mathematics

MDPI

Article

Gaussian Process Regression with Soft Equality Constraints

Didem Kochan [1] and Xiu Yang [1,*]

Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015, USA
* Correspondence: xiy518@lehigh.edu

Abstract: This study introduces a novel Gaussian process (GP) regression framework that probabilistically enforces physical constraints, with a particular focus on equality conditions. The GP model is trained using the quantum-inspired Hamiltonian Monte Carlo (QHMC) algorithm, which efficiently samples from a wide range of distributions by allowing a particle's mass matrix to vary according to a probability distribution. By integrating QHMC into the GP regression with probabilistic handling of the constraints, this approach balances the computational cost and accuracy in the resulting GP model, as the probabilistic nature of the method contributes to shorter execution times compared with existing GP-based approaches. Additionally, we introduce an adaptive learning algorithm to optimize the selection of constraint locations to further enhance the flexibility of the method. We demonstrate the effectiveness and robustness of our algorithm on synthetic examples, including 2-dimensional and 10-dimensional GP models under noisy conditions, as well as a practical application involving the reconstruction of a sparsely observed steady-state heat transport problem. The proposed approach reduces the posterior variance in the resulting model, achieving stable and accurate sampling results across all test cases while maintaining computational efficiency.

Keywords: Gaussian process regression; quantum-inspired Hamiltonian Monte Carlo; equality constraints

MSC: 68U99



Academic Editor: Niansheng Tang

Received: 3 January 2025
Revised: 16 January 2025
Accepted: 19 January 2025
Published: 22 January 2025

Citation: Kochan, D.; Yang, X. Gaussian Process Regression with Soft Equality Constraints. *Mathematics* **2025**, *13*, 353. https://doi.org/10.3390/math13030353

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In many machine learning applications, measuring complex systems or evaluating computational models can be challenging due to the high costs, time requirements, or intensive computational demands. GP regression offers a solution by creating a surrogate model that approximates the behavior of these systems. As a Bayesian approach to supervised learning, GP regression is commonly applied in tasks such as regression and classification. It defines a probability distribution over a set of functions, allowing for predictions based on observed data through Bayesian conditioning [1,2].

Collaborating with Bayesian optimization, GP regression can be used broadly for applications such as industrial process time series forecasting modeling [3] and price predictions [4]. Although conventional GP regression has the disadvantage of computational complexity, various techniques can be used to improve its efficiency. For example, integrating stochastic fully symmetric interpolatory rules and local approximations into the GP regression framework reduces time complexity and improves prediction accuracy [5]. A key advantage of GP regression arises from its ability to incorporate prior information through kernels, enabling the model to make predictions while also providing uncertainty estimates [6]. However, in many cases, the prior information reflects underlying physical

laws [1]. For example, physical properties such as temperature, density, and viscosity must adhere to non-negativity constraints [7]. The standard GP regression framework may not account for these constraints, potentially leading to predictions that violate fundamental physical principles, resulting in unbounded or infeasible values. Therefore, the integration of physical constraints acts as a form of regularization, improving the accuracy and reliability of a GP model in real-world applications [8].

For example, enforcing inequality constraints in GP regression is challenging, as the conditional process does not retain typical GP properties [9]. Approaches to address these types of problems include data augmentation to enforce constraints at multiple locations [10] or using block covariance kernels [11]. Implicitly constrained GP methods show that linear constraints can be satisfied if met by the training data [12], while other methods demonstrate that imposing linear inequality constraints results in a compound GP with a truncated Gaussian mean [13]. Most approaches assume constraints on a finite set of points, approximating the posterior distribution accordingly [9,14,15]. The study in [16] extends the work in [9] by approximating the posterior distribution by a truncated multinormal distribution using various Markov chain Monte Carlo (MCMC) methods including Gibbs, Metropolis-Hastings (MH), and Hamiltonian Monte Carlo (HMC) for sampling. While truncated Gaussian methods provide high accuracy and the flexibility to satisfy multiple inequality conditions, they can be time-consuming, particularly when dealing with large datasets or high-dimensional problems, despite addressing some limitations identified in [9]. In [17], the QHMC algorithm is employed to train the GP model and enforce inequality and monotonicity constraints in a probabilistic manner. This method addresses the computational challenges associated with high-dimensional problems and large datasets. Unlike the truncated Gaussian techniques used in [16] for handling inequality constraints or the additive GP approach in [18] for imposing monotonicity constraints, the proposed method is capable of maintaining efficiency even in higher-dimensional settings.

Furthermore, constraints may also be more tightly integrated with the underlying physics: the GP can be constrained to satisfy constraints representing physical laws expressed as partial different equations (PDE) [8]. An earlier method for constraining GP with differential equations involves constructing a specialized covariance kernel so that the GP satisfies the constraint across the entire domain. This technique traces back to the divergence-free kernel [19] for vector-valued GPs. It not only enforces the constraint strictly but also avoids the computational cost associated with the four-block covariance matrix. However, this approach has limited applicability and requires specialized implementation since it involves analytically deriving a kernel with the necessary constraining properties. Later, a method for designing the covariance function of a multivariate GP subject to known linear operator constraints on the target function was presented in [20]. The method is designed to guarantee that any sample drawn from the resulting process will obey the constraints at all points. While there are other methods for constructing the kernel function, like the one proposed in [21], these approaches often involve specific conditions. This can make it more challenging to generalize the methods across different scenarios.

Traditional approaches for solving differential equations have long relied on analytical and numerical methods under well-posed conditions, constrained by the quality of boundary and forcing conditions. However, recent advancements are fundamentally shifting this paradigm by integrating probabilistic machine learning techniques with differential equations. For example, data-driven algorithms that use GP priors tailored for integro-differential operators enable the solving of general linear equations. These algorithms can work with sparse, noisy, multi-fidelity data that do not need to be located at the domain boundaries [11,22]. This approach not only provides predictive posterior distributions to quantify uncertainty but also enables adaptive solution refinement through active learning.

Building upon the works in [17,22], in this work, we integrate QHMC training into equality-constrained GP regression in a probabilistic manner. Additionally, an adaptive learning algorithm is employed to optimize the selection of constraint locations. Our work addresses the computational limitations caused by high dimensions or large datasets. It offers a flexible alternative to conventional numerical methods while also being scalable to high-dimensional problems. The effectiveness and precision of the QHMC algorithms are demonstrated across various scenarios, including both low-dimensional and high-dimensional synthetic problems, and one real example is provided for a three-dimensional heat transfer problem under PDE constraints. The main contributions of this work can be outlined in the following three points: (i) using QHMC sampling increases the accuracy and reduces the difference between the posterior mean and ground truth; (ii) applying QHMC within a probabilistic framework reduces variance and uncertainty; and (iii) the proposed algorithm serves as a robust, efficient, and flexible approach that can be applied across a wide range of problems.

2. Gaussian Process Under Equality Constraints

2.1. Standard GP Regression Framework

Consider a target function represented by the values $\mathbf{y} = (y^{(1)}, y^{(2)}, ..., y^{(T)})^N$, where each $y^{(i)} \in \mathbb{R}$ corresponds to an observation at the locations $\mathbf{X} = \{x^{(i)}\}_{i=1}^N$. Here, each $x^{(i)}$ is a d-dimensional vector within the domain $\mathcal{D} \in \mathbb{R}^d$. In line with the framework from [2], we approximate the target function using a GP, denoted as $Y(.,.) : D \times \Omega \to \mathrm{R}$. The GP Y can be written as

$$Y(x) := GP[\mu(x), K(x, x')],$$

where $\mu(.)$ is the mean function and $K(x, x')$ is the covariance function, each specified by

$$\mu(x) = \mathbb{E}[Y(x)], \quad \text{and} \quad K(x, x') = \mathbb{E}[Y(x) - \mu(x)][Y(x') - \mu(x')].$$

A typical choice for the covariance kernel function is the standard squared exponential covariance kernel, which is described as

$$K(x, x') = \sigma^2 \exp\left(-\frac{||x - x'||_2^2}{2l^2}\right) + \sigma_n^2 \delta_{x,x'},$$

where σ^2 represents the signal variance, $\delta_{x,x'}$ is the Kronecker delta function, and l is the length-scale. It is assumed that the observations include an additive independent identically distributed (i.i.d.) Gaussian noise term ϵ and having zero mean and variance σ_n^2. The set of hyperparameters, $\theta = (\sigma, l, \sigma_n)$, is estimated using the training data, typically by minimizing the negative marginal log-likelihood [2,23,24]:

$$-\log[p(\mathbf{Y}|\mathbf{X}, \theta)] = \frac{1}{2}[(\mathbf{y} - \mu^{\mathrm{T}} K^{-1}(\mathbf{y} - \mu) + \log|K| + N\log(2\pi)]. \tag{1}$$

In this work, we estimate the set of parameters by performing QHMC updates.

2.2. Quantum-Inspired Hamiltonian Monte Carlo

Quantum-Inspired Hamiltonian Monte Carlo is an advanced variation of the traditional HMC algorithm that introduces an additional random mass matrix for the particles, governed by a probability distribution. In standard HMC, the system's state is described by two main components: the position, represented by the original variables (x), and the Gaussian momentum, modeled through auxiliary variables (q). QHMC expands on this

by incorporating principles from quantum mechanics, particularly the energy–time uncertainty relation, to allow the mass of the particle to vary randomly according to a probability distribution. This modification introduces a new variable, the mass (m), alongside the position and momentum variables. The introduction of this third variable offers the key benefit of enabling the QHMC algorithm to explore a broader range of potential landscapes in the state-space, making it more versatile. Consequently, QHMC can more effectively navigate complex distributions, including those that are discontinuous, non-smooth, or have spiky features—areas where traditional HMC may struggle. This enhanced exploration capability gives QHMC an edge in sampling from challenging distributions, as it can adapt to varying local structures in the target distribution [25,26].

The quantum nature of QHMC can be illustrated through the example of a one-dimensional harmonic oscillator, as described in [26]. Consider a ball with a fixed mass m attached to a spring at the origin. Let x represent the displacement of the ball from the origin. The restoring force acting on the ball is given by $F = -kx$, which pulls the ball back toward the origin, causing it to oscillate with a period $T = 2\pi\sqrt{\frac{m}{k}}$. In standard HMC, the mass m is fixed at a value of 1. QHMC, however, introduces a key enhancement by allowing the mass to vary with time. This enables the ball to have different accelerations and, as a result, explore a wider range of distribution landscapes. Specifically, QHMC can use a shorter time period T, which corresponds to a smaller mass m, to efficiently traverse broad and relatively flat regions of the distribution. On the other hand, when dealing with regions that are spiky or have sharp features, QHMC can adapt by employing a longer time period T, associated with a larger mass m, ensuring that all areas of the landscape are thoroughly explored. This flexibility in adjusting the mass and time scale allows QHMC to better navigate complex and varied distributions compared to standard HMC [26].

The implementation of QHMC is relatively simple: a stochastic process $M(t)$ is constructed for the mass, and at each time t, the mass $M(t)$ is sampled from a distribution $P_M(M)$. The only addition to the standard HMC process is resampling the positive-definite mass matrix at each step. In practice, we assume that $P_M(M)$ is independent of x and q, and we select a mass density function $P_M(M)$ with mean μ_m and variance σ_m^2 as $\log m \sim \mathcal{N}(\mu_m, \sigma_m^2)$, $M = mI$, where I is the identity matrix. Then, the QHMC framework simulates the following dynamical system:

$$d\begin{pmatrix} x \\ q \end{pmatrix} = dt\begin{pmatrix} M(t)^{-1}q \\ -\nabla U(x) \end{pmatrix}.$$

3. Proposed Method

In the QHMC context provided in the previous section, the potential energy function for the QHMC system is given by $U(x) = -\log[p(\mathbf{Y}|\mathbf{X}, \theta)]$, i.e., the negative of marginal log-likelihood. Algorithm 1 summarizes the steps of QHMC sampling, and, here, we consider the location variables $\{x^{(i)}\}_{i=1}^{N}$ in GP model as the position variables x in Algorithm 1. The method evolves the QHMC dynamics to update the locations x. In this work, we implement the QHMC method for equality-constrained GP regression in a probabilistic manner.

Rather than strictly enforcing all constraints, the methods proposed in [7,17] focus on minimizing the negative marginal log-likelihood function from Equation (1) while allowing for occasional violations of the constraints with a small probability. For example, an equality-constrained optimization problem can be modified by introducing the following condition to the problem:

$$P[(\mathbf{Y}(x)|x, \theta) \neq c] \leq \eta, \quad \text{for all} \quad x \in \mathcal{D}, \tag{2}$$

where $0 < \eta << 1$. Unlike methods that enforce constraints using a truncated Gaussian assumption [9] or by applying inference techniques such as the Laplace approximation and expectation propagation [27], the proposed approach maintains the Gaussian posterior typical of standard GP regression. This method introduces only a slight modification to the existing cost function. For a model that follows a Gaussian distribution, the constraint can be reformulated in terms of the posterior mean and posterior standard deviation:

$$y^*(x) + \phi^{-1}(\eta)s(x) = 0, \quad \text{for all} \quad x \in \mathcal{D}, \tag{3}$$

where $y^*(x)$ stands for the posterior mean, s is the standard deviation, and ϕ is the cumulative distribution function of a Gaussian random variable. In this work, we set η 2.2% for demonstration purposes, as in [7]. This gives $\phi^{-1}(\eta) = -2$, indicating that the violation is allowed for two standard deviations. Then, the reformulation of the optimization problem is given as follows:

$$\begin{aligned} \text{argmin}_\theta \quad & -\log[p(\mathbf{Y}|\mathbf{X},\theta)] \quad \text{such that} \\ & c - 2s(x) \leq y^*(x) \leq c + 2s(x). \end{aligned} \tag{4}$$

In this specific formulation of the optimization problem, there is a functional constraint outlined in Equation (4). It may be impractical or even impossible to satisfy this constraint at every point throughout the entire domain. Consequently, we implement a strategy that enforces Equation (4) only at a chosen set of m constraint points, denoted as $\mathbf{X}_c = {x_c^{(i)}}_{i=1}^m$. We reformulate the optimization problem:

$$\begin{aligned} \text{argmin}_\theta \quad & -\log[p(\mathbf{Y}|\mathbf{X},\theta)] \quad \text{such that} \\ & c - 2s(x_c^{(i)}) \leq y^*(x_c^{(i)}) \leq c + 2s(x_c^{(i)}) \quad \text{for all} \quad i = 1, 2, ..., m, \end{aligned} \tag{5}$$

where hyperparameters are estimated to enforce bounds. Solving this optimization problem can be quite challenging, requiring the addition of regularization terms to the approach presented in [7]. Instead of directly solving the optimization problem, this work employs the soft-QHMC method first introduced in [17], which introduces inequality constraints with a high probability (for example, 95%) by selecting a specific set of m constraint points within the domain. Non-negativity is then enforced on the posterior GP at these selected points.

The QHMC algorithm is used to minimize the log-likelihood from Equation (1). By utilizing Bayesian estimation techniques [28], we can approximate the posterior distribution based on the log-likelihood function and the prior probability distribution, as outlined below:

$$p(\mathbf{X}, \theta|\mathbf{Y}) \propto p(\mathbf{X}, \theta, \mathbf{Y}) = p(\theta)p(\mathbf{X}|\theta)p(\mathbf{Y}|\mathbf{X}, \theta). \tag{6}$$

The QHMC training process begins with this Bayesian learning and switch to an MCMC procedure to draw samples generated within the Bayesian framework. The details are provided in Algorithm 1. A general sampling procedure at step t is given as follows:

$$\begin{aligned} X^{(t+1)} &\sim \pi(X|\theta) = p(X|\theta^{(t)}, Y), \\ \theta^{(t+1)} &\sim \pi(\theta|X) = p(\theta|X^{(t+1)}, Y). \end{aligned} \tag{7}$$

Algorithm 1 QHMC Training for GP with Equality Constraints

Input: Initial point x_0, step size ϵ, number of simulation steps L, mass distribution parameters μ_m and σ_m.

for $t = 1, 2, ...$ **do**
  Resample $M_t \sim P_M(M)$
  Resample $q_t \sim N(0, M_t)$
  $(x_0, q_0) = (x^{(t)}, q^{(t)})$
  $q_0 \leftarrow q_0 - \frac{\epsilon}{2}\nabla U(x_0)$
  for $i = 1, 2, ..., L-1$ **do**
    $x_i \leftarrow x_{i-1} + \epsilon M_t^{-1} q_{i-1}$
    $q_i \leftarrow q_{i-1} - \frac{\epsilon}{2}\nabla U(x_i)$
  end for
  $x_L \leftarrow x_{L-1} + \epsilon M_t^{-1} q_{L-1}$
  $q_L \leftarrow q_{L-1} - \frac{\epsilon}{2}\nabla U(x_L)$
  $(\hat{x}, \hat{q}) = (x_L, q_L)$
  MH step: $u \sim \text{Uniform}[0, 1]$;
  $\rho = e^{-H(\hat{x},\hat{q}) + H(x^{(t)}, q^{(t)})}$;
  if $u < \min(1, \rho)$ **then**
    $(x^{(t+1)}, q^{(t+1)}) = (\hat{x}, \hat{q})$
  else
    $(x^{(t+1)}, q^{(t+1)} = (x^{(t)}, q^{(t)})$
  end if
end for
Output: $\{x^{(1)}, x^{(2)}, ...\}$

3.1. Adaptive Learning Mechanism

Instead of randomly selecting m constraint points, the proposed algorithm initiates its process with an empty set of constraints and adaptively determines the locations of the constraint points one by one. This adaptive approach allows for a more sophisticated selection process that can lead to improved performance. Throughout this iterative process, several strategies are employed to effectively add constraints based on the behavior of the model.

The first strategy is known as the constraint-adaptive approach. Specifically, the algorithm continuously examines whether the constraints are being satisfied at various pre-decided locations within the domain. For each candidate location, the function value is calculated, and if a violation of the constraint is detected, a constraint point is then added at that specific location. This real-time evaluation ensures that the algorithm can quickly respond to areas where the constraints are not met, allowing for a more dynamic and responsive optimization process.

The second strategy employed is the variance-adaptive approach. This technique involves calculating the posterior variance within the test set to identify which locations exhibit the highest uncertainty. By focusing on these high-variance areas, the algorithm aims to add constraint points that will help to reduce overall prediction variance, thereby increasing the stability of the model's outputs. This approach ensures that the algorithm prioritizes regions that may contribute most significantly to uncertainty, effectively allowing it to enhance the reliability of its predictions.

Additionally, the algorithm employs a hybrid strategy that combines both constraint and variance adaptation. In this combined approach, a threshold value is established for the variance. The algorithm identifies constraint points at locations where the highest prediction variance is observed, aiming to address and stabilize these areas first. Once the variance for these points is successfully reduced to the predetermined threshold value, the algorithm switches back to the constraint-adaptive approach. In this phase, it focuses on locating constraint points in areas where violations are occurring, ensuring that the model

remains robust and compliant with the necessary constraints while continually refining its predictions. By integrating these strategies, the algorithm effectively navigates the challenges posed by the optimization problem, leading to more accurate and reliable results.

An example of the variance-adaptive approach can be seen in the workflow of soft equality-constrained GP regression, as outlined in Algorithm 2. In this implementation, QHMC sampling, described in Algorithm 1, serves as the training method for the Gaussian process (GP). In this specific version of soft equality-constrained GP regression, constraint points are strategically positioned at locations where the posterior variance is at its highest. We summarize the workflow of the soft equality-constrained GP regression using QHMC in Figure 1, where we give the main steps to use Algorithms 1 and 2.

Algorithm 2 GP Regression with Soft Constraints

1: Specify m constraint points denoted by $\mathbf{X}_c = x_c^{(i)}{}_{i=1}^{m}$, where corresponding observation $y^*(x_c)^{(i)}$.
2: **for** $i = 1, 2, ..., m$ **do**
3: Compute the MSE of $s^2(x_c^{(i)})$ of MLE prediction $y^*(x_c)$ for $x_c \in \mathcal{D}$.
Obtain observation $y^*(x_c)^{(i)}$ at $x_c^{(i)}$
Locate $x_c^{(i+1)}$ for the maximum of $s^2(x_c^{(i)})$ for $x_c \in \mathcal{D}$.
4: **end for**
Construct the MLE prediction of $y^*(x)$ using QHMC training.
Output: $y^*(x)$.

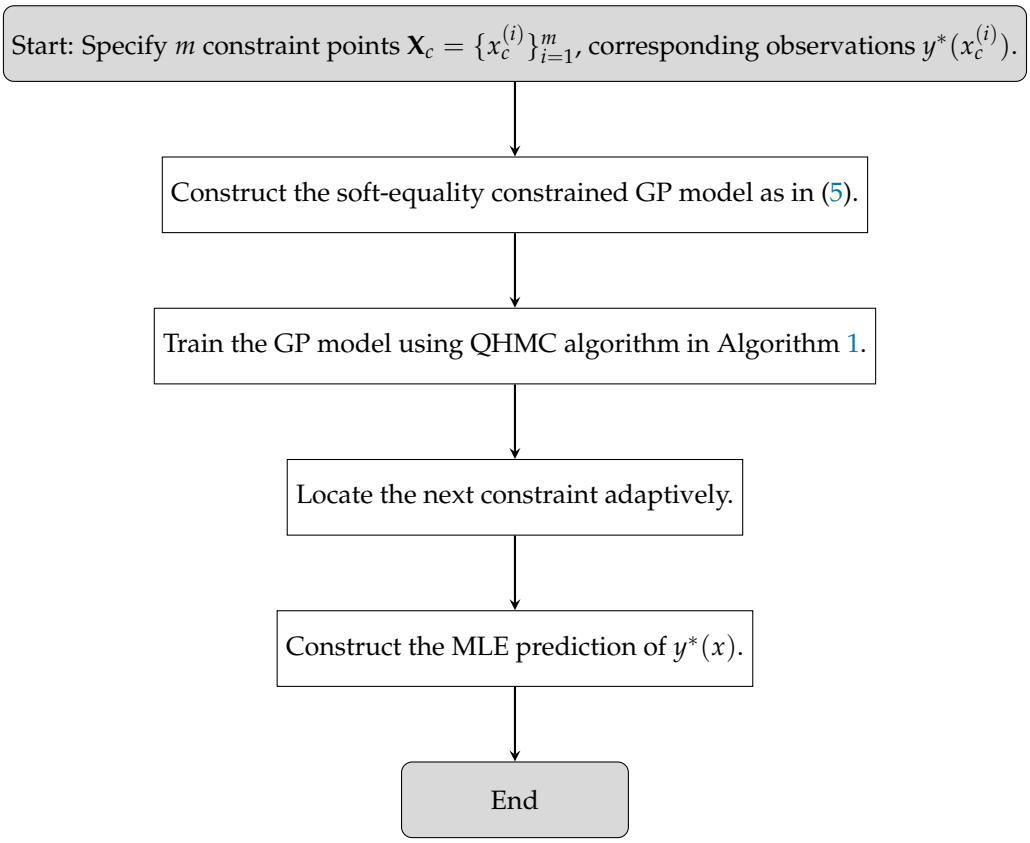


Figure 1. Workflow of soft equality-constrained GP regression using QHMC training.

3.2. Convergence Properties of the Probabilistic Approach

This section establishes that enforcing constraints on a subset of locations x in the domain $\mathcal{D}$ preserves convergence. We will follow the proof provided in [17], and for more

details, we refer the readers to [17,29]. Recall that the equality-constrained optimization problem is as follows:

$$\begin{aligned} \text{argmin}_\theta \quad & -\log[p(\mathbf{Y}|\mathbf{X},\theta)] \quad \text{such that} \\ & c - 2s(x_c^{(i)}) \leq y^*(x_c^{(i)}) \leq c + 2s(x_c^{(i)}) \quad \text{for all} \quad i = 1, 2, ..., m. \end{aligned} \tag{8}$$

The goal here is to show that the solution obtained using the selected set of input locations converges to the solution obtained using all input locations. This convergence guarantees that the probabilistic approach will eventually result in a model that satisfies the desired conditions. For simplicity, we will show this convergence for the left-hand side of Equation (8), and throughout the proof, we will assume that $\mathcal{D}$ is finite. We can construct the proof for two cases:

Case (i): Suppose that the domain $\mathcal{D}$ is a countable set containing N elements. Select a subset $\mathcal{D}_m \in \mathcal{D}$ with m points, where $x_c^{(1)}, x_c^{(2)}, ..., x_c^{(m)} \in \mathcal{D}_m$. We have a Gaussian probability distribution for each $x \in \mathcal{D}$. The set of distributions for $x \in \mathcal{D}$ is denoted by $\mathcal{P}$, while $\mathcal{P}_m$ represents the set of distributions for constraint points $x \in \mathcal{D}_m$. Now, define a set $H(x)$ as

$$H(x) := \{\theta | p(\mathbf{Y}|\mathbf{X},\theta) < 0\}, \tag{9}$$

which identifies regions where the constraint is violated. For a fixed $x \in \mathcal{D}$, let the following apply:

$$\begin{aligned} v(x) &:= \inf_{P\in\mathcal{P}} P(\mathbf{Y}|\mathbf{X},\theta) < 0 \equiv \inf_{P\in\mathcal{P}} P(H(x)), \quad \text{and} \\ v_m(x) &:= \inf_{P\in\mathcal{P}_m} P(\mathbf{Y}|\mathbf{X},\theta) < 0 \equiv \inf_{P\in\mathcal{P}_m} P(H(x)). \end{aligned} \tag{10}$$

Case (ii): Suppose that the domain $\mathcal{D}$ is a finite but uncountable set. Since $\mathcal{D}$ is finite, the set $\mathcal{D} \cup \{x\}$ is also finite. Then, construct a countable subset $\tilde{\mathcal{D}}$ with $x \in \tilde{\mathcal{D}}$. The sets $\mathcal{P}, \mathcal{P}_m$ and $H(x)$ are defined similarly to Equations (9) and (10). To establish convergence, a convergence of v_m over v as $\mathcal{P}_m$ converges to $\mathcal{P}$, and we will use the following metrics [29]:

1. Distance from a point to set:

$$d(x, A) := \inf_{x'\in A} ||x - x'||. \tag{11}$$

Equation (11) defines the distance from point x to set A.

2. Distance between compact sets:

$$\mathbb{D}(A, B) := \sup_{x\in A} d(x, B). \tag{12}$$

Equation (12) defines the distance between two compact sets, A and B.

3. Hausdorff distance:

$$\mathbb{H}(A, B) := \max\{\mathbb{D}(A, B), \mathbb{D}(B, A)\}.$$

Equation (3) defines the Hausdorff distance between A and B.

4. Pseudo-metric for probability distributions: Finally, we define a pseudo-metric $\mathbf{d}$ to describe the distance between two probability distributions, P and $\tilde{P}$, as

$$\mathbf{d}(P, \tilde{P}) := \sup_{x\in\mathcal{D}} |P(H(x)) - \tilde{P}(H(x))|, \tag{13}$$

where $\mathcal{D}$ is the domain specified in Section 3.2.

Assumption 1. *Assume the following:*

1. *There exists a weakly compact set* $\tilde{\mathcal{P}}$ *such that* $\mathcal{P} \subset \tilde{\mathcal{P}}$ *and* $\mathcal{P}_m \subset \tilde{\mathcal{P}}$.
2. $\lim_{m \to N} \mathbf{d}(\mathcal{P}, \mathcal{P}_m) = 0$, *with probability 1.*
3. $\lim_{m \to N} \mathbf{d}(\mathcal{P}_m, \mathcal{P}) = 0$, *with probability 1.*

Theorem 1. *Under the assumptions in Assumption 1,* v_m *converges to* v *as* $\mathcal{P}_m$ *converges to* $\mathcal{P}$*, i.e.,*

$$\lim_{m \to N} \sup_{x \in \mathcal{D}} |v_m(x) - v(x)| = 0.$$

Proof. Let $x \in \mathcal{D}$ be fixed. Define:

$$V := \{P(H(x)) : P \in \mathrm{cl}(\mathcal{P})\}, \quad V_m := \{P(H(x)) : P \in \mathrm{cl}(\mathcal{P}_m)\}.$$

The Hausdorff distance between $\mathrm{conv}(V)$ and $\mathrm{conv}(V_m)$ is:

$$\mathbb{H}(\mathrm{conv}(V), \mathrm{conv}(V_m)) = \max\{|b_m - b|, |a_m - a|\},$$

where:

$$a = \inf_{v \in V} v, \quad b = \sup_{v \in V} v, \quad a_m = \inf_{v \in V_m} v, \quad b_m = \sup_{v \in V_m} v.$$

Using the properties of the Hausdorff distance:

$$|v_m(x) - v(x)| \leq \mathbb{H}(V, V_m) \leq \mathbb{H}(\mathcal{P}, \mathcal{P}_m).$$

Taking the supremum over $x \in \mathcal{D}$:

$$\sup_{x \in \mathcal{D}} |v_m(x) - v(x)| \leq \mathbb{H}(\mathcal{P}, \mathcal{P}_m),$$

and as $m \to N$, $\mathbb{H}(\mathcal{P}, \mathcal{P}_m) \to 0$. Thus:

$$\lim_{m \to N} \sup_{x \in \mathcal{D}} |v_m(x) - v(x)| = 0.$$

This concludes the proof for the left-hand side of the constraint in (8). The proof for the right-hand side follows the same approach. □

4. Numerical Examples

In this section, we evaluate the performance of the proposed algorithms on various examples including synthetic and real data. Synthetic examples are generated to demonstrate the efficacy of the proposed method on larger datasets and higher dimensions. In these numerical examples, the time and accuracy performances of the algorithms are assessed while varying the dataset size and noise level in the data. We consider the relative l_2 error between the posterior mean y^* and the true value of the target function $f(x)$ on a set of test points $\mathbf{X}_t = \{x_T^{(i)}\}_{i=1}^{N_t}$ [7]:

$$E = \sqrt{\frac{\sum_{i=1}^{N_t} [y^*(x_T^{(i)}) - f(x_T^{(i)})]^2}{\sum_{i=1}^{N_t} f(x_T^{(i)})^2}}. \tag{14}$$

We solve the log-likelihood minimization problem in Equation (5) in MATLAB. Furthermore, to demonstrate the advantages of QHMC over standard HMC, the proposed method

is also implemented using the traditional HMC procedure. The comparison includes the relative error, posterior variance, and execution time for both the QHMC and HMC algorithms, with the results presented for each version.

The results for both soft-constrained and hard-constrained versions of QHMC are presented for comparison. The constraint-adaptive, hard-constrained version is labeled as QHMCad, while its soft-constrained counterpart is referred to as QHMCsoftad. Similarly, QHMCvar denotes the variance-focused approach, with its soft-constrained version called QHMCsoftvar. The combination of the two methods, incorporating both hard and soft constraints, is represented by QHMCboth and QHMCsoftboth, respectively.

4.1. Poisson Equation in 2D

Consider the following differential equation [30]:

$$\frac{\partial^2}{\partial x_1^2}u(x) + \frac{\partial^2}{\partial x_2^2}u(x) = f(x). \tag{15}$$

We generate a synthetic dataset that includes a single-fidelity collection of noise-free observations for the forcing term $f(x) = -2\pi^2 \sin(\pi x_1)\sin(\pi x_2)$. Additionally, the initial noise-free locations are generated from the exact solution $u(x) = \sin(\pi x_1)\sin(\pi x_2)$. Beginning with this initial training set, we initiate an active learning iteration loop. In each iteration, a new observation is added to the training set based on the selected sampling policy.

Table 1 shows the relative error and posterior variance comparison of all algorithms alongside their HMC counterparts based on an example of a non-noisy data with 50 samples. All the QHMC algorithms outperform their HMC equivalents, with QHMCboth delivering the highest accuracy. Meanwhile, QHMCsoftboth provides a smaller posterior variance, offering the second most accurate results. Figure 2 presents more a comprehensive accuracy comparison of the algorithms with respect to the increasing signal-to-noise ratio (SNR) and data size. The figure indicates that the proposed algorithms can tolerate the noise in the data, especially when the dataset is larger. Although all the algorithms provide a small error in this simple 2D example, QHMCboth and QHMCsoftboth are the best ones, while QHMCvar and QHMCsoftvar are outperformed by QHMCad and QHMCsoftad. However, Figure 3 shows that the variance-based QHMC approach is faster than both the QHMCad and QHMCboth algorithms.

Table 1. Relative error (Err) and posterior variance (Var) comparison of QHMC and standard HMC [31] versions of algorithms on a 2D Poisson example. QHMCboth is the most accurate, while QHMCsoftboth is the second most accurate, with a reduced variance.

Method	**Err ($\times 10^{-4}$)**	**Var ($\times 10^{-4}$)**	**Method**	**Err ($\times 10^{-4}$)**	**Var ($\times 10^{-4}$)**
QHMCad	0.15	0.11	HMCad	0.19	0.14
QHMCsoftad	0.17	0.10	HMCsoftad	0.21	0.12
QHMCvar	0.16	0.09	HMCvar	0.21	0.11
QHMCsoftvar	0.19	0.08	HMCsoftvar	0.22	0.11
QHMCboth	0.14	0.09	HMCboth	0.17	0.12
QHMCsoftboth	0.15	0.08	HMCsoftboth	0.19	0.11

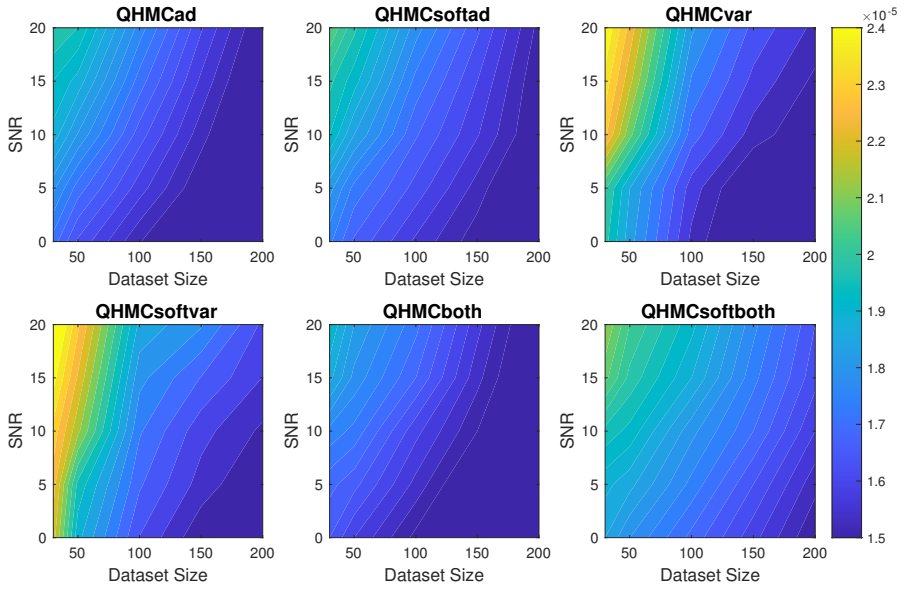


Figure 2. Relative error of the algorithms with increasing SNR and data sizes for a 2D Poisson example. All algorithms achieve high accuracy up to 20% noise, especially when the dataset is larger.

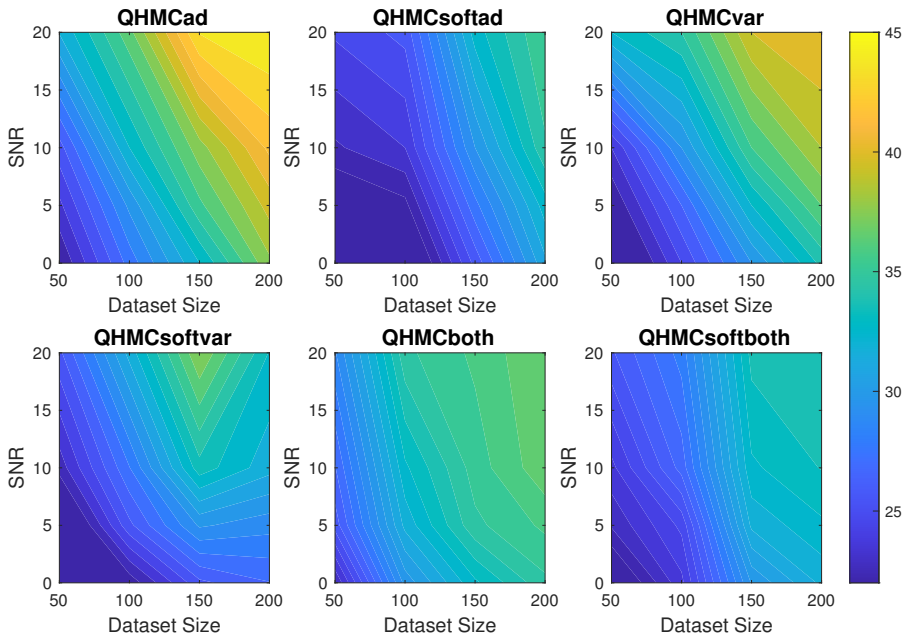


Figure 3. Time comparison of the algorithms with increasing SNR and dataset size for 2D Possion example. Soft-constrained methods can tolerate higher noise and larger datasets, as they are approximately 20% faster than their hard-constrained versions.

4.2. Poisson Equation in 10D

We consider the 10-dimensional Poisson example in [30] given by

$$f_2(x) = -8\pi^2 \sin(2\pi x_1) \sin(2\pi x_3), \tag{16}$$

with the noise term $\{\mathbf{x_2}, \mathbf{y_2}\}$, $\mathbf{y}_2 = f(\mathbf{x}_2) + \epsilon_2$, where $\epsilon_2 \sim N(0, 0.05I)$. The training set contains $n = 80$ observations.

Table 2 presents the error and posterior variance values of the QHMC and HMC versions of all approaches with $n = 80$ observations, where the QHMC approach pro-

vides a significant improvement in accuracy. Specifically, QHMCboth attains the highest accuracy, which is around 98%, and QHMCsoftvar achieves the lowest posterior variance. Additionally, QHMCsoftboth demonstrates overall success, achieving 98% accuracy with a smaller variance compared to QHMCboth. Figure 4 shows the change in relative error of the algorithms for increasing dataset size and noise. Although the error values fluctuated around 2% for all the methods, the QHMCboth results were the most accurate and robust. Based on the comparison in Figure 4, QHMCboth can tolerate noise levels up to 20% with the smallest error, and it can still provide good accuracy (error is around 0.02). Since this is a 10-dimensional example, the time performances are more important than in the previous example, which are shown in Figure 5. Here, we can clearly observe the time advantage of sthe oft-constrained approaches, especially for QHMCsoftvar and QHMCsoftboth. Considering the accuracy and time efficiency together, QHMCsoftboth provides the most efficient and robust method for especially larger datasets.

Table 2. Relative error (Err) and posterior variance (Posterior Var) comparison of QHMC and HMC versions of algorithms on a 10D Poisson example. QHMCboth is the most accurate, while QHMCsoftboth is the second most accurate with reduced variance.

Method	Err	Posterior Var	Method	Err	Posterior Var
QHMCad	0.022	0.016	HMCad	0.027	0.019
QHMCsoftad	0.025	0.014	HMCsoftad	0.030	0.019
QHMCvar	0.023	0.011	HMCvar	0.032	0.016
QHMCsoftvar	0.026	0.010	HMCsoftvar	0.034	0.015
QHMCboth	0.017	0.012	HMCboth	0.025	0.014
QHMCsoftboth	0.020	0.011	HMCsoftboth	0.029	0.015

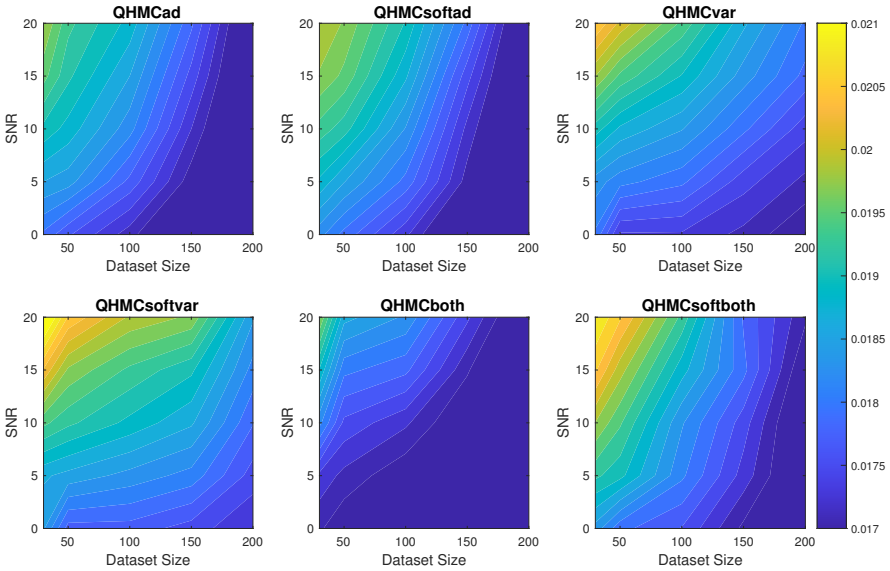


Figure 4. Relative error of the algorithms with different SNR and data sizes for a 10D Poisson example. The algorithms can maintain 98% accuracy up to 20% noise, especially when the dataset is larger.

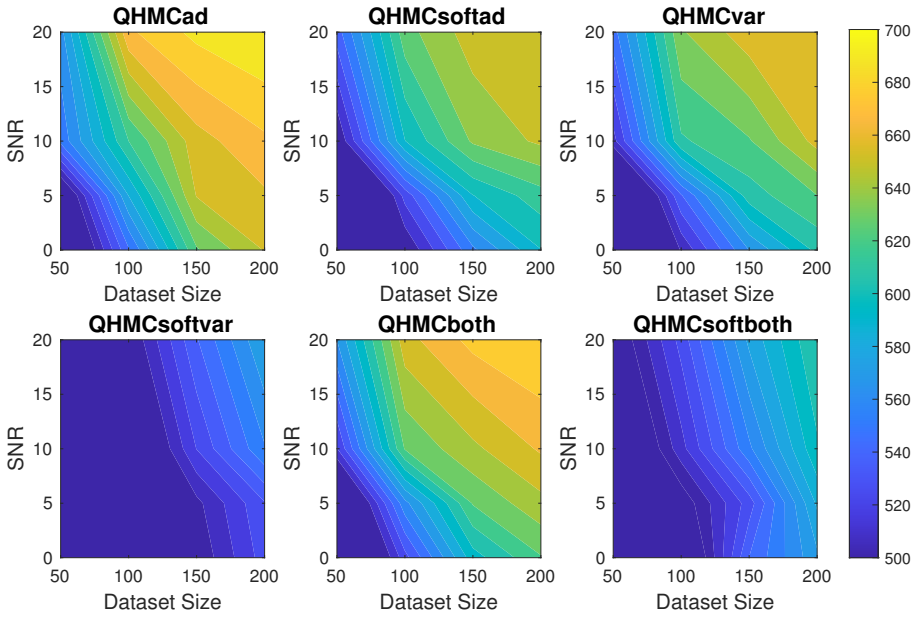


Figure 5. Time comparison of the algorithms with increasing SNR and dataset size for 10D Possion example. Soft-constrained methods can tolerate higher noise and larger datasets, as they are approximately 25% faster than their hard-constrained versions.

4.3. Heat Transfer in a Hallow Sphere

This three-dimensional example considers a heat transfer problem in a hallow sphere. Let $B_r(0)$ represent a ball centered at 0 with radius r. Defining the hallow sphere as $D = B_4(0) - B_2(0)$, the equations are given as follows [32]:

$$\begin{cases} \frac{\partial u(\mathbf{x},t)}{\partial t} - \nabla \cdot (\kappa \nabla u(\mathbf{x},t)) = 0, & \mathbf{x} \in D, \\ \kappa \frac{\partial u(\mathbf{x},t)}{\partial \mathbf{n}} = \theta^2(\pi-\theta)^2\phi^2(\pi-\phi)^2, & \text{if } \|\mathbf{x}|_2 = 4 \text{ and } \phi \geq 0, \\ \kappa \frac{\partial u(\mathbf{x},t)}{\partial \mathbf{n}} = 0, & \text{if } \|\mathbf{x}\|_2 = 4 \text{ and } \phi < 0, \\ u(\mathbf{x},t) = 0, & \text{if } \|\mathbf{x}\|_2 = 2. \end{cases} \tag{17}$$

In this context, $\mathbf{n}$ denotes the normal vector pointing outward, while θ and ϕ represent the azimuthal and elevation angles, respectively, of points within the sphere. We determine the precise heat conductivity using $\kappa = 1.0 + \exp(0.05u)$. Quadratic elements with 12,854 degrees of freedom are employed, and we set $y(\mathbf{x}) = u(\mathbf{x}, 10)$ to solve the partial differential equations (PDEs).

We assume that accurate observation data are available at six locations. We start with these initial locations, then we add six new constraint locations one by one using the adaptive learning approach. Table 3 shows the relative error, posterior variance, and execution time of all types of algorithms, including the HMC versions. For each variant of QHMC and HMC, employing QHMC sampling in a particular version speeds up the process and enhances accuracy. Overall, the comparison reveals that among all the QHMC and HMC variants, QHMCboth delivers the highest accuracy, while QHMCsoftboth and QHMCad rank second in terms of accuracy. We also observe that the soft-constrained counterparts of the algorithms reduce variance. The comparisons among QHMC versions are shown in detail in Figure 6, where a step-by-step decrease shows the effects of the adaptive approaches and the superiority of the QHMCboth and QHMCsoftboth algorithms. In this real data example, we observe the same behavior as in the Poisson examples: QHMCboth is the most accurate, while QHMCsoftboth is the fastest. On the other hand,

QHMCsoftboth ranks second for accuracy; therefore, it should be the most efficient since the error values are close to each other.

Table 3. Relative error (Err) and posterior variance (Var) comparison of QHMC and HMC versions of algorithms on a 3D heat equation example. QHMCboth is the most accurate, while QHMCsoftboth is the second most accurate with reduced variance.

Method	Err	Var	Time	Method	Error	Var	Time
QHMCad	0.031	0.0042	39 s	HMCad	0.036	0.0062	52 s
QHMCsoftad	0.042	0.0040	30 s	HMCsoftad	0.051	0.0059	41 s
QHMCvar	0.032	0.0034	32 s	HMCvar	0.045	0.0044	44 s
QHMCsoftvar	0.042	0.0032	26 s	HMCsoftvar	0.051	0.0043	35 s
QHMCboth	0.027	0.0036	43 s	HMCboth	0.036	0.0040	56 s
QHMCsoftboth	0.031	0.0035	28 s	HMCsoftboth	0.049	0.0039	39 s

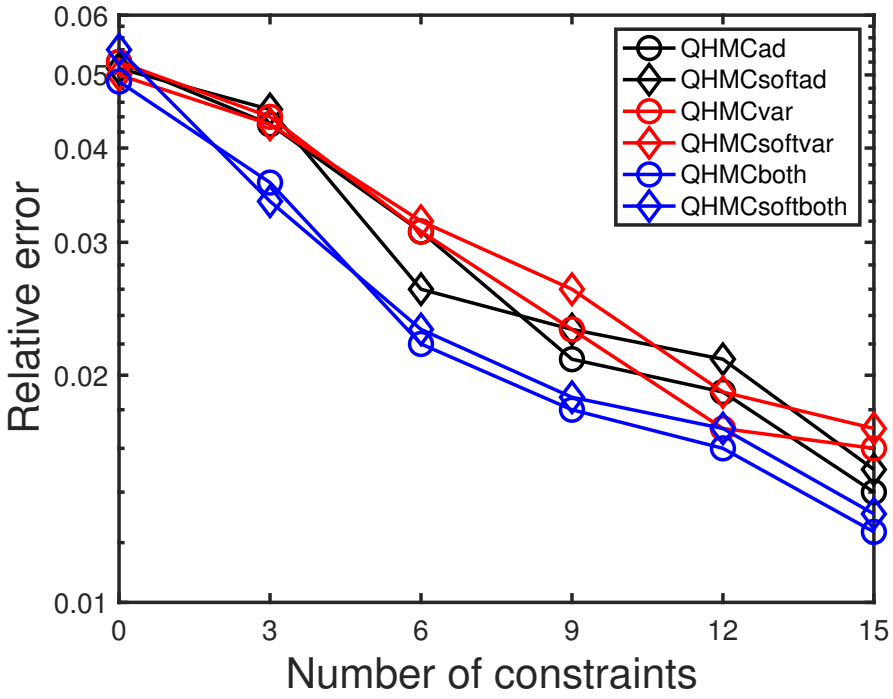


Figure 6. The step-by-step decrease in relative error while adding the constraints adaptively. The smallest error is achieved by QHMCboth.

5. Discussion

Building up on the work in [17], this work evaluates both the advantages and limitations of applying a soft-constrained approach to physics-informed GP regression, focusing on equality constraints. To validate the superiority of this approach, comparisons are also made between modified versions of the proposed algorithm. The key findings and the potential underlying reasons are summarized as follows:

1. To demonstrate the robustness and effectiveness of the proposed methods, synthetic examples are designed with a focus on two main factors: the size of the dataset and the signal-to-noise ratio (SNR). The QHMC-based algorithms were tested across varying levels of SNR, up to 20%. The findings, illustrated in Figures 2 and 4, demonstrate a consistent ability of the method—both in its soft-constrained and hard-constrained forms—to handle noise effectively, especially when the noise remains below 15%. Furthermore, the algorithms demonstrated improved performance with larger datasets,

indicating that increased data size helps mitigate the impacts of noise. These observations from the synthetic examples collectively confirm the robustness of the proposed method under varying conditions.

2. The numerical results for the synthetic examples also provide execution times as the SNR and dataset size increase in each case, aiming to highlight the efficiency of the proposed algorithm. As shown in Figures 3 and 5, the algorithms demonstrate significant time-saving benefits, with the soft-constrained versions exhibiting particularly remarkable advantages. Combined with the results in Figures 2 and 4, where the soft-constrained algorithms maintain high prediction accuracy, the proposed method highlights its capability to provide efficient sampling while preserving performance. We further demonstrated the robustness of the algorithms by including the posterior variance values of the results in Tables 1 and 2.
3. To test the ability of the algorithms to handle higher-dimensional problems while maintaining robustness and efficiency, the evaluations focused on different dimensional settings: 2 dimensions and 10 dimensions. The results confirmed that the proposed methods could consistently deliver accurate results, even as dimensionality increased, and they did so with relatively short execution times. This demonstrates the scalability of the algorithms across a range of problem complexities.
4. To demonstrate the potential of the proposed method to generalize across different types of problems, we selected a real example: a 3D heat transfer problem that requires solutions to partial differential equations. Unlike the synthetic examples, in this part, we had a fixed dataset size that contain no injected Gaussian noise. A thorough comparison of all the methods, including the step-by-step change in the relative error, is presented in Figure 6, validating the success of all the versions. Moreover, as shown in Table 3, the soft-constrained approaches attained approximately 96% accuracy with 35% time efficiency.
5. It is also worth noting that while the numerical results demonstrate the robustness and efficiency of the current QHMC algorithm, further investigation is needed regarding the effects of the probability of constraint violation. Our experiments were conducted with a relatively low constraint release probability (approximately 5%), and accuracy was maintained under these conditions. However, increasing the allowance for violations could introduce limitations.
6. The proposed approach demonstrates strong performance in handling dimensions up to 10, but its scalability to significantly higher dimensions, such as 50 or beyond, has yet to be fully investigated. This question requires systematic study and highlights a valuable direction for extending our work. Previous studies have demonstrated the successful application of the QHMC algorithm to high-dimensional unconstrained problems [26,33], suggesting its potential as a promising method for addressing computational challenges in constrained scenarios.

6. Conclusions

This work combines the accurate training capabilities of QHMC with the efficiency of a probabilistic approach to create a new soft-constrained QHMC algorithm. The algorithm is specifically designed to enforce equality constraints on GP regression. The proposed approach ensures more accurate predictions while also boosting efficiency by delivering successful results in a relatively short time. To further optimize the performance of QHMC algorithms across different problem settings, this work introduces several modified versions of QHMC that incorporate adaptive learning techniques. These adaptations allow the algorithm to dynamically adjust its behavior, improving convergence speed and robustness under varying conditions. By implementing these adaptive versions, the approach provides

greater flexibility, enabling users to choose the most appropriate algorithm based on the specific characteristics and priorities of the problem at hand, such as balancing accuracy, computational time, or the handling of noisy data. This tailored approach broadens the potential applications of the algorithms and ensures more consistent results across a wide range of constrained optimization tasks.

After theoretically justifying their convergence, the methods are implemented to solve a variety of problems, demonstrating their versatility and effectiveness. In each experiment, we started by demonstrating the benefits of QHMC sampling compared to traditional HMC sampling, with QHMC consistently outperforming HMC. Across all tested cases, QHMC sampling delivered approximately a 20% reduction in computation time and a 15% improvement in accuracy. Building on these initial findings, we conducted a more extensive evaluation of the performance of the algorithms across different scenarios. The examples included higher-dimensional problems, illustrating the capability of the algorithms in tackling complex tasks. Additionally, we tested the methods on a real-world application where we solve a PDE. These evaluations highlight the adaptability of the proposed approach and confirm its effectiveness in a broad range of practical and theoretical settings. Overall, this work has shown that the soft-constrained QHMC method is a robust, efficient, and adaptable approach that is capable of handling high-dimensional problems and large datasets. The numerical results indicate that the soft-constrained QHMC has strong potential for generalization across a wide range of applications, including those with varying physical properties.

Author Contributions: Methodology, D.K. and X.Y.; Software, D.K.; Formal analysis, D.K. and X.Y.; Writing—original draft, D.K.; Writing—review & editing, D.K. and X.Y.; Supervision, X.Y.; Funding acquisition, X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NSF CAREER DMS-2143915.

Data Availability Statement: The data analyzed in this study is subject to the following licenses/restrictions: Datasets are from author's one of the previous papers. Requests to access these datasets should be directed to D.K., dik318@lehigh.edu.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GP	Gaussian process
MCMC	Markov Chain Monte Carlo
MH	Metropolis-Hastings
HMC	Hamiltonian Monte Carlo
QHMC	Quantum-inspired Hamiltonian Monte Carlo
HMCad	Hard-constrained Hamiltonian Monte Carlo with adaptivity
HMCsoftad	Soft-constrained Hamiltonian Monte Carlo with adaptivity
HMCvar	Hard-constrained Hamiltonian Monte Carlo with variance
HMCsoftvar	Soft-constrained Hamiltonian Monte Carlo with variance
HMCboth	Hard-constrained Hamiltonian Monte Carlo with both adaptivity and variance
HMCsofboth	Soft-constrained Hamiltonian Monte Carlo with both adaptivity and variance
QHMCad	Hard-constrained Quantum-inspired Hamiltonian Monte Carlo with adaptivity
QHMCsoftad	Soft-constrained Quantum-inspired Hamiltonian Monte Carlo with adaptivity
QHMCvar	Hard-constrained Quantum-inspired Hamiltonian Monte Carlo with variance
QHMCsoftvar	Soft-constrained Quantum-inspired Hamiltonian Monte Carlo with variance
QHMCboth	Hard-constrained Quantum-inspired Hamiltonian Monte Carlo, adaptivity and variance

QHMCsofboth	Soft-constrained Quantum-inspired Hamiltonian Monte Carlo, adaptivity and variance
SNR	Signal-to-noise ratio
PDE	Partial differential equations

References

1. Lange-Hegermann, M. Linearly constrained Gaussian processes with boundary conditions. In Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, San Diego, CA, USA, 13–15 April 2021; pp. 1090–1098.
2. Kuss, M.; Rasmussen, C. Gaussian processes in reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2003**, *16*, 751–759.
3. Liu, T.; Wei, H.; Liu, S.; Zhang, K. Industrial time series forecasting based on improved Gaussian process regression. *Soft Comput.* **2020**, *24*, 15853–15869. [CrossRef]
4. Jin, B.; Xu, X. Machine learning WTI crude oil price predictions. *J. Int. Commer. Econ. Policy* **2024**. [CrossRef]
5. Zhang, H.; Liu, J. Jointly stochastic fully symmetric interpolatory rules and local approximation for scalable Gaussian process regression. *Pattern Recognit.* **2025**, *159*, 111125. [CrossRef]
6. Rasmussen, C.E.; Williams, C.K. *Gaussian Processes for Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 1.
7. Pensoneault, A.; Yang, X.; Zhu, X. Nonnegativity-enforced Gaussian process regression. *Theor. Appl. Mech. Lett.* **2020**, *10*, 182–187. [CrossRef]
8. Swiler, L.P.; Gulian, M.; Frankel, A.L.; Safta, C.; Jakeman, J.D. A survey of constrained Gaussian process regression: Approaches and implementation challenges. *J. Mach. Learn. Model. Comput.* **2020**, 1–2. [CrossRef]
9. Maatouk, H.; Bay, X. Gaussian process emulators for computer experiments with inequality constraints. *Math. Geosci.* **2017**, *49*, 557–582. [CrossRef]
10. Abrahamsen, P.; Benth, F.E. Kriging with inequality constraints. *Math. Geol.* **2001**, *33*, 719–744. [CrossRef]
11. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Machine learning of linear differential equations using Gaussian processes. *J. Comput. Phys.* **2017**, *348*, 683–693. [CrossRef]
12. Salzmann, M.; Urtasun, R. Implicitly Constrained Gaussian Process Regression for Monocular Non-Rigid Pose Estimation. Available online: https://papers.nips.cc/paper_files/paper/2010/hash/115f89503138416a242f40fb7d7f338e-Abstract.html (accessed on 3 January 2025).
13. Agrell, C. Gaussian processes with linear operator inequality constraints. *arXiv* **2019**, arXiv:1901.03134.
14. Da Veiga, S.; Marrel, A. Gaussian process modeling with inequality constraints. *Ann. Fac. Des Sci. Toulouse MathéMatiques* **2012**, *21*, 529–555. [CrossRef]
15. Maatouk, H.; Roustant, O.; Richet, Y. Cross-validation estimations of hyper-parameters of Gaussian processes with inequality constraints. *Procedia Environ. Sci.* **2015**, *27*, 38–44. [CrossRef]
16. López-Lopera, A.F.; Bachoc, F.; Durrande, N.; Roustant, O. Finite-dimensional Gaussian approximation with linear inequality constraints. *SIAM/ASA J. Uncertain. Quantif.* **2018**, *6*, 1224–1255. [CrossRef]
17. Kochan, D.; Yang, X. Gaussian Process Regression with Soft Inequality and Monotonicity Constraints. *arXiv* **2024**, arXiv:2404.02873.
18. López-Lopera, A.; Bachoc, F.; Roustant, O. High-dimensional additive Gaussian processes under monotonicity constraints. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 8041–8053.
19. Narcowich, F.J.; Ward, J.D. Generalized Hermite interpolation via matrix-valued conditionally positive definite functions. *Math. Comput.* **1994**, *63*, 661–687. [CrossRef]
20. Jidling, C.; Wahlström, N.; Wills, A.; Schön, T.B. Linearly Constrained Gaussian Processes. Available online: https://papers.nips.cc/paper_files/paper/2018/hash/68b1fbe7f16e4ae3024973f12f3cb313-Abstract.html (accessed on 3 January 2025)
21. Albert, C.G.; Rath, K. Gaussian process regression for data fulfilling linear differential equations with localized sources. *Entropy* **2020**, *22*, 152. [CrossRef]
22. Ezati, M.; Esmaeilbeigi, M.; Kamandi, A. Novel approaches for hyper-parameter tuning of physics-informed Gaussian processes: Application to parametric PDEs. *Eng. Comput.* **2024**, *40*, 3175–3194. [CrossRef]
23. Stein, M.L. Asymptotically efficient prediction of a random field with a misspecified covariance function. *Ann. Stat.* **1988**, *16*, 55–63. [CrossRef]
24. Zhang, H. Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *J. Am. Stat. Assoc.* **2004**, *99*, 250–261. [CrossRef]
25. Barbu, A.; Zhu, S.C. *Monte Carlo Methods*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 35.
26. Liu, Z.; Zhang, Z. Quantum-inspired Hamiltonian Monte Carlo for Bayesian sampling. *arXiv* **2019**, arXiv:1912.01937.
27. Jensen, B.S.; Nielsen, J.B.; Larsen, J. Bounded Gaussian process regression. In Proceedings of the 2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Southampton, UK, 22–25 September 2013; pp. 1–6.
28. Gelman, A.; Carlin, J.B.; Stern, H.S.; Dunson, D.B.; Vehtari, A.; Rubin, D.B. *Bayesian Data Analysis*; Tyler & Francis Group, Inc.: New York, NY, USA, 2014.

29. Guo, S.; Xu, H.; Zhang, L. Stability analysis for mathematical programs with distributionally robust chance constraint. *SIAM J. Optim.* **2015**. Available online: https://api.semanticscholar.org/CorpusID:16378663 (accessed on 15 January 2025).
30. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Inferring solutions of differential equations using noisy multi-fidelity data. *J. Comput. Phys.* **2017**, *335*, 736–746. [CrossRef]
31. Vishnoi, N.K. An introduction to Hamiltonian Monte Carlo method for sampling. *arXiv* **2021**, arXiv:2108.12107.
32. Yang, X.; Tartakovsky, G.; Tartakovsky, A.M. Physics information aided kriging using stochastic simulation models. *SIAM J. Sci. Comput.* **2021**, *43*, A3862–A3891. [CrossRef]
33. Kochan, D.; Zhang, Z.; Yang, X. A Quantum-Inspired Hamiltonian Monte Carlo Method for Missing Data Imputation. In Proceedings of the Mathematical and Scientific Machine Learning, PMLR, Beijing, China, 15–17 August 2022; pp. 17–32.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.