2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP) | 979-8-3503-7225-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/MLSP58920.2024.10734739

PARALLEL-IN-TIME PROBABILISTIC SOLUTIONS FOR TIME-DEPENDENT NONLINEAR PARTIAL DIFFERENTIAL EQUATIONS

Sahel Iqbal[⋆] *Hany Abdulsamad*[⋆] *Tripp Cator*[†] *Ulisses Braga-Neto*[†] *Simo Särkkä*[⋆]

[⋆] Aalto University, Finland. [†] Texas A&M University, United States of America.

ABSTRACT

We present an efficient probabilistic solver for time-dependent nonlinear partial differential equations. We formulate our method as the maximum a posteriori solver for a constrained risk problem on a reproducing kernel Hilbert space induced by a spatio-temporal Gaussian process prior. We show that for a suitable choice of temporal kernels, the risk objective can be minimized efficiently via a Gauss–Newton algorithm corresponding to an iterated extended Kalman smoother (IEKS). Furthermore, by leveraging a parallel-in-time implementation of IEKS, our algorithm can take advantage of massively parallel graphical processing units to achieve logarithmic instead of linear scaling with time. We validate our method numerically on popular benchmark problems.

Index Terms— partial differential equations, kernel methods, sparse optimization, parallel computation.

1. INTRODUCTION

Partial differential equations (PDEs) are ubiquitous in physics and engineering as a way to model phenomena that correlate over space and time. Apart from a few special cases, PDEs do not admit analytical solutions, and this has led to the development of a vast literature on numerical solutions to PDEs [1]. Recently, a new class of *probabilistic* numerical PDE solvers has garnered attention in the machine learning community [2, 3, 4, 5, 6]. These methods place Gaussian process (GP) priors [7] on the function space of PDE solutions and compute an *a posteriori* solution given a set of collocation points that resemble observed data in Gaussian process regression. In addition to returning numerical solutions of PDEs, such methods confer the unique benefit of providing uncertainty bounds on the solutions, which can often be useful for downstream tasks.

In [5], the authors introduced a method for solving nonlinear PDEs that targets the *maximum a posteriori* (MAP) estimate of a Gaussian process conditioned on a grid of collocation points. The approach is framed as an optimal recovery problem in a reproducing kernel Hilbert space (RKHS) [9] and the resulting solver relies on a *batch* Gauss–Newton algorithm [10, Sec. 10.3]. Their algorithm provides guaranteed convergence for a wide class of PDEs, with the disadvantage of cubic computational complexity in the number of collocation points. An alternative approach introduced in [6], called the *probabilistic numerical method of lines*, first approximates time-dependent PDEs by ordinary differential equations (ODEs). Then, by assuming a Gauss-Markov temporal prior, the approximate ODEs are solved using a Kalman filtering-based ODE solver [11]. The advantage of this method is that it scales linearly with the number of temporal discretization points. However, by using a filtering-based solver, it forgoes the maximum a posteriori solution and well-calibrated posterior uncertainty estimates.

Our contribution in this work is a novel smoothing-based probabilistic numerical solver for time-dependent nonlinear PDEs. It retrieves the maximum a posteriori solution of an approximate ODE formulation while achieving loga-

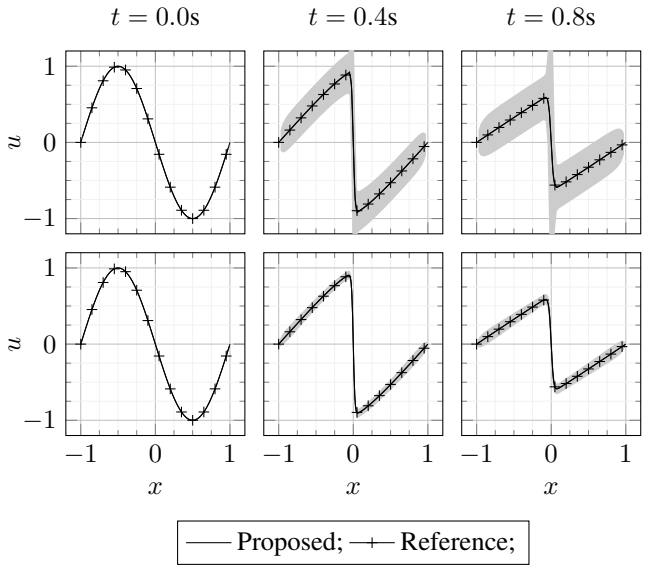

Fig. 1. Probabilistic solutions to the Burgers' equation showing the mean ± twice the standard deviation. The top and bottom rows are for sparse and fine grids respectively. The reference solution is computed according to [8, Sec. 4.4.1].

This work was supported by the National Science Foundation through NSF award CCF-2225507 and by the Research Council of Finland through award 357667.

rithmic complexity in time. Our method differs from [6] in two respects. Firstly, we use the iterated extended Kalman smoother (IEKS) [12] to solve the approximate ODE problem obtained using the method of lines [13]. The IEKS retrieves the MAP estimate and was shown to have favorable theoretical properties for probabilistic ODE solvers in [14]. Secondly, we employ a state-of-the-art, parallel-in-time implementation of the IEKS [15, 16] which can leverage massively parallel graphical processing units (GPUs) to achieve logarithmic span-complexity in the time dimension. The resulting algorithm delivers better-calibrated posterior distributions on PDE solutions while being computationally efficient, both facets which we demonstrate on representative nonlinear PDEs.

2. PROBLEM FORMULATION

Let $t \in [0, T]$ and $x \in [a, b]$ for $T, a, b \in \mathbb{R}$. Our goal is to find a function $u : [0, T] \times [a, b] \to \mathbb{R}^d$, $d \in \mathbb{N}$ that satisfies the following nonlinear PDE

$$\frac{\partial u}{\partial t}(t, x) = f\big(t, x, u(t, x), \mathcal{D}u(t, x)\big), \tag{1}$$

subject to initial and boundary conditions

$$\begin{aligned} u(0, x) &= h(x), \quad \forall x \in [a, b], \\ u(t, x) &= g(x), \quad \forall (t, x) \in [0, T] \times \{a, b\}. \end{aligned}$$

f, g and h are known nonlinear functions and $\mathcal{D}$ is a differential operator. We assume that the initial and boundary conditions are chosen appropriately such that the PDE has a unique solution in the domain.

In addition to finding a solution u, we wish to quantify the uncertainty in the solution arising from the discretization in the numerical solver. For that purpose, we define a Gaussian process prior over PDE solutions, $u \sim \mathcal{GP}(0, k)$, where $k : \Omega \times \Omega \to \mathbb{R}$ is a symmetric, positive-definite *kernel function* with $\Omega := [0, T] \times [a, b]$. Associated with the kernel k is a unique reproducing kernel Hilbert space $\mathcal{H}$, equipped with the norm $\|\cdot\|_{\mathcal{H}}$ [9]. Furthermore, for a finite-dimensional representation of the PDE solution, we assume a symmetric set of collocation points on which the solution is enforced and define the time and space grids $\mathbb{T} := \{t_0, t_1, \ldots, t_N\}$ and $\mathbb{X} := \{x_0, x_1, \ldots, x_M\}$ with $0 = t_0 < t_1 < \cdots < t_N = T$ and $a = x_0 < x_1 < \cdots < x_M = b$, for some $N, M \in \mathbb{N}$.

We follow the setting proposed in [5] and formulate the MAP estimate of the GP constrained by a PDE and its initial and boundary conditions in the discretized domain $\mathbb{T} \times \mathbb{X}$ as a constrained optimization problem in the Hilbert space

$$\begin{aligned} \min_{u \in \mathcal{H}} \quad & \|u\|_{\mathcal{H}} \qquad (2) \\ \text{subject to} \quad & \frac{\partial u}{\partial t}(t, x) = f\big(t, x, u, \mathcal{D}u\big), \ \forall (t, x) \in \mathbb{T} \times \mathbb{X}, \\ & u(t, x) = g(x), \ \forall (t, x) \in \mathbb{T} \times \{a, b\}, \\ & u(0, x) = h(x), \ \forall x \in \mathbb{X}. \end{aligned}$$

In [5], the authors used a batch Gauss–Newton algorithm to solve (2), which involves inverting an $(M \times N)$-dimensional Gram matrix constructed by evaluating the kernel function at the grid points. Exact computation of the inverse of dense matrices has cubic cost in the dimension, resulting in a computational cost of $\mathcal{O}\big(M^3 N^3\big)$ for their algorithm [5, Section 3.4.3]. This approach was extended in [17], bringing the computational complexity to near-linear using approximations of the Cholesky decomposition of the Gram matrix inverse.

While such an approach is generally applicable for both time-dependent and time-independent PDEs, it needlessly enforces a dense structure in both the time and space dimensions for time-dependent PDEs. In the following sections, we show that by relying on a method-of-lines approximation ansatz, and by leveraging a class of Gauss–Markov temporal priors that promote a sparse optimization structure, we can form an approximate optimal recovery problem to (2) whose MAP estimate can be recovered at a linear computational cost in the time dimension. The MAP estimate is computed using an efficient Gauss–Newton algorithm in the form of an iterated extended Kalman smoother (IEKS), with overall complexity of $\mathcal{O}(M^3 N)$. Moreover, by using a parallel implementation of the IEKS for GPUs [15, 16], we reduce complexity further down to $\mathcal{O}(M^3 \log N)$. While similar spatio-temporal GP priors are used in [6] for solving PDEs, the proposed solver does not target a well-defined optimization objective.

3. BACKGROUND

In this section, we describe how to convert a PDE to a system of ODEs using the method of lines, followed by a description of the equivalence between certain Gaussian process priors and stochastic differential equations (SDEs). We then elaborate on our choice of the spatio-temporal prior and the data model, which are similar to [6].

3.1. The Method of Lines

The method of lines (MOL) [13] is a technique to solve PDEs by converting them to a system of ODEs. This is achieved by discretizing all but one dimension of the PDE, typically the time dimension. In our case, we use a finite difference scheme to approximate the differential operator $\mathcal{D}$ on the spatial grid $\mathbb{X}$ as a matrix-vector product

$$(\mathcal{D}u)(t, \mathbb{X}) \approx Du(t, \mathbb{X}), \quad D \in \mathbb{R}^{d(M+1) \times d(M+1)},$$

where we have defined

$$u^\top(t, \mathbb{X}) := [u^\top(t, x_0), u^\top(t, x_1), \ldots, u^\top(t, x_M)].$$

After removing the differential operator, we approximate the function f from (1) by

$$f(t, x, u, \mathcal{D}u) \approx \hat{f}(t, x, u).$$

Since the grid $\mathbb{X}$ is fixed, let us define $\mathbf{u}(t) \coloneqq u(t, \mathbb{X})$ and $\hat{f}(t, \mathbf{u}) \coloneqq \hat{f}(t, \mathbb{X}, \mathbf{u})$, which transforms (1) to an ODE

$$\frac{\partial \mathbf{u}}{\partial t}(t) \approx \hat{f}(t, \mathbf{u}), \tag{3}$$

with initial condition $\mathbf{u}(t_0) = \left[h^\top(x_0), \dots, h^\top(x_M)\right]^\top$. We can now use probabilistic ODE solvers to solve (3), which will yield approximate solutions to the PDE from (1).

3.2. Gaussian Process Priors as Stochastic Differential Equations

A kernel function k is said to be *stationary* if $k(t, t') = k(t - t')$. Certain classes of stationary kernel functions admit equivalent representations as linear time-invariant stochastic differential equations [18]. Let $f : \mathbb{R} \to \mathbb{R}^d$ and

$$\psi^\top(t) = \left[(f(t))^\top, (f^{(1)}(t))^\top, \cdots, (f^{(\nu)}(t))^\top\right]$$

be a state vector consisting of $f(t)$ and its derivatives up to order $\nu \in \mathbb{N}$. Let us also define matrices $E_m \coloneqq e_m \otimes \mathbb{I}_{d(M+1)}$, where $\{e_m\}_{m=0}^{\nu}$ is the standard basis in $\mathbb{R}^{\nu+1}$, $\mathbb{I}_{d(M+1)}$ is the identity matrix in $\mathbb{R}^{d(M+1)}$ and $\otimes$ denotes the Kronecker product. The matrix E_m picks out the m-th order derivative of f from the state vector ψ,

$$f^{(m)}(t) = E_m^\top \psi(t), \quad \forall m \in \{0, 1, \dots, \nu\},$$

with $f^{(0)} \coloneqq f$. Then there exists kernel functions k such that $f \sim \mathcal{GP}(0, k)$ implies the state $\psi(t)$ solves an SDE

$$\mathrm{d}\psi(t) = F\,\psi(t)\,\mathrm{d}t + E_\nu\,\mathrm{d}\beta(t), \quad \psi(t_0) \sim \mathcal{N}(0, \Sigma_0), \tag{4}$$

where $\beta(t)$ is a vector of Wiener processes with diffusion matrix Γ and $\mathcal{N}$ denotes the Gaussian distribution. The ν-times integrated Wiener process and the $(\nu + 1/2)$-order Matérn functions are examples of kernels that admit this form [19, Sec. 12.3]. F, Γ, and Σ_0 are all specified by the choice of the kernel k and are thus dependent on these hyperparameters of the kernel.

Time Discretization. On the time grid $\mathbb{T}$, we can discretize the SDE from (4) to obtain linear-Gaussian transitions [19, Section 6.2]

$$\psi(t_{k+1}) \mid \psi(t_k) \sim \mathcal{N}(A(h_k)\psi(t_k), Q(h_k)),$$

where $h_k = t_{k+1} - t_k$ and

$$A(h_k) = \exp(Fh_k),$$
$$Q(h_k) = \int_0^{h_k} A(h_k - \tau) E_\nu\, \Gamma E_\nu^\top A^\top(h_k - \tau)\,\mathrm{d}\tau.$$

Coupled with a suitable observation model, this forms a state-space model for which inference can be done at $\mathcal{O}(N)$ computational complexity, which is favorable compared to the $\mathcal{O}(N^3)$ required for traditional GP regression [7, Ch. 2].

3.3. Spatio-Temporal Priors on the Solution

We place a separable spatio-temporal GP prior on the solution of the PDE [20],

$$u(t, x) \sim \mathcal{GP}(0, k_t \otimes k_x),$$

where $k_t \otimes k_x$ is the product kernel $(k_t \otimes k_x)(t, t', x, x') = k_t(t, t')k_x(x, x')$. This induces the following prior on the solution on the grid $\mathbf{u}(t)$,

$$\mathbf{u}(t) \sim \mathcal{GP}(0, k_t \otimes K), \quad K_{ij} = k_x(x_i, x_j).$$

If we choose a kernel k_t that admits an equivalent SDE representation as given in (4), we obtain Gaussian transitions on the time grid

$$\begin{aligned}\psi(t_{k+1}) \mid \psi(t_k) &\sim \mathcal{N}(\tilde{A}(h_k)\psi(t_k), \tilde{Q}(h_k)),\\ \tilde{A}(h_k) &= A(h_k) \otimes \mathbb{I}_{d(M+1)},\\ \tilde{Q}(h_k) &= Q(h_k) \otimes K,\end{aligned}$$

where $\psi^\top(t) \coloneqq [(\mathbf{u}(t))^\top, (\mathbf{u}^{(1)}(t))^\top, \dots, (\mathbf{u}^{(\nu)}(t))^\top]$ and $\mathbf{u}^{(\nu)} = \partial^\nu \mathbf{u} / \partial t^\nu$.

3.4. Data Model

Similar to probabilistic ODE solvers [21, 11], our data model enforces the PDE at the time grid points. We construct the observation function

$$\begin{aligned}z(t) &\coloneqq E_1^\top \psi(t) - \hat{f}(t, E_0^\top \psi(t))\\ &= \mathbf{u}^{(1)}(t) - \hat{f}(t, \mathbf{u}(t)),\end{aligned} \tag{5}$$

and *approximately* enforce the PDE by conditioning on fixed observations $z(t_n) = 0$ for all $n = 0, \dots, N$, which corresponds to enforcing the ODE from (3).

4. MAXIMUM A POSTERIORI ESTIMATION

In this section, we detail our algorithm to compute the MAP estimate of the PDE solution under the method-of-lines approximation formulated in Sec. 3.1.

The Markovian prior process and the data model together yield a state-space model (SSM) formulation with linear-Gaussian transition dynamics and a Dirac delta observation model centered at the collocation points

$$\begin{aligned}\psi(t_k) \mid \psi(t_{k-1}) &\sim \mathcal{N}(\tilde{A}(h_{k-1})\psi(t_{k-1}), \tilde{Q}(h_{k-1})),\\ z(t_k) = 0 \mid \psi(t_k) &\sim \delta\big(E_1^\top \psi(t_k) - \hat{f}(t, E_0^\top \psi(t_k))\big).\end{aligned}$$

The MAP estimate for the full state trajectory of the SSM is

defined by the following optimization problem [14]

$$\begin{aligned} \min_{\psi(t_{0:N})} \quad & \mathcal{J}(\psi(t_{0:N})) \\ \text{subject to} \quad & E_0^\top \psi(t_0) = h(\mathbb{X}), \\ & E_1^\top \psi(t_0) = \hat{f}(t_0, h(\mathbb{X})), \\ & u(t_n, x) = g(x) \qquad \text{for } x \in \{a, b\}, \\ & z(t_n, \psi(t_n)) = 0, \quad n = 1, \ldots, N, \end{aligned}$$

where the objective $\mathcal{J}$ is the negative prior log density

$$\mathcal{J}(\cdot) = \sum_{n=0}^{N-1} \|\psi(t_{n+1}) - A(h_n)\psi(t_n)\|^2_{Q(h_n)} + \|\psi(t_0)\|^2_{\Sigma_0}.$$

Here $\|\cdot\|^2_\Sigma$ denotes the squared Mahalanobis distance with respect to a covariance matrix Σ. This optimal recovery problem approximates (2), with the difference being that here we enforce the approximate ODE rather than the true PDE.

When the function $\hat{f}$ in (3) is affine, the observation model (5) is affine and the full posterior $p\big(\psi(t_{0:N}) \mid z(t_{0:N})\big)$, centered at the true MAP, can be obtained through Gaussian smoothing [22]. However, for general nonlinear functions $\hat{f}$, the posterior cannot be computed exactly and we have to resort to approximations. In such cases, iterative algorithms can be used to target the MAP estimate. A prominent example is the iterated extended Kalman smoother (IEKS) [12], which is a Gauss–Newton algorithm with linear complexity in time. Despite primarily targeting the MAP, the IEKS also delivers an approximation of the posterior uncertainty in the neighborhood of the posterior mode.

4.1. Exact Inference for Affine Vector Fields

For affine vector fields $\hat{f}\big(t, \mathbf{u}(t)\big) = \Lambda(t)\,\mathbf{u}(t) + \zeta(t)$, the MAP estimate and the full posterior can be retrieved using the Rauch-Tung-Striebel (RTS) smoother [23]. In the following, the mean and covariance of the state $\psi(t)$ conditioned on the observations until time t are denoted by $\mu_F(t)$ and $\Sigma_F(t)$ respectively, while those conditioned on the observations until time T are denoted by $\mu_S(t)$ and $\Sigma_S(t)$. The smoother consists of a forward and a backward pass. The forward pass computes the filtered marginal distributions by alternating between a *prediction step*,

$$\begin{aligned} \mu_F^-(t_n) &= \tilde{A}(h_{n-1})\mu_F(t_{n-1}), \\ \Sigma_F^-(t_n) &= \tilde{A}(h_{n-1})\Sigma_F(t_{n-1})\tilde{A}^\top(h_{n-1}) + \tilde{Q}(t_{n-1}), \end{aligned}$$

and an *update step*

$$\begin{aligned} H(t_n) &= E_1^\top - \Lambda(t_n)E_0^\top, \\ S(t_n) &= H(t_n)\Sigma_F^-(t_n)H^\top(t_n), \\ K(t_n) &= \Sigma_F^-(t_n)H^\top(t_n)S^{-1}(t_n), \\ \mu_F(t_n) &= \mu_F^-(t_n) + K(t_n)\big(\zeta(t_n) - H(t_n)\mu_F^-(t_n)\big), \\ \Sigma_F(t_n) &= \Sigma_F^-(t_n) - K(t_n)S(t_n)K^\top(t_n). \end{aligned}$$

The initial conditions are enforced by an update step at t_0. The backward pass computes the smoothed marginals via

$$\begin{aligned} G(t_n) &= \Sigma_F(t_n)\tilde{A}^\top(h_n)[\Sigma_F^-(t_{n+1})]^{-1}, \\ \mu_S(t_n) &= \mu_F(t_n) + G(t_n)\big(\mu_S(t_{n+1}) - \mu_F^-(t_{n+1})\big), \\ \Sigma_S(t_n) &= \Sigma_F(t_n) + G(t_n)\big(\Sigma_S(t_{n+1}) - \Sigma_F^-(t_{n+1})\big)G^\top(t_n), \end{aligned}$$

starting with $\mu_S(t_N) = \mu_F(t_N)$ and $\Sigma_S(t_N) = \Sigma_F(t_N)$.

4.2. The Iterated Extended Kalman Smoother

The smoothing algorithm presented in Sec. 4.1 applies only to affine vector fields. To tackle nonlinear PDEs, we use the iterated extended Kalman smoother. The IEKS iteratively linearizes the measurement model around the current smoother estimate, then applies the Gaussian smoothing equations described in the previous section. This process is repeated until convergence [22, Sec. 13.3].

The function $\hat{f}$ is approximated with an affine function $\hat{f}\big(t, \mathbf{u}(t)\big) \approx \Lambda(t)\,\mathbf{u}(t) + \zeta(t)$ and this approximation is used in the update step of the forward pass. At iteration $(l+1)$ of the IEKS, Λ and ζ are obtained using a first-order Taylor series expansion around the smoothing mean from the previous iteration, $\mu_S^l(t_n)$, as follows

$$\begin{aligned} \Lambda^l(t_n) &= J_f(t_n, E_0^\top \mu_S^l(t_n)), \\ \zeta^l(t_n) &= \hat{f}(t_n, E_1^\top \mu_S^l(t_n)) - J_f(t_n, E_0^\top \mu_S^l(t_n))\, E_0^\top \mu_S^l(t_n). \end{aligned}$$

Here, J_f is the Jacobian of $\hat{f}$ with respect to $\mathbf{u}$. The mean and covariance at iteration $(l+1)$ are then computed using the above linearization. We initialize the algorithm with an extended RTS smoother that uses the same first-order Taylor linearization for the measurement model [11].

4.3. Temporal Parallelization of Bayesian Smoothers

The Gaussian filter and smoother presented in Section 4.1 have $\mathcal{O}(N)$ computational complexity in the number of time grid points N. This linear complexity can be a bottleneck for stiff PDEs, for which dense grids are often necessary to achieve satisfactory levels of accuracy. Fortunately, the span-complexity of the algorithm, which is the number of computational steps as measured by a wall clock, can be reduced to $\mathcal{O}(\log N)$ using the parallel-in-time Bayesian smoothers introduced in [15, 16].

The parallel implementation of [15] makes use of the *parallel scan* algorithm [24]. Given a set of elements $\{a_k\}_{k=0}^N$ and a binary associative operator $\otimes$, the parallel scan algorithm computes the prefix sums $\bigotimes_{i=0}^k a_i \coloneqq a_0 \otimes a_1 \otimes \cdots \otimes a_k$ for all $k = 1, \ldots, N$ at $\mathcal{O}(\log N)$ span-complexity. This is conditional on having at least $(N+1)$ computational threads for parallel operations. In [15], the authors identified the elements a_k and operators $\otimes$ such that the prefix sums would correspond to the filtering and smoothing distributions, and

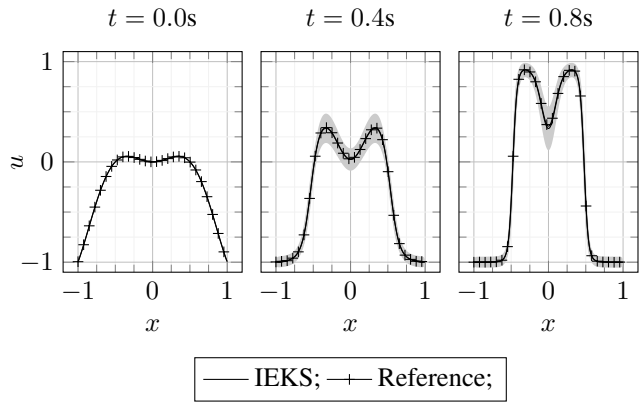

Fig. 2. Solution to the Allen-Cahn equation. We plot the mean $\pm$ twice the standard deviation.

we direct the reader to [15, Sec. 4] for details. The parallel-in-time IEKS algorithm was recently utilized for probabilistic solutions to ODEs [25].

5. NUMERICAL EVALUATION

For the empirical evaluation of our method, we first consider the *viscous Burger's equation*, given by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \eta\frac{\partial^2 u}{\partial x^2} = 0, \quad (t, x) \in [0, 1.5] \times [-1, 1],$$

with $\eta = 1 \times 10^{-2}$, initial condition $u(0, x) = -\sin(\pi x)$ and boundary condition $u(t, -1) = u(t, 1) = 0$. In addition, we consider the *reaction-diffusion Allen-Cahn equation*,

$$\frac{\partial u}{\partial t} - \gamma\frac{\partial^2 u}{\partial x^2} + 5u^3 - 5u = 0, \quad (t, x) \in [0, 1] \times [-1, 1],$$

with $\gamma = 5 \times 10^{-3}$, initial condition $u(0, x) = x^2\cos(\pi x)$ and boundary condition $u(t, -1) = u(t, 1) = -1.0$.

In Fig. 1 and Fig. 2, we visualize the MAP estimate and the approximate posterior distribution of the solution computed by the IEKS for both equations and compare to numerical solutions obtained using finite-difference schemes. Fig. 1 plots the solution for two different discretizations. For the finer discretization, the MAP estimate is closer to the reference solution and the approximate posterior has lower variance, as is expected of a well-calibrated uncertainty estimate.

Next, we demonstrate the computational efficiency of our solver in Fig. 3. Depicted is the runtime in seconds as a function of the number of time discretization points N of the sequential and parallel versions of the IEKS, along with that of a batch solution to the smoothing problem. The batch solver has $\mathcal{O}(N^3)$ computational complexity, similar to the method of [5]. As evident from the figure, at fine discretizations, and consequently higher number of grid points, the parallel IEKS implementation holds a clear advantage over alternatives. [1]

[1] Our code: `https://github.com/hanyas/parallel-pde`.

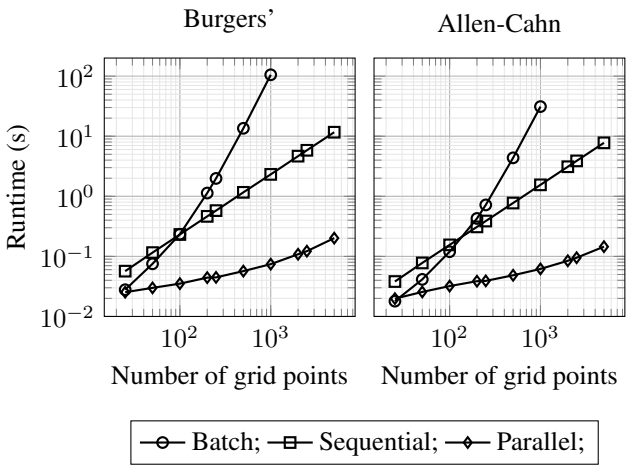

Fig. 3. GPU runtime of the sequential and parallel implementations of the IEKS algorithm, along with that of the batch solution, for the Burgers' and Allen-Cahn equations.

Table 1. Log likelihoods of the reference trajectories under the posteriors and average L_2 norms of the posterior means for the EKF and the IEKS.

	Burgers'		Allen-Cahn	
Algorithm	LLH	L_2	LLH	L_2
EKF	96743	0.038	-137642	0.077
IEKS	113366	0.038	-130540	0.081

Finally, Tab. 1 reports two metrics that compare the solution quality of the IEKS and the extended Kalman filter (EKF) approaches. The first is the log likelihood of the reference solution under the posteriors returned by the algorithms. The IEKS solutions achieve higher log likelihoods for both PDEs, indicating better-calibrated posterior uncertainty. The second metric, at which both the EKF and IEKS demonstrate comparable results, is the average L_2 norm of the posterior mean with respect to the reference solution.

6. CONCLUSION

We have introduced a novel probabilistic numerical solver for time-dependent, nonlinear PDEs that scales logarithmically in time. We use the method of lines to approximate nonlinear PDEs by coupled ODEs and compute the MAP estimate of the solution using a massively parallelizable realization of the iterated extended Kalman smoothing algorithm. Numerical experiments demonstrate the accuracy and computational efficiency of our method.

Our method is not without limitations. Firstly, unlike the method of [5], ours does not enforce the true PDE at the collocation points, but rather the approximate ODE from (3). Secondly, our method does not account for the error induced by discretizing the spatial coordinate. One way to achieve this

would be to model the error similar to [6], which uses a *probabilistic discretization* to track the error from using a spatial grid. Finally, while logarithmic in time, our algorithm has cubic cost in space, which could prove prohibitive for large domains or fine discretizations. Addressing these limitations is identified as future work.

7. REFERENCES

[1] Randall J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*, Society for Industrial and Applied Mathematics, 2007.

[2] Houman Owhadi, "Bayesian numerical homogenization," *Multiscale Modeling & Simulation*, vol. 13, no. 3, pp. 812–828, 2015.

[3] Jon Cockayne, Chris Oates, Tim Sullivan, and Mark Girolami, "Probabilistic numerical methods for PDE-constrained Bayesian inverse problems," *AIP Conference Proceedings*, vol. 1853, no. 1, pp. 060001, 2017.

[4] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis, "Numerical Gaussian processes for time-dependent and nonlinear partial differential equations," *SIAM Journal on Scientific Computing*, vol. 40, no. 1, pp. A172–A198, 2018.

[5] Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart, "Solving and learning nonlinear PDEs with Gaussian processes," *Journal of Computational Physics*, vol. 447, pp. 110668, 2021.

[6] Nicholas Krämer, Jonathan Schmidt, and Philipp Hennig, "Probabilistic numerical method of lines for time-dependent partial differential equations," in *AISTATS*, 2022.

[7] Carl Edward Rasmussen and Christopher K. I. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, 2006.

[8] Lawrence C. Evans, *Partial differential equations*, American Mathematical Society, 2010.

[9] Nachman Aronszajn, "Theory of reproducing kernels," *Transactions of the American mathematical society*, vol. 68, no. 3, pp. 337–404, 1950.

[10] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, Springer New York, 2006.

[11] 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*, vol. 29, no. 6, pp. 1297–1315, 2019.

[12] Bradley M Bell, "The iterated Kalman smoother as a Gauss–Newton method," *SIAM Journal on Optimization*, vol. 4, no. 3, pp. 626–636, 1994.

[13] William E Schiesser, *The numerical method of lines: integration of partial differential equations*, Elsevier, 2012.

[14] Filip Tronarp, Simo Särkkä, and Philipp Hennig, "Bayesian ODE solvers: the maximum a posteriori estimate," *Statistics and Computing*, vol. 31, no. 23, 2021.

[15] Simo Särkkä and Ángel F. García-Fernández, "Temporal parallelization of Bayesian smoothers," *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 299–306, 2021.

[16] Fatemeh Yaghoobi, Adrien Corenflos, Sakira Hassan, and Simo Särkkä, "Parallel iterated extended and sigma-point Kalman smoothers," in *ICASSP*, 2021.

[17] Yifan Chen, Houman Owhadi, and Florian Schäfer, "Sparse Cholesky factorization for solving nonlinear PDEs via Gaussian processes," *arXiv preprint arXiv:2304.01294*, 2024.

[18] Jouni Hartikainen and Simo Särkkä, "Kalman filtering and smoothing solutions to temporal Gaussian process regression models," in *MLSP*, 2010.

[19] Simo Särkkä and Arno Solin, *Applied Stochastic Differential Equations*, Cambridge University Press, 2019.

[20] Arno Solin, Pasi Jylänki, Jaakko Kauramäki, Tom Heskes, Marcel AJ van Gerven, and Simo Särkkä, "Regularizing solutions to the MEG inverse problem using space-time separable covariance functions," *arXiv preprint arXiv:1604.04931*, 2016.

[21] Jon Cockayne, Chris J. Oates, T. J. Sullivan, and Mark Girolami, "Bayesian probabilistic numerical methods," *SIAM Review*, vol. 61, no. 4, pp. 756–789, 2019.

[22] Simo Särkkä and Lennart Svensson, *Bayesian filtering and Smoothing*, Cambridge University Press, 2nd edition, 2023.

[23] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[24] Guy E Blelloch, "Scans as primitive parallel operations," *IEEE Transactions on Computers*, vol. 38, no. 11, pp. 1526–1538, 1989.

[25] Nathanael Bosch, Adrien Corenflos, Fatemeh Yaghoobi, Filip Tronarp, Philipp Hennig, and Simo Särkkä, "Parallel-in-time probabilistic numerical ODE solvers," *arXiv preprint arXiv:2310.01145*, 2023.