
Flexible and Efficient Probabilistic PDE Solvers through Gaussian Markov Random Fields

Tim Weiland

Marvin Pförtner

Philipp Hennig

University of Tübingen, Tübingen AI Center

Abstract

Mechanistic knowledge about the physical world is virtually always expressed via partial differential equations (PDEs). Recently, there has been a surge of interest in probabilistic PDE solvers—Bayesian statistical models mostly based on Gaussian process (GP) priors which seamlessly combine empirical measurements and mechanistic knowledge. As such, they quantify uncertainties arising from e.g. noisy or missing data, unknown PDE parameters or discretization error by design. Prior work has established connections to classical PDE solvers and provided solid theoretical guarantees. However, scaling such methods to large-scale problems remains a challenge, primarily due to dense covariance matrices. Our approach addresses the scalability issues by leveraging the Markov property of many commonly used GP priors. It has been shown that such priors are solutions to stochastic PDEs (SPDEs), which, when discretized, allow for highly efficient GP regression through sparse linear algebra. In this work, we show how to leverage this prior class to make probabilistic PDE solvers practical, even for large-scale nonlinear PDEs, through greatly accelerated inference mechanisms. Additionally, our approach also allows for flexible and physically meaningful priors beyond what can be modeled with covariance functions. Experiments confirm substantial speedups and accelerated convergence of our physics-informed priors in nonlinear settings.

1 INTRODUCTION

In the physical sciences, the laws of nature are mostly expressed through partial differential equations (PDEs). A PDE describes how a function, modelling a quantity of interest, changes over space and time through its partial derivatives. Due to their expressive power, PDEs have found applications in many scientific domains, e.g. climate and weather forecasting (Bauer et al., 2015), oceanography (Chandrasekar, 2022), continuum mechanics (Reddy, 2013) and electromagnetics (Šolín, 2005). As closed-form solutions are typically unavailable, applications employ numerical PDE solvers which compute approximate solutions.

Both in the machine learning and scientific domain communities, there has been an increasing interest in physics-informed machine learning methods (Karniadakis et al., 2021) which learn not only from direct measurements, but also from mechanistic knowledge codified in PDEs. A common promise of such methods is that they predict accurate solutions to PDEs at a fraction of the cost of classic solvers. However, recent worrying reproducibility problems with deep-learning based simulation methods show that a careful connection between classic PDE solvers and machine learning methodology is urgently needed (McGreiv and Hakim, 2024).

Nevertheless, neither classical nor most machine-learned simulation methods handle **uncertainty** gracefully. In real-world applications, simulations of physical systems are subject to various types of uncertainties: (i) Sensors produce noisy, sparse, and potentially incomplete measurements of the system state. (ii) The parameters of the PDE are hardly ever known exactly. (iii) The solution is only an approximated and typically discretized.

Probabilistic PDE solvers are probabilistic numerical methods (Hennig et al., 2022) that treat such uncertainties fundamentally by interpreting the approximation of a PDE solution as a Bayesian inference problem. The most prominent approach is based on Gaussian process priors that are conditioned to solve

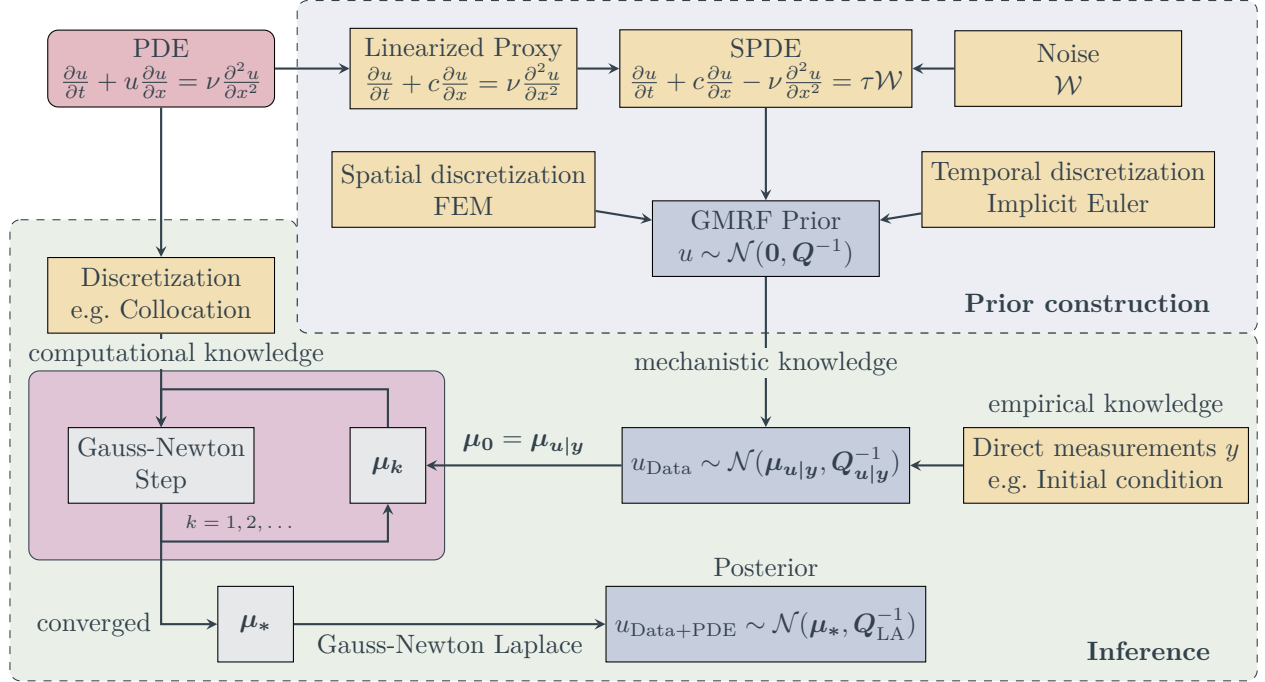


Figure 1: A flowchart of the high-level steps we propose for prior construction and inference. This conceptual sketch uses Burgers’ equation as a concrete example for a simple nonlinear PDE.

a discretization of the PDE. Under specific design choices, these methods exactly reproduce classic numerical solvers in the posterior mean (Pfortner et al., 2022), thus inheriting the convergence guarantees of these methods.

This provides a strong theoretical argument for probabilistic PDE solvers over less structured machine-learned surrogates. However, high computational cost has so far restricted the practical value of general probabilistic solvers. The underlying reason for this is that previous, especially theoretically motivated works, strive for formal generality of the probabilistic framework, to maximize the quantification of discretization error, parameter uncertainty, and noise in the data. Lowering the computational cost of probabilistic PDE solvers to be more in line with that of classic solvers requires a trade-off between flexible and slow, or restrictive and fast models and algorithms. In an example of this direction, some prior works on scalable probabilistic solvers are based on restrictive prior assumptions, e.g. tensor product covariance functions (Weiland et al., 2024; Wang et al., 2021a). These lower cost, but severely limit the expressive power of the prior.

Our central contribution is to equip probabilistic PDE solvers with faster inference mechanisms under flexible, physics-informed priors. The foundation of our method is an approach widely

used in spatial statistics: Compact basis function discretizations of stochastic PDEs (SPDEs) yield finite-dimensional approximations of stochastic processes with sparse precision matrices, enabling substantial computational gains (Lindgren et al., 2011). Our proposed method consists of two stages, which are summarized in Fig. 1. First, we construct a prior: In Section 2, we demonstrate how existing approaches based on Gaussian process priors inspire finite-dimensional approximations. This will then naturally lead to the idea of using physics-informed correlation models as a strong alternative to widely used stationary models. Having constructed a prior, we then revisit in Section 3 the general notion of probabilistic numerics, to phrase Bayesian inference on PDE discretizations as merely yet another source of data, analogous to statistical measurements, and show how to concretely realise it in our specific setting. As we will outline, this viewpoint enables highly efficient inference mechanisms for spatiotemporal, nonlinear PDEs. Finally, we validate our method empirically in Section 5.

2 PRIOR CONSTRUCTION

2.1 GP-based PDE solvers

As a starting point, we briefly outline the approach that is currently used to design probabilistic PDE solvers. In spatiotemporal settings, an initial bound-

ary value problem (IBVP) consists of a PDE along with initial and boundary conditions. Consider as an example an IBVP based on Burgers' equation:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad (1)$$

$$u(0, x) = \phi(x), \quad (2)$$

$$u(t, 0) = u(t, 1), \quad (3)$$

where Eq. (2) with $\phi : [0, 1] \rightarrow \mathbb{R}$ constitutes the initial condition and Eq. (3) the (periodic) boundary condition.

Probabilistic PDE solvers frame the problem of finding the solution to Eqs. (1) to (3) as a Bayesian inference problem by positing a Gaussian process (GP) prior $u \sim \mathcal{GP}(\mu, k)$ over the unknown solution function. The prior mean μ and covariance function k model prior knowledge about the solution. Observe that discretizing the initial condition Eq. (2) simply yields direct observations of the solution, which may be integrated with classic GP regression. This yields a posterior $u^* \sim \mathcal{GP}(\mu^*, k^*)$ with covariance

$$k^*(z_1, z_2) = k(z_1, z_2) - k(z_1, \mathbf{X}) (k(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I})^{-1} k(\mathbf{X}, z_2), \quad (4)$$

with discretization points $\mathbf{X} \in \mathbb{R}^{N \times 2}$ ($N \in \mathbb{N}$) and measurement noise $\sigma^2 > 0$.

In fact, GPs are not only closed under point observations, but—under mild assumptions—also under other linear transformations (Pförtner et al., 2022). The resulting posterior covariance has the same structure as Eq. (4), but in the downdate term occurrences of \mathbf{X} are replaced with applications of the linear operator (see Pförtner et al., 2022, Section 4). Differentiation is a linear operation, and thus, if the PDE of interest is linear, the posterior under a PDE discretization is available in closed form. In the nonlinear case, discretizations may be formulated as nonlinear transformations of linear operations. The posterior is then generally no longer a GP, but it may be approximated, e.g. through a Laplace approximation (Chen et al., 2024).

Practical considerations. These properties make GPs theoretically appealing for obtaining approximate PDE solutions under statistical data. In practice however, we encounter two issues:

1. **The expression of prior knowledge** is restricted to what can be modelled tractably with covariance functions. A standard choice is the Matérn covariance function.
2. **Scalability** of these models is impeded significantly by the (typically dense) matrix inversion

on the right-hand side of Eq. (4). Approaches to circumvent these issues exist, but may require even stronger prior assumptions such as separable covariance functions.

Gauss-Markov priors and SDEs. In the context of probabilistic ODE solvers, the solution to these problems is the use of Markovian priors which are solutions to linear SDEs (Tronarp et al., 2019). This enabled the use of filtering and smoothing algorithms for linear-time inference. Unfortunately, a direct transfer of these ideas to spatiotemporal processes is not possible. However, in the following we leverage the same underlying ideas: We consider multivariate Markov priors which are solutions to linear SPDEs and demonstrate the benefits of this viewpoint for probabilistic PDE solvers.

2.2 SPDE Priors

The SPDE approach, as introduced by Lindgren et al. (2011), is motivated by the example of the Matérn process. It is based on the insight that a Gaussian process with Matérn covariance function is the stationary solution of the Whittle-Matérn SPDE (Whittle, 1954)

$$\underbrace{(\kappa^2 - \Delta)^{\alpha/2}}_{=: \mathcal{D}} u(\mathbf{x}) = \mathcal{W}(\mathbf{x}) \quad (5)$$

on \mathbb{R}^d , where \mathcal{W} is a Gaussian white noise process, $\kappa > 0, \nu > 0, \alpha = \nu + d/2$ and Δ is the Laplacian operator.

In practice, the solution of the SPDE is approximated by a parametric Gaussian process $u(\mathbf{x}) \approx \sum_{j=1}^N u_j \phi_j(\mathbf{x})$ with compactly supported basis / feature functions ϕ_j and $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^{-1})$. To solve for the corresponding precision matrix \mathbf{Q} , the SPDE is typically discretized using the finite element method (FEM), which results in a linear system

$$\mathbf{K} \mathbf{u} = \mathbf{w} \quad (6)$$

with stochastic right-hand side $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{M})$. The system matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ is given by the stiffness matrix corresponding to \mathcal{D} and $\{\phi_j\}_{j=1}^N$ and $\mathbf{M} \in \mathbb{R}^{N \times N}$ is the mass matrix corresponding to the basis functions. Since the basis functions ϕ_j have compact support, both \mathbf{K} and \mathbf{M} are sparse. The solution of the linear system is hence given by

$$\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \underbrace{\mathbf{K}^{-1} \mathbf{M} \mathbf{K}^{-T}}_{=\mathbf{Q}^{-1}}). \quad (7)$$

In FEM, it is common to approximate the mass matrix \mathbf{M} by a diagonal matrix $\tilde{\mathbf{M}}$ through mass lumping. This results in a sparse precision matrix

$$\mathbf{Q} \approx \mathbf{K}^T \tilde{\mathbf{M}}^{-1} \mathbf{K}. \quad (8)$$

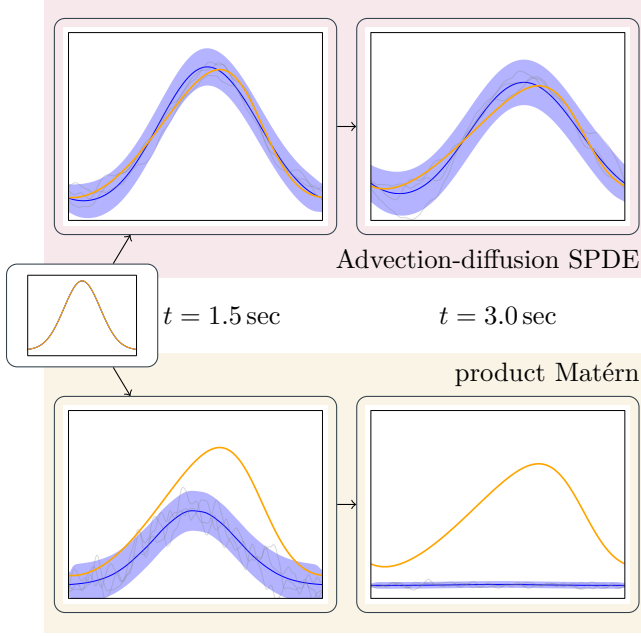


Figure 2: **Advection-diffusion SPDE priors provide physical information.** Starting from the same initial condition, the top row shows the analytic evolution of the advection-diffusion SPDE prior for Burgers’ univariate equation at $t = 1.5$ sec and $t = 3.0$ sec. For comparison the bottom row shows an analogous stationary product Matérn prior used in prior work. Both rows compare to the ground-truth dynamics of a nonlinear Burgers’ equation in orange.

Due to the sparsity of its precision matrix, \mathbf{u} is called a Gaussian Markov Random Field (GMRF). GMRF inference is quite efficient, which has lead to widespread use in spatial statistics (Lindgren et al., 2022).

2.3 Physics-Informed Priors

GMRFs are able to approximate Matérn processes, which are a common prior for probabilistic PDE solvers. While the computational benefits of GMRFs provide a strong motivation for their use (more in Section 3), one may still wonder whether a generic Matérn prior is the optimal choice for a specific probabilistic PDE solver.

Priors in terms of SPDE dynamics. The Matérn process follows the dynamics of the Whittle-Matérn SPDE. The dynamics of this SPDE result in posteriors that revert towards the mean as the distance to the data grows, at an effective range and smoothness controllable by κ and ν , respectively. This is a sensible prior for pure regression settings where no mechanistic knowledge about the function to model is available. However, this is clearly not the case for the applica-

tion of PDE solvers. Figure [...] shows the effect of a separable Matérn prior for a Burgers’ equation IBVP: The effect of the initial condition simply drops off over time, causing a substantial mismatch to the true dynamics of the problem.

Philosophically, a good prior should include all prior knowledge available about the function of interest. But this perspective is insufficient for the design of good numerical methods since it does not account for computational cost (Hennig et al., 2022, §9.3). We know a priori that the function of interest fulfills the PDE, yet encoding this knowledge into the prior incurs computational cost. In practice, the expressive power of a prior is thus necessarily always limited by computational resources.

Beyond the Matérn. The SPDE formulation introduced above actually allows for a broader perspective than just stationary Matérn processes. At its core, it forms priors as solutions to linear SPDEs. For the same discretization resolution, a generic Matérn prior is then comparable in computation time to *any* prior based on arbitrary *other*, but still *linear* SPDEs. We thus propose to construct priors which more closely capture the dynamics of the target (nonlinear) PDE solution through **linear stochastic proxies**, the probabilistic prior class analogous to linearization of nonlinear PDEs, a concept well-known in the context of numerical PDE solvers.

To exemplify this idea, we return to our running example of Burgers’ equation. Burgers’ equation exhibits linear diffusion and **nonlinear** advection dynamics:

$$\frac{\partial u}{\partial t} + \underbrace{u \frac{\partial u}{\partial x}}_{\text{Advection}} = \underbrace{\nu \frac{\partial^2 u}{\partial x^2}}_{\text{Diffusion}}. \quad (9)$$

Define the average initial function value $c := \int_0^1 u(0, x) dx$ and consider the linear PDE

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}. \quad (10)$$

This is a linear proxy for Burgers’ equation that captures the diffusion and the bulk advection. If we further introduce a stochastic forcing term \mathcal{W} scaled by a parameter $\tau \in \mathbb{R}$, we obtain the advection-diffusion SPDE

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} - \nu \frac{\partial^2 u}{\partial x^2} = \tau \mathcal{W}, \quad (11)$$

which we can use to form a GMRF prior. Figure 2 shows that this prior indeed captures the general dynamics of the true solution much more effectively.

Choosing a linear proxy. The success of the above approach relies on our ability to find a linear proxy that approximates the dynamics of the nonlinear PDE

of interest well. Linearization of nonlinear PDEs is well-studied in physics. For example: In fluid dynamics, the linearized shallow-water equations find common use (Vallis, 2017); Monge-Ampère equations may be solved numerically by iteratively solving (linear) Poisson equations (Benamou et al., 2010), and so on. Our suggested approach is not limited to any specific linearization technique. Rather, practitioners may use their domain knowledge to flexibly choose an appropriate proxy for a specific problem.

Boundary conditions. The FEM discretization of the SPDE involves the specification of boundary conditions. In spatial statistics, the resulting boundary effects are often undesirable, and a common approach to avoid these effects is to artificially inflate the problem domain (Lindgren et al., 2011). For a probabilistic PDE solver however, this enables us to embed the boundary condition of the IBVP of interest directly into the prior, as long as the condition is linear. When assembling the stiffness and mass matrices, we eliminate the corresponding degrees of freedom from the system as in classic FEM. Having constructed the precision matrix, we may then tune the certainty of the boundary condition through the diagonal entries corresponding to these degrees of freedom. For more details, refer to Section B.

2.4 Spatiotemporal Models

For purely spatial PDEs, the basis function discretization can be chosen analogously to classic FEM solvers. In spatiotemporal settings however, the GMRF approach differs in the sense that time stepping is not possible and the entire spacetime domain needs to be discretized at once. Conceptually, time may be treated in the same way as space, and thus a FEM basis may also be used along the time dimension. In practice, we find that this adds implementation overhead as existing FEM software is not designed for this use case.

Implicit Euler discretization. Instead, we suggest to use an implicit Euler discretization along time as described by Clarotto et al. (2024) for the special case of an advection-diffusion SPDE. Following Rothe’s method, we first discretize along time through implicit Euler and then discretize the spatial component via FEM. This leads to a discrete state-space model of the form

$$\underbrace{(M + \Delta t K)}_{=:G} \mathbf{u}^{(k+1)} = M \mathbf{u}^{(k)} + \sqrt{\Delta t} M \Sigma^{1/2} \mathbf{z} \quad (12)$$

$$\Leftrightarrow \mathbf{u}^{(k+1)} = \underbrace{G^{-1} M}_{=:A} \mathbf{u}^{(k)} + \underbrace{\sqrt{\Delta t} G^{-1} M \Sigma^{1/2}}_{=:F^{1/2}} \mathbf{z}, \quad (13)$$

Eq. (13) represents a conditional distribution $\mathbf{u}^{(k+1)} |$

$\mathbf{u}^{(k)} \sim \mathcal{N}(A \mathbf{u}^{(k)}, F)$. In combination with an initial distribution $\mathbf{u}^{(0)} \sim \mathcal{N}(\mathbf{0}, Q^{(0)-1})$, we get the joint distribution

$$\begin{pmatrix} \mathbf{u}^{(0)} \\ \vdots \\ \mathbf{u}^{(T)} \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, Q_{ST}^{-1}), \quad (14)$$

where Q_{ST} is a block tridiagonal precision matrix. For a detailed derivation, refer to Section A.

3 INFERENCE MECHANISMS

Section 2 described how to construct customized priors suited for probabilistic PDE solvers. In the following, we describe how to perform Bayesian inference with PDE observations under these priors.

3.1 Affine conditioning

Information operators. In the context of GP-based PDE solvers, *information operators* (Cockayne et al., 2019) may be used to condition a GP prior on information about the PDE. Let $u \sim \mathcal{GP}(\mu, k)$ and \mathcal{L} a linear operator acting on the paths of u and taking values in \mathbb{R}^M . Under mild assumptions on u and \mathcal{L} , the posterior $u | \mathcal{L}[u] = \mathbf{b} \sim \mathcal{GP}(\mu_*, k_*)$ is again a GP with known closed forms for μ_* and k_* (Pförtner et al., 2022). For a PDE $\mathcal{D}[u] = f$ with a linear differential operator \mathcal{D} , we can then construct an information operator $\mathcal{I}_{\mathcal{L}, \mathbf{b}}[u] := (\mathcal{L} \circ \mathcal{D})[u] - \mathbf{b}$ and obtain a closed-form posterior $u | (\mathcal{I}_{\mathcal{L}, \mathbf{b}}[u] = \mathbf{0})$. The most common example is *collocation*: Choose $\mathbf{x} \in \mathbb{R}^d$ and set $\mathcal{L}[u] = u(\mathbf{x})$ and $\mathbf{b} = f(\mathbf{x})$. This choice ensures that the linear PDE is fulfilled *exactly* at the discrete collocation point x .

For general GPs, the closed form for the posterior under such information operators involves applying the linear operator to the covariance function. In our case, this is simplified through our use of a parametric GP $u(\mathbf{x}) = \sum_{j=1}^N u_j \phi_j(\mathbf{x})$, as

$$\begin{aligned} \mathcal{L}[u] &= \sum_{j=1}^N u_j \mathcal{L}[\phi_j] = L_\phi \mathbf{u}, \quad \mathcal{L}[u] = \sum_{j=1}^N u_j \mathcal{L}[\phi_j] \\ &= L_\phi \mathbf{u}, \end{aligned} \quad (15)$$

with $L_\phi \in \mathbb{R}^{M \times N}$ and $(L_\phi)_{ij} = \mathcal{L}[\phi_j]_i$, which means that the action of the linear operator on u is induced by its action on the basis functions.

Affine conditioning of GMRFs. In particular, this means that conditioning a GMRF representation on an information operator is equivalent to conditioning on a transformation under a matrix. Let $\mathbf{u} \sim \mathcal{N}(\mu, Q^{-1})$ be an arbitrary GMRF and let $A \in \mathbb{R}^{M \times N}$, $\mathbf{b} \in \mathbb{R}^M$. Under the likelihood $\mathbf{y} | \mathbf{u} \sim \mathcal{N}(A \mathbf{u} + \mathbf{b}, Q_\epsilon^{-1})$, we

obtain (Rue and Held, 2005):

$$\mathbf{u} \mid \mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_{u|y}, \mathbf{Q}_{u|y}^{-1}), \quad (16)$$

$$\mathbf{Q}_{u|y} = \mathbf{Q} + \mathbf{A}^T \mathbf{Q}_\epsilon \mathbf{A}, \quad (17)$$

$$\boldsymbol{\mu}_{u|y} = \boldsymbol{\mu} + \mathbf{Q}_{u|y}^{-1} \mathbf{A}^T \mathbf{Q}_\epsilon (\mathbf{y} - (\mathbf{A}\boldsymbol{\mu} + \mathbf{b})). \quad (18)$$

These equations are the key to efficient closed-form conditioning on information operator observations. Two key examples include

$$\textbf{Collocation: } A_{ij} := \mathcal{D}\phi_j(\mathbf{x}_i) \text{ and } b_i := f(\mathbf{x}_i) \quad (19)$$

$$\textbf{FEM: } A_{ij} := \int \langle \phi_i(\mathbf{x}), \mathcal{D}\phi_j(\mathbf{x}) \rangle d\mathbf{x}, \quad (20)$$

$$\text{and } b_i := \int \langle \phi_i(\mathbf{x}), f(\mathbf{x}) \rangle d\mathbf{x}. \quad (21)$$

Additionally, we highlight that classic regression also fits into this viewpoint through $A_{ij} := \phi_j(\mathbf{x}_i)$.

3.2 Moments and sampling

Efficient numerical linear algebra is a crucial component of state-of-the-art PDE solvers. In the following, we present how GMRFs as a model class naturally enable efficient algorithms for the computation of their key quantities. Whereas efficiency mostly remains an afterthought for alternative methods for probabilistic PDE solving, GMRF computations are efficient by design.

Sparse Cholesky decomposition. The precision matrix is sparse, symmetric and positive definite, and thus a sparse Cholesky decomposition $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$ may be computed. Through efficient node reorderings that minimize fill-in, such decompositions achieve runtime complexities ranging from $\mathcal{O}(N)$ for temporal models to $\mathcal{O}(N^2)$ for spatiotemporal models (Rue and Held, 2005). Once the Cholesky decomposition has been computed, computing the posterior mean in Eq. (18) requires one sparse forward and one sparse backward solve. Similarly, if $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, then $\boldsymbol{\mu} + \mathbf{L}^{-T}\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}^{-1})$, i.e. sampling requires one backward solve.

Conjugate gradient method. In situations where a sparse Cholesky decomposition is too expensive (due to fill-in), the conjugate gradient (CG) method may be used to solve linear systems iteratively. This is directly applicable to Eq. (18). To sample using CG, we assume that we have access to a left square root \mathbf{L} of $\mathbf{Q}_{u|y}$. Then, if $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we get $\mathbf{L}\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{u|y})$ and thus $\mathbf{Q}_{u|y}^{-1}\mathbf{L}\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{u|y}^{-1})$, which may be solved using CG. All models considered in this work have easily accessible left square roots: For Eq. (8) we use $\mathbf{L} := \mathbf{K}^T \tilde{\mathbf{M}}^{-1/2}$, and for the general spatiotemporal prior in Eq. (14) we derive a square root in Section A. For $\mathbf{Q} = \mathbf{A} + \mathbf{B}$ with $\mathbf{A} = \mathbf{L}_A \mathbf{L}_A^T$ and $\mathbf{B} = \mathbf{L}_B \mathbf{L}_B^T$, a left square root is given by $(\mathbf{L}_A \quad \mathbf{L}_B)$.

Marginal variance computation. The marginal variances are given by the diagonal of the covariance matrix, i.e. the inverse of the precision matrix. Marginal variances can be computed without full inversion of the matrix through the Takahashi recursion (Rue, 2005). While the Takahashi recursion is accurate, it also quickly becomes prohibitively expensive. An alternative is a Rao-Blackwellized Monte Carlo estimator of the variance described by Sidén et al. (2018), which uses the precision matrix to achieve reasonable accuracies even with relatively few samples.

3.3 Nonlinear PDEs

With the individual inference step set up, we can now finally treat the general setting of nonlinear PDEs. Here, we consider a nonlinear function $\mathbf{f} : \mathbb{R}^N \rightarrow \mathbb{R}^M$ which induces a likelihood $\mathbf{y} \mid \mathbf{u} \sim \mathcal{N}(\mathbf{f}(\mathbf{u}), \mathbf{Q}_\epsilon^{-1})$. For example, \mathbf{f} may evaluate the PDE residual at M collocation points. This yields a posterior density proportional to

$$\pi(\mathbf{u}) := \exp \left(-\frac{1}{2} (\mathbf{u} - \boldsymbol{\mu})^T \mathbf{Q} (\mathbf{u} - \boldsymbol{\mu}) - \frac{1}{2} (\mathbf{y} - \mathbf{f}(\mathbf{u}))^T \mathbf{Q}_\epsilon (\mathbf{y} - \mathbf{f}(\mathbf{u})) \right). \quad (22)$$

Then the mode is given by

$$\boldsymbol{\mu}_* := \arg \max_{\mathbf{u}} \pi(\mathbf{u}) = \arg \min_{\mathbf{u}} (-\log \pi(\mathbf{u})). \quad (23)$$

Gauss-Newton optimization. The right-hand side expression in Eq. (23) can be framed as a nonlinear least-squares problem and solved through Gauss-Newton optimization, which is a common technique also for classic PDE solvers. Each Gauss-Newton iteration involves a linear system solve with the Gauss-Newton matrix $\mathbf{H}^{(k)} := \mathbf{Q} + \mathbf{J}^{(k)T} \mathbf{Q}_\epsilon \mathbf{J}^{(k)} \approx -\nabla \nabla^T \log \circ \pi$, where $\mathbf{J}^{(k)} \in \mathbb{R}^{M \times N}$ is the Jacobian of \mathbf{f} in iteration $k \in \mathbb{N}$. Due to the compact basis function representation of \mathbf{u} , and because \mathbf{f} is composed of derivative information, the Jacobian is sparse and $\mathbf{H}^{(k)}$ generally has the same sparsity pattern as \mathbf{Q} . This enables the use of the sparse linear algebra techniques discussed above.

Acceleration techniques. In the sparse Cholesky setting, the node reordering only needs to be computed once and can be reused for all further iterations. In a CG setting, the Gauss-Newton direction computed in the previous iteration may be reused as an initial guess for CG. Independently of the linear solver, we employ a backtracking line search to stabilize the optimization.

Laplace approximation. After converging at a solution for the mode $\boldsymbol{\mu}_*$, we may then form a Gauss-

Newton Laplace approximation of Eq. (22) as

$$\mathbf{x}_* \sim \mathcal{N}\left(\boldsymbol{\mu}_*, (\mathbf{Q} + \mathbf{J}_*^T \mathbf{Q}_\epsilon \mathbf{J}_*)^{-1}\right). \quad (24)$$

To draw samples and compute variances, we employ the strategies described in Section 3.2.

4 RELATED WORK

GP-based PDE solvers There is a growing body of work on GP-based PDE solvers. These methods can be broadly categorized by the strategy used to discretize the PDE, their applicability to weak or strong form PDEs, and by their ability to solve nonlinear equations. Graepel (2003); Cockayne et al. (2017); Raissi et al. (2017) propose collocation-based methods for approximating strong solutions to linear PDEs. Owhadi (2015) casts numerical homogenization of linear weak-form PDEs as a Bayesian inference problem with a prior Gaussian random field. Pförtner et al. (2022) construct a general framework for GP-based solvers for linear PDEs in weak or strong form that is shown to generalize all classical methods of weighted residuals. Weiland et al. (2024) build a GP analogue of the finite volume method. GP-based methods for nonlinear PDEs using finite-difference discretization (Wang et al., 2021b; Krämer et al., 2022) and collocation (Chen et al., 2021, 2024) have also been proposed.

SPDE priors Our work is certainly not the first to explore alternative SPDE priors. Particularly in the context of spatiotemporal models, there has been an increasing interest in such models in recent years (Vergara et al., 2018; Lindgren et al., 2023; Clarotto et al., 2024). The work of Vergara et al. (2018) is of particular interest since it also sets a strong focus on physically inspired models. However, to the best of our knowledge, the relevance and ease of construction of such models in the context of probabilistic PDE solvers has not been explored in prior work.

StatFEM and BFEM The statistical finite element method (statFEM) (Girolami et al., 2021) is a framework for solving PDE-constrained inverse problems in a Bayesian fashion. Just like our work, it employs a FEM-discretized linear SPDE prior. In contrast to our work, this prior is then not conditioned on an information operator, but instead used to identify the parameters of the SPDE.

The Bayesian finite element method (BFEM) (Poot et al., 2024) aims to quantify the discretization error in the finite element method. It also uses a Gaussian prior defined by a linear SPDE. However, as opposed to our approach, it then proceeds with the inference procedure in covariance rather than in precision form.

5 EXPERIMENTS

In the following, we evaluate the scaling behavior of GMRF-based PDE solvers as well as the utility of our physics-informed priors.

Implementation. The code for our experiments is available on GitHub¹. It is based on our own implementation of a framework for GMRFs which utilizes the built-in sparse array routines of Julia².

Datasets. We test various aspects of our method on a variety of example problems. Li et al. (2021) introduced a benchmark to evaluate a neural operator. From this benchmark, we evaluate on 2048 different samples of a **2D Darcy Flow** problem with Dirichlet boundary conditions, which is a purely spatial PDE with highly discrete variable coefficients, as well as 30 different samples of a **1D Burgers’ equation** IBVP with periodic boundary conditions, which is a common benchmark to test the ability of PDE solvers to handle nonlinearities. Furthermore, to test the performance of our solver against a state-of-the-art probabilistic solver, we also run experiments on the **nonlinear elliptic equation** and the **1D Burgers’ equation** presented in Chen et al. (2024). Section C contains further details on the setup of the experiments.

5.1 Accuracy and Efficiency

On the 2D Darcy Flow problem, we start with two versions of a Matérn prior: The first follows the commonly used approach of an inflated boundary (see Section 2.3) and then manually conditions on the Dirichlet boundary conditions, while the second directly embeds the boundary conditions into the prior as detailed in Section 2.3. We then condition both priors on FEM discretizations of Darcy’s flow of different resolutions on the interior of the domain. To validate the hypothesis that our solvers reproduce the accuracy of classic methods and to check the computational overhead of our methods, we also compare to a FEM baseline which uses the same implementation for the assembly of the stiffness matrix.

The work-precision diagram in Fig. 4 demonstrates that GMRF-based solvers with FEM observations indeed reproduce the accuracy of classic FEM solvers, with only minute differences that may be attributed to the noisy observation scheme. More crucially, **the runtime of our GMRF-based solver is at the same order of magnitude as FEM**, with a natural overhead arising from added functionality (namely uncertainty quantification).

¹<https://github.com/timweiland/DiffEqGMRFs.jl>

²<https://github.com/timweiland/GaussianMarkovRandomFields.jl>

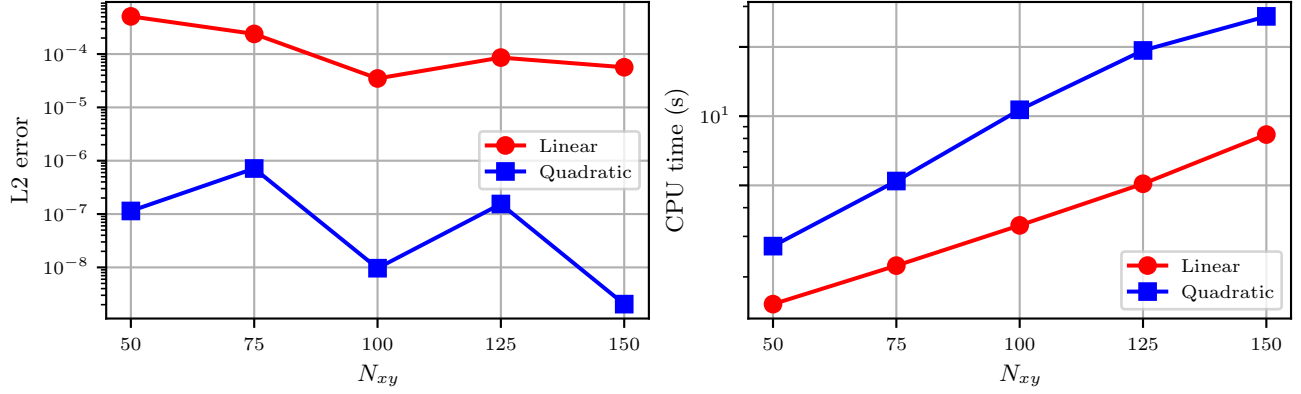


Figure 3: **Performance of our method on the nonlinear elliptic PDE from Chen et al. (2024).** We evaluate the L2 error and runtime of GMRF-FEM with linear and quadratic basis elements for varying mesh resolutions N_{xy} .

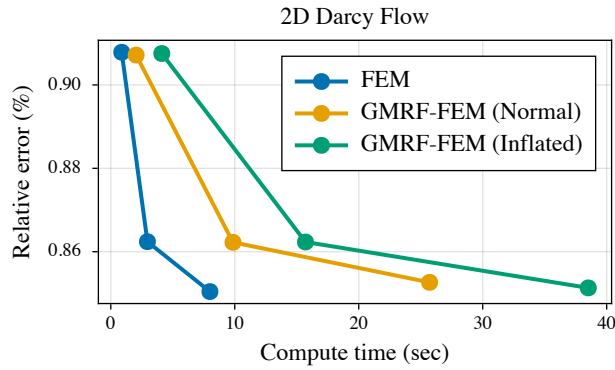


Figure 4: **GMRF-based methods equal classic FEM solvers in accuracy at reasonable computational overheads.** The plot shows the mean relative error across all problem instances for different discretization resolutions.

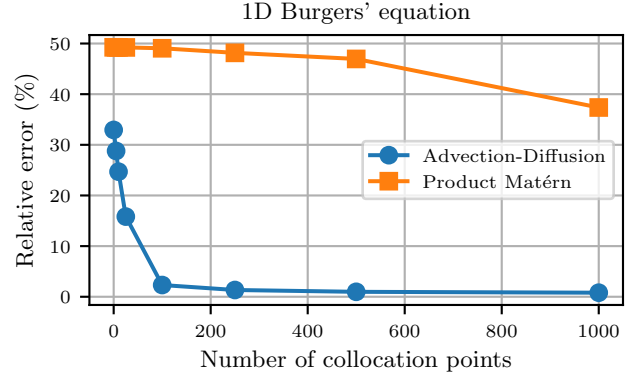


Figure 5: **Physics-informed priors yield faster convergence in terms of the discretization resolution.** The plot shows the mean relative error across all problem instances for different discretization resolutions for the Burgers' equation IBVPs.

Furthermore, we observe that there is no difference in accuracy between inflating the boundary and directly embedding the correct boundary conditions into the prior. However, naturally the inflation of the boundary creates additional degrees of freedom and thus also inflates the size of the precision matrix, increasing the solve time. For this reason, we find that **GMRF-based PDE solvers should—if possible—include the boundary conditions in the prior.**

5.2 Physics-Informed Priors

We validate the value of physics-informed priors on the Burgers' equation benchmark. To this end, we start with a separable Matérn prior as well as an advection-diffusion prior, both of which have an embedded periodic boundary. We then condition on the initial condition and perform Gauss-Newton optimization to solve

for nonlinear collocation observations of the PDE. The results are shown in Fig. 5. We observe that the accuracy for the advection-diffusion prior—which captures the rough dynamics of Burgers' equation—improves greatly even after only adding a few collocation points. By contrast, the separable Matérn prior requires many more collocation points to achieve a substantially improved accuracy. We attribute this to the behavior of the Matérn to revert to the mean, which causes the collocation points to have less of an effect. These findings validate our intuition that **physics-informed priors pay off by requiring fewer PDE observations to converge to an accurate posterior.**

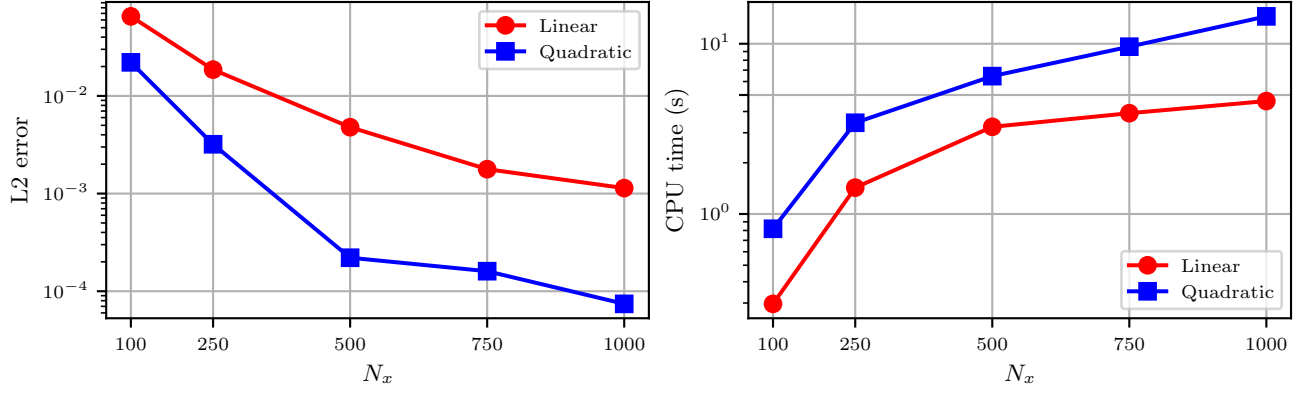


Figure 6: **Performance of our method on the Burgers' equation example from Chen et al. (2024).** We evaluate the L2 error and runtime of GMRF-FEM with linear and quadratic basis elements for varying mesh resolutions N_x .

5.3 Comparison to the state-of-the-art

In this section, we compare to a state-of-the-art probabilistic PDE solver introduced by Chen et al. (2024). Their method is based on finding sparse approximate inverse Cholesky factors to covariance matrices arising from GPs formulated in terms of a covariance function. By contrast, our method begins with an SPDE perspective, a discretization of which *naturally* yields precision matrices with a controllable sparsity.

Nonlinear elliptic equation. Chen et al. (2024) present a purely spatial nonlinear elliptic PDE, which we solve as before using FEM observations combined with Gauss-Newton optimization. Figure 3 shows that using quadratic elements, our method achieves a high accuracy of $1e-9$ L2 error in around 20 seconds. For comparison: The baseline requires a runtime on the order of hundreds of seconds to achieve an L2 error on the order of $1e-8$. Our method achieves the same order of error in 10 seconds. This improvement cannot be attributed to hardware differences, but rather to the fundamental design choices of our method, which induces sparsity by design and does not rely on potentially expensive approximate factorizations.

1D Burgers' equation. Additionally, Chen et al. (2024) evaluate their method on a 1D Burgers' equation. As before, we use an advection-diffusion prior. Along space, we use $N_x \in \mathbb{N}$ finite elements. We discretize along time using the same time step as the baseline of $\Delta t = 0.02$. For the sake of efficiency, we use an implicit Euler discretization to form the prior, and then switch to a Crank-Nicolson discretization for the actual FEM observations of the PDE, **highlighting the flexibility of our method.**

Figure 6 shows the results. Our method roughly equals the baseline in terms of accuracy and runtime. Import-

tantly, we want to highlight that our method differs in terms of its treatment of the spatiotemporal dynamics. To the best of our knowledge, the baseline method does not propagate uncertainties between time steps. This enables them to work with 50 small kernel matrices, whereas our method works with one sparse block tridiagonal system for the entire spacetime system. Hence, our method offers a more principled treatment of spacetime that propagates uncertainties along time.

6 CONCLUSION

We have presented a framework for efficient probabilistic PDE solvers based on GMRFs. The SPDE approach enables us to construct physics-informed priors that more closely match the dynamics of the true solution, at comparable computation time compared to classically used stationary models. Furthermore, GMRFs map inference to sparse linear algebra, allowing for highly efficient computations which enable a comparison to classic solvers, while adding functionality in the form of principled uncertainty quantification.

This necessarily implies an overhead in runtime, which is however massively reduced in contrast to prior work due to the efficient inference mechanisms enabled by the GMRF framework. Our method matches the high accuracy of classic PDE solvers—which surpasses even state-of-the-art deep learning methods—while providing added functionality in the form of a principled treatment of uncertain data and PDE parameters, all in a machine learning-based method. We hope that our work can contribute to the development of hybrid methods which mix the high inference speeds of deep learning surrogates with the high accuracy and theoretical guarantees of classic PDE solvers.

Acknowledgements

The authors gratefully acknowledge co-funding by the European Union (ERC, ANUBIS, 101123955). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. PH is supported by the DFG through Project HE 7114/6-1 in SPP2298/2. PH is a member of the Machine Learning Cluster of Excellence, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC number 2064/1 – Project number 390727645. The authors also gratefully acknowledge the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (FKZ:01IS18039A); and funds from the Ministry of Science, Research and Arts of the State of Baden-Württemberg. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting TW and MP. The authors are also grateful to Tobias Weber for suggesting improvements to the introduction of this text.

References

- Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, Sep 2015. ISSN 1476-4687. doi: 10.1038/nature14956.
- A. Chandrasekar. *Ocean Models*, page 484–508. Cambridge University Press, 2022.
- J. N. Reddy. *An Introduction to Continuum Mechanics*. Cambridge University Press, 2 edition, 2013.
- P. Šolín. *Equations of Electromagnetics*, chapter 7, pages 269–318. John Wiley & Sons, Ltd, 2005. ISBN 9780471764106. doi: <https://doi.org/10.1002/0471764108.ch7>.
- George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, Jun 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00314-5.
- Nick McGreivy and Ammar Hakim. Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations. *Nature Machine Intelligence*, pages 1–14, September 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00897-5.
- Philipp Hennig, Michael A Osborne, and Hans P Kersting. *Probabilistic Numerics: Computation as Machine Learning*. Cambridge University Press, 2022.
- Marvin Pförtner, Ingo Steinwart, Philipp Hennig, and Jonathan Wenger. Physics-Informed Gaussian Process Regression Generalizes Linear PDE Solvers, 2022. URL <https://arxiv.org/abs/2212.12474>.
- Tim Weiland, Marvin Pförtner, and Philipp Hennig. Scaling up Probabilistic PDE simulators with Structured Volumetric Information, 2024. URL <https://arxiv.org/abs/2406.05020>.
- Junyang Wang, Jon Cockayne, Oksana Chkrebti, T. J. Sullivan, and Chris. J. Oates. Bayesian numerical methods for nonlinear partial differential equations. *Statistics and Computing*, 31(5), July 2021a. ISSN 1573-1375. doi: 10.1007/s11222-021-10030-w.
- Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2011.00777.x.
- Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. Gaussian Measures Conditioned on Nonlinear Observations: Consistency, MAP Estimators, and Simulation. <https://arxiv.org/abs/2405.13149v1>, May 2024.
- Filip Tronarp, Hans Kersting, Simo Särkkä, and Philipp Hennig. Probabilistic solutions to ordinary differential equations as nonlinear Bayesian filtering: A new perspective. *Statistics and Computing*, 29(6):1297–1315, November 2019. ISSN 1573-1375. doi: 10.1007/s11222-019-09900-1.
- P. Whittle. On Stationary Processes in the Plane. *Biometrika*, 41(3/4):434–449, 1954. ISSN 0006-3444. doi: 10.2307/2332724.
- Finn Lindgren, David Bolin, and Håvard Rue. The SPDE approach for Gaussian and non-Gaussian fields: 10 years and still running. *Spatial Statistics*, 50:100599, August 2022. ISSN 2211-6753. doi: 10.1016/j.spasta.2022.100599.
- Geoffrey K. Vallis. *Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation*. Cambridge University Press, Cambridge, 2 edition, 2017. ISBN 978-1-107-06550-5. doi: 10.1017/9781107588417.
- Jean-David Benamou, Brittany D. Froese, and Adam M. Oberman. Two Numerical Methods for the elliptic Monge-Ampère equation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(4): 737–758, July 2010. ISSN 0764-583X, 1290-3841. doi: 10.1051/m2an/2010017.
- Lucia Clarotto, Denis Allard, Thomas Romary, and Nicolas Desassis. The SPDE approach for spatio-

- temporal datasets with advection and diffusion. *Spatial Statistics*, 62:100847, August 2024. ISSN 2211-6753. doi: 10.1016/j.spasta.2024.100847.
- Jon Cockayne, Chris J. Oates, T. J. Sullivan, and Mark Girolami. Bayesian Probabilistic Numerical Methods. *SIAM Review*, 61(4):756–789, January 2019. ISSN 0036-1445. doi: 10.1137/17M1139357.
- Håvard Rue and Leonhard Held. *Gaussian Markov Random Fields: Theory and Applications*. Chapman and Hall/CRC, New York, February 2005. ISBN 978-0-429-20882-9. doi: 10.1201/9780203492024.
- Håvard Rue. Marginal Variances for Gaussian Markov Random Fields, 2005.
- Per Sidén, Finn Lindgren, David Bolin, and Mattias Villani. Efficient Covariance Approximations for Large Sparse Precision Matrices. *Journal of Computational and Graphical Statistics*, 27(4):898–909, October 2018. ISSN 1061-8600. doi: 10.1080/10618600.2018.1473782.
- Thore Graepel. Solving noisy linear operator equations by Gaussian processes: Application to ordinary and partial differential equations. In *Proceedings of the 20th International Conference on Machine Learning*, pages 234–241. AAAI Press, 2003.
- Jon Cockayne, Chris Oates, Tim Sullivan, and Mark Girolami. Probabilistic numerical methods for PDE-constrained Bayesian inverse problems. In Geert Verdooolaege, editor, *Proceedings of the 36th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, volume 1853 of *AIP Conference Proceedings*, pages 060001–1 – 060001–8, 2017. doi: 10.1063/1.4985359.
- Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Machine learning of linear differential equations using Gaussian processes. *Journal of Computational Physics*, 348:683–693, 2017. doi: 10.1016/j.jcp.2017.07.050.
- Houman Owhadi. Bayesian numerical homogenization. *Multiscale Modeling & Simulation*, 13(3):812–828, 2015. doi: 10.1137/140974596.
- Junyang Wang, Jon Cockayne, Oksana Chkrebtii, Tim J. Sullivan, and Chris J. Oates. Bayesian Numerical Methods for Nonlinear Partial Differential Equations. *Statistics and Computing*, 31(55), 2021b. doi: 10.1007/s11222-021-10030-w.
- Nicholas Krämer, Jonathan Schmidt, and Philipp Hennig. Probabilistic Numerical Method of Lines for Time-Dependent Partial Differential Equations. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 151, pages 625–639. PMLR, 2022.
- Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. Solving and learning nonlinear PDEs with Gaussian processes. *Journal of Computational Physics*, 447, 2021. doi: <https://doi.org/10.1016/j.jcp.2021.110668>.
- Ricardo Carrizo Vergara, Denis Allard, and Nicolas Desassis. A general framework for SPDE-based stationary random fields, July 2018.
- Finn Lindgren, Haakon Bakka, David Bolin, Elias Krainski, and Håvard Rue. A diffusion-based spatio-temporal extension of Gaussian Matérn fields, April 2023.
- Mark Girolami, Eky Febrianto, Ge Yin, and Fehmi Cirak. The statistical finite element method (stat-FEM) for coherent synthesis of observation data and model predictions. *Computer Methods in Applied Mechanics and Engineering*, 375, 2021. doi: <https://doi.org/10.1016/j.cma.2020.113533>.
- Anne Poot, Pierre Kerfriden, Iuri Rocha, and Frans van der Meer. A Bayesian Approach to Modeling Finite Element Discretization Error. *Statistics and Computing*, 34(5):167, 2024.
- Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations, May 2021.
- Kristoffer Carlsson, Fredrik Ekre, and Ferrite jl contributors. Ferrite.jl. Zenodo, September 2024.

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] Follows directly from the properties of GMRFs. See e.g. Section 3.2.
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No] Will be made available during the author response period.
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes]
 - (b) Complete proofs of all theoretical results. [Yes]
 - (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [No] We do include detailed explanations of our experiments, particularly in the Appendix. However, we have not yet released the code and will do so during the author response period.
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] See Section C.
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] We report standard deviations across IBVPs where applicable, also in Section C.
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] We report the type of hardware and the cluster used for the experiments.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Yes] We use the datasets of [Li et al. \(2021\)](#) and cite them accordingly in our work.
 - (b) The license information of the assets, if applicable. [Not Applicable]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
 - (d) Information about consent from data providers/curators. [Not Applicable]
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
 - (a) The full text of instructions given to participants and screenshots. [Not Applicable]
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Appendix

A SPATIOTEMPORAL PRIOR

A.1 Construction

In the following, we describe the spatiotemporal prior construction introduced in Section 2.4 in more detail. We follow the construction of [Clarotto et al. \(2024\)](#) for a general spatiotemporal SPDE.

Setup. Assume a spatiotemporal domain $\mathbb{D} = [0, T] \times \mathbb{D}_s$ and a PDE of the form

$$\frac{\partial u}{\partial t}(t, \mathbf{x}) + \mathcal{L}_s[u](t, \mathbf{x}) = \mathcal{W}(t, \mathbf{x}), \quad ((t, \mathbf{x}) \in \mathbb{D}) \quad (25)$$

where $\mathcal{L}_s[u]$ is a differential operator that only acts on the spatial dimensions of u and \mathcal{W} is spatiotemporal noise. More specifically, we assume that the noise has the form

$$\mathcal{W}(t, \mathbf{x}) = \mathcal{W}_t(t) \otimes \mathcal{W}_s(\mathbf{x}), \quad (26)$$

where \mathcal{W}_t is Gaussian white noise and \mathcal{W}_s is Gaussian noise. In practice, to ensure sufficient regularity, we choose \mathcal{W}_s as spatial Matérn noise.

Temporal discretization. Following Rothe's method, we first discretize the temporal dimension through an implicit Euler scheme (which exhibits improved stability properties over explicit Euler schemes). Consider a temporal domain $[0, T]$ and a partition $0 = t_1 < t_2 < \dots < t_{N_t-1} < t_{N_t} = T$ with $N_t \in \mathbb{N}$. Then implicit Euler yields equations of the form

$$u^{(i+1)}(\mathbf{x}) - u^{(i)}(\mathbf{x}) + (t_{i+1} - t_i)\mathcal{L}_s[u^{(i+1)}](\mathbf{x}) = \sqrt{t_{i+1} - t_i}\mathcal{W}_s(\mathbf{x}), \quad (27)$$

for $i \in \{1, \dots, N_t - 1\}$.

Spatial discretization. Next, we assume an appropriate FEM basis function representation for the spatial dimensions. In practice, we use Lagrange polynomials of sufficient order such that forming observations using collocation or FEM for the PDE of interest is possible subsequently. We write $u^{(i)} = \sum_{j=1}^N u_j^{(i)} \phi_j$ and integrate Eq. (27) against test functions, which yields

$$\mathbf{M}\mathbf{u}^{(i+1)} - \mathbf{M}\mathbf{u}^{(i)} + (t_{i+1} - t_i)\mathbf{K}\mathbf{u}^{(i+1)} = \sqrt{t_{i+1} - t_i}\mathbf{M}\boldsymbol{\Sigma}^{1/2}\mathbf{z} \quad (28)$$

$$\iff \underbrace{(\mathbf{M} + (t_{i+1} - t_i)\mathbf{K})}_{=: \mathbf{G}_i} \mathbf{u}^{(i+1)} = \mathbf{M}\mathbf{u}^{(i)} + \underbrace{\sqrt{t_{i+1} - t_i}\mathbf{M}\boldsymbol{\Sigma}^{1/2}\mathbf{z}}_{=: \mathbf{F}_i^{1/2}} \quad (29)$$

$$\iff \mathbf{u}^{(i+1)} = \underbrace{\mathbf{G}_i^{-1}\mathbf{M}}_{=: \mathbf{A}_i} \mathbf{u}^{(i)} + \mathbf{F}_i^{1/2}\mathbf{z} \quad (30)$$

where \mathbf{M} is the mass matrix, \mathbf{K} is the stiffness matrix corresponding to \mathcal{L}_s , $\boldsymbol{\Sigma}^{1/2}$ is a square root of the covariance matrix resulting from a FEM discretization of the spatial Matérn noise process, and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

State-space model. Assume an initial distribution with precision \mathbf{Q}_1 and mean zero. In practice, we use a GMRF approximation to a spatial Matérn process. We obtain the state-space model

$$\mathbf{u}^{(1)} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_1), \quad (31)$$

$$\mathbf{u}^{(i+1)} \mid \mathbf{u}^{(i)} \sim \mathcal{N}(\mathbf{A}_i \mathbf{u}^{(i)}, \mathbf{F}_i). \quad (32)$$

This yields the joint distribution

$$\begin{pmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(N_t)} \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\text{ST}}^{-1}), \quad (33)$$

with block tridiagonal precision matrix

$$\mathbf{Q}_{\text{ST}} := \begin{pmatrix} \mathbf{Q}_1^{-1} + \mathbf{A}_1^T \mathbf{F}_1^{-1} \mathbf{A}_1 & -\mathbf{A}_1^T \mathbf{F}_1^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ -\mathbf{F}_1^{-1} \mathbf{A}_1 & \mathbf{F}_1^{-1} + \mathbf{A}_2^T \mathbf{F}_2^{-1} \mathbf{A}_2 & -\mathbf{A}_2^T \mathbf{F}_2^{-1} & \ddots & \vdots \\ \mathbf{0} & \ddots & \ddots & -\mathbf{A}_{N_t-2}^T \mathbf{F}_{N_t-2}^{-1} & \mathbf{0} \\ \vdots & \ddots & -\mathbf{F}_{N_t-2}^{-1} \mathbf{A}_{N_t-2} & \mathbf{F}_{N_t-2}^{-1} + \mathbf{A}_{N_t-1}^T \mathbf{F}_{N_t-1}^{-1} \mathbf{A}_{N_t-1} & -\mathbf{A}_{N_t-1}^T \mathbf{F}_{N_t-1}^{-1} \\ \mathbf{0} & \dots & \mathbf{0} & -\mathbf{F}_{N_t-1}^{-1} \mathbf{A}_{N_t-1} & \mathbf{F}_{N_t-1}^{-1} \end{pmatrix}. \quad (34)$$

For a temporal discretization with uniform step size, we have $\mathbf{F}_1 = \dots = \mathbf{F}_{N_t-1}$ and $\mathbf{A}_1 = \dots = \mathbf{A}_{N_t-1}$.

A.2 Square root

For efficient sampling when using the conjugate gradient method, a left square root of the precision matrix is helpful. Given square roots of \mathbf{Q}_1 and \mathbf{F}_i^{-1} ($i \in \{1, \dots, N_t - 1\}$) – which are available in closed form by construction – we can construct a block bidiagonal square root of Eq. (34) as

$$\mathbf{Q}_{\text{ST}}^{1/2} := \begin{pmatrix} \mathbf{Q}_1^{-1/2} \mid \mathbf{A}_1^T \mathbf{F}_1^{-1/2} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} \mid -\mathbf{F}_1^{-1/2} & \mathbf{0} \mid \mathbf{A}_2^T \mathbf{F}_2^{-1/2} & \mathbf{0} & \dots & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \mid \mathbf{A}_{N_t-1}^T \mathbf{F}_{N_t-1}^{-1/2} & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \mid -\mathbf{F}_{N_t-1}^{-1/2} & \mathbf{0} \end{pmatrix}, \quad (35)$$

which utilizes wide blocks to treat the sums on the diagonal of Eq. (34).

B BOUNDARY CONDITIONS

Setup. In the following, we describe how to implement GMRFs that fulfill linear constraints of the form

$$u_k = \sum_{i=1}^{N_k} c_i u_{h(i)}, \quad (36)$$

where $k \in \{1, \dots, N\}$ is the index of the degree of freedom that is constrained, $N_k < N$, $\mathbf{c} \in \mathbb{R}^{N_k}$ and $h : \{1, \dots, N_k\} \rightarrow \{1, \dots, N\}$ maps to the indices of the degrees of freedom that u_k depends on. Note that this setup includes

$$\textbf{Homogeneous Dirichlet boundary: } N_k = 0, \quad (37)$$

$$\textbf{Periodic boundary: } N_k = 1, c_1 = 1, h(1) = \textit{mirror}(k), \quad (38)$$

where the function *mirror* maps degrees of freedom to their appropriate counterparts on the boundary of the domain.

Adding constraints to linear systems. Any linear system involving the degrees of freedom needs to be adapted to include the linear constraints. Consider a linear system of the form

$$\mathbf{A}\mathbf{u} = \mathbf{b}. \quad (39)$$

The i -th row of this system is the linear equation

$$\sum A_{ij} u_j = b_i. \quad (40)$$

Inserting Eq. (36) yields

$$\sum_{l=1}^{N_k} A_{ik} c_l u_{h(l)} + \sum_{j \neq k} A_{ij} u_j = b_i \quad (41)$$

$$\iff \sum_{l=1}^{N_k} (A_{ik} c_l + A_{i,h(l)}) u_{h(l)} + \sum_{j \neq \{k\} \cup h(\{1, \dots, N_k\})} A_{ij} u_j = b_i. \quad (42)$$

In other words, the linear constraint may be enforced by adding scaled copies of column k to the columns indexed by $h(\{1, \dots, N_k\})$, and then setting column k to zero.

FEM matrix assembly. The constraints also need to be included in the matrix assembly process. Here, we first follow the process described above for a general linear system, and then we set $A_{kk} = 1$, $A_{kj} = 0$ ($j \neq k$). Recall that the construction of a GMRF stems from the relation

$$\mathbf{K}\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{M}). \quad (43)$$

We apply the constraints to both \mathbf{K} and \mathbf{M} . If we now further set $M_{kk} = \varepsilon_k^2$, then we obtain $u_k \sim \mathcal{N}(0, \varepsilon_k^2)$. ε_k^2 may be used to model a noisy boundary condition: Consider the transformed random variable $\tilde{\mathbf{u}}$ with $\tilde{u}_j = u_j$ ($j \neq k$) and $\tilde{u}_k = u_k + \sum_{i=1}^{N_k} c_i u_{h(i)}$. Then $\tilde{u}_k \mid (\tilde{u}_{h(1)}, \dots, \tilde{u}_{h(N_k)}) \sim \mathcal{N}(\sum_{i=1}^{N_k} c_i \tilde{u}_{h(i)}, \varepsilon_k^2)$. In the limit of $\varepsilon_k^2 \rightarrow 0$ this models a hard, deterministic boundary condition.

	Compute time (sec)	Relative error (%)	Time per sample (sec)	Time for std. dev. (sec)
GMRF-FEM (Normal)	25.71	0.85 ± 0.05	0.51 ± 0.17	33.89 ± 4.08
GMRF-FEM (Inflated)	38.51	0.85 ± 0.05	0.65 ± 0.19	44.73 ± 4.67
FEM	8.00	0.85 ± 0.05	-	-
FNO	-	0.98	-	-

Table 1: Detailed benchmark results for 2D Darcy Flow.

	$N_{xy} = 100$	$N_{xy} = 200$	$N_{xy} = 300$
GMRF-FEM (Normal)	4.79	10.37	24.71
GMRF-FEM (Inflated)	6.56	14.61	35.87

Table 2: Prior construction times for Darcy flow (in seconds).

C EXPERIMENT DETAILS

The Darcy Flow experiment was run on an internal compute cluster on 8 cores of an Intel Xeon Gold 6240 CPU. All other experiments were run on a 2021 M1 Max MacBook Pro.

C.1 Darcy Flow

Dataset. We use the Darcy flow dataset introduced in [Li et al. \(2021\)](#), which consists of 2048 samples in total.

FEM discretization. We use a structured mesh of the unit square consisting of quadratic triangular elements with Lagrange basis functions. We regulate the mesh resolution by specifying the number of elements along each dimension; in practice, we use the same number of elements for both spatial dimensions.

FEM baseline. For the FEM baseline, we ensure a fair comparison by using the same stiffness matrix assembly code as for GMRF-FEM, which is based on Ferrite.jl ([Carlsson et al., 2024](#)). Afterwards, we compute a sparse Cholesky decomposition since the stiffness matrix is symmetric and positive definite for this problem. Finally, we use the Cholesky decomposition to solve for the degrees of freedom.

GMRF-FEM prior. For the prior of GMRF-FEM, we use a GMRF approximation of a two-dimensional Matérn covariance with an effective range of $N_{xy}^{-1/2}$, where N_{xy} is the number of elements per dimension, and with $\nu = 2$. In practice, we find that the accuracy of the solution is not very sensitive to the prior hyperparameters due to the strong weight of the observations in this setup.

Boundary inflation. For the version of GMRF-FEM with an inflated boundary, we inflate the boundary by a width of 0.15 and increase the size of the external elements to up to twice the size of the internal elements.

Compute time. The compute times in Fig. 4 include the time required to assemble the stiffness matrix for both methods, as well as the linear solve for FEM and the time required to condition the prior on the FEM observations for GMRF-FEM. Note that we do not include the time required to form the prior for GMRF-FEM, which we only form once at the start to save resources. Table 1 contains more detailed results, and Table 2 contains the concrete times for the prior construction.

C.2 Burgers’ equation

Dataset. We use the Burgers’ equation dataset introduced in [Li et al. \(2021\)](#) and subsample it down to 30 samples.

FEM discretization. We use a structured mesh of the unit interval consisting of quadratic line elements of equal size with Lagrange basis functions. For the implicit Euler discretization, we use exactly the time points used for evaluation of the solution, which uses a uniformly spaced partition of the temporal domain $[0.0, 1.0]$ with 101 time points. In all cases, our prior is based on a spatial discretization consisting of 750 elements.

Product Matérn prior. For the Product Matérn prior, we use $\nu = 7/2$ for the spatial part and $\nu = 1/2$ for the temporal part. We use a temporal range of 3.0 and a spatial range of 0.02.

Number of collocation points	Advection-Diffusion	Product Matern
0	32.94 ± 10.43	49.25 ± 7.49
5	28.79 ± 8.61	49.25 ± 7.50
10	24.70 ± 7.08	49.24 ± 7.49
25	15.83 ± 5.01	49.23 ± 7.49
100	2.32 ± 2.05	49.06 ± 7.47
250	1.34 ± 1.34	48.16 ± 7.33
500	0.98 ± 1.00	46.95 ± 7.04
1000	0.80 ± 0.54	37.37 ± 4.50

Table 3: Mean and standard deviations of the relative errors of the different methods applied to Burgers’ equation, in %.

Advection-Diffusion prior. For the physics-informed prior, we use the model of Eq. (11), which is a special case of the model described in Clarotto et al. (2024). For the spatial noise we use a Matérn process with $\nu = 3/2$ and a range of 0.02.

Quantitative results. Table 3 shows the concrete values used in Fig. 5.

C.3 Comparison to state-of-the-art probabilistic solver

Nonlinear elliptic PDE. We consider the nonlinear elliptic PDE as presented in Chen et al. (2024):

$$\begin{cases} -\Delta u + u^3 = f & \text{in } \Omega := [0, 1]^2, \\ u = g & \text{on } \partial\Omega, \end{cases} \quad (44)$$

with the manufactured solution

$$u(\mathbf{x}) = \sum_{k=1}^{600} \frac{1}{k^6} \sin(k\pi x_1) \sin(k\pi x_2).$$

For $N_{xy} \in \mathbb{N}$, we employ our method with N_{xy}^2 finite elements. We start with a Matérn prior with effective range 0.1 and $\nu = 2$. Then we condition on the boundary points with a uniform spacing of 0.001 and perform Gauss-Newton optimization on FEM observations of the PDE until either the Newton decrement subceeds $1e-5$ or a maximum number of 10 steps is reached.

1D Burgers’ equation. We consider the nonlinear Burgers’ equation problem presented in Chen et al. (2024):

$$\partial_t u + u \partial_x u - 0.001 \partial_x^2 u = 0, \quad \forall (x, t) \in (-1, 1) \times (0, 1), \quad (45)$$

$$u(x, 0) = -\sin(\pi x), \quad (46)$$

$$u(-1, t) = u(1, t) = 0. \quad (47)$$

The ground-truth solution is computed by a Cole-Hopf transformation via the same code used by Chen et al. (2024).

We run the Gauss-Newton optimization until either the Newton decrement subceeds $1e-5$ or a maximum number of 30 steps is reached.