Technical Report TR-2008-08

Efficient Sampling of Dynamical Systems with Spatial Uncertainty

Kyle Schmitt[*], Mihai Anitescu[†], and Dan Negrut[*]

[*] - Simulation Based Engineering Laboratory, University of Wisconsin, Madison, WI

[†] - Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL

October 9, 2008

Abstract

We present two methods for efficiently sampling the response (trajectory space) of dynamical systems operating under spatial uncertainty assumed to be representable with Gaussian processes. The dynamics of such systems depends on spatially indexed uncertain parameters that span infinite dimensional spaces. This places a heavy computational burden on the implementation of existing methodologies, a challenge addressed with two new conditional sampling approaches. When a single instance of the uncertainty is needed in the entire domain, we use a fast Fourier transform technique. When the Gaussian process has a compactly supported kernel, we use an incremental sampling approach, which not only is fast but also has a very small memory footprint. We prove that both methods produce the same distributions as the widely used Cholesky-based approaches while having much less complexity. We illustrate this convergence at a far smaller computational effort and memory cost for a simple vehicle model.

Keywords: Uncertainty quantification, Gaussian process, dynamical system, fast Fourier transform, conditional sampling

Contents

1 INTRODUCTION

The importance and impact of uncertainty in a broad spectrum of science and engineering problems are widely recognized. Uncertainty quantification has been studied for decades, with methods and techniques improving in accuracy and robustness. Painstaking effort has been exerted in the modern computer era to adapt uncertainty quantification methods, leveraging the ever-growing storage and computing power available to the scientific community. Uncertainty quantification processes are now being demanded to evaluate at new levels of completeness and fineness. In order to these demands, a premium is placed on the efficiency of the methods used. This paper addresses the efficiency of uncertainty quantification for dynamical systems operating under spatial uncertainty.

A good overview of uncertainty quantification methods commonly in use can be found in [3]. Probably the most widespread technique is represented by random sampling methods, such as Monte Carlo and Latin hypercube sampling methods [8,9]. Reliability methods can be computationally less intensive than sampling methods [6]. Their increased efficiency is the result of the fact that the entire surface response is updated by any given sample information. This update can be carried out by mean value [6], global Gaussian process [11,12], and other approaches [20]. Polynomial chaos expansion (PCE) methods construct a polynomial surface response approximation [4,5]. They use either a nonlinear projection Galerkin method [5] or a collocation approach [22] to obtain the polynomial approximation. PCE methods have generally been impaired, however, by a relatively low dimension of the uncertainty space that they can span [22]. The other methods are in principle applicable independent of dimension. Nonetheless, their efficiency tends to degrade with the increasing dimension of uncertainty spaces if one uses the implementations presented in the above references.

Spatial uncertainty quantification – learning, inference, and sampling in multidimensional space – is being held to ever-escalating expectations. One of the primary reasons is the unbounded dimensional nature of the problem; problem spaces can always be larger and desired results can always be finer. In this context, the research progress has reached a fork in the methodological road: the continuation of traditional methods or the imposition of conditions and constraints on traditional methods to elicit an exploitable problem structure or pattern. Our methodologies are of the later contingent.

Kriging and its variations, co-kriging and kriging with regression, are example of the developing nature of spatial uncertainty [10, 13, 18]. These "hybrid methods," as they are called, were contrived to address accuracy and robustness concerns, capable of extending to less-structured data (i.e. nonstationary, nonhomogeneous). Still, the methods require the solution of small kriging systems at each sample point or the solution of a large system, using global dual kriging simulation. These methods are suitable for data interpolation of small sample sets from large data sets but cannot be overextended without encountering significant computational effort. Gaussian processes, as presented in [16] and applied in [17], are an especially accurate and usable methodology for interpolating spatial data. Unfortunately, like kriging methods, the Gaussian processes framework traditionally suffers from numerical instability and excessive storage requirements because of the matrix operations it employs.

Our proposed methodologies address these problems. The first presents an analogous approach for Gaussian processes in a matrix-free fashion in the frequency domain. The second presents an incremental approach to data interpolation where interpolation is done only in areas of interest.

Some traditional methods for spatial uncertainty calculations, such as white-noise methods and spectral methods, fare much better than kriging with respect to runtime yet suffer from accuracy and robustness issues. Examples of the first type of method, based on homogeneous random processes, are explained in [7, 15]. While these methods model a large class of problems and may be useful in design and simulation, they are nonetheless not appropriate for situations where the spatial variation has large areas of coherence that are inhomogeneous, a common occurrence in real-life applications. Another original approach is discussed in [19], where the use of traditional spectral methods is supplemented with covariance spectrum phase values conditioned from data. However, the methodology is plagued by accuracy limitations typically associated with spectral methods. For one, accuracy of these methods is dependent on the number of harmonics used and can quickly become hard to implement and manage. Also, spectral techniques cannot easily accommodate rapid variations in the properties of the surface, which is a well-known side effect of the Gibbs phenomenon, thus presenting robustness concerns.

Growing problem complexity and solution resolution have led to creative approaches for conditioned problems, that is, solution methods that gain efficiency in lieu of robustness, and sometimes accuracy. The methods still provide informative results that can be used in industry applications. One common approach is regimenting the data space to a lattice and sampling from the prior; that is the samples are not conditioned on the local deviation from the prior model. The benefit of this simplification is that the covariance matrix is Toeplitz and can be sampled from by leveraging fast Fourier transforms (FFT) [2]. Another perspective gaining wider acceptance is the sparse-grid assumption introduced by Stein in [18]. Stein leverages compact support kernels to limit the training data set size required for statistically accurate and consistent interpolation. This is especially useful with high-frequency data sets, where correlation decays to nearly zero quickly. Stein, however, does not couple sparse grid theory with conditional sampling to allow for truly dynamic interpolation (as we present as our second method).

Our first method, herein referred to as periodic fold sampling (PFS), imposes one condition and one assumption to conduct the entire Gaussian processes in a matrix-free fashion. The condition is that the data is provided on a lattice. The assumption is that the space is periodic, and this is reflected in the autocorrelation function; this assumption is tried and introduces few problems far from boundaries when working on large grid spaces.

Our second method, herein denoted the incremental compact kernel approach (ICKA), imposes a sparse-grid assumption to localize the data interpolation. That is, data need not be interpolated all at once on the entire problem space but rather can be interpolated incrementally only in the proximity of the region of importance. This framework is different from other approaches in that its effort is invariant to problem size.

Both methods are tested in conjunction with the simulation of nonlinear vehicle dynamics.

The spatially uncertain property interpolated with Gaussian processes in this paper is the friction coefficient; imagine, for instance, randomly distributed patches of ice on a road. [1] Both methods prove to be several orders faster than traditional Gaussian process sampling approaches while achieving similar results. Furthermore, the incremental sampling method shows a linear increase of runtime with simulation time, suggesting that interpolation runtime is negligible compared to simulation runtime.

For PFS, we present the specifics of each matrix-free Gaussian processes step conducted in the frequency domain: data organization and data augmentation, estimation of hyper-parameters, computation of posterior distribution characteristics, and multivariate Gaussian sampling. We explain how FFTs are used to map between the time and frequency domains. We note that FFT approaches have been used to sample from prior distributions [1] but not, as far as we know, from the posterior distribution. We show in conjunction with PFS convergence of the resulting sample mean and sample covariance to the respective values computed with traditional Gaussian processes, as in [17]. For ICKA, we describe the compact support kernel in detail, including a discussion of the implementation process, formal proofs, and numerical results. For the latter, there is no descriptive way to compare the resulting data set (on a path) with one computed with traditional methods (on a grid). Instead, we show convergence of the average dynamic behavior of a vehicle model on the uncertain friction space predicted first with traditional methods and then using ICKA.

2 NONLINEAR DYNAMICS APPLICATION

Spatial uncertainty quantification has been proved feasible and insightful in computer experiments where the spatial properties are static or nearly static, which include geology and climate studies. Unfortunately, sophisticated methods such as traditional Gaussian processes are of limited use when dealing with large, rapidly changing uncertainty problems because of the heavy computational burden associated with the methodology. Uncertainty quantification of dynamical system response carried out under these circumstances is hindered less by the numerical simulation cost and more by the prohibitive cost associated with sampling of the posterior. The proposed PFS and ICKA methods to correct this situation.

2.1 Dynamical Systems with Spatially Dependent Uncertainty

We are interested in characterizing the spatial uncertainty effects on a dynamical system

$$\dot{x} = f(x, t, \eta(x)), \tag{1}$$

subject to a given set of initial conditions. The function $f(x, t, u)$ is the intrinsic function that dictates the dynamics of the system; it is a known function obtained as the outcome of

---

[1]Schmitt et al. employed traditional Gaussian processes with this scenario in [17], resulting in accurate results but an exponential runtime vs. simulation time function impeding real-time simulation.

a mathematical modeling stage that is not of interest here. The quantity $\eta(x)$ is a random variable indexed by the space variable x. We want to solve the following problem:

> Characterize the distribution of the trajectory (random variable) $x(t)$ at a given time t or for a collection of times $t_1 < t_2 < \ldots < t_N$.

For example, (1) may represent the equations of motion of a vehicle, whereas $\eta(x)$ can be the uncertain road elevation or friction coefficient.

The same formalism can be applied to differential algebraic equations with virtually no change in the approach. For simplicity, we restrict our presentation to the case of ordinary differential equations (1).

In a stochastic framework, (1) is more rigorously reformulated as

$$\dot{x} = f(x, t, \eta(x, \omega)), \tag{2}$$

where ω is an element of the event space Ω. This results in a trajectory $x(t, \omega)$, which is thus a random variable. An example of quantity to be investigated is $E_\omega[x(T, \omega)]$, the expected state at a time T. The key to efficiently solving the stated problem is to sample $\eta(x, \omega)$ efficiently, that is, produce a sample function $\eta(x, \omega_1)$ for a given event $\omega_1 \in \Omega$ and for any x.

To simplify the notation, we state merely that $\eta(t)$ and $x(t)$ are random variables, and we do not explicitly denote their dependence on ω. Note that (1) is not a stochastic differential equation; rather, it is a differential equation with an infinite-dimensional stochastic parameter. The difficulty of the uncertainty quantification problem presented in this work is that we must accommodate an infinite-dimensional uncertainty space. Each realization is not a sample vector; rather, it is a sample "hypersurface." To sample the dependent trajectory $x(t)$, we must sample the x-indexed random surface $\eta(x)$.

We are interested in quantifying the uncertainty in the system (1) in two circumstances. In one circumstance, we are using a sample surface only once. This is the case if we know the starting point of a vehicle in a random terrain and we aim to characterize the expected end behavior of the vehicle. Here, sample values are needed only in a small area in a neighborhood of the path. Of course, the difficulty is that the path itself depends on the sample. We circumvent this difficulty in the case where the Gaussian process that models the spatial uncertainty has a compact kernel. By using conditional simulation we derive the ICKA, which has both low computational cost and low memory requirements.

In the second circumstance, we are interested in determining the statistics of $x(t)$ when the uncertainty appears in finite-dimensional random parameters (such as initial position or vehicle weight) in addition to the spatially dependent uncertainty. Sampling in an infinite dimensional space can be expensive. From this perspective, an economical approach is to sample conditional on instances of $\eta(x)$ according to the conditional probability density rule:

$$p(x(t)) = p(x(t)|\eta(x)) p(\eta(x)).$$

In that case, computing expectations of functions ψ of the trajectory $x(t)$ results in

$$E[\psi\,(x(t))] = E_\eta E\,[\psi\,(x(t))|\,\eta(x)]\,.$$

Therefore η sample surfaces can be used for multiple values of the other sample variables. As opposed to the preceding regime, samples are needed in the entire sampling region, so whole grid samples need to be provided. We use a fast Fourier transform approach that needs only $O(n\log(n))$ effort to create one sample from the correct distributions.

The two methods presented in this paper for sampling such uncertainty spaces make real-time dynamic simulation of or in an uncertain environment plausible. PFS allows for interpolation to be conducted several orders faster than traditional methods. Moreover, the method can generate infinite realizations from a single evaluation of the posterior; this is beneficial in applications requiring Monte Carlo simulation to understand the average and distributional nature of the dynamics. ICKA also allows for interpolation to be conducted several orders faster than traditional methods. Moreover, it exhibits a nonexponential relationship between runtime and problem size. This suggests that, if implemented correctly, ICKA can have negligible runtime compared to integration runtime. Also, in theory, ICKA's computational storage is invariant to problem size.

2.2   Application Example

To illustrate the versatility of the proposed methods, we use them in this paper to simulate a vehicle on randomly distributed patches of ice, in real time. The specifics of this simulation are detailed in [17], but the essentials are discussed here. First, a model for ice is required; studies in geostatistics suggest that the squared exponential is a representative correlation function for Gaussian random processes [1], and a variation of that function will be used herein. In order to address the natural bounds of ice (between dry friction μ_d and ice friction μ_s), a phase parameter f is used with the Gaussian processes:

$$f = -\ln\left(\frac{\mu_d - \mu}{\mu - \mu_s}\right) \in (-\infty, \infty). \tag{3}$$

Note that the phase parameter is an infinite-dimensional random variable indexed by a spatial variable. Take now a set of points in space $x_1, x_2, \ldots x_M$, and denote by $f_1, f_2, \ldots, f_M$ the random variables that represent the phase parameters at those points in space. Modeling the phase parameter as a Gaussian process means that the vector $F = (f_1, f_2, \ldots, f_M)$ has both a prior and a posterior multivariate normal distribution. The posterior distribution can be determined from well-established statistical considerations applicable for any multivariate normal distributions. The key attribute of Gaussian random processes is that the prior distribution can be generated from a bivariate function, the covariance function, and a univariate function, the mean function.

We use an exponential covariance function with hyperparameters $\theta_2 = \{\gamma, \alpha_{x1}, \alpha_{x2}\}$ (p.

86, [16]), and a linear mean function with hyperparameters $\theta_1 = \{a_0, a_1, a_2\}$:

$$m(x; \theta_1) = a_0 + a_1 x_1 + a_2 x_2, \tag{4}$$

$$k(x, x'; \theta_2) = \exp\left(-\left[\frac{(x_1 - x_1')}{\alpha_{x_1}}\right]^{2/\gamma} - \left[\frac{(x_2 - x_2')}{\alpha_{x_2}}\right]^{2/\gamma}\right). \tag{5}$$

The hyperparameters are determined from data by a maximum likelihood estimation as described in Section 3.1.

The vehicle model used on the ice, shown in Figure 1, has an open-loop steering system set to execute a constant radius turn without any axial forces applied. The model has three degrees of freedom: longitudinal motion V_x, lateral motion V_y, and yaw Ω_z. Three input functions determine the behavior of the model: steer angle δ_f and the front and rear wheel road adhesion coefficients μ_f and μ_r, respectively. The governing equations are as follows:
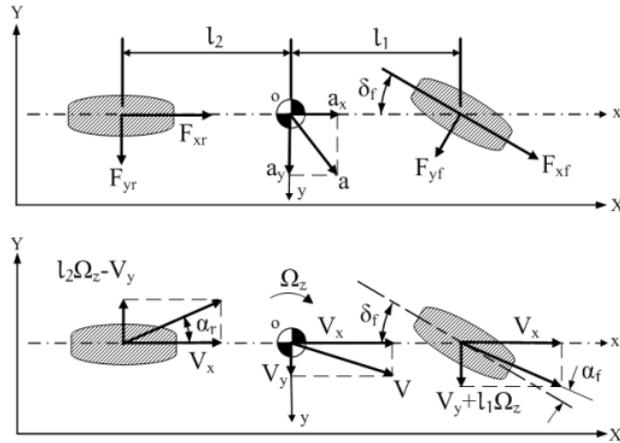

Figure 1: Bicycle model used in preliminary research of methodology [21].

$$\begin{aligned}
m(\dot{V}_x - V_y \Omega_z) &= -F_{yf} \sin \delta_f \\
m(\dot{V}_y + V_x \Omega_z) &= F_{yr} + F_{yf} \cos \delta_f \\
I_z \dot{\Omega}_z &= l_1 F_{yf} \cos \delta_f - l_2 F_{yr}
\end{aligned} \tag{6}$$

$$\begin{aligned}
\dot{X} &= V_x \cos \Theta_z - V_y \sin \Theta_z \\
\dot{Y} &= V_x \sin \Theta_z + V_y \cos \Theta_z \\
\dot{\Theta}_z &= \Omega_z.
\end{aligned} \tag{7}$$

The constitutive equations for the forces acting on the tires were provided by [21].

8

3 GAUSSIAN PROCESS DESCRIPTION OF SPATIAL UNCERTAINTY

Gaussian processes are a versatile approach for simulating infinite-dimensional uncertainty. We say that a spatially distributed random variable $\eta(x)$ is a Gaussian process with mean function $m(x; \theta_1)$ and correlation function $k(x, x'; \theta_2)$ if, for any set of space points $X = \{x_1, x_2, \ldots, x_M\}$, we have that

$$\eta(X) = \begin{pmatrix} \eta(x_1) \\ \eta(x_2) \\ \vdots \\ \eta(x_M) \end{pmatrix} \sim \mathcal{N}\left(m(X; \theta_1), K(X, X; \theta_2)\right).$$

Here we denote by $\mathcal{N}(f, K)$, where $f \in \mathbb{R}^M$ and $K \in \mathbb{R}^{M \times M}$ is the M–variate normal distribution with mean f and variance K given by

$$m(X; \theta_1) = \begin{pmatrix} m(x_1, \theta_1) \\ m(x_2, \theta_1) \\ \vdots \\ m(x_M, \theta_1) \end{pmatrix}, \quad K(X, X'; \theta_2) = \begin{pmatrix} k(x_1, x_1'; \theta_2) & k(x_1, x_2'; \theta_2) & \cdots & k(x_1, x_N'; \theta_2) \\ k(x_2, x_1'; \theta_2) & k(x_2, x_2'; \theta_2) & \cdots & k(x_2, x_N'; \theta_2) \\ \vdots & \vdots & \vdots & \vdots \\ k(x_M, x_1'; \theta_2) & k(x_M, x_2'; \theta_2) & \cdots & k(x_M, x_N'; \theta_2) \end{pmatrix},$$

where $X' = \{x_1', x_2', \ldots, x_N'\}$, and θ_1 and θ_2 are the hyperparameters of the mean and covariance functions, an example of which can be found in Section 2.2.

As in reference [16], we employ a Bayesian point of view in dealing with uncertainty. The hyperparameters θ_1 and θ_2 are obtained from a data set $\eta(D)$ at nodes $D = \{d_1, d_2, \ldots, d_M\}$. The posterior distribution of the variable $\eta(S)$ at node points $S = \{S_1, S_2, \ldots, S_N\}$, consistent with $\eta(D)$, is $\mathcal{N}(f^*, K^*)$ [16], where

$$f^* = K(S, D; \theta_2) \left[K(D, D; \theta_2) + \sigma_N^2 I_M\right]^{-1} (\eta(D) - m(D; \theta_1)) + m(S; \theta_1) \tag{8}$$

$$K^* = K(S, S; \theta_2) - K(S, D; \theta_2) \left[K(D, D; \theta_2) + \sigma_N^2 I_M\right]^{-1} K(D, S; \theta_2). \tag{9}$$

We have included the modification to the posterior distribution that is brought about by the noise in the data with variance σ_N^2.

The key issues in simulating from this posterior model are how to obtain the hyperparameters from data and how to sample from $\mathcal{N}(f^*, K^*)$, especially in the case where M is very large. The classical approach is to do a Cholesky factorization of K^*, a costly order $O(M^3)$ operation.

To simplify the notation, we will use the same symbols for a vector of random variables as for the vector of locations to which these variables are attached: for example, $\eta(D) \to D$, and $\eta(S) \to S$, the use of which will become clear from the context. In addition, we will not explicitly represent the dependence of $m(\cdot)$ and $k(\cdot, \cdot)$ on the hyperparameters θ.

3.1 Calculation of Hyperparameters

At the outset of the interpolation process, a prior mean function and a variogram model are selected as a function of space and several descriptive hyperparameters. Before training the posterior and sampling from it, the hyperparameters must be estimated to evaluate the prior. Since this step is unavoidable, we will briefly discuss and validate our learning method.

A commonly used method for the estimation of the hyperparameters from data is maximum likelihood estimation [16]. The method relies on the maximization of the log-likelihood function. In the multivariate Gaussian with mean $m(\theta)$ and covariance matrix $K_y(\theta)$ case, the log-likelihood function assumes the form

$$\log p(y|\theta) = -\frac{1}{2}W^T K_y(\theta)^{-1}W - \frac{1}{2}\log|K_y(\theta)| - \frac{M}{2}\log 2\pi \tag{10}$$

where $W = y - m(\theta)$ and y is the observed data. In the case of spatial uncertainty, the dependence on the hyperparameters θ appears by means of the spatial coordinates x.

In the example in Section 2.2, we have that $\theta = \{\theta_1, \theta_2, \sigma_n\}$. The gradients of the likelihood function can be computed analytically [16]:

$$\frac{\partial}{\partial\theta_{1j}}\log p(y|\theta) = \frac{1}{2}\text{tr}\left((K(\theta)^{-1}W(K(\theta)^{-1}W)^T - K(\theta)^{-1})\frac{\partial K(\theta)}{\partial\theta_{1j}}\right) \tag{11a}$$

$$\frac{\partial}{\partial\theta_{2j}}\log p(y|\theta) = -\left(\frac{\partial}{\partial\theta_{2j}}m(\theta)\right)^T K(\theta)^{-1}W \tag{11b}$$

$$\frac{\partial}{\partial\sigma_n}\log p(y|\theta) = \frac{1}{2}\sigma_n\text{tr}\left(K(\theta)^{-1}W(K(\theta)^{-1}W)^T - K(\theta)^{-1}\right). \tag{11c}$$

It can be proven that the gradient of $K(\theta)$ with respect to any hyperparameter maintains a periodic matrix structure. This fact suggests that all evaluation inside the trace operator can be done in the frequency domain after the K and $\frac{\partial K}{\partial\theta_{1j}}$ are diagonalized with FFT. Nonetheless, since the conditional sampling part of our procedure is far more time consuming, we use the FFT-based direct diagonalization only for its matrices.

To determine the hyperparameters, we use the MATLAB *fsolve* function. This function implements a quasi-Newton approach for nonlinear equations. We apply it to the nonlinear equations (11) obtained from the optimality conditions of maximizing the likelihood function. We note that using an optimization approach would have resulted in the need to compute the objective function at multiple points in hyperspace. This involves the evaluation of the determinant, which is currently not known to be possible in a matrix-free setting. Note that (11) can be evaluated in matrix-free fashion by using conjugate gradient techniques for linear systems that are equivalent to the application of K^{-1} by using an FFT approach. Although a matrix-free approach for MLE has not been implemented in this paper, this is a major block for using optimization approaches for large problem of the type we intend to solve in the future. We have thus decided to carry out the MLE procedure by a nonlinear equations approach in order to test its feasibility.

3.2 Conditional Density Decomposition

Both of the methodologies presented in this paper hinge on conditional sampling with Gaussian processes; that is, both data and previous samples are used when training new data sets. As the following sections demonstrate, the two new approaches lead to substantial efficiency gains. First, however, this manner of conditional sampling must be validated. The controversy invoked by our conditional sampling is that it learns and infers from previously computed Gaussian samples; later samples are conditioned on earlier ones. It must be shown that this method is statistically representative of the traditional Gaussian process where all samples are computed, all at once, from the observed data only. In statistical notation, our method can be equated to the old method by

$$\mathbb{P}(S_1, S_2|D) = \mathbb{P}(S_2|S_1, D) * \mathbb{P}(S_1|D), \tag{12}$$

where D is the observed data space, S_1 is the data space for the first iteration, and S_2 is the data space for the second iteration. The motivation behind our "divide and conquer" approach is twofold. First, we will show that the aggregate of all conditional simulations is sometimes more efficient to carry out than one comprehensive simulation. Second, the memory requirements of a successive conditioning approach can be much reduced compared to the all-at-once approach. We also note that the conditioning procedure can be applied recursively to result in as small a dimension of the variable to be sampled as needed, provided that the parameters of the conditional distribution can be efficiently determined.

The equality (12) can be expressed with probability densities by using the multivariate Gaussian distribution joint probability density function [16]:

$$p\left(\left(\begin{array}{c} S_1 \\ S_2 \end{array} \right) |m_1, \Sigma_1\right) = p(S_2|m_2, \Sigma_2) * p(S_1|m_3, \Sigma_3) \tag{13}$$

$$p(x|m, \Sigma) = (2 * \pi)^{-n/2}|\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(x - m)^T \Sigma^{-1}(x - m)\right), \tag{14}$$

where m is the posterior mean vector, Σ is the posterior covariance matrix, x is the sample, and n is the number of elements in the sample. The aggregate covariance matrix Σ_1 and the conditional covariance matrices Σ_2 and Σ_3 are defined in equation (33) in Appendix 1.

The proof of (14) can be obtained directly from the conditional probability rule (12) by using the expression of the conditional variance of Gaussian processes [16]. The complicated structure of the matrices in (33), however, makes the identities (13–14), which are central to all calculations in this work, impossible to verify directly.

We therefore undertake to prove (14) using only linear algebra. That proof, which draws on Schur complement techniques, is given in Appendix 1. This result will be taken advantage of in PFS and ICKA.

11

4 SPECTRAL APPROACH

Here we present the procedure and theory of our first method for conducting Gaussian processes on large spaces or fine-grained evaluation grids. As shown, PFS is used to ensure a data structure for which spectral methods may be used. The spectral analogy for each Gaussian processes step is then explained and validated.

4.1 Sampling Procedure: PFS

The question that motivated the PFS is simple: how can one conduct high-fidelity interpolation avoiding the bottleneck of large matrix storage and computation without compromising accuracy and robustness? A satisfactory answer can be produced as soon as two assumptions regarding the spatial distribution of the training input data and input data periodicity are made. The sampling follows a regimented strategy called *folding*.

In terms of the *spatial distribution of the training input data*, PFS is applied strictly to data initially provided on a lattice. This assumption certainly reduces the applicability of the methodology. Still, countless applications exist where data is compiled this way. Small-scale data collection, in custom lab experiments, for instance, is one area where lattice data collection is readily achievable. PFS makes a strong case for the priority of lattice data collection in experiment setup. Another important application of PFS is in satellite-collected data because the range of satellites makes their data collection flexible and adaptive. Note that although the paper focuses on the two-dimensional methodology, little alteration is required to extend the methodology to higher dimensional spaces.

The *input data periodicity* assumption is quantitatively described in the covariance function that we use when implementing the example in Section 2.2. We use a variation of 5 that is forced to be periodic:

$$
K_p(x, x^*) = \begin{cases} \prod_{s=1}^{l} \exp\left(-\left[\frac{|x_s - x_s^*|}{\alpha_s}\right]^{2/\gamma}\right) & |x_s - x_s^*| < L_s/2 \\ \prod_{s=1}^{l} \exp\left(-\left[\frac{L_s - |x_s - x_s^*|}{\alpha_s}\right]^{2/\gamma}\right) & |x_s - x_s^*| > L_s/2, \end{cases}
\tag{15}
$$

where l is the number of dimensions, x is a point in the first data set, x^* is a point in the second data set, L_s is the dimension respective length, and α_s is the dimension respective characteristic length-scale. Because this class of periodic matrices is an algebra, the periodic assumption ensures that the Gaussian processes-computed posterior covariance matrix (shown in 9) will also be periodic. This periodic nature will allow us to diagonalize the matrix with FFT.

The periodic assumption has been used in a broad range of research and applications fields to achieve significant efficiency gains. The aspect that adversely impacts this approach is the Gibbs phenomenon: the correlative misrepresentation of sample points near the space boundary, especially those within two characteristic length-scales of the boundary. Points near the boundary are predicted to have large correlation with boundary points on opposite
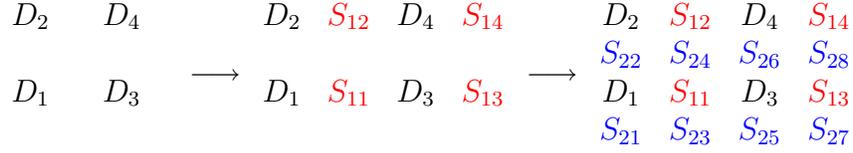
$$
\begin{array}{cc}
D_2 & D_4 \\
\\
D_1 & D_3
\end{array}
\quad\longrightarrow\quad
\begin{array}{cccc}
D_2 & S_{12} & D_4 & S_{14} \\
\\
D_1 & S_{11} & D_3 & S_{13}
\end{array}
\quad\longrightarrow\quad
\begin{array}{cccc}
D_2 & S_{12} & D_4 & S_{14} \\
S_{22} & S_{24} & S_{26} & S_{28} \\
D_1 & S_{11} & D_3 & S_{13} \\
S_{21} & S_{23} & S_{25} & S_{27}
\end{array}
$$

Figure 2: Pictorial representation of first two iterations of sampling method.

sides of the space, which is of course false unless the space is innately periodic, spherical or cylindrical, for instance. In natural distributions, this drawback is critical only if the inspected space is on the order of the characteristic length-scale, which is rarely the case. If this is the case, less sophisticated regression methods can be used to predict data. It is important to note the erroneous nature of our method near the boundaries, however, so that information is drawn only from accurate samples and simulations are conducted away from the boundaries.

The regimented sampling strategy required by PFS, introduced above as *folding*, is done iteratively. Each iteration folds or shifts the sample points half a resolution from the existing data; thus, each iteration doubles the sampled space. Subsequent iterations then use the previously interpolated data to learn from. This strategy is illustrated in Figure 2. Figure 3 shows in detail the direction and magnitude of each consecutive shift; each time the direction of shift is rotated by $-\pi/2$, and the magnitude of shift is reduced 50 percent. The conditional nature of sampling was discussed in detail in Section 3.2. The fold method of sampling ensures that the K_{12} covariance matrix will be square and FFT diagonalizable as proven and leveraged in Section 4.2.
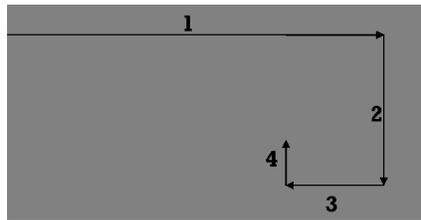

Figure 3: Direction and magnitude of the iterative shift.

Formally, we have a series of increasing grids of data, Γ_D, Γ_1, Γ_2, ..., Γ_n. We are given the data vector D on Γ_D, and we want to sample the output vectors S_1 on Γ_1, S_2 on Γ_2, ..., S_n on Γ_n.

We use iteratively the conditional probability density formula. This results in

$$
p\left(S_1, S_2, \ldots, S_n | D\right) = \prod_{i=1}^{n} p\left(S_i | D, S_1, S_2, \ldots S_{i-1}\right). \tag{16}
$$

13

The product rule allows for incremental sampling. Starting from D, we sample for sets S_i of increasing index i conditionally on the preceding sample values. A key observation is that, for Gaussian processes, the conditional distribution is algebraically easy to obtian (33).

We note that

$$\dim(S_i) = \dim(D) + \sum_{j=1}^{i-1} \dim(S_j), \quad i = 1, 2, \ldots, n. \tag{17}$$

This results in the vectors to the left and right of the conditional symbol having the same size. In turn, this allows us to show that a conditional sample can be easily calculated by using FFT techniques. Note, however, that it is quite possible that the FFT approach can work in some form even where (17) does not hold. The proof in that case will be the subject of further investigation.

This manner of sampling is not suitable for computing realizations at ad hoc locations because only coordinates $(N_1 s_x/2^n, N_2 s_y/2^n)$ are interpolated, where n is the number of fold iterations, s_x and s_y are the respective dimension observed data resolutions, and N_1 and N_2 are positive integers. Kriging methods are better suited for the computation of single realizations at predefined coordinates. However, as long as the coordinate requirements can tolerate small error, PFS is extremely effective because of the fineness that is achievable in small runtimes. In Section 6.4, plots are shown of runtime as a function of iterations or analogously sample resolution $s/2^n$.

4.2 Matrix-Free Computation of Posterior Distribution Characteristics

After the estimation of the hyperparameters, the conditional mean vector and covariance matrix are computed. The key to implementing an efficient version of the recursive conditioning method (16) is to design an efficient method for sampling from the conditional distribution $p(S_i | D, S_1, S_2, \ldots, S_{i-1})$.

In the following, subscript 1 refers to the coordinates of the observed and simulated data set, $\Gamma_1 = \{D, S_1, S_2, \ldots, S_{i-1}\}$, and subscript 2 refers to the half-resolution shift of those coordinates, $\Gamma_2 = S_i$. With this block notation, the prior distribution is

$$(\Gamma_1, \Gamma_2) \sim p(D, S_1, S_2, \ldots, S_i) \sim \mathcal{N}\left(\begin{pmatrix} m_1 \\ m_2 \end{pmatrix}, \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_W \end{bmatrix} \right).$$

Reference [16] presents a time-domain framework for computing the Gaussian posterior distribution characteristics from the observed data y incorporating measurement noise σ_n, using the prior mean, (4), and the prior covariance matrix, (5). This results in $p(\Gamma_2|\Gamma_1) \sim \mathcal{N}(f^*, K^*)$, where

$$f_* = m_2 + K_{21} K_W^{-1}(y - m_1) \in \mathbb{R}^N \tag{18a}$$

$$K_W = K_{11} + \sigma_n^2 I_D \in \mathbb{R}^{M \times M} \tag{18b}$$

$$K_* = K_{22} - K_{21} K_W^{-1} K_{12} \in \mathbb{R}^{N \times N}. \tag{18c}$$

14

Here M is the number of observed and currently simulated data, and N is the dimension of data vector to be sampled. Note that, in this notation, I_D is the identity operator when reduced to the D subblock of Γ_1, and it is 0 otherwise; it is not the identity matrix of dimension M. The matrix blocks are obtained from the covariance function as $K_{11} = k(\Gamma_1, \Gamma_1)$, $K_{12} = k(\Gamma_1, \Gamma_2)$ and $K_{22} = k(\Gamma_2, \Gamma_2)$ Note that (18) is a condensed version of (8) and (9), where the subscripts qualify which variate is on which covariance matrix axis.

Clearly, the operations involved in this process, matrix inverses and matrix products, will lead to excessive runtimes to compute the posterior. Furthermore, storage capabilities begin to falter when the number of sample points N is in the neighborhood of 10^4 because of the N^2 entries in the covariance matrices.

Our method emulates the framework outlined above, maintaining its robustness and accuracy, but uses the impositions discussed in Section 4.1 to conduct all expensive operations in the frequency domain. It is based on the key observation that all covariance matrices, K_{11}, K_{12}, and K_{22} (herein called the shift covariance matrix), are FFT diagonalizable; that is, $K_a = Q * D_a * Q'$, where D_a is a diagonal matrix and a is one of the pairs 11, 12, 22, and W. Q is a unitary matrix such that $Q = \mathcal{F}(I_N)$, where $\mathcal{F}$ is FFT over the space bearing the uncertainty. In a two-dimensional space, FFT is defined as (using the array indexing from 0)

$$\{x_{lq}\} \xrightarrow{\mathcal{F}} \{\hat{x}_{l_f q_f}\}; \quad \hat{x}_{l_f q_f} \triangleq \frac{1}{N^2} \left(\sum_{l,q=0}^{N-1} x_{lq} e^{-i\left(\frac{2\pi}{N} l l_f + \frac{2\pi}{N} q q_f\right)} \right), \quad \forall l_f, q_f = 0, 1, \ldots, N-1. \quad (19)$$

In the following, we denote by D the set of either data grid points or data random variables and by D_a a diagonal matrix identified by its subscript a. Note that N need not be the same in both the l and q variable, but the proofs are carried out under this assumption for algebraic convenience.

The matrices K_{11}, K_{12}, and K_{22} are periodic matrices. If the spatial dimension is one, they are circulant matrices. In two dimensions, they satisfy the following property, described by a generic matrix K, indexed by the grid index pairs

$$K_{lq,l'q'} = u\left((l - l')\mathrm{mod}N, (q - q')\mathrm{mod}N\right), \quad l, l', q, q' = 0, 1, \ldots, N-1. \quad (20)$$

Here u is a function defined on $\{0, 1, \ldots, N-1\} \times \{0, 1, \ldots, N-1\}$. In particular, K_{11}, K_{12}, and K_{22} satisfy (20) for

$$u_{11}(l, q) = u_{22}(l, q) = k\left(\frac{l L_x}{N}, \frac{q L_y}{N}\right), \quad u_{12}(l, q) = k\left(\frac{(l + 0.5) L_x}{N}, \frac{q L_y}{N}\right), \quad l, q = 0, 1, \ldots, N-1.$$

Recall that we take the covariance function k to be periodic over a rectangle of dimension $L_x \times L_y$. We consider only an x-shift, but the y-shift conclusions follow as well by the ensuing argument. The matrix K_W is also representable as (20), with the function

$$u_W(l, q) = u_{11}(l, q) + \sigma_n^2 \delta((l)\mathrm{mod}N_D, (q)\mathrm{mod}N_D), \quad l, q = 0, 1, \ldots, N-1,$$

where $\delta(l,q) = 1$ if $l = 0, q = 0$, and it is 0 otherwise.

We now provide a short proof of the fact that matrices with the structure described by (20) can be diagonalized by FFT. While this result is simple and ubiquitous for covariance matrices such as K_{11} and K_{22} [1], we are not aware of a simple proof in the Gaussian process literature based on the representation (20). The latter is central to our method, since it shows that K_{12} is also diagonalizable by FFT.

Indeed, take the FFT basis vectors: the FFT transform of the columns of the identity matrix. They are, up to a scaling parameter,

$$V^{l_f,q_f} = \left\{ v_{l,q}^{l_f q_f} \right\}_{l,q}; \quad v_{l,q}^{l_f q_f} = e^{-i\left(\frac{2\pi}{N} l l_f + \frac{2\pi}{N} q q_f \right)}, \quad \forall l, l_f, q, q_f = 0, 1, \ldots, N-1.$$

Here the variables l, q index the entries in the FFT basis vector whereas l_f, q_f index the vectors themselves. We then have that

$$
\begin{aligned}
\left[K V^{l_f q_f} \right]_{lq} &= \sum_{l',q'=0}^{N-1} K_{lq,l'q'} v_{l'q'}^{l_f q_f} \\
&\overset{(20)}{=} \sum_{l',q'=0}^{N-1} u\left((l - l') \bmod N, (q - q') \bmod N \right), e^{-i\left(\frac{2\pi}{N} l' l_f + \frac{2\pi}{N} q' q_f \right)} \\
&= e^{-i\left(\frac{2\pi}{N} l l_f + \frac{2\pi}{N} q q_f \right)} \sum_{l',q'=0}^{N-1} u\left((l - l') \bmod N, (q - q') \bmod N \right) e^{-i\left(\frac{2\pi}{N} (l-l') l_f + \frac{2\pi}{N} (q-q') q_f \right)} \\
&= e^{-i\left(\frac{2\pi}{N} l l_f + \frac{2\pi}{N} q q_f \right)} \psi(l_f, q_f) = \psi(l_f, q_f) v_{l,q}^{l_f q_f},
\end{aligned}
$$

for all $l_f, q_f, l, q = 0, 1, \ldots, N-1$. Here,

$$\psi(l_f, q_f) = \sum_{l,q=0}^{N-1} u(l, q) e^{-i\left(\frac{2\pi}{N} l l_f + \frac{2\pi}{N} q q_f \right)}. \tag{21}$$

Therefore,

$$K V^{l_f q_f} = \psi(l_f, q_f) V^{l_f q_f}, \quad \forall l_f, q_f = 0, 1, \ldots, N-1.$$

This shows that the matrix K is indeed diagonalizable by the FFT basis vectors and that $\psi(l_f, q_f)$ are its eigenvalues and, thus, its diagonal entries in that basis. Note that the diagonal elements will be real for K_{11} and K_{22} because of their symmetry but they will likely be complex for K_{12} because of the $\frac{L_x}{2N}$ shift.

To demonstrate the diagonalization property numerically, we use the diagonalization-by-FFT error operator Ψ, which is defined, by using the MATLAB *diag* function convention, as

$$\Psi(K) = \| Q'KQ - \text{diag}\left(\text{diag}(Q'KQ) \right) \|. \tag{22}$$

As a first check, at each iteration the priors K_{11} and K_{22} should be equal because of the PFS sampling strategy; the nonhomogeneity of the data is not factored in until the

posterior. Table 1 in Section 6.2.1 reflects this. Second, the diagonalizable-by-FFT property of K_{11}, K_{22}, and K_{12} is demonstrated in Table 1.

Next, using the results of our proof of diagonalizability for periodic matrix structures (20), we create a framework for direct diagonalization of the covariance matrices. The elements of the FFT diagonal of the covariance matrix can be generated directly by using (21), as opposed to constructing Q first. After some bookkeeping, these values are ordered into the FFT diagonal D_a, where a is either 12/22, or W. This direct diagonalization is necessary because the generation of the unitary matrix Q and the computation of the product $Q'KQ$ proves to be a substantial bottleneck for large values of N. The subscript notation we used for the time-domain covariance matrices is used for the directly diagonalized matrices D_{22}, D_W, D_{12}. The operator Γ is used, where

$$\Gamma(K_a) = \|Q'K_aQ - D_a\|, \quad a = 22, 12, W, \tag{23}$$

to show equivalence of the directly and indirectly derived diagonals for different iterations in the Section 6.2.1.

Now, all of the expensive operations in Gaussian processes described in (18a) can be conducted in the frequency domain after direct diagonalization by using (21). With our methods, the adjustment to the posterior mean f' is computed in the frequency domain and then mapped back to the time domain with a two-dimensional inverse FFT and added to the posterior mean.

$$
\begin{aligned}
f^* &= m_2 + \hat{f} \in \mathbb{R}^N \\
\hat{f} &= \mathcal{F}^{-1}(D_{21}D_W{}^{-1}\hat{W}) \in \mathbb{R}^N \\
D_W &= D_{11} + \sigma_n^2 I_M \in \mathbb{R}^{N \times N},
\end{aligned}
\tag{24}
$$

where $\hat{W} = \mathcal{F}(y - m(x))$.

The FFT diagonal of the covariance matrix can be used for sampling and thus does not ever have to be mapped to the time domain. This fact has significant implications for our runtime because we avoid large matrix products and inverses in posterior computation, and we avoid having to take large matrix square roots when sampling from $\mathcal{N}(f^*, K^*)$ (18):

$$D_* = D_{22} - D_{21}D_W{}^{-1}D_{12} \in \mathbb{R}^{N \times N}. \tag{25}$$

4.3   Matrix-Free Multivariate Gaussian Sampling

Sampling from a multivariate Gaussian distribution matrix such as $\mathcal{N}(f^*, K^*)$ is a mathematically simple, but computationally expensive, operation as it requires a large matrix square root of K^* to be taken. In the time domain, a Cholesky decomposition of the covariance matrix is taken, and the upper triangular matrix $L^{1/2}$ is multiplied by a vector of standard normals u. This procedure brings in the $O(N^3)$ computational effort for Cholesky for kernels that are not compactly supported.

The posterior distribution sampling can be done more efficiently in the frequency domain first by creating $\hat{u} = \mathcal{F}(u)$ and simulating

17

$$
\begin{aligned}
S^* &= f^* + \hat{f}^* \in \mathbb{R}^N \\
\hat{f}^* &= \mathcal{F}^{-1}((D(f^*))^{1/2}\hat{u}) \in \mathbb{R}^N; \ \hat{u} \sim \mathcal{N}\left(0_N, I_N\right).
\end{aligned}
\tag{26}
$$

Now, the Cholesky decomposition is avoided entirely, leading to remarkable efficiency gains, especially when Monte Carlo simulation is employed to compute trends and errors in dynamic simulations. Plots of the sample covariance matrix normalized error vs. number of samples, in Section 6.2.2, show convergence equivalent to that of traditional methods. These results demonstrate the accuracy of our process from start to finish. Since these methods are based entirely on FFT operations, they have a theoretical computational effort bound of $O(N log N)$, compared to the $O(N^3)$ bound of Cholesky.

5 INCREMENTAL COMPACT KERNEL APPROACH

The conditional sampling of Gaussian processes approach discussed in detail in Appendix 1 allows for a unique approach where small grid spaces can be interpolated dynamically *at runtime* throughout a simulation. This can replace the all-at-once presimulation, allowing for constant computational effort regardless of problem size. The idea of dynamic grid building is especially useful for Eulerian perspective dynamic simulation. In the sections that follow, our sampling framework for ICKA is described in detail.

5.1 Incremental Simulation

An interesting simplification appears when the prior covariance structure has compact support. That is, we have that

$$
\mathrm{Cov}\left[\gamma(x), \gamma(x')\right] = 0 \text{ for } \|x - x'\| \geq \epsilon.
$$

Consider now the initial point x_0 and an open set Γ, $x_0 \in \Gamma$. Denote by $\delta\Gamma$ the boundary of the set Γ and by $d(\cdot, \cdot)$ the Euclidean distance. Then trajectories that start at x_0 are independent of values of γ at points x' outside Γ, as soon as

$$
d(x(t), \delta\Gamma) > \epsilon.
$$

This situation points to the following strategy for incrementally sampling a trajectory. We take a ball $B_l(x_0, r)$ with a radius $r > \epsilon$. Here, $1 \leq l \leq \infty$ is the norm used in the definition of the ball. We simulate the Gaussian process at all grid points of fixed size inside $B_l(x_0, r)$. We advance the trajectory until it reaches a point no farther than $r - \epsilon$ from x_0 in the l norm. We denote that point x_1. We construct $B(x_1, r)$, and we sample on the grid in it, conditional on all the values sampled before, using (13). We advance the simulation up to a point x_2 no farther than $r - \epsilon$ from x_1, which we denote x_2. We continue the process until

we reach the prescribed time interval or other termination criterion. This should produce the same distribution of $x(t)$ as if the whole sampled surface were computed at once.

Since only a small fraction of sampling points needs to be computed, we expect this approach to be much faster than computing the whole surface. Moreover, the conditional distribution should consider conditioning only with respect to sample points on the grid no farther than $r + \epsilon$ from the center of the current ball. If the dynamical system progresses without a lot of "winding," that number is very small. Therefore, even the conditional simulation needs both low computational effort and low memory.

In our simulations we condition only with respect to the sample points in the ball preceding the current one. While one cannot a priori prove that this is approach sufficient, for our case the results give distributions indistinguishable from whole-domain sampling. In any case, the total number of points with respect to which we condition does not have to be any larger than the number of sampling grid points inside of a ball of radius $r + \epsilon$ in the l norm.

5.2 Compact Support Kernel

To allow for only local consideration of the sample space, one must assume that the covariance between points becomes exactly zero when their distance exceeds a certain threshold; this threshold quantifies the allowable magnitude of locality. Rasmussen and Willams [16, Ch. 4.2] suggest a family of piecewise polynomial covariance functions with compact support that guarantee positive definiteness in $\mathbb{R}^D$. We use a variation of the $\mathbb{R}^2$ equation for our numerical experiments,

$$k(r) = \begin{cases} (1 - \frac{r}{\epsilon})^{j+1} & r < \epsilon \\ 0 & r > \epsilon, \end{cases} \tag{27}$$

where $r = \sqrt{x_1^2 + x_2^2}$, ϵ defines the compact threshold, and j quantifies the function smoothness. The compact kernel allows for computational exploitation, but the question still lingers of how sensitive the simulations are to the use of a compact kernel; we explore this in Section 6.3.2.

For our convergence study we wanted to use a compact kernel that approximated our baseline γ-exponential function. Some trial and error was used to select the hyperparameters for both covariance functions to allow for likeness. If $\gamma = 2$ in (5), and $j = 3$ in (27), then likeness is achieved when $\epsilon = 5 * \alpha_{x_1} = 5 * \alpha_{x_2}$. Two like kernels are shown in Figure 4. Note that the compact kernel decays to zero at $r = \epsilon$.

5.3 Sampling Procedure: ICKA

As discussed, the sampling procedure of our methodology involves the dynamic interpolation of small grids in the regions of interest. Imagine a vehicle traveling in uncertain space. Values are needed only at the points of contact of the vehicle; thus, only data and previously sampled values (as stipulated by the proof in Appendix 1) within the compact kernel threshold, ϵ,
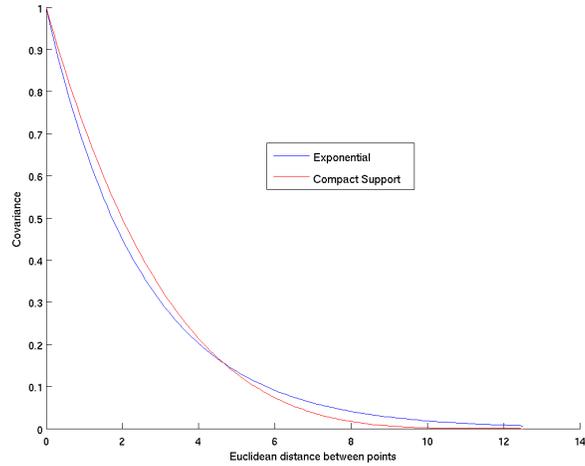
Figure 4: Comparison of exponential kernel and compact kernel; $\alpha_{x_1} = \alpha_{x_2} = 2.5$ and $\epsilon = 12.5$.

need to be used for training the interpolation. Since it would computationally expensive to interpolate with Gaussian processes for each integration step, we instead interpolate on small grids large enough to accommodate the simulation for multiple time steps; this strategy is presented pictorially in Figure 5. Further research is required to determine an optimal grid size to capture the efficiency of small grid interpolation without excess interpolation.
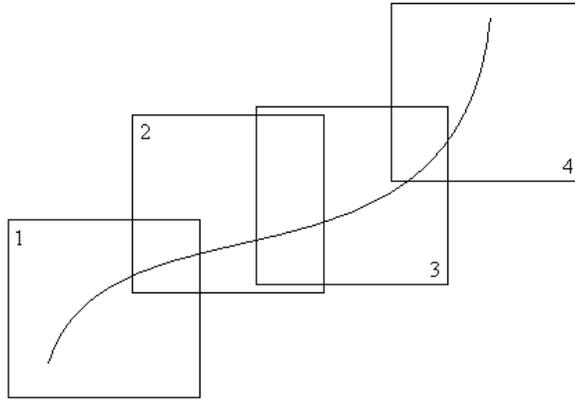
Figure 5: Pictorial representation of incremental sampling procedure.

For the interpolation of grid 1 in Figure 5, only the observed data within ϵ of the grid boundaries were used for training the new data set S. For grid 2, the observed data and S within ϵ of the grid boundaries were used for training the new data set S; S values were updated each time a new grid was built, and old values were discarded. For this reason, our

sampling procedure is valid only if the location of interest is such that $\frac{\partial x_D}{\partial t} > 0$. This way, our sampling procedure does, in theory, result in constant computational effort, invariant of problem size.

Another issue with our sampling procedure is the risk of discontinuities at grid changes. When the grid is updated over an integration step, one cannot guarantee that points of interest will undergo a continuous change, especially if a low evaluation grid resolution is used; the spline fit to the evaluation grid may change significantly at points of interest in a single time step. Note that, in order to curb this, it would be advantageous to include the coordinates of the points of interest in the training and test sets to translate continuity over a grid update. We could not do so, however, because our spline fitting procedure was strictly for data on a lattice and points of interest are not necessarily on this lattice. We found that it is helpful to use an explicit integrator to attenuate the discontinuities at grid changes. An alternative is to use an implicit integrator but stop and restart the integration at the grid updates, transferring the end conditions from the current integration to the initial conditions of the next, preventing the implicit integrator from encountering discontinuities.

6 NUMERICAL RESULTS

Numerical results are reported below to support the methodologies presented in this paper, as well as their use in the dynamic application described in Section 2.2. We present convergence studies demonstrating the accuracy of the methods together with effort vs. problem-size plots to illustrate the significant gains in terms of runtime and storage.

6.1 Validation of Hyperparameter Estimation

A leave-one-out cross-validation procedure was used to verify the accuracy and reliability of our hyperparameter estimation method. To start, we generated data from known hyperparameters and divided this data into ten unique training and testing sets. The training sets were used to estimate the hyperparameters, and the resulting posteriors were compared to the test set data. For each test, we verified the expected behavior of the prediction method: that the test data $f \sim \mathcal{N}(\hat{f}, \Sigma)$, where $\hat{f}$ and Σ are the mean and covariance matrix predicted from the training set, respectively. Using the central limit theorem, one can expect that

$$(f - \hat{f})^T \Sigma^{-1} (f - \hat{f}) \sim \mathcal{N}(n, \sqrt{2n}). \tag{28}$$

This is explained in detail in Appendix 2.

The performance of our estimation was checked for two unique sets of hyperparameters. For each hyperparameter test there were ten cross-validations, generating ten Z-scores. The resulting CDFs for the Z-scores are plotted against the expected CDF (given infinite cross-validation tests) and are shown in Figure 6. The plot qualitatively validates the results of our estimation method.

6.2 Validation of the Spectral Approach Results

For the spectral approach resulting in our PFS method, we investigate only the sampling of $\eta(x)$ itself and not the dynamical simulation. Once such sample surfaces consistent with the Gaussian process prior are obtained, simulations can be done by virtually any engineering package followed by postprocessing of the sampling trajectories. Such an approach was followed in [17], where the highly nonlinear dynamics of a vehicle model was simulated in the commercial multibody dynamics package ADAMS [14]. The value added by our work is in the sampling efficiency accompanied by the proof of the fact that our method is consistent with the state prior. This part will be validated by numerical examples below.

6.2.1 Validation of Diagonalizability and Diagonal Equivalence

The PFS method is such that the covariance matrices exhibit predictable and exploitable structure. Two key expectations are that the K_{11} and the K_{22} will be equivalent and that all priors K_{11}, K_{22}, and K_{12} will be diagonalizable with the unitary matrix Q for each fold iteration. Table 1 demonstrates these expectations for six fold iterations using the diagonalization error operator Ψ (22).

Table 1: Verification of Covariance Matrix Structure

Iteration	$\|K_{11} - K_{22}\|$	$\Psi(K_{22})$	$\Psi(K_{12})$
1	0	.0003e-10	.0001e-10
2	0	.0027e-10	.0010e-10
3	0	.0079e-10	.0036e-10
4	0	.0275e-10	.0466e-10
5	0	.1943e-10	.1351e-10
6	0	.4960e-10	.7803e-10

To leverage the operations in the frequency domain, we needed to be able to map to the FFT diagonal without producing the unitary matrix Q. We did so using the explicit formula for the diagonal entries in the FFT basis, (21). Table 2 compares the directly and indirectly diagonalized matrices, using (23), to show their equivalence for seven fold iterations.

6.2.2 Convergence Study of Spectral Approach

To verify accuracy and representativeness, we compare the mean and covariance computed from the sampled data generated by our PFS method with the mean and covariance posteriors computed with traditional methods, all samples computed at once. We expect to see the sample posteriors and the actual posteriors converge as the number of samples increases. The plots in Figure 7 show that our method's computed posterior covariance matrices converge at an approximate rate of $1/\sqrt{N}$, the same as with Cholesky simulation with PFS.

Table 2: Verification of Equivalence of Directly Diaganolized Matrices

Iteration	$\Gamma(K_{11})$	$\Gamma(K_{22})$	$\Gamma(K_{12})$
1	.0133e-13	.0133e-13	.0022e-13
2	.0178e-13	.0178e-13	.0266e-13
3	.0155e-13	.0155e-13	.0355e-13
4	.0355e-13	.0355e-13	.0375e-13
5	.2842e-13	.2842e-13	.0711e-13
6	.3553e-13	.3553e-13	.4263e-13
7	.5684e-12	.5684e-12	.5969e-12

Furthermore, the sample mean should converge to the traditional mean vector as computed in (18a). This property is shown in Figure 8.

6.3   Dynamic Simulation

The nonlinear vehicle simulation discussed previously was supplemented with both of the uncertainty quantification methodologies presented in this paper. In this subsection we present evaluation output for modeling the uncertain space encompassing the simulation. Then, we show a convergence study for ICKA with vehicle trajectory (instead of posterior characteristics). An efficiency analysis concludes this subsection.

6.3.1   Evaluation Grids for Dynamic Simulation

To demonstrate the versatility of our methods for the two-dimensional application discussed, we present several grid inputs and outputs. Also, we briefly discuss the legitimacy of spline fitting of sampled data to interpolate in continuous space, essential for our computer experiments.

Two cases are shown in Figure 9. In both, Gaussian processes with PFS sampling are used to interpolate the data, refining the space 2^4-fold; one realization is shown for both data sets. The first data set is stationary with high frequency, and the second is nonstationary with low frequency. The resulting refinements were computed in approximately 0.4 seconds.

For each refined sample, a cubic spline f is used to make predictions outside the N-node grid. This is important in continuous simulation. During the simulation, $f(\hat{X}(t_k))$ is interpolated to evaluate trajectory at all points at any time. Further, Monte Carlo simulation is used to understand the average behavior of the simulation.

Of course, at points away from the evaluation grid, the field function f approximated by splines no longer obeys the Gaussian process model; it is only an approximation of it. One can show, however, that, in the limit of the evaluation grid spacing going to zero, the results of the simulator converge to those that would be obtained if proper Gaussian process sampling had been employed at the points required by the integration procedure.

This convergence is due to the fact that almost any sample f surface is smooth [16]. This convergence is illustrated numerically in [17].

Figure 10 shows a single realization of the incremental approach during the dynamic simulation. Using the grid sizes we prescribed, we interpolate less than half of the rectangular space encompassing the simulation, leading to improvements in time and storage. The overlaps of the grids are not exactly equivalent, but the resolution of the evaluation is fine enough to maintain a relatively smooth transition; the type of smoothness required in grid transition depends entirely on the sensitivity of the system being simulated.

6.3.2   Convergence Study of the Incremental Approach

The incremental approach is validated numerically with a trajectory convergence study from the nonlinear dynamic simulation; this is done because the interpolated data itself (on a path) cannot be compared to the interpolated data from traditional methods (on a grid). Figures 11 and 12 show convergence of the cumulative distributions of the rotational velocity of the vehicle at the end of simulation for the traditional method and the incremental approach, both employing the compact support kernel (27). Figures 13 and 14 show the same convergence when the traditional method employs the exponential covariance function (5).

The same type of convergence is observed in both studies, with slightly less error in the case where both sampling procedures used the compact support kernel. This was expected because of the likeness between the covariance functions used, as demonstrated in Figure 4. We note, however, that while this likeness was hypothesized in [16], it was also left as an open question to be demonstrated by numerical experiments. Our work adds a body of evidence that backs up that hypothesis.

An interesting characteristic of the end trajectory samples is that they are not normally distributed, as shown by the deviation of the CDFs from normal in Figures 12 and 14. This behavior implicates complexities and sensitivities in our system. Further analytical and numerical studies remain to be done to more thoroughly understand the sensitivity of systems to the substitution of compact support, but for our vehicle it appears that dynamic grid simulation is a viable approach.

6.4   Efficiency Comparisons

Without doubt, uncertainty quantification methods need to become faster. Decades of work have spanned the mathematical and statistical frontiers, developing robust, reliable, and accurate methods. An understanding of areas where one can afford to make assumptions or impose conditions allowed us to develop two substantially faster methods. The plot shown in Figure 15 demonstrates the computational efficiency of PFS in terms of runtime and memory storage. PFS runtime starts lower and has a smaller exponential growth than that of traditional methods conducted in the time domain. Moreover, PFS demonstrates reduced storage requirements. Traditional methods implemented in MATLAB became impractical at eight iterations or approximately 9,000 data points; this requires the storage of matrices

with over 81×10^6 entries. In the frequency domain, all information is contained in the FFT diagonal values, so the largest storage is equivalent to N instead of N^2.

The plot shown in Figure 16 demonstrates the immense computational gains achievable with the incremental approach for long simulations. The simulation shown is that of a vehicle executing a ninety degree constant radius turn, longer simulation times leading to larger problem spaces. Note that the times given in this plot are for a single realization; the incremental method is less useful when Monte Carlo simulation is applied because new realizations require completely fresh simulations as opposed to a single Cholesky decomposition for all realizations as with traditional methods. As shown before, the traditional method exhibits exponential growth in runtime and storage with increasing problem size. It is shown that the incremental approach has a nearly linear growth in runtime with simulation time; that is, the Gaussian process computation time is negligible with respect to the simulation integration time. The computational storage effort, in theory, is invariant with respect to simulation time; the maximum storage remains constant because old grid information is purged when outside the compact kernel threshold.

7 CONCLUSIONS AND FUTURE WORK

The research presented in this paper introduces two conditional sampling methods for efficient handling of spatial uncertainty in dynamical systems. We prove that the methods correctly sample from the underlying Gaussian process. We demonstrate that the methods have both small computational costs and low memory requirements and are suitable for large-space and fine-grid spatial uncertainty sampling.

The two new methods developed are called periodic fold sampling (PFS) and the incremental compact kernel approach (ICKA). PFS (based on an FFT approach) is effective for interpolation and quantification of spatial data provided on a lattice. With only a periodic data assumption, reflected in the covariance matrix, and a conditional sampling regiment, PFS can do all the steps of traditional Gaussian processes in the frequency domain. Fast Fourier transforms are employed to map between the frequency domain and time domain when needed. Each step in this method – conditional sampling, hyperparameter estimation, computation of posterior with noise incorporated, and multivariate Gaussian sampling – is proven analytically and verified with numerical results. The computational effort is $O(n \log n)$ to sample a vector of length n compared to the $O(n^3)$ effort of the Cholesky-based approach. To our knowledge, the issue of efficiently sampling the posterior distribution on a grid has not previously been developed or demonstrated with this level of detail. We also note that the periodic assumption is not limiting if we slightly oversample our domain. The approach is limited by the fact that the data provided must reside on a coarse grid, many such example applications exist. In addition, the approach can benefit uncertainty quantification in many areas of mechanical engineering with infinite-dimensional uncertainty subspaces

with smooth realizations. These include contact mechanics and boundary roughness effects in continuum and fluid dynamics.

ICKA is effective for Eulerian perspective dynamic simulation. The method dynamically updates small interpolation spaces in proximity of the location of interest and thus avoids large grid interpolation completely. The method is only accurate statistically if a compact kernel is used to quantify covariance; that is, covariance between points becomes zero beyond some threshold distance. The compact kernel and implementation of the approach were discussed in detail.

For PFS, plots showing convergence of our sample covariance matrix and mean vector with traditionally computed posteriors demonstrate the accuracy of the method for a variety of data types in two-dimensional space. Convergence of dynamic system behavior under uncertainty interpolated with traditional methods and then our incremental method demonstrates the accuracy of ICKA.

The most important metric presented in this paper is the plot of effort vs. number of samples. It is shown that PFS leads to substantial runtime benefits, especially when the sample set is large. Furthermore, traditional methods are shown to run out of memory between 5,000 and 10,000 sample points, while PFS in theory should be able handle up to 10^7 sample points. Even more promising effort results are shown for the incremental approach. The effort of the Gaussian process interpolation appears to be negligible compared to the integration resulting in linear relationship between runtime and simulation duration. With only a few impositions, our frameworks combine the accuracy and reliability of Gaussian processes, with the speed achievable with spectral or white-noise methods. Furthermore, the restrictions of our methods are not severe, making one if not both of them valid and usable in nearly all data environments.

The combined speed and accuracy of PFS and ICKA should prove relevant in industry applications, such as vehicle dynamics and nuclear reaction simulation. In terms of future work this contribution opens up exciting new research opportunities. Since ICKA relies on a balance between grid size and the number of small grid interpolations required over an entire simulation, an optimization study would be challenging but would bring spatial data interpolation to unsurpassable efficiency levels. Research should focus on the multi-variate spatial uncertainty problem from both a "how" and "how faster" perspective.

Acknowledgments

A Multivariate Gaussian Conditional Sampling Proof

Prove

$$p\left(\left(\begin{array}{c} S_1 \\ S_2 \end{array}\right) | m_1, \Sigma_1\right) = p(S_2|m_2, \Sigma_2) * p(S_1|m_3, \Sigma_3),\tag{29}$$

where

$$p(x|m, \Sigma) = (2*\pi)^{-n/2}|\Sigma|^{-1/2}\exp\left(-\frac{1}{2}(x-m)^T\Sigma^{-1}(x-m)\right),\tag{30}$$

where m is the posterior mean vector, Σ is the posterior covariance matrix, x is the sample, and n is the number of elements in the sample.

The expressions involved in this multiplicative equality can thought of as composed of three parts for our proof: 1) the scalar, 2) the determinant scalar, and 3) the exponential term. This proof will demonstrate the equalities separately.

1. Numeric Scalar Equality

$$(2\pi)^{-n_1/2} = (2\pi)^{-n_2/2} * (2\pi)^{-n_3/2},$$

which follows from $n_1 = n_2 + n_3$. QED

2. Determinant Scalar Equality

Directly, we must prove

$$|\Sigma_1|^{-1/2} = |\Sigma_2|^{-1/2}|\Sigma_3|^{-1/2}.\tag{31}$$

It can be shown that this is identical to the equality

$$|\Sigma_1| = |\Sigma_2||\Sigma_3|.\tag{32}$$

From [16], the posteriors are

$$\Sigma_1 = k(\Omega_1, \Omega_1) - k(\Omega_1, D)\left(k(D,D) + \sigma_n^2 I\right)^{-1} k(D, \Omega_1)\tag{33a}$$

$$\Sigma_2 = k(S_2, S_2) - k(S_2, \Omega_2)\left(k(\Omega_2, \Omega_2) + \sigma_n^2 I_D\right)^{-1} k(\Omega_2, S_2)\tag{33b}$$

$$\Sigma_3 = k(S_1, S_1) - k(S_1, D)\left(k(D,D) + \sigma_n^2 I\right)^{-1} k(D, S_1),\tag{33c}$$

where $\Omega_1 = \left(\begin{array}{c} S_1 \\ S_2 \end{array}\right)$ and $\Omega_2 = \left(\begin{array}{c} D \\ S_1 \end{array}\right)$.

Next, we define

$$A = \begin{bmatrix} k(D,D) + \sigma_n^2 I & k(D,S_1) & k(D,S_2) \\ k(S_1,D) & k(S_1,S_1) & k(S_1,S_2) \\ k(S_2,D) & k(S_2,S_1) & k(S_2,S_2) \end{bmatrix} \tag{34}$$

$$B = \begin{bmatrix} k(D,D) + \sigma_n^2 I & k(D,S_1) \\ k(S_1,D) & k(S_1,S_1) \end{bmatrix} \tag{35}$$

$$C = k(D,D) + \sigma_n^2 I. \tag{36}$$

In this proof, we make heavy use of the Schur determinant formula

$$\det \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \det A_{11} \cdot \det (A_{22} - A_{21}A_{11}^{-1}A_{12}). \tag{37}$$

When using the Schur formula while solving for one of the right-hand side terms, we call it the inverse Schur formula.

It then follows that

$$\Sigma_1 = \frac{|A|}{|C|} \tag{38}$$

$$\Sigma_2 = \frac{|A|}{|B|} \tag{39}$$

$$\Sigma_3 = \frac{|B|}{|C|}. \tag{40}$$

Finally,

$$\frac{|A|}{|C|} = \frac{|A|}{|B|} * \frac{|B|}{|C|} \tag{41}$$

$$\frac{|A|}{|C|} = \frac{|A|}{|C|}.$$

$$QED$$

3. Exponent Equality

Some new notation will be used for this proof because of its higher complexity. We introduce the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F}$ to represent the respective posteriors. Note, that boldface is used to distinguish from other variables previously used in this proof;

especially, note the difference between D for data and $\mathbf{D}$ and the difference between A for the matrix in (34) and $\mathbf{A}$.

$$\mathbf{A} = x_1 - m_1 = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} - k(\Omega_1, D)k(D, D)^{-1}Y_D \tag{42a}$$

$$\mathbf{B} = \Sigma_1 \tag{42b}$$

$$\mathbf{C} = x_2 - m_2 = Y_2 - k(S_2, \Omega_2)k(\Omega_2, \Omega_2)^{-1}\begin{pmatrix} Y_D \\ Y_1 \end{pmatrix} \tag{42c}$$

$$\mathbf{D} = \Sigma_2 \tag{42d}$$

$$\mathbf{E} = x_3 - m_3 = Y_1 - k(S_1, D)k(S_D, S_D)^{-1}Y_D \tag{42e}$$

$$\mathbf{F} = \Sigma_3 \tag{42f}$$

Note that, for clarity, measurement noise is not included in this portion of the proof which can be shown to hold even when $\sigma_n \neq 0$. With the notation introduced, the hypothesis concerning exponents of (29) is

$$\mathbf{A}^T\mathbf{B}^{-1}\mathbf{A} = \mathbf{C}^T\mathbf{D}^{-1}\mathbf{C} + \mathbf{E}^T\mathbf{F}^{-1}\mathbf{E}. \tag{43}$$

Result α

Taking the inverse Schur with respect to $\mathbf{B}$, we get

$$-\mathbf{A}^T\mathbf{B}^{-1}\mathbf{A} = \frac{\det \begin{bmatrix} \mathbf{B} & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix}}{\det \mathbf{B}}. \tag{44}$$

Then ,taking the inverse Schur with respect to $k(D, D)$, we get

$$\det \mathbf{B} = \frac{\det A}{\det k(D, D)} \tag{45}$$

$$\det \begin{bmatrix} \mathbf{B} & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix} = \frac{\det \begin{bmatrix} \cdot & \cdot & \cdot & Y_D \\ \cdot & A & \cdot & Y_1 \\ \cdot & \cdot & \cdot & Y_2 \\ Y_D^T & Y_1^T & Y_2^T & Y_D^T k(D, D)^{-1}Y_D \end{bmatrix}}{\det k(D, D)}. \tag{46}$$

Here and in the following we denote by $\cdot$ an empty matrix block.

So, from (44-46), we obtain that

$$
-\mathbf{A}^T\mathbf{B}^{-1}\mathbf{A} = \frac{\det\begin{bmatrix} \cdot & \cdot & \cdot & Y_D \\ \cdot & A & \cdot & Y_1 \\ \cdot & \cdot & \cdot & Y_2 \\ Y_D{}^T & Y_1{}^T & Y_2{}^T & Y_D{}^T k(D,D)^{-1}Y_D \end{bmatrix}}{\det A}. \tag{47}
$$

Result β

Using the Schur complement of $\begin{pmatrix} \mathbf{D} & 0 \\ 0 & \mathbf{F} \end{pmatrix}$, we have

$$
\mathbf{C}^T\mathbf{D}^{-1}\mathbf{C} + \mathbf{E}^T\mathbf{F}^{-1}\mathbf{E} = -\frac{\det\begin{bmatrix} \mathbf{D} & 0 & \mathbf{C} \\ 0 & \mathbf{F} & \mathbf{E} \\ \mathbf{C}^T & \mathbf{E}^T & 0 \end{bmatrix}}{\det\mathbf{D}\cdot\det\mathbf{F}}. \tag{48}
$$

Result δ

Taking the inverse Schur with respect to $k(D,S_1)$ of D and inverse Schur with respect to $k(D,D)$ of $\mathbf{F}$, we obtain

$$
\begin{aligned}
\det\mathbf{D} &= \frac{\det A}{\det k(\Omega_2,\Omega_2)} \\
\det\mathbf{F} &= \frac{\det k(\Omega_2,\Omega_2)}{\det k(D,D)}
\end{aligned} \tag{49}
$$

$$
\det\mathbf{D}\cdot\det\mathbf{F} = \frac{\det A}{\det k(D,D)}. \tag{50}
$$

Result δ

First, execute a row transformation, then an inverse Schur with respect to $k(D,D)$, then a Schur with respect to $k(\Omega_2,\Omega_2)$:

$$
\det \begin{bmatrix} \mathbf{D} & 0 & \mathbf{C} \\ 0 & \mathbf{F} & \mathbf{E} \\ \mathbf{C}^T & \mathbf{E}^T & 0 \end{bmatrix} = \det \begin{bmatrix} \mathbf{F} & 0 & \mathbf{E} \\ 0 & \mathbf{D} & \mathbf{C} \\ \mathbf{E}^T & \mathbf{C}^T & 0 \end{bmatrix} \tag{51}
$$

$$
= \frac{1}{\det k(D,D)} \cdot \det \begin{bmatrix} k(\Omega_2,\Omega_2) & \cdot & 0 & Y_D \\ \cdot & \cdot & 0 & Y_1 \\ 0 & 0 & \mathbf{D} & \mathbf{C} \\ Y_D{}^T & Y_1{}^T & \mathbf{C}^T & Y_D{}^T k(D,D)^{-1} Y_D \end{bmatrix}
$$

$$
= \frac{1}{\det k(D,D)} \cdot \det k(\Omega_2,\Omega_2) \cdot \det \begin{bmatrix} \mathbf{D} & \mathbf{C} \\ \mathbf{C}^T & \Xi \end{bmatrix}, \tag{52}
$$

where

$$
\Xi = Y_D{}^T k(D,D)^{-1} Y_D - \begin{pmatrix} Y_D \\ Y_1 \end{pmatrix}^T k(\Omega_2,\Omega_2)^{-1} \begin{pmatrix} Y_D \\ Y_1 \end{pmatrix}. \tag{53}
$$

Taking the inverse Schur with respect to $k(\Omega_2,\Omega_2)$, $k(\Omega_2,\Omega_2)$ cancels out, and we have

$$
\det \begin{bmatrix} \mathbf{D} & 0 & \mathbf{C} \\ 0 & \mathbf{F} & \mathbf{E} \\ \mathbf{C}^T & \mathbf{E}^T & 0 \end{bmatrix} = -\frac{1}{\det k(D,D)} \cdot \det \begin{bmatrix} \cdot & \cdot & \cdot & Y_D \\ \cdot & A & \cdot & Y_1 \\ \cdot & \cdot & \cdot & Y_2 \\ Y_D{}^T & Y_1{}^T & Y_2{}^T & Y_D{}^T k(D,D)^{-1} Y_D \end{bmatrix}. \tag{54}
$$

Combining Results α, β, γ, and δ we prove our original hypothesis (43). QED

B Cross-Validation Hypothesis

We start by assuming $f \sim \mathcal{N}(\hat{f}, \Sigma)$. Thus,

$$
u = (f - \hat{f}) \sim \mathcal{N}(0, \Sigma). \tag{55}
$$

Then, we use $\Sigma^{-1/2}$ to make a linear map of u.

$$
m(\Sigma^{-1/2} u) = \Sigma^{-1/2} m(u) = 0 \tag{56}
$$

$$
\begin{aligned}
COV(\Sigma^{-1/2} u) &= (\Sigma^{-1/2}) COV(u)(\Sigma^{-1/2})^T \\
&= (\Sigma^{-1/2}) \Sigma (\Sigma^{-1/2})^T \\
&= \Sigma^{-1} \Sigma \\
&= I
\end{aligned} \tag{57}
$$

So, if $v = \Sigma^{-1/2}u \sim \mathcal{N}(0, I) \in \mathbb{R}^{(m \times 1)}$, we expect that $v^T v \sim \chi_n^2$, where n is the number of elements in the test set. By applying the central limit theorem to the χ_n^2 distribution (as $n \to \infty$), we can expect

$$(\hat{f} - f)^T \Sigma^{-1}(\hat{f} - f) \sim N(n, \sqrt{2n}). \tag{58}$$

References

[1] J. P. Chiles and P. Delfiner. *Geostatistics: modeling spatial uncertainty.* Wiley, New York, 1999.

[2] C. R. Dietrich and G. N. Newsam. A fast and exact method for multidimensional Gaussian stochastic simulations. *Water Resources Research*, 29:2861–2870, 1993.

[3] Michael S. Eldred, Anthony A. Giunta, Bart G. van Bloemen Waanders, Steven F. Wojtkiewicz, William E. Hart, and Mario P. Alleva. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis. SAND Report SAND2001-3796, Sandia National Laboratories, april 2002.

[4] Roger Ghanem and John Red-Horse. Propagation of uncertainty in complex physical system using a stochastic finite element approach. *Physica D*, 133:137–144, 1999.

[5] Roger Ghanem and Pol Spanos. *The Stochastic Finite Element Method: A Spectral Approach.* Springer, New York, 1991.

[6] A. Haldar and S. Mahadevan. *Probability, reliability, and statistical methods in engineering design.* John Wiley Chichester [England, 2000.

[7] J. K. Hammond and R. F. Harrison. *Covariance equivalent forms and evolutionary spectra for nonstationary random processes*, volume 62. Springer Berlin/Heidelberg, 1984.

[8] JC Helton and FJ Davis. Sampling-based methods. *Sensitivity Analysis*, pages 101–153, 2000.

[9] JC Helton and FJ Davis. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering and System Safety*, 81(1):23–69, 2003.

[10] Heuvelink G. Hengl T. and Stein A. Comparison of kriging with external drift and regression-kriging. technical note., 2003.

[11] D. Huang, T.T. Allen, W.I. Notz, and N. Zeng. Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models. *Journal of Global Optimization*, 34(3):441–466, 2006.

[12] D.R. Jones, M. Schonlau, and W.J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[13] Brus D.J. Knotters M. and Voshaar J.H.O. A comparison of kriging, co-kriging and kriging combined with regression for spatial interpolation of horizon depth with censored observations. *Geoderma*, 67:227–246(20), August 1995.

[14] MSC.Software. ADAMS User Manual. Available online at http://www.mscsoftware.com, 2005.

[15] S. Narayanan. Stochastic optimal-control of nonstationary response of a single degree-of freedom vehicle model. *Journal of Sound and Vibration*, 141:449–463, 1990.

[16] C.E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[17] Negrut D. Schmitt K., Madsen J. and Anitescu M. A gaussian process-based approach for handling uncertainty in vehicle dynamics simulation. Abstract accepted to 2008 ASME Int. Mech. Eng. Congress and Exp. Final acceptance pending, September 2008.

[18] M.L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, 1999.

[19] Yao T. Conditional spectral simulation with phase identification. *Mathematical Geology*, 30:285–308(24), April 1998.

[20] J. Tu, KK Choi, and YH Park. A New Study on Reliability-Based Design Optimization. *Journal of Mechanical Design*, 121:557, 1999.

[21] J.Y. Wong. *Theory of Ground Vehicles*. John Wiley & Sons, Inc., 2001.

[22] D. Xiu and J.S. Hesthaven. High-Order Collocation Methods for Differential Equations with Random Inputs. *SIAM Journal on Scientific Computing*, 27:1118, 2005.
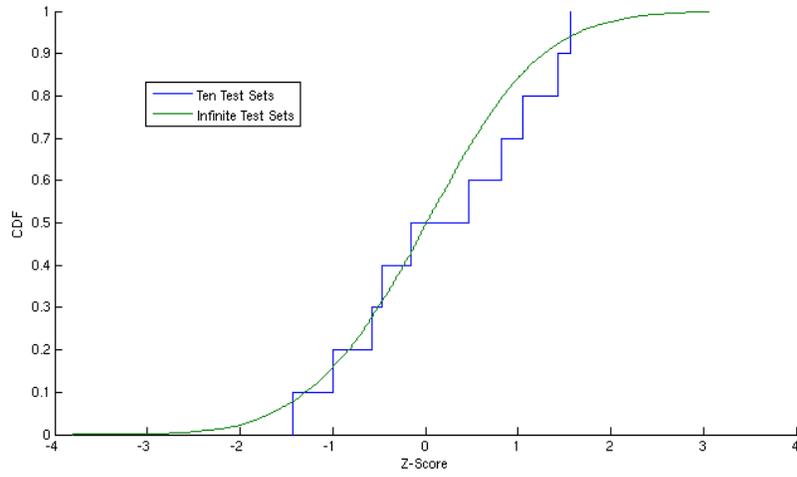
(a) Stationary data: $\gamma = 4, \alpha_x = \alpha_y = 1.5, a_0 = 1, a_1 = a_2 = 0$.
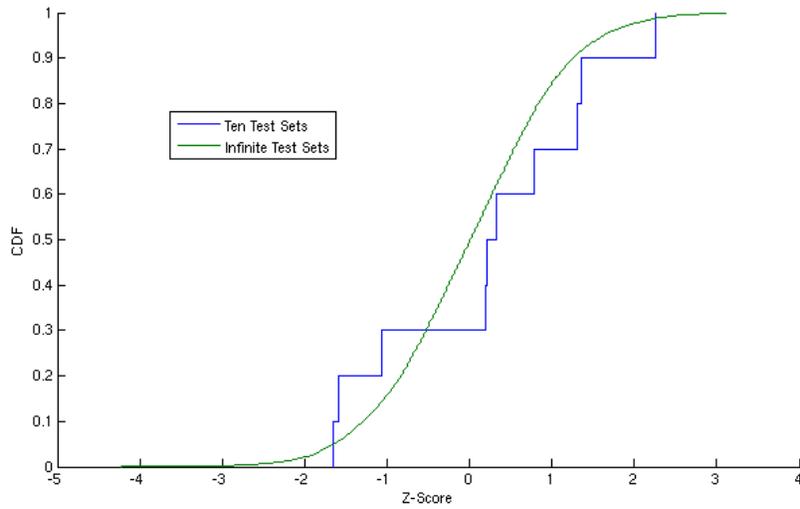
(b) Nonstationary data: $\gamma = 4, \alpha_x = \alpha_y = 1.5, a_0 = 3, a_1 = a_2 = 0.1$.
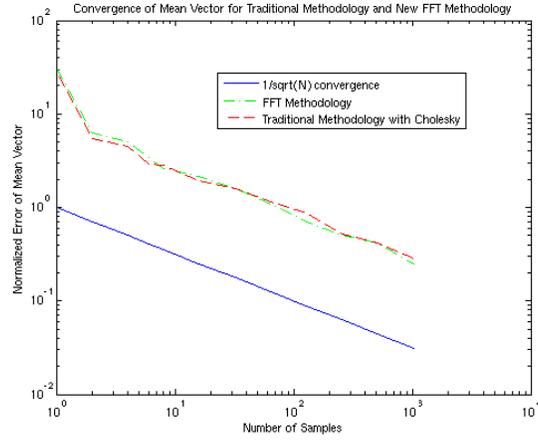
Figure 6: Cross validation CDF comparison.
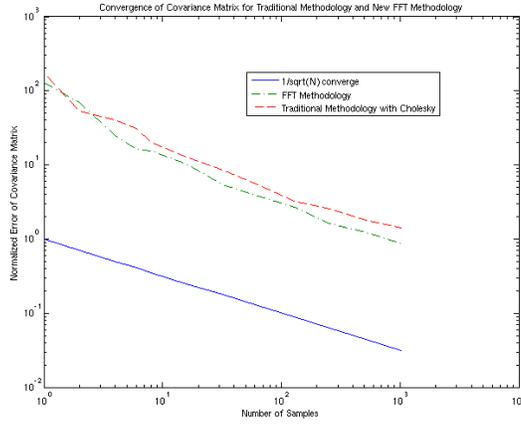
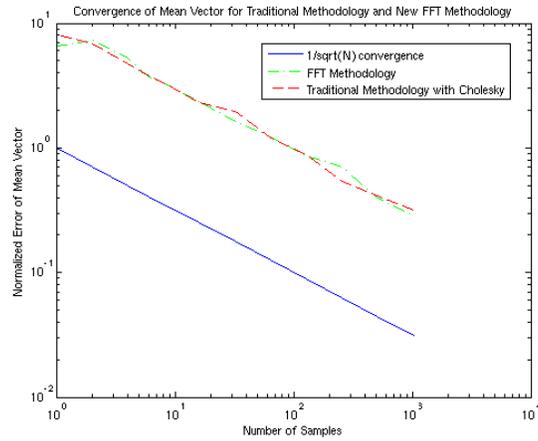(a) Stationary, high-frequency data: $\gamma = 4, \alpha_x = \alpha_y = 1, a_0 = a_1 = a_2 = 0$.
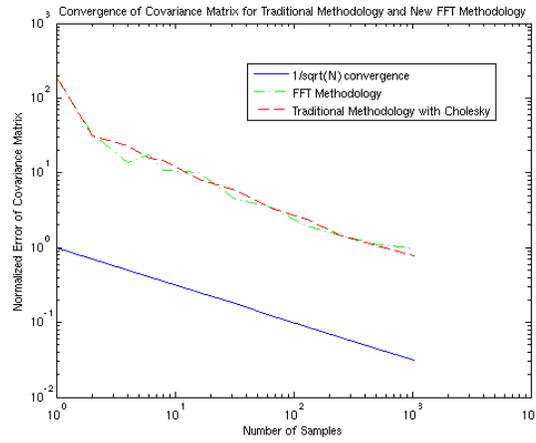
(b) Non-stationary, low-frequency data $\gamma = 4, \alpha_x = \alpha_y = 3, a_0 = 0.5, a_1 = a_2 = 0.1$.
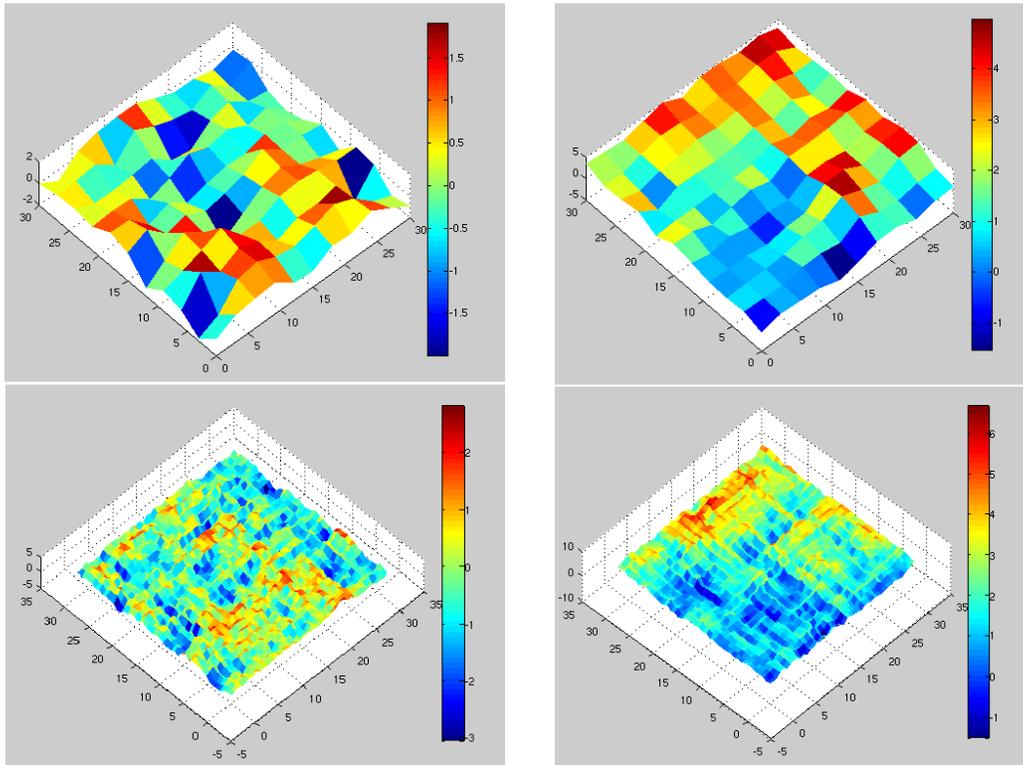
Figure 7: PFS derived covariance matrix convergence.

(a) Non-stationary, low-frequency data: $\gamma = 4, \alpha_x = \alpha_y = 3, a_0 = 0, a_1 = a_2 = .5$

(b) Stationary, high-frequency data: $\gamma = 4, \alpha_x = \alpha_y = 1, a_0 = 1, a_1 = a_2 = 0$

Figure 8: PFS derived mean vector convergence.

(a) $\gamma = 4, \alpha_x = \alpha_y = 3, a_0 = a_1 = a_2 = 0$.

(b) $\gamma = 4, \alpha_x = \alpha_y = 5, a_0 = 0, a_1 = a_2 = 0.06$.

Figure 9: Grid refinements.
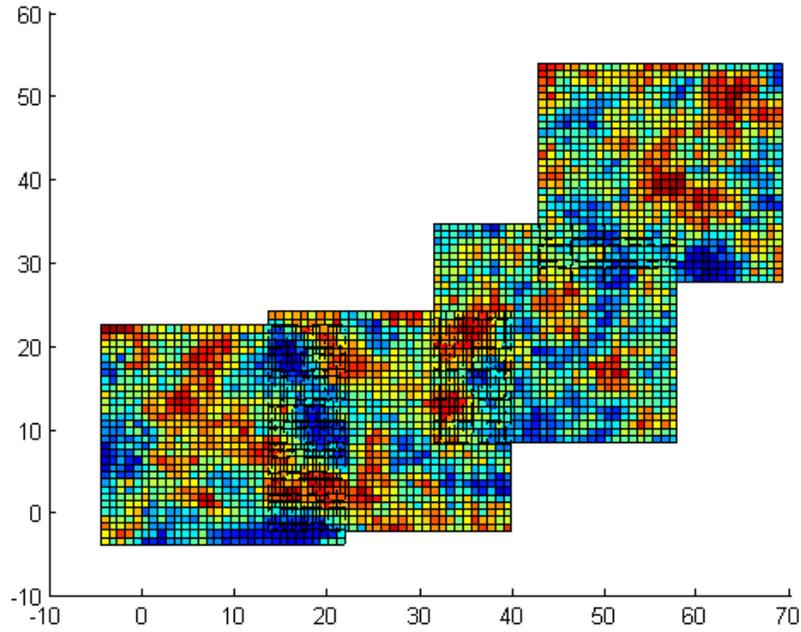
Figure 10: Example of a grid refinement with ICKA.
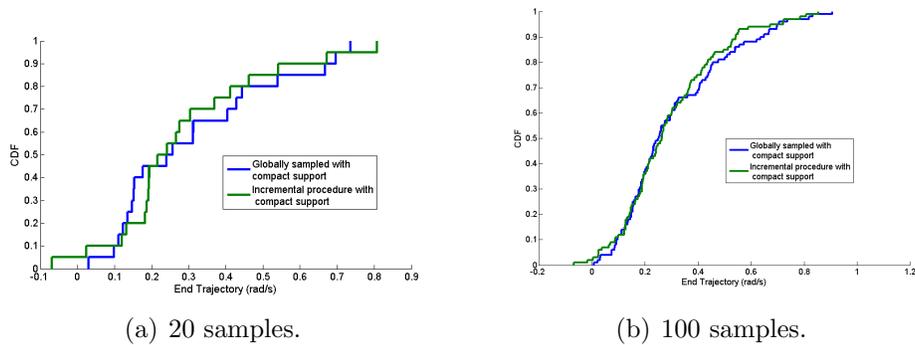
(a) 20 samples.

(b) 100 samples.

Figure 11: Convergence study of dynamic grid simulation when convergence target uses compact support.
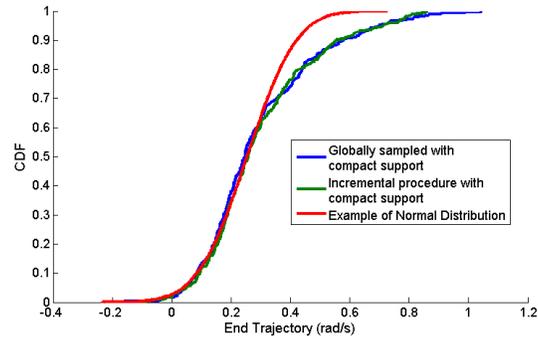
Figure 12: Convergence study of dynamic grid simulation when convergence target uses compact support: 500 samples.
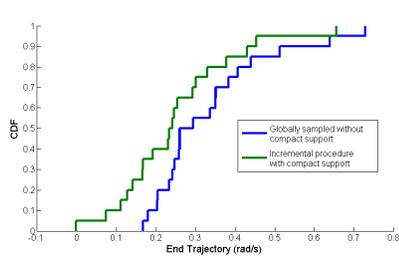
(a) 20 samples.
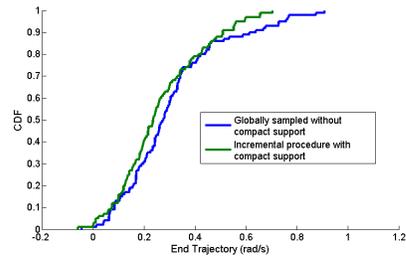
(b) 100 samples.

Figure 13: Convergence study of dynamic grid simulation when convergence target uses Eq. 5.
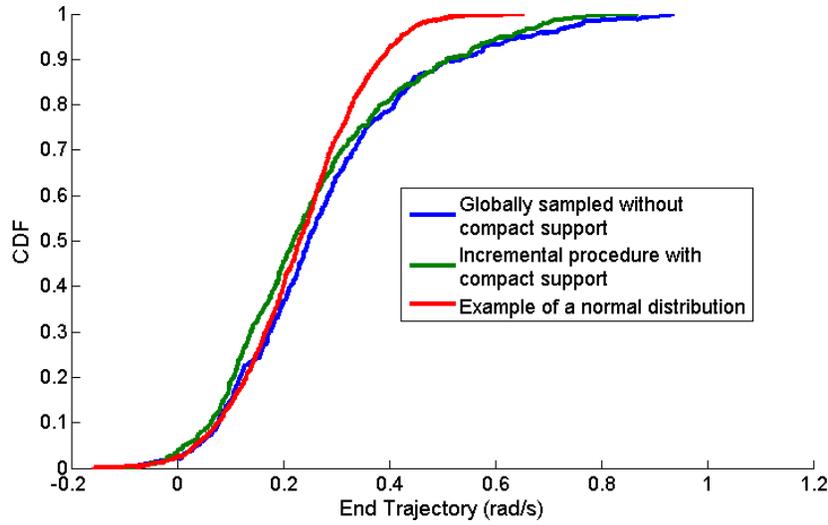
Figure 14: Convergence study of dynamic grid simulation when convergence target uses Eq. 5: 500 samples.
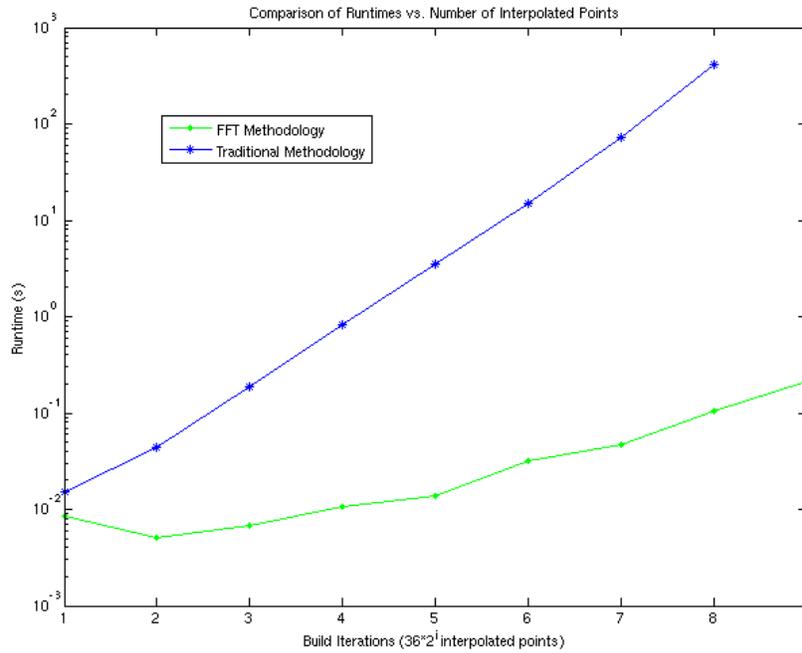

Figure 15: FFT methodology for periodic data shown to be several orders faster than traditional method. The traditional method reaches a memory limit after eight iterations.
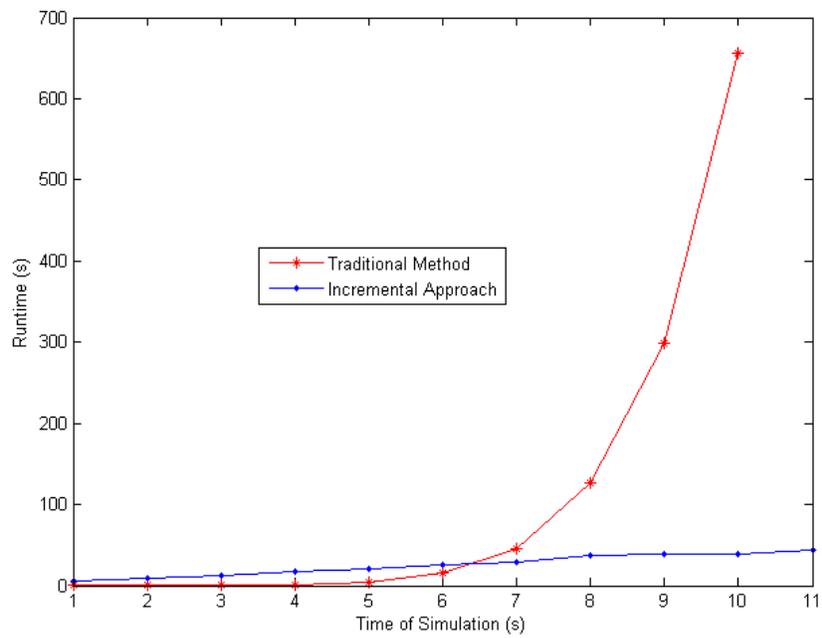
Figure 16: Runtime of traditional method and incremental approach shown as a function of simulation time or space to accommodate simulation. The traditional method reaches a memory limit after ten seconds in this simulation.