SUPPLEMENTARY MATERIAL FOR “ADDITIVE MIXED MODELS FOR CORRELATED FUNCTIONAL DATA”

By Fabian Scheipl∗, Ana-Maria Staicu† and Sonja Greven∗

Ludwig-Maximilians-Universität München∗ and North Carolina State University†
OVERVIEW

This supplement is divided into three parts. Section A gives details about suitable identifiability constraints for the additive models for functional responses and an illustrative example in Section A.2. Section B provides more details about the simulation study in the main article: Section B.1 offers a detailed description of the data generating processes used for the simulation, Section B.2 gives the unabridged simulation results, and Section B.3 shows examples of the generated data and fitted models for the various settings. Section C is a fully reproducible and extended treatment of the Canadian Weather Data example.

APPENDIX A: IDENTIFIABILITY CONSTRAINTS FOR ADDITIVE MODELS FOR FUNCTIONAL RESPONSES

A.1. Deriving and imposing suitable constraints. The issue is that

\[ Y_{ij}(t) = g_0(t) + g(z_{1ij}, t) + B_{0i}(t) + \epsilon_{ij}(t), \]

is not identifiable in the sense that

\[ Y_{ij}(t) = g_0(t) + \bar{g}_{z1}(t) + (g(z_{1ij}, t) - \bar{g}_{z1}(t)) + \bar{B}_0(t) + (B_{0i}(t) - \bar{B}_0(t)) + \epsilon_{ij}(t) \]

with \( \bar{B}_0(t) = (\sum_i n_i)^{-1} \sum_i n_i B_{0i}(t), \bar{g}_{z1}(t) = n^{-1} \sum_{i,j} g(z_{1ij}, t) \) etc., with

\[ n = \sum_i n_i \]

yields exactly the same fit.

If we fit the model above with the constraints that \( \bar{g}_{z1}(t), \bar{B}_0(t) \) are constant zero, i.e. with \( (\sum_i n_i)^{-1} \sum_i n_i B_{0i}(t) = n^{-1} \sum_{i,j} g(z_{1ij}, t) = 0 \forall t \), we get a model with interpretable effects in the sense that:

- \( g_0(t) \) is the (smoothed) sample mean of \( Y(t) \),
- effects that vary over the index of \( Y \) are directly interpretable as deviations from this sample mean trajectory.

The default sum-to-zero constraints \( \sum_{i,t} f(z_i, t) = 0 \) implemented in \texttt{mgcv} do not yield effects that are interpretable like this.
Consider a function

\[ f(z, t) = \begin{pmatrix} f(z_1, t_1) \\ f(z_1, t_2) \\ \vdots \\ f(z_1, t_G) \\ f(z_2, t_1) \\ \vdots \\ f(z_n, t_G) \end{pmatrix} \approx B\theta \]

with a \( n \times K = K_zK_t \) tensor product basis function matrix \( B \), where each row of \( B \) is the tensor product of the associated marginal basis functions \( B'_z|_{z=z_i} \otimes B'_t|_{t=t_G} \), with \( K_z \) marginal basis functions in \( z \) and \( K_t \) marginal basis functions in \( t \). For this representation, the constraint above can be written as a linear constraint on the associated spline coefficients \( \theta \):

\[ C\theta = 0 \]  

(i.e.,

\[ (1'_n \otimes I_G) = \begin{pmatrix} 1 & \cdots & 1 \\ \cdots & \ddots & \cdots \\ 1 & \cdots & 1 \end{pmatrix} \]

does a summation of basis function values for every observed timepoint). Constraints are absorbed into a modified design matrix by post-multiplying \( B \) with the Q-factor of the QR-decomposition of \( C^\top \) (Wood, 2006, ch. 1.8.1). Since we can write \( C \) as

\[ C = (B'_1(z), \ldots, B'_{K_z}(\bar{z}))) \otimes \begin{pmatrix} B'_1(t_1) & \cdots & B'_{K_t}(t_1) \\ \vdots & \ddots & \vdots \\ B'_1(t_G) & \cdots & B'_{K_t}(t_G) \end{pmatrix}, \]

with \( B'_i(\bar{z}) = \sum_{i=1}^{n} B'_i(z_i) \), the rank of \( C \) is \( K_t \). In practice it is sufficient to enforce the constraint for \( K_t \) of the timepoints spread across \( t_1, \ldots, t_G \) and rely on the smoothness of the function estimates to make sure that the constraint is fulfilled (approximately) in between. This allows us to avoid numerical issues that occur for the otherwise rank-deficient \( C \).

Our \texttt{pfr()} function overrides the inappropriate default constraint implemented in \texttt{mgcv}'s \texttt{gam()} function and instead imposes the constraints given above for terms that vary smoothly over the index of the response.
A.2. Data Example. The code for the following example is included in the accompanying R-script file `lfpr_supplement.R`.

We generate an artificial data set \( Y_i(t) = \beta_0(t) + f(z_i, t) + \beta x_i + \epsilon_{it} \) with \( n = 80 \) observations on \( G = 60 \) gridpoints, with a global functional intercept \( \beta_0(t) \), a smooth functional effect \( f(z, t) \) (see top row in Figure 1 for the shapes) and a constant linear effect \( \beta x_i \) with \( \beta = 3 \). Covariates \( x_i \) and \( z_i \) are standard uniform variables, \( \epsilon_{it} \) are i.i.d. Gaussian with negligible variance for almost noiseless data with signal-to-noise ratio 105. The code below fits an overspecified model \( E(Y_i(t)) = \beta_0(t) + f(z_i, t) + f(x_i, t) + \epsilon_{it} \), once with the standard sum-to-zero constraints \( \sum_{i,t} f(z_i, t) = 0 \) (fit with a deprecated version `pffrOld` of `pffr`), and once with the modified constraints \( \sum_{i,t} f(z_i, t) = 0 \forall t \), both with 10 marginal cubic P-spline basis functions and first order difference penalties.

```r
m.OldConstr <- pffrOld(Y ~ s(z) + s(x),
                      yind=t, data=d, method="REML",
                      bsy.default = list(bs="ps", m=c(2, 1), k=10))
m <- pffr(Y ~ s(z) + s(x),
          yind=t, data=d, method="REML",
          bs.int = list(bs="ps", m=c(2, 1), k=10),
          bs.yind= list(bs="ps", m=c(2, 1), k=10))
```

The fits for the two models are almost equivalent, the correlation between their fitted values is about 0.996. Figure 1, however, shows that the model without the appropriate centering constraints completely misses the true shape of the effects, that the functional intercept in the model with sum-to-zero-\( \forall -t \) constraints is directly interpretable as the mean trajectory in the data, and that estimation uncertainty is reduced by using the appropriate constraints (c.f. the widths of the confidence intervals around \( \hat{\beta}_0(t) \)).
**Fig 1:** True (top row) and estimated effects for the data described in section A.2. Middle row shows effects estimated with the default sum-to-zero constraints, while the bottom row shows effects estimated under the sum-to-zero-$\forall$-t constraints.
APPENDIX B: SIMULATION STUDY

B.1. Data generation. This subsection describes in detail the data generating process used for the simulation study in the article. We use

\[ \alpha(t) = t^2 - \sqrt{t} + \phi(t, 0.2, 0.1) + \phi(t, 0.6, 0.4) - f_B(t, 7, 4) + f_B(t, 9, 9), \]

where \( \phi(\cdot, \mu, \sigma) \) is a \( N(\mu, \sigma^2) \)-density and \( f_B(\cdot, a, b) \) is a Beta\((a, b)\)-density,

\[
\beta_1(s, t) = \pi^{0.12} \left( 1.2 \exp \left( - \frac{(s-0.2)^2}{0.09} - \frac{(t-0.3)^2}{0.16} \right) + 0.8 \exp \left( - \frac{(s-0.7)^2}{0.09} - \frac{(t-0.8)^2}{0.16} \right) \right),
\]

\[
\beta_2(s, t) = 2 + \sin(2\pi st) \log(1 + s + t) + 2st + \exp(s)t^2,
\]

\[
\gamma_1(z_1, t) = 5\phi \left( 2z_1 - \sin \left( \frac{\pi}{2t} \right) \right),
\]

\[
\delta_2(t) = \cos(\pi t) + t - 0.5.
\]

Functional random intercepts \( B_{i0}(t) \) are generated from a cubic B-spline basis with 5 basis functions whose coefficients are i.i.d. \( N(0, 1) \) and centered so that \( \sum_i n_i B_{i0}(t) = 0 \forall t \). Functional random slopes \( B_{i1}(t) \) are generated independently from \( B_{i0}(t) \) as a weighted sum of basis functions \((0.5, (t - 0.5), (t - 0.2)^2, (t - 0.8)^3)\) with i.i.d. \( N(0, 1) \) weights and centered as well. The functional covariates \( X_1(t) \) and \( X_2(t) \) are generated from a natural cubic spline basis with fifteen basis functions whose i.i.d. coefficients are drawn from \( U[-3, 3] \). For the scalar covariates, \( z_{1,ij} \sim i.i.d. U[-0.5, 0.5] \) and \( z_{2,ij} \sim i.i.d. N(0, 1) \). Residual \( \varepsilon_{ij}(t) \) are i.i.d. Gaussian, with variance \( \sigma^2_{\varepsilon} \) determined by the signal-to-noise ratio SNR\(c\) of the setting.

B.2. Relative integrated mean square errors. The following graphs display the relative integrated mean squared error rIMSE\((\hat{f}(t)) = \frac{\int (\hat{f}(t) - f(t))^2 dt}{\int f(t)^2 dt} \) for the various scenarios and settings. Tables give estimated coefficients for main effect models for \( \log_2(\text{rIMSE}) \) with the various settings under the different scenarios. Note that we are evaluating the estimation accuracy of the effects on the scale of the response, not on the scale of the coefficient function itself, e.g. for the effect of a functional covariate \( X(s) \) we consider the error \( \int_T \int_S (X(s)\hat{\beta}(s, t) - X(s)\beta(s, t))^2 dsdt \) (and not \( \int_T \int_S (\hat{\beta}(s, t) - \beta(s, t))^2 dsdt \)).
**Table 1**

<table>
<thead>
<tr>
<th>scenario</th>
<th>setting</th>
<th>high</th>
<th>low</th>
<th>estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>rIMSE(Y(t))</td>
<td>1 baseline</td>
<td>0.14</td>
<td>0.12</td>
<td>0.13</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>2 baseline</td>
<td>0.08</td>
<td>0.07</td>
<td>0.07</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>3 baseline</td>
<td>0.09</td>
<td>0.07</td>
<td>0.08</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>4 baseline</td>
<td>0.09</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>1 M: 10 → 100</td>
<td>0.97</td>
<td>0.86</td>
<td>0.91</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>2 M: 10 → 100</td>
<td>0.72</td>
<td>0.64</td>
<td>0.68</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>3 M: 10 → 100</td>
<td>0.74</td>
<td>0.66</td>
<td>0.70</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>4 M: 10 → 100</td>
<td>0.76</td>
<td>0.67</td>
<td>0.71</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>1 n: 3 → 20</td>
<td>0.28</td>
<td>0.25</td>
<td>0.26</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>2 n: 3 → 20</td>
<td>0.35</td>
<td>0.31</td>
<td>0.33</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>3 n: 3 → 20</td>
<td>0.32</td>
<td>0.28</td>
<td>0.30</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>4 n: 3 → 20</td>
<td>0.32</td>
<td>0.28</td>
<td>0.30</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>1 G: 30 → 100</td>
<td>0.43</td>
<td>0.38</td>
<td>0.40</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>2 G: 30 → 100</td>
<td>0.48</td>
<td>0.43</td>
<td>0.45</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>3 G: 30 → 100</td>
<td>0.50</td>
<td>0.44</td>
<td>0.47</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>4 G: 30 → 100</td>
<td>0.48</td>
<td>0.43</td>
<td>0.45</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>1 SNR_B: 0.1 → 1</td>
<td>0.49</td>
<td>0.42</td>
<td>0.45</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>2 SNR_B: 0.1 → 1</td>
<td>0.47</td>
<td>0.40</td>
<td>0.43</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>3 SNR_B: 0.1 → 1</td>
<td>0.93</td>
<td>0.80</td>
<td>0.86</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>4 SNR_B: 0.1 → 1</td>
<td>0.84</td>
<td>0.72</td>
<td>0.78</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>1 SNR_B: 0.1 → 10</td>
<td>0.12</td>
<td>0.11</td>
<td>0.11</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>2 SNR_B: 0.1 → 10</td>
<td>0.22</td>
<td>0.19</td>
<td>0.21</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>3 SNR_B: 0.1 → 10</td>
<td>0.48</td>
<td>0.42</td>
<td>0.45</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>4 SNR_B: 0.1 → 10</td>
<td>0.38</td>
<td>0.33</td>
<td>0.36</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>1 SNR_ε: 1 → 5</td>
<td>0.09</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>2 SNR_ε: 1 → 5</td>
<td>0.13</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>3 SNR_ε: 1 → 5</td>
<td>0.12</td>
<td>0.11</td>
<td>0.11</td>
</tr>
<tr>
<td>rIMSE(Y(t))</td>
<td>4 SNR_ε: 1 → 5</td>
<td>0.11</td>
<td>0.10</td>
<td>0.11</td>
</tr>
</tbody>
</table>

Exponentiated coefficient estimates (i.e., multiplication factors) for the log₂-linear model for rIMSE(Y(t)). “High” and “low” estimates are estimated coefficients ± 2 standard deviations, exponentiated.
Fig 2: rIMSE for \( \hat{Y}(t) \) for all combinations of the various settings.
<table>
<thead>
<tr>
<th>scenario</th>
<th>setting</th>
<th>high</th>
<th>low</th>
<th>estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>rIMSE(B₀(t))</td>
<td>baseline</td>
<td>0.19</td>
<td>0.15</td>
<td>0.17</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>baseline</td>
<td>0.07</td>
<td>0.05</td>
<td>0.06</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>baseline</td>
<td>0.07</td>
<td>0.06</td>
<td>0.07</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>baseline</td>
<td>0.08</td>
<td>0.06</td>
<td>0.07</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>baseline</td>
<td>1.36</td>
<td>1.14</td>
<td>1.25</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>M : 10 → 100</td>
<td>1.01</td>
<td>0.85</td>
<td>0.93</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>M : 10 → 100</td>
<td>1.10</td>
<td>0.92</td>
<td>1.01</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>M : 10 → 100</td>
<td>1.14</td>
<td>0.96</td>
<td>1.05</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>n₁ : 3 → 20</td>
<td>0.15</td>
<td>0.13</td>
<td>0.14</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>n₁ : 3 → 20</td>
<td>0.32</td>
<td>0.27</td>
<td>0.29</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>n₁ : 3 → 20</td>
<td>0.27</td>
<td>0.23</td>
<td>0.25</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>n₁ : 3 → 20</td>
<td>0.27</td>
<td>0.22</td>
<td>0.24</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>G : 30 → 100</td>
<td>0.57</td>
<td>0.48</td>
<td>0.52</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>G : 30 → 100</td>
<td>0.53</td>
<td>0.45</td>
<td>0.49</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>G : 30 → 100</td>
<td>0.52</td>
<td>0.44</td>
<td>0.48</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>G : 30 → 100</td>
<td>0.49</td>
<td>0.41</td>
<td>0.45</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>SNR_B : 0.1 → 1</td>
<td>1.74</td>
<td>1.40</td>
<td>1.56</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>SNR_B : 0.1 → 1</td>
<td>5.48</td>
<td>4.41</td>
<td>4.92</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>SNR_C : 0.1 → 1</td>
<td>1.83</td>
<td>1.47</td>
<td>1.64</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>SNR_C : 0.1 → 1</td>
<td>1.93</td>
<td>1.56</td>
<td>1.74</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>SNR_B : 0.1 → 10</td>
<td>19.09</td>
<td>15.36</td>
<td>17.13</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>SNR_B : 0.1 → 10</td>
<td>96.06</td>
<td>77.30</td>
<td>86.17</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>SNR_B : 0.1 → 10</td>
<td>37.60</td>
<td>30.26</td>
<td>33.73</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>SNR_B : 0.1 → 10</td>
<td>36.59</td>
<td>29.43</td>
<td>32.81</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>SNR_C : 1 → 5</td>
<td>0.16</td>
<td>0.13</td>
<td>0.15</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>SNR_C : 1 → 5</td>
<td>0.14</td>
<td>0.12</td>
<td>0.13</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>SNR_C : 1 → 5</td>
<td>0.12</td>
<td>0.10</td>
<td>0.11</td>
</tr>
<tr>
<td>rIMSE(B₀(t))</td>
<td>SNR_C : 1 → 5</td>
<td>0.11</td>
<td>0.10</td>
<td>0.10</td>
</tr>
</tbody>
</table>

Table 2
Exponentiated coefficient estimates (i.e., multiplication factors) for the log-linear model for rIMSE(B₀(t)). “High” and “low” estimates are estimated coefficients ± 2 standard deviations, exponentiated.

<table>
<thead>
<tr>
<th>scenario</th>
<th>setting</th>
<th>high</th>
<th>low</th>
<th>estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>rIMSE(B₁(t))</td>
<td>baseline</td>
<td>0.50</td>
<td>0.33</td>
<td>0.40</td>
</tr>
<tr>
<td>rIMSE(B₁(t))</td>
<td>M : 10 → 100</td>
<td>1.55</td>
<td>1.13</td>
<td>1.32</td>
</tr>
<tr>
<td>rIMSE(B₁(t))</td>
<td>G : 30 → 100</td>
<td>0.59</td>
<td>0.43</td>
<td>0.51</td>
</tr>
<tr>
<td>rIMSE(B₁(t))</td>
<td>SNR_B : 1 → 5</td>
<td>0.17</td>
<td>0.12</td>
<td>0.14</td>
</tr>
<tr>
<td>rIMSE(B₁(t))</td>
<td>SNR_B : 0.1 → 1</td>
<td>1.78</td>
<td>1.21</td>
<td>1.47</td>
</tr>
<tr>
<td>rIMSE(B₁(t))</td>
<td>SNR_B : 0.1 → 10</td>
<td>16.19</td>
<td>11.00</td>
<td>13.35</td>
</tr>
<tr>
<td>rIMSE(B₁(t))</td>
<td>n₁ : 3 → 20</td>
<td>0.10</td>
<td>0.07</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Table 3
Exponentiated coefficient estimates (i.e., multiplication factors) for the log₂-linear models for rIMSE(B₁(t)). “High” and “low” estimates are estimated coefficients ± 2 standard deviations, exponentiated.
Fig 3: rIMSE for $B_0(t)$ for all combinations of the various settings.
Fig 4: rIMSE for $\hat{B}_1(t)$ for all combinations of the various settings.
### Table 4

Exponentiated coefficient estimates (i.e., multiplication factors) for the log<sub>2</sub>-linear models for rIMSE(\( \int X_1(s)\beta_1(s,t) ds \)). “High” and “low” estimates are estimated coefficients ± 2 standard deviations, exponentiated.

<table>
<thead>
<tr>
<th>scenario</th>
<th>setting</th>
<th>high</th>
<th>low</th>
<th>estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>2</td>
<td>baseline</td>
<td>0.47</td>
<td>0.36</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>3</td>
<td>baseline</td>
<td>3.22</td>
<td>2.50</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>4</td>
<td>baseline</td>
<td>2.66</td>
<td>2.07</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>2</td>
<td>(M: 10 \rightarrow 100)</td>
<td>0.36</td>
<td>0.30</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>3</td>
<td>(M: 10 \rightarrow 100)</td>
<td>0.28</td>
<td>0.23</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>4</td>
<td>(M: 10 \rightarrow 100)</td>
<td>0.29</td>
<td>0.24</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>2</td>
<td>(n_t: 3 \rightarrow 20)</td>
<td>0.37</td>
<td>0.31</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>3</td>
<td>(n_t: 3 \rightarrow 20)</td>
<td>0.29</td>
<td>0.24</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>4</td>
<td>(n_t: 3 \rightarrow 20)</td>
<td>0.32</td>
<td>0.27</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>2</td>
<td>(G: 30 \rightarrow 100)</td>
<td>0.58</td>
<td>0.48</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>3</td>
<td>(G: 30 \rightarrow 100)</td>
<td>0.55</td>
<td>0.46</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>4</td>
<td>(G: 30 \rightarrow 100)</td>
<td>0.58</td>
<td>0.48</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>2</td>
<td>(\text{SNR}_B: 0.1 \rightarrow 1)</td>
<td>0.29</td>
<td>0.23</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>3</td>
<td>(\text{SNR}_B: 0.1 \rightarrow 1)</td>
<td>0.10</td>
<td>0.08</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>4</td>
<td>(\text{SNR}_B: 0.1 \rightarrow 1)</td>
<td>0.10</td>
<td>0.08</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>2</td>
<td>(\text{SNR}_B: 0.1 \rightarrow 10)</td>
<td>0.27</td>
<td>0.21</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>3</td>
<td>(\text{SNR}_B: 0.1 \rightarrow 10)</td>
<td>0.07</td>
<td>0.05</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>4</td>
<td>(\text{SNR}_B: 0.1 \rightarrow 10)</td>
<td>0.07</td>
<td>0.06</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>2</td>
<td>(\text{SNR}_\varepsilon: 1 \rightarrow 5)</td>
<td>0.24</td>
<td>0.20</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>3</td>
<td>(\text{SNR}_\varepsilon: 1 \rightarrow 5)</td>
<td>0.19</td>
<td>0.16</td>
</tr>
<tr>
<td>rIMSE((\beta_1(s,t)))</td>
<td>4</td>
<td>(\text{SNR}_\varepsilon: 1 \rightarrow 5)</td>
<td>0.19</td>
<td>0.16</td>
</tr>
</tbody>
</table>

### Table 5

Exponentiated coefficient estimates (i.e., multiplication factors) for the log<sub>2</sub>-linear models for rIMSE(\( \int X_2(s)\beta_2(s,t) ds \)). “High” and “low” estimates are estimated coefficients ± 2 standard deviations, exponentiated.

<table>
<thead>
<tr>
<th>scenario</th>
<th>setting</th>
<th>high</th>
<th>low</th>
<th>estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>rIMSE((\beta_2(s,t)))</td>
<td>3</td>
<td>baseline</td>
<td>1.64</td>
<td>1.30</td>
</tr>
<tr>
<td>rIMSE((\beta_2(s,t)))</td>
<td>3</td>
<td>(M: 10 \rightarrow 100)</td>
<td>0.26</td>
<td>0.21</td>
</tr>
<tr>
<td>rIMSE((\beta_2(s,t)))</td>
<td>3</td>
<td>(n_t: 3 \rightarrow 20)</td>
<td>0.26</td>
<td>0.22</td>
</tr>
<tr>
<td>rIMSE((\beta_2(s,t)))</td>
<td>3</td>
<td>(G: 30 \rightarrow 100)</td>
<td>0.53</td>
<td>0.44</td>
</tr>
<tr>
<td>rIMSE((\beta_2(s,t)))</td>
<td>3</td>
<td>(\text{SNR}_B: 0.1 \rightarrow 1)</td>
<td>0.09</td>
<td>0.07</td>
</tr>
<tr>
<td>rIMSE((\beta_2(s,t)))</td>
<td>3</td>
<td>(\text{SNR}_B: 0.1 \rightarrow 10)</td>
<td>0.06</td>
<td>0.05</td>
</tr>
<tr>
<td>rIMSE((\beta_2(s,t)))</td>
<td>3</td>
<td>(\text{SNR}_\varepsilon: 1 \rightarrow 5)</td>
<td>0.17</td>
<td>0.14</td>
</tr>
</tbody>
</table>
\[ \text{rIMSE}(\int X_1(s)\beta_1(s,t)ds) \]

**Fig 5:** rIMSE for \( \int X_1(s)\beta_1(s,t)ds \) for all combinations of the various settings.

<table>
<thead>
<tr>
<th>scenario</th>
<th>setting</th>
<th>high</th>
<th>low</th>
<th>estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>rIMSE(\gamma_1(z_1,t))</td>
<td>4</td>
<td>baseline</td>
<td>3.62</td>
<td>2.77</td>
</tr>
<tr>
<td>rIMSE(\gamma_1(z_1,t))</td>
<td>4</td>
<td>M: 10 \rightarrow 100</td>
<td>0.21</td>
<td>0.17</td>
</tr>
<tr>
<td>rIMSE(\gamma_1(z_1,t))</td>
<td>4</td>
<td>n_i: 3 \rightarrow 20</td>
<td>0.21</td>
<td>0.17</td>
</tr>
<tr>
<td>rIMSE(\gamma_1(z_1,t))</td>
<td>4</td>
<td>G: 30 \rightarrow 100</td>
<td>0.45</td>
<td>0.37</td>
</tr>
<tr>
<td>rIMSE(\gamma_1(z_1,t))</td>
<td>4</td>
<td>SNR_B: 0.1 \rightarrow 1</td>
<td>0.07</td>
<td>0.06</td>
</tr>
<tr>
<td>rIMSE(\gamma_1(z_1,t))</td>
<td>4</td>
<td>SNR_B: 0.1 \rightarrow 10</td>
<td>0.05</td>
<td>0.04</td>
</tr>
<tr>
<td>rIMSE(\gamma_1(z_1,t))</td>
<td>4</td>
<td>SNR_e: 1 \rightarrow 5</td>
<td>0.13</td>
<td>0.10</td>
</tr>
</tbody>
</table>

**Table 6**

Exponentiated coefficient estimates (i.e., multiplication factors) for the log_2-linear models for rIMSE(\gamma_1(z_1,t)). “High” and “low” estimates are estimated coefficients ± 2 standard deviations, exponentiated.
Fig 6: rIMSE for \( \int X_2(s) \beta_2(s, t) \) for all combinations of the various settings.

Table 7

<table>
<thead>
<tr>
<th>scenario</th>
<th>setting</th>
<th>high</th>
<th>low</th>
<th>estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>rIMSE(( \delta(t)z_2 ))</td>
<td>4</td>
<td>baseline</td>
<td>6.39</td>
<td>4.41</td>
</tr>
<tr>
<td>rIMSE(( \delta(t)z_2 ))</td>
<td>4</td>
<td>( M : 10 \rightarrow 100 )</td>
<td>0.16</td>
<td>0.12</td>
</tr>
<tr>
<td>rIMSE(( \delta(t)z_2 ))</td>
<td>4</td>
<td>( n_i : 3 \rightarrow 20 )</td>
<td>0.15</td>
<td>0.11</td>
</tr>
<tr>
<td>rIMSE(( \delta(t)z_2 ))</td>
<td>4</td>
<td>( G : 30 \rightarrow 100 )</td>
<td>0.43</td>
<td>0.33</td>
</tr>
<tr>
<td>rIMSE(( \delta(t)z_2 ))</td>
<td>4</td>
<td>( SNR_B : 0.1 \rightarrow 1 )</td>
<td>0.04</td>
<td>0.03</td>
</tr>
<tr>
<td>rIMSE(( \delta(t)z_2 ))</td>
<td>4</td>
<td>( SNR_B : 0.1 \rightarrow 10 )</td>
<td>0.02</td>
<td>0.01</td>
</tr>
<tr>
<td>rIMSE(( \delta(t)z_2 ))</td>
<td>4</td>
<td>( SNR_e : 1 \rightarrow 5 )</td>
<td>0.08</td>
<td>0.06</td>
</tr>
</tbody>
</table>

Exponentiated coefficient estimates (i.e., multiplication factors) for the log_2-linear models for rIMSE(\( \delta(t)z_2 \)). “High” and “low” estimates are estimated coefficients ± 2 standard deviations, exponentiated.
Fig 7: rMSE for $\hat{\gamma}_1(z_1, t)$ for all combinations of the various settings.

Fig 8: rMSE for $\hat{\delta}_2(t)_{z_2}$ for all combinations of the various settings.
B.3. Exemplary data sets and fits. For each scenario, we present the setting and replication with rIMSE values closest to the median rIMSE values, followed by those with minimal and maximal rIMSE values across the various combinations of $n_i$, $M$, $G$, $\text{SNR}_B$ and $\text{SNR}_e$. For all plots, the left column shows the observed or true quantities, while the right column shows their estimates. The bottom row displays the observed functional responses on the left and the estimated residual curves on the right, note that they are on different vertical scales. Trajectories are colour-coded for subject. For larger data sets, only a sample of at most 300 observations is plotted.

B.3.1. Scenario 1.
Scenario 1 (median error): $M = 10$, $n_{ij} = 3$, $G = 30$, $\text{SNR}_e = 5$, $\text{SNR}_B = 1$

**Fig 9:** Example of data and fit for scenario 1 with median error.
Scenario 1 (minimum error) : $M = 10$, $n_i = 20$, $G = 100$, $SNR_{\epsilon} = 5$, $SNR_B = 1$

Fig 10: Example of data and fit for scenario 1 with minimum error.
Scenario 1 (maximum error): $M = 10$, $n_i = 3$, $G = 30$, $\text{SNR}_c = 1$, $\text{SNR}_B = 1$

**Fig 11:** Example of data and fit for scenario 1 with maximum error.
B.3.2. Scenario 2.

Fig 12: Example of data and fit for scenario 2 with median error.
Scenario 2 (minimum error): $M = 100$, $n_i = 20$, $G = 100$, $\text{SNR}_e = 5$, $\text{SNR}_B = 1$

Fig 13: Example of data and fit for scenario 2 with minimum error.
Scenario 2 (maximum error): $M = 10, n_i = 3, G = 30, \text{SNR}_e = 1, \text{SNR}_B = 10$

Fig 14: Example of data and fit for scenario 2 with maximum error.
Scenario 3 (median error): $M = 10$, $n_i = 20$, $G = 100$, SNR$_{\varepsilon} = 1$, SNR$_{B} = 1$

Fig 15: Example of data and fit for scenario 3 with median error.

B.3.3. Scenario 3.
Scenario 3 (minimum error) : $M = 100$, $n_i = 20$, $G = 100$, $\text{SNR}_\varepsilon = 5$, $\text{SNR}_B = 1$

Fig 16: Example of data and fit for scenario 3 with minimum error.
Scenario 3 (maximum error) : \( M = 10, n_i = 3, G = 30, \text{SNR}_e = 1, \text{SNR}_B = 10 \)

**Fig 17:** Example of data and fit for scenario 3 with maximum error.
B.3.4. Scenario 4.

Fig 18: Example of data and fit for scenario 4 with median error.
Scenario 4 (minimum error) : \( M = 100, \ n_i = 20, \ G = 100, \ \text{SNR}_e = 5, \ \text{SNR}_B = 1 \)

\[ E(Y_{ij}(t)) - \hat{Y}_{ij}(t) \]

\[ \delta_2(t) \text{ and } \hat{\delta}_2(t) \text{ for } z_2: \text{rMSE}=1e-04 \]

\[ \gamma_1(z_{ij}, t) \]

\[ \hat{\gamma}_1(z_{ij}, t): \text{rMSE}\approx4e-04 \]

\[ \gamma_1(z_{ij}, t) \]

\[ \hat{\gamma}_1(z_{ij}, t): \text{rMSE}=4e-04 \]

\[ \beta_1(s, t) \text{ and } \hat{\beta}_1(s, t) \]

\[ \beta_1(s, t) \text{ and } \hat{\beta}_1(s, t) \text{ for } z_1: \text{rMSE}=6e-04 \]

\[ \int X_{1,ij}(s)\beta_1(s, t)ds \]

\[ \int X_{1,ij}(s)\hat{\beta}_1(s, t)ds: \text{rMSE}\approx0.0043 \]

\[ E(Y_{ij}(t)) \]

\[ \hat{Y}_{ij}(t) \]

\[ E(Y_i(t)) \]

\[ \hat{Y}_i(t) \]

\[ E(Y_i(t)) - \hat{Y}_i(t) \]

\[ \text{Fig 19: Example of data and fit for scenario 4 with minimum error.} \]
Scenario 4 (maximum error) : $M = 10$, $n_i = 3$, $G = 30$, SNR$_\varepsilon = 1$, SNR$_B = 10$

Fig 20: Example of data and fit for scenario 4 with maximum error.
APPENDIX C: PREDICTING PRECIPITATION PROFILES FROM TEMPERATURE CURVES FOR THE CANADIAN WEATHER DATA

This section is primarily intended to show the flexibility and performance of pffr on this small toy data set with some example code and graphical summaries, and not to attempt a stringent model criticism or model comparisons. R-Code used to perform the analysis is set in typewriter font and put in light-grey boxes, output returned by the R-console is indicated by ##. Comments in the code are indicated by #'.

```
#'load data:
data(CanadianWeather)
dataM <- with(CanadianWeather, 
  list(
    temp = t(monthlyTemp),
    l10precip = t(log10(monthlyPrecip)),
    lat = coordinates[,"N.latitude"],
    lon = coordinates[,"W.longitude"],
    region = factor(region),
    place = factor(place)
  )
)
#' correct Prince George location
#' (wrong at least until fda_2.2.7):
dataM$lon["Pr. George"] <- 122.75
dataM$lat["Pr. George"] <- 53.9
#' center temperature curves:
dataM$tempRaw <- dataM$temp
dataM$temp <- sweep(dataM$temp, 2, colMeans(dataM$temp))
#' define function indices
month.t <- 1:12
month.s <- 1:12
```

The Canadian weather data consists of temperature and precipitation curves, measured as the monthly average over several years at 35 Canadian weather stations (see Figure 21, top). The data has been used extensively in the functional data analysis literature. As it is available as part of the R-package fda (Ramsay et al., 2011), we can make our analysis fully reproducible, the full source code for this section is in the CanadianWeather_Long.R file included in this supplement.

We will here focus on both the functional relationship between temperature and precipitation profiles as well as on the spatial nature of the data, clearly visible from the locations of the weather stations depicted in Fig-
ure 21 (middle left). Ramsay and Silverman (2005) propose a concurrent model to predict precipitation profiles from temperature curves, where temperature is allowed to influence log-precipitation linearly at the same time point \( t \). Within the framework of our model, we can investigate more flexible functional regression models that allow for lagged temperature effects. Additionally, we can take into account the spatial correlation structure between weather stations.

C.1. Model 1: Time-varying smooth effect for smooth spatially correlated residual curves. We consider the model

\[
Y_i(t) = g_0(t) + \gamma(c_i, d_i, t) + b_0i + \int X_i(s) \beta(s, t) ds + \varepsilon_{it}, \quad \varepsilon_{it} \sim N(0, \sigma^2_\varepsilon),
\]

where \( Y_i(t) \) and \( X_i(t) \) denote the log-precipitation and the (centered) temperature at location \( i \) and time \( t \), and \( c_i \) and \( d_i \) denote longitude and latitude of location \( i \). As the temperature curves were centered pointwise across stations and \( \gamma(c_i, d_i, t) \) is constrained to sum to zero for each \( t \), \( g_0(t) \) indicates the mean log-precipitation curve for a station with average temperature profile. The spatio-temporal term \( \gamma(c_i, d_i, t) \) yields a smooth cyclic residual curve for each station, with a spatial covariance structure induced by the bivariate spline basis for longitude and latitude and its smoothness penalty. Alternatively, it can be viewed as a smooth spatial effect that varies cyclically throughout the year. Small scale local differences in the levels of precipitation not captured by these spatially correlated residuals are modelled with a scalar random intercept \( b_0i \overset{i.i.d.}{\sim} N(0, \sigma^2_b) \). It is important to note that the temperature and precipitation profiles considered here are averaged over several years. Our model thus does not have the problem of ‘the future influencing the past’, but relates general weather patterns to one another.

To avoid identifiability issues, we have to use a fairly small number of basis functions for the temperature effect in this case (c.f. Scheipl and Greven, 2012):

```r
# check effective rank of covariance of temperature deviations
# to avoid identifiability issues:
cov.temp <- crossprod(dataM$temp)
ev.cov.temp <- eigen(cov.temp)$values
cumsum(ev.cov.temp)/sum(ev.cov.temp)
## [1] 0.89238 0.97506 0.99335 0.99800 0.99892 0.99949 0.99976
## [8] 0.99984 0.99991 0.99996 0.99999 1.00000
# first 4 eigenfunctions represent >.995 of total variability
# --> use only 4 basis functions to avoid identifiability issues
```
To take into account the cyclic nature of both the response and the predictor curves, we use cyclic basis functions in both $s$ and $t$ direction. The model can then be fit using the `pfr()` function in the `refund` package (Crainiceanu et al., 2011) as

```r
mM <- pfr(l10precip ~ s(lat,lon) + c(s(place, bs="re"))) +
```
ff(temp, yind=month.t, xind=month.s,  
splinepars=list(bs=c("cc", "cc"), k=c(4, 4)),  
check.ident=FALSE),  
bs.int = list(bs = "cc", k=10), bs.yindex = list(bs="cc"),  
data=dataM, yind = month.t,  
knots=list(month.t.vec=c(0.5,12.5), temp.tmat=c(0.5,12.5),  
temp.smat=c(0.5,12.5)))

## imposing constraints..
summary(mM)
##
## Family: gaussian
## Link function: identity
##
## Formula:
## l10precip ~ s(lat, lon) + c(s(place, bs = "re")) + ff(temp, yind = month.t,  
## xind = month.s, splinepars = list(bs = c("cc", "cc"), k = c(4,  
##)) <environment: 0x0cccc524>  
##
## Constant coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 0.20852 0.00228 91.3 <2e-16 ***  
## ---  
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##
## Smooth terms & functional coefficients:  
## edf Ref.df F p-value  
## Intercept(month.t) 6.87 7.62 206.8 <2e-16 ***  
## s(lat,lon) 113.84 131.76 26.9 <2e-16 ***  
## c(s(place)) 22.70 22.95 109.7 <2e-16 ***  
## ff(temp,month.t,month.s) 5.66 6.33 1.7 0.12  
## ---  
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##
## R-sq.(adj) = 0.981 Deviance explained = 98.8%  
## REML score = -416.02 Scale est. = 0.0023158 n = 420(35 x 12)

' get estimated coefficients, predictor components, and responses:
coefs <- coef(mM, n1=100, n2=80)
terms <- predict(mM, type="terms")
fit <- fitted(mM)

Here, l10precip and temp are 35×12 matrices containing the log-precipitation and centered temperature profiles, respectively; lat and lon are vectors
containing the latitude and longitude of each location, respectively, and
\texttt{month.s} and \texttt{month.t} are vectors containing the months 1 to 12. A smooth
overall mean function \( \alpha(t) \) (i.e, \texttt{Intercept(month.t)}) is added by default.
\( s(\text{lat,lon}) \) yields a smooth spatio-temporal surface \( \gamma(c_i,d_i,t) \). \texttt{c(s(place, bs="re"))} yields a scalar random intercept \( b_{0i} \) for the stations. \texttt{ff(temp, ..., splinepars=list(bs=c("cc", "cc"), k=c(4, 4)))} fits the linear func-
tional regression term \( \int X_i(s) \beta(s,t) \, ds \), with \( bs=c("cc", "cc") \) specifying a
cyclic cubic regression spline basis with wrapped around penalty in both \( s \)
and \( t \) direction and 4 marginal basis functions each \( (k=c(4,4)) \). We specify
\texttt{check.ident=FALSE} in this case to switch off the identifiability check included
in \texttt{ff()} that would otherwise try to reduce the number of basis functions for
this low-rank predictor to 3. \texttt{bs.int = list(bs = "cc", k=10) and bs.yindex
= list(bs="cc")} specify a cyclic spline basis for the intercept curve and all
other terms that are functions in \( t \). The \texttt{knots}-statement specifies the time-
points at which cyclic basis functions are “wrapped around”, i.e. timepoint
0.5 is equivalent to timepoint 12.5 in this case.

Most of the variation in the data is explained by the effect of temperature,
followed by the spatially varying functional random intercept \( \gamma(c_i,d_i,t) \) and
the scalar random intercept. The estimated overall mean function (Figure
21, bottom left) shows a seasonal pattern, with slightly lower precipitation
in the spring and higher precipitation in the fall, above and beyond what is
explained by the effect of temperature.

The effect of temperature on log-precipitation is shown in Figure 21 (bot-
tom right). In \( s \) direction, a clear seasonal pattern is visible, with higher
temperatures in winter and especially autumn associated with an increase
of precipitation, and higher temperatures in the spring and summer
associated with a decrease of precipitation. In \( t \) direction, effects are much
stronger for the winter months, and much weaker for the summer months,
especially the effect of winter temperatures on precipitation in the middle
of the year; a plausible result.

The spatially correlated smooth residual plus the scalar random inter-
cept for each weather station is depicted in Figure 21 (middle right – see
Figure 22 for a version without overlapping). It shows some interesting local
features, which clearly illustrate that regional effects cannot capture the spa-
tially varying structure of precipitation curves adequately. For example, the
Arctic station Inuvik in the very north-west shows a similar error pattern to
Continental stations in the north-west (precipitation higher in winter and
lower in the summer and especially in spring). These northern Continental
stations, in turn, exhibit a pattern which is completely different than that of the more southern Continental stations (higher in the summer and lower in the winter). Another perspective on $\gamma(c_i, d_i, t)$ is to view it as a smoothly time-varying surface estimate for a spatial effect. Figure 23 displays the temporal evolution of $\gamma(c_i, d_i, t)$ over the year. We can distinguish essentially two phases, an autumn/winter phase (top row, left three panels of bottom row) with higher precipitation in the coastal regions and lower precipitation in the interior than what can be explained by the mean temperature deviations, followed by a short transition phase and then a late spring/summer phase with increased precipitation in the interior, especially in the south and relative to the Atlantic Coast region. The right panel in the bottom row shows BLUPS for the estimated uncorrelated scalar random intercepts $b_{i0}$ on the same vertical scale as the remainder of the panels. Note the fairly high small-scale variability in mean precipitation levels that has about the same magnitude as the smooth time-varying spatial effect.
Fig 21: Log-precipitation (top left) and temperature (top right) at 35 Canadian weather stations. Middle left: Stations’ locations. Middle right: Estimated spatially correlated smooth residual for each weather station. Bottom left: Estimated overall mean effect. Bottom right: Estimated functional effect $\hat{\beta}(s, t)$ of temperature in month $s$ on log-precipitation in month $t$, color-coded for sign and pointwise significance (95%): blue if significantly $< 0$, lightblue if $< 0$, lightred if $> 0$, red if sig. $> 0$. 
Fig 22: Solid lines: Spatially correlated smooth residual curves plus scalar random intercept for each weather station. Points: Observed errors $Y_i(t) - \hat{Y}_i(t)$. Stations are roughly ordered from north-west to south-east within regions. Color coding is as in Figure 21.
\[ \hat{\gamma}(c,d,t), \ t = 1 \]
\[ \hat{\gamma}(c,d,t), \ t = 1.58 \]
\[ \hat{\gamma}(c,d,t), \ t = 2.16 \]
\[ \hat{\gamma}(c,d,t), \ t = 3.32 \]
\[ \hat{\gamma}(c,d,t), \ t = 3.89 \]
\[ \hat{\gamma}(c,d,t), \ t = 4.47 \]
\[ \hat{\gamma}(c,d,t), \ t = 5.63 \]
\[ \hat{\gamma}(c,d,t), \ t = 6.21 \]
\[ \hat{\gamma}(c,d,t), \ t = 6.79 \]
\[ \hat{\gamma}(c,d,t), \ t = 7.95 \]
\[ \hat{\gamma}(c,d,t), \ t = 8.53 \]
\[ \hat{\gamma}(c,d,t), \ t = 9.11 \]
\[ \hat{\gamma}(c,d,t), \ t = 10.26 \]
\[ \hat{\gamma}(c,d,t), \ t = 10.84 \]
\[ \hat{\gamma}(c,d,t), \ t = 12 \]

Fig 23: Time-varying spatial effect \( \hat{\gamma}(c_i, d_i, t) \) over the course of the year and scalar random intercepts \( \hat{b}_{0i} \). Station locations given by the dots at the top of the plots.
C.2. Model 2: Time-varying regional effects and spatially uncorrelated residual curves. For comparison, a simpler model ignoring the spatial correlation

\[ Y_i(t) = \alpha_{g_i}(t) + \int X_i(s)\beta(s,t)ds + E_i(t) + \varepsilon_{it}, \quad \varepsilon_{it} \sim N(0, \sigma^2) \]

could be fit. Here, \( g_i \) indicates which of the four climate regions (Atlantic, Continental, Pacific and Arctic) \( i \) belongs to and \( \alpha_{g_i}(t) \) thus denotes a region-specific intercept curve. \( E_i(t) \) is a location-specific smooth residual centered at zero for each \( t \). This model can be fit via

```r
mR <- pffr(l10precip ~ 0 + c(0) + region + c(region) +
            s(place, bs="re") +
            ff(temp, yind=month.t, xind=month.s,
               splinepars=list(bs=c("cc", "cc"), k=c(4, 4)),
               check.ident=FALSE),
            bs.int = list(bs = "cc", k=10), bs.yindex = list(bs="cc"),
            data=dataM, yind = month.t,
            knots=list(month.t.vec=c(0.5,12.5), temp.tmat=c(0.5,12.5),
                       temp.smat=c(0.5,12.5)))
```

Specifically, \( \text{region} + \text{c(region)} \) yields time-varying region effects not centered at zero, made all estimable by dropping the constant and time-varying intercepts via \( \text{0 + c(0)} \), and \( \text{s(place, bs = "re")} \) is used to obtain location-specific smooth residuals \( E_i(t) \). In the notation of section 2.4 in the main article, this corresponds to estimating a functional random intercept for each observation \( i \), with no inter-subject correlation (\( P^0 = I_{35} \)).

Estimated regional effects and the coefficient surface for temperature deviations are displayed in Figure 24. Estimated temperature effects are very similar for the two models, indicating that either model formulation captures the spatial component of the responses’ variability well enough to allow for a reliable estimate of the temperature effect. The spatially uncorrelated smooth residuals are shown in Figure 25. Note that they are fairly similar for closely neighboring stations despite the fact that no spatial correlation structure was modeled.
Fig 24: Estimated effects for a model with region-specific intercepts and spatially uncorrelated smooth residuals. Left: Estimated region effects and approximate pointwise 95% confidence intervals (Arctic, Atlantic, Continental, Pacific). Right: Estimated functional effect $\hat{\beta}(s,t)$ of temperature in month $s$ on log-precipitation in month $t$, color-coded for sign and pointwise significance (95%): blue if significantly $< 0$, light blue if $< 0$, light red if $> 0$, red if sig. $> 0$. 
Fig 25: Solid lines: Spatially uncorrelated smooth residual curves for each weather station for the model with regional effects. Points: Observed errors \( Y_i(t) - \hat{Y}_i(t) \). Stations are roughly ordered from north-west to south-east within regions. Color coding as in previous figures.
C.3. Models 3-5: Time-varying regional effects and spatially correlated residual curves with fixed correlation structures. We can modify model

\[ Y_i(t) = \alpha_{g_i(t)} + \int X_i(s) \beta(s, t) ds + E_i(t) + \varepsilon_{it}, \quad \varepsilon_{it} \sim N(0, \sigma^2_{\varepsilon}) \]

from the previous subsection by specifying a marginal spatial correlation structure for the smooth residual curves \( E_i(t) \) to encourage similarity of residual curves for stations that are in close proximity. In pffr, we can achieve this by using the inverse of a given correlation matrix as the marginal precision for a Markov random field (more precisely: a Gaussian random field) across the different stations. In the notation of section 2.4 in the main article, we now use the inverse of the correlation matrix of the stations as \( P^q \) instead of \( P^q = I_{35} \) as in the previous subsection.

Specifically, we use Matèrn correlation functions with smoothness parameters \( \nu = .5 \) (i.e., the exponential correlation function), \( \nu = 1 \) and \( \nu = 10 \) and range parameters chosen so that the correlation drops to about 0.2 for great circle distances of 500 km, 1500 km and 3000 km, respectively.

```r
# get great circle distances:
locations <- cbind(dataM$lon, dataM$lat)
# fix location names s.t. they correspond to levels in places
rownames(locations) <- as.character(dataM$place)
dist <- rdist.earth(locations, miles=FALSE, R=6371)

# construct Matern correlation matrices as
# marginal penalty for a GRF over the locations:

# find ranges for nu = .5, 1 and 10
# where the correlation drops to .2 at a distance of 500/1500/3000 km
# (about the 10%/40%/70% quantiles of distances here)
r.5 <- Matern.cor.to.range(500, nu=0.5, cor.target=.2)
r1 <- Matern.cor.to.range(1500, nu=1.0, cor.target=.2)
r10 <- Matern.cor.to.range(3000, nu=10.0, cor.target=.2)
# compute correlation matrices
corr_nu.5 <- apply(dist, 1, Matern, nu=.5, range=r.5)
corr_nu1 <- apply(dist, 1, Matern, nu=1, range=r1)
corr_nu10 <- apply(dist, 1, Matern, nu=10, range=r10)

# invert to get precisions
P_nu.5 <- solve(corr_nu.5)
P_nu1 <- solve(corr_nu1)
P_nu10 <- solve(corr_nu10)
```
Figure 26 shows the shapes of the three correlation functions – we fit one model with a very quickly decreasing correlation structure ($\nu = .5$), one intermediate correlation structure ($\nu = 1$), and one model with a very persistent spatial correlation structure ($\nu = 10$).

![Figure 26: The three different Matérn correlation functions used. Range parameters are 311, 624 and 377, respectively, for $\nu = .5$, $\nu = 1$ and $\nu = 10$](image)

To fit models where entry $(i, j)$ in $(P^q)^{-1}$ is $\rho(d((c_i, d_i), (c_j, d_j)); \nu, \text{range})$, with latitudes $c_i$ and longitudes $d_i$, great circle distance function $d()$, and Matérn correlation function $\rho(d; \nu, \text{range})$ we specify an mrf-term (“Markov” random field, actually a conventional Gaussian random field in this case) for the different stations, and supply the inverse of the Matérn correlation matrix as the precision of the Markov random field, i.e. we specify $s(\text{place, bs="mrf"}, \ k=35, \ xt=list(list(penalty=P.nu1)))$. Specifying the $k$-argument to equal the number of locations avoids the low-rank approximation of the MRF used by default in mgcv, because this approximation only worked for positive semi-definite precision matrices at the time of writing.
mR_nu.5 <- pffr(l10precip ~ 0 + c(0) + region + c(region) + 
  s(place, bs="mrf", k=35, xt=list(list(penalty=P_nu.5))) + 
  ff(temp, yind=month.t, xind=month.s, 
  splinepars=list(bs="cc", "cc"), k=c(4, 4)), 
  bs.int = list(bs = "cc", k=10), bs.yindex = list(bs="cc"), 
  data=dataM, yind = month.t, 
  knots=list(month.t.vec=c(0.5,12.5), temp.tmat=c(0.5,12.5), 
  temp.smat=c(0.5,12.5)))
## imposing constraints..
summary(mR_nu.5)
##
## Family: gaussian
## Link function: identity
##
## Formula: 
## l10precip ~ 0 + c(0) + region + c(region) + s(place, bs = "mrf", 
## k = 35, xt = list(list(penalty = P_nu.5))) + ff(temp, yind = month.t, 
## xind = month.s, splinepars = list(bs = c("cc", "cc"), k = c(4, 
## 4)), check.ident = FALSE)
## <environment: 0x199f0888>
##
## Constant coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## regionArctic                   -0.06774  0.01134  -5.97  8.8e-09 ***
## regionAtlantic                 0.37988  0.00400   94.90  < 2e-16 ***
## regionContinental              0.09383  0.00444   21.11  < 2e-16 ***
## regionPacific                  0.17714  0.00945   18.75  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Smooth terms & functional coefficients:
##                                    edf Ref.df  F    p-value
## regionArctic(month.t)             2.67  3.33 1.88   0.13
## regionAtlantic(month.t)           5.84  6.70 17.13 < 2e-16 ***
## regionContinental(month.t)        7.17  7.68 64.49 < 2e-16 ***
## regionPacific(month.t)            5.18  6.03 5.48  2.4e-05 ***
## s(place)                          160.68 188.22 32.93 < 2e-16 ***
## ff(temp,month.t,month.s)          5.45  5.48 71.59 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.986  Deviance explained = 99.3%
## REML score =  427.56  Scale est. =  0.001696  n = 420(35 x 12)
mR_nu1 <- pffr(l10precip ~ 0 + c(0) + region + c(region) +
  s(place, bs="mrf", k=35, xt=list(penalty=P_nu1))) +
  ff(temp, yind=month.t, xind=month.s,
   splinepars=list(bs=c("cc", "cc"), k=c(4, 4)),
   check.ident=FALSE),
  bs.int = list(bs = "cc", k=10), bs.yindex = list(bs="cc"),
  data=dataM, yind = month.t,
  knots=list(month.t.vec=c(0.5,12.5), temp.tmat=c(0.5,12.5),
  temp.smat=c(0.5,12.5)))
## imposing constraints..
summary(mR_nu1)
##
## Family: gaussian
## Link function: identity
##
## Formula:
## l10precip ~ 0 + c(0) + region + c(region) + s(place, bs = "mrf",
## k = 35, xt = list(penalty = P_nu1))) + ff(temp, yind = month.t,
## xind = month.s, splinepars = list(bs = c("cc", "cc"), k = c(4,
## 4)), check.ident = FALSE)
## <environment: 0x199f0888>
##
## Constant coefficients:
##
##             Estimate  Std. Error t value Pr(>|t|)
## regionArctic  -0.06182   0.01158  -5.34  2.2e-07 ***
## regionAtlantic  0.37963   0.00397  95.56 < 2e-16 ***
## regionContinental  0.09528   0.00446 21.35 < 2e-16 ***
## regionPacific  0.17084   0.00981  17.41 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Smooth terms & functional coefficients:
##
##            edf Ref.df    F    p-value
## regionArctic(month.t)  2.96   3.67  2.86   0.029 *
## regionAtlantic(month.t)  5.82   6.67 16.74 < 2e-16 ***
## regionContinental(month.t)  7.16   7.66 65.87 < 2e-16 ***
## regionPacific(month.t)  5.19   6.03   5.54  2.1e-05 ***
## s(place)   160.85 189.29 33.37  < 2e-16 ***
## ff(temp,month.t,month.s)  5.24   5.28 101.09 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.987  Deviance explained = 99.3%
## REML score = -432.08  Scale est. = 0.0016655  n = 420(35 x 12)
mR_nu10 <- pffr(l10precip ~ 0 + c(0) + region + c(region) +
  s(place, bs="mrf", k=35, xt=list(penalty=P_nu10))) +
  ff(temp, yind=month.t, xind=month.s,
  splinepars=list(bs=c("cc", "cc"), k=c(4, 4)),
  check.ident=FALSE),
  bs.int = list(bs = "cc", k=10), bs.yindex = list(bs="cc"),
  data=dataM, yind = month.t,
  knots=list(month.t.vec=c(0.5,12.5), temp.tmat=c(0.5,12.5),
  temp.smat=c(0.5,12.5)))
# imposing constraints..
summary(mR_nu10)
#
## Family: gaussian
## Link function: identity
##
## Formula:
## l10precip ~ 0 + c(0) + region + c(region) + s(place, bs = "mrf",
##   k = 35, xt = list(penalty = P_nu10))) + ff(temp, yind = month.t,
##   xind = month.s, splinepars = list(bs = c("cc", "cc"), k = c(4,
##   4)), check.ident = FALSE)
## <environment: 0x199f0888>
##
## Constant coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## regionArctic           -0.06118  0.01187  -5.15  5.2e-07 ***
## regionAtlantic         0.37999  0.00407  93.32 < 2e-16 ***
## regionContinental      0.09536  0.00458  20.82 < 2e-16 ***
## regionPacific          0.16917  0.01009  16.77 < 2e-16 ***
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Smooth terms & functional coefficients:
##                      edf Ref.df   F  p-value
## regionArctic(month.t) 2.75  3.47  3.16 0.02 *
## regionAtlantic(month.t) 5.74  6.60 17.05 < 2e-16 ***
## regionContinental(month.t) 7.06  7.61 63.95 < 2e-16 ***
## regionPacific(month.t)  5.52  6.39  6.09 3.6e-06 ***
## s(place)             140.10 168.11 35.53 < 2e-16 ***
## ff(temp,month.t,month.s)  6.06  6.36 74.67 < 2e-16 ***
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.986  Deviance explained =  99.2%
## REML score = -414.42  Scale est. =  0.0017526  n = 420(35 x 12)
As the summaries show, the fits are very similar and robust against different specifications of the spatial correlation structure.

Fig 27: Estimated region effects for the model with independent smooth residuals and the models with spatially correlated smooth residuals.

Figures 27 to 29 compare the estimated effects for the model with region effects and independent smooth residuals to those for the models with correlated smooth residuals. As seen in figure 27, region effects are very robust against the different specifications for the smooth residual terms. Figure 28 shows that, as the spatial correlation of the smooth residuals increases (and thus the flexibility of the residual terms decreases), the temperature effect becomes somewhat larger, while retaining its shape fairly exactly. Our interpretation of this phenomenon is that, since temperature curves of course correlate strongly with the spatial locations, including more flexible spatial effects tends to attenuate the effect of temperature. The estimated spatial
Fig 28: Estimated functional effect $\hat{\beta}(s,t)$ of temperature in month $s$ on log-precipitation in month $t$, color-coded for sign and pointwise significance (95%): blue if sig. $< 0$, lightblue if $< 0$, lightred if $> 0$, red if sig. $> 0$. From top left to bottom right for the model with independent smooth residuals and the models with spatially correlated smooth residuals.
Fig 29: Spatially uncorrelated and correlated smooth residual curves for each weather station for the model with regional effects. Stations are roughly ordered from north-west to south-east within regions. Color coding for regions as in previous figures, line types code for the 4 different models.
Fig 30: Observed residuals $Y_i(t) - \hat{Y}_i(t)$ for each weather station for the model with regional effects. Stations are roughly ordered from north-west to south-east within regions. Color coding for regions as in previous figures, line types code for the 4 different models.
effects (see Figure 29) are very similar for the four different model specifi-
cations because residual curves of neighboring locations are very similar even
for the model with independence assumption in this case (see also Figure
25) so that the specification of a spatial correlation structure seems to make
fairly little difference in terms of the BLUPs. As expected, observed residu-
als $\hat{\epsilon}_{it}$ are occasionally somewhat larger for the models with stronger spatial
autocorrelation (see Figure 30). The advantage of specifying a correlation
structure for the random effects in this case is that it allows interpolation or
prediction for previously unobserved locations and improves precision of the
estimates if the specified correlation structure approximates the true one.

COMPUTATIONAL DETAILS

This section was compiled with knitr (Xie, 2012), under the following
setup:

```r
sessionInfo()
# R version 2.14.1 (2011-12-22)
## Platform: i386-pc-mingw32/i386 (32-bit)
##
## locale:
## 
## attached base packages:
## [1] splines grid stats graphics grDevices utils datasets
## [8] methods base
##
## other attached packages:
## [1] fda_2.2.7 zoo_1.7-7 mapdata_2.2-1 maps_2.2-5
## [5] fields_6.6.3 spam_0.28-0 tikzDevice_0.6.2 filehash_2.2
## [9] animation_2.0-6 knitr_0.4 ggplot2_0.8.9 proto_0.3-9.2
## [13] reshape_0.8.4plyr_1.7.1 gamm4_0.1-5 mgcv_1.7-14
## [17] lme4_0.999375-42 Matrix_1.0-3 lattice_0.19-13 mvtnorm_0.9-9992
## [21] rj_1.0.3-7
##
## loaded via a namespace (and not attached):
## [1] codetools_0.2-8 digest_0.5.1 evaluate_0.4.2 formatR_0.4
## [5] highlight_0.3.1 nlme_3.1-97 parser_0.0-14 Rcpp_0.9.10
## [9] rj.gd_1.0.3-3 stats4_2.14.1 stringr_0.6 tools_2.14.1
```

The version of pffr used for this supplement is included in the code folder.
REFERENCES


