Goal-Oriented Real-Time Bayesian Inference for Linear Autonomous Dynamical Systems With Application to Digital Twins for Tsunami Early Warning

Stefan Henneking*, Sreeram Venkat, Omar Ghattas
Oden Institute, The University of Texas at Austin

January 23, 2025

Abstract

We present a goal-oriented framework for creating digital twins with the following properties: (1) they employ discretizations of high-fidelity partial differential equation (PDE) models governed by autonomous dynamical systems, leading to large-scale forward problems; (2) they solve a linear inverse problem to assimilate observational data to infer uncertain model components followed by a forward prediction of the evolving dynamics; and (3) the entire end-to-end, data-to-inference-to-prediction computation is carried out in real time through a Bayesian framework that rigorously accounts for uncertainties. Realizations of such a framework are faced with several challenges that stem from the extreme scale of forward models and, in some cases, slow eigenvalue decay of the parameter-to-observable map. In this paper, we introduce a methodology to overcome these challenges by exploiting the autonomous structure of the forward model. As a result, we can move the PDE solutions, which dominate the cost for solving the Bayesian inverse problem, to an offline computation and leverage the high-performance dense linear algebra capabilities of GPUs to accelerate the online prediction of quantities of interest. We seek to apply this framework to create digital twins for the Cascadia subduction zone as a means of providing early warning for tsunamis generated by subduction zone megathrust earthquakes. To that end, we demonstrate how our methodology can be used to employ seafloor pressure observations, along with the coupled acoustic-gravity wave equations, to infer the earthquake-induced spatiotemporal seafloor motion (discretized with $\mathcal{O}(10^9)$ parameters) and forward predict the tsunami propagation toward populated regions along coastlines. We present results of an end-to-end inference, prediction, and uncertainty quantification for a representative test problem for which this goal-oriented Bayesian inference, based on the discretized full-order model, is accomplished without loss of accuracy in real time, that is, in a matter of seconds.

Keywords: Large-scale inverse problems, Goal-oriented Bayesian inference, Data assimilation, Uncertainty quantification, Digital twins

1 Introduction

Motivation. This paper presents a novel framework for solving massive scale linear inverse problems governed by autonomous dynamical systems using high-fidelity partial differential equation (PDE) models, with rigorously quantified uncertainties, in real time. Our target application is the inference of megathrust earthquake-induced seafloor motion from seafloor measurements in subduction zones, and subsequent forecasting of tsunamis in vulnerable coastal regions, with particular

[*] Corresponding author: stefan@oden.utexas.edu

focus on the Cascadia subduction zone (CSZ) in the Pacific Northwest. Tsunamis resulting from megathrust earthquakes are capable of massive destruction and loss of human life. Two recent examples are the tsunamis unleashed by the 2011 Tōhoku earthquake, resulting in 22,000 deaths and 410 B in damages, and the 2004 Sumatra–Andaman earthquake, resulting in 230,000–280,000 total deaths. Strain has been accumulating on the CSZ since its last major rupture in 1700. Statistical recurrence intervals suggest a significant probability of a major rupture (up to magnitude 9) on this fault in the next 50 years [27], which would have an enormous impact on millions of people living in the Pacific Northwest [12, 36].

With the increasingly dense instrumentation of subduction zones, including Cascadia [37, 7], and the advance of earthquake-tsunami simulations enabled by high-performance computing [41, 26, 1], the availability of real-time observational data promises to enhance tsunami early warning systems. Our methodology aims to employ pressure observations obtained from seafloor sensors, along with the coupled acoustic-gravity wave equations that describe the generation of tsunamis [47], to infer the earthquake-induced spatiotemporal seafloor motion. Employing a model that accounts for acoustic wave propagation is critical for extracting the maximum amount of information from the acoustic pressure sensors, particularly in the near field and during the transients that follow a rupture on the fault [28, 29]. The solution of this inverse problem then provides the boundary forcing to forward propagate the tsunamis toward populated and vulnerable regions along coastlines. Other tsunami early warning methods that quantify uncertainties in tsunamigenic earthquake sources (e.g. [17] and references therein) typically rely on statistical tools (rather than full-order PDE solutions) to enable rapid forecasting of the propagating tsunami under uncertainty. To provide sufficient early warning to guide evacuation of lower lying regions and inform early responders, the entire inverse problem and subsequent forecasting of coastal inundation must be carried out in real time, i.e. in a few seconds. This is because tsunamis generated on the CSZ may reach coastal communities in as little as 20 minutes [28, 33]. Such early warning systems would have saved innumerable lives in the Tōhoku and Sumatra–Andaman earthquakes. However, solution of this inverse problem satisfying the discretized high-fidelity PDE model exactly in real time represents a grand challenge.

Large-scale inverse problems. A reasonable discretization of the acoustic-gravity PDE forward model for the 1000 km long CSZ gives rise to a system with $\mathcal{O}(10^9)$ parameters that describe the dynamics of the earthquake-induced spatiotemporal seafloor motion. Scalable algorithms for large-scale inverse problems stemming from discretizations of PDEs have been developed and successfully employed for a number of challenging application problems. These include ice sheet dynamics [23], mantle convection [46], poroelasticity [22], atmospheric transport [15], ocean dynamics [24], tumor growth modeling [40], and others. Typically, these algorithms rely on exploiting some low-dimensional structure of the map from parameters (inputs) to observations (outputs) in order to construct a cheaper-to-evaluate surrogate approximation of the high-fidelity model. Numerous surrogate modeling techniques exist, including Gaussian processes [25], radial basis functions [44], projection-based model reduction [8], as well as machine learning techniques such as Fourier neural operators [30], derivative-informed neural operators [35], and many others. Generally speaking, these techniques work well when the dynamics governed by the full-order PDE model can be well-approximated in a lower-dimensional solution manifold, which is indeed a feature inherent to many problems in science and engineering [16].

For the tsunami inverse problem, in addition to the need for real-time inference, we are faced with the major difficulty that developing accurate and predictive surrogates over a 10^9-dimensional parameter space for tsunami dynamics described by hyperbolic PDEs is seemingly intractable; such systems do not readily admit low-dimensional subspace representations [19]. Moreover, wave propagation inverse problems are ill-posed in the high wavenumber components of the parameter field [10, 11, 9] due to the band-limited observational data; additional "noise" stemming from the surrogate (due to its typical inability to capture high frequencies) will be amplified in the inverse solution and may lead to instability, even if instrument noise is low. The high-rank structure of the input–output map and the corresponding inverse operator—i.e. the Hessian matrix of the negative log posterior—renders existing techniques largely inefficient and therefore inapplicable to real-time inference of high-dimensional parameter fields for these types of inverse problems. The need for forward predicting specific quantities of interest (QoIs) and quantifying their uncertainties within a goal-oriented Bayesian framework further compounds the difficulty of the problem. In recent years, some methods that address high-rank Hessians have been developed, including those that exploit the pseudo-differential [34, 14], augmented Lagrangian [4], product-convolution [3], H-matrix [5], and point spread function [2] structure of the Hessians of particular classes of inverse problems.

However, we can entirely circumvent the need for surrogate approximations and solve the inverse problem exactly (up to discretization error) if the inverse problem is governed by autonomous dynamical systems. The evolution of such systems with respect to any given input may depend on the system's current state but does not explicitly depend on the independent variable. The coupled acoustic-gravity PDE model that governs the tsunami dynamics is a time-invariant dynamical system, which is a subclass of autonomous systems where the independent variable is time. For such problems, we can by exploit the intrinsic structure of the problem and move most of the computationally expensive parts of the Bayesian inverse problem offline. As a key component of this framework, we employ a multi-GPU-accelerated fast Fourier transform (FFT)-based algorithm for computing Hessian matvecs that provides a $\mathcal{O}(10^4)\times$ speedup over conventional methods (after a one-time setup cost) [42].

Contributions. Solution of large-scale inverse problems from observations and high-fidelity PDE models, together with the prediction of QoIs under uncertainty, belongs to the general class of digital twins. This paper introduces an efficient and scalable digital twin framework to perform goal-oriented real-time Bayesian inference for linear autonomous dynamical systems. The framework is applicable to inverse problems with high-rank Hessians for which existing surrogate techniques are largely inefficient. Moreover, contrary to methods based on surrogate modeling, our approach solves the inverse problem exactly, i.e. satisfying the (discretized) full-order PDE model. The framework can be used to infer infinite-dimensional parameter fields (high-dimensional when discretized) but assumes that the observational data are relatively sparse. In particular, the number of spatial observers (e.g. sensor locations) needs to be small enough so that the discretized PDE model can be solved offline in a precomputation step once per sensor. The Bayesian inverse problem is then posed in a goal-oriented setting that forward predicts QoIs—including their uncertainties—from the inferred parameter field.

Our digital twin framework for goal-oriented real-time Bayesian inference combines the following key ideas: (1) an offline–online decomposition of the inverse solution that precomputes various mappings to enable the real-time application of the inverse operator and entirely circumvents the need for PDE solutions at the time of inference; (2) representation of the inverse operator in the data space instead of the high-dimensional parameter space; (3) extension of this real-time inference methodology to the goal-oriented setting for the prediction of QoIs under uncertainty; (4) exploitation of the shift invariance of autonomous dynamical systems to reduce the required number of PDE solutions by several orders of magnitude; and (5) exploitation of this same property to perform efficient FFT-based and multi-GPU-accelerated Hessian matvecs. We show how this framework can be applied to create digital twins for tsunami early warning from sea bottom pressure observations and a coupled acoustic-gravity PDE model. For a representative test problem with more than 10^8 parameters and 800 QoIs, both the parameter inference and the full end-to-end, data-to-inference-to-prediction computations are carried out within a fraction of a second.

2 Background

Notation. Lower-case italic letters (a) are used to denote infinite-dimensional functions or variables. Discretized functions (vectors in $\mathbb{R}^n$) are denoted by lower-case upright boldface ($\mathbf{a}$) notation. Upper-case calligraphic letters ($\mathcal{A}$) indicate infinite-dimensional operators. Discretized operators (matrices in $\mathbb{R}^{m\times n}$) are indicated by upper-case upright boldface ($\mathbf{A}$) notation. Lower-case italic (c) or upper-case italic (C) letters may also denote constants. Greek letters are used in different contexts, but their definitions will be made clear as they appear. Any exceptions to the above will be clear from context. The notation $\|\mathbf{a}\|_{\mathbf{A}}$ refers to the weighted Euclidean inner product $\mathbf{a}^T\mathbf{A}\mathbf{a}$; if no weight is given, the weight is identity. Estimates for computation times are given in h (hours), m (minutes), s (seconds), or ms (milliseconds) of wall-clock time. Storage estimates are based on double precision (i.e. eight bytes per floating point number) and given in PB (Petabytes), TB (Terabytes), GB (Gigabytes), or MB (Megabytes).

Linear time-invariant dynamical system. While our framework applies to linear autonomous dynamical systems in general, we consider the case of a linear time-invariant (LTI) dynamical system, which is applicable to the tsunami early warning application discussed in this paper. In particular, we consider an LTI dynamical system of the form

$$\begin{aligned} \frac{\partial w}{\partial t} &= \mathcal{A}w + \mathcal{C}m && \text{in } \Omega \times (0,T), \\ w &= w_0 && \text{in } \Omega \times \{0\}, \\ d &= \mathcal{B}w && \text{in } \Omega \times (0,T), \end{aligned} \tag{2.1}$$

with appropriate boundary conditions on $\partial\Omega \times (0,T)$, where Ω is the spatial domain, $(0,T)$ is the time domain, $w(x,t)$ is the system's state with initial state $w_0(x)$, parameter (input) $m(x,t)$ represents the source or forcing of the system and is independent of the state, and both $\mathcal{A}$ and $\mathcal{C}$ are time-invariant differential operators; $d(x,t)$ describes the observables (output) of the system, which are extracted from the state w via a time-invariant observation operator $\mathcal{B}$.

Discrete LTI system. Consider a discrete version of the LTI system obtained by discretizing (2.1) in time with a single-step explicit method[1] and with some spatial discretization scheme,

$$\mathbf{w}_{k+1} = \mathbf{A}\mathbf{w}_k + \mathbf{C}\mathbf{m}_k, \quad k = 0, 1, \cdots, N_t - 1, \tag{2.2}$$

where $\mathbf{w}_k \in \mathbb{R}^{N_w}$, $\mathbf{m}_k \in \mathbb{R}^{N_m}$, and the discrete time-stepping operator $\mathbf{A} \in \mathbb{R}^{N_w\times N_w}$ and $\mathbf{C} \in \mathbb{R}^{N_w\times N_m}$ both depend on the particular time-stepping scheme.[2] Then, using (2.2) the discretized LTI system can be written as follows:

[1] Note that the methodology is easily extended to multi-step explicit and implicit methods.

[2] For example, forward Euler time-stepping implies $\mathbf{A}$ and $\mathbf{C}$ respectively spatially discretize $(\mathcal{I} + \Delta t\mathcal{A})$ and $\Delta t\mathcal{C}$, where $\mathcal{I}$ is identity and Δt is the (uniform) time step size.

$$
\begin{aligned}
\mathbf{w}_1 &= \mathbf{A}\mathbf{w}_0 + \mathbf{C}\mathbf{m}_0, \\
\mathbf{w}_2 &= \mathbf{A}\mathbf{w}_1 + \mathbf{C}\mathbf{m}_1, \\
&= \mathbf{A}(\mathbf{A}\mathbf{w}_0 + \mathbf{C}\mathbf{m}_0) + \mathbf{C}\mathbf{m}_1 = \mathbf{A}^2\mathbf{w}_0 + \mathbf{A}^1\mathbf{C}\mathbf{m}_0 + \mathbf{A}^0\mathbf{C}\mathbf{m}_1, \\
&\vdots \\
\mathbf{w}_{k+1} &= \mathbf{A}^{k+1}\mathbf{w}_0 + \sum_{i=0}^{k} A^i C\mathbf{m}_{k-i}, \\
\mathbf{d}_{k+1} &= \mathbf{B}\mathbf{w}_{k+1},
\end{aligned}
\tag{2.3}
$$

where $\mathbf{d}_k \in \mathbb{R}^{N_d}$ and $\mathbf{B} \in \mathbb{R}^{N_d \times N_w}$ is the discrete observation operator.

Without loss of generality, assume homogeneous initial condition $\mathbf{u}_0 = \mathbf{0}$. We can then write the discretized LTI system in the following way:

$$
\begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \vdots \\ \mathbf{d}_{k+1} \\ \vdots \\ \mathbf{d}_{N_t} \end{bmatrix}
=
\begin{bmatrix}
\mathbf{B}\mathbf{A}^0\mathbf{C} & & & & & \\
\mathbf{B}\mathbf{A}^1\mathbf{C} & \mathbf{B}\mathbf{A}^0\mathbf{C} & & & & \\
\vdots & \vdots & \ddots & & & \\
\mathbf{B}\mathbf{A}^k\mathbf{C} & \mathbf{B}\mathbf{A}^{k-1}\mathbf{C} & \cdots & \mathbf{B}\mathbf{A}^0\mathbf{C} & & \\
\vdots & \vdots & & \vdots & \ddots & \\
\mathbf{B}\mathbf{A}^{N_t-1}\mathbf{C} & \mathbf{B}\mathbf{A}^{N_t-2}\mathbf{C} & \cdots & \mathbf{B}\mathbf{A}^{N_t-(k+1)}\mathbf{C} & \cdots & \mathbf{B}\mathbf{A}^0\mathbf{C}
\end{bmatrix}
\begin{bmatrix} \mathbf{m}_0 \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_k \\ \vdots \\ \mathbf{m}_{N_t-1} \end{bmatrix}.
\tag{2.4}
$$

We define $\mathbf{F}_{ij} \coloneqq \mathbf{B}\mathbf{A}^{i-j}\mathbf{C} \in \mathbb{R}^{N_d \times N_m}$, $i, j = 1, 2, \ldots, N_t, i \geq j$; the LTI system (2.2) can then be written more compactly as follows:

$$
\begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \\ \vdots \\ \mathbf{d}_{N_t} \end{bmatrix}
=
\begin{bmatrix}
\mathbf{F}_{11} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{F}_{21} & \mathbf{F}_{11} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{F}_{31} & \mathbf{F}_{21} & \mathbf{F}_{11} & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \mathbf{0} \\
\mathbf{F}_{N_t,1} & \mathbf{F}_{N_t-1,1} & \cdots & \mathbf{F}_{21} & \mathbf{F}_{11}
\end{bmatrix}
\begin{bmatrix} \mathbf{m}_0 \\ \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_{N_t-1} \end{bmatrix},
\tag{2.5}
$$

or very concisely as

$$
\mathbf{d} \coloneqq \mathbf{F}\mathbf{m}. \tag{2.6}
$$

We refer to $\mathbf{F}$ as the (discrete) *parameter-to-observable* (p2o) map; $\mathbf{m}$ is the parameter vector and $\mathbf{d}$ is the vector of observables or data vector. Then,

- $\mathbf{m} \in \mathbb{R}^{N_m N_t}$ with blocks $\mathbf{m}_j \in \mathbb{R}^{N_m}$, $j = 0, 1, \ldots, N_t - 1$;
- $\mathbf{d} \in \mathbb{R}^{N_d N_t}$ with blocks $\mathbf{d}_i \in \mathbb{R}^{N_d}$, $i = 1, 2, \ldots, N_t$;
- $\mathbf{F} \in \mathbb{R}^{(N_d N_t) \times (N_m N_t)}$ with blocks $\mathbf{F}_{ij} \in \mathbb{R}^{N_d \times N_m}$, $i, j = 1, 2, \ldots, N_t$.

It is clear from (2.5) that the p2o map $\mathbf{F}$ is shift-invariant with respect to its blocks $\mathbf{F}_{ij}$. In particular, $\mathbf{F}$ is *block Toeplitz*. Additionally, time causality implies that $\mathbf{F}$ is block lower-triangular. Section 4 describes how this special structure of $\mathbf{F}$ can be effectively exploited in the context of solving inverse problems.

Bayesian inverse problem. Given a system of the form (2.6), we consider the inverse problem of inferring the parameters $\mathbf{m}$ from the observed data $\mathbf{d}^{\text{obs}}$. In finite dimensions, Bayes theorem is given by:

$$\pi_{\text{post}}(\mathbf{m}|\mathbf{d}^{\text{obs}}) \propto \pi_{\text{like}}(\mathbf{d}^{\text{obs}}|\mathbf{m})\pi_{\text{prior}}(\mathbf{m}), \tag{2.7}$$

where $\pi_{\text{prior}}(\mathbf{m})$ is the prior probability density of the model parameters, $\pi_{\text{like}}(\mathbf{d}^{\text{obs}}|\mathbf{m})$ is the likelihood of the data $\mathbf{d}^{\text{obs}}$ given parameters $\mathbf{m}$, and $\pi_{\text{post}}(\mathbf{m}|\mathbf{d}^{\text{obs}})$ denotes the posterior density reflecting the probability of the parameters conditioned on the data.

We assume a Gaussian prior

$$\pi_{\text{prior}}(\mathbf{m}) \propto \exp\left\{-\frac{1}{2}\|\mathbf{m}-\mathbf{m}_{\text{prior}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{prior}(m)}}\right\}, \tag{2.8}$$

where $\mathbf{m}_{\text{prior}}$ and $\boldsymbol{\Gamma}_{\text{prior}(m)}$ are respectively the mean and covariance of the prior distribution.

Given data $\mathbf{d}^{\text{obs}}$ perturbed by additive Gaussian noise, i.e. $\mathbf{d}^{\text{obs}} = \mathbf{F}\mathbf{m} + \boldsymbol{\nu}$, $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}_{\text{noise}})$, the likelihood is

$$\pi_{\text{like}}(\mathbf{d}^{\text{obs}}|\mathbf{m}) \propto \exp\left\{-\frac{1}{2}\|\mathbf{F}\mathbf{m}-\mathbf{d}^{\text{obs}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}}\right\}, \tag{2.9}$$

For this choice of Gaussian prior and additive Gaussian noise, the posterior is then given by:

$$\pi_{\text{post}}(\mathbf{m}|\mathbf{d}^{\text{obs}}) \propto \exp\left\{-\frac{1}{2}\|\mathbf{F}\mathbf{m}-\mathbf{d}^{\text{obs}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} - \frac{1}{2}\|\mathbf{m}-\mathbf{m}_{\text{prior}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{prior}(m)}}\right\}. \tag{2.10}$$

The maximum-a-posterior (MAP) point $\mathbf{m}_{\text{map}}$ maximizes the posterior distribution over admissible parameters $\mathbf{m}$:

$$\mathbf{m}_{\text{map}} \coloneqq \operatorname*{arg\,min}_{\mathbf{m}\in\mathbb{R}^{N_m N_t}} \left(-\log \pi_{\text{post}}(\mathbf{m})\right) = \operatorname*{arg\,min}_{\mathbf{m}\in\mathbb{R}^{N_m N_t}} \frac{1}{2}\|\mathbf{F}\mathbf{m}-\mathbf{d}^{\text{obs}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} + \frac{1}{2}\|\mathbf{m}-\mathbf{m}_{\text{prior}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{prior}(m)}}. \tag{2.11}$$

The MAP point $\mathbf{m}_{\text{map}}$ is thus given by the solution to the linear inverse problem

$$\left(\mathbf{F}^*\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{F} + \boldsymbol{\Gamma}^{-1}_{\text{prior}(m)}\right)\mathbf{m}_{\text{map}} = \mathbf{F}^*\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{d}^{\text{obs}} + \boldsymbol{\Gamma}^{-1}_{\text{prior}(m)}\mathbf{m}_{\text{prior}}, \tag{2.12}$$

where $\mathbf{F}^*$ denotes the adjoint of the p2o map and $\mathbf{H} \coloneqq \left(\mathbf{F}^*\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{F} + \boldsymbol{\Gamma}^{-1}_{\text{prior}(m)}\right)$ is the Hessian of the negative log-posterior evaluated at $\mathbf{m}_{\text{map}}$.

The posterior for this linear inverse problem is then a Gaussian centered at the MAP point, i.e. $\pi_{\text{post}}(\mathbf{m}|\mathbf{d}^{\text{obs}}) \propto \mathcal{N}(\mathbf{m}_{\text{map}}, \boldsymbol{\Gamma}_{\text{post}(m)})$, with posterior covariance

$$\boldsymbol{\Gamma}_{\text{post}(m)} \coloneqq \mathbf{H}^{-1} = \left(\mathbf{F}^*\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{F} + \boldsymbol{\Gamma}^{-1}_{\text{prior}(m)}\right)^{-1}. \tag{2.13}$$

For a more thorough discussion of Bayesian inverse problems, see [16, 43] and references therein.

Goal-oriented Bayesian inference. In the goal-oriented setting, we consider an extension to the LTI system (2.1):

$$q = \mathcal{B}_q w \text{ in } \Omega \times (0, T), \tag{2.14}$$

where some quantity of interest (QoI), denoted by q, is extracted from the system's state via a time-invariant QoI observation operator $\mathcal{B}_q$.

Analogous to (2.6), the discrete LTI system for the QoI can be defined concisely by

$$\mathbf{q} \coloneqq \mathbf{F}_q\mathbf{m}, \tag{2.15}$$

where $\mathbf{F}_q$ is the discrete *parameter-to-QoI* (p2q) map and $\mathbf{q}$ is the QoI vector; that is,

- $\mathbf{q} \in \mathbb{R}^{N_q N_t}$ with blocks $\mathbf{q}_i \in \mathbb{R}^{N_q}$, $i = 1, 2, \ldots, N_t$;
- $\mathbf{F}_q \in \mathbb{R}^{(N_q N_t) \times (N_m N_t)}$ with blocks $(\mathbf{F}_q)_{ij} \in \mathbb{R}^{N_q \times N_m}$, $i, j = 1, 2, \ldots, N_t$.

Similar to the block-wise definition of the p2o map $\mathbf{F}$ in (2.5), $\mathbf{F}_q$ is defined by its blocks,

$$(\mathbf{F}_q)_{ij} \coloneqq \mathbf{B}_q \mathbf{A}^{i-j} \mathbf{C} \in \mathbb{R}^{N_q \times N_m}, \quad i, j = 1, 2, \ldots, N_t, i \geq j, \tag{2.16}$$

where $\mathbf{B}_q \in \mathbb{R}^{N_q \times N_w}$ is the discrete QoI observation operator. Like the p2o map $\mathbf{F}$, the p2q map $\mathbf{F}_q$ is block lower-triangular and shift-invariant with respect to its block, i.e. $\mathbf{F}_q$ is also block Toeplitz.

Remark. *In general, the number of time steps used by the time-stepping algorithm (N_t) may be different from the temporal discretization of the parameters (N_t^m), data (N_t^d), and QoIs (N_t^q). The block structure of the p2o and p2q maps then changes correspondingly but the same algorithmic techniques apply. For ease of presentation, we use $N_t = N_t^m = N_t^d = N_t^q$. The numerical example in Section 5 presents a case where $N_t^q \ll N_t^d = N_t^m \ll N_t$.*

Now, consider the problem of predicting the QoIs $\mathbf{q}$ from the observed data $\mathbf{d}^{\text{obs}}$ via the inference of parameters $\mathbf{m}$. Since the Bayesian prior and posterior of the parameter field are Gaussians, and a linear transformation of a Gaussian is still Gaussian, we can explicitly write down the prior and posterior for $\mathbf{q}$:

$$\pi_{\text{prior}}(\mathbf{q}) \propto \mathcal{N}\left(\mathbf{q}_{\text{prior}}, \mathbf{\Gamma}_{\text{prior}(q)}\right) = \mathcal{N}\left(\mathbf{F}_q \mathbf{m}_{\text{prior}}, \mathbf{F}_q \mathbf{\Gamma}_{\text{prior}(m)} \mathbf{F}_q^*\right), \tag{2.17}$$

$$\pi_{\text{post}}(\mathbf{q}|\mathbf{d}^{\text{obs}}) \propto \mathcal{N}\left(\mathbf{q}_{\text{map}}, \mathbf{\Gamma}_{\text{post}(q)}\right) = \mathcal{N}\left(\mathbf{F}_q \mathbf{m}_{\text{map}}, \mathbf{F}_q \mathbf{\Gamma}_{\text{post}(m)} \mathbf{F}_q^*\right), \tag{2.18}$$

where $\mathbf{F}_q^*$ is the adjoint p2q map. In other words, the means of the prior and posterior of the QoIs can be computed from the means of the prior and posterior of the inferred parameters via a push-forward through the p2q map; the respective covariances require an additional application of the adjoint p2q map.

Ultimately, goal-oriented Bayesian inference aims to estimate predictions of the QoIs from observed data without the need for inference of the parameters. In Section 4, we discuss how this can be accomplished in our mathematical and computational framework for real-time inference and prediction.

3 Tsunami Inverse Model

The Bayesian framework for real-time inference and prediction presented in this paper is applicable to a variety of application problems that are governed by PDE operators with underlying autonomous structure. Some examples will be given in the concluding remarks. However, rather than introducing this framework abstractly, we believe it is more descriptive to the reader to follow the presentation along with a concrete application problem in mind. For this reason, we introduce our motivating application in this section before discussing the framework step-by-step in the succeeding section, where this application will serve as an example both in the description of the framework and in giving concrete estimates of the memory and computational complexities involved in realizing it for a large-scale application (even if those estimates may differ considerably for other application problems).

Acoustic-gravity wave equations. The tsunami dynamics are modeled by the linear acoustic-gravity wave equations that couple the propagation of ocean acoustic waves with the surface gravity wave. This PDE model is derived by linearization of the conservation of mass and momentum around hydrostatic pressure in the compressible ocean, and the coupling to the surface gravity wave comes from a modified free surface boundary condition at the sea surface [32]. The model assumes that the surface gravity wave height is much smaller than the ocean depth, so the model may only be a good approximation in the deep ocean sufficiently far away from the coastline. We refer to [32] and references therein for details on the model derivation.

The model is formulated as a mixed problem in terms of the vector-valued velocity field u, the scalar-valued pressure field p, and the surface gravity wave height η:

$$\begin{cases} \rho \dfrac{\partial u}{\partial t} + \nabla p = 0, & \Omega \times (0,T), \\ K^{-1}\dfrac{\partial p}{\partial t} + \nabla \cdot u = 0, & \Omega \times (0,T), \\ p = \rho g \eta, & \partial\Omega_{\mathrm{s}} \times (0,T), \\ \dfrac{\partial \eta}{\partial t} = u \cdot n, & \partial\Omega_{\mathrm{s}} \times (0,T), \\ u \cdot n = -\dfrac{\partial b}{\partial t}, & \partial\Omega_{\mathrm{b}} \times (0,T), \\ u \cdot n = Z^{-1} p, & \partial\Omega_{\mathrm{a}} \times (0,T), \end{cases} \tag{3.1}$$

with homogeneous initial conditions; K and ρ are respectively the bulk modulus and density of seawater, $Z = \rho c$ is the acoustic wave impedance, $c = \sqrt{K/\rho}$ is the speed of sound in seawater, and $\partial b/\partial t$ is the seafloor motion; the spatial domain is denoted by Ω with boundaries $\partial\Omega_{\mathrm{s}}$ (sea surface), $\partial\Omega_{\mathrm{b}}$ (sea bottom), and $\partial\Omega_{\mathrm{a}}$ (lateral, absorbing boundaries); n is the outward unit normal; and the temporal domain is $(0,T)$.

The acoustic-gravity model (3.1) represents a LTI dynamical system of the form (2.1) and fits into the Bayesian inference framework laid out in the preceding section. The model parameter m, defined as the source for the acoustic-gravity wave problem (3.1), is given by the spatiotemporal seafloor motion, $m(x,t) \coloneqq -\partial b(x,t)/\partial t$, $(x,t) \in \partial\Omega_{\mathrm{b}} \times (0,T)$. The state is comprised of the velocity, pressure, and surface gravity wave height unknowns, $w \coloneqq (u,p,\eta)$, with homogeneous initial state. The observables of the system are pointwise observations of the pressure field, assumed to be obtained from pressure sensors mounted on the sea bottom, i.e. $d = \mathcal{B}w = p(x_j^d, t_i^d)$ where $x_j^d \in \partial\Omega_{\mathrm{b}}$, $j = 1,\dots,N_d$, are the seafloor sensor locations and $t_i^d \in (0,T)$, $i = 1,\dots,N_t^d$, are the time instances at which observations are made. The model QoIs are observations of the surface wave height, i.e. $q = \mathcal{B}_q w = \eta(x_j^q, t_i^q)$ where $x_j^q \in \partial\Omega_{\mathrm{s}}$, $j = 1,\dots,N_q$, and $t_i^q \in (0,T)$, $i = 1,\dots,N_t^q$, are specific locations and time instances pertinent to tsunami early warning.

The forward problem (3.1) can be cast into a mixed variational formulation and then discretized with finite elements and explicit time-stepping (see [21] for details). The discrete expressions of the p2o map, $\mathbf{d} = \mathbf{F}\mathbf{m}$ (cf. (2.6)), and the p2q map, $\mathbf{q} = \mathbf{F}_q\mathbf{m}$ (cf. (2.15)), are then clearly defined as the mappings of the parameter input $\mathbf{m}$ (seafloor motion) to the observables $\mathbf{d}$ (sea bottom pressure) and the QoIs $\mathbf{q}$ (surface gravity wave height) via the solution to the discretized equations of the acoustic-gravity PDE model (3.1).

The inverse problem can then be stated as follows: given pressure recordings $\mathbf{d}^{\mathrm{obs}}$ from sensors on the seafloor, infer the spatiotemporal seafloor motion $\mathbf{m}$ in the subduction zone. And the tsunami prediction problem is: given inferred parameters $\mathbf{m}$, forward predict the surface gravity wave heights $\mathbf{q}$. In the goal-oriented setting, the task is to directly predict $\mathbf{q}$ from the observations $\mathbf{d}^{\mathrm{obs}}$ without the need to explicitly infer $\mathbf{m}$ (even if this inverse problem is embedded within the goal-oriented inference of $\mathbf{q}$).

Adjoint equations. Solving the goal-oriented inverse problem (i.e. characterizing the Bayesian posterior (2.18)) requires us to have access to the Hessian action (2.13) which involves the adjoints of the p2o and p2q maps, $\mathbf{F}^*$ and $\mathbf{F}_q^*$. There are two common approaches for deriving the adjoint problems: discretize-then-optimize (DTO) and optimize-then-discretize (OTD). Each approach has its benefits and drawbacks, see [20] for a detailed discussion. In general, the discretized adjoint problems derived by DTO and OTD do not match.[3] If we had direct access to the discretized maps $\mathbf{F}$ and $\mathbf{F}_q$, then the DTO approach could be realized by simply transposing these maps. Typically, this is not a practical method because we usually are not able to precompute and store these matrices but only have access to their action on a vector which requires solving the PDEs.[4] Here, we show the adjoint equations on the infinite-dimensional level, consistent with an OTD approach. Similar to the forward problem (3.1), the adjoint PDEs are given as a mixed problem in terms of the adjoint velocity field τ, the adjoint pressure v, and the adjoint surface gravity wave height ξ:

$$\begin{cases} \rho \dfrac{\partial \tau}{\partial t} + \nabla v = 0, & \Omega \times (0, T), \\ K^{-1} \dfrac{\partial v}{\partial t} + \nabla \cdot \tau = 0, & \Omega \times (0, T), \end{cases} \tag{3.2}$$

with homogeneous terminal conditions; the BCs for the adjoint p2o map are

$$\begin{cases} v = \rho g \xi, & \partial\Omega_{\mathrm{s}} \times (0, T), \\ \dfrac{\partial \xi}{\partial t} = \tau \cdot n, & \partial\Omega_{\mathrm{s}} \times (0, T), \\ \tau \cdot n = -\mathcal{B}^* \tilde{d}, & \partial\Omega_{\mathrm{b}} \times (0, T), \\ \tau \cdot n = -Z^{-1} v, & \partial\Omega_{\mathrm{a}} \times (0, T), \end{cases} \tag{3.3}$$

where $\tilde{d}$ is the source of the adjoint problem, defined at the observation points.[5]

For the adjoint p2q map, the BCs change; in particular, the boundary source term, denoted by $\tilde{q}$, is now defined on the sea surface, and we instead obtain homogeneous BCs on the sea bottom:

$$\begin{cases} v = \rho g \xi, & \partial\Omega_{\mathrm{s}} \times (0, T), \\ \dfrac{\partial \xi}{\partial t} = \tau \cdot n + \mathcal{B}_q^* \tilde{q}, & \partial\Omega_{\mathrm{s}} \times (0, T), \\ \tau \cdot n = 0, & \partial\Omega_{\mathrm{b}} \times (0, T), \\ \tau \cdot n = -Z^{-1} v, & \partial\Omega_{\mathrm{a}} \times (0, T). \end{cases} \tag{3.4}$$

Analogous to the forward problem, these adjoint problems can be cast into mixed variational formulations and then discretized with finite elements and a time-stepping scheme to obtain discrete representations of the adjoint p2o and p2q maps $\mathbf{F}^*$ and $\mathbf{F}_q^*$. Each action of $\mathbf{F}^*$ and $\mathbf{F}_q^*$ then corresponds to mapping the respective source terms $\tilde{\mathbf{d}}$ and $\tilde{\mathbf{q}}$ to the parameter space via the solution to the discretized equations of the adjoint acoustic-gravity PDEs (3.2)+(3.3) and (3.2)+(3.4), respectively.

[3] Depending the discretization scheme, DTO and OTD may yield consistent adjoint problems. For the numerical example presented in Section 5, this is in fact the case.

[4] In the case where $\mathbf{F}$ and $\mathbf{F}_q$ represent discretized LTI systems, they can be compactly stored; however, precomputing them efficiently still requires access to the action of the adjoints $\mathbf{F}^*$ and $\mathbf{F}_q^*$ (as will be shown in Section 4).

[5] Technically, the adjoint observation operator $\mathcal{B}^*$ maps to the adjoint state space, so $\mathcal{B}^* \tilde{d}$ could be written as a source term on the right-hand side of (3.2).

Hessian action. The linear inverse problem (2.12) can be solved with an iterative solver like the conjugate gradient method, which requires numerous actions of the Hessian on a vector. Explicit construction of the discrete Hessian itself is usually prohibitive, because it necessitates as many forward PDE solves as there are parameters, leading to the need for performing matrix-free Hessian matvecs instead. Each action of the adjoint-based Hessian $\mathbf{H}$ involves matvecs with both $\mathbf{F}$ and $\mathbf{F}^*$ (cf. (2.13)). Matrix-free Hessian matvecs therefore require solution of a pair of forward/adjoint PDE solves per Hessian action. Unfortunately, solving (2.12) in this way is not feasible in the case of the tsunami inverse problem. Its massive scale and the need for real-time solution preclude solving the forward/adjoint PDE model at the time of inference. Instead, as will be shown in Section 4, we exploit the block Toeplitz structure of $\mathbf{F}$ and $\mathbf{F}^*$, as well as $\mathbf{F}_q$ and $\mathbf{F}_q^*$, which implies that we can precompute and store these matrices, and it enables Hessian matvecs that are orders of magnitude faster and more efficient than matrix-free actions via pairs of forward/adjoint PDE solves.

Discretization of the CSZ tsunami inverse problem. In order to present the reader with concrete examples of the memory and computational complexities involved in our real-time inference and prediction methodology (presented in Section 4), we provide estimates based on the problem size for the full-scale CSZ digital twin for tsunami early warning. We reemphasize that the methodology presented below can be applied to many other applications whose problem dimensions may be quite different from these estimates.

The CSZ is more than 1000 km long, stretching from off the coast of Northern California to beyond the Northwest peak of Vancouver Island. Discretizing the full-order acoustic-gravity PDE model along the entire subduction zone and extending near the Pacific–Northwest coastline yields a forward problem with $N_w \sim 10^{10}$ states and $N_m \sim 10^6$ parameters per time step (i.e. 10^{10} and 10^6 spatial degrees of freedom, respectively).[6]

Sea bottom sensors in the Cascadia region are very sparse but various efforts are proceeding to instrument the CSZ, as well as other subduction zones around the world, to provide real-time data for tsunami early warning. For the CSZ digital twin, observations are assumed to be obtained from seafloor measurements recorded by hundreds of sensors, i.e. $N_d \sim 10^2$, and tsunami predictions are made in a few specified forecasting locations, i.e. $N_q \sim 10$ (e.g. near major population centers). The dominant frequencies of earthquake-induced seafloor motion and the ensuing ocean acoustic waves are typically below 1 Hz (though sea bottom pressure sensors may record higher frequencies) [29]. Given the proximity of the CSZ to the coastline, where the tsunami may reach population centers within 20 minutes of the earthquake, the observation time prior to inference and prediction with the CSZ digital twin to provide early warning is on the order of a few hundred or at most one thousand seconds, so $N_t \sim 10^3$.

4 Methodology

The Bayesian posteriors of the parameters $\mathbf{m}$ and the QoIs $\mathbf{q}$, derived in Section 2, are respectively given by $\pi_{\text{post}}(\mathbf{m}|\mathbf{d}^{\text{obs}}) \propto \mathcal{N}(\mathbf{m}_{\text{map}}, \mathbf{\Gamma}_{\text{post}(m)})$ and $\pi_{\text{post}}(\mathbf{q}|\mathbf{d}^{\text{obs}}) \propto \mathcal{N}\left(\mathbf{q}_{\text{map}}, \mathbf{\Gamma}_{\text{post}(q)}\right)$, where

$$\mathbf{\Gamma}_{\text{post}(m)} = \left(\mathbf{F}^*\mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{F} + \mathbf{\Gamma}_{\text{prior}(m)}^{-1}\right)^{-1} \quad , \quad \mathbf{m}_{\text{map}} = \mathbf{\Gamma}_{\text{post}}\left(\mathbf{F}^*\mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{d}^{\text{obs}} + \mathbf{\Gamma}_{\text{prior}(m)}^{-1}\mathbf{m}_{\text{prior}}\right), \tag{4.1}$$

$$\mathbf{\Gamma}_{\text{post}(q)} = \mathbf{F}_q\mathbf{\Gamma}_{\text{post}(m)}\mathbf{F}_q^* \quad , \quad \mathbf{q}_{\text{map}} = \mathbf{F}_q\mathbf{m}_{\text{map}}. \tag{4.2}$$

The ultimate goal of Bayesian inference is to characterize these posteriors; for example, computing their means, estimating their covariances, or drawing samples. For inverse problems where the

[6] The number of time steps for the state is determined by the Courant–Friedrichs–Lewy (CFL) condition.

effective rank of the Hessian $\mathbf{H} := \mathbf{\Gamma}_{\text{post}(m)}^{-1}$ is high (e.g. hyperbolic systems like the acoustic-gravity wave equations), it takes many iterations for an iterative solver such as the conjugate gradient method to solve for $\mathbf{m}_{\text{map}}$ in (4.1) using actions of $\mathbf{H}$. Moreover, conventionally each application of $\mathbf{H}$ requires a pair of forward and adjoint wave equation solves, which is expensive and precludes real-time inference. Here, we present an alternative: the tsunami dynamics are governed by an autonomous system whose operator does not have explicit time dependence. This fact (and causality) manifests in the time-shift invariance of the p2o and p2q maps ($\mathbf{F}$ and $\mathbf{F}_q$), which results in $\mathbf{F}$ and $\mathbf{F}_q$ being block lower-triangular Toeplitz matrices (cf. (2.5)), enabling extremely fast and efficient matvecs with $\mathbf{F}$, $\mathbf{F}_q$, and their respective adjoints (see below). In this section, we introduce a 4-phase framework for real-time inference and prediction; this framework relies on the following key ideas:

- Efficient and fast FFT-based multi-GPU-accelerated actions of the discretized p2o and p2q operators and their respective adjoints.
- Compact storage of the p2o and p2q maps in block Toeplitz form.
- An offline-online decomposition of the inference and prediction problem into precomputation phases (offline) and a real-time inference and prediction phase (online).
- Shift of the computation of the Hessian inverse from the very large parameter space to the much lower-dimensional data space.

In the remainder of this section, N_t refers to the number of time steps in the discrete parameter field, data, and QoIs;[7] N_m refers to the number of spatial degrees of freedom in the discrete parameter field; N_d to the number of sensor locations; and N_q to the number of spatial locations for QoI prediction. While our methodology does not assume specific values for any of these parameters, an efficient realization of the framework does require the following assumptions:

$$N_d \ll N_m, \;\; N_q \ll N_m, \;\; \text{and} \;\; N_t \gg 1. \tag{4.3}$$

In the context of inverse problems governed by autonomous dynamical systems, these assumptions are not very restrictive. In particular, many applications are data-constrained, i.e. they seek to infer high-dimensional parameter fields from sparse measurements. And in the goal-oriented setting, we often aim to predict a limited number of QoIs which are typically much fewer than the parameter degrees of freedom. Recall from the preceding section that our estimates for the full-scale CSZ digital twin are $N_m \sim 10^6$, $N_d \sim 10^2$, $N_q \sim 10$, $N_t \sim 10^3$. It is easily verified that these problem dimensions fit well within the intended scope of the framework.

FFT-based GPU-accelerated p2o and p2q matvecs. To solve the parameter inference and QoI prediction problem (i.e. characterize the corresponding Bayesian posteriors), we require many matvecs with the p2o and p2q maps ($\mathbf{F}$ and $\mathbf{F}_q$) and their respective adjoints ($\mathbf{F}^*$ and $\mathbf{F}_q^*$). Matrix-free actions of those maps and their adjoints on a vector involve the solution of the forward and adjoint acoustic-gravity wave equations, respectively, making them computationally expensive.

However, if we have direct access to the matrices $\mathbf{F}$ and $\mathbf{F}_q$, we can exploit their block lower-triangular Toeplitz structure (2.5). Fast matvecs are achieved by embedding the respective block Toeplitz matrix within a block circulant matrix, which is block-diagonalized by the discrete Fourier

[7] For ease of presentation, WLOG, we use the same time step for parameters, observables, and QoIs. In practice, each one may have a different time step as needed.

transform (DFT) [18]. The DFT only needs to be done once for each matrix. The matvec then becomes a block-diagonal matvec operation in Fourier space.[8] Moreover, because the FFT is a unitary operator, the action of the adjoint p2o and p2q maps correspond to simply applying the complex conjugates in Fourier space, eliminating the need to separately store the Fourier-transformed forward and adjoint maps. These FFT-based p2o and p2q matvecs can be implemented efficiently on multi-GPU clusters [42].

In comparison to the conventional matrix-free actions involving forward and adjoint PDE solves, the FFT-based algorithm trades off computational complexity (PDE solves) with memory complexity (storing the matrices). Conventionally, computing or storing the full p2o map is not tractable for large-scale inverse problems, because doing so requires one forward PDE solve per column of the matrix, and the number of columns corresponds to the dimension of the parameter space ($\sim 10^9$). For linear inverse problems governed by autonomous dynamical systems, however, the autonomous structure of the p2o and p2q maps both reduces the number of PDE solves required and enables compact storage of these maps (see Phase 1 below).

The main advantages of the FFT-based matvec algorithm are that it (1) avoids PDE solves entirely, thus making it independent of the cost of state discretization, time-stepping constraints from CFL condition, and computing on unstructured mesh-based data structures dictated by the PDE discretization; and (2) relies on arithmetic operations that involve data that is contiguous in memory and leverages routines from libraries such as cuBLAS that are well known to achieve high performance on GPUs. A quantitative comparison between the FFT-based GPU-accelerated algorithm and the conventional approach of matrix-free actions via forward and adjoint PDE solves for a representative test problem is presented in Section 5.

Phase 1 (offline): Precompute p2o and p2q maps. Phase 1 of our framework is concerned with precomputing and storing the p2o and p2q maps, so that their actions are later available (during Phases 2–4) at the much lower computational cost enabled by the FFT-based algorithm and with the speedup provided due to the algorithm's suitability for multi-GPU execution.

The p2o and p2q maps can be computed either from forward PDE solutions (one per column) or from adjoint PDE solutions (one per row). Because the data dimension ($N_d N_t \sim 10^5$) and QoI dimension ($N_q N_t \sim 10^4$) are each much smaller than the parameter dimension ($N_d N_t \sim 10^9$), computing rows via adjoint PDE solutions is the more efficient approach. The time-shift invariance of the PDEs (3.1) means that only one adjoint PDE solve per sensor location is required to create $\mathbf{F}$, and one adjoint PDE solve per QoI spatial location to create $\mathbf{F}_q$. For $\mathbf{F}$, these adjoint PDEs are solved backward in time with pressure "sources" at each sensor location; for $\mathbf{F}_q$, they are solved with pressure sources at each QoI point in space. Translating these solutions in time gives the entire $\mathbf{F}$ and $\mathbf{F}_q$. The block Toeplitz structure of $\mathbf{F}$ and $\mathbf{F}_q$ also reduces the memory requirements to store these matrices by a factor of N_t. This means only $\sim$0.8 TB memory is required to store $\mathbf{F}$ instead of $\sim$800 TB, and $\sim$80 GB to store $\mathbf{F}_q$ instead of $\sim$80 TB.

While $N_d + N_q \sim \mathcal{O}(10^2)$ adjoint acoustic-gravity wave-equation solves incur considerable computational expense, this entire computation happens offline and only needs to be done once. For the discretized CSZ digital twin (with $\sim 10^{10}$ states and $\sim 10^3$ time steps), a scalable explicit finite element solver employed on a moderately large compute cluster is able to compute one adjoint acoustic-gravity wave-equation solution per hour.[9]

[8] The matvec in Fourier space can also be computed as an elementwise vector operation for each block followed by a row-wise reduction over blocks. However, this requires strided memory access patterns which negatively impacts GPU performance; see [42] for details.

[9] This time estimate was obtained from a benchmark of the acoustic-gravity wave equation solve performed on 512 compute nodes of TACC's *Frontera* supercomputer using an explicit finite element solver implemented in MFEM.

Phase 2 (offline): Compute a compact representation of $\boldsymbol{\Gamma}_{\mathbf{post}(m)}$. For the CSZ digital twin, the discretized parameter dimension is $N_m N_t \sim 10^9$. The linearity of the inverse problem enables us, in principle, to precompute both the mapping from observed data $\mathbf{d}^{\text{obs}}$ to the MAP point $\mathbf{m}_{\text{map}}$ as well as the posterior covariance $\boldsymbol{\Gamma}_{\text{post}(m)}$:

$$\mathbf{m}_{\text{map}} = \underbrace{\mathbf{H}^{-1}\mathbf{F}^*\boldsymbol{\Gamma}_{\text{noise}}^{-1}}_{\text{precompute}}\mathbf{d}^{\text{obs}} + \underbrace{\mathbf{H}^{-1}\boldsymbol{\Gamma}_{\text{prior}(m)}^{-1}\mathbf{m}_{\text{prior}}}_{\text{precompute}}, \tag{4.4}$$

$$\boldsymbol{\Gamma}_{\text{post}(m)} = \mathbf{H}^{-1} = \underbrace{\left(\mathbf{F}^*\boldsymbol{\Gamma}_{\text{noise}}^{-1}\mathbf{F} + \boldsymbol{\Gamma}_{\text{prior}(m)}^{-1}\right)^{-1}}_{\text{precompute}}. \tag{4.5}$$

Unlike the p2o map, the inverse of the Hessian does not inherit the block Toeplitz structure from the discretized autonomous dynamical system, so computing and storing the full covariance matrix $\boldsymbol{\Gamma}_{\text{post}(m)}$ of size $\sim 10^9 \times 10^9$ is intractable. Moreover, the forward model (3.1) is hyperbolic, implying a slow decay of the eigenvalues of the Hessian, thereby prohibiting the effectiveness of traditional low-rank decomposition methods [16].

To circumvent this issue, we employ the Sherman–Morrison–Woodbury formula [38, 45], with which $\boldsymbol{\Gamma}_{\text{post}(m)}$ can be rewritten as

$$\boldsymbol{\Gamma}_{\text{post}(m)} = \boldsymbol{\Gamma}_{\text{prior}(m)}\left(\mathbf{I} - \mathbf{F}^*\mathbf{K}^{-1}\mathbf{F}\boldsymbol{\Gamma}_{\text{prior}(m)}\right), \tag{4.6}$$

where $\mathbf{K} = \boldsymbol{\Gamma}_{\text{noise}} + \mathbf{F}\boldsymbol{\Gamma}_{\text{prior}(m)}\mathbf{F}^*$. Expressing $\boldsymbol{\Gamma}_{\text{post}(m)}$ in this form shifts the problem of inverting an operator of the dimension of the parameter space $N_m N_t$ to one of the dimension of the data space $N_d N_t$. For the CSZ digital twin, this corresponds to a matrix $\mathbf{K}$ of size $10^5 \times 10^5$, whereas the original $\boldsymbol{\Gamma}_{\text{post}(m)}$ was of size $10^9 \times 10^9$. This matrix can be Cholesky-factorized offline, and triangular solves are done in the online phase.

To compute matrix $\mathbf{K}$, Phase 2 of our framework relies on the efficient FFT-based algorithm to perform actions of the p2o map $\mathbf{F}$ and its adjoint $\mathbf{F}^*$ precomputed in Phase 1. Computing $\mathbf{K}$ column-by-column requires 10^5 matvecs with $\mathbf{F}$ and $\mathbf{F}^*$ each. Using the conventional approach of matrix-free actions via forward and adjoint PDE solves to perform 10^5 matvecs would have made this task infeasible. As we will see for a numerical example in Section 5, the FFT-based algorithm can perform these matvecs for large-scale problems within a reasonably short time on moderate compute resources.

Computing $\mathbf{K}$ also requires actions of the noise covariance $\boldsymbol{\Gamma}_{\text{noise}}$ and the prior covariance $\boldsymbol{\Gamma}_{\text{prior}(m)}$. We assume that $\boldsymbol{\Gamma}_{\text{noise}}$ is a diagonal matrix, making the cost of applying $\boldsymbol{\Gamma}_{\text{noise}}$ or $\boldsymbol{\Gamma}_{\text{noise}}^{-1}$ to a vector negligible. For the prior covariance, a common choice is to employ a smoothing prior given in the form of a discretized inverse elliptic differential operator [16, 39, 31]. Performing 10^5 solves of a discretized elliptic PDE operator of size 10^9 is a tractable problem thanks to the existence of fast and scalable solvers for such systems, but it can become a bottleneck in the computation of $\mathbf{K}$. However, many times the elliptic PDE operator whose inverse generates the prior covariance may itself be time-invariant (we use one such example in Section 5), so that the matrix $\mathbf{G}^* := \boldsymbol{\Gamma}_{\text{prior}(m)}\mathbf{F}^*$ is also block Toeplitz. In this case, $\mathbf{G}^*$ can be precomputed at the cost of only $N_d \sim 10^2$ prior solves (which can be batched over time steps) and stored at the same expense as storing $\mathbf{F}$. Matvecs with $\mathbf{G}^*$ can then be performed with the FFT-based algorithm and incur the same computational cost as matvecs with $\mathbf{F}^*$. With the definition of $\mathbf{G}^*$, the updated expression for the Hessian inverse is

$$\boldsymbol{\Gamma}_{\text{post}(m)} = \left(\mathbf{I} - \mathbf{G}^*\mathbf{K}^{-1}\mathbf{F}\right)\boldsymbol{\Gamma}_{\text{prior}(m)} \tag{4.7}$$

where $\mathbf{K} = \boldsymbol{\Gamma}_{\text{noise}} + \mathbf{F}\mathbf{G}^*$.

It must be emphasized that Hessian actions given by (4.7) are exact up to discretization error. In other words, compact representation of $\mathbf{\Gamma}_{\text{post}(m)}$ here only refers to an efficient representation of the inverse operator in the data space dimension, which was obtained through applying the Sherman–Morrison–Woodbury formula to $\mathbf{\Gamma}_{\text{post}(m)}$ (cf. (4.6)), and it does *not* refer to an approximation of the Hessian action through representing $\mathbf{\Gamma}_{\text{post}(m)}$ in some lower-dimensional space.

Having precomputed $\mathbf{G}^*$ and prefactorized $\mathbf{K}$, Hessian actions via (4.7) with the multi-GPU-accelerated FFT-based matvecs of $\mathbf{F}$ and $\mathbf{G}^*$ are extremely fast and efficient, and they enable the real-time inference of the parameter MAP point, as will be shown in Phase 4. In fact, as we will demonstrate with numerical results in Section 5, the Hessian matvec computation time is now primarily limited by the prior solve of $\mathbf{\Gamma}_{\text{prior}(m)}$. Despite this efficiency of Hessian matvecs, computing the full posterior covariance may still be unattainable due to the memory cost of storing columns of $\mathbf{\Gamma}_{\text{post}(m)}$ as well as the computational cost of performing as many as $\sim 10^9$ Hessian matvecs. However, obtaining a subset of the parameter uncertainties from the pointwise variance field, given by $\text{diag}(\mathbf{\Gamma}_{\text{post}(m)})$, is now feasible by computing columns of $\mathbf{\Gamma}_{\text{post}(m)}$.

Lastly, if the prior mean is non-zero ($\mathbf{m}_{\text{prior}} \neq \mathbf{0}$), we can now easily precompute its contribution to the MAP point of the posterior (cf. (4.4)):

$$\mathbf{m}_{\text{map}}^{\text{prior}} \coloneqq \mathbf{\Gamma}_{\text{post}(m)}\mathbf{\Gamma}_{\text{prior}(m)}^{-1}\mathbf{m}_{\text{prior}} = \left(\mathbf{I} - \mathbf{G}^*\mathbf{K}^{-1}\mathbf{F}\right)\mathbf{m}_{\text{prior}}. \tag{4.8}$$

Phase 3 (offline): Compute QoI uncertainties and data-to-QoI map. Many applications of interest, including tsunami early warning, ultimately require a real-time forecast (with uncertainties) for one or multiple specified QoIs. To achieve this, we can exploit the fast Hessian actions enabled by the compact representation of $\mathbf{\Gamma}_{\text{post}(m)}$, precomputed in Phase 2, to compute uncertainties for the QoIs $\mathbf{q}$ via

$$\mathbf{\Gamma}_{\text{post}(q)} = \mathbf{F}_q\mathbf{\Gamma}_{\text{post}(m)}\mathbf{F}_q^* = \mathbf{F}_q\left(\mathbf{I} - \mathbf{G}^*\mathbf{K}^{-1}\mathbf{F}\right)\mathbf{\Gamma}_{\text{prior}(m)}\mathbf{F}_q^*. \tag{4.9}$$

In the case of a time-invariant prior, we can use the same trick as in computing $\mathbf{G}^*$ (for faster Hessian matvecs) to accelerate computation of (4.9) by precomputing the block Toeplitz matrix $\mathbf{G}_q^* \coloneqq \mathbf{\Gamma}_{\text{prior}(m)}\mathbf{F}_q^*$ via $N_q \sim 10$ prior solves (one for each column of the last block column of $\mathbf{F}_q^*$). The posterior covariance for the QoIs can then be written as

$$\mathbf{\Gamma}_{\text{post}(q)} = \mathbf{F}_q\left(\mathbf{I} - \mathbf{G}^*\mathbf{K}^{-1}\mathbf{F}\right)\mathbf{G}_q^*. \tag{4.10}$$

The covariance matrix $\mathbf{\Gamma}_{\text{post}(q)}$ of size $\sim 10^4 \times 10^4$ can now be computed through $\sim 10^4$ actions of $\mathbf{\Gamma}_{\text{post}(q)}$ via (4.10), giving access to the Bayesian posterior of $\mathbf{q}$ once $\mathbf{q}_{\text{map}}$ has been computed.

During the online phase of real-time inference and prediction, described in Phase 4, the QoI MAP point $\mathbf{q}_{\text{map}}$ could, in principle, be computed through push-forward of the inferred parameters $\mathbf{m}_{\text{map}}$ via the operator $\mathbf{F}_q$ (cf. (4.2)). Using the FFT-based matvec of $\mathbf{F}_q$, this is entirely feasible in real time. However, this requires (1) first solving the real-time inference of $\mathbf{m}_{\text{map}}$; (2) storing the matrices $\mathbf{K}$, $\mathbf{F}$, $\mathbf{F}_q$, and $\mathbf{G}^*$ for use during the online phase; and (3) online access to a multi-GPU cluster to compute fast actions of these operators.

A more efficient alternative for the real-time QoI forecast is available by precomputing the data-to-QoI (d2q) map

$$\mathbf{Q} \coloneqq \mathbf{F}_q\mathbf{\Gamma}_{\text{post}(m)}\mathbf{F}^*\mathbf{\Gamma}_{\text{noise}}^{-1} = \mathbf{F}_q\left(\mathbf{I} - \mathbf{G}^*\mathbf{K}^{-1}\mathbf{F}\right)\mathbf{G}^*\mathbf{\Gamma}_{\text{noise}}^{-1}, \tag{4.11}$$

which can then be used to directly map observed data to the QoI without the need to first infer the parameter $\mathbf{m}_{\text{map}}$. This in turn drastically reduces the storage and compute requirements for the QoI prediction during online phase (see Phase 4 below). The d2q map is of size $\sim 10^4 \times 10^5$, so it can be computed through $\sim 10^4$ actions of $\mathbf{Q}$ via (4.11).

We remark that a similar computation could have been performed to precompute the data-to-parameter map given by $\boldsymbol{\Gamma}_{\text{post}(m)}\mathbf{F}^*\boldsymbol{\Gamma}_{\text{noise}}^{-1}$. However, for the problem scale we are interested in, the memory requirements for storing this mapping (~ 800 TB) together with the I/O time required for reading it from disk during the online phase make this difficult to use in real time. In other applications—particularly where the parameter dimension is a few orders of magnitude smaller than for the CSZ digital twin—this may be an attractive approach.

Finally, the contribution of a non-zero prior mean ($\mathbf{m}_{\text{prior}} \neq \mathbf{0}$) to the MAP point of the QoI can also be precomputed (cf. (4.2) and (4.8)):

$$\mathbf{q}_{\text{map}}^{\text{prior}} \coloneqq \mathbf{F}_q\mathbf{m}_{\text{map}}^{\text{prior}} = \mathbf{F}_q\boldsymbol{\Gamma}_{\text{post}(m)}\boldsymbol{\Gamma}_{\text{prior}(m)}^{-1}\mathbf{m}_{\text{prior}} = \mathbf{F}_q\left(\mathbf{I} - \mathbf{G}^*\mathbf{K}^{-1}\mathbf{F}\right)\mathbf{m}_{\text{prior}}. \tag{4.12}$$

Phase 4 (online): Real-time inference and prediction. After the various stages of offline computation during Phases 1–3, the online phase reduces to the tasks of (1) real-time inference of the parameters $\mathbf{m}_{\text{map}}$ and (2) real-time prediction of the QoIs $\mathbf{q}_{\text{map}}$.

Using the derived expression (4.7) for fast Hessian matvecs, the inference of $\mathbf{m}_{\text{map}}$ from real-time observations $\mathbf{d}^{\text{obs}}$ can be calculated via

$$\mathbf{m}_{\text{map}} = \left(\mathbf{I} - \mathbf{G}^*\mathbf{K}^{-1}\mathbf{F}\right)\mathbf{G}^*\boldsymbol{\Gamma}_{\text{noise}}^{-1}\mathbf{d}^{\text{obs}} + \mathbf{m}_{\text{map}}^{\text{prior}} \tag{4.13}$$

in real time (order of seconds) for $\sim 10^9$ parameters. This real-time inference relies on the FFT-based GPU-accelerated matvecs with the precomputed block Toeplitz matrices $\mathbf{F}$ and $\mathbf{G}^*$, and triangular solves of the prefactorized matrix $\mathbf{K}$.

The real-time prediction for the QoI employs the precomputed d2q map $\mathbf{Q}$ (a dense matrix of size $\sim 10^4 \times 10^5$). The prediction of $\mathbf{q}_{\text{map}}$ can then be obtained directly from observations $\mathbf{d}^{\text{obs}}$ via

$$\mathbf{q}_{\text{map}} = \mathbf{Q}\mathbf{d}^{\text{obs}} + \mathbf{q}_{\text{map}}^{\text{prior}} \tag{4.14}$$

for $\sim 10^4$ QoIs within fractions of a second. Importantly, due to its relatively small size, the d2q map can be stored with about ~ 8 GB of memory, and the dense matvec of $\mathbf{Q}$ is computationally cheap. This makes the action of $\mathbf{Q}$ available for online computation on a laptop. With precomputed $\mathbf{Q}$, this QoI prediction is accomplished without the need for inference of $\mathbf{m}_{\text{map}}$; therefore, (4.14) enables high-fidelity (exact up to discretization error) goal-oriented Bayesian inference in real time without the need for online access to large-scale compute resources.

If this goal-oriented setting allows us to obtain real-time predictions of the QoIs directly from the observed data, then one might ask why we still care for the inference of the parameters. Inferring the earthquake-induced spatiotemporal seafloor motion is of interest, because this inverse solution can be used as a source for tsunami propagation in other models, including nonlinear tsunami propagation models that account for coastal inundation and other features that are not represented in the linearized acoustic-gravity PDE model.

5 Numerical Example

In this section, the end-to-end digital twin framework—from noisy observations via parameter inference to QoI prediction with uncertainties—is demonstrated with a 3D test configuration of the acoustic-gravity wave equations. We employ synthetic observations (obtained from the PDE model) with additive noise as pressure recordings from seafloor sensors to infer the underlying spatiotemporal seafloor motion and forward predict tsunami wave heights at specified locations in real time.

For the Gaussian prior, we use prior mean $\mathbf{m}_{\text{prior}} = \mathbf{0}$, and the prior covariance $\mathbf{\Gamma}_{\text{prior}(m)}$ is generated from the inverse of a discretized elliptic differential operator:

$$\mathbf{\Gamma}_{\text{prior}(m)} \coloneqq (\alpha_1 I_h - \alpha_2 \Delta_h)^{-2}, \tag{5.1}$$

where $\alpha_1 > 0$, $\alpha_2 > 0$, I is identity, Δ denotes the Laplacian $(\partial^2/\partial x^2 + \partial^2/\partial y^2)$, and h indicates discretization; we employ Robin boundary conditions to mitigate boundary artifacts [13]. This choice of prior is equivalent to a Matérn covariance kernel [31] and it guarantees bounded pointwise variance [39]. The prior covariance (5.1) smoothes rough components of the parameter field. The premise for using a smoothing prior operator is that the inference of rough components is unstable because they are not sufficiently informed by the data [16]. Note that this prior has time-invariant parameters α_1 and α_2, and it uses spatial correlation only, leading to a shift-invariant prior covariance matrix with block-diagonal structure.[10]

Test configuration. Figure 1 illustrates the spatial domain and the locations of sensors and QoIs for the test configuration. The spatial domain is given by $\Omega \coloneqq (0, 128)$ km × $(0, 128)$ km × $(0, 4)$ km, and it is discretized on a spatial grid with resolution of 0.25 km × 0.25 km × 0.125 km. Each grid point on the seafloor has one parameter degree of freedom, leading to a spatial parameter dimension of $N_m = 513 \times 513 = 263\,169$. The total number of sensors is $N_d = 49$; the sensors are placed uniformly on a 7×7 grid:

$$\begin{aligned} x_d &\in \{0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875\}(x_{\max} - x_{\min}),\\ y_d &\in \{0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875\}(y_{\max} - y_{\min}), \end{aligned} \tag{5.2}$$

where $x_{\min} = y_{\min} = z_{\min} = 0$, $x_{\max} = y_{\max} = 128$, $z_{\max} = 4$, and $(x_d, y_d, z_{\min})$ are the sensor coordinates on the seafloor. The QoI is the surface-gravity wave height sampled at $N_q = 16$ spatial locations with coordinates $(x_q, y_q, z_{\max})$ on the sea surface given by

$$\begin{aligned} x_q &\in \{0.5, 0.625, 0.75, 0.875\}(x_{\max} - x_{\min}),\\ y_q &\in \{0.5, 0.625, 0.75, 0.875\}(y_{\max} - y_{\min}). \end{aligned} \tag{5.3}$$

The time domain is given by $(0, T) \coloneqq (0, 50)$ s. Parameters and data are sampled at the same temporal frequency of 10 Hz, leading to $N_t^m = N_t^d =: N_t = 500$. The QoI is sampled at a lower temporal resolution of 1 Hz, so that $N_t^q = 50$. The total (spatiotemporal) dimensions are respectively $N_m N_t = 131\,584\,500$ parameters, $N_d N_t = 24\,500$ data, and $N_q N_t^q = 800$ QoIs.

Phase 1 (offline). In Phase 1, the adjoint PDEs of the acoustic-gravity wave equations (3.2) must be solved repeatedly to precompute the p2o map ($N_d = 49$ adjoint PDE solves) and the p2q map ($N_q = 16$ adjoint PDE solves). The discrete form of the equations is obtained by discretizing a weak variational formulation of the mixed acoustic-gravity wave problem with Galerkin finite elements in space and explicit 4th-order Runge–Kutta (RK4) time-stepping. The conforming finite element discretization employs third-order continuous (H^1-conforming) elements for the scalar-valued pressure unknown and second-order discontinuous (L^2-conforming) elements for the vector-valued velocity unknown (see [21] for details). For the test configuration, this discretization leads to a total state dimension of $N_w = 908\,627\,041$ and RK4 time-stepping with 5 000 time steps. The

[10] The shift invariance of the prior covariance $\mathbf{\Gamma}_{\text{prior}(m)}$ enables efficient matvecs with $\mathbf{G}^* \coloneqq \mathbf{\Gamma}_{\text{prior}(m)} \mathbf{F}^*$ and $\mathbf{G}_q^* \coloneqq \mathbf{\Gamma}_{\text{prior}(m)} \mathbf{F}_q^*$ by precomputing and storing $\mathbf{G}^*$ and $\mathbf{G}_q^*$ as block Toeplitz matrices. Without this structure of $\mathbf{\Gamma}_{\text{prior}(m)}$, the number of prior solves is significantly larger during Phases 2–3, making these offline computations more costly.

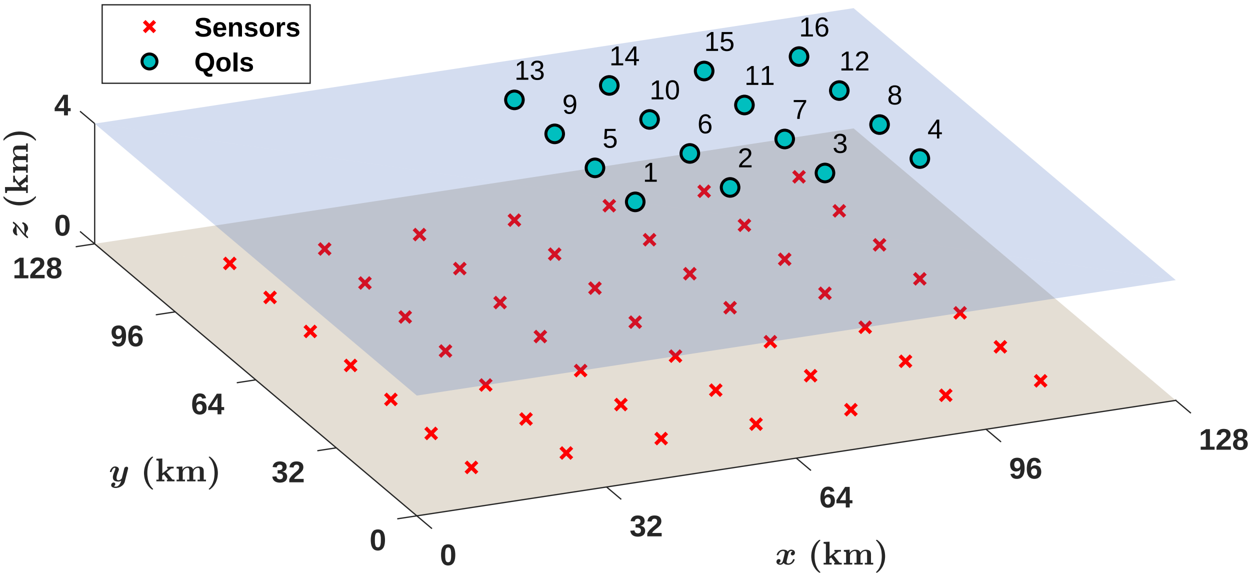

Figure 1: Test configuration: 3D spatial domain with the locations of 49 seafloor pressure sensors and 16 predicted QoIs (surface-gravity wave height).

explicit finite element solver for the acoustic-gravity wave equations is implemented in MFEM [6]. The implementation is optimized using fast integration via partial assembly with tensor-product hexahedral elements, and it is scalable supporting both CPU and GPU-based computation.

For this numerical example, the adjoint PDEs were solved on the *Stampede3* supercomputer at the Texas Advanced Computing Center (TACC) using 32 SPR compute nodes with a total of 3 584 Sapphire Rapids cores (112 cores per node).[11] Each one of the 49 + 16 adjoint PDE solves computes and stores one column of the adjoint p2o and p2q map, respectively, each of the size of the parameter dimension (∼1.05 GB), requiring total storage of 52 GB for the p2o map and 17 GB for the p2q map. The time per solve is approximately eight minutes, leading to a total Phase 1 compute time of ∼6.5 hours for the p2o map and ∼2.1 hours for the p2q map; however, all of these solves can be done simultaneously to accelerate the computation if needed.

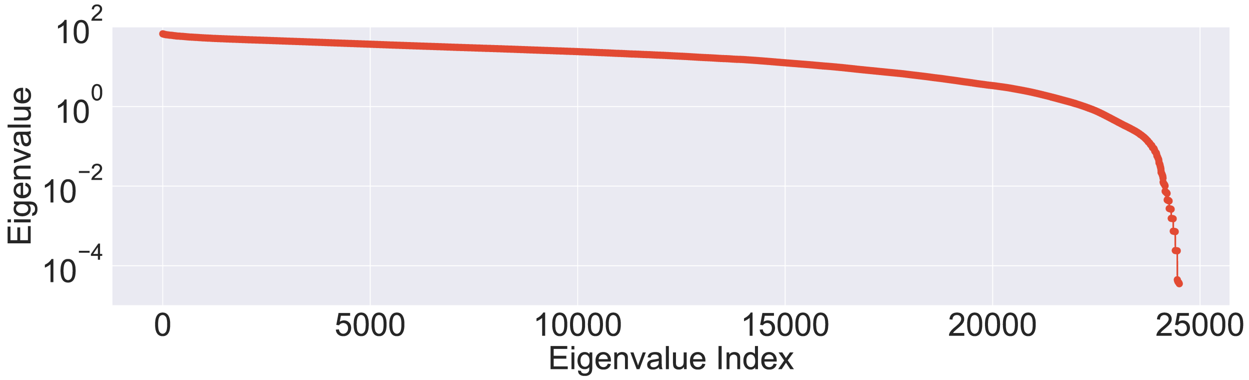

Figure 2: Eigenvalues of $\mathbf{FF}^*$ for the test configuration. The slow eigenvalue decay is indicative of hyperbolic PDE operators which do not readily admit compact representations.

Figure 2 depicts the eigenvalues of $\mathbf{FF}^*$ for the test configuration. The plot shows that the eigenvalues decay by only two orders of magnitude within the first 90 percent of the eigenvalues.

[11] See https://docs.tacc.utexas.edu/hpc/stampede3 for TACC's *Stampede3* system configuration.

This slow eigenvalue decay, indicative of hyperbolic PDE operators, exemplifies the difficulty of finding accurate compact representations of the operator in a lower-dimensional space via low-rank approximation or similar techniques (see our discussion on the Kolmogorov N-width problem in Section 1).

Phase 2 (offline). After completion of Phase 1, the finite element solver is no longer required. Instead of using forward and adjoint PDE solves, the actions of the p2o and p2q maps (and their respective adjoints) are now performed using the multi-GPU-accelerated FFT-based algorithm. On TACC's *Lonestar6* GPU cluster[12], with 24 Nvidia A100 GPUs (distributed across eight compute nodes), the runtime reduces to ~ 11 ms per $\mathbf{F}$ or $\mathbf{F}^*$ matvec and ~ 3 ms per $\mathbf{F}_q$ or $\mathbf{F}_q^*$ matvec. Compared to performing (adjoint) PDE solves with the explicit finite element solver, this corresponds respectively to a $\sim 44\,000\times$ and $\sim 160\,000\times$ speedup for the (adjoint) p2o and (adjoint) p2q matvecs. These incredibly fast actions of the p2o and p2q maps (and their respective adjoints) empower the remaining computations done offline (Phases 2–3) and the real-time inference and prediction performed online (Phase 4). All of the reported runtimes in Phases 2–4 are based on the same system configuration with 24 Nvidia A100 GPUs; in some cases, which will be pointed out, computations are done on a single A100 GPU.

Next up is precomputing $\mathbf{K} = \mathbf{\Gamma}_{\text{noise}} + \mathbf{F}\mathbf{G}^*$, where $\mathbf{G}^* = \mathbf{\Gamma}_{\text{prior}(m)}\mathbf{F}^*$. With our choice of prior, $\mathbf{G}^*$ is block Toeplitz, so we only need to perform N_d prior solves to precompute and store $\mathbf{G}^*$. Storage costs of $\mathbf{G}^*$ are equivalent to storing $\mathbf{F}$ ($\sim$52 GB). The time per prior solve, performed on a single A100 GPU with Nvidia cuDSS[13], is $\sim$84 ms. Thus, the total time needed to compute $\mathbf{G}^*$ is $N_d \cdot 84$ ms $\sim$4.2 s. The matrix $\mathbf{K}$ is of size $N_dN_t \times N_dN_t$, so precomputing it requires a total of $N_dN_t \sim 24\,500$ matvecs with $\mathbf{F}$ and $\mathbf{G}^*$. Because $\mathbf{G}^*$ has the same dimensions and block Toeplitz structure as $\mathbf{F}^*$, matvecs with $\mathbf{G}^*$ come at the same computational expense as matvecs with $\mathbf{F}$ or $\mathbf{F}^*$ ($\sim$0.011 s on 24 A100 GPUs). Therefore, the total time to compute and store $\mathbf{K}$ is $N_dN_t \cdot 0.022$ s $\sim$9 m.[14] $\mathbf{K}$ is a dense matrix without any obvious structure, so storing it requires storing all of its entries ($\sim$ 4.8 GB). Once $\mathbf{K}$ has been computed, it can be Cholesky-factorized ($\mathbf{K} = \mathbf{L}\mathbf{L}^*$) which takes $\sim$ 1.18 s on a single A100 GPU; the Cholesky factor $\mathbf{L}$ may overwrite $\mathbf{K}$ and can be stored at half its memory cost ($\sim$2.4 GB).[15]

After $\mathbf{K}$ has been factorized, parameter uncertainties can be sampled from the posterior covariance $\mathbf{\Gamma}_{\text{post}(m)} = \left(\mathbf{I} - \mathbf{G}^*\mathbf{K}^{-1}\mathbf{F}\right)\mathbf{\Gamma}_{\text{prior}(m)}$. Each action of $\mathbf{\Gamma}_{\text{post}(m)}$ requires a matvec with $\mathbf{G}^*$ and $\mathbf{F}$ (11 ms each), a triangular solve of the prefactorized $\mathbf{K}$ matrix (20 ms), and a prior solve (84 ms), for a total time of $\sim$126 ms. This enables us to rapidly sample parameter covariances. However, computing the full covariance matrix is still too costly, both in memory and computation time. Each of the $N_mN_t = 131\,584\,500$ columns has a storage cost of $\sim$1.05 GB thus totaling $\sim$138 PB for the entire $\mathbf{\Gamma}_{\text{post}(m)}$, and computing this many matvecs would take $N_mN_t \cdot 126$ ms $\sim$192 days. The storage issue could be overcome by limiting ourselves to computing pointwise variances from the diagonal of $\mathbf{\Gamma}_{\text{post}(m)}$, and the matvecs could be performed simultaneously (with each matvec performed on 24 A100 GPUs) to reduce the computation time to less than two days on a large-scale machine with 2400 A100 GPUs. So computing parameter uncertainties may be indeed feasible for a problem of the size of this numerical example. Unfortunately, the parameter space dimension for the CSZ digital twin is at least one order of magnitude larger ($\mathcal{O}(10^9)$), making a full computation of the diagonal of $\mathbf{\Gamma}_{\text{post}(m)}$ computationally intractable.

[12] See https://docs.tacc.utexas.edu/hpc/lonestar6 for TACC's *Lonestar6* system configuration.

[13] https://developer.nvidia.com/cudss

[14] Note that $\mathbf{\Gamma}_{\text{noise}}$ is assumed to be diagonal, so its computational expense in computing $\mathbf{K}$ is negligible.

[15] In practice, many solvers store the Cholesky factor as a lower (or upper) triangular part of the full matrix, in which case the memory cost for storing $\mathbf{L}$ equals storing $\mathbf{K}$.

Phase 3 (offline). As a preliminary step to computing posterior covariance of the QoIs, we precompute $\mathbf{G}_q^* = \mathbf{\Gamma}_{\text{prior}(m)}\mathbf{F}_q^*$. Because $\mathbf{G}_q^*$ is block Toeplitz, we only need to perform N_q prior solves to compute and store $\mathbf{G}_q^*$. Storage costs of $\mathbf{G}_q^*$ are equivalent to storing $\mathbf{F}_q$ ($\sim$17 GB). The time per prior solve, performed on a single A100 GPU with Nvidia cuDSS, is $\sim$0.084 s. Therefore, the total time needed to compute $\mathbf{G}_q^*$ is $N_q \cdot 0.084$ s $\sim$1.4 s.

To precompute the posterior covariance of the QoIs, $\mathbf{\Gamma}_{\text{post}(q)} = \mathbf{F}_q\left(\mathbf{I} - \mathbf{G}^*\mathbf{K}^{-1}\mathbf{F}\right)\mathbf{G}_q^*$, its action needs to be evaluated $N_q N_t^q = 800$ times. Each action of $\mathbf{\Gamma}_{\text{post}(q)}$ requires one matvec each with $\mathbf{F}_q$ (3 ms), $\mathbf{G}^*$ (11 ms), $\mathbf{F}$ (11 ms), and $\mathbf{G}_q^*$ (3 ms), as well as a triangular solve of the prefactorized $\mathbf{K}$ matrix (20 ms). So each $\mathbf{\Gamma}_{\text{post}(q)}$ matvec takes $\sim$48 ms, leading to a total time of $N_q N_t \cdot 48$ ms $\sim$38.4 s to compute the full QoI covariance matrix. For this QoI dimension of 800, the storage cost for $\mathbf{\Gamma}_{\text{post}(q)}$ is negligible ($\sim$5.2 MB).

Precomputing the d2q map $\mathbf{Q} = \mathbf{F}_q\left(\mathbf{I} - \mathbf{G}^*\mathbf{K}^{-1}\mathbf{F}\right)\mathbf{G}^*\mathbf{\Gamma}_{\text{noise}}^{-1}$ requires $N_q N_t^q = 800$ matvecs with $\mathbf{Q}$. Each matvec takes 56 ms, so the total time to compute $\mathbf{Q}$ is 44.8 s, and the cost for storing the matrix is 157 MB.

Phase 4 (online). To test the inference and prediction framework, we use synthetic data generated from the PDE model. For the numerical example, we use a source function that describes a spatiotemporal seafloor deformation from three mixed Gaussians:

$$\frac{\partial b(x,y,t)}{\partial t} = \sum_{i=1}^{3} G^{(i)}(x,y,t), \tag{5.4}$$

where

$$G^{(i)}(x,y,t) = \begin{cases} A^{(i)} \exp\left(-\left(\dfrac{x - X_\mathrm{c}^{(i)}}{X_\mathrm{r}^{(i)}}\right)^2 - \left(\dfrac{y - Y_\mathrm{c}^{(i)}}{Y_\mathrm{r}^{(i)}}\right)^2\right) \dfrac{\pi}{2T_\mathrm{r}^{(i)}} \sin\left(\dfrac{\pi t}{T_\mathrm{r}^{(i)}}\right) & , \quad t \le T_\mathrm{r}^{(i)}, \\ 0 & , \quad t > T_\mathrm{r}^{(i)}. \end{cases} \tag{5.5}$$

The parameters for these Gaussians are given in Table 1. While each of the Gaussians varies smoothly in both space and time, the second and third ($i = 2, 3$) introduce higher-frequency components (both spatially and temporally) into the solution than the seafloor motion due to the first Gaussian ($i = 1$).

Table 1: Parameters of the Gaussians defining synthetic seafloor deformation for the numerical example.

Symbol	Description	Value			Unit
		$i=1$	$i=2$	$i=3$	
$A^{(i)}$	Amplitude	4	1	-0.5	m
$T_\mathrm{r}^{(i)}$	Rise time	20	10	10	s
$X_\mathrm{r}^{(i)}$	Rise width in x	16	4	4	km
$Y_\mathrm{r}^{(i)}$	Rise width in y	32	4	8	km
$X_\mathrm{c}^{(i)}$	Center in x	64	64	70	km
$Y_\mathrm{c}^{(i)}$	Center in y	64	88	56	km

Noisy data are generated from the true data (obtained from the PDE model) by adding Gaussian noise. In particular, for each sensor, the magnitude of the added Gaussian noise is taken as a certain percentage (from now on referred to as "noise level") of the infinity norm (over all time steps) of the sensor's true data. Given these noisy observations, the total online compute time to solve the inference problem for 131 584 500 parameters (via (4.13)) is ∼ 53 ms on 24 A100 GPUs.[16] Solving the prediction problem for 800 QoIs can then be done either through push-forward of the inferred parameters via the p2q map (∼ 3 ms on 24 A100 GPUs) or, more efficiently, directly from the noisy observations (via (4.14)) using the precomputed d2q map (∼ 1 ms on a laptop). In other words, the full end-to-end digital twin, from observations via inference to prediction, with the full-order model and including uncertainties, is realized within fractions of a second for a test configuration of nearly 132 M parameters.

Table 2: Compute times and storage requirements for offline computation (Phases 1–3) and online computation (Phase 4) of the Bayesian inference and prediction framework for a numerical example with 132 M parameters.

Phase	Task	Time	Memory	System
1	Precompute p2o map $\mathbf{F}$ (49 adjoint PDE solves)	6.5 h	52 GB	3584 Sapphire Rapids Cores
	Precompute p2q map $\mathbf{F}_q$ (16 adjoint PDE solves)	2.1 h	17 GB	
2	Form $\mathbf{G}^*$ (49 prior solves)	4.2 s	52 GB	24 Nvidia A100 (40 GB) GPUs
	Form $\mathbf{G}_q^*$ (16 prior solves)	1.4 s	17 GB	
	Compute and factorize $\mathbf{K}$	10 m	4.8 GB	
3	Compute QoI cov. $\mathbf{\Gamma}_{\text{post}(q)}$	1 m	5.2 MB	24 Nvidia A100 (40 GB) GPUs
	Precompute d2q map $\mathbf{Q}$	1 m	0.16 GB	
4	Inference of $\mathbf{m}_{\text{map}}$	57 ms	1.1 GB	24 Nvidia A100 (40 GB) GPUs
	Prediction of $\mathbf{q}_{\text{map}}$	1 ms	0.1 MB	Laptop

Table 2 summarizes the computational costs for each phase of the framework, both in storage requirements and compute times (obtained on systems as specified in the table), for the test configuration. While the spatial domain of this numerical example was about 20× smaller than the dimensions of the full CSZ digital twin, the computational resources can be scaled up accordingly. Importantly, all of the large-scale computations done offline—particularly the PDE solves in Phase 1—scale effectively to larger compute clusters.

Table 3: Relative errors of the inferred parameters $\mathbf{m}_{\text{map}}$, the predicted QoIs $\mathbf{q}_{\text{map}}$, and the pressures at sensor locations reconstructed from $\mathbf{m}_{\text{map}}$ via the p2o map $\mathbf{F}$, obtained via the real-time inference and prediction framework from synthetic observations with varying levels of relative additive noise.

Noise level	Inferred parameters	Predicted QoIs	Pressure reconstruction at sensors
2%	0.0776	0.0108	0.0195
4%	0.0836	0.0167	0.0397
6%	0.0948	0.0161	0.0596

For the particular source function given in (5.4)–(5.5) and three examples of synthetic observations with varying additive noise levels (2%, 4%, 6%), Table 3 shows the relative errors observed for

[16] Neglecting cost for I/O, the total compute time is composed of ∼ 11 ms per $\mathbf{G}^*$ and $\mathbf{F}$, and ∼ 20 ms for solving prefactorized $\mathbf{K}$.

the mean of the inferred parameters, the mean of the predicted QoIs, and the pressures at sensor locations reconstructed from the mean of the inferred parameter field via the p2o map. The relative error of the reconstructed pressures, $\|\mathbf{Fm}_{\text{map}} - \mathbf{Fm}_{\text{true}}\|/\|\mathbf{Fm}_{\text{true}}\|$, corresponds roughly to the noise level; this is expected because the highly underdetermined inverse problem ($N_m \gg N_d$) closely fits the sparse and noisy data, even with the smoothing prior. The relative errors of the inferred mean of the parameter field, $\|\mathbf{m}_{\text{map}} - \mathbf{m}_{\text{true}}\|/\|\mathbf{m}_{\text{true}}\|$ ($\sim$ 7–10%), are larger than those of the predicted mean of the QoIs, $\|\mathbf{q}_{\text{map}} - \mathbf{q}_{\text{true}}\|/\|\mathbf{q}_{\text{true}}\|$ ($\sim$ 1–2%). Since the QoIs are predicted via the inference of the parameters, this results suggests that some components of the error of the inferred parameter field are not forward propagated to the QoIs via the p2q map.

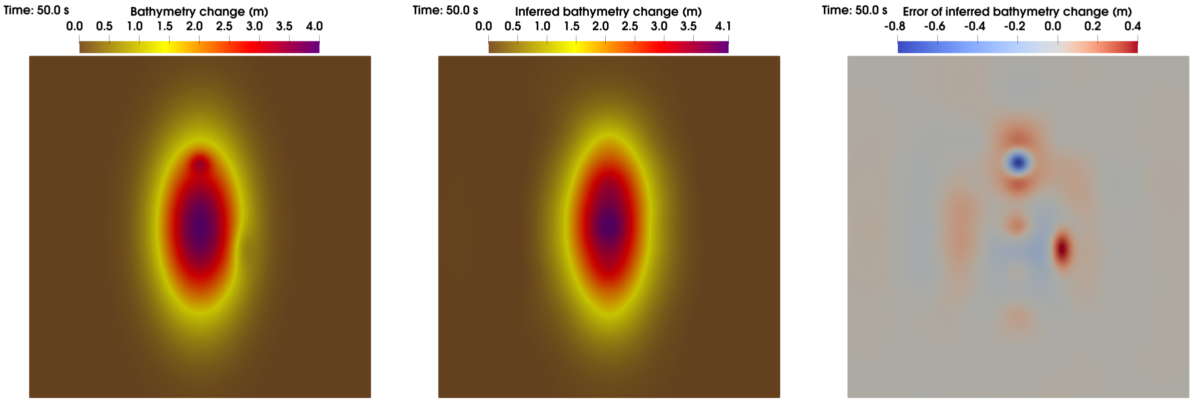

Figure 3: Total seafloor displacement (time-integrated parameter field). Left: True field; Middle: Inferred mean from synthetic observations with 6% relative additive noise; Right: Error between true and inferred fields.

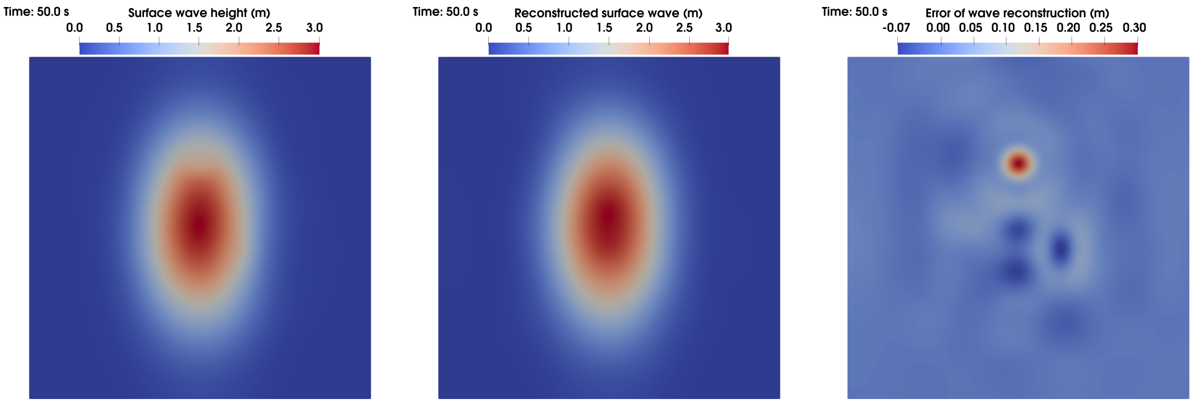

Figure 4: Surface gravity wave. Left: True field; Middle: Reconstructed field based on inferred mean of seafloor motion (from synthetic observations with 6% relative additive noise); Right: Error between true and reconstructed surface gravity wave based on inferred mean of seafloor motion.

Figure 3 illustrates that the components of the parameter field that are not inferred correctly predominantly correspond to the higher-frequency spatial components. These rough components cannot be reconstructed because they are not well informed by the sparse data.[17] On the other hand, the lower-frequency modes of the seafloor motion are inferred with good accuracy. A similar observation can be made for the surface gravity wave reconstructed based on the inferred mean of the seafloor motion, which is depicted in Figure 4. However, the rough components of the relative

[17] This is a fundamental limitation of an ill-posed inverse problem [16].

error for the surface gravity wave are smaller than those for the parameter field, suggesting that the corresponding higher-frequency modes are dampened by the p2q map. In other words, the tsunami formation appears to be driven mostly by the lower-frequency components of the seafloor motion, making relatively accurate QoI predictions possible even when the inferred parameters have larger relative errors (cf. Table 3) and those errors are dominated by the rough components. The inferred mean of some of the predicted QoIs (cf. Figure 1), as well as their respective 95% confidence intervals, are shown in Figure 5.

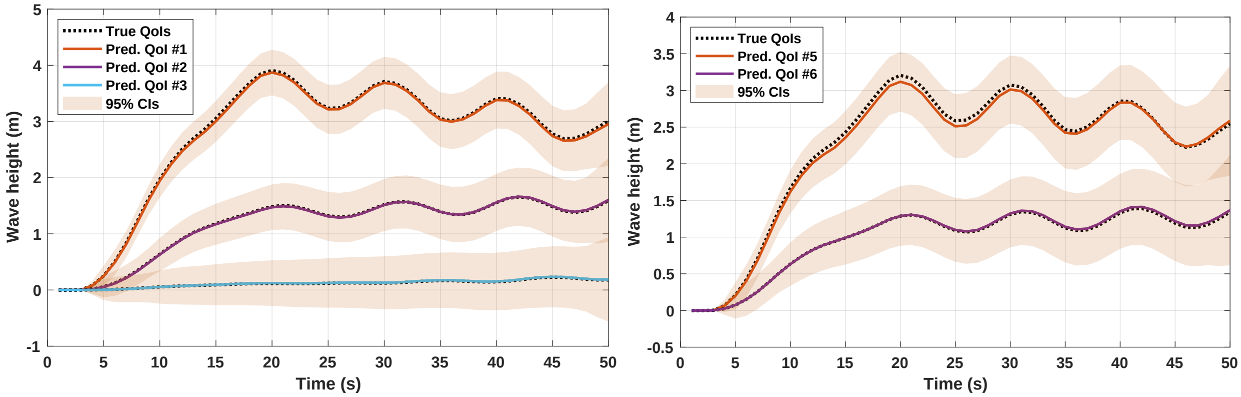

Figure 5: True and predicted QoIs (tsunami wave heights at specified locations) with uncertainties depicted in the form of 95% confidence intervals. Left: QoI locations #1–3; Right: QoI locations #5–6 (cf. Figure 1).

6 Conclusions and ongoing work

By exploiting the autonomous structure inherent to the governing forward PDE model, we have developed a digital twin framework for real-time Bayesian inversion at extreme scale while remaining faithful to high-fidelity, physics-based models. As a core component of this framework, we employ an FFT-based algorithm to compute Hessian matvecs at speeds over four orders of magnitude faster than those of conventional methods and circumvent the need for PDE solutions at the time of inference. In addition to making real-time Hessian matvecs possible, this algorithm can also be used to accelerate offline portions of the computation, which includes precomputing an efficient representation of the inverse operator in the data space. The digital twin framework is extended to the goal-oriented setting to enable real-time predictions of QoIs under uncertainty.

As a target application, we utilized this framework for a proof-of-concept tsunami early warning system that infers the earthquake-induced seafloor motion from near-field observations of the acoustic pressure on the seafloor and an acoustic-gravity PDE model, followed by a forward prediction of the tsunami wave heights at specified locations. The entire end-to-end inference, prediction, and uncertainty quantification for a system with over one hundred million parameters was computed in fractions of a second. For this tsunami inversion, the results indicate that sparse observations of the pressure transients in the near field are sufficient to accurately infer the smooth components of the seafloor motion and that these smooth components appear to account well for the surface gravity wave formation.

In our ongoing work, we leverage this framework to create a digital twin for tsunami forecasting from potential rupture events in the CSZ. Implementation of the CSZ digital twin requires offline computations of extreme-scale adjoint wave propagation PDE solutions and online inference of more than one billion parameters (i.e. at least one order of magnitude larger than the test model problem solved in this paper). Beyond tsunami early warning, the core components of our digital twin framework are applicable to many other classes of dynamical system inverse problems, including nuclear testing treaty verification, contaminant transport, and atmospheric CO2 monitoring. Extending the framework to create digital twins for these problems, and others, is ongoing work.

Acknowledgements. This research was supported in part by DOE ASCR grant DE-SC0023171 and DOD MURI grants FA9550-21-1-0084 and FA9550-24-1-0327. Supercomputing resources were provided by the Texas Advanced Computing Center (TACC) at UT Austin on its *Frontera*, *Lonestar6*, *Stampede3*, and *Vista* systems, as well as by NERSC award ALCC-ERCAP0030671 at NERSC on its *Perlmutter* system. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 2137420.

References

[1] L. S. Abrahams, L. Krenz, E. M. Dunham, A.-A. Gabriel, and T. Saito. "Comparison of methods for coupled earthquake and tsunami modelling". In: *Geophysical Journal International* 234.1 (2023), pp. 404–426. DOI: 10.1093/gji/ggad053.

[2] N. Alger, T. Hartland, N. Petra, and O. Ghattas. "Point Spread Function Approximation of High-Rank Hessians with Locally Supported Nonnegative Integral Kernels". In: *SIAM Journal on Scientific Computing* 46.3 (2024), A1658–A1689.

[3] N. Alger, V. Rao, A. Myers, T. Bui-Thanh, and O. Ghattas. "Scalable matrix-free adaptive product-convolution approximation for locally translation-invariant operators". In: *SIAM Journal on Scientific Computing* 41.4 (2019), A2296–A2328.

[4] N. Alger, U. Villa, T. Bui-Thanh, and O. Ghattas. "A data scalable augmented Lagrangian KKT preconditioner for large-scale inverse problems". In: *SIAM Journal on Scientific Computing* 39.5 (2017), A2365–A2393.

[5] I. Ambartsumyan, W. Boukaram, T. Bui-Thanh, O. Ghattas, D. Keyes, G. Stadler, G. Turkiyyah, and S. Zampini. "Hierarchical Matrix Approximations of Hessians Arising in Inverse Problems Governed by PDEs". In: *SIAM Journal on Scientific Computing* 42.5 (2020), A3397–A3426.

[6] R. Anderson, J. Andrej, A. Barker, J. Bramwell, J.-S. Camier, J. Cerveny, V. Dobrev, Y. Dudouit, A. Fisher, T. Kolev, W. Pazner, M. Stowell, V. Tomov, I. Akkerman, J. Dahm, D. Medina, and S. Zampini. "MFEM: A modular finite element methods library". In: *Computers & Mathematics with Applications* 81 (2021), pp. 42–74.

[7] C. Becerril, A. Sladen, J.-P. Ampuero, P. J. Vidal-Moreno, M. Gonzalez-Herraez, F. Kutschera, A.-A. Gabriel, and F. Bouchette. "Towards tsunami early-warning with distributed acoustic sensing: expected seafloor strains induced by tsunamis". In: *Authorea Preprints* (2024). DOI: 10.22541/essoar.171052484.46251426/v1.

[8] P. Benner, S. Gugercin, and K. Willcox. "A survey of projection-based model reduction methods for parametric dynamical systems". In: *SIAM review* 57.4 (2015), pp. 483–531. DOI: 10.1137/130932715.

[9] T. Bui-Thanh and O. Ghattas. "An analysis of infinite dimensional Bayesian inverse shape acoustic scattering and its numerical approximation". In: *SIAM/ASA Journal of Uncertainty Quantification* 2.1 (2014), pp. 203–222. DOI: 10.1137/120894877.

[10] T. Bui-Thanh and O. Ghattas. “Analysis of the Hessian for Inverse Scattering Problems. Part I: inverse shape scattering of acoustic waves”. In: *Inverse Problems* 28.5 (2012), p. 055001. DOI: 10.1088/0266-5611/28/5/055001.

[11] T. Bui-Thanh and O. Ghattas. “Analysis of the Hessian for Inverse Scattering Problems. Part II: inverse medium scattering of acoustic waves”. In: *Inverse Problems* 28.5 (2012), p. 055002. DOI: 10.1088/0266-5611/28/5/055002.

[12] J. Y. Cherniawsky, V. V. Titov, K. Wang, and J.-Y. Li. “Numerical simulations of tsunami waves and currents for southern Vancouver Island from a Cascadia megathrust earthquake”. In: *Pure and Applied Geophysics* 164 (2007), pp. 465–492.

[13] Y. Daon and G. Stadler. “Mitigating the influence of the boundary on PDE-based covariance operators”. In: *Inverse Problems and Imaging* 12.5 (2018), pp. 1083–1102.

[14] L. Demanet, P.-D. Létourneau, N. Boumal, H. Calandra, J. Chiu, and S. Snelson. “Matrix probing: A randomized preconditioner for the wave-equation Hessian”. In: *Applied and Computational Harmonic Analysis* 32.2 (2012), pp. 155–168.

[15] H. P. Flath, L. C. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, and O. Ghattas. “Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations”. In: *SIAM Journal on Scientific Computing* 33.1 (2011), pp. 407–432. DOI: 10.1137/090780717.

[16] O. Ghattas and K. Willcox. “Learning physics-based models from data: perspectives from inverse problems and model reduction”. In: *Acta Numerica* 30 (2021), pp. 445–554.

[17] D. Giles, D. Gopinathan, S. Guillas, and F. Dias. “Faster than real time tsunami warning with associated hazard uncertainties”. In: *Frontiers in Earth Science* 8 (2021), p. 597865. DOI: 10.3389/feart.2020.597865.

[18] R. M. Gray et al. “Toeplitz and circulant matrices: A review”. In: *Foundations and Trends in Communications and Information Theory* 2.3 (2006), pp. 155–239.

[19] C. Greif and K. Urban. “Decay of the Kolmogorov N-width for wave problems”. In: *Applied Mathematics Letters* 96 (2019), pp. 216–222.

[20] M. D. Gunzburger. *Perspectives in flow control and optimization*. SIAM, 2002.

[21] S. Henneking and O. Ghattas. *A tsunami inverse problem for seafloor motion from near-field observations and an acoustic-gravity wave propagation model.* In Preparation. 2025.

[22] M. A. Hesse and G. Stadler. “Joint inversion in coupled quasi-static poroelasticity”. In: *Journal of Geophysical Research: Solid Earth* 119.2 (2014), pp. 1425–1445. DOI: 10.1002/2013JB010272.

[23] T. Isaac, N. Petra, G. Stadler, and O. Ghattas. “Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the Antarctic ice sheet”. In: *Journal of Computational Physics* 296 (2015), pp. 348–368. DOI: 10.1016/j.jcp.2015.04.047.

[24] A. G. Kalmikov and P. Heimbach. “A Hessian-based method for uncertainty quantification in global ocean state estimation”. In: *SIAM Journal on Scientific Computing* 36.5 (2014), S267–S295. DOI: 10.1137/130925311.

[25] M. C. Kennedy and A. O’Hagan. “Bayesian calibration of computer models”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 63.3 (2001), pp. 425–464. DOI: 10.1111/1467-9868.00294.

[26] L. Krenz, C. Uphoff, T. Ulrich, A.-A. Gabriel, L. S. Abrahams, E. M. Dunham, and M. Bader. “3D acoustic-elastic coupling with gravity: the dynamics of the 2018 Palu, Sulawesi earthquake and tsunami”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2021, pp. 1–14.

[27] L. J. Leonard, C. A. Currie, S. Mazzotti, and R. D. Hyndman. “Rupture area and displacement of past Cascadia great earthquakes from coastal coseismic subsidence”. In: *GSA Bulletin* 122.11-12 (2010), pp. 2079–2096.

[28] R. J. LeVeque, P. Bodin, G. Cram, B. W. Crowell, F. I. González, M. Harrington, D. Manalang, D. Melgar, D. A. Schmidt, J. E. Vidale, et al. “Developing a warning system for inbound tsunamis from the Cascadia subduction zone”. In: *OCEANS 2018 MTS/IEEE Charleston*. IEEE. 2018, pp. 1–10.

[29] W. Li, H. Yeh, K. Hirata, and T. Baba. “Ocean-bottom pressure variations during the 2003 Tokachi-Oki earthquake”. In: *Nonlinear wave dynamics*. World Scientific. 2009, pp. 109–126.

[30] Z. Li, N. B. Kovachki, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. “Fourier neural operator for parametric partial differential equations”. In: *International Conference on Learning Representations*. 2021.

[31] F. Lindgren, H. Rue, and J. Lindström. “An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 73.4 (2011), pp. 423–498.

[32] G. C. Lotto and E. M. Dunham. “High-order finite difference modeling of tsunami generation in a compressible ocean from offshore earthquakes”. In: *Computational Geosciences* 19.2 (2015), pp. 327–340.

[33] D. Melgar, R. J. LeVeque, D. S. Dreger, and R. M. Allen. “Kinematic rupture scenarios and synthetic displacement data: An example application to the Cascadia subduction zone”. In: *Journal of Geophysical Research: Solid Earth* 121.9 (2016), pp. 6658–6674.

[34] R. Nammour. “Approximate multi-parameter inverse scattering using pseudodifferential scaling”. PhD thesis. Rice University, 2011.

[35] T. O’Leary-Roseberry, P. Chen, U. Villa, and O. Ghattas. “Derivative-informed neural operator: an efficient framework for high-dimensional parametric derivative learning”. In: *Journal of Computational Physics* 496 (2024), p. 112555. DOI: 10.1016/j.jcp.2023.112555.

[36] K. B. Olsen, W. J. Stephenson, and A. Geisselmeyer. “3D crustal structure and long-period ground motions from a M9.0 megathrust earthquake in the Pacific Northwest region”. In: *Journal of Seismology* 12 (2008), pp. 145–159.

[37] D. Schmidt, W. Wilcock, R. LeVeque, F. Gonzalez, G. Cram, D. Manalang, M. Harrington, E. Roland, and P. Bodin. “Earthquake and tsunami early warning on the Cascadia subduction zone: a feasibility study for an offshore geophysical monitoring network”. In: (2023). URL: https://hdl.handle.net/1773/50968.

[38] J. Sherman and W. J. Morrison. “Adjustment of an inverse matrix corresponding to a change in one element of a given matrix”. In: *The Annals of Mathematical Statistics* 21.1 (1950), pp. 124–127.

[39] A. M. Stuart. “Inverse problems: a Bayesian perspective”. In: *Acta Numerica* 19 (2010), pp. 451–559.

[40] S. Subramanian, K. Scheufele, M. Mehl, and G. Biros. “Where did the tumor start? An inverse solver with sparse localization for tumor growth models”. In: *Inverse problems* 36.4 (2020), p. 045006. DOI: 10.1088/1361-6420/ab649c.

[41] C. Uphoff, S. Rettenberger, M. Bader, E. H. Madden, T. Ulrich, S. Wollherr, and A.-A. Gabriel. “Extreme scale multi-physics simulations of the tsunamigenic 2004 Sumatra megathrust earthquake”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis.* 2017, pp. 1–16.

[42] S. Venkat, M. Fernando, S. Henneking, and O. Ghattas. “Fast and scalable FFT-based GPU-accelerated algorithms for Hessian actions arising in linear inverse problems governed by autonomous dynamical systems”. In: *arXiv preprint arXiv:2407.13066* (2024).

[43] U. Villa, N. Petra, and O. Ghattas. “hIPPYlib: an extensible software framework for large-scale inverse problems governed by PDEs: part I: deterministic inversion and linearized Bayesian inference”. In: *ACM Transactions on Mathematical Software* 47.2 (2021), pp. 1–34.

[44] S. M. Wild, R. G. Regis, and C. A. Shoemaker. “ORBIT: Optimization by radial basis function interpolation in trust-regions”. In: *SIAM Journal on Scientific Computing* 30.6 (2008), pp. 3197–3219. DOI: 10.1137/070691814.

[45] M. A. Woodbury. *Inverting modified matrices.* Department of Statistics, Princeton University, 1950.

[46] J. Worthen, G. Stadler, N. Petra, M. Gurnis, and O. Ghattas. “Towards adjoint-based inversion for rheological parameters in nonlinear viscous mantle flow”. In: *Physics of the Earth and Planetary Interiors* 234 (2014), pp. 23–34. DOI: 10.1016/j.pepi.2014.06.006.

[47] T. Yamamoto. “Gravity waves and acoustic waves generated by submarine earthquakes”. In: *International Journal of Soil Dynamics and Earthquake Engineering* 1.2 (1982), pp. 75–82.