Linear mixed model implementation in lme4

Douglas Bates
Department of Statistics
University of Wisconsin – Madison
Bates@wisc.edu

January 16, 2006

Abstract

Expressions for the evaluation of the profiled log-likelihood or profiled log-restricted-likelihood of a linear mixed model, the gradients and Hessians of these criteria, and update steps for an ECME algorithm to optimize these criteria are given in Bates and DebRoy (2004). A representation of linear mixed models using positive semidefinite symmetric matrices and dense matrices is given in Bates (2004). In this paper we present details of that representation and those computational methods in the `lme4` package for R.

1 Introduction

General formulae for the evaluation of the profiled log-likelihood and profiled log-restricted-likelihood in a linear mixed model are given in Bates and DebRoy (2004) and the use of a sparse matrix representation for such models is described in Bates (2004). The purpose of this paper is to describe the details of the implementation of this representation and those computational methods in the `lme4` package for R.

Because we concentrate on the computational methods and the representation, the order and style of presentation will be based on the sequence of calculations, not on the sequence in which the results would be derived. We will emphasize "what" and not "why". For the "why", refer to the papers cited above.

In §2 we describe the form and representation of the model. The calculation of the criteria to be optimized by the parameter estimates and related quantities is discussed in §3. Details of the calculation of the ECME step and the evaluation of the gradients of the criteria are given in §6 and those of the Hessian in §7. In §8 we give the details of an unconstrained parameterization for the model and the transformation of our results to this parameterization.

2 Form and representation of the model

We consider linear mixed models of the form

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{b} + \boldsymbol{\epsilon} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I}), \boldsymbol{b} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{\Omega}^{-1}), \boldsymbol{\epsilon} \perp \boldsymbol{b} \tag{1}$$

where $\boldsymbol{y}$ is the n-dimensional response vector, $\boldsymbol{X}$ is an $n \times p$ model matrix for the p dimensional fixed-effects vector $\boldsymbol{\beta}$, $\boldsymbol{Z}$ is the $n \times q$ model matrix for the q dimensional random-effects vector $\boldsymbol{b}$ that has a Gaussian distribution with mean $\boldsymbol{0}$ and relative precision matrix $\boldsymbol{\Omega}$ (i.e., $\boldsymbol{\Omega}$ is the precision of $\boldsymbol{b}$ relative to the precision of $\boldsymbol{\epsilon}$), and $\boldsymbol{\epsilon}$ is the random noise assumed to have a spherical Gaussian distribution. The symbol $\perp$ indicates independence of random variables. The matrix $\boldsymbol{X}$ has full column rank and $\boldsymbol{\Omega}$ is positive definite.

The random effects vector $\boldsymbol{b}$ and the columns of $\boldsymbol{Z}$ are divided into k outer blocks associated with grouping factors $\boldsymbol{f}_i, i = 1, \ldots, k$ in the data. Because so many of the expressions that we will use involve the grouping factors and quantities associated with them, we will reserve the letter i to index the grouping factors and related quantities. We will not explicitly state the range of i, which will always be $i = 1, \ldots, k$.

The outer blocks in $\boldsymbol{b}$ and the columns of $\boldsymbol{Z}$ are further subdivided into m_i inner blocks of size q_i where m_i is the number of levels of $\boldsymbol{f}_i$ (i.e. the number of distinct values in $\boldsymbol{f}_i$). Each grouping factor has associated with it an $n \times q_i$ model matrix $\boldsymbol{Z}_i$. The full model matrix $\boldsymbol{Z}$ is derived from the grouping factors $\boldsymbol{f}_i$ and these submodel matrices $\boldsymbol{Z}_i$.

Random effects in different outer blocks are independent. Within each of these blocks, the inner blocks of random effects are independent and identically distributed with mean $\boldsymbol{0}$ and $q_i \times q_i$ variance-covariance matrix $\sigma^2\boldsymbol{\Omega}_i^{-1}$. Thus $\boldsymbol{\Omega}$ is block diagonal in k blocks of size $m_i q_i \times m_i q_i$ and each of these blocks is itself block diagonal in m_i blocks, each of which is the positive defi-

nite symmetric $q_i \times q_i$ matrix $\mathbf{\Omega}_i$. We call this a *repeated block/block diagonal* matrix.

2.1 Model specification and representation

Like most model fitting functions in the S language, the `lme` function in the `lme4` package uses a formula/data specification. The formula specifies how to evaluate the response $\boldsymbol{y}$ and the fixed-effects model matrix $\boldsymbol{X}$ in the data frame provided as the `data` argument. An additional argument named `random` specifies the names of the grouping factors and the formulae for evaluation of the model matrices $\boldsymbol{Z}_i$.

Standard optional arguments, such as `na.action` and `subset`, are passed through to the `model.frame` function that returns the model frame in which to evaluate all the formulae of the model. The `drop.unused.levels` optional argument is set to `TRUE` so that unused levels in any factors are dropped during creation of the model frame. Thus there is no ambiguitiy regarding the number of levels m_i of each of the $\boldsymbol{f}_i$. Every level in each of the $\boldsymbol{f}_i$ must occur at least once in the model frame.

2.1.1 Pairwise crosstabulation

We begin by ordering the grouping factors so that $m_1 \geq m_2 \geq \cdots \geq m_k$ and, if $k > 1$, forming their *pairwise crosstabulation*. This is the crossproduct matrix, $\boldsymbol{T} = \tilde{\boldsymbol{Z}}^\mathsf{T}\tilde{\boldsymbol{Z}}$, for the *variance components model* determined by these grouping factors. (In a variance components model $q_1 = q_2 = \cdots = q_k = 1$ and each of the $\tilde{\boldsymbol{Z}}_i$ is the $n \times 1$ matrix of 1s. The ith outer block of columns in $\tilde{\boldsymbol{Z}}$ is the set of indicator columns for the levels of $\boldsymbol{f}_i$. The name "variance components" reflects the fact that, in this model, each of the $\mathbf{\Omega}_i$, which are scalars, is the relative precision of the component of the variation in the response attributed to factor $\boldsymbol{f}_i$.)

In fact it is not necessary to form and store $\boldsymbol{T}$. All that we need are the locations of the nonzeros in the off-diagonal blocks of the representation

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_{(1,1)} & \boldsymbol{T}_{(2,1)}^\mathsf{T} & \cdots & \boldsymbol{T}_{(k,1)}^\mathsf{T} \\ \boldsymbol{T}_{(2,1)} & \boldsymbol{T}_{(2,2)} & \cdots & \boldsymbol{T}_{(k,2)}^\mathsf{T} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{T}_{(k,1)} & \boldsymbol{T}_{(k,2)} & \cdots & \boldsymbol{T}_{(k,k)}^\mathsf{T} \end{bmatrix}. \tag{2}$$

(Because the ith outer block of columns in $\tilde{\boldsymbol{Z}}$ is a set of indicators, the diagonal blocks will themselves be diagonal and their patterns of nonzeros are trivial.) The blocks in the strict lower triangle, $\boldsymbol{T}_{(i,j)}, i > j$ are stored as a list of compressed, sparse, column-oriented matrices (see Appendix A for details). Only the column pointers and the row indices of these sparse matrices are used..

These off-diagonal blocks are easily calculated because the integer representations of the factors $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ form the row and column indices in a (possibly redundant) triplet representation of $\boldsymbol{T}_{(i,j)}$. All that we need to do is convert the triplet representation to the compressed, sparse, column-oriented representation. This common conversion is easily accomplished with standard software.

We check whether each of the matrices $\boldsymbol{T}_{(j,j-1)}, j = 2, \ldots, k$ has exactly one nonzero in each column. If so, the grouping factors form a *nested sequence* in that each level of factor $\boldsymbol{f}_i$ occurs with exactly one level of factor $\boldsymbol{f}_j, i < j \leq k$. In the compressed, sparse, column-oriented format it is easy to determine the number of nonzeros in each column because these are the successive differences in the column pointers.

If the grouping factors form a nested sequence (and a single grouping factor is trivially a nested sequence) there is no need for further symbolic analysis. If not, which is to say that we have multiple, non-nested grouping factors, we continue the symbolic analysis for the unit lower triangular factor $\tilde{\boldsymbol{L}}$ in the LDL form of the Cholesky decomposition of $\boldsymbol{T}$ (Davis, 2004).

This symbolic analysis is performed on rows of blocks in the lower triangle of $\tilde{\boldsymbol{L}}$, where the blocks of $\tilde{\boldsymbol{L}}$ correspond to those of $\boldsymbol{T}$

$$\tilde{\boldsymbol{L}} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \tilde{\boldsymbol{L}}_{(2,1)} & \tilde{\boldsymbol{L}}_{(2,2)} & \ldots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\boldsymbol{L}}_{(k,1)} & \tilde{\boldsymbol{L}}_{(k,2)} & \ldots & \tilde{\boldsymbol{L}}_{(k,k)} \end{bmatrix}. \tag{3}$$

We emphasize that we are only determining the positions of the nonzeros in the blocks of $\tilde{\boldsymbol{L}}$, not performing an actual decomposition. (The decomposition would fail for $k > 1$ because $\boldsymbol{T}$ is only positive semidefinite, not positive definite. It is composed of multiple blocks of indicators and the row sums within each block are always unity.)

Because $\boldsymbol{T}_{(1,1)}$ is diagonal, the block in the $(1,1)$ position of $\tilde{\boldsymbol{L}}$ will be both diagonal and unit, which is to say that it is the identity matrix of size

m_1. A consequence of the $(1,1)$ block being the identity is that the structure of the first column of blocks of $\tilde{\boldsymbol{L}}$ is the same as the corresponding block of $\boldsymbol{T}$. That is, the nonzeros in $\tilde{\boldsymbol{L}}_{(j,1)}$ are in the same positions as those in $\boldsymbol{T}_{(j,1)}$ for $1 < j \leq k$.

The off-diagonal nonzero positions in $\tilde{\boldsymbol{L}}_{(2,2)}$ are determined from the union of the off-diagonal nonzeros positions in $\boldsymbol{T}_{(2,2)}$, of which there are none, and those in $\tilde{\boldsymbol{L}}_{(2,1)}\tilde{\boldsymbol{L}}_{(2,1)}^{\mathsf{T}} = \boldsymbol{T}_{(2,1)}\boldsymbol{T}_{(2,1)}^{\mathsf{T}}$. The number of nonzeros in $\tilde{\boldsymbol{L}}_{(2,2)}$ can be changed by permuting the levels of $\boldsymbol{f}_2$, which causes a permutation of the rows of the blocks $\boldsymbol{T}_{(2,i)}$ and the columns of the blocks $\boldsymbol{T}_{(i,2)}$. We determine a fill-reducing permutation of the levels of $\boldsymbol{f}_2$ using routines from the Metis (Karapis, 2003) graph-partitioning package applied to the incidence graph of $\boldsymbol{T}_{(2,1)}\boldsymbol{T}_{(2,1)}^{\mathsf{T}}$. (This is the graph of m_2 nodes in which nodes s and t are connected by an edge if and only if $\left\{\boldsymbol{T}_{(2,1)}\boldsymbol{T}_{(2,1)}^{\mathsf{T}}\right\}_{s,t}$ is nonzero.) We apply this permutation to the columns of all blocks $\boldsymbol{T}_{(i,2)}$ and the rows of all blocks $\boldsymbol{T}_{(2,i)}$.

The symbolic analysis function from the LDL package applied to $\boldsymbol{T}_{(2,1)}\boldsymbol{T}_{(2,1)}^{\mathsf{T}}$ (after permuting the rows of $\boldsymbol{T}_{(2,1)}$) provides the positions of the nonzeros in $\tilde{\boldsymbol{L}}_{(2,2)}$, from which we determine the positions of the nonzeros in $\tilde{\boldsymbol{L}}_{(2,2)}^{-1}$. The positions of the nonzeros in $\tilde{\boldsymbol{L}}_{(3,2)}$ are those of $\boldsymbol{T}_{(3,2)}\tilde{\boldsymbol{L}}_{(2,2)}^{-1}$. The next step in the iteration is to form the union of the nonzero positions of $\tilde{\boldsymbol{L}}_{(3,2)}\tilde{\boldsymbol{L}}_{(3,2)}^{\mathsf{T}}$ and the nonzero positions in $\boldsymbol{T}_{(3,1)}\boldsymbol{T}_{(3,1)}^{\mathsf{T}}$ from which we determine a fill-reducing permutation for the levels of $\boldsymbol{f}_3$. The process is continued until a fill-reducing permutation for the levels of $\boldsymbol{f}_k$ and the structure of $\tilde{\boldsymbol{L}}_{(k,k)}$ and $\tilde{\boldsymbol{L}}_{(k,k)}^{-1}$ are determined.

As part of the symbolic analysis of each diagonal block, the elimination tree (Davis, 2004) of the block is determined and stored as an integer array of length m_i. It is straightforward to check the number of terminal nodes in this tree. If the elmination tree for $\tilde{\boldsymbol{L}}_{(i,i)}$ is found to have only one terminal node then $\tilde{\boldsymbol{L}}_{(i,i)}^{-1}$ will be dense. Furthermore, all subsequent diagonal blocks will be dense so there is no purpose in checking for a fill-reducing permutation of the levels of $\boldsymbol{f}_j, i < j \leq k$, and the symbolic analysis can be terminated.

2.1.2 Allocating storage

In the numeric representation of the model we will write the augmented model matrix of size $n \times (p+1)$ obtained by appending $\boldsymbol{y}$ to $\boldsymbol{X}$ as $\tilde{\boldsymbol{X}} = [\boldsymbol{X}, \boldsymbol{y}]$. We

can allocate the storage for the sparse matrix representation of the model using the results of the symbolic analysis and the numbers of columns in the model matrices, $q_i, i = 1, \ldots, k$ and $(p + 1)$.

We will store $\boldsymbol{\Omega}$, $\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}$, $\boldsymbol{Z}^\mathsf{T}\tilde{\boldsymbol{X}}$, $\tilde{\boldsymbol{X}}^\mathsf{T}\tilde{\boldsymbol{X}}$ and components $\boldsymbol{R}_{ZZ}$, $\tilde{\boldsymbol{R}}_{ZX}$ and $\tilde{\boldsymbol{R}}_{XX}$ of the Cholesky decomposition

$$\begin{bmatrix} \boldsymbol{Z}^\mathsf{T}\boldsymbol{Z} + \boldsymbol{\Omega} & \boldsymbol{Z}^\mathsf{T}\tilde{\boldsymbol{X}} \\ \tilde{\boldsymbol{X}}^\mathsf{T}\boldsymbol{Z} & \tilde{\boldsymbol{X}}^\mathsf{T}\tilde{\boldsymbol{X}} \end{bmatrix} = \boldsymbol{R}^\mathsf{T}\boldsymbol{R} \quad \text{where} \quad \boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_{ZZ} & \tilde{\boldsymbol{R}}_{ZX} \\ \boldsymbol{0} & \tilde{\boldsymbol{R}}_{XX} \end{bmatrix}. \tag{4}$$

in one of four possible formats: as a dense matrix, as a block/block sparse matrix, as a block/block diagonal matrix, or as a repeated block/block diagonal matrix.

For example, we have already indicated that $\boldsymbol{\Omega}$ is repeated block/block diagonal so we store it in the `Omega` slot as a list of k symmetric matrices of sizes $q_i \times q_i$ (only the upper triangles of the symmetric matrices are used).

The matrices $\boldsymbol{Z}^\mathsf{T}\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{X}}^\mathsf{T}\tilde{\boldsymbol{X}}$, and the decomposition components $\tilde{\boldsymbol{R}}_{ZX}$, and $\tilde{\boldsymbol{R}}_{XX}$ matrices are stored as dense matrices in slots named `ZtX`, `XtX`, `RZX` and `RXX`, respectively.

The matrix $\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}$ is stored in a symmetric, sparse column-oriented format like that of $\boldsymbol{T}$ in (2) except that $\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}$ has both inner and outer blocks. The $k \times k$ grid of outer blocks $(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})_{(i,j)}$ is determined by the grouping factors. These outer blocks correspond to the blocks $\boldsymbol{T}_{(i,j)}$ in $\boldsymbol{T}$. The diagonal outer block $(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})_{(i,i)}$ is itself block diagonal in m_i blocks of size $q_i \times q_i$. We store the upper triangles of these inner blocks in an array of size $q_i \times q_i \times m_i$. Block $(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})_{(i,j)}$ for $j < i \leq k$ is subdivided into inner blocks of size $q_i \times q_j$ corresponding to the levels of grouping factors i and j. Because an inner block of $(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})_{(i,j)}$ is nonzero if and only if the corresponding element of $\boldsymbol{T}_{(i,j)}$ is nonzero, we can use the column pointers and row indices from $\boldsymbol{T}_{(i,j)}$ for $(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})_{(i,j)}$ with the convention that they index the inner blocks, not the individual elements, of $(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})_{(i,j)}$. The inner blocks are stored in a dense array of dimension $q_i \times q_j \times n_{(i,j)}$ where $n_{(i,j)}$ is the number of nonzeros in $\boldsymbol{T}_{(i,j)}$. This is the block/block sparse matrix format.

Instead of calculating $\boldsymbol{R}_{ZZ}$ we calculate the LDL form

$$\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z} + \boldsymbol{\Omega} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^\mathsf{T} \tag{5}$$

where $\boldsymbol{L}$ is block/block unit lower triangular (i.e. block/block lower triangular with all the inner diagonal blocks in the ith outer diagonal block being the $q_i \times q_i$ identity matrix) and $\boldsymbol{D}$ is block/block diagonal.

Just as the column pointers and row indices of the blocks of $\boldsymbol{T}$ can be used for the outer blocks of $\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}$, we can use the column pointers and row indices of $\tilde{\boldsymbol{L}}$, which we have determined in the symbolic analysis, for the outer blocks of

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{L}_{(2,1)} & \boldsymbol{L}_{(2,2)} & \dots & \boldsymbol{0} \\ \vdots & \vdots & & \boldsymbol{0} \\ \boldsymbol{L}_{(k,1)} & \boldsymbol{L}_{(k,2)} & \dots & \boldsymbol{L}_{(k,k)} \end{bmatrix}. \tag{6}$$

That is, the same structure as $\tilde{\boldsymbol{L}}_{(i,j)}$ applies to the blocks $\boldsymbol{L}_{(i,i)}, 1 < i \le k$, $\boldsymbol{L}_{(i,i)}, 1 < i \le k$, and $\boldsymbol{L}_{(i,j)}, 1 \le j < i \le k$ except that each nonzero in $\tilde{\boldsymbol{L}}_{(i.j)}$ corresponds to a block of size $q_i \times q_j$ in $\boldsymbol{L}_{(i.j)}$.

From the symbolic analysis we also have the column pointers and row indices for $\tilde{\boldsymbol{L}}^{-1}_{(i,i)}$, corresponding to the diagonal outer blocks of $\boldsymbol{L}^{-1}$. We allocate storage for these diagonal outer blocks only. Note that $\boldsymbol{L}^{-1}$ is block/block unit lower triangular just like $\boldsymbol{L}$. Because the inner diagonal blocks of these matrices are always the identity, these inner blocks are not explicitly stored. Neither $\boldsymbol{L}_{(1,1)} = \boldsymbol{I}$ nor $(\boldsymbol{L}^{-1})_{(1,1)} = \boldsymbol{I}$ require any storage to be allocated.

We have used fill-reducing permutations of the levels of the grouping factors $\boldsymbol{f}_j, j = 2, \dots, k$. These can decrease, sometimes dramatically, the amount of storage required for the $\tilde{\boldsymbol{L}}_{(i,i)}$ but generally they do not result in compact storage of $\tilde{\boldsymbol{L}}^{-1}_{(i,i)}$. In the worst case the matrix $\tilde{\boldsymbol{L}}^{-1}_{(2,2)}$ will be dense or nearly dense, resulting in a storage requirement of approximately $q_2^2 m_2(m_2+1)/2$ elements for the array holding the numeric values for the $(2,2)$ outer block of $\boldsymbol{L}^{-1}$.

The matrix $\boldsymbol{D}$ is block/block diagonal. It is stored as a list of k arrays of sizes $q_i \times q_i \times m_i$.

After allocating the storage we evaluate $\boldsymbol{y}$, $\boldsymbol{X}$ and the $\boldsymbol{Z}_i$ then update the contents of the `XtX`, `ZtX`, and `ZtZ` slots. We allocate the storage and update the contents of the storage in separate steps so we can update the numeric values without reallocating storage during iterative algorithms for fitting generalized linear mixed models or nonlinear mixed models.

3 Evaluation of the objective

If prior estimates of the $\boldsymbol{\Omega}_i$ are available, we set the `Omega` slot accordingly. Otherwise we form initial estimates from the matrices $\boldsymbol{Z}_i$, as described in

Pinheiro and Bates (2000, ch. 3). We then begin iterative optimization of the estimation criterion with respect to the $\boldsymbol{\Omega}_i$ or with respect to the unconstrained parameter $\boldsymbol{\theta}$ that determines the $\boldsymbol{\Omega}_i$, as described in §8.

In this section we will describe the steps in evaluating the objective function (i.e. the function to be optimized w.r.t. the $\boldsymbol{\Omega}_i$ or w.r.t. $\boldsymbol{\theta}$) given the current values of the $\boldsymbol{\Omega}_i$. Recall that after setting values of the $\boldsymbol{\Omega}_i$ we form the Cholesky decomposition (4)

$$\begin{bmatrix} \boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}+\boldsymbol{\Omega} & \boldsymbol{Z}^\mathsf{T}\tilde{\boldsymbol{X}} \\ \tilde{\boldsymbol{X}}^\mathsf{T}\boldsymbol{Z} & \tilde{\boldsymbol{X}}^\mathsf{T}\tilde{\boldsymbol{X}} \end{bmatrix} = \boldsymbol{R}^\mathsf{T}\boldsymbol{R} \quad \text{where} \quad \boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_{ZZ} & \tilde{\boldsymbol{R}}_{ZX} \\ \boldsymbol{0} & \tilde{\boldsymbol{R}}_{XX} \end{bmatrix}.$$

and $\tilde{\boldsymbol{X}}$ incorporates both $\boldsymbol{X}$ and $\boldsymbol{y}$. In some cases it will be convenient to consider $\boldsymbol{X}$ and $\boldsymbol{y}$ separately and we will write the components of the Cholesky decomposition as if they were

$$\begin{bmatrix} \boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}+\boldsymbol{\Omega} & \boldsymbol{Z}^\mathsf{T}\boldsymbol{X} & \boldsymbol{Z}^\mathsf{T}\boldsymbol{y} \\ \boldsymbol{X}^\mathsf{T}\boldsymbol{Z} & \boldsymbol{X}^\mathsf{T}\boldsymbol{X} & \boldsymbol{X}^\mathsf{T}\boldsymbol{y} \\ \boldsymbol{y}^\mathsf{T}\boldsymbol{Z} & \boldsymbol{y}^\mathsf{T}\boldsymbol{X} & \boldsymbol{y}^\mathsf{T}\boldsymbol{y} \end{bmatrix} = \boldsymbol{R}^\mathsf{T}\boldsymbol{R} \quad \text{where} \quad \boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_{ZZ} & \boldsymbol{R}_{ZX} & \boldsymbol{r}_{Zy} \\ \boldsymbol{0} & \boldsymbol{R}_{XX} & \boldsymbol{r}_{Xy} \\ \boldsymbol{0} & \boldsymbol{0} & r_{yy} \end{bmatrix} \tag{7}$$

even though they are stored in the form of (4).

Recall also that $\boldsymbol{R}_{ZZ}$ is calculated and stored in the LDL form. Instead of storing the symmetric positive definite inner blocks of the block/block diagonal $\boldsymbol{D}$, we calculate and store their upper Cholesky factors. We write the block/block diagonal matrix of upper Cholesky factors as $\boldsymbol{D}^{1/2}$ and its transpose as $\boldsymbol{D}^{\mathsf{T}/2}$. (Note that transposing a block/block diagonal matrix only requires transposing the inner blocks.) The quantity $\log|\boldsymbol{D}|$ is evaluated as the sum of the logarithms of the squares of the diagonal elements of the inner diagonal blocks of $\boldsymbol{D}^{1/2}$.

The next step in the factorization is to solve for $\tilde{\boldsymbol{R}}_{ZX}$ in

$$\boldsymbol{R}_{ZZ}^\mathsf{T}\tilde{\boldsymbol{R}}_{ZX} = \boldsymbol{L}\boldsymbol{D}^{\mathsf{T}/2}\tilde{\boldsymbol{R}}_{ZX} = \boldsymbol{Z}^\mathsf{T}\tilde{\boldsymbol{X}} \tag{8}$$

using blocked sparse matrix techniques (see Appendix B), storing the result in the `RZX` slot. Finally, dense matrix operations are used to downdate the densely stored $\tilde{\boldsymbol{X}}^\mathsf{T}\tilde{\boldsymbol{X}}$ and obtain the upper Cholesky factor $\tilde{\boldsymbol{R}}_{XX}$ that satisfies

$$\tilde{\boldsymbol{R}}_{XX}^\mathsf{T}\tilde{\boldsymbol{R}}_{XX} = \tilde{\boldsymbol{X}}^\mathsf{T}\tilde{\boldsymbol{X}} - \tilde{\boldsymbol{R}}_{ZX}^\mathsf{T}\tilde{\boldsymbol{R}}_{ZX}, \tag{9}$$

and provides $\log\left(|\boldsymbol{R}_{XX}|^2\right)$ and r_{yy}.

The `status` slot is a pair of logical values called `factored` and `inverted`. It is set to `(TRUE, FALSE)` indicating that the factorization is current and that it has not been inverted.

In a separate calculation the logarithm of the determinant

$$\log|\boldsymbol{\Omega}| = \sum_{i=1}^{k} m_i \log|\boldsymbol{\Omega}_i| \tag{10}$$

is evaluated from the Cholesky factors of the $\boldsymbol{\Omega}_i$. The function to be minimized w.r.t. $\boldsymbol{\theta}$, called the *profiled deviance*, is

$$f(\boldsymbol{\theta}) = \log|\boldsymbol{D}| - \log|\boldsymbol{\Omega}| + n\left[1 + \log\left(\frac{2\pi r_{yy}^2}{n}\right)\right] \tag{11}$$

for ML estimation or

$$f_R(\boldsymbol{\theta}) = \log|\boldsymbol{D}| + \log\left(|\boldsymbol{R}_{XX}|^2\right) - \log|\boldsymbol{\Omega}| + (n-p)\left[1 + \log\left(\frac{2\pi r_{yy}^2}{n-p}\right)\right] \tag{12}$$

for REML.

3.1 Inverting the factorization

Evaluatation of the objective function does not require inversion of the factorization nor does evaluation of many other quantities of interest, such as the conditional variance estimates $\widehat{\sigma^2} = r_{yy}^2/n$ and $\widehat{\sigma}_R^2 = r_{yy}^2/(n-p)$, the conditional fixed effects estimates, $\widehat{\boldsymbol{\beta}}$, which satisfy $\boldsymbol{R}_{XX}\widehat{\boldsymbol{\beta}} = \boldsymbol{r}_{Xy}$, and the conditional modes of the random effects, $\widehat{\boldsymbol{b}}$, which satisfy

$$\boldsymbol{D}^{1/2}\boldsymbol{L}^{\mathsf{T}}\widehat{\boldsymbol{b}} = \boldsymbol{r}_{Zy} - \boldsymbol{R}_{ZX}\widehat{\boldsymbol{\beta}} \tag{13}$$

However, if we wish to evaluate the ECME step or the gradient and/or the Hessian of the objective, it is convenient to invert some parts of the decomposition. We invert the dense upper triangular matrix $\tilde{\boldsymbol{R}}_{XX}$ in place, producing $\boldsymbol{R}_{XX}^{-1}$ in the first p rows and columns, $1/r_{yy}$ in the $(p+1, p+1)$ position, and $-\widehat{\boldsymbol{\beta}}/r_{yy}$ in the first p rows of the $p+1$st column. We also invert each of the triangular inner blocks of $\boldsymbol{D}^{1/2}$ in place producing $\boldsymbol{D}^{-1/2}$. The inverses of the outer blocks on the diagonal of $\boldsymbol{L}$ are also determined and stored separately. (In general the diagonal outer blocks cannot be inverted

in place. The number of nonzero inner blocks in the inverse of an outer block on the diagonal can be different from the number of nonzero inner blocks in the outer diagonal block itself.) Notice that this last operation is trivial in the case of a single grouping factor or multiple, nested grouping factors because all the outer diagonal blocks in $\boldsymbol{L}$ are identity matrices.

The matrix $\tilde{\boldsymbol{R}}_{ZX}\tilde{\boldsymbol{R}}_{XX}^{-1}$ is evaluated as a dense matrix product then each column of $-\boldsymbol{L}^{-\mathsf{T}}\boldsymbol{D}^{-1/2}\left(\tilde{\boldsymbol{R}}_{ZX}\tilde{\boldsymbol{R}}_{XX}^{-1}\right)$ is evaluated using blocked sparse matrix operations (see Appendix B for details) and stored in the `RZX` slot. The first p columns of the result, which are $-\boldsymbol{L}^{-\mathsf{T}}\boldsymbol{D}^{-1/2}\boldsymbol{R}_{ZX}\boldsymbol{R}_{XX}^{-1}$, are used to create products involving the matrix $\boldsymbol{V_b}$, which is the unscaled, conditional, REML variance-covariance of the random effects

$$
\begin{aligned}
\boldsymbol{V_b} &= \begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\end{bmatrix}\left(\begin{bmatrix}\boldsymbol{R}_{ZZ}^{\mathsf{T}} & \boldsymbol{0}\\ \boldsymbol{R}_{ZX}^{\mathsf{T}} & \boldsymbol{R}_{XX}^{\mathsf{T}}\end{bmatrix}\begin{bmatrix}\boldsymbol{R}_{ZZ} & \boldsymbol{R}_{ZX}\\ \boldsymbol{0} & \boldsymbol{R}_{XX}\end{bmatrix}\right)^{-1}\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}\\
&= \begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{R}_{ZZ}^{-1} & -\boldsymbol{R}_{ZZ}^{-1}\boldsymbol{R}_{ZX}\boldsymbol{R}_{XX}^{-1}\\ \boldsymbol{0} & \boldsymbol{R}_{XX}^{-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{R}_{ZZ}^{-\mathsf{T}} & \boldsymbol{0}\\ -\boldsymbol{R}_{XX}^{-\mathsf{T}}\boldsymbol{R}_{ZX}^{\mathsf{T}}\boldsymbol{R}_{ZZ}^{-\mathsf{T}} & \boldsymbol{R}_{XX}^{-\mathsf{T}}\end{bmatrix}\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}\\
&= \begin{bmatrix}\boldsymbol{R}_{ZZ}^{-1} & -\boldsymbol{R}_{ZZ}^{-1}\boldsymbol{R}_{ZX}\boldsymbol{R}_{XX}^{-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{R}_{ZZ}^{-\mathsf{T}}\\ -\boldsymbol{R}_{XX}^{-\mathsf{T}}\boldsymbol{R}_{ZX}^{\mathsf{T}}\boldsymbol{R}_{ZZ}^{-\mathsf{T}}\end{bmatrix}\\
&= \boldsymbol{R}_{ZZ}^{-1}\boldsymbol{R}_{ZZ}^{-\mathsf{T}} + \boldsymbol{R}_{ZZ}^{-1}\boldsymbol{R}_{ZX}\boldsymbol{R}_{XX}^{-1}\boldsymbol{R}_{XX}^{-\mathsf{T}}\boldsymbol{R}_{ZX}^{\mathsf{T}}\boldsymbol{R}_{ZZ}^{-\mathsf{T}}\\
&= \left(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z}+\boldsymbol{\Omega}\right)^{-1} + \boldsymbol{R}_{ZZ}^{-1}\boldsymbol{R}_{ZX}\boldsymbol{R}_{XX}^{-1}\boldsymbol{R}_{XX}^{-\mathsf{T}}\boldsymbol{R}_{ZX}^{\mathsf{T}}\boldsymbol{R}_{ZZ}^{-\mathsf{T}}\\
&= \left(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z}+\boldsymbol{\Omega}\right)^{-1} + \boldsymbol{L}^{-\mathsf{T}}\boldsymbol{D}^{-1/2}\boldsymbol{R}_{ZX}\boldsymbol{R}_{XX}^{-1}\boldsymbol{R}_{XX}^{-\mathsf{T}}\boldsymbol{R}_{ZX}^{\mathsf{T}}\boldsymbol{D}^{-\mathsf{T}/2}\boldsymbol{L}^{-1}
\end{aligned}
\tag{14}
$$

The $p+1$st column is $-\widehat{\boldsymbol{b}}/r_{yy}$.

The `status` flag is changed to `(TRUE, TRUE)` indicating that the factorization is current and that it is the inverted components that are stored in the `RZX`, `RXX`, and `Dfac` slots and that the `LIx` slot if current.

4 Examples

To illustrate this representation and these operations we consider some examples of common types of mixed effects models.

4.1 Single grouping factor

The `Early` data set in the `lme4` package is from a longitudinal study on an early childhood intervention program. These data are described in Singer

and Willett (2003, ch. 3). The response is a cognitive development score, `cog`, measured at ages 1, 1.5, and 2 years (`age`) on 103 infants (identified by `id`); 58 of whom were in the treatment group and 45 in the control group (identified by `trt`). We convert the `age` measurement to "time on study", `tos` as the treatment began at age 0.5 years.

```
> Early$tos <- Early$age - 0.5
> str(Early)
```

```
`data.frame':        309 obs. of  5 variables:
 $ id : Factor w/ 103 levels "86","87","77",..: 12 12 12 17 17 17 22 22 22 8 ...
 $ cog: int  103 119 96 106 107 96 112 86 73 100 ...
 $ age: num  1 1.5 2 1 1.5 2 1 1.5 2 1 ...
 $ trt: Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
 $ tos: num  0.5 1 1.5 0.5 1 1.5 0.5 1 1.5 0.5 ...
```

A lattice plot of the longitudinal scores is given in Figure 1.

These data are grouped by subject, as is common in longitudinal data. The only grouping factor for random effects is `id`. (The subjects are themselves grouped into a treatment group and a control group but that dichotomy would be modeled with fixed effects, not random effects.) As an initial model an analyst may fit an *unconditional growth* model (Singer and Willett, 2003, ch. 4)

```
> fm1 <- lmer(cog ~ tos + (tos | id), Early)
```

or proceed immediately to a model that allows for the effects of the treatment

```
> fm2 <- lmer(cog ~ tos * trt + (tos | id), Early)
```

The structure of the resulting object is

```
> str(fm2)
```

```
Formal class 'lmer' [package "Matrix"] with 35 slots
  ..@ assign  : int(0)
  ..@ frame   :`data.frame':        309 obs. of  4 variables:
  .. ..$ cog: int [1:309] 103 119 96 106 107 96 112 86 73 100 ...
  .. ..$ tos: num [1:309] 0.5 1 1.5 0.5 1 1.5 0.5 1 1.5 0.5 ...
  .. ..$ trt: Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
  .. ..$ id : Factor w/ 103 levels "86","87","77",..: 12 12 12 17 17 17 22 22 22 8 ...
  .. ..- attr(*, "terms")=Classes 'terms', 'formula' length 3 cog ~ tos * trt + (tos + id)
  .. .. .. ..- attr(*, "variables")= language list(cog, tos, trt, id)
  .. .. .. ..- attr(*, "factors")= int [1:4, 1:4] 0 1 0 0 0 0 1 0 0 0 ...
  .. .. .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. .. .. ..$ : chr [1:4] "cog" "tos" "trt" "id"
  .. .. .. .. .. ..$ : chr [1:4] "tos" "trt" "id" "tos:trt"
  .. .. .. ..- attr(*, "term.labels")= chr [1:4] "tos" "trt" "id" "tos:trt"
  .. .. .. ..- attr(*, "order")= int [1:4] 1 1 1 2
  .. .. .. ..- attr(*, "intercept")= int 1
  .. .. .. ..- attr(*, "response")= int 1
  .. .. .. ..- attr(*, ".Environment")=length 3 <environment>
  .. .. .. ..- attr(*, "predvars")= language list(cog, tos, trt, id)
  .. .. .. ..- attr(*, "dataClasses")= Named chr [1:4] "numeric" "numeric" "factor" "factor"
  .. .. .. .. ..- attr(*, "names")= chr [1:4] "cog" "tos" "trt" "id"
```

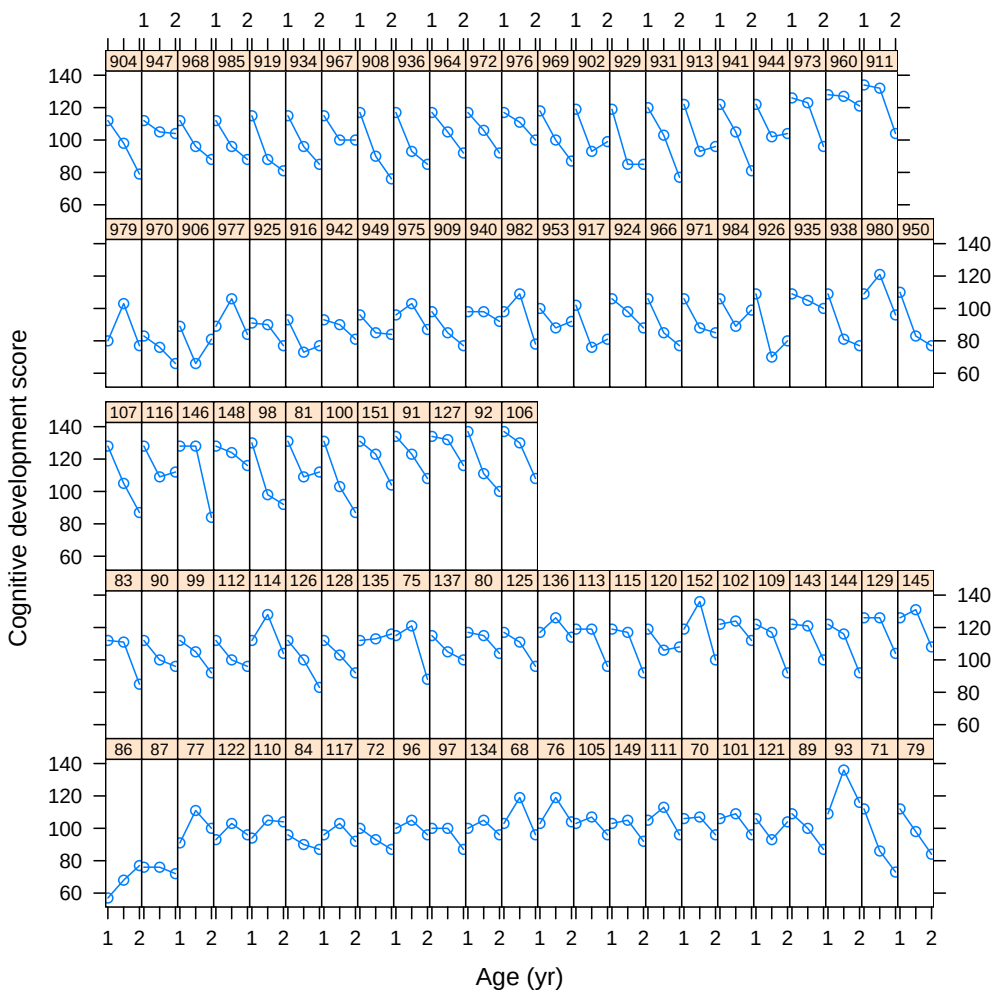

Figure 1: Cognitive development scores for 103 infants; 45 in the control group (identification numbers above 900) and 58 in the treatment group. The data from each infant are displayed in separate panels. Those for the infants in the treatment group are in the lower three rows of panels; the control group are in the upper two rows. Within each group the panels are ordered by increasing initial cognitive score.

```
..@ terms   :Classes 'terms', 'formula' length 3 cog ~ tos * trt
.. .. ..- attr(*, "variables")= language list(cog, tos, trt)
.. .. ..- attr(*, "factors")= int [1:3, 1:3] 0 1 0 0 0 1 0 1 1
.. .. .. ..- attr(*, "dimnames")=List of 2
.. .. .. .. ..$ : chr [1:3] "cog" "tos" "trt"
.. .. .. .. ..$ : chr [1:3] "tos" "trt" "tos:trt"
.. .. ..- attr(*, "term.labels")= chr [1:3] "tos" "trt" "tos:trt"
.. .. ..- attr(*, "order")= int [1:3] 1 1 2
.. .. ..- attr(*, "intercept")= int 1
.. .. ..- attr(*, "response")= int 1
.. .. ..- attr(*, ".Environment")=length 3 <environment>
.. .. ..- attr(*, "predvars")= language list(cog, tos, trt)
.. .. ..- attr(*, "dataClasses")= Named chr [1:3] "numeric" "numeric" "factor"
.. .. .. ..- attr(*, "names")= chr [1:3] "cog" "tos" "trt"
..@ flist   :List of 1
.. ..$ id: Factor w/ 103 levels "86","87","77",..: 12 12 12 17 17 17 22 22 22 8 ...
..@ Zt      :Formal class 'dgCMatrix' [package "Matrix"] with 6 slots
.. .. ..@ i       : int [1:618] 22 23 22 23 22 23 32 33 32 33 ...
.. .. ..@ p       : int [1:310] 0 2 4 6 8 10 12 14 16 18 ...
.. .. ..@ Dim     : int [1:2] 206 309
.. .. ..@ Dimnames:List of 2
.. .. .. ..$ : NULL
.. .. .. ..$ : NULL
.. .. ..@ x       : num [1:618] 1 0.5 1 1 1 1.5 1 0.5 1 1 ...
.. .. ..@ factors : list()
..@ X       : num [1:309, 1:4] 1 1 1 1 1 1 1 1 1 1 ...
.. ..- attr(*, "dimnames")=List of 2
.. .. ..$ : chr [1:309] "1" "2" "3" "4" ...
.. .. ..$ : chr [1:4] "(Intercept)" "tos" "trtY" "tos:trtY"
.. ..- attr(*, "assign")= int [1:4] 0 1 2 3
.. ..- attr(*, "contrasts")=List of 1
.. .. ..$ trt: chr "contr.treatment"
..@ y       : num [1:309] 103 119 96 106 107 96 112 86 73 100 ...
..@ wts     : num [1:309] 1 1 1 1 1 1 1 1 1 1 ...
..@ wrkres  : num [1:309] 103 119 96 106 107 96 112 86 73 100 ...
..@ method  : chr "REML"
..@ useScale: logi TRUE
..@ family  :List of 10
.. ..$ family    : chr "gaussian"
.. ..$ link      : chr "identity"
.. ..$ linkfun   :function (mu)
.. ..$ linkinv   :function (eta)
.. ..$ variance  :function (mu)
.. ..$ dev.resids:function (y, mu, wt)
.. ..$ aic       :function (y, n, mu, wt, dev)
.. ..$ mu.eta    :function (eta)
.. ..$ initialize:  expression({     n <- rep.int(1, nobs)     mustart <- y })
.. ..$ validmu   :function (mu)
.. ..- attr(*, "class")= chr "family"
..@ call    : language lmer(formula = cog ~ tos * trt + (tos | id), data = Early)
..@ cnames  :List of 2
.. ..$ id    : chr [1:2] "(Intercept)" "tos"
.. ..$ .fixed: chr [1:4] "(Intercept)" "tos" "trtY" "tos:trtY"
..@ nc      : Named int 2
.. ..- attr(*, "names")= chr "id"
..@ Gp      : int [1:2] 0 206
..@ XtX     :Formal class 'dpoMatrix' [package "Matrix"] with 5 slots
```

```
.. .. ..@ x       : num [1:16] 309   0   0   0 309 ...
.. .. ..@ Dim     : int [1:2] 4 4
.. .. ..@ Dimnames:List of 2
.. .. .. ..$ : chr [1:4] "(Intercept)" "tos" "trtY" "tos:trtY"
.. .. .. ..$ : chr [1:4] "(Intercept)" "tos" "trtY" "tos:trtY"
.. .. ..@ factors : list()
.. .. ..@ uplo    : chr "U"
..@ ZtZ     :Formal class 'dsCMatrix' [package "Matrix"] with 7 slots
.. .. ..@ i       : int [1:309] 0 0 1 2 2 3 4 4 5 6 ...
.. .. ..@ p       : int [1:207] 0 1 3 4 6 7 9 10 12 13 ...
.. .. ..@ Dim     : int [1:2] 206 206
.. .. ..@ Dimnames:List of 2
.. .. .. ..$ : NULL
.. .. .. ..$ : NULL
.. .. ..@ x       : num [1:309] 3 3 3.5 3 3 3.5 3 3 3.5 3 ...
.. .. ..@ uplo    : chr "U"
.. .. ..@ factors : list()
..@ ZtX     :Formal class 'dgeMatrix' [package "Matrix"] with 4 slots
.. .. ..@ x       : num [1:824] 3 3 3 3 3 3 3 3 3 3 ...
.. .. ..@ Dim     : int [1:2] 206 4
.. .. ..@ Dimnames:List of 2
.. .. .. ..$ : NULL
.. .. .. ..$ : chr [1:4] "(Intercept)" "tos" "trtY" "tos:trtY"
.. .. ..@ factors : list()
..@ Zty     : num [1:206] 202 212 224 222 302 ...
..@ Xty     : num [1:4] 31709 30774 18577 18117
..@ Omega   :List of 1
.. ..$ id:Formal class 'dpoMatrix' [package "Matrix"] with 5 slots
.. .. .. ..@ x       : num [1:4] 1.01e+09 0.00e+00 4.02e+09 1.60e+10
.. .. .. ..@ Dim     : int [1:2] 2 2
.. .. .. ..@ Dimnames:List of 2
.. .. .. .. ..$ : chr [1:2] "(Intercept)" "tos"
.. .. .. .. ..$ : chr [1:2] "(Intercept)" "tos"
.. .. .. ..@ factors :List of 1
.. .. .. .. ..$ Cholesky:Formal class 'Cholesky' [package "Matrix"] with 5 slots
.. .. .. .. .. .. ..@ x       : num [1:4] 3.18e+04 0.00e+00 1.26e+05 2.67e+00
.. .. .. .. .. .. ..@ Dim     : int [1:2] 2 2
.. .. .. .. .. .. ..@ Dimnames:List of 2
.. .. .. .. .. .. .. ..$ : NULL
.. .. .. .. .. .. .. ..$ : NULL
.. .. .. .. .. .. ..@ uplo    : chr "U"
.. .. .. .. .. .. ..@ diag    : chr "N"
.. .. .. ..@ uplo    : chr "U"
..@ L       :Formal class 'dCHMsuper' [package "Matrix"] with 7 slots
.. .. ..@ x    : num [1:412] 3.18e+04 1.26e+05 0.00e+00 5.84e+00 3.18e+04 ...
.. .. ..@ super: int [1:104] 0 2 4 6 8 10 12 14 16 18 ...
.. .. ..@ pi   : int [1:104] 0 2 4 6 8 10 12 14 16 18 ...
.. .. ..@ px   : int [1:104] 0 4 8 12 16 20 24 28 32 36 ...
.. .. ..@ s    : int [1:206] 0 1 2 3 4 5 6 7 8 9 ...
.. .. ..@ perm : int [1:206] 0 1 2 3 4 5 6 7 8 9 ...
.. .. ..@ type : int [1:6] 2 1 1 1 1 1
..@ RZX     :Formal class 'dgeMatrix' [package "Matrix"] with 4 slots
.. .. ..@ x       : num [1:824]  9.43e-05 -1.53e+00  9.43e-05 -1.53e+00  9.43e-05 ...
.. .. ..@ Dim     : int [1:2] 206 4
.. .. ..@ Dimnames:List of 2
.. .. .. ..$ : NULL
.. .. .. ..$ : chr [1:4] "(Intercept)" "tos" "trtY" "tos:trtY"
```

```
.. .. ..@ factors : list()
..@ RXX     :Formal class 'dtrMatrix' [package "Matrix"] with 5 slots
.. .. ..@ x       : num [1:16] 8.32 0.00 0.00 0.00 9.94 ...
.. .. ..@ Dim     : int [1:2] 4 4
.. .. ..@ Dimnames:List of 2
.. .. .. ..$ : chr [1:4] "(Intercept)" "tos" "trtY" "tos:trtY"
.. .. .. ..$ : chr [1:4] "(Intercept)" "tos" "trtY" "tos:trtY"
.. .. ..@ uplo    : chr "U"
.. .. ..@ diag    : chr "N"
..@ rZy     : num [1:206]  6.35e-03 -1.01e+02  7.04e-03 -1.14e+02  9.49e-03 ...
..@ rXy     : num [1:4]  824.6 -126.0   43.4   18.1
..@ devComp : Named num [1:7] 3.09e+02 4.00e+00 3.33e+06 1.00e+01 2.50e+03 ...
.. ..- attr(*, "names")= chr [1:7] "n" "p" "yty" "logryy2" ...
..@ deviance: Named num [1:2] 2370 2359
.. ..- attr(*, "names")= chr [1:2] "ML" "REML"
..@ fixef   : num [1:4] 118.41 -21.13   4.22   5.27
..@ ranef   : num [1:206] -42.99  10.83 -33.98   8.56  -7.77 ...
..@ RZXinv  :Formal class 'dgeMatrix' [package "Matrix"] with 4 slots
.. .. ..@ x       : num [1:824] -0.1246  0.0314 -0.1246  0.0314 -0.1246 ...
.. .. ..@ Dim     : int [1:2] 206 4
.. .. ..@ Dimnames:List of 2
.. .. .. ..$ : NULL
.. .. .. ..$ : chr [1:4] "(Intercept)" "tos" "trtY" "tos:trtY"
.. .. ..@ factors : list()
..@ bVar    :List of 1
.. ..$ id: num [1:2, 1:2, 1:103]  0.4622  0.0000 -0.1164  0.0293  0.4622 ...
..@ gradComp:List of 1
.. ..$ id: num [1:2, 1:2, 1:4] 227.3   0.0 -57.3  14.4 176.1 ...
..@ status  : Named logi [1:4] FALSE FALSE FALSE FALSE
.. ..- attr(*, "names")= chr [1:4] "factored" "secondary" "gradComp" "Hesscomp"
```

5 Frechet derivatives

We will denote the Frechet derivative of the matrix $\boldsymbol{\Omega}$ with respect to $\boldsymbol{\Omega}_i$ by $\mathsf{D}_{\boldsymbol{\Omega}_i}\boldsymbol{\Omega}$. This is a four dimensional array that can be regarded as an indexed set of q_i^2 matrices, each the same size as $\boldsymbol{\Omega}$. These matrices, which we call the q_i^2 "faces" of the array, are indexed by the rows r and the columns c, $r, c = 1, \ldots, q_i$ of $\boldsymbol{\Omega}_i$. The (r, c)th face, which we write as $\mathsf{D}_{i:r,c}\boldsymbol{\Omega}$, is the derivative of $\boldsymbol{\Omega}$ with respect to the (r, c)th element of $\boldsymbol{\Omega}_i$. It is a repeated block/block diagonal matrix where all of the outer diagonal blocks are zero except for the (i, i) diagonal block which is block diagonal in m_i blocks of $\boldsymbol{e}_r\boldsymbol{e}_c^{\mathsf{T}}$, where $\boldsymbol{e}_j$ is the jth column of the identity matrix of size q_i.

An expression of the form $\mathrm{tr}\left[\left(\mathsf{D}_{\boldsymbol{\Omega}_i}\boldsymbol{\Omega}\right)\boldsymbol{M}\right]$ where $\boldsymbol{M}$ is a matrix of the same size as $\boldsymbol{\Omega}$ evaluates to q_i^2 scalars indexed by r and c, $r, c = 1, \ldots, q_i$, which we convert to a $q_i \times q_i$ matrix in the obvious way. This type of expression

can be simplified as

$$\mathrm{tr}\left[(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega})\boldsymbol{M}\right] = \sum_{j=1}^{m_i} \mathrm{tr}\left[(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}_i)\boldsymbol{M}_{i,i,j,j}\right] = \sum_{j=1}^{m_i} \boldsymbol{M}_{i,i,j,j}^{\mathsf{T}} \tag{15}$$

where $\boldsymbol{M}_{i,i,j,j}$ is the jth inner diagonal block in the ith outer diagonal block of $\boldsymbol{M}$. To establish the last equality in (15) note that for any $q_i \times q_i$ matrix $\boldsymbol{A}$ the entry in row r and column c of $\mathrm{tr}\left[(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}_i)\boldsymbol{A}\right]$ is

$$\left\{\mathrm{tr}\left[(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}_i)\boldsymbol{A}\right]\right\}_{r,c} = \mathrm{tr}\left[\boldsymbol{e}_r\boldsymbol{e}_c^{\mathsf{T}}\boldsymbol{A}\right] = \mathrm{tr}\left[\boldsymbol{e}_c^{\mathsf{T}}\boldsymbol{A}\boldsymbol{e}_r\right] = \boldsymbol{e}_c^{\mathsf{T}}\boldsymbol{A}\boldsymbol{e}_r = \boldsymbol{A}_{c,r} \tag{16}$$

If $\boldsymbol{M}$ is symmetric, as is the case in all the expressions of this form that we consider, then $\mathrm{tr}\left[(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega})\,\boldsymbol{M}\right]$ will be symmetric.

Expressions of this form that we will require include

$$\begin{aligned}\mathrm{tr}\left[(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega})\,(\boldsymbol{\Omega})^{-1}\right] &= \sum_{j=1}^{m_i} \left(\boldsymbol{\Omega}_{i,i,j,j}\right)^{-1} \\ &= m_i\boldsymbol{\Omega}_i^{-1}\end{aligned}, \tag{17}$$

$$\mathrm{tr}\left[(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega})\left(\frac{\widehat{\boldsymbol{b}}}{r_{yy}}\frac{\widehat{\boldsymbol{b}}}{r_{yy}}^{\mathsf{T}}\right)\right] = \boldsymbol{B}_i\boldsymbol{B}_i^{\mathsf{T}} \tag{18}$$

where $\boldsymbol{B}_i$ is the $q_i \times m_i$ matrix whose j column is $\widehat{\boldsymbol{b}}_{i,j}/r_{yy}, j = 1,\ldots,m_i$ (recall that these vectors are in the $p+1$st column of the `RZX` slot after the inversion step), and $\mathrm{tr}\left[(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega})\,(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z}+\boldsymbol{\Omega})^{-1}\right]$.

The last term requires evaluation of the inner diagonal blocks of

$$\left(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z}+\boldsymbol{\Omega}\right)^{-1} = \left(\boldsymbol{R}_{ZZ}^{\mathsf{T}}\boldsymbol{R}_{ZZ}\right)^{-1} == \boldsymbol{L}^{-\mathsf{T}}\boldsymbol{D}^{-1/2}\boldsymbol{D}^{-\mathsf{T}/2}\boldsymbol{L}^{-1} \tag{19}$$

When $k=1$, $\boldsymbol{L}$ is an identity matrix and the result is simply

$$\mathrm{tr}\left[(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega})\,\boldsymbol{D}^{-1/2}\boldsymbol{D}^{-\mathsf{T}/2}\right] = \sum_{j=1}^{m_1} \boldsymbol{D}_{1:j}^{-1/2}\boldsymbol{D}_{1:j}^{-\mathsf{T}/2}.$$

When $k>1$ we must evaluate the crossproduct of each of the m_i inner blocks of q_i columns in the ith outer block of columns of $\boldsymbol{D}^{-\mathsf{T}/2}\boldsymbol{L}^{-1} = \boldsymbol{R}_{ZZ}^{-\mathsf{T}}$, which we do using blocked, sparse matrix techniques. See Appendix B.1 for details.

For REML results we also need

$$\begin{aligned}&\operatorname{tr}\left[\left(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}\right)\boldsymbol{V}_{\boldsymbol{b}}\right]=\\&\quad\operatorname{tr}\left[\left(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}\right)\boldsymbol{R}_{ZZ}^{-1}\boldsymbol{R}_{ZZ}^{-\mathsf{T}}\right]+\operatorname{tr}\left[\left(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}\right)\boldsymbol{R}_{ZZ}^{-1}\boldsymbol{R}_{ZX}\boldsymbol{R}_{XX}^{-1}\boldsymbol{R}_{XX}^{-\mathsf{T}}\boldsymbol{R}_{ZX}^{\mathsf{T}}\boldsymbol{R}_{ZZ}^{-\mathsf{T}}\right]\end{aligned}\tag{20}$$

We have just described how to calculate the first term in (20). The second term is evaluated as the crossproducts of the m_i inner blocks of q_i rows in the ith outer block of the rows of $-\boldsymbol{R}_{ZZ}^{-1}\boldsymbol{R}_{ZX}\boldsymbol{R}_{XX}^{-1}$, which is calculated and stored in the `RZX` slot during the inversion of the factorization, as described in §3.1.

6 ECME updates

ECME iterations provide a stable, but only linearly convergent, optimization algorithm for the objective functions f and f_R. We use a moderate number (the default is 20) of them to refine our initial estimates $\boldsymbol{\Omega}_i^{(0)}$ and bring us into the region of the optimum before switching to Newton iterations on the unconstrained parameter vector $\boldsymbol{\theta}$.

The ECME iterations can be defined in terms of the $\boldsymbol{\Omega}_i$. At the νth iteration, $\boldsymbol{\Omega}^{(\nu+1)}$ is defined to be the repeated block/block diagonal, symmetric, positive definite matrix that satisfies the k systems of equations

$$\operatorname{tr}\left[\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}\left(\frac{\widehat{\boldsymbol{b}}^{(\nu)}}{\widehat{\sigma}^{(\nu)}}\frac{\widehat{\boldsymbol{b}}^{(\nu)\,\mathsf{T}}}{\widehat{\sigma}^{(\nu)}}+\left(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z}+\boldsymbol{\Omega}^{(\nu)}\right)^{-1}-\left(\boldsymbol{\Omega}^{(\nu+1)}\right)^{-1}\right)\right]=\mathbf{0}\tag{21}$$

for ML estimates or the k systems

$$\operatorname{tr}\left[\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}\left(\frac{\widehat{\boldsymbol{b}}^{(\nu)}}{\widehat{\sigma}_R^{(\nu)}}\frac{\widehat{\boldsymbol{b}}^{(\nu)\,\mathsf{T}}}{\widehat{\sigma}_R^{(\nu)}}+\boldsymbol{V}_{\boldsymbol{b}}^{(\nu)}-\boldsymbol{\Omega}^{(\nu+1)^{-1}}\right)\right]=\mathbf{0}\tag{22}$$

for REML.

The results of §5 provide

$$\boldsymbol{\Omega}_i^{(\nu+1)}=m_i\left\{\operatorname{tr}\left[\left(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}\right)\left(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z}+\boldsymbol{\Omega}^{(\nu)}\right)^{-1}\right]+n\boldsymbol{B}_i^{(\nu)}\boldsymbol{B}_i^{(\nu)\,\mathsf{T}}\right\}^{-1}\tag{23}$$

for ML and

$$\boldsymbol{\Omega}_i^{(\nu+1)}=m_i\left\{\operatorname{tr}\left[\left(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}\right)\boldsymbol{V}_{\boldsymbol{b}}\right]+(n-p)\boldsymbol{B}_i^{(\nu)}\boldsymbol{B}_i^{(\nu)\,\mathsf{T}}\right\}^{-1}\tag{24}$$

for REML.

6.1 Gradient evaluations

The gradients of (11) and (12) with respect to $\boldsymbol{\Omega}_i$ are

$$\nabla_{\boldsymbol{\Omega}_i} f = \operatorname{tr}\left[\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}\left((\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}+\boldsymbol{\Omega})^{-1} - \boldsymbol{\Omega}^{-1} + \frac{\widehat{\boldsymbol{b}}}{\widehat{\sigma}}\frac{\widehat{\boldsymbol{b}}}{\widehat{\sigma}}^\mathsf{T}\right)\right] \tag{25}$$

$$\nabla_{\boldsymbol{\Omega}_i} f_R = \operatorname{tr}\left[\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}\left(\boldsymbol{V}_{\boldsymbol{b}} - \boldsymbol{\Omega}^{-1} + \frac{\widehat{\boldsymbol{b}}}{\widehat{\sigma}_R}\frac{\widehat{\boldsymbol{b}}}{\widehat{\sigma}_R}^\mathsf{T}\right)\right] \tag{26}$$

and we have already described how to calculate all those terms.

7 Evaluation of the Hessian

The Hessians of the scalar functions f and f_R are symmetric matrices of second derivatives. As for the calculation of the gradient, we first evaluate the Hessian with respect to the $Q = \sum_{i=1}^{k} q_i^2$ dimensional vector $\boldsymbol{\omega} = \left[\boldsymbol{\omega}_1^\mathsf{T}, \ldots, \boldsymbol{\omega}_k^\mathsf{T}\right]^\mathsf{T}$ where $\boldsymbol{\omega}_i = \operatorname{vec}\boldsymbol{\Omega}_i$ and 'vec' is the vectorization function that produces a column vector from a matrix by concatenating the columns. That is, we temporarily ignore the fact that the $\boldsymbol{\Omega}_i$ must be positive definite and symmetric and we consider each of the elements in these matrices separately. In §8 we describe an unconstrained parameter vector $\boldsymbol{\theta}$ of length $\sum_{i=1}^{k}\binom{q_i+1}{2}$ and the conversion of the gradient and Hessian for $\boldsymbol{\omega}$ to the corresponding quantities for $\boldsymbol{\theta}$.

For convenience we index the Q elements of $\boldsymbol{\omega}$ by triplets $i : r, c$ denoting the element at row r and column c of $\boldsymbol{\Omega}_i$. Bates and DebRoy (2004) show that the element in row $i : r, c$ and column $j : s, t$ of the Hessian is

$$\begin{aligned}
\left\{\nabla^2_{\boldsymbol{\omega}} f\right\}_{i:r,c;j:s,t} = &\operatorname{tr}\left[\mathsf{D}_{i:r,c}\left(\mathsf{D}_{j:s,t}\,\boldsymbol{\Omega}\right)\left(\left(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}+\boldsymbol{\Omega}\right)^{-1} - \boldsymbol{\Omega}^{-1} + \frac{\widehat{\boldsymbol{b}}}{\widehat{\sigma}}\frac{\widehat{\boldsymbol{b}}^\mathsf{T}}{\widehat{\sigma}}\right)\right] \\
&- \operatorname{tr}\left[\mathsf{D}_{i:r,c}\,\boldsymbol{\Omega}\left(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}+\boldsymbol{\Omega}\right)^{-1}\mathsf{D}_{j:s,t}\,\boldsymbol{\Omega}\left(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}+\boldsymbol{\Omega}\right)^{-1}\right] \\
&+ \operatorname{tr}\left[\left(\mathsf{D}_{i:r,c}\,\boldsymbol{\Omega}\right)\boldsymbol{\Omega}^{-1}\left(\mathsf{D}_{j:s,t}\,\boldsymbol{\Omega}\right)\boldsymbol{\Omega}^{-1}\right] \\
&- 2\frac{\widehat{\boldsymbol{b}}^\mathsf{T}}{\widehat{\sigma}}\left(\mathsf{D}_{i:r,c}\,\boldsymbol{\Omega}\right)\boldsymbol{V}_{\boldsymbol{b}}\left(\mathsf{D}_{j:s,t}\,\boldsymbol{\Omega}\right)\frac{\widehat{\boldsymbol{b}}}{\widehat{\sigma}} \\
&- \frac{1}{n}\left(\frac{\widehat{\boldsymbol{b}}^\mathsf{T}}{\widehat{\sigma}}\left(\mathsf{D}_{i:r,c}\,\boldsymbol{\Omega}\right)\frac{\widehat{\boldsymbol{b}}}{\widehat{\sigma}}\right)\left(\frac{\widehat{\boldsymbol{b}}^\mathsf{T}}{\widehat{\sigma}}\left(\mathsf{D}_{j:s,t}\,\boldsymbol{\Omega}\right)\frac{\widehat{\boldsymbol{b}}}{\widehat{\sigma}}\right)
\end{aligned} \tag{27}$$

for ML estimates, and

$$\begin{aligned}\left\{\nabla^2_{\boldsymbol{\omega}} f_R\right\}_{i:r,c;j:s,t} = &\operatorname{tr}\left[\mathsf{D}_{i:r,c}\left(\mathsf{D}_{j:s,t}\,\boldsymbol{\Omega}\right)\left(\boldsymbol{V}_{\boldsymbol{b}} - \boldsymbol{\Omega}^{-1} + \frac{\widehat{\boldsymbol{b}}}{\widehat{\sigma}}\frac{\widehat{\boldsymbol{b}}^{\mathsf{T}}}{\widehat{\sigma}}\right)\right] \\ &- \operatorname{tr}\left[\mathsf{D}_{i;r,c}\,\boldsymbol{\Omega}\boldsymbol{V}_{\boldsymbol{b}}\,\mathsf{D}_{j;s,t}\,\boldsymbol{\Omega}\boldsymbol{V}_{\boldsymbol{b}}\right] \\ &+ \operatorname{tr}\left[\left(\mathsf{D}_{i;r,c}\,\boldsymbol{\Omega}\right)\boldsymbol{\Omega}^{-1}\left(\mathsf{D}_{j;s,t}\,\boldsymbol{\Omega}\right)\boldsymbol{\Omega}^{-1}\right] \\ &- 2\frac{\widehat{\boldsymbol{b}}^{\mathsf{T}}}{\widehat{\sigma}_R}\left(\mathsf{D}_{i;r,c}\,\boldsymbol{\Omega}\right)\boldsymbol{V}_{\boldsymbol{b}}\left(\mathsf{D}_{j;s,t}\,\boldsymbol{\Omega}\right)\frac{\widehat{\boldsymbol{b}}}{\widehat{\sigma}_R} \\ &- \frac{1}{n}\left(\frac{\widehat{\boldsymbol{b}}^{\mathsf{T}}}{\widehat{\sigma}_R}\left(\mathsf{D}_{i;r,c}\,\boldsymbol{\Omega}\right)\frac{\widehat{\boldsymbol{b}}}{\widehat{\sigma}_R}\right)\left(\frac{\widehat{\boldsymbol{b}}^{\mathsf{T}}}{\widehat{\sigma}_R}\left(\mathsf{D}_{j;s,t}\,\boldsymbol{\Omega}\right)\frac{\widehat{\boldsymbol{b}}}{\widehat{\sigma}_R}\right)\end{aligned} \tag{28}$$

for REML.

The matrix $\mathsf{D}_{j:s,t}\,\boldsymbol{\Omega}$ is constant so the first term in both (27) and (28) is zero.

8 Unconstrained parameterization

The vector $\boldsymbol{\theta} = \left[\boldsymbol{\theta}_1^{\mathsf{T}}, \ldots, \boldsymbol{\theta}_k^{\mathsf{T}}\right]^{\mathsf{T}}$ is an unconstrained parameterization of $\boldsymbol{\Omega}$ based on the LDL form of the Cholesky decomposition of the $\boldsymbol{\Omega}_i$

$$\boldsymbol{\Omega}_i = \boldsymbol{L}_i \boldsymbol{D}_i \boldsymbol{L}_i^{\mathsf{T}}. \tag{29}$$

where $\boldsymbol{L}_i$ is a $q_i \times q_i$ unit lower triangular matrix and $\boldsymbol{D}_i$ is a $q_i \times q_i$ diagonal matrix with positive diagonal elements. We use the q_i logarithms of the diagonal elements of $\boldsymbol{D}_i$, written $\boldsymbol{\delta}_i$, and, when $q_i > 1$, the row-wise concatenation of the $\binom{q_i}{2}$ elements in the strict lower triangle of $\boldsymbol{L}_i$, written $\boldsymbol{\lambda}_i$, as the $\binom{q+1}{2}$ dimensional unconstrained parameter $\boldsymbol{\theta}_i = \left[\boldsymbol{\delta}_i^{\mathsf{T}}, \boldsymbol{\lambda}_i^{\mathsf{T}}\right]^{\mathsf{T}}$

Let g be any scalar function of the $\boldsymbol{\Omega}_i$ such that the $q_i \times q_i$ gradient matrices $\nabla_{\boldsymbol{\delta}_i} f$ are symmetric. (Both f, the objective for ML estimation, and f_R, the objective for REML estimation, are such functions.) The components $\nabla_{\boldsymbol{\delta}_i} g$ and $\nabla_{\boldsymbol{\lambda}_i} g$ of the gradient vector $\nabla_{\boldsymbol{\theta}_i} g$ can be evaluated from the symmetric gradient matrix $\nabla_{\boldsymbol{\Omega}_i} g$ and the derivative of the product (29). For

example,

$$\begin{aligned}
\left\{\nabla_{\boldsymbol{\delta}_i} g\right\}_j &= \operatorname{tr}\left[\left(\nabla_{\boldsymbol{\Omega}_i} g\right) \frac{\partial \boldsymbol{\Omega}_i}{\partial \left\{\boldsymbol{\delta}_i\right\}_j}\right] \\
&= \operatorname{tr}\left[\left(\nabla_{\boldsymbol{\Omega}_i} g\right) \boldsymbol{L}_i \frac{\partial \boldsymbol{D}_i}{\partial \left\{\boldsymbol{\delta}_i\right\}_j} \boldsymbol{L}_i^{\mathsf{T}}\right] \\
&= \exp\left\{\boldsymbol{\delta}_i\right\}_j \boldsymbol{e}_j^{\mathsf{T}} \boldsymbol{L}_i \left(\nabla_{\boldsymbol{\Omega}_i} g\right)^{\mathsf{T}} \boldsymbol{L}_i \boldsymbol{e}_j \\
&= \left\{\boldsymbol{R}_i \left(\nabla_{\boldsymbol{\Omega}_i} g\right) \boldsymbol{R}_i^{\mathsf{T}}\right\}_{j,j}
\end{aligned} \tag{30}$$

for $j = 1, \ldots, q_i$, where $\boldsymbol{R}_i$ is the upper Cholesky factor of $\boldsymbol{\Omega}_i$. The partial derivative with respect to element t of $\boldsymbol{\lambda}_i$, which determines the r, cth element of $\boldsymbol{L}_i$, is

$$\begin{aligned}
\left\{\nabla_{\boldsymbol{\lambda}_i} g\right\}_t &= \operatorname{tr}\left[\left(\nabla_{\boldsymbol{\Omega}_i} g\right) \frac{\partial \boldsymbol{\Omega}_i}{\partial \left\{\boldsymbol{\lambda}_i\right\}_t}\right] \\
&= \operatorname{tr}\left[\left(\nabla_{\boldsymbol{\Omega}_i} g\right) \boldsymbol{e}_r \boldsymbol{e}_c^{\mathsf{T}} \boldsymbol{D}_i \boldsymbol{L}_i^{\mathsf{T}} + \left(\nabla_{\boldsymbol{\Omega}_i} g\right) \boldsymbol{L}_i \boldsymbol{D}_i \boldsymbol{e}_c \boldsymbol{e}_r^{\mathsf{T}}\right] \\
&= \boldsymbol{e}_c^{\mathsf{T}} \boldsymbol{D}_i \boldsymbol{L}_i^{\mathsf{T}} \left(\nabla_{\boldsymbol{\Omega}_i} g\right) \boldsymbol{e}_r + \boldsymbol{e}_r^{\mathsf{T}} \left(\nabla_{\boldsymbol{\Omega}_i} g\right) \boldsymbol{L}_i \boldsymbol{D}_i \boldsymbol{e}_c \\
&= 2\left\{\left(\nabla_{\boldsymbol{\Omega}_i} g\right) \boldsymbol{L}_i \boldsymbol{D}_i\right\}_{c,r} \\
&= 2\left\{\left(\nabla_{\boldsymbol{\Omega}_i} g\right) \boldsymbol{R}_i^{\mathsf{T}} \boldsymbol{D}_i^{1/2}\right\}_{c,r}
\end{aligned} \tag{31}$$

9 Acknowledgements

This work was supported by U.S. Army Medical Research and Materiel Command under Contract No. DAMD17-02-C-0119. The views, opinions and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy or decision unless so designated by other documentation.

A Sparse matrix formats

The basic form of sparse matrix storage, called the *triplet* form, represents the matrix as three arrays of length `nz`, which is the number of nonredundant nonzero elements in the matrix. The `i` array contains the row indices, the `j` array the column indices, and the `x` array the values of the nonzero elements. A *column-oriented* triplet form requires that the elements of these arrays be

such that `j` is nondecreasing. A sorted, column-oriented triplet form is such that elements in `i` corresponding to the same `j` index are in increasing order. That is, the arrays are ordered by column and, within column, by row.

In column-oriented storage, multiple elements in the same column produce runs in `j`. A *compressed*, column-oriented storage form replaces `j` by `p`, an array of pointers to the indices (in `i` and `x`) of the first element in each column. If there are n columns in the matrix, this array is of length $n + 1$, with the last element of `p` being one greater than the index of last element in the last column. Then the differences of successive elements of `p` give the number of nonzeros in each column.

In the implementation in the `Matrix` package for R, the indices `i`, `j`, and `p` are `0`-based, as in the C language, (i.e. the first element of an array has index `0`) not `1`-based, as in the S language. Thus the first element of `p` is always zero.

B Blocked sparse matrix operations

We take advantage of the blocked sparse structure of $\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}$, $\boldsymbol{\Omega}$, $\boldsymbol{L}$ and $\boldsymbol{D}$ in operations involving these matrices. Consider, for example, the operation of solving for $\tilde{\boldsymbol{R}}_{ZX}$ in the system (8), $\boldsymbol{L}\boldsymbol{D}^{\mathsf{T}/2}\tilde{\boldsymbol{R}}_{ZX} = \boldsymbol{Z}^\mathsf{T}\tilde{\boldsymbol{X}}$. Let $\boldsymbol{u}$ be a column of $\boldsymbol{D}^{\mathsf{T}/2}\tilde{\boldsymbol{R}}_{ZX}$ and $\boldsymbol{v}$ be the corresponding column of $\boldsymbol{Z}^\mathsf{T}\tilde{\boldsymbol{X}}$. The corresponding column of the result, which is $\boldsymbol{D}^{-\mathsf{T}/2}\boldsymbol{u}$, is easily evaluated from $\boldsymbol{u}$. (Recall that $\boldsymbol{D}^{-1/2}$ is block/block diagonal, i.e. block diagonal in k outer blocks of sizes $m_i q_i \times m_i q_i$ each of which is itself block diagonal in m_i blocks of size $q_i \times q_i$, and that $\boldsymbol{D}^{-1/2}$ is calculated and stored during the inversion step.) Both $\boldsymbol{u}$ and $\boldsymbol{v}$ can be divided into k outer blocks of sizes $m_i q_i$, which we denote $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$, and each of the outer blocks can be divided into m_i inner blocks of size q_i, which we denote $\boldsymbol{u}_{i:j}$ and $\boldsymbol{v}_{i:j}$.

Because $\boldsymbol{L}$ is lower triangular, the blocks of $\boldsymbol{v}$ can be evaluated in a blockwise forward solve. That is, the blocked equations are solved in the order

$$\begin{aligned}
\boldsymbol{L}_{(1,1)}\boldsymbol{u}_1 &= \boldsymbol{v}_1 \quad \Longrightarrow \quad \boldsymbol{I}\boldsymbol{u}_1 = \boldsymbol{u}_1 = \boldsymbol{v}_1 \\
\boldsymbol{L}_{(2,2)}\boldsymbol{u}_2 &= \boldsymbol{v}_2 - \boldsymbol{L}_{(2,1)}\boldsymbol{u}_1 \\
&\vdots \\
\boldsymbol{L}_{(k,k)}\boldsymbol{u}_k &= \boldsymbol{v}_k - \boldsymbol{L}_{(k,k-1)}\boldsymbol{u}_{k-1} - \cdots - \boldsymbol{L}_{(k,1)}\boldsymbol{u}_1
\end{aligned} \tag{32}$$

Notice that the solution in the first block does not require any calculation

because $\boldsymbol{L}_{(1,1)}$ is always the identity. In many applications $k = 1$ and the operation of solving for $\boldsymbol{L}\boldsymbol{u} = \boldsymbol{v}$ can be skipped entirely. In most other applications the size of $\boldsymbol{u}_1$ dominates the size of the other components so that being able to set $\boldsymbol{u}_1 = \boldsymbol{v}_1$ represents a tremendous savings in computational effort.

We take advantage of the blocked, sparse columns of the outer blocks of $\boldsymbol{L}$ in evaluations of the form $\boldsymbol{v}_2 - \boldsymbol{L}_{(2,1)}\boldsymbol{u}_1$. That is, we evaluate $\boldsymbol{L}_{(2,1)}\boldsymbol{u}_1$ using the blocks of size $q_2 \times q_1$ skipping blocks that we know will be zero as we update the inner blocks of $\boldsymbol{u}_2$ in place. The solutions of systems of the form $\boldsymbol{L}_{(i,i)}\boldsymbol{u}_i = \tilde{\boldsymbol{v}}_i$ where $\tilde{\boldsymbol{v}}_i$ is the updated $\boldsymbol{v}_i$ are also done in inner blocks taking advantage of the sparsity of $\boldsymbol{L}_{(i,i)}^{-1}$. Recall that in the case of nested grouping factors the $\boldsymbol{L}_{(i,i)}$ are all identity matrices and this step is skipped.

B.1 Accumulating diagonal blocks of the inverse

The ECME steps and the gradient evaluation require

$$\operatorname{tr}\left[\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}\left(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z}+\boldsymbol{\Omega}\right)^{-1}\right] = \operatorname{tr}\left[\left(\mathsf{D}_{\boldsymbol{\Omega}_i}\,\boldsymbol{\Omega}\right)\left(\boldsymbol{D}^{-\mathsf{T}/2}\boldsymbol{L}^{-1}\right)^{\mathsf{T}}\left(\boldsymbol{D}^{-\mathsf{T}/2}\boldsymbol{L}^{-1}\right)\right] \quad (33)$$

which will be the sum of the crossproducts of the m_i inner blocks of q_i columns in the ith outer block of columns of $\boldsymbol{D}^{-\mathsf{T}/2}\boldsymbol{L}^{-1}$.

When $i = k = 1$ the result is $\sum_{j=1}^{m_1} \boldsymbol{D}_{1:j}^{-1/2}\boldsymbol{D}_{1:j}^{-\mathsf{T}/2}$, which can be evaluated in a single call to the level-3 BLAS routine `dsyrk`. (Having the 3 dimensional array containing the m_i inner blocks of the ith outer block of $\boldsymbol{D}^{-1/2}$ in the form that allows this to be done in a single, level-3 BLAS call is the reason that we store and invert the upper triangular factors of inner blocks of $\boldsymbol{D}$.)

When $i = 1$ and $k > 1$ we initialize the result from the $(1,1)$ block as above, then add the crossproducts of inner blocks of columns in the outer $(2,1)$ block, the outer $(3,1)$ block, and so on. These blocks of columns are calculated sparsely. During the initial decomposition we evaluate and store (in blocked, sparse format) $\boldsymbol{L}_{(i,i)}^{-1}$ for $1 < i \le k$.

References

Douglas Bates. Sparse matrix representations of linear mixed models. *J. of Computational and Graphical Statistics*, 2004. submitted.

Douglas M. Bates and Saikat DebRoy. Linear mixed models and penalized least squares. *J. of Multivariate Analysis*, 2004. to appear.

Timothy A. Davis. Algorithm 8xx: A concise sparse Cholesky factorization package. Technical report, Department of Computer and Information Science and Engineering, University of Florida, 2004.

George Karapis. Metis: Family of multilevel partioning algorithms. http://www-users.cs.umn.edu/ karypis/metis/, 2003.

José C. Pinheiro and Douglas M. Bates. *Mixed-Effects Models in S and S-PLUS*. Springer, 2000. ISBN 0-387-98957-9.

Judith D. Singer and John B. Willett. *Applied Longitudinal Data Analysis*. Oxford University Press, 2003. ISBN 0-19-515296-4.