# Demystifying the SPDE approach to spatial modelling

Andy Seaton

University of St Andrews

# Acknowledgements

**Prof Finn Lindgren**, University of Edinburgh

**Dr David Miller**, University of St Andrews

**Dr Richard Glennie**, University of St Andrews

**Dr Janine Illian**, University of St Andrews

**Prof David Borchers**, University of St Andrews

**Mr Rick Camp**, University of St Andrews and US Geological Survey

**Dr Ben Stevenson**, University of Auckland

# Overview

1. The SPDE approach

# Overview

1. The SPDE approach
2. The basis-penalty smoother approach

# Overview

1. The SPDE approach
2. The basis-penalty smoother approach
3. Examples

# Introduction

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

# Introduction

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

- $y(x)$ is the response variable

# Introduction

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

- $y(x)$ is the response variable
- $\eta(x)$ are the fixed effects

# Introduction

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

- $y(x)$ is the response variable
- $\eta(x)$ are the fixed effects
- $\epsilon(x)$ is the unstructured error

# Introduction

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

- $y(x)$ is the response variable
- $\eta(x)$ are the fixed effects
- $\epsilon(x)$ is the unstructured error
- $f(x)$ is the “structured random effect”

# Introduction

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

# Introduction

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

- A flexible model for $f(x)$ is a Gaussian process

# Introduction

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

- A flexible model for $f(x)$ is a Gaussian process
- Common notation: $f \sim \mathcal{GP}(0, c(x, x'))$

# Introduction

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

- A flexible model for $f(x)$ is a Gaussian process
- Common notation: $f \sim \mathcal{GP}(0, c(x, x'))$
- $c(x, x') = \text{Cov}\left[f(x), f(x')\right]$ is the **covariance function**

# Introduction

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

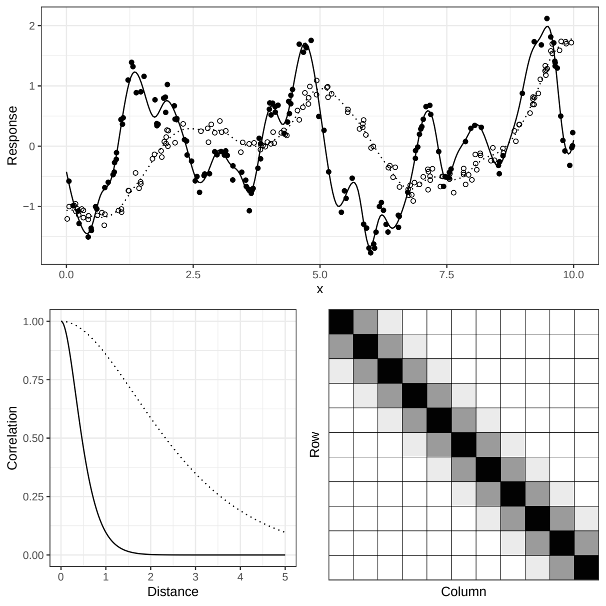
- A flexible model for $f(x)$ is a Gaussian process
- Common notation: $f \sim \mathcal{GP}(0, c(x, x'))$
- $c(x, x') = \mathrm{Cov}\left[f(x), f(x')\right]$ is the **covariance function**
- $c(x, x')$ often chosen to decay with increasing distance between $x$ and $x'$

# Introduction

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

- A flexible model for $f(x)$ is a Gaussian process
- Common notation: $f \sim \mathcal{GP}(0, c(x, x'))$
- $c(x, x') = \mathrm{Cov}\,[f(x), f(x')]$ is the **covariance function**
- $c(x, x')$ often chosen to decay with increasing distance between $x$ and $x'$
- $[f(x_1), \ldots, f(x_M)] \sim \mathcal{N}(0, \Sigma^{-1})$ where $\Sigma_{ij} = c(x_i, x_j)$

# 1D Gaussian Process Example



# The SPDE Approach

# The SPDE Approach

Instead of $f \sim \mathcal{GP}(0, c(x, x'))$, we say $f$ is a solution to a stochastic partial differential equation (SPDE)

# The SPDE Approach

Instead of $f \sim \mathcal{GP}(0, c(x, x'))$, we say $f$ is a solution to a stochastic partial differential equation (SPDE)

$$\mathcal{D}f(x) = \epsilon(x)$$

# The SPDE Approach

Instead of $f \sim \mathcal{GP}(0, c(x, x'))$, we say $f$ is a solution to a stochastic partial differential equation (SPDE)

$$\mathcal{D}f(x) = \epsilon(x)$$

- $\mathcal{D}$ is a **linear differential operator**

# The SPDE Approach

Instead of $f \sim \mathcal{GP}(0, c(x, x'))$, we say $f$ is a solution to a stochastic partial differential equation (SPDE)

$$\mathcal{D}f(x) = \epsilon(x)$$

- $\mathcal{D}$ is a **linear differential operator**
- Examples: $\mathcal{D} = \frac{\mathrm{d}}{\mathrm{d}x}$, $\mathcal{D} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$

# The SPDE Approach

Instead of $f \sim \mathcal{GP}(0, c(x, x'))$, we say $f$ is a solution to a stochastic partial differential equation (SPDE)

$$\mathcal{D}f(x) = \epsilon(x)$$

- $\mathcal{D}$ is a **linear differential operator**
- Examples: $\mathcal{D} = \frac{\mathrm{d}}{\mathrm{d}x}$, $\mathcal{D} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$
- $[\epsilon(x_1), \ldots, \epsilon(x_M)] \sim \mathcal{N}(0, \boldsymbol{I}_M)$ (a.k.a Gaussian white noise process)

# The SPDE Approach

Instead of $f \sim \mathcal{GP}(0, c(x, x'))$, we say $f$ is a solution to a stochastic partial differential equation (SPDE)

$$\mathcal{D}f(x) = \epsilon(x)$$

- $\mathcal{D}$ is a **linear differential operator**
- Examples: $\mathcal{D} = \frac{\mathrm{d}}{\mathrm{d}x}$, $\mathcal{D} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$
- $[\epsilon(x_1), \ldots, \epsilon(x_M)] \sim \mathcal{N}(0, \boldsymbol{I}_M)$ (a.k.a Gaussian white noise process)
- $f$ is a Gaussian process

# The SPDE Approach

Instead of $f \sim \mathcal{GP}(0, c(x, x'))$, we say $f$ is a solution to a stochastic partial differential equation (SPDE)

$$\mathcal{D}f(x) = \epsilon(x)$$

- $\mathcal{D}$ is a **linear differential operator**
- Examples: $\mathcal{D} = \frac{\mathrm{d}}{\mathrm{d}x}$, $\mathcal{D} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$
- $[\epsilon(x_1), \ldots, \epsilon(x_M)] \sim \mathcal{N}(0, \boldsymbol{I}_M)$ (a.k.a Gaussian white noise process)
- $f$ is a Gaussian process
- The **covariance function** of $f$ can be derived from $\mathcal{D}$ and vice versa

# The SPDE Approach

Instead of $f \sim \mathcal{GP}(0, c(x, x'))$, we say $f$ is a solution to a stochastic partial differential equation (SPDE)

$$\mathcal{D}f(x) = \epsilon(x)$$

- $\mathcal{D}$ is a **linear differential operator**
- Examples: $\mathcal{D} = \frac{\mathrm{d}}{\mathrm{d}x}$, $\mathcal{D} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$
- $[\epsilon(x_1), \dots, \epsilon(x_M)] \sim \mathcal{N}(0, \boldsymbol{I}_M)$ (a.k.a Gaussian white noise process)
- $f$ is a Gaussian process
- The **covariance function** of $f$ can be derived from $\mathcal{D}$ and vice versa
- The SPDE is an **equivalent way of defining the Gaussian process**

# The SPDE Approach - Finite Element Methods

Solving the SPDE:

$$\mathcal{D}f(x) = \epsilon(x)$$

# The SPDE Approach - Finite Element Methods

Solving the SPDE:

$$\mathcal{D}f(x) = \epsilon(x)$$
$$\int \mathcal{D}f(x)\phi(x)\mathrm{d}x = \int \epsilon(x)\phi(x)\mathrm{d}x$$

# The SPDE Approach - Finite Element Methods

Solving the SPDE:

$$\mathcal{D}f(x) = \epsilon(x)$$
$$\int \mathcal{D}f(x)\phi(x)\mathrm{d}x = \int \epsilon(x)\phi(x)\mathrm{d}x$$
$$\langle \mathcal{D}f, \phi \rangle = \langle \epsilon, \phi \rangle$$

# The SPDE Approach - Finite Element Methods

Solving the SPDE:

$$\mathcal{D}f(x) = \epsilon(x)$$
$$\int \mathcal{D}f(x)\phi(x)\mathrm{d}x = \int \epsilon(x)\phi(x)\mathrm{d}x$$
$$\langle \mathcal{D}f, \phi \rangle = \langle \epsilon, \phi \rangle$$

for any **test function** $\phi(x)$

# The SPDE Approach - Finite Element Methods

Solving the SPDE:

$$\mathcal{D}f(x) = \epsilon(x)$$
$$\int \mathcal{D}f(x)\phi(x)\mathrm{d}x = \int \epsilon(x)\phi(x)\mathrm{d}x$$
$$\langle \mathcal{D}f, \phi \rangle = \langle \epsilon, \phi \rangle$$

for any **test function** $\phi(x)$

This is not just a trick, this is the *definition* of what is meant when we write $\mathcal{D}f(x) = \epsilon(x)$

# The SPDE Approach - Finite Element Methods

Solving the SPDE:

$$\mathcal{D}f(x) = \epsilon(x)$$
$$\int \mathcal{D}f(x)\phi(x)\mathrm{d}x = \int \epsilon(x)\phi(x)\mathrm{d}x$$
$$\langle \mathcal{D}f, \phi \rangle = \langle \epsilon, \phi \rangle$$

for any **test function** $\phi(x)$

This is not just a trick, this is the *definition* of what is meant when we write $\mathcal{D}f(x) = \epsilon(x)$

(see - Generalised Functions/Distributions)

# The SPDE Approach - Finite Element Methods

Solving the SPDE:

$$\mathcal{D}f(x) = \epsilon(x)$$

$$\langle \mathcal{D}f, \phi \rangle = \langle \epsilon, \phi \rangle$$

# The SPDE Approach - Finite Element Methods

Solving the SPDE:

$$\mathcal{D}f(x) = \epsilon(x)$$
$$\langle \mathcal{D}f, \phi \rangle = \langle \epsilon, \phi \rangle$$

Pick a set of **basis functions** $\phi_1(x), \ldots, \phi_M(x)$ to represent $f(x)$

# The SPDE Approach - Finite Element Methods

Solving the SPDE:

$$\mathcal{D}f(x) = \epsilon(x)$$
$$\langle \mathcal{D}f, \phi \rangle = \langle \epsilon, \phi \rangle$$

Pick a set of **basis functions** $\phi_1(x), \ldots, \phi_M(x)$ to represent $f(x)$

Also use this basis as the set of test functions

# The SPDE Approach - Finite Element Methods

Solving the SPDE:

$$\mathcal{D}f(x) = \epsilon(x)$$
$$\langle \mathcal{D}f, \phi \rangle = \langle \epsilon, \phi \rangle$$

Pick a set of **basis functions** $\phi_1(x), \ldots, \phi_M(x)$ to represent $f(x)$

Also use this basis as the set of test functions

$$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x) \implies \sum_{i=1}^{M} \beta_i \langle \mathcal{D}\phi_i, \phi_j \rangle = \langle \epsilon, \phi_j \rangle$$

for $j = 1, \ldots, M$

# The SPDE Approach - Finite Element Methods

Solving the SPDE:

$$\mathcal{D}f(x) = \epsilon(x)$$
$$\langle \mathcal{D}f, \phi \rangle = \langle \epsilon, \phi \rangle$$

Pick a set of **basis functions** $\phi_1(x), \ldots, \phi_M(x)$ to represent $f(x)$

Also use this basis as the set of test functions

$$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x) \implies \sum_{i=1}^{M} \beta_i \langle \mathcal{D}\phi_i, \phi_j \rangle = \langle \epsilon, \phi_j \rangle$$

for $j = 1, \ldots, M$

i.e. A set of $M$ linear equations we can write in matrix-vector notation:

$$\boldsymbol{P}\boldsymbol{\beta} = \boldsymbol{e}$$

where $\boldsymbol{P}_{ij} = \langle \mathcal{D}\phi_i, \phi_j \rangle$ and $\boldsymbol{e}_j = \langle \epsilon, \phi_j \rangle$

# The SPDE Approach - Finite Element Methods

- Given $\mathcal{D}f(x) = \epsilon(x)$

# The SPDE Approach - Finite Element Methods

- Given $\mathcal{D}f(x) = \epsilon(x)$
- Choose a **basis** for $f$: $\phi_1, \ldots, \phi_M$ with associated coefficients $\beta_1, \ldots, \beta_M$

# The SPDE Approach - Finite Element Methods

- Given $\mathcal{D}f(x) = \epsilon(x)$
- Choose a **basis** for $f$: $\phi_1, \ldots, \phi_M$ with associated coefficients $\beta_1, \ldots, \beta_M$
- Choose a set of **test functions**: also $\phi_1, \ldots, \phi_M$

# The SPDE Approach - Finite Element Methods

- Given $\mathcal{D}f(x) = \epsilon(x)$
- Choose a **basis** for $f$: $\phi_1, \ldots, \phi_M$ with associated coefficients $\beta_1, \ldots, \beta_M$
- Choose a set of **test functions**: also $\phi_1, \ldots, \phi_M$
- Represent the SPDE as matrix-vector equation: $\boldsymbol{P}\boldsymbol{\beta} = \boldsymbol{e}$

# The SPDE Approach - Finite Element Methods

- Given $\mathcal{D}f(x) = \epsilon(x)$
- Choose a **basis** for $f$: $\phi_1, \ldots, \phi_M$ with associated coefficients $\beta_1, \ldots, \beta_M$
- Choose a set of **test functions**: also $\phi_1, \ldots, \phi_M$
- Represent the SPDE as matrix-vector equation: $\boldsymbol{P}\boldsymbol{\beta} = \boldsymbol{e}$

- $\boldsymbol{P}$ is fixed and known

# The SPDE Approach - Finite Element Methods

- Given $\mathcal{D}f(x) = \epsilon(x)$
- Choose a **basis** for $f$: $\phi_1, \ldots, \phi_M$ with associated coefficients $\beta_1, \ldots, \beta_M$
- Choose a set of **test functions**: also $\phi_1, \ldots, \phi_M$
- Represent the SPDE as matrix-vector equation: $\boldsymbol{P}\boldsymbol{\beta} = \boldsymbol{e}$

- $\boldsymbol{P}$ is fixed and known
- $\boldsymbol{e}$ has known distribution $\mathcal{N}(0, \boldsymbol{Q_e})$

# The SPDE Approach - Finite Element Methods

- Given $\mathcal{D}f(x) = \epsilon(x)$
- Choose a **basis** for $f$: $\phi_1, \ldots, \phi_M$ with associated coefficients $\beta_1, \ldots, \beta_M$
- Choose a set of **test functions**: also $\phi_1, \ldots, \phi_M$
- Represent the SPDE as matrix-vector equation: $\boldsymbol{P}\boldsymbol{\beta} = \boldsymbol{e}$

- $\boldsymbol{P}$ is fixed and known
- $\boldsymbol{e}$ has known distribution $\mathcal{N}(0, \boldsymbol{Q_e})$
- Can show $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{Q})$, where $\boldsymbol{Q} = \boldsymbol{P}^T \boldsymbol{Q_e} \boldsymbol{P}$

# The SPDE Approach - Finite Element Methods

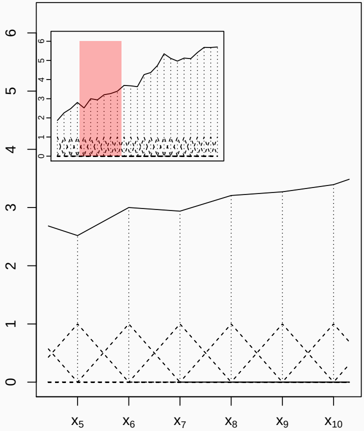
- Given $\mathcal{D}f(x) = \epsilon(x)$
- Choose a **basis** for $f$: $\phi_1, \ldots, \phi_M$ with associated coefficients $\beta_1, \ldots, \beta_M$
- Choose a set of **test functions**: also $\phi_1, \ldots, \phi_M$
- Represent the SPDE as matrix-vector equation: $\boldsymbol{P}\boldsymbol{\beta} = \boldsymbol{e}$

- $\boldsymbol{P}$ is fixed and known
- $\boldsymbol{e}$ has known distribution $\mathcal{N}(0, \boldsymbol{Q}_{\boldsymbol{e}})$
- Can show $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{Q})$, where $\boldsymbol{Q} = \boldsymbol{P}^T \boldsymbol{Q}_{\boldsymbol{e}} \boldsymbol{P}$

Given a basis representation $f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$,

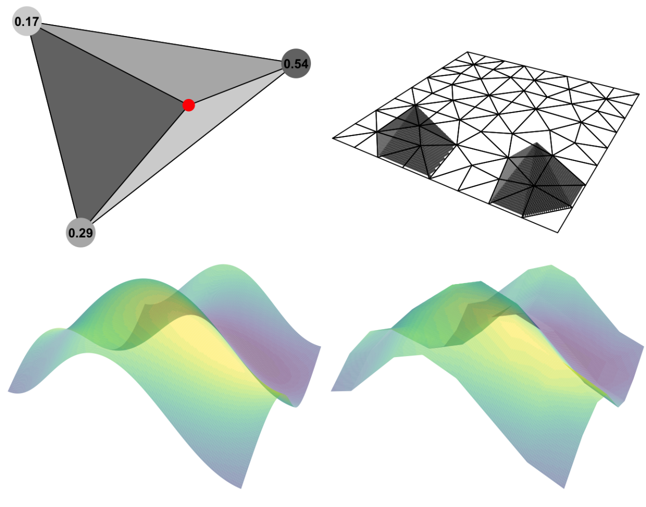
**the SPDE imposes a multivariate-normal prior on the $\beta$'s**

# The SPDE Approach - Finite Element Method 1D




Piecewise linear basis example

# The SPDE Approach - Finite Element Method 2D




source: Advanced Spatial Modelling with SPDEs Using R and INLA, Krainski et al. (2018)

# The Basis-Penalty Smoothing Approach

# The Basis-Penalty Smoothing Approach

Penalised log-likelihood:

$$\ell_p(\boldsymbol{\beta}, \lambda) = \ell(\boldsymbol{\beta}) - \lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$$

# The Basis-Penalty Smoothing Approach

Penalised log-likelihood:

$$\ell_p(\boldsymbol{\beta}, \lambda) = \ell(\boldsymbol{\beta}) - \lambda \int (\mathcal{D}f(x))^2 \,\mathrm{d}x$$

- $\ell(\beta)$ is a measure of fit (log-likelihood)

# The Basis-Penalty Smoothing Approach

Penalised log-likelihood:

$$\ell_p(\boldsymbol{\beta}, \lambda) = \ell(\boldsymbol{\beta}) - \lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$$

- $\ell(\beta)$ is a measure of fit (log-likelihood)
- $\lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$ is a **smoothing penalty**

# The Basis-Penalty Smoothing Approach

Penalised log-likelihood:

$$\ell_p(\boldsymbol{\beta}, \lambda) = \ell(\boldsymbol{\beta}) - \lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$$

- $\ell(\beta)$ is a measure of fit (log-likelihood)
- $\lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$ is a **smoothing penalty**
- $\lambda$ is a **smoothing paramater** that controls the amount of penalisation

# The Basis-Penalty Smoothing Approach

Penalised log-likelihood:

$$\ell_p(\boldsymbol{\beta}, \lambda) = \ell(\boldsymbol{\beta}) - \lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$$

- $\ell(\beta)$ is a measure of fit (log-likelihood)
- $\lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$ is a **smoothing penalty**
- $\lambda$ is a **smoothing paramater** that controls the amount of penalisation
- Optimise the combination of log-likelihood and penalty

# The Basis-Penalty Smoothing Approach

Penalised log-likelihood:

$$\ell_p(\boldsymbol{\beta}, \lambda) = \ell(\boldsymbol{\beta}) - \lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$$

- $\ell(\beta)$ is a measure of fit (log-likelihood)
- $\lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$ is a **smoothing penalty**
- $\lambda$ is a **smoothing paramater** that controls the amount of penalisation
- Optimise the combination of log-likelihood and penalty

We can play the same game. Choose a basis representation for $f(x)$:

# The Basis-Penalty Smoothing Approach

Penalised log-likelihood:

$$\ell_p(\boldsymbol{\beta}, \lambda) = \ell(\boldsymbol{\beta}) - \lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$$

- $\ell(\beta)$ is a measure of fit (log-likelihood)
- $\lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$ is a **smoothing penalty**
- $\lambda$ is a **smoothing paramater** that controls the amount of penalisation
- Optimise the combination of log-likelihood and penalty

We can play the same game. Choose a basis representation for $f(x)$:

$$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x) \implies \lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x = \boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta}$$

where $(\boldsymbol{S}_\lambda)_{ij} = \lambda \int \mathcal{D}\phi_i(x) \mathcal{D}\phi_j(x) \mathrm{d}x$

# The Basis-Penalty Smoothing Approach

$$\ell_p(\boldsymbol{\beta}, \lambda) = \ell(\boldsymbol{\beta}) - \boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta}$$

# The Basis-Penalty Smoothing Approach

$$\ell_p(\boldsymbol{\beta}, \lambda) = \ell(\boldsymbol{\beta}) - \boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta}$$

$$\mathcal{L}_p(\boldsymbol{\beta}, \boldsymbol{\lambda}) = \mathcal{L}(\boldsymbol{\beta}) \exp(-\boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta})$$

# The Basis-Penalty Smoothing Approach

$$\ell_p(\boldsymbol{\beta}, \lambda) = \ell(\boldsymbol{\beta}) - \boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta}$$

$$\mathcal{L}_p(\boldsymbol{\beta}, \boldsymbol{\lambda}) = \mathcal{L}(\boldsymbol{\beta}) \exp(-\boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta})$$

- $\exp(-\boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta})$ is proportional to a multivariate normal density with precision matrix $\boldsymbol{S}_\lambda$

# The Basis-Penalty Smoothing Approach

$$\ell_p(\boldsymbol{\beta}, \lambda) = \ell(\boldsymbol{\beta}) - \boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta}$$

$$\mathcal{L}_p(\boldsymbol{\beta}, \boldsymbol{\lambda}) = \mathcal{L}(\boldsymbol{\beta}) \exp(-\boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta})$$

- $\exp(-\boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta})$ is proportional to a multivariate normal density with precision matrix $\boldsymbol{S}_\lambda$
- Take a Bayesian interpretation of the penalised log-likelihood and view this as a prior

# The Basis-Penalty Smoothing Approach

$$\ell_p(\boldsymbol{\beta}, \lambda) = \ell(\boldsymbol{\beta}) - \boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta}$$

$$\mathcal{L}_p(\boldsymbol{\beta}, \boldsymbol{\lambda}) = \mathcal{L}(\boldsymbol{\beta}) \exp(-\boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta})$$

- $\exp(-\boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta})$ is proportional to a multivariate normal density with precision matrix $\boldsymbol{S}_\lambda$
- Take a Bayesian interpretation of the penalised log-likelihood and view this as a prior

Given a basis representation $f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$,

**the smoothing penalty imposes a multivariate-normal prior on the $\beta$'s**

# The SPDE and Smoothing Penalty do the same thing

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

# The SPDE and Smoothing Penalty do the same thing

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

**SPDE approach**

# The SPDE and Smoothing Penalty do the same thing

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

**SPDE approach**

$\mathcal{D}f(x) = \frac{\epsilon(x)}{\sqrt{\lambda}}$

# The SPDE and Smoothing Penalty do the same thing

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

**SPDE approach**

$\mathcal{D}f(x) = \frac{\epsilon(x)}{\sqrt{\lambda}}$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

# The SPDE and Smoothing Penalty do the same thing

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

**SPDE approach**

$\mathcal{D}f(x) = \frac{\epsilon(x)}{\sqrt{\lambda}}$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

$\boldsymbol{P}_\lambda \boldsymbol{\beta} = \boldsymbol{e}$

# The SPDE and Smoothing Penalty do the same thing

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

**SPDE approach**

$\mathcal{D}f(x) = \frac{\epsilon(x)}{\sqrt{\lambda}}$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

$\boldsymbol{P}_\lambda \boldsymbol{\beta} = \boldsymbol{e}$

$\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{Q}_\lambda)$

# The SPDE and Smoothing Penalty do the same thing

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

**SPDE approach**

$\mathcal{D}f(x) = \frac{\epsilon(x)}{\sqrt{\lambda}}$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

$\boldsymbol{P}_\lambda \boldsymbol{\beta} = \boldsymbol{e}$

$\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{Q}_\lambda)$

**Basis-Penalty Smooth**

# The SPDE and Smoothing Penalty do the same thing

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

**SPDE approach**

$\mathcal{D}f(x) = \frac{\epsilon(x)}{\sqrt{\lambda}}$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

$\boldsymbol{P}_\lambda \boldsymbol{\beta} = \boldsymbol{e}$

$\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{Q}_\lambda)$

**Basis-Penalty Smooth**

$\lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$

# The SPDE and Smoothing Penalty do the same thing

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

**SPDE approach**

$\mathcal{D}f(x) = \frac{\epsilon(x)}{\sqrt{\lambda}}$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

$\boldsymbol{P}_\lambda \boldsymbol{\beta} = \boldsymbol{e}$

$\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{Q}_\lambda)$

**Basis-Penalty Smooth**

$\lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

# The SPDE and Smoothing Penalty do the same thing

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

**SPDE approach**

$\mathcal{D}f(x) = \frac{\epsilon(x)}{\sqrt{\lambda}}$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

$\boldsymbol{P}_\lambda \boldsymbol{\beta} = \boldsymbol{e}$

$\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{Q}_\lambda)$

**Basis-Penalty Smooth**

$\lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

$\boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta}$

# The SPDE and Smoothing Penalty do the same thing

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

**SPDE approach**

$\mathcal{D}f(x) = \frac{\epsilon(x)}{\sqrt{\lambda}}$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

$\boldsymbol{P}_\lambda \boldsymbol{\beta} = \boldsymbol{e}$

$\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{Q}_\lambda)$

**Basis-Penalty Smooth**

$\lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

$\boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta}$

$\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{S}_\lambda)$

# The SPDE and Smoothing Penalty do the same thing

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

**SPDE approach**

$\mathcal{D}f(x) = \frac{\epsilon(x)}{\sqrt{\lambda}}$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

$\boldsymbol{P}_\lambda \boldsymbol{\beta} = \boldsymbol{e}$

$\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{Q}_\lambda)$

**Basis-Penalty Smooth**

$\lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

$\boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta}$

$\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{S}_\lambda)$

A wonderful thing: $\boldsymbol{Q}_\lambda = \boldsymbol{S}_\lambda$

# The SPDE and Smoothing Penalty do the same thing

$$y(x) = \eta(x) + f(x) + \epsilon(x)$$

**SPDE approach**

$\mathcal{D}f(x) = \frac{\epsilon(x)}{\sqrt{\lambda}}$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

$\boldsymbol{P}_\lambda \boldsymbol{\beta} = \boldsymbol{e}$

$\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{Q}_\lambda)$

**Basis-Penalty Smooth**

$\lambda \int \left(\mathcal{D}f(x)\right)^2 \mathrm{d}x$

$f(x) = \sum_{i=1}^{M} \beta_i \phi_i(x)$

$\boldsymbol{\beta}^T \boldsymbol{S}_\lambda \boldsymbol{\beta}$

$\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{S}_\lambda)$

A wonderful thing: $\boldsymbol{Q}_\lambda = \boldsymbol{S}_\lambda$

(The optimal smoothing spline is an estimator of the posterior mean of the Gaussian process)

# The SPDE and Smoothing Penalty do the same thing

We can use `R-INLA` functions to construct the precision matrix $\boldsymbol{Q}$

# The SPDE and Smoothing Penalty do the same thing

We can use `R-INLA` functions to construct the precision matrix $\boldsymbol{Q}$

Then use $\boldsymbol{Q}$ in software most suited to our needs/experience

# The SPDE and Smoothing Penalty do the same thing

We can use `R-INLA` functions to construct the precision matrix $\boldsymbol{Q}$

Then use $\boldsymbol{Q}$ in software most suited to our needs/experience

E.g. `mgcv`, `TMB`, `JAGS`, `stan`

# The SPDE and Smoothing Penalty do the same thing

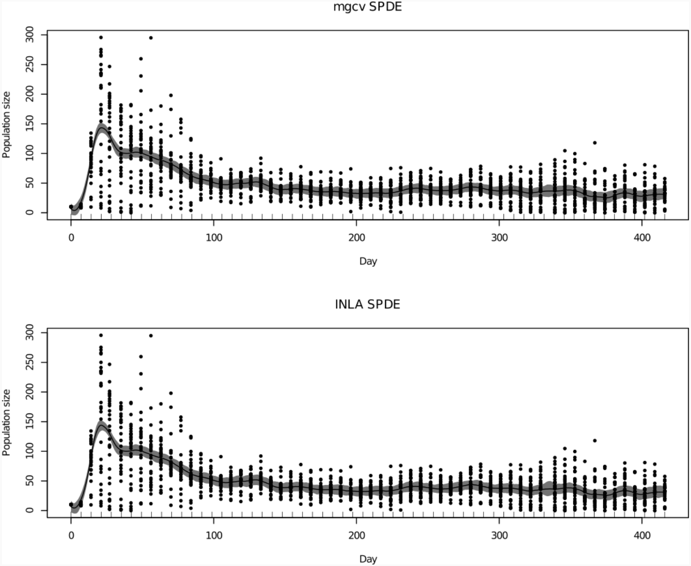
We can use `R-INLA` functions to construct the precision matrix $\boldsymbol{Q}$

Then use $\boldsymbol{Q}$ in software most suited to our needs/experience

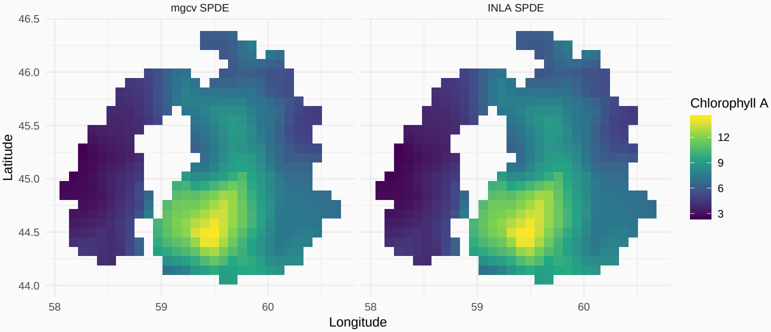
E.g. `mgcv`, `TMB`, `JAGS`, `stan`

Because $\boldsymbol{Q}$ is sparse, most efficiency advantages in software that can make use of the sparsity

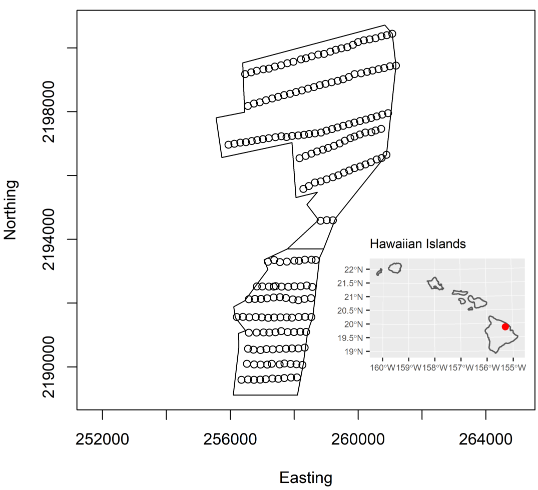
# Zooplankton Population Size



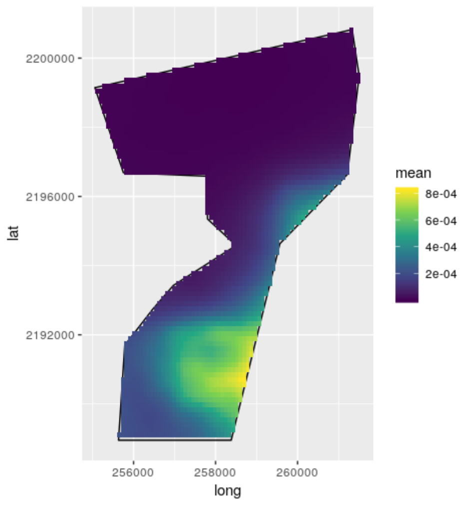
# Chlorophyll A



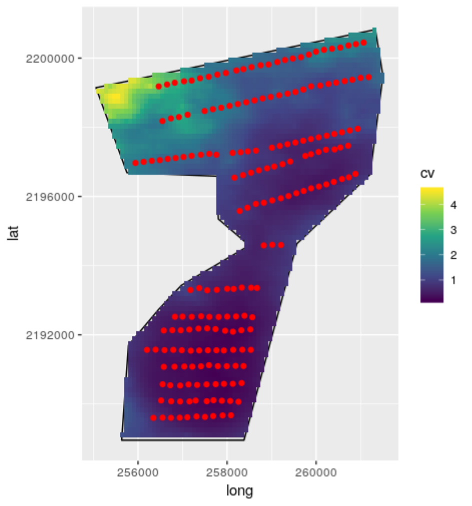
# Point transect distance sampling - Hawaiian Akepa survey



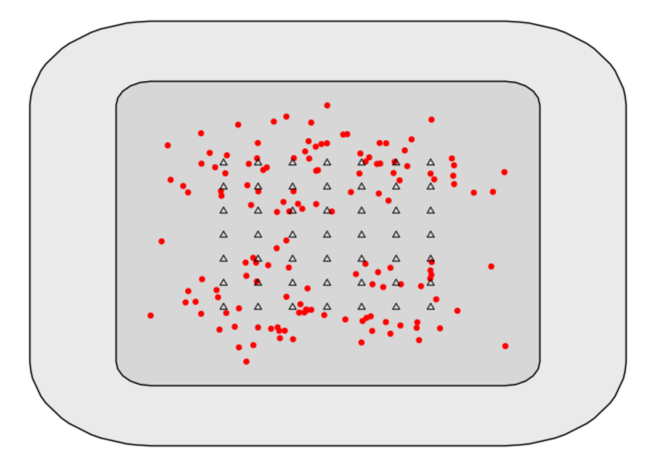
# Point transect distance sampling - Hawaiian Akepa survey



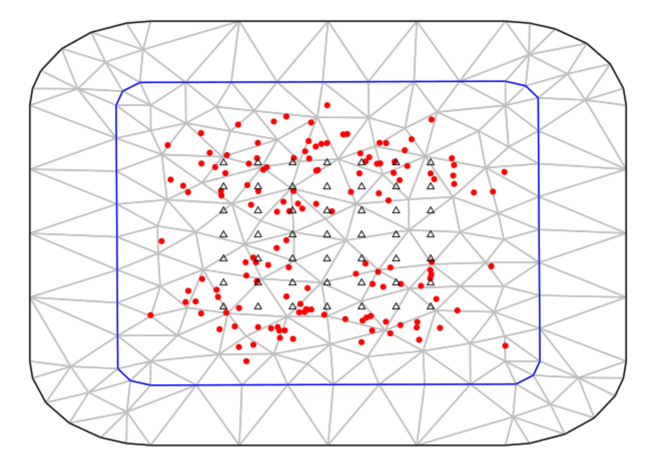
# Point transect distance sampling - Hawaiian Akepa survey



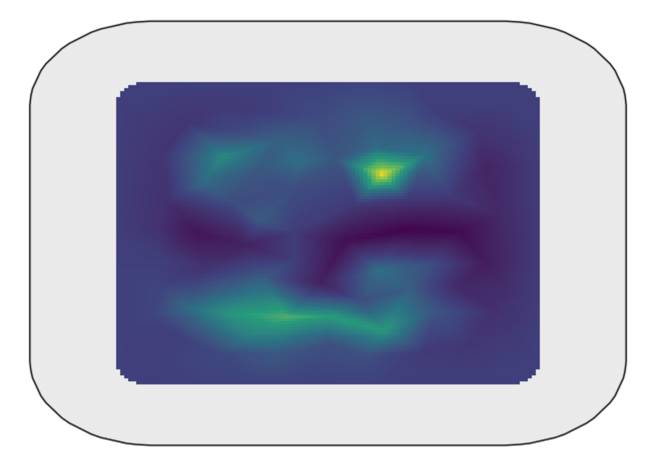
# Spatial Capture-Recapture - SPDE with TMB



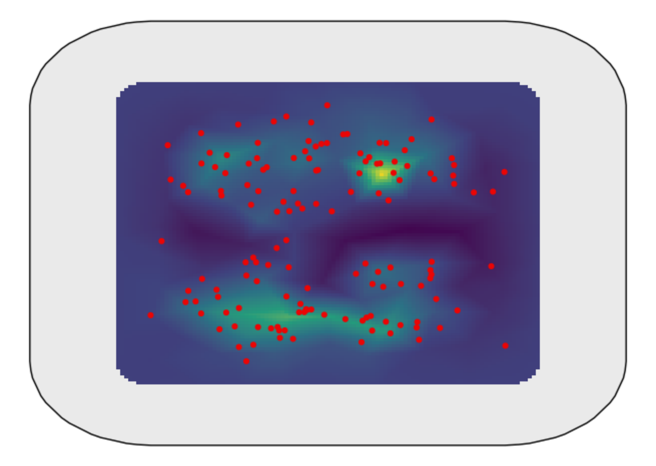
# Spatial Capture-Recapture - SPDE with TMB



# Spatial Capture-Recapture - SPDE with TMB



# Spatial Capture-Recapture - SPDE with TMB



# Summary

1. The **SPDE** places a prior on basis function coefficients

# Summary

1. The **SPDE** places a prior on basis function coefficients
2. Locations nearby are more correlated than locations far apart

# Summary

1. The **SPDE** places a prior on basis function coefficients
2. Locations nearby are more correlated than locations far apart
3. The **smoothing penalty** places a prior on basis function coefficients

## Summary

1. The **SPDE** places a prior on basis function coefficients
2. Locations nearby are more correlated than locations far apart
3. The **smoothing penalty** places a prior on basis function coefficients
4. The penalty induces smooth functions (see point 2)

# Summary

1. The **SPDE** places a prior on basis function coefficients
2. Locations nearby are more correlated than locations far apart
3. The **smoothing penalty** places a prior on basis function coefficients
4. The penalty induces smooth functions (see point 2)
5. Same differential operator $\mathcal{D}$ gives same prior in both cases

# Summary

1. The **SPDE** places a prior on basis function coefficients
2. Locations nearby are more correlated than locations far apart
3. The **smoothing penalty** places a prior on basis function coefficients
4. The penalty induces smooth functions (see point 2)
5. Same differential operator $\mathcal{D}$ gives same prior in both cases
6. Spread those SPDE wings far and wide - use $\boldsymbol{Q}$ wherever you have need

# Summary

1. The **SPDE** places a prior on basis function coefficients
2. Locations nearby are more correlated than locations far apart
3. The **smoothing penalty** places a prior on basis function coefficients
4. The penalty induces smooth functions (see point 2)
5. Same differential operator $\mathcal{D}$ gives same prior in both cases
6. Spread those SPDE wings far and wide - use $\boldsymbol{Q}$ wherever you have need

Final point: there is value in ignorance! Being new to a complicated topic is a chance to help others

# References


H. Bakka.
**How to solve the stochastic partial differential equation that gives a mat\'ern random field using the finite element method.**
*arXiv preprint arXiv:1803.03765*, 2018.

G. S. Kimeldorf and G. Wahba.
**A correspondence between Bayesian estimation on stochastic processes and smoothing by splines.**
*The Annals of Mathematical Statistics*, 41(2):495–502, 1970.

E. Krainski, V. Gmez-Rubio, H. Bakka, A. Lenzi, D. Castro-Camilo, D. Simpson, F. Lindgren, and H. Rue.
**Advanced Spatial Modeling with Stochastic Partial Differential Equations Using R and INLA.**
CRC Press, Boca Raton, FL, USA, 2019.

F. Lindgren, H. Rue, and J. Lindstrm.
**An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach.**
*Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

D. W. Nychka.
**Spatial-Process Estimates as Smoothers, chapter 13, pages 393–424.**
Wiley, Chichester, UK, 2012.

H. Rue, S. Martino, and N. Chopin.
**Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations.**
*Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392, 2009.

D. Simpson, J. B. Illian, F. Lindgren, S. H. Srbye, and H. Rue.
**Going off grid: computationally efficient inference for log-Gaussian Cox processes.**
*Biometrika*, 103(1):49–70, 2016.

D. Simpson, F. Lindgren, and H. Rue.
**In order to make spatial statistics computationally feasible, we need to forget about the covariance function.**
*Environmetrics*, 23(1):65–74, 2012.

D. Simpson, F. Lindgren, and H. Rue.
**Think continuous: Markovian Gaussian models in spatial statistics.**
*Spatial Statistics*, 1:16–29, 2012.

G. Wahba.
**Spline models for observational data.**
SIAM, Philadephia, PA, USA, 1990.

S. N. Wood.
**Generalized Additive Models. An Introduction with R.**
Texts in Statistical Science. CRC Press, Boca Raton, FL, USA, 2nd edition, 2017.

Y. Yuan, F. E. Bachl, F. Lindgren, D. L. Borchers, J. B. Illian, S. T. Buckland, H. Rue, T. Gerrodette, et al.
**Point process models for spatio-temporal distance sampling data from a large-scale survey of blue whales.**
*The Annals of Applied Statistics*, 11(4):2270–2297, 2017.

Y. R. Yue, D. Simpson, F. Lindgren, and H. Rue.
**Bayesian adaptive smoothing splines using stochastic differential equations.**
*Bayesian Analysis*, 9(2):397–424, 2014.