

The Statistical Finite Element Method: A Theoretical Foundation for Digital Twins

Mark Girolami^{1,2}

Probabilistic Numerics Spring School, April 9 2024

¹Department of Engineering
University of Cambridge

²The Alan Turing Institute

Talk outline

1. Digital Twin Modelling: what is it?
2. Examples of Digital Twin Modelling.
3. Finite Elements: The Backbone of Digital Twins.
4. Extending Finite Elements to Handle Model Misspecification.
5. The Statistical Finite Element Method (statFEM).
6. Convergence & Theoretical Guarantees of statFEM.
7. Methodology: Nonlinear problems, high dimensions.
8. Examples in Oceanography, Structural Mechanics, Nonlinear Oscillators.
9. Extensions: Machine Learning methods for unstructured data.
10. Conclusions.

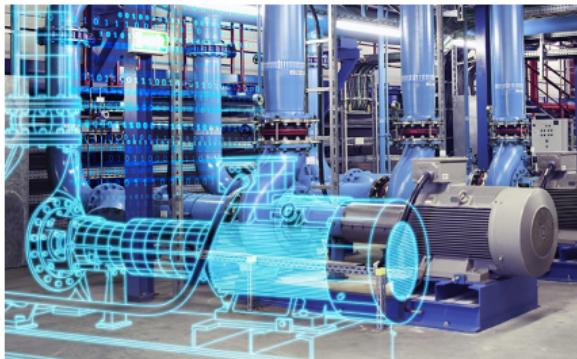
Digital Twins: Not Just Hype

- Digital twins are a powerful step forward, following the computational modelling revolution of the 20th century.
- They allow for us to combine powerful mechanistic descriptions of reality with data, with massive impact across engineering and healthcare (among other fields).
- A lot of attention has been focussed on digital twins in the last decade, but there still remains ambiguity as to what they **are**.
- So what is a digital twin?

Digital Twins: Not Just Hype

- Digital twins are a powerful step forward, following the computational modelling revolution of the 20th century.
- They allow for us to combine powerful mechanistic descriptions of reality with data, with massive impact across engineering and healthcare (among other fields).
- A lot of attention has been focussed on digital twins in the last decade, but there still remains ambiguity as to what they **are**.
- So what is a digital twin?

Digital Twins

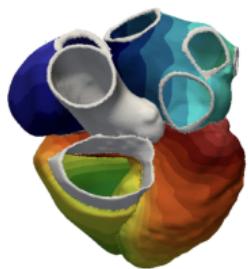


- A digital twin is virtual duplicate of a real system, coupling a mathematical model to observations.
- Simple paradigm common in science and engineering, which is highly versatile and universal.
- However, building a DT in practice is complex and challenging.
- Require knowledge of mechanistic models underlying the system, and the relations of these to data.

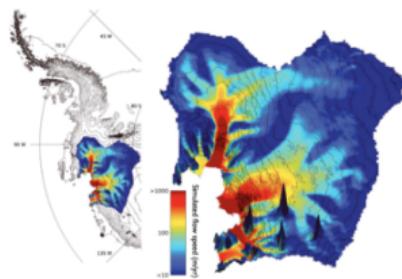
Digital Twins: Key Components

1. Time evolving; ideally over the complete lifecycle of the asset, from design through to end of life.
2. Continued connectivity; data, and information passed between the physical and digital twin continuously.
3. Versatility; enables highly context dependent digital twins: Entirely bespoke to each individual asset considered.
4. Universality; can be applied to all fields of human activity.
5. New functionalities; prediction; learning; management; autonomy.
6. Reduces uncertainty; can quantify the uncertainties present using the digital twin so decisions can more readily be supported.

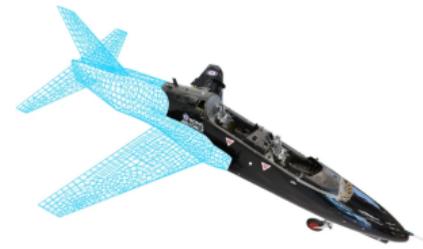
Digital Twins: Application Areas



Health



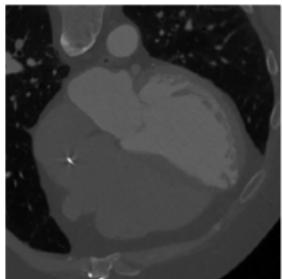
Environment



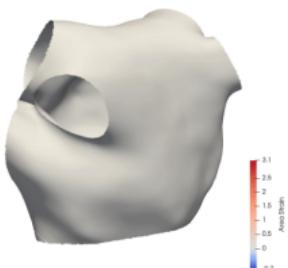
Infrastructure

Digital Twins are **driving innovation** across **health**, **environment**, and **infrastructure** (see, e.g., the Turing Research and Innovation Cluster in Digital Twins).

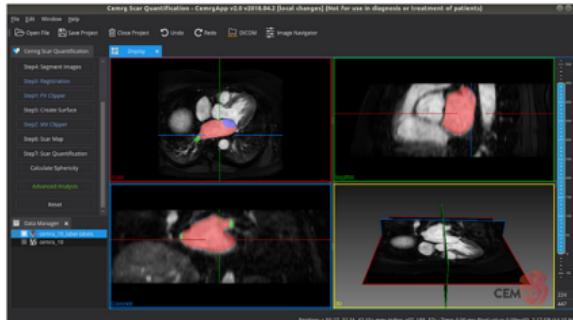
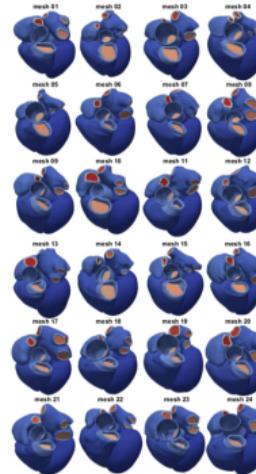
Digital Twins: Health Examples



Data to Model
→



Current Cohorts of 20 hearts



The
Alan Turing
Institute

Digital Twins: Environment Examples in EU



Digital Twins: Infrastructure Examples in UK



DTNet⁺

[Join community](#)

UKRI Digital Twinning NetworkPlus: DTNet+

DTNet+ is a UKRI funded, UK-wide interdisciplinary research network designed to harness the latest advancements that will contribute to the next generation of intelligent, resilient, and trusted digital twins.

[Join community](#)

An illustration showing various digital twin components: a smartphone, a laptop, a cloud icon, a person working on a computer, and a person holding a tablet. There are also icons representing data storage and processing.

DIGITAL TWIN HUB

About Us ▾ News ▾ Resources ▾ Milestones ▾ Get Involved ▾

THE DIGITAL TWIN HUB

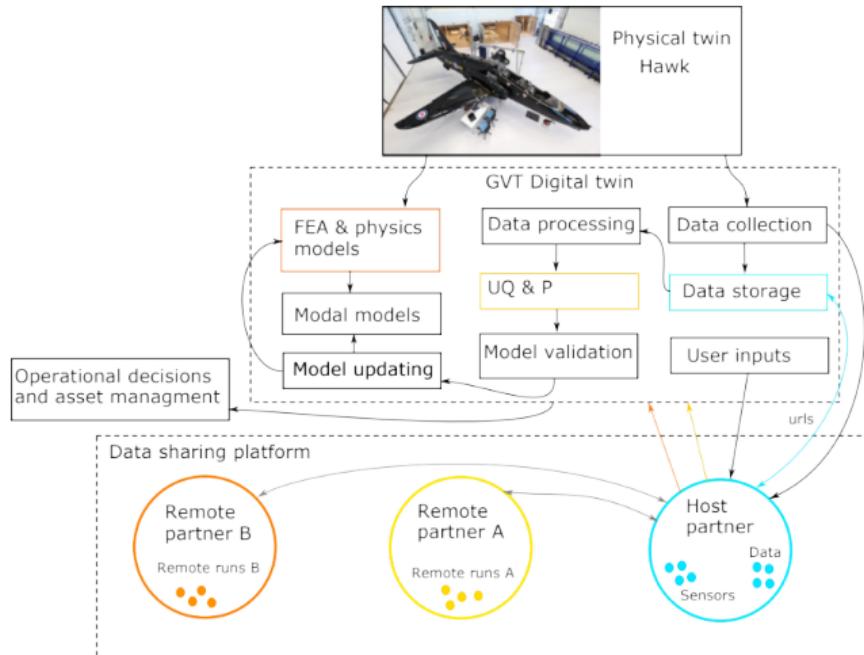
Our motto: 'Learning by doing and progressing through sharing'

The Digital Twin (DT) Hub was created in 2020 by the Centre for Digital Britain at the University of Cambridge. In 2022, the DT Hub transitioned to a multi-sector Industry and Catapult Network partnership housed at the Connected Places Catapult.



Finite Element Simulation: The Workhorse of DTs

Hawk Aircraft Digital Twin Structure



In the Digital Twin of the Hawk aircraft, a core component of the model is **Finite Element Simulation**.

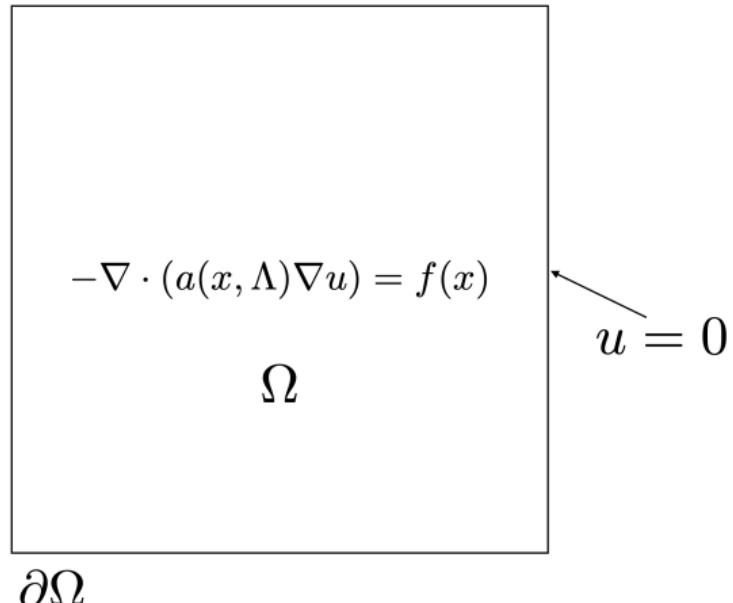
- Underneath FEM simulations are **partial differential equations** which describe the **physical processes** which are occurring within the domain of interest.
- These transcribe physical laws into mathematical objects which we can use and realise analytically, or, **computationally**.
- For example, conservation of momentum and conservation of mass of fluids gives us the famous Navier-Stokes equations.
- Often numerical methods are the only recourse to interacting with these systems; the **most popular** being the **Finite Element Method**.

Finite Element Methods

To illustrate FEM, consider the Poisson equation:

$$\begin{aligned}-\nabla \cdot (a(x, \Lambda) \nabla u) &= f(x), \quad x \in \Omega \\ u(x) &= 0, \quad x \in \partial\Omega.\end{aligned}$$

Note: Ω can be an arbitrary shape, but for a 2D square, the setup looks like:



FEM: Function Space Setup

- Now we want to look for solutions $u \in H_0^1(\Omega)$, and we assume $a \in L^\infty(\Omega)$ and $f \in L^2(\Omega)$. To do so we introduce the **weak form**.
- Weak form of PDE enables “weak solutions” which lower differentiability requirements and ease solving.
- Letting $v \in \mathcal{V}(\Omega) := \{v \in H^1(\Omega) : v = 0 \text{ on } \partial\Omega\}$, we multiply and integrate to give the **weak form**:

$$\int_{\Omega} (a(x, \Lambda) \nabla u) \cdot \nabla v \, dx = \int_{\Omega} f \cdot v \, dx, \quad \forall v \in \mathcal{V}(\Omega).$$

Which we can write as the shorthand bilinear form $\mathcal{A}_\Lambda(u, v) = \langle f, v \rangle$.

- This is the beginning of our **discretization** of the PDE.

FEM: Discretization from the Weak Form

Introduce degree- r polynomial basis functions $\{\phi_i(x)\}_{i=1}^M$, parameterized by a mesh refinement parameter h . Subsequently, $\mathcal{V}_h(\Omega) := \text{span}\{\phi_i(x)\}_{i=1}^M$; the space we “look” for solutions in.

Noting that $\mathcal{V}_h(\Omega) \subset \mathcal{V}(\Omega)$, then the variational problem becomes finite-dimensional, $u_h = \sum_{i=1}^M u_{h,i} \phi_i(x)$, and so

$$\mathcal{A}_\Lambda(u_h, \phi_j) = \langle f, \phi_j \rangle, \quad \forall j = 1, \dots, M.$$

Theorem (FEM convergence)

Let $\|\cdot\|_s$ be the Sobolev norm of degree s . Under technical assumptions for degree-1 polynomial bases $\{\phi_i\}_{i=1}^M$ (i.e. piecewise continuous)

$$\|u - u_h\|_{L^2(\Omega)} \leq Ch^2 \|f\|_{L^2(\Omega)}.$$

FEM: Solving

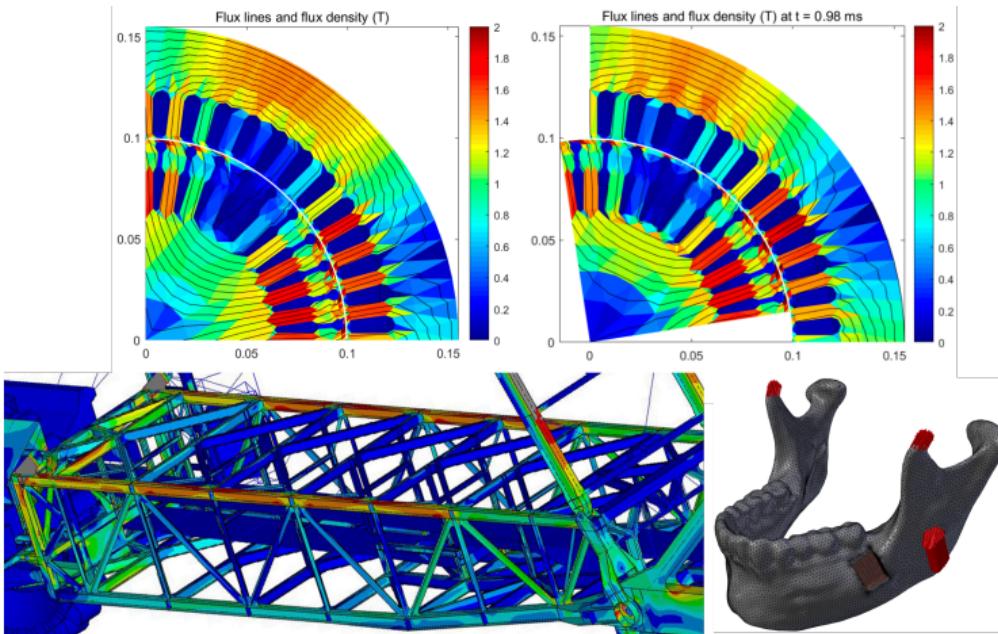
The finite-dimensional weak form defines a linear system to solve:

$$\mathbf{A}\mathbf{u} = \mathbf{b},$$

$$\mathbf{A}_{ji} = \mathcal{A}_\Lambda(\phi_i, \phi_j), \mathbf{u}_i = u_{h,i}, \mathbf{b}_i = \langle f, \phi_i \rangle.$$

- Deterministic system of equations to solve for the solution vector $\mathbf{u} = \mathbf{A}^{-1}\mathbf{b}$.
- Powerful framework to turn PDE descriptions of physical systems to linear — or nonlinear — deterministic systems of equations.
- Advanced numerical linear algebra to solve these systems: for example, Krylov methods, multigrid methods/preconditioners.

FEM: Classical Simulation Examples



FEM is widely used across all engineering, physics, and biomechanics.¹

¹ Figures from <https://link.springer.com/article/10.1007/s00784-018-2671-z>, <https://www.mdpi.com/2073-8994/13/2/254/htm>, https://www.twscon.com/en/fem_analysis.html

Requirement for Statistical Construction of FEM

- However we know that models are often idealized representations of reality. There is inherent **uncertainty** due to possible misspecification.
- Roughly, our models should be of the form (see, e.g., Kennedy and O'Hagan)

$$\mathbf{u} \approx \mathbf{A}^{-1}\mathbf{b} + \boldsymbol{\xi},$$

where $\boldsymbol{\xi}$ describes **structural, stochastic, model error**.

- This allows for **uncertainty quantification**, through the admission of **possibly misspecified model parameters**, enabling:
 1. The assimilation of data;
 2. The solving of the inverse problem;
 3. The combination of deep learning approaches with physical systems.

How do we construct such **statistical descriptions of FEM**? We will now go through one such approach.

Requirement for Statistical Construction of FEM

- However we know that models are often idealized representations of reality. There is inherent **uncertainty** due to possible misspecification.
- Roughly, our models should be of the form (see, e.g., Kennedy and O'Hagan)

$$\mathbf{u} \approx \mathbf{A}^{-1}\mathbf{b} + \boldsymbol{\xi},$$

where $\boldsymbol{\xi}$ describes **structural, stochastic, model error**.

- This allows for **uncertainty quantification**, through the admission of **possibly misspecified model parameters**, enabling:
 1. The assimilation of data;
 2. The solving of the inverse problem;
 3. The combination of deep learning approaches with physical systems.

How do we construct such **statistical descriptions of FEM**? We will now go through one such approach.

The Statistical Finite Element Method (statFEM) Construction

statFEM Prior Measure

Consider a stochastically forced elliptic PDE,

$$\begin{aligned}\mathcal{L}u &= -\nabla \cdot (a(x, \Lambda) \nabla u(x)) = \xi, \quad x \in \Omega \\ u &= 0, \quad x \in \partial\Omega,\end{aligned}$$

where $\xi \sim \mathcal{GP}(f, k_\theta(x, x'))$, and $\Omega \subset \mathbb{R}^d$ is open and bounded.

We shall assume that a has no unknown parameters and is thus fully deterministic, so $a(x, \Lambda) := a(x)$. It is also assumed $a(\cdot) \in C(\bar{\Omega}) \cap C^1(\Omega)$ and $a(x) > a_0 > 0$ for $x \in \Omega$.

Because \mathcal{L} is linear and deterministic, we have, formally that

$$u \sim \mathcal{GP}(\mathcal{L}^{-1}f, k_{\mathcal{L}, \theta}),$$

where

$$k_{\mathcal{L}, \theta}(x, y) = (-\mathcal{L})_x^{-1}(-\mathcal{L})_y^{-1}k_\theta(x, y).$$

Gaussian process ξ induces **hierarchical prior distribution** on the space of fields u .

From the Gaussian process we can look at the associated Gaussian measure on $L^2(\Omega)$. Via the weak interpretation of elliptic example we have $\mathcal{L}^{-1}: L^2(D) \rightarrow L^2(D)$, and thus

$$u \sim \mu_0 = \mathcal{N}(\mathcal{L}^{-1}f, \mathcal{L}^{-1}C_\theta\mathcal{L}^{-*}),$$

where C_θ is the covariance operator with kernel $k_\theta(\cdot, \cdot)$.

In the function-space setting Gaussian measure provides an appropriate prior measure μ_0 , which can be updated to give the posterior measure μ^y .

Next suppose that data, $\mathbf{y} \in \mathbb{R}^{n_y}$, is observed via $\mathbf{y} = \mathcal{H}u + \eta$, where \mathcal{H} is the continuous observation operator, and $\eta \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. From this we can define

$$\frac{d\mu^y}{d\mu_0}(u) \propto \exp(-\Phi(u; \mathbf{y})), \quad \Phi(u; \mathbf{y}) = \frac{1}{2\sigma^2} \|\mathbf{y} - \mathcal{H}u\|_2^2.$$

The Radon-Nikodym derivative

$$\frac{d\mu^y}{d\mu_0}(u) \propto \exp(-\Phi(u; \mathbf{y})), \quad \Phi(u; \mathbf{y}) = \frac{1}{2\sigma^2} \|\mathbf{y} - \mathcal{H}u\|_2^2.$$

defines the posterior Gaussian measure μ^y , on the $L^2(\Omega)$ function space.

This posterior is absolutely continuous with respect to the prior, $\mu_0 \ll \mu$, inheriting “information” from the prior distribution. The prior defines the starting point for inference and should be well-specified!

Using statFEM, data can update our prior beliefs in the model solution.

Now, when discretised, can we be sure that this prior is converging to something sensible?

The Radon-Nikodym derivative

$$\frac{d\mu^y}{d\mu_0}(u) \propto \exp(-\Phi(u; \mathbf{y})), \quad \Phi(u; \mathbf{y}) = \frac{1}{2\sigma^2} \|\mathbf{y} - \mathcal{H}u\|_2^2.$$

defines the posterior Gaussian measure μ^y , on the $L^2(\Omega)$ function space.

This posterior is absolutely continuous with respect to the prior, $\mu_0 \ll \mu$, inheriting “information” from the prior distribution. The prior defines the starting point for inference and should be well-specified!

Using statFEM, data can update our prior beliefs in the model solution.

Now, when discretised, can we be sure that this prior is converging to something sensible?

Analysis: Set-Up

Given a triangulated mesh Ω_h with mesh-size $h > 0$ we can introduce a finite element approximation with M degrees of freedom.

This leads to approximate prior:

$$u_h(x) \sim \mathcal{GP}(\Phi_h(x)\mathbf{A}^{-1}\mathbf{b}, \Phi_h(x)\mathbf{C}_u\Phi_h(x)^*),$$

where $\Phi_h : \Omega \rightarrow \mathbb{R}^M$ is the basis-to-coordinate map. This is an intrinsically finite dimensional distribution as

$$u_h(\cdot) \sim \Phi_h(\cdot)\mathcal{N}(\mathbf{A}^{-1}\mathbf{b}, \mathbf{C}_u),$$

where \mathbf{A} is the stiffness matrix and \mathbf{C}_u is the induced covariance by the FEM basis.

Analysis of the prior

The random fields u and u_h are two Gaussian processes on the same compact domain Ω .

We expect that $u_h \rightarrow u$ as $h \rightarrow 0$ in an appropriate sense. Can we quantify the discretisation error?

For risk-assessment applications, it is important that we can control the error over all aspects of the distributions, i.e. we should be able to control the error in:

- mean
- covariance
- quantiles
- extrema (spatial maximum, spatial minimum distributions)
- other appropriate quantities of interest.

Analysis strategy

What is the appropriate metric/distance in which to assess the discrepancy between fields u and u_h which captures all of the above?

Challenge: We have no guarantees that the measures u_h and u are not mutually singular on the Hilbert space on which they are both supported.

This precludes analysis based on KL, Fisher, Hellinger or related distances/divergences.

Proposed Approach: Analyse error in p -Wasserstein distance, for $p \geq 2$.

Main Results

To upper-bound the Wasserstein distance, we exploit a connection between the Wasserstein distance between Gaussian measures and the Procrustes Metric on the respective covariance operators (Masarotto, Panaretos, and Zemel)².

Theorem (Vastly simplified main result)

Suppose that $f \in L^2(\Omega)$ where Ω is open and bounded with Lipschitz boundary and the covariance operator k_θ satisfies some technical conditions. Then for given a regular triangulation Ω_h of Ω with mesh size h sufficiently small, there exists $\gamma > 0$ independent of h such that:

$$W_2(\mathcal{D}[u], \mathcal{D}[u_h]) \leq \gamma h^2,$$

where $\mathcal{D}[u]$ is the measure associated with the Gaussian random field u .

²Yanni Papandreou et al. "Theoretical Guarantees for the Statistical Finite Element Method". *SIAM/ASA Journal on Uncertainty Quantification* (Dec. 2023): 1278–1307.

Main Results

This bound is consistent with $L^2(\Omega)$ a-priori error estimates for deterministic finite element approximations obtained via Aubin-Nitche. In particular, as the variance of the noisy forcing goes to zero, we have that

$$\mathcal{D}[u] \rightarrow \delta_{\mathcal{L}^{-1}f}, \text{ and } \mathcal{D}[u_h] \rightarrow \delta_{\mathcal{L}_h^{-1}f_h},$$

and

$$W_2(\delta_{\mathcal{L}^{-1}f}, \delta_{\mathcal{L}_h^{-1}f_h}) = \|\mathcal{L}^{-1}f - \mathcal{L}_h^{-1}f_h\|_{L^2(\Omega)} = O(h^2)$$

Realizing statFEM

statFEM: Resultant Discretization

Of course, to work with these measures they must be realized in practice. We can write our discretized (stochastic) weak form as

$$\mathcal{A}_\Lambda(u_h, \phi_j) = \langle \xi, \phi_j \rangle, \quad \forall j = 1, \dots, M.$$

Which gives the Gaussian law

$$(\mathbf{u} | \Lambda, \theta) \sim \mathcal{N}(\mathbf{A}^{-1}\mathbf{b}, \mathbf{A}^{-1}\mathbf{G}_\theta\mathbf{A}^{-\top}),$$

where $\mathbf{b}_j = \langle f, \phi_j \rangle$, $\mathbf{A}_{jk} = \mathcal{A}_\Lambda(\phi_j, \phi_k)$, $\mathbf{G}_{\theta,jk} = \langle \phi_j, \langle k_\theta(\cdot, \cdot) \phi_k \rangle \rangle$.

Combining with data

- Now suppose that we have observed some data $\mathbf{y} \in \mathbb{R}^N$. Observed on a grid $\mathbf{X} = (\mathbf{x}_1^{\text{obs}}, \dots, \mathbf{x}_N^{\text{obs}})$, with some measurement error.
- Measurement process could be described if full knowledge of true process available.
- Data is a linear combination of the measurement error and the response of the true unknown generating process.
- Define $\mathbf{H} : \mathbb{R}^M \rightarrow \mathbb{R}^N$, the **discrete observation operator** which maps from the domain of the FEM solution: $\mathbf{H}\mathbf{u} = (u_h(\mathbf{x}_1^{\text{obs}}), \dots, u_h(\mathbf{x}_N^{\text{obs}}))$.

Combining with data

To deal with possible model misspecification we posit the following data generating process (DGP): $\mathbf{y} = \rho \mathbf{H}\mathbf{u} + \mathbf{d} + \boldsymbol{\varepsilon}$,

- $\mathbf{y} \in \mathbb{R}^N$: observations observed on $\mathbf{X} \in \Omega$.
- $\mathbf{u} \in \mathbb{R}^M$: statFEM model, $p(\mathbf{u} | \Lambda, \theta) \sim \mathcal{N}(\mathbf{A}^{-1}\mathbf{b}, \mathbf{A}^{-1}\mathbf{G}_\theta\mathbf{A}^{-\top})$.
- $\mathbf{H} : \mathbb{R}^M \rightarrow \mathbb{R}^N$: observation operator.
- $\mathbf{d} \sim \mathcal{GP}(0, \mathbf{K}_d)$: systematic model bias/discrepancy/mismatch (similar ideas from Kennedy-O'Hagan type models — models the functional difference between FEM and observed data).
- $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma_y^2 \mathbf{I})$: observation noise, could be known from measurement devices or able to be inferred using e.g. marginal likelihood.
- Denote any hyperparameters of \mathbf{d} , $\boldsymbol{\varepsilon}$ as \mathbf{w} , and assumed $\mathbf{u} \perp \mathbf{d} \perp \boldsymbol{\varepsilon}$.

Likelihood and posterior

This gives the likelihood $(\mathbf{y} | \mathbf{u}, \mathbf{w}) \sim \mathcal{N}(\mathbf{H}\mathbf{u}_h, \mathbf{K}_d + \sigma^2 \mathbf{I})$, which can be combined with the prior $p(\mathbf{u}_h | \mathbf{f}, \Lambda, \theta)$ to give the posterior

$$p(\mathbf{u} | \mathbf{y}, \mathbf{w}, \Lambda, \theta) \propto p(\mathbf{y} | \mathbf{u}, \mathbf{w}) \cdot p(\mathbf{u}_h | \Lambda, \theta).$$

Statistically coherent combination of prior physical knowledge *and* observed data, taking into account model mismatch.

Allows for known information from the physics of the problem to be rigorously incorporated in the inference procedure.

Marginal likelihood:

$$p(\mathbf{y} | \mathbf{w}, \Lambda, \theta) = \mathcal{N}(\mathbf{H}\mathbf{m}_u, \mathbf{H}\mathbf{C}_u\mathbf{H}^\top + \mathbf{K}_d + \sigma^2 \mathbf{I}).$$

We optimize the marginal likelihood to learn the mismatch hyperparameters.

Sampling the statFEM Prior Distribution

Characterise the statFEM measures through sampling methods which avoid computing matrix square-roots: **unadjusted Langevin methods**. Let $\Psi(\mathbf{u}) = \frac{1}{2} \|\mathbf{A}\mathbf{u} - \mathbf{b}\|_{\mathbf{G}^{-1}}^2$, $\nabla_{\mathbf{u}}\Psi(\mathbf{u}) = \mathbf{A}^\top \mathbf{G}^{-1} (\mathbf{A}\mathbf{u} - \mathbf{b})$ and³

$$d\mathbf{u} = -\nabla_{\mathbf{u}}\Psi(\mathbf{u}) dt + \sqrt{2} d\mathbf{B}_t,$$

where $(\mathbf{B}_t)_{t \geq 0}$ is a SBM.

The asymptotic law is $(\mathbf{u} | \Lambda, \theta)$, samples $\{\mathbf{u}^{(k)}\}$ generated using the **Euler-Maruyama step**:

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \eta \nabla_{\mathbf{u}}\Psi(\mathbf{u}^{(k)}) + \sqrt{2\eta} \mathbf{Z}^{(k)}.$$

Under Lipschitz gradients this also extends to nonlinear forward models i.e.

$$\Psi(\mathbf{u}) = \frac{1}{2} \|\mathbf{A}(\mathbf{u}) - \mathbf{b}\|_{\mathbf{G}^{-1}}^2.$$

³Ömer Deniz Akyildiz et al. "Statistical Finite Elements via Langevin Dynamics". *SIAM/ASA Journal on Uncertainty Quantification* (Dec. 2022): 1560–1585.

Sampling the Prior

Denoting the law of the **unadjusted Langevin algorithm** as $p_k(\mathbf{u} | \Lambda, \theta)$, we can provide guarantees for the convergence to the conditional target measure $p(\mathbf{u} | \Lambda, \theta)$:

$$\text{KL}(p_k || p) \leq \mathcal{O}(e^{-\eta m k}) + \mathcal{O}(\eta),$$

for some $m > 0$ (due to strong convexity) and η is the step-size of the ULA.

The bias can be made arbitrarily small with small $\eta > 0$ and $k \rightarrow \infty$.

Sampling the Prior

More precisely, we obtain guarantees for the convergence to the conditional target measure $p(\mathbf{u} | \Lambda, \theta)$:

$$\text{KL}(p_k || p) \leq e^{-\lambda_{\min}(\mathbf{A}^\top \mathbf{G}^{-1} \mathbf{A}) \eta k} \text{KL}(p_0, p) + 8\eta d \frac{\lambda_{\max}(\mathbf{A}^\top \mathbf{G}^{-1} \mathbf{A})^2}{\lambda_{\min}(\mathbf{A}^\top \mathbf{G}^{-1} \mathbf{A})},$$

where $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the maximum and minimum eigenvalue of a given matrix, respectively.

The same scheme can be used to sample from the posterior, giving a similar convergence rate:

$$\text{KL}(p_k || p) \leq e^{-\lambda_{\min}(\mathbf{C}_y) \eta k} \text{KL}(p_0, p) + 8\eta d \frac{\lambda_{\max}(\mathbf{C}_y^{-1})^2}{\lambda_{\min}(\mathbf{C}_y^{-1})}.$$

Powerful sampling methodology can be used to generate samples and characterise the prior and posterior statFEM measures.

statFEM Numerical Results: Samplers

Empirical Results

For our experiments, we consider the Poisson problem

$$\begin{aligned} -\nabla \cdot (a(x)\nabla u(x)) &= f(x) + \xi(x), & x \in \Omega, \\ u &= 0, & x \in \partial\Omega, \end{aligned}$$

where $\Omega = [0, 1] \times [0, 1]$ and $f(x) \equiv 1$. Stochasticity is defined as

$$\begin{aligned} \xi &\sim \mathcal{GP}(0, \beta^2 \delta(x - x')), \quad \beta = 0.05, \\ \log a(x) &\sim \mathcal{GP}(\log(1 + 0.3 \sin(\pi(x_0 + x_1))), k_\theta(x, x')) \end{aligned}$$

where

$$k_\theta = 0.1^2 \exp\left(-\frac{\|x - x'\|^2}{2 \cdot 0.2^2}\right).$$

Empirical Results

Upon discretization we have conditional Gaussian

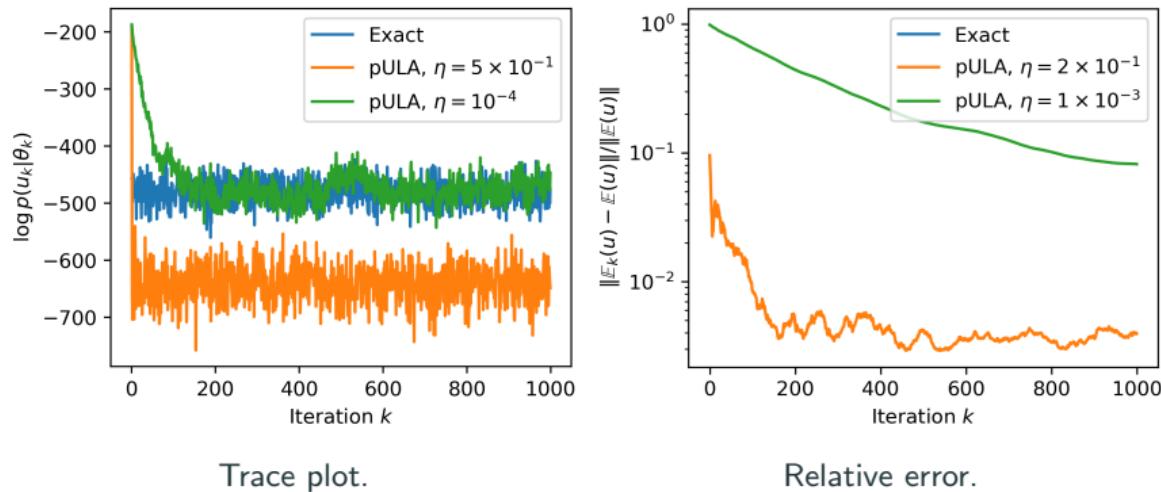
$$p(\mathbf{u}|\theta) = \mathcal{N}(\mathbf{A}^{-1}\mathbf{b}, \mathbf{A}^{-1}\mathbf{G}\mathbf{A}^{-\top}).$$

We construct \mathbf{G} by noting that

$$\begin{aligned}\tilde{G}_{ij} &= \beta^2 \int_{\Omega} \phi_i(x) \int \delta(x - x') \phi_j(x') \, dx' \, dx \\ &= \beta^2 \int_{\Omega} \phi_i(x) \phi_j(x) \, dx' \, dx = \beta^2 \mathbf{M}_{ij},\end{aligned}$$

for the mass matrix \mathbf{M} . We next make a diagonal approximation such that $\mathbf{G}_{ii} = \sum_j \tilde{\mathbf{G}}_{ij}$ (*lumping*).

Empirical Results



Prior results for state dimension $d = 1089$, plotting the sampled values of the log-target (a), and the relative errors on the mean (b). For ULA, the stepsize offers a tradeoff between bias and rate of convergence.

Empirical results

Consider the prior of the previous examples, and now suppose that the data are observed according to a nonlinear observational model:

$$\mathbf{y}_i = \mathbf{H}(\mathbf{u}) + \boldsymbol{\varepsilon}_i, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}).$$

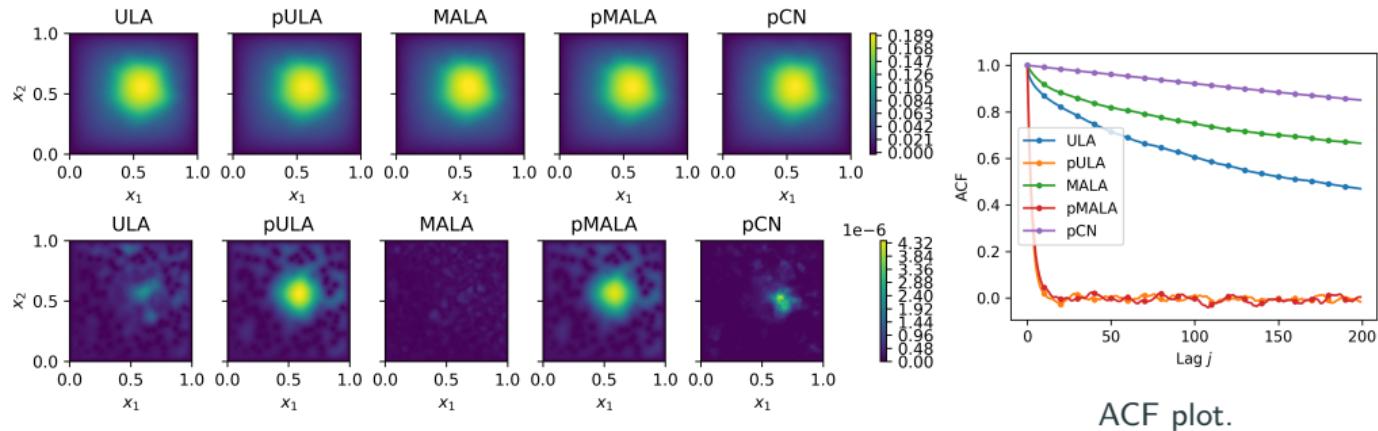
We take $\mathbf{H}(\cdot)$ to be a sigmoid function to mimic sensor “saturation” past a certain level.

Take n_{obs} observations in total, so the potential for $p(\mathbf{u}|\mathbf{y}, \theta)$ is given by

$$\Psi(\mathbf{u}) = \frac{1}{2} \sum_{i=1}^{n_{\text{obs}}} (\mathbf{y}_i - \mathbf{H}(\mathbf{u}))^\top \mathbf{R}^{-1} (\mathbf{y}_i - \mathbf{H}(\mathbf{u})) + \frac{1}{2} (\mathbf{A}\mathbf{u} - \mathbf{b})^\top \mathbf{G}^{-1} (\mathbf{A}\mathbf{u} - \mathbf{b}).$$

Compare MALA, pMALA, ULA, pULA, and pCN.

Empirical results



Posterior means ($\mathbb{E}(\mathbf{u})$, top), and variance fields ($\text{var}(\mathbf{u})$, bottom).

Nonlinear likelihood: posterior results. The ACF plot is shown for the samples from the FEM coefficient $\mathbf{u}^{(100)}$. All samplers are accurate in the mean, but preconditioning captures the variance.

Time-varying, Nonlinear Dynamics

Time-varying statFEM

Use **stochastic dynamics**. Take a nonlinear, time-evolving system:

$$\partial_t u + \mathcal{L}u + \mathcal{N}(u) = \xi_\theta,$$

we have $u := u(\mathbf{x}, t)$, $\mathbf{x} \in \Omega$, $t \in [0, T]$. Model **uncertainty** with a GP (with known θ):

$$\xi_\theta(\mathbf{x}, t) \sim \mathcal{GP}(0, \delta(t - t') \cdot k_\theta(\mathbf{x}, \mathbf{x}')).$$

Making a spatial discretisation with FEM, and time-discretisation gives a **transition model** over the FEM coefficients $\mathbf{u}_n := \mathbf{u}(n\Delta_t)$:

$$\mathcal{M}(\mathbf{u}_n, \mathbf{u}_{n-1}) = \mathbf{e}_n, \quad \mathbf{e}_n \sim \mathcal{N}(\mathbf{0}, \Delta_t \mathbf{G}).$$

Nonlinear dynamics encoded in $\mathcal{M}(\cdot, \cdot)$, implied **transition densities** $p(\mathbf{u}_n | \mathbf{u}_{n-1}, \Lambda, \theta)$.

Time-dependent data synthesis

With data $\mathbf{y}_n \in \mathbb{R}^{n_y}$, we write $\mathbf{y}_n = \mathbf{H}\mathbf{u}_n + \boldsymbol{\varepsilon}_n$. This gives the state-space model

$$\mathcal{M}(\mathbf{u}_n, \mathbf{u}_{n-1}) = \mathbf{e}_n, \quad \mathbf{e}_n \sim \mathcal{N}(\mathbf{0}, \Delta_t \mathbf{G}), \quad (\text{Transition})$$

$$\mathbf{y}_n = \mathbf{H}\mathbf{u}_n + \boldsymbol{\varepsilon}_n, \quad \boldsymbol{\varepsilon}_n \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I}). \quad (\text{Observation})$$

Compute the posterior $p(\mathbf{u}_n | \mathbf{y}_{1:n}, \Lambda, \theta)$ using nonlinear filtering methods such as the **extended** Kalman filter (ExKF)⁴.

⁴Connor Duffin et al. “Statistical Finite Elements for Misspecified Models”. *Proceedings of the National Academy of Sciences* (Jan. 2021).

Time-varying statFEM

With the **observation model** this gives a **nonlinear Gaussian state-space model**.

Assuming $p(\mathbf{u}_{n-1}|\mathbf{y}_{1:n-1}) = \mathcal{N}(\mathbf{m}_{n-1}, \mathbf{C}_{n-1})$ we compute the posterior $p(\mathbf{u}_n|\mathbf{y}_{1:n})$ via filtering:

1. Prediction: $p(\mathbf{u}_n|\mathbf{y}_{1:n-1}) = \int p(\mathbf{u}_n|\mathbf{u}_{n-1}, \mathbf{y}_{1:n-1})p(d\mathbf{u}_{n-1}|\mathbf{y}_{1:n-1}).$
2. Update: $p(\mathbf{u}_n|\mathbf{y}_{1:n}) \propto p(\mathbf{y}_n|\mathbf{u}_n)p(\mathbf{u}_n|\mathbf{y}_{1:n-1}).$

This gives the Gaussian $p(\mathbf{u}_n|\mathbf{y}_{1:n}) = \mathcal{N}(\mathbf{m}_n, \mathbf{C}_n) \implies$ describes **uncertainty with solution** given all information **up to and including** current time $n\Delta_t$. Scaled with low-rank approaches.

Conditioning procedure for time-dependent problems

The ExKF is given by, if one assumes that $p(\mathbf{u}_{n-1} \mid \mathbf{y}_{1:n-1}, \boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{m}_{n-1}, \mathbf{C}_{n-1})$:

1. (*Prediction step*) Solve for $\hat{\mathbf{m}}_n$, $\mathcal{M}_\Lambda(\hat{\mathbf{m}}_n, \mathbf{m}_{n-1}) = \mathbf{0}$, and update the covariance matrix

$$\hat{\mathbf{C}}_n = \mathbf{J}_n^{-1} \left(\mathbf{J}_{n-1} \mathbf{C}_{n-1} \mathbf{J}_{n-1}^\top \right) \mathbf{J}_n^{-\top} + \Delta_t \mathbf{J}_n^{-1} \mathbf{G} \mathbf{J}_n^{-\top},$$

$\mathbf{J}_.$ is the Jacobian of the the nonlinear $\mathcal{M}(\cdot, \cdot)$.

2. (*Update step*) Set the observed marginal likelihood covariance

$$\mathbf{S}_n = \mathbf{H}_n \hat{\mathbf{C}}_n \mathbf{H}_n^\top + \sigma_n^2 \mathbf{I}, \text{ and compute a standard Kalman update on the mean}$$

$$\mathbf{m}_n = \hat{\mathbf{m}}_n + \hat{\mathbf{C}}_n \mathbf{H}_n^\top \mathbf{S}_n^{-1} (\mathbf{y}_n - \mathbf{H}_n \hat{\mathbf{m}}_n).$$

and the covariance

$$\mathbf{C}_n = \hat{\mathbf{C}}_n - \hat{\mathbf{C}}_n \mathbf{H}_n^\top \mathbf{S}_n^{-1} \mathbf{H}_n \hat{\mathbf{C}}_n.$$

Note that this general procedure is exact for linear models - the Jacobians \mathbf{J}_n just become the appropriate linear combinations of FEM matrices (e.g. the mass/stiffness matrices).

Aside: scaling to high dimensions

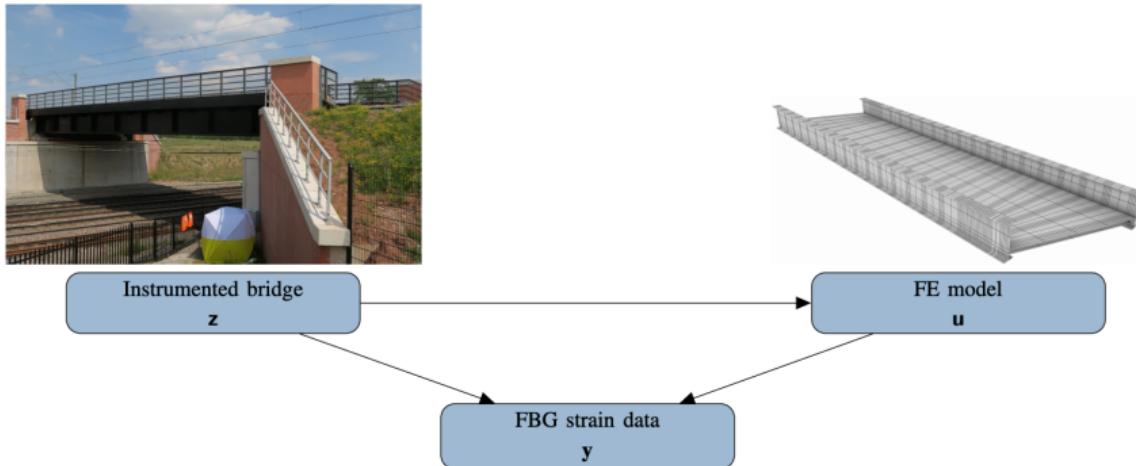
ExKF works for low-dimensional systems but is **not scalable!** How to scale the method?

- To compute posterior $p(\mathbf{u}_n | \mathbf{y}_{1:n}, \Lambda)$ we use a **low-rank Extended Kalman filter (LR-ExKF)**.
- Idea: \mathcal{GP} covariance matrices (typically) only need a few dominant modes (eigenvector/value pairs) to describe the system. Leverage this inside of ExKF.
- LR-ExKF constructs approximate measure $p(\mathbf{u}_n | \mathbf{y}_{1:n}, \Lambda) = \mathcal{N}(\mathbf{m}_n, \mathbf{L}_n \mathbf{L}_n^\top)$,
 $\mathbf{m}_n \in \mathbb{R}^{n_u}$, $\mathbf{L}_n \in \mathbb{R}^{n_u \times k}$, $k \ll n_u$ ⁵.
- Low-rank approximation is optimal in the ℓ^2 sense so UQ is sensible (not the case with, e.g., EnKF).

⁵Connor Duffin et al. “Low-Rank Statistical Finite Elements for Scalable Model-Data Synthesis”. *Journal of Computational Physics* (Aug. 2022).

Examples

Staffordshire bridge



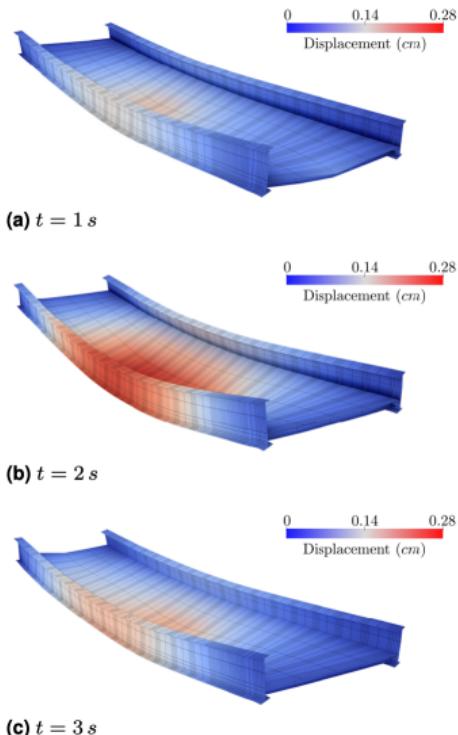
- Digital twin of Staffordshire bridge: how to combine sensor data with FEM model, acknowledging model misspecification and wanting UQ⁶.

⁶Eky Febrianto et al. "Digital Twinning of Self-Sensing Structures Using the Statistical Finite Element Method". *Data-Centric Engineering* (2022): e31.

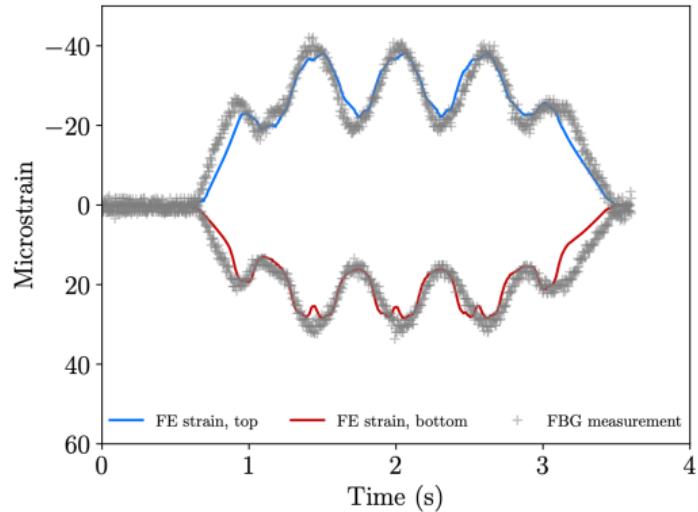
Staffordshire bridge

- Use FEM model with subdivision surfaces, to arrive at the usual Gaussian prior:
 $p(\mathbf{u}) = \mathcal{N}(\mathbf{A}^{-1}\mathbf{b}, \mathbf{A}^{-1}\mathbf{G}\mathbf{A}^{-\top})$. In this example there are no unknown parameters Λ, θ , so uncertainty in the prior only comes in stochastic forcing.
- Posit the same DGP: $\mathbf{y} = \rho \mathbf{H}\mathbf{u} + \mathbf{d} + \boldsymbol{\eta}$, assuming as previous $\mathbf{d} \sim \mathcal{N}(0, \mathbf{K}_d)$, and $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma_e^2 \mathbf{I})$.
- Covariance \mathbf{K}_d is given from a squared-exponential kernel, with parameters σ_d , and ℓ_d , so hyperparameters that need to be estimated are $\mathbf{w} = \{\rho, \sigma_d, \ell_d\}$.
- Data $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{n_o}] \in \mathbb{R}^{n_y \times n_o}$ consists of $n_o = 501$ observations at $n_y = 40$ sensors over a two-second observation window $t \in [1, 3]$ s, in which a T1-type train with four carriages passes over the bridge.
- MCMC estimates \mathbf{w} , setting them to fixed $\mathbf{w}^* = \mathbb{E}[\mathbf{w} | \mathbf{Y}]$, using prior $p(\mathbf{w}) \propto 1$.
- QoI: posterior measure $p(\mathbf{H}\mathbf{u} | \mathbf{y}, \mathbf{w}^*)$, at times $t = 1$ s, $t = 2$ s, and $t = 3$ s.

Staffordshire bridge

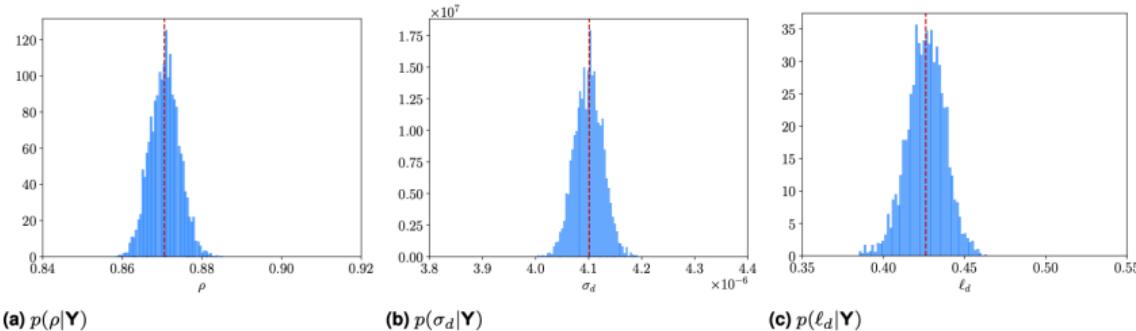


Simulated bridge displacements, using the FEM forward model.

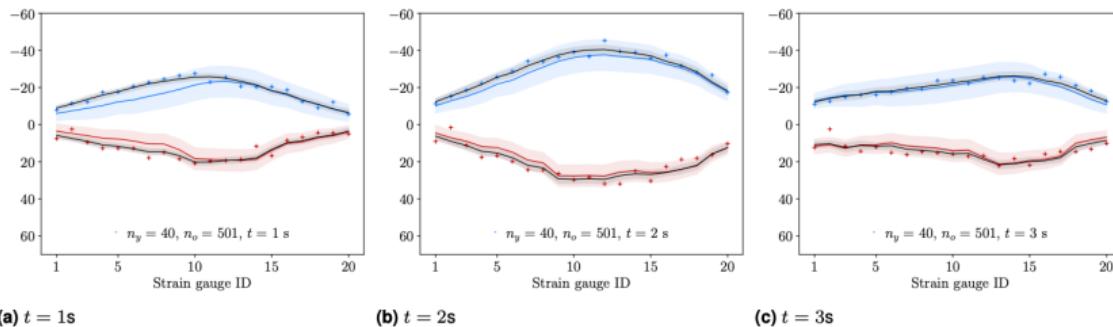


FEM strain results with experimental data, across all observation times.

Staffordshire bridge



Hyperparameter \mathbf{w} estimates from MCMC, $\mathbf{w}^* = \mathbb{E}[\mathbf{w}]$ shown as a dashed red line.

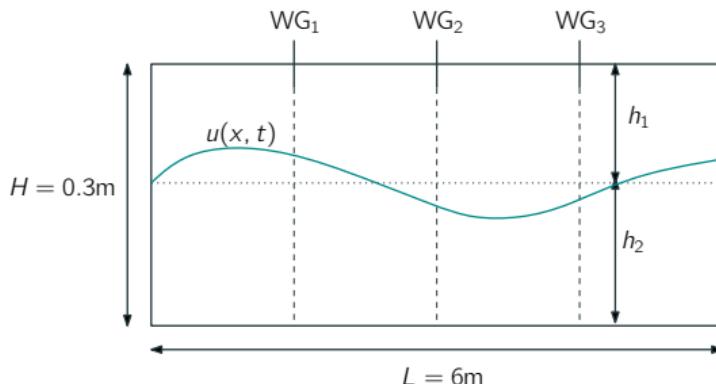


Posterior results for the statFEM posterior, shown as the projected measure $p(\mathbf{H}\mathbf{u} | \mathbf{Y}, \mathbf{w}^*)$.

Staffordshire Bridge Example: Takeaways

- Statistically coherent FEM model of Staffordshire railway bridge, with complete uncertainty quantification, via statFEM
- Acknowledgement of misspecification enables inference, model hyperparameters calibrated with data.
- Sensible UQ with models calibrated with data.

Case Study: Nonlinear Internal Waves

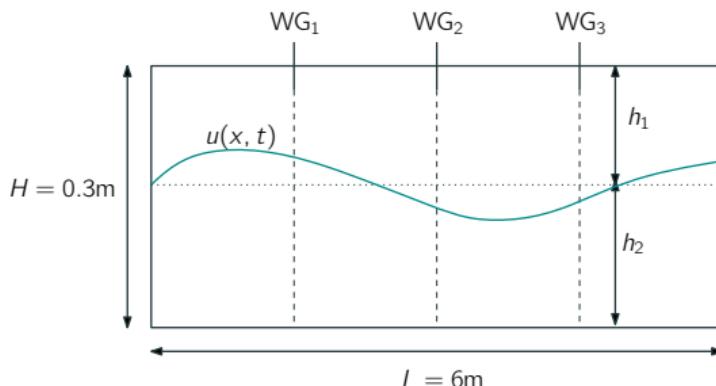


- Internal waves flow between layers of density-varying water (mean depths h_1 , h_2), in a tank of length L and total depth $H = h_1 + h_2$.
- KdV equation models the internal wave profile $u(x, t)$ (deviations from rest):

$$\partial_t u + \alpha uu_x + \beta u_{xxx} + cu_x + \nu u = \dot{\xi}.$$

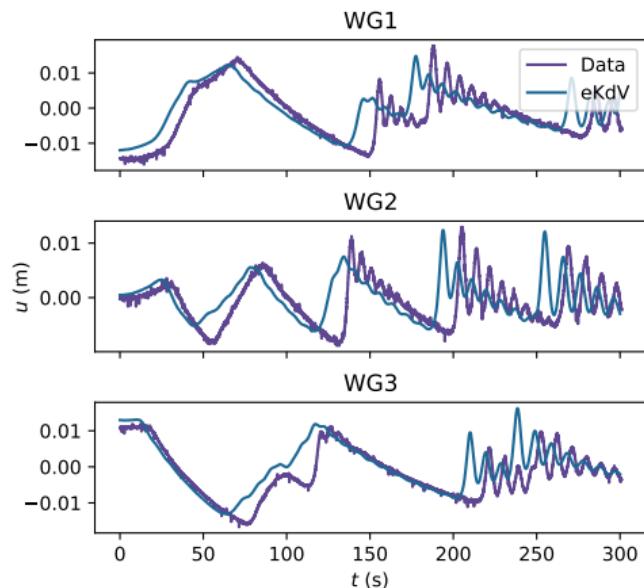
- Can we synthesise this PDE with measurements $\mathbf{y}_{1:n} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$ obtained at wave gauges (labelled above)?

Case Study: Nonlinear Internal Waves

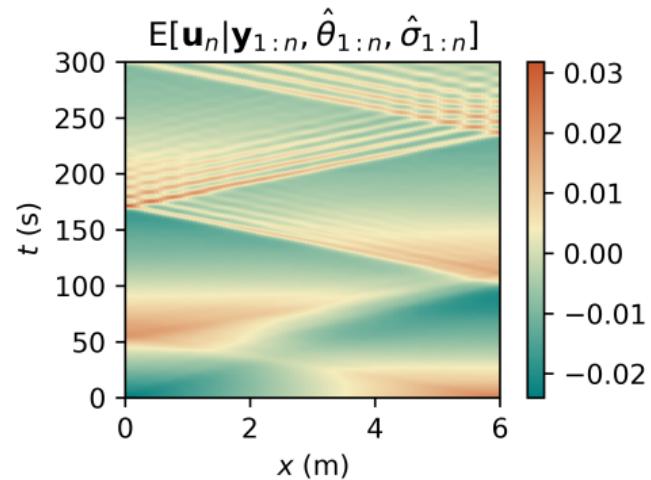


Apply statFEM to compute posterior $p(\mathbf{u}_n \mid \mathbf{y}_{1:n}, \Lambda)$ given the observations at each timestep. Observations $\mathbf{y}_n = (u_n^{\text{WG}_1}, u_n^{\text{WG}_2}, u_n^{\text{WG}_3})$, taking each of the $n_T = 1001$ timesteps for $0 \leq t \leq 300$ s.

Case Study: internal waves

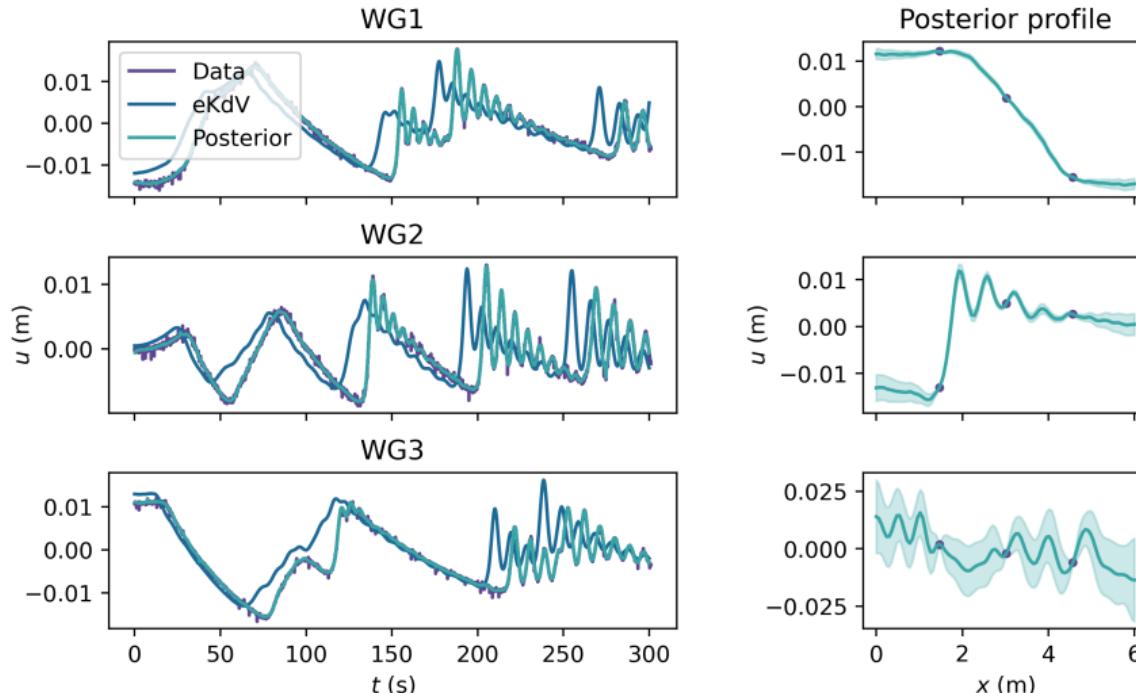


Experimental data and prior mean, up to time $t = 300$ s.



KdV posterior mean across space-time grid.

Case study: internal waves



StatFEM posterior measure $p(\mathbf{u}_n | \mathbf{y}_{1:n}, \Lambda)$ for the KdV equation: posterior at WG locations (left); posterior wave profile $u(s, t)$ for $t = \{75, 150, 225\}$ s (right).

Internal Waves: Takeaways

- Assimilated data with KdV equation: allows for physics-informed interpolator, with an interpretable posterior distribution.
- Uncertainty quantification is sensible and enables the calibration of simpler physical models with potentially sparse data.
- Conjecture: physically sensible way of incorporating nonstationarity into GP-type models → nonstationarity driven by dynamics.

Case Study: Reaction-Diffusion

Apply the statFEM methodology to the Oregonator system of equations:⁷

$$\begin{aligned} u_t &= \frac{1}{\varepsilon} \left(u(1-u) - fv \frac{u-q}{u+q} \right) + D_u \nabla^2 u + \dot{\xi}^u, \\ v_t &= u - v + D_v \nabla^2 v + \dot{\xi}^v, \end{aligned}$$

$$u := u(x, t), \quad v := v(x, t), \quad x \in [0, L] \times [0, L], \quad t \in [0, T],$$

using zero-flux boundary conditions.

Sparse measurements and misspecified initial conditions; “spiral-wave” regime:
 $(f, q, \varepsilon) = (2, 0.002, 0.02)$, $(D_u, D_v) = (1, 0.6)$.

Stochastic Oregonator system implicitly defines the prior distribution: set $\xi^u = 0$,
induce uncertainty through v -component:

$$\xi^v(x, t) \sim \mathcal{GP}(0, \delta(t - t') \cdot k_\theta(x, x')), \quad k_\theta(x, x') = \rho^2 \exp(-\|x - x'\|_2^2 / (2\ell^2)).$$

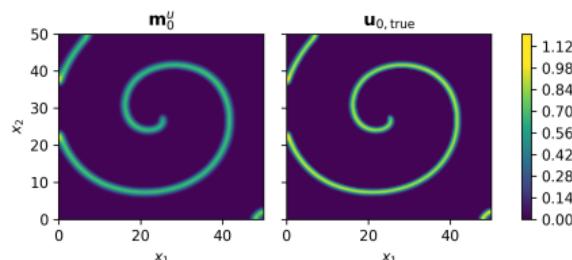
⁷Connor Duffin et al. “Low-Rank Statistical Finite Elements for Scalable Model-Data Synthesis”. *Journal of Computational Physics* (Aug. 2022).

Case Study: Reaction-Diffusion

- Discretise to give \mathbf{u}_n and \mathbf{v}_n — the FEM coefficients at time $n\Delta_t$ — and compute joint posterior with LR-ExKF:

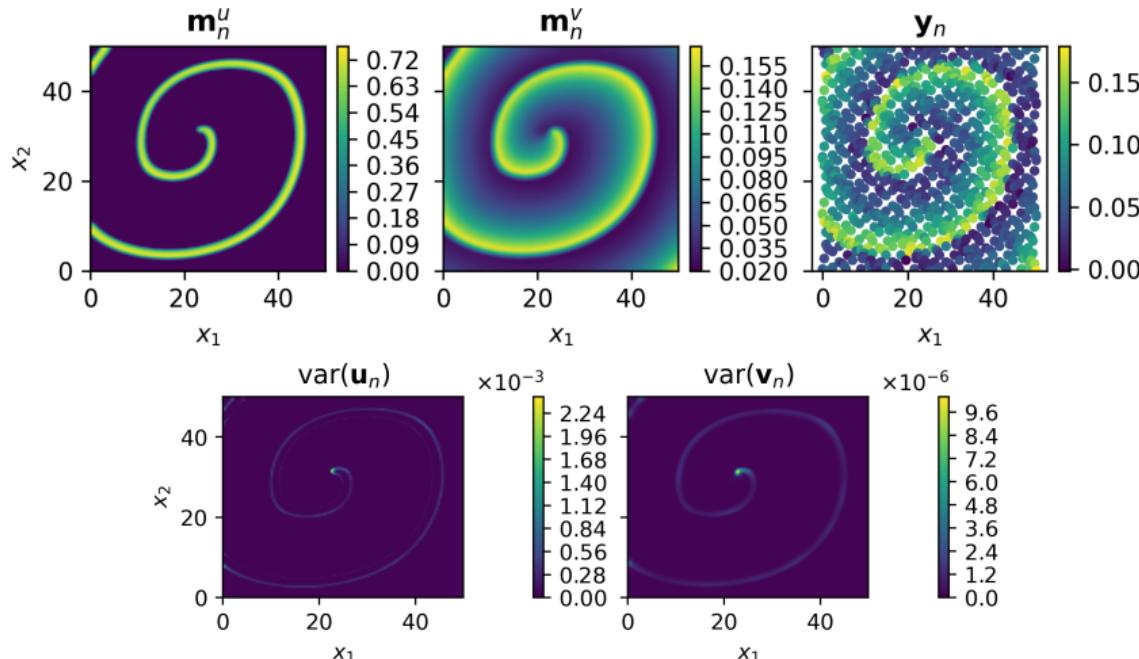
$$p(\mathbf{u}_n, \mathbf{v}_n \mid \mathbf{y}_{1:n}, \Lambda, \theta) \sim \mathcal{N}(\mathbf{m}_n, \mathbf{L}_n \mathbf{L}_n^\top).$$

- StatFEM posterior mean has blurred initial condition: model is misspecified through incorrect initialisation.
- Noisy data, only single component observed: $\mathbf{y}_n = \mathbf{H}\mathbf{v}_n + \boldsymbol{\eta}_n$, $\boldsymbol{\eta}_n \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$.



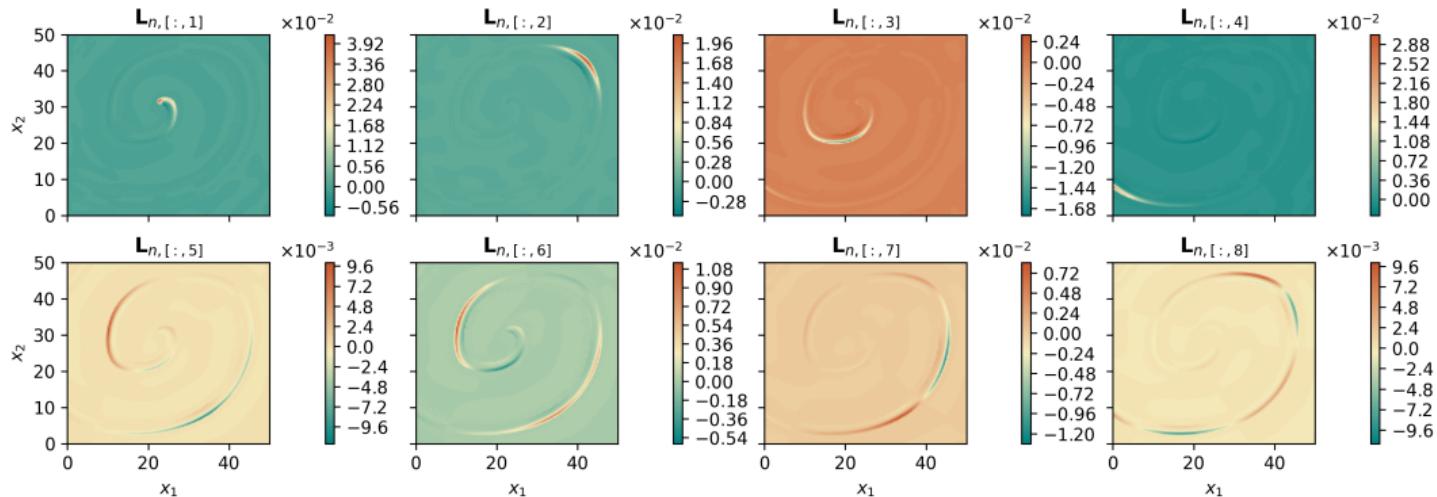
Initial conditions: statFEM (left) and true (right). We observe the \mathbf{v}_n component only $\implies \mathbf{u}_n$ misspecified and unobserved.

Case Study: Reaction-Diffusion



Top: statFEM posterior means ($\mathbf{m}_n^u, \mathbf{m}_n^v$) and observed data (\mathbf{y}_n) at time $t = 5$. Bottom: statFEM posterior variances ($\text{diag}(\mathbf{C}_n)$) on the u and v components, at time $t = 5$.

Case Study: Reaction-Diffusion

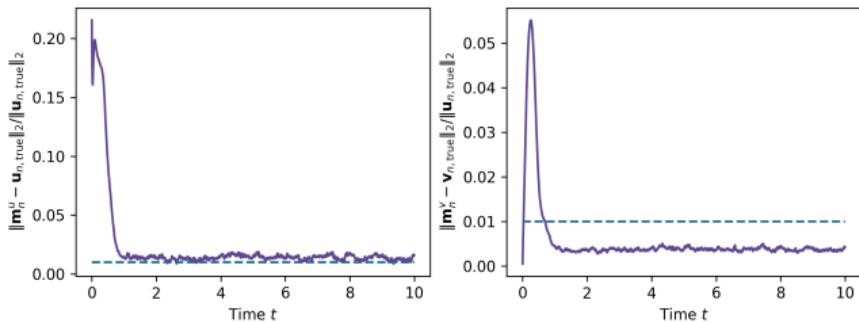


Leading-order covariance modes for the spiral-wave problem \implies variance decomposes surrounding the main spiral feature.

Case Study: Reaction-Diffusion

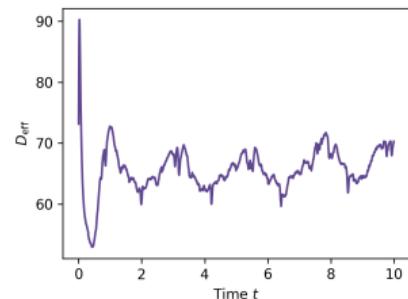
Relative errors show the **data generating process** is tracked:

$$\frac{\|\mathbf{m}_n^u - \mathbf{u}_n^{\text{DGP}}\|_2}{\|\mathbf{u}_n^{\text{DGP}}\|_2}, \quad \frac{\|\mathbf{m}_n^v - \mathbf{v}_n^{\text{DGP}}\|_2}{\|\mathbf{v}_n^{\text{DGP}}\|_2}.$$



Effective rank of covariance:

$$D_{\text{eff}} = \frac{\sum_{j=1}^k \sqrt{\lambda_j}}{\sum_{j=1}^k \lambda_j}.$$



Reaction Diffusion: Takeaways

- Low-rank approximation scales the method to high dimensionalities, enabling application of the method to complex “real world” systems.
- Uncertainty quantification is again sensible, and initial condition misspecification is calibrated with data.
- Enables inference in systems which may be **partially observed**.

Unknown Indirect Observation Operator

What if observation operator is unknown?

- That is, what if $\mathbf{y}_n = \mathcal{G}_\theta(\mathbf{u}_n) + \varepsilon_n$, for some **learnable** function $\mathcal{G}_\theta(\cdot)$.
- Use neural nets to learn this embedding from unstructured data into known mechanistic description.
- Mechanistic information used to identify the embedding: **not** to learn approximations to **solution fields**.
- **Example:** process is recorded with video camera, multi-channel recordings are taken (e.g., audio data).

How can we synthesise the phenomena with the mechanistic representation when we do not have an observation model?

Unknown Indirect Observation Operator

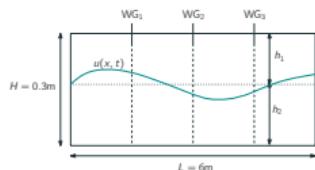
What if observation operator is unknown?

- That is, what if $\mathbf{y}_n = \mathcal{G}_\theta(\mathbf{u}_n) + \varepsilon_n$, for some **learnable** function $\mathcal{G}_\theta(\cdot)$.
- Use neural nets to learn this embedding from unstructured data into known mechanistic description.
- Mechanistic information used to identify the embedding: **not** to learn approximations to **solution fields**.
- **Example:** process is recorded with video camera, multi-channel recordings are taken (e.g., audio data).

How can we synthesise the phenomena with the mechanistic representation when we do not have an observation model?

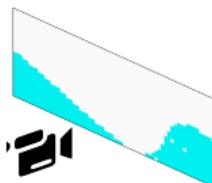
Unknown observation operator: some examples

Measurement



Observed internal wave.

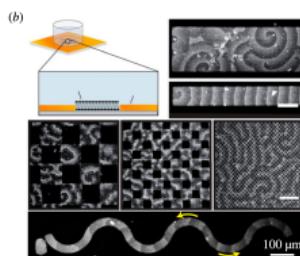
Phenomena



Mechanistic Representation

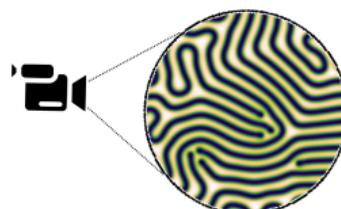
Example: Korteweg-de Vries equation:

$$\partial_t u + \alpha uu_x + \beta u_{xxx} + cu_x = 0.$$



Observed species concentrations.

Example: Gray-Scott equation:

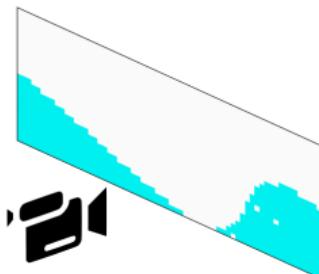


$$\partial_t u = D_u \nabla^2 u - uv^2 + F(1 - u),$$

$$\partial_t v = D_v \nabla^2 v + uv^2 - (F + k)v.$$

Unknown Observation Operator

Phenomena



Mechanistic Representation

KdV equation:

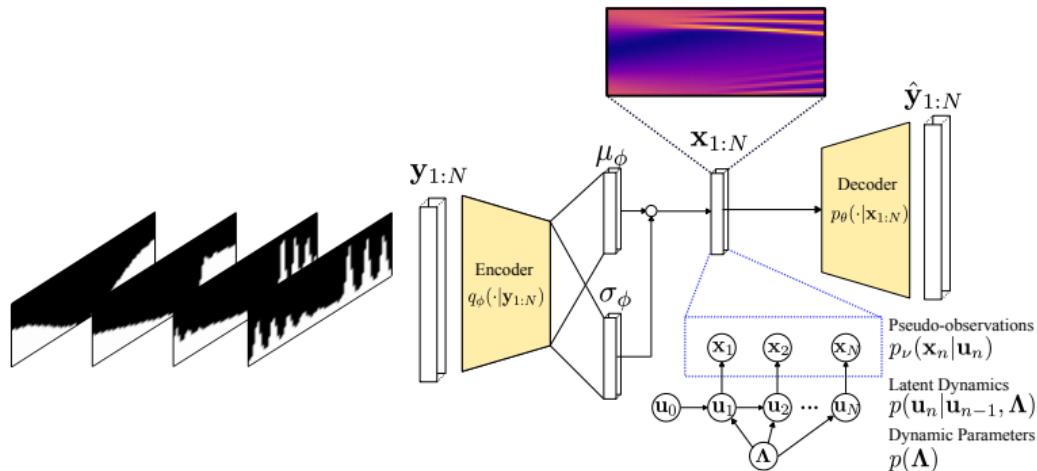
$$\partial_t u + \alpha uu_x + \beta u_{xxx} + cu_x = 0.$$

- Observation operator can be **approximated** with deep neural networks.
- We *posit* an observation operator of the form:

$$p(\mathbf{y}_n | \mathbf{u}_n) = \mathcal{N}(\mathcal{G}_\phi(\mathbf{u}_n), \mathbf{R}), \quad \mathcal{G} : \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y \times n_c}.$$

- Learn this embedding of the data to observations of the mechanistic system in a **variational inference framework**.

- Phenomenological data received through time: $\mathbf{y}_{1:N}$ (e.g., video feeds).
- Encoded to latent mechanistic observations $\mathbf{x}_{1:N}$ using a **variational autoencoder (VAE)** (Kingma and Welling).
- Mechanistic representation embedded into latent space, driving latent stochastic dynamics with statFEM.



⁸Alex Glyn-Davies et al. *Φ-DVAE: Physics-Informed Dynamical Variational Autoencoders for Unstructured Data Assimilation*, July 2023. arXiv: 2209.15609

ϕ -DVAE: Probabilistic Model Structure

We propose the following hierarchical probabilistic model:

- Parameter prior: $\Lambda \sim p(\Lambda)$.
- Transition density: $\mathbf{u}_n | \mathbf{u}_{n-1}, \Lambda \sim p(\mathbf{u}_n | \mathbf{u}_{n-1}, \Lambda)$ (assumed known form).
- Pseudo-observations: $\mathbf{x}_n | \mathbf{u}_n \sim p(\mathbf{x}_n | \mathbf{u}_n)$ (assumed known form).
- Decoder distribution: $\mathbf{y}_n | \mathbf{x}_n \sim p_\theta(\mathbf{y}_n | \mathbf{x}_n)$.

Following VAEs, we also introduce the “encoder” variational approximation,
 $q_\phi(\mathbf{x}_{1:N} | \mathbf{y}_{1:N}) = \mathcal{N}(\mu_\phi, \sigma_\phi)$, and the parameter posterior $p(\Lambda | \mathbf{y}_{1:N}) \approx q_\lambda(\Lambda)$.

How can we conduct joint parameter inference over $\{\Lambda, \theta, \phi\}$?

ϕ -DVAE: Probabilistic Model Structure

We propose the following hierarchical probabilistic model:

- Parameter prior: $\Lambda \sim p(\Lambda)$.
- Transition density: $\mathbf{u}_n | \mathbf{u}_{n-1}, \Lambda \sim p(\mathbf{u}_n | \mathbf{u}_{n-1}, \Lambda)$ (assumed known form).
- Pseudo-observations: $\mathbf{x}_n | \mathbf{u}_n \sim p(\mathbf{x}_n | \mathbf{u}_n)$ (assumed known form).
- Decoder distribution: $\mathbf{y}_n | \mathbf{x}_n \sim p_\theta(\mathbf{y}_n | \mathbf{x}_n)$.

Following VAEs, we also introduce the “encoder” variational approximation,
 $q_\phi(\mathbf{x}_{1:N} | \mathbf{y}_{1:N}) = \mathcal{N}(\mu_\phi, \sigma_\phi)$, and the parameter posterior $p(\Lambda | \mathbf{y}_{1:N}) \approx q_\lambda(\Lambda)$.

How can we conduct **joint parameter inference** over $\{\Lambda, \theta, \phi\}$?

ϕ -DVAE: Variational Lower Bound

Encoding, decoding, and model parameters are all jointly learnt through optimising a variational lower bound.

Evidence lower bound provides a tractable target for optimisation:

$$\log p(\mathbf{y}_{1:N}) \geq \mathbb{E}_{q_\phi(\mathbf{x}_{1:N}|\mathbf{y}_{1:N})} \left[\log \frac{p_\theta(\mathbf{y}_{1:N}|\mathbf{x}_{1:N})}{q_\phi(\mathbf{x}_{1:N}|\mathbf{y}_{1:N})} + \mathbb{E}_{q_\lambda} \left[\log p(\mathbf{x}_{1:N} | \Lambda) + \log \frac{p(\Lambda)}{q_\lambda(\Lambda)} \right] \right].$$

First term: encoder/decoder. Second term: pseudo-observations

$$p(\mathbf{x}_{1:N} | \Lambda) = \int p(\mathbf{u}_{1:N}, \mathbf{x}_{1:N} | \Lambda) d\mathbf{u}_{1:N}.$$

Marginalising over the dynamics acts as a “physics informed regulariser”. Third term: variational parameter posterior KL divergence.

Φ -DVAE: Simulation Studies

- We now go through a **selection of simulation studies** using Φ -DVAE.
- We look at (variational) parameter inference and filtering inference, $p(\mathbf{u}_{1:n}|\mathbf{x}_{1:n})$.
- We look at 2 particular systems: the classic Lorenz-63 system, and the (hopefully, now-familiar) KdV equation.
- We simulate **synthetic data** consisting of velocity fields, for the Lorenz-63 case, and video data, for the KdV case. These are our $\mathbf{y}_{1:N}$.
- We aim to learn the mapping from $\mathbf{y}_{1:N} \rightarrow \mathbf{x}_{1:N}$, thus **inferring the latent state** \mathbf{u}_n , conditioned on $\mathbf{y}_{1:n}$.

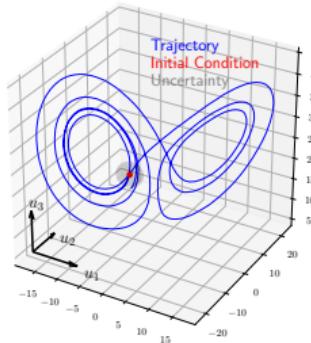
Lorenz-63 Dynamical System: Illustrative Example

Data $\mathbf{y}_{1:N}$ are simulated velocity field measurements, which are modulated by the first dimension of a latent stochastic Lorenz-63 system:

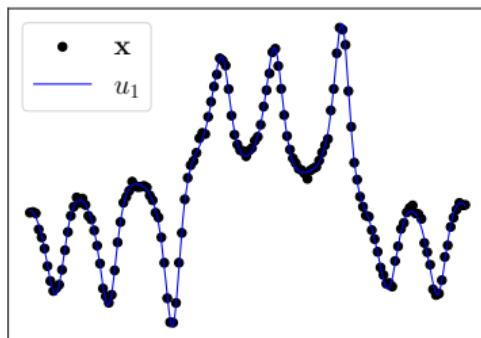
$$du_1 = -\sigma u_1 + \sigma u_2 + dw_1, \quad du_2 = -u_1 u_3 + ru_1 - u_2 + dw_2, \quad du_3 = u_1 u_2 - bu_3 + dw_3,$$

so now $\Lambda = \sigma$, $p(\Lambda) = \mathcal{N}(30, 5^2)$, and $q_\lambda(\Lambda) = \mathcal{N}(\mu_\lambda, \sigma_\lambda^2)$.

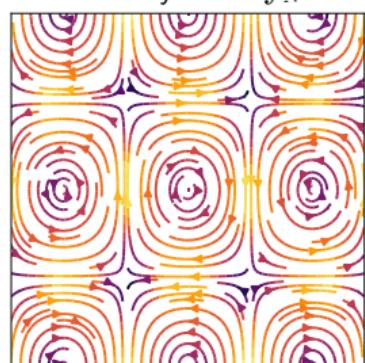
Latent Dynamics: $\mathbf{u}_{1:N}$



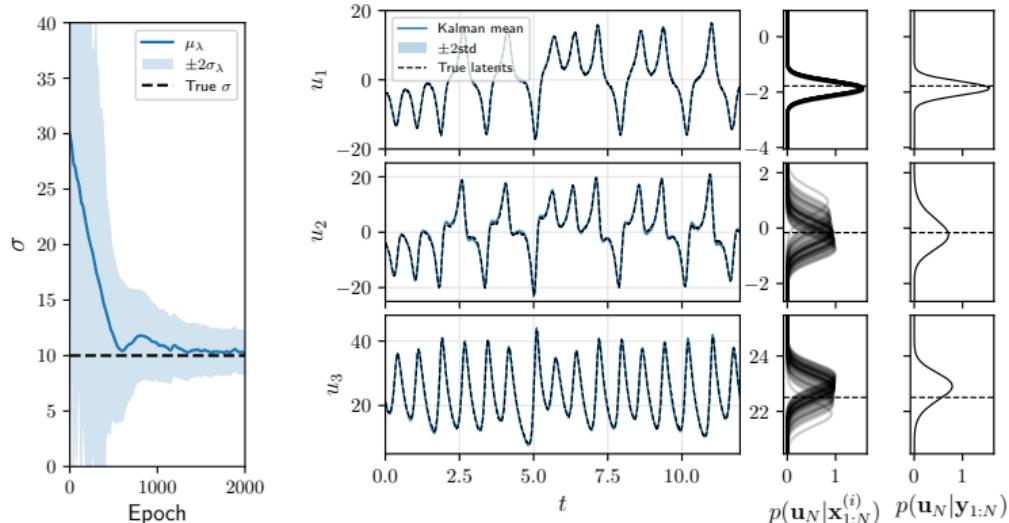
Pseudo-Observations: $\mathbf{x}_{1:N}$



Velocity Field: \mathbf{y}_N

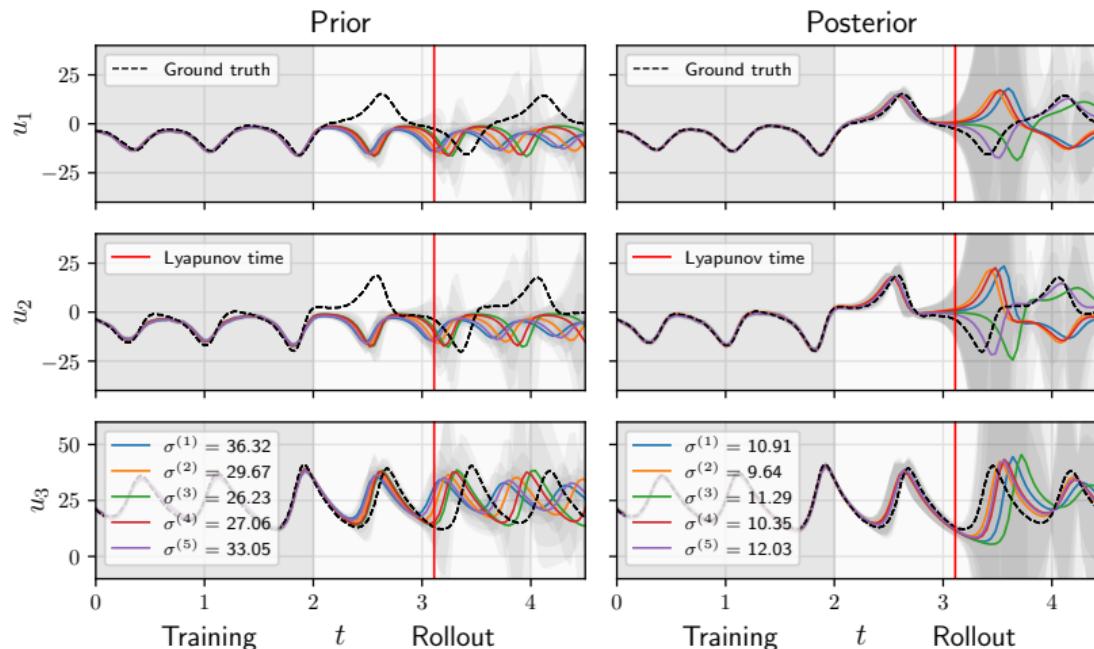


Lorenz-63: State and Parameter Inference



Left: “trace plot” of parameter variational distribution $q_\lambda(\Lambda) = \mathcal{N}(\mu_\lambda, \sigma_\lambda^2)$, with mean (blue) and ± 2 standard deviations (blue fill). Right: filtering inference for latent states $\mathbf{u}_{1:N}$, where the filtering distribution $p(\mathbf{u}_n | \mathbf{x}_{1:n})$ is plotted with the ground truth $\mathbf{u}_n^{\text{true}}$.

Lorenz-63: Rolling Out Beyond Training



“Rollout”: training time indicated with grey-fill, with (left) showing samples generated with the prior (left), and the posterior (right) variational distribution $q_\lambda(\cdot)$.

KdV: Learning the Observation Operator and Drag Coefficient

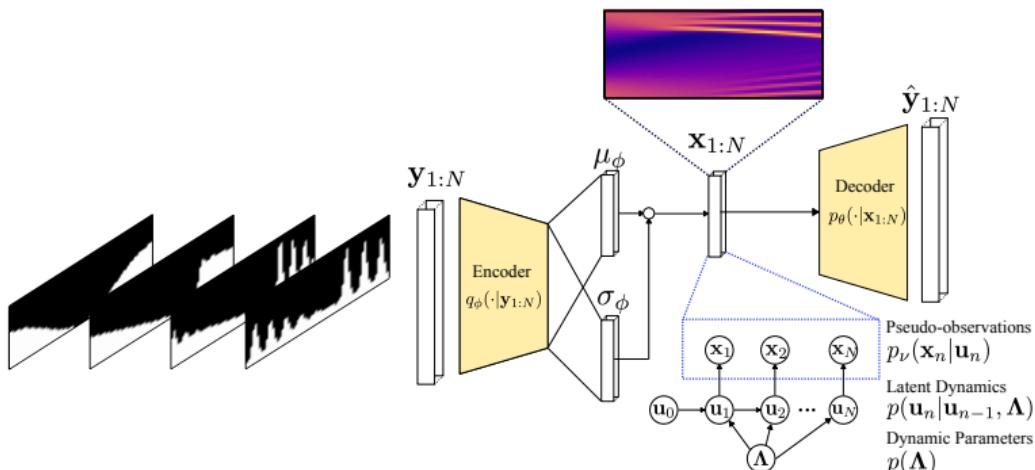
- In this final example we return to KdV: we generate **synthetic video data** (a sequence of images), giving our $\mathbf{y}_{1:N}$, from a governing KdV equation:

$$\partial_t u + \alpha uu_x + \beta u_{xxx} + \nu u = \xi_\theta.$$

We **jointly estimate** the embedding and the drag coefficient ν , so $\Lambda = \nu$, $p(\Lambda) = \mathcal{LN}(2, 0.5^2)$, $q_\lambda(\Lambda) = \mathcal{LN}(\mu_\lambda, \sigma_\lambda^2)$.

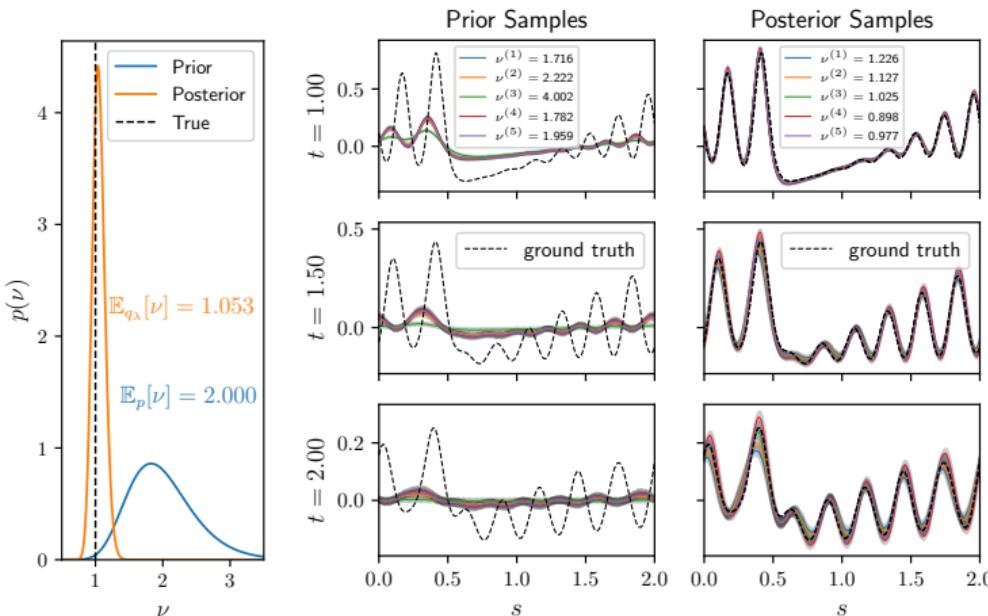
- Weakly-informative **log-normal prior** for the drag coefficient as $\nu > 0$.
- Encoding and decoding networks are MLPs with 3 hidden layers of width 128.

KdV: Learning the Observation Operator and Drag Coefficient



A reminder: video frames $\mathbf{y}_{1:N}$ are encoded to **pseudo-observations** $\mathbf{x}_{1:N}$ of a latent dynamical system with a known transition density $p(\mathbf{u}_n | \mathbf{u}_{n-1}, \Lambda)$. **Φ -DVAE infers** the encoder $q_\phi(\cdot)$, the decoder $p_\theta(\cdot | \mathbf{y}_{1:N})$, and parameters $q_\lambda(\cdot)$.

KdV: Results with drag coefficient estimation



Results for KdV with drag: (left) comparison of prior and variational posterior for model parameter $\nu = 1$. Right: latent filtering distribution for prior and posterior parameter estimates.

Conclusions

StatFEM synthesises data and FEM models via the posterior $p(\mathbf{u}_n | \mathbf{y}_{1:n}, \Lambda)$.

- Works in nonlinear, time-dependent models, enabling **interpolation** and **inference** in sparse data settings.
- Quantifies uncertainty and **robust** to **model misspecification**.
- Demonstrated through structural mechanics, fluid dynamics, nonlinear oscillators, and machine learning extensions.

More broadly: statFEM broadly provides a **fundamental methodology** to underpin modern digital twin models; which is derived from a statistically coherent construction.



All code on Github!

References i

Acknowledging content from Prof. Steve Niederer and Prof. David Wagg.

-  Akyildiz, Ömer Deniz, et al. "Statistical Finite Elements via Langevin Dynamics". *SIAM/ASA Journal on Uncertainty Quantification* 10, no. 4 (Dec. 2022): 1560–1585. <https://doi.org/10.1137/21M1463094>.
-  Duffin, Connor, et al. "Low-Rank Statistical Finite Elements for Scalable Model-Data Synthesis". *Journal of Computational Physics* 463 (Aug. 2022).
<https://doi.org/10.1016/j.jcp.2022.111261>.
-  – . "Statistical Finite Elements for Misspecified Models". *Proceedings of the National Academy of Sciences* 118, no. 2 (Jan. 2021).
<https://doi.org/10.1073/pnas.2015006118>.
-  Febrianto, Eky, et al. "Digital Twinning of Self-Sensing Structures Using the Statistical Finite Element Method". *Data-Centric Engineering* 3 (2022): e31.

References ii

-  Glyn-Davies, Alex, et al. *Φ -DVAE: Physics-Informed Dynamical Variational Autoencoders for Unstructured Data Assimilation*, arXiv:2209.15609, July 2023.
<https://doi.org/10.48550/arXiv.2209.15609>. arXiv: 2209.15609 [physics, stat].
-  Kennedy, Marc C., and Anthony O'Hagan. "Bayesian Calibration of Computer Models". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63, no. 3 (Aug. 2001): 425–464. <https://doi.org/10.1111/1467-9868.00294>.
-  Kingma, Diederik P., and Max Welling. *Auto-Encoding Variational Bayes*, arXiv:1312.6114, May 2014. <https://doi.org/10.48550/arXiv.1312.6114>. arXiv: 1312.6114 [cs, stat].
-  Masarotto, Valentina, et al. "Procrustes Metrics on Covariance Operators and Optimal Transportation of Gaussian Processes". *Sankhya A* 81, no. 1 (Feb. 2019): 172–213. <https://doi.org/10.1007/s13171-018-0130-1>.

References iii

-  Papandreou, Yanni, et al. "Theoretical Guarantees for the Statistical Finite Element Method". *SIAM/ASA Journal on Uncertainty Quantification* 11, no. 4 (Dec. 2023): 1278–1307. <https://doi.org/10.1137/21M1463963>.