Genet. Sel. Evol. 40 (2008) 3–24
© INRA, EDP Sciences, 2008
DOI: 10.1051/gse:2007032

Available online at:
www.gse-journal.org

Original article

Parameter expansion for estimation of reduced rank covariance matrices *(Open Access publication)*

Karin MEYER*

Animal Genetics and Breeding Unit**, University of New England,
Armidale NSW 2351, Australia

(Received 14 December 2006; accepted 25 June 2007)

Abstract – Parameter expanded and standard expectation maximisation algorithms are described for reduced rank estimation of covariance matrices by restricted maximum likelihood, fitting the leading principal components only. Convergence behaviour of these algorithms is examined for several examples and contrasted to that of the average information algorithm, and implications for practical analyses are discussed. It is shown that expectation maximisation type algorithms are readily adapted to reduced rank estimation and converge reliably. However, as is well known for the full rank case, the convergence is linear and thus slow. Hence, these algorithms are most useful in combination with the quadratically convergent average information algorithm, in particular in the initial stages of an iterative solution scheme.

restricted maximum likelihood / reduced rank estimation / algorithms / expectation maximisation / average information

1. INTRODUCTION

Restricted maximum likelihood (REML) is one of the preferred methods for estimation of genetic parameters in animal breeding applications. Algorithms available to locate the maximum of the likelihood function differ in efficiency, computational requirements, ease of implementation and sensitivity to starting values in iterative schemes. The so-called 'average information' algorithm has been found to be highly effective, often converging in few rounds of iteration [40]. However, there have been some, albeit largely anecdotal, observations of convergence problems for analyses with 'bad' starting values, many

* Corresponding author: kmeyer@didgeridoo.une.edu.au
** AGBU is a joint venture between the NSW Department of Primary Industries and the University of New England.

random effects or large numbers of traits. On the other hand, 'expectation-maximisation' (EM) type methods are noted for their stability, yielding estimates within the parameter space and an increase in likelihood with each iterate. Unfortunately, these desirable features often come at the price of rather slow convergence rates.

Over the last decade or so, a number of new, 'fast' EM procedures have been proposed. Of particular interest is the PX-EM or 'parameter expanded' algorithm of Liu *et al.* [20]. Foulley and van Dyk [6] considered its application for several types of mixed model analyses, demonstrating a dramatic increase in speed of convergence over the standard EM algorithm. Yet, there has been virtually no practical use in variance component estimation so far.

Covariance matrices in multivariate analyses by and large have been treated as 'unstructured', *i.e.* apart from symmetry and requiring eigenvalues to be non-negative, no further assumption is made. There has been growing interest, however, in analyses considering the leading 'factors' or 'principal components' of a set of correlated effects only. As discussed by Kirkpatrick and Meyer [16], omitting any factors explaining negligible variation reduces the number of parameters to be estimated, yielding a highly parsimonious model. The resulting estimates of covariance matrices then have a factor-analytic structure *e.g.* [15] or, assuming specific variances are zero, have reduced rank (RdR). Average information algorithms for these scenarios have been described by Thompson *et al.* [39] and Meyer and Kirkpatrick [29], respectively.

On closer inspection, it is evident that the PX-EM algorithm [20] involves a reparameterisation of the standard, linear mixed model of the same form as REML algorithms to estimate RdR covariance matrices [29]. This can be exploited to obtain EM type estimators for factorial and RdR models. After a brief review of pertinent algorithms, this paper extends the approach of Foulley and van Dyk [6] to EM and PX-EM estimation for models fitting the leading principal components only. Convergence behaviour of the resulting algorithms is examined for a number of practical examples, and contrasted to that of the average information algorithm.

2. REVIEW

Maximum likelihood estimation of variance components almost invariably represents a constrained optimisation problem which needs to be solved iteratively [8].

2.1. Average information algorithm

A widely used optimisation procedure is the Newton-Raphson (NR) algorithm. It utilises both first and second derivatives of the function to be optimised, and thus provides an efficient search strategy *e.g.* [35]. A particular variant of NR used in REML analyses is the 'average information' (AI) algorithm, proposed by Thompson and co-workers (see [40]), which replaces second derivatives of $\log \mathcal{L}$ by the average of observed and expected values. NR algorithms perform unconstrained optimisation while REML estimates are required to be within the bounds of the parameter space [8]. Fortunately, constraints are readily implemented by estimating functions of the variance components for which the parameter space is not limited. Pinheiro and Bates [36] compare several options. The most commonly used is a parameterisation to the elements of the Cholesky decompositions of the covariance matrices, taking logarithmic values of the diagonal elements [19, 31]. As well as enforcing permissible estimates, this can improve rates of convergence of iterative maximisation schemes [7, 24]. In addition, NR type algorithms do not guarantee $\log \mathcal{L}$ to increase. While an initial, small step in the 'wrong direction' might result in a better position for subsequent steps, NR algorithms frequently do not recover from steps away from the maximum of $\log \mathcal{L}$ ($\log \mathcal{L}_{max}$). The step size in a NR iterate is proportional to the product of the inverse of the information (or AI) matrix and the vector of first derivatives of $\log \mathcal{L}$. A simple modification to control 'overshooting' is to reduce the step size until an increase in $\log \mathcal{L}$ is achieved.

Optimisation theory divides the convergence of NR algorithms into two phases [1]: Phase I comprises iterates sufficiently far away from $\log \mathcal{L}_{max}$ that step sizes need to be 'damped' to increase $\log \mathcal{L}$. Convergence in this phase is generally at least linear. Jennrich and Sampson [14] suggested a simple strategy of successive 'step halving' for this purpose. More sophisticated, 'backtracking' line search algorithms are available which attempt to optimise step sizes and guarantee convergence; see, for instance, Boyd and Vandenberghe [1], Chapter 9. In particular, Dennis and Schnabel [4] describe a quadratic approximation to choose a scale factor τ. Utilising derivatives of $\log \mathcal{L}$ yields an estimate of τ without the need for an additional function evaluation. If this step size fails to improve $\log \mathcal{L}$, updates can be obtained using a cubic approximation. Phase II, the 'pure' Newton phase, is reached when no further step size modifications are required. Typically, this phase shows quadratic convergence rates and involves relatively few iterates.

In addition, successful optimisation *via* NR algorithms requires the Hessian matrix (or its approximation) to be positive definite. While this is guaranteed

for the AI matrix, which is a matrix of sums of squares and crossproducts, it can have eigenvalues close to zero or a large condition number (*i.e.* ratio of largest to smallest eigenvalue). Such ill-conditioning can result in a vector of overly large step sizes which, in turn, may need excessive scaling ($\tau \ll 1$) to enforce an increase in $\log \mathcal{L}$, and thus hamper convergence. It is then advisable to modify the Hessian to ensure that it is 'safely' positive definite. Strategies based on the Cholesky decomposition of the Hessian matrix have been described [5, 37] that are suitable for large optimisation problems. For problems small enough to compute the eigenvalues of the Hessian matrix, we can directly modify the vector of eigenvalues and compute a corresponding modified Hessian matrix, or add a small multiple of the identity matrix. The latter results in an update of the parameters intermediate between that from a NR step and a method of steepest descent algorithm. Choices of modification and for minimum eigenvalues are discussed by Nocedahl and Wright [35], Chapter 6.

2.2. Expectation maximisation algorithm

A widely used alternative to NR for maximum likelihood estimation is the EM algorithm, described by Dempster *et al.* [3]. It involves computing the expectation of the (log) likelihood, pretending any 'missing data' are known, the so-called E-step. Secondly, in the M-step, this expectation is maximised with respect to the parameters to be estimated; see, for example, Ng *et al.* [34] for an exposé, or McLachlan and Krishnan [21] for an in-depth treatment. The popularity of the EM type algorithm is, in part at least, due to its property of monotone convergence under fairly general conditions, *i.e.* that the likelihood increases in each iterate. In addition, for variance component problems based on the linear, mixed model, estimates are guaranteed to be within the parameter space, and terms in the estimators are usually much easier to calculate than those for NR type methods. An early formulation for an EM type algorithm to estimate covariances for multiple trait models has been presented by Henderson [11].

The main disadvantage of EM type algorithms is that they can be rather slow to converge. While NR methods are expected to exhibit quadratic rates of convergence, EM algorithms are expected to converge linearly [34]. This behaviour has motivated numerous modifications of the basic EM algorithm, aimed at improving its rate of convergence. In the simplest cases, it is attempted to predict changes in parameters based on changes over the past iterates, *e.g.* the 'accelerated EM' [17], which employs a multivariate form of Aitken acceleration. Other modifications involve approximations to derivatives

of the likelihood to yield Quasi-Newton *e.g.* [13, 22] or gradient type procedures *e.g.* [12, 18]. In addition, several generalised EM type algorithms have been proposed over the last decade. Strategies employed in these include maximisation of the likelihood conditional on subsets of the parameters, switching between the complete and observed likelihoods, or alternating between schemes to augment the observed by the missing data; see Meng and van Dyk [23] for a review.

Less attention has been paid to the effects of choice of parameterisation on convergence behaviour of EM type algorithms. Thompson and Meyer [38] showed that estimation of linear functions of variance components, similar in form to mean squares between random effects in balanced analyses of variance, instead of the variance components could dramatically improve convergence of the EM algorithm. While a reparameterisation to the non-zero elements of Cholesky factors of covariance matrices is routinely used with NR and Quasi-Newton type algorithms *e.g.* [31,33], this has found virtually no use in practical EM estimation of variance components. Largely this is due to the fact that estimates are ensured to be within the parameter space, so that there is no pressing need for a reparameterisation.

Lindstrom and Bates [19] described an EM algorithm for maximum likelihood and REML estimation in linear mixed models which utilised the Cholesky factorisation of the covariance matrices to be estimated. More recently, Meng and van Dyk [24] and van Dyk [41] proposed EM type algorithms which transformed the vector of random effects in the mixed model to a vector with diagonal covariance matrix, showing that substantial reductions in numbers of iteration could be achieved. The transformation utilised was the inverse of the Cholesky factor of the covariance matrix among random effects, and parameters estimated were the elements of the Cholesky factor.

2.3. Parameter expansion

Probably the most interesting proposal among the modern ‘fast’ EM type methods is the Parameter Expanded (PX) algorithm of Liu *et al.* [20]. Like the approach of Meng and van Dyk [24] it involves conceptual rescaling of the vector of random effects. However, there are no specific assumptions about the structure of the matrix α defining the transformation. Liu *et al.* [20] considered application of PX-EM for a number of examples, including a random coefficient, mixed model. Foulley and van Dyk [6] derived detailed formulae for PX-EM based on the standard mixed model equations for common univariate

models. As for the standard EM algorithm, the likelihood is ensured to increase in each iterate of the PX-EM algorithm [20].

Briefly, the basic procedure for PX-EM estimation of variance components is as follows [6]: The E-step of the PX-EM algorithm is the same as for standard EM. Similarly, in the first part of the M-step, covariance matrices for random effects, $\mathbf{\Sigma}$, are estimated 'as usual', *i.e.* assuming $\boldsymbol{\alpha}$ is equal to an identity matrix. Subsequently, the elements of $\boldsymbol{\alpha}$ are estimated as additional parameters – this represents the expansion of the parameter vector. However, expansion is only temporary: pre- and postmultiplying the estimate of $\mathbf{\Sigma}$ by $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\alpha}}'$, respectively, then yields an updated estimate of $\mathbf{\Sigma}$, effectively collapsing the parameter vector again to its original size. Finally, estimates of the residual covariances are obtained as in the standard EM algorithm, after adjusting estimates of random effects for $\hat{\boldsymbol{\alpha}}$.

For most algorithms, computational requirements of REML estimation increase with the number of parameters, both per iterate and overall. Hence it seems somewhat counter-intuitive to estimate a substantial number of additional parameters. For instance, if we have q traits in a multivariate analysis, there are $q(q+1)/2$ elements of $\mathbf{\Sigma}$ to be estimated and, making no assumptions about the structure of $\boldsymbol{\alpha}$, an additional q^2 elements of $\boldsymbol{\alpha}$. However, the PX-EM algorithm can yield dramatically faster convergence than the standard EM algorithm [6, 20].

Loosely speaking, the efficacy of the PX-EM algorithm can be attributed to the additional parameters capturing 'information' which is not utilised in the standard EM algorithm. In each iterate of the EM algorithm we treat the current values of the parameters as if they were the 'true' values, *i.e.* the values maximising the likelihood. Hence, before convergence, in the E-step the 'missing data' are imputed and the expectation of the complete likelihood is computed with error. This error is larger, the further away we are from $\log \mathcal{L}_{max}$. The deviation of $\hat{\boldsymbol{\alpha}}$ from the identity matrix gives a measure of the error. Adjusting the estimate of $\mathbf{\Sigma}$ for $\hat{\boldsymbol{\alpha}}$ effectively involves a regression of the vector of parameters on the vector of differences between $\hat{\boldsymbol{\alpha}}$ and its assumed value in the E-step. Liu *et al.* [20] described this as a 'covariance adjustment'.

3. ALGORITHMS

3.1. Standard EM

Consider the standard linear, mixed model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \mathbf{e} \tag{1}$$

with $\mathbf{y}$, $\boldsymbol{\beta}$, $\mathbf{u}$ and $\mathbf{e}$ denoting the vectors of observations, fixed effects, random effects and residuals, respectively, and $\mathbf{X}$ and $\mathbf{Z}$ the corresponding incidence matrices.

The model given by (Eq. 1) is general and encompasses multiple random effects, as well as standard multivariate and random regression models. However, for simplicity of presentation, let $\mathbf{u}$ represent a single random effect for q traits, with subvectors $\mathbf{u}_i$ for $i = 1, \ldots, q$ and covariance matrix $\mathbf{G} = \boldsymbol{\Sigma}_U \otimes \mathbf{A}$. For $\mathbf{u}$ representing animals' genetic effects, $\mathbf{A}$ is the numerator relationship matrix. $\boldsymbol{\Sigma}_U$ is the $q \times q$ covariance matrix between random effects with elements $\sigma_{U\,ij}$, and $\otimes$ denotes the direct matrix product. Assume $\mathbf{u}$ and $\mathbf{e}$ are uncorrelated, and let $\mathrm{Var}(\mathbf{e}) = \mathbf{R}$. Further, let $\boldsymbol{\Sigma}_E$ be the matrix of residual covariances with elements $\sigma_{E\,ij}$ for $i, j = 1, \ldots, q$. Ordering $\mathbf{e}$ according traits within individuals, $\mathbf{R}$ is block-diagonal with the k-th block equal to the submatrix of $\boldsymbol{\Sigma}_E$ corresponding to the traits recorded for individual k.

This gives the vector of parameters to be estimated, $\boldsymbol{\theta}' = (\mathrm{vech}\,(\boldsymbol{\Sigma}_U)' \; |\mathrm{vech}\,(\boldsymbol{\Sigma}_E)')$ of length p (with vech the operator which stacks the columns in the lower triangle of a symmetric matrix into a vector *e.g.* [9]). Standard formulation considers the likelihood of $\boldsymbol{\theta}$, given the data. Vectors $\mathbf{u}$ and $\boldsymbol{\beta}$ in (Eq. 1) cannot be observed and are thus treated as 'missing data' in the EM algorithm. In the E-step, we need to compute the expectation of the complete data log likelihood ($\log Q$), *i.e.* the likelihood of $\boldsymbol{\theta}$ given $\mathbf{y}$, $\boldsymbol{\beta}$ and $\mathbf{u}$. This can be split into a part due to the random effects, $\mathbf{u}$, and a part due to residuals, $\mathbf{e}$, [6],

$$\begin{aligned} \log Q &= -\frac{1}{2}\Big(const. + E[\log|\mathbf{G}| + \mathbf{u}'\mathbf{G}^{-1}\mathbf{u} + \log|\mathbf{R}| + \mathbf{e}'\mathbf{R}^{-1}\mathbf{e}]\Big) \\ &= const. + \log Q_U + \log Q_E \end{aligned} \tag{2}$$

with $\mathbf{e} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{u}$. Each part comprises a quadratic form in the respective random vector and the inverse of its covariance matrix, and the log determinant of the latter. Strictly speaking, (Eq. 2) (and the following equations) should be given conditional on $\boldsymbol{\theta}$ being equal to some current value, $\boldsymbol{\theta}^t$, but this has been omitted for clarity; see, for instance, Foulley and van Dyk [6] or Ng *et al.* [34] for more rigorous formulations.

In the M-step, we take first derivatives of $\log Q$ with respect to the elements of $\boldsymbol{\theta}$, θ_k. The resulting expressions are equated to zero and solved for θ_k, $k = 1, \ldots, p$.

3.1.1. Random effects covariances

For $\theta_k = \sigma_{U\,ij}$ and $\mathbf{\Delta}^U_{ij} = \partial\mathbf{\Sigma}_U/\partial\sigma_{U\,ij}$,

$$\frac{\partial \log Q_U}{\partial \sigma_{U\,ij}} = -\frac{1}{2}\Big(\mathrm{tr}\left(\mathbf{\Sigma}_U^{-1}\mathbf{\Delta}^U_{ij}\otimes\mathbf{I}\right) - E[\hat{\mathbf{u}}'\left(\mathbf{\Sigma}_U^{-1}\mathbf{\Delta}^U_{ij}\mathbf{\Sigma}_U^{-1}\otimes\mathbf{A}^{-1}\right)\hat{\mathbf{u}}]\Big) = 0. \quad (3)$$

Matrix $\mathbf{\Delta}^U_{ij}$ has elements of unity in position i, j and j, i, and zero otherwise. With all subvectors of $\mathbf{u}$ of the same length, N_U, and using that $E[\hat{\mathbf{u}}_i'\mathbf{A}^{-1}\hat{\mathbf{u}}_j] = \hat{\mathbf{u}}_i'\mathbf{A}^{-1}\hat{\mathbf{u}}_j + \mathrm{tr}\left(\mathbf{A}^{-1}\mathbf{C}^{UU}_{ij}\right)$, we obtain – after some rearranging – the well known estimators [11]

$$\hat{\sigma}_{U\,ij} = (\hat{\mathbf{u}}_i'\mathbf{A}^{-1}\hat{\mathbf{u}}_j + \mathrm{tr}\left(\mathbf{A}^{-1}\mathbf{C}^{UU}_{ij}\right))/N_U \quad (4)$$

where $\mathbf{C}$ is the inverse of the coefficient matrix in the mixed model equations (MME) pertaining to (Eq. 1), and $\mathbf{C}^{UU}_{ij}$ is the submatrix of $\mathbf{C}$ corresponding to the vectors of random effects for traits i and j, $\mathbf{u}_i$ and $\mathbf{u}_j$.

3.1.2. Residual covariances

Similarly, estimators for the residual covariances $\sigma_{E\,ij}$ are obtained setting $\partial \log Q_E/\partial\sigma_{E\,ij} = 0$. Inserting $\mathbf{R}^{-1}\mathbf{R}$ into the trace term (in Eq. 3) and rearranging, yields [11]

$$\mathrm{tr}\left(\mathbf{E}_{ij}\mathbf{R}\right) = \hat{\mathbf{e}}'\mathbf{E}_{ij}\hat{\mathbf{e}} + \mathrm{tr}\left(\mathbf{E}_{ij}\mathbf{W}\mathbf{C}\mathbf{W}'\right) \quad (5)$$

with $\mathbf{E}_{ij} = \mathbf{R}^{-1}\left(\partial\mathbf{R}/\partial\sigma_{E\,ij}\right)\mathbf{R}^{-1}$ and $\mathbf{W} = (\mathbf{X}\,\mathbf{Z})$.

Expand $\mathbf{\Sigma}_E$ as $\sum_{m=1}^{q}\sum_{n=m}^{q}\mathbf{\Delta}^E_{mn}\sigma_{E\,mn}$, with $\mathbf{\Delta}^E_{mn} = \partial\mathbf{\Sigma}_E/\partial\sigma_{E\,mn}$. Using that $\mathbf{R}$ is block-diagonal, we can then rewrite the left hand side of (Eq. 5) as

$$\mathrm{tr}\left(\mathbf{E}_{ij}\mathbf{R}\right) = \sum_{m=1}^{q}\sum_{n=m}^{q}\sum_{k=1}^{N}\mathrm{tr}\Big(\mathbf{\Sigma}_E^{-1}(\mathbf{\Delta}^E_{ij})^k\,\mathbf{\Sigma}_E^{-1}(\mathbf{\Delta}^E_{mn})^k\Big)\sigma_{E\,mn} = \sum_{m=1}^{q}\sum_{n=m}^{q}F^E_{ij,mn}\,\sigma_{E\,mn} \quad (6)$$

with N the number of individuals, and $(\mathbf{\Delta}^E_{ij})^k$ for the k-th individual equal to $\mathbf{\Delta}^E_{ij}$ with rows and columns pertaining to traits not recorded set to zero. Likewise, the right hand side of (Eq. 5) can be accumulated across individuals,

$$\begin{aligned}\hat{\mathbf{e}}'\mathbf{E}_{ij}\hat{\mathbf{e}} + \mathrm{tr}\left(\mathbf{E}_{ij}\mathbf{W}\mathbf{C}\mathbf{W}'\right) = \sum_{k=1}^{N}\mathrm{tr}\Big(\Big(&\hat{\mathbf{e}}^k\hat{\mathbf{e}}^{k'} + \mathbf{X}^k\mathbf{C}^{XX}\mathbf{X}^{k'} + \mathbf{X}^k\mathbf{C}^{XU}\mathbf{Z}^{k'} \\ &+ \mathbf{Z}^k\mathbf{C}^{UX}\mathbf{X}^{k'} + \mathbf{Z}^k\mathbf{C}^{UU}\mathbf{Z}^{k'}\Big)\mathbf{\Sigma}_E^{-1}\left(\mathbf{\Delta}^E_{ij}\right)^k\mathbf{\Sigma}_E^{-1}\Big) = t^E_{ij}\end{aligned} \quad (7)$$

with $\mathbf{X}^k$, $\mathbf{Z}^k$ and $\mathbf{e}^k$ the sub-matrices and -vector of $\mathbf{X}$, $\mathbf{Z}$ and $\mathbf{e}$, respectively, for the k-th individual. This yields a system of $q(q+1)/2$ linear equations to be solved to obtain estimates of $\boldsymbol{\theta}_E = \text{vech}(\boldsymbol{\Sigma}_E)$

$$\hat{\boldsymbol{\theta}}_E = \mathbf{F}_E^{-1}\mathbf{t}_E \tag{8}$$

with elements $F^E_{ij,mn}$ and t^E_{ij} of $\mathbf{F}_E$ and $\mathbf{t}_E$ as defined in (Eq. 6) and (Eq. 7), respectively.

3.2. PX-EM

For the 'Parameter Expanded' EM algorithm, (Eq. 1) is reparameterised to

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}(\mathbf{I}\otimes\boldsymbol{\alpha})\mathbf{u}^+ + \mathbf{e} \tag{9}$$

with $\text{Var}(\mathbf{u}^+) = \boldsymbol{\Sigma}^+_U \otimes \mathbf{A}$. The elements of $\boldsymbol{\alpha}$ represent the additional parameters to be estimated, *i.e.* the expanded parameter vector is $\boldsymbol{\Theta}' = \left(\text{vech}(\boldsymbol{\Sigma}^+_U)' \,|\text{vech}(\boldsymbol{\Sigma}_E)' \,|\text{vec}(\boldsymbol{\alpha})'\right)$ (with vec the operator which stacks the columns of a matrix into a vector [9]). Depending on assumptions on the structure of $\boldsymbol{\alpha}$, there are up to q^2 additional parameters.

In the E-step, $\log Q$ is conditioned on $\boldsymbol{\alpha} = \boldsymbol{\alpha}_0$. Choosing $\boldsymbol{\alpha}_0 = \mathbf{I}$, the E-step is identical to that described above for the standard EM algorithm, *i.e.* the difference between $\mathbf{u}^+$ and $\mathbf{u}$ is merely conceptual. This implies that steps to set up and manipulate the MME are largely 'as usual', making implementation of the PX-EM algorithm a straightforward extension to standard EM. For the reparameterised model (Eq. 9), $\mathbf{e} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}(\mathbf{I}\otimes\boldsymbol{\alpha})\mathbf{u}^+$. Hence, for $\boldsymbol{\Theta}_k = \alpha_{ij}$ only derivatives of $\log Q_E$ are non-zero. For unstructured $\boldsymbol{\alpha}$, $\partial \log Q_E / \partial \alpha_{ij}$ has a single non-zero element of unity in position i, j. As shown by Foulley and van Dyk [6], equating derivatives to zero then yields – after some manipulations – a linear system of q^2 equations to be solved, $\hat{\boldsymbol{\theta}}_\alpha = \mathbf{F}_\alpha^{-1}\mathbf{t}_\alpha$ with $\boldsymbol{\theta}_\alpha = \text{vec}(\boldsymbol{\alpha})$. Elements of $\mathbf{F}_\alpha$ and $\mathbf{t}_\alpha$ are

$$F^\alpha_{ij,mn} = \text{tr}\left(\mathbf{Z}_j'\mathbf{R}^{-1}\mathbf{Z}_n\left(\hat{\mathbf{u}}^+_m(\hat{\mathbf{u}}^+_i)' + \mathbf{C}^{UU}_{mi}\right)\right) \tag{10}$$

$$t^\alpha_{ij} = \hat{\mathbf{u}}^+_i\mathbf{Z}_j'\mathbf{R}^{-1}\mathbf{y} - \text{tr}\left(\mathbf{Z}_j'\mathbf{R}^{-1}\mathbf{X}\left(\hat{\boldsymbol{\beta}}(\hat{\mathbf{u}}^+)' + \mathbf{C}^{XU}_i\right)\right) \tag{11}$$

where $\mathbf{u}^+_i$ and $\mathbf{Z}_i$ denote the subvector and -matrix of $\mathbf{u}^+$ and $\mathbf{Z}$, respectively, for trait i, and $\mathbf{C}^{XU}_i$ is the submatrix of $\mathbf{C}$ corresponding to the fixed effects and random effects levels for trait i.

$\boldsymbol{\Sigma}^+_U$ is estimated assuming $\boldsymbol{\alpha} = \mathbf{I}$, *i.e.* estimators are as given in Section 3.1.1 (replacing $\sigma_{U\,ij}$ with $\sigma^+_{U\,ij}$). Similarly, estimates of the residual covariances

are obtained as for the standard EM algorithm (Sect. 3.1.2). Foulley and van Dyk [6] recommended to use $\hat{\mathbf{e}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{Z}(\mathbf{I} \otimes \hat{\boldsymbol{\alpha}})\hat{\mathbf{u}}^{+}$, *i.e.* to adjust for the current estimate $\hat{\boldsymbol{\alpha}} \neq \mathbf{I}$. The M-step is completed by obtaining estimates for $\boldsymbol{\Sigma}_U$, collapsing $\boldsymbol{\Theta}$ into $\boldsymbol{\theta}$. The reduction function is $\hat{\boldsymbol{\Sigma}}_U = \hat{\boldsymbol{\alpha}}\hat{\boldsymbol{\Sigma}}_U^{+}\hat{\boldsymbol{\alpha}}'$ [20].

3.3. Reduced rank estimation

Considering the direct estimation of principal components (PCs), Meyer and Kirkpatrick [29] reparameterised (Eq. 1) to

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}(\mathbf{I} \otimes \mathbf{Q})\mathbf{u}^{\star} + \mathbf{e} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}^{\star}\mathbf{u}^{\star} + \mathbf{e}. \tag{12}$$

The eigenvalue decomposition of the covariance matrix among random effects is $\boldsymbol{\Sigma}_U = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}'$, with $\mathbf{E}$ the matrix of eigenvectors of $\boldsymbol{\Sigma}_U$ and $\boldsymbol{\Lambda}$ the diagonal matrix of corresponding eigenvalues, λ_i. As it is standard practice, let eigenvectors and -values be in descending order of λ_i.

For $\mathbf{Q} = \mathbf{E}$, $\mathbf{u}^{\star}$ comprises random effect values for the PCs of the q traits considered. For $\mathbf{Q} = \mathbf{E}\boldsymbol{\Lambda}^{1/2}$, PCs are standardised to variances of unity and $\boldsymbol{\Sigma}_U = \mathbf{Q}\mathbf{Q}'$. This is the parameterisation used by Meyer and Kirkpatrick [29], who truncated $\mathbf{Q}$ to columns $1, \ldots, r < q$ to obtain reduced rank estimates of $\boldsymbol{\Sigma}_U$. A more convenient alternative is $\mathbf{Q} = \mathbf{L}$ with $\mathbf{L}$ the Cholesky factor of $\boldsymbol{\Sigma}_U$. This uses that $\mathbf{L} = \mathbf{E}\boldsymbol{\Lambda}^{1/2}\mathbf{T}$ with $\mathbf{T}\mathbf{T}' = \mathbf{I}$ [9]. Assuming that the Cholesky decomposition has been carried out pivoting on the largest diagonals, this implies that we can obtain reduced rank estimates of a matrix considering the leading PCs only, by estimating the non-zero elements of corresponding columns of $\mathbf{L}$.

At full rank (Eq. 12) gives an equivalent model to (Eq. 1). Truncating $\mathbf{Q}$ to the first $r < q$ columns, yields an estimate of $\boldsymbol{\Sigma}_U$ which has, at most, rank r. Clearly, (Eq. 12) is of the same form as (Eq. 9). However, there is a major conceptual difference: essentially, the roles of extra parameters and those of interest are reversed. The 'modifiers' of $\mathbf{Z}$ are now the parameters to be estimated, rather than auxiliary quantities. Conversely, the covariance matrix of random effects, $\mathrm{Var}(\mathbf{u}^{\star})$ is assumed to be an identity matrix for standard EM and AI REML algorithms. In a PX-EM algorithm, these covariances are estimated as additional parameters, $\mathrm{Var}(\mathbf{u}^{\star}) = \boldsymbol{\alpha}^{\star}$, which is symmetric with $r(r+1)/2$ elements $\alpha_{ij}^{\star}$.

3.3.1. Random effects parameters

The mechanics of taking derivatives of $\log Q_E$ with respect to the elements of $\mathbf{Q}$ are analogous to those for α_{ij} in the full rank PX-EM algorithm. However,

there is no conditioning on $\mathbf{Q} = \mathbf{Q}_0 = \mathbf{I}$. Consequently, we need to distinguish MME involving $\mathbf{Z}$ and $\mathbf{Z}^\star$. For generality, let $\Theta_k = f\left(q_{ij}\right)$ where q_{ij} is the ij-th element of $\mathbf{Q}$ and $f(\cdot)$ is some function of q_{ij} (but not involving any other elements of $\mathbf{Q}$). This gives a matrix of derivatives $\mathbf{\Delta}^Q_{ij} = \partial\mathbf{Q}/\partial\Theta_k$ which has a single non-zero element $\omega_{ij} = \partial q_{ij}/\partial f\left(q_{ij}\right)$ in position i, j. In most cases, ω_{ij} is unity. However, if we choose to take logarithmic values of the diagonal elements of $\mathbf{L}$, $\omega_{ii} = \log(q_{ii})$.

For $\partial\mathbf{Z}^\star/\partial\Theta_k = \mathbf{Z}\left(\mathbf{I}\otimes\mathbf{\Delta}^Q_{ij}\right)$,

$$\frac{\partial \log Q_E}{\partial\Theta_k} = E\Big[\omega_{ij}\hat{\mathbf{u}}^{\star\prime}\left(\mathbf{I}\otimes\mathbf{\Delta}^Q_{ij}\right)'\mathbf{Z}'\mathbf{R}^{-1}\hat{\mathbf{e}}\Big]. \tag{13}$$

Using that $\hat{\mathbf{e}} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}^\star\mathbf{u}^\star$, expanding $\mathbf{Q}$ to $\mathbf{Q} = \sum_{m=1}^{r}\sum_{n=m}^{q}\mathbf{\Delta}^Q_{mn} f\left(q_{mn}\right)$ and equating (Eq. 13) to zero then yields, after some rearrangement,

$$\sum_{m=1}^{r}\sum_{n=m}^{q}\omega_{mn}\mathrm{tr}\Big(\mathbf{Z}_j'\mathbf{R}^{-1}\mathbf{Z}_n E\Big[\hat{\mathbf{u}}^\star_m\hat{\mathbf{u}}^{\star\prime}_i\Big]\Big)f\left(q_{mn}\right) = \hat{\mathbf{u}}^{\star\prime}_i\mathbf{Z}_j'\mathbf{R}^{-1}\mathbf{y} - \mathrm{tr}\Big(\mathbf{Z}_j'\mathbf{R}^{-1}\mathbf{X}E\Big[\hat{\boldsymbol{\beta}}\hat{\mathbf{u}}^{\star\prime}_i\Big]\Big) \tag{14}$$

with $\mathbf{u}^\star_i$ the subvector of $\mathbf{u}^\star$ for the i-th principal component. Subscript ranges, $i = 1,\ldots,r$ and $j = i,\ldots,q$ as well as $m = 1,\ldots,r$ and $j = m,\ldots,q$ in (Eq. 14), pertain to $\mathbf{Q}$ consisting of the first r columns of the Cholesky factor $\mathbf{L}$, and are readily adapted to other choices of $\mathbf{Q}$.

This gives a system of $r(2q - r + 1)/2$ linear equations to estimate $\boldsymbol{\theta}_Q$ consisting of the non-zero elements of vech $(\mathbf{Q})$,

$$\mathbf{F}_Q\,\hat{\boldsymbol{\theta}}_Q = \mathbf{t}_Q \tag{15}$$

with elements

$$F^Q_{ij,mn} = \omega_{mn}\mathrm{tr}\Big(\mathbf{Z}_j'\mathbf{R}^{-1}\mathbf{Z}_n\left(\hat{\mathbf{u}}^\star_m\hat{\mathbf{u}}^{\star\prime}_i + \mathbf{C}^{UU}_{mi}\right)\Big) \tag{16}$$

$$t^Q_{ij} = \hat{\mathbf{u}}^{\star\prime}_i\mathbf{Z}_j'\mathbf{R}^{-1}\mathbf{y} - \mathrm{tr}\Big(\mathbf{Z}_j'\mathbf{R}^{-1}\mathbf{X}\left(\hat{\boldsymbol{\beta}}\,\hat{\mathbf{u}}^{\star\prime}_i + \mathbf{C}^{XU}_i\right)\Big). \tag{17}$$

$\mathbf{C}$ in (Eq. 16) and (Eq. 17) is the inverse of the coefficient matrix in the MME pertaining to (Eq. 12), *i.e.* involving $\mathbf{Z}^\star$ rather than $\mathbf{Z}$, and with numbers of equations proportional to r rather than q, with submatrices as defined above. Similarly, $\mathbf{u}^\star_i$ and $\boldsymbol{\beta}$ are the (sub-)vectors of effects in (Eq. 12). Terms $\mathbf{Z}_j'\mathbf{R}^{-1}\mathbf{Z}_n$, $\mathbf{Z}_j'\mathbf{R}^{-1}\mathbf{X}$ and $\mathbf{Z}_j'\mathbf{R}^{-1}\mathbf{y}$, however, are submatrices and -vectors of the data part of coefficient matrix and right hand side of the mixed model equations on the 'original scale', *i.e.* pertaining to (Eq. 1). Hence, implementation of an EM algorithm for reduced rank estimation requires part of a second set of MME – proportional to the number of traits q – to be set up for each iterate.

Table I. Characteristics of the data structure and model for examples.

	Example 1	Example 2	Example 3
No. of traits or RR[a] coefficients	8	4	13
No. of records	20 171	8845	28 637
No. of animals in data	5605	3743	908
No. of animals in pedigree[b]	8044	3786	1150
Random effects fitted[c]	A	A, M, C	A, P
No. of covariance components[d]	56	40	194
Source	[29]	[32]	[26]

[a] Random regression.
[b] After pruning.
[c] A: Direct additive genetic, M: maternal additive genetic, P: direct permanent environmental, and C: maternal permanent environmental.
[d] For full rank analysis.

3.3.2. PX-EM: auxiliary parameters

Estimates of $\boldsymbol{\alpha}^\star$ can be obtained in the same way as the estimates of covariance components due to random effects in the standard EM algorithm (see Sect. 3.1.1 above).

$$\hat{\alpha}^\star_{ij} = \left(\hat{\mathbf{u}}_i^{\star\prime}\mathbf{A}^{-1}\hat{\mathbf{u}}_j^\star + \mathrm{tr}\left(\mathbf{A}^{-1}\mathbf{C}_{ij}^{UU}\right)\right)/N_U \tag{18}$$

for $i = 1, \ldots, r$ and $j = i, \ldots, r$, and with $\mathbf{C}$ as in (Eq. 16) and (Eq. 17).

Updated estimates of $\mathbf{Q}$ are then obtained as the first r columns of the Cholesky decomposition of $\hat{\mathbf{Q}}\hat{\alpha}^\star\hat{\mathbf{Q}}'$.

3.3.3. Residual covariances

Again, residual covariances are estimated as in the standard EM algorithm (Sect. 3.1.2), but with $\hat{\mathbf{e}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{Z}^\star\hat{\mathbf{u}}^\star$.

4. APPLICATION

4.1. Examples

The performance of algorithms described above was examined for three, relatively small practical examples analysed previously. Table I summarises characteristics of the data and analyses. Further details can be found in the respective publications.

Example 1 (from Meyer and Kirkpatrick [29]) consisted of four 'carcass traits' measured by live ultra-sound scanning of beef cattle in a single herd. Treating records for males and females as different traits, resulted in 8 traits in a multivariate analysis. With distinct subsets, the 16 residual covariances between traits measured on animals of different sex were zero. The model of analysis was a simple animal model, fitting animals' direct additive genetic effects as the only random effect.

Example 2 comprised records for birth, weaning, yearling and final weights of Polled Hereford cattle in the Wokalup selection experiment see [32]. While most animals had records for the first two weights, only replacement animals remaining in the herd after weaning had records for the later weights (35–40% of those with birth weight). The model of analysis fitted direct and maternal additive genetic effects, assuming direct-maternal covariances were zero, as well as maternal permanent environmental effects as random effects.

Example 3 considered repeated records for mature cow weights, also from the Wokalup selection experiment, taken between 19 and 84 months of age. Cows were weighed monthly, except during the calving season. This resulted in up to 63 records per animal, with 75% of cows having at least 13 records. With short mating and calving periods in the experiment, there was a strong association between age at and month of weighing. Previous analyses at the phenotypic level [25] thus had found a strong annual, cyclic pattern in both weights and variances. Hence, analyses fitted a random regression (RR) on quadratic B-splines of age at weighing, with 11 equi-distant knots at 6 months intervals resulting in 13 RR coefficients, for both additive genetic and permanent environmental effects of the animal. Measurement error variances were assumed to be heterogeneous with 12 classes, corresponding to the calendar month of recording.

4.2. Analyses

Full rank and RdR estimates of covariance matrices were obtained by REML, employing an AI, standard EM and PX-EM algorithm as well as a combination, consisting of 4 initial iterates of the PX-EM algorithm followed by AI (PX+AI). Residual covariance matrices were assumed to have full rank throughout. The same set of starting values for the covariance components to be estimated was used in all analyses for a particular example. Calculations were carried out using our REML program WOMBAT [28].

All analyses parameterised to the leading columns of the Cholesky decomposition of the covariance matrices to be estimated, pivoting on the largest

diagonal elements. PX-EM and standard EM-algorithms for RdR estimation were implemented as described above (Sect. 3.3). In calculating the sparse inverse of the coefficient matrix ($\mathbf{C}$), only the elements corresponding to the non-zero elements in the Cholesky factorisation of the original matrix were determined. Any other elements which might have been required to compute the terms in (Eq. 14) were treated as if they were zero. Convergence was assumed to have been reached when the change in $\log \mathcal{L}$ between iterates ($\boldsymbol{\Delta}\mathcal{L}$) was less than 10^{-6} or if the relative change in the vector of parameters to be estimated, $\sqrt{|\hat{\boldsymbol{\theta}}^t - \hat{\boldsymbol{\theta}}^{t-1}|/|\hat{\boldsymbol{\theta}}^t|}$, was less than 10^{-7} [6] (with $|\cdot|$ denoting the vector norm, and $\hat{\boldsymbol{\theta}}^t$ the estimate of $\boldsymbol{\theta}$ from iterate t).

The AI algorithm used was as described by Meyer and Kirkpatrick [29], but parameterising to the leading columns of Cholesky factors (see Sect. 3.3) and calculating the average information as described in the Appendix. Pivots were constrained to a minimum value of 10^{-6} and transformed to logarithmic scale if small values (< 0.2) were encountered during the course of iterations. In each iterate, $\log \mathcal{L}$ was forced to increase by scaling step sizes if necessary, using the line search procedure of Dennis and Schnabel [4]. In addition, the AI matrix was ensured to be 'safely' positive definite, by adding an appropriate multiple of the identity matrix to it, if the smallest eigenvalue was less than the minimum of 0.002 and $10^{-6} \times \lambda_1$, with λ_1 representing the largest eigenvalue of the AI matrix. The AI algorithm was deemed to have converged if the $\boldsymbol{\Delta}\mathcal{L} < 10^{-5}$ and the corresponding Newton decrement [1] was greater than -0.01.

4.3. Results

4.3.1. Example 1

Starting values for covariance components for Example 1 were the set of 'bad' values used by Meyer [28] to compare PX-EM, EM and AI algorithms for standard, full-rank multivariate REML analyses. These consisted of estimates from four-trait analyses for measures on females, repeated for males and all genetic covariances set to 0.01. Analyses were carried out fitting from $1, \ldots, 8$ principal components for additive genetic effects. Characteristics of the convergence patterns are summarised in Table II, and Figure 1 shows values of the relative log likelihood, *i.e.* $\log \mathcal{L}$ deviated from the highest value found across all corresponding analyses ($\log \mathcal{L}_{max}$), for selected numbers of PCs fitted. With very stringent convergence criteria, almost all analyses for a given number of PCs converged to the same value, up to the third decimal.

Table II. Number of iterates (N) needed and deviation of log likelihood ($\log \mathcal{L}$) from best value (D, multiplied by 1000) for change in $\log \mathcal{L}$ between iterates to reach a minimum value, and N for $\log \mathcal{L}$ to reach a given D, for Example 1.

Fit[a]		Change in $\log \mathcal{L}$ less than								Deviation less than		
		0.00001		0.00005		0.00010		0.00050		–0.20	–0.10	–0.05
		N	D	N	D	N	D	N	D	N	N	N
8	AI[b]	15	0[c]	14	0	13	0	11	–1	4	5	5
	PX+AI	46	–1	24	–1	18	–2	14	–2	7	8	8
	PX-EM	573	–1	374	–4	313	–8	205	–33	114	143	178
	EM	600	–1	401	–4	338	–9	221	–35	124	156	196
7	AI	10	0	9	0	8	0	7	0	4	4	4
	PX+AI	16	0	14	0	13	0	12	0	7	7	7
	PX-EM	601	–1	402	–4	338	–9	219	–36	120	153	195
	EM	604	–1	405	–4	342	–9	222	–36	122	156	198
5	AI	15	0	14	0	13	0	12	0	7	7	8
	PX+AI	16	0	14	0	14	0	13	0	8	8	9
	PX-EM	481	0	346	–2	301	–5	211	–26	115	144	177
	EM	499	0	364	–2	318	–6	225	–27	126	157	192
3	AI	76	0	71	0	68	0	63	–1	46	49	51
	PX+AI	40	0	35	0	33	0	28	–1	14	15	17
	PX-EM	571	0	367	–4	299	–8	172	–37	86	111	150
	EM	620	0	415	–4	348	–8	209	–40	105	142	191
2	AI	84	0	81	0	80	0	77	0	66	67	68
	PX+AI	49	0	45	0	44	0	41	0	30	31	32
	PX-EM	578	0	446	–2	402	–5	305	–28	195	232	271
	EM	595	0	464	–2	419	–5	322	–28	210	249	289

[a] No of genetic principal components.

[b] AI: Average information, EM: expectation maximisation, PX-EM: parameter expanded EM, PX+AI: 4 PX-EM steps followed by AI.

[c] A value of 0 denotes a deviation < 0.001.

Both EM and PX-EM required hundreds of iterates to locate the maximum of $\log \mathcal{L}$. With a linear convergence pattern, reaching a stage where the $\mathbf{\Delta}\mathcal{L}$ dropped to less than 10^{-5} generally doubled the amount of iterates required, compared to a less stringent value of 0.005, while increasing $\log \mathcal{L}$ by less than 0.04. For all orders of fit, estimates of the matrix of auxiliary parameters for PX-EM, $\alpha^{\star}$, approached an identity matrix in relatively few iterates. While the PX-EM yielded slightly bigger improvements in $\log \mathcal{L}$ than the EM algorithm initially, there was only little advantage over standard EM overall, even when all PCs were fitted. In stark contrast, there were substantial differences between the two algorithms for full rank estimation on the original scale [28],

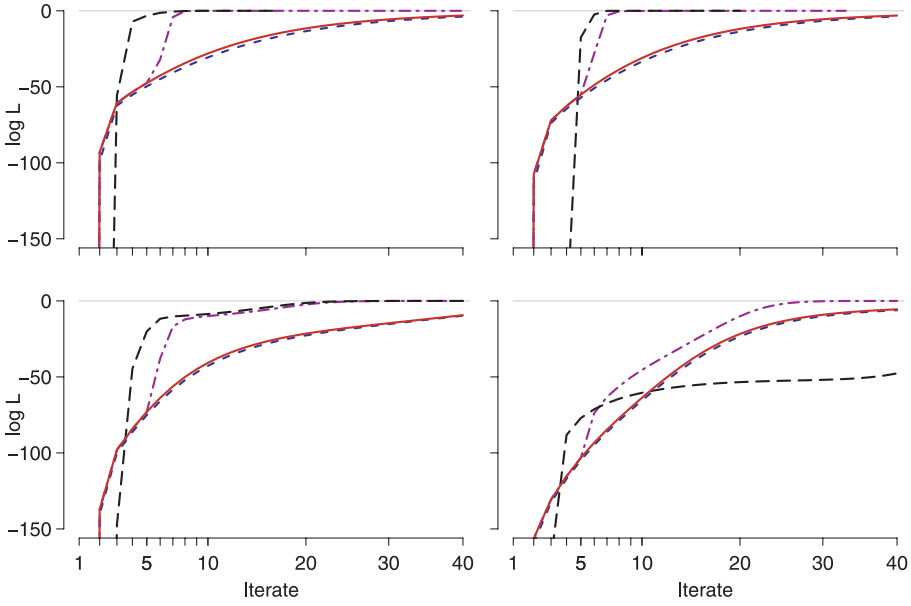

Figure 1. Change in relative log likelihood (log $\mathcal{L}$) for Example 1 in the first 40 iterates for various algorithms, fitting 8 (top left), 6 (top right), 4 (bottom left) and 2 (bottom right) principal components. (—— PX-EM, - - - - EM, — — — AI, and —·—·— PX+AI algorithm)

i.e., as suggested by Meng and van Dyk [23], parameterisation to the elements of the Cholesky factor greatly improved convergence of the EM algorithm.

In contrast, the AI algorithm converged in few iterates. With a quadratic convergence pattern, generally only a few additional iterates were required when increasing the stringency of the convergence criterion tenfold or more. The eigenvalue for the last PC of the 8 traits was very small (< 0.001). In turn, this yielded an AI matrix with small minimum eigenvalue, so that a constant needed to be added to its diagonal and multiple steps requiring step size scaling. Omitting this PC (Fit 7) removed the need for these control measures and improved the rate of convergence. Reducing the rank of fit further had comparatively little effect on the convergence of the AI algorithm, as long as the eigenvalues of the PCs not fitted were small. Fitting less than 5 PCs, however, there was a trend for the number of iterates required to increase with the number of PCs omitted. This was especially evident for an analysis fitting 2 PCs (see Fig. 1). While this did not cause a need for step size scaling or modification of the AI matrix, there was a sequence iterates with small changes in log $\mathcal{L}$

only. For these scenarios, a few initial iterates of the PX-EM algorithm tended to 'bypass' this area of search and thus reduced the number of iterates required by roughly 40%.

4.3.2. Example 2

For Example 2, analyses were carried out fitting all 4 PCs for direct genetic (A), maternal genetic (M), permanent environmental (C) and residual (E) covariance matrices (Model 4444), fitting 3 PCs for A and M and 2 PCs for C (Model 3324), and fitting 2 PCs for A, M and C (Model 2224), yielding 40, 33 and 30 parameters to be estimated, respectively. Convergence characteristics are summarised in Table III. As for Example 1, the PX-EM and EM (not shown) algorithms required substantial numbers of iterates to locate the maximum of $\log \mathcal{L}$, while the AI algorithm converged in about 20 iterates. With multiple random effects and highly correlated traits, both RdR analyses shown omitted only PCs with small eigenvalues and thus converged more quickly than the full rank analysis.

4.3.3. Example 3

For Example 3, RdR analyses considered 7 and 9 PCs (Model 79), 5 and 7 PCs (Model 57), and 5 PCs (Model 55) for both genetic and permanent environmental covariances, respectively *c.f.* [26]. For this example, the number of iterates required for the (PX-)EM algorithm were excessive, especially for the analysis fitting only 5 PCs for both random effects. With relative 'good' starting values, full rank AI (Model 13 13) converged quickly despite representing a highly overparameterised model, requiring 30 iterates for $\Delta\mathcal{L}$ to drop below 0.0005 with a corresponding deviation from $\log \mathcal{L}_{max}$ of -0.01; see Table III. For RdR analyses, the number of AI iterates required was again reduced at first (Model 79) but tended to increase when PCs with non-negligible eigenvalues were omitted. The latter was due to a series of AI steps with small, monotonically declining improvements in $\log \mathcal{L}$, yielding more a linear than a quadratic convergence pattern.

5. DISCUSSION

RdR estimation of covariance matrices decreases the number of parameters to be estimated. Moreover, omitting PCs with negligible eigenvalues alleviates

Table III. Convergence characteristics for Examples 2 and 3.

Fit[a]		Change in log $\mathcal{L}$ less than						Deviation less than		
		0.00005		0.00010		0.00050		–0.20	–0.10	–0.05
		N[b]	D[b]	N	D	N	D	N	N	N
Example 2										
4444	AI[b]	29	0	23	–1	15	–2	6	7	8
	PX+AI	21	0	17	–1	16	–1	8	9	10
	PX-EM	591	–8	500	–14	323	–55	190	259	353
3324	AI	21	0	21	0	18	0	11	12	13
	PX+AI	20	0	20	0	19	0	12	12	13
	PX-EM	546	–5	468	–10	284	–54	173	221	293
2224	AI	20	0	19	0	17	0	10	11	12
	PX+AI	22	0	21	0	19	0	12	13	14
	PX-EM	734	–2	701	–4	631	–21	535	564	593
Example 3										
13 13	AI	62	–3	62	–3	30	–10	10	11	14
	PX+AI	75	–2	52	–5	33	–10	16	17	19
	PX-EM	1690	–38	1346	–61	792	–185	763	1062	1476
79	AI	25	–30	25	–30	25	–30	13	18	22
	PX+AI	39	–1	33	–4	33	–4	15	20	22
	PX-EM	3198	–36	2663	–72	1632	–320	1947	2422	2936
57	AI	60	0	56	0	48	–2	26	29	32
	PX+AI	76	0	73	0	67	–1	46	50	53
	PX-EM	7923	–22	7551	–47	3623	–1611	6818	7172	7518
55	AI	115	–1	107	–1	88	–6	47	54	62
	PX+AI	116	–1	108	–1	89	–6	47	55	63
	PX-EM	7250	–111	5605	–221	2689	–874	5828	7495	9249

[a] Numbers of principal components fitted for covariance matrices estimated, numbers for A, M, C and E for Example 2, and A and R for Example 3; *cf.* Table I.

[b] See Table II for abbreviations.

problems associated with attempting to estimate parameters close to the boundary of their permissible space, and tends to improve convergence rates compared to full rank analyses. One of the main obstacles in multivariate analyses involving more than a few traits is the computational effort involved. While the size of the MME to be manipulated in REML estimation is proportional to the number of PCs fitted for random effects, the number of operations required in each iterate increases more than quadratically with the number of PCs. Thus even a small reduction in the number of PCs considered can have a dramatic effect on the computational requirements *e.g.* [27]. For Example 1, for instance,

total computing times required using the AI algorithm (with a convergence criterion of $\Delta\mathcal{L} < 0.0005$) were 2678, 1076, 723 and 624 seconds for analyses fitting 8, 7, 6 and 5 PCs, respectively (using a 64-bit dual core processor, rated at 2.6 Ghz). The combination of greater stability and faster convergence in estimation and reduction in computational requirements per iterate makes RdR analysis a powerful strategy for higher dimensional multivariate analyses.

Caution is required, however, when reducing the number of PCs fitted beyond those with negligible eigenvalues. As results show, this can increase the number of REML iterates required. Moreover, estimates of both the directions and eigenvalues of the subset of PCs fitted tend to be biassed in this case [30].

The examples chosen represent diverse and difficult analyses involving many parameters and, at full rank, somewhat overparameterised models, applied to relatively small data sets. All algorithms examined were capable of maximising $\log \mathcal{L}$. The AI algorithm generally required substantially fewer iterates than the PX-EM or EM algorithm, but stringent control of the AI steps and care in choosing an appropriate parameterisation were needed throughout. Earlier work [2, 28], considering the PX-EM algorithm for full rank estimation found it to be most useful in conjunction with the AI algorithm, replacing the first few iterates to reduce problems due to poor starting values or initial overshooting. As shown, the PX-EM algorithm is readily adapted to RdR estimation, and again is most useful combined with the AI algorithm for scenarios where AI performs relatively poorly initially.

6. CONCLUSION

The PX-EM algorithm is a useful, additional ‘weapon’ in our armoury for REML estimation of variance components. Reduced rank estimation is highly appealing and can reduce the number of iterates required as well as the computational requirements per iterate, thus making multivariate analyses involving more than a few traits more feasible.

ACKNOWLEDGEMENTS

This work was supported by grant BFGEN.100B of Meat and Livestock Australia Ltd (MLA) and funds from the International Livestock Resources and Information Centre (ILRIC).

REFERENCES

[1] Boyd S., Vandenberghe L., Convex Optimization, Cambridge University Press (2004).
[2] Cullis B.R., Smith A.B., Thompson R., Perspectives of ANOVA, REML and a general linear mixed model, in: Methods and Models in Statistics in Honour of Professor John Nelder, FRS, Imperial College Press, London, 2004, pp. 53–94.
[3] Dempster A.P., Laird N.M., Rubin D.B., Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Stat. Soc. B 39 (1977) 1–39.
[4] Dennis J.E., Schnabel R.B., Numerical methods for Unconstrained Optimization and Nonlinear Equations, SIAM Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, 1996.
[5] Forsgren A., Gill P.E., Murray W., Computing modified Newton directions using a partial Cholesky factorization, SIAM J. Sci. Statist. Comp. 16 (1995) 139–150.
[6] Foulley J.L., van Dyk D.A, The PX-EM algorithm for fast stable fitting of Henderson's mixed model, Genet. Sel. Evol. 32 (2000) 143–163.
[7] Groeneveld E., A reparameterisation to improve numerical optimisation in multivariate REML (co)variance component estimation, Genet. Sel. Evol. 26 (1994) 537–545.
[8] Harville D.A., Maximum likelihood approaches to variance component estimation and related problems, J. Amer. Stat. Ass. 72 (1977) 320–338.
[9] Harville D.A., Matrix Algebra from a Statistician's Perspective, Springer Verlag, 1997.
[10] Henderson C.R., A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values, Biometrics 32 (1976) 69–83.
[11] Henderson C.R., Estimation of variances and covariances under multiple trait models, J. Dairy Sci. 67 (1984) 1581–1589.
[12] Jamshidian M., Jennrich R.I, Conjugate gradient acceleration of the EM algorithm, J. Amer. Stat. Ass. 88 (1993) 221–228.
[13] Jamshidian M., Jennrich R.I., Acceleration of the EM algorithm using Quasi-Newton methods, J. Roy. Stat. Soc. B 59 (1997) 569–587.
[14] Jennrich R.I., Sampson P.F., Newton-Raphson and related algorithms for maximum likelihood variance component estimation, Technometrics 18 (1976) 11–17.
[15] Jennrich R.I., Schluchter M.D., Unbalanced repeated-measures models with structured covariance matrices, Biometrics 42 (1986) 805–820.
[16] Kirkpatrick M., Meyer K., Simplified analysis of complex phenotypes: Direct estimation of genetic principal components, Genetics 168 (2004) 2295–2306.
[17] Laird N., Lange N., Stram D., Maximum likelihood computations with repeated measures: applications of the EM algorithm, J. Amer. Stat. Ass. 82 (1987) 97–105.
[18] Lange K., A gradient algorithm locally equivalent to the EM algorithm, J. Roy. Stat. Soc. B 57 (1995) 425–438.

[19] Lindstrom M.J., Bates D.M., Newton-Raphson and EM algorithms for linear mixed-effects models for repeated-measures data, J. Amer. Stat. Ass. 83 (1988) 1014–1022.
[20] Liu C., Rubin D.B., Wu Y.N., Parameter expansions to accelerate EM: The PX-EM algorithm, Biometrika 85 (1998) 755–770.
[21] McLachlan G.J., Krishnan T., The EM algorithm and extensions, Wiley Series in Probability and Statistics, Wiley, New York, 1997.
[22] Meilijson I., A fast improvement of the EM algorithm on its own terms, J. Roy. Stat. Soc. B 51 (1989) 127–138.
[23] Meng X.L., van Dyk D., The EM algorithm - an old folk-song sung to a new fast tune, J. Roy. Stat. Soc. B 59 (1997) 511–567.
[24] Meng X.L., van Dyk D., Fast EM-type implementations for mixed-effects models, J. Roy. Stat. Soc. B 60 (1998) 559–578.
[25] Meyer K., Random regressions to model phenotypic variation in monthly weights of Australian beef cows, Livest. Prod. Sci. 65 (2000) 19–38.
[26] Meyer K., Advances in methodology for random regression analyses, Austr. J. Exp. Agric. 45 (2005a) 847–858.
[27] Meyer K., Genetic principal components for live ultra-sound scan traits of Angus cattle, Anim. Sci. 81 (2005b) 337–345.
[28] Meyer K., PX × AI: algorithmics for better convergence in restricted maximum likelihood estimation, CD-ROM Eighth World Congr. Genet. Appl. Livest. Prod., August 13–18 2006, Belo Horizonte, Brasil, Communication No. 24-15.
[29] Meyer K., Kirkpatrick M., Restricted maximum likelihood estimation of genetic principal components and smoothed covariance matrices, Genet. Sel. Evol. 37 (2005) 1–30.
[30] Meyer K., Kirkpatrick M., A note on bias in reduced rank estimates of covariance matrices, Proc. Ass. Advan. Anim. Breed. Genet. 17 (2007) 154–157.
[31] Meyer K., Smith S.P., Restricted maximum likelihood estimation for animal models using derivatives of the likelihood, Genet. Sel. Evol. 28 (1996) 23–49.
[32] Meyer K., Carrick M.J., Donnelly B.J.P., Genetic parameters for growth traits of Australian beef cattle from a multi-breed selection experiment, J. Anim. Sci. 71 (1993) 2614–2622.
[33] Neumaier A., Groeneveld E., Restricted maximum likelihood estimation of covariance components in sparse linear models, Genet. Sel. Evol. 30 (1998) 3–26.
[34] Ng S.K., Krishnan T., McLachlan G.J., The EM algorithm, in: Gentle J.E., Härdle W., Mori Y., (Eds.), Handbook of Computational Statistics, vol. I, Springer Verlag, New York, 2004, pp. 137–168.
[35] Nocedahl J., Wright S.J., Numerical Optimization, Springer Series in Operations Research, Springer Verlag, New York, Berlin Heidelberg, 1999.
[36] Pinheiro J.C., Bates D.M., Unconstrained parameterizations for variance-covariance matrices, Stat. Comp. 6 (1996) 289–296.
[37] Schnabel R.B., Estrow E., A new modified Cholesky factorization, SIAM J. Sci. Statist. Comp. 11 (1990) 1136–1158.
[38] Thompson R., Meyer K., Estimation of variance components: What is missing in the EM algorithm? J. Stat. Comp. Simul. 24 (1986) 215–230.

[39] Thompson R., Cullis B.R., Smith A.B., Gilmour A.R., A sparse implementation of the Average Information algorithm for factor analytic and reduced rank variance models, Austr. New Zeal. J. Stat. 45 (2003) 445–459.
[40] Thompson R., Brotherstone S., White I.M.S., Estimation of quantitative genetic parameters, Phil. Trans. R. Soc. B 360 (2005) 1469–1477.
[41] van Dyk D.A., Fitting mixed-effects models using efficient EM-type algorithms, J. Comp. Graph. Stat. 9 (2000) 78–98.

APPENDIX

The mn-th element of the average information is calculated as $\mathbf{b}'_m\mathbf{P}\mathbf{b}_n$, with work vector $\mathbf{b}_m = (\partial\mathbf{V}/\partial\theta_m)\,\mathbf{P}\mathbf{y}$ and projection matrix $\mathbf{P} = \mathbf{V}^{-1} - \mathbf{V}^{-1}\mathbf{X}\left(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}\right)^{-}\mathbf{X}\mathbf{V}^{-1}$. As $\mathbf{P}\mathbf{y} = \mathbf{R}^{-1}\hat{\mathbf{e}}$ [8], $\mathbf{b}_m$ is readily determined from the vector of residuals. Moreover, for parameters representing covariance components (or functions thereof) due to random effects fitted and full rank estimation, $\mathbf{b}_m$ can conveniently be calculated from the corresponding vector of random effects. For the model given by (Eq. 1), $\mathbf{b}_m = \mathbf{Z}\left((\partial\mathbf{\Sigma}_U/\partial\theta_m)\mathbf{\Sigma}_U^{-1} \otimes \mathbf{I}\right)\hat{\mathbf{u}}$. The reduced rank equivalent suggested earlier [29] (Eqs. A.14, A.15), however, is inappropriate, resulting in poor convergence rates.

Hence, (in the notation of Sect. 3.3) for $\theta_m = q_{ij}$ and $r < q$, $\mathbf{b}_m$ needs to be evaluated as

$$\mathbf{b}_m = \mathbf{Z}\left(\left(\mathbf{\Delta}_{ij}^{Q}\mathbf{Q}' + \mathbf{Q}(\mathbf{\Delta}_{ij}^{Q})'\right) \otimes \mathbf{A}\right)\mathbf{Z}'\mathbf{R}^{-1}\hat{\mathbf{e}}. \tag{19}$$

For genetic effects, this requires the numerator relationship matrix which can be quite dense. Hence, (Eq. 19) is best obtained in two steps, using that $\mathbf{A} = \mathbf{L}_A\mathbf{L}'_A$, with $\mathbf{L}_A$ the Cholesky factor of $\mathbf{A}$ which can be set up from a list of pedigree information *e.g.* [10].