Better Batch Optimizer

Yui Chun Leung

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Wolfson College August 2019

I would like to dedicate this dissertation to my loving parents . . .

Declaration

I, Yui Chun Leung of Wolfson College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report 'Better Batch Optimizer' and the work described in it are my own work, unaided except as may be specified below, and the report does not contain material that has already been used to any substantial extent for a comparable purpose.

MATLAB is the software used to develop codes for my work. The codes and notes from github directory `cambridge-mlg/minimize` are the primary starting points for the project. The report uses the template 'PhD Thesis Template for Cambridge University Engineering Department (CUED) v2.3.1' from Cambridge University Engineering Department.

This dissertation contains 14,145 words excluding bibliography, photographs, diagrams and declarations including appendices, bibliography, footnotes, figure captions, tables and equations.

Yui Chun Leung
August 2019

Acknowledgements

I would like to thank my supervisors Carl Edward Rasmussen and Manon Kok for help and guidance. Without their knowledge of Bayesian Optimization, I would not be able to draft my algorithm design. Learning from and Working with them have been an enjoyable and and intellectually stimulating experience for me.

A shout out goes out to the MLMI 2018-2019 cohort as well which has been very supportive and taught me much about sharing ideas and having fun together. I would never forget how we worked and struggled together in the MPhil room.

Abstract

BFGS [1] is one of the algorithms which can solve non-linear unconstrainted multivariate optimization problems. It firstly finds the search direction with the steepest gradient descent and performs line search along that direction to find the optimal step size minimizing the objective function with the satisfaction of strong Wolfe Condition. But due to its problematic termination criteria, the algorithm can stop prematurely under noisy search region.

We built a probabilistic line search algorithm by combining the smoothing quintic spline model with Bayesian optimization. Experiments showed that our algorithm can improve the accuracy of predicted step-size value from the line search algorithm but there were implementations problems causing a long processing time.

Contents

List of Figures **v**

List of Tables **vii**

1 Introduction **1**

1.1 Motivation 1

1.2 Dissertation Contribution 2

1.3 Dissertation Organization 2

2 Background **5**

2.1 BFGS 5

2.2 Line Searches 6

2.3 Challenges 6

2.4 Polynomial Regression 7

2.5 Spline model 8

2.5.1 Natural Cubic Spline 8

2.5.2 Natural Quintic Spline 9

2.5.3 Smoothing Spline 9

2.6 Bayesian Optimization 11

2.6.1 Gaussian Process 11

2.6.2 Sampling new point by Expected Improvement 18

3 Algorithm Design **21**

3.1 Related Works 21

3.2 Outline 22

3.3 Step-samples generation 25

3.4 Scaling factor 25

3.5 Prediction 27

3.6 Selection, Shift, Zoom and Retain 28

3.7 Range of stepsize 32

4 Experiments 35

4.1 One Dimensional Quadratic Problem 36

4.2 One Dimensional x^4 Problem 41

4.3 Rosenbrock Function Problem 46

4.4 Sigmoid Optimization Problem 48

4.5 Sinc Function 51

5 Conclusion 57

Bibliography 59

Appendix A pseudocode 61

A.1 BFGS 62

A.2 Line Search 63

A.3 zoom 63

Appendix B Smoothing Spline 65

Appendix C Numerical Considerations 71

Appendix D Negative log marginal likelihood 79

Appendix E Hyperparameter restimation 81

List of Figures

2.1 BFGS transforms the n-D problem into the 1-D line search problem. [1] denotes p_k as search direction, x_k as the starting vector and x^* as the optimal step-size 5

2.2 Differences between original curve, natural cubic spline and smoothing cubic spline. 11

2.3 Prediction difference between the cubic spline and the quintic spline. 14

2.4 Examples of expected improvement [2] predicted by 2 points. 19

2.5 Examples of expected improvement predicted by 3 points. 20

3.1 The example of rescaling the entries and their projected gradients. 26

3.2 How to initialize τ^2 and optimize it for the quintic spline model. 27

3.3 The new evaluation point is usually the location with the highest utility. . . 28

3.4 The example of zoom. 29

3.5 The example of the new starting subset leading to extrapolation for the next iteration. 30

3.6 The example of retaining data for the next iteration i. 31

3.7 Evolution of the potential interval $[\kappa, \kappa+\Delta]$ over iteration i for an one dimensional quadratic problem with $\sigma^2 = 0.01$. 33

4.1 Evolution of the potential interval $[\kappa, \kappa+\Delta]$ over iteration i for the one dimensional quadratic problem with $\sigma^2 = 0.01$ and different starting point. 38

4.2 Evolution of the potential interval $[\kappa, \kappa+\Delta]$ over iteration i in the one dimensional x^4 problem with $\sigma^2 = 0$. 41

4.3 Evolution of the potential interval $[\kappa, \kappa+\Delta]$ over iteration i in the one dimensional x^4 problem with $\sigma^2 = 0$ and different start settings. 44

4.4 Evolution of the potential interval $[\kappa, \kappa+\Delta]$ over iteration i in the rosenbrock problem with $\sigma^2 = 0.1$. 46

4.5 Evolution of the potential interval $[\kappa, \kappa + \Delta]$ over iteration in the sigmoid problem with $\sigma^2 = 0.0004$. 49
4.6 Evolution of the potential interval $[\kappa, \kappa + \Delta]$ over iteration in the sinc problem with $\sigma^2 = 0$. 52
4.7 Evolution of the potential interval $[\kappa, \kappa + \Delta]$ over iteration in the sinc problem with $\sigma^2 = 0$. 54

List of Tables

4.1 Evolution of the signal variance changed over iteration in the search range $[\kappa, \kappa + \delta\Delta]$ from Figure 3.7. 36

4.2 Influence of the function noise variance towards the performance. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic. 36

4.3 Influence of the retained data towards performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0.01,$ rescale=True,spline=quintic. 37

4.4 Influence of the range factor towards performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, N = 3,$ rescale=True,spline=quintic. 37

4.5 Influence of the rescale mode towards performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, \delta = 5, N = 4,$ spline=quintic. 37

4.6 The influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, \delta = 5, N = 4,$ rescale=True. 38

4.7 Evolution of the signal variance over iteration i in the search range $[\kappa, \kappa + \delta\Delta]$ from Figure 4.1. 39

4.8 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic. 39

4.9 Influence of the number of retained data towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0.01,$ rescale=True,spline=quintic. . . . 40

4.10 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, N = 3,$ rescale=True,spline=quintic. 40

4.11 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, \delta = 5, N = 4,$ spline=quintic. 40

4.12 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, \delta = 5, N = 4,$ rescale=True. 41

4.13 Evolution of the signal variance over iteration in the search range $[\kappa, \kappa + \delta\Delta]$ from Figure 4.2. 42

4.14 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic. 42
4.15 Influence of N towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0.1,$ rescale=True,spline=quintic. 42
4.16 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.1, N = 3,$ rescale=True,spline=quintic. 43
4.17 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.1, \delta = 5, N = 4,$ spline=quintic. 43
4.18 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.1, \delta = 5, N = 4,$ rescale=True. 43
4.19 Evolution of the signal variance over iteration i in the search range $[\kappa, \kappa + \delta\Delta]$ from Figure 4.3. 44
4.20 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic. 44
4.21 Influence of N towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0.1,$ rescale=True,spline=quintic. 45
4.22 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.1, N = 3,$ rescale=True,spline=quintic. 45
4.23 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.1, \delta = 5, N = 4,$ spline=quintic. 45
4.24 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.1, \delta = 5, N = 4,$ rescale=True. 46
4.25 Evolution of the signal variance over iteration in the search range $[\kappa, \kappa + \delta\Delta]$ from Figure 4.4. 47
4.26 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic. 47
4.27 Influence of N towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0.01,$ rescale=True,spline=quintic. 47
4.28 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, N = 3,$ rescale=True,spline=quintic. 48
4.29 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, \delta = 5, N = 3,$ spline=quintic. 48
4.30 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, \delta = 6, N = 3,$ rescale=True. 48
4.31 Evolution of the signal variance over iteration in the search range $[\kappa, \kappa + \delta\Delta]$ from Figure 4.5. 49

4.32 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic. 50
4.33 Influence of N towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0.0001,$ rescale=True,spline=quintic. 50
4.34 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.0001, N = 3,$ rescale=True,spline=quintic. 50
4.35 Influence of the retain condition towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 4, N = 3, \sigma^2 = 0.0001,$ rescale=True,spline=quintic. 51
4.36 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.0001, \delta = 4, N = 3,$ spline=quintic. 51
4.37 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.0001, \delta = 4, N = 3,$ rescale=True. 51
4.38 Evolution of the signal variance over iteration in the search range $[\kappa, \kappa + \delta\Delta]$ from Figure 4.6. 52
4.39 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic. 53
4.40 Influence of N towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0,$ rescale=True,spline=quintic. 53
4.41 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0, N = 3,$ rescale=True,spline=quintic. 53
4.42 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0, \delta = 5, N = 4,$ spline=quintic. 54
4.43 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0, \delta = 5, N = 4,$ rescale=True. 54
4.44 Evolution of the signal variance over iteration in the search range $[\kappa, \kappa + \delta\Delta]$ from Figure 4.7. 55
4.45 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic. 55
4.46 Influence of N towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0,$ rescale=True,spline=quintic. 55
4.47 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0, N = 3,$ rescale=True,spline=quintic. 56
4.48 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0, \delta = 5, N = 4,$ spline=quintic. 56
4.49 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0, \delta = 5, N = 4,$ rescale=True. 56

Chapter 1

Introduction

1.1 Motivation

Numerical optimization of continuous multivariate objectives is a standard requirement in most numerate fields (including machine learning). It is useful to find optimal solution on different kinds of optimization problems, such as finding solutions for minimizing manufacturing cost and optimizing the hyperparameters settings in the deep neural network.
In recent years, a great success of the deep neural network has influenced and transformed many industries in the world. Its robust performance has not been possible without the batch optimizer behind the architecture, for example, Stochastic Gradient Descent (SGD) [3] and Adaptive Moment Estimation (Adam) [4], in order to find out the optimal weight settings to minimize the cost function. However, the parameters settings behind most of the state-of-art optimizers require manual specification by developers or teams, which is not efficient because it requires many testings on the algorithm with different settings. These algorithms and designs can be viewed as an optimization of a black-box objective function which is sometimes unknown and expensive to evaluate. And the input are the hyperparameters, and the objective function value is the output performance such as accuracy [5].
Bayesian Optimization, with the use of Gaussian Process [6] and Expected Improvement [2], is a sample-efficient method for global optimization for such black-box functions using probabilistic methods and is gaining great popularity to improve the performance of machine learning algorithms [7][8][9]. A recent study [10] has shown that one of the important parameters in SGD, the learning rate, can be specified automatically by Bayesian Optimization method. Their method involves modelling an univariate objective (the loss function in the neural network) as a stochastic process and employing a probabilistic line search (an extension of the line search algorithm [1]) to find the optimal settings.
On other hand, BFGS [1] is an algorithm which is built directly upon the line search method.

It relies on a number of idealized assumptions including noise-free observations, which are seldom realistic in practice because the observations can be influenced by the stochastic noise. This causes it to slow down or stop prematurely, thus failing to deliver a better solution in the application. To develop a better batch optimizer, we can scale down the project into developing a better line search algorithm which resembles the approach of the probabilistic line search and can improve the performance of the BFGS algorithm.
In comparison to the probabilistic line search [10] with the commonly-used smoothing cubic spline model [11] which implicitly assumes the second derivative of the objective is not continuously differentiable, a different implementation of the probabilistic line search with the smoothing quintic spline which does not have that implicit assumption is adopted to improve the performance of BFGS in this project. This approach should also solve the problems caused by the noisy observations to BFGS.

1.2 Dissertation Contribution

In this dissertation, we have a review of theories behind Bayesian Optimization, including Gaussian Process, Expected Improvement, spline models and their variations. Based on the theories, we design a probabilistic line search algorithm which is capable to improve the accuracy of the prediction from the line search and the performance of BFGS.
We incorporate the theory of quintic spline, which is not commonly used in the field, into the stochastic process, and develop a relevant set of codes for it. The work also has included a discussion about the stopping criteria of the expected improvement which is seldom mentioned in other research works, and the proposed algorithm is tested with different optimization problems in this paper. Although the expected performance is not observed due to implementation problems, we carry out a detailed analysis on the experimental results and point out the potential design problems. The improvements which can be made are also presented in the future work.
The codes developed are uploaded to the github directory and can be reused in the future work if other researchers would like to modify the state-of-art optimization algorithm with the probabilistic line search and the quintic spline instead of the cubic spline.

1.3 Dissertation Organization

The rest of this dissertation is organised as follows. Chapter 2 briefly introduces the background knowledge for BFGS algorithm, Line search algorithm, Polynomial regression, Spline model and Bayesian Optimization. Chapter 3 presents the algorithm design of probabilistic line search based on the theories mentioned in Chapter 2. The performances of the algorithm with various parameters settings on different optimization problems are discussed in chapter 4. Chapter 5 gives the conclusion and suggests the future work.

Chapter 2

Background

2.1 BFGS

Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [1] (Appendix A.1) is an iterative method for solving continuously-differentiable unconstrainted non-linear multivariate optimization problems.
Consider, with access to the objective function $f(\mathbf{s}) \in \mathbb{R}$ as a function of multiple inputs denoted by a vector $\mathbf{s} \in \mathbb{R}^D$ and a starting vector $\mathbf{s_e}$, it firstly computes the gradient $\nabla f(\mathbf{s_e}) \in \mathbb{R}^D$ to find the search direction $\mathbf{t_e}$ with the steepest descent. Along that search direction, with access to the objective function $\phi(x) = f(\mathbf{s_e} + x\mathbf{t_e})$ and the projected gradient $\phi'(x) = \mathbf{t_e} \cdot \nabla f(\mathbf{s_e} + x\mathbf{t_e})$, the multivariate optimization problem turns into an univariate problem of step-size x. The algorithm performs line searches [1] (Appendix A.2) to find the optimal step-size x_e which can minimize the objective function along that direction.

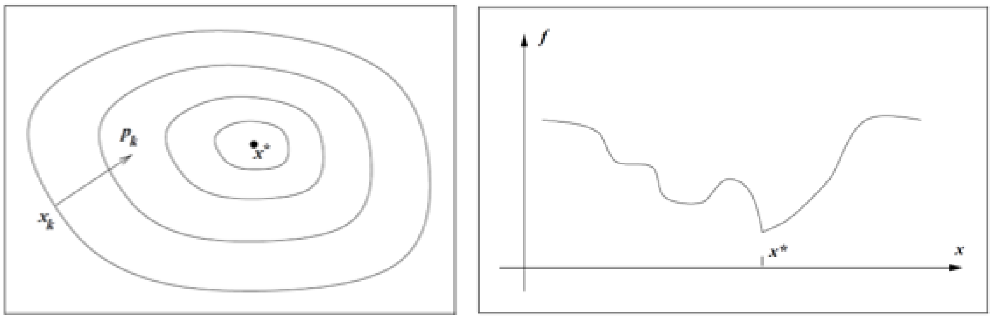

Figure 2.1 BFGS transforms the n-D problem into the 1-D line search problem. [1] denotes p_k as search direction, x_k as the starting vector and x^* as the optimal step-size

There are a set of conditions, named the strong Wolfe Conditions, which are required to satisfy during this process. These strong Wolfe Conditions include the sufficient decrease condition (eq.2.1) and the curvature condition (eq.2.2), which ensures the sufficient iterative improvement.

$$\phi(x) \leq \phi(0) + c_1 x \phi'(0) \tag{2.1}$$

$$|\phi'(x)| \geq c_2 |\phi'(0)| \tag{2.2}$$

where $c_1 = 0.05$ and $c_2 = 0.8$ are the constants chosen by the designer of the line search and satisfy the condition $0 \leq c_1 \leq c_2 \leq 1$.
The search then repeats the process with the new starting vector $\mathbf{s}'_\mathbf{e} = \mathbf{s}_\mathbf{e} + x_e \mathbf{t}_\mathbf{e}$ until it is terminated by the stopping criteria.
The main idea of this algorithm is to transform D-dimensional problem into several one-dimensional line search problems which have less tuning parameters. So, it requires less computation power.

2.2 Line Searches

The line search [12] is a method to optimize one-dimensional problem $\phi(x)$ by exploring the positive univariate domain of step-size $x \in \mathbb{R}_+$ until an acceptable step-size $x' = x_e$ is reached.
The main idea of the algorithm is to collect the scalar function values and the projected gradient values of location x_j to predict the optimal solution $x^* \approx x_e$.
Most line searches begin with an initial extrapolation to find the maximum search range x_r which violates strong Wolfe conditions to localize one of the possible local minima. Mutliple repetitive interpolations with various entries $x \in [0, x_r]$, their objective and gradient values in the interval $[0, x_r]$ then follow to narrow down the search interval $[x_{r,min}, x_{r,max}]$ iteratively (Appendix A.3) until the acceptable entry $x' \in [x_{r,min}, x_{r,max}]$ satisfying the conditions is found.

2.3 Challenges

There are three main issues in the line search algorithm.
Firstly, its limited capability of explorations affects its effectiveness of finding an optimal solution. Consider the algorithm usually takes the first step which violates the strong Wolfe Conditions to define the maximum search range $x_1 = x_r$, this search interval can be defined improperly when the objective and gradient values are noisy. Especially when the maximum range is defined mistakenly as $x_r < x^*$,the line search should not be able to find x^* because it cannot extrapolate out of the interval once the interval is decided. To make the search more

accurate, the search can be defined as wide as possible randomly, i.e. $x_r >> x^*$, but it may slow down the search process. Thus, it is difficult for the algorithm to define the best range which can affect the performance.
Secondly, the algorithm is sensitive to the stochastic noise which can terminate the search prematurely due to the wrong satisfaction of the strong Wolfe Condition.
Thirdly, the interpolation with polynomial regression or natural cubic spline has a limited capability to model the observations effectively. More specifically, it does not take the noise or the deviation in the observations into account so the prediction can be problematic.
Apart from the issues, when developing a rock-stable production-ready implementation, it is challenging to ensure the accuracy and the precision of the predictions. So that the performance of the algorithm is not susceptible to different initializations and it should not have different predicted values with great deviation. Also, the processing time is an essential criteria which determines the robustness of the algorithm because a single line search takes less than a second, the modified model should not add more than a second of processing time and finding the better solution should not require infinite processing time.
Before encountering those challenges, we firstly analyze the problems behind the line search algorithm by discussing the disadvantages of the traditional interpolation methods, including the polynomial regression and the natural spline model. Next, we introduce the more advanced methods, such as the smoothing spline model and the Bayesian Optimization, which are useful for the construction of the modified model to resolve the issues and satisfy the criteria mentioned above.

2.4 Polynomial Regression

To interpolate the search region, one of the commonly used method is the polynomial regression model. The noisy objective function y can be modelled as n^{th} degree polynomial as follows [13],

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_n x^n + \varepsilon \tag{2.3}$$

where $\beta = \{\beta_i\}_{i=0}^{n}$ are the unknown parameters, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and σ^2 is the function noise variance.
By stacking the entries and their observations from x, y to $\mathbf{X}, \mathbf{Y}$ respectively, the polynomial regression coefficients can be computed with the least squared analysis by,

$$\beta = (\mathbf{X}^\top \mathbf{X})^{-\mathbf{1}} \mathbf{X}^\top \mathbf{y} \tag{2.4}$$

But the textbook [11] proves mathematically that, Inputting the uniformly-spaced data into the polynomial regression model can lead to bad prediction when the number of data and the order of polynomial increase, which is problematic since the line search relies on evaluating the function many times to find the optimal solution.

2.5 Spline model

To overcome the problem from the polynomial regression, the spline model [11] is introduced. Given a set of entries $\{x_j\}_{j=0}^{n}$ and their observations, a piecewise N-th order of polynomial g is fit within an interval of two consecutive entries $[x_{j'}, x_{j'+1}]$, where $j' \in [0, n-1]$. The solution should satisfy the boundary conditions stating that the function value of g and its first $(N-1)$th derivatives must be continuous. As the number of constraints is not enough for solving all the unknowns in the spline model, there are two extra conditions added stating that, the $(N-1)$th derivatives of the polynomial with the entries of x_0 and x_n should equal to zero. And this spline model is named the natural spline.

2.5.1 Natural Cubic Spline

A cubic spline is a spline constructed of piecewise third-order polynomials. With a set of $n+1$ observation points $(y_0, ...y_n)$, the i-th piece of the natural cubic spline has the following form [14][15],

$$g_i(t) = a_i + b_i t + c_i t^2 + d_i t^3 \tag{2.5}$$

where t is a parameter $t \in [0,1]$, $i = 0, ...n-1$ and a_i, b_i, c_i, d_i are the unknown coefficients of the natural cubic spline. The function has the following properties,

1. $g_i(t)$ is a third degree polynomial in each interval.
2. $g_i(t)$ and its first two derivatives are continuous, for example, $g_i(0) = g_{i-1}(i) = y_i, i = 1, 2, ..., n-1$.
3. $g_0''(0) = g_{n-1}''(1) = 0$

These should give sufficient constraints to solve for the unknowns give by,

$$
\begin{aligned}
a_i &= y_i \\
b_i &= D_i = g_i'(0) \\
c_i &= 3(y_{i+1} - y_i) - 2D_i - D_{i+1} \\
d_i &= 2(y_i - y_{i+1}) + D_i + D_{i+1}
\end{aligned}
$$

2.5.2 Natural Quintic Spline

A quintic spline is a spline constructed of piecewise fifth-order polynomials. Simiar to the formulation of natural cubic spline, the natural quintic splines are of the form

$$g_i(t) = y_i + b_i t + c_i t^2 + d_i t^3 + e_i t^4 + f_i t^5 \tag{2.6}$$

where t is a parameter $t \in [0,1]$, $i = 0, ...n-1$ and $a_i, b_i, c_i, d_i, e_i, f_i$ are the unknown coefficients of the natural quintic spline. The function has the following properties,

1. $g_i(t)$ is a fifth degree polynomial in each interval.
2. $g_i(t)$ and its first four derivatives, upto and including $g_i''''(t)$ are continuous.
3. $g_0'''(0) = g_{n-1}'''(1) = g_0''''(0) = g_{n-1}''''(1) = 0$

which are enough to solve the unknowns.
The difference between natural cubic and quintic splines is the constraints they have. In theory, the natural quintic splines are capable to model more structured data with more constraints, while the natural cubic splines can model data with higher flexibility.
However, the natural splines have the problem of overfitting under least squared estimation when the data is noisy even though they perform better than the polynomial regression.

2.5.3 Smoothing Spline

The overfitting problem in the natural spline can be resolved by adding a regularized term (or smoothing term) which integrates the m-th derivative of the spline model. This model is named as the smoothing spline.
The whole process of fitting data with smoothing spline is described as finding the function

which minimizes the following,

$$\frac{1}{n}\sum_{i=1}^{n}(g(x_i)-y(x_i))^2+\lambda\int_{-\infty}^{\infty}(g^{(m)}(x))^2dx \tag{2.7}$$

where g is the smoothing spline function, $g^{(m)}$ is the m-th derivative of the smoothing spline function, $x_i \in (0,1)$ the entry, n is the number of entries and λ smoothing parameter which can affect how much deviation (observation noise) the smoothing spline should model.
The solution of the smoothing spline is a polynomial of degree $m-1$ for $x \in [-\infty,0] \cup [1,\infty]$ and a piecewise of degree $2m-1$ with $2m-2$ continuous derivatives for the interval of $x \in (0,1)$. The settings of $m=2$ correspond to cubic spline and $m=3$ corresponds to quintic spline.
The smoothing term can be regarded as integrating the white noise in the m-th derivative of the objective. So it implies that, the cubic spline implicitly assumes the 2nd derivative is not continuously differentiable and the quintic spline implicitly assumes the 3rd derivative is not continuously differentiable. In theory, the quintic spline should be able to retain more structured information from the objective.
The same solution can be obtained by using Bayesian inference of $g(x)$ with an uninformative prior, which is further discussed in the next chapter, as follows,

$$g_{prior}(x)=f(x)+\sum_{j=0}^{m-1}\beta_j x^j=\int_0^1\frac{(x-u)_+^{m-1}}{(m-1)!}Z(u)du+\sum_{j=0}^{m-1}\beta_j x^j \tag{2.8}$$

where the first term $f(x)$ is the $(m-1)$-fold integrated random walk, $Z(u)$ is a Gaussian white noise process with covariance $\delta(u-u')$, the second term is a polynomial of degree $m-1$ with uninformative prior $\beta_j \sim \mathcal{N}(0,\sigma_\beta^2), \sigma_\beta \to \infty$.

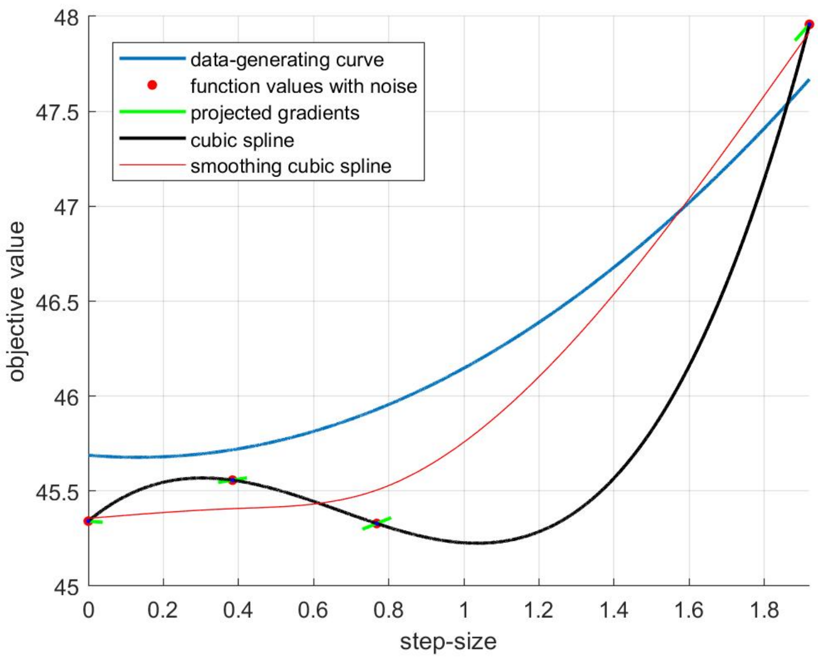

Figure 2.2 Differences between original curve, natural cubic spline and smoothing cubic spline.

With comparison of the natural spline with the smoothing spline, Figure 2.2 shows that fitting noisy data with the natural cubic spline model can cause overfitting and is more susceptible to noise, causing the bad predictive curve. But with the smoothing cubic spline, it can give a better prediction on how the data-generating curve looks like, thus improving the performance of line searches.

2.6 Bayesian Optimization

2.6.1 Gaussian Process

The objective function can be modelled as a Gaussian Process by the following equation,

$$f(\mathbf{x}) \sim \mathscr{GP}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}')) \tag{2.9}$$

with a mean function of 0 and covariance function of $k(x, x')$.
The noisy observation value y (with objective noise variance σ^2) can be modelled as,

$$y(\mathbf{x}) \sim \mathscr{N}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}') + \sigma^2) \tag{2.10}$$

By minimizing the negative log-marginal likelihood (nlml) with respect to the hyperparameters (e.g. σ^2), the best settings of the hyperparameters can be computed to fit the data with the model which can give the predictive distribution of the noise-free data f_*.
The predictive distribution is given by,

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{matrix} k(\mathbf{x},\mathbf{x})+\sigma^2 & k(\mathbf{x},\mathbf{x}_*) \\ k(\mathbf{x}_*,\mathbf{x}) & k(\mathbf{x}_*,\mathbf{x}_*) \end{matrix}\right) \tag{2.11}$$

So,

$$\begin{aligned} & f_*|y,\mathbf{x},\mathbf{x}_* \sim \mathcal{N}(\mu_*, cov_*) \qquad \text{, where} \\ & \mu_* = k(\mathbf{x}_*,\mathbf{x})[k(\mathbf{x},\mathbf{x})+\sigma^2]^{-1}y \\ & cov_* = k(\mathbf{x}_*,\mathbf{x}_*) - k(\mathbf{x}_*,\mathbf{x})[k(\mathbf{x},\mathbf{x})+\sigma^2]^{-1}k(\mathbf{x},\mathbf{x}_*) \end{aligned} \tag{2.12}$$

By stacking the entries $\mathbf{x},\mathbf{x}_*$ and their observations y, f_*, it can be rewritten as,

$$\begin{aligned} & \mathbf{f}_*|\mathbf{y},\mathbf{X},\mathbf{X}_* \sim \mathcal{N}(\mu_*, \mathbf{cov}_*) \qquad \text{, where} \\ & \mu_* = \mathbf{K}(\mathbf{X}_*,\mathbf{X})[\mathbf{K}(\mathbf{X},\mathbf{X})+\sigma^2\mathbf{I}]^{-1}\mathbf{y} \\ & \mathbf{cov}_* = \mathbf{K}(\mathbf{X}_*,\mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*,\mathbf{X})[\mathbf{K}(\mathbf{X},\mathbf{X})+\sigma^2\mathbf{I}]^{-1}\mathbf{K}(\mathbf{X},\mathbf{X}_*) \end{aligned} \tag{2.13}$$

where we denote $\mathbf{K}_\mathbf{f}^* = \mathbf{K}(\mathbf{X}_*,\mathbf{X})$ and $\mathbf{K}_\mathbf{y} = \mathbf{K}(\mathbf{X},\mathbf{X})+\sigma^2\mathbf{I} = \mathbf{K}_\mathbf{f}+\sigma^2\mathbf{I}$.
To express the prior information, the use of explicit basis functions is a way to specify a non-zero mean over the Gaussian process. This can be accomplished by specifying a few fixed basis functions, whose coefficients β, to be inferred from the data. Consider [6]

$$g(\mathbf{x}) = f(\mathbf{x}) + \mathbf{h}(\mathbf{x})^\top \beta \tag{2.14}$$

where $f(\mathbf{x})$ is a zero mean $\mathcal{GP}$, $\mathbf{h}(\mathbf{x}) = (1, x, ...)$ is a set of fixed basis functions and $\beta \sim \mathcal{N}(\mathbf{b},\mathbf{B})$ are additional parameters (as mentioned in Section 2.5.3). Therefore, we obtain another $\mathcal{GP}$,

$$g(\mathbf{x}) \sim \mathcal{GP}(\mathbf{h}(\mathbf{x})^\top\mathbf{b}, k(\mathbf{x},\mathbf{x}') + \mathbf{h}(\mathbf{x})^\top\mathbf{B}\mathbf{h}(\mathbf{x}')) \tag{2.15}$$

With the uninformative prior limit of $\mathbf{B} \to \mathbf{0}$, it is assumed that β can take any settings in an uniform distribution ($\mathcal{N} \sim \mathbb{U}$), the predictive distribution becomes [6],

$$\begin{aligned} \mu_{x^\star|\mathbf{y},\mathbf{X}} &= \mathbf{K_f^*}\mathbf{K_y^{-1}}\left(\mathbf{y} - \mathbf{H}^\top\overline{\beta}\right) + \mathbf{H}_*^\top\overline{\beta} \\ \sigma^2{}_{x^\star|\mathbf{y},\mathbf{X}} &= \mathbf{K_f}(\mathbf{x}^\star, \mathbf{x}^\star) - \mathbf{K_f^*}\mathbf{K_y^{-1}}(\mathbf{K_f^*})^\top + \mathbf{R}^\top\mathbf{A}^{-1}\mathbf{R} \end{aligned} \tag{2.16}$$

where $\mathbf{H} = \mathbf{H}(\mathbf{X})$ is the stacked vector of the basis functions $\mathbf{h}$, $\mathbf{H}_* = \mathbf{H}(\mathbf{X}_*)$, $\bar{\beta} = \mathbf{A}^{-1}\mathbf{H}\mathbf{K_f^{-1}}\mathbf{y}$, $\mathbf{R} = \mathbf{H}_* - \mathbf{H}\mathbf{K_y^{-1}}\mathbf{K_f^*}$ and $\mathbf{A} = \mathbf{H}\mathbf{K_y^{-1}}\mathbf{H}^\top$.

Another important step to model the objective as the stochastic process is choosing the covariance function. It is assumed that the prediction should not be stationary and uniform over the positive univariate domain x. This means that, the local minima should not occur everywhere and the pattern of the covariance function should not be repetitive over the positive univariate domain. Therefore, the stationary covariance function, such as the squared exponetial, can not be used. It is also assumed that the local minima should be located in the interval where the data exists, the observations outside that interval diverge. This allows the algorithm to interpolate within the confident interval where the data exists, and extrapolate outside the interval due to uncertainty.

As the smoothing spline model (Subsection 2.5.3) has different solutions for the region within and outside the data interval and its formulation is related to the uninformative prior described by equations 2.14, 2.15 and 2.16, the characteristics of the smoothing spline covariance function are well-suited to the application.

Another advantage of using this covariance function over others is that it has only one hyperparameter to be estimated, which is shown later in this chapter. This can reduce the computation power and the processing time to make the algorithm more robust.

Given we proceed with the smoothing spline covariance function, as our algorithm relies on both the objective and gradient information to find the optimal solution, the projected gradient value should also be modelled as a Gaussian Process. Therefore, the Gaussian process can then be written as,

$$\begin{bmatrix} f \\ f' \end{bmatrix} \sim \mathcal{GP}\left(\mathbf{0}, \begin{bmatrix} k & k^{\partial} \\ {}^{\partial}k & {}^{\partial}k^{\partial} \end{bmatrix}\right) \tag{2.17}$$

where ${}^{\partial^i}k^{\partial^j} = \frac{\partial^{i+j}k(x,x')}{\partial x^i \partial x'^j}$. This equation is an extent of equation 2.9.

Another reason of choosing the spline model is that the deviations (noises) of the objective values and projected gradient values can be modelled by the basis functions. There are two spline models used in this paper, such as cubic spline and quintic spline mentioned before.

The main difference between these two spline models is that the quintic spline uses the quadratic basis function to model the objective and gradient values' deviations, while the cubic spline uses the linear basis function.

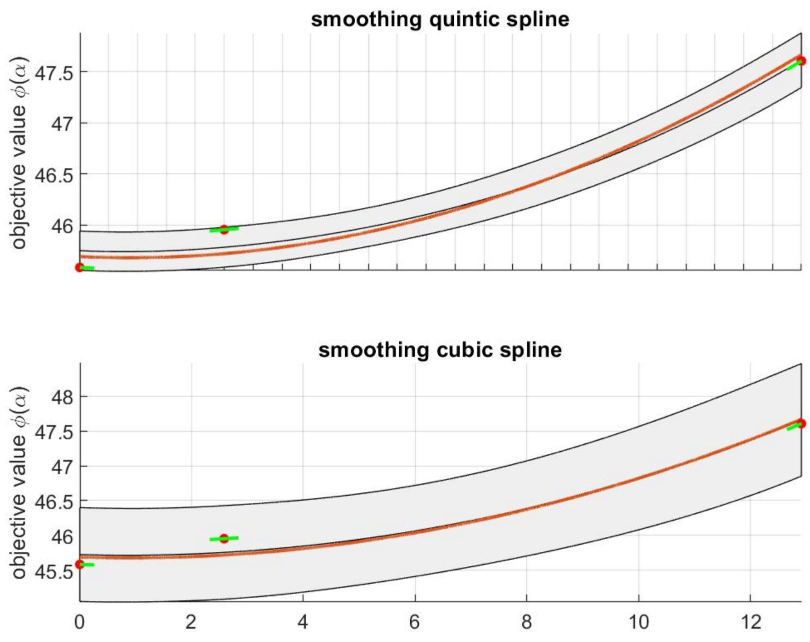

Figure 2.3 Prediction difference between the cubic spline and the quintic spline.

Refer to Figure 2.3, it can be observed that the uncertainty bar denoted by the grey shaded area of the quintic spline is thinner than that of the cubic spline. This shows the quintic spline can model more prior from the data.
Smoothing cubic spline covariance has the form of,

$$
\begin{aligned}
k &= \theta^2 \left(1/3 \min(\mathbf{x},\mathbf{x}')^3 + 1/2|\mathbf{x}-\mathbf{x}'| \min(\mathbf{x},\mathbf{x}')^2)\right) \\
k^{\partial} &= \theta^2/2((\mathbf{x}-\mathbf{x}') \min(\mathbf{x},\mathbf{x}') + \mathbf{x}\mathbf{x}') \\
{}^{\partial}k &= \theta^2/2((\mathbf{x}'-\mathbf{x}) \min(\mathbf{x},\mathbf{x}') + \mathbf{x}\mathbf{x}') \\
{}^{\partial}k^{\partial} &= \theta^2 \min(\mathbf{x},\mathbf{x}')
\end{aligned}
\tag{2.18}
$$

Smoothing quintic spline covariance has the form of,

$$\begin{aligned}
k &= \theta^2/4(1/5\min(\mathbf{x},\mathbf{x}')^5 + 1/2|\mathbf{x}-\mathbf{x}'|\min(\mathbf{x},\mathbf{x}')^4 + 1/3(\mathbf{x}-\mathbf{x}')^2\min(\mathbf{x},\mathbf{x}')^3) \\
k^{\partial} &= \theta^2/4(2/3(\mathbf{x}'-\mathbf{x})\min(\mathbf{x},\mathbf{x}')^4 + 1/2(\mathbf{x}^2-\mathbf{x}'^2)\min(\mathbf{x},\mathbf{x}')^2 + 1/2(\mathbf{x}\mathbf{x}')^2) \\
{}^{\partial}k &= \theta^2/4(2/3(\mathbf{x}-\mathbf{x}')\min(\mathbf{x},\mathbf{x}')^4 + 1/2(\mathbf{x}'^2-\mathbf{x}^2)\min(\mathbf{x},\mathbf{x}')^2 + 1/2(\mathbf{x}\mathbf{x}')^2) \\
{}^{\partial}k^{\partial} &= \theta^2(1/3\min(\mathbf{x},\mathbf{x}')^3 + 1/2|\mathbf{x}-\mathbf{x}'|\min(\mathbf{x},\mathbf{x}')) \qquad (2.19)
\end{aligned}$$

where θ^2 is the signal variance, and we denote the covariance function as follows,

$$\mathbf{K_f} = \mathbf{K_{spline}} = \begin{bmatrix} k & k^{\partial} \\ {}^{\partial}k & {}^{\partial}k^{\partial} \end{bmatrix} \qquad (2.20)$$

These equations for the smoothing spline covariances can be derived from equation 2.7. The detailed derivations can be seen from Appendix B.
The algorithm can access only noisy objective and projected gradient values y_x, y'_x at location x, with Gaussian likelihood given the Gaussian process f,

$$\begin{bmatrix} y_x \\ y'_x \end{bmatrix} \sim \mathscr{N}\left(\begin{bmatrix} f_x \\ f'_x \end{bmatrix}, \begin{bmatrix} \sigma^2 & 0 \\ 0 & \rho^2 \end{bmatrix}\right) \qquad (2.21)$$

where σ^2 is the function value noise variance and ρ^2 the gradient value noise variance.
Given a set of evaluations $(\mathbf{x},\mathbf{y},\mathbf{y}')$(training set of vectors, with elements x_i, y_{x_i}, y'_{x_i} and $1 \leq i \leq T$, where T is the number of entries), the vector of observations can be modelled as follows,

$$\begin{aligned}
\mathbf{Y} = \begin{bmatrix} \mathbf{y} \\ \mathbf{y}' \end{bmatrix} &\sim \mathscr{N}\left(\begin{bmatrix} f \\ f' \end{bmatrix}, \begin{bmatrix} \sigma^2\mathbf{I} & 0 \\ 0 & \rho^2\mathbf{I} \end{bmatrix}\right) \\
&\sim \mathscr{N}\left(\mathbf{0}, \mathbf{K_f} + \begin{bmatrix} \sigma^2\mathbf{I} & 0 \\ 0 & \rho^2\mathbf{I} \end{bmatrix}\right) \\
&\sim \mathscr{N}\left(\mathbf{0}, \mathbf{K_f} + \Sigma_{noise}\right) \qquad (2.22)
\end{aligned}$$

Similar to equation 2.15, another Gaussian process can be obtained,

$$\begin{bmatrix} g(\mathbf{x}) \\ g'(\mathbf{x}) \end{bmatrix} \sim \mathscr{GP}\left(\mathbf{H}^\top\mathbf{b}, \mathbf{K_f} + \mathbf{H}^\top\mathbf{B}\mathbf{H}\right)$$

where $\mathbf{H} = [\mathbf{h}(\mathbf{x}), \mathbf{h}'(\mathbf{x})]^\top$ and we denote the new covariance function as follows,

$$\mathbf{K} = \mathbf{K_f} + \mathbf{HBH}^\top \tag{2.23}$$

where its inverse is denoted as $\mathbf{Z}$. It can lead to the same predictive distribution $\mathbf{Y}$ mentioned by equation 2.16 with the limit of uninformative prior.
Consider $\mathbf{K_y} = \mathbf{K_f} + \Sigma_{noise}$, there is a problem to evaluate $\mathbf{K_y^{-1}}$ because the line search always starts from the location $x_1 = 0$ and $\rho^2 = 0$ causing zero walls due to $\min(\mathbf{x}, \mathbf{x}')$ in the covariance function, and thus zero determinant. The expressions of $\mathbf{A^{-1}}$ and $\mathbf{Z}$ cannot be evaluated properly as the result.

$$\mathbf{K_y} = \begin{bmatrix} \sigma^2 & \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0}' & > \mathbf{0} & \mathbf{0}' & > \mathbf{0} \\ 0 & \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0}' & > \mathbf{0} & \mathbf{0}' & > \mathbf{0} \end{bmatrix} \tag{2.24}$$

where $\mathbf{0} = \mathbf{0}_{1\times T-1}$ is the zero matrix and $> \mathbf{0} = > \mathbf{0}_{T-1\times T-1}$ is the non zero matrix.
To solve these numerical problems, we have rearranged the positions of zero elements in the matrices, and provided the new forms of $\mathbf{K_y^{-1}}, \mathbf{A}, \mathbf{A^{-1}}, \mathbf{Z}, \mathbf{Y}$ and $\mathbf{H}$. The details are mentioned in Appendix C.
The negative log marginal likelihood (nlml) is rewritten as (derived from Appendix D),

$$\begin{aligned} -\log p(\mathbf{Y}|\mathbf{X}) &= \frac{1}{2}\mathbf{Y}^\top\mathbf{K}^{-1}\mathbf{Y} + \frac{1}{2}\log|\mathbf{K}| + \frac{n}{2}\log 2\pi \\ &= \frac{1}{2}\mathbf{Y}^\top\mathbf{ZY} + \frac{1}{2}(\log|\mathbf{K_y}| + \log|\mathbf{B}| + \log|\mathbf{A}|) + \frac{n}{2}\log 2\pi \\ &\approx \frac{1}{2}\mathbf{Y}^\top\mathbf{ZY} + \frac{n}{2}\log 2\pi + \frac{1}{2}|\mathbf{K_{22}}| \\ &+ \frac{1}{2}\begin{cases} \text{if cubic spline,} \\ \quad \log[(\sigma^2 a + 1)(\rho^2 c + 1) - \rho^2\sigma^2 b^2] \\ \text{if quintic spline,} \\ \quad \log[(\sigma^2 a + 1)((\rho^2 c + 1)f - \rho^2 e^2) \\ \quad +\sigma^2\rho^2 b(2ed - bf) - \sigma^2 d^2(\rho^2 c + 1)] \end{cases} \end{aligned} \tag{2.25}$$

where a, b, c, d, e, f are the coefficients mentioned in Appendix D.
To fit the Gaussian Process model needs the best settings of hyperparameters which should minimize the negative log marginal likelihood.
From equations 2.18, 2.19 and 2.25, there are three hyperparameters $(\theta^2, \sigma^2, \rho^2)$ required to

be optimized. In this paper, we assume that the observations from the projected gradients are noise-free, i.e. $\rho^2 \to 0$ because the objective is more susceptible to noise than the projected gradient in the real-life problem, and this can further simplify the optimization problem as a problem of τ^2 by defining the function value noise to signal variance ratio as $\tau^2 = \frac{\sigma^2}{\theta^2}$ described in the details of Appendix E.
As seen from Appendix E, to estimate the hyperparameters, an analytic solution for θ^2 is derived as a function of τ^2. And τ^2 is optimized by Newton Optimization.
The closed form solution for θ^2 is given by,

$$\hat{\theta}^2 = \frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathbf{y}\bar{\mathbf{K}}_\mathbf{y})}\mathbf{Y}^\top\bar{\mathbf{Z}}_\mathbf{y}\mathbf{Y} \tag{2.26}$$

where

$$\begin{aligned}
&\mathrm{Tr}(\bar{\mathbf{Z}}_\mathbf{y}\bar{\mathbf{K}}_\mathbf{y}) = \begin{cases} 2n-2 & \text{if cubic spline} \\ 2n-3 & \text{if quintic spline} \end{cases} \\
&\bar{\mathbf{Z}}_\mathbf{y} = \bar{\mathbf{K}}_\mathbf{y}^{-1} - \bar{\mathbf{K}}_\mathbf{y}^{-1}\mathbf{H}^\top(\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-1}\mathbf{H}^\top)^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-1} \\
&\bar{\mathbf{K}}_\mathbf{y} = \mathbf{K}_\mathbf{f} + \tau^2\mathbf{F} \\
&\mathbf{F} = \Sigma_{noise}, \qquad \text{when } \rho^2 \to 0
\end{aligned}$$

As the hyperparameters must be positive, they are mapped into the log-scale before applying Newton Optimization over τ^2.

$$\tau^2 = \exp 2\bar{\tau} \tag{2.27}$$

$$\theta^2 = \exp 2\bar{\theta} \tag{2.28}$$

The Newton Optimization of τ^2 is given by the following equation,

$$\bar{\tau}[t+1] = \bar{\tau}[t] - H^{-1}g \tag{2.29}$$

where H and g are the Hessian and gradient respectively. The negative sign can be replaced by the positive sign as the backup optimization process.
The gradient is given by,

$$-\frac{\partial}{\partial\bar{\tau}}\log p(\mathbf{Y}|\mathbf{X}) = \frac{1}{2}\mathrm{Tr}\left((\mathbf{Z} - \mathbf{z}\mathbf{z}')\frac{\partial\mathbf{K}}{\partial\bar{\tau}}\right) \tag{2.30}$$

where $\mathbf{Z} = \mathbf{K}^{-1}$, $\mathbf{z} = \mathbf{Z}\mathbf{Y}$ and $\mathbf{B} = \mathbf{Z} - \mathbf{z}\mathbf{z}'$.
The Hessian is given by,

$$\begin{aligned}
-\frac{\partial^2}{\partial\theta^2}\log p(\mathbf{Y}|\mathbf{X}) &= \frac{1}{2}\frac{\partial}{\partial\theta}\mathrm{Tr}\left(\mathbf{B}\frac{\partial\mathbf{K}}{\partial\theta}\right) \\
&= \frac{1}{2}\mathrm{Tr}\left(\frac{\partial\mathbf{K}}{\partial\bar{\tau}}\mathbf{Z}\frac{\partial\mathbf{K}}{\partial\bar{\tau}}(2\mathbf{z}\mathbf{z}^\top - \mathbf{Z})\right) + \frac{1}{2}\frac{\partial^2\hat{\theta}^2}{\partial\bar{\tau}^2}\mathrm{Tr}(\mathbf{B}\bar{\mathbf{K}}_\mathbf{y}) \\
&\quad + 2\tau^2\frac{\partial\hat{\theta}^2}{\partial\bar{\tau}}\mathrm{Tr}(\mathbf{BF}) + 2\tau^2\hat{\theta}^2\mathrm{Tr}(\mathbf{BF})
\end{aligned} \tag{2.31}$$

where $\frac{\partial\bar{K}_y}{\partial\bar{\tau}} = 2\tau^2\mathbf{F}$, $\frac{\partial^2\bar{K}_y}{\partial\bar{\tau}^2} = 4\tau^2\mathbf{F}$,

$$\begin{aligned}
\frac{\partial\mathbf{K}}{\partial\bar{\tau}} &= \hat{\theta}^2\frac{\partial\bar{K}_y}{\partial\bar{\tau}} + \frac{\partial\hat{\theta}^2}{\partial\bar{\tau}}\bar{\mathbf{K}}_y \\
\frac{\partial\hat{\theta}^2}{\partial\bar{\tau}} &= -\frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathbf{y}\bar{\mathbf{K}}_\mathbf{y})}\mathbf{Y}^\top\bar{\mathbf{Z}}_\mathbf{y}\frac{\partial\bar{\mathbf{K}}_\mathbf{y}}{\partial\bar{\tau}}\bar{\mathbf{Z}}_\mathbf{y}\mathbf{Y} \\
\frac{\partial^2\hat{\theta}^2}{\partial\bar{\tau}^2} &= \frac{2}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathbf{y}\bar{\mathbf{K}}_\mathbf{y})}\mathbf{Y}^\top\bar{\mathbf{Z}}_\mathbf{y}\frac{\partial\bar{\mathbf{K}}_\mathbf{y}}{\partial\bar{\tau}}\bar{\mathbf{Z}}_\mathbf{y}\frac{\partial\bar{\mathbf{K}}_\mathbf{y}}{\partial\bar{\tau}}\bar{\mathbf{Z}}_\mathbf{y}\mathbf{Y} \\
&\quad - \frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathbf{y}\bar{\mathbf{K}}_\mathbf{y})}\mathbf{Y}^\top\bar{\mathbf{Z}}_\mathbf{y}\frac{\partial\bar{\mathbf{K}}_\mathbf{y}}{\partial\bar{\tau}}\bar{\mathbf{Z}}_\mathbf{y}\mathbf{Y}
\end{aligned}$$

2.6.2 Sampling new point by Expected Improvement

From the previous part, the predictive distribution over the search region is computed.
To improve the prediction, the next evaluation point within the region of interest is required so that the algorithm can have a better understanding about the landscape of the search region. It is not a random sampling process like the one in line search but is governed by the Bayesian Optimization.
Consider the prior and the data have induced a posterior over functions (i.e. the predictive distribution), to avoid evaluating the expensive objective repeatedly and randomly, the concept of the acquisition function ($u(x)$, or sometimes called the utility) acting as a replacement of the objective is introduced. To select the new potential point requires finding the entry which maximizes the utility as follows,

$$x_{j+1} = \arg\max_x u(x) \tag{2.32}$$

The utility u is defined to take the mean and the covariance of the predictive distribution into account, and has to make a decision that can balance the trade-off between exploration

and exploitation so that the algorithm can either investigate the uncertain region or find the optimized point (in our case, the minimum point).
There are several choices for the acquisition function. One is the probability of improvement [16] which is given by,

$$u_{PI}(x) = \frac{1}{2}\left(1 + \operatorname{erf}\frac{\eta - \mu}{\sqrt{2\mathbb{V}(x)}}\right) \tag{2.33}$$

where $\eta = \min_{i=1,\dots,T}\{\mu(x_i)\}$, μ is the predictive mean and $\mathbb{V} = \sigma^2_{\mathbf{X}*|\mathbf{Y},\mathbf{X}}$ is the predictive covariance.
Another option is the expected improvement [2], the proper formulation is stated as follows, The utility [10] is given by,

$$\begin{aligned} u_{EI}(x) &= \mathbb{E}_{p(f_t|y,y')}[\min\{0, \eta - f(t)\}] \\ &= \frac{\eta - \mu(x)}{2}\left(1 + \operatorname{erf}\frac{\eta - \mu(x)}{\sqrt{2\mathbb{V}(x)}}\right) \\ &+ \sqrt{\frac{\mathbb{V}(x)}{2\pi}}\exp\left(-\frac{(\eta - \mu(x))^2}{2\mathbb{V}(x)}\right) \end{aligned} \tag{2.34}$$

The complete derivation is shown by [5]. Many research studies [7][10] have shown and suggested that the expected improvement is the better option than the probability of improvement, so the project proceeds with the expected improvement as the acquisition function (utility). Below is the example of using the expected improvement to improve the prediction.

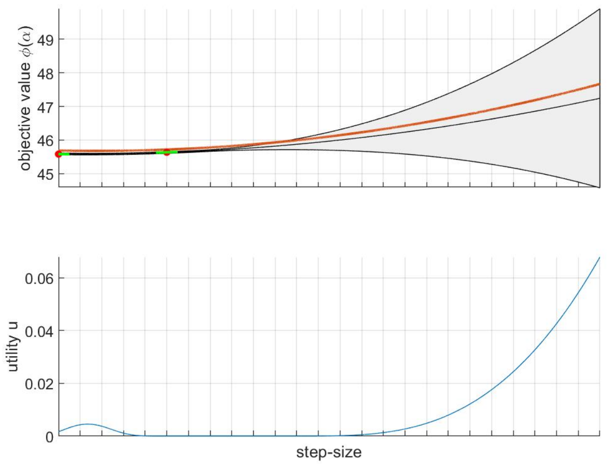

Figure 2.4 Examples of expected improvement [2] predicted by 2 points.

From Figure 2.4, given two points (with the red dots as the objectives and the green bars as the projected gradients), the prediction (the red curve is the true mean and the black one is the predictive mean, the grey region is the uncertainty) is made as shown in the top subplot. It gives the corresponding utility which peaks at the region far away from the existing points as seen from the bottom subplot. Investing the new point in that region can result in an extrapolation step which is similar to that in the original line search algorithm from Appendix A.2.

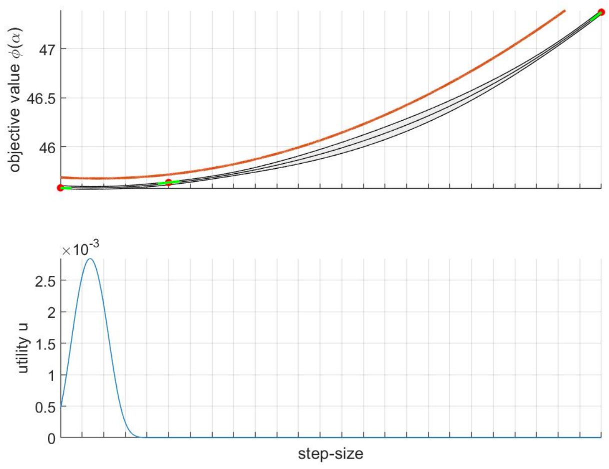

Figure 2.5 Examples of expected improvement predicted by 3 points.

Immediately after investing the new point, the new prediction and its utility are made as shown in Figure 2.5. Both the profiles of the prediction and the utility are changed. With the smaller predictive error (the grey area) within the search region, the algorithm can now look for the entry with the lowest predictive mean to find the local minimum.

Therefore, with the predictive distribution, the new evaluation point with the highest utility can be computed, sampled and added into the training data, to improve the prediction in the next iteration as seen from Figures 2.4 and 2.5. The decision made by the expected improvement can provide both interpolation and extrapolation steps.

Chapter 3

Algorithm Design

3.1 Related Works

There is a discussion about how to terminate the probabilistic line search algorithm. In the design of line search algorithm, it terminates if the acceptable step-size satisfying the strong Wolfe Condition is found. But this comes with the problem of early stopping due to the noisy observations.
The research study [10] which models the observations as the stochastic process suggests that the Wolfe Conditions can be expressed in a probabilistic way, which triggers termination if the Wolfe probability is greater than a threshold. Their implementation does have a promising performance.
On other hand, there are some research studies [5][17][18][19][20] suggesting that the expected improvement may have a convergence property but most of them only work with certain assumptions, such as noise-free observations or kernels (covariance functions) in Hilbert space. Despite the popular use of the expected improvement, only a few research studies have considered the stopping criteria for it. There is one research [5] suggesting a possible termination criteria. Based on the observations that the utility value decreases with the number of evaluation points and iterations, they terminate the expected improvement when the new evaluated point has the utility lower than the threshold 0.5. The settings of their algorithm perform better than the benchmark settings both in computation complexity and accuracy. Therefore, It is worthwhile to investigate the stopping criteria of the expected improvement.
After some tests on the decisions made by the expected improvement, it is observed that it always begin with an extrapolation step to the region far away from where the existing entries locate because of the high uncertainty, and performs multiple interpolation steps within the search range. And most of the interpolation steps fall into the region nearby the

local minimum point, which is regarded as the potential interval. If it is assumed that the bounds of this potential interval can converge to a termination point after a finite number of iterations, it is possible to replace the properties of the Wolfe Conditions with those of the expected improvement. It is because the expected improvement built upon the stochastic model has already taken the observations (function values for sufficient decrease condition and projected gradients for curvature condition) into account to determine where to invest the new point, for example, the extrapolation due to high uncertainty can replace that due to violation of the Wolfe Conditions and the interpolation due to low predictive mean can ensure the iterative improvement guaranteed by the Wolfe Conditions. The only difference is that, the line search uses the Wolfe Condition to localize the interval and select a termination point from it, meanwhile, my algorithm uses the expected improvement to narrow down the potential interval until it converges to a termination point.
This assumption of replacing the Wolfe Conditions with the expected improvement can be appealing because the early stopping due to the satisfaction of Wolfe Conditions, the additional calculation of the probabilistic Wolfe Conditions, which might potentially reduce the processing time, and the parameter settings c_1, c_2 can be avoided.

3.2 Outline

The design for the probabilistic line search is proposed here. It is assumed that the expected improvement can lead to the convergence of the potential interval $[x_{min}, x_{max}] = [\kappa, \kappa + \Delta]$ where the ideal step-size x'_e is located, so that the bounds are assumed to converge to a termination point which approximates the ideal step-size. κ is denoted as the lower bound of the potential interval and Δ is the width of the potential interval.
So the termination criteria of the probabilistic line search is not any extent of the strong Wolfe Conditions or their variations, but instead the width of the potential interval Δ when it is lower than an argument tolerance 1×10^{-3}. This termination criteria resembles one of the stopping criteria which terminates BFGS when the change of the two consecutive starting vectors is too small $|\mathbf{s_{e+1}} - \mathbf{s_e}| < 1 \times 10^{-8}$. Undoubtedly, this stopping criteria is weaker than that from BFGS because $10^{-3} >> 10^{-8}$ but more restrictive than the Wolfe Conditions because the bounds have to converge to a termination point.
It is interesting to investigate how optimization algorithm can operate without Wolfe Conditions but with Bayesian Optimization.
The stopping criteria is given by,

$$x_{max} - x_{min} = \Delta \leq 10^{-3} \tag{3.1}$$

Let's consider, BFGS begins with the inputs of function f and the starting vector $\mathbf{s_1}$. A while loop with a counter variable e, named epoch, is initialized to execute gradient descent and line searches repetitively until it is terminated by the stopping criteria.
For each epoch e (starting from 1), BFGS takes the starting point $\mathbf{s_{e(=1)}}$ (which is a step-size of 0, i.e. $x^e_{n=1} = 0$), computes the search direction $\mathbf{t_e}$ and outputs the acceptable step-size $x^e_{n=2} = x_e$. n denotes the indexing of initial data (i.e. the total number of initial entries is $N = 2$ so n is either 1 or 2).
Our algorithm then inherits those 2 starting entries (x^e_1 and x^e_2), the starting vector $\mathbf{s_e}$ and search direction $\mathbf{t_e}$, and initializes another while loop with a counter variable i. It is assumed that the 2 starting entries are the bounds of the potential interval in each iteration i, i.e. $[x^e_1, x^e_2] = [x^{ei}_1, x^{ei}_2] = [x_{min}, x_{max}]$. One may argue this assumption may restrict exploration. But since those two starting entries are renewed in each iteration i by a selection scheme, which is described later, such that they can be different from the previous entries and not within the previous potential interval, the algorithm is able to extrapolate as well, i.e. $x^{e,i+1}_{1:2} \notin [x^{e,i}_1, x^{e,i}_2]$ and $|x^{e,i+1}_1 - x^{e,i+1}_2| > |x^{e,i}_1 - x^{e,i}_2|$.
Within this while loop i, there is another while loop with a counter variable j which will reset to 0 for each iteration of i. Based on the Bayesian Optimization, each iteration j of the while loop can sample a new evaluation point x^{ei}_{N+j} from the search range $[\kappa, \kappa + \delta\Delta]$ (δ is the range factor) and its observations $y^{ei}_{N+j}, y'^{ei}_{N+j}$. This loop terminates if the new evaluated point is too close to one existing point, which may indicate the region where those two close points are located is the new potential interval.
Thus, in each new iteration $i+1$, the bounds of the potential interval $[\kappa, \kappa + \Delta]$ are re-estimated. The process repeats until the width of the potential interval satisfies the condition from 3.1. At the end of the probabilistic line search algorithm, it should be able to output the better predicted value of the step-size $x'_e = \frac{1}{2}(2\kappa + \Delta)$.
The new optimal step-size should ensure that BFGS can start with a better starting vector $\mathbf{s_{e+1}} = \mathbf{s_e} + x'_e \mathbf{t_e}$ for the next epoch $e+1$ to improve its performance.
The pseudocode (Algorithm 1) is the outline of my algorithm. The rest of the chapter will focus on explaining the rationales behind each important step of the algorithm.

Algorithm 1 Probabilistic Line Search

```
1:  Input: $\mathbf{s_e}, \mathbf{t_e}, x_1^e, x_2^e, f, \nabla f$.
2:  Default Settings: Spline = Quintic, rescale = on, $\delta = 5$ and $N = 4$
3:  Initialize $\Delta = 10, i = 0, \kappa = 0$, $\mathbf{s_e}, x_1^e, x_2^e \rightarrow \mathbf{s_{e1}}, x_1^{e1}, x_2^{e1}$
4:  **while** $\Delta > 0.001$ **do** $i = i + 1, j = 0$
5:      **if** $i > 1$ **then** Inherit **stepsamples:** $\mathbf{T},\mathbf{B},\mathbf{C} \leftarrow \mathbf{s_{ei}},\mathbf{t_{ei}},x_{1:N}^{ei},y_{1:N}^{ei},y_{1:N}^{\prime ei}$
6:      **end if**
7:      Store location data. **orgls**, **stepls** $\leftarrow x_{1:N}^{ei}$
8:      **while** true **do** $j = j + 1$
9:          **if** $i = 1$ and $j = 1$ **then** Initialize **stepsamples:** $\mathbf{T},\mathbf{B},\mathbf{C} \leftarrow \mathbf{s_{ei}},\mathbf{t_{ei}},x_{1:2}^{ei},f,\nabla f$
10:         **end if**
11:         **if** j=1 **then** $a = x_2^{ei}$
12:         **else** Add new **stepsamples:** $\mathbf{T},\mathbf{B} \leftarrow \mathbf{s_{ei}},\mathbf{t_{ei}},\mathbf{T},x_{N+j}^{ei},f,\nabla f$
13:         **end if**
14:         **rescale:** $\mathbf{T}',\mathbf{B}',\mathbf{C}',\mathbf{orgls}',\mathbf{stepls}',scale \leftarrow \mathbf{T},\mathbf{B},\mathbf{C},\mathbf{orgls},\mathbf{stepls}$
15:         **predict:** $\mu, \mathbf{cov} \leftarrow \mathbf{T}', \mathbf{C}'$
16:         $\hat{\alpha} \xleftarrow{\max} \mathbf{u} \leftarrow \mathbf{EI}(\mu, \mathbf{cov})$
17:         **selection:** $\mathbf{stepls}' \leftarrow \alpha \leftarrow \mathbf{T}',\mathbf{C}',\mathbf{u},a,scale,\delta$
18:         **if** $\min|\mathbf{orgls}' - \hat{\alpha}| < \frac{a \times scale}{\delta}$ **then**
19:             Cluster two close data $x_{1:2}^{\prime e,i+1} \leftarrow$**stepls'** and sort them in ascending order
20:             **unscale:** $\mathbf{T},\mathbf{C},x_{1:2}^{e,i+1} \leftarrow \mathbf{T}',\mathbf{C}',scale,x_{1:2}^{\prime e,i+1}$
21:             Update the potential interval $\kappa = \kappa + x_1^{e,i+1}, \mathbf{s_{e,i+1}} = \mathbf{s_{e,i}} + x_1^{e,i+1}\mathbf{t_e}$, $\Delta = x_2^{e,i+1} - x_1^{e,i+1}$
22:             Remain $N$ more data $\mathbf{T} \rightarrow x_{3:N+2}^{e,i+1}$
23:             **shift:** $x_{1:N+2}^{e,i+1} = x_{1:N+2}^{e,i+1} - x_1^{e,i+1}$
24:             **return** $x_{1:N}^{ei},y_{1:N}^{ei},y_{1:N}^{\prime ei}$(dummy:$N+2 \rightarrow N$) and **break**
25:         **end if**
26:         $\mathbf{orgls}',x_{N+j}^{\prime ei} \leftarrow \hat{\alpha}$
27:         **unscale:** $\mathbf{T},\mathbf{B},\mathbf{C},\mathbf{orgls},\mathbf{stepls},x_{N+j}^{ei} \leftarrow \mathbf{T}',\mathbf{B}',\mathbf{C}',\mathbf{orgls}',\mathbf{stepls}',scale,x_{N+j}^{\prime ei}$
28:     **end while**
29: **end while**
30: Output: $x_e' = \frac{1}{2}(2\kappa + \Delta)$.
```

There are some parameters that can affect the performance of the probabilistic line search algorithm, including δ and N. δ is the search factor which determines the maximum search range for each i-th iteration (i.e. $\delta \times x_2^{ei}$) and the minimum separation tolerance between two location datapoints (i.e. x_2^{ei}/δ) which trigger termination of the j loop. N is the maximum

number of data (except the trigger datapoints) can be remained from the previous iteration *i*. This parameter is also used to alter the retain range defined in the later section, which allows us to check which region most of the retain data comes from. Different rescale modes and various kind of splines should alter the performance of the algorithm.
The following sections are the notes-worthy details in the algorithm.

3.3 Step-samples generation

For each *j*-th iteration, a new evaluation point x^{ei}_{N+j} is added. The noisy observations, including the objective value y^{ei}_j and the projected gradient value y'^{ei}_j, are necessary for the prediction. Assume that the noise is Gaussian-distributed, they are given by,

$$y^{ei}_{N+j} = f(\mathbf{s_{ei}} + x^{ei}_{N+j}\mathbf{t_e}) + \varepsilon^{ei}_{N+j} = \phi(x^{ei}_{N+j}) + \varepsilon^{ei}_{N+j} \tag{3.2}$$

$$y'^{ei}_{N+j} = \mathbf{t_e} \cdot \nabla f(\mathbf{s_{ei}} + x^{ei}_{N+j}\mathbf{t_e}) + \nu^{ei}_{N+j} = \phi'(x^{ei}_{N+j}) + \nu^{ei}_{N+j} \tag{3.3}$$

, where $\varepsilon^{ei}_{N+j} \sim \sigma^2\mathcal{N}(\mathbf{0},\mathbf{I})$ and $\nu^{ei}_{N+j} \sim \rho^2\mathcal{N}(\mathbf{0},\mathbf{I})$.
The collection of all $x^{ei}_{N+j}, y^{ei}_{N+j}, y'^{ei}_{N+j}$ is named as the train data $\mathbf{T}$.
The collection of the curve entries and their noise-free observations is sampled directly from the function and named as the curve data $\mathbf{C}$. Its location x^{ei}_{*j} is ranged from $x^{ei}_1 = 0$ to δx^{ei}_2 (or $\delta\Delta$). The curve data is important because its location entries can be used to form $\mathbf{K_f}(\mathbf{x},\mathbf{x}_*), \mathbf{K_f}(\mathbf{x}_*,\mathbf{x}_*), \mathbf{H}_*$ and its observations are for plotting purpose. The slope bar data $\mathbf{B}$ is also generated to show the gradient information of the train data in the diagram as follows, set $\vartheta = 0.02$ (which controls how wide the slope bar is),

$$\begin{aligned}
y^{ei}_{N+j} &= \phi'^{ei}_{N+j} x^{ei}_{N+j} + \xi^{ei}_{N+j} \\
\xi^{ei}_{N+j} &= y^{ei}_{N+j} - \phi'^{ei}_{N+j} x^{ei}_{N+j} \\
(x^{ei}_{N+j,low}, x^{ei}_{N+j,high}) &= x^{ei}_{N+j} + (-\vartheta, \vartheta) \\
(y^{ei}_{N+j,low}, y^{ei}_{N+j,high}) &= \xi^{ei}_{N+j} + \phi'^{ei}_{N+j}(x^{ei}_{N+j,low}, x^{ei}_{N+j,high})
\end{aligned}$$

where ξ is the intercept of the linear equation. This part corresponds to lines 5, 9 and 12 of Algorithm 1.

3.4 Scaling factor

Since the stepsize sample always is greater than 0, if it is in the range of $(0,1)$, the elements of covariance matrix become very small due to the 3rd power of cubic spline and 5th power

of quintic spline, causing badly-scaled matrix.
To solve the following problem, every entry is re-scaled by the 2^{nd} smallest location since the smallest location is always 0. As the location is scaled, the slope (projected gradient) also has to be re-scaled as follows,

$$\begin{aligned} x^{ei}_{N+j} &\leftarrow \frac{\gamma}{x^{ei}_2} x^{ei}_{N+j} \\ y'^{ei}_{N+j} &\leftarrow \frac{x^{ei}_2}{\gamma} y'^{ei}_{N+j} \\ \text{, where} \quad \gamma &= \begin{cases} \sqrt{3} & \text{if cubic spline} \\ \sqrt{\frac{20}{3}} & \text{if quintic spline} \end{cases} \end{aligned} \tag{3.4}$$

Therefore, from Figure 3.1, the location entries and their projected gradients are rescaled.

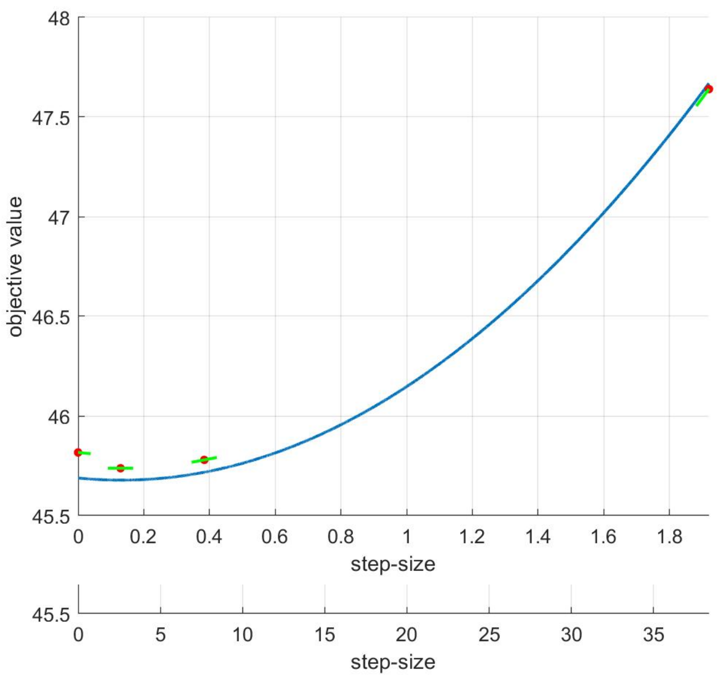

Figure 3.1 The example of rescaling the entries and their projected gradients.

The γ factor is obtained from the squared-root-ratio of the smallest coefficients between k and $^{\partial}k^{\partial}$ from equations 2.18 and 2.19. It is used to eliminate the coefficients in the covariance functions so the location of the 2^{nd} smallest entry can be rescaled to the value greater than 1. The scale $\frac{\gamma}{x^{ei}_2}$ can be re-used for un-scaling. It is important to carefully apply re-scaling and un-scaling since mixing up rescaled or unscaled data can cause poor prediction.
This part corresponds to lines 14, 20 and 27 of Algorithm 1.

3.5 Prediction

Before computing the predictive distribution, the settings of hyperparameters (i.e. θ^2, τ^2) are required.
Consider, the algorithm is initialized by only 2 location data (i.e. x_1^{e1},x_2^{e1}) with 4 known observations (i.e. $y_1^{e1}, y_2^{e1}, y_1^{\prime e1}, y_2^{\prime e1}$), they are not enough to solve the unknowns of the quintic spline model but enough for those of the cubic spline because of 2 extra unknowns (coefficients) from linear basis function of cubic spline and 3 extra unknowns from quadratic function of quintic spline.
Therefore, Newton Optimization (equation 2.29) is only applied to solve τ^2 when there are 3 location data (6 knowns) for quintic spline and 2 for cubic spline. To compute the preliminary settings, τ^2 is set to 1 so that the value of θ^2 and nlml can be computed from equations 2.26 and 2.25.
We set the default setting of the function noise to signal variance ratio as $\tau^2 = 1$ implying the prior assumptions as it is not possible to estimate hyperparameters uniquely.
Once there are enough observations to infer the unknowns, a while loop is set up to improve τ^2 by Newton Optimization, and compute its corresponding θ^2 and nlml. The loop terminates until a particular setting of τ^2 results in a lower nlml value or the counter of the loop exceeds the threshold. If there is no better solution found after the loop, this might be due to the bad estimate estimate of the Hessian sign so another loop is set up to improve it by Newton Optimiation with the positive sign as a backup plan. But the reasons leading to bad Hessian has to be investigated in the future work.

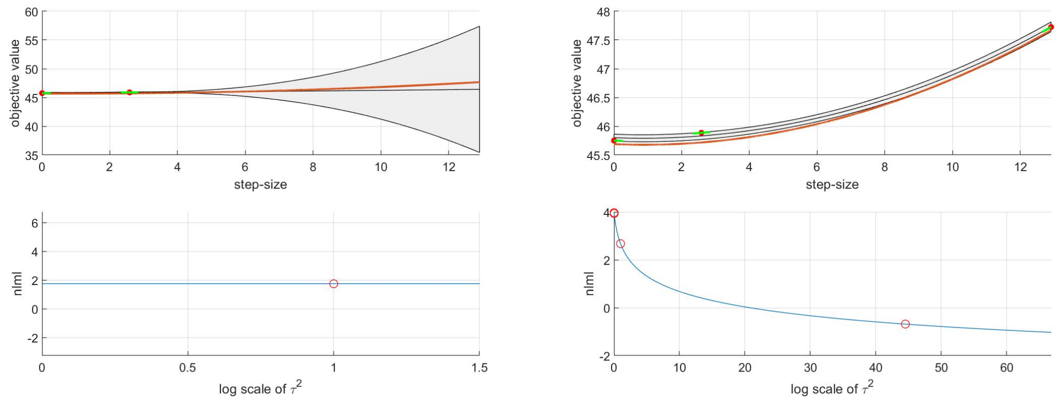

Figure 3.2 How to initialize τ^2 and optimize it for the quintic spline model.

From Figure 3.2, the left bottom subplot shows that the nlml profile is flat (i.e. ~ 2) when there are only 2 entries subjected to the prediction from the left top subplot, as not enough information is available to solve the unknowns of the quintic spline. Once a new evaluated

point is added at the right top subplot, the nlml profile from the right bottom subplot is no longer flat, so that a new setting of τ^2 can be optimized. But the Newton Optimization does not work properly with the negative sign, then the backup plan is activated with positive sign to solve the problem.
With the settings of hyperparameters, the predictive distribution can be computed from equation 2.16 to fit the model with the data. This part refers to line 15 of Algorithm 1.

3.6 Selection, Shift, Zoom and Retain

As the predictive distribution is given, the utility of each location can be computed (i.e. line 16 of Algorithm 1). The algorithm usually selects the location with the highest utility as the new evaluation point (lines 16 and 26 of Algorithm 1) to improve the prediction for the next iteration $j+1$ as seen from Figure 3.3.

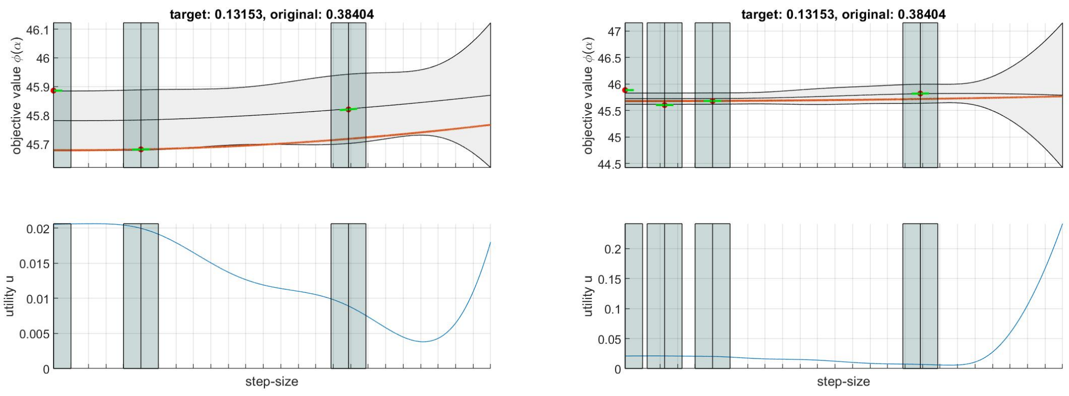

Figure 3.3 The new evaluation point is usually the location with the highest utility.

Figure 3.3 shows the predictive distributions at the top subplots and the utilities of their corresponding distributions above at the bottom subplots. The blue shaded region denotes the trigger regions $[-\frac{x_2^{ei}}{\delta}, +\frac{x_2^{ei}}{\delta}]$ where the new evaluated points should not locate, or otherwise, it triggers a set of shifting, zooming and retaining actions which are discussed later in this section.
The left bottom subplot shows that the utility peaks at the left hand side of the plot, so the algorithm invests a new point right there leading to the better predictive distribution with thinner uncertainty at the left hand side of the right top subplot.

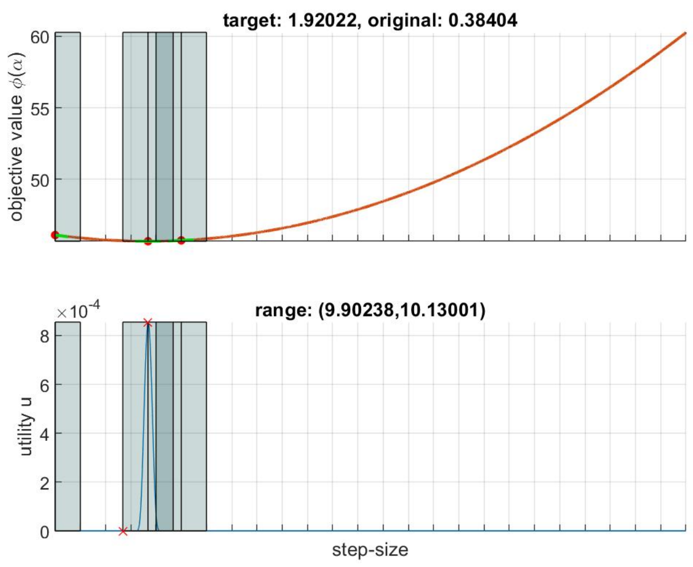

Figure 3.4 The example of zoom.

If the new evaluated point is close to one of the existing point (i.e. within the blue shaded region) by the following unscaled condition (refers to line 18 of Algorithm 1),

$$|x^{ei}_{-(N+j+1)} - x^{ei}_{N+j+1}| < \frac{x^{ei}_2}{\delta}$$

where x^{ei}_{N+j+1} is the new evaluated point and $x^{ei}_{-(N+j+1)}$ is the existing evaluated points except the new one for iteration $j+1$, adding this new location entry may cause bad condition number for the matrix in a long run. But the theory of expected improvement implies that the region around this new evaluated point has a potential to explore.

Therefore, when a new evaluation point x^{ei}_{N+j+1} is close to the existing point x^{ei}_{N+j}, the algorithm is designed to automatically terminate the while loop j, collect x^{ei}_{N+j} as the first entry of the starting subset $(x_1^{e,i+1}, x_2^{e,i+1})$, re-sample another point which has the highest utility but is located on the margin of or outside the blue shaded region as the second entry and then collect N previous points within the new retain range (refers to lines 17, 19 and 22 of Algorithm 1). The reason of not choosing x^{ei}_{N+j+1} and x^{ei}_{N+j} directly as the new starting subset is that, if they are equivalent, the algorithm cannot make the prediction based on one datum, or if they are very close, the algorithm may converge too quickly since $\Delta \to 0$.

In most of the cases, the utility is continuous across the search range so the second entry of the starting subset is either $x^{ei}_{N+j} + x^{ei}_2/\delta$ or $x^{ei}_{N+j} - x^{ei}_2/\delta$ as shown in Figure 3.4 which illustrates that the first two entries of the new starting subset are those two red crosses in the bottom subplot as the utility peaks within the blue shaded region.

There is another advantage of choosing x^{ei}_{N+j} and another location (which is outside the blue region and has the highest utility) as the new starting subset. If the algorithm has already investigated the search region $[0, \delta x^{ei}_2]$ thoroughly, there are multiple blue shaded regions across the search region. When the algorithm determines to extrapolate, the second entry for the starting subset should be selected from the non-blue region, then it is more likely to be far away from the first entry, resulting in a larger width of search interval $\Delta = |x^{e,i+1}_1 - x^{e,i+1}_2|$ to extrapolate which can possibly lead to a better sub-optimal solution.
Therefore, these two entries are clustered to update the potential interval, the new interval width Δ and lower bound κ.

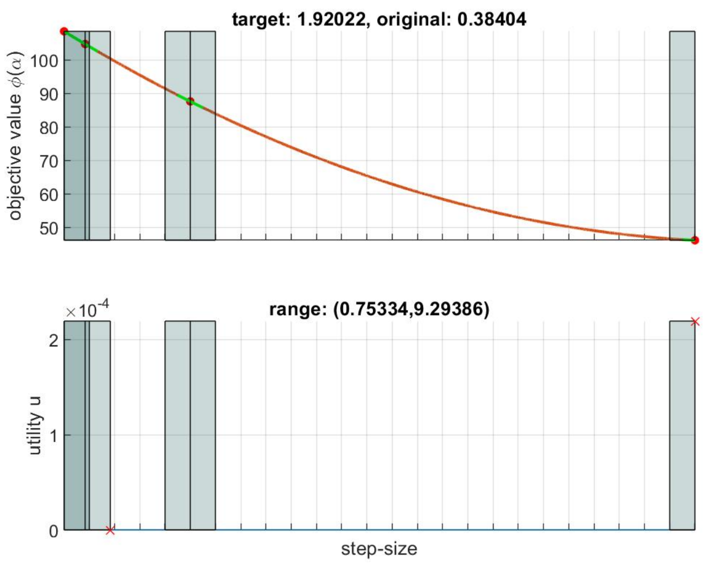

Figure 3.5 The example of the new starting subset leading to extrapolation for the next iteration.

From Figure 3.5, the algorithm invests two points at the right end region. So one point is sampled from the right end. Another is sampled according to the rule that it should be outside the trigger region and have the highest utility. The algorithm then takes the red cross on the left side of the bottom subplot as the second entry. The two starting entries are far away from each other. They are sorted in the ascending order, the smallest entry becomes the new starting point, the gap difference between two entries becomes the new width of the potential interval. The algorithm shifts to this potential interval $[\kappa, \kappa + \Delta]$ for the next search in iteration $i+1$. The whole process corresponds to lines 19-24 of Algorithm 1.

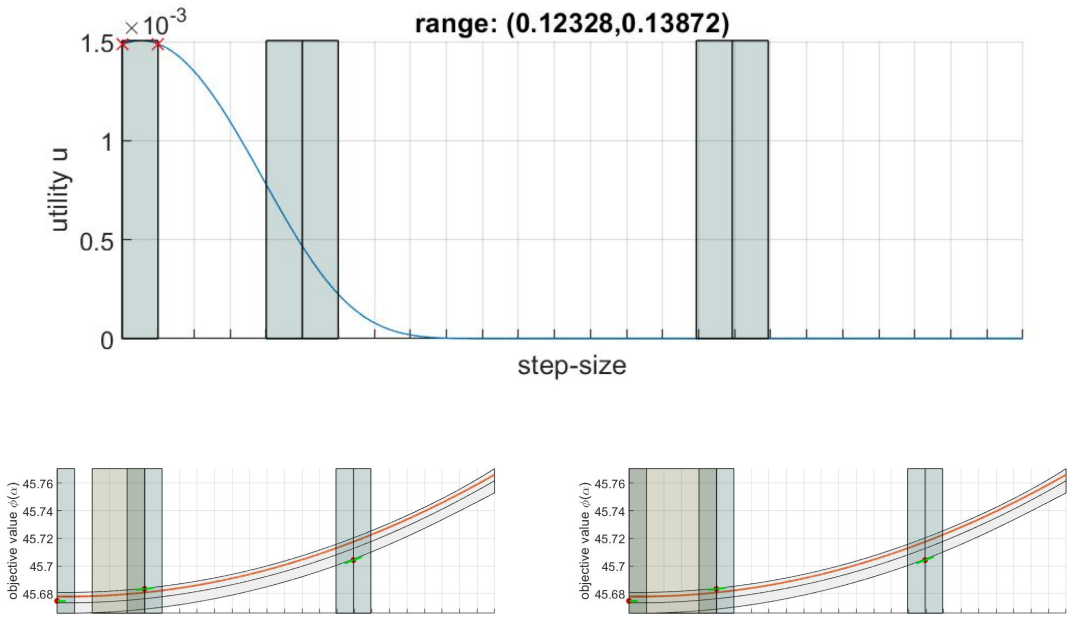

Figure 3.6 The example of retaining data for the next iteration *i*.

To avoid the search process to be very random between consecutive iterations of the while loop *i*, some data must be retained from the previous iterations. *N* datapoints are selected randomly to form part of the new starting subset $x^{e,i+1}_{3:N+2}$ if they satisfy the following conditions,

$$x^{e,i+1}_{3:N+2} \neq x^{e,i+1}_{1:2} \tag{3.5}$$

and

$$x^{e,i+1}_{3:N+2} \in x^{e,i+1}_{1} + (\delta - N, \delta)(x^{e,i+1}_{2} - x^{e,i+1}_{1}) \tag{3.6}$$

where δ is the range factor, N is the maximum number of data remained (except the first two entries which trigger the termination of the while loop j) and $\delta \geq N$.

This retaining region is defined as the region behind the smallest location entry (i.e. $x^{e,i+1}_{1}$) of the starting subset as shown in Figure 3.6 (the green shaded area). The reason of choosing $(\delta - N)\Delta$ as the lower limit is to observe which retaining region behind the smallest location entry is the most informative one for the next iteration. With the lower value of N, the lower bound of the retaining area is shifted to the right.

There is another condition can be tested to replace the condition $x^{e,i+1}_{3:N+2} \in x^{e,i+1}_{1} + (\delta - N, \delta)(x^{e,i+1}_{2} - x^{e,i+1}_{1})$. It is given by,

$$x^{e,i+1}_{3:N+2} \in x^{e,i+1}_{1} + (0, N)(x^{e,i+1}_{2} - x^{e,i+1}_{1}) \tag{3.7}$$

So reversely, the lower limit remains unchanged, but the upper limit is increased with N. This retaining process refers to line 22 of Algorithm 1.
From Figure 3.6, the top subplot shows the utility of the prediction, and the algorithm determines to cluster that two points with red crosses at the left hand side of the top subplot as the first two starting entries. The left bottom subplot shows the region (green shaded area) which retains the data when $N = 3$. The right bottom subplot shows the retain region when $N = 5$. These examples indicate that N affects both the number of data remained and the retain region range. Both of them determine to retain one point in the second blue shaded region. If $N = 0$ or 1, then that point is not retained because it is not within the defined retain range.

3.7 Range of stepsize

Each iteration of the while loop i generates at least two new entries $x_1^{e,i+1}, x_2^{e,i+1}$ ($x_1^{e,i+1} < x_2^{e,i+1}$ and $|x_1^{e,i+1} - x_2^{e,i+1}| < x_2^{ei}/\delta$). Consider the probabilistic line search is initialized by $\mathbf{s_{e1}}, \mathbf{t_e}, x_{1,2}^{e1}$, it should iterate with the updates of the new starting vector $\mathbf{s_{e,i+1}}$, the lower bound κ and the potential interval width Δ,

$$\mathbf{s_{e,i+1}} = \mathbf{s_{e,i}} + x_1^{e,i+1}\mathbf{t_e} \tag{3.8}$$

$$\kappa = \kappa + x_1^{e,i+1} \tag{3.9}$$

$$\Delta = x_2^{e,i+1} - x_1^{e,i+1} \tag{3.10}$$

Therefore, the potential interval of the stepsize can be given by,

$$(x'_{e,min}, x'_{e,max}) = \sum_i (x_1^{e,i}, x_2^{e,i}) = (\kappa, \kappa + \Delta) \tag{3.11}$$

The potential interval width should decrease over iterations, if $i \to \infty$, $\Delta \to 0$.
All the entries of the starting subset $x_{1:2+N}^{e,i+1}$ are also shifted accordingly. This corresponds to lines 21, 23 and 24 of Algorithm 1.
As illustrated in Figure 3.7 below, the area of blue shaded region is decreasing over iterations i until its (x_{min}, x_{max}) width is less than or equal to 0.001. Therefore, the following algorithm should give an approximation of where the optimal step-size is located.

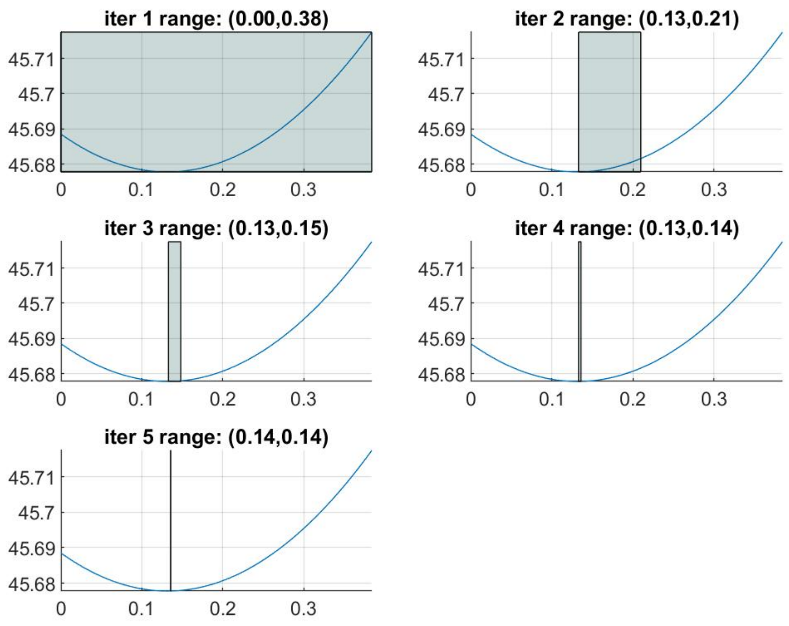

Figure 3.7 Evolution of the potential interval $[\kappa, \kappa+\Delta]$ over iteration i for an one dimensional quadratic problem with $\sigma^2 = 0.01$.

Based on this algorithm design, several detailed experiments were conducted to evaluate its performance in the next Chapter.

Chapter 4

Experiments

Experiments were conducted on the algorithm to assess its abilities to interpolate, extrapolate and terminate in different optimization problems. Since the algorithm was still in the development stage, the following experiments focused on identifying the potential design problems, characteristics, strengths and weaknesses of the algorithm. Thus, the algorithm had not been merged with the line search and BFGS to run in various optimization problems. The default settings of the algorithm were, the derivative noise-to-signal ratio was set to $\rho^2 = 0$, it used the smoothing quintic spline with rescaling, the range factor was set to $\delta = 5$ and the maximum number of retained data was set to $N = 4$.
There were few performance metrics designed for this experiment to assess the performance of the algorithm. One was the average precision measured by variance (var) to compare how deviate the 100 predictions were.

$$\text{var} = \frac{\sum_l^{100}(x'_e - x'_{el})^2}{N} \tag{4.1}$$

, where x'_{el} is the predicted step-size for the l-th loop. The lower the predictions variance is, the higher the precision the predictions have.
Next was the average accuracy measured by the difference between the optimal step-size and the average predicted step-size (avg) over 100 iterations in the loop l. The smaller the difference is, the higher the accuracy of the predictions is.
Other useful metrics were, the average processing time (t) measured how long each loop took, the success rate (SR) within 100 predictions (it counted how many runs were able to output a predicted value and was useful to measure how often the bugs occurred when running the algorithm), and the within rate (WR) within 100 predictions (it counted how many times the predicted step-size was within a certain range, which was particularly useful if there were multiple local minima within the search range).

There were multiple settings can be altered during the experiments, including the spline polynomials (either cubic or quintic), the rescale on-off settings, the range factor δ, the maximum number N of retained entries after each iteration i and the function noise variance of the generated data σ^2, to identify which can give the best performance.
The followings were the optimization problems had been experimented on.

4.1 One Dimensional Quadratic Problem

$$f(\mathbf{x}) = 3x_1^2 + x_2^2 + 55x_3^2 + 2x_4^2 + x_5^2 \tag{4.2}$$

These experiments were to test whether my algorithm can find the optimal solution for a simple convex optimization problem (without multiple local minima and wide flat region).

i-th	1	2	3	4-5
θ^2	7.77×10^{-1}	1.59×10^{-3}	1.00×10^{-6}	0

Table 4.1 Evolution of the signal variance changed over iteration in the search range $[\kappa, \kappa + \delta\Delta]$ from Figure 3.7.

The search was begun with $\mathbf{s_e} = [0.8561, 4.6657, -0.0115, -2.0839, 3.6095]'$, $x_e = 0.3840$, $\mathbf{t_e} = [0.2064, 0.1797, 0.0114, 0.4667, 0.1391]'$, the ideal step-size was 0.1315. As seen from Figure 3.7, the algorithm was able to interpolate to improve the original step-size. The prediction was $x_e' = 0.135$ which was better than the original one $x_e = 0.384$. In this run, the function noise variance was set to 0.01 which became more dominant than the signal variance starting from the second iteration in Table 4.1. But the algorithm was able to converge in the high function noise to signal variance region.
Some experiments were run to examine how different parameters settings may affect the performance.

σ^2	avg	var	SR	t (s)
0.001	0.139	4.81×10^{-5}	92	5.10
0.01	0.152	5.05×10^{-4}	91	5.79
0.1	0.224	7.78×10^{-3}	97	6.43

Table 4.2 Influence of the function noise variance towards the performance. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic.

When the function noise variance was decreased, the average predicted value became closer to 0.1315. The accuracy and the precision got improved. The success rate and the processing time were also increased with the noise variance. This indicated that the algorithm took more time to process the noisy observations, and the lower noise variance may cause more precision error in the calculations so the success rate decreased.

N	avg	var (10^{-4})	SR	t (s)
0	0.153	5.66	97	7.73
1	0.151	4.52	97	6.52
2	0.152	5.65	93	5.95
3	0.157	7.37	94	6.10
4	0.152	5.05	91	5.79
5	0.152	6.62	89	5.93

Table 4.3 Influence of the retained data towards performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0.01,$ rescale=True,spline=quintic.

Table 4.3 showed that the parameter N did not affect the precision and the average prediction at all. But taking no data from the previous iteration took more time to process. A larger value of N can reduce the success rate, maybe because when $x^{ei}_{3:N+2}$ were too close to $x^{ei}_{1:2}$, the algorithm failed to make the prediction. This was quite counter-intuitive because investing points in the new potential interval should give more information.

δ	avg	var (10^{-4})	SR	t (s)
4	0.156	3.84	92	6.95
5	0.157	7.37	94	6.10
6	0.149	4.20	95	7.03

Table 4.4 Influence of the range factor towards performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, N = 3,$ rescale=True,spline=quintic.

The success rate increased with the range factor δ as observed from Table 4.4. But the effect was not significant, and no other trend was observed.

mode	avg	var	SR	t (s)
on	0.152	5.05×10^{-4}	91	5.79
off	0.132	7.38×10^{-13}	100	4.40

Table 4.5 Influence of the rescale mode towards performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, \delta = 5, N = 4,$ spline=quintic.

The algorithm without rescaling mode showed a better general performance than that with rescaling.
But theoretically, the algorithm with rescaling should perform equally or better than that without rescaling. This might indicate that the scaling was implemented incorrectly.

spline	avg	var (10^{-4})	SR	t (s)
quintic	0.152	5.05	91	5.79
cubic	0.146	4.68	95	6.96

Table 4.6 The influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, \delta = 5, N = 4,$ rescale=True.

The quintic spline was shown to perform better in processing time but worse in other performance metrics than the cubic spline, which was out of the expectation because we assume that the quintic spline should perform better than the cubic spline in all metrics due to the theory of the smoothing spline.

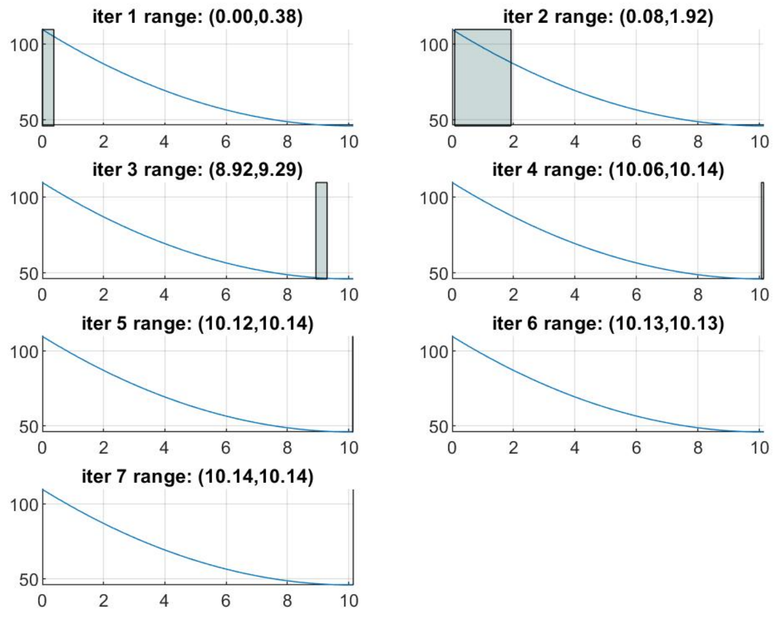

Figure 4.1 Evolution of the potential interval $[\kappa, \kappa + \Delta]$ over iteration i for the one dimensional quadratic problem with $\sigma^2 = 0.01$ and different starting point.

Another experiment was conducted by shifting the starting point further away, i.e. $\mathbf{s_e} = \mathbf{s_e} - 10\mathbf{t_e}$, as shown in Figure 4.1. When the starting point was further away from the ideal step-size, the algorithm was able to extrapolate until it reached the region of interest. Given that the new ideal step-size x_e^* was 10.132 and the initial step was $x_e = 0.384$, the prediction x_e' was 10.14. This showed the algorithm design can allow multiple extrapolations during the search.

i-th	1	2	3	4	5	6-7
θ^2	5.55×10^1	3.56×10^2	5.84×10^{-2}	7.74×10^{-4}	1.00×10^{-6}	0

Table 4.7 Evolution of the signal variance over iteration i in the search range $[\kappa, \kappa + \delta\Delta]$ from Figure 4.1.

From Table 4.7, the increase of signal variance from iteration 1 to 2 showed the potential interval width became wider, leading to a big step shown in the second iteration subplot of Figure 4.1.
Some experiments were run to examine how different parameters settings may affect the performance in this shifted settings.

σ^2	avg	var	SR	t (s)
0.001	10.137	4.38×10^{-5}	92	15.8
0.01	10.152	3.95×10^{-4}	95	13.0
0.1	3.750	9.28×10^{0}	100	7.82

Table 4.8 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic.

For $\sigma^2 = 0.1$, the algorithm was not able to extrapolate back to the region of interest. Similarly to Table 4.2, the smaller value of noise variance made the average prediction closer to the ideal prediction, and increased the accuracy and precision of the prediction. The success rate increased with the function noise variance. But the processing time decreased with the changes of noise variance probably due to premature stopping.

N	avg	var (10^{-4})	SR	t (s)
0	10.153	3.39	95	14.2
1	10.149	3.63	93	13.2
2	10.154	4.45	93	13.2
3	10.152	3.53	95	13.2
4	10.152	3.95	95	13.0
5	10.156	4.54	95	11.8

Table 4.9 Influence of the number of retained data towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0.01,$ rescale=True,spline=quintic.

No obvious trend was seen from Table 4.9. But without retaining data, the algorithm took longer time to process. The reason of the parameter N not affecting the performance is that, the number of data existing in the new search range is always is always 0 or less than N. So, this might indicate that the retain region used to cluster the data was wrong or defined incorrectly.

δ	avg	var (10^{-4})	SR	t (s)
4	10.155	4.01	93	16.9
5	10.152	3.53	95	13.2
6	10.150	4.11	94	11.6

Table 4.10 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, N = 3,$ rescale=True,spline=quintic.

The greater range factor δ resulted in the better average prediction which was closer to the ideal value, and the less processing time. This was reasonable because the wider search range allowed the algorithm to discover the minimum point quicker.

mode	avg	var	SR	t (s)
on	10.152	3.95×10^{-4}	95	13.0
off	9.10	1.19×10^{-8}	59	7.12

Table 4.11 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, \delta = 5, N = 4,$ spline=quintic.

The algorithm without rescaling had a lower success rate and worse prediction than the one with rescaling. The experiments begun to show the problem for the algorithm without rescaling. But the precision and the processing time of the algorithm without was better that those with probably due to premature termination.

spline	avg	var (10^{-4})	SR	t (s)
quintic	10.152	3.95	95	13.0
cubic	10.146	4.70	96	12.0

Table 4.12 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, \delta = 5, N = 4,$ rescale=True.

When compared Table 4.12 with Table 4.6, cubic spline was shown to have better processing time, success rate and accuracy than quinctic spline.

4.2 One Dimensional x^4 Problem

$$f(x) = (x-1)^4 \tag{4.3}$$

This optimization problem was to assess how the algorithm performed in a wide flat region $(0,2)$.

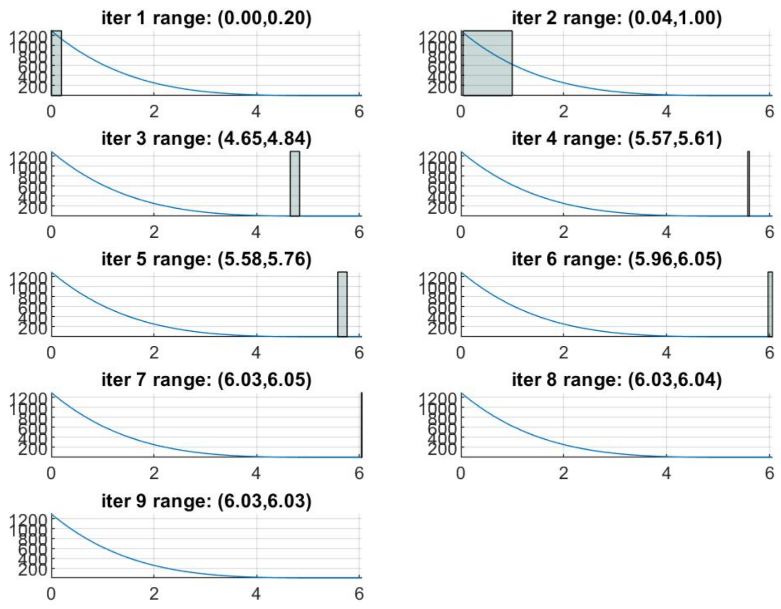

Figure 4.2 Evolution of the potential interval $[\kappa, \kappa + \Delta]$ over iteration i in the one dimensional x^4 problem with $\sigma^2 = 0$.

i	1-2	3	4
θ^2	$>1.00\times10^4$	8.46×10^{-1}	8.80×10^{-5}
i	5	6	7-9
θ^2	2.43×10^{-3}	2.17×10^{-4}	0

Table 4.13 Evolution of the signal variance over iteration in the search range $[\kappa, \kappa + \delta\Delta]$ from Figure 4.2.

The experiment was firstly conducted by initializing the search with a point close to the ideal step-size ($x_e^* = 6$), i.e. $s_e = -5, t_e = 1, x_e = 0.2$. The algorithm was able to move to and zoom into the region of interest to find the acceptable solution. The prediction x_e' from Figure 4.2 was 6.03.
Other experiments to test various parameters settings were also conducted.

σ^2	avg	var	SR	t (s)
0.001	5.80	1.74×10^{-4}	100	11.3
0.1	5.79	9.05×10^{-3}	98	10.6
1	3.86	4.97×10^{0}	100	8.80

Table 4.14 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic.

Table 4.14 showed the greater the noise level was, the further the prediction was away from the ideal step-size and the less precise the prediction was. Low noise variance resulted in longer processing time. These indicated that the extrapolation capability was better and the processing time was longer if the noise level was smaller.

N	avg	var	SR	t (s)
0	5.80	1.05×10^{-2}	98	10.8
1	5.80	6.88×10^{-3}	98	10.8
2	5.81	1.15×10^{-2}	98	10.9
3	5.81	1.35×10^{-2}	100	11.0
4	5.79	9.05×10^{-3}	98	10.6
5	6.09	3.73×10^{-2}	100	11.7

Table 4.15 Influence of N towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0.1,$ rescale=True,spline=quintic.

With the settings of $N = 5$, the performance of the algorithm became better. This might indicate the data from region $[x_1^{ei}, x_2^{ei}]$ was useful to the prediction for the next iteration.

δ	avg	var (10^{-2})	SR	t (s)
4	6.15	2.87	100	13.8
5	5.81	1.35	100	11.0
6	6.12	1.29	99	9.61

Table 4.16 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.1, N = 3,$ rescale=True,spline=quintic.

The precision and the processing time were improved with the range factor. These revealed that the greater factor boosted the performance of extrapolation.

mode	avg	var	SR	t (s)
on	5.79	9.05×10^{-3}	98	10.6
off	/	/	/	/

Table 4.17 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.1, \delta = 5, N = 4,$ spline=quintic.

The algorithm without rescaling was not able to converge and terminate. So it can be concluded that the algorithm without rescaling does not work in some cases.

spline	avg	var	SR	t (s)
quintic	5.79	9,05e-3	98	10.6
cubic	4.10	1.31e-1	100	13.6

Table 4.18 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.1, \delta = 5, N = 4,$ rescale=True.

The quintic spline was shown to have better general performance metrices than the cubic spline, including higher accuracy, precision and processing time, which satisfied the theoretical expectation stating that the quintic spline should perform better than the cubic spline.

If the search was initialized with a starting point further away from the ideal step-size and a slightly larger step, i.e. $s_e = -105, t_e = 2, x_e = 1$,the algorithm was still able to extrapolate back to the region of interest, and then interpolate to find the acceptable step-size. The ideal step-size in this case was $x_e^* = 53$, the prediction made was $x_e' = 53.25$.

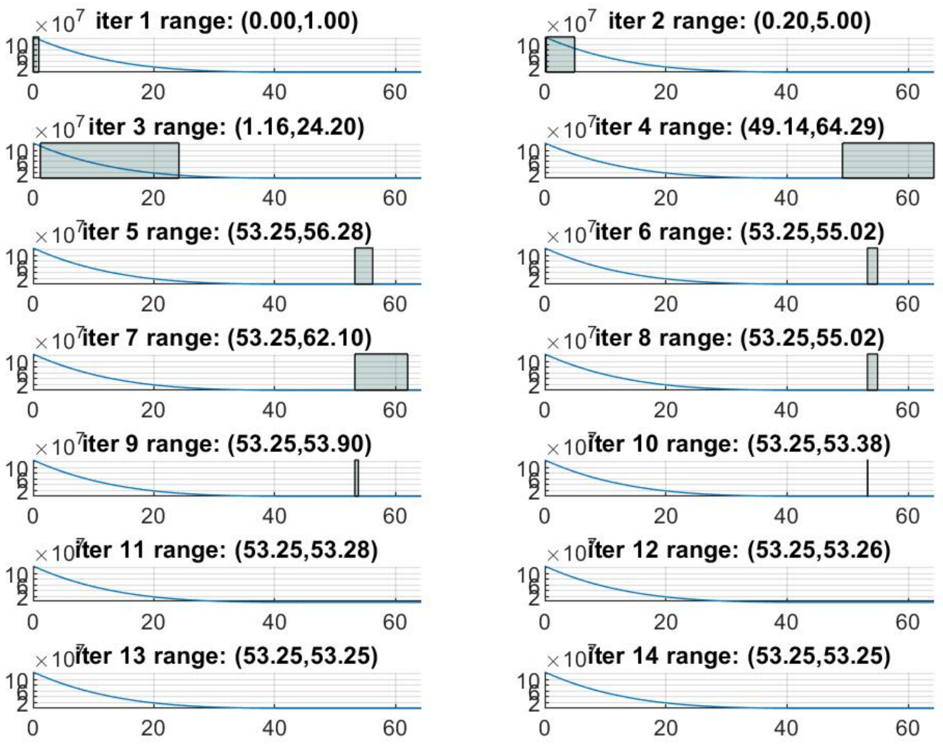

Figure 4.3 Evolution of the potential interval $[\kappa, \kappa+\Delta]$ over iteration i in the one dimensional x^4 problem with $\sigma^2 = 0$ and different start settings.

i	1-10	11	12	13	14
θ^2	$>1.00\times10^1$	1.16×10^{-2}	1.21×10^{-4}	3.61×10^{-6}	1.37×10^{-7}

Table 4.19 Evolution of the signal variance over iteration i in the search range $[\kappa, \kappa+\delta\Delta]$ from Figure 4.3.

The algorithm with different parameters settings was tested below.

σ^2	avg	var	SR	t (s)
0.001	52.7	4.79×10^{-10}	100	22.2
0.1	51.9	1.12×10^{-4}	100	22.6
1	50.8	2.28×10^{-3}	100	22.9

Table 4.20 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic.

As seen before, the predicted value and its precision was improved with the decrease of the function noise variance. Conversely, the processing time increased with the function noise variance.

N	avg	var	SR	t (s)
0	53.2	7.46×10^{-4}	100	21.8
1	53.1	1.08×10^{-2}	100	25.8
2	53.1	8.05×10^{-3}	100	21.6
3	53.1	8.61×10^{-3}	100	20.5
4	53.3	1.12×10^{-4}	100	22.6
5	54.6	1.98×10^{0}	72	22.5

Table 4.21 Influence of N towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0.1,$ rescale=True,spline=quintic.

For the setting of $N = 5$, the performance of the algorithm was reduced significantly in terms of success rate, predicted value and its precision.

δ	avg	var	SR	t (s)
4	53.1	7.20×10^{-3}	100	17.8
5	53.1	8.61×10^{-3}	100	20.5
6	54.8	2.34×10^{-12}	100	14.2

Table 4.22 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.1, N = 3,$ rescale=True,spline=quintic.

No specific trend was observed from Table 4.22 about tuning the range factor. But the maximum setting of range factor resulted in the shorter processing time and better precision again, but it had the bad accuracy.

mode	avg	var	SR	t (s)
on	53.3	1.12×10^{-4}	100	22.6
off	53.3	0	1	10.6

Table 4.23 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.1, \delta = 5, N = 4,$ spline=quintic.

In general, the algorithm without rescaling was performing poorly in the x^4 optimization problem, only 1% of runs were successful when the search was begun with the far-away starting point, and none of the runs were successful when the search was initialized with the closer starting point.

spline	avg	var	SR	t (s)
quintic	53.3	1.12×10^{-4}	100	22.6
cubic	51.9	3.88×10^{-2}	100	60.2

Table 4.24 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.1, \delta = 5, N = 4,$ rescale=True.

The quintic spline had better performances in predicted value, precision and processing time than the cubic spline.

4.3 Rosenbrock Function Problem

$$f(\mathbf{x}) = \sum_{i+1}^{N-1} [100(x_{i+1} - x_i)^2 + (1 - x_i)^2] \tag{4.4}$$

The algorithm was tested with this standard optimization problem (rosenbrock function). The search was initialized with $\mathbf{s_e} = [1.0594, 1.1099, 1.2323, 1.5156, 2.2731]', x_e = 1.8915$ and $\mathbf{t_e} = [-0.0154, -0.0196, -0.0435, -0.1036, -0.2902]'$. The ideal stepsize x_e^* was 1.8915.

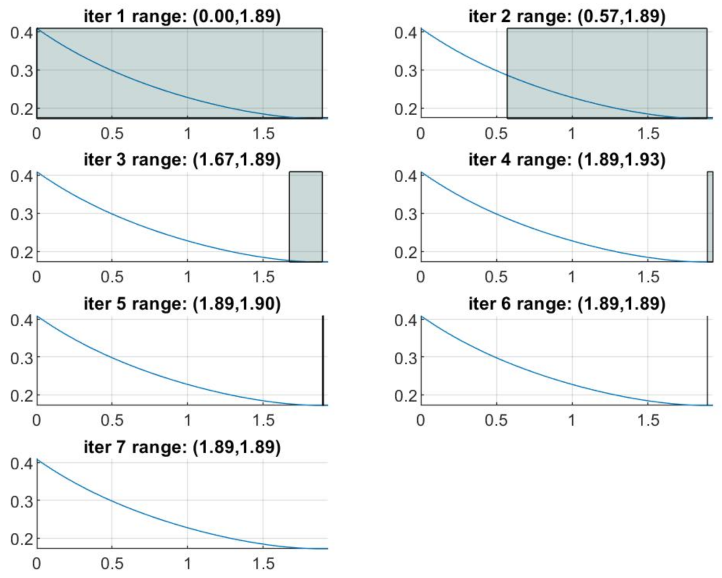

Figure 4.4 Evolution of the potential interval $[\kappa, \kappa + \Delta]$ over iteration i in the rosenbrock problem with $\sigma^2 = 0.1$.

i-th	1-2	3	4	6-7
θ^2	$>1.00\times10^2$	6.66×10^{-3}	4.00×10^{-6}	0

Table 4.25 Evolution of the signal variance over iteration in the search range $[\kappa, \kappa + \delta\Delta]$ from Figure 4.4.

Figure 4.4 showed that the algorithm was able to interpolate towards the region of interest. The predicted value x'_e was 1.89.
Various experiments were run with different parameters settings as follows.

σ^2	avg	var	SR	t (s)
0.0001	1.89	2.69×10^{-5}	100	6.35
0.001	1.90	1.93×10^{-4}	100	8.51
0.01	1.94	1.93×10^{-3}	98	8.94

Table 4.26 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic.

When the function noise variance was lower, the algorithm can perform better in terms of predicted value, precision of the predicted values and the processing time.

N	avg	var	SR	t (s)
0	1.94	1.25×10^{-3}	100	10.3
1	1.94	1.59×10^{-3}	100	8.43
2	1.94	1.52×10^{-3}	100	8.57
3	1.94	1.11×10^{-3}	100	8.54
4	1.94	1.93×10^{-3}	98	8.94
5	1.97	1.57×10^{-1}	100	8.85

Table 4.27 Influence of N towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0.01,$ rescale=True,spline=quintic.

The setting of $N = 5$ caused the predicted value less accurate and precise, while the setting without remain data $N = 0$ increased the processing time. This may indicate that $N = 1$ was already the best setting in this case.

δ	avg	var (10^{-3})	SR	t (s)
4	1.95	2.67	100	9.24
5	1.94	1.11	100	8.54
6	1.93	1.27	100	8.93

Table 4.28 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, N = 3,$ rescale=True,spline=quintic.

The accuracy of the prediction increased with the range factor.

mode	avg	var	SR	t (s)
on	1.94	1.93×10^{-3}	98	8.94
off	1.90	1.40×10^{-4}	99	4.98

Table 4.29 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, \delta = 5, N = 3,$ spline=quintic.

The algorithm had better general performances without rescaling.

spline	avg	var	SR	t (s)
quintic	1.94	1.93×10^{-3}	98	8.94
cubic	2.02	2.69×10^{-2}	100	16.5

Table 4.30 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.01, \delta = 6, N = 3,$ rescale=True.

The cubic spline performed badly in this optimization problem, except the success rate.

4.4 Sigmoid Optimization Problem

$$f(x) = \frac{e^{-2x}}{1+e^{-2x}} \tag{4.5}$$

This optimization problem was to test whether the algorithm can extrapolate out of one flat region, move to next lower flat region and continue searching, to find the better sub-optimal solution.

The experiment was initialized with $s_e = -5, t_e = 1$ and $x_e = 0.5$. When training with the noisy observations, the algorithm was able to reach the region $[5, \infty]$ and slowed down after reaching a lower plateau in the 4-th iteration since the function noise variance became more dominant than the signal variance as illustrated in the Figure 4.5 and Table 4.31.

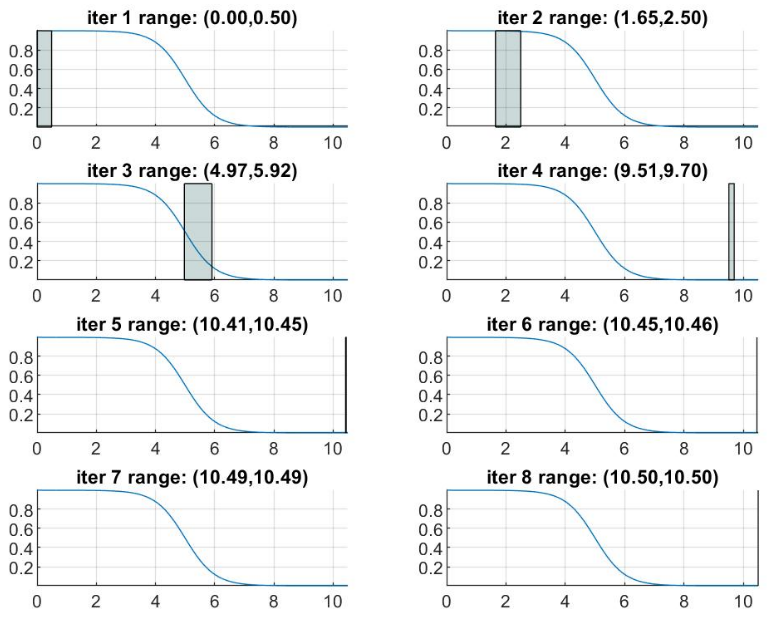

Figure 4.5 Evolution of the potential interval $[\kappa, \kappa+\Delta]$ over iteration in the sigmoid problem with $\sigma^2 = 0.0004$.

i	1	2	3	4
θ^2	1.83×10^{-5}	1.32×10^{-1}	1.44×10^{-2}	9.19×10^{-10}
i	5	6	7	8
θ^2	4.36×10^{-12}	2.13×10^{-13}	7.90×10^{-15}	3.13×10^{-16}

Table 4.31 Evolution of the signal variance over iteration in the search range $[\kappa, \kappa+\delta\Delta]$ from Figure 4.5.

In this set of experiments, the within rate (WR) was introduced to count the number of loops that can successfully predict the step-size greater than 5. With this new performance metric, the algorithm with different parameters settings were tested. The difference between conditions equations 3.6 and 3.7 for retaining data was also tested to assess how they affect the extrapolation capability.

σ^2	avg	var	SR	WR(>5)	t (s)
0	11.8	0	100	100	15.3
0.0001	9.10	8.32×10^0	100	83	11.1
0.0004	5.42	1.40×10^1	100	31	7.94
0.001	3.83	5.08×10^0	100	12	7.11

Table 4.32 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic.

The within rate and the processing time decreased with the function noise variance. The precision reached the minimum when $\sigma^2 = 0.0004$ because the average prediction stuck at the cliff $x'_e = 5$, causing the predicted value can either be 0 or 1.

N	avg	var	SR	WR(>5)	t (s)
0	8.75	9.09	100	80	11.0
1	9.29	7.28	100	86	11.3
2	8.96	8.25	100	83	11.1
3	8.86	8.86	100	82	10.9
4	9.10	8.32	100	83	11.1
5	9.42	30.8	100	84	11.5

Table 4.33 Influence of N towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0.0001,$ rescale=True,spline=quintic.

No particular trend was observed from Table 4.33. The setting of $N = 5$ had the worst precision. This might indicate there was not any retained data for every iteration i because the retain region was defined incorrectly. The algorithm was abandoning too much data whenever the potential interval was shifting towards the right and the previous data in the new negative domain was abandoned.

δ	avg	var	SR	WR(>5)	t (s)
4	5.06	8.17	100	45	9.79
5	8.86	8.86	100	82	10.9
6	9.77	4.67×10^{-3}	99	99	8.09

Table 4.34 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.0001, N = 3,$ rescale=True,spline=quintic.

The algorithm was able to extrapolate further away with the greater value of range factor δ so the setting of $N = 6$ had achieved the best performances.

cond	avg	var	SR	WR(>5)	t (s)
upper (new: Equation 3.7)	4.60	8.08	100	36	9.45
lower (Equation 3.6)	5.06	8.17	100	45	9.79

Table 4.35 Influence of the retain condition towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 4, N = 3, \sigma^2 = 0.0001$, rescale=True,spline=quintic.

With the new condition to retain the data, the within-rate greater than 5 was dropped. This indicated that, retaining data from the region closer to $x_1^{ei} + \delta(x_2^{ei} - x_1^{ei})$ improved the prediction for the next iteration more than those closer to x_1^{ei} because it can avoid the initial extrapolation for each iteration.

mode	avg	var	SR	WR(>5)	t (s)
off	/	/	/	/	/
on	5.06	8.17	100	45	9.79

Table 4.36 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.0001, \delta = 4, N = 3$, spline=quintic.

The algorithm was not able to operate without rescaling.

spline	avg	var	SR	WR(>5)	t (s)
C	6.92	3.08	97	87	10.9
Q	5.06	8.17	100	45	9.79

Table 4.37 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0.0001, \delta = 4, N = 3$, rescale=True.

The cubic spline had better general performance than the quintic spline. It extrapolated further away, had higher within-rate and precision. But the quintic spline had higher success rate and shorter processing time.

4.5 Sinc Function

$$f(x) = -\frac{\sin(x)}{x} \tag{4.6}$$

My algorithm was subjected to this optimization problem to assess how it performed when there were many local minima within the search region.

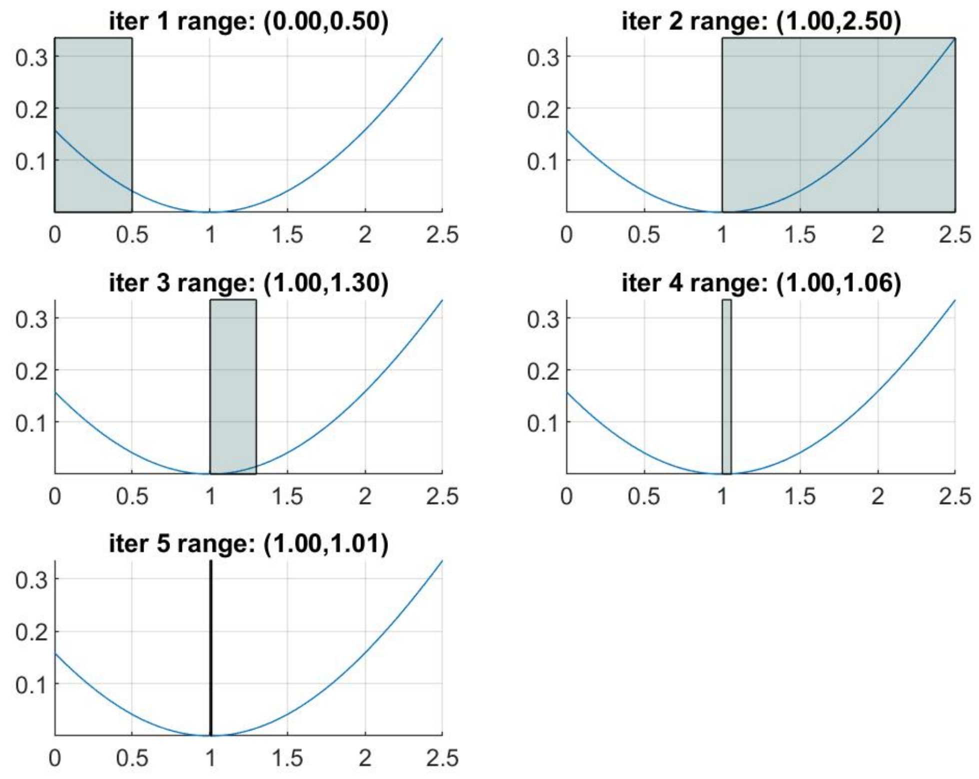

Figure 4.6 Evolution of the potential interval $[\kappa, \kappa+\Delta]$ over iteration in the sinc problem with $\sigma^2 = 0$.

i	1	2	3	4	5
θ^2	$>1.00\times10^{0}$	1.35×10^{-1}	1.97×10^{-2}	4.20×10^{-5}	7.22×10^{-8}

Table 4.38 Evolution of the signal variance over iteration in the search range $[\kappa, \kappa+\delta\Delta]$ from Figure 4.6.

The first search was initialized with $s_e = -1, t_e = 1$ and $x_e = 0.5$. The global minimum was in $x_e^* = 1$. From the Figure 4.6, the algorithm was able to converge into the closest minimum so the predicted value x_e' was 1.00.

Some experiments were conducted to test how different parameters settings affect the performance.

σ^2	avg	var	SR	WR(0,2)	t (s)
0	1.01	0	100	100	5.27
0.0001	1.00	1×10^{-5}	100	100	6.83
0.01	1.04	4×10^{-3}	100	100	6.93

Table 4.39 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic.

The higher the function noise variance was, the more the processing time was required and the lower precision the prediction had. But their predicted values were able to converge to the optimal value $x_e^* = 1$.

N	avg	var	SR	WR(0,2)	t (s)
0	1.01	0	100	100	5.22
1	1.01	0	100	100	5.24
2	1.01	0	100	100	5.26
3	1.01	0	100	100	5.28
4	1.01	0	100	100	5.27
5	1.39	8.67	89	87	13.2

Table 4.40 Influence of N towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0,$ rescale=True,spline=quintic.

Only the setting with the most remain entries ($N = 5$) had the worst general performance, other settings were having similar performance. This may indicate that the retained data from region (x_1^{ei}, x_2^{ei}) did not improve the prediction for the next iteration. This might be because those data did not improve the prediction for extrapolation.

δ	avg	var	SR	WR(0,2)	t (s)
4	0.999	0	100	100	3.79
5	1.01	0	100	100	5.28
6	1.00	0	100	100	6.79

Table 4.41 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0, N = 3,$ rescale=True,spline=quintic.

The greater the search factor δ was, the longer the processing time needed for the search.

mode	avg	var	SR	WR(0,2)	t (s)
off	0.996	0	100	100	2.97
on	1.01	0	100	100	5.27

Table 4.42 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0, \delta = 5, N = 4,$ spline=quintic.

The algorithm without rescaling had a better predicted value and processing time than that with rescaling. But the accuracy difference was not large.

spline	avg	var	SR	WR(0,2)	t (s)
C	1.01	0	100	100	6.54
Q	1.01	0	100	100	5.27

Table 4.43 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0, \delta = 5, N = 4,$ rescale=True.

The processing time of the algorithm with quintic spline was shorter than that with cubic spline.

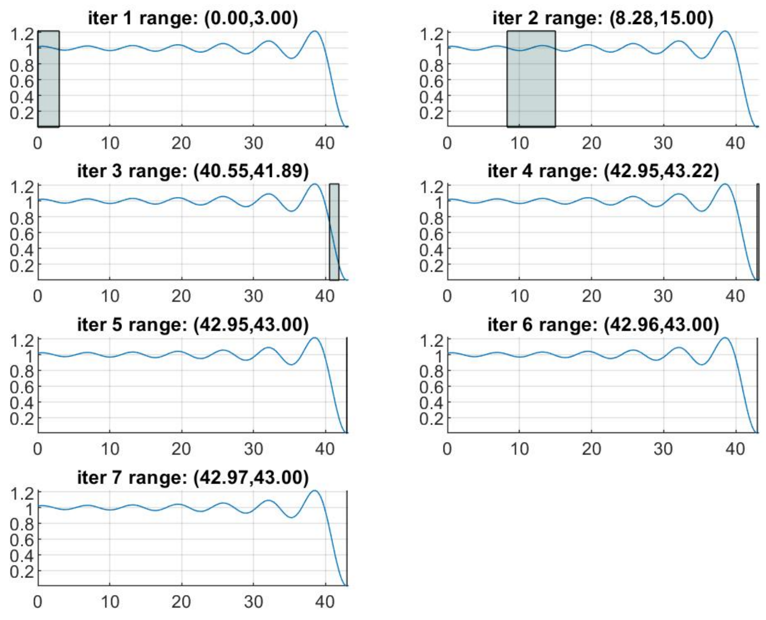

Figure 4.7 Evolution of the potential interval $[\kappa, \kappa + \Delta]$ over iteration in the sinc problem with $\sigma^2 = 0$.

i	1	2	3	4
θ^2	4.63×10^{-4}	6.98×10^{-2}	1.99×10^{-1}	1.20×10^{-2}
i	5	6	7	/
θ^2	1.31×10^{-5}	5.37×10^{-6}	2.21×10^{-6}	/

Table 4.44 Evolution of the signal variance over iteration in the search range $[\kappa, \kappa+\delta\Delta]$ from Figure 4.7.

In the second case studies, the experiments were initialized with the settings of $s_e = -43, t_e = 1$ and $x_e = 3$. Unexpectedly, the algorithm was able to reach the global minimum $x_e^* = 43$ far away from the starting point $s_e = -43$.

σ^2	avg	var	SR	WR(42,44)	t (s)
0	43.0	0	100	100	8.90
0.0001	45.9	7.04×10^{1}	100	89	12.0
0.01	45.5	3.99×10^{3}	98	40	14.4

Table 4.45 Influence of the noise variance towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, N = 4,$ rescale=True,spline=quintic.

Without noisy observations, the algorithm was more able to converge towards the global minimum, and took less processing time and had a better within-rate. The precision and the within rate were decreased and the processing time was increased with the noise level.

N	avg	var	SR	WR(42,44)	t (s)
0	43.0	0	100	100	8.91
1	43.0	0	100	100	8.92
2	43.0	0	100	100	8.91
3	43.0	0	100	100	8.91
4	43.0	0	100	100	8.90
5	43.0	0	100	100	8.88

Table 4.46 Influence of N towards various performance metrics. Settings were: $\rho^2 = 0, \delta = 5, \sigma^2 = 0,$ rescale=True,spline=quintic.

The performance remained unchanged with different settings of retain number N, which revealed that the number of data was always 0.

δ	avg	var	SR	WR(42,44)	t (s)
4	22.6	0	100	0	11.8
5	43.0	0	100	100	8.91
6	/	/	/	/	/

Table 4.47 Influence of the range factor towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0, N = 3,$ rescale=True,spline=quintic.

With $\delta = 4$, the algorithm was not able to reach the global minimum. With $\delta = 6$, it was not able to operate, which indicated that there was an unexpected bug in the algorithm. The algorithm with $\delta = 5$ was having the best general performance among others.

mode	avg	var	SR	WR(42,44)	t (s)
off	43.5	0	100	100	21.1
on	43.0	0	100	100	8.90

Table 4.48 Influence of the rescale mode towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0, \delta = 5, N = 4,$ spline=quintic.

The algorithms with and without rescaling were able to reach the global minimum. But the one with rescaling had a better prediction and shorter processing time.

spline	avg	var	SR	WR(42,44)	t (s)
C	50.9	0	100	0	11.8
Q	43.0	0	100	100	8.90

Table 4.49 Influence of the spline model towards various performance metrics. Settings were: $\rho^2 = 0, \sigma^2 = 0, \delta = 5, N = 4,$ rescale=True.

The quintic spline was having better general performance metrices than the cubic spline. The algorithm with cubic spline had extend the search range further away from both the starting point and the global minimum, and was not able to find the global minimum.

Chapter 5

Conclusion

From the experimental results, it can be concluded that the algorithm was able to interpolate, extrapolate and terminate automatically. Although the probabilistic line search algorithm was able to improve the accuracy of the prediction, the processing time of this implementation was not acceptable as it took more than 5 seconds to run.

When increasing the function value noise variance, the accuracy and precision of the prediction were decreased. Changing the maximum number of retained data did not affect the performance at all, except N was set to 5 which sometimes worsened the performance. This might imply there was actually no extra data (except the new starting entries) retained after each iteration i, so that the performance of $N = 0$ was the same as that of $N > 0$. Sometimes when the range factor was increased, it improved the accuracy. The algorithm with rescaling was more stable than that without. When comparing the quintic spline with the cubic spline, it was found that the quintic spline did successfully out-performe the cubic-spline in most of the cases, but sometimes the cubic spline can still have better performance.

The fact that the processing time of the algorithm dealing with different problems was long may imply that there were some design problems. First of all, whenever the potential interval was redefined each iteration in the while loop i, some previous data which were in the positive domain in the previous iteration were redefined to enter the negative domain in the next iteration and got abandoned. This caused the algorithm to re-sample new data from the objective, which increased the processing time. And the region where the retained data should exist was defined incorrectly as well, so there were either no or small number of data retained.

Although the experiments had proved that the algorithm was able to terminate with expected improvement, the convergence rate was still too slow, compared to the state-of-art optimizers which took only less than a second. Narrowing down the potential interval until it converged to a termination point was a long process because the algorithm had to re-estimate the bounds repetitively, On the other hand, choosing a termination point using the Wolfe conditions and their variations did not require the potential interval width to converge and be small than a threshold (argument tolerance).

Moreover, the rescaling was not done properly, which was indicated by the fact that sometimes the algorithm without rescaling can outperform that with.

There were also some numerical problems while running the algorithm. Sometimes the covariance of the predictive distribution may lead to the complex number and made the algorithm fail to compute the expected improvement, which stopped to suggest the new evaluation point.

During the experiments, the Newton Optimization with negative sign always failed to re-estimate the new hyperparameters settings with more than 2 entries, which were not expected. This is not good since the better re-estimation can allow the algorithm to terminate earlier when the predicted τ^2 was less than a threshold. The good settings of hyperparameters are also helpful to avoid the overfitting of the data, and predict the optimal solution.

In this circumstance, the algorithm was designed to use the backup plan with the positive sign of Newton Optimization, which was given as follows,

$$\bar{\tau}[t+1] = \bar{\tau}[t] + H^{-1}g \tag{5.1}$$

It was found that the Newton Optimization was more likely to compute a lower nlml with the equation above. This might be due to the fact that under certain conditions, the Hessian was evaluated badly, causing an opposite sign. So changing negative to positive might work. But simply backing up with an opposite sign casually did not analyze why the original version of Newton Optimization did not work, more investigations are necessary to examine the conditions leading to the malfunctioning Newton Optimization in the future.

Some researchers [21] suggested that to encounter the ill-conditioned Hessian and gradient problem, the equation should be rewritten as

$$\theta \leftarrow \theta + g(H + \lambda I)^{-1} V$$

This implementation is also worthwhile to investigate.

After these experiments, the probabilistic termination criteria for the strong Wolfe conditions suggested by some research studies [10] should be considered. It is necessary to retain data in a more efficient strategy to reduce the resampling.

For the future work, it is also possible to model the second derivative of the objective as a stochastic process. Our algorithm should be integrated with BFGS and standard line search to test with various optimization problems to assess its performance.

Bibliography

[1] J. Nocedal and S. J. Wright, *Numerical Optimization.* New York, NY, USA: Springer, second ed., 2006.

[2] E. Brochu, V. M. Cora, and N. de Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning,"

[3] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *in COMPSTAT*, 2010.

[4] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[5] V. Nguyen, S. Gupta, S. Rana, C. Li, and S. Venkatesh, "Regret for expected improvement over the best-observed value and stopping condition," in *Proceedings of the Ninth Asian Conference on Machine Learning*, pp. 279–294, 2017.

[6] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning).* The MIT Press, 2005.

[7] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 2951–2959, Curran Associates, Inc., 2012.

[8] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, pp. 148–175, 2016.

[9] S. Rana, C. Li, S. Gupta, V. Nguyen, and S. Venkatesh, "High dimensional Bayesian optimization with elastic Gaussian process," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 2883–2891, PMLR, 06–11 Aug 2017.

[10] M. Mahsereci and P. Hennig, "Probabilistic line searches for stochastic optimization," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 181–189, Curran Associates, Inc., 2015.

[11] C. De Boor, *A practical guide to splines; rev. ed.* Applied mathematical sciences, Berlin: Springer, 2001.

[12] J. J. Moré and D. J. Thuente, “Line search algorithms with guaranteed sufficient decrease,” *ACM Trans. Math. Softw.*, vol. 20, pp. 286–307, Sept. 1994.

[13] E. Ostertagová, “Modelling using polynomial regression,” *Procedia Engineering*, vol. 48, pp. 500 – 506, 2012. Modelling of Mechanical and Mechatronics Systems.

[14] E. W. Weisstein, “Cubic spline.” Last visited on 7/8/2019.

[15] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics &Amp; Geometric Modeling*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987.

[16] K. HJ., *A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise*. ASME, 1964.

[17] E. Vazquez and J. Bect, “Convergence properties of the expected improvement algorithm with fixed mean and covariance functions,” *Journal of Statistical Planning and Inference*, vol. 140, pp. 3088–3095, 11 2010.

[18] A. D. Bull, “Convergence rates of efficient global optimization algorithms,” *J. Mach. Learn. Res.*, vol. 12, pp. 2879–2904, Nov. 2011.

[19] Z. Wang and N. de Freitas, “Theoretical analysis of bayesian optimisation with unknown gaussian process hyper-parameters,” 06 2014.

[20] I. O. Ryzhov, “On the Convergence Rates of Expected Improvement Methods,” *Operations Research*, vol. 64, pp. 1515–1528, December 2016.

[21] Y. N. Dauphin, R. Pascanu, Ç. Gülçehre, K. Cho, S. Ganguli, and Y. Bengio, “Identifying and attacking the saddle point problem in high-dimensional non-convex optimization,” *CoRR*, vol. abs/1406.2572, 2014.

Appendix A

pseudocode

A.1 BFGS

Algorithm 2 BFGS

```
1: Initialize starting point $\mathbf{x_0}$, Hessian $\mathbf{H_0} = \mathbf{I}$
2: Set convergence $\varepsilon = 1 \times 10^{-4}$, $k \leftarrow 0$
3: while $\|\nabla f_k\| > \varepsilon$ do
4:     Compute search direction $\mathbf{d_k}$, $\mathbf{d_k} = -\mathbf{H_k^{-1}}\nabla f(\mathbf{x_k})$
5:     Compute step-size $\alpha_k$ via line search, $\alpha_k = arg\min_\alpha f(\mathbf{x_k} + \alpha\mathbf{d_k})$
6:     Update $\mathbf{x_{k+1}} = \mathbf{x_k} + \alpha_k\mathbf{d_k}$, $\mathbf{s_k} = \alpha_k\mathbf{d_k}$
7:     Update $\mathbf{q_k} = \nabla f(\mathbf{x_{k+1}}) - \nabla f(\mathbf{x_k})$.
8:     Compute $\mathbf{H_{k+1}}$ by, $\mathbf{H_{k+1}} = \mathbf{H_k} + \frac{\mathbf{q_k q_k^T}}{\mathbf{q_k^T s_k}} - \frac{\mathbf{H_k s_k s_k^T H_k^T}}{\mathbf{s_k^T H_k s_k}}$
9:     Check sufficient decrease condition via,
           $\mathbf{B_1} = f(\mathbf{x_{k+1}}) \leq f(\mathbf{x_k}) + c_1\alpha_k\mathbf{d_k^T}\nabla f_k$
10:    Check curvature condition via,
           $\mathbf{B_2} = \|\mathbf{d_k^T}\nabla f_{k+1}\| \leq c_2\|\mathbf{d_k^T}\nabla f_k\|$
11:    if $\mathbf{B_1}$ and $\mathbf{B_2}$ then
12:        Store the progress in the memory via,
13:        $\mathbf{M} \leftarrow \mathbf{x_{k+1}}$, $k \leftarrow k + 1$
14:    else
15:        return $\mathbf{M}$
16:    end if
17: end while
```

A.2 Line Search

Algorithm 3 Line Search

```
1: Set $\alpha_0 \leftarrow 0$, choose $\alpha_1 > 0$ and $\alpha_{max}$, $i \leftarrow 1$
2: while True do
3:     Evaluate $\phi(\alpha_i)$
4:     if $\phi(\alpha_i) > \phi(0) + c_1\alpha_i\phi'(0)$ or $[\phi(\alpha_i) \geq \phi(\alpha_{i-1})$and $i > 1]$ then
5:         $\alpha_* \leftarrow zoom(\alpha_{i-1}, \alpha_i)$ and $stop$
6:     end if
7:     Evaluate $\phi'(\alpha_i)$
8:     if $|\phi'(\alpha_i)| \leq -c_2\phi'(0)$ then $set$ $\alpha_* \leftarrow \alpha_i$ and $stop$
9:     end if
10:    if $\phi'(\alpha_i) \geq 0$ then $set$ $\alpha_* \leftarrow zoom(\alpha_i, \alpha_{i-1})$ and $stop$
11:    end if
12:    Choose $\alpha_{i+1} \in (\alpha_i, \alpha_{max})$, $i \leftarrow i+1$
13: end while
```

A.3 zoom

Algorithm 4 zoom

```
1: while True do
2:     Interpolate (using quadratic,cubic,or bisection) to find a trial step length $\alpha_j$ between
   $\alpha_{low}$ and $\alpha_{high}$
3:     Evaluate $\phi(\alpha_j)$
4:     if $\phi(\alpha_j) > \phi(0) + c_1\alpha_j\phi'(0)$ or $\phi(\alpha_j) \geq \phi(\alpha_{low})$ then
5:         $\alpha_{high} \leftarrow \alpha_j$
6:     else
7:         Evaluate $\phi'(\alpha_j)$
8:         if $|\phi'(\alpha_j)| \leq -c_2\phi'(0)$ then Set $\alpha_* \leftarrow \alpha_j$ and $stop$
9:         end if
10:        if $\phi'(\alpha_j)(\alpha_{high} - \alpha_{low}) \geq 0$ then $\alpha_{high} \leftarrow \alpha_{low}$
11:        end if
12:        $\alpha_{low} \leftarrow \alpha_j$
13:    end if
14: end while
```

Appendix B

Smoothing Spline

To fit the data with the smoothing splines involves finding the function $g(x)$ which minimizes,

$$\frac{1}{n}\sum_{i=1}^{n}(g(x_i)-y(x_i))^2+\lambda\int_{-\infty}^{\infty}(g^{(m)}(x))^2dx$$

where we assume that each location $x_i \in x$, $x \in (0,c)$ and c is the maximum entry. Therefore, the solution is a polynomial of degree $m-1$ for $x \in [-\infty, 0]$ and $[c,\infty]$ and a piecewise polynomial of degree $2m-1$ with $2m-2$ continuous derivatives for $x \in (0,c)$.
When $m=2$, the solution is a smoothing cubic spline; when $m=3$, the solution is a smoothing quintic spline.
The same solution can be obtained by using Bayesian inference of $g(x)$ with a prior from eq 2.14.

$$\begin{aligned} g(x) &= f(x)+h(x)^\top\beta \\ &= f(x)+\sum_{j=0}^{m-1}\beta_j x^i \end{aligned}$$

where $\beta_j \sim \mathcal{N}(\mathbf{b},\mathbf{B})$ with $\mathbf{B}\to\infty$ (uninformative prior about a polynomial of degree $m-1$) and $f(x)$ is given by (m-1)-fold integrated random walk,

$$f(x)=\int_0^c \frac{(x-u)_+^{m-1}}{(m-1)!}Z(u)du$$

where $Z(u)$ is a Gaussian white noise process with covariance $\delta(u-u')$.
If consider eq 2.15 and the covariance of $h(x)^\top\mathbf{B}h(x')$ is given by $(h(x)^\top\beta)(h(x)^\top\beta)^\top$, then

$k(x,x') = f(x)f(x')^\top$ so,

$$k(x,x') = \frac{\theta^2}{((m-1)!)^2}\int_0^c (x-u)_+^{m-1}(x'-u)_+^{m-1}du$$

To integrate the integral above , the condition $u > x,x'$ must be fulfilled. If $x > x'$, $c \to x'$; if $x < x'$, $c \to x$. By ignoring the max function $max(0,\cdot) = (\cdot)_+$, the covariance function becomes,

$$\begin{aligned}k(x,x') &= \frac{\theta^2}{((m-1)!)^2}\int_0^c (x-u)^{m-1}(x'-u)^{m-1}du \\ &= \frac{\theta^2}{((m-1)!)^2}\int_0^{\min(x,x')} (x-u)^{m-1}(x'-u)^{m-1}du\end{aligned}$$

For cubic spline $m = 2$,

$$\begin{aligned}k(x,x') &= \theta^2\int_0^{\min(x,x')}(x-u)(x'-u)du \\ &= \theta^2\int_0^{\min(x,x')}(x-u)(x'-u)du \\ &= \theta^2\int_0^{\min(x,x')}(xx'+u^2-(x+x')u)du \\ &= \theta^2(xx'\min(x,x')+1/3\min(x,x')^3 \\ &\qquad -1/2(x+x')\min(x,x')^2)\end{aligned}$$

If $x > x'$, then

$$\begin{aligned}k(x,x') &= \theta^2(1/3\min(x,x')^3+\min(x,x') \\ &\qquad (xx'-xx'/2-x'^2/2)) \\ &= \theta^2(1/3\min(x,x')^3+1/2\min(x,x')x'(x-x'))\end{aligned}$$

Conversely,

$$k(x,x') = \theta^2(1/3\min(x,x')^3+1/2\min(x,x')x(x'-x))$$

So, in general,

$$k(x,x') = \theta^2(1/3\min(x,x')^3+1/2|x-x'|\min(x,x')^2)$$

With ${}^{\partial^i}k^{\partial^j} = \frac{\partial^{i+j}k}{\partial x^i \partial x'^j}$, the remaining covariance functions can be derived.
For $x > x'$, setting $\theta^2 = 1$

$$\begin{aligned}
k &= 1/3x'^3 + 1/2(x-x')x'^2 \\
k^{\partial} &= x'^2 + x'x - 3/2x'^2 = x'x - 1/2x'^2 \\
&= 1/2(2x'x - x'^2) \\
{}^{\partial}k^{\partial} &= x'
\end{aligned}$$

For $x < x'$,

$$\begin{aligned}
k &= 1/3x^3 + 1/2(x'-x)x^2 \\
k^{\partial} &= 1/2x^2 \\
{}^{\partial}k^{\partial} &= x
\end{aligned}$$

So, in general,

$$\begin{aligned}
k^{\partial} &= 1/2((x-x')\min(x,x') + xx') \\
{}^{\partial}k &= 1/2((x'-x)\min(x,x') + xx') \\
{}^{\partial}k^{\partial} &= \min(x,x')
\end{aligned}$$

Similarly for quintic spline $m = 3$, setting $\theta^2 = 4$ to ignore the factor,

$$\begin{aligned}
k(x,x') &= \theta^2/4 \int_0^{\min(x,x')} (x-u)^2(x'-u)^2 du \\
&= \int_0^{\min(x,x')} (x^2 + u^2 - 2ux)(x'^2 + u^2 - 2ux')du \\
&= \int_0^{\min(x,x')} (x^2x'^2 + u^4 + (x^2 + x'^2)u^2 \\
&\quad - 2u^3(x+x') - 2uxx'(x+x') + 4u^2xx')du \\
&= x^2x'^2 \min(x,x') + 1/5\min(x,x')^5 \\
&+ 1/3(x^2 + x'^2 + 4xx')\min(x,x')^3 \\
&- 1/2(x+x')\min(x,x')^4 - xx'(x+x')\min(x,x')^2 \\
&= 1/5\min(x,x')^5 + 1/3(x-x')^2\min(x,x')^3 \\
&+ 2xx'\min(x,x')^3 + x^2x'^2\min(x,x') \\
&- 1/2(x+x')\min(x,x')^4 - xx'(x+x')\min(x,x')^2
\end{aligned}$$

If $x > x'$, the last 4 terms become,

$$\begin{aligned}&2xx'^4 + x^2x'^3 - 1/2xx'^4 - 1/2x'^5 - x^2x'^3 - xx'^4\\&= 1/2xx'^4 - 1/2x'^5\end{aligned}$$

If $x < x'$, the last 4 terms become,

$$\begin{aligned}&2x^4x' + x^3x'^2 - 1/2x^5 - 1/2x'x^4 - x^4x' - x^3x'^2\\&= 1/2x^4x' - 1/2x^5\end{aligned}$$

In general, the last 4 terms are summarized as,

$$\begin{aligned}&2xx'\min(x,x')^3 + x^2x'^2\min(x,x')\\&-1/2(x+x')\min(x,x')^4 - xx'(x+x')\min(x,x')^2\\&= 1/2|x-x'|\min(x,x')^4\end{aligned}$$

Thus, the covariance k is given by,

$$\begin{aligned}k = \theta^2/4(&1/5\min(x,x')^5 + 1/2|x-x'|\min(x,x')^4\\&+1/3(x-x')^2\min(x,x')^3)\end{aligned}$$

For $x > x'$, setting $\theta^2 = 4$, other covariance matrices are,

$$\begin{aligned}k &= 1/5x'^5 + 1/2xx'^4 - 1/2x'^5 + 1/3x^2x'^3 + 1/3x'^5\\&\quad -2/3xx'^4\\&= 1/30x'^5 - 1/6xx'^4 + 1/3x^2x'^3\\k^{\partial} &= 1/6x'^4 - 2/3xx'^3 + x^2x'^2\\&= 2/3(x'-x)x'^3 - 1/2x'^4 + x^2x'^2\\&= 2/3(x'-x)x'^3 + 1/2x^2x'^2 + 1/2x^2x'^2 - 1/2x'^4\\&= 2/3(x'-x)x'^3 + 1/2(x^2 - x'^2)x'^2 + 1/2x^2x'^2\\{}^{\partial}k^{\partial} &= -2/3x'^3 + 2xx'^2 = 1/3x'^3 - x'^3 + 2xx'^2\\&= 1/3x'^3 + 1/2(4x - 2x')x'^2\\&= 1/3x'^3 + (x-x')x'^2 + xx'^2\end{aligned}$$

For $x < x'$,

$$\begin{aligned}
k &= 1/5x^5 + 1/2x'x^4 - 1/2x^5 + 1/3x^5 + 1/3x'^2x^3 \\
&\quad - 2/3x^4x' \\
&= 1/30x^5 - 1/6x^4x' + 1/3x^3x'^2 \\
k^{\partial} &= 2/3x^3x' - 1/6x^4 \\
&= 2/3(x'-x)x^3 + 2/3x^4 - 1/6x^4 \\
&= 2/3(x'-x)x^3 + 1/2x^4 \\
&= 2/3(x'-x)x^3 + 1/2x^4 - 1/2x'^2x^2 + 1/2x'^2x^2 \\
&= 2/3(x'-x)x^3 + 1/2(x^2 - x'^2)x^2 + 1/2x'^2x^2 \\
{}^{\partial}k^{\partial} &= 2x^2x' - 2/3x^3 = 1/3x^3 + 2x^2x' - x^3 \\
&= 1/3x^3 + 1/2(4x' - 2x)x^2 \\
&= 1/3x^3 + (x'-x)x^2 + x'x^2
\end{aligned}$$

Therefore,

$$\begin{aligned}
k^{\partial} &= \theta^2/4(2/3(x'-x)\min(x,x')^3 + 1/2x^2x'^2 \\
&\quad + 1/2(x^2 - x'^2)\min(x,x')^2) \\
{}^{\partial}k &= \theta^2/4(2/3(x-x')\min(x,x')^3 + 1/2x^2x'^2 \\
&\quad + 1/2(x'^2 - x^2)\min(x,x')^2) \\
{}^{\partial}k^{\partial} &= \theta^2(1/3\min(x,x')^3 + 1/2|x-x'|\min(x,x')^2)
\end{aligned}$$

Alternatively,

$$\begin{aligned}
{}^{\partial}k^{\partial} &= \theta^2(1/3\min(x,x')^3 + |x-x'|\min(x,x')^2) \\
&\quad + \max(x,x')\min(x,x')^2
\end{aligned}$$

Appendix C

Numerical Considerations

To encounter this problem, we have to re-arrange $\mathbf{K_y}$ as follows,

$$\mathbf{K_y} = \begin{bmatrix} \mathbf{K_{11}} & \\ & \mathbf{K_{22}} \end{bmatrix}$$

where the empty elements are zeros and,

$$\mathbf{K_{11}} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \rho^2 \end{bmatrix}$$

$$\mathbf{K_{22}} = K_{22}^{spline} + \Sigma_{22}$$

$$\mathbf{K_{22}^{spline}} = \begin{bmatrix} k_{T-1\times T-1} & k^{\partial}_{T-1\times T-1} \\ {}^{\partial}k_{T-1\times T-1} & {}^{\partial}k^{\partial}_{T-1\times T-1} \end{bmatrix}$$

$$\Sigma_{22} = \begin{bmatrix} \sigma^2\mathbf{I_{T-1\times T-1}} & \mathbf{0}' \times \mathbf{0} \\ \mathbf{0}' \times \mathbf{0} & \rho^2\mathbf{I_{T-1\times T-1}} \end{bmatrix}$$

All other matrices also have to be re-arranged, including the feature vector $\mathbf{H}$ and the observation vector $\mathbf{Y}$.
Firstly, for the feature vector, since we also have to consider the gradient of the basis function. For cubic spline,

$$\mathbf{H} = \begin{bmatrix} \mathbf{1}_{1\times T} & \mathbf{0} \\ x_{1:T}^T & \mathbf{1}_{1\times T} \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 0 & \mathbf{1}_{1\times T-1} & \mathbf{0}_{1\times T-1} \\ x_1 & 1 & x_{2:T}^T & \mathbf{1}_{1\times T-1} \end{bmatrix} = \begin{bmatrix} \mathbf{H}(\mathbf{x_1}) & \mathbf{H}(\mathbf{x_{2:T}}) \end{bmatrix}$$

Since $x_1 = 0$,

$$\mathbf{H}(\mathbf{x_1}) = \begin{bmatrix} 1 & 0 \\ x_1 & 1 \end{bmatrix} = \mathbf{I}$$

and

$$\mathbf{H}(\mathbf{x_{2:T}}) = \begin{bmatrix} \mathbf{1}_{1\times T-1} & \mathbf{0}_{1\times T-1} \\ x_{2:T}^T & \mathbf{1}_{1\times T-1} \end{bmatrix}$$

For quintic spline,

$$\mathbf{H} = \begin{bmatrix} \mathbf{1}_{1\times T} & \mathbf{0} \\ x_{1:T}^T & \mathbf{1}_{1\times T} \\ (x_{1:T}^2)^T & 2x_{1:T} \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 0 & \mathbf{1}_{1\times T-1} & \mathbf{0}_{1\times T-1} \\ x_1 & 1 & x_{2:T}^T & \mathbf{1}_{1\times T-1} \\ x_1^2 & 2x_1 & (x_{2:T}^2)^T & 2x_{2:T}^T \end{bmatrix} = \begin{bmatrix} \mathbf{H}(\mathbf{x_1}) & \mathbf{H}(\mathbf{x_{2:T}}) \end{bmatrix}$$

Since $x_1 = 0$,

$$\mathbf{H}(\mathbf{x_1}) = \begin{bmatrix} 1 & 0 \\ x_1 & 1 \\ x_1^2 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

and

$$\mathbf{H}(\mathbf{x_{2:T}}) = \begin{bmatrix} \mathbf{1}_{1\times T-1} & \mathbf{0}_{1\times T-1} \\ x_{2:T}^T & \mathbf{1}_{1\times T-1} \\ (x_{2:T}^2)^T & 2x_{2:T}^T \end{bmatrix}$$

The observation vector is also modified,

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}(x_{1:T}) \\ \mathbf{y}'(x_{1:T}) \end{bmatrix} \rightarrow \begin{bmatrix} y(x_1) \\ y'(x_1) \\ \mathbf{y}(x_{2:T}) \\ \mathbf{y}'(x_{2:T}) \end{bmatrix} = \begin{bmatrix} \mathbf{Y}(x_1) \\ \mathbf{Y}(x_{2:T}) \end{bmatrix}$$

Some important scalars and matrices $(q, \bar{\beta}, \mathbf{Q}, \mathbf{A^{-1}}, \mathbf{W}, \mathbf{Z})$ are derived as follows, Cubic Spline: Given $\mathbf{H_1} = \mathbf{I}$, $\mathbf{A} = \mathbf{H}\mathbf{K_y^{-1}}\mathbf{H}^\top$

$$\mathbf{A} = \mathbf{K_{11}^{-1}} + \mathbf{H_2}\mathbf{K_{22}^{-1}}\mathbf{H_2^\top} = \begin{pmatrix} a+\sigma^{-2} & b \\ b & c+\rho^{-2} \end{pmatrix}$$

$$\det(\mathbf{A}) = (a+\sigma^{-2})(c+\rho^{-2}) - b^2 = \frac{1}{\sigma^2\rho^2}((\sigma^2 a+1)(\rho^2 c+1) - \sigma^2\rho^2 b^2)$$

$$\mathbf{A}^{-1} = \frac{\sigma^2\rho^2}{\begin{pmatrix}(\sigma^2 a+1)(\rho^2 c+1)\\ -\sigma^2\rho^2 b^2\end{pmatrix}} \begin{pmatrix} c+\rho^{-2} & -b \\ -b & a+\sigma^{-2}\end{pmatrix}$$

$$= \frac{1}{\begin{pmatrix}(\sigma^2 a+1)(\rho^2 c+1)\\ -\sigma^2\rho^2 b^2\end{pmatrix}} \begin{pmatrix} \sigma^2(\rho^2 c +1) & -\sigma^2\rho^2 b \\ -\sigma^2\rho^2 b & \rho^2(\sigma^2 a +1)\end{pmatrix} = q\mathbf{Q}$$

$$\bar{\beta} = \mathbf{A}^{-1}\mathbf{H}\mathbf{K}^{-1}\mathbf{y} = q\mathbf{Q}\mathbf{H}\mathbf{K}^{-1}\mathbf{y} = q\mathbf{Q}\begin{pmatrix}\mathbf{H_1} & \mathbf{H_2}\end{pmatrix}\begin{pmatrix}\mathbf{K_{11}^{-1}} & \mathbf{0}\\ \mathbf{0} & \mathbf{K_{22}^{-1}}\end{pmatrix}\begin{pmatrix}\mathbf{Y_1}\\ \mathbf{Y_2}\end{pmatrix}$$

$$= q\mathbf{Q}\begin{pmatrix}\mathbf{H_1} & \mathbf{H_2}\end{pmatrix}\begin{pmatrix}\mathbf{K_{11}^{-1}Y_1}\\ \mathbf{K_{22}^{-1}Y_2}\end{pmatrix} = q\underbrace{\mathbf{QH_1K_{11}^{-1}}}_{\mathbf{W}}\mathbf{Y_1} + q\mathbf{QH_2K_{22}^{-1}Y_2}$$

$$= q\mathbf{WY_1} + q\mathbf{QH_2K_{22}^{-1}Y_2}$$

$$\mathbf{W} = \mathbf{QH_1K_{11}^{-1}} = \begin{pmatrix}\sigma^2(\rho^2 c+1) & -\sigma^2\rho^2 b\\ -\sigma^2\rho^2 b & \rho^2(\sigma^2 a+1)\end{pmatrix}\begin{pmatrix}\sigma^{-2} & 0\\ 0 & \rho^{-2}\end{pmatrix}$$

$$= \begin{pmatrix}\rho^2 c+1 & -\sigma^2 b\\ -\rho^2 b & \sigma^2 a+1\end{pmatrix}$$

$$\mathbf{Z} = \mathbf{K}^{-1} - \mathbf{K}^{-1}\mathbf{H}^\top\mathbf{A}^{-1}\mathbf{H}\mathbf{K}^{-1} = \mathbf{K}^{-1} - q\mathbf{K}^{-1}\mathbf{H}^\top\mathbf{Q}\mathbf{H}\mathbf{K}^{-1}$$

$$= \mathbf{K}^{-1} - q\begin{pmatrix}\mathbf{K_{11}^{-1}} & \mathbf{0}\\ \mathbf{0} & \mathbf{K_{22}^{-1}}\end{pmatrix}\begin{pmatrix}\mathbf{H_1^\top}\\ \mathbf{H_2^\top}\end{pmatrix}\mathbf{Q}\begin{pmatrix}\mathbf{H_1} & \mathbf{H_2}\end{pmatrix}\begin{pmatrix}\mathbf{K_{11}^{-1}} & \mathbf{0}\\ \mathbf{0} & \mathbf{K_{22}^{-1}}\end{pmatrix}$$

$$= \mathbf{K}^{-1} - q\begin{pmatrix}\mathbf{K_{11}^{-1}H_1^\top}\\ \mathbf{K_{22}^{-1}H_2^\top}\end{pmatrix}\mathbf{Q}\begin{pmatrix}\mathbf{H_1K_{11}^{-1}} & \mathbf{H_2K_{22}^{-1}}\end{pmatrix}$$

$$= \begin{pmatrix}\mathbf{K_{11}^{-1}} & \mathbf{0}\\ \mathbf{0} & \mathbf{K_{22}^{-1}}\end{pmatrix} - \begin{pmatrix} q\mathbf{K_{11}^{-1}H_1^\top QH_1K_{11}^{-1}} & q\mathbf{K_{11}^{-1}H_1^\top QH_2K_{22}^{-1}}\\ q\mathbf{K_{22}^{-1}H_2^\top QH_1K_{11}^{-1}} & q\mathbf{K_{22}^{-1}H_2^\top QH_2K_{22}^{-1}}\end{pmatrix}$$

$$= \begin{pmatrix}\mathbf{K_{11}^{-1}} - q\mathbf{K_{11}^{-1}QK_{11}^{-1}} & -q\mathbf{W}^\top\mathbf{H_2K_{22}^{-1}}\\ -q\mathbf{K_{22}^{-1}H_2^\top W} & \mathbf{K_{22}^{-1}} - q\mathbf{K_{22}^{-1}H_2^\top QH_2K_{22}^{-1}}\end{pmatrix}$$

$$\mathbf{Z_{11}} = \mathbf{K_{11}^{-1}} - q\mathbf{K_{11}^{-1}QK_{11}^{-1}} = \mathbf{K_{11}^{-1}} - q\mathbf{K_{11}^{-1}W} = q\begin{pmatrix}Z_{11a} & Z_{11b}\\ Z_{11b} & Z_{11c}\end{pmatrix}$$

$$\mathbf{K_{11}^{-1}W} = \begin{pmatrix}\sigma^{-2} & 0\\ 0 & \rho^{-2}\end{pmatrix}\begin{pmatrix}\rho^2 c+1 & -\sigma^2 b\\ -\rho^2 b & \sigma^a+1\end{pmatrix}$$

$$= \begin{pmatrix}\sigma^{-2}(\rho^2 c+1) & -b\\ -b & \rho^{-2}(\sigma^2 a+1)\end{pmatrix}$$

$$
\begin{aligned}
\mathbf{K_{11}^{-1}} &= q\begin{pmatrix} K_{11a}^{-1} & 0 \\ 0 & K_{11c}^{-1} \end{pmatrix} \\
K_{11a}^{-1} &= \sigma^{-2}(\sigma^2 a+1)(\rho^2 c+1)-\rho^2 b^2 \\
K_{11c}^{-1} &= \rho^{-2}(\sigma^2 a+1)(\rho^2 c+1)-\sigma^2 b^2 \\
Z_{11a} &= \sigma^{-2}(\sigma^2 a+1)(\rho^2 c+1)-\rho^2 b^2-\sigma^{-2}(\rho^2 c+1) \\
&= \cancel{\sigma^{-2}}(\cancel{\sigma^2} a\cancel{+1}\cancel{-1})(\rho^2 c+1)-\rho^2 b^2 \\
&= a(\rho^2 c+1)-\rho^2 b^2 \\
Z_{11b} &= 0-b=b \\
Z_{11c} &= \rho^{-2}(\sigma^2 a+1)(\rho^2 c+1)-\sigma^2 b^2-\rho^{-2}(\sigma^2 a+1) \\
&= \cancel{\rho^{-2}}(\sigma^2 a+1)(\cancel{\rho^2} c\cancel{+1}\cancel{-1})-\sigma^2 b^2 \\
&= c(\sigma^2 a+1)-\sigma^2 b^2
\end{aligned}
$$

Quintic Spline: Given

$$
\mathbf{A} = \mathbf{H}\mathbf{K_y^{-1}}\mathbf{H}^\top = \begin{pmatrix} a+\sigma^{-2} & b & d \\ b & c+\rho^{-2} & e \\ d & e & f \end{pmatrix}
$$

$$
\begin{aligned}
\det(\mathbf{A}) = (a+\sigma^{-2})\underbrace{[(c+\rho^{-2})f-e^2]}_{\hat{a}} + b\underbrace{(de-bf)}_{b} \\
+\, d\underbrace{[be-(c+\rho^{-2})d]}_{\hat{d}}
\end{aligned}
$$

$$
\mathbf{A^{-1}} = \frac{1}{\det(\mathbf{A})}\begin{pmatrix} \hat{a} & \hat{b} & \hat{d} \\ \hat{b} & \hat{c} & \hat{e} \\ \hat{d} & \hat{e} & \hat{f} \end{pmatrix} = q\mathbf{Q}, \text{where}
$$

$$
\begin{aligned}
\hat{a} &= (c+\rho^{-2})f-e^2 & \hat{b} &= ed-bf \\
\hat{c} &= (a+\sigma^{-2})f-d^2 & \hat{d} &= be-(c+\rho^{-2})d \\
\hat{e} &= bd-e(a+\sigma^{-2}) & \hat{f} &= (a+\sigma^{-2})(c+\rho^{-2})-b^2
\end{aligned}
$$

By multiplying with $\sigma^2\rho^2$, the expression of q, $\mathbf{Q}$ can be obtained.

$$\begin{aligned}
&q^{-1} = \sigma^2\rho^2\det(\mathbf{A})\\
&= (\sigma^2 a+1)[(\rho^2 c+1)f-\rho^2 e^2] + \sigma^2\rho^2 b(de-bf)\\
&+\sigma^2 d[\rho^2 be-(\rho^2 c+1)d]\\
&Q_{11} = \sigma^2[(\rho^2 c+1)f-\rho^2 e^2] = \sigma^2\bar{a}\\
&Q_{12} = \sigma^2\rho^2(ed-bf) = \sigma^2\rho^2\bar{b}\\
&Q_{13} = \sigma^2[\rho^2 be-(\rho^2 c+1)d] = \sigma^2\bar{d}\\
&Q_{22} = \rho^2[(\sigma^2 a+1)f-\sigma^2 d^2] = \rho^2\bar{c}\\
&Q_{23} = \rho^2[\sigma^2 bd-e(\sigma^2 a+1)] = \rho^2\bar{e}\\
&Q_{33} = (\sigma^2 a+1)(\rho^2 c+1)-\sigma^2\rho^2 b^2 = \bar{f}\\
&q = [(\sigma^2 a+1)\bar{a}+\sigma^2\rho^2 b\bar{b}+\sigma^2 d\bar{d}]^{-1}\\
&\mathbf{Q} = \begin{pmatrix} \sigma^2\bar{a} & \sigma^2\rho^2\bar{b} & \sigma^2\bar{d}\\ \sigma^2\rho^2\bar{b} & \rho^2\bar{c} & \rho^2\bar{e}\\ \sigma^2\bar{d} & \rho^2\bar{e} & \bar{f}\end{pmatrix}\\
&\mathbf{W} = \mathbf{Q}\mathbf{H_1}\mathbf{K_{11}^{-1}} = \mathbf{Q}\begin{pmatrix}1&0\\0&1\\0&0\end{pmatrix}\begin{pmatrix}\sigma^{-2}&0\\0&\rho^{-2}\end{pmatrix}\\
&= \begin{pmatrix} \sigma^2\bar{a} & \sigma^2\rho^2\bar{b} & \sigma^2\bar{d}\\ \sigma^2\rho^2\bar{b} & \rho^2\bar{c} & \rho^2\bar{e}\\ \sigma^2\bar{d} & \rho^2\bar{e} & \bar{f}\end{pmatrix}\begin{pmatrix}\sigma^{-2}&0\\0&\rho^{-2}\\0&0\end{pmatrix} = \begin{pmatrix}\bar{a}&\sigma^2\bar{b}\\ \rho^2\bar{b}&\bar{c}\\ \bar{d}&\bar{e}\end{pmatrix}\\
&\mathbf{Z_{11}} = \mathbf{K_{11}^{-1}} - q\mathbf{K_{11}^{-1}}\mathbf{H_1^\top}\mathbf{Q}\mathbf{H_1}\mathbf{K_{11}^{-1}}\\
&= \mathbf{K_{11}^{-1}} - q\mathbf{K_{11}^{-1}}\mathbf{H_1^\top}\mathbf{W}\\
&= \mathbf{K_{11}^{-1}} - q\begin{pmatrix}\sigma^{-2}&0\\0&\rho^{-2}\end{pmatrix}\begin{pmatrix}1&0&0\\0&1&0\end{pmatrix}\mathbf{W}\\
&= \mathbf{K_{11}^{-1}} - q\begin{pmatrix}\sigma^{-2}&0&0\\0&\rho^{-2}&0\end{pmatrix}\begin{pmatrix}\bar{a}&\sigma^2\bar{b}\\ \rho^2\bar{b}&\bar{c}\\ \bar{d}&\bar{e}\end{pmatrix}\\
&= \mathbf{K_{11}^{-1}} - q\begin{pmatrix}\sigma^{-2}\bar{a}&\bar{b}\\ \bar{b}&\rho^{-2}\bar{c}\end{pmatrix}\\
&K_{11a}^{-1} = \sigma^{-2}[(\sigma^2 a+1)\bar{a}+\sigma^2\rho^2 b\bar{b}+\sigma^2 d\bar{d}]\\
&= \sigma^{-2}(\sigma^2 a+1)\bar{a}+\rho^2 b\bar{b}+d\bar{d}
\end{aligned}$$

$$
\begin{aligned}
&K^{-1}_{11c} = \rho^{-2}[(\sigma^2 a+1)\bar{a}+\sigma^2\rho^2 b\bar{b}+\sigma^2 d\bar{d}]\\
&= \rho^{-2}(\sigma^2 a+1)\bar{a}+\sigma^2 b\bar{b}+\rho^{-2}\sigma^2 d\bar{d}\\
&Z_{11a} = \sigma^{-2}(\sigma^2 a+1)\bar{a}+\rho^2 b\bar{b}+d\bar{d}-\sigma^{-2}\bar{a}\\
&= \cancel{\sigma^{-2}}(\cancel{\sigma^2} a\cancel{+1}\cancel{-1})\bar{a}+\rho^2 b\bar{b}+d\bar{d} = a\bar{a}+\rho^2 b\bar{b}+d\bar{d}\\
&Z_{11b} = -\bar{b}\\
&Z_{11c} = \rho^{-2}(\sigma^2 a+1)\bar{a}+\sigma^2 b\bar{b}+\rho^{-2}\sigma^2 d\bar{d}-\rho^{-2}\bar{c}\\
&= \rho^{-2}[(\sigma^2 a+1)[(\rho^2 c+1)f-\rho^2 e^2]+\sigma^2 d[\rho^2 be\\
&\qquad -(\rho^2 c+1)d]-(\sigma^2 a+1)f+\sigma^2 d^2]+\sigma^2 b\bar{b}\\
&= \rho^{-2}[(\sigma^2 a+1)[(\rho^2 c\cancel{+1})f-\rho^2 e^2\cancel{-f}]\\
&\qquad +\sigma^2 d[\rho^2 be-(\rho^2 c\cancel{+1})d\cancel{-d}]+\sigma^2 b\bar{b}\\
&= \cancel{\rho^{-2}}\cancel{\rho^2}[(\sigma^2 a+1)(cf-e^2)+\sigma^2 d(be-cd)]+\sigma^2 b\bar{b}\\
&= (\sigma^2 a+1)(cf-e^2)+\sigma^2 d(be-cd)+\sigma^2 b\bar{b}
\end{aligned}
$$

Notes: a,b and c are calculated by $\mathbf{H_2 K_{22}^{-1} H_2^\top}$ for cubic spline, and a,b,c,d,e, and f are for quintic spline.

To compute the mean and covariance of the predictive distribution, we need to further consider $\mathbf{K}_*$ and $\mathbf{H}_*$. It is important to note that $\mathbf{K}_*$ and $\mathbf{H}_*$ are not in the same form of $\mathbf{K}$ and $\mathbf{H}$.

For cubic spline, the feature vector $\mathbf{H}_*$ of the prediction location $x*$ is given by,

$$
\mathbf{H}_* = \begin{bmatrix} \mathbf{1}_{1\times N} \\ x_{1:T}^T \end{bmatrix}
$$

The forms are different because we would like to predict the objective values only, but not also the gradient values.

The covariance function $\mathbf{K}_*$ should be in the form of,

$$
\mathbf{K}_* = \mathbf{K_y}(\mathbf{x}*, \mathbf{X}) = \begin{bmatrix} \mathbf{K_{1*}} & \\ & \mathbf{K_{2*}} \end{bmatrix}
$$

where

$$
\mathbf{K_{1*}} = \begin{bmatrix} k \\ \partial_k \end{bmatrix}^\top_{(x_1^*, x_1)=(0,0)} ; [\mathbf{K_{1*}}] = [1 \times 2]
$$

,

$$\mathbf{K_{2*}} = \begin{bmatrix} k \\ \partial_k \end{bmatrix}^{\top}_{(x^*_{2:N}, x_{2:T})} ; [\mathbf{K_{1*}}] = [N-1 \times 2T-2]$$

and N is the number of prediction location.

For quintic spline, the feature vector $\mathbf{H}_*$ of the prediction location $x*$ is given by,

$$\mathbf{H}_* = \begin{bmatrix} \mathbf{1}_{1\times N} \\ x^T_{1:T} \\ (x^2_{1:T})^T \end{bmatrix}$$

Appendix D

Negative log marginal likelihood

The negative log marginal likelihood is given by,

$$-\log p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}\mathbf{y}^\top \mathbf{K}^{-\mathbf{1}}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{n}{2}\log 2\pi$$

Given that the covariance function which includes spline model, basis function and noise is denoted as $\mathbf{K} = \theta^2\mathbf{K_f} + \sigma^2\mathbf{F} + \mathbf{HBH}^\top = \theta^2\bar{\mathbf{K}}_\mathbf{y} + \mathbf{HBH}^\top = \mathbf{K_y} + \mathbf{HBH}^\top$, $\bar{\mathbf{K}}_\mathbf{y} = \mathbf{K_f} + \tau^2\mathbf{F}$, $\mathbf{Z} = \mathbf{K}^{-\mathbf{1}}$, and consider the non-informative limit gives $\mathbf{B} \to \infty$ and $\mathbf{B}^{-\mathbf{1}} \to 0$,

$$\begin{aligned}
\mathbf{Z} = \mathbf{K}^{-\mathbf{1}} &= (\mathbf{K_y} + \mathbf{HBH}^\top)^{-\mathbf{1}} \\
&= \mathbf{K_y^{-1}} - \mathbf{K_y^{-1}}\mathbf{H}^\top(\mathbf{B}^{-\mathbf{1}} + \mathbf{HK_y^{-1}}\mathbf{H}^\top)^{-\mathbf{1}}\mathbf{HK_y^{-1}} \\
&= \mathbf{K_y^{-1}} - \mathbf{K_y^{-1}}\mathbf{H}^\top(\mathbf{HK_y^{-1}}\mathbf{H}^\top)^{-\mathbf{1}}\mathbf{HK_y^{-1}} \\
&= \mathbf{K_y^{-1}} - \mathbf{K_y^{-1}}\mathbf{H}^\top\mathbf{A}^{-\mathbf{1}}\mathbf{HK_y^{-1}}
\end{aligned}$$

The second term can be rewritten as,

$$\begin{aligned}
\frac{1}{2}\log|\mathbf{K}| &= \frac{1}{2}\log|\mathbf{K_y} + \mathbf{HBH}^\top| \\
&= \frac{1}{2}\log|\mathbf{K_y}(\mathbf{I} + \mathbf{K_y^{-1}}\mathbf{HBH}^\top)| \\
&= \frac{1}{2}\log|\mathbf{K_y}(\mathbf{I} + \mathbf{K_y^{-1}}\mathbf{HBH}^\top)| \\
&\approx \frac{1}{2}\log|\mathbf{K_y}\mathbf{BHK_y^{-1}}\mathbf{H}^\top)| \\
&= \frac{1}{2}(\log|\mathbf{K_y}| + \log|\mathbf{B}| + \log|\mathbf{A}|)
\end{aligned}$$

For cubic spline, the determinant is given by,

$$|\mathbf{A}| = (a+\sigma^{-2})(c+\rho^{-2}) - b^2$$

$$|\mathbf{K_{11}}| = \sigma^2\rho^2$$

Consider $\rho \to 0$, it is still difficult to compute $\log|\mathbf{K_y}|$ and $\log|\mathbf{A}|$ individually, to deal with this, we can rewrite them as,

$$\begin{aligned}\log|\mathbf{K_y}| + \log|\mathbf{A}| &= \log|\mathbf{K_{22}}| + \log|\mathbf{A}||\mathbf{K_{11}}| \\ &= \log|\mathbf{K_{22}}| + \log[(a+\sigma^{-2})(c+\rho^{-2}) - b^2]\sigma^2\rho^2 \\ &= \log|\mathbf{K_{22}}| + \log[(\sigma^2 a+1)(\rho^2 c+1) - \rho^2\sigma^2 b^2]\end{aligned}$$

For the quintic spline, the determinant is given by,

$$\begin{aligned}|\mathbf{A}| \;=\; & (a+\sigma^{-2})\big((c+\rho^{-2})f - e^2\big) - b^2 f + 2bde \\ & -d^2(c+\rho^{-2})\end{aligned}$$

$$|\mathbf{K_{11}}| = \sigma^2\rho^2$$

Consider $\rho \to 0$, it is still difficult to compute $\log|\mathbf{K_y}|$ and $\log|\mathbf{A}|$ individually, to deal with this, we can rewrite them as,

$$\begin{aligned}\log|\mathbf{K_{11}A}| \;=\; & \log[(\sigma^2 a+1)((\rho^2 c+1)f - \rho^2 e^2) \\ & +\sigma^2\rho^2 b(2ed - bf) - \sigma^2 d^2(\rho^2 c+1)]\end{aligned}$$

and $\mathbf{Z} = (\theta^2\bar{\mathbf{K}}_\mathbf{y} + \mathbf{HBH}^\top)^{-1} = \frac{1}{\theta^2}\bar{\mathbf{Z}}_\mathbf{y}$,

$$\begin{aligned}\mathbf{Z} = \frac{\mathbf{1}}{\theta^2}\bar{\mathbf{Z}}_\mathbf{y} &= \mathbf{K_y^{-1}} - \mathbf{K_y^{-1}H}^\top\mathbf{A^{-1}HK_y^{-1}} \\ &= \frac{\mathbf{1}}{\theta^2}\bar{\mathbf{K}}_\mathbf{y}^{-1} - \frac{\mathbf{1}}{\theta^2}\bar{\mathbf{K}}_\mathbf{y}^{-1}\mathbf{H}^\top(\mathbf{H}\frac{\mathbf{1}}{\theta^2}\bar{\mathbf{K}}_\mathbf{y}^{-1}\mathbf{H}^\top)^{-1}\mathbf{H}\frac{\mathbf{1}}{\theta^2}\bar{\mathbf{K}}_\mathbf{y}^{-1} \\ &= \frac{\mathbf{1}}{\theta^2}\bar{\mathbf{K}}_\mathbf{y}^{-1} - \frac{\mathbf{1}}{\theta^2}\bar{\mathbf{K}}_\mathbf{y}^{-1}\mathbf{H}^\top(\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-1}\mathbf{H}^\top)^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-1}\end{aligned}$$

Therefore,

$$\bar{\mathbf{Z}}_\mathbf{y} = \bar{\mathbf{K}}_\mathbf{y}^{-1} - \bar{\mathbf{K}}_\mathbf{y}^{-1}\mathbf{H}^\top(\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-1}\mathbf{H}^\top)^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-1}$$

Appendix E

Hyperparameter restimation

The negative log marginal likelihood is approximately given by,

$$-\log p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}\mathbf{y}^\top \mathbf{K}^{-\mathbf{1}}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}|$$

Given that a random independent variable θ, $\frac{\partial}{\partial\theta}\ln|\mathbf{X}| = \mathrm{Tr}(\mathbf{X}^{-\mathbf{1}}\frac{\partial\mathbf{X}}{\partial\theta})$ and $\frac{\partial}{\partial\theta}\mathbf{X}^{-\mathbf{1}} = \mathbf{X}^{-\mathbf{1}}\frac{\partial\mathbf{X}}{\partial\theta}\mathbf{X}^{-\mathbf{1}}$,

$$\begin{aligned}
&-\frac{\partial}{\partial\theta}\log p(\mathbf{y}|\mathbf{X}) = \frac{\partial}{\partial\theta}(\frac{1}{2}\mathbf{y}^\top\mathbf{K}^{-\mathbf{1}}\mathbf{y} + \frac{1}{2}\ln|\mathbf{K}|) \\
&= -\frac{1}{2}\underbrace{\mathbf{y}^\top\mathbf{K}^{-\mathbf{1}}\frac{\partial\mathbf{K}}{\partial\theta}\mathbf{K}^{-\mathbf{1}}\mathbf{y}}_{scalar} + \frac{1}{2}\mathrm{Tr}(\mathbf{K}^{-\mathbf{1}}\frac{\partial\mathbf{K}}{\partial\theta}) = -\frac{1}{2}\mathrm{Tr}(\mathbf{y}^\top\mathbf{K}^{-\mathbf{1}}\frac{\partial\mathbf{K}}{\partial\theta}\mathbf{K}^{-\mathbf{1}}\mathbf{y}) + \frac{1}{2}\mathrm{Tr}(\mathbf{K}^{-\mathbf{1}}\frac{\partial\mathbf{K}}{\partial\theta}) \\
&= -\frac{1}{2}\mathrm{Tr}(\mathbf{K}^{-\mathbf{1}}\mathbf{y}\mathbf{y}^\top\mathbf{K}^{-\mathbf{1}}\frac{\partial\mathbf{K}}{\partial\theta}) + \frac{1}{2}\mathrm{Tr}(\mathbf{K}^{-\mathbf{1}}\frac{\partial\mathbf{K}}{\partial\theta}) = \frac{1}{2}\mathrm{Tr}((\mathbf{K}^{-\mathbf{1}} - \mathbf{K}^{-\mathbf{1}}\mathbf{y}\mathbf{y}^\top\mathbf{K}^{-\mathbf{1}})\frac{\partial\mathbf{K}}{\partial\theta}) \\
&= \frac{1}{2}\mathrm{Tr}(\mathbf{B}\frac{\partial\mathbf{K}}{\partial\theta})
\end{aligned}$$

where $\mathbf{B} = \mathbf{K}^{-\mathbf{1}} - \mathbf{K}^{-\mathbf{1}}\mathbf{y}\mathbf{y}^\top\mathbf{K}^{-\mathbf{1}} = \mathbf{Z} - \mathbf{z}\mathbf{z}^\top$ and $\mathbf{z} = \mathbf{Z}\mathbf{y}$. Therefore,

$$-\frac{\partial}{\partial\theta}\log p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}\mathrm{Tr}(\mathbf{B}\frac{\partial\mathbf{K}}{\partial\theta})$$

There are several hyperparameters in this optimization problem, i.e. signal variance θ^2, function value noise variance σ^2, derivative noise variance ρ^2, function noise to signal variance $\tau^2 = \frac{\sigma^2}{\theta^2}$ and derivative noise to signal variance $\zeta^2 = \frac{\rho^2}{\theta^2}$. To avoid all the hyperparameters to be complex and ensure they are positive, they can be re-defined in the log-scale,

$$\theta^2 = \exp 2\bar{\theta} \qquad \sigma^2 = \exp 2\bar{\sigma} \qquad \rho^2 = \exp 2\bar{\rho}$$
$$\tau^2 = \exp 2\bar{\tau} \qquad \zeta^2 = \exp 2\bar{\zeta}$$

Here are some useful expressions for later derivations,

$$\frac{\partial \mathbf{K}}{\partial \bar{\theta}} = \frac{\partial}{\partial \bar{\theta}}(\exp(2\bar{\theta})\bar{\mathbf{K}}_\mathrm{y} + \mathbf{HBH}^\top) = 2\exp(2\bar{\theta})\bar{\mathbf{K}}_\mathrm{y} = 2\theta^2\bar{\mathbf{K}}_\mathrm{y}$$
$$\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}} = \frac{\partial}{\partial \bar{\tau}}(\mathbf{K}_\mathrm{f} + \tau^2\mathbf{F}) = \frac{\partial}{\partial \bar{\tau}}(\mathbf{K}_\mathrm{f} + \exp(2\bar{\tau})\mathbf{F}) = 2\exp(2\bar{\tau})\mathbf{F} = 2\tau^2\mathbf{F}$$
$$\frac{\partial^2 \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}^2} = \frac{\partial}{\partial \bar{\tau}}(2\exp(2\bar{\tau})\mathbf{F}) = 4\exp(2\bar{\tau})\mathbf{F} = 4\tau^2\mathbf{F}$$
$$\frac{\partial \bar{\mathbf{K}}_\mathrm{y}^{-1}}{\partial \bar{\tau}} = -\bar{\mathbf{K}}_\mathrm{y}^{-1}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\bar{\mathbf{K}}_\mathrm{y}^{-1} = -2\tau^2\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{F}\bar{\mathbf{K}}_\mathrm{y}^{-1}$$
$$\frac{\partial \mathbf{K}}{\partial \bar{\tau}} = \frac{\partial}{\partial \bar{\tau}}(\hat{\theta}^2\bar{\mathbf{K}}_\mathrm{y}) = \frac{\partial \hat{\theta}^2}{\partial \bar{\tau}}\bar{\mathbf{K}}_\mathrm{y} + \hat{\theta}^2\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}$$
$$\frac{\partial^2 \mathbf{K}}{\partial \bar{\tau}^2} = 2\frac{\partial \hat{\theta}^2}{\partial \bar{\tau}}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}} + \frac{\partial^2 \hat{\theta}^2}{\partial \bar{\tau}^2}\bar{\mathbf{K}}_\mathrm{y} + \hat{\theta}^2\frac{\partial^2 \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}^2}$$
$$\frac{\partial \mathbf{Z}}{\partial \bar{\tau}} = \frac{\partial \mathbf{K}^{-1}}{\partial \bar{\tau}} = -\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\mathbf{K}^{-1} = -\mathbf{Z}\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\mathbf{Z}$$

Let $\mathbf{C} = \mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top$,

$$\begin{aligned}\frac{\partial}{\partial \bar{\tau}}\mathbf{C}^{-1} &= -\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \bar{\tau}}\mathbf{C}^{-1} = -\mathbf{C}^{-1}\mathbf{H}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}^{-1}}{\partial \bar{\tau}}\mathbf{H}^\top\mathbf{C}^{-1}\\ &= \mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\mathbf{H}^\top\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{C}^{-1} = 2\tau^2\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{F}\mathbf{H}^\top\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{C}^{-1}\end{aligned}$$

To prove $\frac{\partial \bar{\mathbf{Z}}_\mathrm{y}}{\partial \bar{\tau}} = -\bar{\mathbf{Z}}_\mathrm{y}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\bar{\mathbf{Z}}_\mathrm{y} = -2\tau^2\bar{\mathbf{Z}}_\mathrm{y}\mathbf{F}\bar{\mathbf{Z}}_\mathrm{y}$,

$$\begin{aligned}\frac{\partial \bar{\mathbf{Z}}_\mathrm{y}}{\partial \bar{\tau}} &= \frac{\partial}{\partial \bar{\tau}}(\bar{\mathbf{K}}_\mathrm{y}^{-1} - \bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top(\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top)^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1}) = \frac{\partial}{\partial \bar{\tau}}(\bar{\mathbf{K}}_\mathrm{y}^{-1} - \bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1})\\
&= \frac{\partial}{\partial \bar{\tau}}\bar{\mathbf{K}}_\mathrm{y}^{-1} - \frac{\partial}{\partial \bar{\tau}}(\bar{\mathbf{K}}_\mathrm{y}^{-1})\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1} - \bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top\frac{\partial \mathbf{C}^{-1}}{\partial \bar{\tau}}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1} - \bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\frac{\partial}{\partial \bar{\tau}}\bar{\mathbf{K}}_\mathrm{y}^{-1}\\
&= -2\tau^2\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{F}\bar{\mathbf{K}}_\mathrm{y}^{-1} - 2\tau^2\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{F}\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1}\\
&\qquad - 2\tau^2\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{F}\bar{\mathbf{K}}_\mathrm{y}^{-1} - 2\tau^2\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{F}\mathbf{H}^\top\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1}\\
&= -2\tau^2(\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{F}(\bar{\mathbf{K}}_\mathrm{y}^{-1} - \bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1}) - \bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{F}(\bar{\mathbf{K}}_\mathrm{y}^{-1} - \bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1}))\\
&= -2\tau^2(\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{F}\bar{\mathbf{Z}}_\mathrm{y} - \bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{F}\bar{\mathbf{Z}}_\mathrm{y}) = -2\tau^2(\bar{\mathbf{K}}_\mathrm{y}^{-1} - \bar{\mathbf{K}}_\mathrm{y}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_\mathrm{y}^{-1})\mathbf{F}\bar{\mathbf{Z}}_\mathrm{y} = R.H.S\end{aligned}$$

A closed form solution in terms of τ^2 and ζ^2 can be defined for θ^2.

$$
\begin{aligned}
&-\frac{\partial}{\partial\bar{\theta}}\log p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}\mathrm{Tr}((\mathbf{Z}-\mathbf{Z}\mathbf{y}\mathbf{y}^\top\mathbf{Z})\frac{\partial\mathbf{K}}{\partial\bar{\theta}}) = \frac{1}{2}\mathrm{Tr}((\frac{\mathbf{1}}{\theta^2}\bar{\mathbf{Z}}_\mathrm{y} - \frac{\mathbf{1}}{\theta^4}\bar{\mathbf{Z}}_\mathrm{y}\mathbf{y}\mathbf{y}^\top\bar{\mathbf{Z}}_\mathrm{y})2\theta^2\bar{\mathbf{K}}_\mathrm{y})\\
&= \mathrm{Tr}((\bar{\mathbf{Z}}_\mathrm{y} - \frac{\mathbf{1}}{\theta^2}\bar{\mathbf{Z}}_\mathrm{y}\mathbf{y}\mathbf{y}^\top\bar{\mathbf{Z}}_\mathrm{y})\bar{\mathbf{K}}_\mathrm{y}) = \mathrm{Tr}\,(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y} - \frac{\mathbf{1}}{\theta^2}\bar{\mathbf{Z}}_\mathrm{y}\mathbf{y}\mathbf{y}^\top\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})\\
&= \mathrm{Tr}\,(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y} - \underbrace{\frac{1}{\theta^2}\mathbf{y}^\top\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y}\bar{\mathbf{Z}}_\mathrm{y}\mathbf{y}}_{\textbf{scalar}}) = 0
\end{aligned}
$$

Therefore, $\theta^2 = \frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})}\mathbf{y}^\top\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y}\bar{\mathbf{Z}}_\mathrm{y}\mathbf{y}$.
Further consider the form of $\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y}\bar{\mathbf{Z}}_\mathrm{y}$,

$$
\begin{aligned}
&(\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}} - \bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top(\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top)^{-\mathbf{1}}\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}})\bar{\mathbf{K}}_\mathrm{y}\bar{\mathbf{Z}}_\mathrm{y} = (\mathbf{I} - \bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top(\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top)^{-\mathbf{1}}\mathbf{H})\bar{\mathbf{Z}}_\mathrm{y}\\
&= \bar{\mathbf{Z}}_\mathbf{y} - \bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top(\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top)^{-\mathbf{1}}\mathbf{H}\bar{\mathbf{Z}}_\mathbf{y}\\
&= \bar{\mathbf{Z}}_\mathbf{y} - \bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top(\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top)^{-\mathbf{1}}\mathbf{H}(\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}} - \bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top(\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top)^{-\mathbf{1}}\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}})\\
&= \bar{\mathbf{Z}}_\mathbf{y} - \bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top(\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top)^{-\mathbf{1}}\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\\
&\quad + \bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top\cancel{(\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top)^{-\mathbf{1}}\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top}(\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top)^{-\mathbf{1}}\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}})\\
&= \bar{\mathbf{Z}}_\mathbf{y} - \cancel{\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top(\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top)^{-\mathbf{1}}\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}} + \cancel{\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top(\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}\mathbf{H}^\top)^{-\mathbf{1}}\mathbf{H}\bar{\mathbf{K}}_\mathbf{y}^{-\mathbf{1}}} = \bar{\mathbf{Z}}_\mathbf{y}
\end{aligned}
$$

Finally,

$$\hat{\theta}^2 = \frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})}\mathbf{y}^\top\bar{\mathbf{Z}}_\mathrm{y}\mathbf{y}$$

Newton Optimization is utilized to optimize the log-scale of τ^2 so the gradient $g = -\frac{\partial}{\partial\bar{\tau}}\log p(\mathbf{y}|\mathbf{X})$ and the Hessian $H = -\frac{\partial^2}{\partial\bar{\tau}^2}\log p(\mathbf{y}|\mathbf{X})$ are required to improve the hyperparameter iteratively,

$$\bar{\tau}[t+1] = \bar{\tau}[t] - H^{-1}g$$

As derived before, the gradient is given by,

$$-\frac{\partial}{\partial\bar{\tau}}\log p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}\mathrm{Tr}(\mathbf{B}\frac{\partial\mathbf{K}}{\partial\bar{\tau}})$$

The derivative $\frac{\partial\mathbf{K}}{\partial\bar{\tau}}$ is given by,

$$\frac{\partial\mathbf{K}}{\partial\bar{\tau}} = \frac{\partial}{\partial\bar{\tau}}(\hat{\theta}^2\bar{\mathbf{K}}_\mathrm{y} + \mathbf{H}\mathbf{B}\mathbf{H}^\top) = \frac{\partial}{\partial\bar{\tau}}(\hat{\theta}^2\bar{\mathbf{K}}_\mathrm{y}) = \frac{\partial}{\partial\bar{\tau}}(\frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})}\mathbf{y}^\top\bar{\mathbf{Z}}_\mathrm{y}\mathbf{y}\bar{\mathbf{K}}_\mathrm{y})$$

According to the product rule of differentiation, there are three derivatives required, i.e. $\frac{\partial}{\partial\bar{\tau}}(\frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}})})$, $\frac{\partial}{\partial\bar{\tau}}\bar{\mathbf{Z}}_{\mathrm{y}}$ and $\frac{\partial}{\partial\bar{\tau}}\bar{\mathbf{K}}_{\mathrm{y}}$, to give the whole proper expression.
Firstly,

$$\frac{\partial}{\partial\bar{\tau}}(\frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}})}) = -\frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}})^2}\mathrm{Tr}(\frac{\partial}{\partial\bar{\tau}}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}}))$$

Consider,

$$\begin{aligned}&\frac{\partial}{\partial\bar{\tau}}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}}) = \frac{\partial}{\partial\bar{\tau}}\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}} + \bar{\mathbf{Z}}_{\mathrm{y}}\frac{\partial}{\partial\bar{\tau}}\bar{\mathbf{K}}_{\mathrm{y}} = -2\tau^2\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{F}\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}} + 2\tau^2\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{F}\\ &= 2\tau^2\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{F}(\mathbf{I}-\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}}) = 2\tau^2\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{F}(\mathbf{I}-\mathbf{I}+\bar{\mathbf{K}}_{\mathrm{y}}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}) = 2\tau^2\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{F}\bar{\mathbf{K}}_{\mathrm{y}}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\end{aligned}$$

so,

$$\begin{aligned}&\mathrm{Tr}(\frac{\partial}{\partial\bar{\tau}}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}})) = 2\tau^2\,\mathrm{Tr}(\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{F}\bar{\mathbf{K}}_{\mathrm{y}}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}) = 2\tau^2\,\mathrm{Tr}(\bar{\mathbf{K}}_{\mathrm{y}}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{F})\\ &= 2\tau^2\,\mathrm{Tr}(\bar{\mathbf{K}}_{\mathrm{y}}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}(\bar{\mathbf{K}}_{\mathrm{y}}^{-1}-\bar{\mathbf{K}}_{\mathrm{y}}^{-1}\mathbf{H}^\top\mathbf{C}^{-1}\mathbf{H}\bar{\mathbf{K}}_{\mathrm{y}}^{-1})\mathbf{F}) = 0\end{aligned}$$

Then,

$$\begin{aligned}\frac{\partial\mathbf{K}}{\partial\bar{\tau}} &= \frac{\partial}{\partial\bar{\tau}}(\frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}})}\mathbf{y}^\top\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{y}\bar{\mathbf{K}}_{\mathrm{y}})\\ &= \frac{\partial}{\partial\bar{\tau}}(\frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}})})\mathbf{y}^\top\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{y}\bar{\mathbf{K}}_{\mathrm{y}} + \frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}})}\mathbf{y}^\top\frac{\partial\bar{\mathbf{Z}}_{\mathrm{y}}}{\partial\bar{\tau}}\mathbf{y}\bar{\mathbf{K}}_{\mathrm{y}} + \frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}})}\mathbf{y}^\top\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{y}\frac{\partial\bar{\mathbf{K}}_{\mathrm{y}}}{\partial\bar{\tau}}\\ &= \frac{2\tau^2}{\mathrm{Tr}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}})}\mathbf{y}^\top(-\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{F}\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{y}\bar{\mathbf{K}}_{\mathrm{y}} + \bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{y}\mathbf{F}) = \frac{2\tau^2}{\mathrm{Tr}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}})}(\mathbf{y}^\top\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{y}\mathbf{F} - \mathbf{y}^\top\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{F}\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{y}\bar{\mathbf{K}}_{\mathrm{y}})\\ &= 2\tau^2\hat{\theta}^2\mathbf{F} - \frac{2\tau^2}{\mathrm{Tr}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}})}\mathbf{y}^\top\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{F}\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{y}\bar{\mathbf{K}}_{\mathrm{y}}\end{aligned}$$

Finally,

$$-\frac{\partial}{\partial\bar{\tau}}\log p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}\mathrm{Tr}(\mathbf{B}\frac{\partial\mathbf{K}}{\partial\bar{\tau}}) = \hat{\theta}^2\tau^2\,\mathrm{Tr}(\mathbf{B}\mathbf{F}) - \frac{\tau^2}{\mathrm{Tr}(\bar{\mathbf{Z}}_{\mathrm{y}}\bar{\mathbf{K}}_{\mathrm{y}})}\mathbf{y}^\top\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{F}\bar{\mathbf{Z}}_{\mathrm{y}}\mathbf{y}\,\mathrm{Tr}(\mathbf{B}\bar{\mathbf{K}}_{\mathrm{y}})$$

For the Hessian,

$$-\frac{\partial^2}{\partial\bar{\tau}^2}\log p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}\frac{\partial}{\partial\bar{\tau}}\mathrm{Tr}(\mathbf{B}\frac{\partial\mathbf{K}}{\partial\bar{\tau}}) = \frac{1}{2}\mathrm{Tr}\frac{\partial}{\partial\bar{\tau}}(\mathbf{B}\frac{\partial\mathbf{K}}{\partial\bar{\tau}}) = \frac{1}{2}\mathrm{Tr}(\frac{\partial\mathbf{B}}{\partial\bar{\tau}}\frac{\partial\mathbf{K}}{\partial\bar{\tau}} + \mathbf{B}\frac{\partial^2\mathbf{K}}{\partial\bar{\tau}^2})$$

Consider,

$$\begin{aligned}\frac{\partial \mathbf{B}}{\partial \bar{\tau}} &= \frac{\partial}{\partial \bar{\tau}}(\mathbf{Z}-\mathbf{Z}\mathbf{y}\mathbf{y}^\top\mathbf{Z}) = \frac{\partial \mathbf{Z}}{\partial \bar{\tau}} - \frac{\partial \mathbf{Z}}{\partial \bar{\tau}}\mathbf{y}\mathbf{y}^\top\mathbf{Z} - \mathbf{Z}\mathbf{y}\mathbf{y}^\top\frac{\partial \mathbf{Z}}{\partial \bar{\tau}} \\ &= \mathbf{Z}\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\mathbf{Z} + \mathbf{Z}\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\mathbf{Z}\mathbf{y}\mathbf{y}^\top\mathbf{Z} + \mathbf{Z}\mathbf{y}\mathbf{y}^\top\mathbf{Z}\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\mathbf{Z}\end{aligned}$$

So the first trace part can be re-written as,

$$\begin{aligned}&\frac{1}{2}\mathrm{Tr}\left(\frac{\partial \mathbf{B}}{\partial \bar{\tau}}\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\right) = \frac{1}{2}\mathrm{Tr}\left(\left(-\mathbf{Z}\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\mathbf{Z} + \mathbf{Z}\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\mathbf{z}\mathbf{z}^\top + \mathbf{z}\mathbf{z}^\top\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\mathbf{Z}\right)\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\right) \\ &= \frac{1}{2}\mathrm{Tr}\left(-\mathbf{Z}\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\mathbf{Z}\frac{\partial \mathbf{K}}{\partial \bar{\tau}} + 2\mathbf{Z}\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\mathbf{z}\mathbf{z}^\top\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\right) = \frac{1}{2}\mathrm{Tr}\,\mathbf{Z}\frac{\partial \mathbf{K}}{\partial \bar{\tau}}(-\mathbf{Z}+2\mathbf{z}\mathbf{z}^\top)\frac{\partial \mathbf{K}}{\partial \bar{\tau}} \\ &= -\frac{1}{2}\mathrm{Tr}\,\mathbf{Z}\frac{\partial \mathbf{K}}{\partial \bar{\tau}}(\mathbf{Z}-2\mathbf{z}\mathbf{z}^\top)\frac{\partial \mathbf{K}}{\partial \bar{\tau}} = -\frac{1}{2}\mathrm{Tr}\,\mathbf{Z}\frac{\partial \mathbf{K}}{\partial \bar{\tau}}(\mathbf{B}-\mathbf{z}\mathbf{z}^\top)\frac{\partial \mathbf{K}}{\partial \bar{\tau}}\end{aligned}$$

The second trace part is given by,

$$\frac{1}{2}\mathrm{Tr}\left(\mathbf{B}\frac{\partial^2 \mathbf{K}}{\partial \bar{\tau}^2}\right) = \frac{1}{2}\mathrm{Tr}\left(\mathbf{B}\left(2\frac{\partial \hat{\theta}^2}{\partial \bar{\tau}}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}} + \frac{\partial^2 \hat{\theta}^2}{\partial \bar{\tau}^2}\bar{\mathbf{K}}_\mathrm{y} + \hat{\theta}^2\frac{\partial^2 \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}^2}\right)\right)$$

Consider,

$$\begin{aligned}&\frac{\partial \hat{\theta}^2}{\partial \bar{\tau}} = \frac{\partial}{\partial \bar{\tau}}\left(\frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})}\mathbf{y}^\top\bar{\mathbf{Z}}_\mathrm{y}\mathbf{y}\right) = \cancel{\frac{\partial}{\partial \bar{\tau}}\left(\frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})}\right)}\mathbf{y}^\top\bar{\mathbf{Z}}_\mathrm{y}\mathbf{y} + \frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})}\mathbf{y}^\top\frac{\partial \bar{\mathbf{Z}}_\mathrm{y}}{\partial \bar{\tau}}\mathbf{y} \\ &= \frac{-1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})}\mathbf{y}^\top\bar{\mathbf{Z}}_\mathrm{y}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\bar{\mathbf{Z}}_\mathrm{y}\mathbf{y} \\ &\frac{\partial^2 \hat{\theta}^2}{\partial \bar{\tau}^2} = \frac{\partial}{\partial \bar{\tau}}\left(\frac{-1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})}\mathbf{y}^\top\bar{\mathbf{Z}}_\mathrm{y}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\bar{\mathbf{Z}}_\mathrm{y}\mathbf{y}\right) \\ &= \cancel{\frac{\partial}{\partial \bar{\tau}}\left(\frac{-1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})}\right)}\mathbf{y}^\top\bar{\mathbf{Z}}_\mathrm{y}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\bar{\mathbf{Z}}_\mathrm{y}\mathbf{y} + \frac{-1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})}\mathbf{y}^\top\frac{\partial}{\partial \bar{\tau}}\left(\bar{\mathbf{Z}}_\mathrm{y}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\bar{\mathbf{Z}}_\mathrm{y}\right)\mathbf{y} \\ &= \frac{-1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})}\mathbf{y}^\top\left(\frac{\partial \bar{\mathbf{Z}}_\mathrm{y}}{\partial \bar{\tau}}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\bar{\mathbf{Z}}_\mathrm{y} + \bar{\mathbf{Z}}_\mathrm{y}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\frac{\partial \bar{\mathbf{Z}}_\mathrm{y}}{\partial \bar{\tau}} + \bar{\mathbf{Z}}_\mathrm{y}\frac{\partial^2 \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}^2}\bar{\mathbf{Z}}_\mathrm{y}\right)\mathbf{y} \\ &= \frac{-1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})}\mathbf{y}^\top\left(-\bar{\mathbf{Z}}_\mathrm{y}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\bar{\mathbf{Z}}_\mathrm{y}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\bar{\mathbf{Z}}_\mathrm{y} - \bar{\mathbf{Z}}_\mathrm{y}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\bar{\mathbf{Z}}_\mathrm{y}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\bar{\mathbf{Z}}_\mathrm{y} + \bar{\mathbf{Z}}_\mathrm{y}\frac{\partial^2 \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}^2}\bar{\mathbf{Z}}_\mathrm{y}\right)\mathbf{y} \\ &= \frac{1}{\mathrm{Tr}(\bar{\mathbf{Z}}_\mathrm{y}\bar{\mathbf{K}}_\mathrm{y})}\mathbf{y}^\top\bar{\mathbf{Z}}_\mathrm{y}\left(2\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}}\bar{\mathbf{Z}}_\mathrm{y}\frac{\partial \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}} - \frac{\partial^2 \bar{\mathbf{K}}_\mathrm{y}}{\partial \bar{\tau}^2}\right)\bar{\mathbf{Z}}_\mathrm{y}\mathbf{y}\end{aligned}$$

Finally,

$$
\begin{aligned}
\frac{1}{2}\operatorname{Tr}(\mathbf{B}\frac{\partial^{\mathbf{2}}\mathbf{K}}{\partial\bar{\tau}^{\mathbf{2}}}) &= \frac{1}{2}\operatorname{Tr}(\mathbf{B}(2\frac{\partial\hat{\theta}^{\mathbf{2}}}{\partial\bar{\tau}}\frac{\partial\bar{\mathbf{K}}_{\mathbf{y}}}{\partial\bar{\tau}} + \frac{\partial^{\mathbf{2}}\hat{\theta}^{\mathbf{2}}}{\partial\bar{\tau}^{\mathbf{2}}}\bar{\mathbf{K}}_{\mathbf{y}} + \hat{\theta}^{\mathbf{2}}\frac{\partial^{\mathbf{2}}\bar{\mathbf{K}}_{\mathbf{y}}}{\partial\bar{\tau}^{\mathbf{2}}}))\\
&= 2\tau^2\frac{\partial\hat{\theta}^2}{\partial\bar{\tau}}\operatorname{Tr}(\mathbf{BF}) + \frac{1}{2}\frac{\partial^2\hat{\theta}^2}{\partial\bar{\tau}^2}\operatorname{Tr}(\mathbf{B}\bar{\mathbf{K}}_{\mathbf{y}}) + 2\hat{\theta}^2\tau^2\operatorname{Tr}(\mathbf{BF})\\
&= 2\tau^2(\hat{\theta}^2 + \frac{\partial\hat{\theta}^2}{\partial\bar{\tau}})\operatorname{Tr}(\mathbf{BF}) + \frac{1}{2}\frac{\partial^2\hat{\theta}^2}{\partial\bar{\tau}^2}\operatorname{Tr}(\mathbf{B}\bar{\mathbf{K}}_{\mathbf{y}})
\end{aligned}
$$