo
      do i=1,Nindex
        Lresidual(i)=Lhelpv(i)-Lresidual(i)
      enddo

      open(77,file=HVprogress,form='formatted',position='append')
      write (77,*) ' distribute rhs ... '
      close(77)

!     distribute the helpv, ie, the right hand side
      Call Distribute(Phelpv, Chelpv, Lhelpv)

      open(77,file=HVprogress,form='formatted',position='append')
      write (77,*) ' distribute residual ... '
      close(77)

!     call MiX99phdX('iodpcgHV: Going Distribute', my_id)
!     distribute the residual
      Call Distribute(Presidual, Cresidual, Lresidual)

!     call MiX99phdX('iodpcg: Going ListZero', my_id)
      Call ListZero(Psearch_d, Csearch_d, Lsearch_d)

      open(77,file=HVprogress,form='formatted',position='append')
      write (77,*) ' calculate preconditioner * rhs ... '
      close(77)
```

```
        allocate(Bbuffer2(Blimit2))
        allocate(Bblocks(Bdumps))

        read(87) (Bblocks(i),i=1,Bdumps)
        close (87)
      else
        print*,'Cannot open file "B1/ARlog" with block-structure of B-model
        stop
      endif

!   get information about the trait subgroup structure
      open(unit=64,file='Tralog',form='formatted',status='old',action='read
      read(64,*)
      do i=1,nt
         read(64,*)  (dummy(j),j=1,ntrgr) ! reading not needed in mix99p.f
      enddo
srch: do
        if (levcod/=a_current%code) then
           bpos=bpos+1
           if (btest(levcod,bpos)) then
              if (.not.associated(a_current%left)) then
                 nlev=nlev+1
                 new=.true.
                 allocate(a_new)
                 a_new%code=levcod
                 nullify(a_new%left,a_new%right,a_new%next)
                 a_current%left=>a_new
                 if (root==1) then
                   a_last1%next=>a_new
                   a_last1=>a_new
                 elseif (root==2) then
                   a_last2%next=>a_new
                   a_last2=>a_new
                 elseif (root==3) then
                   a_last3%next=>a_new
                   a_last3=>a_new
                 endif
                 exit srch
              else
                 a_current=>a_current%left
                 cycle srch
              endif
           else
```

THANK YOU