Multidimensional smoothing using sparse mixed models in R package LMMsolver

BMS-ANed: Modern Mixed Models

Martin Boer and Bart-Jan van Rossum

Biometris, Wageningen University and Research

Rotterdam, 13 May 2022

Introduction

- Aim is to keep all the calculations **sparse**.
- Solving **sparse** mixed model equations C: use R-packages like **spam**, **Matrix**.
- Calculation of Effective dimensions, using **reverse** Automated Differentiation of Cholesky.
- This is connected to idea of **sparse inverse** of C, used in software like asreml and INLA.
- Use a **sparse tranformation** of P-splines to mixed models, using the properties of B-splines.
- Extension to **multidimensional** P-splines.
- Implemented in CRAN R-package **LMMsolver**.

Linear Mixed Models and REML

Robin Thompson, Wageningen 2008

- **REML**: Residual Maximum Likelihood
- Patterson, H. D., & Thompson, R. (1971). *Recovery of inter-block information when block sizes are unequal.* Biometrika, 58(3), 545-554.
- Johnson, D. L., & Thompson, R. (1995). *Restricted maximum likelihood estimation of variance components for univariate animal models using sparse matrix techniques and a quasi-Newton procedure* Journal of Dairy Science, 78, 449–456.
- Thompson, R. (2019). *Desert island papers—A life in variance parameter and quantitative genetic parameter estimation reviewed using 16 papers* Journal of Animal Breeding and Genetics, 136(4), 230-242.

Example Sparse Linear System

$$\begin{pmatrix} a_{11} & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 \\ 0 & 0 & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

Forward solution:

$$\begin{aligned} x_1 &= b_1/a_{11} \\ x_2 &= b_2/a_{22} \\ x_3 &= b_3/a_{33} \\ x_4 &= (b_4 - a_{41}x_1 - a_{42}x_2 - a_{43}x_3)/a_{44} \end{aligned}$$

Sparse Linear System (2)

$$\begin{pmatrix} a_{11} & 0 & 0 & a_{14} \\ 0 & a_{22} & 0 & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

Backwards solution:

$$\begin{aligned} x_4 &= b_4/a_{44} \\ x_3 &= (b_3 - a_{34}x_4)/a_{33} \\ x_2 &= (b_2 - a_{24}x_4)/a_{22} \\ x_1 &= (b_4 - a_{41}x_4)/a_{11} \end{aligned}$$

Sparse Symmetric Matrix

$$\begin{pmatrix} a_{11} & 0 & 0 & a_{14} \\ 0 & a_{22} & 0 & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

- Cholesky decomposition $A = LL'$, L lower triangle matrix.
- Solve $LL'x = b$ in two steps:
  1. $Lz = b$ (forwards)
  2. $L'x = z$ (backwards)

Automated Differentiation

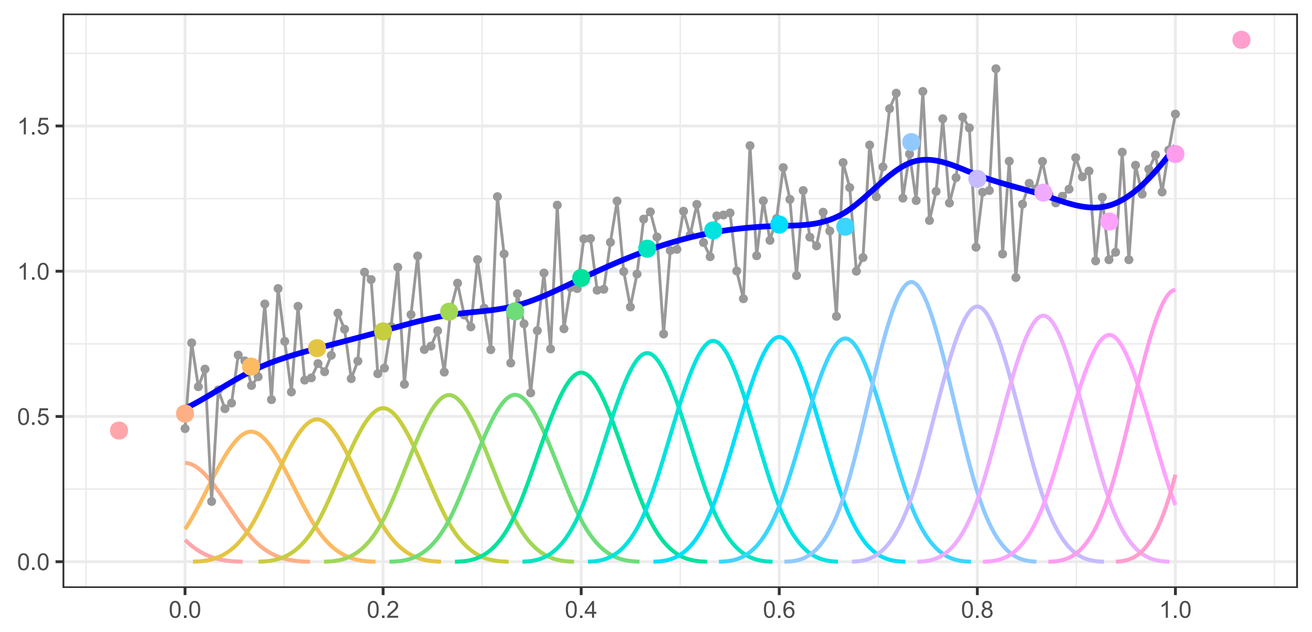
Speelpenning example (book Evaluating Derivatives, Griewank and Walther 2008), determinant of diagonal matrix, with dimension $n = 4$ as example here: `

$$f(x_1, x_2, x_3, x_4) = \left| \begin{pmatrix} x_1 & 0 & 0 & 0 \\ 0 & x_2 & 0 & 0 \\ 0 & 0 & x_3 & 0 \\ 0 & 0 & 0 & x_4 \end{pmatrix} \right| = x_1 x_2 x_3 x_4$$

Calculation of partial derivatives $\frac{\partial f}{\partial x_i}$ using reverse Automated Differentiation:

- $v_0 = 1$
- $v_i = v_{i-1} x_i$ for $i = 1 ... n$ (forward)
- $w_{n+1} = 1$
- $w_i = w_{i+1} x_i$ for $i = n...1$ (reverse)
- $\frac{\partial f}{\partial x_i} = v_{i-1} w_{i+1}$ for $i = 1 ... n$.

Example P-splines



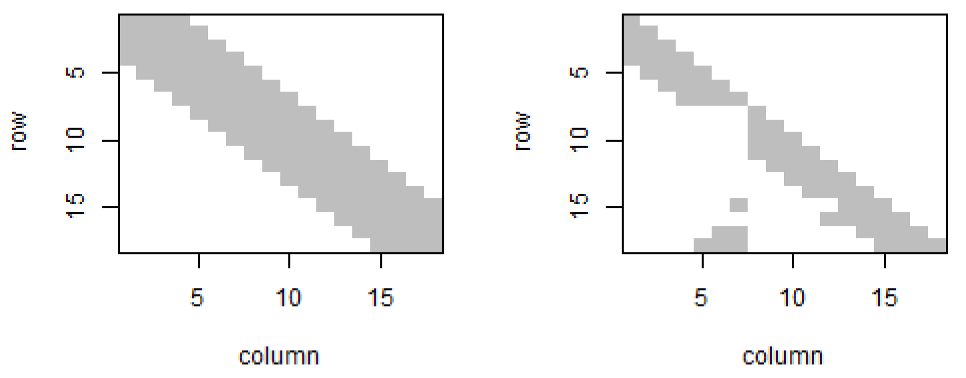
- Figure 1.1 (page 2) in The Book by Eilers and Marx (2021).
- Number of segments: nseg = 15
- Number of B-splines: $q = 18$
- Penalty: $\lambda = 0.1$

Sparse structure of P-splines

Minimize the objective function: $S = ||y - Ba||^2 + \lambda a' D' D a$ gives:

$$(B'B + \lambda D'D)\hat{a} = B'y$$

The grey area are the non-zero's of $C = B'B + \lambda D'D$, with Cholesky decomposition $LL' = C$.



Effective Dimensions

```
theta = c(1, lambda)
C = theta[1]*BtB + theta[2] * DtD
H1 = theta[1]*solve(C, BtB) # this step expensive!
H2 = theta[2]*solve(C, DtD) # this step expensive!
ed = c(sum(diag(H1)), sum(diag(H2)))
round(ed,2)
```

```
## [1] 12.87  5.13
```

The effective dimensions are given by partial derivatives of log-determinant:

$$\mathrm{ed}(\theta_1) = \theta_1 \frac{\partial \log|C|}{\partial \theta_1} = \theta_1 \mathrm{trace}[C^{-1}B'B]$$

$$\mathrm{ed}(\theta_2) = \theta_1 \frac{\partial \log|C|}{\partial \theta_2} = \theta_2 \mathrm{trace}[C^{-1}D'D]$$

$$\mathrm{ed}(\theta_1) + \mathrm{ed}(\theta_2) = q, \quad C = \theta_1 B'B + \theta_2 D'D$$

Automated Differentiation

If C is sparse, then both $\log|C|$ and the partial derivatives $\frac{\partial \log|C|}{\partial \theta_i}$ can be calculated in an efficient way, by **reverse Automated Differentiation of Cholesky algorithm** (S.P. Smith, 1995), implemented in C++/Rcpp in **LMMsolver**.

Initialization of **ADchol** object, list of semi-positive definite matrices with same dimension, in this example list of two:

```
lP <- list(BtB, DtD)
ADcholC <- LMMsolver::ADchol(lP)
```

Calculates partial derivatives of log-determinant, plus Effective Dimensions

```
pDeriv <- LMMsolver::dlogdet(ADcholC, theta)
ED <- as.numeric(theta * pDeriv)
round(ED, 2)
```

```
## [1] 12.87  5.13
```

Transformation to mixed model

Objective function $S = ||y - Ba||^2 + \lambda a' D' D a$

The matrix $D'D$ is singular, and in standard transformations to a mixed model the vector a is split in a **fixed** and **random** part. For example, the eigen decomposition

$$D'D = U\Gamma U^{'} = U_+\Gamma_+ U_+^{'}$$

can be used, resulting in

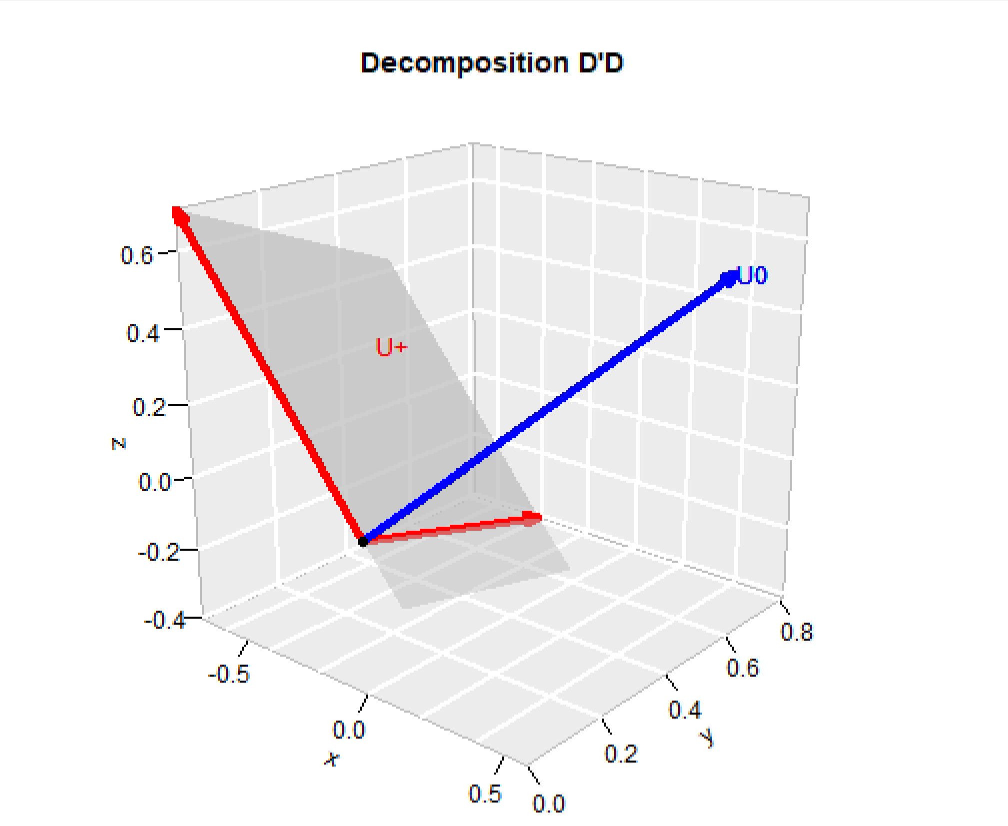
$$y = X\beta + Zu + e, \quad X = BU_0, \quad Z = BU_+\Gamma^{-1/2},$$

with

$$u \sim N(0, \sigma_u^2 I) \quad e \sim N(0, \sigma_e^2 I)$$

and U_0 is the null space of $D'D$.

Split in fixed and random part



Here first order differences (pord=1) are used, with $U_0 = 1_q$

Sparse Transformation

- The standard transformations split the q-dimensional space of the B-splines basis in:
  - d dimensional fixed effect, corresponding to the null-space U_0.
  - $q-d$ dimensional random effect.
- Another approach is to **add** a d-dimensional space:

$$S = ||y - Ba||^2 + \lambda(a'D'Da + b'b)$$

where b is a vector of length d, and is only added to the penalized part of S.

- The object function S can be rewritten as

$$S = ||y - Ba||^2 + \lambda \begin{pmatrix} a \\ b \end{pmatrix}' \begin{pmatrix} D'D & 0 \\ 0 & I_d \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$

- Or

$$S = ||y - B[I|0]u||^2 + \lambda u' \begin{pmatrix} D'D & 0 \\ 0 & I_d \end{pmatrix} u, \quad u = \begin{pmatrix} a \\ b \end{pmatrix}$$

Sparse Transformation (2)

- Continued..

$$S = ||y - B[I|0]u||^2 + \lambda u' \begin{pmatrix} D'D & 0 \\ 0 & I_d \end{pmatrix} u, \quad u = \begin{pmatrix} a \\ b \end{pmatrix}$$

- Define the following transformation, with A a $q \times d$ matrix ($|A'U_0| \neq 0$).

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} I_q & U_0 \\ A' & 0 \end{pmatrix} \begin{pmatrix} v \\ \beta \end{pmatrix}$$

- Substitution gives:

$$S = ||y - BU_0\beta - Bv||^2 + \lambda v'(D'D + AA')v$$

- Now the transformation to a mixed model is easy:

$$y = X\beta + Bv + e, \quad v \sim N(0, G), \quad G^{-1} = \lambda(D'D + AA')$$

Sparse transformation (3)

- Continued..

$$y = X\beta + Bv + e, \quad v \sim N(0, G), \quad G^{-1} = \lambda(D'D + AA')$$

- β is a vector of length d, and v is a vector of length q, so the total dimension is $d + q$. However, there are built-in constraints:

$$A'\hat{v} = 0_d$$

- We can choose any A ($|A'U_0| \neq 0$), in **LMMsolver** for second order differences and cubical B-splines

$$A' = \begin{pmatrix} 1/6 & 2/3 & 1/6 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1/6 & 2/3 & 1/6 \end{pmatrix}$$

which makes the random part of prediction function $f(x)$ at the boundaries equal to zero:

$$f_r(x_{min}) = f_r(x_{max}) = 0$$

Linear Mixed Models

- Mixed Model:

$$y = X\beta + Zu + e, \quad \mathrm{var}(u) = G, \quad \mathrm{var}(e) = R$$

- Mixed Model Equations:

$$\begin{pmatrix} X'R^{-1}X & X'R^{-1}Z \\ Z'R^{-1}X & Z'R^{-1}Z + G^{-1} \end{pmatrix} \begin{pmatrix} \hat{\beta} \\ \hat{u} \end{pmatrix} = \begin{pmatrix} X'R^{-1}y \\ Z'R^{-1}y \end{pmatrix}$$

- If matrices R^{-1}, G^{-1} and C are sparse, the mixed models are easy to solve using sparse Cholesky Decomposition.

Linear Mixed Models (2)

- Mixed Model:

$$y = X\beta + Zu + e, \quad \mathrm{var}(u) = G, \quad \mathrm{var}(e) = R$$

- REML log-likelihood:

$$\log L = \frac{1}{2}\left(-\log|R^{-1}| - \log|G^{-1}| + \log|C| + y'R^{-1}(y - X\hat{\beta} - Z\hat{u})\right)$$

$$C = \begin{pmatrix} X'R^{-1}X & X'R^{-1}Z \\ Z'R^{-1}X & Z'R^{-1}Z + G^{-1} \end{pmatrix}$$

- REML loglikelihood calculations fast, if matrices R^{-1}, G^{-1} and C are sparse.
- Same is true for the partial derivatives and the Effective Dimensions, using Automated Differentiation of the Cholesky Algorithm (S.P. Smith, 1995)
- Most transformations of penalized splines to mixed models are **dense**.
- The method implemented in **LMMsolver** use sparse mixed model P-splines.

LMMsolver

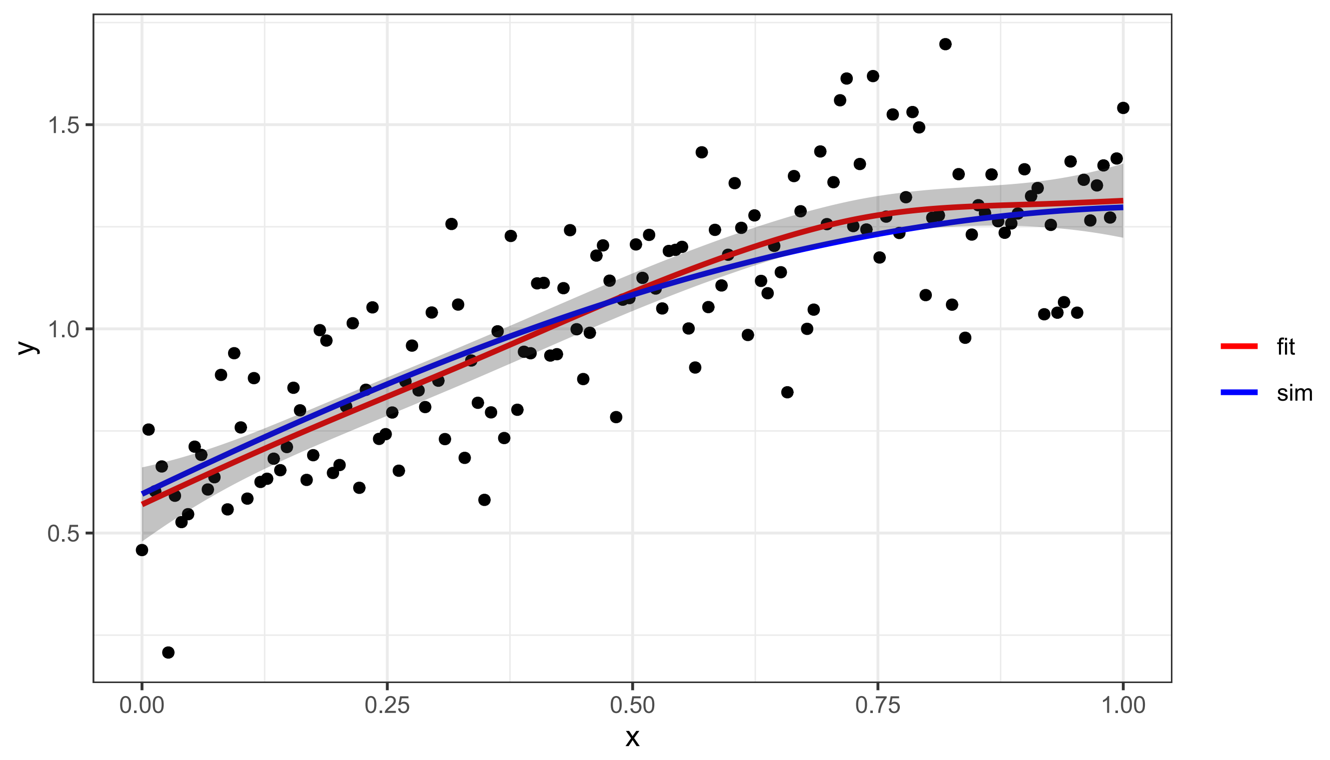
```r
library(LMMsolver)
obj <- LMMsolve(fixed = y~1,
                spline = ~spl1D(x, nseg = 15, pord=2),
                data = Data)
EDtbl <- summary(obj)
```

Term	Effective	Model	Nominal	Ratio	Penalty
(Intercept)	1.00000	1	1	1.0000000	0.0000000
lin(x)	1.00000	1	1	1.0000000	0.0000000
s(x)	2.25185	18	16	0.1407406	0.7750018
residual	145.74815	150	148	0.9847848	41.0211814

$$\mathrm{ed}_{\mathrm{tot}} = 150, \quad \hat{\sigma}_e^2 = \frac{1}{\theta_r} = 0.024, \quad \hat{\lambda} = \frac{\theta_s}{\theta_r} = 0.02$$

Make predictions

```
pred <- obtainSmoothTrend(obj, grid=ng, includeIntercept = TRUE)
```



Oats field trial example

- data from `agridat` package, and discussed in detail in Boer et al. 2020.
- There are 24 varieties in 3 replicates
- The plots were laid out in a single row

```
## Load data.
data(john.alpha, package = "agridat")
dat <- john.alpha
head(dat)
```

```
##   plot rep block gen  yield row col
## 1    1  R1    B1 G11 4.1172   1   1
## 2    2  R1    B1 G04 4.4461   2   1
## 3    3  R1    B1 G05 5.8757   3   1
## 4    4  R1    B1 G22 4.5784   4   1
## 5    5  R1    B2 G21 4.6540   5   1
## 6    6  R1    B2 G10 4.1736   6   1
```

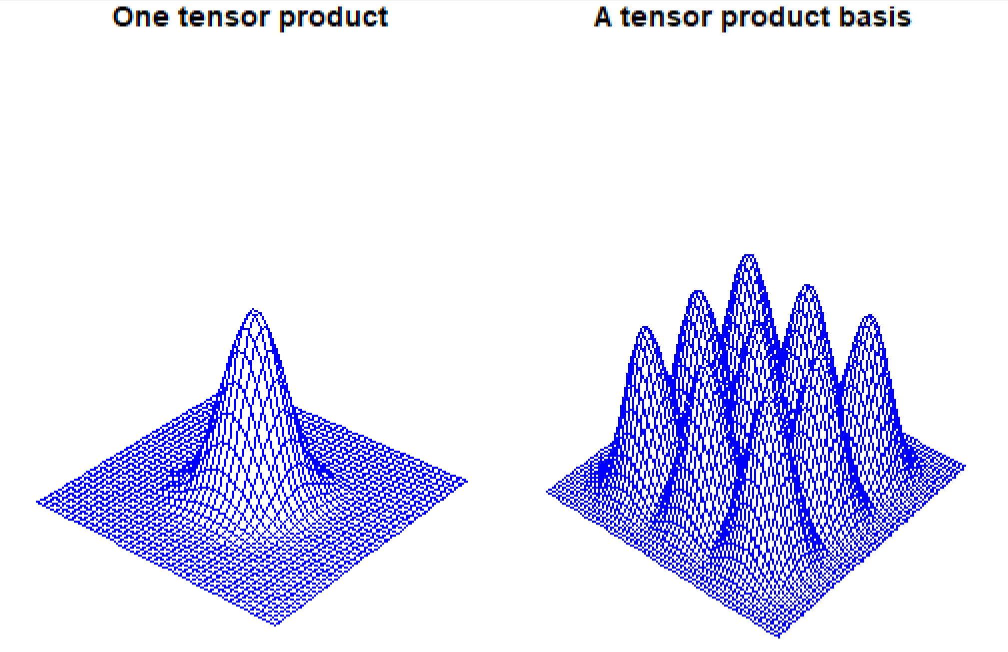
LMMsolver analysis oats data

```r
obj1 <- LMMsolve(fixed = yield ~ rep,
                 random = ~gen,
                 spline = ~spl1D(x=plot, nseg = 50),
                 data = dat)
EDtbl <- summary(obj1)
```

Term	Effective	Model	Nominal	Ratio	Penalty
(Intercept)	1.000000	1	1	1.0000000	0.000000
rep	2.000000	2	2	1.0000000	0.000000
lin(plot)	1.000000	1	1	1.0000000	0.000000
gen	19.498369	24	23	0.8477552	6.455573
s(plot)	4.331908	53	51	0.0849394	3261.172380
residual	44.169723	72	68	0.6495548	13.385048

$$\mathrm{ed}_{\mathrm{tot}} = 72$$

2D P-splines



Penalty: $P = \lambda_1(D_1^{'}D_1 \otimes I_2) + \lambda_2(I_1 \otimes D_2^{'}D_2)$

Eilers and Marx (2021), page 70-71.

2D example US precipitation

```
# Get precipitation data from spam
data(USprecip)
dat = data.frame(USprecip)
# only use observed data, # 5906 true records, as in SAP2014 paper.
dat = subset(dat, infill==1)
tic("LMMsolve time")
obj2 <- LMMsolve(fixed = anomaly ~ 1,
                 spline = ~spl2D(x1 = lon, x2 = lat, nseg = c(41, 41)),
                 data = dat)
toc()
```

```
## LMMsolve time: 1.67 sec elapsed
```

Same model, but not sparse takes several minutes to run: *Fast smoothing parameter separation in multidimensional generalized P-splines: the SAP algorithm*, Rodríguez-Álvares et al. Stat Comput, 2014.

The difference between computation time of original approach in **SAP2014**-paper and **LMMsolver** further increases if number of segments is increased.

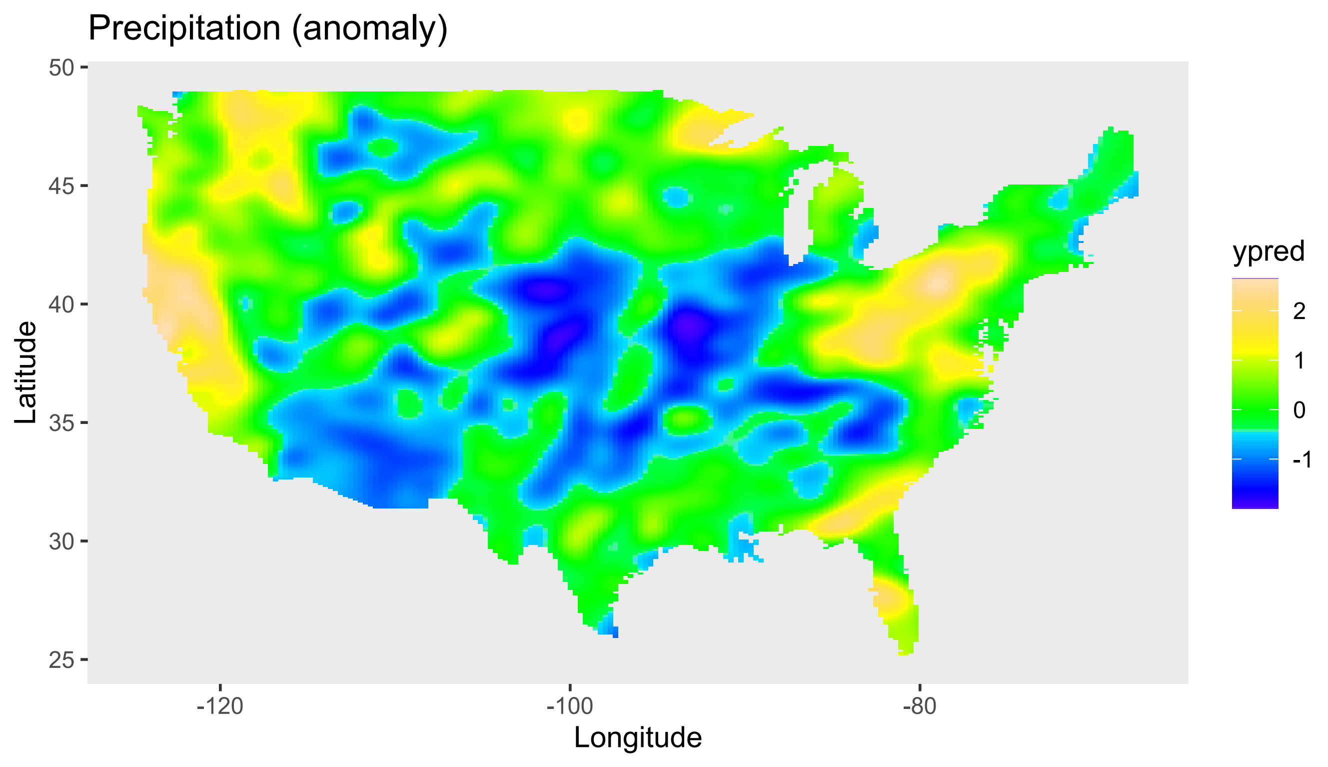
ED table US data

Term	Effective	Model	Nominal	Ratio	Penalty
(Intercept)	1.0000	1	1	1.0000000	0.0000000
lin(lon, lat)	3.0000	3	3	1.0000000	0.0000000
s(lon)	302.5973	1936	1932	0.1566238	0.2589899
s(lat)	409.0913	1936	1932	0.2117450	0.0768224
residual	5190.3114	5906	5902	0.8794157	13.5258044

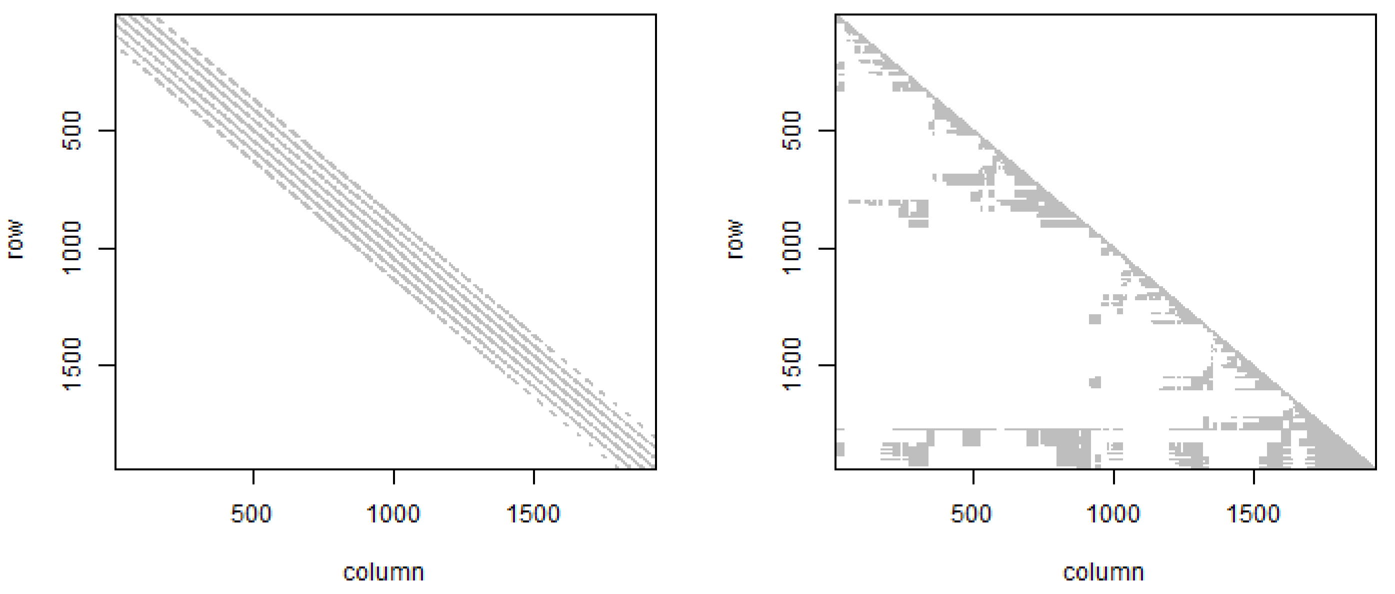
Total effective dimension: 5906

Predictions smooth trend

```
plotDat <- obtainSmoothTrend(obj2,
                             grid = c(200, 300),
                             includeIntercept = TRUE)
```



Mixed Model Eqn are sparse



Mixed Model Eqn are sparse (2)

```
LMMsolver::diagnosticsMME(obj2)
```

```
## Summary of matrix C
## Matrix object of class 'spam' of dimension 1940x1940,
##     with 75172 (row-wise) nonzero elements.
##     Density of the matrix is 2%.
## Class 'spam' (32-bit)
##
##  Summary of cholesky decomposition of C
## Matrix object of class 'spam' of dimension 1940x1940,
##     with 170112 (row-wise) nonzero elements.
##     Density of the matrix is 4.52%.
## Class 'spam' (32-bit)
```

Some references

- Boer, M. P., Wright, D., ..., van Eeuwijk, F. A. (2007). *A mixed-model quantitative trait loci (QTL) analysis for multiple-environment trial data using environmental covariables for QTL-by-environment interactions, with an example in maize.* Genetics, 177(3), 1801-1813.
- Li, W., Boer, M. P., Zheng, C., Joosen, R. V., & van Eeuwijk, F. A. (2021). *An IBD-based mixed model approach for QTL mapping in multiparental populations*. Theoretical and Applied Genetics, 134(11), 3643-3660.
- Rodriguez-Alvarez, M. X., Boer, M. P., van Eeuwijk, F. A., & Eilers, P. H. (2018). *Correcting for spatial heterogeneity in plant breeding experiments with P-splines. Spatial Statistics*, 23, 52-71.
- Boer, M. P., Piepho, H. P., & Williams, E. R. (2020). *Linear variance, P-splines and neighbour differences for spatial adjustment in field trials: how are they related?.* JABES, 25(4), 676-698.
- Pérez-Valencia, D. M., Rodríguez-Álvarez, M. X., Boer, M. P., ...,van Eeuwijk, F. A. (2022). *A two-stage approach for the spatio-temporal analysis of high-throughput phenotyping data.* Scientific Reports, 12(1), 1-16.

LMMsolver summary

- Can be used for solving LMM in general, with important role for (tensor product) P-splines.
- LMMsolver keeps most of the calculations **sparse**.
- Aim is to keep the syntax simple, not too many different options.
- Current release on CRAN: 1.0.2
- Further details: https://biometris.github.io/LMMsolver/index.html

To do list for next releases

- Make sure all the calculations are sparse:
  - For example, use `sparse.model.matrix()` function from Matrix library to make the design matrices for factors.
- Calculation of standard errors, without calculating C^{-1}.
- Add `predict` function
- Generalized Linear Mixed Models.

E-mail: **martin.boer@wur.nl**