

# Package ‘HMMEsolver’

January 20, 2025

**Type** Package

**Title** A Fast Solver for Henderson Mixed Model Equation via Row Operations

**Version** 0.1.2

**Description**

Consider the linear mixed model with normal random effects. A typical method to solve Henderson's Mixed Model Equations (HMME) is recursive estimation of the fixed effects and random effects. We provide a fast, stable, and scalable solver to the HMME without computing matrix inverse. See Kim (2017) <[arXiv:1710.09663](#)> for more details.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**Imports** Rcpp, Rdpack

**LinkingTo** Rcpp, RcppArmadillo

**RdMacros** Rdpack

**RoxygenNote** 6.1.1

**NeedsCompilation** yes

**Author** Jiwoong Kim [aut, cre]

**Maintainer** Jiwoong Kim <[jwboys26@gmail.com](mailto:jwboys26@gmail.com)>

**Repository** CRAN

**Date/Publication** 2019-01-05 00:40:03 UTC

## Contents

HMMEsolver-package . . . . .	2
SolveHMME . . . . .	2

<b>Index</b>	<b>4</b>
--------------	----------

---

HMMesolver-package      *HMMesolver Package*

---

### Description

Consider the linear mixed model with normal random effects,

$$Y = X\beta + Zv + \epsilon$$

where  $\beta$  and  $v$  are vectors of fixed and random effects. One of most popular methods to solve the Henderson's Mixed Model Equation related to the problem is EM-type algorithm. Its drawback, however, comes from repetitive matrix inversion during recursive estimation steps. Kim (2017) proposed a novel method of avoiding such difficulty, letting the estimation more fast, stable, and scalable.

---

SolveHMME      *Solve Henderson's Mixed Model Equation.*

---

### Description

Consider a linear mixed model with normal random effects,

$$Y_{ij} = X_{ij}^T \beta + v_i + \epsilon_{ij}$$

where  $i = 1, \dots, n$ ,  $j = 1, \dots, m$ , or it can be equivalently expressed using matrix notation,

$$Y = X\beta + Zv + \epsilon$$

where  $Y \in \mathbb{R}^{nm}$  is a known vector of observations,  $X \in \mathbb{R}^{nm \times p}$  and  $Z \in \mathbb{R}^{nm \times n}$  design matrices for  $\beta$  and  $v$  respectively,  $\beta \in \mathbb{R}^p$  and  $v \in \mathbb{R}^n$  unknown vectors of fixed effects and random effects where  $v_i \sim N(0, \lambda_i)$ , and  $\epsilon \in \mathbb{R}^{nm}$  an unknown vector random errors independent of random effects. Note that  $Z$  does not need to be provided by a user since it is automatically created accordingly to the problem specification.

### Usage

SolveHMME(X, Y, Mu, Lambda)

### Arguments

X	an $(nm \times p)$ design matrix for $\beta$ .
Y	a length- $nm$ vector of observations.
Mu	a length- $nm$ vector of initial values for $\mu_i = E(Y_i)$ .
Lambda	a length- $n$ vector of initial values for $\lambda$ , variance of $v_i \sim N(0, \lambda_i)$

**Value**

a named list containing

**beta** a length- $p$  vector of BLUE  $\hat{\beta}_a$ .

**v** a length- $n$  vector of BLUP  $\hat{v}$ .

**leverage** a length- $(mn + n)$  vector of leverages.

**References**

Henderson CR, Kempthorne O, Searle SR, von Krosigk CM (1959). “The Estimation of Environmental and Genetic Trends from Records Subject to Culling.” *Biometrics*, **15**(2), 192. ISSN 0006341X, doi: [10.2307/2527669](https://doi.org/10.2307/2527669), <http://www.jstor.org/stable/2527669?origin=crossref>.

Robinson GK (1991). “That BLUP is a Good Thing: The Estimation of Random Effects.” *Statistical Science*, **6**(1), 15–32. ISSN 0883-4237, doi: [10.1214/ss/1177011926](https://doi.org/10.1214/ss/1177011926), <http://projecteuclid.org/euclid.ss/1177011926>.

McLean RA, Sanders WL, Stroup WW (1991). “A Unified Approach to Mixed Linear Models.” *The American Statistician*, **45**(1), 54. ISSN 00031305, doi: [10.2307/2685241](https://doi.org/10.2307/2685241), <http://www.jstor.org/stable/2685241?origin=crossref>.

Kim J (2017). “A Fast Algorithm for Solving Henderson’s Mixed Model Equation.” *ArXiv e-prints*.

**Examples**

```
## small setting for data generation
n = 100; m = 2; p = 2
nm = n*m; nmp = n*m*p

## generate artificial data
X = matrix(rnorm(nmp, 2,1), nm,p) # design matrix
Y = rnorm(nm, 2,1) # observation

Mu = rep(1, times=nm)
Lambda = rep(1, times=n)

## solve
ans = SolveHMME(X, Y, Mu, Lambda)
```

# Index

HMMesolver-package, [2](#)

SolveHMM, [2](#)