PROBABILISTIC NUMERICAL INTEGRATION WITH APPLICATIONS IN MACHINE LEARNING

by

Kelly Hirschbeck Smalenberger

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Applied Mathematics

Charlotte

2020

ABSTRACT

KELLY HIRSCHBECK SMALENBERGER. PROBABILISTIC NUMERICAL INTEGRATION WITH APPLICATIONS IN MACHINE LEARNING. (Under the direction of DR. XINGJIE (HELEN) LI)

In this dissertation, we consider the approximation of unknown or intractable integrals using quadrature, especially when the evaluation of these integrals is considered very costly. This is a central problem both within and without machine learning, including model averaging, (hyper-)parameter marginalization, and computing posterior predictive distributions.

Recently Batch Bayesian Quadrature has successfully combined the probabilistic integration techniques of Bayesian Quadrature with the parallelization techniques of Batch Bayesian Optimization, resulting in improved performance when compared to state-of-the-art Markov Chain Monte Carlo techniques, especially when parallelization is increased. While the selection of batches in Batch Bayesian Quadrature mitigates costs associated with individual point selection, every point within every batch is nevertheless chosen serially, which impedes the realization of the full potential of batch selection. We resolve this shortcoming.

We have developed a novel Batch Bayesian Quadrature method which allows us to update points within a batch without incurring the costs traditionally associated with non-serial point selection. To implement this, we also devise a novel dynamic domain decomposition. Combining these, we show that this efficiently reduces uncertainty, leads to lower error estimates of the integrand, and therefore results in more numerically robust estimates of the integral. Furthermore, we close an open question in the Batch Bayesian Quadrature literature about the cessation criterion, which we determine and support using numerical methods on commonly used test functions in one and two dimensions.

We present our findings within the context of the history of quadrature, show how our novel methods significantly improve what currently exists in the literature, and provide recommendations how where improvements can be made in the future.

ACKNOWLEDGMENTS

There are many people who deserve acknowledgement and thanks for helping me throughout this academic journey.

First, I would like to genuinely thank my advisor Dr. Helen Li. Without her understanding, support, and depth of knowledge, this would not have been possible. The flexibility she has provided me to work an learn at my own pace, and yet has always made herself available when I would contact her, made me feel like I had the independence and support that facilitated my growth into an independent producer of knowledge. It has been an honor to work with Dr. Li, the insights I have gained from her have been invaluable to my research and my position as a professor.

Additionally, I would like to thank Dr. Hae-Soo Oh, Dr. Duan Chen, and Dr. Milind Khire for being on my dissertation committee. Without their flexibility, kindness, and support during this challenging process I would not have been able to reach this goal. Their help is extremely appreciated. I would also like to thank Dr. Deng for his prompt replies to any inquiry and support which shows that he truly cares about the success of graduate students and the graduate program. He has worked diligently to grow the graduate program, which can be seen by the placements graduate students have had during his tenure.

I would also like to acknowledge Dr. Tanya Cofer and Dr. Stephanie Levi for their encouragement and support throughout my undergraduate studies. They provided me with the opportunities necessary to pursue my graduate studies. It has always been a goal of mine to provide that same support for my students.

I would also like to thank my family and friends. My parents have always worked hard for their family and their children to ensure that we have every opportunity available to us. My parents, friends, and siblings have always been beyond supportive and loving. Especially my mother who is a true example of strength, honesty, and compassion.

I would also like to thank my children, Jane, Betty, and the baby which I am currently carrying. They have brought me joy, motivation, tons of distraction, and the ability to laugh at anything. Though I know you are likely too small to remember this time, I thank you for your patience in being born while I have been in graduate school and working full time. I hope that I can help you see the joys of learning and encourage your life's pursuits.

Lastly, I would like to thank my husband Michael whose kindness, support, persistence, and love have helped me reach and complete each milestone of my graduate and professional journey. You are my rock and I am forever grateful.

TABLE OF CONTENTS

INTRODUCTION

In mathematics, the term 'quadrature' can trace its origins back to the Pythagorean doctrine of ancient Greece, and has as its primary objective the determination of area. Specifically, ancient Greek mathematicians understood the calculation of the area of a figure as a process of geometrically constructing a square (squaring) or a set of rectangles which when summed have the same area. Therefore, the creation of these rectangles eventually morphologically evolved into the term quadrature, and provided the foundation for the development of calculus and specifically integration.

While advances in integration can be traced back to ancient Greece, it wasn't until Gottfried Wilhelm Leibnitz and Sir Isaac Newton independently formulated the integral as an infinite sum of rectangles of infinitesimal width that we obtain the definition used in calculus. Therefore, an integral assigns a numeric value to functions which can be used to describe displacement, area, volume, or other concepts that arise by combining infinitesimal data.

Despite the fact that integration provides modern practitioners with a powerful and valuable tool which has applications in a seemingly endless list of fields and topics, it nevertheless has its challenges. For example, suppose it is desired to determine the signed area of the region bounded by a function, f(x), and a second limiting constraint, say the x-axis. This task would be futile using basic calculus integration if function, f(x), governing the area to be evaluated is unknown or analytically intractable. In these scenarios, quadrature still has an important application.

Modern quadrature techniques are predicated on the selection of points used to partition a function. Much consideration has been given in the literature on how to most effectively and efficiently select these quadrature points, and several techniques have been proposed. A common theme throughout these techniques is to select quadrature points sequentially. Even the most recent developments in the quadrature literature, termed Batch Bayesian Quadrature where a batch of quadrature points are selected, select points within each batch sequentially. An oft employed justification for the sequential selection of quadrature points is that non-sequentially selecting points would increase the dimensionality of the problem significantly and therefore the computational cost of the quadrature technique in the integration process would be unjustifiable at best, or insurmountable at worst. Furthermore, when the integrand is considered costly to evaluate, the selection of a large number of points is untenable.

Beginning with this notion as our premise, it is here that we make our most important contribution to the literature. We show that computationally simplistic modifications can be done to update the selection of quadrature points, which in turn leads to better estimates of the integrand, and hence more robust integral estimates.

In order to provide a precise definition of the problem, a holistic exploration of the solutions already proposed in the literature, and thorough and convincing justification for the superiority of our approach to what already exists in the literature, this document proceeds as follows: In chapter 1 we describe past accomplishments in quadrature and use this as prelude to our novel and significant contributions. In chapter 2 we highlight the most recent advances in the literature in order to contrast the results of those with our own. In chapter 3 we begin with a thorough explanation of our novel method, along with why our technique is superior to those which already exists. We continue this chapter by implementing our technique to well-known test functions, and convey numerical results which make this superiority explicit in practice. We also make explicit other novel contributions which allow for this implementation, including a novel form of dynamic domain decomposition. Furthermore, this chapter also answers open questions in the quadrature literature on batch Bayesian quadrature, such as numerically founded and supported cessation criterion. This chapter concludes with an overview of applications of our novel contributions on the cutting edge of machine learning, and describe potential expansions in future work. This document concludes with a conclusion section, followed by a list of references and an appendix containing data from our numerical applications.

CHAPTER 1: PAST AS PRELUDE

The goal is to compute integrals such as

$$\int_{\Omega} f(\omega)p(\omega)d\Omega$$

where $\omega \in \Omega$, $\int_{\Omega} p(\omega)d\Omega = 1$, and without loss of generality $f(\omega) \geq 0$. So in the one dimensional case we have

$$Z = \int_{a}^{b} f(x)p(x)dx \tag{1.1}$$

If the product $f(x)p(x)$ are easy functions, such as polynomials, then the integral I will be easy to compute and the answer will be obtained in closed form using methods taught in calculus II. However in practice, the product $f(x)p(x)$ is rarely an easy function and f and p are most often unrelated. Therefore, numerical techniques must be employed in order to approximate Z.

Because the integral Z is the product of two functions $f(x)$ and $p(x)$, it can also be seen as an estimate of the expectation of the function. Where $p(x)$ is the probability density function or pdf and $f(x)$ is the function one would like to integrate. However, using this integration for Bayesian models, $p(x)$ is the posterior distribution and $f(x)$ is the prediction made by a model (Rasmussen et al., 2003).

1.1 History of Quadrature

A student of Galileo, Bonaventura Cavelieri (1598-1647) was one of the first to determine the area beneath a curve of the form $y = x^k$ where k is a positive integer.

Cavalieri's process was arithmetized by John Wallis (1616-1703) in the book Arithmetics Infinitorum. The approach Wallis used involved exploring the limit-sum of the kth powers of the first n positive integers (Burton, 2011). For example when approximating $y = x^k$ Wallis determined the area was equal to $lim_{n\to\infty}\frac{0^k+1^k+2^k+....+n^k}{n^k+n^k+n^k+...+n^k}$. In particular this lead him to conclude that $\int_0^a x^k dx = \frac{a^{k+1}}{k+1}$ which is the power rule for integrations taught in mahy calculus II classes. Wallis imagined the area beneath a curve as being made up of n infinitely narrow vertical rectangles each of width $\frac{b-a}{n}$. Though Wallis was able to derive the classic power rule for integration, this idea of approximating the area under a curve by narrower and narrower rectangles is the first appearance of classic quadrature formulas (Burton, 2011).

The use of randomness in a deterministic manner can be traced back to the 18th century. A French scientist named Georges Louis LeClerc used randomness in a number of studies including "Baffon's needle" in which he attempted to estimate π. This is considered, by some, to be the first use of Monte Carlo simulations, though the use of this term was not used until much later (Harrison, 2010).

The term Monte Carlo was first used by John Von Neumann, Stanislaw Ulam, and Nicholas Metropolis to refer to stochastic simulations. The three were working at Los Alamos National Laboratory on the Manhattan Project during World War II. Von Neumann and Ulam were attempting to model possible outcomes from detonation of the atomic bomb. Because of the implications, their model of what would happen in a chain reaction caused by highly enriched uranium needed to be have little error. Because of the complexity of the reaction, von Neumann, Ulam, and Metropolis needed to use numerical methods instead of algebraic methods which were typically used at the time.

The difficulty they found was that the problem had so many dimensions and variables that because computational methods and programs were underdeveloped at the time, calculations were far too time consuming. Thus the Monte Carlo method of using random numbers was developed. Because this method used random numbers and chance, Ulam suggested naming it the Monte Carlo method after the gambling complex located in Monaco frequented by his uncle. Calculations were done as many times as possible and basic statistics were then used in order to model with high accuracy the possible outcomes of the chain reactions of uranium. Because of the high accuracy of the Monte Carlo method, it was used to help guide the design of the atomic bomb.

Now Monte Carlo is used for a multitude of numerical simulation methods. This includes stochastic simulations, Monte Carlo Tests, and as will be used discussed in this dissertation, Monte Carlo Integration, as defined by Anderson (1999) "Monte Carlo is the art of approximating an expectation hy the sample mean of a function of simulated random variables."

According to Karvonen et al. (2017), Bayesian Monte Carlo or Bayesian Quadrature dates back to at least the 1970's with the work of Larking. Larking proposed a new method for numerical approximation which contrasted the classical approach of constructing low degree polynomials to best fit the function which cannot be integrated analytically.

1.2 Classical Quadrature Methods

For simplicity, in this chapter we will let $f(x)p(x) = m(x)$ and our integral from the introduction will become

$$Z = \int_a^b f(x)p(x)dx = \int_a^b m(x)dx$$

There are multiple well known classical numerical methods for approximating an integral. Quadrature formulas refer to the numerical approximation of Z to a finite summation $\sum_{i=1}^{n} w_i m(x_i)$ where x_i are points in $[a, b]$, and w_i are weights.

A few well know quadrature methods include the midpoint formula, the trapezoidal formula, Simpson's formula, and Gauss Quadrature. Note that in order to use these formulas, m must be continuous on the interval $[a, b]$. The interval $[a, b]$ is then divided into subintervals $Z_k = [x_{k-1}, Z_k]$, where $x_k = a + kH$, $H = (b - a)/M$ is the length of each subinterval, and $k = 1, 2, ..., M$.

Additionally, the function m must be approximated on each subinterval Z_k. Typically m is interpolated using one or many evaluation points in Z_k in order to construct a polynomial. Polynomials are frequently used as the interpolating function because they are easy to integrate. This new function $\tilde{m}$ is chosen slightly differently based the quadrature formula used. For midpoint, trapezoidal, and Simpson's Formula, the interval $[a, b]$ has been divided into subintervals of equal length. This strongly affects the weights in each of the following quadrature formulas (Quarteroni et al., 2010). It is also important to note that accuracy of quadrature methods increase as the number of evaluation points increase or as the length of the sub-intervals Z_k decreases.

Though the following quadrature methods work well in most cases, they are not easily extended when integrating in higher dimensions.

1.2.1 Midpoint Quadrature Formula

To employ the midpoint quadrature formula, a point $\bar{x}_k$ in each subinterval $[x_{k-1}, x_k]$ must be chosen such that $\bar{x}_k$ is the midpoint of that subinterval. In other words $\bar{x}_k = \frac{x_{k-1} - x_k}{2}$. $\tilde{m}$ is chosen as polynomial interpolation if m on Z_k. Then the midpoint quadrature formula can be easily seen as

$$Z_{mp} = H \sum_{k=1}^{M} \tilde{m}(\bar{x}_k)$$

This formula has also been referred to as the composite rectangle quadrature formula when $\tilde{m}$ is chosen to be the constant polynomial approximation of m at $\bar{x}_k$ because of its graphical representation.

The midpoint quadrature formula has second order accuracy with respect to H. That is, if m is in $C^2([a,b])$ then

$$Z - Z_{mp} = \frac{b-a}{24} H^2 m''(\xi) \tag{1.2}$$

where ξ is a point in $[a,b]$.

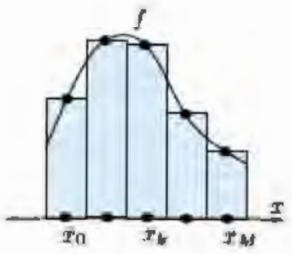


Figure 1.1: Example of the midpoint rule (Chorin et al., 2013)

1.2.2 Trapezoidal Quadrature Formula

To use the trapezoidal quadrature formula, $\tilde{m}$ must be chosen as the linear polynomial interpolation of m at the end points of each subinterval. Then the trapezoidal quadrature formula becomes

$$Z_t = \frac{H}{2} \sum_{k=1}^{M} [\tilde{m}(x_{k-1}) - \tilde{m}(x_k)] \tag{1.3}$$

The trapezoidal midpoint quadrature formula has second order accuracy with respect to H. That is, if m is in $C^2([a,b])$ then

$$Z - Z_t = -\frac{b-a}{12} H^2 m''(\xi) \tag{1.4}$$

where ξ is a point in $[a,b]$.

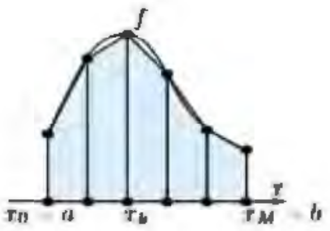


Figure 1.2: Example of the trapezoidal rule (Chorin et al., 2013)

1.2.3 Simpson's Quadrature Formula

To use Simpson's quadrature formula, $\tilde{m}$ must be chosen as the quadratic polynomial interpolation of m at the end points x_{k-1}, and x_k, as well as at the midpoint $\bar{x}_k$ of each subinterval Z_k. That is

$$\tilde{m}(x) = \frac{2(x-\bar{x}_k)(x-x_k)}{H^2}m(x_{k-1}) + \frac{(x_{k-1}-x)(x-x_k)}{H^2}m(\bar{x}_k) + \frac{(x-\bar{x}_k)(x_k - xk - 1)}{H^2}m(x_k)$$

Then the Simpson's quadrature formula becomes

$$Z_S = \frac{H}{6}\sum_{k=1}^{M}[\tilde{m}(x_{k-1}) + 4\tilde{m}(\bar{x}_k) + \tilde{m}(x_k)] \tag{1.5}$$

This formula has fourth order accuracy with respect to H. That is, if m is in $C^4([a,b])$ then

$$Z - Z_S = -\frac{b-a}{180}\frac{H^4}{16}m^{(4)}(\xi)$$

where ξ is a point in $[a,b]$.

1.2.4 General Quadrature Formula

Note that the above three formulas were described by partitioning the interval $[a,b]$ into equally distributed sub-intervals Z_k of length H. However, this may not always be the most prudent choice. There are a multitude of ways different choose x_i and/or ω_i in order to approximate Z. One way is to vary ω_i based on the variability of the range of m. The more a function varies in the range, the more the domain should be partitioned. Similarly, less variability in the range would indicate fewer partitions being required with little effect on accuracy. This partitioning based on a functions variability in the range would create subintervals of varying length but would allow for increased efficiency and accuracy when modeling m. The general quadrature formula is given by

$$Z = \sum_{i=0}^{n} \omega_i m(x_i) \tag{1.6}$$

where ω_i are the quadrature weights, or lengths of the sub-intervals, and x_i are the quadrature nodes, or the points which will be used to try to model the function m. It is also important to note that $\sum_{i=0}^{n} \omega_1 = b - a$.

One type of general quadrature rule is Gauss quadrature. Gauss quadrature is calculated using n evaluation points x_i and corresponding weights ω_i on the interval $[-1, 1]$. These points are the roots of the Legendre polynomial $P_n(x)$ of order n. Weights are calculated by computing

$$\omega_i = \frac{2}{(1 - x_i^2)(P_n'(x_i))^2}$$

Because of the way in which x_i and ω_i are determined, Gauss quadrature provides the exact integral if m is a polynomial of degree $2n - 1$ (Weisstein, 2018B).

Though Gauss quadrature is typically computed on the interval $[-1, 1]$, it can be easily transformed to the interval $[a, b]$. Additionally, the weights can be transformed to the interval $[a, b]$. The formulas are as follows;

$$x_{Ti} = a + \frac{b-a}{2}(1 + x_i) \qquad\qquad \omega_{Ti} = \frac{b-a}{2}\omega_i$$

Additionally, Gauss quadrature is easily adaptable to higher dimensions and non-uniform partitioning of the domain (Deng, 2015).

Another type of general quadrature rule is Gauss-Hermite quadrature. Gauss-Hermite quadrature differs from Gauss quadrature in that it is integrals of the form $\int_{-\infty}^{\infty} e^{-x^2} f(x)dx$. Gauss-Hermite quadrature is calculated using n evaluation points x_i and corresponding weights ω_i on the interval $(-\infty, \infty)$. These points are the roots of the Hermite polynomial $H_n(x)$ of order n. Weights are calculated (Weisstein, 2018B) by computing

$$\omega_i = \frac{2^{n-1} n 1 \sqrt{\pi}}{n^2 (H_{n-1}(x_i))^2}$$

Gauss-Hermite quadrature is worth mentioning because of the numerical simulations computed in this work which have a similar form, with weighting function is $W(x) = e^{-x^2}$ and $f(x) = 1 + \sin x$.

1.2.5 Gauss Quadrature Points and Weights

Below are the Gauss Quadrature point x_i and weights ω_i on the interval $[-1, 1]$.

Gauss Quadrature Points and Weights		
number of points n	points x_i	weights ω_i
1	0	2
2	$\pm\sqrt{\frac{1}{3}}$	1
3	0 $\pm\sqrt{\frac{3}{5}}$	$\frac{8}{9}$ $\frac{5}{9}$
4	$\pm\sqrt{\frac{3}{7}-\frac{2}{7}\sqrt{\frac{6}{5}}}$ $\pm\sqrt{\frac{3}{7}+\frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{18+\sqrt{30}}{36}$ $\frac{18-\sqrt{30}}{36}$
5	0 $\pm\frac{1}{3}\sqrt{5-2\sqrt{\frac{10}{7}}}$ $\pm\frac{1}{3}\sqrt{5+2\sqrt{\frac{10}{7}}}$	$\frac{128}{225}$ $\frac{322+13\sqrt{70}}{900}$ $\frac{322-13\sqrt{70}}{900}$

1.2.6 Numerical Simulations

Note that the following calculations were computed for the function $\frac{1}{\sqrt{2\pi}}(1+\sin x)e^{\frac{-x^2}{2}}$ on the interval $[0,1]$ with 100 sub-intervals. For the computation of Gauss quadrature, three Gauss points were found on each sub-interval.

Evaluation of Quadrature Formulas				
	Midpoint Formula	Trapezoidal Formula	Simpson's Formula	Gaussian Quadrature
# of Points	201	101	201	401
Error	$8.065802e-06$	$1.613167e-05$	$2.307243e-11$	$4.896639e-13$

The information provided in this table shows how the accuracy of each of the methods. Simpsons Formula and Gaussian Quadrature perform significantly better than the midpoint and trapezoidal formulas.

1.3 Modern Quadrature Methods

1.3.1 Monte Carlo Method

Monte Carlo methods are a class of methods which can be applied to two different types of problems involving statistical inference. Those are problems involving optimization and problems involving integration which are typically associated with a Bayesian approach. Monte Carlo methods used to numerically approximate integrals are known for their ease of use, fitness for use in higher dimensions, and because of its use of the Law of Large Numbers (Wasserman, 2004).

We would like to apply the Monte Carlo method to integrals of the form $Z = \int_0^1 f(x)p(x)dx$, where $p(x)$ is the probability density function and f can be sampled at arbitrary points. More specifically $f(x) \geq 0$ and $\int_a^b p(x)dx = 1$, then $Z = E[f(\eta)]$. Where η is a random variable with a pdf $p(x)$. That is, Monte Carlo method can be used to evaluate a non-random quantity as the expected value of a random variable. Moreover, if η is sampled so that there are n independent experiments with outcomes $\eta_1, \eta_2, ..., \eta_n$, then from the Chebyshev inequality, one can make the following approximation;

$$Z = E[f(\eta)] \sim \frac{1}{n}\sum_{i=1}^{n} f(\eta_i) \tag{1.7}$$

The error that occurs from this approximation is of order $\sigma(f(\eta))/\sqrt{n}$, where $\sigma(f(\eta))$ is the standard deviation of $f(\eta)$ (Chorin et al., 2013). Additionally, it is worth pointing out that this approximation uses the Law of Large Numbers.

Theorem 1.1. *(Law of Large Numbers) Let $\xi_1, \xi_2, ...$ be an independent identically distributed random variable. If $E[f(\xi_1] < \infty$ then*

$$lim_{n->\infty}\frac{1}{n}\sum_{i=1}^{n}\xi_i = E[f(\xi_1)]$$

This is according to Wasserman Wasserman (2004). The Law of Large Numbers basically says that after many trials, the average of the results should be close to the expected value and will be more accurate with more trials.

The drawbacks of the Monte Carlo approximation include that that it typically converges very slowly. So the sample of the random variable η must be large. Additionally, because each sample point is weighted equally, a functional value for random numbers that are sampled very close together are essentially counted as double. Additionally, the more the function f fluctuates, the slower the Monte Carlo approximation will converge.

1.3.2 Monte Carlo Method with Importance Sampling

The error that occurs with the basic Monte Carlo Method can be reduced in the following ways; increasing the sample size, and reducing the variance of $f(\eta)$. In order to reduce the variance, importance sampling can be done. Additionally, the Monte Carlo method requires the function p to be sampled, but this is not always optimal. An alternative to directly sampling p is importance sampling.

A function $h(x)$ must be found which is similar to $p(x)$ in order to sample from h and make computations simpler. According to Chorin et al. (2013), h must be found with the following properties;

1. $Z_1 = \int_a^b g(x)h(x)dx$ can be computed
2. $h(x) \geq 0$
3. A random variable with pdf $g(x)h(x)/Z_1$ can be sampled easily
4. $f(x)/h(x)$ varies little

Then according to Chorin et al. (2013),

$$\begin{aligned} Z = \int_a^b f(x)p(x)dx = \int_a^b \frac{p(x)}{h(x)} f(x)h(x)dx = Z_1 \int_a^b \frac{p(x)}{h(x)} \frac{p(x)h(x)}{Z_1} dx \\ = Z_1 E[\frac{p}{h}(\eta)] \approx \frac{Z_1}{n} \sum_{i=1}^{n} \frac{p(\eta_i)}{h(\eta_i)} \qquad (1.8) \end{aligned}$$

where η has pdf $f(x)h(x)/Z_1$. Since the last requirement dictates that $p(x)/h(x)$ varies little, the subsequent error will be smaller. Additionally, importance sampling gets its name from the new random variable that places more points where f is large or "important" (Chorin et al., 2013).

1.3.3 Numerical Simulations

A problem similar to that in Chorin et al. (2013) is best used to explain this method and the subsequent computations. Suppose $Z = \int_{-10}^{10} \sin(x/7)e^{-x/3}dx$. This can be done by applying the basic Monte Carlo method. That would mean sampling a uniform random variable ξ n times. ξ is a random number generated on the interval $[-10, 10]$. Then the Monte Carlo approximating for Z is given by

$$Z \approx \sum_{i=1}^{n} \frac{1}{n} \sin \xi_i/7e^{-\xi_i/3}$$

Alternatively, the Monte Carlo method with importance sampling can be done. The first step is to let $Z_1 = \int_{-10}^{10} e^{-x/3}dx = 3(e^{10/3} - e^{-10/3})$. Then $Z = \int_{-10}^{10} \sin(x/7)e^{-x/3}dx = Z_1 \int_{-10}^{10} \sin(x/7)\frac{e^{-x/3}}{Z_1}dx$. Let η be a random variable with pdf

$$f(x) = \begin{cases} \frac{e^{-x/3}}{Z_1} & \text{if } 0 \leq x \leq 1 \\ 0 & \text{else} \end{cases}$$

then Z can be written as $Z = Z_1E[\sin \eta/7]$. As can be seen graphically, the function $\sin x/7$ has smaller variation in the range of integration $[-10, 10]$ than the previous integrand. In order to perform the Monte Carlo integration, the random variable η needs to be sampled n times. This can be done by solving the equation $\int_{-10}^{\eta} e^{-x/3}/Z_1$, where where η is a random variable consisting of random numbers generated on the interval $[-10, 10]$. A straight forward calculation gives $\eta = -3ln(\xi - 3e^{10/3}) - 3ln(\frac{-Z_1}{3})$. This can be used to sample η n times. The resulting Monte Carlo approximation with importance sampling becomes

$$Z \approx \frac{Z_1}{n} \sum_{i=1}^{n} \sin \eta_i / 7$$

Additionally, because the sample generated for were random numbers each computation of the Monte Carlo method gives a different result. Therefore, the Monte Carlo method was calculated 100 times. Those calculations were then used to calculate the average error.

Evaluation of Error for Quadrature Formulas			
# of Points	100	$1,000$	$10,000$
Basic MC	0.101572	$1.002346e{-}02$	$1.001022e{-}03$
MC with Importance sampling	...	...	...

Note that the following calculations were computed for the function $\frac{1}{\sqrt{2\pi}}(1 + \sin x)e^{-\frac{x^2}{2}}$ on the interval $[0, 1]$. Due to the slow nature of convergence of the Monte Carlo method and the Monte Carlo method with importance sampling, two different cases were computed for each method.

Computations for the basic Monte Carlo Method were done as follows; a random number η_i was generated on the interval $[0, 1]$. η was sampled 100 and $10,000$ times. Then the Monte Carlo approximation was calculated as

$$Z = E[f(\eta)] \sim \frac{1}{n} \sum_{i=1}^{n} (1 + \sin(\eta_i))e^{-\frac{\eta_i^2}{2}}$$

However, because the sample generated for η were random numbers each computation of the Monte Carlo method gives a different result. Therefore, the Monte Carlo method was calculated 100 times. Those calculations were then used to calculate the average error.

Computations for Monte Carlo method with importance sampling were not able to be done in the method described by (1.8) because of the inability to integrate $e^{-x^2/2}$ on the interval $[0,1]$ in closed form. Therefore, it is essential to recognize that the function $p(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$ is the Gaussian or normal random variable with mean, $\mu = 0$, and variance, $\sigma^2 = 1$. So for importance sampling, a normal random variable η was constructed and truncated to the interval $[-10, 10]$ which was sampled 100 and $10,000$ times. Then the Monte Carlo with importance sampling approximations was calculated to be

$$Z = E[f(\eta)] \approx \frac{1}{n}\sum_{i=1}^{n}(1 + \sin x)$$

In Monte Carlo with importance sampling, computing the truncated Gaussian random variable also depended on random numbers. Therefore, this method was calculated 100 times. Those calculations were then used to calculate the average error.

Evaluation of Error for Quadrature Formulas			
# of Points	100	$1,000$	$10,000$
Basic MC	0.21966	0.06545	0.01986
MC with Importance sampling	$5.11073 - 02$	$1.73823e - 02$	$5.10225e - 03$

The information provided in this table shows how the accuracy of each of the methods as well as he rate of convergence. Clearly the Monte Carlo method with importance sampling out performs the basic Monte Carlo method. Additionally, the basic Monte Carlo method converge significantly more slowly than Monte Carlo with importance sampling.

1.4 Probabilistic Integration

$$Z = \int_a^b f(x)p(x)dx$$

Bayesian Monte Carlo or Bayesian quadrature is known for treating the function $f(x)$ as a random variable with the goal of reducing the variance of f, and for its use of Bayes' Theorem. Because Bayesian Monte Carlo uses Bayes' theorem, updating is done and a much smaller sample is needed. This in turn saves computational abilities.

Since this section is about Bayesian Monte Carlo, it is worth noting Bayes Theorem.

Theorem 1.2. *(Bayes Theorem) Let $A_l, ..., A_k$ be a partition of Ω such that $P(Ai) > 0$ for each i. If $P(B) > 0$ then, for each $i = 1, ..., k$,*

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)}$$

This is according to Wasserman Wasserman (2004).

The proof of Bayes Theorem comes directly from the application of the definition of conditional probability. That is, if $P(B) > 0$ then the conditional probability of A given B is $P(A|B) = \frac{P(A \cap B)}{P(B)}$.

Here $P(A_i)$ is the prior prohability distrihution of A. The prior is often the original probability or an educated guess. However, once an experiment is done or data is obtained the prior is improved and the posterior is obtained. Here $P(A_i|B)$ the conditional probability or posterior probability of A which improved the prior based on data.

Additionally, Bayesian Monte Carlo uses Gaussian process. A Gaussian process, written as $GP(m(x), k(x, x'))$, is a collection of random variahles, any finite number of which have a joint Gaussian distribution. It is completely described by its mean, $m(x)$ and covariance function $k(x, x')$. The mean and covariance functions of a real process $f(x)$ are defined as; $m(x) = \mathbb{E}[f(x)]$, and $k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$ respectively.

A general name for a function k of two arguments that maps $x, x' \in X \subset \mathbb{R} \to \mathbb{R}$

is a kernel. Kernels act like filters in a Gaussian process and determine different levels of smoothness. The kernel is a covariance function that controls the properties of a Gaussian process. Possibly the most commonly used kernel function is the squared exponential kernel function which is defined as $\text{cov}_f(x, x') = k(x, x') = \exp(-\frac{|x-x'|^2}{2l^2})$ where l is the characteristic length. Another widely used kernel function is the dot product kernel function which is defined as $\text{cov}_f(x, x') = k(x, x') = \sigma_0^2 + x^T \Sigma x'$ where Σ is a general covariance matrix.

1.4.1 Bayesian Quadrature

The Bayesian approach to the Monte Carlo Method is a probabilistic approach to numerically computing the integral $I = \int_a^b f(x)p(x)dx$. As with Bayesian inference, one is able to make better choices or more precise computations based on different states of knowledge or updating of data.

In Bayesian Monte Carlo one must think of I as being random. This way of thinking is consistent with a Bayesian approach which views uncertainty through probabilistic representation, states of knowledge, and expectations.

As described by Huszar et al. (2012) Bayesian Monte Carlo or BMC begins by placing a prior probability distribution on f, written as $p(f)$ though a Gaussian Process. For ease of calculations, let this prior have kernel function k and mean 0. Then I is estimated by combining the prior on f with conditioned observations on f at the sample points $D = (x^{(i)}, f(x^{(i)})|i = 1, 2, ...$ BMC allows for these sample points to be selected as desired. The result is a posterior probability distribution $p(f|D)$ which is also Gaussian since integration is a linear projection.

The posterior expectation of f from the Gaussian Process allows for the following computation ;

$$E_{f|D}[I] = E_{f|D}(\int f(x)p(x)dx)$$

$$= \int \int f(x)p(x)dxp(f|D)df$$
$$= \int \int f(x)p(f|D)dfp(x)dx$$
$$= \int \bar{f}_D(x)p(x)dx \tag{1.9}$$

where $\bar{f}_D$ is the posterior mean function. But the standard result from the Gaussian Process gives

$$\bar{f}_D(x) = k(x, \mathbf{x})K^{-1}\mathbf{f}$$

where $\mathbf{x}$ and $\mathbf{f}$ are vectors containing the observed inputs and corresponding functional values. k is the kernel function, or the form of the pdf which omits factors which are not functions of any variables in the domain. And $K = k(x^{(i)}, x^{(i)})$ for $i = 1, 2, ...n$ is the covariant matrix where $K_{i,j} = \text{cov}(x_i, x_j)$. One of the benefits to BMC is that any distribution can be sampled as long as $p(x)$ can be evaluated (Rasmussen et al., 2003). However, if $p(x)$ is Gaussian, more specifically, if $p(x)$ has mean b and variance B and the Gaussian kernel on the data points are $N(x_i, A = \text{diag}(\omega_1^2, ...\omega_M^2))$ then

$$E_{f|D}[I] = z^T K^{-1}\mathbf{f} \tag{1.10}$$

where

$$z = \omega_0|A^{-1}B + I|^{-1/2}\text{exp}[-0.5(a-b)^T(A+B)^{-1}(a-b)] \tag{1.11}$$

Note that ω_0 and a are scalars that come from the Gaussian Process. This formulation of the expectation of I is a linear combination of observed functional values of $f(x)$. Similarly as with all quadrature methods, this estimations can be viewed as a sum of weights $w^{(i)} = \sum_m z_j^T K_{im}^{-1}$ and points. That is

$$E_{f|D}[I] = \sum_i w^{(i)}\mathbf{f}^{(i)} \tag{1.12}$$

1.4.2 Optimal Sampling

As mentioned above, we must find the value of f at certain values of x. But in order to not was computational abilities during this computation, x must be chosen in a specific way. To determine the sample $x^{(i)}$ at which to evaluate f,Huszar et al. (2012) recommend trying to minimize the posterior variance. The more the posterior variance can be minimized, the fewer the number of sample points will be required because minimizing the posterior variance also minimize the error. Therefore, it is essential to express the posterior variance of I conditioned on $f(x^{(i)})$ in closed form. That is

$$\begin{aligned} V_{f|D}[I] &= V_{f|D}[\int f(x)p(x)dx] \\ &= \int [\int f(x)p(x)dx - \int \bar{f}(x')p(x')dx']^2 p(f|D)df \\ &= \int\int\int [f(x)-\bar{f}(x)][f(x')-\bar{f}(x')]p(f|D)df\, p(x)p(x')dxdx' \\ &= \int\int \mathrm{Cov}_D(f(x),f(x'))p(x)p(x')dxdx' \qquad (1.13) \end{aligned}$$

where the standard results for the Gaussian Process gives

$$\mathrm{Cov}_D(f(x),f(x')) = k(x,x') - k(x,\mathbf{x})K^{-1}k(\mathbf{x},x')$$

with $\mathbf{x}$ and $\mathbf{f}$ as defined above. Similarly if $p(x)$ is Gaussian with mean b and variance B and the Gaussian kernel on the data points are $N(x_i, A = \mathrm{diag}(\omega_1^2, ...\omega_M^2))$ then

$$V_{f|D}[I] = \omega_0|2A^{-1}B + I|^{-1/2} - z^T K^{-1} z \qquad (1.14)$$

where z as defined in (1.11). It is worth noting that this posterior variance does not depend on $f(x^{(i)})$, but only depends on $x^{(i)}$(Huszar et al., 2012). So rewriting $V_{f|D}[I]$ as $V_{f|x^{(i)}}[I]$ is also helpful.

The sample is constructed by adding one piece, x, to the sample $x_1, x_2...x_n$ at a time. This is referred to as Sequential Bayesian Quadrature.

$$\begin{aligned} x^{(n+1)} &= \text{argmin} V_{f|D}[I] \\ &= \text{argmin} V_{f|(x_1,x_2 \ldots x_n,x)}[I] \end{aligned} \tag{1.15}$$

Since Bayesian Monte Carlo uses a Gaussian process which ensures a type of smoothness, points in sparsely sampled areas are much more informative than points in highly sample regions. As noted in Huszar et al. (2012) there are some trade-offs between accuracy and the diversity of the sample. They point out that when the sampling close to high density regions under p, this results in a decrease in the variance. Also, when the distance between the samples decreases, this results in an increase in the variance.

1.4.3 Numerical Simulations

Note that the following calculations were computed for the function $\frac{1}{\sqrt{2\pi}}(1+\sin x)e^{\frac{-x^2}{2}}$ on the interval $[-10, 10]$ with a different number of sub-intervals. Similar to previous calculations, the basic Monte Carlo or MC method was computed 100 times because it depends on a random number generator. Those calculations were then used to calculate the average error.

Error Calculated			
	Monte Carlo	MC with importance sampling	Bayesian MC
Error	$2.19259e-04$	$2.38755e-05$	$8.49210e-05$
# of points	$100,000,000$	100	10

The information provided in this table shows how the accuracy of each of the methods. Clearly the Bayesian Monte Carlo method performs significantly better than the basic Monte Carlo method.

CHAPTER 2: Batch Bayesian Quadrature

Similar to Bayesian Quadrature (BQ) in a previous chapter, Batch Bayesian Quadrature (BBQ) also concerns itself with integrals of the form

$$Z = \int f(x)p(x)dx \tag{2.1}$$

where both $f(x)$ (e.g., a likelihood) and $p(x)$ (e.g., a prior) are non-negative. These integrals are usually intractable, and we must therefore turn to approximation to solve them.

Until recently, point selection strategies in BQ have been sequential in nature, and are therefore collectively referred to as Sequential Bayesian Quadrature (SBQ). That is, a single quadrature point (x_1) is selected, the model is updated, and a subsequent quadrature point (x_2) is selected based on the updated model (Rasmussen et al., 2003; Huszar et al., 2012). Therefore, it is important to note that the cost associated with SBQ has two primary sources, namely 1) updating the model, and 2) choosing the optimal quadrature points. We will take these in turn.

2.1 Model Updating

Since point sampling and evaluation are expected to be costly, sequentially selecting points and updating the model is considered to be greedy. That is, the least number of iterations of sampling and evaluation are undertaken in order to achieve a desired result. BQ provides not only a mean estimate of the integral, but a full Gaussian posterior distribution. The variance of this distribution quantifies the uncertainty in the integral estimate. Therefore, when selecting quadrature points, a frequently used stopping criterion for SBQ is the posterior variance of the integral estimate

$$V[Z|f(x_s)] = \int\int k(x,x')p(x)p(x')dxdx' - z^T K^{-1} z, \tag{2.2}$$

as was illustrated in a previous chapter. To this end, a predetermined posterior variance is chosen, and once the integral estimate's posterior variance reaches this threshold, sampling and updating cease. A generally accepted posterior variance threshold is $1.0x10^{-5}$ (Garnett et al., 2010).

Insights from the field of Bayesian Optimization (BO) have shown that sequential model updating may not be efficient. In BO, so-called batch methods have been proposed as an effective way of allowing parallelization of model updating, and are called Batch Bayesian Optimization (BBO) (Ginsbourger et al., 2010; Gonzáles et al., 2016; Nguyen et al., 2017). In BBO, a batch of samples is taken and its points are used to simultaneously update the model. Scenarios where BBO has been shown to be practically efficient are in alloy quality testing and machine learning, among many others (Nguyen et al., 2017). For example, in order to test alloy quality, a sample of the alloy must be placed in a large oven where it is subjected to high temperatures for an extensive period. In practice this process is costly, and hence many alloy samples are placed in an oven in order to evaluate a batch simultaneously. Similarly, in machine learning various algorithms are developed in order to test their efficiency in obtaining a desired result. Decision nodes in these processes can be run in parallel at the same time using multiple cores.

Due to the similarities between BQ and BO, ideas from BBO lend themselves well to BQ. BO uses a probabilistic model to guide exploration of a search space by defining an acquisition function to be maximized. In BBO, so-called batch methods have been proposed as an effective way of allowing parallelization of this exploration. Once a batch of samples has been chosen, the model can be updated using the samples simultaneously and therefore reduce the cost associated with multiple iterations of the model updating process. In this regard, SBQ is inefficient when parallel evaluations are possible.

2.2 Point Selection

The ideas of batch selection in BBO have recently been applied to Bayesian Quadrature, and are called Batch Bayesian Quadrature (BBQ) (Wagstaff, 2018A). In these batch methods, rather than selecting one point to sample at each step, a batch of points of size n are selected. The entire batch is then used to update the model before which a subsequent batch could be selected.

As was illustrated in the previous chapter, the posterior variance of the integral estimate shown in Equation 6.2 does not depend on the function values, $f(x)$, only on the location of the the quadrature points. This allows the optimal samples in BQ to be computed ahead of observing any values of the integrand (Huszar et al., 2012; Minka, 2000; Osborne et al., 2012A). Therefore, in BBQ, one could theoretically predetermine quadrature points, select a batch of size n of these points, and use these to update the model simultaneously. However, choosing quadrature points irrespective of the functional form of $f(x)$ has obvious shortcomings. To this end, recent developments in BBQ have devised more sophisticated batch selection techniques.

2.2.1 Posterior of Integral Estimate versus Posterior of Gaussian Process

BQ puts a prior distribution on f, then estimates integral (1) by inferring a posterior distribution over the function f, conditioned on the observations $f(x_s)$ at some query points x_s. The posterior distribution over f then implies a distribution over Z. This method allows us to choose sample locations x_s in any desired manner (Huszar et al., 2012). That is, we may choose to not predetermine quadrature points to minimize the posterior variance of the integral estimate Z, and instead take into account the function values $f(x_s)$ and maximize some acquisition function of our choosing. One such acquisition function could have the objective of minimizing the posterior variance of the Gaussian Process on f.

An important advantage of Bayesian Quadrature is that the posterior variance of the Gaussian Process (GP) can be used to model uncertainty of the integrand, and therefore encourage the exploration of the sample space more efficiently. We will discuss later how it is possible to develop more sophisticated models where, for example, the covariance of better-informed posteriors relies on the sampled quadrature points, among other benefits (Wagstaff, 2018A).

It is crucially important to note that, as referenced earlier in regards to BBO efficiency in alloy testing and machine learning, the simultaneous evaluation of a batch of points in updating the model is one primary source of potentially significant cost. It is, however, not the only source of potentially substantial cost. Specifically, how the batch of points are selected can also incur substantial cost. It is in this regard that we make a significant contribution to the existing literature.

2.2.2 Sequential Batch Points Selection Methods

One way a set of n points could be selected per batch is to naively select the n highest points on the acquisition function. However, this has the obvious downfall that the points selected in this manner will all be nearly the same as to the maximum of the acquisition function, leading to a potentially uninformative batch. That is, the gain in information of the entire batch entails more than merely the sum of the information of each separate point within the batch. Put differently, is that we require an informative batch, and hence we must select a batch of diverse points (Wagstaff, 2018A).

To find the optimal batch we could theoretically define an acquisition function over all sets of n points and maximize it. In practice this is too costly since it increases the dimensionality of the problem by a factor of n. This would lead to the acquisition function very costly to compute. One way past research in the literature has side-stepped this problem is to use heuristics where each quadrature point is selected sequentially. Subsequently the acquisition function updated after a point is identified and selected to be included in the batch (Wagstaff, 2018A). Two common methods for the sequential selection of batch points are the so-called Kriging Believer strategy (Ginsbourger et al., 2010; Wagstaff, 2018A), and Local Penalisation (Gonzáles et al., 2016; Wagstaff, 2018A). These will be discussed in turn.

Kriging Believer

In utilizing the Kriging Believer batch selection method, we select a point which maximizes an acquisition function, set it's value equal to the posterior mean. Subsequently the GP model is updated to reflect the new information. We select the subsequent point based on the posterior of our updated GP. This process is repeated until a batch of the requisite size has been gathered. Only then do we use our selected batch points to evaluate the integrand and use these observations and evaluations to update our model. This strategy allows for two things, namely 1) it naturally forces further exploration of the space as the region immediately around each point will no longer be informative, and 2) it allows for the selection of an informative batch while only once incurring the cost of evaluating the integrand in order to update our model (Wagstaff, 2018A; Ginsbourger et al., 2010).

Local Penalisation

Another strategy that has been adopted from BO in selecting batch points is the so-called Local Penalisation (Gonzáles et al., 2016; Wagstaff, 2018A). This strategy directly penalizes the acquisition function around a chosen point. This leads to not using the entire model or space, as in Kriging Believer, but instead lowering the acquisition function around each selected point, which in turn encourages diversity in the batch. Similar to Kriging Believer, once the batch points have been selected and the model updated, the penalization begins anew. Therefore, as before with Kriging Believer, this strategy also allows for the selection of an informative batch while only once incurring the cost of evaluating the integrand in order to update our model (Wagstaff, 2018A).

2.3 Point Selection with Batch Updating

As previously stated, the justification for the sequential selection of batch points in Kriging Believer and Local Penalisation - as opposed to defining an acquisition function over all sets of n points (e.g., by considering the expected reduction in variance) and maximizing it - is that in practice it is less expensive since otherwise the dimensionality of the problem would increase by a factor of n making the acquisition function very costly to compute (Wagstaff, 2018A). However, this argument has obvious flaws. It is in the mitigation of these flaws that we improve on the existing literature.

Since a posterior on f implies a posterior on Z, in Kriging Believer and Local Penalisation alike, an acquisition function is maximized that seeks to minimize the posterior variance of the GP. The flaw in the argument for sequentially selecting batch points is that the only way to achieve a more optimal set of points would increase the dimensionality of the problem, and therefore cost, significantly. As we will show, this is not true.

2.3.1 Stationarity of GP Posterior Mean

Note that in Kriging Believer and Local Penalisation alike, the posterior mean of the GP is stationary. That is, since Kriging Believer is a form of simple kriging, when a quadrature point (x_i) is "sampled", the functional value associated with this point prior to selecting a subsequent quadrature point (x_j), where $i < j$, is the GP posterior mean at (x_1). This allows the GP posterior mean to remain unchanged. Similarly, in Local Penalisation, the acquisition function is penalized as previously discussed. This, again, allows the GP posterior mean to remain stationary.

2.3.2 Symmetry of GP Posterior Variance between Training Points

As discussed in a previous chapter, the noiseless GP posterior covariance matrix is given by

$$k(X_2, X_2) - k(X_2, X_1)k(X_1, X_1)^{-1}k(X_1, X_2), \tag{2.3}$$

where (X_1) is the training set, (X_2) is the test set, k is the kernel function, and the posterior variance at the test set conditioned on the training set is the diagonal of this covariance matrix. Recall that analytic results for $Z = \int f(x)p(x)dx$ can be obtained from certain combinations of kernel functions and priors $p(x)$ (Rasmussen et al., 2003). One such combination is a Gaussian prior and the Squared Exponential (SE) kernel. We continue with that premise here.

First, note that the Gaussian prior naturally encourages the selection of quadrature points near the center of the distribution. Furthermore, the SE kernel is given by

$$k(x, x') = \sigma^2 \exp(-\frac{(x - x')^2}{2l^2}), \tag{2.4}$$

where σ^2 is a scale factor, l is the lengthscale, and x and x' may be vectors. It is important to note that since the value of the SE decreases with distance and ranges between 0 (in the limit) and 1 ($x = x'$), it can be interpreted as a similarity measure.

This summary of batch Bayesian quadrature will now serve as the cornerstone upon which we build our novel method in the next chapter.

CHAPTER 3: Batch Bayesian Quadrature with Selection Updating

Problems frequently encountered in machine learning call for the approximation of intractable integrals of the form

$$Z = \int l(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} \tag{3.1}$$

where both $l(\mathbf{x})$ (e.g., a likelihood) and $\pi(\mathbf{x})$ (e.g., a prior) are non-negative. Various approaches exist to complete this task, including Laplace approximation, variational inference, and Markov Chain Monte Carlo (MCMC). However, all of these approximation methods have their drawbacks (Blei et al., 2016; O'Hagan, 1987) and are unsuitable where evaluating the desired likelihood is expensive, such as when estimates based on only a few evaluations must be made.

Bayesian Quadrature (BQ), on the other hand, considers the approximation of intractable integrals as a problem of inference from limited data to which probability theory can be applied. Additionally, Gunter et al. (2014) and Wagstaff (2018A) have shown that BQ techniques achieve faster convergence and smaller absolute errors when compared to state of the art MCMC methods.

3.1 Warping

Bayesian Quadrature utilizes a probabilistic model to induce both the functional form of the integrand and a probability distribution over the value of the integral. Gaussian Process (GP) is a commonly used method for placing probability distributions over functions (Diaconis, 1988; O'Hagan, 1987; Minka, 2000; Rasmussen et al., 2003; Wagstaff, 2018A), and we will use it in our probabilistic model.

Given $Z = \int l(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$, we choose to place the GP on the likelihood, l, though we could place it on the entire integrand. The GP is parameterized by a mean $\mu(x)$ and a scaled Gaussian covariance $K(x, x') = \lambda^2 \exp(-\frac{1}{2}\frac{(x-x')^2}{\sigma^2})$ where the output length-scale λ and input length-scale σ control the standard deviation of the output and the autocorrelation range of each function evaluation, respectively, and will be jointly denoted by $\theta = \{\lambda, \sigma\}$. Conditioned on samples $x_d = \{x_1, ..., x_N\}$ and corresponding functional values $l(x_d)$, our GP is given by

$$l \mid D \sim GP(m_D, C_D) \tag{3.2a}$$

$$m_D(x) = \mu(x) + K(x, x_d)K^{-1}(x_d, x_d)(l(x_d) - \mu(x_d)) \tag{3.2b}$$

$$C_D(x, x') = K(x, x') - K(x, x_d)K^{-1}(x_d, x_d)K(x_d, x') \tag{3.2c}$$

where $D = \{x_d, l(x_d), \theta\}$. This leads to a Gaussian distribution of the value of Z since GPs are closed under affine transformations (Rasmussen et al., 2006). The mean and variance of Z are given by

$$\mathbb{E}_{l|D}[Z] = \int m_D(x)\pi(x)dx \tag{3.3a}$$

$$\mathbb{V}_{l|D}[Z] = \int\int C_D(x, x')\pi(x)\pi(x')dxdx' \tag{3.3b}$$

These are analytic given our K and when $\pi(x)$ is Gaussian, which we use here (Briol et al., 2015). Other combinations of K and $\pi(x)$ also lead to analytic results (Gunter et al., 2014).

Since we place the GP on l, utilizing a standard GP prior would ignore the range and non-negativity of l leading to pathologies (Rasmussen et al., 2003). To address this, several prior works have investigated warping the output space of the GP (Osborne et al., 2012B; Gunter et al., 2014; Chai et al., 2018). That is, instead of modeling l as a GP, a transformed likelihood $g(l)$ is modeled, where for example $g = \log(x)$ or $g = \sqrt{x}$, such that $g^{-1}(g(l))$ is non-negative. While this leads to a posterior on l that is not a GP, it is possible to derive a GP which closely approximates it. To that end, we follow the method of Gunter et al. (2014) termed WSABI-L, which makes use of the square-root transformation $g = \sqrt{x}$.

Specifically, we define $\tilde{l}(x) = \sqrt{2(l(x) - \alpha)}$, where α is a small scalar. Prior investigations found that the performance was insensitive to the choice of this parameter and used $\alpha = 0.8$ x min $l(x_d)$ (Gunter et al., 2014), which we also implement here.

We take a GP prior on $\tilde{l}(x) : \tilde{l}(x) \sim GP(0, K)$, for which the posterior is

$$p(\tilde{l} \mid D) = GP(\tilde{l};\, \tilde{m}_D(\cdot),\, \tilde{C}_D(\cdot,\cdot)) \tag{3.4}$$

$$\tilde{m}_D(x) = K(x, x_d)K^{-1}(x_d, x_d)\tilde{l}(x_d) \tag{3.5}$$

$$\tilde{C}_D(x, x') = K(x, x') - K(x, x_d)K^{-1}(x_d, x_d)K(x_d, x') \tag{3.6}$$

However, with this transformation we arrive at a GP whose marginal distribution for any $l(x)$ is a non-central χ^2 with one degree of freedom, and hence the posterior of our integral is not closed-form.

Since GPs are closed under linear transformations and given our GP on $\tilde{l}$, a local linearization of the form $f : \tilde{l} \mapsto l = \alpha + \frac{1}{2}\tilde{l}^2$ will give us a GP for l. That is, lineariation around $\tilde{l}_0$ results in $l(x) \simeq f(\tilde{l}_0) + f'(\tilde{l}_0)(\tilde{l} - \tilde{l}_0)$. We choose $\tilde{l} = \tilde{m}_D$ such that

$$l(x) \simeq (\alpha + \tfrac{1}{2}\tilde{m}_D(x)^2) + \tilde{m}_D(x)(\tilde{l}(x) - \tilde{m}_D(x)) \tag{3.7}$$

$$\simeq \alpha - \tfrac{1}{2}\tilde{m}_D(x)^2 + \tilde{m}_D(x)\tilde{l}(x) \tag{3.8}$$

and therefore l is an affine transformation of $\tilde{l}$, which results in the following posterior:

$$p(l \mid D) \simeq GP(l; m_D^L(\cdot), C_D^L(\cdot, \cdot)) \tag{3.9}$$

$$m_D^L(x) = \alpha + \tfrac{1}{2}\tilde{m}_D(x)^2 \tag{3.10}$$

$$C_D^L(x, x') = \tilde{m}_D(x)\tilde{C}_D(x, x')\tilde{m}_D(x') \tag{3.11}$$

Since $\tilde{m}_D$ and $\tilde{C}_D$ are mixtures of un-normalized Gaussians K, m_D^L and C_D^L are also mixtures of un-normalized Gaussians. Therefore, substituting (8) and (9) into (3a) and (3b), respectively, yields closed-form expressions for the mean and variance of Z.

3.2 Active Sampling

One characteristic of the standard BQ model is that the posterior covariance only depends on the locations sampled, not on the functional values at those points (Rasmussen et al., 2003).

However, given the Bayesian model of the likelihood, Bayesian decision theory can be used to define an acquisition function to guide the selection of sample locations, including the reduction in uncertainty about either the integrand or the integral.

One possibility in selecting the next sample location x_* would be to follow Osborne et al. (2012A) in minimizing the expected entropy of the integral by selecting $x_* = \arg\min_x \langle \mathbb{V}_{l|D, l(x)}[Z] \rangle$, where

$$\langle \mathbb{V}_{l|D, l(x)}[Z] \rangle = \int \mathbb{V}_{l|D, l(x)}[Z] N(l(x); m_D(x), C_D(x, x)) \mathrm{d}l(x). \tag{3.12}$$

Another possibility would be what is known as *uncertainty sampling*. Here the reduction of entropy in the integrand is targeted by selecting x_* with the largest uncertainty, i.e., $x_* = \arg\max_x \mathbb{V}_{l|D}[l(x)\pi(x)]$, where for our warped integrand we have

$$\mathbb{V}^L_{l|D}[l(x)\pi(x)] = \pi(x)^2 \tilde{C}_D(x,x)\tilde{m}_D(x)^2. \tag{3.13}$$

It should be noted, as the work by Gunter et al. (2014) correctly stated, that uncertainty sampling reduces the entropy of the GP to $p(l)$ rather than the true intractable distribution, and that the computation of (10) is considerably more expensive than that of (11).

3.2.1 Future Uncertainty Sampling

What we now describe is what we believe to be a novel sampling technique in Bayesian Quadrature which we call *future uncertainty sampling*, and our first contribution with this paper. Uncertainty sampling is a sequential process, i.e., x_* is selected, and only after $x_* \in x_d$ is the subsequent x_* chosen. We will see in the next section that prior work exists where a batch of x_* are chosen before they are included in x_d, but even there each x_* within each batch is chosen sequentially.

Instead of choosing x_* to maximize the reduction in entropy of the integrand as in uncertainty sampling, we instead minimize the uncertainty of the integrand by selecting sample locations x'_* such that

$$\mathbf{x}'_* = \arg\min_{\mathbf{x}_*} \left[\max_x \mathbb{V}_{l|D}[l(x)\pi(x)]\right] \tag{3.14}$$

where again $\mathbb{V}_{l|D}[l(x)\pi(x)]$ is as in equation 3.13. Now x_* is to be understood as any point which has been evaluated as if it had been chosen as a quadrature point. This definition will be used throughout this research when discussing the implementation of our novel technique, and should be distinguished from the quadrature point of choice when using uncertainty sampling.

While future uncertainty sampling can be implemented as a sequential sampling technique, it also allows for multiple sample locations to be chosen simultaneously. A common argument against the simultaneous selection of a set of n points is that it increases the dimensionality of the selection problem by a factor of n, and therefore in practice becomes too costly to compute.

A mitigating factor in quadrature is that locations close to sampled points become uninformative. To that end, we can decompose the domain and perform a discretized search over sample locations without incurring the usual costs of non-serial point selection. We discuss our implementation of this domain decomposition in section 5.

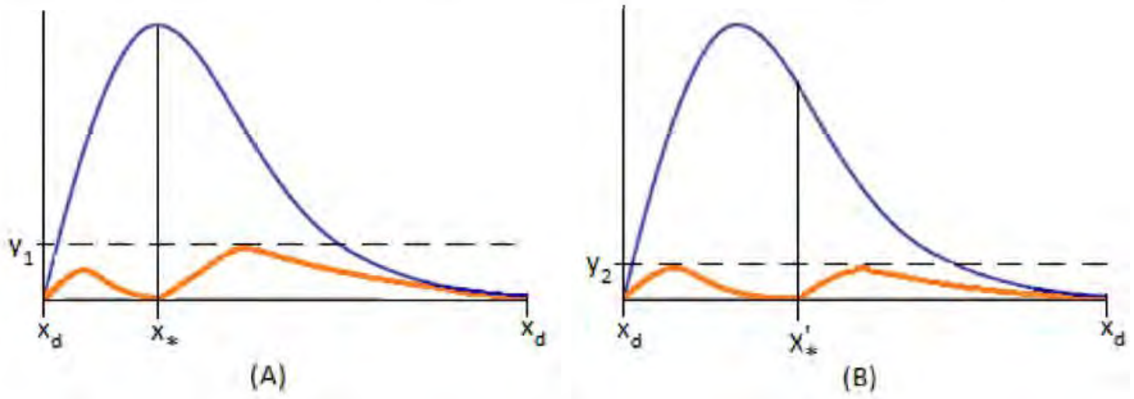


Figure 3.3: Warped Gaussian process posterior covariance

Uncertainty sampling targets the reduction of entropy in the integrand by selecting x_* with the largest uncertainty, i.e., $x_* = \underset{x}{\arg\max}\mathbb{V}_{l|D}[l(x)\pi(x)]$, where for our warped integrand we have $\mathbb{V}^{L}_{l|D}[l(x)\pi(x)] = \pi(x)^2\tilde{C}_D(x,x)\tilde{m}_D(x)^2$ as in equation 3.13. Specifically, the sample location x_* is chosen where the posterior covariance of the GP is largest, without consideration about the resulting magnitude of the posterior covariance of the GP.

Contrary to uncertainty sampling, future uncertainty sampling instead minimizes the uncertainty of the integrand by selecting sample locations x'_* such that $\mathbf{x}'_* = \underset{\mathbf{x}_*}{\arg\min}\left[\max_{x}\mathbb{V}_{l|D}[l(x)\pi(x)]\right]$ where again $\mathbb{V}_{l|D}[l(x)\pi(x)]$ is as in equation 3.13. That is, x'_* is chosen such that the posterior covariance of the GP is smallest within the region bounded by both x_d after the new quadrature point x'_* is selected, without consideration about whether x'_* also demarcates where the posterior covariance of the GP was largest prior to the selection of x'_*.

Figure 3.3 shows an example section of the posterior covariance of a GP, given by the large blue curves. Here x_d represents previously chosen sample locations that the model has used to update the GP, and y demarcates the highest posterior covariance of the GP in this example section after the next quadrature point is chosen. The posterior covariance of the GP after the next quadrature point is chosen are indicated by the two brown curves, each of which terminates at an x_d and the next quadrature point, either x_* in panel (A) or x'_* in panel (B).

Panel (A) shows the selection of the next quadrature point x_* using uncertainty sampling. Notice here that x_* is chosen where the posterior covariance of the GP is highest, without regard to the posterior covariance of the GP after the selection of this next quadrature point. That is, the quadrature point is chosen which reduces the posterior covariance of the GP the most. Panel (B), on the other hand, shows the selection of the next quadrature point x'_* using future uncertainty sampling. Here the next quadrature point is selected without regard to where the posterior covariance of the GP is highest, but instead minimizes the posterior covariance of the GP after x'_* is chosen.

This illustrates that while both use the posterior covariance of the GP as consideration in the selection of quadrature points, there are fundamental differences between uncertainty sampling and future uncertainty sampling. While uncertainty sampling will always chose the quadrature point with the largest reduction in entropy, future uncertainty sampling will always chose the quadrature point which results in the lowest posterior covariance of the GP, as illustrated by $y_1 > y_2$. Furthermore, since we warp the GP as previously discussed, and therefore the posterior covariance of the GP is not symmetric, it cannot be that $y_1 \leq y_2$ whenever our search space is all real numbers within our domain.

3.2.2 Superiority of Future Uncertainty Sampling versus Uncertainty Sampling (Part 1)

The seemingly minor difference between selecting the next quadrature point based on future uncertainty sampling versus uncertainty sampling - that is, minimizing the posterior covariance of the GP versus maximizing the reduction in the posterior covariance of the GP - in practice allows for very significant changes. One of these changes is that quadrature point selection need not proceed in a sequential manner.

A characteristics of the standard BQ model that is important to our paper is that the choice of sample locations is a sequential process, i.e., first x_* is chosen, and only after $x_* \in x_d$ is the subsequent x_* chosen (Rasmussen et al., 2010) Since each sample is expensive, this sequential process is very costly.

Recently Wagstaff (2018A) drew on the Bayesian Optimization literature to implement what they referred to as Batch Bayesian Quadrature (BBQ), where several samples are taken in parallel, i.e., each $\mathbf{x}_*$ is a batch of sample locations. To accomplish this, the authors utilized the probabilistic model of BQ to guide exploration of a search space by defining an acquisition function to be maximized. Two such methods are Kriging Believer and Local Penalization.

Utilizing GP model

In Kriging Believer (Ginsbourger et al., 2010), instead of updating the BQ model after every sample, the functional value of a sample location is set equal to the posterior mean, then the GP model is updated before selecting a subsequent sample location. This process is repeated until the batch of sample locations chosen is sufficiently large, and then the evaluations of those sample locations are made and the BQ model is updated.

Acquisition Function Penalization

Unlike Kriging Believer which utilizes the entire GP model, Local Penalization (Gonzáles et al., 2016; Wagstaff, 2018A) only directly modifies the acquisition function around each selected point. This also allows for the selection of a batch of sample locations before those location are evaluated and the BQ model is updated.

A common characteristic of the batch selection procedures in BBQ is that sample locations within a batch are selected serially. While such parallelization procedures reduce the cost of sample selection, they do not consider how changes in sample locations within a batch impact the efficiency of the reduction in uncertainty of the integrand, *ceteris paribus*. In the next section we describe our method of taking this into consideration, and subsequently show that it results in lower absolute error estimates of the integral.

Regardless of whether the entire GP model (Kriging Believer) is used, or a technique which directly penalizes the acquisition function (Local Penalization), uncertainty sampling is an inherently sequential procedure. See also Chapter 2 for a discussion on Kriging and Local Penalization. To see this, it is important to make the distinction between selecting quadrature points simultaneously and selecting quadrature points within the same batch. Since we are implementing batch Bayesian quadrature, were are working by the premise that all points used to update the model will be selected within a batch of other points, but that not every point will come from the same batch. However, updating a model simultaneously with a batch of points is not the same thing as selecting points within a batch simultaneously. Within each batch, points can be selected sequentially or simultaneously.

If uncertainty sampling is used to select quadrature points, as figure 3.3 panel (A) shows, x_* is selected where the posterior covariance of the GP is largest. If instead of picking one quadrature point, two quadrature points were to be selected simultaneously, then the second highest point would be immediately adjacent to x_*.

To see this, consider figure 3.4, where the top panel shows a function with an example GP posterior after 6 points are chosen and the model is updated, where the black dots are the chosen points, the blue dotted line is the function, the red solid line is the GP posterior mean, and the red shaded area is two standard deviations of the GP posterior covariance. The bottom panel of figure 3.4 shows the magnitude of the posterior covariance at each x in the search space, and therefore also represents the acquisition function.

As can be seen in figure 3.4 since uncertainty sampling selects the location with the highest posterior covariance, if two points were to be selected simultaneously, then the two points with the largest posterior covariance of the GP would be chosen, which using a unimodal distribution of posterior covariance of the GP would need to be immediately adjacent to each other. That is, x_{*_1} and x_{*_2} would be chosen. Using immediately adjacent points in quadrature is very uninformative, and much better batches of points can be selected.

Similarly, as can be seen in figure 3.5 panel (A), if uncertainty sampling were used to select the point with the largest posterior covariance of the GP, as in points x_{*_1} and x_{*_3}, this would only be possible if first x_{*_1} is chosen, the model updated or the penalization to the acquisition function applied, and then choose x_{*_3}. However, this would then be a sequential process, and the points would no longer be chosen simultaneously even though they are within the same batch.

Alternatively, if the restriction were relaxed that the point with the largest posterior covariance of the GP must be chosen, then it would be possible to find methods where points could be chosen simultaneously. As previously mentioned, one such method is future uncertainty sampling, which we devise and apply in this research.

To that end, if two or more points can be chosen simultaneously, then the question arises of how to chose these points. Since our acquisition function utilizes the posterior covariance of the GP, we select quadrature points which minimize the posterior variance of the GP, as described by equation 3.14.

3.2.3 Superiority of Future Uncertainty Sampling versus Uncertainty Sampling (Part 2)

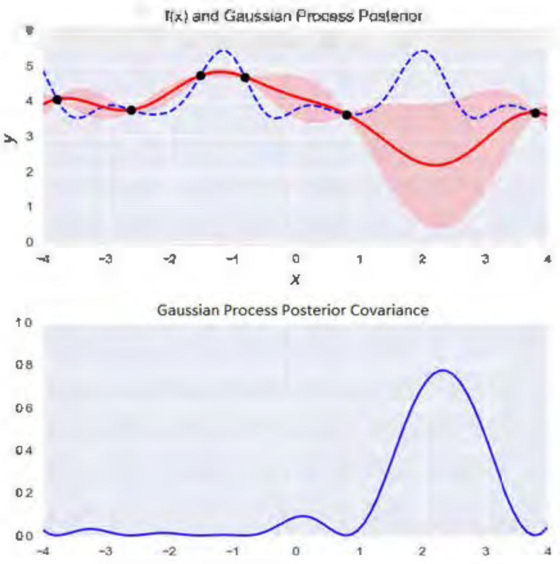


Figure 3.4: f(x) and Gaussian process posterior

A significant benefit of selecting points simultaneously within a batch is the ability for quadrature points to better work in harmony to reduce the posterior covariance of the GP. To see this, as previously discussed, using uncertainty sampling would first chose the quadrature point which corresponds to the maximum posterior covariance of the GP, and after the model is updated or a penalization is applied to the acquisition function, a subsequent quadrature point would be selected. Diagrammatically, this would proceed by first selecting the quadrature point x_{*_1} near $x = 2.4$ in figure 3.4. For illustrative purposes it will be noted that this first quadrature point lies within the neighborhood formed by the points $x_{d_1} = 0.8$ and $x_{d_2} = 3.8$, where both of these represent quadrature points selected from previous batches. Since uncertainty sampling is applied, once x_{*_1} is chosen and after the model is updated or a penalization is applied, the subsequent largest posterior covariance of the GP would be as in figure

3.5 panel (A). Here uncertainty sampling would select the next quadrature point x_{*_2} near $x = 1.6$ as this is now the location corresponding to the largest posterior covariance of the GP.

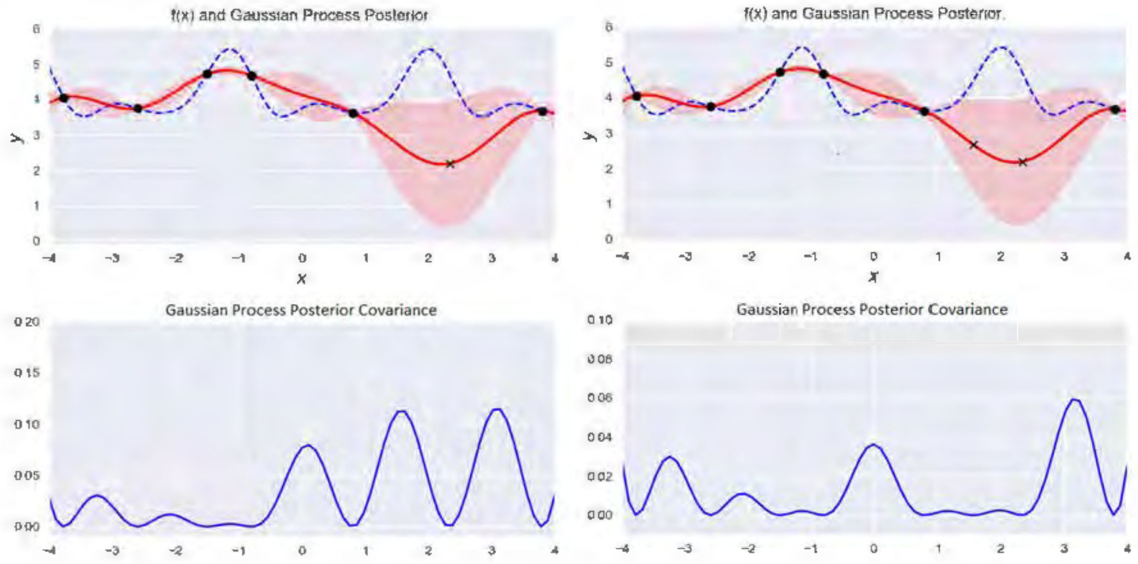


Figure 3.5: f(x) and Gaussian process posterior

It is extremely crucial to note, as can be seen in figure 3.5 that after x_{*_2} is selected and the GP model updated or a penalization applied, that the largest posterior covariance of the GP still lies within the neighborhood defined by x_{d_1} and x_{d_2}. Therefore, if uncertainty sampling would still be applied, the next quadrature point x_{*_3} would be selected within this neighborhood. Therefore, only after these three quadrature points are selected would uncertainty sampling allow for the exploration of other neighborhoods, and the selection of subsequent quadrature points outside of the neighborhood defined by x_{d_1} and x_{d_2}. Specifically, uncertainty sampling and the acquisition function would lead to $x_{*_4} = 0$.

Alternatively, since we implement future uncertainty sampling instead of uncertainty sampling, we not only can choose quadrature points simultaneously, but we also do not need to choose the location where the posterior variance of the GP is largest. Instead, as can be seen in figure 3.6, we can chose points x'_{*_1} near $x = 1.8$

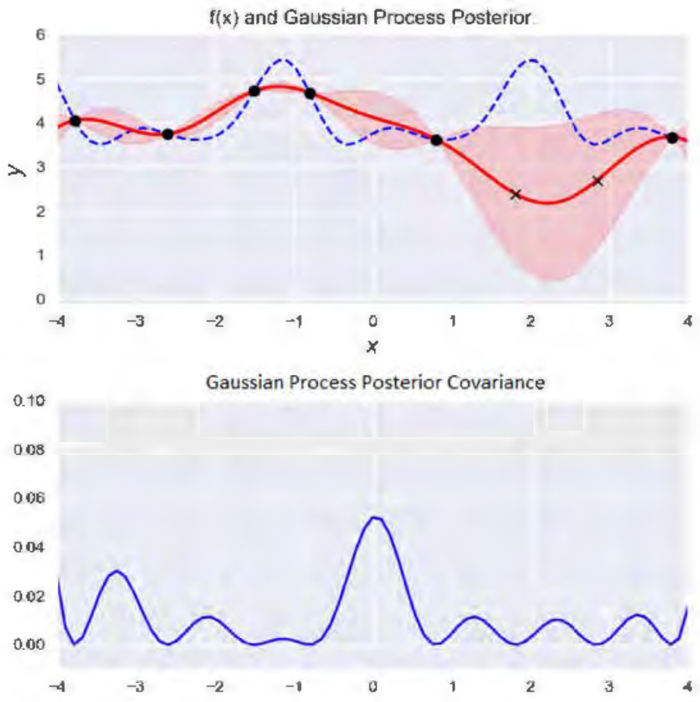


Figure 3.6: f(x) and Gaussian process posterior

and x'_{*_2} near $x = 2.8$. First note that $x'_{*_1} \neq x_{*_1}$ and $x'_{*_2} \neq x_{*_2}$. Most importantly, note that the largest posterior covariance of the GP within the neighborhood bounded by x_{d_1} and x_{d_2} is smaller than the largest posterior covariance of the GP not within this neighborhood. This means that the location for the next quadrature point x_{*_3} will not be in the neighborhood bounded by x_{d_1} and x_{d_2}.

Therefore another consequence which arises from using future uncertainty sampling instead of uncertainty sampling, is that the posterior covariance of the GP is more efficiently reduced, allowing us to explore the search space more efficiently. Said differently, in order to minimize the posterior variance of the GP within the neighborhood defined by x_{d_1} and x_{d_2}, in this example uncertainty sampling requires the selection of three points within this neighborhood before it can explore the search space outside of this neighborhood. On the other hand, future uncertainty sampling in this example only requires two points within the neighborhood defined by x_{d_1} and x_{d_2} in order to reduce the posterior covariance of the GP to a level which allows for the exploration of the search space outside of this neighborhood. Therefore, despite using the same acquisition function, the seemingly minor change required to implement future uncertainty sampling yields this tremendous benefit.

3.3 Test Functions

Since we are interested in intractable integrals of the form $Z = \int l(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$ where both $l(\mathbf{x})$ (e.g., a likelihood) and $\pi(\mathbf{x})$ (e.g., a prior) are non-negative, it is important to evaluate our novel procedure using test functions appropriate to this scenario. Specifically, our GP is parameterized by a mean $\mu(x)$ and a scaled Gaussian covariance $K(x, x') = \lambda^2 exp(-\frac{1}{2}\frac{(x-x')^2}{\sigma^2})$ where the output length-scale is given by λ and the input input length-scale is given by σ. Despite our warping $\tilde{l}(x) = \sqrt{2(l(x) - \alpha)}$ where α is a small scalar, and subsequent linearization $f : \tilde{l} \mapsto l = \alpha + \frac{1}{2}\tilde{l}^2$ we have shown that a GP can be approximated for our likelihood $l(x)$ given by $p(l \mid D) \simeq GP(l;\, m^L_D(\cdot),\, C^L_D(\cdot,\cdot))$ where $m^L_D(x) = \alpha + \frac{1}{2}\tilde{m}_D(x)^2$ and $C^L_D(x, x') = \tilde{m}_D(x)\tilde{C}_D(x, x')\tilde{m}_D(x')$ and $\tilde{m}_D(x) = K(x, x_d)K^{-1}(x_d, x_d)\tilde{l}(x_d)$ and $\tilde{C}_D(x, x') = K(x, x') - K(x, x_d)K^{-1}(x_d, x_d)K(x_d, x')$.

Furthermore, our prior $\pi(x)$ is Gaussian. Given our kernel K in our GP approximation for $l(x)$ and prior $\pi(x)$ are both Gaussian, it is well established that analytic results can be obtained for both the mean and variance of Z given by $\mathbb{E}_{l|D}[Z] = \int m_D(x)\pi(x)dx$ and $\mathbb{V}_{l|D}[Z] = \int\int C_D(x, x')\pi(x)\pi(x')dxdx'$, respectively. (Rasmussen et al., 2006; Briol et al., 2015). Other combinations of K and $\pi(x)$ also lead to analytic results (Gunter et al., 2014), however these are not considered here.

It is important to note that any function can be chosen for $l(x)$ as long as it meets the restrictions applied in our procedure, namely that it conforms to the properties of a likelihood. Comparisons and evaluations of validity, efficiency, and reliability are frequently carried out by using a set of standard benchmarks or test functions prevalent in the literature. The number of test functions used varies widely depending on the evaluation performed, but typically varies between the low single digits and several dozen. The primary consideration made in comparisons and evaluations therefore rests not on the number of test functions used, but instead on the selection of test functions which allow for the evaluation and comparison of the characteristics of interest. In order to establish that our comparison of batch updating versus not updating is complete, we first discuss the properties which distinguish test functions. Subsequently we choose and discuss test functions for our evaluation and comparison of batch updating versus not updating.

3.3.1 Test Function Properties

The literature abounds in benchmark test functions for global optimization problems and in the quadrature literature. However, many of the test function originate in Bayesian Optimization. Despite the similarities in the theoretical underpinnings between Bayesian Optimization and Bayesian Quadrature, their divergence in objectives presents a challenge. Specifically, optimization seeks to identify one global maximum whereas quadrature does not. Therefore, the optimization literature makes distinctions among test functions which are imperative to its objective, but quadrature may omit such distinctions, and *vice versa*. In order to show that the test functions we selected are necessary and sufficient, we begin with a discussion of the relevant characteristics of benchmark test functions and they pertain to both optimization and quadrature. For reference see Suganthan et al. (2005) and Jamil et al. (2013).

Linearity

The linearity of all or part of a test function has significant implications in both optimization and quadrature. As linearity increases, the accuracy of estimates increase more quickly.

Cyclicality

Functions which are not uniformly linear may be cyclic or piece-wise cyclic, where cyclicality is defined as

$$f(x_1, x_2, x_3, ..., x_{n-1}, x_n) = f(x_i, x_{i+1}, ..., x_{n-1}, x_n, x_1, x_2, ...x_{i-1}),$$

where $1 < i \leq n$. Cyclicality plays an important role both in optimization for the identification of local versus global maxima, and in quadrature for symmetry implications.

Modality

Modality plays a more crucial role in optimization than in quadrature since optimization seeks to identify global maxima and quadrature does not. Nevertheless, it is important to note that the search algorithm applied in quadrature may be similar to that of quadrature, and therefore if the algorithm has a strong propensity to remain in such intervals, this can have a negative impact.

Basins and Valleys

Basins are relatively large areas surrounded by steep declines. Similarly, valleys are relatively narrow areas of little change surrounded by regions of steep decent. In optimization, search algorithms functioning as minimizers are attracted to basins and valleys alike, however these algorithms may be significantly slowed in such regions. While the consequences are much less detrimental in quadrature as compared to optimization, the application of similar search algorithms can again have a negative impact.

Oscillation

The frequency and magnitude of oscillations of a test function on an interval, I, in the domain has significant implications both in optimization and quadrature, where an oscillation of a test function is defined as

$$\omega_f(I) = sup f(x) - inf f(x),$$

where $x \in I$. In regards to optimization, algorithms intended to explore the search space can become stuck in sizable intervals defining basins, valleys, and local maxima. Since quadrature may employ similar algorithms, these same encumbrances may occur.

However, in quadrature frequent and large oscillations may potentially lead to a much more significant detriment. While optimization concerns itself with the efficiency with which a global maximum is identified, quadrature concerns itself with the entire shape of the function. Sudden and large oscillations significantly affect the quadrature results.

Separability

A separable function is a function which contains variables or parameters which are independent of its other variables. If all n variables within a function are independent, then a sequence of n independent processes can be performed both in optimization and quadrature. Therefore, separability is a measure of difficulty of test functions, where separable functions are relatively easier to solve.

Dimensionality

The number of variables or parameters corresponds to the dimensionality of the test functions. As the dimensionality of the test function increases, performing the search algorithm becomes increasingly cumbersome, especially for non-separable test functions as discussed above. This stems from the fact that the search space increases drastically, which clearly applies to both optimization and quadrature. However, since in this investigation we restrict ourselves to two dimensions, we omit many of the difficulties of separability and dimensionality alike.

3.3.2 Ackley, Weierstrass and Synthetic Test Functions

Given the common consideration in the selection of test functions, we choose two test functions very prevalent in the literature, namely the Ackley test function and the Weierstrass test function. Subsequently we also use a synthetic test function also used in the literature on Batch Bayesian Quadrature.

Ackley test function

In the quadrature literature, and in many similar fields, "black box" problems are those where a function is either unknown or intractable, but nevertheless needs to be evaluated at strategically relevant locations. In the optimization literature, for example, so called "black box optimization" problems require the optimization of an unknown function, where the optimum is either a global maximum or minimum depending on the evaluation objective.

To perform the search for points to evaluate the unknown function, not knowing the structure of the function provides particular challenges. Such challenges are not uncommon in the parallel field of quadrature, since the problems encountered in this field similarly arises due to either not knowing the functional form of the entire integrand or the $l(x)$ portion thereof, or the fact that the analytic evaluation of the resulting integral in intractable.

In the optimization literature, and subsequently borrowed into the Batch Bayesian Quadrature literature, two vital facets of such evaluation have been articulated. The first is the ability to learn from the information received while searching for strategically relevant location to evaluate. This can be referred to as exploitation in the relevant literature.

Secondly, the ability to continue to explore the search space is vital both in the optimization literature and in quadrature. If the possibilities within a region have either been exhausted or have been deemed unsuitable for further exploitation by the search process, then other regions must be sampled. More importantly, whenever the current region has not been exhausted and further exploitation is warranted, having the ability to nevertheless divert efforts to explore other regions for possible higher rewards is important. This balance, which is highly relevant in both optimization and quadrature, is important to balance. Many techniques have been devised to achieve a suitable balance to reach a specified objective.

One such mechanism was proposed by Ackley (1987). He proposed a connectionist learning machine which produces a search strategy called stochastic iterated genetic hill-climbing (SIGH). Considered over a short time horizon, SIGH conducts a coarse-to-fine searching strategy, similar to simulated annealing and genetic algorithms. A notable difference is that using SIGH it is possible to reverse the convergence process. This implementation therefore makes it possible to unpack the search after it completes. This allows for the recovery of information about the search space obtained during convergence. To that end, SIGH can be viewed as a genetic and a stochastic hill-climbing algorithm, in which genetic search reveals points for hill-climbing, and hill-climbing influences subsequent genetic search. Testing was conducted by Ackley on a set of illustrative functions. Not only has SIGH been shown to be competitive with genetic algorithms and simulated annealing in most cases, and markedly superior where the uphill directions of functions lead away from the global maximum, but his work has also contributed to the test function literature and become frequently used test functions.

The most commonly used Ackley test function is given by

$$-20e^{-0.02\sqrt{D^{-1}\sum_{i=1}^{D} x_i^2}} - e^{D^{-1}\sum_{i=1}^{D} cos(2\pi x_i)} + 20 + e \tag{3.15}$$

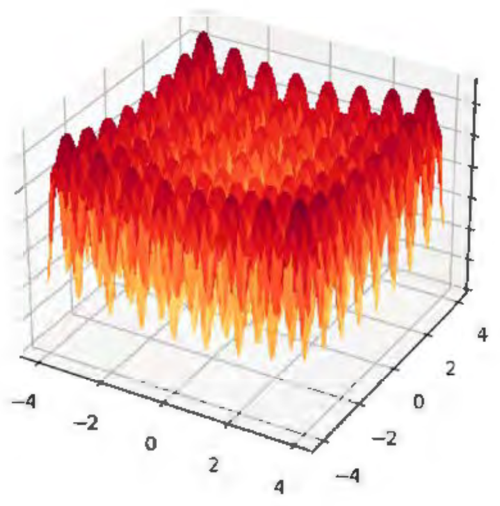


Figure 3.7: Ackley test function in 2D

Pertaining to the discussion about differences in test function properties above, it should be noted that the Ackley test function is Continuous, differentiable, non-separable, scalable, and multi-modal. It can be applied in many dimensions, and if necessary parameters can be changed to achieve specific effects. For example, the addition of $20 + e$ was appended in some literature applications of the Ackley function in order to center the function at the origin. Nevertheless, in our application there are no basins or valleys. Furthermore, there is a not a cyclicality in the original sense, however there is a periodicity of some facets due to the use of the cosine function, and therefore it is also not linear.

Of particular importance are the significant number of often large local fluctuations. Not only are these difficult to traverse in optimization, but also in quadrature they present a particular challenge. Since the posterior covariance of the GP will be large in sections with significant fluctuations, search algorithms may be stuck in these regions. Our novel method of more efficiently reducing the posterior covariance of the GP will allow us to be much less susceptible to becoming stuck in these regions.

Having a test function that will allow us to exploit this added benefit of our novel method makes the selection of the Ackley test function particularly relevant. This is in addition to the many applications already discussed for the Ackely test function, and its wide use in the literature.

Weierstrass test function

The Weierstrass test function was originally used in the late 1800s as an example of a real-valued function that is continuous everywhere but differentiable nowhere. Due to it being a fractal curve, it has often been utilized as a so-called pathological function, that is one which possesses counter-intuitive or irregular properties.

Its original purpose was to investigate the concept that every continuous function is differentiable except on a set of isolated points. This function served as an example that proved the concept that continuity did not guarantee almost-everywhere differentiability. Due to the difficulty of visualizing such functions, they did not gain wide usage until models of Brownian motion required jagged functions. Due to the fact that Brownian motion, among other applications, has characteristics lending itself well to the movement of particles within a medium or random walks, the use of the Weierstrass test function has many far-reaching applications in many fields. These include, finance, statistics, astronomy, physics, electromagnetism, etc. Therefore, devising a method which can better estimate the functional form in black box problems as described above, and determine more robust estimates of the integral involving such functions, makes our novel contribution widely applicable and very consequential.

The Weierstrass test function is given by

$$\sum_{i=1}^{n}[\sum_{k=0}^{kmax} a^k cos(2\pi b^k(x_i+0.5)) - n\sum_{k=0}^{kmax} a^k cos(\pi b^k)] \tag{3.16}$$

where the parameter value for a, b, k, and n can be set as desired or necessitated by the application or investigation being conducted.

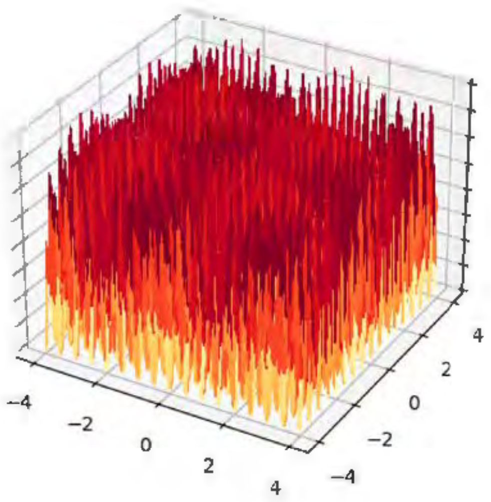


Figure 3.8: Weierstrass test function in 2D

Similar to our discussion above, the Weierstrass test function has several properties which must be noted since their are relevant to our investigation. The Weierstrass test function stated above is continuous, differentiable only on a set of points, separable, scalable depending on the choice of parameters stated above, and multi-modal. Its original applications were primarily in one dimensions for reasons already articulated, but it most certainly can be used in multiple dimensions including very high dimensions. It lacks any basins or valleys. It is not cyclical in the original sense defined, but does have a repetitive aspect to it due to the use of the cosine function. It is therefore also not linear.

For the Weierstrass test function, as in the Ackley test function, an important component of this test function, and therefore also a reason for its inclusion in this study, are the frequent and often significant fluctuations. These transitions are often difficult for optimization since search algorithms can get stuck in such regions. Similarly in quadrature, search function can get stuck in these regions due to the fact that the posterior covariance of the GP can be significant in such regions. Therefore, in addition to its many application described above, the functional form of the Weierstrass test function lends itself well to differentiate the novel application devised in this work compared to those which already exists in the literature. We will see later in the numerical results section that the novel method devised in this study does significantly better in not only estimating the function form of the integrand which utilizes the Weierstrass test function in place of the likelihood, but therefore also gives more robust numerical estimates of the integral.

Synthetic test function

The synthetic test function has the important distinction of being a test function used in the evaluation of Batch Bayesian Quadrature in the literature. While Batch Bayesian Optimization has been in existence for nearly a decade, Batch Bayesian Quadrature was devised in 2018 (Wagstaff, 2018A). Due to the relative newness of this technique, having a function used in multiple studies relevant to this topic will serve as an easy yardstick in the comparison of various techniques and applications. The synthetic test function is given by

$$\frac{\sin(\mathbf{x}) + 0.5\cos(3\mathbf{x})^2}{(\frac{\mathbf{x}}{2})^2 + 0.3} \tag{3.17}$$

As the graph of the synthetic test function shows, there are much fewer fluctuations in the synthetic test function compared to both the Ackley test function and the Weierstrass test function. Nevertheless, especially in the two dimensional application employed here, there are regions which, if unexplored, will significantly influence the determination of both the functional form of the integrand as well as the expected value of the integral. An added benefit of having significantly fewer fluctuations is that the computational cost of running both our novel algorithm and that already present in the literature is significantly lower. This allows for the potential comparison

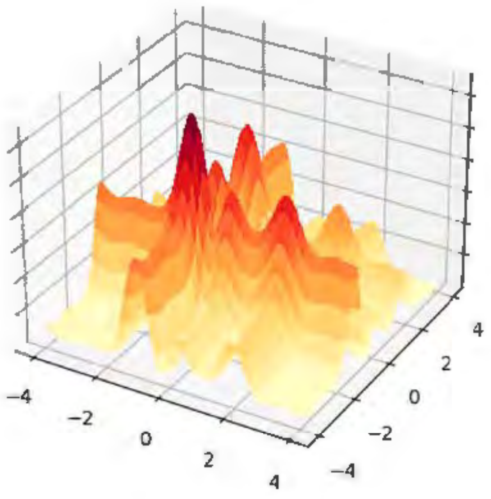


Figure 3.9: Synthetic test function in 2D

in higher dimensions as well as that of increased batch size, or even the changing of the stopping criterion employed, as we will see later.

3.4 Dynamic Domain Decomposition

A common argument raised in simultaneous point selection is the increase in computational cost. The selection of $\mathbf{x}'_*$ has very high computational cost as the size of batches is increased. Fortunately in quadrature, locations close to sampled points are uninformative. Therefore, to mitigate the computational cost, we descretize the search space by designating neighborhoods within our domain, and ensure each neighborhood maintains a quantity of locations which balances information gleaned from location selection with computational cost of location inclusion in the search space. The optimal balance of these competing facets remains an open question, but will be discussed in section 3.8. To begin our decomposition of the domain into neighborhoods, it should be noted our model ensures that before any evaluations of the integrand its uncertainty is Gaussian (Rasmussen et al., 2003; Gunter et al., 2014).

3.4.1 One Dimension

Sample locations x'_* are chosen according to equation 3.14. Once the model is updated, these sampled locations become partitions which form neighborhoods within the domain. At the onset of each iteration of our algorithm, each neighborhood is populated with 100 equally spaced locations. Our search in restricted to the neighborhood which initially contains $\arg\max_x \mathbb{V}_{l|D}[l(x)\pi(x)]$. As in Kriging Believer, after selecting x'_* we set its functional value equal to the posterior mean of the GP, and update our GP model to take this new evidence into account. If prior to the completion of point selection for a given batch this neighborhood again contains $\arg\max_x \mathbb{V}_{l|D}[l(x)\pi(x)]$, then all sample locations within this neighborhood are updated while all sample locations outside of this neighborhood remain unchanged. That is, we select $\mathbf{x}'_* = \arg\min_{\mathbf{x}_*} \left[\max_x \mathbb{V}_{l|D}[l(x)\pi(x)]\right]$, where now $\mathbf{x}'_*$ is a vector of points and everything else remains as previously defined.

It should be noted that the selection of x'_* within a given neighborhood affects the uncertainty of the integrand outside of this neighborhood. However, given our model, the reduction of the uncertainty of locations in adjacent neighborhoods will stem primarily from the partitions of the neighborhood containing x'_* and less from x'_* itself. The impact of x'_* on the uncertainty of the integrand will continue to diminish in neighborhoods farther from x'_*. Therefore, any fluctuation in uncertainty in neighborhoods not containing x'_* but arising from its selection is not taken into consideration.

3.4.2 Two Dimensions: Part 1

The dynamic domain decomposition method in one dimension does not lend itself well to multiple dimensions. Specifically, the use of sampled locations from prior batches in the formation of neighborhoods to aid in the search process for sample locations of the current batch, where the partitions of a neighborhood affect the uncertainty of the integrand more than points outside of that neighborhood, would increase the computational cost of the dynamic domain decomposition unjustifiable high potentially without inducing better estimates. While an optimal domain decomposition in this setting remains an open question that will be discussed in section 3.8, we implement a method suitable for our intentions.

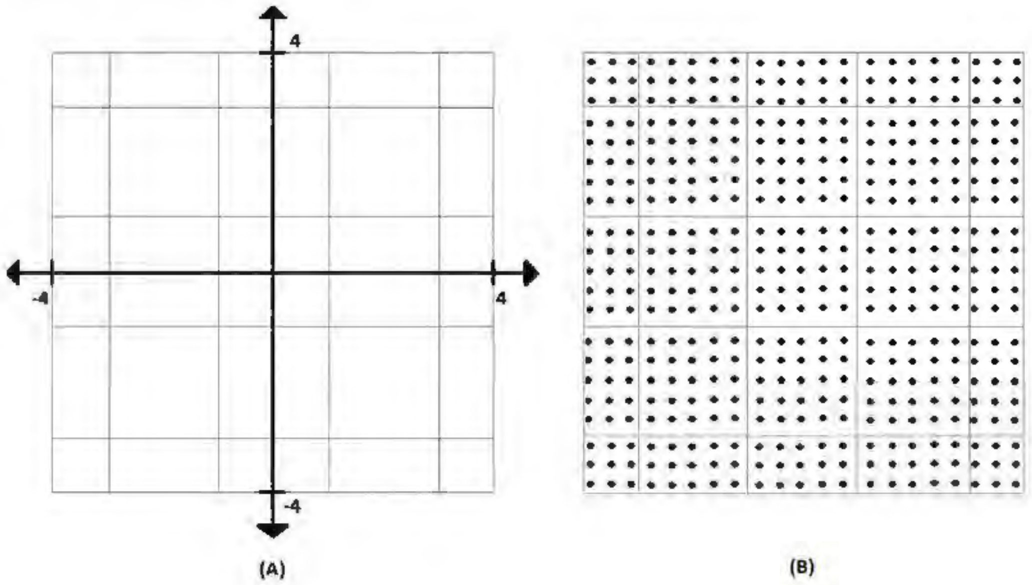


Figure 3.10: 2D domain decomposition

We naively partition the domain, opting for determining the number of partitions rather than the length of the dimensions of each partition. That is, consider an integrand, and hence a domain, in two dimensions. We allow the domain of our integrand to be between -4 and 4 in both dimensions. To begin, we center our first partition on the origin, and then determine how far away from the origin our partition goes, where we determine this distance by the length of partitions we intend on achieving in a direction instead of the number of partitions. However, this determination of partition location could be accomplished *vice versa*. That is, we could determine how far away from the origin our partition goes, where we determine this distance by the number of partitions we intend on achieving in a direction instead of the length of the partition. Nevertheless, we implement the former in this research.

It should be noted that every time we form a partition we form a rectangle, and therefore for the first partition described above the axis in a Cartesian coordinate system bifurcated the sides of this rectangle, as can be seen in panel (A) in figure 3.10. The remaining partitions replicate the dimensions of the first, until either the end of the domain is reached, or the end of the domain is reached after the formation of a partial partition. We used the same number of partitions for each dimension, and therefore also had the same length on each side of every whole partition. Clearly, whenever the domain was not perfectly divisible by the number of partitions chosen, then there remained a fraction of a partition which consisted of different dimensions that the partition encompassing the origin.

It should be noted that the decomposition of the domain by either length or number of partitions in this study was viewed as a preference, and did not impact the results of our investigation. It was also not considered when or how the choice of one is preferential to the other.

We subsequently form neighborhoods encompassing each partition, where the length of the each dimension of the neighborhood is a multiple of its respective partition dimension. In our research this multiple was 0.5 of the length of the partition, were again 0.5 was a number chosen as preference and yielded suitable results for our purposes but the optimal additional length was not investigated. This can be seen in figure 3.11, where the brown section engulfing the blue section are the neighborhoods.

Prior to the commencement of our batch selection algorithm, we populate each partition with 25 equally spaced sample locations, as can be seen in figure 3.10 panel (B). This discretization of the search space allows us to minimize the computational cost while still retaining sample locations which will be informative.

3.4.3 Batch Selection

In this section we introduce, to our knowledge, the first Batch Bayesian Quadrature routine where sample locations do not need to be selected sequentially, or what we refer to as *selection updating.* To achieve this, we also continue to introduce a novel form of dynamic domain decomposition which allows us to reduce the computational cost of our search. In combination, these results facilitate a more efficient reduction in the posterior covariance of the GP and hence lead to more robust numerical estimates of the integral of interest.

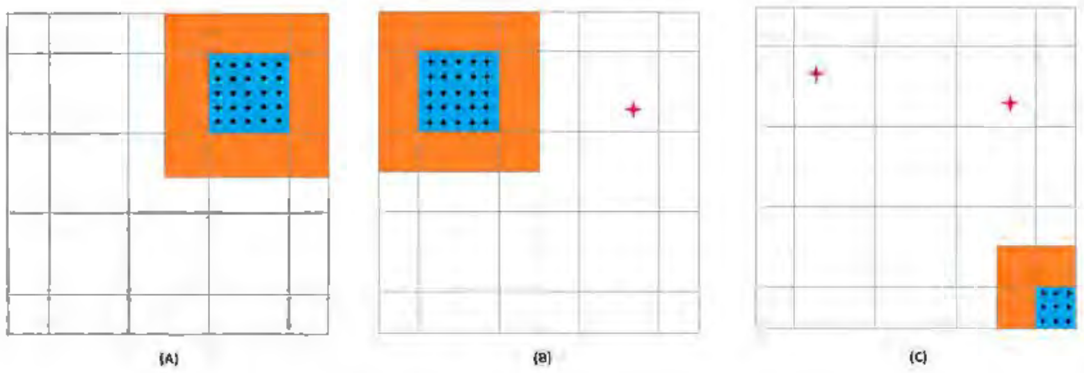


Figure 3.11: Batch selection in 2D

To begin, we identify the partition of our domain which contains $x_* = \arg\max_x \mathbb{V}_{l|D}[l(x)\pi(x)]$, where x_*, l, D, π are as previously defined, and $x \in \mathbb{R}$ in the domain. This is identified by the blue box in panel (A) in figure 3.11. We then determine which one of the 25 possible sample locations within this partition leads to the smallest posterior covariance of the GP within our neighborhood, where the neighborhood is as previously defined and represented by the combination of the blue partition and brown adjacent regions in figure 3.11. That is, we find $x'_* = \arg\min_{x_*} \left[\max_x \mathbb{V}_{l|D}[l(x)\pi(x)]\right]$, where x'_* is as previously defined in the section describing our novel method of future uncertainty sampling, and must also be one of the 25 possible sample locations. Specifically, x_* is to be understood as any point from our possible sample locations which has been evaluated as if it had been chosen as a quadrature point, and x'_* is the point among those which leads to the smallest posterior covariance of the GP. This point is identified as a red star in figure 3.11.

To identify a subsequent quadrature point, we again identify the partition of our domain which contains $x_* = \arg\max_x \mathbb{V}_{l|D}[l(x)\pi(x)]$ after the selection of any previous sampled points have been taken into consideration, such as through Kriging Believer of Local Penalization. We then again find $x'_* = \arg\min_{x_*} \left[\max_x \mathbb{V}_{l|D}[l(x)\pi(x)]\right]$ in this partition, as shown in panel (B) and panel (C) in figure 3.11. These subsequent quadrature points within the current batch are also identified by red stars in figure 3.11 through figure 3.13.

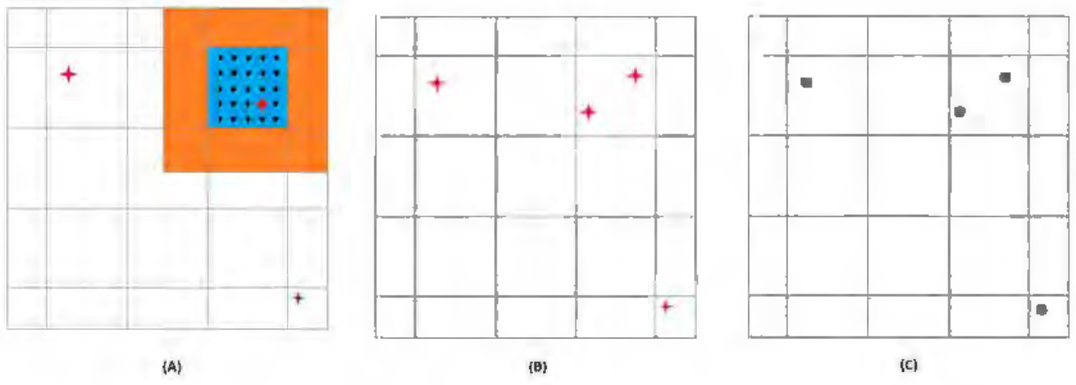


Figure 3.12: Batch selection in 2D

If, however, $x_* = \arg\max_x \mathbb{V}_{l|D}[l(x)\pi(x)]$ is contained within a partition which already possesses a quadrature point in the *current* batch being selected, then we can update our quadrature point selection *for that region*, where any other quadrature points which may have already been identified in other regions remain unchanged.

Pictorially this can best be seen in the comparison of the partition near the top right in figure 3.12 panel (A) to the same partition in figure 3.12 panel (B). As can be seen is panel (A), in the current batch a quadrature point has already been sampled in the lower right section of this partition, which is is represented by the red star. When the search algorithm again identified this partition, two quadrature points are selected in this partition that are *different* than the first quadrature point. That is, the location of the two red stars, while still within the partition being discussed, are in different locations than the initial quadrature point in that partition in panel (A). Therefore, while the quadrature point locations within that partition will be updated and hence change in order to work better in harmony to reduce the posterior covariance of the GP, this process does not affect the location of the quadrature points already sampled in other regions, which also aids in reducing computational cost.

To accomplish this, we find $\mathbf{x}'_* = \arg\min_{\mathbf{x}_*} \left[\max_x \mathbb{V}_{l|D}[l(x)\pi(x)]\right]$. Note here that we are finding a vector of points using future uncertainty sampling. Specifically, we evaluate all combinations of two sample locations points to determine which selection of these pairs of points would lead to the smallest posterior covariance of the GP. Similarly, the number of point combinations would increase depending on how many times a region contained $x_* = \arg\max_x \mathbb{V}_{l|D}[l(x)\pi(x)]$ during the selection of a single batch.

This last step described is of tremendous importance. Unlike the methods in the literature on batch Bayesian quadrature, this facet not only allows us to update the selection of quadrature points through a simultaneous evaluation of multiple sample locations, but even though computational cost can still be high in certain cases where large batches are selected and the posterior covariance of the GP is disproportionately high in one or a few regions, it also allows us to significantly lower the computational cost so that simultaneous selection of quadrature points is feasible.

Once the requisite number of quadrature points has been selected, in our illustrative example in figures 3.11 through 3.12 the requisite number is a batch size of 4, then these points are used to update the model. That is, the quadrature point locations are used to evaluate the integrand, and these values are used to update both the GP and BQ estimates. Once this has been accomplished, the location of these quadrature points cannot be changed. As we said earlier, in one dimension these quadrature points become the new demarcations of the partitions of our domain, but this does not lend itself well to two or multiple dimensions for previously stated reasons. To depict that these quadrature point locations cannot be changed, in figure 3.12 panel (C) we transition their representation from red starts to purple rectangles.

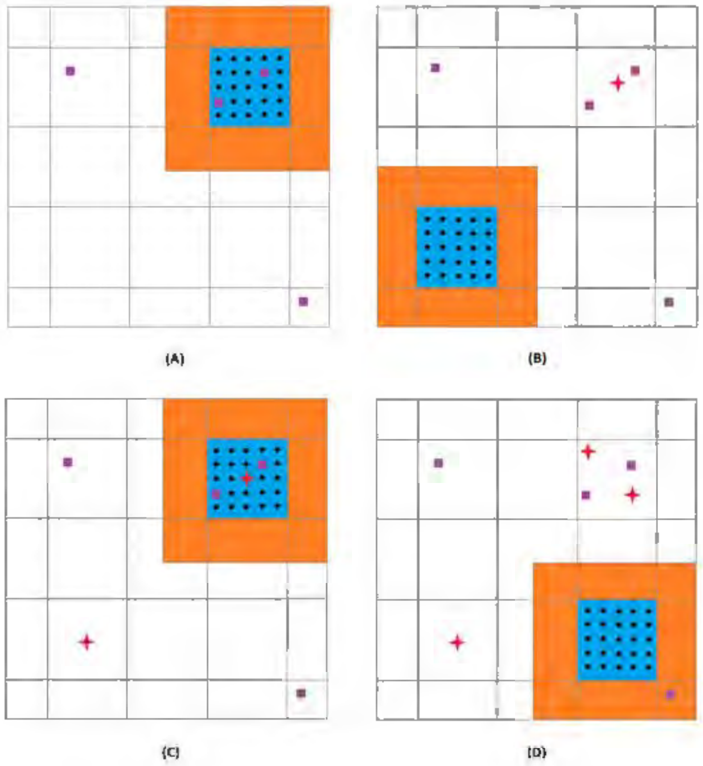


Figure 3.13: Batch selection in 2D

Once a batch of points is complete and the model is updated, any subsequent batch selections proceed similar to the first batch described with the exception that all quadrature points from any previous batch which have already been used to update the model cannot change. Again, pictorially these are the quadrature point location represented by purple rectangles. This is shown in figure 3.13 were one batch has been used to update the model, and now a second batch is being selected.

As can be seen, while each partition was initially populated by 25 equally spaced quadrature points, once a batch has been selected and the model updated, any quadrature points locations that were part of that batch are no longer available to subsequent batches. That is, once a location is used once it cannot be used again since using that location again would provide our model with absolutely no additional new information. For specific examples and accompanying numerical results, see the section of this document pertaining to numerical simulations and the appropriate tables at the end of this document.

This has two consequences regarding our novel batch selection method and novel dynamic domain decomposition technique. First, it reduces computational cost during the batch selection phase. If we again consider our previous example, when a second point was selected within a region as in figure 3.12 panel (A), we needed to evaluate all possible combinations of size two of possible quadrature points within that region in order to determine which pair minimized the posterior covariance of the GP. However, as can be seen in figure 3.13 panel (C), only 23 of the 25 locations have not been used by quadrature points in previous batches. Therefore, when selecting multiple quadrature points in the current batch from this region, say two quadrature points, then only all combinations of pairs of quadrature points using the 23 unused locations must be evaluated to see which pairs minimize the posterior variance of the GP. Hence, this requires less computational cost.

Second, It is possible, especially for large batch sizes and in regions which have a disproportionately high posterior covariance of the GP compared to other regions, that as the number of batches taken increases that regions may require more than 25 points. This will be addressed in the next section.

3.4.4 Two Dimensions: Part 2

A defining feature of our novel domain decomposition is that it is dynamic. As was stated above, each partition is initially populated with 25 equally spaced points, however it is possible that these points may be exhausted before the cessation criterion is met. For a discussion on the cessation criterion, please see that section later in this

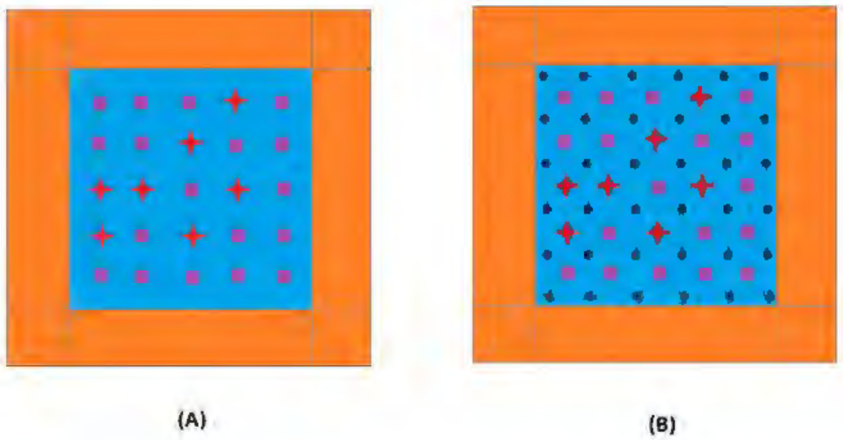


Figure 3.14: Dynamic domain decomposition in 2D

document.

If this exhaustion occurs for one region in the domain, then it is replenished with additional quadrature points. To see this, consider figure 3.14. In panel (A) all possible quadrature location have been used, either in the current batch (red stars) or from previous batches (purple rectangles). It may, however, still be the case that $x_* = \underset{x}{\arg\max}\mathbb{V}_{l|D}[l(x)\pi(x)]$ is still in this region, and hence the search algorithm would lead to this region. In order to facilitate the selection of a quadrature point in this region which are informative, additional sample locations must be added. Specifically, the distance between existing used sample points is halved, and in these new locations additional sample locations are added, as can be seen in figure 3.14 panel (B).

There are several noteworthy characteristics of our procedure that should be addressed. First is the consideration of when to add additional quadrature points. We have chosen, out of preference, that additional quadrature points will only be added when the previously allocated points have all been exhausted. While this allowed us to achieve our objective and lead to meaningful and significant results, it is possible to add these points earlier, say, when only a certain fraction of the original quadrature points remain instead of after they have all been exhausted. This remains an open question.

Furthermore, another consideration is where to add these quadrature points. We have chosen, out of computational convenience, to half the distance between existing sample locations to determine the locations for the replenishment of sample locations. Implications of this were not investigated.

Another consideration, which goes hand in hand with the consideration of where to place the replenishment of quadrature points, is how many quadrature points should be added. By halving the distance between existing quadrature points, we add significantly more new sample locations than the 25 with which the region was originally populated. We thought this was warranted because any region which exhausts its initially allocated 25 sample locations is of particular interest to our search algorithm, and hence should have the possibility of considering more points. However, this significant increase in quadrature points increases the computational cost of future quadrature point selections, especially when the replenishment must happen more than once.

All of these considerations, while not explicitly investigated in this research, will be discussed in the future work section of this document. We believe that these questions are of high importance, and may result in significant and large increases in efficiency.

Finally, one aspect that aided in the reduction of computational cost and also was crucial in the achievement of our objective, is in which regions replenishment occurred. It is particularly important that the addition on new sample locations only occurred in the regions where it was necessary, that is, in regions which exhausted its initially allocated sample locations. If any time one region exhausted its initially allocated sample location all regions would be replenished, this would unnecessarily increase the computational cost of regions which still had some of its initially allocated sample locations. For that reason, this was not done in our research.

It should be noted then, that since we decompose our domain into discrete sample locations, and that these sample locations are updated as required, our domain decomposition is dynamic. Also, since this domain decomposition was designed to specifically address our requirements to meet our objective, and other similar types of decomposition where not available in the literature, in the batch Bayesian quadrature literature this is novel and significant. Despite the fact that it is not the most significant contribution we make in this research, we believe that other fields with similar objectives could use our novel contribution.

3.4.5 Higher Dimensions

Our novel dynamic domain decomposition as outlined earlier in this section is applicable in higher dimensions. That is, nothing about of decomposition process would need to change in order to apply it to multiple dimensions.

Nevertheless, what is important to note is that, specifically for our purpose in the application of this dynamic decomposition, it makes applications in lower dimensions computationally feasible. In higher dimensions, computational cost becomes increasingly prohibitive. While this is not unlike other methods used in the literature to integrate intractable functions in high dimensions, for our specific objective it limits the number of dimensions for which this will allow us to reach our objective. Therefore, in order to be able to apply our novel quadrature method to higher dimensions, our dynamic domain decomposition may need to be refined. specifically, the limiting factor in the application to high dimensions is not our novel quadrature technique, but instead the suitability of the dynamic domain decomposition which we have successfully employed in low dimensions. This will be discussed in the future work section of this document.

3.4.6 Superiority of Future Uncertainty Sampling versus Uncertainty Sampling (Part 3)

Despite the fact that Future Uncertainty Sampling was discussed in section 3.2, followed by two subsection on the superiority of future uncertainty sampling over uncertainty sampling, this section highlights an additional aspect of the superiority of future uncertainty sampling building upon information from the concepts discussed in the sections since then, including the test functions, our novel dynamic domain decomposition, and our novel batch selection process.

To briefly reiterate the salient points, future uncertainty sampling minimizes the uncertainty of the integrand by selecting sample locations x'_* such that $\mathbf{x}'_* = \arg\min_{\mathbf{x}_*} \left[\max_{x} \mathbb{V}_{l|D}[l(x)\pi(x)]\right]$ where again $\mathbb{V}_{l|D}[l(x)\pi(x)]$ is as in equation 3.13, and x_* is to be understood as any point which has been evaluated as if it had been chosen as a quadrature point.

Two very significant benefits of using future uncertainty sampling instead of uncertainty sampling have already been discussed which already make the former superior to the latter. The first which was discussed is that future uncertainty sampling allows us the significant benefit of being able to select points simultaneously within a batch where previous techniques only allowed for the sequential selection of quadrature points within a batch. The other very significant consequence discussed is that by using future uncertainty sampling we can chose points which work better in harmony, and hence the posterior covariance of the GP is more efficiently reduced. Again, for a more thorough discussion, consult the appropriate sections of this research document.

A third significant benefit which has yet to be discussed of using future uncertainty sampling as opposed to uncertainty sampling or any other technique mentioned in the batch Bayesian quadrature literature is its ability to more quickly explore the search space while also reducing the posterior covariance of the GP more efficiently, and hence lead to superior estimates of the integrand.

To see this, it is now clear from our discussion of our dynamic domain decomposition and batch selection, that future uncertainty sampling requires less points within a partition in order to obtain lower posterior covariance of the GP. Therefore, it can select quadrature points within regions that other search techniques would not yet be able to explore due to their relative inefficiency.

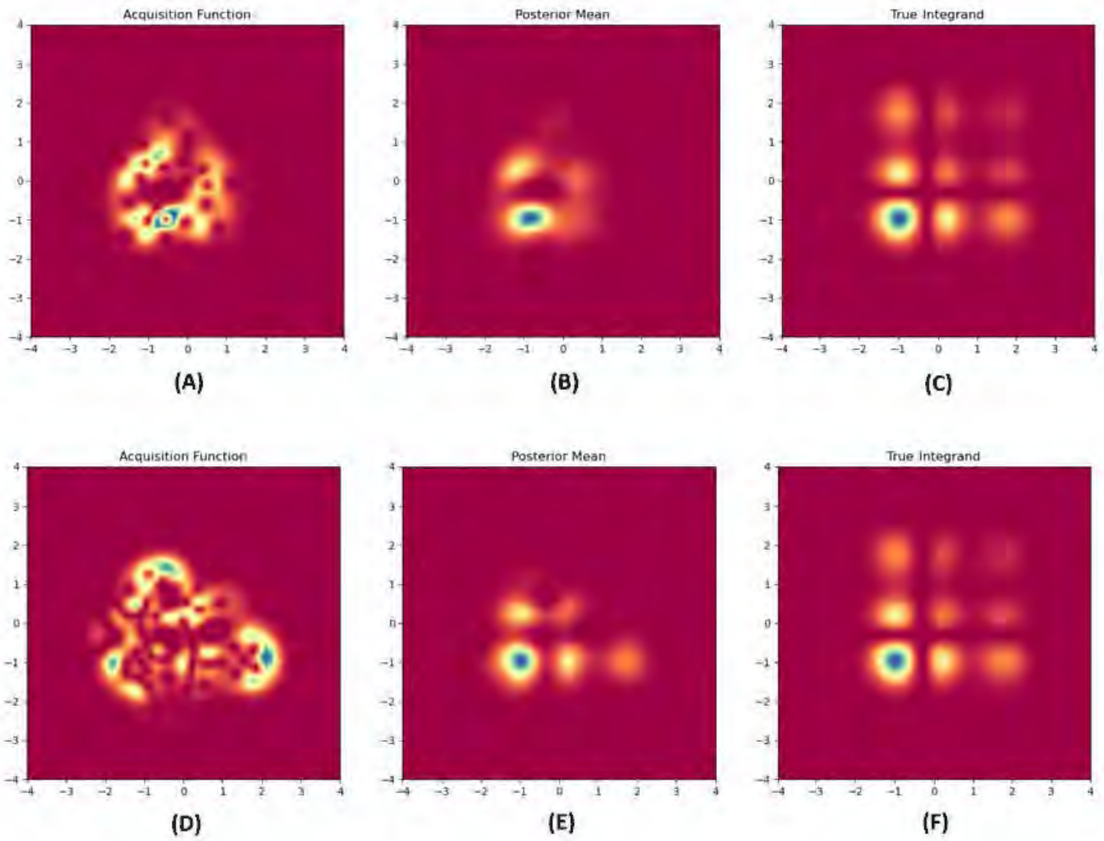


Figure 3.15: Synthetic test function and Gaussian process posterior in 2D

Pictorially this can be seen from graphs stemming from our numerical simulations. In figure 3.15, panel (A) and (D) represent the posterior covariance of the GP, where panel (A) is utilizing uncertainty sampling, and panel (D) is using our novel future uncertainty sampling. What is crucially important to note here is that our novel future uncertainty sampling has explored much more of the search space than uncertainty sampling. That is, the yellow near the center of panels (A) and (D) are much more diffused in panel (D) than in panel (A).

This has significant consequences for the estimates of the integrand. Panel (B) and panel (E) in figure 3.15 represent the posterior mean of the GP, that is, the estimate of the integrand. Furthermore, panel (C) and panel (F) in figure 3.15 represent the actual integrand, and are identical. Therefore, in order to see that future uncertainty sampling better estimates the integrand as compared to the most state-of-the-art batch quadrature techniques in the literature, we compare panel (B) to panel (C), and panel (E) to panel (F). It should be clear that panel (E) much more closely represents panel (F) than does panel (B) represent panel (C).

It should be noted here that figure 3.15 represents only a few batches, and hence only the beginning stages of estimation. Figure 3.16, on the other hand, has completed the estimation and reached the cessation criterion which will be discussed in a later section in this research document.

The test function being evaluated in figure 3.15 and figure 3.16 is the synthetic test function. As was discussed in the section dedicated to test functions, this test function has many fewer fluctuations that both the Ackley test function and the Weierstrass test function. Therefore, while our novel method allows for the more rapid and better estimation of the integrand, and hence also more robust estimates of the integral, by the time the cessation criterion is met, both our novel technique and those already explored in the literature achieve accurate estimates. While our novel method achieves the cessation criterion slightly sooner, the improvement of evaluating such types of functions is only modest. We will see that the next set of examples allows for much more significant improvements.

The general setup for figure 3.16 is very similar as for figure 3.15, where panels (A) through (C) represent the simulations for uncertainty sampling, panels (D) through (F) represent the simulations for future uncertainty sampling, panels (A) and (D) represent the posterior covariance of the GP, panels (B) and (E) represent the poste-

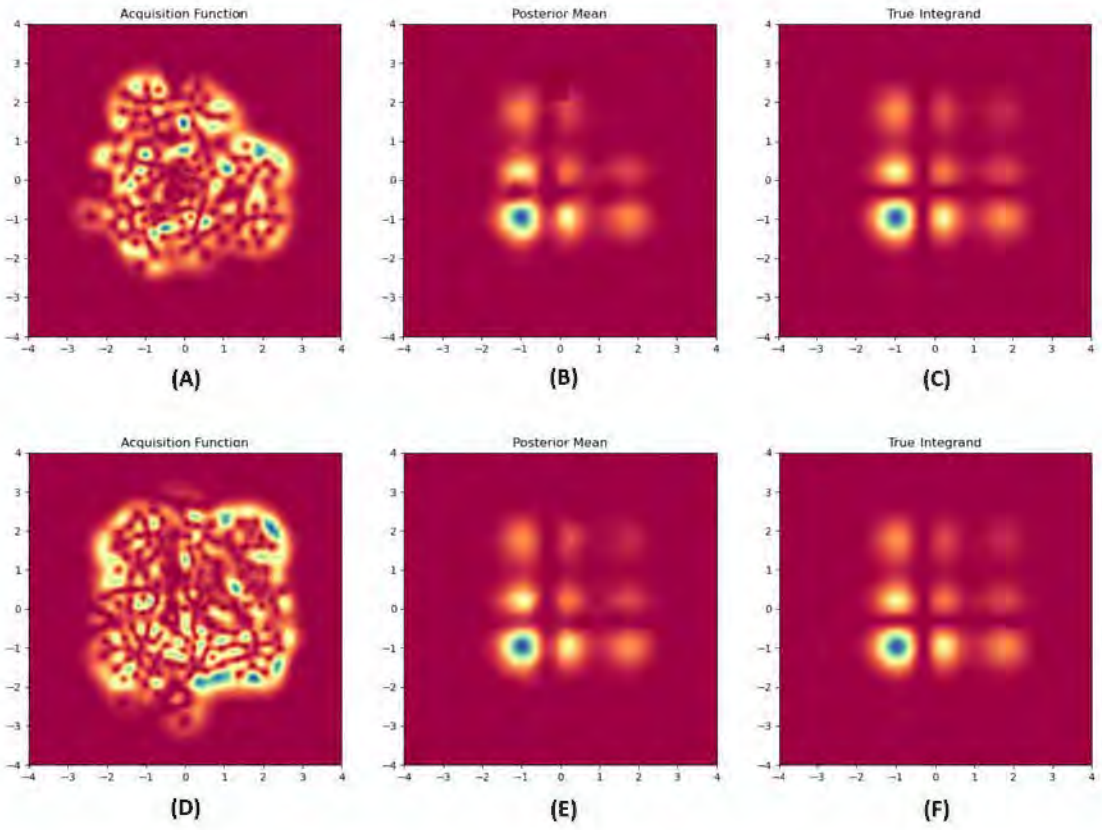


Figure 3.16: Synthetic test function and Gaussian process posterior in 2D

rior mean of the GP and hence the estimate of the integrand, and finally panels (C) and (F) represent the true integrand and hence are once again identical.

What is significant again is that our novel method explores the search space much more quickly and efficiently, again represented by the fact that the yellow in panel (D) is much more diffused that the yellow in panel (A). Also, our novel method estimates the true integrand much more closely, represented by the fact that panel (E) more closely represents panel (F) than does panel (B) represent panel (F). These graphs again represent just a few batches being selected and not yet reaching the cessation criterion. Nevertheless, these graphs show that even at the very beginning of the estimation process, our novel method significantly outperforms the methods in the literature. This is a significant improvement, since we aim to estimate the integrand

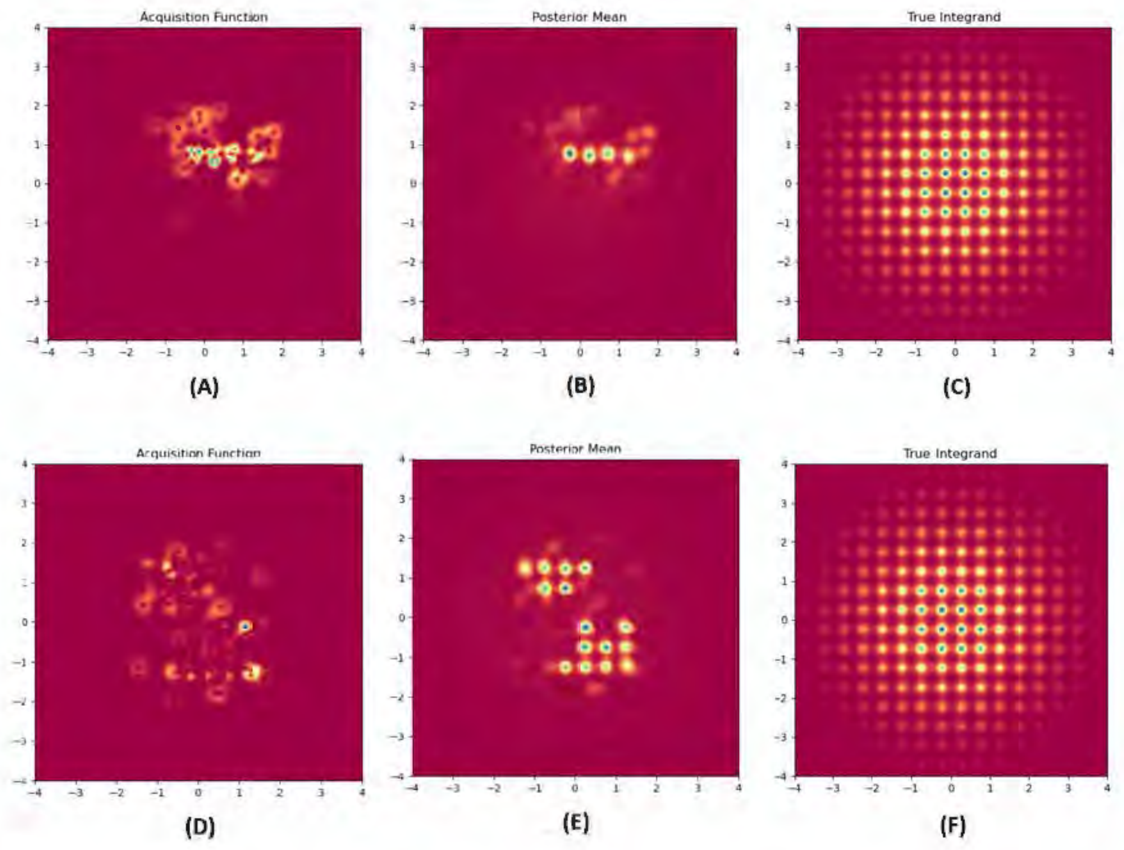


Figure 3.17: Weierstrass test function and Gaussian process posterior in 2D

with the fewest number of batches as possible.

One obvious distinction between figures 3.15 and 3.16 with figures 3.17 and 3.18 is that the latter now represent a different test function, namely the Weierstrass test function which has many more fluctuations than the synthetic test function. These fluctuations mean that there will be higher uncertainty in the Weierstrass test function estimates than in those for the synthetic test function. This larger uncertainty is represented by larger and more frequent oscillating posterior cavariances of the GP. Therefore, while the estimates using our novel method outperform those for the synthetic test function, the magnitude with which our method outperforms the already existing methods grows significantly in test functions which have larger and more frequent fluctuations, such as the Weierstrass test function and the Ackley test function.

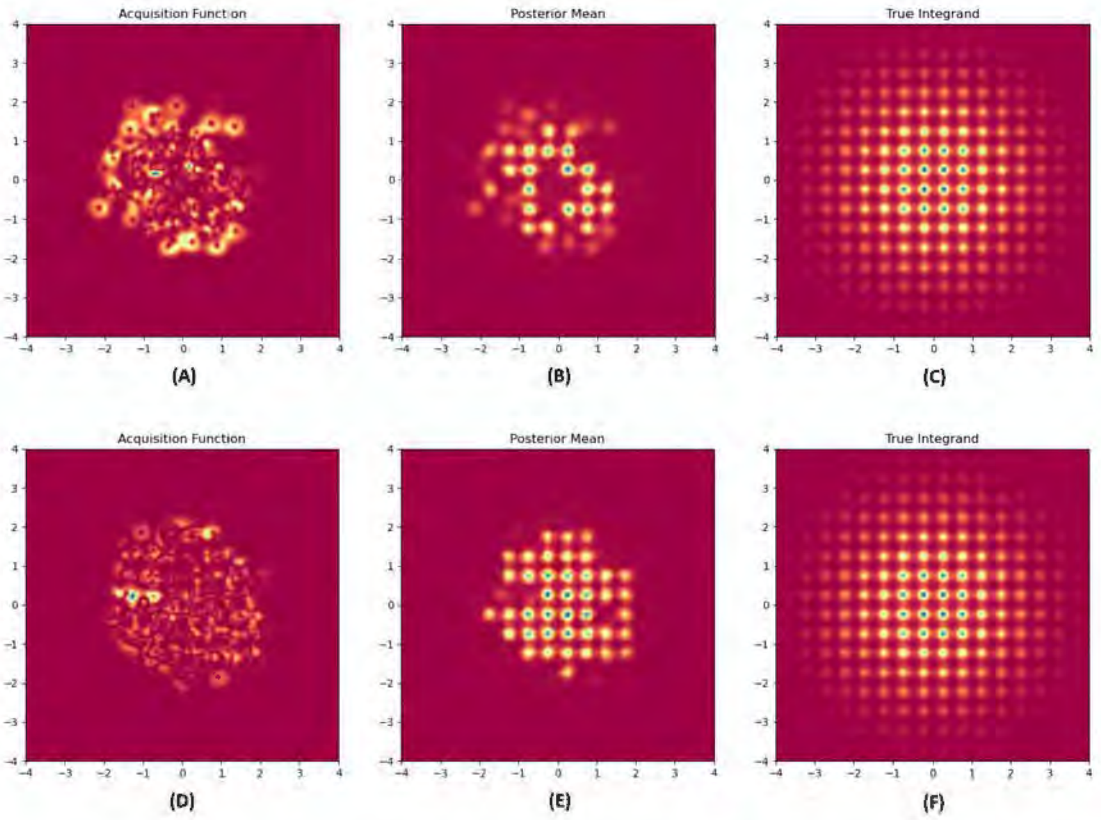


Figure 3.18: Weierstrass test function and Gaussian process posterior in 2D

What is especially important to note is that our technique continues to outperform even later in the estimation process, and can be seen in figure 3.18. Panel (E) which uses our novel technique looks much more like panel (F) than does panel (B) look like panel (C) despite having taken many more batches and therefore being much farther in the estimation process. This shows that our method not only will require fewer batches to reach the cessation criterion, but that even early on we have much better results which persist well into the estimation process. Again, this is particularly true for function which have many more and larger fluctuations, such as the Weierstrass test function depicted here, or the Ackley test function. Therefore, our method is much better for a large class of functions, and modestly better for all here. For exact numerical results, consult that section of this research document as well as the tables in the Appendix

3.5 Numerical Results

This section is broken down as follows: Since there are several novel contributions as well as contrasting applications of these contributions, first it will be noted which differences will be investigated in this section. Subsequently we will demonstrate the application of these novel contributions in both one dimension and two dimensions, and conclude with a discussion on the superiority of our novel contributions.

3.5.1 Differences to be Highlighted

It is important to note that Bayesian Quadrature utilizes a probabilistic model to induce two things, namely both 1) the functional form of the integrand, and 2) a probability distribution over the value of the integral. While both of these aspects play an integral role in accomplishing the objective of Batch Bayesian Quadrature (BBQ), those two components can be distinguished from each other. Specifically, a novel contribution to the existing BBQ literature can arise from either an improvement in determining the functional form of the integrand, or by better estimating the integral.

Since our process assumes that evaluating the integrand is costly, we devise a novel technique which reduces this cost. Specifically, we not only employ batch methods to select sample locations for our quadrature points, but we devise a novel method which allows us to more efficiently select the most informative quadrature points. That is, given

$$Z = \int l(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$$

and our choice to place the GP on the likelihood, l. Conditioned on samples $x_d =$

$\{x_1, ..., x_N\}$ and corresponding functional values $l(x_d)$, we have

$$l \mid D \sim GP(m_D, C_D)$$
$$m_D(x) = \mu(x) + K(x, x_d)K^{-1}(x_d, x_d)(l(x_d) - \mu(x_d))$$
$$C_D(x, x') = K(x, x') - K(x, x_d)K^{-1}(x_d, x_d)K(x_d, x')$$

where

$$K(x, x') = \lambda^2 exp(-\frac{1}{2}\frac{(x - x')^2}{\sigma^2})$$

such that the output length-scale λ and input length-scale σ control the standard deviation of the output and the autocorrelation range of each function evaluation, respectively, and will be jointly denoted by $\theta = \{\lambda, \sigma\}$.

Furthermore, since utilizing a standard GP prior would ignore the range and non-negativity of l. We therefore define $\tilde{l}(x) = \sqrt{2(l(x) - \alpha)}$, where α is a small scalar, in our case $\alpha = 0.8$ x min $l(x_d)$. We take a GP prior on $\tilde{l}(x) : \tilde{l}(x) \sim GP(0, K)$, for which the posterior is

$$p(\tilde{l} \mid D) = GP(\tilde{l}; \tilde{m}_D(\cdot), \tilde{C}_D(\cdot, \cdot))$$
$$\tilde{m}_D(x) = K(x, x_d)K^{-1}(x_d, x_d)\tilde{l}(x_d)$$
$$\tilde{C}_D(x, x') = K(x, x') - K(x, x_d)K^{-1}(x_d, x_d)K(x_d, x')$$

which leads to a GP whose marginal distribution for any $l(x)$ is a non-central χ^2 with one degree of freedom. Therefore, we perform a local linearization of the form

$$f : \tilde{l} \mapsto l = \alpha + \frac{1}{2}\tilde{l}^2$$

This linearization around $\tilde{l}_0$ results in

$$l(x) \simeq f(\tilde{l}_0) + f'(\tilde{l}_0)(\tilde{l} - \tilde{l}_0)$$

We choose $\tilde{l} = \tilde{m}_D$ such that

$$\begin{aligned} l(x) &\simeq (\alpha + \tfrac{1}{2}\tilde{m}_D(x)^2) + \tilde{m}_D(x)(\tilde{l}(x) - \tilde{m}_D(x)) \\ &\simeq \alpha - \tfrac{1}{2}\tilde{m}_D(x)^2 + \tilde{m}_D(x)\tilde{l}(x) \end{aligned}$$

and therefore l is an affine transformation of $\tilde{l}$, which results in the following posterior:

$$\begin{aligned} p(l \mid D) &\simeq GP(l;\, m_D^L(\cdot),\, C_D^L(\cdot,\cdot)) \\ m_D^L(x) &= \alpha + \tfrac{1}{2}\tilde{m}_D(x)^2 \\ C_D^L(x,x') &= \tilde{m}_D(x)\tilde{C}_D(x,x')\tilde{m}_D(x') \end{aligned}$$

Since $\tilde{m}_D$ and $\tilde{C}_D$ are mixtures of un-normalized Gaussians K, m_D^L and C_D^L are also mixtures of un-normalized Gaussians. Rasmussen et al. (2006) have shown that this procedure leads to analytic results for

$$\mathbb{E}_{l|D}[Z] = \int m_D(x)\pi(x)dx$$

and therefore so does

$$\mathbb{E}_{l|D}[Z] = \int m_D^L(x)\pi(x)dx \tag{3.18}$$

An important distinction can be made about how to select quadrature points. We can select points which either better approximate the integral or the integrand. Since both uncertainty sampling and future uncertainty sampling consider

$$\mathbb{V}_{l|D}^L[l(x)\pi(x)] = \pi(x)^2\tilde{C}_D(x,x)\tilde{m}_D(x)^2 \tag{3.19}$$

instead of

$$\langle \mathbb{V}_{l|D,l(x)}[Z] \rangle = \int \mathbb{V}_{l|D,l(x)}[Z] N(l(x);\, m_D(x),\, C_D(x,x)) \mathrm{d}l(x) \tag{3.20}$$

a compelling argument can be made for selecting points which better determine the integrand. As we will see, since the integrand obviously plays a large role in determining the integral, better approximating the integrand can lead to more robust estimates of the integral.

Specifically, we can choose quadrature points to either

$$min \left| \mathbb{E}_{l|D}[Z] - Z \right| \tag{3.21}$$

or

$$min \left| m_D^L - l \right| \tag{3.22}$$

where equation 3.19 would be better accomplished considering equation 3.22, and equation 3.20 would be better accomplished by considering equation 3.21. Since both uncertainty sampling and future uncertainty sampling consider equation 3.19, we will focus our evaluation on equation 3.22, but we will present both for completeness of discussion.

3.5.2 One Dimension

A common evaluation criterion in estimating an integral is to determine the absolute error an estimation is from the truth. In our case that is $\left| \mathbb{E}_{l|D}[Z] - Z \right|$. Since our test function used for our integrand are analytically intractable in traditional integration, we evaluate the ground truth using a fine grid. To accomplish this, our evaluations can either only take place in one dimension or two dimensions. Therefore, we restrict ourselves here to only two dimensions.

Similarly, while there are a plethora of differing evaluation techniques regarding the approximation of an unknown function, a common criterion used in estimating the accuracy in determining the function in black box scenarios is the sum of squared error (SSE) from the estimate to the true function. We apply this evaluation criterion in this study to determine $min\left|m_D^L - l\right|$.

For the second evaluation criterion it should be noted that in quadrature, especially after the model is updated, using the SSE for the quadrature points would result in no information gleaned. To see this, it is important to note that when quadrature points are selected using the full GP model, the posterior mean of the GP is used for functional values of those quadrature points in the batch selection process.

Specifically, in Batch Bayesian Quadrature (BBQ) several samples are taken in parallel, i.e., each $\mathbf{x}_*$ is a batch of sample locations. To accomplish this, the probabilistic model of BQ is used to guide exploration of a search space by defining an acquisition function to be maximized. Two such methods are Kriging Believer and Local Penalization.

In Kriging Believer (Ginsbourger et al., 2010), instead of updating the BQ model after every sample, the functional value of a sample location is set equal to the posterior mean, then the GP model is updated before selecting a subsequent sample location. This process is repeated until the batch of sample locations chosen is sufficiently large, and then the evaluations of those sample locations are made and the BQ model is updated. Therefore, if the SSE were to be calculated in the quadrature point selection process, all of these value would be equal to the posterior mean of the GP, and hence no information could be gained. That is, any point selection method, including uncertainty sampling and future uncertainty sampling would use as the functional evaluation of the quadrature points the same posterior mean of the GP, and hence one could not evaluate the superiority of one selection method over the other.

Similarly, unlike Kriging Believer which utilizes the entire GP model, Local Penalization (Gonzáles et al., 2016; Wagstaff, 2018A) only directly modifies the acquisition function around each selected point. While this technique also allows for the selection of a batch of sample locations before those locations are evaluated and the BQ model is updated, the obstacle of not being able to compare point selection techniques remains. That is, despite not utilizing the posterior mean of the GP as the functional evaluations of quadrature points, the posterior mean of the GP remains stationary as in Kriging Believer. Therefore, using this quadrature point selection method, regardless of whether coupled with uncertainty sampling or future uncertainty sampling, would not allow us to compare point selection algorithms.

A consequence of this process is therefore that when the model is updated, the integrand is evaluated at the selected quadrature points, where again this step is considered costly and a driving aspect of our research. Subsequently, these functional evaluations of the integrand at these quadrature points are used to update the GP model. Hence the posterior mean of the GP will traverse through these points, making their error to the true equal to zero for all sampled locations. This, in turn, leads to no information of the accuracy of one technique over the other.

To circumvent this obstacle, we define 2,000 equally spaced points in our domain in one dimension, and a grid of 2,000 equally spaced points in two dimensions as our evaluation points. While some of these points may overlap with the quadrature points selected, especially in large batch sizes and if many batches are selected, the vast majority will not. Therefore, there may be some evaluation points for which we gain no information in the comparison of quadrature point selection methods, but for many others we will.

It should be noted that the selection of 2,000 equally spaced evaluation points in one dimension and a grid of 2,000 equally spaced evaluation points in two dimensions was a determination suitable for our investigation. While it allowed us to perform the necessary comparisons, the ideal number of points to select remains an open question, but likely not one of high importance. However, what is important to note is that the selection of this many evaluation points, and the calculation of the sum of squared errors for all of them, will lead to rather large numbers even when there are only minor deviations is the estimated function to the true function. While these figures may be unsightly, the potentially large fluctuations allows us to notice even small differences between either both techniques, or when even minor deviations occur between the estimated function and the true function.

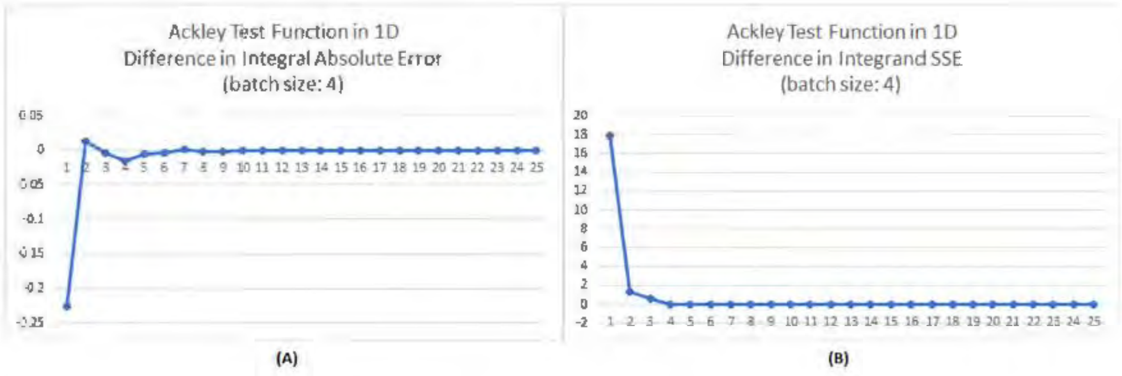


Figure 3.19: Ackley test function in 1D and batch size of 4

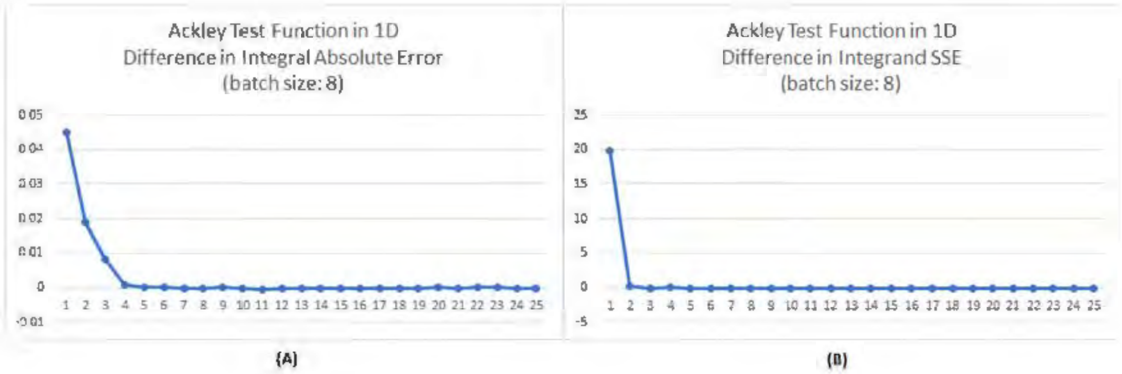


Figure 3.20: Ackley test function in 1D and batch size of 8

In figure 3.19, panel (A) represents the difference in the absolute error of the estimated integral to the true integral value determined as stated above. Panel (B), on the other hand, depicts the difference in SSE of the estimated function for the integrand versus the true function. Both of these graphs represent the results where batches of size 4 are selected, and a total of 25 batches were taken. Furthermore, these estimates represent the average of ten runs of this process. Each subsequent graph in this section follows the same description, where either the test function changes, the size of the batch changes or the dimension our method is being applied to changes. Note the title of each graph for details.

The calculations these graphs represent is that first quadrature points were selected using uncertainty sampling and Kriging Believer, but the model was not updated once the batch was completed. Subsequently, the point locations and estimated functional values were stored, and the posterior of the GP was reset so that the model to be evaluated appeared identical to when the points were selected using uncertainty sampling and Kriging Believer commenced. Then a batch of quadrature points was selected using future uncertainty sampling and Kriging Believer. Once the requisite number of quadrature points was selected for the batch, then the model was evaluated separately using these two batches of points.

Subsequently, the estimated integral values and the posterior mean of the GP were calculated and compared to the true values for each method. That is, first the estimated integral value and posterior mean of the GP were calculated for uncertainty sampling and Kriging Believer, and subsequently the estimated integral value and posterior mean of the GP were calculated for future uncertainty sampling and Kriging Believer. Therefore, two values were calculated for each technique, both representing different measures.

These pairs of values were then separately compared to the ground truth in order to ascertain an error for each value. The pair of error values for future uncertainty sampling and Kriging Believer, which again represent some of our novel contributions, were individually subtracted from their counterparts resulting from uncertainty sampling and Kriging Believer. Therefore, if the error for either measure was larger for the results arising from uncertainty sampling and Kriging Believer than those arising from future uncertainty sampling and Kriging Believer, then this difference would be positive. Conversely, if the error for either measure was smaller for the results arising from uncertainty sampling and Kriging Believer than those arising from future uncertainty sampling and Kriging Believer, then this difference would be negative.

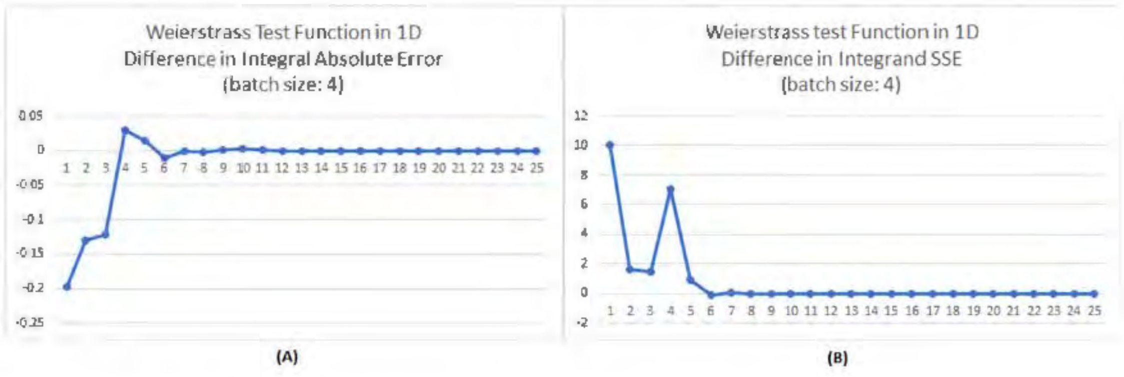


Figure 3.21: Weierstrass test function in 1D and batch size of 4

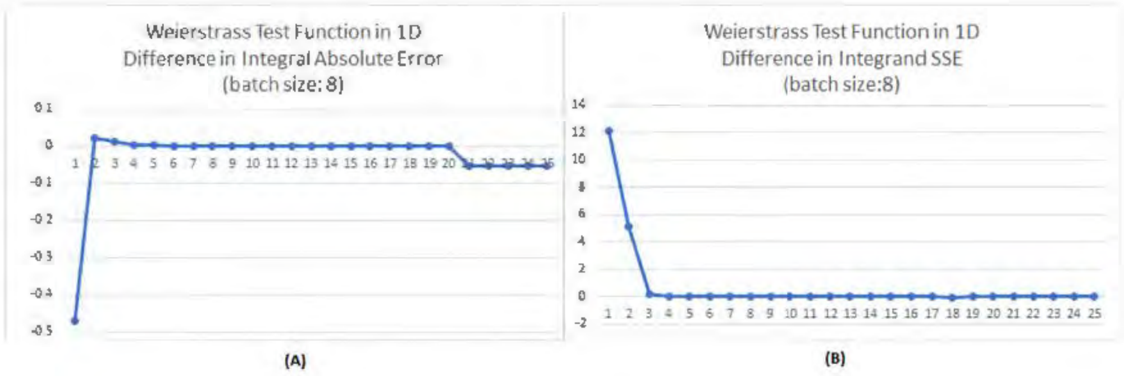


Figure 3.22: Weierstrass test function in 1D and batch size of 8

That is, panel (A) in figure 3.19 to 3.24 is calculated by

$$\left|\mathbb{E}_{l|D;KB,US}[Z] - Z\right| - \left|\mathbb{E}_{l|D;KB,FUS}[Z] - Z\right| \tag{3.23}$$

where Z, l, and D are as previously defined, and KB, US, and FUS stand for Kriging Believer, uncertainty sampling, and future uncertainty sampling, respectively.

Panel (B) in figure 3.19 to 3.24 is calculated by

$$\sum_{x}[m^{L}_{D,KB,US}(x)\pi(x) - l(x)\pi(x)] - \sum_{x}[m^{L}_{D,KB,FUS}(x)\pi(x) - l(x)\pi(x)] \qquad (3.24)$$

where m^{L}_{D}, π, l, D, KB, US, and FUS are as previously defined, and x represents the 2,000 equally spaced points on which the error estimates are to be calculated as also previously defined. It is important to note that since our research is driven by the desire to better estimate the integrand, we place higher priority on panel (B) than on panel (A). We will also see that this leads to more robust integral estimates.

First, note that in panel (B) in figures 3.19 to 3.24 our estimate are always initially superior to the techniques which already exist in the BBQ literature, that is our estimates are initially always positive. Again, this is calculated using equation 3.24. Specifically, when we investigate test functions with frequent and often large fluctuations in magnitude, our novel method reduces the posterior covariance of the GP more quickly, allowing us to explore our search space more efficiently. This efficiency allows us to better estimate the integrand in neighborhoods where prior techniques have yet to explore. It is due to this more efficient exploitation and quicker exploration that gives us superior results in estimating highly fluctuating integrands. It is particularly those types of integrands which prior techniques in the literature have done particularly poorly in estimating.

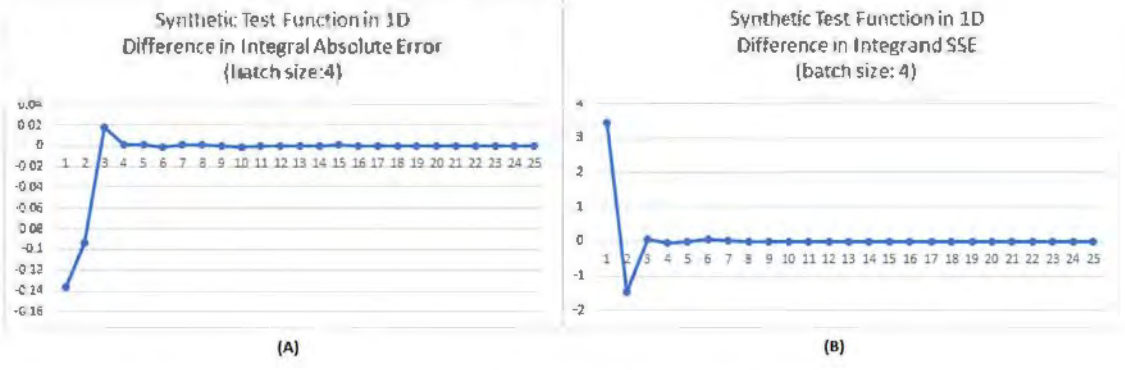


Figure 3.23: Synthetic test function in 1D and batch size of 4

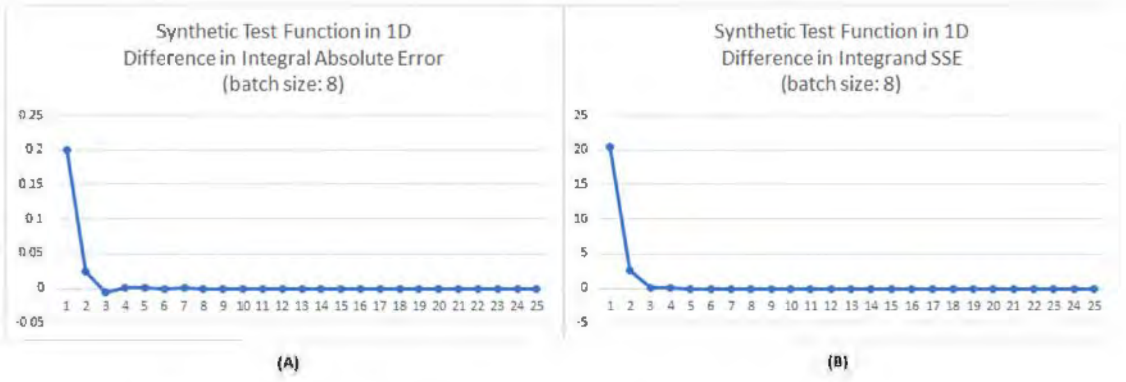


Figure 3.24: Synthetic test function in 1D and batch size of 8

Second, regarding panel (A) in figures 3.19 to 3.24, this is calculated using equation 3.23. While it appears that our novel method sometimes does better at estimating the integral, and sometimes worse, this can be deceiving. Since our novel method better estimates the integrand, while our estimates for the integral may have a larger error in comparison to previous techniques, our estimates of the integral are nevertheless more robust. To see this, consider figure 3.25. In the top part of panel (A) we begin by showing that the true integrand is given by line A, line B is the estimate of the integrand using the techniques in the literature, and line C is the estimate of the integrand using our novel technique. Here we see that the SSE estimate of our estimate would be smaller. In the bottom half of panel A we indicate that the integral of our estimate of the integrand given by line C would clearly be smaller that the true integral value given by line A. On the other hand, we use the case where the estimate of the integrand using previous techniques in the literature given by line B coincidentally happens to have the same integral value as that of the true integrand A despite clearly representing very different functions. For the actual numerical results using well-known test functions, see the tables at the end of this document.

As we continue our quadrature point selection and model updating process, these results can change, as can be seen in going from panel (A) to panel (B) in figure 3.25. We again emphasize that our estimates of the integrand are superior. That is, as in panel (A), the SSE of our estimate is smaller that the SSE using the techniques already in the literature, as can be seen in panel (B) in figures 3.19 to 3.24. The error of our estimate of the integral as well as the error of the estimate of the integral using techniques already in the literature both decrease in the top part of panel (B), while our errors continue to be smaller. In the bottom part of panel (B), while our estimate of the integral still has error, the error now is smaller that that of the techniques previously used in the literature.

Finally, as we progress from panel (B) to panel (C), we can once again see in the top part of panel (C) that the SSE for our estimate of the integrand decreases as well as that for the estimate of the integrand using techniques previously shown in the BBQ literature. What is striking now, however, is that the SSE for our estimate is significantly smaller than that of the other estimates, but the integral value has more error for our estimate that that of the other methods. This can be seen in the bottom half of panel (C) where the integral estimate of the techniques in the literature again equals the true integral value, however the integral estimate using our novel technique is again lower than the true integral value.

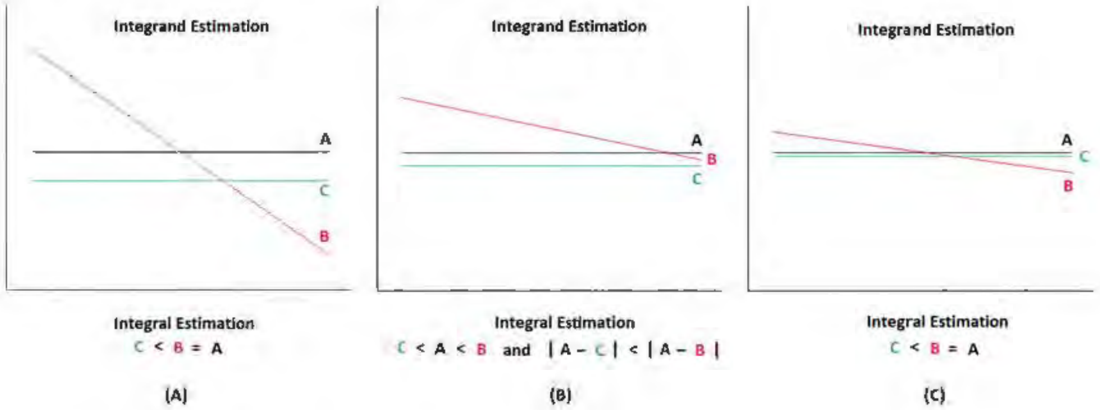


Figure 3.25: Integral vs. integrand estimation

It has now been shown that we not only estimate the integrand better using our novel technique, but that our technique therefore also has more robust estimates of the integral. That is, our estimate of the integral converges better to the true integral value that that of the BBQ techniques already mentioned in the literature, even if our integral value at times will have a larger error. Therefore, despite the fact that panel (A) in figures 3.19 through 3.24 show a negative value since the driving consideration of our research is to better determine the integrand, our novel method is superior since any larger errors that those of previous techniques are merely by chance. Again, for the actual numerical results supporting these conclusions using well-known test functions, see the tables at the end of this document.

3.5.3 Two Dimensions

Our novel method is applicable not just in one dimension, but in multiple dimensions too. In fact, it is applicable even in high dimensional problems, however the suitability of BBQ in these situations is questionable. This will be further discussed in the section on the cessation criterion. Here we show that the results produce similar results in two dimensions, and can therefore be extrapolated into multiple dimensions.

It should be noted here that much of this discussion uses the discussion in one dimension as a pretext, and therefore will proceed very similarly to that discussion. However, it should also be noted that there are differences between one dimension and two dimensions that are relevant to our research, including those already discussed in the domain decomposition section, and will be discussed in the cessation criterion section.

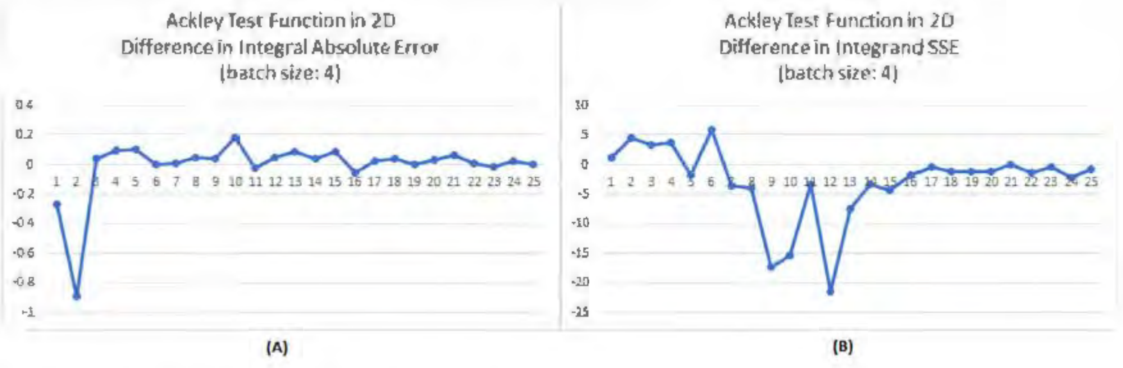


Figure 3.26: Ackley test function comparisons

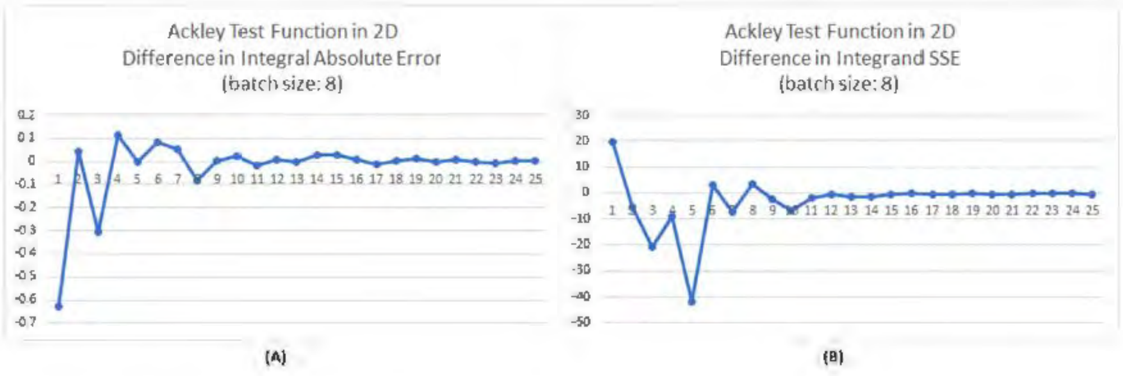


Figure 3.27: Ackley test function comparisons

Similar to the case in the discussion on the numerical results in one dimension above, figures 3.26 through figure 3.31, panel (A) represents the difference in the absolute error of the estimated integral to the true integral value determined as stated above, and panel (B) depicts the difference in SSE of the estimated function for the integrand versus the true function. Both of these graphs represent the results where batches of quadrature points are selected, and a total of 25 batches were taken. Furthermore, these estimates represent the average of ten runs of this process. Each subsequent graph in this section follows the same description, where either the test function changes between the Ackley test function, the Weierstrass test function, or the synthetic test function, or the size of the batch changes from either size 4 or 8. Note the title of each graph for details. Also, for specific numerical results see the tables section of this document.

The calculations these graphs represent is similar to that for one dimension. The first quadrature points were selected using uncertainty sampling and Kriging Believer, but the model was not updated once the batch was completed. Subsequently, the point locations and estimated functional values were stored, and the posterior of the GP was reset so that the model to be evaluated appeared identical to when the points were selected using uncertainty sampling and Kriging Believer commenced. Then a batch of quadrature points was selected using future uncertainty sampling and Kriging

Believer. Once the requisite number of quadrature points was selected for the batch, then the model was evaluated separately using these two batches of points.

Subsequently, the estimated integral values and the posterior mean of the GP were calculated and compared to the true values for each method. That is, first the estimated integral value and posterior mean of the GP were calculated for uncertainty sampling and Kriging Believer, and subsequently the estimated integral value and posterior mean of the GP were calculated for future uncertainty sampling and Kriging Believer. Therefore, two values were calculated for each technique, both representing different measures.

These pairs of values were then separately compared to the ground truth in order to ascertain an error for each value. The pair of error values for future uncertainty sampling and Kriging Believer, which again represent some of our novel contributions, were individually subtracted from their counterparts resulting from uncertainty sampling and Kriging Believer. Therefore, if the error for either measure was larger for the results arising from uncertainty sampling and Kriging Believer than those arising from future uncertainty sampling and Kriging Believer, then this difference would be positive. Conversely, if the error for either measure was smaller for the results arising from uncertainty sampling and Kriging Believer than those arising from future uncertainty sampling and Kriging Believer, then this difference would be negative.

Panel (A) in figure 3.26 to 3.31 is calculated by

$$\left|\mathbb{E}_{l|D;KB,US}[Z] - Z\right| - \left|\mathbb{E}_{l|D;KB,FUS}[Z] - Z\right|$$

as in equation 3.23, where Z, l, and D are as previously defined, and KB, US, and FUS stand for Kriging Believer, uncertainty sampling, and future uncertainty sampling, respectively.

Panel (B) in figure 3.26 to 3.31 is calculated by

$$\sum_x [m^L_{D,KB,US}(x)\pi(x) - l(x)\pi(x)] - \sum_x [m^L_{D,KB,FUS}(x)\pi(x) - l(x)\pi(x)]$$

as in equation 3.24, where m^L_D, π, l, D, KB, US, and FUS are as previously defined, and x represents the 2,000 equally spaced points on which the error estimates are to be calculated as also previously defined. Again, for specific numerical results see the tables section of this document.

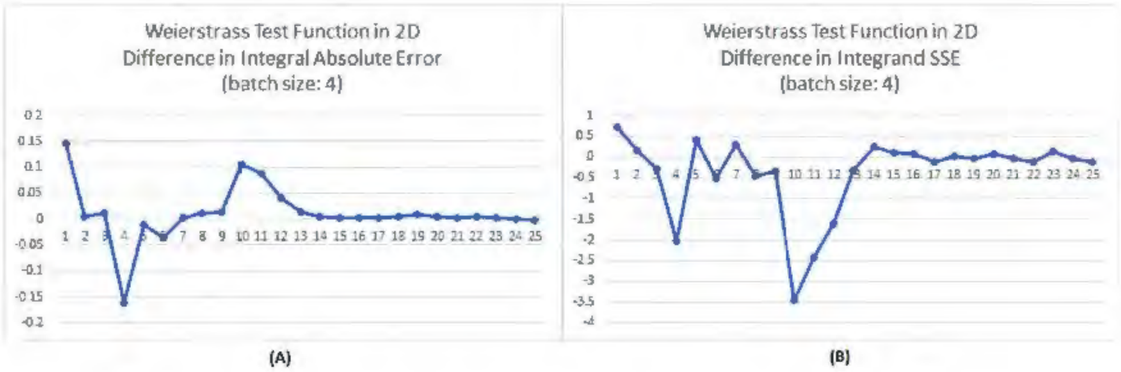


Figure 3.28: Weierstrass test function comparisons

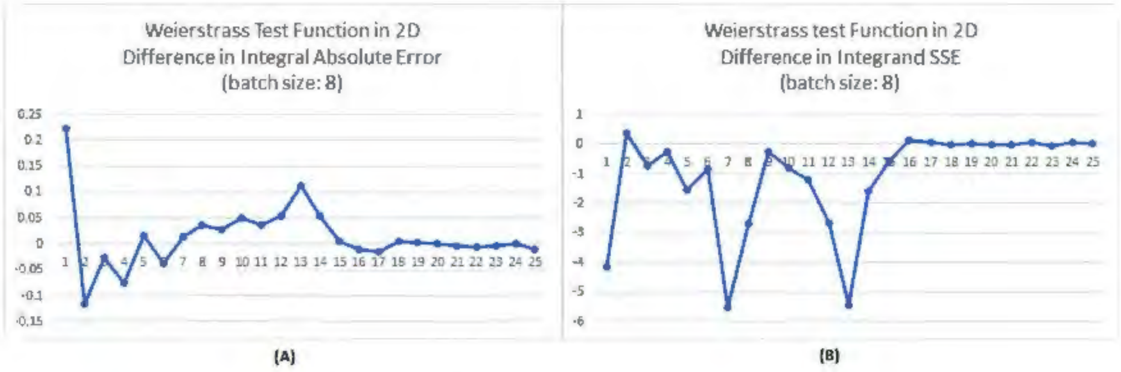


Figure 3.29: Weierstrass test function comparisons

Again, it is important to note that since our research is driven by the desire to better estimate the integrand, we place higher priority on panel (B) than on panel (A). Also, particularly noteworthy is the fact that a batch size, if applied in two dimensions as opposed to one dimension, will result in much sparser quadrature point allocations. This will not only affect the cessation criterion discussed later in this research, but also the rate at which the estimate converges with the true value for the integrand and integral. Furthermore, in multiple dimensions, when fluctuations can span a larger space, it is reasonable that fewer quadrature points points will not be as efficient in reducing the posterior covariance of the GP as is the case in either fewer dimensions of one dimension. Since our novel process reduces the posterior covariance of the GP more efficiently, and therefore has a stronger push to explore the search space than the methods which already exist in the literature, it is possible that such further exploration is premature. Should more quadrature points within a region be more prudent that fewer, then the previously existing methods in the literature will achieve superior results as can be seen in select panels in the graphs in this and the previous sections, as well as in other potential test functions with appropriate dimensionalities and batch sizes. Since in certain circumstances the integrand is assumed to be unknown, which method will obtain superior results may be indeterminable *a priori*.

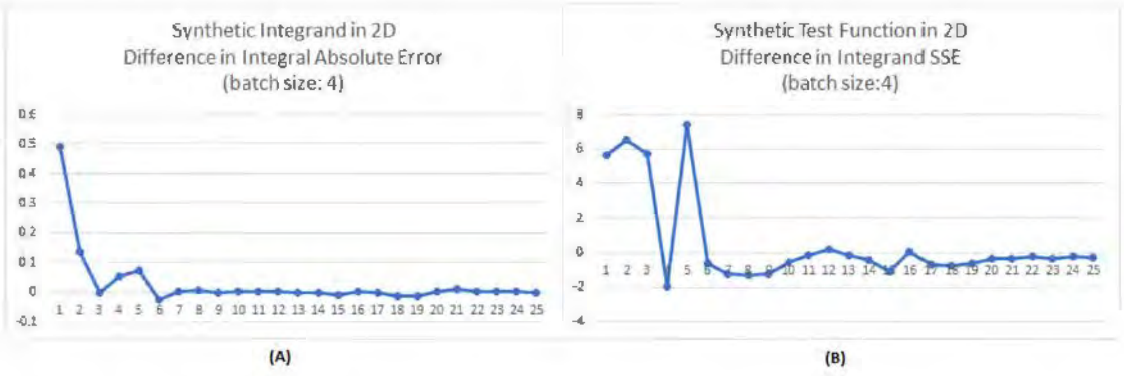


Figure 3.30: Synthetic test function comparisons

3.6 Cessation Criterion

When performing the search algorithm for identifying points within a batch, and also subsequently updating the model and filling subsequent batches, an eventual question

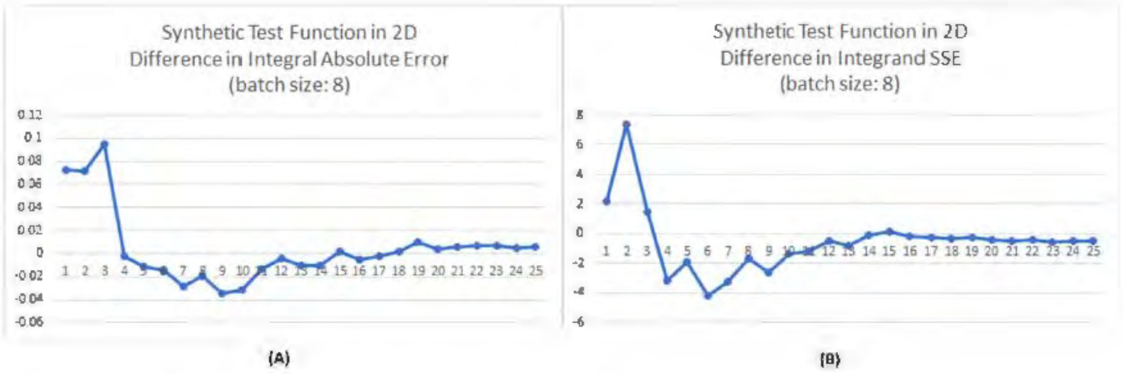


Figure 3.31: Synthetic test function comparisons

that must be posed is when to end the process. There are several potential cessation criterion which can be used. For example, given $Z = \int l(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$, since standard BQ determines a variance for its integral estimate, we have

$$\mathbb{V}_{l|D}[Z] = \int\int C_D(x, x')\pi(x)\pi(x')dxdx'$$

where $C_D(x, x') = K(x, x') - K(x, x_d)K^{-1}(x_d, x_d)K(x_d, x')$ and the scaled Gaussian covariance K(x, x') $= \lambda^2 \exp(-\frac{1}{2}\frac{(x-x')^2}{\sigma^2})$ such that the output length-scale λ and input length-scale σ control the standard deviation of the output and the autocorrelation range of each function evaluation, respectively and jointly denoted by $\theta = \{\lambda, \sigma\}$, $x_d = \{x_1, ..., x_N\}$ represent the samples taken, leading to $D = \{x_d, l(x_d), \theta\}$, and l represents the likelihood and π represents the prior as previously defined (Rasmussen et al., 2006).

Previous applications have set a stopping criterion of $\sqrt{\mathbb{V}_D[Z]} = 0.015$, and stopped once below this threshold (Garnett et al., 2010). That is, the posterior variance of the integral estimate was used when the driving determinant was the estimate of the integral itself. As was previously shown, there are ways in which better informed decision can be made in the quadrature point selection process. Furthermore, when the driving indicator of the search algorithm was not the integral estimate or its posterior variance, other cessation criterion have been implemented.

One possibility in selecting sample locations and cumulatively batches would be to follow Osborne et al. (2012B) in minimizing the expected entropy of the integral by selecting $x_* = \arg\min_x \langle \mathbb{V}_{l|D,\,l(x)}[Z] \rangle$, where

$$\langle \mathbb{V}_{l|D,\,l(x)}[Z] \rangle = \int \mathbb{V}_{l|D,\,l(x)}[Z] N(l(x);\, m_D(x),\, C_D(x,x))\mathrm{d}l(x)$$

as was given in equation 3.12 where, in addition to what has been previously defined, we have $m_D(x) = \mu(x) + K(x, x_d)K^{-1}(x_d, x_d)(l(x_d) - \mu(x_d))$ where $\mu(x)$ is the partial parameterization of the GP given by $l \mid D \sim GP(m_D, C_D)$.

This approach has difficulties. One of which for our application is the high computational cost associated with this method. Again, whenever the integrand is considered expensive to evaluate, selecting points which require the evaluation of the integrand, which here occurs through the approximation of the integral and hence its variance, is extremely costly.

Another possibility would be to target the uncertainty in the integrand, where for our warped integrand we have

$$\mathbb{V}^L_{l|D}[l(x)\pi(x)] = \pi(x)^2 \tilde{C}_D(x,x)\tilde{m}_D(x)^2$$

where $\tilde{m}_D(x) = K(x, x_d)K^{-1}(x_d, x_d)\tilde{l}(x_d)$ and
$\tilde{C}_D(x, x') = K(x, x') - K(x, x_d)K^{-1}(x_d, x_d)K(x_d, x')$ where $\tilde{l}(x) = \sqrt{2(l(x) - \alpha)}$ and $\alpha = 0.8$ x $\min\ l(x_d)$, and a local linearization of the form $f : \tilde{l} \mapsto l = \alpha + \frac{1}{2}\tilde{l}^2$ to achieve a GP and hence also facilitate analytic results, as was previously stated and discussed regarding equation 3.13. Given our constraints and prerogatives, it is this criterion which we consider in our acquisition of quadrature points and batches, and therefore also as the consideration for our investigation into the most suitable cessation criterion.

First, it should be noted, as the work by Gunter et al. (2014) correctly stated, that uncertainty sampling reduces the entropy of the GP to $p(l)$ rather than the true intractable distribution, and that the computation of equation 3.12 is considerably more expensive than that of equation 3.13.

Also, similar to our application of the square-root transformation in the warping of the GP on the likelihood, halving the dynamic range of the function we model provides both benefits and pitfalls. On one hand, this warping mitigates typically large variations in the likelihood, and extends the autocorrelation range of the GP yielding improved predictive power when extrapolating away from the data. On the other hand, the model is overconfident away from the data causing the BQ variance to be erroneously low, making the typical threshold unsuitable as a stopping criterion.

Previous applications of BBQ have sidestepped this issue by simply setting a fixed computational budget (Wagstaff, 2018A). That is, neither the posterior variance of the integral estimate nor the posterior covariance of the GP were considered, but instead a predetermined number of batches were selected regardless of the number of quadrature points per batch. Once this predetermined number of batches was reached, the search algorithm ceased.

While this certainly is a method to limit computational cost, better informed cessation criterion can be formulated. To do so, we should note two important consideration in accomplishing this. First, the main considerations in these kind of black box applications of integration usually consist of the combination of finding the most accurate estimate of the integral, coupled with having the lowest uncertainty in this estimate, and the fewest evaluations of the integrand in this process. While there are certainly other consideration which must be made, including the dimensionality of the function to be integrated, the size of the search space, computational costs, etc., most if not all studying in such application take the former three considerations into account for obvious reasons. In our application in particular, the integrand is considered to be extremely costly to evaluate, and hence this criterion is of very high importance.

Second, a driving aspect which allowed us to implement our novel contribution, which in the previous section were shown to lead to superior results, is our novel dynamic domain decomposition. This decomposition allowed us to significantly reduce computational cost while keeping our search space informative for our search algorithm. However, of particular importance here again is the discretization of the search space. This discretization allowed us to eliminate sample locations which would not have yielded significant new information in our search space, however the elimination of these points therefore also set a lower limit on the reduction in entropy.

Nevertheless, as the combination of the three main driving factors articulated above will show, given our implementation of our novel method and novel domain decomposition, we not only had significant numerical results in the previous section, but we are also able to determine for the first time ever in the literature on Batch Bayesian Quadrature a sufficient cessation criterion. As we will see in the following two subsections on one dimensional evaluations and two dimensional evaluations, our cessation criterion is established and supported by numerical results. Improvements on these results, and possibilities for my accurate cessation criterion given other implications of the dynamic domain decomposition follows the next two subsections. This includes implications for higher dimensions.

3.6.1 One Dimension

For illustrative purposes several different test functions and batch sizes in one dimension will be presented in graphical form, however many different test functions and batch sizes could be selected. Tabular versions of the data are included in the Appendix. It can be seen both in graphical form and in tabular form that the numerical results are similar when batch sizes remain relatively close, however can diverge for significantly different batch sizes as expected. This will be further highlighted and discussed in the last subsection of this section.

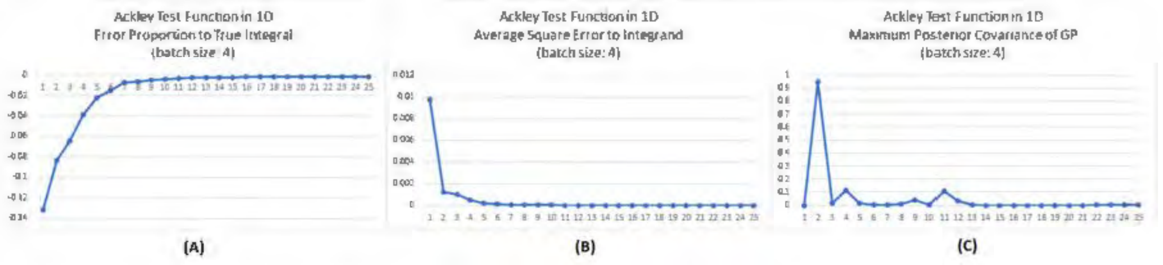


Figure 3.32: Ackley test function in 1D and batch size 4

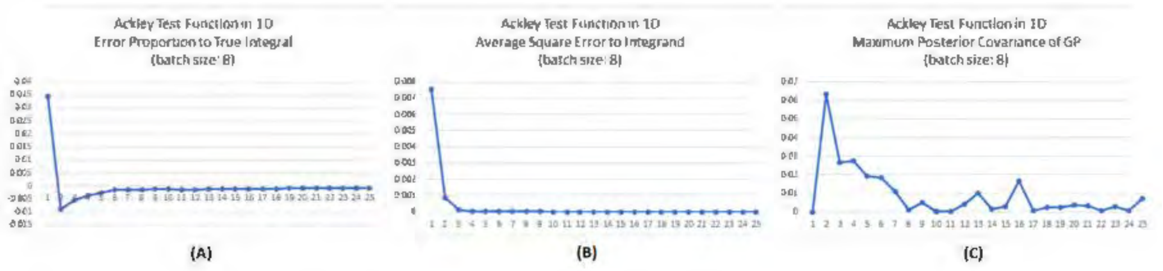


Figure 3.33: Ackley test function in 1D and batch size 8

The first set of numerical results which will be analyzed to support our cessation criterion is for the Ackley test function in one dimension. These results will be presented for batch size 4 in figure 3.32 and batch size 8 in figure 3.33, however many different kinds of batch sizes can be selected. Again, for numerical results see the Appendix.

Of particular importance to note is first the error proportion to the true integral, or panel (A) in figure 3.32 through figure 3.33 and figure 3.36 through figure 3.38. That is, the integral estimate is calculated as previously discussed. Due to the fact that these calculation are for one dimension, the true value of the integral is calculated using a fine grid. Since different choices of likelihood function will result in sometimes significantly different values of the integral, determining the absolute difference to the true can be misleading. Specifically, if an integral estimate is, for example, 0.001 away from the actual integral, this would be a good result if the integral value were 1 as it would represent a 0.1% error, but a bad result if the integral value were 0.002 as this would represent a 50% error. Therefore, instead of reporting the error to the true integral value, we report the error proportion to the true integral. Furthermore, while results would be equally elucidating for our purposes, we do not take the absolute value of this error proportion. Our method instead allows us to convey in which direction the estimate has an error.

The average square error to the integrand given in panel (B) in figure 3.32 through figure 3.33 and figure 3.36 through figure 3.38 is of particular importance to this study. First the sum of square error is a standard measure for the accuracy of an estimated function to the true function. This is calculated by taking the difference of the estimate and the true function, squaring this value, and summing it for all test location. It should be noted that if the SSE were to be calculated in the quadrature point selection process using Kriging Believer, all of these value would be equal to the posterior mean of the GP, and hence no information could be gained. That is, any point selection method, including uncertainty sampling and future uncertainty sampling would use as the functional evaluation of the quadrature points the same posterior mean of the GP, and hence only using these points would not allow us to determine a sufficient cessation criterion.

Similarly, even if a penalization technique were employed such as Local Penalization (Gonzáles et al., 2016; Wagstaff, 2018A) which only directly modifies the acquisition function around each selected point, the obstacle of not being able to only use the selected quadrature points to evaluate a proposed cessation criterion remains. That is, while this technique also allows for the selection of a batch of sample locations before those location are evaluated and the BQ model is updated, despite not utilizing the posterior mean of the GP as the functional evaluations of quadrature points, the posterior mean of the GP remains stationary as in Kriging Believer. Therefore, using this quadrature point selection method, regardless of whether coupled with uncertainty sampling or future uncertainty sampling, would not allow us to use only these points in evaluating a suitable cessation criterion.

A consequence of this process is therefore that when the model is updated, the integrand is evaluated at the selected quadrature points, where again this step is considered costly and a driving aspect of our research. Subsequently, these functional evaluation of the integrand at these quadrature points is used to update the GP model. Hence the posterior mean of the GP will traverse through these points, making their error to the true equal to zero for all locations. This, in turn, leads to no information of the accuracy of one technique over the other.

To circumvent this obstacle, similar to our procedure in the numerical results section, we define 2,000 equally spaced points in our domain in one dimension, and a grid of 2,000 equally spaced points in two dimensions as our evaluation points. While some of these points may overlap with the quadrature points selected, especially in large batch sizes and if many batches are selected, the vast majority will not. Therefore, there may be some evaluation points for which we gain no information in the comparison of quadrature point selection methods, but for many others we will.

It should be noted that the selection of 2,000 equally spaced evaluation points in one dimension and a grid of 2,000 equally spaced evaluation points in two dimensions was a determination suitable for our investigation, as previously argued in the numerical results section of this research. While it allowed us to perform the necessary comparisons, the ideal number of points to select remains an open question, but likely not one of high importance. However, what is important to note is that the selection of this many evaluation points, and the calculation of the sum of squared errors for all of them, will lead to rather large numbers even when there are only minor deviations is the estimated function to the true function. While these figures may be unsightly, the potentially large fluctuations allows us to notice even small differences between either both techniques, or when even minor deviations occur between the estimated function and the true function. However, since we are particularly interested in the overall convergence of our integrand estimate to the true, we average the square error over the equally spaced evaluation points in our domain. this allows us to therefore determine the average square error from our estimate to the true integrand.

The maximum posterior covariance of the GP is given in panel (C) in figure 3.32 through figure 3.33 and figure 3.36 through figure 3.38. The quadrature points are selected according to our novel process articulated earlier, namely using Kriging Believer and future uncertainty sampling. That is, during the point selection phase of our novel method, we select quadrature points in order to minimize the posterior covariance of the GP, and set the functional value of these quadrature points equal to the posterior mean of the GP. Specifically, we choose sample locations x'_* such that

$$\mathbf{x}'_* = \arg\min_{\mathbf{x}_*} \left[\max_x \mathbb{V}_{l|D}[l(x)\pi(x)] \right]$$

where again $\mathbb{V}_{l|D}[l(x)\pi(x)]$ is as in 3.13. As previously discussed, pictorially this is given by panel (B) in the following graph.

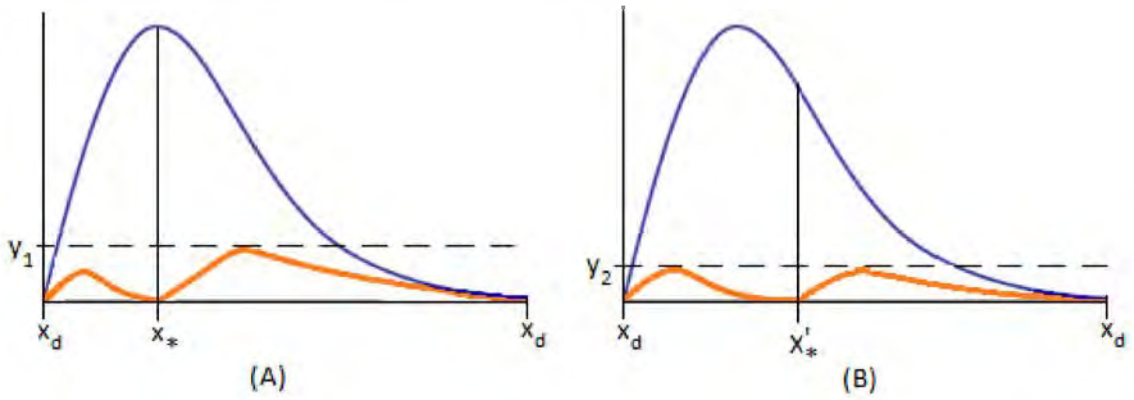


Figure 3.34: Warped Gaussian process posterior covariance

It is important to note that during the point selection phase of our novel method, the posterior covariance of the GP will uniformly decrease because the assumption is made that the functional values of these points are equal to the posterior mean of the GP since we use Kriging Believer. However, once the batch is complete and the integrand is evaluated using the quadrature points in the newly selected batch, we obtain the true functional values that can be significantly different than the posterior mean of the GP. This will be especially significant whenever our function to be estimated has frequent and relatively large fluctuations, as is the case in the Ackley test function and Weierstrass test function.

To be clear, when our search procedure selects quadrature points in a previously unexplored region, if the functional value of the integrand is significantly different that the posterior mean of the GP, then when the model is updated using these quadrature points, the posterior covariance of the GP will increase despite the fact that during the point selection phase our acquisition of quadrature points decreases the maximum posterior covariance of the GP. This is because the acquisition of the quadrature points takes the posterior mean of the GP as the functional values of the quadrature points, however when the model updates it uses the true functional values from the evaluation of the integrand. When the posterior mean of the GP is significantly different the the true integrand values at the quadrature points, then the posterior covariance of the GP will increase once the model is updated. Since panel (C) figure 3.32 through figure 3.33 and figure 3.36 through figure 3.38 represent the maximum posterior covariance of the GP *after* the model is updated with the new quadrature points, this graph need not be uniformly decreasing. Instead, if there is an increase in panel (C) in figure 3.32 through figure 3.33 and figure 3.36 through figure 3.38 is simply means that the process discovered a new significant fluctuation in its estimate of the integrand which it had not previously thought existed.

While panel (B) is of particular importance to our research, panel (C) represents the information which we will use to apply our cessation criterion. That is, panel (A) and panel (B) are only known to us because we know the integrand represented in part by our test functions. Knowing the functions, and therefore being able to determine their integral on a fine grid and measure the errors of our estimates to both the value of the integral and the integrand, is not an option in black box scenarios. Therefore, we will use panel (A) and panel(B) to inform us in finding a cessation criterion represented by a threshold on the maximum posterior covariance of the GP represented in panel (C). It should be noted here specifically for the posterior variance of the GP that, similar to our application of the square-root transformation in the warping of the GP on the likelihood, halving the dynamic range of the function we model provides both benefits and pitfalls. On one hand, this warping mitigates typically large variations in the likelihood, and extends the autocorrelation range of the GP yielding improved predictive power when extrapolating away from the data. On the other hand, the model is overconfident away from the data causing the the estimates of the uncertainty to be low. Hence our cessation criterion here may be lower than those proposed in reference to the application of different Bayesian quadrature techniques, and will in fact vary based on the dimensionality of the problem at hand.

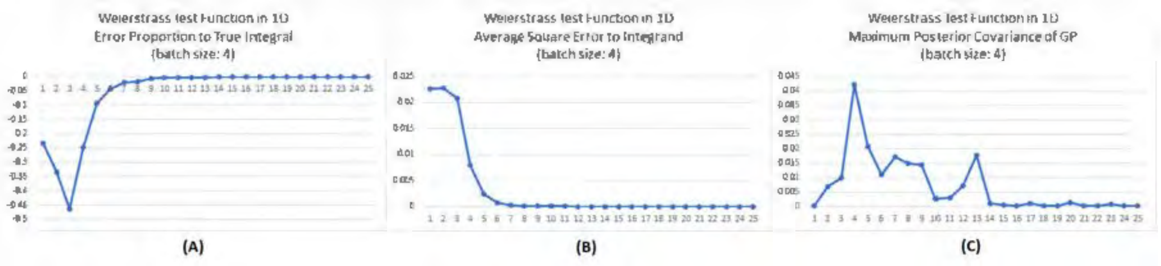


Figure 3.35: Weierstrass test function in 1D and batch size 4

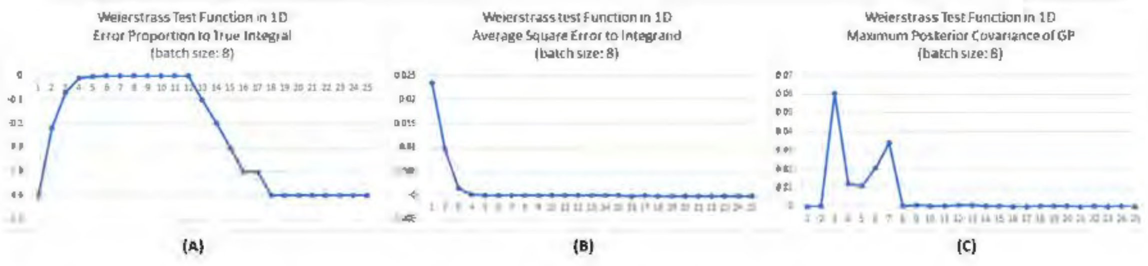


Figure 3.36: Weierstrass test function in 1D and batch size 8

The threshold for maximum posterior covariance of the GP as the cessation criterion will therefore depend on the values achieved by the error proportion of the estimated integral to the true integral, and the average square error of the estimated integrand to the true integrand. For our purposes we select that the error proportion of the estimated integral to the true integral should be less that 5%, and that the average square error of the estimated integrand to the true integrand should be less than 0.001. To accomplish this, the cessation criterion in one dimension is once the maximum posterior covariance of the GP falls below 0.015 after selecting at least 4 batches.

Something that should be noted is the significant difference in the error proportion of the estimated integral to the true integral, namely 0.05, and the average square error of the estimated integrand to the true integrand, namely 0.001. To justify this difference, and particularly the very low value for the average square error of the estimated integrand to the true integrand, is that a driving aspect of our research and technique is that we can estimate the integrand significantly better than currently available techniques in batch Bayesian Quadrature, especially when the cost of evaluating the integrand is very high. Placing such stringent requirements on the average square error of the estimated integrand to the true integrand is therefore justified since it addresses our primary concern.

Another aspect which should be noted is that we have also stated that our cessation criterion includes a minimum number of batches. Something that is particularly important for our novel contributions versus what has already been established in the literature regarding other batch Bayesian quadrature techniques is that since our method is able to minimize the posterior covariance of the GP more efficiently as shown in the numerical results section, other evaluation techniques may require more - and in fact significantly more - batches than our method.

Furthermore, a question which can be raised is whether the minimum number of batches should be specified or the minimum number of quadrature points. While this, on its surface, is a legitimate question with relevance to other quadrature techniques and their applications in many other fields, this is a question which has previously been answered in the batch Bayesian quadrature literature (Wagstaff, 2018A; Gunter et al., 2014). Again, since a driving aspect of our research is that the integrand is very costly to evaluate, the number of times we evaluate our integrand is extremely important. However, the number of points with which we chose to evaluate our integrand is only a secondary concern. Therefore, for example, for our purposes evaluating the integrand with a batch of size four quadrature points will have the approximate same cost as evaluating the integrand with a batch of size, say, 6 quadrature points. To be very specific, as was briefly mentioned earlier, this holds true whenever batch sizes remain relatively similar. In the batch Bayesian quadrature literature, as in this study here, batch sizes have ranged from one quadrature point to eight quadrature points, and therefore the above mentioned implications hold. Varying the size of the batches significantly and determining such implications may lead to different results. However, it may be difficult to find practical application where such peculiar circumstances may apply, and therefore were also not investigated here and remains an open question where it can be argued that it is not one of high practical importance.

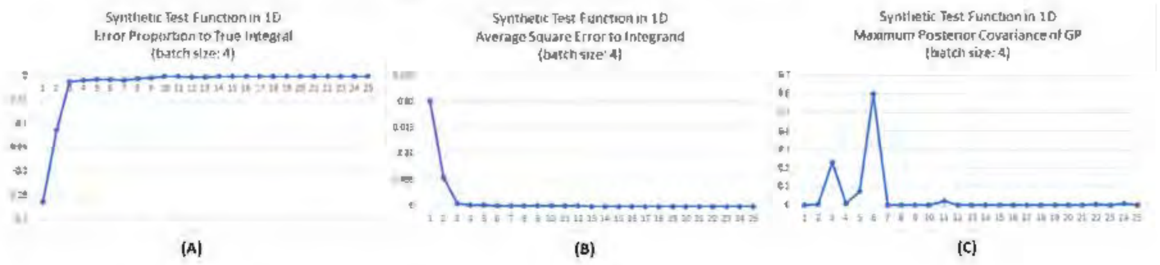


Figure 3.37: Synthetic test function in 1D and batch size 4

Nevertheless, an aspect which carries significance is when integrands significantly differ, and this implication on the cessation criterion. To investigate this, we consider numerical results from the application of our method to the three test functions already discussed, namely the Ackley test function, the Weierstrass test function, and the synthetic test function depicted in figures 3.32 and 3.33, 3.35 and 3.36, and 3.37

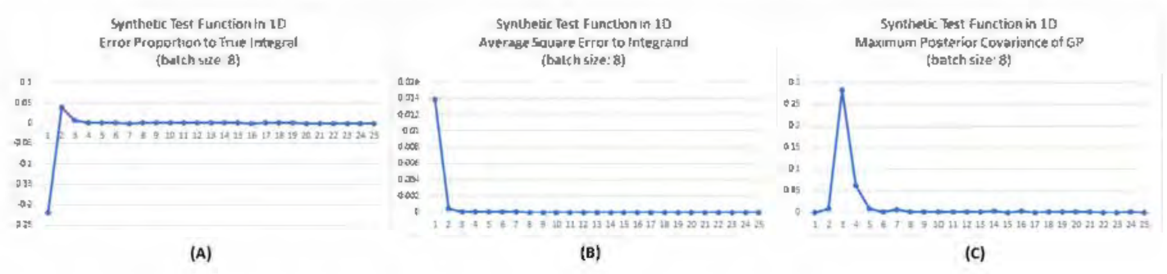


Figure 3.38: Synthetic test function in 1D and batch size 8

and 3.38, respectively. Here figure 3.32, figure 3.35 and figure 3.37 represent batch size of 4 and figure 3.33, figure 3.36 and figure 3.38 represent batch size of 8. For each graph, an ten runs of 25 batches were made and averaged. Despite the similarities in conducting the experiment, the difference in test function properties, specifically the number of oscillations, also only had modest impacts. While refinements can be made to such aspects as domain decomposition, size of batches, etc., our results and resulting cessation criterion hold firm for our results. To our knowledge, this establishes the first cessation criterion in batch Bayesian quadrature in the literature.

3.6.2 Two Dimensions

Many of the considerations discussed in regards to one dimension also hold true for two dimensions. However, there are some differences with significant implications that must be addressed. In two dimensions, a set of quadrature points are much more sparsely distributed than the same number of quadrature points in one dimension. This also holds true for higher dimensions. Therefore, in order to estimate the integrand, and therefore also the integral, efficiently, more quadrature points are needed as the dimensionality of the problem increases, *ceteris paribus*. However, as was previously argued, a driving consideration of our research isn't the number of batch points selected, but instead the number of evaluations of the integrand. To that end, two dimensional and higher dimension problems will require more batches due to the sparsity of the point locations compared to one dimension.

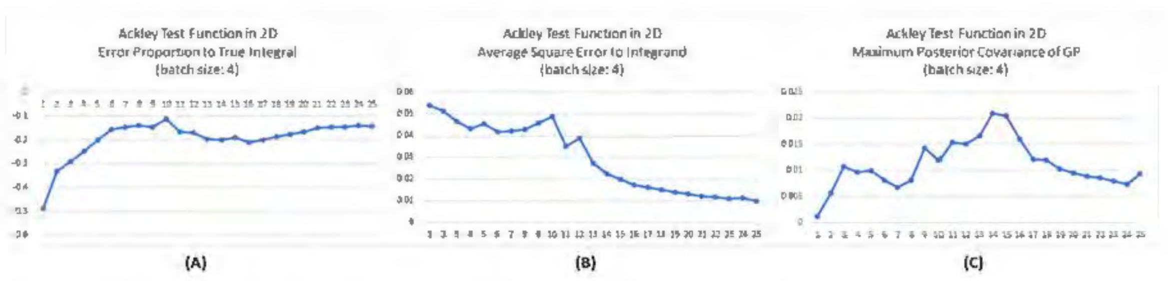


Figure 3.39: Ackley test function in 2D and batch size 4

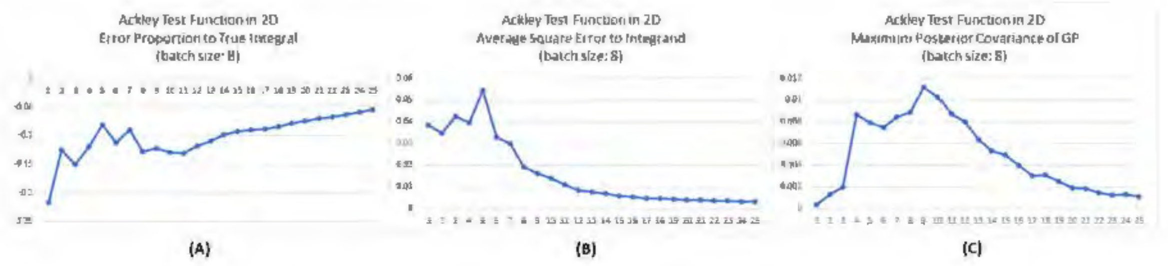


Figure 3.40: Ackley test function in 2D and batch size 8

In order to achieve results that support our constraints of achieving an error proportion of the integral estimate to the true integral of no more that 5%, and an average square error of the integrand estimate to the true integrand of no more that 0.001 as we did in one dimension, our numerical results determine and support a cessation criterion for two dimensions of a posterior covariance of the GP of no more than 0.001 after at least 16 batches are taken. This, again, is to our knowledge the first cessation criterion in the literature for batch Bayesian quadrature.

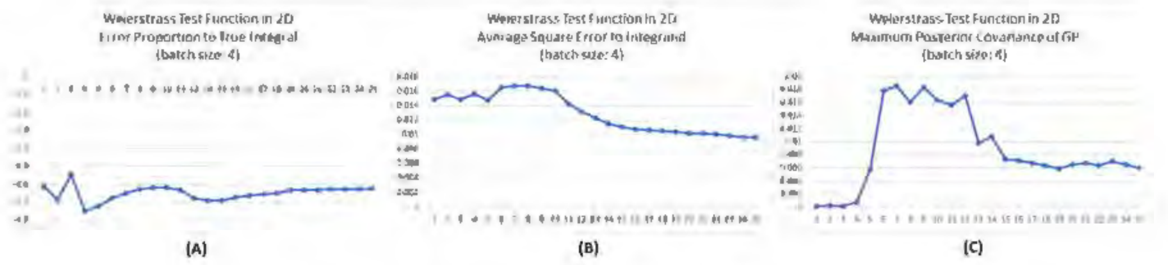


Figure 3.41: Weierstrass test function in 2D and batch size 4

There are several subtleties which must be noted and discussed. First among those is the fact that errors tend to not increase linearly as dimensions are increased. That

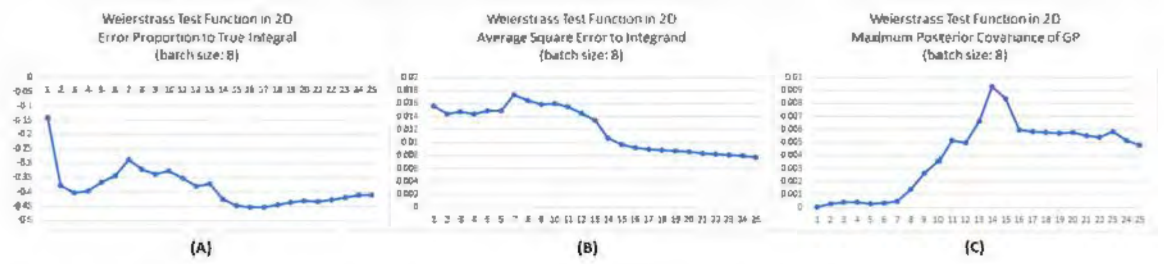


Figure 3.42: Weierstrass test function in 2D and batch size 8

is, an error of a certain size, say 0.001, in one dimension can have smaller implications that an the same size error in all directions in multiple dimensions. Therefore, it is important to lower the threshold of the uncertainty in order to ensure continued robust estimates. Furthermore, increasing the size of batches may have a significant impact in multiple dimensions as compared to one dimension. As already discussed, the sparsity of the quadrature points in multiple dimensions is much more profound than in one dimension, or even for lower dimensions compared to higher dimensions. Since a driving constraint of our problem is that evaluating the integrand is expensive, in the case of multiple or higher dimensions it may not only be warranted but perhaps even advisable in significantly increase batch sizes. This will be discussed in the next subsection.

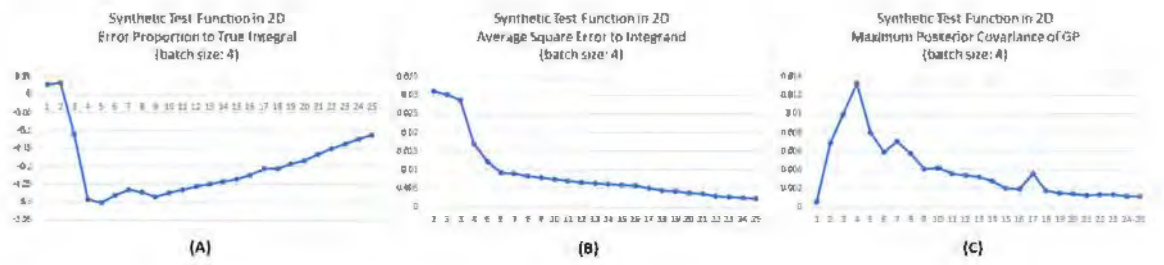


Figure 3.43: Synthetic test function in 2D and batch size 4

It should again be noted that one aspect the test function significantly differ is in the number of oscillations they have, where again here the test functions we employ are the Ackley test function, Weierstrass test function, and a synthetic test function all in two dimensions. These are depicted in figures 3.39 and 3.40, 3.41 and 3.42,

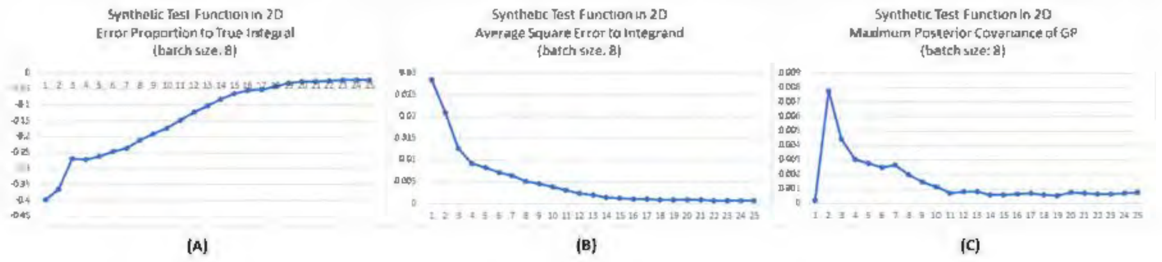


Figure 3.44: Synthetic test function in 2D and batch size 8

and 3.43 and 3.44, respectively. Here figure 3.39, figure 3.41 and figure 3.43 represent batch size of 4 and figure 3.40, figure 3.42 and figure 3.43 represent batch size of 8. For each graph, ten runs of 25 batches were made and averaged. Despite the similarities in conducting the experiment, the difference in test function properties, specifically the number of oscillations, had a growing impact as compared to one dimension. Increases in the dimensionality of the problem also significantly increases the number of regions where significant fluctuations in the posterior covariance of the GP can exist, the implications of which are discussed in the next subsection. While refinements can be made to such aspects as domain decomposition, size of batches, etc., our results and cessation criterion nevertheless hold firm for our results. To our knowledge, this establishes the first cessation criterion in batch Bayesian quadrature in the literature.

3.6.3 Higher Dimensions

An important distinction between our numerical results in one dimension versus two dimensions, which will also bear of higher dimensions, is the significant increase in number of batches that must be selected in order to achieve the desired results. Specifically, in one dimension we determined that in order to meet our defined objective that the cessation criterion would be once the posterior covariance of the GP fell below the 0.01 threshold after at least 4 batches were taken. However, in two dimensions we stated that in order to meet our stated objectives that the cessation criterion would be that the maximum posterior covariance of the GP must be below a threshold of 0.001 after at least 16 batches were selected. The reason for the lower threshold of the posterior covariance of the GP was already discussed. However, the significant increase in the minimum number of batches is particularly noteworthy given the constraints imposed by batch Bayesian quadrature, namely that evaluating the integrand is costly.

The underlying impetus for this significant increase in the number of batches was already discussed. It is clear that these impetus will continue to have increasing implications as the dimensionality of the problem is increased. Therefore, while it was not explicitly investigated in this research, as the dimensionality of the problem increases, *ceteris paribus*, it would be reasonable to expect that the number of batches which must be selected in three dimensions in order to meet our defined objective would exceed 50. This is a staggering number in batch Bayesian quadrature. One possibility of mitigating this would be to significantly increase the number of quadrature points within a batch.

A distinction must be made here from what was stated earlier regarding increasing batch sizes significantly. It should be noted that finding a practical application of the comparison of, say, a batch size of 4 to a batch size of 40, *ceteris paribus*, would be difficult to find. However, finding an application of only a batch size of 40 would be possible. Therefore, it is not necessarily that large batch sizes in batch Bayesian quadrature do not have practical applications, but rather a scenario where *ceteris paribus* the batch size significantly increases would be peculiar.

It is therefore possible not only to apply our novel technique to higher dimensions, but also to use our novel method of finding cessation criterion for these. While one possibility of accomplishing these tasks would be to significantly increase the size of the batches selected, such application of batch Bayesian quadrature do not currently exist in the literature and were not explicitly investigated here. For that reason, further discussion of the issue is left the the section of this research addressing future work.

3.7 Applications and Relevance of Novel Procedure

Two specific applications were discussed in reference to the test functions used in the numerical evaluation of our novel method. Specifically, the Ackley test function and Weierstrass test function have widespread applications in optimization, quadrature, and black box algorithms that have heavily employed in the machine learning literature. Several other applications in machine learning will be highlighted, but at a more high level that what was previously discussed in the numerical applications section and throughout this document.

What is of particular interest are applications in machine learning that make use of integration of unknown or intractable integrals, the prime focus of our research. The most prominent and common instances where such problems arise involve marginalizations. Marginalization is the process of summing over possible values of one or more variables or parameters to determine the marginal contribution or effect of another variable or parameter. This summation would clearly be the sum for discrete variable or the integral for continuous variables, where we focus on the latter in this research. It should therefore be obvious that these kinds of applications are not limited to applications of machine learning, though they feature prominently in such applications. In fact, marginalization is a concept learned in introductory statistics courses which highlights their ubiquitous applications.

A common instance where marginalization occurs is its application to latent variables or parameters. Such latent variables or parameters are often considered to be nuisance variables or parameters, and clearly can take on many different forms depending on the field of interest. Similarly, this application can be expanded to include marginalizing over likelihoods which themselves can represent sets of variables or parameters.

We can take an initial step in going from integrating over variables or parameters, to marginalizing over likelihoods, but we can also continue to take steps and include model averaging into the realm of possible applications for our novel technique. Model averaging, or more specifically ensemble averaging in machine learning fields such as artificial neural networks, is when multiple models are combined to represent an output being investigated with the intended outcome that the ensemble better represents the phenomena.

Lastly, there are a host of related yet distinct sub-areas of application, which include computing posterior predictive distributions, computing model evidences, and partition functions.

It is important to note that other techniques in machine learning have made significant strides in addressing these same topics. For example, deep learning, currently a particularly hot topic in machine learning, solves similar problems in the areas outlined above. However, what should also be noted is that the fields of batch Bayesian quadrature and deep learning, while both are in the broader field of machine learning, have very different constraints. Our application of batch Bayesian quadrature, and the superior novel techniques we have devised, have a central definition that evaluating the integrand is considered to be very expensive, and therefore the fewest number of batches should be chosen. In our numerical section we chose at most 25 batches. In deep learning, on the other hand, a driving consideration in its application is using high dimensional functions - often six or higher. This is many more dimensions than would be feasible with batch Bayesian quadrature for one striking reason, namely that some of the most state-of-the-art techniques in deep learning evaluate the integrand well over 100,000 times, and often several million times! Therefore, while while batch Bayesian quadrature as of yet cannot match the dimensionality of deep learning, deep learning also cannot match the efficiency of batch Bayesian quadrature.

In conclusion, the problem space adhering to the constraints of batch Bayesian quadrature is very large. Our novel contributions significantly improve upon what is available in the literature to tackle these problems. For these reasons, our novel contributions are very significant and relevant to many fields, including the field of machine learning in general, and quadrature specifically.

3.8 Future Work

The main aspect of work that can and should be accomplished, as stated several times throughout this document, primarily pertains to the dynamic domain decomposition and its impacts on working in higher dimensions and the cessation criterion.

To that end, one aspect of interest is how many sample locations should be allocated to each partition of our domain. For computational convinience we utilized 100 equally spaced sample locations per partition in one dimension, and 25 equally spaced sample locations in two dimensions. While this yielded suitable results for our objective, there was not a determination made on whether or not different numbers would be more beneficial. Specifically, by increasing sample locations we could have chosen quadrature points which better illustrated the superiority of our novel method, however this would have come at the expenditure of increased computational cost. Conversely, we could have saved additional computational cost by allocating fewer sample locations.

One way this may be researched is by considering the parameters $\theta = \{\lambda, \sigma\}$, where these represent the output length-scale λ and input length-scale σ control the standard deviation of the output and the autocorrelation range of each function evaluation, respectively. Based on what these parameters are, especially after they have been optimized as is standard practice, could inform the optimal number of sample locations. This holds true both in the original allocation before search begins, as well as when sample locations have been replenished. The the appropriate sections for a more detailed discussion on how those procedures were implemented in this research.

A similar consideration is when to replenish sample locations within partitions. Our current practice, which showed promising results, is to replenish the sample locations within a partition only after the previous allocation within that partition had been completely exhausted. This may also lead to inefficiencies, particularly when partitions are large. To see this, if a partition is relatively large, and sample locations are sparsely distributed, it is possible that the location corresponding to the maximum posterior covariance of the GP may reside in this partition, but not near where the remaining sample locations exist within that partition. Therefore, it may be optimal to still chose the available sample location due to its proximity to the location corresponding to the maximum posterior covariance of the GP - that is, other sample locations outside of the partition would be worse choices - however efficiency may be gained if sample locations within that partition existed which are closer to the location which corresponds to the maximum posterior of the GP. Therefore, replenishing sample locations more quickly would provide such points, but also increase computational cost earlier.

Another consideration, which goes hand-in-hand with these, is the size of the partitions. In our process here we naively established where partitions would be, by either determining the number of partitions or their length *a priori*. Clearly forming larger partitions while keeping sample locations constant would potentially decrease computational cost while perhaps leading to less accurate results. Conversely, decreasing the size of partitions would perhaps lead to more accurate results, while potentially increasing computational cost. Again, considering the parameters $\theta = \{\lambda, \sigma\}$ may lead to increased efficiency.

An interesting consideration to the size of the partitions is to research adding an additional dynamic component to our already dynamic domain decomposition, namely altering the size of the partitions in multiple dimensions similarly to that which we employed in one dimension. While the technique we used in one dimension cannot directly translate to multiple dimensions, it would be interesting to investigate how altering partitions could improve our process. Again, see the appropriate sections in this document for a more thorough description of our novel process.

Achieving higher efficiency from an improved dynamic domain decomposition may allow us to increase the dimensionality of the functions which can be considered using batch Bayesian quadrature. As was shown in our numerical results section, because a driving consideration of batch Bayesian quadrature is that evaluating the integrand is expensive and hence the least number of batches possible should be taken, coupled with the fact that quadrature points become more sparse as the dimensions of the problem increase, utilizing batch Bayesian quadrature in higher dimensions becomes unfeasible. However, if sample locations could be allocated more efficiently, and hence quadrature points which are more informative can be chosen, it may be possible to slightly increase the dimensionlity of the problems which can be undertaken. However, to be clear, morphing batch Bayesian quadrature into something like deep learning where very high dimensional problems can be solved, should not be the objective. These fields have different constraints, and hence are applicable to different scenarios. Nevertheless, increasing the number of scenarios where batch Bayesian quadrature is applicable may be feasible.

Finally, something these improvements would allow to achieve is to further refine our numerically supported cessation criterion. Having more accurate results would allow us to better find and support a cessation criterion. If the aforementioned improvements from future work also allow us to expand batch Bayesian quadrature to higher dimensions, then we could also elaborate our existing cessation criterion to those dimensions.

CONCLUSION

Quadrature can trace its origins all the way back to ancient Greece, and yet it is anything but outmoded. It is a topic that has reinvented itself, including contributions by Gottfried Leibnitz and Sir Isaac Newton. In the 20th century, quadrature found new, modern applications in the realm of probability, first using Monte Carlo, and later using Bayesian principles. As more and more other bastions of knowledge have been used in conjunction with quadrature, this once ancient idea has found its way into even the most cutting-edge technology in such fields as machine learning. And yet, new discoveries in this field continue to grow.

To find evidence of this one need not look any further than this document. It in we have described a significant step forward in the use of quadrature, namely the used of batch Bayesian quadrature with *selection updating.* Perhaps surprisingly, this is the first batch quadrature method where point selection need not be sequential, and it allows us to achieve remarkable improvements. To accomplish this, we devise a novel point selection algorithm which uses *future uncertainty sampling.* Unlike other modern techniques which often aim to select quadrature points which maximize the reduction in uncertainty, our method instead selects quadrature points which minimizes the resulting uncertainty. This seemingly trivial distinction and minor change in method in fact leads to significant improvements in the efficiency of uncertainty reduction, accuracy of integrand estimation, and more robust numerical estimates of the integral. These represent significant strides forward!

In order to implement our novel technique, we devised a novel dynamic domain decomposition which not only enabled us to achieve superior results, but can also find applications in other fields. Of particular interest is the ability of this dynamic domain decomposition to be applied to many different dimensions. This can allow batch Bayesian quadrature with selection updating to be applied even to higher dimensions than presented in this document, and may be a promising avenue to further expand this resilient field into the future.

Furthermore, the previous instances of batch Bayesian quadrature in the literature left the cessation criterion as an open question. Therefore, our research presents the first numerically founded and supported cessation criterion in batch Bayesian quadrature. We present these for applications in one dimension and two dimensions.

Despite our numerous novel contributions and significant improvements these achieve, there remains work to be accomplished. We highlight in the future work section of this document that chief among these is a refinement of our dynamic domain decomposition. While this aspect also represents a novel contribution of ours, it did not represent the most significant contribution. Nevertheless, it may present the best means to facilitate the application of batch Bayesian quadrature in higher dimensions. If this is in fact possible, which we believe it is, this would be another significant step forward in this field.

While it should be clear that with our contributions we enable the longstanding field of quadrature to take a significant step forward, it should also be clear that despite its longevity, this field still has many more places to go. Our contributions demarcate a milestone on that journey, and we look forward to continue to propel the field of quadrature forward!

REFERENCES

Ackley, D. (1987) A conectionist machine for genetic hillclimbing. Kluwer Academic Publishers, Boston, MA.

Anderson, E. (1999) Monte Carlo Methods and Importance Sampling. Lecture Notes for Stat 578C Statistical Genetics, University of California, Berkeley.

Blei, D., Kucukelbir, A., McAuliffe, J. (2016). Variational Inference: A Review for Statisticians. Journal of the American Statistical Association, 112(518), 859-877. arXiv:1601.00670

Briol, F., Oates, C., Girolami, M., Osborne, M., Sejdinovic, D. (2015). Probabilistic Integration: A Role in Statistical Computation?, 1-49. arXiv:1512.00933.

Brooks, S., Roberts, G. (1998) Convergence assessment techniques for Markov chain Monte Carlo. Technical Report CRG-TR-93-1, University of Toronto.

Burton, M. (2011) Elementary Numerical Analysis, 7th Edition. New York: McGraw-Hill.

Chai, H., Garnett, R. (2018) An Implroved Bayesian Framework for Quadrature. arXiv:1802.04782

Chen, M., Shao, Q., Ibrahim, J. (2000) Monte Carol methods in Bayesian computation. Springer.

Chorin, A. J, Hald, O. (2013) Stochastic Tools for Mathematics and Science. New York: Springer.

Cisewski, J. "Importance Sampling." From Penn State University. http://astrostatistics.psu.edu/su14/lectures/cisewski_is.pdf

Cowles, M., Roberts, G., Rosenthal, J. (1999) Possible biases induced by MCMC convergence diagnostics. Journal of Statistical Computation and Simulation, 64(1): 87.

Deng, S. "Quadrature Formulas in Two Dimensions." From Math 5172. http://math2.uncc.edu/ shaodeng/TEACHING/math5172/Lectures/Lect_15.PDF

Diaconis, P. (1988). Bayesian Numerical Analysis. In Statistical decision theory and related topics iv (pp. 163-175). New York, NY: Springer New York.

Duvenaud, D. (2013) Bayesian Quadrature: Model-based Approximate Integration. Presentation, University of Cambridge.

Fouss, F., Pirotte, A., Renders, J-M., Saerens, M. (2007) Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Transactions on Knowledge and Data Engineering, 19(3): 355-369.

Garnett, R., Osborne, M., Reece, S., Rogers, A., Roberts, S. (2010) Sequential bayesian prediction in the precense of changepoints and faults. The Computer Journal, 53(9): 1430.

Garnett, R., Krishnamurthy, Y., Xiong, X., Schneider, J., Mann, R. (2012) Bayesian optimal active search and surveying. In J. Langford and J. Pineau (eds.), Proceedings of the 29th International Conference on Machine Learning (ICML 2012). Omnipress, Madison, WI, USA.

Ginsbourger, D., Le Riche, R., Carraro, L. (2010). Kriging Is Well-Suited to Parallelize Optimization. Computational Intelligence in Expensive Optimization Problems, 2, 131-162.

Gonzáles, J., Dai, Z., Hennig, P., Lawrence, N. (2016). Batch bayesian optimization via local penalization. In Artificial intellegence and statistics (pp. 648-657).

Gunter, T., Osborne, M., Garnett, R., Hennig, P., Roberts, S. (2014) Sampling for Inference in Probabilistic Models with Fast Bayesian Quadrature. In Advances in neural information processing systems (nips) (pp. 1-9). arXiv: 1411.0439v1.

Hansen, N., Müller, Koumoutsakos, P. (2003) Reducing the time complexity of the derandomized evolutin strategy with covariance matrix adaptation (CMA-ES). Evolutionary Computation, 11(1): 1-18.

Harrison, R. (2010). Introduction To Monte Carlo Simulation. AIP Conference Proceedings 1204, 17.

Hastings, W. (1970). Monte carlo sampling methods using markov chains and their applications. Biometrika, 57(1), 97-109.

Hennig, P., Osborne, M., Girolami, M. (2015). Probabilistic Numerics and Uncertainty in Computations. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 471(2179). arXiv:1506.01326

Hoffman, M., Gelman, A. (2014) The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. Journal of Machine Learning Research, 15(1), 1593-1623.

Huszar, F., Duvenaud, D. (2012) Optimally-Weighted Herding in Bayesian Quadrature. From Proceedings of the Twenty-Eight Conference on Uncertainty in Artificial Intelligence. Arlington, Virginia: AUAI Press.

Jamil, M., Yang, X. (2013) A Literature Survey of Benchmark Functions For Global Optimization Problems. Int. Journal of Mathematical Modeling and Numerical Optimization, Vol. 4, No. 2, pp. 150-194.

Jones, D. (2001). A taxonomy of global optimization methods based on response surfaces. Journal of global optimization, 21(4), 345-383.

Karvonen, T., Särkkä, S. (2017). Classical quadrature rules via Gaussian processes. In Ieee international workshop on machine learning for signal processing (mlsp) (pp. 1-6).

Kennedy, M. (1998) Bayesian Quadrature with non-normal approximating functions. Statistics and Computing, 8(4): 365-375.

Meng, X., Wong, W. (1996) Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. Statistica Sinica, 6(4): 831-860.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E. (1953). Equation of state calculations by fast computing machines. The journal of chemical physics, 21(6), 1087-1092.

Minka, T. (2000). Deriving quadrature Rules from Gaussian processes. Technical report, Statistics Department, Carnegie Mellon University, 1-21.

Neal, R. Probabilistic inference using Markov Chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

Neal, R. (2001) Annealed importance Sampling. Statistics and Computing, 11(2): 125-139.

Nguyen, V., Rana, S., Gupta, S., Li, C., Venkatesh, S. (2017). Budgeted Batch Bayesian Optimization With Unknown Batch Sizes. In Ieee international conference on data mining, icdm (pp. 1107-1112). arXiv:1703.04842.

Lambers, J. "MAT 460/560 Lecture 31 Notes." From MAT 460/560 Numerical Analysis I. http://www.math.usm.edu/lambers/mat460/fall09/lecture31.pdf

O'Hagan, A. (1987). Monte Carlo is Fundamentally Unsound. Journal of the Royal Statistical Society. Series D (The Statistician), 36(2), 247-249.

O'Hagan, A. Bayes-Hermite quadrature. Journal of Statistical Planning and Inference, 29: 245-260, 1991

Osborne, M., Duvenaud, D., Garnett, R., Rasmussen, C., Roberts, S., Ghahramani, Z. (2012) Active Learning of Model Evidence Using Bayesian Quadrature. Advances in Neural Information Processing Systems, 1, 46-54.

Osborne, M., Garnett, R., Roberts, S., Hart, C., Aigrain, S., Gibson, N., Aigrain, S. (2012) Bayesian quadrature for ratios. In Proceedings of the Fifteenth International Conference oon Artificial Intelligence and Statistics (AISTATS 2012).

Quarteroni, A., Saleri, F., Gervasio, P. (2010) Scientific Computing with MATLAB and Octave. Heidelberg: Springer.

Rasmussen, C. E., Ghahramani, Z. In Becker, S. and Obermayer, K. (eds.), Advances in Neural Information Processing Systems, volume 15. MIT Press, Cambridge, MA, 2003.

Rasmussen, C., Nickisch, H. (2010) Gaussian process for machine learning (GPML) toolbox. The Journal of Machine Learning Research, 11(2010): 3011-3015.

Rasmussen, C. E., Williams, C. K. I., (2006) Gaussian Processes for Machine Learning. Cambridge, MA. MIT Press.

Skilling, J. (2004). Nested Sampling. Bayesian inference and maximum entropy methods in science and engineering, 735: 395-405.

Suganthan, P., Hansen, N., Liang, J., Deb, K., Chen, Y., Augeer, A., Tiwari, S. (2005) Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Technical Report, Nanyang Technological University, Singapore.

Wagstaff, E. (2018) bayesquad Python code. Retrieved from: https://github.com/OxfordML/bayesquad.

Wagstaff, E., Hamid, S., Osborne, M. (2018). Batch Selection for Parallelisation of bayesian Quadrature. arXiv: 1812.01553v1.

Wasserman, L. (2004) All of Statistics: A Concise Course in Statistical Inference. New York: Springer.

Weisstein, E. "Hermite-Gauss Quadrature." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Hermite-GaussQuadrature.html

Weisstein, E. "Legendre-Gauss Quadrature." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Legendre-GaussQuadrature.html

APPENDIX

Ackley Test Function in 1D with Batch Size 4

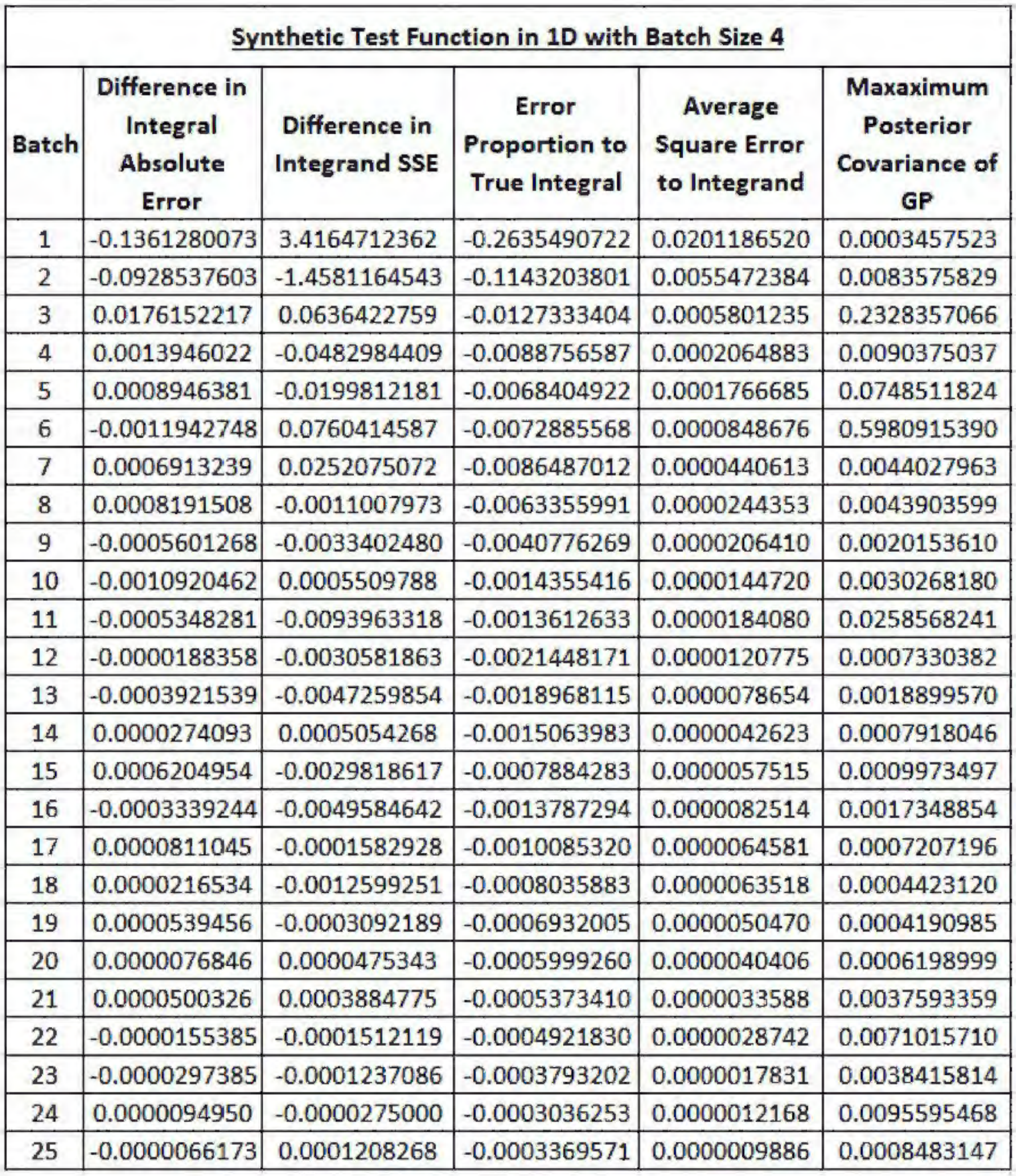
Batch	Difference in Integral Absolute Error	Difference in Integrand SSE	Error Proportion to True Integral	Average Square Error to Integrand	Maxaximum Posterior Covariance of GP
1	-0.2261911383	17.8443556123	-0.1312642397	0.0097173292	0.0004091749
2	0.0129189224	1.3510756804	-0.0837295416	0.0011931602	0.9495427393
3	-0.0045716621	0.5905490549	-0.0645012615	0.0010053787	0.0185090812
4	-0.0149577403	-0.0051948531	-0.0388990992	0.0004514216	0.1169537530
5	-0.0060276378	-0.0170247624	-0.0221422760	0.0001538036	0.0189320921
6	-0.0038647323	0.0046404469	-0.0152295914	0.0001100138	0.0057602211
7	0.0005946268	-0.0180733133	-0.0077217733	0.0000425500	0.0016714235
8	-0.0020276256	-0.0087717749	-0.0068342916	0.0000197151	0.0087990653
9	-0.0014980251	0.0016789052	-0.0053956007	0.0000152754	0.0402880869
10	0.0000529123	0.0018228189	-0.0040009136	0.0000067421	0.0039323340
11	-0.0002467146	-0.0009683063	-0.0032039848	0.0000045384	0.1107873288
12	0.0001065174	0.0009031263	-0.0024684407	0.0000046580	0.0317631216
13	0.0003038706	-0.0011813034	-0.0027087784	0.0000047102	0.0029868954
14	-0.0002304466	-0.0005758259	-0.0024640431	0.0000041704	0.0001428920
15	-0.0001554301	-0.0000674221	-0.0024494163	0.0000038845	0.0001321398
16	0.0002968752	0.0003030088	-0.0022360454	0.0000028479	0.0000399041
17	-0.0000959180	-0.0002813925	-0.0021311556	0.0000026164	0.0001958980
18	0.0000335466	-0.0000497903	-0.0021694364	0.0000022272	0.0000153831
19	0.0002538417	0.0002165716	-0.0020083744	0.0000021969	0.0000198240
20	0.0000692969	0.0000531792	-0.0017548436	0.0000016841	0.0002751336
21	0.0000040718	0.0000131710	-0.0016979434	0.0000015542	0.0000544795
22	-0.0000171771	-0.0000375494	-0.0017505104	0.0000012769	0.0008892339
23	-0.0001431537	-0.0000788855	-0.0017346621	0.0000012565	0.0018647648
24	0.0000320901	0.0001458354	-0.0017094792	0.0000011907	0.0024558868
25	0.0000149590	0.0000338177	-0.0016400609	0.0000011519	0.0052629896

Ackley Test Function in 1D with Batch Size 8

Batch	Difference in Integral Absolute Error	Difference in Integrand SSE	Error Proportion to True Integral	Average Square Error to Integrand	Maxaximum Posterior Covariance of GP
1	0.0449855093	19.7150419740	0.0342908292	0.0075499546	0.0000027706
2	0.0191942618	0.3240857147	-0.0087715155	0.0008489455	0.0631956798
3	0.0083705726	0.0142684309	-0.0055371341	0.0001175101	0.0263909979
4	0.0007322090	0.0221427124	-0.0037749442	0.0000161322	0.0272701049
5	0.0002690452	0.0002825483	-0.0027515181	0.0000073570	0.0194390454
6	0.0001458738	-0.0039139944	-0.0015202557	0.0000057758	0.0182798377
7	-0.0002971362	-0.0003303579	-0.0014803920	0.0000039105	0.0112088403
8	-0.0000861417	-0.0000309304	-0.0013738248	0.0000034741	0.0012213867
9	0.0001673591	-0.0025509428	-0.0011887078	0.0000036006	0.0049956738
10	-0.0002281049	-0.0001878728	-0.0013228768	0.0000026571	0.0001393237
11	-0.0004842887	-0.0009385252	-0.0016095468	0.0000027304	0.0001313637
12	-0.0001377455	-0.0006200024	-0.0013741768	0.0000014800	0.0042433956
13	-0.0000770363	-0.0001115601	-0.0012579025	0.0000009542	0.0102348541
14	-0.0001057274	-0.0000739932	-0.0011866881	0.0000007310	0.0016699822
15	0.0000337281	0.0000808499	-0.0011235817	0.0000006561	0.0028563528
16	-0.0000617476	-0.0000629784	-0.0011031453	0.0000005657	0.0166149976
17	-0.0000221144	-0.0000467650	-0.0010333136	0.0000005250	0.0006017478
18	0.0000151187	-0.0000480576	-0.0010426913	0.0000005271	0.0022054510
19	-0.0000091094	-0.0000278724	-0.0009857568	0.0000004827	0.0021980196
20	0.0000550761	0.0000425563	-0.0009463045	0.0000004347	0.0038163023
21	0.0000256694	0.0000081754	-0.0008832476	0.0000004023	0.0034287240
22	0.0000642110	0.0000507359	-0.0008292828	0.0000003586	0.0006009087
23	0.0001151844	0.0000727707	-0.0007820067	0.0000003311	0.0026959774
24	-0.0000915511	-0.0000680761	-0.0008125402	0.0000003383	0.0005942807
25	0.0000014360	-0.0000146217	-0.0007569263	0.0000003103	0.0069981206

Weierstrass Test Function in 1D with Batch Size 4					
Batch	Difference in Integral Absolute Error	Difference in Integrand SSE	Error Proportion to True Integral	Average Square Error to Integrand	Maxaximum Posterior Covariance of GP
1	-0.1976498024	10.0161711161	-0.2353065173	0.0226549913	0.0000810261
2	-0.1295446246	1.6658259519	-0.3354658509	0.0228057559	0.0067675230
3	-0.1210833161	1.4441602636	-0.4648146481	0.0207600175	0.0097586654
4	0.0303854555	7.0235667858	-0.2493316523	0.0080303849	0.0420181006
5	0.0149421655	0.9695670431	-0.0949653490	0.0024724907	0.0206349647
6	-0.0098235611	-0.0615369779	-0.0438323609	0.0006641019	0.0108721580
7	-0.0007861315	0.0633721461	-0.0220155865	0.0002008195	0.0169755615
8	-0.0013643017	0.0062686793	-0.0171953405	0.0000892690	0.0149309119
9	0.0013140082	0.0073604742	-0.0077376674	0.0000381214	0.0143597928
10	0.0037177795	0.0072856173	-0.0034948070	0.0000185800	0.0027709254
11	0.0014712267	0.0004091951	-0.0029644666	0.0000120282	0.0030580794
12	-0.0007765743	0.0019923253	-0.0036976934	0.0000094169	0.0072038159
13	-0.0003505051	-0.0008038236	-0.0028332435	0.0000084732	0.0175643545
14	-0.0005339547	-0.0006743633	-0.0025444848	0.0000065005	0.0009198271
15	-0.0001440368	-0.0013647999	-0.0019964188	0.0000043621	0.0005358876
16	0.0001358926	0.0011402280	-0.0021023283	0.0000025916	0.0001540769
17	-0.0004915219	-0.0015675525	-0.0020421511	0.0000028348	0.0009733221
18	-0.0002763248	-0.0007894934	-0.0019166828	0.0000026128	0.0001586974
19	-0.0002080959	-0.0004914639	-0.0018475428	0.0000024653	0.0002104691
20	-0.0001950800	-0.0003020012	-0.0018799984	0.0000023309	0.0013161176
21	-0.0000470279	0.0001504097	-0.0017472318	0.0000018932	0.0000495482
22	0.0002317384	0.0002401413	-0.0014550256	0.0000016625	0.0002025687
23	0.0000546363	0.0000016951	-0.0013772714	0.0000017012	0.0006123512
24	-0.0000656648	-0.0000157495	-0.0014886902	0.0000016556	0.0002197820
25	-0.0000347120	-0.0000232570	-0.0014850715	0.0000016295	0.0000261336

Weierstrass Test Function in 1D with Batch Size 8					
Batch	**Difference in Integral Absolute Error**	**Difference in Integrand SSE**	**Error Proportion to True Integral**	**Average Square Error to Integrand**	**Maxaximum Posterior Covariance of GP**
1	-0.4700899944	12.0664692192	-0.5023805741	0.0234613609	0.0000013060
2	0.0205409611	5.1232274883	-0.2202045058	0.0095426703	0.0002065309
3	0.0123927922	0.2060148600	-0.0730912742	0.0015205148	0.0601761428
4	0.0031477100	0.0452405402	-0.0120079501	0.0001273965	0.0124771140
5	0.0010281360	-0.0002595079	-0.0034018253	0.0000197166	0.0112113239
6	-0.0006983866	-0.0003706931	-0.0022248798	0.0000090286	0.0205189228
7	-0.0001774984	-0.0009198887	-0.0013640993	0.0000058614	0.0340256591
8	0.0006412138	-0.0000941840	-0.0012544456	0.0000023855	0.0001035538
9	0.0006027291	-0.0002137631	-0.0010936660	0.0000019769	0.0007993566
10	0.0000892631	-0.0003792043	-0.0011897167	0.0000018998	0.0000777706
11	0.0001247604	-0.0000369196	-0.0011360898	0.0000017328	0.0001489373
12	-0.0000602844	-0.0000354100	-0.0013245064	0.0000015140	0.0008207028
13	-0.0000243995	0.0003658964	-0.1011890340	-0.0000162084	0.0008621404
14	-0.0000031718	0.0001946900	-0.2010607815	-0.0000301454	0.0000390791
15	-0.0000248217	0.0004316107	-0.3009551582	-0.0000456814	0.0000837294
16	0.0000624220	-0.0169078104	-0.4007902885	-0.0000653493	0.0000184698
17	0.0000063416	0.0124022387	-0.4008843870	-0.0000642688	0.0000188417
18	-0.0000291056	-0.0217263982	-0.5006793213	-0.0000802804	0.0001976906
19	-0.0000407929	0.0111688341	-0.5006327251	-0.0000850517	0.0001831060
20	0.0000990414	0.0235220799	-0.5005691962	-0.0000816882	0.0001064097
21	-0.0532698386	-0.0054896792	-0.5005636825	-0.0000688531	0.0000180303
22	-0.0532381751	0.0079038693	-0.5005239688	-0.0000771728	0.0000312258
23	-0.0532703105	0.0110658410	-0.5005197832	-0.0000722352	0.0000126655
24	-0.0532305259	-0.0075470981	-0.5004803100	-0.0000766208	0.0003032580
25	-0.0533027025	0.0037270246	-0.5004436232	-0.0000752995	0.0000048353



Synthetic Test Function in 1D with Batch Size 4

Batch	Difference in Integral Absolute Error	Difference in Integrand SSE	Error Proportion to True Integral	Average Square Error to Integrand	Maxaximum Posterior Covariance of GP
1	-0.1361280073	3.4164712362	-0.2635490722	0.0201186520	0.0003457523
2	-0.0928537603	-1.4581164543	-0.1143203801	0.0055472384	0.0083575829
3	0.0176152217	0.0636422759	-0.0127333404	0.0005801235	0.2328357066
4	0.0013946022	-0.0482984409	-0.0088756587	0.0002064883	0.0090375037
5	0.0008946381	-0.0199812181	-0.0068404922	0.0001766685	0.0748511824
6	-0.0011942748	0.0760414587	-0.0072885568	0.0000848676	0.5980915390
7	0.0006913239	0.0252075072	-0.0086487012	0.0000440613	0.0044027963
8	0.0008191508	-0.0011007973	-0.0063355991	0.0000244353	0.0043903599
9	-0.0005601268	-0.0033402480	-0.0040776269	0.0000206410	0.0020153610
10	-0.0010920462	0.0005509788	-0.0014355416	0.0000144720	0.0030268180
11	-0.0005348281	-0.0093963318	-0.0013612633	0.0000184080	0.0258568241
12	-0.0000188358	-0.0030581863	-0.0021448171	0.0000120775	0.0007330382
13	-0.0003921539	-0.0047259854	-0.0018968115	0.0000078654	0.0018899570
14	0.0000274093	0.0005054268	-0.0015063983	0.0000042623	0.0007918046
15	0.0006204954	-0.0029818617	-0.0007884283	0.0000057515	0.0009973497
16	-0.0003339244	-0.0049584642	-0.0013787294	0.0000082514	0.0017348854
17	0.0000811045	-0.0001582928	-0.0010085320	0.0000064581	0.0007207196
18	0.0000216534	-0.0012599251	-0.0008035883	0.0000063518	0.0004423120
19	0.0000539456	-0.0003092189	-0.0006932005	0.0000050470	0.0004190985
20	0.0000076846	0.0000475343	-0.0005999260	0.0000040406	0.0006198999
21	0.0000500326	0.0003884775	-0.0005373410	0.0000033588	0.0037593359
22	-0.0000155385	-0.0001512119	-0.0004921830	0.0000028742	0.0071015710
23	-0.0000297385	-0.0001237086	-0.0003793202	0.0000017831	0.0038415814
24	0.0000094950	-0.0000275000	-0.0003036253	0.0000012168	0.0095595468
25	-0.0000066173	0.0001208268	-0.0003369571	0.0000009886	0.0008483147

Synthetic Test Function in 1D with Batch Size 8

Batch	Difference in Integral Absolute Error	Difference in Integrand SSE	Error Proportion to True Integral	Average Square Error to Integrand	Maxaximum Posterior Covariance of GP
1	0.1997518523	20.4981533237	-0.2183502144	0.0138958231	0.0000053126
2	0.0252264999	2.6730432895	0.0382245146	0.0004753056	0.0081843178
3	-0.0049647045	0.0955792897	0.0065510773	0.0000447612	0.2830179350
4	0.0010754001	0.1409790348	0.0011972702	0.0000308036	0.0612295079
5	0.0002710380	-0.0043383031	0.0010257402	0.0000162236	0.0090969558
6	0.0001264232	-0.0005067285	0.0010815930	0.0000101947	0.0010742733
7	0.0004963298	-0.0093019300	-0.0003666776	0.0000101525	0.0066426848
8	-0.0000933205	-0.0038065532	0.0001613410	0.0000063575	0.0006955538
9	-0.0003538183	0.0004721351	0.0003864965	0.0000038450	0.0001610468
10	-0.0002878909	-0.0006875733	0.0002913798	0.0000022874	0.0008381878
11	-0.0000972435	0.0000876522	0.0002011362	0.0000016277	0.0002453695
12	-0.0000224203	-0.0010828379	0.0001734083	0.0000025070	0.0002468292
13	0.0000683536	-0.0001513750	0.0001522090	0.0000015911	0.0004538721
14	0.0000134711	-0.0000079170	0.0000223860	0.0000010788	0.0020265781
15	0.0000361861	0.0000168871	-0.0000334142	0.0000005473	0.0001005942
16	-0.0000285289	0.0000793290	-0.0000721702	0.0000005042	0.0020567131
17	0.0000422554	-0.0003976773	0.0000197929	0.0000006501	0.0000893210
18	-0.0000091952	0.0000091428	0.0000336782	0.0000001642	0.0002036150
19	-0.0000087455	0.0001189272	0.0000103677	0.0000001210	0.0009003861
20	0.0000158467	-0.0000129252	-0.0000941379	0.0000000976	0.0001995000
21	0.0000148122	-0.0000122212	-0.0000908590	0.0000000708	0.0009288070
22	-0.0000269187	-0.0000059846	-0.0000968570	0.0000000622	0.0000483915
23	0.0000020892	-0.0000328451	-0.0000793299	0.0000000583	0.0000181073
24	-0.0000181468	0.0000098836	-0.0000789912	0.0000000231	0.0001341907
25	0.0000060819	-0.0000019648	-0.0000712697	0.0000000210	0.0000723267

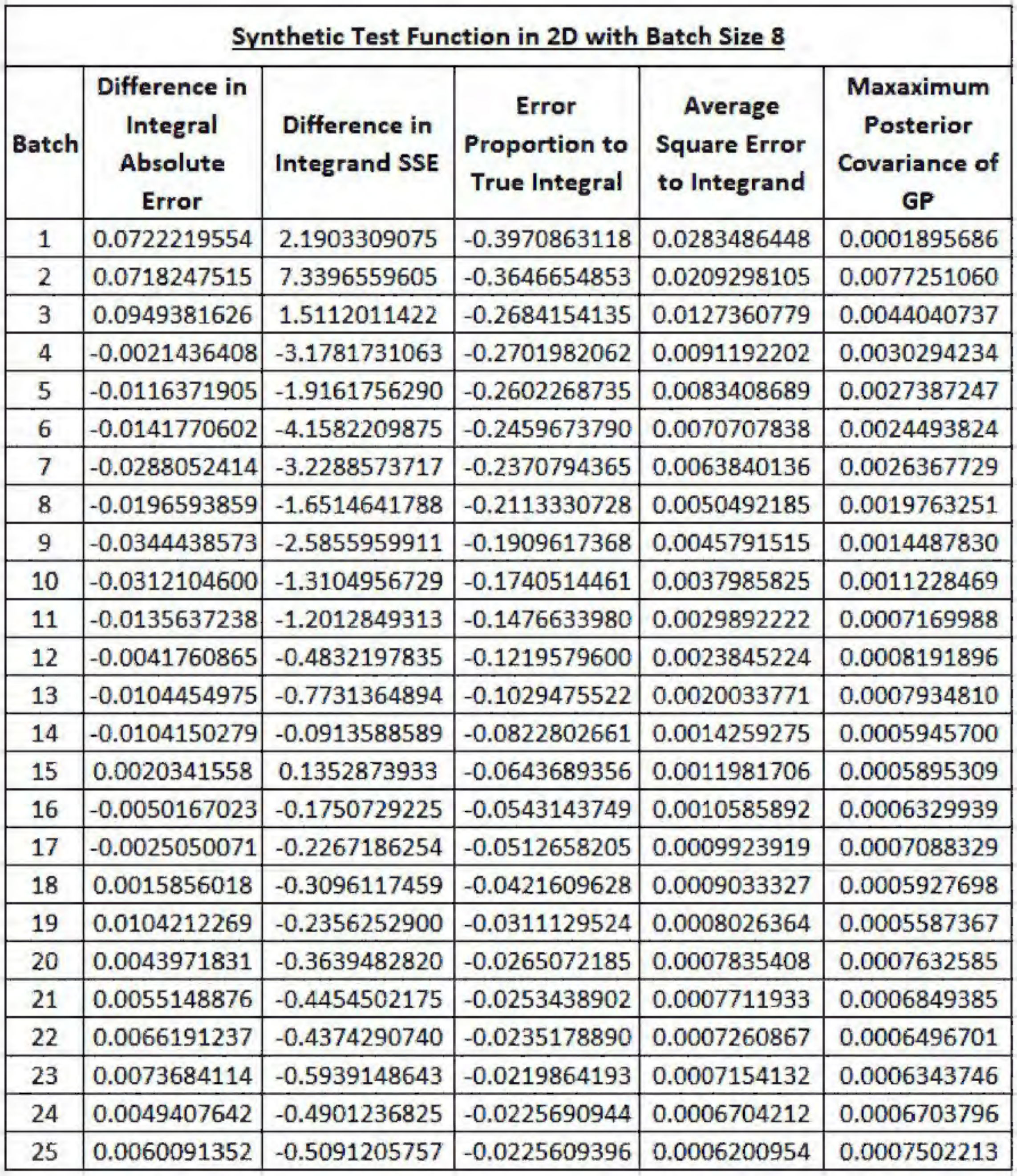
Ackley Test Function in 2D with Batch Size 4					
Batch	**Difference in Integral Absolute Error**	**Difference in Integrand SSE**	**Error Proportion to True Integral**	**Average Square Error to Integrand**	**Maxaximum Posterior Covariance of GP**
1	-0.2697484043	1.1092955552	-0.4859724226	0.0537546428	0.0011607158
2	-0.8861923021	4.4446270491	-0.3315811467	0.0513606346	0.0057083343
3	0.0362038111	3.3527106793	-0.2914732902	0.0465330273	0.0107107053
4	0.0953826289	3.7716597052	-0.2474160281	0.0429855797	0.0096251892
5	0.1057614791	-1.8321006079	-0.2003331638	0.0452055851	0.0099908500
6	-0.0038565490	5.8513578913	-0.1560909847	0.0416576667	0.0080538034
7	0.0058204229	-3.4888020986	-0.1462147790	0.0422060466	0.0066881198
8	0.0482557594	-3.8800043519	-0.1392449320	0.0429426572	0.0080664057
9	0.0394362073	-17.2422722851	-0.1466398968	0.0456916716	0.0142112902
10	0.1803029838	-15.2557620186	-0.1120965475	0.0486126035	0.0119272644
11	-0.0225733035	-3.3080006371	-0.1678247128	0.0348990369	0.0153496562
12	0.0453448743	-21.3439710094	-0.1710593615	0.0387449721	0.0150023532
13	0.0861037841	-7.4881802796	-0.1969773770	0.0271018098	0.0165291936
14	0.0380249111	-3.2884473158	-0.1989156870	0.0223253629	0.0209346523
15	0.0882564529	-4.2813601663	-0.1884224387	0.0197898261	0.0204738898
16	-0.0585392023	-1.7352906980	-0.2089103090	0.0171927030	0.0159125782
17	0.0220631929	-0.3696179175	-0.2013573149	0.0159856490	0.0121690890
18	0.0367412589	-1.0937190340	-0.1853035162	0.0151139119	0.0119157555
19	0.0035595787	-1.2477546761	-0.1752057208	0.0140462593	0.0103413527
20	0.0285498122	-1.2606077029	-0.1663752874	0.0131485759	0.0094870774
21	0.0590357037	0.0199790937	-0.1505808235	0.0121928426	0.0089223935
22	0.0083975694	-1.3465611812	-0.1467455589	0.0117471420	0.0085866088
23	-0.0169379763	-0.4028592976	-0.1456997649	0.0109762507	0.0079354975
24	0.0263263877	-2.1813657845	-0.1409749495	0.0113127628	0.0073713113
25	0.0022414524	-0.7580021456	-0.1423431604	0.0097597440	0.0093920302

Ackley Test Function in 2D with Batch Size 8					
Batch	**Difference in Integral Absolute Error**	**Difference in Integrand SSE**	**Error Proportion to True Integral**	**Average Square Error to Integrand**	**Maxaximum Posterior Covariance of GP**
1	-0.6271206768	19.8007458860	-0.2164559408	0.0383240295	0.0003168938
2	0.0414783294	-5.3944075913	-0.1247744958	0.0347532594	0.0012996318
3	-0.3036123709	-20.7816461340	-0.1501724526	0.0423845628	0.0019988587
4	0.1119594587	-9.0741875660	-0.1203159921	0.0392780795	0.0086086667
5	-0.0023198321	-41.6405303413	-0.0826580091	0.0544209111	0.0079041650
6	0.0863038806	3.1981778175	-0.1134730259	0.0326253872	0.0074768313
7	0.0527540004	-7.0541773716	-0.0898853588	0.0299112526	0.0084053430
8	-0.0837316246	3.7206501254	-0.1284728010	0.0191795263	0.0088272605
9	0.0056233891	-2.2476213780	-0.1231315072	0.0161509463	0.0111786176
10	0.0240472902	-6.5935180564	-0.1291889990	0.0140663231	0.0102399752
11	-0.0172925144	-1.6642542652	-0.1309013465	0.0107969208	0.0087412220
12	0.0106976701	-0.5239523696	-0.1188909040	0.0083334097	0.0079614999
13	-0.0013574026	-1.1775771390	-0.1107091463	0.0076557289	0.0063537598
14	0.0300589216	-1.2801099234	-0.0991903161	0.0067737133	0.0053110564
15	0.0277336738	-0.6591152350	-0.0935999042	0.0059123020	0.0049292702
16	0.0095387566	-0.1700346155	-0.0903090829	0.0052119041	0.0039534732
17	-0.0129353402	-0.7042739758	-0.0894259497	0.0047821197	0.0029958186
18	0.0050873454	-0.3801383505	-0.0851313229	0.0044914012	0.0030629690
19	0.0154264624	-0.0584448493	-0.0792211324	0.0041975549	0.0024966321
20	-0.0023053142	-0.2832487585	-0.0756248595	0.0038404011	0.0019258987
21	0.0092239546	-0.3672024125	-0.0704857605	0.0039115407	0.0017986851
22	-0.0026387207	-0.2027642732	-0.0675638313	0.0036289573	0.0014773500
23	-0.0054449577	-0.2202868779	-0.0641323870	0.0034716156	0.0012613977
24	0.0018788744	-0.2218090870	-0.0597378705	0.0033140590	0.0012841384
25	0.0042651483	-0.4074019517	-0.0555712086	0.0033409489	0.0010484945

Weierstrass Test Function in 2D with Batch Size 4					
Batch	Difference in Integral Absolute Error	Difference in Integrand SSE	Error Proportion to True Integral	Average Square Error to Integrand	Maxaximum Posterior Covariance of GP
1	0.1460791661	0.7104674418	-0.6059178979	0.0148439272	0.0001399625
2	0.0057853516	0.1513132555	-0.6832951731	0.0154815536	0.0003244136
3	0.0106412763	-0.2879987713	-0.5431104240	0.0148719025	0.0002141362
4	-0.1612670767	-2.0174280800	-0.7456111877	0.0156192775	0.0007709935
5	-0.0099201944	0.4128517269	-0.7193095813	0.0147256599	0.0057253819
6	-0.0356213796	-0.5098788607	-0.6731904086	0.0165144485	0.0178563161
7	0.0020302217	0.2956828996	-0.6500382474	0.0166916725	0.0186217321
8	0.0116779282	-0.4473776036	-0.6274593398	0.0167647107	0.0161275948
9	0.0140974530	-0.3434357241	-0.6187997430	0.0164072003	0.0183250772
10	0.1067129458	-3.4489963530	-0.6171990534	0.0160760369	0.0163848077
11	0.0873801781	-2.4138462471	-0.6307481586	0.0143102956	0.0156669760
12	0.0404429315	-1.6155381166	-0.6743799036	0.0132036343	0.0171400417
13	0.0141182580	-0.3293850561	-0.6907417742	0.0122481890	0.0098443011
14	0.0057784368	0.2481311540	-0.6867889315	0.0114968760	0.0108152249
15	0.0022530162	0.0949998488	-0.6699641556	0.0110139985	0.0074086707
16	0.0023848728	0.0692243349	-0.6599497279	0.0107541762	0.0071196827
17	0.0027347441	-0.1332709432	-0.6536454953	0.0106407157	0.0066885456
18	0.0058539377	0.0205359750	-0.6485738276	0.0105216907	0.0063142190
19	0.0086581610	-0.0412236902	-0.6313073578	0.0103720702	0.0058467881
20	0.0039002346	0.0733052417	-0.6320722813	0.0101771150	0.0065262614
21	0.0017534827	-0.0291804952	-0.6295477420	0.0101210552	0.0067432883
22	0.0056129610	-0.1325126982	-0.6260549648	0.0100097386	0.0063620792
23	0.0016555671	0.1238418605	-0.6260013819	0.0098109576	0.0069426358
24	-0.0003882930	-0.0268016813	-0.6246879522	0.0096522580	0.0064858777
25	-0.0020380638	-0.1270868185	-0.6226745779	0.0095692253	0.0060173723

Weierstrass Test Function in 2D with Batch Size 8					
Batch	**Difference in Integral Absolute Error**	**Difference in Integrand SSE**	**Error Proportion to True Integral**	**Average Square Error to Integrand**	**Maxaximum Posterior Covariance of GP**
1	0.2229448390	-4.1395781927	-0.1403357833	0.0156402090	0.0000150221
2	-0.1158816822	0.3662005370	-0.3773755645	0.0144261021	0.0003031120
3	-0.0264352745	-0.7358956813	-0.4012159025	0.0147197361	0.0004336770
4	-0.0753970948	-0.2556423403	-0.3954222491	0.0143265068	0.0004173613
5	0.0166336818	-1.5554422323	-0.3663922047	0.0148906473	0.0002776468
6	-0.0363137944	-0.8454332098	-0.3433893310	0.0149328770	0.0003686661
7	0.0134269188	-5.5025367604	-0.2871976294	0.0174009093	0.0004894313
8	0.0369480099	-2.6739186877	-0.3199283823	0.0164555547	0.0014200075
9	0.0278508123	-0.2550622154	-0.3365342055	0.0158741777	0.0026047297
10	0.0513853200	-0.7957888912	-0.3256655694	0.0160051766	0.0035513574
11	0.0372915118	-1.2027238024	-0.3505370575	0.0155241038	0.0051292929
12	0.0557344950	-2.6488984897	-0.3805234118	0.0145312027	0.0050014014
13	0.1140675937	-5.4416289355	-0.3725253727	0.0134547091	0.0066683380
14	0.0542622354	-1.5831720455	-0.4258908484	0.0106736620	0.0092806498
15	0.0052577936	-0.5648210154	-0.4481998290	0.0097209942	0.0083643703
16	-0.0102572921	0.1503112249	-0.4526037651	0.0092381049	0.0059803261
17	-0.0144287516	0.0744662308	-0.4539569974	0.0089921747	0.0058300432
18	0.0060948256	-0.0318304135	-0.4440393360	0.0088588317	0.0057759265
19	0.0040651484	0.0304596931	-0.4350335557	0.0087135738	0.0056971093
20	0.0005761738	-0.0110086975	-0.4313718270	0.0085556498	0.0057870561
21	-0.0023642826	-0.0116308034	-0.4326217929	0.0083497141	0.0055012049
22	-0.0046149600	0.0606771003	-0.4286218009	0.0081699289	0.0054178832
23	-0.0027016284	-0.0634594616	-0.4202546762	0.0081077905	0.0058318927
24	0.0002300446	0.0631077373	-0.4113234125	0.0079454545	0.0051710214
25	-0.0090801244	0.0384275510	-0.4093506445	0.0077651659	0.0048096228

Synthetic Test Function in 2D with Batch Size 4					
Batch	Difference in Integral Absolute Error	Difference in Integrand SSE	Error Proportion to True Integral	Average Square Error to Integrand	Maxaximum Posterior Covariance of GP
1	0.4909067456	5.6261557379	0.0283189671	0.0310070259	0.0005450577
2	0.1367001159	6.5026739843	0.0330355280	0.0301858016	0.0068629980
3	-0.0020192034	5.6907667277	-0.1109867653	0.0286795182	0.0099896125
4	0.0537547393	-1.9475348216	-0.2922299141	0.0169642301	0.0132361705
5	0.0749475797	7.3656356175	-0.3005283834	0.0122694714	0.0080100870
6	-0.0238579002	-0.6357073408	-0.2813043196	0.0091377168	0.0058975938
7	0.0025079476	-1.2413061946	-0.2642893151	0.0088877103	0.0070782593
8	0.0061903745	-1.2869127476	-0.2708360200	0.0082486037	0.0057262902
9	-0.0024818167	-1.2390812461	-0.2853486780	0.0079361167	0.0041429380
10	0.0028036129	-0.5550466654	-0.2729912476	0.0075456359	0.0042039181
11	0.0023751667	-0.1712527683	-0.2641046669	0.0070741793	0.0036236264
12	0.0029020100	0.2021279696	-0.2561501718	0.0066475081	0.0034568605
13	0.0009680435	-0.1302047971	-0.2500443567	0.0064441302	0.0032265062
14	-0.0007648100	-0.4299069470	-0.2428018733	0.0062064619	0.0028553607
15	-0.0078225896	-1.0788999080	-0.2356405479	0.0059481421	0.0020007454
16	0.0044161793	0.0539978682	-0.2249498607	0.0056961166	0.0019399315
17	-0.0007866910	-0.6910130348	-0.2061202979	0.0049918141	0.0036158203
18	-0.0105444101	-0.7697351776	-0.2057781853	0.0044461528	0.0017797710
19	-0.0124716069	-0.6368493176	-0.1938308272	0.0041519599	0.0015049717
20	0.0046270620	-0.3430142027	-0.1841226272	0.0038399579	0.0014604766
21	0.0121084963	-0.3234460616	-0.1658872005	0.0035440265	0.0012855121
22	0.0053003816	-0.2091796361	-0.1515657803	0.0029810545	0.0013402055
23	0.0050497006	-0.3649404792	-0.1374280846	0.0027212375	0.0013503382
24	0.0048488933	-0.1770908710	-0.1244029599	0.0024393248	0.0011818882
25	0.0007162649	-0.2793686693	-0.1132013060	0.0022381192	0.0011735159



Synthetic Test Function in 2D with Batch Size 8

Batch	Difference in Integral Absolute Error	Difference in Integrand SSE	Error Proportion to True Integral	Average Square Error to Integrand	Maxaximum Posterior Covariance of GP
1	0.0722219554	2.1903309075	-0.3970863118	0.0283486448	0.0001895686
2	0.0718247515	7.3396559605	-0.3646654853	0.0209298105	0.0077251060
3	0.0949381626	1.5112011422	-0.2684154135	0.0127360779	0.0044040737
4	-0.0021436408	-3.1781731063	-0.2701982062	0.0091192202	0.0030294234
5	-0.0116371905	-1.9161756290	-0.2602268735	0.0083408689	0.0027387247
6	-0.0141770602	-4.1582209875	-0.2459673790	0.0070707838	0.0024493824
7	-0.0288052414	-3.2288573717	-0.2370794365	0.0063840136	0.0026367729
8	-0.0196593859	-1.6514641788	-0.2113330728	0.0050492185	0.0019763251
9	-0.0344438573	-2.5855959911	-0.1909617368	0.0045791515	0.0014487830
10	-0.0312104600	-1.3104956729	-0.1740514461	0.0037985825	0.0011228469
11	-0.0135637238	-1.2012849313	-0.1476633980	0.0029892222	0.0007169988
12	-0.0041760865	-0.4832197835	-0.1219579600	0.0023845224	0.0008191896
13	-0.0104454975	-0.7731364894	-0.1029475522	0.0020033771	0.0007934810
14	-0.0104150279	-0.0913588589	-0.0822802661	0.0014259275	0.0005945700
15	0.0020341558	0.1352873933	-0.0643689356	0.0011981706	0.0005895309
16	-0.0050167023	-0.1750729225	-0.0543143749	0.0010585892	0.0006329939
17	-0.0025050071	-0.2267186254	-0.0512658205	0.0009923919	0.0007088329
18	0.0015856018	-0.3096117459	-0.0421609628	0.0009033327	0.0005927698
19	0.0104212269	-0.2356252900	-0.0311129524	0.0008026364	0.0005587367
20	0.0043971831	-0.3639482820	-0.0265072185	0.0007835408	0.0007632585
21	0.0055148876	-0.4454502175	-0.0253438902	0.0007711933	0.0006849385
22	0.0066191237	-0.4374290740	-0.0235178890	0.0007260867	0.0006496701
23	0.0073684114	-0.5939148643	-0.0219864193	0.0007154132	0.0006343746
24	0.0049407642	-0.4901236825	-0.0225690944	0.0006704212	0.0006703796
25	0.0060091352	-0.5091205757	-0.0225609396	0.0006200954	0.0007502213