Supplementary materials to understanding the stochastic partial differential equation approach to smoothing

David L Miller * Richard Glennie* Andrew E Seaton*

1 Supplementary results

Below are more detailed derivations of results referred to in the main paper.

Proposition 1. *For a linear differential operator D, stochastic process f, and white noise process ϵ, there exists a function w such that $f(x) = \int w(x-u)\epsilon(u)\,\mathrm{d}u$.*

Proof. To demonstrate this, we use the Fourier transform, $\mathcal{F}$. Suppose D has the form $D = \sum_{k=1}^{K} \alpha_K \partial^{m_k}/\partial x^{m_k}$. From the basic properties of the Fourier transform, it follows that $\mathcal{F}(Df) = \left(\sum_{k=1}^{K} \alpha_k \xi^{m_k}\right)\mathcal{F}(f)$ for frequency variable ξ.

Let w be the function such that $\mathcal{F}(w)^{-1} = \sum_{k=1}^{K} \alpha_k \xi^{m_k}$. We then have that $\mathcal{F}(f) = \mathcal{F}(w)\mathcal{F}(\epsilon)$. Hence, by properties of Fourier transform, the function f is the convolution of w and ϵ: $f(x) = \int w(x-u)\epsilon(u)\,\mathrm{d}u$.

This demonstration omits the necessary justification that the Fourier transform of stochastic processes is well defined and the usual properties of the Fourier transform apply in this context.

□

Proposition 2. *Given that $f(x) = \int w(x-u)\epsilon(u)\,\mathrm{d}u$ for white noise process ϵ and weighting function w, the covariance between $f(x), f(y)$ is given by $k(x,y) = \int w(x-u)w(y-u)\,\mathrm{d}u$.*

*Joint first author. *Address:* Centre for Research into Ecological & Environmental Modelling and School of Mathematics & Statistics, University of St Andrews, St Andrews, Fife, Scotland

Proof. It is easily seen that as a convolution of white noise $\mathbb{E}(f(x)) = 0$ for all x. Thus,

$$\begin{aligned} k(x,y) &= \mathbb{E}(f(x)f(y)) \\ &= \mathbb{E}\left\{\int w(x-u)\epsilon(u)\,\mathrm{d}u \int w(x-v)\epsilon(v)\,\mathrm{d}v\right\} \\ &= \mathbb{E}\left\{\int w(x-u)w(y-u)\,\mathrm{d}u\right\} \qquad \text{by Itô's isometry.} \\ &= \int w(x-u)w(y-u)\,\mathrm{d}u \end{aligned}$$

□

Proposition 3. *Let D be a differential operator, f be a stochastic process, and ϵ be the white noise process. Let $\psi_1, \ldots, \psi_M$ be a basis over the space of interest and suppose $f(x) = \sum_{j=1}^{M} \beta_j \psi_j(x)$.*

For the SPDE, $Df = \epsilon$, let the precision matrix of $\boldsymbol{\beta}$ approximated by the finite element method be $\mathbf{Q}$. Furthermore, let $\mathbf{S}$ be the smoothing penalty matrix associated with the penalty $\langle Df, Df\rangle$.

In this case, $\mathbf{Q} = \mathbf{S}$.

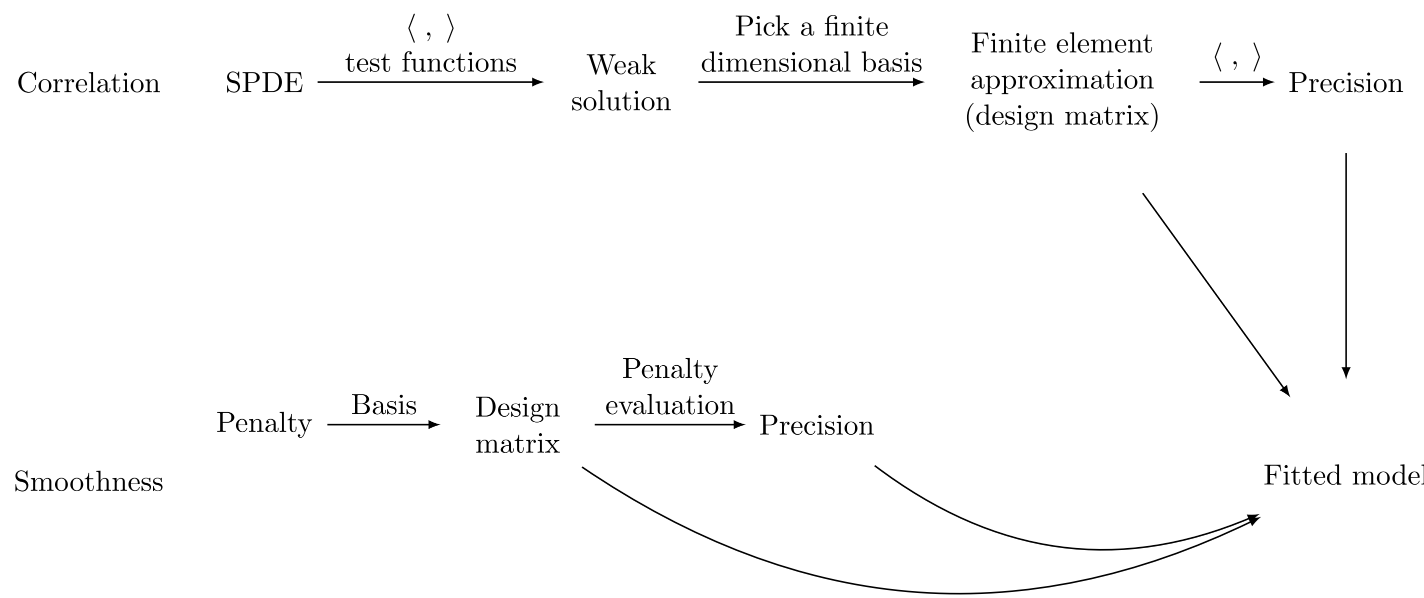
Proof. In the main paper, it is shown that $\mathbf{Q} = \mathbf{P}^\top \mathbf{Q}_e \mathbf{P}$ where $\mathbf{P}$ has $(i,j)^{\text{th}}$ entry $\langle D\psi_i, \psi_j\rangle$. It is also easily shown that $\mathbf{Q}_e^{-1}$ has $(i,j)^{\text{th}}$ entry $\langle \psi_i, \psi_j\rangle$.

Consider a function $g(x) = \sum_{j=1}^{M} \alpha_j \psi_j$. It follows that $\mathbf{Q}_e^{-1}\boldsymbol{\alpha}$ has i^{th} entry $\sum_{j=1}^{M} \alpha_j \langle \psi_i, \psi_j\rangle = \langle \psi_i, g\rangle$. Looking at this the other way around, we can see that $\mathbf{Q}_e$ transforms vectors with i^{th} element $\langle \psi_i, g\rangle$ to vectors with i^{th} element α_i when $g = \sum_{i=1}^{M} \alpha_i \psi_i$.

The j^{th} column of $\mathbf{P}$ has i^{th} element $\langle D\psi_i, \psi_j\rangle$ and so $\mathbf{Q}_e\mathbf{P}$ has $(i,j)^{\text{th}}$ element $\alpha_{i,j}$ such that $D\psi_i = \sum_{j=1}^{M} \alpha_{i,j}\psi_j$. We therefore have that $\mathbf{P}^\top \mathbf{Q}_e \mathbf{P}$ has $(i,j)^{\text{th}}$ entry $\sum_{k=1}^{M} \alpha_{i,k}\langle \psi_k, D\psi_j\rangle = \langle \sum_{k=1}^{M} \alpha_{i,k}\psi_k, D\psi_j\rangle = \langle D\psi_i, D\psi_j\rangle$. This coincides with the definition of $\mathbf{S}$. □

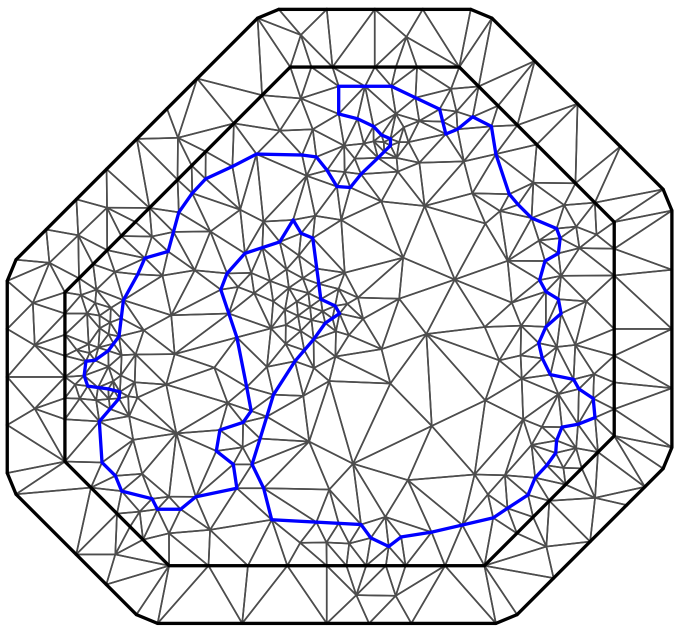
2 Flow Diagram

Supplementary Figure 1 shows a flow diagram of the modelling workflow for the SPDE and basis-penalty approaches.

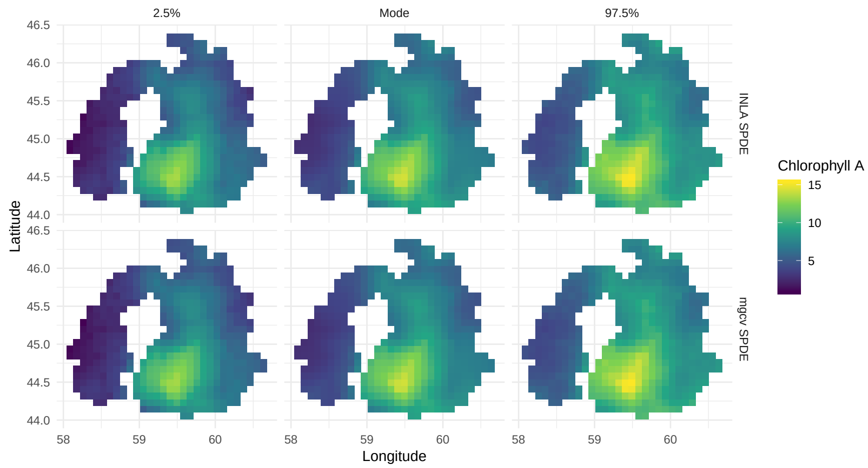


Supplementary Figure 1. Flow diagram showing the two approaches (correlation and smoothness) to fitting flexible, structured models.

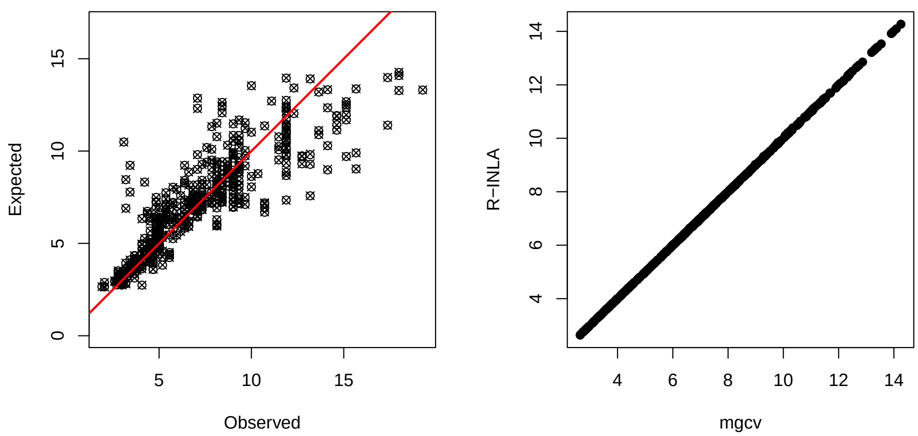
3 Examples



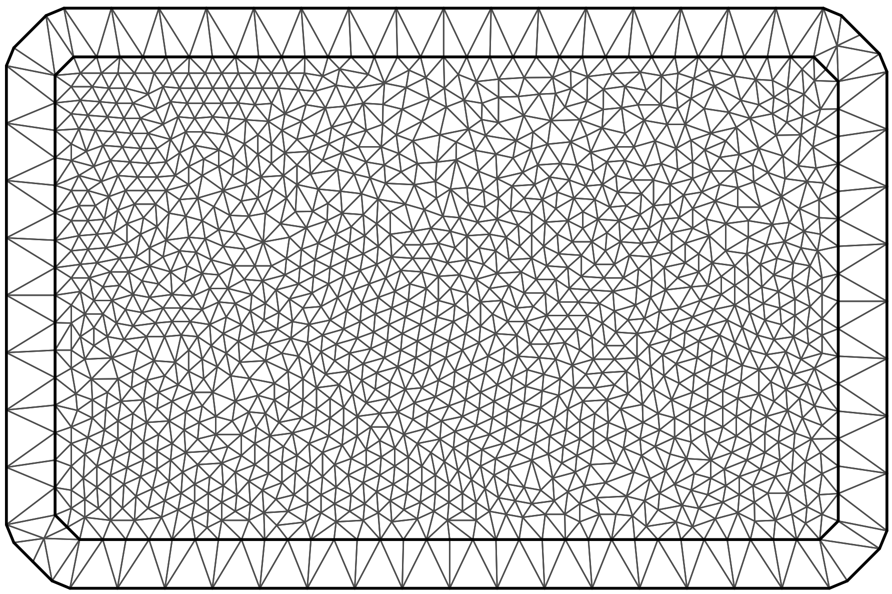
Supplementary Figure 2. The mesh used to fit the Aral sea data. The mesh was generated using the `meshbuilder` Shiny application in the R package `fmesher`.



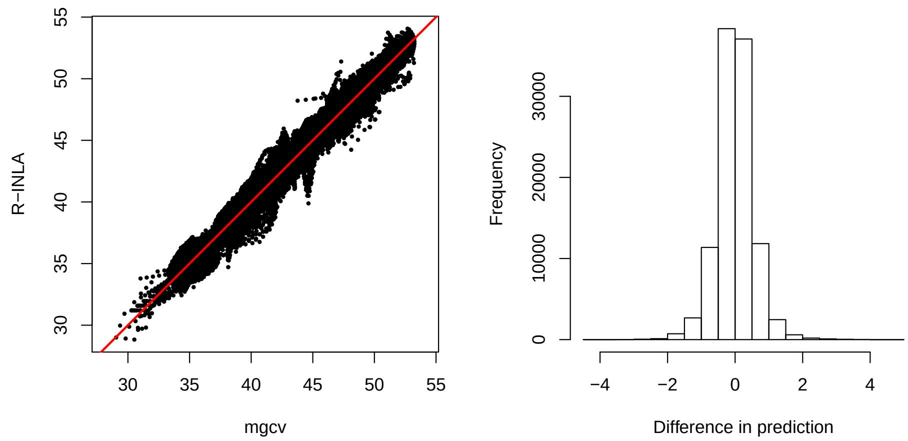
Supplementary Figure 3. Credible intervals for the two models fitted to the chlorophyll Aral sea example. Top row shows predictions from the SPDE fitted using `R-INLA` and bottom row shows the predictions from the SPDE model fitted using `mgcv`. Left plot shows the 2.5% credible surface, middle shows the mode and right shows the 97.5% credible surface.



Supplementary Figure 4. Comparison of `R-INLA` and mgcv SPDE results for the Aral sea example. Left shows the raw data (log chlorophyll A) against model predictions for `R-INLA` (crosses) and mgcv (circles), showing relatively good fit to the data and good match-up between the results. The right plot shows predictions from `R-INLA` versus those from mgcv again showing that the results match.



Supplementary Figure 5. The mesh used to fit the MODIS temperature data. The mesh was generated using the `meshbuilder` Shiny application in the R package `fmesher`.



Supplementary Figure 6. Comparison of predictions for the two models fitted the MODIS temperature data. Left plot shows the predictions plotted against each other, with a red line showing perfect agreement for reference. Right plot shows a histogram of the differences between the model predictions. The largest absolute difference was 4.761, small compared to the range of the data.

4 Matérn penalty parameterisation

This parametrisation allows us to simultaneously estimate τ and κ within `mgcv`. This relies on a reparametrisation in terms of the log smoothing parameters. Going back to our precision matrix, if we write $\lambda_1 = \tau^2\kappa^4$, $\lambda_2 = \kappa^2\tau^2$ and $\lambda_3 = \tau^2$ we can then use the following parametrisation:

$$\begin{aligned}
\log(\lambda_1) &= 2\log(\tau) + 4\log(\kappa),\\
\log(\lambda_2) &= 2\log(\tau) + 2\log(\kappa),\\
\log(\lambda_3) &= 2\log(\tau),
\end{aligned}$$

i.e., the log smoothing parameters ($\log(\lambda_j)$) are a linear transformation of the "raw" smoothing parameters $\log(\tau)$ and $\log(\kappa)$. We can write this transformation in matrix form as:

$$\mathbf{L}\begin{bmatrix}\tau\\ \kappa\end{bmatrix} = \boldsymbol{\lambda} \quad \text{where } \mathbf{L} = \begin{bmatrix}2 & 4\\ 2 & 2\\ 2 & 0\end{bmatrix}$$

The matrix $\mathbf{L}$ can be supplied as an argument to `paraPen`, in this case the smoothing parameters returned by `mgcv` are τ and κ.

Code for this parameterisation is provided as part of the online supplementary material. Thanks to Simon Wood for suggesting this approach.

5 Notation

This Paper	**Lindgren et al. (2011)**	**Meaning**
z	y	Response variable
$x, x_1, x_2, \ldots$	u, v	Location in space or time
$\eta(x)$	–	Fixed effect linear predictor at x
$f(x)$	$x(u)$	Random field value at location x
$\epsilon(x)$	$\mathcal{W}(u)$	Gaussian white noise at location x
$c(x_i, x_j)$	$r(u, v)$	Covariance between locations x_i, x_j
Σ	Q^{-1}	Variance-covariance matrix
Q	Q	Precision matrix
ψ	ψ	Basis function
ϕ	ϕ	Test function
λ	–	Smoothing parameter
Δ	Δ	Laplace operator
∇	∇	Gradient operator
w	–	Weighting function
κ	κ	Inverse range parameter
τ	–	Variance parameter
α	α	Smoothness parameter
ν	ν	Smoothness parameter related to α
d	d	Dimension 1D, 2D,...
β_i	β_i and w_i	Basis coefficients
M	n	Number of basis functions / test functions
C	C	Matrix of inner products $\langle \psi_i, \psi_j \rangle$
G	G	Matrix of inner products $\langle \nabla\psi_i, \nabla\phi_i \rangle$

Supplementary Table 1. Notation conversion table between our paper and Lindgren et al. (2011)