NAG Library Function Document

nag_hier_mixed_init (g02jcc)

1 Purpose

nag_hier_mixed_init (g02jcc) preprocesses a dataset prior to fitting a linear mixed effects regression model of the following form via either nag_reml_hier_mixed_regsn (g02jdc) or nag_ml_hier_mixe d_regsn (g02jec).

2 Specification

```
#include <nag.h>
#include <nagg02.h>
```

```
void nag_hier_mixed_init (Nag_OrderType order, Integer n, Integer ncol,
     const double dat[], Integer pddat, const Integer levels[],
     const double y[], const double wt[], const Integer fixed[],
     Integer lfixed, Integer nrndm, const Integer rndm[], Integer lrndm,
     Integer *nff, Integer *nlsv, Integer *nrf, double rcomm[],
     Integer lrcomm, Integer icomm[], Integer licomm, NagError *fail)
```

3 Description

nag_hier_mixed_init (g02jcc) must be called prior to fitting a linear mixed effects regression model with either nag_reml_hier_mixed_regsn (g02jdc) or nag_ml_hier_mixed_regsn (g02jec).

The model fitting functions nag_reml_hier_mixed_regsn (g02jdc) and nag_ml_hier_mixed_regsn (g02jec) fit a model of the following form:

$$y = X\beta + Z\nu + \epsilon$$

where y is a vector of n observations on the dependent variable,

X is an n by p design matrix of *fixed* independent variables,

β is a vector of p unknown *fixed effects*,

Z is an n by q design matrix of *random* independent variables,

ν is a vector of length q of unknown *random effects*,

ϵ is a vector of length n of unknown random errors,

and ν and ϵ are Normally distributed with expectation zero and variance/covariance matrix defined by

$$\mathrm{Var}\begin{bmatrix} \nu \\ \epsilon \end{bmatrix} = \begin{bmatrix} G & 0 \\ 0 & R \end{bmatrix}$$

where $R = \sigma_R^2 I$, I is the $n \times n$ identity matrix and G is a diagonal matrix.

Case weights can be incorporated into the model by replacing X and Z with $W_c^{1/2} X$ and $W_c^{1/2} Z$ respectively where W_c is a diagonal weight matrix.

4 References

None.

5 Arguments

1: **order** – Nag_OrderType *Input*

On entry: the **order** argument specifies the two-dimensional storage scheme being used, i.e., row-major ordering or column-major ordering. C language defined storage is specified by **order** = Nag_RowMajor. See Section 2.3.1.3 in How to Use the NAG Library and its Documentation for a more detailed explanation of the use of this argument.

Constraint: **order** = Nag_RowMajor or Nag_ColMajor.

2: **n** – Integer *Input*

On entry: n, the number of observations.

The effective number of observations, that is the number of observations with nonzero weight (see **wt** for more detail), must be greater than the number of fixed effects in the model (as returned in **nff**).

Constraint: **n** ≥ 1.

3: **ncol** – Integer *Input*

On entry: the number of columns in the data matrix, **dat**.

Constraint: **ncol** ≥ 0.

4: **dat**[*dim*] – const double *Input*

Note: the dimension, *dim*, of the array **dat** must be at least

$\max(1, \mathbf{pddat} \times \mathbf{ncol})$ when **order** = Nag_ColMajor;
$\max(1, \mathbf{n} \times \mathbf{pddat})$ when **order** = Nag_RowMajor.

Where $\mathbf{DAT}(i, j)$ appears in this document, it refers to the array element

$\mathbf{dat}[(j-1) \times \mathbf{pddat} + i - 1]$ when **order** = Nag_ColMajor;
$\mathbf{dat}[(i-1) \times \mathbf{pddat} + j - 1]$ when **order** = Nag_RowMajor.

On entry: a matrix of data, with $\mathbf{DAT}(i, j)$ holding the ith observation on the jth variable. The two design matrices X and Z are constructed from **dat** and the information given in **fixed** (for X) and **rndm** (for Z).

Constraint: if $\mathbf{levels}[j-1] \neq 1, 1 \leq \mathbf{DAT}(i, j) \leq \mathbf{levels}[j-1]$.

5: **pddat** – Integer *Input*

On entry: the stride separating row or column elements (depending on the value of **order**) in the array **dat**.

Constraints:

if **order** = Nag_ColMajor, **pddat** $\geq$ **n**;
if **order** = Nag_RowMajor, **pddat** $\geq$ **ncol**.

6: **levels**[**ncol**] – const Integer *Input*

On entry: $\mathbf{levels}[i-1]$ contains the number of levels associated with the ith variable held in **dat**.

If the ith variable is continuous or binary (i.e., only takes the values zero or one) then $\mathbf{levels}[i-1]$ must be set to 1. Otherwise the ith variable is assumed to take an integer value between 1 and $\mathbf{levels}[i-1]$, (i.e., the ith variable is discrete with $\mathbf{levels}[i-1]$ levels).

Constraint: $\mathbf{levels}[i-1] \geq 1$, for $i = 1, 2, \ldots, \mathbf{ncol}$.

7: **y**[**n**] – const double *Input*

On entry: y, the vector of observations on the dependent variable.

8: **wt**[**n**] – const double *Input*

On entry: optionally, the weights to be used in the weighted regression.

If $\mathbf{wt}[i-1] = 0.0$, the ith observation is not included in the model, in which case the effective number of observations is the number of observations with nonzero weights.

If weights are not provided then **wt** must be set to the null pointer, i.e., `(double *)0`, and the effective number of observations is **n**.

Constraint: if **wt** is not **NULL**, $\mathbf{wt}[i-1] \geq 0.0$, for $i = 1, 2, \ldots, \mathbf{n}$.

9: **fixed**[**lfixed**] – const Integer *Input*

On entry: defines the structure of the fixed effects design matrix, X.

fixed[0]
: The number of variables, N_F, to include as fixed effects (not including the intercept if present).

fixed[1]
: The fixed intercept flag which must contain 1 if a fixed intercept is to be included and 0 otherwise.

$\mathbf{fixed}[2+i-1]$
: The column of **DAT** holding the ith fixed variable, for $i = 1, 2, \ldots, \mathbf{fixed}[0]$.

See Section 9.1 for more details on the construction of X.

Constraints:

$\mathbf{fixed}[0] \geq 0$;
$\mathbf{fixed}[1] = 0$ or 1;
$1 \leq \mathbf{fixed}[2+i-1] \leq \mathbf{ncol}$, for $i = 1, 2, \ldots, \mathbf{fixed}[0]$.

10: **lfixed** – Integer *Input*

On entry: length of the vector **fixed**.

Constraint: $\mathbf{lfixed} \geq 2 + \mathbf{fixed}[0]$.

11: **nrndm** – Integer *Input*

On entry: the second dimension of the random effects design matrix **RNDM**.

Constraint: $\mathbf{nrndm} > 0$.

12: **rndm**[**lrndm** × **nrndm**] – const Integer *Input*

Note: where $\mathbf{RNDM}(i, j)$ appears in this document, it refers to the array element

$\mathbf{rndm}[(j-1) \times \mathbf{lrndm} + i - 1]$ when **order** = Nag_ColMajor;
$\mathbf{rndm}[(i-1) \times \mathbf{nrndm} + j - 1]$ when **order** = Nag_RowMajor.

On entry: $\mathbf{RNDM}(i, j)$ defines the structure of the *random effects* design matrix, Z. The bth column of **RNDM** defines a block of columns in the design matrix Z.

$\mathbf{RNDM}(1, b)$
: The number of variables, N_{R_b}, to include as random effects in the bth block (not including the random intercept if present).

$\mathbf{RNDM}(2, b)$
: The random intercept flag which must contain 1 if block b includes a random intercept and 0 otherwise.

$\mathbf{RNDM}(2+i, b)$
: The column of **DAT** holding the ith random variable in the bth block, for $i = 1, 2, \ldots, \mathbf{RNDM}(1, b)$.

RNDM$(3 + N_{R_b}, b)$
: The number of subject variables, N_{S_b}, for the bth block. The subject variables define the nesting structure for this block.

RNDM$(3 + N_{R_b} + i, b)$
: The column of **DAT** holding the ith subject variable in the bth block, for $i = 1, 2, \ldots,$ **RNDM**$(3 + N_{R_b}, b)$.

See Section 9.2 for more details on the construction of Z.

Constraints:

RNDM$(1, b) \geq 0$;
RNDM$(2, b) = 0$ or 1;
at least one random variable or random intercept must be specified in each block, i.e., **RNDM**$(1, b)$ + **RNDM**$(2, b) > 0$;
the column identifiers associated with the random variables must be in the range 1 to **ncol**, i.e., $1 \leq$ **RNDM**$(2 + i, b) \leq$ **ncol**, for $i = 1, 2, \ldots,$ **RNDM**$(1, b)$;
RNDM$(3 + N_{R_b}, b) \geq 0$;
the column identifiers associated with the subject variables must be in the range 1 to **ncol**, i.e., $1 \leq$ **RNDM**$(3 + N_{R_b} + i, b) \leq$ **ncol**, for $i = 1, 2, \ldots,$ **RNDM**$(3 + N_{R_b}, b)$.

13: **lrndm** – Integer *Input*

On entry: maximum number of entries in any column of **RNDM**.

Constraint: **lrndm** $\geq \max_b (3 + N_{R_b} + N_{S_b})$.

14: **nff** – Integer * *Output*

On exit: p, the number of fixed effects estimated, i.e., the number of columns in the design matrix X.

15: **nlsv** – Integer * *Output*

On exit: the number of levels for the overall subject variable (see Section 9.2 for a description of what this means). If there is no overall subject variable, **nlsv** $= 1$.

16: **nrf** – Integer * *Output*

On exit: the number of random effects estimated in each of the overall subject blocks. The number of columns in the design matrix Z is given by $q =$ **nrf** $\times$ **nlsv**.

17: **rcomm**[**lrcomm**] – double *Communication Array*

On exit: communication array as required by the analysis functions nag_reml_hier_mixed_regsn (g02jdc) and nag_ml_hier_mixed_regsn (g02jec).

18: **lrcomm** – Integer *Input*

On entry: the dimension of the array **rcomm**.

Constraint: **lrcomm** $\geq$ **nrf** $\times$ **nlsv** + **nff** + **nff** $\times$ **nlsv** + **nrf** $\times$ **nlsv** + **nff** + 2.

19: **icomm**[**licomm**] – Integer *Communication Array*

On exit: if **licomm** $= 2$, **icomm**[0] holds the minimum required value for **licomm** and **icomm**[1] holds the minimum required value for **lrcomm**, otherwise **icomm** is a communication array as required by the analysis functions nag_reml_hier_mixed_regsn (g02jdc) and nag_ml_hier_mix ed_regsn (g02jec).

20: **licomm** – Integer *Input*

On entry: the dimension of the array **icomm**.

Constraint: **licomm** $= 2$ or
licomm $\geq 34 + N_F \times (\mathrm{MFL} + 1) +$ **nrndm** $\times \mathrm{MNR} \times \mathrm{MRL} + (\mathrm{LRNDM} + 2) \times$ **nrndm** $+$ **ncol** $+ \mathrm{LDID} \times \mathrm{LB}$,

where

$$\mathrm{MNR} = \max_b (N_{R_b}),$$

$$\mathrm{MFL} = \max_i (\textbf{levels}[\textbf{fixed}[2 + i - 1] - 1]),$$

$$\mathrm{MRL} = \max_{b,i} (\textbf{levels}[\textbf{RNDM}(2 + i, b) - 1]),$$

$$\mathrm{LDID} = \max_b N_{S_b},$$

$\mathrm{LB} =$ **nff** $+$ **nrf** $\times$ **nlsv**, and

$$\mathrm{LRNDM} = \underset{b}{max}(3 + N_{R_b} + N_{S_b})$$

21: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_BAD_PARAM

On entry, argument $\langle$*value*$\rangle$ had an illegal value.

NE_INT

On entry, **lfixed** $= \langle$*value*$\rangle$.
Constraint: **lfixed** $\geq \langle$*value*$\rangle$.

On entry, **licomm** $= \langle$*value*$\rangle$.
Constraint: **licomm** $\geq \langle$*value*$\rangle$.

On entry, **lrcomm** $= \langle$*value*$\rangle$.
Constraint: **lrcomm** $\geq \langle$*value*$\rangle$.

On entry, **lrndm** $= \langle$*value*$\rangle$.
Constraint: **lrndm** $\geq \langle$*value*$\rangle$.

On entry, **n** $= \langle$*value*$\rangle$.
Constraint: **n** ≥ 1.

On entry, **ncol** $= \langle$*value*$\rangle$.
Constraint: **ncol** ≥ 0.

On entry, **nrndm** $= \langle$*value*$\rangle$.
Constraint: **nrndm** > 0.

NE_INT_2

On entry, **pddat** $= \langle value \rangle$ and **n** $= \langle value \rangle$.
Constraint: **pddat** $\geq$ **n**.

On entry, **pddat** $= \langle value \rangle$ and **ncol** $= \langle value \rangle$.
Constraint: **pddat** $\geq$ **ncol**.

NE_INT_ARRAY

On entry, index of fixed variable j is less than 1 or greater than **ncol**: $j = \langle value \rangle$, index $= \langle value \rangle$ and **ncol** $= \langle value \rangle$.

On entry, index of random variable j in random statement i is less than 1 or greater than **ncol**: $i = \langle value \rangle$, $j = \langle value \rangle$, index $= \langle value \rangle$ and **ncol** $= \langle value \rangle$.

On entry, invalid value for fixed intercept flag: value $= \langle value \rangle$.

On entry, invalid value for random intercept flag for random statement i: $i = \langle value \rangle$, value $= \langle value \rangle$.

On entry, **levels**$[\langle value \rangle] = \langle value \rangle$.
Constraint: **levels**$[i-1] \geq 1$.

On entry, must be at least one parameter, or an intercept in each random statement i: $i = \langle value \rangle$.

On entry, nesting variable j in random statement i has one level: $j = \langle value \rangle$, $i = \langle value \rangle$.

On entry, number of fixed parameters, $\langle value \rangle$ is less than zero.

On entry, number of random parameters for random statement i is less than 0: $i = \langle value \rangle$, number of parameters $= \langle value \rangle$.

On entry, number of subject parameters for random statement i is less than 0: $i = \langle value \rangle$, number of parameters $= \langle value \rangle$.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 2.7.6 in How to Use the NAG Library and its Documentation for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 2.7.5 in How to Use the NAG Library and its Documentation for further information.

NE_REAL_ARRAY

On entry, no observations due to zero weights.

On entry, variable j of observation i is less than 1 or greater than **levels**$[j-1]$: $i = \langle value \rangle$, $j = \langle value \rangle$, value $= \langle value \rangle$, **levels**$[j-1] = \langle value \rangle$.

On entry, **wt**$[\langle value \rangle] = \langle value \rangle$.
Constraint: **wt**$[i-1] \geq 0.0$.

NE_TOO_MANY

On entry, more fixed factors than observations, **n** $= \langle value \rangle$.
Constraint: **n** $\geq \langle value \rangle$.

7 Accuracy

Not applicable.

8 Parallelism and Performance

nag_hier_mixed_init (g02jcc) is not threaded in any implementation.

9 Further Comments

9.1 Construction of the *fixed effects* design matrix, *X*

Let

N_F denote the number of fixed variables, that is **fixed**$[0] = N_F$;

F_j denote the jth fixed variable, that is the vector of values held in the kth column of **DAT** when **fixed**$[2 + j - 1] = k$;

F_{ij} denote the ith element of F_j;

$L(F_j)$ denote the number of levels for F_j, that is $L(F_j) =$ **levels**[**fixed**$[2 + j - 1] - 1]$;

$D_v(F_j)$ denoted an indicator function that returns a vector of values whose ith element is 1 if $F_{ij} = v$ and 0 otherwise.

The design matrix for the *fixed effects*, X, is constructed as follows:

set k to zero and the flag done_first to false;

if a fixed intercept is included, that is **fixed**$[1] = 1$,

set the first column of X to a vector of 1s;

set $k = k + 1$;

set done_first to true;

loop over each fixed variable, so for each $j = 1, 2, \ldots, N_F$,

if $L(F_j) = 1$,

set the kth column of X to be F_j;

set $k = k + 1$;

else

if done_first is false then

set the $L(F_j)$ columns, k to $k + L(F_j) - 1$, of X to $D_v(F_j)$, for $v = 1, 2, \ldots, L(F_j)$;

set $k = k + L(F_j)$;

set done_first to true;

else

set the $L(F_j) - 1$ columns, k to $k + L(F_j) - 2$, of X to $D_v(F_j)$, for $v = 2, 3, \ldots, L(F_j)$;

set $k = k + L(F_j) - 1$.

The number of columns in the design matrix, X, is therefore given by

$$p = 1 + \sum_{j=1}^{N_F} (\textbf{levels}[\textbf{fixed}[2 + j - 1] - 1] - 1).$$

This quantity is returned in **nff**.

In summary, nag_hier_mixed_init (g02jcc) converts all non-binary categorical variables (i.e., where $L(F_j) > 1$) to dummy variables. If a fixed intercept is included in the model then the first level of all such variables is dropped. If a fixed intercept is not included in the model then the first level of all such variables, other than the first, is dropped. The variables are added into the model in the order they are specified in **fixed**.

9.2 Construction of *random effects* design matrix, *Z*

Let

N_{R_b} denote the number of random variables in the bth random statement, that is $N_{R_b} = \mathbf{RNDM}(1, b)$;

R_{jb} denote the jth random variable from the bth random statement, that is the vector of values held in the kth column of **DAT** when $\mathbf{RNDM}(2 + j, b) = k$;

R_{ijb} denote the ith element of R_{jb};

$L(R_{jb})$ denote the number of levels for R_{jb}, that is $L(R_{jb}) = \mathbf{levels}[\mathbf{RNDM}(2 + j, b) - 1]$;

$D_v(R_{jb})$ denoted an indicator function that returns a vector of values whose ith element is 1 if $R_{ijb} = v$ and 0 otherwise;

N_{S_b} denote the number of subject variables in the bth random statement, that is $N_{S_b} = \mathbf{RNDM}(3 + N_{R_b}, b)$;

S_{jb} denote the jth subject variable from the bth random statement, that is the vector of values held in the kth column of **DAT** when $\mathbf{RNDM}(3 + N_{R_b} + j, b) = k$;

S_{ijb} denote the ith element of S_{jb};

$L(S_{jb})$ denote the number of levels for S_{jb}, that is $L(S_{jb}) = \mathbf{levels}[\mathbf{RNDM}(3 + N_{R_b} + j, b) - 1]$;

$I_b\left(s_1, s_2, \ldots, s_{N_{S_b}}\right)$ denoted an indicator function that returns a vector of values whose ith element is 1 if $S_{ijb} = s_j$ for all $j = 1, 2, \ldots, N_{S_b}$ and 0 otherwise.

The design matrix for the *random effects*, Z, is constructed as follows:

set k to zero;

loop over each random statement, so for each $b = 1, 2, \ldots, \mathbf{nrndm}$,

loop over each level of the last subject variable, so for each $s_{N_{S_b}} = 1, 2, \ldots, L\left(R_{N_{S_b} b}\right)$,

⋮

loop over each level of the second subject variable, so for each $s_2 = 1, 2, \ldots, L(R_{2b})$,

loop over each level of the first subject variable, so for each $s_1 = 1, 2, \ldots, L(R_{1b})$,

if a random intercept is included, that is $\mathbf{RNDM}(2, b) = 1$,

set the kth column of Z to $I_b\left(s_1, s_2, \ldots, s_{N_{S_b}}\right)$;

set $k = k + 1$;

loop over each random variable in the bth random statement, so for each $j = 1, 2, \ldots, N_{R_b}$,

if $L(R_{jb}) = 1$,

set the kth column of Z to $R_{jb} \times I_b\left(s_1, s_2, \ldots, s_{N_{S_b}}\right)$ where $\times$ indicates an element-wise multiplication between the two vectors, R_{jb} and $I_b(\ldots)$;

set $k = k + 1$;

else

set the $L(R_{bj})$ columns, k to $k + L(R_{bj})$, of Z to $D_v(R_{jb}) \times I_b\left(s_1, s_2, \ldots, s_{N_{S_b}}\right)$, for $v = 1, 2, \ldots, L(R_{jb})$. As before, $\times$ indicates an element-wise multiplication between the two vectors, $D_v(\ldots)$ and $I_b(\ldots)$;

set $k = k + L(R_{jb})$.

In summary, each column of **RNDM** defines a block of consecutive columns in Z. nag_hier_mixed_init (g02jcc) converts all non-binary categorical variables (i.e., where $L(R_{jb})$ or $L(S_{jb}) > 1$) to dummy variables. All random variables defined within a column of **RNDM** are nested within all subject variables defined in the same column of **RNDM**. In addition each of the subject variables are nested within each other, starting with the first (i.e., each of the $R_{jb}, j = 1, 2, \ldots, N_{R_b}$ are nested within S_{1b} which in turn is nested within S_{2b}, which in turn is nested within S_{3b}, etc.).

If the last subject variable in each column of **RNDM** are the same (i.e., $S_{N_{S_1}1} = S_{N_{S_2}2} = \ldots = S_{N_{S_b}b}$) then all random effects in the model are nested within this variable. In such instances the last subject variable ($S_{N_{S_1}1}$) is called the overall subject variable. The fact that all of the random effects in the model are nested within the overall subject variable means that $Z^{\mathrm{T}}Z$ is block diagonal in structure. This fact can be utilised to improve the efficiency of the underlying computation and reduce the amount of internal storage required. The number of levels in the overall subject variable is returned in $\mathbf{nlsv} = L\left(S_{N_{S_1}1}\right)$.

If the last k subject variables in each column of **RNDM** are the same, for $k > 1$ then the overall subject variable is defined as the interaction of these k variables and

$$\mathbf{nlsv} = \prod_{j=N_{S_1}-k+1}^{N_{S_1}} L(S_{j1}).$$

If there is no overall subject variable then $\mathbf{nlsv} = 1$.

The number of columns in the design matrix Z is given by $q = \mathbf{nrf} \times \mathbf{nlsv}$.

9.3 The rndm argument

To illustrate some additional points about the **rndm** argument, we assume that we have a dataset with three discrete variables, V_1, V_2 and V_3, with 2, 4 and 3 levels respectively, and that V_1 is in the first column of **DAT**, V_2 in the second and V_3 the third. Also assume that we wish to fit a model containing V_1 along with V_2 nested within V_3, as random effects. In order to do this the **RNDM** matrix requires two columns:

$$\mathbf{RNDM} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 2 \\ 0 & 1 \\ 0 & 3 \end{pmatrix}$$

The first column, $(1, 0, 1, 0, 0)$, indicates one random variable ($\mathbf{RNDM}(1, 1) = 1$), no intercept ($\mathbf{RNDM}(2, 1) = 0$), the random variable is in the first column of **DAT** ($\mathbf{RNDM}(3, 1) = 1$), there are no subject variables; as no nesting is required for V_1 ($\mathbf{RNDM}(4, 1) = 0$). The last element in this column is ignored.

The second column, $(1, 0, 2, 1, 3)$, indicates one random variable ($\mathbf{RNDM}(1, 2) = 1$), no intercept ($\mathbf{RNDM}(2, 2) = 0$), the random variable is in the second column of **DAT** ($\mathbf{RNDM}(3, 2) = 2$), there is one subject variable ($\mathbf{RNDM}(4, 2) = 1$), and the subject variable is in the third column of **dat** ($\mathbf{RNDM}(5, 2) = 3$).

The corresponding Z matrix would have 14 columns, with 2 coming from V_1 and 12 (4×3) from V_2 nested within V_3. The, symmetric, $Z^{\mathrm{T}}Z$ matrix has the form

$$\begin{pmatrix}
- & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & - & - & - & - & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & - & - & - & - & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & - & - & - & - & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
- & - & 0 & 0 & 0 & 0 & - & - & - & - & 0 & 0 & 0 & 0 \\
- & - & 0 & 0 & 0 & 0 & - & - & - & - & 0 & 0 & 0 & 0 \\
- & - & 0 & 0 & 0 & 0 & - & - & - & - & 0 & 0 & 0 & 0 \\
- & - & 0 & 0 & 0 & 0 & - & - & - & - & 0 & 0 & 0 & 0 \\
- & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & - & - & - \\
- & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & - & - & - \\
- & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & - & - & - \\
- & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & - & - & -
\end{pmatrix}$$

where 0 indicates a structural zero, i.e., it always takes the value 0, irrespective of the data, and $-$ a value that is not a structural zero. The first two rows and columns of $Z^{\mathrm{T}}Z$ correspond to V_1. The block diagonal matrix in the 12 rows and columns in the bottom right correspond to V_2 nested within V_3. With the 4×4 blocks corresponding to the levels of V_2. There are three blocks as the subject variable (V_3) has three levels.

The model fitting functions, nag_reml_hier_mixed_regsn (g02jdc) and nag_ml_hier_mixed_regsn (g02jec), use the sweep algorithm to calculate the log-likelihood function for a given set of variance components. This algorithm consists of moving down the diagonal elements (called pivots) of a matrix which is similar in structure to $Z^{\mathrm{T}}Z$, and updating each element in that matrix. When using the k diagonal element of a matrix A, an element $a_{ij}, i \neq k, j \neq k$, is adjusted by an amount equal to $a_{ik}a_{ij}/a_{kk}$. This process can be referred to as sweeping on the kth pivot. As there are no structural zeros in the first row or column of the above $Z^{\mathrm{T}}Z$, sweeping on the first pivot of $Z^{\mathrm{T}}Z$ would alter each element of the matrix and therefore destroy the structural zeros, i.e., we could no longer guarantee they would be zero.

Reordering the **RNDM** matrix to

$$\mathbf{RNDM} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 2 & 1 \\ 1 & 0 \\ 3 & 0 \end{pmatrix}$$

i.e., the swapping the two columns, results in a $Z^{\mathrm{T}}Z$ matrix of the form

$$\begin{pmatrix}
- & - & - & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & - \\
- & - & - & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & - \\
- & - & - & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & - \\
- & - & - & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & - \\
0 & 0 & 0 & 0 & - & - & - & - & 0 & 0 & 0 & 0 & - & - \\
0 & 0 & 0 & 0 & - & - & - & - & 0 & 0 & 0 & 0 & - & - \\
0 & 0 & 0 & 0 & - & - & - & - & 0 & 0 & 0 & 0 & - & - \\
0 & 0 & 0 & 0 & - & - & - & - & 0 & 0 & 0 & 0 & - & - \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & - & - & - & - & - \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & - & - & - & - & - \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & - & - & - & - & - \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & - & - & -
\end{pmatrix}$$

This matrix is identical to the previous one, except the first two rows and columns have become the last two rows and columns. Sweeping a matrix, $A = \{a_{ij}\}$, of this form on the first pivot will only affect those elements a_{ij}, where $a_{i1} \neq 0$ and $a_{1j} \neq 0$, which is only the 13th and 14th row and columns, and the top left hand block of 4 rows and columns. The block diagonal nature of the first 12 rows and columns therefore greatly reduces the amount of work the algorithm needs to perform.

nag_hier_mixed_init (g02jcc) constructs the $Z^{\mathrm{T}}Z$ as specified by the **RNDM** matrix, and does not attempt to reorder it to improve performance. Therefore for best performance some thought is required on what ordering to use. In general it is more efficient to structure **RNDM** in such a way that the first row relates to the deepest level of nesting, the second to the next level, etc..

10 Example

See Section 10 in nag_reml_hier_mixed_regsn (g02jdc) and nag_ml_hier_mixed_regsn (g02jec).