

Article

# Gradient Profile Estimation Using Exponential Cubic Spline Smoothing in a Bayesian Framework

Kushani De Silva <sup>1,\*</sup>, Carlo Cafaro <sup>2</sup> and Adom Giffin <sup>3,†</sup>

<sup>1</sup> Department of Mathematics, Iowa State University, Ames, IA 50011, USA

<sup>2</sup> Department of Mathematics and Physics, SUNY Polytechnic Institute, Albany, NY 12203, USA; carlocafaro2000@yahoo.it

<sup>3</sup> Naval Nuclear Laboratory, Schenectady, NY 12309, USA; physics101@gmail.com

\* Correspondence: kushani.desilva0314@gmail.com

† These authors contributed equally to this work.

**Abstract:** Attaining reliable gradient profiles is of utmost relevance for many physical systems. In many situations, the estimation of the gradient is inaccurate due to noise. It is common practice to first estimate the underlying system and then compute the gradient profile by taking the subsequent analytic derivative of the estimated system. The underlying system is often estimated by fitting or smoothing the data using other techniques. Taking the subsequent analytic derivative of an estimated function can be ill-posed. This becomes worse as the noise in the system increases. As a result, the uncertainty generated in the gradient estimate increases. In this paper, a theoretical framework for a method to estimate the gradient profile of discrete noisy data is presented. The method was developed within a Bayesian framework. Comprehensive numerical experiments were conducted on synthetic data at different levels of noise. The accuracy of the proposed method was quantified. Our findings suggest that the proposed gradient profile estimation method outperforms the state-of-the-art methods.

**Keywords:** computational techniques; inference methods; probability theory



**Citation:** De Silva, K.; Cafaro, C.; Giffin, A. Gradient Profile Estimation Using Exponential Cubic Spline Smoothing in a Bayesian Framework. *Entropy* **2021**, *23*, 674. <https://doi.org/10.3390/e23060674>

Academic Editor: Antonio M. Scarfone

Received: 25 March 2021

Accepted: 26 May 2021

Published: 27 May 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Estimating the gradient of a system from a discrete set of data has a vast number of applications in many fields, such as biology, engineering, and physics. For instance, determining the particle velocity from the discrete time-position data in particle image velocimetry and particle tracking velocimetry experiments is important in plasma physics [1–4]. Applications of velocity estimation in motion control systems using discrete-time data have increased with the invention of microprocessors (see [5], and the references therein). Moreover, with improved technology, faster equipment is now available to measure high-speed discrete data [1]. The state-of-the-art method for determining the gradient from data is to estimate an underlying smoothing function of the data and to take its subsequent analytic derivative [6–13] (similar approaches with Bayesian methods are used in [14–19]). Finite differencing is also used for this purpose [2]. However, when there is noise in the data, the results can be ill-posed because the derivative tends to blow up the uncertainty in the estimates [5]. The estimates become worse when these data are measured in shorter intervals, especially when position data are measured using high-speed cameras in particle tracking experiments [1,2].

On the other hand, the cubic splines used in these methods of estimating the data can produce drastic results depending on the recorded speed of data or noise levels. Exponential cubic splines are superior to cubic splines because of their capacity to capture abrupt changes in the data due to its extra parameter of smoothness [20,21]. However, special attention must be paid to this extra smoothing parameter because its extreme values

can produce unrealistic results. In the literature, Jeffreys prior was used for this smoothing parameter as a means of scaling down [1,20,21].

In our study, motivated by the work in [1], we present a detailed investigation on gradient profile estimation as a Bayesian inference problem by directly estimating the gradient, which avoids taking the analytic derivative of noisy data. Moreover, we use exponential cubic spline as the underlying smoothing function of the gradient. We also introduce a more meaningful choice of prior distribution for the smoothing parameter of this spline. Moreover, we present a comprehensive sensitivity analysis of noise on the gradient estimates and, additionally, we present the estimates for position and acceleration obtained by the subsequent integration and derivative of our gradient estimates.

The paper is organized as follows. Section 2 develops the idea of separating spaces with mathematical background. The Bayesian framework of the algorithm is demonstrated in Section 3. Thereafter, the computational details are presented in Sections 4 and 5. In particular, in these sections, our findings are compared to those obtained by means of more traditional methods, such as smoothing data with the subsequent analytic derivative. Our concluding remarks are given in Section 6.

## 2. Separating Spaces in Bayesian Context

In this section, we explain how the spaces are separated, which permits one to directly infer the gradient along with its mathematical forms. From this point forward, we use the terms (and mathematical notations)  $x$ -space ( $\mathbb{U}$ ) and  $v$ -space ( $\mathbb{V}$ ) to denote the space where measured data lives and the space where the gradient resides, respectively. We then use Bayes' theorem to map the information between these two spaces (examples are given in [22–28]). The mapping is done using the obvious relationship between an object's position and its velocity,

$$x(t) \stackrel{\text{def}}{=} \int v(t) dt, \quad (1)$$

where  $x(t)$  and  $v(t)$  represent the object's position and velocity, respectively. By the use of our notations,  $x(t)$  lives in  $x$ -space and  $v(t)$  lives in  $v$ -space. In our approach, a functional form (mathematical model) is given to velocity, rather than to measured data as in the traditional approach. Therefore, our mapping is such that the positions in  $x$ -space are 'obtained' by integrating the functional form placed in  $v$ -space. The Bayes' theorem allows us to do this as an inference problem to infer the unknowns of the functional form in  $v$ -space with the information available in  $x$ -space.

The exponential cubic spline is proven to capture abrupt changes due to its high flexibility compared to the cubic spline [1,20,21]; thus we use the exponential cubic spline (denoted by  $S_v$ ) as the functional form to represent the velocity, i.e., the exponential cubic spline lives in  $v$ -space (shown by the subscript  $v$ ). This spline is the solution to the following minimal bending energy functional [1]:

$$\int_{t_1}^{t_n} \left\{ \left[ \frac{d^2 S_v(t)}{dt^2} \right]^2 + \lambda(t)^2 \left[ \frac{dS_v(t)}{dt} \right]^2 \right\} dt \quad \text{for } \xi_{v_i} \leq t \leq \xi_{v_{i+1}}. \quad (2)$$

for  $S_v(\xi_{v_i}) = f_{v_i}$ ,  $i = 1, \dots, E_v$  and  $\lambda(t) = \lambda_{v_i}$  for  $\xi_i \leq t \leq \xi_{i+1}$ , where  $t_1 = \xi_{v_1}$  and  $t_n = \xi_{v_{E_v}}$ . Here,  $f_{v_i}$  is the function value at  $\xi_{v_i}$  and  $\lambda_{v_i}$  is proportional to tension between two knots,  $\xi_{v_i}, \xi_{v_{i+1}}$ . A numerically convenient reformulation of the spline at the  $i$ -th interval is then given by [21,29]

$$S_{v_i}(t, f_v, \lambda_v, \xi_v, E_v) \stackrel{\text{def}}{=} f_{v_i}(1 - h_v) + f_{v_{i+1}} h_v + \frac{M_{v_i}}{\lambda_{v_i}^2} \left\{ \frac{\sinh[\mu_{v_i}(1 - h_v)]}{\sinh(\mu_{v_i})} - (1 - h_v) \right\} + \frac{M_{v_{i+1}}}{\lambda_{v_i}^2} \left[ \frac{\sinh(\mu_{v_i} h_v)}{\sinh(\mu_{v_i})} - h_v \right], \quad (3)$$

for  $i = 1, \dots, E_v$ . The quantities  $h_v, \mu_{v_i}$ , and  $M_{v_i}$  in Equation (3) are defined as

$$h_v \stackrel{\text{def}}{=} \frac{t - \xi_{v_i}}{\xi_{v_{i+1}} - \xi_{v_i}}, \mu_{v_i} \stackrel{\text{def}}{=} \lambda_{v_i} (\xi_{v_{i+1}} - \xi_{v_i}), \text{ and} \tag{4}$$

$$M_{v_i} \stackrel{\text{def}}{=} \left\{ \frac{d^2}{dt^2} [S_{v_i}(t, f_v, \lambda_v, \xi_v, E_v)] \right\}_{t=\xi_{v_i}},$$

respectively. Using the arrow notations for vectorial quantities, the exponential cubic spline in Equation (3) can also be viewed in matrix form as [1]

$$S_v(\vec{t}) = W(\vec{t}, \vec{\lambda}_v, \vec{\xi}_v) \vec{f}_v. \tag{5}$$

In Equation (5),  $W$  denotes the design matrix of  $\vec{t}$ -locations of the function values  $\vec{f}_v$ , the support points (knots)  $\vec{\xi}_v$ , and tension parameters  $\vec{\lambda}_v$ . The second derivative of  $S_v, M_v$ , can be found explicitly by solving the tri-diagonal system of equations in Equation (5).

We used the exponential cubic spline to model the gradient. Since the gradient is the unknown quantity here, the variables,  $\vec{f}_v, \vec{\lambda}_v, \vec{\xi}_v, E_v$  in Equation (3) are unknown. Therefore, it is important at this stage to identify data and parameters in this inference problem and they are given in Table 1.

**Table 1.** The definition of data and parameters of the inference problem. The spaces where they reside are given in parentheses.

Data ( $x$ -space)	$\vec{t}, \vec{x}$
Parameters ( $v$ -space)	$\vec{f}_v, \vec{\lambda}_v, \vec{\xi}_v, E_v$

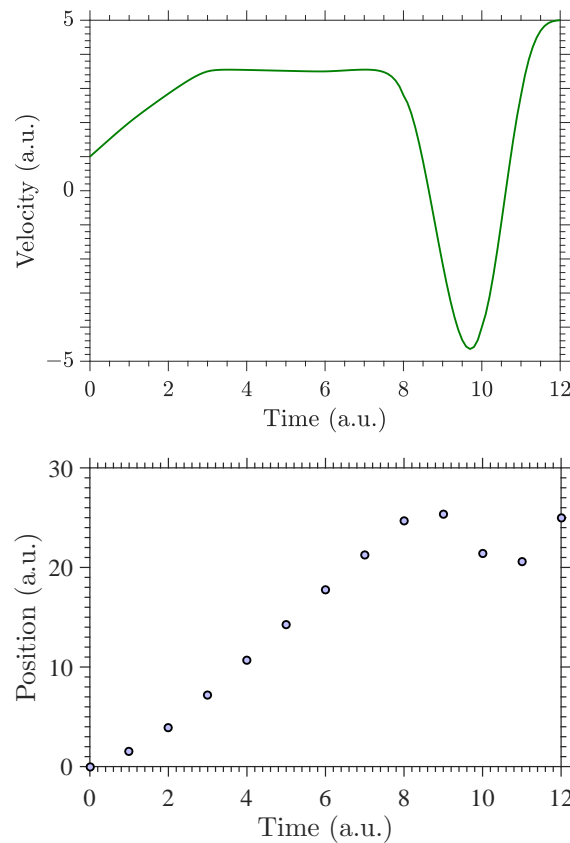
Now, the general relationship given in Equation (1) can be rewritten precisely using to our model for an arbitrary  $i$ -th data point ( $i = 1, \dots, n$  where  $n$  is the total number of measured data values) as

$$x_i = \int_{t_1}^{t_i} S_v(t, f_v, \lambda_v, \xi_v, E_v) dt, \tag{6}$$

with  $S_v(t, f_v, \lambda_v, \xi_v, E_v)$  given in Equation (3). That is,  $i$ -th position ( $i$ -th data point) is the integrated spline from first time point (time of the initial position) to the  $i$ -th time point (time of the  $i$ -th position). The relationship in Equation (6) can be further reduced to

$$x_i = x_{i-1} + \int_{t_{i-1}}^{t_i} S_{v_{i-1}}(t, f_v, \lambda_v, \xi_v, E_v) dt, \quad 2 < i < n. \tag{7}$$

The starting and ending values of  $x_i$  in Equation (7) are the initial and the last positions recorded. Here,  $S_{v_{i-1}}$  is the spline in  $(i - 1)^{th}$  interval. We point out that the exponential cubic spline is analytically integrable and its technical details are given in Appendix A. When a relationship (as in (1) and (7)) is used to match an unknown quantity (gradient/velocity) and the observable data (position), the relationship should be able to generate data as close as possible to the observable information if the desired/unknown quantity is known. This is called the *forward problem* in Bayesian language. However, what is required here in our study is to be able to solve the *inverse problem*. That is, making inferences about the desired/unknown quantity using the observed information and the functional form of the unknown (here, the exponential cubic spline) [30–32]. An example of solving a forward problem by computing positions when the velocity is known is shown in Figure 1.



**Figure 1.** An example of solving a forward problem from a velocity curve (**top**) to calculate positions/distances (**bottom**). The velocity curve is obtained by the exponential cubic spline computed at  $\vec{\xi} = [0, 1, 3, 6, 8, 10, 11, 12]$ . The positions were calculated at  $n = 13$  different time instances from the velocity by solving the forward problem. The time axis is given at arbitrary units (a.u.).

### 3. The Bayesian Recipe

The inverse problem for the position-velocity problem discussed in the previous section can be written using Bayes’ rule as follows:

$$p(\text{velocity} \in \mathbb{V} | \text{position} \in \mathbb{U}) = \frac{p(\text{velocity} \in \mathbb{V})p(\text{position} \in \mathbb{U} | \text{velocity} \in \mathbb{V})}{p(\text{position} \in \mathbb{U})}, \quad (8)$$

which shows how the joint posterior probability distribution is built to infer the velocity (gradient) profile when the position profile is known. Bayes’ rule in Equation (8) can be rewritten using data and parameters given in Table 1 as follows:

$$p(\vec{f}_v, \vec{\lambda}_v, \vec{\xi}_v, E_v | \vec{t}, \vec{x}, I) = \frac{p(\vec{f}_v, \vec{\lambda}_v, \vec{\xi}_v, E_v | I) p(\vec{x} | \vec{f}_v, \vec{\lambda}_v, \vec{\xi}_v, E_v, \vec{t}, I)}{p(\vec{x} | I)}, \quad (9)$$

where  $p(\vec{x} | \vec{f}_v, \vec{\lambda}_v, \vec{\xi}_v, E_v, \vec{t}, I)$  is the likelihood and the quantity  $I$  represents all the relevant background information. We assume the spline variables are independent. It then enables us to write the following:

$$p(\vec{f}_v, \vec{\lambda}_v, \vec{\xi}_v, E_v | I) = p(\vec{f}_v | I) p(\vec{\lambda}_v | I) p(\vec{\xi}_v | I) p(E_v | I). \quad (10)$$

The special case of when the number of knots and the position of knots are known (decided prior to computing the spline), we can write (10)

$$p(\vec{f}_v, \vec{\lambda}_v, \vec{\xi}_v, E_v | I) = p(\vec{f}_v | E_v, \vec{\xi}_v, I) p(\vec{\lambda}_v | E_v, \vec{\xi}_v, I). \quad (11)$$

Substituting Equation (10) into Equation (9), we get the general form of the posterior distribution:

$$p(\vec{f}_v, \vec{\lambda}_v, \vec{\zeta}_v, E_v | \vec{t}, \vec{x}, I) = \frac{p(\vec{f}_v | I) p(\vec{\lambda}_v | I) p(\vec{\zeta}_v | I) p(E_v | I) p(\vec{x} | \vec{f}_v, \vec{\lambda}_v, \vec{\zeta}_v, E_v, \vec{t}, I)}{p(\vec{x} | I)}. \tag{12}$$

The evidence  $p(\vec{x} | I)$  in Equation (9) is given by

$$p(\vec{x} | I) \stackrel{\text{def}}{=} \iiint p(\vec{f}_v | I) p(\vec{\lambda}_v | I) p(\vec{\zeta}_v | I) p(E_v | I) p(\vec{x} | \vec{f}_v, \vec{\lambda}_v, \vec{\zeta}_v, E_v, \vec{t}, I) d\vec{f}_v d\vec{\lambda}_v d\vec{\zeta}_v dE_v.$$

The relationship between the gradient and position in Equation (7) is included in the likelihood. We then assume the noise (or the uncertainty) in the measured data,  $\vec{\epsilon} \stackrel{\text{def}}{=} (\epsilon_1, \dots, \epsilon_n)$ , has a Gaussian process with

$$\vec{\epsilon} \sim N(\vec{\mu}, \Sigma), \tag{13}$$

where the quantity  $\vec{\mu}$  is the  $(n \times 1)$ -dimensional mean vector and  $\Sigma$  is a  $(n \times n)$ -dimensional covariance matrix of the noise. We further assume  $\vec{\mu} = 0$  and the noise is uncorrelated which, in turn, makes  $\Sigma$  a diagonal matrix with diagonal elements,  $\{\sigma_1, \dots, \sigma_n\}$ . Omitting the arrow notation for vectorial quantities, we can write the following:

$$x_i = \begin{cases} x_1 + \sum_{\gamma=1}^{i-1} \left( \int_{\zeta_{v_\gamma}}^{\zeta_{v_{\gamma+1}}} S_{v_\gamma}(t, f_v, \lambda_v, \zeta_v, E_v) dt \right) \\ \quad + \int_{\zeta_{v_i}}^{t_i} S_{v_i}(t, f_v, \lambda_v, \zeta_v, E_v) dt + \epsilon_i, & \text{for } \zeta_{v_i} < t_i < \zeta_{v_{i+1}} \\ x_1 + \sum_{\gamma=1}^{i-1} \left( \int_{\zeta_{v_\gamma}}^{\zeta_{v_{\gamma+1}}} S_{v_\gamma}(t, f_v, \lambda_v, \zeta_v, E_v) dt \right) + \epsilon_i, & \text{for } t_i = \zeta_{v_i} \end{cases} \tag{14}$$

In Equation (14), the first line (case 1) showcases when the time  $t_i$ , in which  $x_i$  is measured, falls between two knots of the spline, in which case  $S_{v_i}$ , calculated from  $\zeta_{v_i}$  to  $t_i$ , is a partial interval. The second line (case 2) showcases when  $t_i$  falls exactly on a knot, in which case  $S_{v_\gamma}$ , calculated at  $t_i = \zeta_{v_i}$ , is a full interval. Note that case 2 is a special case of case 1. In other words, case 2 can be achieved via case 1 when  $t_i = \zeta_{v_i}$ . Therefore, we rewrite Equation (14) for the general case subjecting the noise:

$$\epsilon_i = x_i - \left( x_1 + \sum_{\gamma=1}^{i-1} \int_{\zeta_{v_\gamma}}^{\zeta_{v_{\gamma+1}}} S_{v_\gamma}(t, f_v, \lambda_v, \zeta_v, E_v) dt + \int_{\zeta_{v_i}}^{t_i} S_{v_i}(t, f_v, \lambda_v, \zeta_v, E_v) dt \right), \tag{15}$$

for  $i = 1, \dots, n$ . By the assumption of iid (independent and identically distributed), we write our likelihood function:

$$p(\vec{x} | \vec{f}_v, \vec{\lambda}_v, \vec{\zeta}_v, E_v, \vec{t}, I) = \prod_{i=1}^n p(x_i | \vec{f}_v, \vec{\lambda}_v, \vec{\zeta}_v, E_v, t_i, I). \tag{16}$$

Combining Equations (15) and (16) together with the additional assumption that  $\sigma_i = \sigma_e = \text{constant}$  for any  $1 \leq i \leq n$ , the likelihood distribution becomes (case 1 of Equation (14) is assumed here for generality)

$$\begin{aligned} p(\vec{x} | \vec{f}_v, \vec{\lambda}_v, \vec{\zeta}_v, E_v, \vec{t}, I) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma_e^2}} \exp\left[-\frac{1}{2\sigma_e^2}(\epsilon_i)^2\right] \\ &= (2\pi\sigma_e^2)^{-n/2} \exp\sum_{i=1}^n \mathcal{G}(x_i, \vec{f}_v, \vec{\lambda}_v, \vec{\zeta}_v, E_v, \vec{t}), \end{aligned} \tag{17}$$

where,

$$\mathcal{G}(x_i, \vec{f}_v, \vec{\lambda}_v, \vec{\zeta}_v, E_v, \vec{t}) \stackrel{\text{def}}{=} -\frac{1}{2\sigma_e^2} \left[ x_i - \left( x_1 + \sum_{\gamma=1}^{i-1} \int_{\zeta_{v_\gamma}}^{\zeta_{v_{\gamma+1}}} S_{v_\gamma}(\cdot) dt + \int_{\zeta_{v_i}}^{t_i} S_{v_i}(\cdot) dt \right) \right]^2. \tag{18}$$

### The Choice of Prior Probability Distributions

According to Equation (11), we have three prior distributions to look at. We assume that there is no prior information about the spline values,  $p(\vec{f}_v|E_v, I)$ , or the position of knots,  $p(\vec{\zeta}_v|E_v, I)$ . However, we give our attention to the tension parameter,  $\vec{\lambda}_v$ , because the shape of the spline between knots depends on it. The tension value is a scale parameter and it can take any real number. With the change in  $\lambda_{v_i}$  from 0 to  $\infty$ , the spline changes from a polygonal function to a cubic spline [21,29]. A polygonal function lacks the smoothness required by a moving object and the cubic spline can sometimes overestimate the curvature by producing cubic curves when knots are placed close. Hence, estimating an optimum tension value plays a big role in getting a reliable gradient estimate profile and choosing an appropriate prior adds a great value to it.

Previous studies in the literature [21,33] used Jeffreys prior to merely scale down large tension values to avoid unrealistic results. In this work, we used a Gaussian distribution as a prior for the tension (Jeffreys and uniform priors were used in previous work by the same authors related to this problem [34]) because: (1) it allows one to have a higher probability for sensible values for the spline and lower probability for nonsensible values for the spline; and (2) a conjugate prior makes the posterior a Gaussian family of distribution (the known parametric form), thus making the computations easy. As stated earlier, polygonal functions lack the smoothness of a moving object and thus values of tension that produce polygonal functions are considered nonsensible values in this problem.

Assuming the prior distribution of tension parameters,  $\vec{\lambda}_v$ , are iid Gaussian distributions with mean  $\mu_i$  and variance  $\sigma_{\theta_{\lambda_i}}$ , we can write

$$\begin{aligned} p(\vec{\lambda}_v|E_v, I) &= \prod_{i=1}^{E_v-1} p(\lambda_{v_i}|E_v, I) \\ &= \prod_{i=1}^{E_v-1} \frac{1}{\sqrt{2\pi\sigma_{\theta_{\lambda_i}}^2}} \exp\left[-\frac{(\lambda_{v_i} - \mu_i)^2}{2\sigma_{\theta_{\lambda_i}}^2}\right] \\ &= (2\pi\sigma_{\theta_{\lambda}}^2)^{-(E_v-1)/2} \exp\left[-\sum_{i=1}^{E_v-1} \frac{(\lambda_{v_i} - \mu_i)^2}{2\sigma_{\theta_{\lambda}}^2}\right] \quad (19) \\ &\text{(by taking } \sigma_{\theta_{\lambda_i}} = \sigma_{\theta_{\lambda}} \forall i \text{ with } 1 \leq i \leq E_v - 1\text{).} \end{aligned}$$

Finally, the posterior distribution  $p(\vec{f}_v, \vec{\lambda}_v, \vec{\zeta}, E_v|\vec{t}, \vec{x}, I)$  in Equation (9) can be written as

$$\begin{aligned} p(\vec{f}_v, \vec{\lambda}_v, \vec{\zeta}, E_v|\vec{t}, \vec{x}, I) &\propto p(\vec{\lambda}_v|E_v, I) p(\vec{x}|\vec{f}_v, \vec{\lambda}_v, \vec{\zeta}, E_v, \vec{t}, I) \\ &\propto (2\pi\sigma_{\theta_{\lambda}}^2)^{-(E_v-1)/2} (2\pi\sigma_e^2)^{-(n/2)} \exp\left[-\sum_{i=1}^{E_v-1} \frac{(\lambda_{v_i} - \mu_i)^2}{2\sigma_{\theta_{\lambda}}^2} + \sum_{i=1}^n \mathcal{G}(x_i, \vec{f}_v, \vec{\lambda}_v, \vec{\zeta}, E_v, \vec{t})\right], \end{aligned}$$

where  $\mathcal{G}(x_i, \vec{f}_v, \vec{\lambda}_v, \vec{\zeta}, E_v, \vec{t})$  is given in Equation (18). In this work, the Gaussian prior distribution uses  $\mu_i = 0$  and  $\sigma_{\theta_{\lambda}} = 5$ . As noted previously [34], even though lambda has a range of 0 to  $\infty$ , the impact of lambda on the spline starts to flatten around 10. Thus, by setting  $\sigma_{\theta_{\lambda}} = 5$ ,  $2\sigma_{\theta_{\lambda}}$  covers 95% of the prior distribution.

### 4. Numerical Simulation

In this section, we discuss the numerical method followed to simulate the posterior distribution for inferring the velocity/gradient using positions/distances. We first created synthetic data from the velocity curve showcased for forward problem in Figure 1. We generated a sample of  $n = 121$  position data points. Thereafter, we added Gaussian noise of  $\mu = 0$  and different levels of  $\sigma_e$  to create different datasets of different noise levels ( $\sigma_e$ ). In this work, we hand-picked the number of knots and their positions for simplicity and to

demonstrate the inference (Equation (11)). Thus, we have  $2E_v - 1$  number of parameters to infer: spline values,  $f_v$ , and tension values  $\lambda_v$ . However, in future work, it is important that these two sets of parameters are inferred.

For the synthetic data, following the work in [29], we selected  $E_v = 8$  knots at  $\vec{\zeta}_v = [0, 1, 3, 6, 8, 10, 11, 12]$  in inferring the spline. Then, the posterior distribution with different priors (see Equation (12)) were simulated using a hybrid MCMC algorithm called DRAM [35,36]. The simulation begins with an initial minimization process. It should be noted that the posterior distribution is undefined at  $\lambda_v = 0$  due to numerical instability of the spline. Therefore, the smallest value that was used for  $\lambda_v$  was 0.001. Thus, the values of the parameters were initialized with the following values for the initial minimization process:

$$f_{v_{\text{initial}}} = 0, \text{ and } \lambda_{v_{\text{initial}}} = 0.001. \tag{20}$$

The initial minimization acts as a jump start to the DRAM algorithm. The proposal distributions of the parameters used in DRAM that generate the candidate sample values are Gaussian distributions. The parameter values of the proposal distributions (i.e., the mean and variance of Gaussian distributions) are user inputs and are showcased in Table 2, where  $LB$  and  $UB$  are the lower and upper bounds of the parameters, respectively.

**Table 2.** The sets of parameters in initial probability distributions of DRAM.

Parameter	Proposal Distribution	Prior Distribution
$f_{v_i}$ for $i = 1, \dots, E_v$	Gaussian $(\mu_{v_i}, \sigma_{v_i}^2)$ $\in (LB_{v_i}, UB_{v_i})$	$\mu_{\theta_{v_i}}, \sigma_{\theta_{v_i}}$
$\lambda_{v_i}$ for $i = 1, \dots, E_v - 1$	Gaussian $(\mu_{\lambda_i}, \sigma_{\lambda_i}^2)$ $\in (LB_{\lambda_i}, UB_{\lambda_i})$	$\mu_{\theta_{\lambda_i}}, \sigma_{\theta_{\lambda_i}}$

The optimum values  $\vec{f}_v^*$  and  $\vec{\lambda}_v^*$  produced by the initial minimization were used as the mean values of the proposal distributions  $(\mu_{v_i}, \mu_{\lambda_i})$ , so that the initial proposal distributions are centered around  $\vec{f}_v^*$  and  $\vec{\lambda}_v^*$ . The information of prior distributions can be specified with prior means  $\mu_{\theta_{v_i}}, \mu_{\theta_{\lambda_i}}$  and prior variances  $\sigma_{\theta_{v_i}}, \sigma_{\theta_{\lambda_i}}$ . The DRAM procedure takes the variance of the prior as the initial variance of the parameters if the covariance matrix is not specified by the user. When the prior variance is not specified, the variances of the proposal distributions are calculated as

$$\sigma_{v_i}^2 = (\mu_{v_i} \times 0.05)^2, \text{ and } \sigma_{\lambda_i}^2 = (\mu_{\lambda_i} \times 0.05)^2. \tag{21}$$

In other words, if no prior information is available about the variance of the parameter, 5% of the mean of the initial distribution is taken as the initial variance. Finally, samples were drawn from these Gaussian proposals and the marginal densities of the parameters were computed. The MCMC simulations were run until convergence was observed. Each sampling procedure of length  $\Omega$  was repeated  $M = 100$  times. At each of the  $M$ -th repetitions, a different dataset of same noise level was created followed by an initial minimization process. All the sample chains were stored and, in addition, for a given sample chain, the first few samples were burnt during  $\Omega \times p\%$  period. The quantity  $p$  denotes the burn-in time. The average of  $M$  marginal densities of each parameter were then computed. In what follows, we assume that  $\omega$  denotes the remaining number of samples after the burn-in period, that is,  $\omega \stackrel{\text{def}}{=} (1 - p)\% \times \Omega$ . Then, for any  $k = 1, \dots, E_v$ , the mean values  $\{f_{v_k}^*\}$  of those average marginal densities were calculated as

$$f_{v_k}^* \stackrel{\text{def}}{=} \frac{1}{\omega} \sum_{i=1}^{\omega} \overline{f_{v_{ki}}}, \tag{22}$$

where  $\overline{f_{v_{ki}}}$  denote the average sample values of  $f_{v_k}$  after  $M$  number of iterations,

$$\bar{f}_{vki} \stackrel{\text{def}}{=} \frac{1}{M} \sum_{j=1}^M f_{vki j}. \quad (23)$$

Following the same line of reasoning presented for Equations (22) and (23), we also have

$$\lambda_{v_s}^* \stackrel{\text{def}}{=} \frac{1}{\omega} \sum_{i=1}^{\omega} \bar{\lambda}_{vsi} = \frac{1}{\omega M} \sum_{i=1}^{\omega} \sum_{j=1}^M \lambda_{v_{sij}}, \quad (24)$$

for any  $s = 1, \dots, E_v - 1$  with  $\bar{\lambda}_{vsi}$  denoting the average sample values of tension after M number of iterations:

$$\bar{\lambda}_{vsi} \stackrel{\text{def}}{=} \frac{1}{M} \sum_{j=1}^M \lambda_{v_{sij}}. \quad (25)$$

Finally, the standard deviations  $\{\sigma_{v_k}^*\}$  and  $\{\sigma_{\lambda_s}^*\}$  of the parameters are calculated from the average of all chains after M iterations as

$$\sigma_{v_k}^* \stackrel{\text{def}}{=} \sqrt{\frac{1}{\omega - 1} \sum_{i=1}^{\omega} (\bar{f}_{vki} - f_{v_k}^*)^2}, \quad (26)$$

for any  $k = 1, \dots, E_v$  and

$$\sigma_{\lambda_s}^* \stackrel{\text{def}}{=} \sqrt{\frac{1}{\omega - 1} \sum_{i=1}^{\omega} (\bar{\lambda}_{vsi} - \lambda_{v_s}^*)^2}, \quad (27)$$

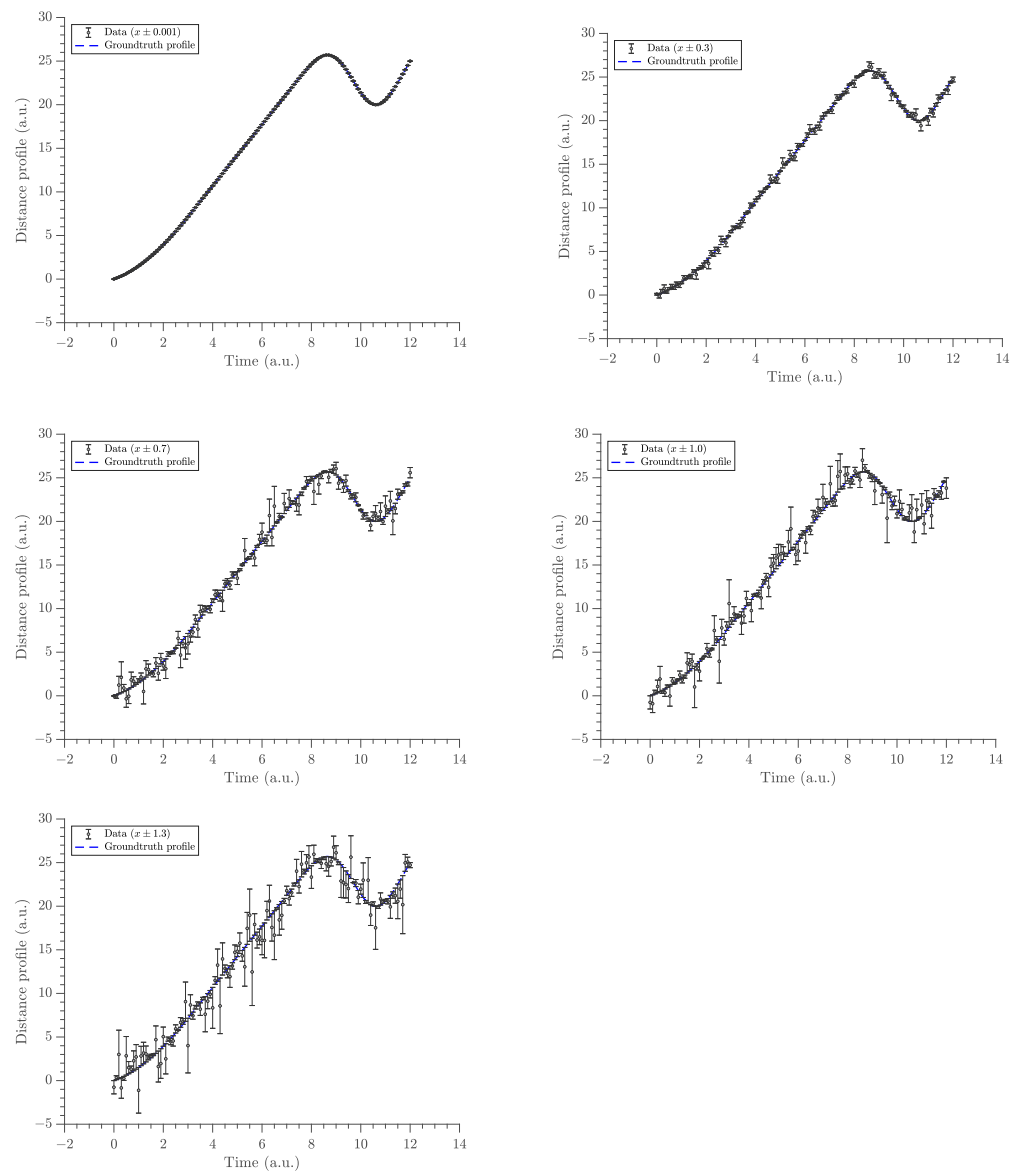
for any  $s = 1, \dots, E_v - 1$ , respectively.

## 5. Results

This section presents the results obtained from the methodology built throughout the previous sections. The sensitivity of gradient estimates at different noise levels (of the position data) were tested. The noise levels (denoted by  $\sigma_e$ ) on  $x$ -space are depicted in Figure 2 with five different noise levels:  $\sigma_e \in \{0.001, 0.3, 0.7, 1.0, 1.3\}$ . The noise levels were selected so that they cover almost all possibilities of the standard Gaussian noise. That is, by the definition of the Gaussian distribution, the first four noise levels cover a probability of 68.2% and the last noise level covers a probability of 95%.

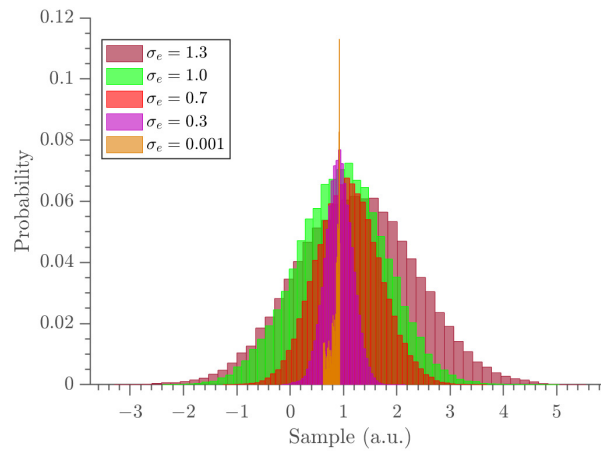
The sensitivity of noise levels to the posterior distribution was investigated when the Gaussian prior was used for the tension parameter. These results of the sensitivity analysis were compared against a common traditional method of fitting the same type of exponential cubic spline in  $x$ -space and the gradient was obtained by analytically differentiating the resulting fitted spline function. The notation “ $i$ -fit ( $i$ -space) with  $i = x, v$ ” denotes that  $i$ -fit was fitted on the  $i$ -space, where the  $x$ -fit is the distance profile, while the  $v$ -fit is the gradient profile.



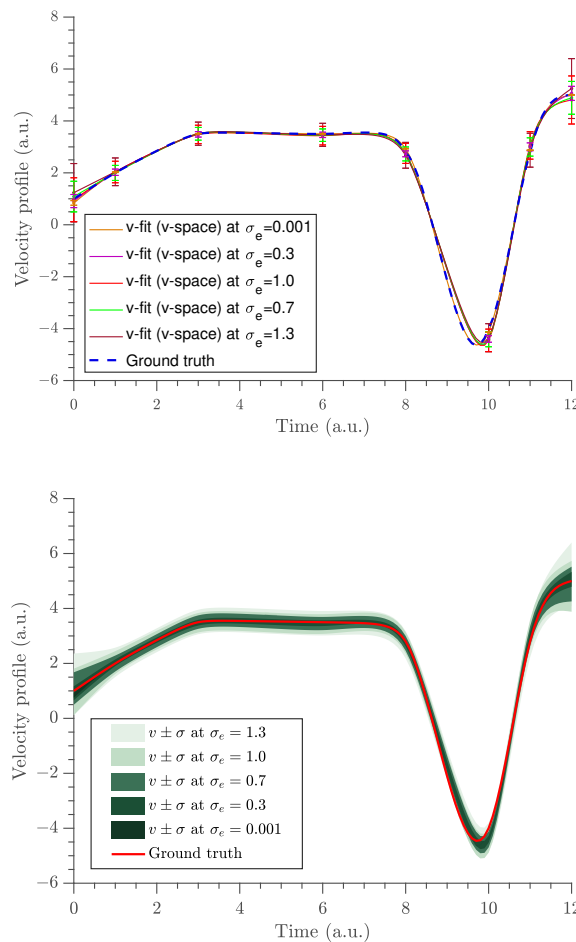


**Figure 2.** The panel of time-position data with error bars at five different noise levels. Noise levels increase from left to right and top to bottom. The length of error bars is relative to the magnitude of the noise.

A constant force was observed between the times 0–8 (a.u.) and short impulses were observed between the times 8–11 (a.u.). Moreover, it was observed that as the noise level increased, the data around the short impulses tended to become fuzzier so that it was hard to identify the trajectory. When the noise increased, the width of the marginal distributions were expanded. This, in turn, resulted in an increased length of the error bars of the parameter estimates. As a demonstration, we show the marginal probability distribution of  $f_{v_1}$  in Figure 3 at all noise levels. The gradient estimates at all noise levels are depicted in Figure 4 and the uncertainties of  $f_{v_i}$  parameters are shown in the same figure using error bars (top panel) and Bayesian credible intervals (bottom panel). The estimates almost overlap with the ground truth data at all noise levels, except at the boundaries and near  $t = 10$  (a.u.), where the short impulses were present in the position profile.

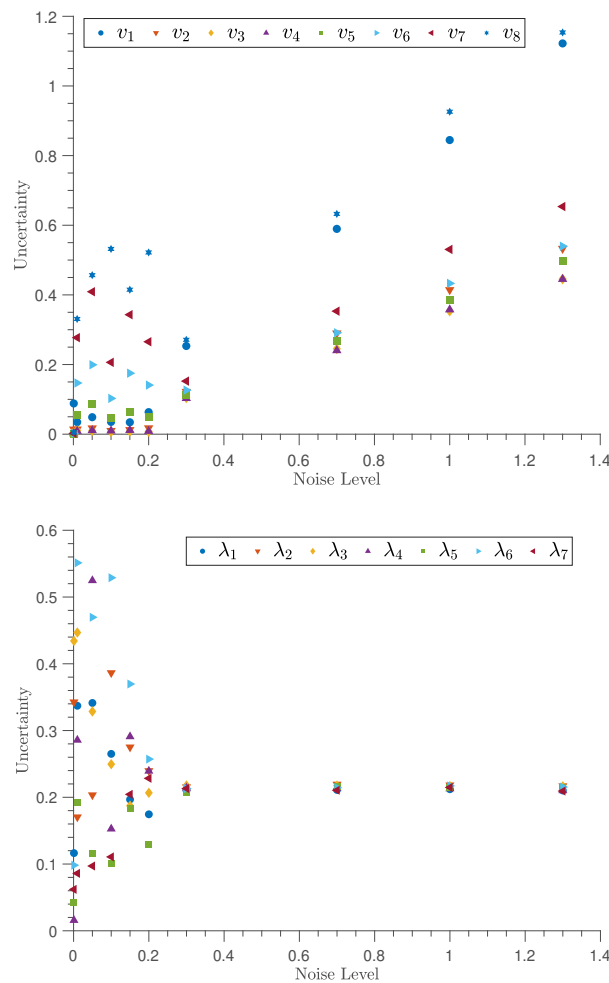


**Figure 3.** The comparison of marginal probability densities of the  $f_{v_1}$  parameter at all noise levels.



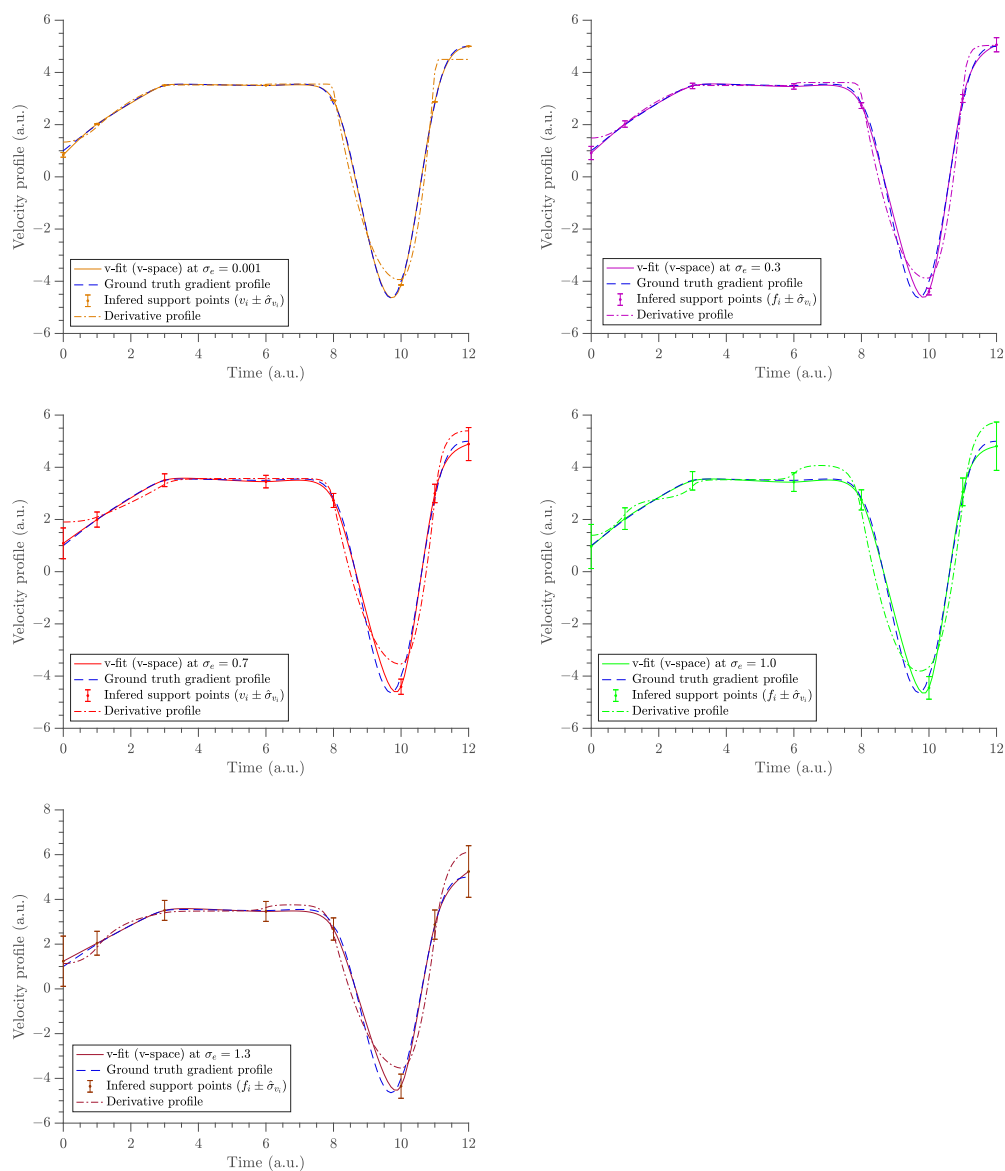
**Figure 4.** The gradient estimates obtained at the five different noise levels. The uncertainty of the velocity estimates ( $\hat{\sigma}_v$ ) are shown using error bars (**top** panel) and credible intervals (**bottom** panel).

The uncertainties of both velocity and tension parameters are characterized in Figure 5. The uncertainties of the velocity parameters with the noise level exhibit a linear relationship reflecting the linear relationship of velocity parameters in the likelihood function. The uncertainties of the tension parameters converge to the Gaussian prior uncertainty when the data noise exceeds the prior uncertainty. To help illustrate this, additional noise levels were added in these graphs in Figure 5 to show the relationship more clearly.

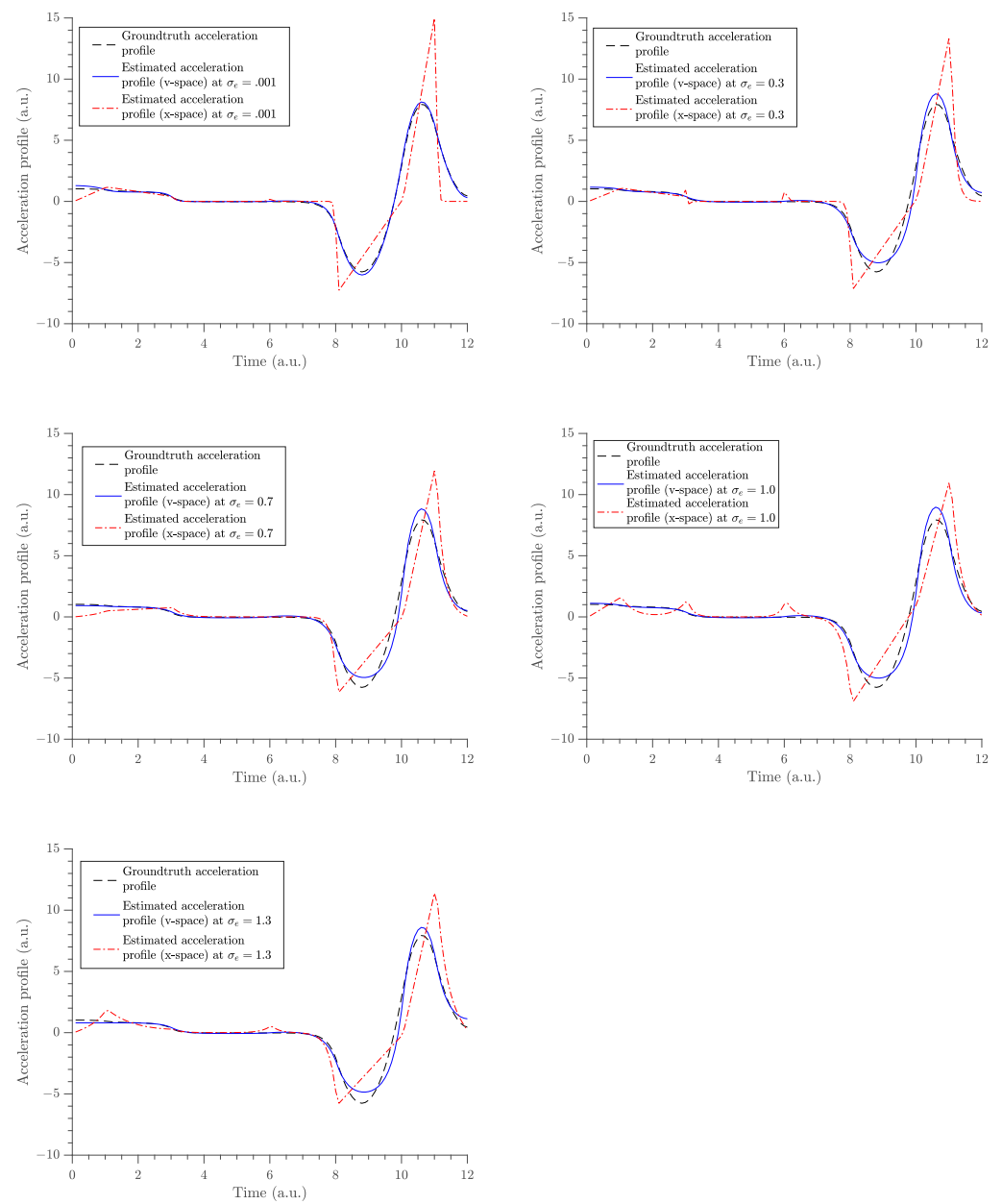


**Figure 5.** The relationship between uncertainty of the estimates and the noise level of the system. The top panel depicts the relationship of velocity parameters (here,  $v_i$  indicates  $f_{v_i}$ ), whereas the bottom panel shows the relationship of tension parameters (here,  $\lambda_i$  indicates  $\lambda_{v_i}$ ). Note that additional noise levels are included to demonstrate how the uncertainty in the estimated lambda converges to the Gaussian prior uncertainty when the data noise exceeds the prior uncertainty.

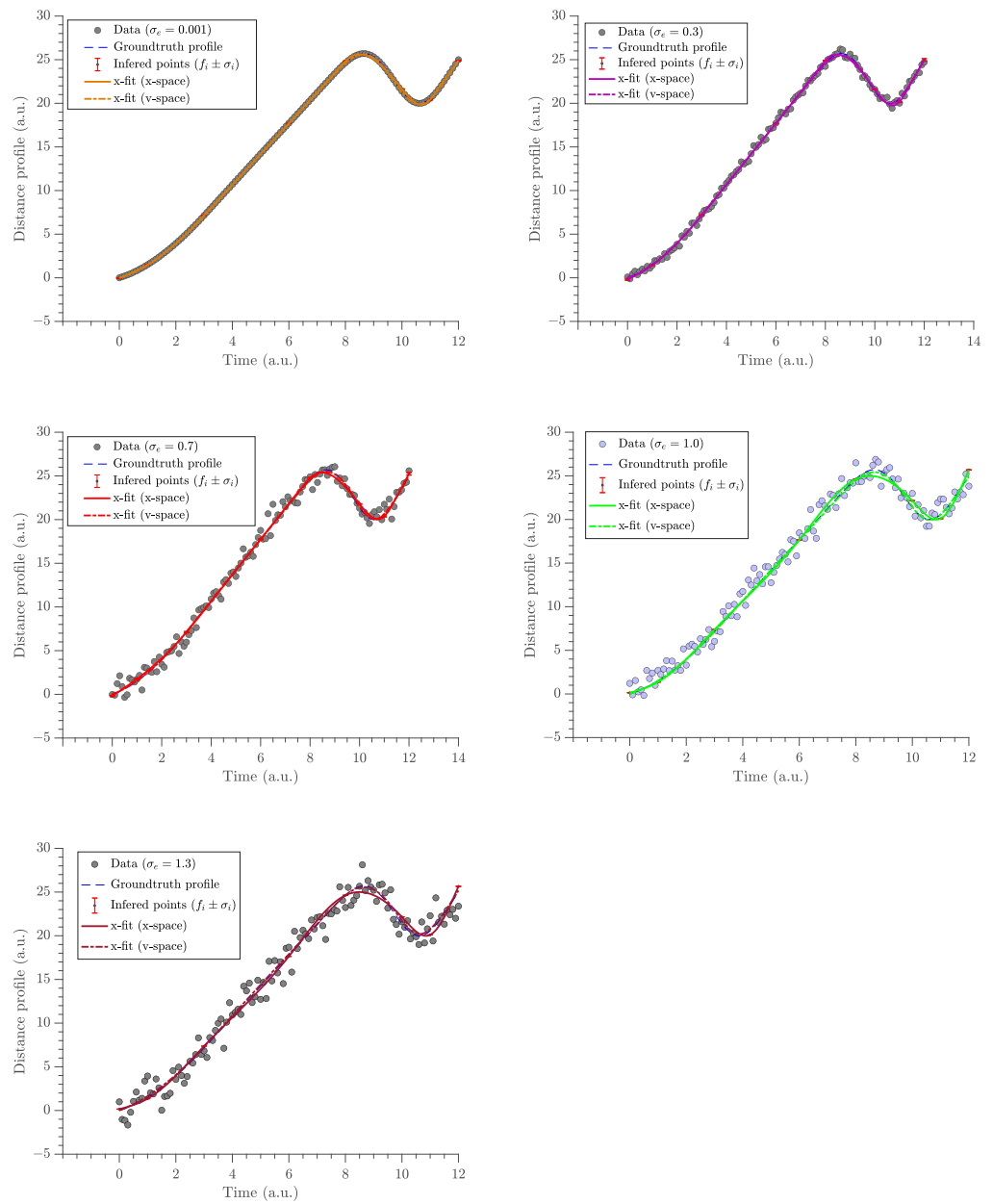
The gradient estimates are compared with the analytic derivative of the x-fit obtained in  $\mathbb{U}$ . The x-fit is obtained by fitting the same exponential cubic spline to the time-position data in  $\mathbb{U}$ . This was performed at the same knot positions. The gradient profile, the acceleration profile, and the position profile were obtained from the two methods (via v-fit from  $\mathbb{V}$  and x-fit from  $\mathbb{U}$ ) and are compared in Figures 6–8, respectively. The acceleration profile from our method (v-fit in  $\mathbb{V}$ ) was obtained by differentiating the exponential cubic spline, i.e., the analytic derivative of exponential cubic spline. It should be noted that ideally the acceleration should be inferred directly. Then, velocity and position could be found via integration. However, the focus and motivation of this paper was velocity. We are only using this as a quick comparison to illustrate how poor the acceleration fit would be when differentiating twice. The same was obtained using the traditional method by finite-differencing its gradient profile, i.e., finite differencing the analytic derivative of the exponential cubic spline.



**Figure 6.** The panel of velocity estimates from the two spaces compared at five different noise levels. The noise level increases from left to right and top to bottom. v-fit ( $v$ -space) shows the gradient inferred by placing the exponential cubic spline in  $v$ -space (solid lines). The derivative profile shows the gradient estimates obtained by differentiating the exponential cubic spline fit in  $x$ -space (dashed-dot lines).



**Figure 7.** The panel of acceleration estimates obtained from the two methods are compared at five different noise levels. Solid blue lines shows the estimates obtained from our method and dashed-dot red lines shows the estimates obtained from the traditional method.



**Figure 8.** The panel of  $x$ -fit estimates obtained from two different methods compared at five different noise levels. Dashed-dot lines shows the estimates obtained from our method and solid lines shows the estimates obtained from the traditional method.

The gradient estimates that emerge from the two spaces (namely, the  $x$ -space and  $v$ -space) clearly show some remarkable differences. In particular, the difference in accuracy is shown in Table 3 using the error norm:

$$||e||^2 \stackrel{\text{def}}{=} \sum_{i=1}^L \left( f_{v_{\text{estimated}_i}} - f_{v_{\text{true}_i}} \right)^2, \tag{28}$$

where  $L$  is the sample size [37].

**Table 3.** The comparison of accuracy of velocity fit from our method (in  $v$ -fit ( $v$ -space)) and the traditional method (derivative of  $x$ -fit in  $x$ -space).

Noise Level	$\ e\ ^2$	
	$v$ -Fit ( $v$ -Space)	Derivative of $x$ -Fit ( $x$ -Space)
0.001	0.2077	14.4107
0.3	1.9679	15.0792
0.7	3.1793	20.9974
1	3.1569	30.1122
1.3	3.8159	31.2719

The acceleration characterizes the acting forces on the object. The acceleration estimate ( $x$ -space) could not achieve constant force where it is expected. The errors at short impulses around  $t = 10$  a.u. started becoming very large, which indicates the possibility of infinite forces. Moreover, the acceleration profile from the  $x$ -space (red dashed-dot lines) is not realistic with its very sharp turns. Therefore, the acceleration estimate from the velocity space is a reliable estimator of acting forces. The goodness of fit was tested by studying the squared sum of errors (SSE) defined as [38]

$$\text{SSE} \stackrel{\text{def}}{=} \sum_{i=1}^n (\hat{f}_i - f_i)^2 \quad (29)$$

where  $\hat{f}_i$  represent the estimated function values. The quantity  $f_i$  in Equation (29) is replaced with noisy  $f_i$  to obtain the second and the third columns in Table 4, while  $f_i$  is replaced with the original  $f_i$  without noise to calculate the fourth and fifth columns in the same table.

**Table 4.** The comparison of the quality of  $x$ -fit from  $x$ - and  $v$ -spaces respectively using squared sum of errors (SSE).

$\sigma_e$	SSE with Noisy Data		SSE with True Data	
	$x$ -Fit ( $\mathbb{U}$ )	$x$ -Fit ( $\mathbb{V}$ )	$x$ -Fit ( $\mathbb{U}$ )	$x$ -Fit ( $\mathbb{V}$ )
0.001	1.1743	0.6082	1.1753	0.6089
0.3	12.4677	12.6049	2.1946	0.8541
0.7	62.4582	63.4059	4.1855	0.8176
1.0	121.0864	120.8161	13.3478	1.1253
1.3	163.1339	264.3176	18.9582	3.1651

The trajectory estimate (position profiles) is a bonus product from the gradient profile ( $v$ -space) and is obtained by integrating the gradient estimate ( $v$ -space). They are compared in Figure 8. The  $x$ -fit ( $x$ -space) follows the noisy data closely. This is because, when fitting a curve in the same space as data, that curve is trying to minimize the SSE, which means it tries to match the data as much as possible. Therefore, the  $x$ -fit from  $\mathbb{U}$  tries to satisfy the noisy data as much as possible. However, the  $x$ -fit ( $v$ -space) follows the ground truth data more than it follows the noisy data (see Table 4). When integrating the  $v$ -fit ( $v$ -space), an additional order of differentiability is added to the trajectory estimate. Therefore, the  $x$ -fit ( $v$ -space) has a higher order of differentiability than that of the  $x$ -fit ( $x$ -space). This additional order of differentiability ensures the continuity of the velocity and, therefore, it satisfies the constraint of the finite force of the object.

The calculated SSE with true data and noisy data are shown in the Table 4. The SSE values for noisy data are smaller for the  $x$ -fit ( $x$ -space). This, in turn, reflects the fact that the  $x$ -fit ( $x$ -space) better follows noisy data. However, the SSE values for true data are smaller for the  $x$ -fit ( $v$ -space). This, instead, reflects the fact that the  $x$ -fit ( $v$ -space) better follows true data.

## 6. Conclusions

In this paper, we proposed a method to compute the gradient profile of a noisy system using the Bayesian inference method. It allowed us to infer the unobservable quantity, the gradient (velocity), by building a meaningful relationship between the unobservable quantity in velocity space and the observable quantity in data space. Furthermore, the unobservable quantity (the gradient) was modeled using the exponential cubic spline in velocity space. Using Bayesian methods, the parameters of this exponential cubic spline were inferred. The results of this new method were compared against those of a traditional method of modeling noisy data, which uses the exponential cubic spline in data space and takes the subsequent analytic derivative to obtain the gradient.

The results show that the gradient estimates obtained by modeling in velocity space are more accurate than the estimates obtained via the traditional method (see Table 3). Moreover, our method was able to produce better acceleration estimates with reliable and accurate values, where a constant force is expected (see Figure 7). We also compared the trajectory profile. We conclude that when the traditional method is used, the trajectory estimates tend to follow noisy data, whereas when our method is used, the estimates tend to follow the ground truth data, which, in return, produce more accurate estimates (see Figure 8 and Table 4). It is argued that by integrating the model in velocity space to compute the trajectory values, an extra order of differentiability is added. This argument can be further extended to suggest that the acceleration should be inferred first. Then, velocity and trajectories can be inferred by integration. However, this was not the focus of the paper. In conclusion, the method demonstrated in this paper is superior when estimating the velocity of a moving object under finite force as compared to others in the literature (for instance, see [1] and the references therein). It provided better results in all three estimates: trajectory, velocity, and acceleration.

It is necessary to note that, although our main focus in this paper was on the proposal of a novel theoretical method to estimate a gradient profile from discrete noisy data, a number of improvements can be sought in its practical implementation. For instance, the use of an MCMC algorithm gains precision at the expense of speed. A faster algorithm (such as the expectation maximization algorithm [39]) may be needed for real-time estimation. Furthermore, the amount and placement of the knots lacks a systematic guiding principle. In the future, we plan to use Bayesian model selection for determining the amount of knots. For the placement issue, we can adopt a hierarchical approach by including spline knot location algorithms [40] in conjunction with our main algorithm. This would provide estimates for the values associated with the knots and where they should be located. Finally, we hope to apply our Bayesian estimation technique to more realistic problems in which acceleration, velocity, and trajectory estimations are needed. In particular, from both theoretical and applied perspectives, in the future, we think it may be worth exploring the possibility of characterizing velocity profiles in ferromagnetic fluids [41] with the use of entropic inference methods that encompass Bayesian techniques [42] and, moreover, show promising results in inferring the ferromagnetic properties of materials [43].

**Author Contributions:** Conceptualization, A.G.; Methodology, K.D.S.; Supervision, A.G.; Writing—original draft, K.D.S.; Writing—review & editing, C.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** The authors would like to thank the meaningful and helpful discussions had with Udo Von Toussaint at the Max-Planck Institute in Garching, Germany.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

In this appendix, we present some details on the integration of the exponential cubic spline. The integration is performed in two cases. In one case, one integrates the spline



function between two knots, and it is referred to as  $I_{\text{full}}$ . In the other case, one integrates the spline function from a knot to a random point before the next knot, and it is referred to as  $I_{\text{partial}}$ . The two types of integrals yield

$$I_{\text{full}} = \int_{\xi_{v_k}}^{\xi_{v_{k+1}}} S_v(t, f_v, \lambda_v, \xi_v, E_v) dt, \quad (\text{A1})$$

with  $k = 1, \dots, E_v - 1$ , and

$$I_{\text{partial}} = \int_{\xi_{v_k}}^{t_i} S_v(t, f_v, \lambda_v, \xi_v, E_v) dt \quad (\text{A2})$$

with  $k = 1, \dots, E_v - 1$ ,  $i = 1, \dots, n$ , and  $\xi_{v_k} \leq t_i < \xi_{v_{k+1}}$ , respectively. The solutions to these two cases presented in Equations (A1) and (A2) can be found analytically and result in the following expressions:

$$I_{\text{partial}} = \frac{f_{v_{i+1}}}{2h_{v_i}}(t_i - \xi_{v_i})^2 + \frac{f_{v_i}(t_i - \xi_{v_i})}{2h_{v_i}}(2\xi_{v_{i+1}} - t_i - \xi_{v_i}) - \frac{M_{v_i}(t_i - \xi_{v_i})}{2\lambda_{v_i}^2 h_{v_i}}(2\xi_{v_{i+1}} - t_i - \xi_{v_i}) \quad (\text{A3})$$

$$+ \frac{M_{v_i}}{\lambda_{v_i}^3 \sinh(\lambda_{v_i} h_{v_i})} \{ \cosh(\lambda_{v_i} h_{v_i}) - \cosh[\lambda_{v_i}(t_i - \xi_{v_{i+1}})] \} + \frac{M_{v_{i+1}}}{\lambda_{v_i}^3 \sinh(\lambda_{v_i} h_{v_i})} \{ \cosh[\lambda_{v_i}(t_i - \xi_{v_i})] - 1 \} - \frac{M_{v_{i+1}}}{2\lambda_{v_i}^2 h_{v_i}}(t_i - \xi_{v_i})^2, \quad (\text{A4})$$

and

$$I_{\text{full}} = \frac{f_{v_i} h_{v_i}}{2} + \frac{f_{v_{i+1}} h_{v_i}}{2} - \frac{M_{v_i} h_{v_i}}{2\lambda_{v_i}^2} - \frac{M_{v_{i+1}} h_{v_i}}{2\lambda_{v_i}^2} + \frac{(M_{v_{i+1}} + M_{v_i})}{\lambda_{v_i}^3} \tanh\left(\frac{\lambda_{v_i} h_{v_i}}{2}\right). \quad (\text{A5})$$

## Appendix B

In this appendix, as it is in the framework of Markov chain Monte Carlo (MCMC) methods, we present some details on the Delayed Rejection Adaptive Metropolis (DRAM) algorithm used to compute posterior distributions in our work. DRAM is a hybrid algorithm in which the concepts of Delayed Rejection (DR) and Adaptive Metropolis (AM) are combined for enhancing the performance of Metropolis-Hastings (MH)-type MCMC algorithms [35]. In the basic MH algorithm, upon the rejection of the new candidate at a  $(j + 1)$ th state, the sample remains at the current point  $(j)$ . However, when the Delayed Rejection is introduced, upon rejection, another new candidate  $z$  is drawn. Then, one finds out whether it can be accepted. The AM feature, instead, is a global adaptive strategy, which is combined with the partial local adaptation caused by the DR strategy. The idea of AM is to tune the proposal distribution at each time step depending on the past. In other words, the covariance matrix of the proposal distribution depends on the history of the sample chain (sequence of samples). Usually, the adaptation starts after an initial period of time  $(0, t)$ .

The basic steps of the DRAM algorithm are given in Algorithm A1 with an arbitrary initial point  $\Theta^{(0)}$ . The acceptance probabilities at steps 6 and 15 are shown in Equations (A6a) and (A6b), respectively.

$$\alpha_1(x, y) = \begin{cases} \min\left[\frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}, 1\right] & \text{if } \pi(x)q(x, y) > 0, \\ 1 & \text{otherwise.} \end{cases} \quad (\text{A6a})$$

$$\alpha_2(x, y, z) = \begin{cases} \min\left[\frac{\pi(z)}{\pi(x)} \frac{q(z, y)}{q(x, y)} \frac{q(z, y, x)}{q(x, y, z)} \frac{[1 - \alpha_1(z, y)]}{[1 - \alpha_1(x, y)]}, 1\right] & \text{if } \pi(x)q(x, y)q(x, y, z)[1 - \alpha_1(x, y)] > 0, \\ 1 & \text{otherwise.} \end{cases} \quad (\text{A6b})$$

where  $q(\theta)$  represents the proposal distribution and  $\Theta^{(j)}$  represents the set of parameters of the proposal distribution at  $j$ -th iteration.

---

#### Algorithm A1 DRAM

---

```

1: for  $j = 1 : N$  do
2:   Generate  $y$  from  $q(\Theta^{(j)}, \cdot)$  and  $u$  from  $U(0, 1)$ 
3:   if  $y$  is outside  $(L, U)$  then
4:     Reject  $y$  (level 1)
5:   else
6:     Calculate  $\alpha_1(\Theta^{(j)}, y)$ 
7:     if  $u \leq \alpha_1(\Theta^{(j)}, y)$  then
8:        $\Theta^{(j+1)} = y$ 
9:     else
10:      Reject  $y$  (level 2) {starts delayed rejection}
11:      Generate new candidate  $z$  from  $q(y, \cdot)$ 
12:      if  $z$  is outside  $(L, U)$  then
13:        Reject  $z$  (level 3)
14:      else
15:        Calculate  $\alpha_2(\Theta^{(j)}, y, z)$  and  $u$  from  $U(0, 1)$ 
16:        if  $u \leq \alpha_2(\Theta^{(j)}, y, z)$  then
17:           $\Theta^{(j+1)} = z$ 
18:        else
19:          Reject  $z$  (level 4)
20:           $\Theta^{(j+1)} = \Theta^{(j)}$ 
21:        end if
22:      end if
23:    end if
24:  end if
25: end for
26: Return the values  $\{\Theta^{(0)}, \Theta^{(1)}, \Theta^{(2)}, \dots, \Theta^{(N)}\}$ .

```

---

#### References

1. von Toussaint, U.; Gori, S. Digital particle image velocimetry using splines in tension. In Proceedings of the 30th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (MaxEnt 2010), Chamonix, France, 4–9 July 2010.
2. Feng, Y.; Goree, J.; Liu, B. Errors in particle tracking velocimetry with high-speed cameras. *Rev. Sci.* **2011**, *82*, 053707. [[CrossRef](#)] [[PubMed](#)]
3. Chilenski, M.A.; Greenwald, M.; Marzouk, Y.; Howard, N.; White, A.E.; Rice, J.E.; Walk, J.R. Improved profile fitting and quantification of uncertainty in experimental measurements of impurity transport coefficients using Gaussian process regression. *Nucl. Fusion* **2015**, *55*, 023012. [[CrossRef](#)]
4. Ziegenhein, T.; Lucas, D. On sampling bias in multiphase flows: Particle image velocimetry in bubbly flows. *Flow Meas. Instrum.* **2016**, *48*, 36–41. [[CrossRef](#)]
5. Brown, R.H.; Schneider, S.C.; Mulligan, M.G. Analysis of algorithms for velocity estimation from discrete position versus time data. *IEEE Trans. Ind. Electron.* **1992**, *39*, 11–19. [[CrossRef](#)]
6. Reinsch, C.H. Smoothing by spline functions. *Numer. Math.* **1967**, *10*, 177–183. [[CrossRef](#)]
7. Reinsch, C.H. Smoothing by spline functions. II. *Numer. Math.* **1971**, *16*, 451–454. [[CrossRef](#)]

8. THastie, J.; Tibshirani, R.J. Generalized additive models. In *Monographs on Statistics and Applied Probability*; Taylor & Francis: Abingdon, UK, 1990; Volume 43.
9. Wahba, G. *Spline Models for Observational Data*; SIAM: Philadelphia, PA, USA, 1990; Volume 59.
10. Wand, M.P.; Jones, M.C. *Kernel Smoothing*; CRC Press: Boca Raton, FL, USA, 1994.
11. Green, P.J.; Silverman, B.W. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*; CRC Press: Boca Raton, FL, USA, 1993.
12. Wold, S. Spline functions in data analysis. *Technometrics* **1974**, *16*, 1–11. [[CrossRef](#)]
13. Silverman, B.W. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *J. R. Stat. Soc.* **1985**, *47*, 1–21. [[CrossRef](#)]
14. Wahba, G. Improper priors, spline smoothing and the problem of guarding against model errors in regression. *J. R. Stat. Soc.* **1978**, *40*, 364–372. [[CrossRef](#)]
15. Kimeldorf, G.S.; Wahba, G. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Ann. Math. Stat.* **1970**, *41*, 495–502. [[CrossRef](#)]
16. Ansley, C.F.; Kohn, R. Estimation, filtering, and smoothing in state space models with incompletely specified initial conditions. *Ann. Stat.* **1985**, *13*, 1286–1316. [[CrossRef](#)]
17. Berry, S.M.; Carroll, R.J.; Ruppert, D. Bayesian smoothing and regression splines for measurement error problems. *J. Am. Stat. Assoc.* **2002**, *97*, 160–169. [[CrossRef](#)]
18. Fischer, R.; Hanson, K.M.; Dose, V.; von Der Linden, W. Background estimation in experimental spectra. *Phys. Rev. E* **2000**, *61*, 1152. [[CrossRef](#)] [[PubMed](#)]
19. Heinecke, A.; Tallarita, M.; De Iorio, M. Bayesian splines versus fractional polynomials in network meta-analysis. *BMC Med. Res. Methodol.* **2020**, *20*, 261. [[CrossRef](#)]
20. Dose, V.; Fischer, R. Function estimation employing exponential splines. *AIP Conf. Proc.* **2005**, *803*, 67–71.
21. Fischer, R.; Dose, V. Flexible and reliable profile estimation using exponential splines. *AIP Conf. Proc.* **2006**, *872*, 296–303.
22. Bayes, T. LII. An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S. *Philos. Trans. R. Soc. Lond.* **1963**, *53*, 370–418.
23. Box, G.E.P.; Tiao, G.C. *Bayesian Inference in Statistical Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2011; Volume 40.
24. Geweke, J. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica* **1989**, *57*, 1317–1339. [[CrossRef](#)]
25. Dempster, A.P. A generalization of Bayesian inference. *J. R. Stat. Soc.* **1986**, *30*, 205–232.
26. Gamerman, D.; Lopes, H.F. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*; CRC Press: Boca Raton, FL, USA, 2006.
27. Knill, D.C.; Richards, W. *Perception as Bayesian Inference*; Cambridge University Press: Cambridge, UK, 1996.
28. Newton, M.A.; Raftery, A.E. Approximate Bayesian inference with the weighted likelihood bootstrap. *J. R. Stat. Soc.* **1994**, *56*, 3–26. [[CrossRef](#)]
29. Rentrop, P. An algorithm for the computation of the exponential spline. *Numer. Math.* **1980**, *35*, 81–93. [[CrossRef](#)]
30. Mohammad-Djafari, A. Bayesian inference for inverse problems. *AIP Conf. Proc.* **2002**, *617*, 477–496.
31. Sivia, D.; Skilling, J. *Data Analysis: A Bayesian Tutorial*; Oxford University Press: Oxford, UK, 2006.
32. Gelman, A.; Carlin, J.B.; Stern, H.S.; Rubin, D.B. *Bayesian Data Analysis*; Taylor & Francis: Abingdon, UK, 2014; Volume 2.
33. Fischer, R.; Dinklage, A.; Turkin, Y. Non-parametric profile gradient estimation. In Proceedings of the 33rd EPS Conference on Plasma Physics, Rome, Italy, 19–23 June 2006; Volume 30I, p. 2.127.
34. De Silva, K. P. *Bayesian Inference of Profile Gradient with Exponential Cubic Splines*; Clarkson University: Potsdam, NY, USA, 2017.
35. Haario, H.; Laine, M.; Mira, A.; Saksman, E. DRAM: Efficient adaptive MCMC. *Stat. Comput.* **2006**, *16*, 339–354. [[CrossRef](#)]
36. *MATLAB*; Version R2015b; The MathWorks Inc.: Natick, MA, USA, 2015.
37. Hansen, P.C. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Rev.* **1992**, *34*, 561–580. [[CrossRef](#)]
38. Merriman, M. *A Text Book on the Method of Least Squares*; John Wiley & Sons: Hoboken, NJ, USA, 1909.
39. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B Methodol.* **1977**, *39*, 1–38.
40. Montoya, E.L.; Ulloa, N.; Miller, V. A simulation study comparing knot selection methods with equally spaced knots in a penalized regression spline. *Int. J. Stat. Probab.* **2014**, *3*, 96. [[CrossRef](#)]
41. Li, M.; Liang, K. The velocity distribution for ferromagnetic fluids. *Adv. Mater. Res.* **2014**, *860*, 1506–1509. [[CrossRef](#)]
42. Giffin, A.; Caticha, A. Updating probabilities with data and moments. *AIP Conf. Proc.* **2007**, *954*, 74–84.
43. Giffin, A.; Cafaro, C.; Ali, S.A. Application of the maximum relative entropy method to the physics of ferromagnetic materials. *Physica A* **2016**, *455*, 11–26. [[CrossRef](#)]