



Aalto University
School of Electrical
Engineering

Stochastic (Partial) Differential Equations and Gaussian Processes

Simo Särkkä

Aalto University, Finland

Contents

- 1 Motivating Applications
- 2 Using SPDE solvers on Gaussian processes
- 3 What do the SPDE methods then look like?
- 4 Discussion and summary

Contents

- 1 Motivating Applications
- 2 Using SPDE solvers on Gaussian processes
- 3 What do the SPDE methods then look like?
- 4 Discussion and summary

Contents

- 1 Motivating Applications
- 2 Using SPDE solvers on Gaussian processes
- 3 What do the SPDE methods then look like?
- 4 Discussion and summary

Contents

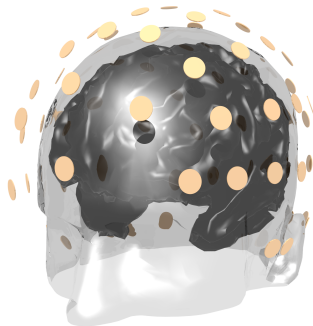
- 1 Motivating Applications
- 2 Using SPDE solvers on Gaussian processes
- 3 What do the SPDE methods then look like?
- 4 Discussion and summary

Contents

- 1 Motivating Applications
- 2 Using SPDE solvers on Gaussian processes
- 3 What do the SPDE methods then look like?
- 4 Discussion and summary

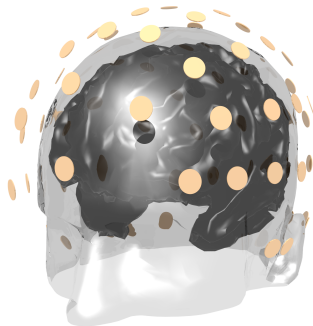
Magneto- and Electroencephalography (MEG & EEG)

- **Unknown:** Time course of amplitudes of dipole sources.
- **Observed:** Electromagnetic field (potential / flux).



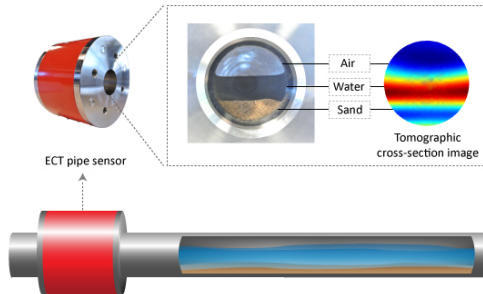
Magneto- and Electroencephalography (MEG & EEG)

- **Unknown:** Time course of amplitudes of dipole sources.
- **Observed:** Electromagnetic field (potential / flux).



Electrical Impedance and Capacitance Tomography (EIT & ECT)

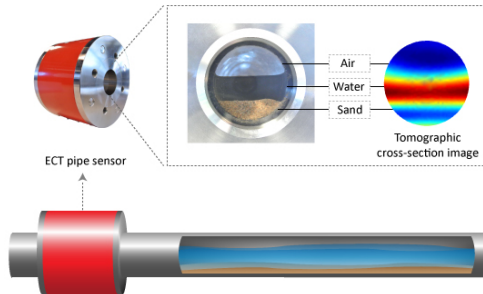
- **Unknown:** Conductivity/permittivity constants.
- **Observed:** Impedances on boundary.



Source: Rocsole Ltd. – www.rocsole.com

Electrical Impedance and Capacitance Tomography (EIT & ECT)

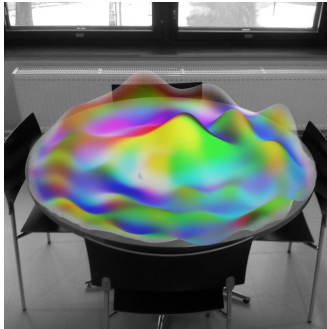
- **Unknown:** Conductivity/permittivity constants.
- **Observed:** Impedances on boundary.



Source: Rocsole Ltd. – www.rocsole.com

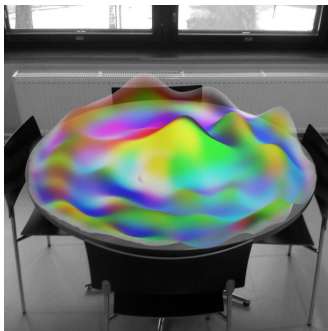
Magnetic field mapping

- **Unknown:** Magnetic field (or actually the corresponding potential).
- **Observed:** Point measurements of the magnetic field.



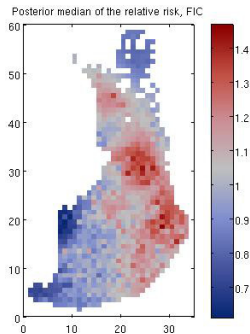
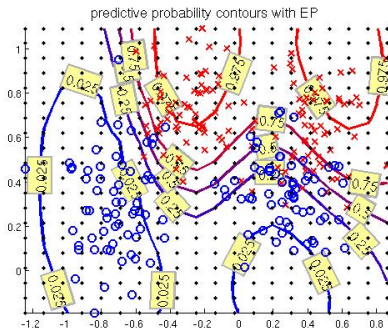
Magnetic field mapping

- **Unknown:** Magnetic field (or actually the corresponding potential).
- **Observed:** Point measurements of the magnetic field.



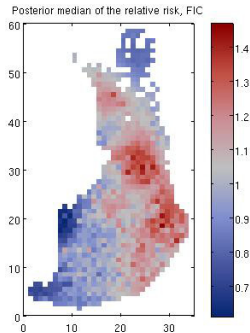
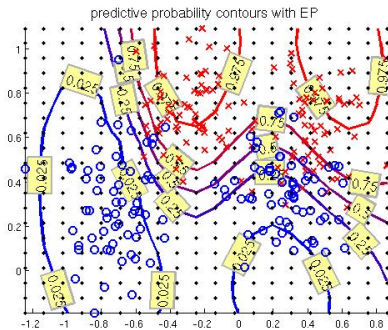
Machine learning and Kriging

- **Unknown:** Class labels or missing spatial values.
- **Observed:** Subset (training set) of class labels, spatial values, and predictors.



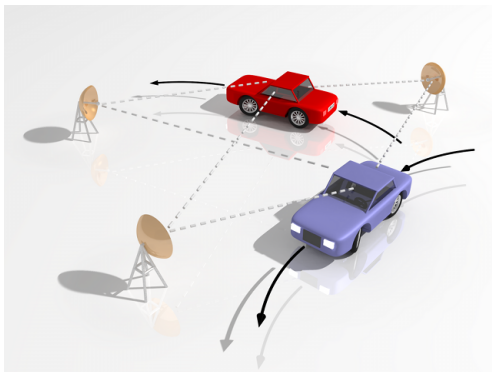
Machine learning and Kriging

- **Unknown:** Class labels or missing spatial values.
- **Observed:** Subset (training set) of class labels, spatial values, and predictors.



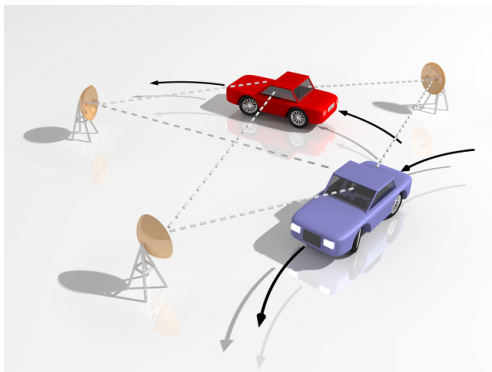
Target Tracking and Location Sensing

- **Unknown:** Target trajectories and velocities.
- **Observed:** Direction or inertial measurements.



Target Tracking and Location Sensing

- **Unknown:** Target trajectories and velocities.
- **Observed:** Direction or inertial measurements.



Contents

- 1 Motivating Applications
- 2 Using SPDE solvers on Gaussian processes
- 3 What do the SPDE methods then look like?
- 4 Discussion and summary

Gaussian process regression models and inverse problems

- A typical statistical inverse problem:

$$f(\mathbf{x}) \sim \text{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$
$$\mathbf{y} = \mathcal{H} f(\mathbf{x}) + \varepsilon$$

- The operator matrix \mathcal{H} encodes physics into the model.
- The mean $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ encode the prior information on $f(\mathbf{x})$.
- In plain Gaussian process regression we have

$$\mathcal{H} f(\mathbf{x}) = \begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_N) \end{pmatrix}$$

Gaussian process regression models and inverse problems

- A typical statistical inverse problem:

$$f(\mathbf{x}) \sim \text{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$
$$\mathbf{y} = \mathcal{H} f(\mathbf{x}) + \varepsilon$$

- The operator matrix \mathcal{H} encodes physics into the model.
- The mean $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ encode the prior information on $f(\mathbf{x})$.
- In plain Gaussian process regression we have

$$\mathcal{H} f(\mathbf{x}) = \begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_N) \end{pmatrix}$$

Gaussian process regression models and inverse problems

- A typical statistical inverse problem:

$$f(\mathbf{x}) \sim \text{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$
$$\mathbf{y} = \mathcal{H} f(\mathbf{x}) + \varepsilon$$

- The operator matrix \mathcal{H} encodes physics into the model.
- The mean $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ encode the prior information on $f(\mathbf{x})$.
- In plain Gaussian process regression we have

$$\mathcal{H} f(\mathbf{x}) = \begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_N) \end{pmatrix}$$

Gaussian process regression models and inverse problems

- A typical statistical inverse problem:

$$f(\mathbf{x}) \sim \text{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$
$$\mathbf{y} = \mathcal{H} f(\mathbf{x}) + \varepsilon$$

- The operator matrix \mathcal{H} encodes physics into the model.
- The mean $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ encode the prior information on $f(\mathbf{x})$.
- In plain Gaussian process regression we have

$$\mathcal{H} f(\mathbf{x}) = \begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_N) \end{pmatrix}$$

Gaussian process regression models and inverse problems (cont.)

- **Temporal** models (models with single input):

$$f(t) \sim \text{GP}(m(t), k(t, t'))$$

$$\mathbf{y} = \mathcal{H} f(t) + \varepsilon$$

- **Spatio-temporal** models:

$$f(\mathbf{x}, t) \sim \text{GP}(m(\mathbf{x}, t), k(\mathbf{x}, t; \mathbf{x}', t'))$$

$$\mathbf{y} = \mathcal{H} f(\mathbf{x}, t) + \varepsilon$$

- All of these can be recasted as **statistical estimation on a stochastic partial differential equation (SPDE) model**.

Gaussian process regression models and inverse problems (cont.)

- **Temporal** models (models with single input):

$$f(t) \sim \text{GP}(m(t), k(t, t'))$$

$$\mathbf{y} = \mathcal{H} f(t) + \varepsilon$$

- **Spatio-temporal** models:

$$f(\mathbf{x}, t) \sim \text{GP}(m(\mathbf{x}, t), k(\mathbf{x}, t; \mathbf{x}', t'))$$

$$\mathbf{y} = \mathcal{H} f(\mathbf{x}, t) + \varepsilon$$

- All of these can be recasted as statistical estimation on a stochastic partial differential equation (SPDE) model.

Gaussian process regression models and inverse problems (cont.)

- **Temporal** models (models with single input):

$$f(t) \sim \text{GP}(m(t), k(t, t'))$$

$$\mathbf{y} = \mathcal{H} f(t) + \varepsilon$$

- **Spatio-temporal** models:

$$f(\mathbf{x}, t) \sim \text{GP}(m(\mathbf{x}, t), k(\mathbf{x}, t; \mathbf{x}', t'))$$

$$\mathbf{y} = \mathcal{H} f(\mathbf{x}, t) + \varepsilon$$

- All of these can be recasted as **statistical estimation on a stochastic partial differential equation (SPDE) model**.

Why use S(P)DE solvers for GPs?

- The $O(n^3)$ computational complexity is always a challenge.
- Latent force models combine PDE/ODEs with GPs.
- What do we get:
 - Sparse approximations developed for SPDEs.
 - Reduced rank Fourier/basis function approximations.
 - The use of Markov properties and Markov approximations.
 - State-space methods for SDEs/SPDEs.
 - Path to non-Gaussian processes.
- Downsides:
 - Approximations of non-parametric models with parametric models.
 - Approximations of a non-Markovian models as Markovian.
 - Mathematics can become messy.

Why use S(P)DE solvers for GPs?

- The $O(n^3)$ computational complexity is always a challenge.
- Latent force models combine PDE/ODEs with GPs.
- What do we get:
 - Sparse approximations developed for SPDEs.
 - Reduced rank Fourier/basis function approximations.
 - The use of Markov properties and Markov approximations.
 - State-space methods for SDEs/SPDEs.
 - Path to non-Gaussian processes.
- Downsides:
 - Approximations of non-parametric models with parametric models.
 - Approximations of a non-Markovian models as Markovian.
 - Mathematics can become messy.

Why use S(P)DE solvers for GPs?

- The $O(n^3)$ computational complexity is always a challenge.
- Latent force models combine PDE/ODEs with GPs.
- What do we get:
 - Sparse approximations developed for SPDEs.
 - Reduced rank Fourier/basis function approximations.
 - The use of Markov properties and Markov approximations.
 - State-space methods for SDEs/SPDEs.
 - Path to non-Gaussian processes.
- Downsides:
 - Approximations of non-parametric models with parametric models.
 - Approximations of a non-Markovian models as Markovian.
 - Mathematics can become messy.

Why use S(P)DE solvers for GPs?

- The $O(n^3)$ computational complexity is always a challenge.
- Latent force models combine PDE/ODEs with GPs.
- What do we get:
 - Sparse approximations developed for SPDEs.
 - Reduced rank Fourier/basis function approximations.
 - The use of Markov properties and Markov approximations.
 - State-space methods for SDEs/SPDEs.
 - Path to non-Gaussian processes.
- Downsides:
 - Approximations of non-parametric models with parametric models.
 - Approximations of a non-Markovian models as Markovian.
 - Mathematics can become messy.

Why use S(P)DE solvers for GPs?

- The $O(n^3)$ computational complexity is always a challenge.
- Latent force models combine PDE/ODEs with GPs.
- What do we get:
 - Sparse approximations developed for SPDEs.
 - Reduced rank Fourier/basis function approximations.
 - The use of Markov properties and Markov approximations.
 - State-space methods for SDEs/SPDEs.
 - Path to non-Gaussian processes.
- Downsides:
 - Approximations of non-parametric models with parametric models.
 - Approximations of a non-Markovian models as Markovian.
 - Mathematics can become messy.

Why use S(P)DE solvers for GPs?

- The $O(n^3)$ computational complexity is always a challenge.
- Latent force models combine PDE/ODEs with GPs.
- What do we get:
 - Sparse approximations developed for SPDEs.
 - Reduced rank Fourier/basis function approximations.
 - The use of Markov properties and Markov approximations.
 - State-space methods for SDEs/SPDEs.
 - Path to non-Gaussian processes.
- Downsides:
 - Approximations of non-parametric models with parametric models.
 - Approximations of a non-Markovian models as Markovian.
 - Mathematics can become messy.

Why use S(P)DE solvers for GPs?

- The $O(n^3)$ computational complexity is always a challenge.
- Latent force models combine PDE/ODEs with GPs.
- What do we get:
 - Sparse approximations developed for SPDEs.
 - Reduced rank Fourier/basis function approximations.
 - The use of Markov properties and Markov approximations.
 - State-space methods for SDEs/SPDEs.
 - Path to non-Gaussian processes.
- Downsides:
 - Approximations of non-parametric models with parametric models.
 - Approximations of a non-Markovian models as Markovian.
 - Mathematics can become messy.

Why use S(P)DE solvers for GPs?

- The $O(n^3)$ computational complexity is always a challenge.
- Latent force models combine PDE/ODEs with GPs.
- What do we get:
 - Sparse approximations developed for SPDEs.
 - Reduced rank Fourier/basis function approximations.
 - The use of Markov properties and Markov approximations.
 - State-space methods for SDEs/SPDEs.
 - Path to non-Gaussian processes.
- Downsides:
 - Approximations of non-parametric models with parametric models.
 - Approximations of a non-Markovian models as Markovian.
 - Mathematics can become messy.

Why use S(P)DE solvers for GPs?

- The $O(n^3)$ computational complexity is always a challenge.
- Latent force models combine PDE/ODEs with GPs.
- What do we get:
 - Sparse approximations developed for SPDEs.
 - Reduced rank Fourier/basis function approximations.
 - The use of Markov properties and Markov approximations.
 - State-space methods for SDEs/SPDEs.
 - Path to non-Gaussian processes.
- Downsides:
 - Approximations of non-parametric models with parametric models.
 - Approximations of a non-Markovian models as Markovian.
 - Mathematics can become messy.

Why use S(P)DE solvers for GPs?

- The $O(n^3)$ computational complexity is always a challenge.
- Latent force models combine PDE/ODEs with GPs.
- What do we get:
 - Sparse approximations developed for SPDEs.
 - Reduced rank Fourier/basis function approximations.
 - The use of Markov properties and Markov approximations.
 - State-space methods for SDEs/SPDEs.
 - Path to non-Gaussian processes.
- Downsides:
 - Approximations of non-parametric models with parametric models.
 - Approximations of a non-Markovian models as Markovian.
 - Mathematics can become messy.

Why use S(P)DE solvers for GPs?

- The $O(n^3)$ computational complexity is always a challenge.
- Latent force models combine PDE/ODEs with GPs.
- What do we get:
 - Sparse approximations developed for SPDEs.
 - Reduced rank Fourier/basis function approximations.
 - The use of Markov properties and Markov approximations.
 - State-space methods for SDEs/SPDEs.
 - Path to non-Gaussian processes.
- Downsides:
 - Approximations of non-parametric models with parametric models.
 - Approximations of a non-Markovian models as Markovian.
 - Mathematics can become messy.

Why use S(P)DE solvers for GPs?

- The $O(n^3)$ computational complexity is always a challenge.
- Latent force models combine PDE/ODEs with GPs.
- What do we get:
 - Sparse approximations developed for SPDEs.
 - Reduced rank Fourier/basis function approximations.
 - The use of Markov properties and Markov approximations.
 - State-space methods for SDEs/SPDEs.
 - Path to non-Gaussian processes.
- Downsides:
 - Approximations of non-parametric models with parametric models.
 - Approximations of a non-Markovian models as Markovian.
 - Mathematics can become messy.

Kernel vs. SPDE representations of GPs

GP model $\mathbf{x} \in \mathbb{R}^d, t \in \mathbb{R}$	Equivalent Static SPDE model
Homogenous $k(\mathbf{x}, \mathbf{x}')$	SPDE model $\mathcal{L} f(\mathbf{x}) = w(\mathbf{x})$
Stationary $k(t, t')$	State-space/Itô-SDE model $d\mathbf{f}(t) = \mathbf{A} \mathbf{f}(t) dt + \mathbf{L} dW(t)$
Homogenous/stationary $k(\mathbf{x}, t; \mathbf{x}', t')$	Stochastic evolution equation $\partial_t \mathbf{f}(\mathbf{x}, t) = \mathcal{A}_x \mathbf{f}(\mathbf{x}, t) dt + \mathbf{L} dW(\mathbf{x}, t)$

Contents

- 1 Motivating Applications
- 2 Using SPDE solvers on Gaussian processes
- 3 What do the SPDE methods then look like?
- 4 Discussion and summary

Basic idea of SPDE inference on GPs [1/2]

- Consider e.g. the stochastic partial differential equation:

$$\frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} - \lambda^2 f(x, y) = w(x, y)$$

- Fourier transforming gives the spectral density:

$$S(\omega_x, \omega_y) \propto (\lambda^2 + \omega_x^2 + \omega_y^2)^{-2}.$$

- Inverse Fourier transform gives the covariance function:

$$k(x, y; x', y') = \frac{\sqrt{(x - x')^2 + (y - y')^2}}{2\lambda} K_1(\lambda \sqrt{(x - x')^2 + (y - y')^2})$$

- But this is just the Matérn covariance function.
- The corresponding RKHS is a Sobolev space.

Basic idea of SPDE inference on GPs [1/2]

- Consider e.g. the stochastic partial differential equation:

$$\frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} - \lambda^2 f(x, y) = w(x, y)$$

- Fourier transforming gives the spectral density:

$$S(\omega_x, \omega_y) \propto \left(\lambda^2 + \omega_x^2 + \omega_y^2 \right)^{-2}.$$

- Inverse Fourier transform gives the covariance function:

$$k(x, y; x', y') = \frac{\sqrt{(x - x')^2 + (y - y')^2}}{2\lambda} K_1(\lambda \sqrt{(x - x')^2 + (y - y')^2})$$

- But this is just the Matérn covariance function.
- The corresponding RKHS is a Sobolev space.

Basic idea of SPDE inference on GPs [1/2]

- Consider e.g. the **stochastic partial differential equation**:

$$\frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} - \lambda^2 f(x, y) = w(x, y)$$

- Fourier transforming gives the **spectral density**:

$$S(\omega_x, \omega_y) \propto \left(\lambda^2 + \omega_x^2 + \omega_y^2 \right)^{-2}.$$

- Inverse Fourier transform gives the **covariance function**:

$$k(x, y; x', y') = \frac{\sqrt{(x - x')^2 + (y - y')^2}}{2\lambda} K_1(\lambda \sqrt{(x - x')^2 + (y - y')^2})$$

- But this is just the **Matérn covariance function**.
- The corresponding **RKHS** is a **Sobolev space**.

Basic idea of SPDE inference on GPs [1/2]

- Consider e.g. the stochastic partial differential equation:

$$\frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} - \lambda^2 f(x, y) = w(x, y)$$

- Fourier transforming gives the spectral density:

$$S(\omega_x, \omega_y) \propto \left(\lambda^2 + \omega_x^2 + \omega_y^2 \right)^{-2}.$$

- Inverse Fourier transform gives the covariance function:

$$k(x, y; x', y') = \frac{\sqrt{(x - x')^2 + (y - y')^2}}{2\lambda} K_1(\lambda \sqrt{(x - x')^2 + (y - y')^2})$$

- But this is just the Matérn covariance function.
- The corresponding RKHS is a Sobolev space.

Basic idea of SPDE inference on GPs [1/2]

- Consider e.g. the **stochastic partial differential equation**:

$$\frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} - \lambda^2 f(x, y) = w(x, y)$$

- Fourier transforming gives the **spectral density**:

$$S(\omega_x, \omega_y) \propto \left(\lambda^2 + \omega_x^2 + \omega_y^2 \right)^{-2}.$$

- Inverse Fourier transform gives the **covariance function**:

$$k(x, y; x', y') = \frac{\sqrt{(x - x')^2 + (y - y')^2}}{2\lambda} K_1(\lambda \sqrt{(x - x')^2 + (y - y')^2})$$

- But this is just the **Matérn covariance function**.
- The corresponding **RKHS** is a **Sobolev space**.

Basic idea of SPDE inference on GPs [2/2]

- More generally, SPDE for some linear operator \mathcal{L} :

$$\mathcal{L} f(\mathbf{x}) = w(\mathbf{x})$$

- Now f is a GP with precision and covariance operators:

$$\mathcal{K}^{-1} = \mathcal{L}^* \mathcal{L}$$

$$\mathcal{K} = (\mathcal{L}^* \mathcal{L})^{-1}$$

- Idea: approximate \mathcal{L} or \mathcal{L}^{-1} using PDE/ODE methods:

- Finite-differences/FEM methods lead to sparse precision approximations.
- Fourier/basis-function methods lead to reduced rank covariance approximations.
- Spectral factorization leads to state-space (Kalman) methods which are time-recursive (or sparse in precision).

Basic idea of SPDE inference on GPs [2/2]

- More generally, SPDE for some linear operator \mathcal{L} :

$$\mathcal{L} f(\mathbf{x}) = w(\mathbf{x})$$

- Now f is a GP with precision and covariance operators:

$$\mathcal{K}^{-1} = \mathcal{L}^* \mathcal{L}$$

$$\mathcal{K} = (\mathcal{L}^* \mathcal{L})^{-1}$$

- Idea: approximate \mathcal{L} or \mathcal{L}^{-1} using PDE/ODE methods:

- Finite-differences/FEM methods lead to sparse precision approximations.
- Fourier/basis-function methods lead to reduced rank covariance approximations.
- Spectral factorization leads to state-space (Kalman) methods which are time-recursive (or sparse in precision).

Basic idea of SPDE inference on GPs [2/2]

- More generally, SPDE for some linear operator \mathcal{L} :

$$\mathcal{L} f(\mathbf{x}) = w(\mathbf{x})$$

- Now f is a GP with precision and covariance operators:

$$\mathcal{K}^{-1} = \mathcal{L}^* \mathcal{L}$$

$$\mathcal{K} = (\mathcal{L}^* \mathcal{L})^{-1}$$

- Idea: approximate \mathcal{L} or \mathcal{L}^{-1} using PDE/ODE methods:

- 1 Finite-differences/FEM methods lead to sparse precision approximations.
- 2 Fourier/basis-function methods lead to reduced rank covariance approximations.
- 3 Spectral factorization leads to state-space (Kalman) methods which are time-recursive (or sparse in precision).

Basic idea of SPDE inference on GPs [2/2]

- More generally, SPDE for some linear operator \mathcal{L} :

$$\mathcal{L} f(\mathbf{x}) = w(\mathbf{x})$$

- Now f is a GP with precision and covariance operators:

$$\mathcal{K}^{-1} = \mathcal{L}^* \mathcal{L}$$

$$\mathcal{K} = (\mathcal{L}^* \mathcal{L})^{-1}$$

- Idea: approximate \mathcal{L} or \mathcal{L}^{-1} using PDE/ODE methods:

- 1 Finite-differences/FEM methods lead to sparse precision approximations.
- 2 Fourier/basis-function methods lead to reduced rank covariance approximations.
- 3 Spectral factorization leads to state-space (Kalman) methods which are time-recursive (or sparse in precision).

Basic idea of SPDE inference on GPs [2/2]

- More generally, **SPDE** for some linear operator \mathcal{L} :

$$\mathcal{L} f(\mathbf{x}) = w(\mathbf{x})$$

- Now f is a GP with **precision and covariance operators**:

$$\mathcal{K}^{-1} = \mathcal{L}^* \mathcal{L}$$

$$\mathcal{K} = (\mathcal{L}^* \mathcal{L})^{-1}$$

- **Idea**: approximate \mathcal{L} or \mathcal{L}^{-1} using PDE/ODE methods:

- 1 **Finite-differences/FEM** methods lead to **sparse precision approximations**.
- 2 **Fourier/basis-function** methods lead to **reduced rank covariance approximations**.
- 3 **Spectral factorization** leads to state-space (Kalman) methods which are **time-recursive** (or sparse in precision).

Basic idea of SPDE inference on GPs [2/2]

- More generally, **SPDE** for some linear operator \mathcal{L} :

$$\mathcal{L} f(\mathbf{x}) = w(\mathbf{x})$$

- Now f is a GP with **precision and covariance operators**:

$$\mathcal{K}^{-1} = \mathcal{L}^* \mathcal{L}$$

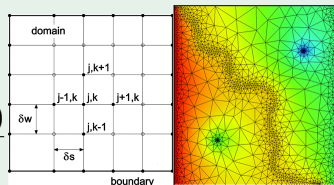
$$\mathcal{K} = (\mathcal{L}^* \mathcal{L})^{-1}$$

- **Idea**: approximate \mathcal{L} or \mathcal{L}^{-1} using PDE/ODE methods:
 - 1 **Finite-differences/FEM** methods lead to **sparse precision approximations**.
 - 2 **Fourier/basis-function** methods lead to **reduced rank covariance approximations**.
 - 3 **Spectral factorization** leads to state-space (Kalman) methods which are **time-recursive** (or sparse in precision).

Finite-differences/FEM – sparse precision

- Basic idea:

$$\frac{\partial f(x)}{\partial x} \approx \frac{f(x+h) - f(x)}{h}$$
$$\frac{\partial^2 f(x)}{\partial x^2} \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$



- We get an SPDE approximation $\mathcal{L} \approx \mathbf{L}$, where \mathbf{L} is sparse
- The precision operator approximation is then sparse:

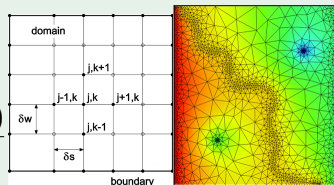
$$\mathcal{K}^{-1} \approx \mathbf{L}^T \mathbf{L} = \text{sparse}$$

- \mathcal{L} need to be approximated as integro-differential operator.
- Requires formation of a grid, but parallelizes well.

Finite-differences/FEM – sparse precision

- Basic idea:

$$\frac{\partial f(x)}{\partial x} \approx \frac{f(x+h) - f(x)}{h}$$
$$\frac{\partial^2 f(x)}{\partial x^2} \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$



- We get an SPDE approximation $\mathcal{L} \approx \mathbf{L}$, where \mathbf{L} is **sparse**
- The **precision operator** approximation is then **sparse**:

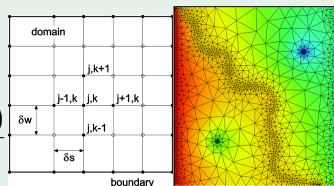
$$\mathcal{K}^{-1} \approx \mathbf{L}^T \mathbf{L} = \text{sparse}$$

- \mathcal{L} need to be approximated as **integro-differential operator**.
- Requires formation of a **grid**, but parallelizes well.

Finite-differences/FEM – sparse precision

- Basic idea:

$$\frac{\partial f(x)}{\partial x} \approx \frac{f(x+h) - f(x)}{h}$$
$$\frac{\partial^2 f(x)}{\partial x^2} \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$



- We get an SPDE approximation $\mathcal{L} \approx \mathbf{L}$, where \mathbf{L} is **sparse**
- The **precision operator** approximation is then **sparse**:

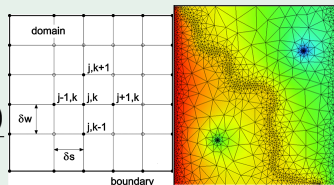
$$\mathcal{K}^{-1} \approx \mathbf{L}^T \mathbf{L} = \text{sparse}$$

- \mathcal{L} need to be approximated as **integro-differential operator**.
- Requires formation of a **grid**, but parallelizes well.

Finite-differences/FEM – sparse precision

- Basic idea:

$$\frac{\partial f(x)}{\partial x} \approx \frac{f(x+h) - f(x)}{h}$$
$$\frac{\partial^2 f(x)}{\partial x^2} \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$



- We get an SPDE approximation $\mathcal{L} \approx \mathbf{L}$, where \mathbf{L} is **sparse**
- The **precision operator** approximation is then **sparse**:

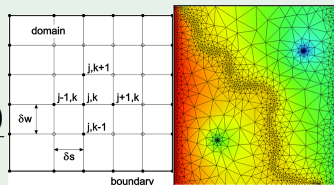
$$\mathcal{K}^{-1} \approx \mathbf{L}^T \mathbf{L} = \text{sparse}$$

- \mathcal{L} need to be approximated as **integro-differential operator**.
- Requires formation of a **grid**, but parallelizes well.

Finite-differences/FEM – sparse precision

- Basic idea:

$$\frac{\partial f(x)}{\partial x} \approx \frac{f(x+h) - f(x)}{h}$$
$$\frac{\partial^2 f(x)}{\partial x^2} \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$



- We get an SPDE approximation $\mathcal{L} \approx \mathbf{L}$, where \mathbf{L} is **sparse**
- The **precision operator** approximation is then **sparse**:

$$\mathcal{K}^{-1} \approx \mathbf{L}^T \mathbf{L} = \text{sparse}$$

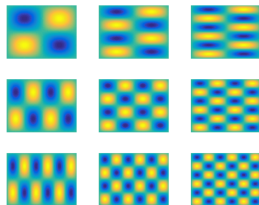
- \mathcal{L} need to be approximated as **integro-differential operator**.
- Requires formation of a **grid**, but parallelizes well.

Classical and random Fourier methods – reduced rank approximations and FFT

- Approximation:

$$f(\mathbf{x}) \approx \sum_{\mathbf{k} \in \mathbb{N}^d} \alpha_{\mathbf{k}} \exp(2\pi i \mathbf{k}^T \mathbf{x})$$

$$\alpha_{\mathbf{k}} \sim \text{Gaussian}$$



- We use less coefficients $\alpha_{\mathbf{k}}$ than the number of data points.
- Leads to reduced-rank covariance approximations

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{|\mathbf{k}| \leq N} \sigma_{\mathbf{k}}^2 \exp(2\pi i \mathbf{k}^T \mathbf{x}) \exp(2\pi i \mathbf{k}^T \mathbf{x}')^*$$

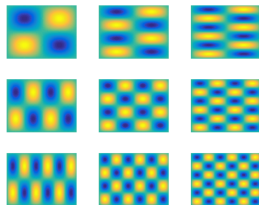
- Truncated series, random frequencies, FFT, ...

Classical and random Fourier methods – reduced rank approximations and FFT

- Approximation:

$$f(\mathbf{x}) \approx \sum_{\mathbf{k} \in \mathbb{N}^d} \alpha_{\mathbf{k}} \exp(2\pi i \mathbf{k}^T \mathbf{x})$$

$$\alpha_{\mathbf{k}} \sim \text{Gaussian}$$



- We use **less coefficients** $\alpha_{\mathbf{k}}$ than the **number of data points**.
- Leads to **reduced-rank covariance approximations**

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{|\mathbf{k}| \leq N} \sigma_{\mathbf{k}}^2 \exp(2\pi i \mathbf{k}^T \mathbf{x}) \exp(2\pi i \mathbf{k}^T \mathbf{x}')^*$$

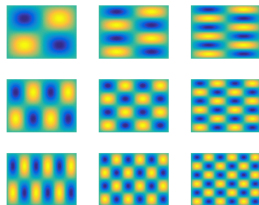
- Truncated series, random frequencies, FFT, ...

Classical and random Fourier methods – reduced rank approximations and FFT

- Approximation:

$$f(\mathbf{x}) \approx \sum_{\mathbf{k} \in \mathbb{N}^d} \alpha_{\mathbf{k}} \exp(2\pi i \mathbf{k}^T \mathbf{x})$$

$$\alpha_{\mathbf{k}} \sim \text{Gaussian}$$



- We use **less coefficients** $\alpha_{\mathbf{k}}$ than the **number of data points**.
- Leads to **reduced-rank covariance approximations**

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{|\mathbf{k}| \leq N} \sigma_{\mathbf{k}}^2 \exp(2\pi i \mathbf{k}^T \mathbf{x}) \exp(2\pi i \mathbf{k}^T \mathbf{x}')^*$$

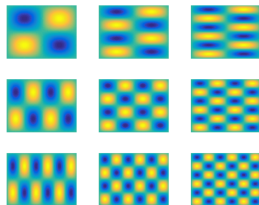
- Truncated series, random frequencies, FFT, ...

Classical and random Fourier methods – reduced rank approximations and FFT

- Approximation:

$$f(\mathbf{x}) \approx \sum_{\mathbf{k} \in \mathbb{N}^d} \alpha_{\mathbf{k}} \exp(2\pi i \mathbf{k}^T \mathbf{x})$$

$$\alpha_{\mathbf{k}} \sim \text{Gaussian}$$



- We use **less coefficients** $\alpha_{\mathbf{k}}$ than the **number of data points**.
- Leads to **reduced-rank covariance approximations**

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{|\mathbf{k}| \leq N} \sigma_{\mathbf{k}}^2 \exp(2\pi i \mathbf{k}^T \mathbf{x}) \exp(2\pi i \mathbf{k}^T \mathbf{x}')^*$$

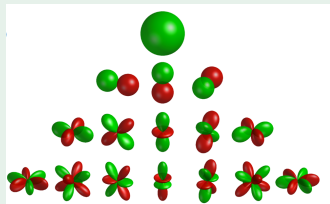
- Truncated series, random frequencies, FFT, ...

Hilbert-space/Galerkin methods – reduced rank approximations

- Approximation:

$$f(\mathbf{x}) \approx \sum_i c_i \phi_i(\mathbf{x})$$

$$\langle \phi_i, \phi_j \rangle_H \approx \delta_{ij}, \text{ e.g. } \nabla^2 \phi_i = -\lambda_i \phi_i$$



- Again, use less coefficients than the number of data points.
- Reduced-rank covariance approximations such as

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{i=1}^N \sigma_i^2 \phi_i(\mathbf{x}) \phi_i(\mathbf{x}').$$

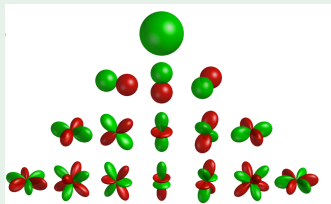
- Wavelets, Galerkin, finite elements, ...

Hilbert-space/Galerkin methods – reduced rank approximations

- Approximation:

$$f(\mathbf{x}) \approx \sum_i c_i \phi_i(\mathbf{x})$$

$$\langle \phi_i, \phi_j \rangle_H \approx \delta_{ij}, \text{ e.g. } \nabla^2 \phi_i = -\lambda_i \phi_i$$



- Again, use **less coefficients** than the **number of data points**.
- Reduced-rank covariance approximations such as

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{i=1}^N \sigma_i^2 \phi_i(\mathbf{x}) \phi_i(\mathbf{x}').$$

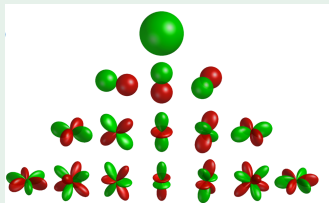
- Wavelets, Galerkin, finite elements, ...

Hilbert-space/Galerkin methods – reduced rank approximations

- Approximation:

$$f(\mathbf{x}) \approx \sum_i c_i \phi_i(\mathbf{x})$$

$$\langle \phi_i, \phi_j \rangle_H \approx \delta_{ij}, \text{ e.g. } \nabla^2 \phi_i = -\lambda_i \phi_i$$



- Again, use **less coefficients** than the **number of data points**.
- **Reduced-rank covariance approximations** such as

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{i=1}^N \sigma_i^2 \phi_i(\mathbf{x}) \phi_i(\mathbf{x}').$$

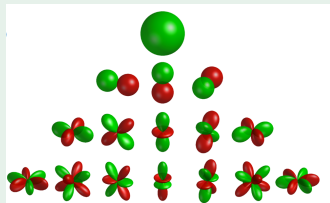
- Wavelets, Galerkin, finite elements, ...

Hilbert-space/Galerkin methods – reduced rank approximations

- Approximation:

$$f(\mathbf{x}) \approx \sum_i c_i \phi_i(\mathbf{x})$$

$$\langle \phi_i, \phi_j \rangle_H \approx \delta_{ij}, \text{ e.g. } \nabla^2 \phi_i = -\lambda_i \phi_i$$



- Again, use **less coefficients** than the **number of data points**.
- **Reduced-rank covariance approximations** such as

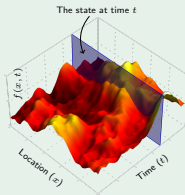
$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{i=1}^N \sigma_i^2 \phi_i(\mathbf{x}) \phi_i(\mathbf{x}').$$

- Wavelets, Galerkin, finite elements, ...

State-space methods – Kalman filters and sparse precision

- Approximation:

$$S(\omega) \approx \frac{b_0 + b_1 \omega^2 + \dots + b_M \omega^{2M}}{a_0 + a_1 \omega^2 + \dots + a_N \omega^{2N}}$$



- Results in a linear stochastic differential equation (SDE)

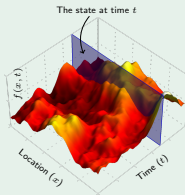
$$d\mathbf{f}(t) = \mathbf{A} \mathbf{f}(t) dt + \mathbf{L} d\mathbf{W}$$

- More generally stochastic evolution equations.
- $O(n)$ GP regression with Kalman filters and smoothers.
- Parallel block-sparse precision methods $\rightarrow O(\log n)$.

State-space methods – Kalman filters and sparse precision

- Approximation:

$$S(\omega) \approx \frac{b_0 + b_1 \omega^2 + \dots + b_M \omega^{2M}}{a_0 + a_1 \omega^2 + \dots + a_N \omega^{2N}}$$



- Results in a linear stochastic differential equation (SDE)

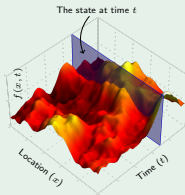
$$d\mathbf{f}(t) = \mathbf{A} \mathbf{f}(t) dt + \mathbf{L} d\mathbf{W}$$

- More generally stochastic evolution equations.
- $O(n)$ GP regression with Kalman filters and smoothers.
- Parallel block-sparse precision methods $\rightarrow O(\log n)$.

State-space methods – Kalman filters and sparse precision

- Approximation:

$$S(\omega) \approx \frac{b_0 + b_1 \omega^2 + \dots + b_M \omega^{2M}}{a_0 + a_1 \omega^2 + \dots + a_N \omega^{2N}}$$



- Results in a linear stochastic differential equation (SDE)

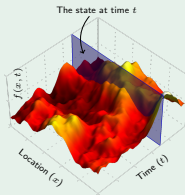
$$d\mathbf{f}(t) = \mathbf{A} \mathbf{f}(t) dt + \mathbf{L} d\mathbf{W}$$

- More generally stochastic evolution equations.
- $O(n)$ GP regression with Kalman filters and smoothers.
- Parallel block-sparse precision methods $\rightarrow O(\log n)$.

State-space methods – Kalman filters and sparse precision

- Approximation:

$$S(\omega) \approx \frac{b_0 + b_1 \omega^2 + \dots + b_M \omega^{2M}}{a_0 + a_1 \omega^2 + \dots + a_N \omega^{2N}}$$



- Results in a linear stochastic differential equation (SDE)

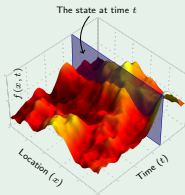
$$d\mathbf{f}(t) = \mathbf{A} \mathbf{f}(t) dt + \mathbf{L} d\mathbf{W}$$

- More generally stochastic evolution equations.
- $O(n)$ GP regression with Kalman filters and smoothers.
- Parallel block-sparse precision methods $\rightarrow O(\log n)$.

State-space methods – Kalman filters and sparse precision

- Approximation:

$$S(\omega) \approx \frac{b_0 + b_1 \omega^2 + \dots + b_M \omega^{2M}}{a_0 + a_1 \omega^2 + \dots + a_N \omega^{2N}}$$



- Results in a linear stochastic differential equation (SDE)

$$d\mathbf{f}(t) = \mathbf{A} \mathbf{f}(t) dt + \mathbf{L} d\mathbf{W}$$

- More generally stochastic evolution equations.
- $O(n)$ GP regression with Kalman filters and smoothers.
- Parallel block-sparse precision methods $\rightarrow O(\log n)$.

State-space methods – Kalman filters and sparse precision (cont.)

Example (Matérn class 1d)

The Matérn class of covariance functions is

$$k(t, t') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\ell} |t - t'| \right)^\nu K_\nu \left(\frac{\sqrt{2\nu}}{\ell} |t - t'| \right).$$

When, e.g., $\nu = 3/2$, we have

$$\begin{aligned} d\mathbf{f}(t) &= \begin{pmatrix} 0 & 1 \\ -\lambda^2 & -2\lambda \end{pmatrix} \mathbf{f}(t) dt + \begin{pmatrix} 0 \\ q^{1/2} \end{pmatrix} dW(t), \\ f(t) &= (1 \quad 0) \mathbf{f}(t). \end{aligned}$$

State-space methods – Kalman filters and sparse precision (cont.)

Example (2D Matérn covariance function)

- Consider a space-time Matérn covariance function

$$k(x, t; x', t') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{\rho}{l} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{\rho}{l} \right).$$

where we have $\rho = \sqrt{(t - t')^2 + (x - x')^2}$, $\nu = 1$ and $d = 2$.

- We get the following representation:

$$d\mathbf{f}(x, t) = \begin{pmatrix} 0 & 1 \\ \frac{\partial^2}{\partial x^2} - \lambda^2 & -2\sqrt{\lambda^2 - \frac{\partial^2}{\partial x^2}} \end{pmatrix} \mathbf{f}(x, t) dt + \begin{pmatrix} 0 \\ 1 \end{pmatrix} dW(x, t).$$

State-space methods – Kalman filters and sparse precision (cont.)

Example (2D Matérn covariance function)

- Consider a space-time Matérn covariance function

$$k(x, t; x', t') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{\rho}{l} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{\rho}{l} \right).$$

where we have $\rho = \sqrt{(t - t')^2 + (x - x')^2}$, $\nu = 1$ and $d = 2$.

- We get the following representation:

$$d\mathbf{f}(x, t) = \begin{pmatrix} 0 & 1 \\ \frac{\partial^2}{\partial x^2} - \lambda^2 & -2\sqrt{\lambda^2 - \frac{\partial^2}{\partial x^2}} \end{pmatrix} \mathbf{f}(x, t) dt + \begin{pmatrix} 0 \\ 1 \end{pmatrix} dW(x, t).$$

Contents

- 1 Motivating Applications
- 2 Using SPDE solvers on Gaussian processes
- 3 What do the SPDE methods then look like?
- 4 Discussion and summary

What then?

- We can **exchange approximations** between the fields.
- **Inference** on the basis functions/point-locations/etc.
- **Non-Gaussian** processes, non-Gaussian likelihoods.
- **Hierarchical** (deep) **SPDE models**.
- **Combined first-principles** and nonparametric models – latent force models (LFM).
- **State-space stochastic control** in Gaussian processes and LFM.

What then?

- We can **exchange approximations** between the fields.
- **Inference** on the basis functions/point-locations/etc.
- Non-Gaussian processes, non-Gaussian likelihoods.
- Hierarchical (deep) SPDE models.
- Combined first-principles and nonparametric models – latent force models (LFM).
- State-space stochastic control in Gaussian processes and LFM.

What then?

- We can **exchange approximations** between the fields.
- **Inference** on the basis functions/point-locations/etc.
- **Non-Gaussian** processes, non-Gaussian likelihoods.
- Hierarchical (deep) **SPDE models**.
- Combined first-principles and nonparametric models – latent force models (LFM).
- **State-space stochastic control** in Gaussian processes and LFM.

What then?

- We can **exchange approximations** between the fields.
- **Inference** on the basis functions/point-locations/etc.
- **Non-Gaussian** processes, non-Gaussian likelihoods.
- **Hierarchical** (deep) **SPDE models**.
- Combined first-principles and nonparametric models – latent force models (LFM).
- State-space stochastic control in Gaussian processes and LFM.

What then?

- We can **exchange approximations** between the fields.
- **Inference** on the basis functions/point-locations/etc.
- **Non-Gaussian** processes, non-Gaussian likelihoods.
- **Hierarchical** (deep) **SPDE models**.
- **Combined first-principles** and nonparametric models – latent force models (LFM).
- **State-space stochastic control** in Gaussian processes and LFM.

What then?

- We can **exchange approximations** between the fields.
- **Inference** on the basis functions/point-locations/etc.
- **Non-Gaussian** processes, non-Gaussian likelihoods.
- **Hierarchical** (deep) **SPDE models**.
- **Combined first-principles** and nonparametric models – latent force models (LFM).
- **State-space stochastic control** in Gaussian processes and LFMs.

Summary

- Gaussian processes (GPs) can be used as **priors in inverse problems**.
- Applications in e.g. **brain imaging, tomography, positioning, and Kriging**.
- GPs have representations as **solutions to SPDEs**.
- SPDE methods can be used to **speed up GP inference**.
- In temporal models we can use **Kalman/Bayesian filters and smoothers**.
- Opens up new paths towards **non-linear GP models**.

Summary

- Gaussian processes (GPs) can be used as priors in inverse problems.
- Applications in e.g. brain imaging, tomography, positioning, and Kriging.
- GPs have representations as solutions to SPDEs.
- SPDE methods can be used to speed up GP inference.
- In temporal models we can use Kalman/Bayesian filters and smoothers.
- Opens up new paths towards non-linear GP models.

Summary

- Gaussian processes (GPs) can be used as priors in inverse problems.
- Applications in e.g. brain imaging, tomography, positioning, and Kriging.
- GPs have representations as solutions to SPDEs.
- SPDE methods can be used to speed up GP inference.
- In temporal models we can use Kalman/Bayesian filters and smoothers.
- Opens up new paths towards non-linear GP models.

Summary

- Gaussian processes (GPs) can be used as priors in inverse problems.
- Applications in e.g. brain imaging, tomography, positioning, and Kriging.
- GPs have representations as solutions to SPDEs.
- SPDE methods can be used to speed up GP inference.
- In temporal models we can use Kalman/Bayesian filters and smoothers.
- Opens up new paths towards non-linear GP models.

Summary

- Gaussian processes (GPs) can be used as priors in inverse problems.
- Applications in e.g. brain imaging, tomography, positioning, and Kriging.
- GPs have representations as solutions to SPDEs.
- SPDE methods can be used to speed up GP inference.
- In temporal models we can use Kalman/Bayesian filters and smoothers.
- Opens up new paths towards non-linear GP models.

Summary

- Gaussian processes (GPs) can be used as priors in inverse problems.
- Applications in e.g. brain imaging, tomography, positioning, and Kriging.
- GPs have representations as solutions to SPDEs.
- SPDE methods can be used to speed up GP inference.
- In temporal models we can use Kalman/Bayesian filters and smoothers.
- Opens up new paths towards non-linear GP models.