

## Smoothing Splines for Discontinuous Signals

Martin Storath & Andreas Weinmann

To cite this article: Martin Storath & Andreas Weinmann (2024) Smoothing Splines for Discontinuous Signals, *Journal of Computational and Graphical Statistics*, 33:2, 651-664, DOI: [10.1080/10618600.2023.2262000](https://doi.org/10.1080/10618600.2023.2262000)

To link to this article: <https://doi.org/10.1080/10618600.2023.2262000>



[View supplementary material](#)



Published online: 21 Nov 2023.



[Submit your article to this journal](#)



Article views: 130



[View related articles](#)



CrossMark

[View Crossmark data](#)



## Smoothing Splines for Discontinuous Signals

Martin Storath<sup>a</sup> and Andreas Weinmann<sup>b</sup>

<sup>a</sup>Lab for Mathematical Methods in Computer Vision and Machine Learning, Technische Hochschule Würzburg-Schweinfurt, Schweinfurt, Germany;  
<sup>b</sup>Department of Mathematics and Natural Sciences, Hochschule Darmstadt, Darmstadt, Germany

### ABSTRACT

Smoothing splines are twice differentiable by construction, so they cannot capture potential discontinuities in the underlying signal. In this work, we consider a special case of the weak rod model of Blake and Zisserman that allows for discontinuities penalizing their number by a linear term. The corresponding estimates are cubic smoothing splines with discontinuities (CSSD) which serve as representations of piecewise smooth signals and facilitate exploratory data analysis. However, computing the estimates requires solving a non-convex optimization problem. So far, efficient and exact solvers exist only for a discrete approximation based on equidistantly sampled data. In this work, we propose an efficient solver for the continuous minimization problem with non-equidistantly sampled data. Its worst case complexity is quadratic in the number of data points, and if the number of detected discontinuities scales linearly with the signal length, we observe linear growth in runtime. This efficient algorithm allows to use cross-validation for automatic selection of the hyperparameters within a reasonable time frame on standard hardware. We provide a reference implementation and supplementary material. We demonstrate the applicability of the approach for the aforementioned tasks using both simulated and real data. Supplementary materials for this article are available online.

### ARTICLE HISTORY

Received June 2022

Accepted September 2023

### KEYWORDS

Algorithms; Nonparametric regression; Numerical optimization; Penalized optimization; Signal processing

## 1. Introduction

Assume that we are given the approximate values  $y_i = g(x_i) + \epsilon_i$  of a function  $g$  at data sites  $x_1, \dots, x_N$ , and an estimate  $\delta_i$  of the standard deviation of the errors  $\epsilon_i$  which are uncorrelated with zero mean. If we can assume that  $g$  is a smooth function, then we may try to estimate  $g$  using a (cubic) smoothing spline, a flexible and widely applicable approach to curve estimation (Silverman 1985; Hastie et al. 2009). Using the notations of De Boor (2001), a cubic smoothing spline is the solution  $\hat{f}$  of the variational problem

$$\min_{f \in C^2} p \sum_{i=1}^N \left( \frac{y_i - f(x_i)}{\delta_i} \right)^2 + (1-p) \int_{x_1}^{x_N} (f''(t))^2 dt. \quad (1)$$

Here, the minimum is taken over all twice continuously differentiable functions on  $[x_1, x_N]$ . The minimizer is a compromise between the conflicting goals smoothness, measured by the squared Euclidean norm of the derivative, and closeness to data. The “stiffness” parameter  $p \in (0, 1)$  controls the relative weight of the two goals. It is well known that the minimizer  $\hat{f}$  of (1) is a cubic spline, that is,  $\hat{f}$  is a piecewise cubic polynomial function which is twice continuously differentiable (De Boor 2001). If the underlying function  $g$  has discontinuities (or breaks/jumps), or in other words,  $g$  is only piecewise smooth, the classical smoothing spline cannot capture these discontinuities.

Signals with discontinuities appear in numerous applications; for example, the cross-hybridization of DNA (Snijders et al. 2001; Drobyshev et al. 2003; Hupé et al. 2004), the reconstruction of brain stimuli (Winkler et al. 2005), single-molecule analysis (Joo et al. 2008; Loeff et al. 2021), cellular ion channel functionalities (Hotz et al. 2013), photo-emission spectroscopy (Frick, Munk, and Sieling 2014) and the rotations of the bacterial flagellar motor (Sowa et al. 2005); see also Kleinberg and Tardos (2006) and Frick, Munk, and Sieling (2014) for further examples.

We are interested in the case where the locations of the discontinuities are unknown. (If the locations of the discontinuities are known a priori, one simply may compute smoothing splines on the intervals between two points of discontinuities.) In their landmark work, Blake and Zisserman (1987) proposed a variational model for this task based on piecewise regression with smoothing splines and linear penalties on the number of jumps and creases, called the *weak rod model*. Here, we study the special case of the weak rod model without creases, and for discrete, potentially non-equidistant, sampling points. Using the above notation of smoothing splines, it can be formulated as

$$\min_{f, J} p \sum_{i=1}^N \left( \frac{y_i - f(x_i)}{\delta_i} \right)^2 + (1-p) \int_{[x_1, x_N] \setminus J} (f''(t))^2 dt + \gamma |J|. \quad (2)$$

Here the minimum is taken over all possible sets of discontinuities between two data sites  $J \subset [x_1, x_N] \setminus \{x_1, \dots, x_N\}$  and all

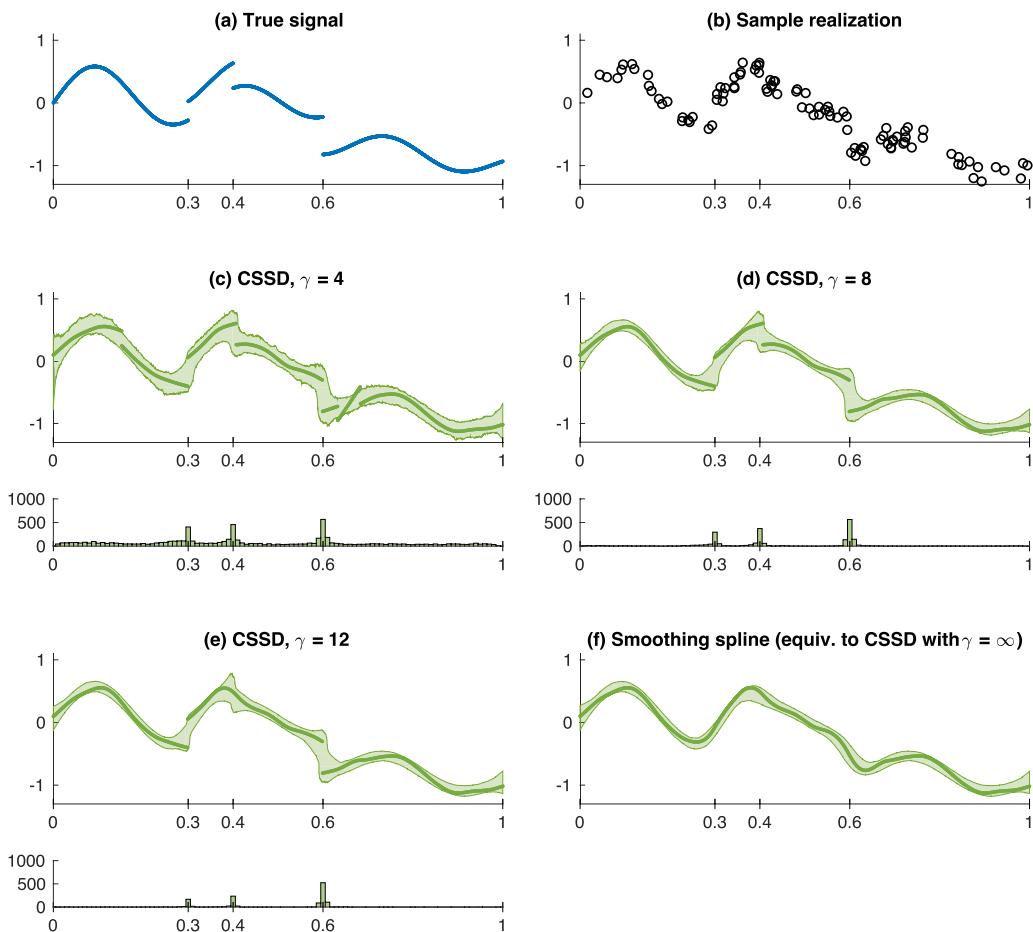
functions  $f$  that are twice continuously differentiable away from the discontinuities. The last term is a penalty for the number of discontinuities  $|J|$  weighted by a parameter  $\gamma > 0$ . In (2), we search for a global minimizer which consists of a discontinuity set  $\hat{J}$  and a piecewise cubic spline  $\hat{f}$  with discontinuities at  $\hat{J}$ . The solution  $\hat{f}$  of the specific minimization problem (2) is a *cubic smoothing spline with discontinuities* which we abbreviate as CSSD. A first example of a CSSD is provided in Figure 1. A CSSD may be used for exploratory data analysis, as predictor, and as function estimator; just like a classical smoothing spline (Silverman 1985), but with the extension that the underlying function may have discontinuities. Additionally, a detected discontinuity can be seen as a type of changepoint.

The considered instance of the weak rod model (2) is interesting and useful on its own, because it comprises two widely used regression models: For sufficiently large  $\gamma$ , a CSSD coincides with the  $C^2$  continuous cubic smoothing spline. In the limit  $p \rightarrow 0$ , a CSSD tends to a piecewise linear regression function, which were studied for example in Kleinberg and Tardos (2006), Friedrich et al. (2008), Storath, Kiefer, and Weinmann (2019). If the second order derivative is replaced by the first order derivative in (2), we obtain the Mumford-Shah model (Mumford and Shah 1989), and the according limit  $p \rightarrow 0$  leads to a piecewise constant regression model, also referred to as Potts

model (Winkler and Liebscher 2002; Winkler 2003), and studied in a series of further works, including the works of Jackson et al. (2005), Boysen et al. (2009), and Killick, Fearnhead, and Eckley (2012), to mention only a few.

The literature describes algorithms for a variant of (2) with a discretized roughness penalty for equidistantly sampled data sites, that is  $x_{i+1} - x_i = \text{const}$  for all  $i$ . For solving this discrete problem, Blake and Zisserman (1987) proposed a graduated non-convexity algorithm, an algorithm based on Hopfield's neural model and a Viterbi-type algorithm but these algorithms are not guaranteed to yield a global minimum in general. Straightforward adaption of the slightly different dynamic programming approaches of Blake (1989) or Winkler and Liebscher (2002), one may obtain a global minimizer in cubic worst case time complexity. Storath, Kiefer, and Weinmann (2019) proposed an algorithm which solves the discretized problem for equidistantly sampled data in quadratic complexity.

However, restriction to equidistantly sampled data means a major limitation because (i) many types of data can be acquired only with varying distances of the measurements sites, and (ii) even when the data sites are equidistantly sampled, non-equidistant data points occur naturally when performing  $K$ -fold cross-validation for parameter selection. The present formulation with continuous roughness penalty (2) naturally deals with



**Figure 1.** A synthetic signal is sampled at  $N = 100$  random data sites  $x_i$  and corrupted by zero mean Gaussian noise with standard deviation 0.1. The results of the discussed model are shown for  $p = 0.999$  and different parameters of  $\gamma$ , where  $\gamma = \infty$  corresponds to classical smoothing splines. The thick lines represent the results of the shown sample realization. The shaded areas depict the 2.5% to 97.5% (pointwise) quantiles of 1000 realizations. The histograms under the plots show the frequency of the detected discontinuity locations over all realizations.



non-equidistant data sites  $x_i$ , but solving it is more challenging than the discrete equidistant setting. There is—to the authors knowledge—no dedicated paper presenting an exact solver for (2). After the reformulation to a reduced search space (see Section 2.2), one may combine methods of dynamic programming (Blake 1989; Winkler and Liebscher 2002; Killick, Fearnhead, and Eckley 2012) and smoothing splines (Reinsch 1967), to obtain an exact solver for (2). However, such a “baseline” solver has worst case cubic time complexity, and hence applying it is costly even for problems of moderate size. This hampers in particular usage of standard parameter selection strategies, such as cross-validation, since they require solving (2) for numerous model parameters  $\gamma$  and  $p$ . This article develops an efficient exact solver for (2). Details on the contribution of this article follow after a further discussion of prior and related work.

### 1.1. Prior and Related Work

Blake and Zisserman (1987) have introduced the weak rod model in the context of image processing along with a two-dimensional generalization which they called the weak plate model. Equation (2) is a special case of the weak rod model in the sense that it does not allow creases (corresponding to the weak rod model with a sufficiently high crease penalty), and that it is formulated with a potentially weighted discrete data term which naturally appears when working with sampled data. Blake and Zisserman (1987) used the penalty factors  $\mu, \alpha > 0$  which are related to the factors  $p, \gamma$  in (2) by  $\mu = \sqrt[4]{(1-p)/p}$  and  $\alpha = \gamma/p$ . (This article adopts the  $p$ -parameterization used in a standard reference on splines by De Boor (2001).) Blake and Zisserman (1987) have obtained results on scale and sensitivity in discontinuity detection for the fully continuous version. In particular, they derived a contrast threshold referring to the minimum detectable jump height for an isolated bi-infinite step function in dependence of the model parameters which is given by  $h_0 = 2^{\frac{3}{4}} \sqrt{\alpha/\mu}$ .

If we replace the second order derivative with the first order derivative in (2), we obtain the weak string model which coincides with the one-dimensional case of the famous Mumford-Shah model for edge preserving smoothing (Mumford and Shah 1985, 1989). This first order model has been investigated in more detail in the literature than the second order model; in particular, an exact algorithm of cubic complexity for the discretized first order model has been proposed in an early work by Blake (1989). Blake and Zisserman (1987) have described a fundamental limitation of the first order model termed the gradient limit; it refers to the undesired introduction of spurious discontinuities when the gradient exceeds a threshold. The second order model considered here does not have this limitation.

The present model is closely connected to smoothing splines which were developed in the works of Schoenberg (1964) and Reinsch (1967), and the basic idea can be tracked back to the work of Whittaker (1923). Silverman (1984) has shown that spline smoothing approximately corresponds to smoothing by a kernel method with bandwidth depending on the local density of the data sites. Comprehensive treatments on smoothing splines are given in the books by De Boor (2001), by Wahba (1990), and by Green and Silverman (1993), as well as in the paper

by Silverman (1985). (In this article, we use the notations and conventions of De Boor 2001.) The signal and image processing point of view is discussed in the papers of Unser (1999, 2002). Smoothing splines have become a standard method for statistical data processing, and they are discussed in standard literature on the topic (Hastie et al. 2009). Recent contributions on splines in a statistical context are generalizations for Riemannian manifolds (Kim et al. 2021), locally adaptive splines for Bayesian model selection in additive partial linear models (Jeong, Park, and van Dyk 2021), knot estimation for linear splines, (Yang, Zhang, and Zhang 2021), and efficiency improvements for splines with multiple predictors (Meng et al. 2022), to mention only a few.

Besides the above mentioned models of spline-type, many other piecewise regression models with discontinuity or jump penalty have been used. They typically lead to combinatorial optimization problems, and fast solvers are a central question (Winkler and Liebscher 2002; Killick, Fearnhead, and Eckley 2012; Frick, Munk, and Sieling 2014). Efficient algorithms for piecewise constant or piecewise polynomial regression functions have been proposed by Auger and Lawrence (1989), Winkler and Liebscher (2002), Jackson et al. (2005), Friedrich et al. (2008), Little and Jones (2011b), and Killick, Fearnhead, and Eckley (2012), and extended to indirectly measured data (Weinmann and Storath 2015; Storath, Weinmann, and Demaret 2014) and to nonlinear data spaces (Storath, Weinmann, and Unser 2017; Weinmann, Demaret, and Storath 2016). Parallelized versions have been considered by Tickle et al. (2020). For the piecewise constant case, Boysen et al. (2009) have obtained consistency and convergence rates of the estimates in  $L^2([0, 1])$ . A detailed treatment of piecewise constant regression is given by Little and Jones (2011a, 2011b).

Regarding automatic selection of the model parameters, there are some common approaches for related piecewise regression models; for example information based criteria (Zhang and Siegmund 2007; Yao 1988), an interval criterion (Winkler et al. 2005) and several variants of cross-validation (Arlot and Celisse 2011). For classical smoothing splines, generalized cross-validation (Silverman 1985; Craven and Wahba 1979; Golub, Heath, and Wahba 1979) is frequently used. The authors are not aware of an automatic parameter selection strategy specifically developed for the considered model (2).

There are also conceptually different methods for estimating discontinuous regression functions of spline type. One such approach was proposed by Koo (1997) where knots are stepwise refined by knot addition, basis deletion, knot-merging, and the final model is determined by the Bayes information criterion. Another possibility is a changepoint based approach: one might first use an existing method for changepoint detection to determine discontinuity locations; for example the CUSUM method (Page 1954), Bayesian changepoint inference (Fearnhead 2006), wild binary segmentation (Fryzlewicz 2014), a narrowest over the threshold method (Baranowski, Chen, and Fryzlewicz 2019), a Bayesian ensemble approach (Zhao et al. 2019), nonparametric maximum likelihood approaches (Zou et al. 2014; Haynes, Fearnhead, and Eckley 2017), or multiscale testing (Frick, Munk, and Sieling 2014) could be used. (See van den Burg and Williams (2020) and the references therein for an overview and a comparison of selected changepoint detection methods.) Then, splines

can be fitted between two detected changepoints; either smoothing splines directly to the data or interpolating splines to the corresponding signal estimates. We point out that such a two stage approach emphasizes the changepoint detection aspect, and as splines are typically not involved in the detection process, they are not necessarily the “natural” models for the data between two changepoints.

For denoising nonsmooth signals, shrinkage of wavelet coefficients is frequently used (Donoho and Johnstone 1994). The book by Mallat (2008) provides an overview on wavelet shrinkage. By their multiscale subdivision algorithms, wavelet methods are computationally extremely efficient. In contrast to the model considered in this work, they typically rely on equidistant sampling points and do not result in piecewise smooth regression functions.

## 1.2. Contribution

We first discuss basic properties of a CSSD, that is, the properties of the minimizers of (2). In particular, in view of the well-definedness of the estimator, we provide uniqueness results for the optimization problem with respect to both function evaluations and partitions in an almost everywhere sense.

The main contribution of this article is an efficient algorithm for computing a solution of the problem (2), meaning a global minimum of the target function. The algorithm is developed in three steps: (i) We show that we may restrict the search space for the discontinuity set  $J$  to the midpoints of the data sites, and we use dynamic programming to reduce the number of possible configurations. (ii) We propose a procedure that computes the spline energies for a signal of increasing length in constant time per new element. It allows to obtain the required spline energies on all (discrete) intervals efficiently. We point out that using the smoothing spline algorithms of Reinsch (1967) and of De Hoog and Hutchinson (1986) lead to significantly higher computational costs for this specific task. (iii) We show that the computation of the spline energies is compatible with two different pruning strategies, the PELT pruning of Killick, Fearnhead, and Eckley (2012) and the FPVI pruning of Storath and Weinmann (2014). The worst case time complexity of the proposed algorithm is  $O(N^2)$  where  $N$  is the number of data points. If the number of detected discontinuities scales linearly with  $N$ , we observe linear scaling of the runtime.

We provide a ready-to-use reference implementation in Matlab, available for download on Github.<sup>1</sup> The novel solver is notably faster than a baseline solver which relies on standard Python modules and which does not use of the methods developed in the present work. We further implement a strategy for selecting the model parameters ( $p$  and  $\gamma$ ) automatically based on  $K$ -fold cross-validation. The proposed fast algorithm allows to perform this selection strategy within a reasonable time frame on standard hardware. Numerical examples with synthetic and real data demonstrate the potential of CSSD as function estimator for discontinuous signals, as basis for a changepoint detector, and as a tool for exploratory data analysis.

<sup>1</sup><https://github.com/mstorath/CSSD>

## 2. Efficient Computation of a CSSD and Uniqueness Result

### 2.1. Reformulation and Basic Properties of the Solutions

Throughout this article, we assume that the data sites satisfy  $x_1 < x_2 < \dots < x_N$ . If the data does not satisfy this constraint, we may merge data sites into a single data point by weighted averaging over the  $y$ -values of coinciding  $x$ -values; see for example Hutchinson (1986).

Let  $J \subset [x_1, x_N] \setminus \{x_1, \dots, x_N\}$  be a set of discontinuities of size  $|J| = K$ . It is convenient to sort the elements of  $J$  ascendingly so that we can access the elements by an index, for example  $J_1$  for the smallest element in  $J$ . Its complement in  $[x_1, x_N]$  consists of  $K + 1$  half open and open intervals, denoted by  $I_1, \dots, I_{K+1}$

$$[x_1, x_N] \setminus J = I_1 \cup \dots \cup I_{K+1}.$$

We denote the (ordered) set of these intervals by  $\mathcal{P}(J)$

$$\mathcal{P}(J) = \{I_1, \dots, I_{K+1}\}.$$

The minimization problem (2) now can be reformulated equivalently in terms of the discontinuity set and the corresponding intervals as

$$\min_{J \subset [x_1, x_N] \setminus \{x_1, \dots, x_N\}} \sum_{I \in \mathcal{P}(J)} \mathcal{E}_I + \gamma |J| \quad (3)$$

where  $\mathcal{E}_I$  is given by

$$\mathcal{E}_I = \min_{f \in C^2(I)} p \sum_{i: x_i \in I} \left( \frac{y_i - f(x_i)}{\delta_i} \right)^2 + (1-p) \int_I (f''(t))^2 dt. \quad (4)$$

As in (1), the model parameter  $p \in (0, 1)$  controls the relative weight of the smoothness term and the data fidelity term, and  $\delta_i$  is an estimate of the standard deviation of the errors at data site  $x_i$ . Note that (4) describes the minimum energy of a classical smoothing spline on the interval  $I$ .

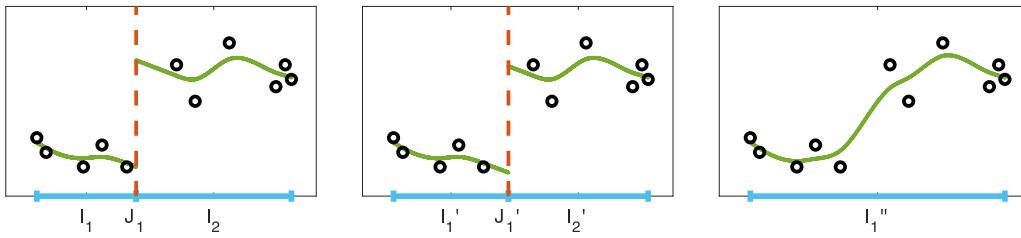
A standard procedure to obtain  $\mathcal{E}_I$  consists in computing a minimizer  $f_I$  of the functional in (4) using the algorithm of Reinsch (1967) on the data in the interval  $I$ , and evaluating (4) with that  $f_I$ . The computational costs of this approach are linear in the number of data points falling in the interval  $I$ . A key element of the fast algorithm we are going to propose in the next section is that it bypasses the costly explicit computation of the  $f_I$ , and computes  $\mathcal{E}_I$  directly using a recurrence relation.

From (3) we see that every discontinuity set  $J$  has a unique piecewise cubic spline function  $f_J$  which is defined for all  $I \in \mathcal{P}(J)$  by the unique minimizer of (4). So once an optimal discontinuity set is known, the corresponding  $f_J$  can be computed by a standard method such as Reinsch's algorithm. In opposite direction,  $J$  consists of the discontinuities of  $f$  and the discontinuities of  $f'$ .

Next we state two basic properties of an optimal discontinuity set  $\hat{J}$ .

**Lemma 1.** Let  $\hat{J}$  be a solution of the minimization problem (3).

1.  $\hat{J}$  has at most one element between two adjacent data sites and
2.  $\hat{J}$  has at most  $\lceil N/2 \rceil - 1$  elements.



**Figure 2.** A toy dataset (circles) and three possible discontinuity configurations (visualized on abscissa) along with their corresponding piecewise splines (solid curves) are shown ( $x$  on abscissa,  $y$  on ordinate). The first two configurations give the same function value  $\mathcal{E}_{I_1} + \mathcal{E}_{I_2} + \gamma = \mathcal{E}_{I'_1} + \mathcal{E}_{I'_2} + \gamma$ , when plugged into the target function in (2). Hence, these configurations are equally good in the sense of the model (2). So we consider the discontinuity locations  $J_1$  and  $J'_1$  as equivalent and use the midpoint (here  $J'_1$ , middle tile) as natural representative. The third tile shows a configuration with no discontinuity which yields the target function value  $\mathcal{E}_{I''_1}$ . If  $\gamma$  is sufficiently small ( $\gamma < \mathcal{E}_{I''_1} - \mathcal{E}_{I'_1} - \mathcal{E}_{I'_2}$ ) the configurations with discontinuity have a better target function value than the one without discontinuity.

The proof is given in the supplementary material.

We develop an efficient solver for (3)—the equivalent formulation of (2)—in three steps: (i) reduction of the search space for the discontinuities to a discrete set and reducing the number of configurations using dynamic programming, (ii) efficient update scheme for the necessary spline energies, (iii) search space pruning which is compatible with the order of computation of the energy updates.

## 2.2. Reduction of the Search Space for the Discontinuity Set

*Reducing the discontinuity set to midpoints of data sites.* It is a basic property of the classical smoothing spline that its energy  $\mathcal{E}_I$  (defined in (4)) only depends on the data sites that are contained in the interval  $I$ . The reason for this is that the spline can be extended linearly beyond the extremal data sites without increasing the energy; see for example Silverman (1985). So if  $I$  and  $I'$  are two intervals which contain the same data sites, that is, if  $\{x_i : x_i \in I\} = \{x_i : x_i \in I'\}$ , then  $\mathcal{E}_I = \mathcal{E}_{I'}$ . As a direct consequence, a shift of a discontinuity location between two data sites  $x_i, x_{i+1}$  does not change the functional value (3). So a discontinuity may be located at an arbitrary position between two data points. Having noticed this, we consider two discontinuity sets as equivalent, if all their respective elements lie between the same data sites. The midpoint between two data sites is a natural choice for a representative of the equivalence class. This is illustrated in Figure 2 using a toy example. Hence, it is sufficient to consider the discrete set of midpoints  $\mathcal{M}$  of the data sites,

$$\mathcal{M} := \{\frac{1}{2}(x_i + x_{i+1}) : i = 1, \dots, N-1\}, \quad (5)$$

as search space for  $J$ . Thus, we have reduced the optimization problem (3) to the discrete optimization problem

$$\min_{J \subset \mathcal{M}} \sum_{I \in \mathcal{P}(J)} \mathcal{E}_I + \gamma |J|. \quad (6)$$

Because  $\mathcal{M}$  is a finite set it follows that the minimization problem (6), and so the problems (2) and (3), indeed have a minimizer.

**Remark 1.** For the algorithm, which we are developing here, the concrete representative of the discontinuity set is irrelevant. The choice of the representative comes into play when displaying

the final result. For this purpose, it seems natural to take the midpoints. Yet, there might be scenarios where other points—or even the entire interval—are reasonable representatives for the discontinuity locations. (One may think of a similar situation when dealing with the ordinary median which can be defined as  $\operatorname{argmin}_\mu |y_i - \mu|$ . For data of even length, all values between the two central data points satisfy the minimality property, and most commonly, the mean value between the central points is chosen. However, for implementing the method of Weinmann, Storath, and Demaret (2015), considering the upper and lower points as median turned out to be useful.)

*Reducing the number of configurations by dynamic programming.* The reduced form (6) constitutes a discrete one-dimensional partitioning problem which can be solved by dynamic programming. The approach works similar to the corresponding algorithms on related partitioning problems; see Friedrich et al. (2008), Kleinberg and Tardos (2006), Winkler and Liebscher (2002), and Jackson et al. (2005). For completeness we briefly describe the procedure. For the formulation it is convenient, to identify an interval  $I$  with the indices of the contained data sites. For example, the interval that contains the data sites  $x_l, x_{l+1}, \dots, x_r$  is identified with the (ordered) set of indices  $\{l, l+1, \dots, r\}$ , or abbreviated in Matlab-type notation  $\{l:r\}$ . In particular, we write  $\mathcal{E}_{\{l:r\}}$  for  $\mathcal{E}_I$ .

We now consider the functional in (6) on a reduced dataset  $(x_1, y_1), \dots, (x_r, y_r)$  for  $r \leq n$  and denote it by  $F_r$ ; that is  $F_r(J) = \sum_{I \in \mathcal{P}(J)} \mathcal{E}_I + \gamma |J|$ , where  $J \subset \mathcal{M} \cap [x_1, x_r]$  and  $\mathcal{E}_I$  is only evaluated for the data points  $x_1, \dots, x_r$ . The minimal functional value of  $F_r$ , denoted by  $F_r^* = \min_J F_r(J)$ , satisfies the Bellman equation

$$F_r^* = \min_{l=1, \dots, r} \left\{ \mathcal{E}_{\{l:r\}} + \gamma + F_{l-1}^* \right\} \quad (7)$$

where we let  $F_0^* = -\gamma$ , see Friedrich et al. (2008). As  $\mathcal{E}_{\{l:r\}} = 0$  for  $r-l \leq 1$  and  $F_l^*$  nondecreasing in  $l$ , the minimum on the right hand side actually only has to be taken over the values  $l = 1, \dots, r-1$ . By the dynamic programming principle, we successively compute  $F_1^*, F_2^*, \dots$ , until we reach  $F_N^*$ . As our primary interest are the optimal discontinuities  $J$ , rather than the minimal functional values  $F_N^*$ , we keep track of these locations. An economic way to do so is to store at step  $r$  the minimizing argument  $l^*$  of (7) as the value  $Z_r$  so that  $Z$  encodes the boundaries of an optimal partition. We refer to Friedrich et al. (2008) for a more detailed description of the data structure

and a visualization. We record that the number of configurations that have to be checked for computing the minimum value is  $O(N^2)$ .

### 2.3. Efficient Computation of the Spline Energies for All Intervals

We next develop an efficient procedure to compute the functional values (4) for all discrete intervals  $\{l : r\}$  with  $1 \leq l \leq r \leq N$ . Before starting the development, we discuss why we need a specialized method for this specific task: The algorithm of Reinsch (1967) aims for computing the minimizing argument of (4), and the minimum value has to be computed from that minimizer. This procedure needs linear time for a single value of  $\mathcal{E}_{\{l,r\}}$ . The method described by De Hoog and Hutchinson (1986) gives direct access to the minimum energy  $\mathcal{E}_{\{l,r\}}$  which is achieved by using orthogonal transformations. However, it is designed for fast computation of the energies for different  $p$ -parameters for signals of fixed length; no efficient algorithm for signals of increasing lengths is given. Hence, computing a single  $\mathcal{E}_{\{l,r\}}$  needs linear time as well. Either approach would result in an algorithm of cubic complexity when used for computing all values  $\mathcal{E}_{\{l,r\}}$ .

We now develop a specialized method that updates the spline energy on the interval  $\{l : r\}$  to that on the interval  $\{l : r + 1\}$  in constant time. To keep the notation simple, we describe the procedure for  $l = 1$  noting that the procedure works analogously for any other  $l = 2, \dots, N$ .

Consider the functional in (4) for an interval  $I$  which contains exactly the data sites  $x_1, \dots, x_r$ , where  $2 \leq r \leq N-1$ , and denote its unique minimizing argument by  $\hat{f}_I$ . The solution  $\hat{f}_I$  is a cubic spline with natural boundary conditions which means that  $p_i := \hat{f}_I|_{[x_i, x_{i+1}]}$  is a polynomial of degree at most 3 for all  $i = 1, \dots, r-1$ , that  $\hat{f}_I$  is in  $C^2(I)$ , and that  $\hat{f}_I''(x_1) = \hat{f}_I''(x_r) = 0$  (De Boor 2001, chap. XIV). Let  $f, f' \in \mathbb{R}^r$  be the vector of Hermite control points of  $\hat{f}_I$ , meaning that  $f_i := \hat{f}_I(x_i)$  and  $f'_i := \hat{f}'_I(x_i)$ , for  $i = 1, \dots, r$ . Each polynomial  $p_i$  is uniquely determined by  $f_i, f'_i, f_{i+1}$ , and  $f'_{i+1}$ ; see for example (Cheney 1998, p. 61). Hence,  $\hat{f}_I$  and its  $2r$  Hermite control points  $f, f'$  are one-to-one. We may express the polynomials  $p_i$  in terms of the Hermite control points as

$$p_i(x) = \sum_{k=0}^3 c_k (x - x_i)^k \quad (8)$$

where  $c_0 = f_i, c_1 = f'_i, c_2 = -\frac{f'_{i+1} + 2f'_i}{d_i} + 3\frac{f_{i+1} - f_i}{d_i^2}, c_3 = \frac{f'_{i+1} + f'_i}{d_i^2} + 2\frac{f_i - f_{i+1}}{d_i^3}$ , and  $d_i = x_{i+1} - x_i$  is the distance between two data sites (Dougherty, Edelman, and Hyman 1989). Using this representation, we obtain the integral of the square of  $p_i''$  on the interval  $[x_i, x_{i+1}]$  in terms of  $f_i, f'_i, f_{i+1}$ , and  $f'_{i+1}$ :

$$\int_{x_i}^{x_{i+1}} (p_i''(x))^2 dx = \frac{12f_i^2}{d_i^3} - \frac{24f_{i+1}f_i}{d_i^3} + \frac{12f_{i+1}^2}{d_i^3} + \frac{12f'_i f_i}{d_i^2} + \frac{12f'_{i+1} f_i}{d_i^2} - \frac{12f'_i f_{i+1}}{d_i^2} - \frac{12f_{i+1} f'_{i+1}}{d_i^2} + \frac{4(f'_i)^2}{d_i} + \frac{4(f'_{i+1})^2}{d_i} + \frac{4f'_i f'_{i+1}}{d_i}.$$

This is a quadratic form which can be written in matrix form as  $\int_{x_i}^{x_{i+1}} (p_i''(x))^2 dx = v_i^T B v_i$ , where  $v_i = [f_i, f'_i, f_{i+1}, f'_{i+1}]$  and

$$B_i = \begin{bmatrix} \frac{12}{d_i^3} & \frac{6}{d_i^2} & -\frac{12}{d_i^3} & \frac{6}{d_i^2} \\ \frac{6}{d_i^2} & \frac{4}{d_i} & -\frac{6}{d_i^2} & \frac{2}{d_i} \\ -\frac{12}{d_i^3} & -\frac{6}{d_i^2} & \frac{12}{d_i^3} & -\frac{6}{d_i^2} \\ \frac{6}{d_i^2} & \frac{2}{d_i} & -\frac{6}{d_i^2} & \frac{4}{d_i} \end{bmatrix}.$$

The matrix  $B_i$  is symmetric and positive semidefinite with two zero eigenvalues. Invoking a semidefinite variant of the Cholesky decomposition we obtain the factorization

$$B_i = U_i^T U_i, \quad \text{with } U_i = \begin{bmatrix} \frac{2\sqrt{3}}{d_i^{3/2}} & \frac{\sqrt{3}}{\sqrt{d_i}} & -\frac{2\sqrt{3}}{d_i^{3/2}} & \frac{\sqrt{3}}{\sqrt{d_i}} \\ 0 & \frac{1}{\sqrt{d_i}} & 0 & -\frac{1}{\sqrt{d_i}} \end{bmatrix} \in \mathbb{R}^{2 \times 4}. \quad (9)$$

For the next step, it is convenient to decompose  $U_i$  as  $U_i = [V_i, W_i]$  with  $V_i, W_i \in \mathbb{R}^{2 \times 2}$  and to define  $e_1^T = [1, 0] \in \mathbb{R}^{1 \times 2}$ . Let us define  $A^{(r)} \in \mathbb{R}^{2r \times s}$ , with  $s = 3r - 2$ , by

$$A^{(r)} = \begin{bmatrix} \alpha_1 e_1^T & 0 & 0 & \cdots & 0 \\ \beta V_1 & \beta W_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 e_1^T & 0 & \cdots & 0 \\ 0 & \beta V_2 & \beta W_2 & \cdots & 0 \\ 0 & 0 & \alpha_3 e_1^T & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \beta V_{r-1} & \beta W_{r-1} \\ 0 & \cdots & 0 & 0 & \alpha_r e_1^T \end{bmatrix} \quad (10)$$

with  $\alpha_i = \frac{\sqrt{p}}{\delta_i}$  and  $\beta = \sqrt{1-p}$ . Now the optimal functional value  $\mathcal{E}_{\{1:r\}}$  can be expressed as follows:

**Lemma 2.** Let  $\mathcal{E}_{\{1:r\}}$  be given by (4) with  $I = \{1 : r\}$ . Then,

$$\mathcal{E}_{\{1:r\}} = \min_{u \in \mathbb{R}^{2r}} \|A^{(r)} u - \tilde{y}^{(r)}\|_2^2, \quad (11)$$

where  $A^{(r)}$  is defined in (10), and  $\tilde{y}^{(r)} \in \mathbb{R}^s$  is a vector of zeros except  $\tilde{y}_{3i-2}^{(r)} = \alpha_i y_i$  for  $i = 1, \dots, r$ .

The proof is given in the supplementary material.

The form (11) has a key property for our purposes: it is a least squares problem in matrix form so that  $A^{(r)}$  is a submatrix of  $A^{(r+1)}$  and  $\tilde{y}^{(r)}$  is a subvector of  $\tilde{y}^{(r+1)}$ . This submatrix relation allows us to update the QR-decomposition of the system  $[A^{(r)} | \tilde{y}^{(r)}]$  to the QR-decomposition of the system  $[A^{(r+1)} | \tilde{y}^{(r+1)}]$  by a constant number of Givens rotations. Let  $Q \in \mathbb{R}^{s \times s}$  orthogonal and  $R \in \mathbb{R}^{s \times r}$ , with  $R_{1:r, 1:r}^T$  upper triangular and  $R_{r+1:s, 1:r} = 0$ , such that  $A^{(r)} = QR$ , and let  $z = Q^T \tilde{y}^{(r)}$ . Then the minimizer of (11) is the solution of the first  $r$  rows of the system

$$[R|z] \quad (12)$$

and  $\mathcal{E}_{\{1:r\}}$  is equal to the sum of squares of the last entries,  $r+1, \dots, s$ , of the right hand side  $Q^T \tilde{y}^{(r)}$ . Passing from  $r$  to  $r+1$ , extends the system by three rows and two columns:

$$\left[ \begin{array}{cc|c} R_{1:s, 1:r-2} & R_{1:s, r-1:r} & 0 \\ 0 & \beta V_{r+1} & \beta W_{r+1} \\ 0 & 0 & \alpha_{r+1} e_1^T \end{array} \right] \quad (13)$$

This system can be brought to upper triangular form using Givens rotations. By the upper triangular form of  $R$ , the Givens rotations only act on the small subsystem

$$\left[ \begin{array}{cc|c} R_{r-1:r,r-1:r} & 0 & z_{r-1:r} \\ \beta V_{r+1} & \beta W_{r+1} & 0 \\ 0 & \alpha_{r+1} e_1^T & \alpha_{r+1} y_{r+1} \end{array} \right] \quad (14)$$

where the left hand side is in  $\mathbb{R}^{5 \times 4}$ . Having applied the Givens rotations, the system has the form  $[R'|z']$  with  $R' \in \mathbb{R}^{5 \times 4}$ ,  $R'_{1:4,1:4}$  upper triangular and  $R'_{5,1:4} = 0$ . So the residual is coded in the last entry of  $z'$ . Hence, we get the update

$$\mathcal{E}_{\{1:r+1\}} = \mathcal{E}_{\{1:r\}} + (z'_5)^2. \quad (15)$$

The update procedure is initialized (for the index  $r = 2$ ) by  $\mathcal{E}_{\{1:2\}} = 0$ , and the system describing the initial state is given  $[R|Q^T \tilde{y}^{(2)}]$  where  $QR = A^{(2)}$  is the QR-decomposition of  $A^{(2)} \in \mathbb{R}^{4 \times 4}$ .

The above derivation leads to the following theorem.

**Theorem 3.** The procedure described above ((12) to (15)) computes the array  $[\mathcal{E}_{\{1:1\}}, \mathcal{E}_{\{1:2\}}, \dots, \mathcal{E}_{\{1:N\}}]$  in  $O(N)$  time complexity and  $O(N)$  memory complexity.

The proof is given in the supplementary material.

For any other starting index  $l$  with  $1 < l \leq r$  the values  $\mathcal{E}_{\{l:r\}}$  are computed in the same fashion by calling the above procedure with the starting index  $l$ .

A note on the practical implementation seems useful at this point: The described procedure uses Givens rotations for elimination. Unfortunately, by their rowwise action, applying Givens rotations sequentially turns out to be relatively slow when implemented in standard linear algebra packages as used by Matlab. However, as the Givens rotations applied to (13) only act on the subsystem (14), we may use Householder reflections directly on (14). This allows to use standard Householder-based QR decompositions. Another possibility is a hardware-near implementation of the Givens rotation. For maintainability and readability reasons, we decided to use the standard linear algebra system as used by Matlab for the reference implementation.

If the sampling distances between adjacent data points  $d_i$  have very different orders of magnitude, so that the global mesh ratio  $(\max_{i=1,\dots,N-1} d_i) / (\min_{i=1,\dots,N-1} d_i)$  is large, the matrix  $A^{(r)}$ , defined by (10), may have a large condition number. Then, solving (11) may get numerically unstable. In such cases, one may try data binning (meaning merging the data points with the smallest distances) to reduce the global mesh ratio.

## 2.4. Compatibility with Pruning Strategies

In the supplementary material, we show that the fast update strategy is compatible with the specific order of the computation of two different pruning strategies, the PELT strategy from Killick, Fearnhead, and Eckley (2012) and the pruning from Storath and Weinmann (2014), abbreviated here by FPVI. Comparing the two strategies, the effectiveness of FPVI essentially grows with the value of  $\gamma$  whereas the effectiveness of PELT essentially grows with the number of detected discontinuities.

## 2.5. Overall Algorithm and Computational Effort

Putting together the procedures described in the last sections, we obtain an algorithm which computes a solution  $\hat{J}$  of the minimization problem (3); that is,  $\hat{J}$  is a global minimizer of the target function in (3). Having computed the optimal discontinuity set  $\hat{J}$ , the corresponding CSSD  $\hat{f}$  is computed using the algorithm of Reinsch (1967). This completes the solution of (2). Altogether, we obtain:

**Theorem 4.** The algorithm described above (Sections 2.2–2.5) computes a solution of (2), that is, a global minimizer of the target function in (2). The worst case time complexity is  $\mathcal{O}(N^2)$  and the memory complexity is  $\mathcal{O}(N)$ .

The proof is given in the supplementary material.

For details on the implementation, we refer to the provided reference implementation, see Section 3.1.

We point out that the scaling in runtime is often more favorable than the worst case scenario. This is an effect of the pruning strategies described in Section 2.4. We observed in our experiments that the runtime scales approximately linearly if the number of detected discontinuities grows linearly with the signal's length. If there are no or very few detected discontinuities the runtime grows approximately quadratically. We refer to the experimental section for an illustrating example. Further, Killick, Fearnhead, and Eckley (2012) made similar observations for piecewise estimators and gave theoretical justifications for the PELT pruning strategy.

**Remark 2.** It is straightforward to extend the method to work for vector-valued data meaning that  $y_i \in \mathbb{R}^D$  for all  $i = 1, \dots, N$ . In this case,  $\mathcal{E}_{\{l:r\}}$  is simply the sum of the energies over all vector components  $\sum_{j=1}^D \mathcal{E}_{\{l:r\},j}$ , where  $\mathcal{E}_{\{l:r\},j}$  is the spline energy of the  $j$ th component of  $y$  on  $\{l : r\}$ . It is clear that the computational effort scales linearly with the dimension  $D$  (See supplementary material for a numerical example).

## 2.6. Uniqueness Result

As discussed in Section 2.2, the precise location of discontinuities between data sites does not influence the penalty function, so that a solution can only be unique up to shifts of discontinuities in between the data sites. Having reduced the potential discontinuities to midpoints (see (6)), the optimal discontinuity set  $\hat{J}$  may still not be unique. A simple example is the data  $x = [0, 1, 2]$  and  $y = [0, 1, 0]$ . For sufficiently small  $\gamma > 0$ , both  $J = \{\frac{1}{2}\}$  and  $\hat{J} = \{\frac{3}{2}\}$  are optimal because they result in the same target functional value  $\gamma$ . A similar situation is found whenever the signal has odd length and has  $\lceil N/2 \rceil - 1$  discontinuities. As it can be seen from the proof of Lemma 1, a solution with  $\lceil N/2 \rceil - 1$  discontinuities is a piecewise linear function interpolating at most two data points. The ambiguities in the examples can be attributed to segments of length two or one, on which a smoothing spline acts interpolatory. If partitions with segments of length less than 3 can be ruled out, we obtain a uniqueness result for a.e. input  $y \in \mathbb{R}^N$  (in a Lebesgue almost everywhere sense). The result is inspired by a related theorem of Wittich et al. (2008) on the piecewise constant model.

**Theorem 5.** We consider the minimization problem (2) for the CSSD in its formulation (6) (with discontinuity set reduced to midpoints). Then, the minimizing values  $f(x_i)$  at the data sites  $x_i$  are uniquely determined for a.e. input  $y \in \mathbb{R}^N$  (w.r.t. Lebesgue measure.) Further, minimizing segments  $I^*$  of length at least three are uniquely determined for (Lebesgue-) a.e. input  $y \in \mathbb{R}^N$ . In particular, the minimizing function  $f$  is unique in  $[x_{\min}, x_{\max}]$  where  $x_{\min}, x_{\max}$  denote the minimal and maximal argument value of the  $x_i$  in the minimizing segment  $I^*$  of length at least three. Consequently, minimizing partitions with all segments being of length at least three and corresponding minimizing functions (outside the intervals with the jumps) are uniquely determined for a.e. input  $y \in \mathbb{R}^N$ .

The proof is given in the supplementary material.

In the present setup, a minimizer of (6) with at least 3 data sites between two discontinuities is obtained if  $\gamma$  is chosen sufficiently large. We note that one could also constrain the search space of (6) w.r.t. the size of the discontinuity sets as an alternative approach.

### 3. Experiments and Parameter Selection

#### 3.1. Implementation and Setup

The proposed algorithm was implemented in Matlab R2022b. For implementation details, we refer to the commented source code provided on Github<sup>2</sup>. The numerical experiments were performed on a laptop with 2.4 GHz 8-Core Intel Core i9, 32 GB RAM. The choice of the model parameters  $p$  and  $\gamma$  depends on the sampling distances on the abscissa, on the scale of the data on the ordinate, and, of course, on the shape of the underlying signal. We set  $\delta_i = \sigma$  for the synthetic experiments, where  $\sigma$  is the standard deviation of the noise, and  $\delta_i = 1$  for the other experiments, for  $i = 1, \dots, N$ .

#### 3.2. Simulated Experiments with a Manually Selected Parameter Set

As the proposed method is designed for estimating smoothly varying functions with discontinuities, we primarily choose test signals with these properties. A classical signal of this class is the “HeaviSine” function used in the work of Donoho and Johnstone (1994). It is given by  $g_2(x) := 4 \sin(4\pi x) - \text{sign}(x - 0.3) - \text{sign}(0.72 - x)$ . As second function of this class, we use the function  $g_1(x) := J_1(20x) + x \cdot 1_{[0.3, 0.4]}(x) - x \cdot 1_{[0.6, 1]}(x)$  as test signal, where  $J_1$  is the Bessel function of the first kind and  $1_S$  is the indicator function on  $S$ . Compared to the HeaviSine, it has one discontinuity more and a varying amplitude. The discontinuity locations of  $g_1$  are chosen such that  $g_1$  shares one discontinuity with  $g_2$  and such that the segments have different lengths. For the experiment shown in Figure 1 (in the introduction), we sampled the signal at  $N = 100$  random points in the interval  $[0, 1]$  (uniformly distributed) and corrupted them by additive zero mean Gaussian noise of standard deviation  $\sigma = 0.1$ . For a classical spline, a visually reasonable result was obtained using

the parameter  $p = 0.999$ . Using this  $p$ -parameter, we report the results for several  $\gamma$ -values.

Experimental results for vector-valued input data are given in the supplementary material.

As the resulting function is a cubic spline with discontinuities it can be described by a piecewise cubic polynomial with discontinuities in  $J$ . Therefore, the result may serve as functional description of the discontinuous signal.

#### 3.3. Computational Costs

Next, we investigate the computational costs of the algorithm. As mentioned in the introduction, there is no freely available software package for solving (2) or its reduced form (6), to the authors knowledge. To assess the benefits of the proposed method, we implemented a baseline solver which relies on standard libraries for solving the reduced form (6). To this end, we combine a dynamic programming approach with the PELT-pruning strategy (Winkler and Liebscher 2002; Killick, Fearnhead, and Eckley 2012), implemented in the Python module *ruptures*<sup>3</sup> (Truong, Oudre, and Vayatis 2020), for computing the optimal partition in (6). The necessary values of  $\mathcal{E}_I$  on each interval (see (4)) were obtained as follows. The smoothing spline module *csaps*<sup>4</sup>, which is described as a Python replication of De Boor’s corresponding Fortran method, was used to compute a minimizer  $\hat{f}_I$  of (4) which in turn was used to compute  $\mathcal{E}_I$  using the corresponding equation given by De Boor (2001). As this baseline solver does not use the fast update scheme developed in this work, its worst case time complexity is  $O(N^3)$  because the number of discrete intervals grows quadratically and the computational effort for a smoothing spline grows linearly in the interval length.

For the runtime comparison, we consider again the HeaviSine signal  $g_2$  used in the last experiment, and we distinguish two experimental setups: In the first scenario, we increase the number of data points  $N$  by increasing the number of samples of the HeaviSine so that the number of discontinuities remains two. In the second scenario, we increase the number of data points by repeating the HeaviSine signal so that the number of discontinuities grows linearly with the signal length. We computed the CSSD for the parameters  $p = 0.9999$  and  $\gamma = 20$  which resulted in reasonably good estimates for the underlying true signal.

In Table 1, we report two measures of the computational effort, the runtimes and the frequency each data point is “visited” for computing all necessary  $\mathcal{E}_{[l:r]}$ . (The baseline algorithm needs the vector  $y_{l:r}$  of length  $r - l + 1$  while the proposed method only needs a single data point,  $y_l$  or  $y_r$ , depending on the order of update.)

We start our analysis by examining the frequencies at which each data point is visited. In the first scenario, where the number of discontinuities is held constant, the baseline method shows a pattern more akin to cubic growth, whereas the proposed method rather shows quadratic growth. For the second scenario, the growth in  $N$  trends toward linear for all considered methods. However, the hidden constant depends on the average segment length, and the relation is quadratic for the baseline method, and

<sup>2</sup>Reference implementation of CSSD provided at <https://github.com/mstorath/CSSD>.

<sup>3</sup><https://github.com/deepcharles/ruptures>

<sup>4</sup><https://github.com/espdev/csaps>

**Table 1.** Computational effort of the baseline solver and the proposed solver with different pruning strategies for computing a CSSD.

Signal length $N$	250	500	1000	2000	4000	8000
<b>Constant number of discontinuities</b>						
Baseline	$1.1 \cdot 10^6$	$2.7 \cdot 10^6$	$2.0 \cdot 10^7$	$1.5 \cdot 10^8$	$1.3 \cdot 10^9$	$7.3 \cdot 10^9$
	45.7	110.1	453.4	1909.1	8704.0	33331.3
Prop. + FPVI	$2.5 \cdot 10^4$	$10.0 \cdot 10^4$	$4.0 \cdot 10^5$	$1.7 \cdot 10^6$	$7.0 \cdot 10^6$	$2.9 \cdot 10^7$
	0.4	1.0	4.2	17.3	72.6	297.1
Prop. + PELT	$1.8 \cdot 10^4$	$4.3 \cdot 10^4$	$1.7 \cdot 10^5$	$6.8 \cdot 10^5$	$2.7 \cdot 10^6$	$9.5 \cdot 10^6$
	0.3	0.6	2.2	8.7	34.5	120.9
<b>Increasing number of discontinuities</b>						
Baseline	$3.9 \cdot 10^5$	$8.0 \cdot 10^5$	$7.4 \cdot 10^5$	$1.4 \cdot 10^6$	$2.9 \cdot 10^6$	$5.7 \cdot 10^6$
	28.6	57.8	81.8	158.9	323.6	652.3
Prop. + FPVI	$2.1 \cdot 10^4$	$9.1 \cdot 10^4$	$1.6 \cdot 10^5$	$2.4 \cdot 10^5$	$5.0 \cdot 10^5$	$1.3 \cdot 10^6$
	0.2	1.0	1.7	2.5	5.3	13.7
Prop. + PELT	$1.2 \cdot 10^4$	$2.4 \cdot 10^4$	$3.5 \cdot 10^4$	$6.9 \cdot 10^4$	$1.4 \cdot 10^5$	$2.8 \cdot 10^5$
	0.2	0.3	0.5	1.0	2.0	3.9

NOTE: The top rows represent the frequency a data item is visited by the algorithm, and the bottom rows the total runtime in seconds, respectively. The subtables show two scenarios: Increasing sampling density which results in a constant number of discontinuities, and repeating the signal so that the number of discontinuities scales linearly with the signal length. The reported values are averages over three runs with different noise realizations and a fixed parameter set.

linear for the proposed method. Importantly, in both scenarios, the proposed method significantly reduces computational effort with respect to the total frequencies at which each data point is visited.

We next look at the corresponding execution times. In the first scenario, we note that the corresponding runtimes for the baseline method exhibit a more favorable growth pattern compared to the growth pattern of the frequencies. This appears to be a consequence of the fact that the `csaps` function, which is used for computing the smoothing splines, shows a predominantly constant growth, and this diverges from the asymptotic linear growth for the majority of the signal lengths evaluated by the algorithm. An asymptotic cubic growth in runtime is expected for even longer signals than the ones currently used for comparison. Overall, the proposed method is much faster than the baseline method. In particular, the proposed methods allows to process signals of moderate size on a laptop repeatedly within a reasonable time frame.

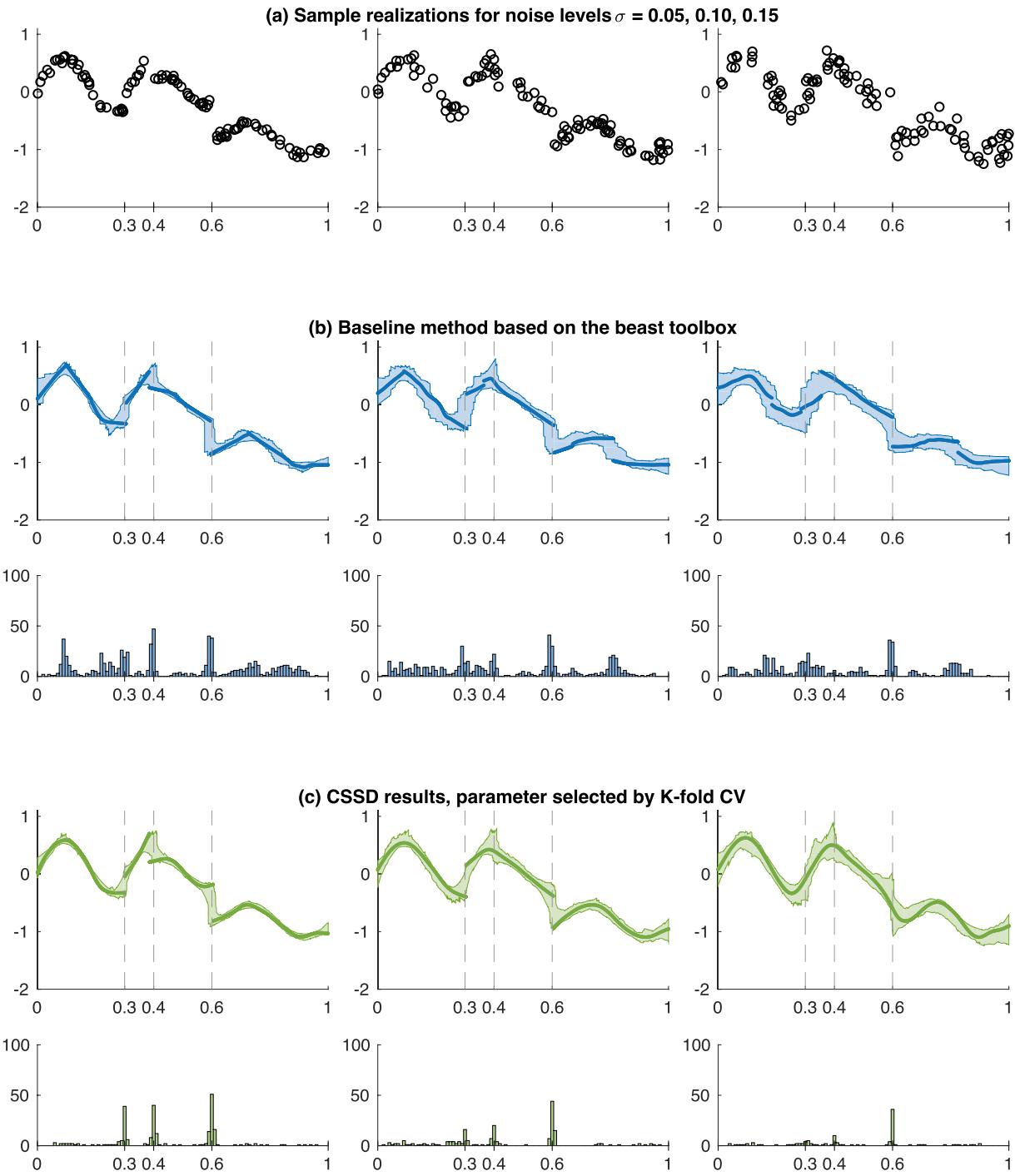
In both scenarios of the above example, the PELT pruning is more effective than the FPVI pruning. Similar observations were made for other signals (not shown in this article) in which the CSSD detects at least a few discontinuities. Interestingly, for the parameter selection by cross-validation on the real datasets (Section 3.5) it is the other way round: the runtimes with FPVI pruning were 2.1 min (geyser data) and 5.8 hr (stock data) whereas runtimes with PELT pruning were 2.7 min and 10.7 hr, respectively. (A detailed table of the runtimes for each sampled  $(p, \gamma)$  value of the Geyser data is given in the supplementary material.) The reasons for the lower runtimes of FPVI pruning in these cases are that a relatively low number of discontinuities are detected and that the optimization procedure frequently evaluates the target functional for large values of  $\gamma$  where FPVI is more effective than PELT.

### 3.4. Automatic Parameter Selection

As for classical smoothing splines, it may be sufficient for some practical purposes to choose the model parameters by visual inspection; yet a procedure for automatic parameter selection is useful (see the discussion on this topic by Silverman (1985,

sec. 4)). To obtain an automatic proposal for the model parameters  $p$  and  $\gamma$ , we here use K-fold cross validation (CV) as follows. We partition the data randomly into  $K$  folds of approximately equal size. The K-fold cross-validation score for  $p$  and  $\gamma$  is given by  $CV(p, \gamma) = \frac{1}{N} \sum_{k=1}^K \sum_{i \in Fold_k} ((\hat{f}_{p, \gamma}^{-k}(x_i) - y_i) / \delta_i)^2$ . Here,  $\hat{f}_{p, \gamma}^{-k}$  denotes the result of the proposed algorithm applied to the all data except those in fold  $k$ . (Recall from Section 2.6 that the solutions of (2) might be nonunique. Yet, the proposed algorithm provides a unique solution, namely the one with the largest possible right-most interval, followed by the largest possible penultimate interval, and so on.) If an evaluation point  $x_i$  coincides with a discontinuity of  $f_{p, \gamma}^{-k}$ , we take the mean of the lefthand and the righthand limits:  $f_{p, \gamma}^{-k}(x_i) = \lim_{t \rightarrow 0+} \frac{1}{2} (f_{p, \gamma}^{-k}(x_i - t) + f_{p, \gamma}^{-k}(x_i + t))$ . A typical choice is  $K = 5$  folds (Hastie et al. 2009) which we adopt here. Optimal parameters  $p, \gamma$  in the sense of K-fold cross-validation are minima of the scoring function  $(p, \gamma) \mapsto CV(p, \gamma)$ . To find a good parameter set in the sense of CV, we improve a starting value  $(p_0, \gamma_0)$  using standard derivative-free optimizers: a global search via simulated annealing followed by a local refinement using the Nelder-Mead method. We use the implementations of Matlab with default options. The simulated annealing algorithm uses a balance of exploration (looking for new, possibly better solutions) and exploitation (optimizing around the best solution found so far), regulated by the temperature and its cooling schedule. By the mechanism of being able to escape local minima, it is less sensitive to the starting value than local methods. To save computation time, it is reasonable to use a starting value that gives a visually reasonable result based on the domain knowledge or prior experience. To enhance parallel processing, simulated annealing could be replaced by a grid search of the  $(p, \gamma)$  domain. For the optimization,  $\gamma$  is parameterized by  $\gamma = p \frac{q}{1-q}$  with  $q \in [0, 1]$ . (This parameterization is obtained by dividing the functional by  $p$  and reparameterizing the resulting penalty  $\gamma' = \gamma/p$  by  $\gamma' = \frac{q}{1-q}$ ).

The procedure gives reasonable results in the conducted experiments (see results further below), but it comes with the limitations that the standard optimizers often need a high number of function evaluations leading to long runtimes, and that the resulting parameters are not guaranteed to be global minimizers of the CV scoring function. The second limitation can be

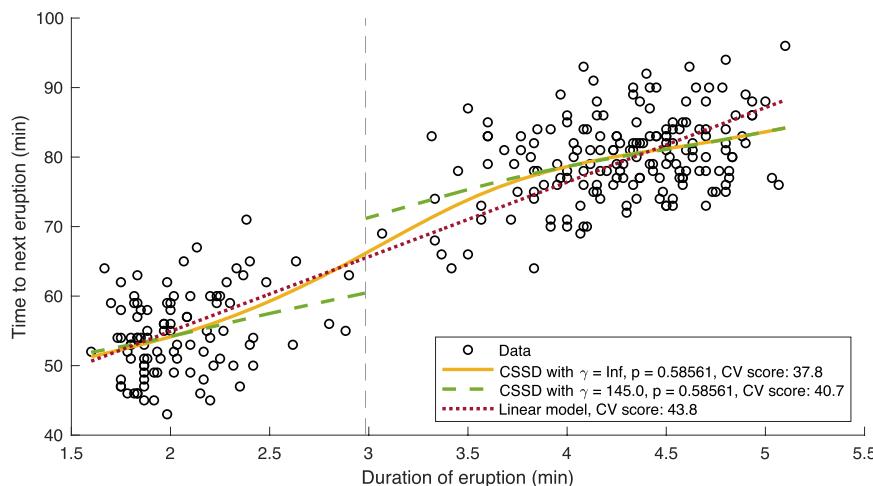


**Figure 3.** CSSD results with automatically selected parameter based on K-fold CV in comparison with a baseline method based on the beast toolbox. (a) Noisy samples of the test signal  $g_1$  shown in Figure 1(a) with the noise levels  $\sigma = 0.05, 0.1, 0.15$  (from left to right). (b) Results of the baseline method consisting of a method for piecewise regression with automatic parameter selection and subsequent spline interpolation. The dashed vertical lines indicate the true discontinuity locations. (As in Figure 1, the solid line represents the result for the sample signal, and the shaded regions represent the 2.5% to 97.5% quantiles, and the histograms represent the frequencies of the discontinuity locations.) (c) Results of the CSSD with automatic parameter selection based on K-fold cross-validation.

mitigated by restarting the optimization with different starting values until no further improvement of the CV score is observed. As the costs for this improvement are additional computation time and/or manual refinement, we use this strategy only for the two real data experiments presented further below. For the synthetic example described further below, the initial values  $p_0 = 0.99$  and  $\gamma_0 = 1$  are used. (These parameters were chosen based on their ability to produce visually acceptable results for a set of sample signals.)

We compare the results of the proposed method to the following baseline method. We use the Bayesian ensemble method of Zhao et al. (2019), implemented in the toolbox `beast`,<sup>5</sup> to estimate changepoints and a corresponding piecewise estimate of the signal. To this end, the function `beast_irreg` (variant for nonuniform sample distances) is called with deactivated

<sup>5</sup>Source code retrieved from <https://github.com/zhaokg/Rbeast> on November 2, 2022.



**Figure 4.** Fitting a CSSD to the Old Faithful data (circles): If the parameter is selected based on K-fold CV we obtain a result without discontinuities which coincides with a classical smoothing spline (solid curve). Keeping the selected  $p$ -parameter and lowering the  $\gamma$  parameter sufficiently gives a two-phase regression curve (dashed curves) with a breakpoint near  $x = 3$  (dashed vertical line), and the two curve segments are nearly linear. Both of the above parameter sets yield better CV-scores than a linear model (dotted line).

seasonality option (because the signal has no seasonality), time step option 0.01 (binning the non-equidistant samples to 100 bins of same length), and default parameters otherwise. (As internal preprocessing, `beast_irreg` aggregates data over bins of fixed width to obtain equidistant data sites. The procedure returns the first bin locations after the corresponding change; so to match the “midpoint convention” used in this article, we work with the midpoint between the returned bin and the preceding non-empty bin.) We then fit a piecewise interpolating spline to the estimated values between the changepoints to obtain a functional estimate on the entire domain  $[0, 1]$ . (The interpolating spline is preferred over the smoothing spline here, because a smoothing spline would give different  $y$ -values at the data sites than those estimated by `beast` and would introduce an additional parameter.)

We use the experimental setup used for Figure 1 and two additional noise levels to compare the results of the two methods. Figure 3 shows the results for 100 signal realizations. We observe that the baseline method returns principally more changepoint candidates than the proposed method, and that a considerable portion of them are located away from the true discontinuity locations, for example near the local extrema of the test signal. The proposed method exhibits less spurious discontinuities than the baseline method and the cluster points in the histograms are more sharply located around the true discontinuity locations.

### 3.5. Results on Real Data Using Automatic Parameter Selection

The next experiment follows an example presented in the work of Silverman (1985) on smoothing splines. The dataset contains the duration of eruptions along with the waiting time to the next eruption of the Old Faithful geyser in the Yellowstone National Park, USA; see Figure 4.<sup>6</sup> When fitting a single straight

line to the data, curvature effects in the residuals are observed. Silverman (1985) argues that fitting a smoothing spline is a useful exploratory step toward the choice of a reasonable model. Inspecting the shape of the resulting spline suggest a two phase linear regression model as plausible alternative. Remarkably, the CSSD model comprises both of the above models: The automatic parameter selection results in  $\gamma = \infty$  so that the corresponding CSSD coincides with a classical smoothing spline. An appropriately chosen smaller  $\gamma$ -parameter gives a two phase model which is nearly linear on the two segments, and tends to a piecewise linear model for  $p \rightarrow 0$ . Ranking these three models by their CV scores, the classical spline can be considered as the best one, the linear model as the worst, and the two phase model lies in the middle. The example illustrates that the CSSD can serve as a useful tool for exploratory data analysis.

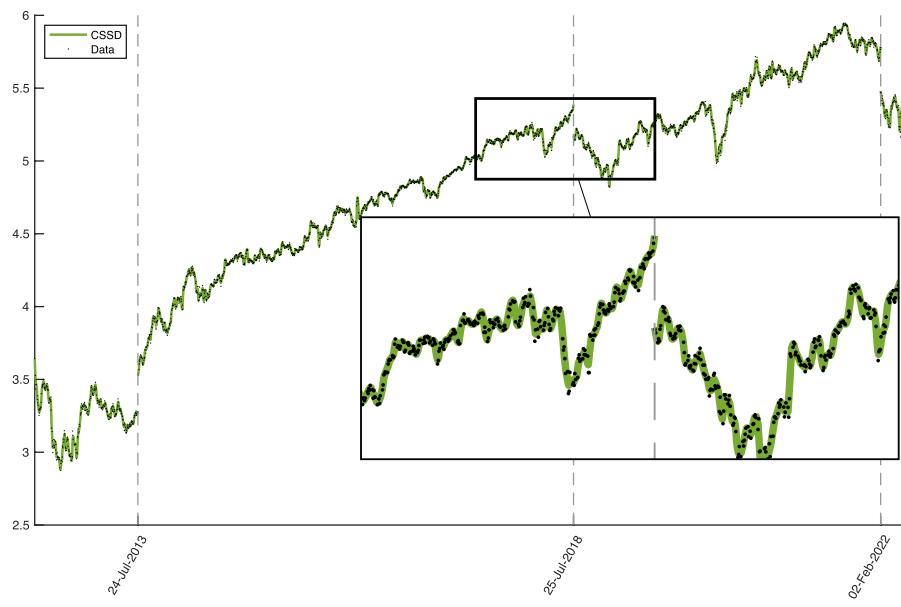
Eventually we report the results on a stock market time series (Figure 5). The present data shows the logarithm of the closing price of the Meta/Facebook stock from May 18, 2012, to May 19, 2022.<sup>7</sup> The dates of the discontinuities can be related to strong market reactions after business reports: For example, on July 24, 2013, Facebook announced remarkable rises in revenues (Wilhelm 2013). On July 25, 2018, Meta announced lower profit margins (Sharma and Vengattil 2018), and on February 2, 2022, the price of Meta shares dropped strongly as Facebook reported a decline in daily users for the first time (Culliford and Balu 2022). In this scenario, detected discontinuities could be interpreted as abrupt changepoints of the signal. The computation times are reported at the end of Section 3.3.

## 4. Conclusion and Outlook

We have studied a variational model for cubic smoothing splines with unknown discontinuity locations which is a special case of the weak rod model. This model comprises classical continuous cubic splines (for sufficiently large  $\gamma$  parameters) and piecewise

<sup>6</sup>The dataset was retrieved from the supplementary material of Wasserman (2004) <https://www.stat.cmu.edu/~larry/all-of-statistics/=data/faithful.dat>

<sup>7</sup>Data source: <https://www.macrotrends.net/stocks/charts/FB/meta-platforms/stock-price-history>



**Figure 5.** The dots represent the logarithm of the closing prices of the Meta stock from May 18, 2012, to May 19, 2022. The curve represents the CSSD with parameters determined by K-fold CV ( $p = 0.4702$ ,  $\gamma = 0.0069$ ). The dashed vertical lines indicate the discontinuities of the CSSD, and the ticks correspond to the date before the discontinuity.

linear regression (for  $p \rightarrow 0$ ). We have shown that the solutions are unique in an almost everywhere sense on segments comprising at least three data sites. We developed an efficient algorithm for computing the global minimizer of the model's underlying optimization problem. The runtime experiments indicated that the method is particularly efficient for signals with many discontinuities. But also in the general case, the algorithm can be executed on a laptop in reasonable time for signals of moderate lengths. The algorithm is applicable to data with nonequidistant sampling points and to vector-valued data. Automatic parameter selection based on K-fold cross-validation gave reasonable results for synthetic and real data; the main limitation of this selection strategy is currently the relatively high runtime. The numerical examples have illustrated potential applications of a CSSD: as function estimator for a discontinuous signal, as basis for a changepoint detector, or as tool for exploratory data analysis.

Open questions for future work include faster algorithms for automatically selecting the model parameters, extension to splines of higher orders and higher dimensions, and further investigation of theoretical properties of the estimator.

## Supplementary Materials

**Supplementary document:** The supplementary document contains proofs of the theorems, justifications for the compatibility with the pruning strategies, and additional experiments.

## Disclosure Statement

The authors report there are no competing interests to declare.

## Funding

MS was supported by the research program “Informations- und Kommunikationstechnik” of the Bavarian State Ministry of Economic Affairs,

Regional Development and Energy (DIK-2105-0044 / DIK0264). AW acknowledges support of Deutsche Forschungsgemeinschaft (DFG) under project number 514177753.

## References

- Arlot, S., and Celisse, A. (2011), “Segmentation of the Mean of Heteroscedastic Data via Cross-Validation,” *Statistics and Computing*, 21, 613–632. [653]
- Auger, I. E., and Lawrence, C. E. (1989), “Algorithms for the Optimal Identification of Segment Neighborhoods,” *Bulletin of Mathematical Biology*, 51, 39–54. [653]
- Baranowski, R., Chen, Y., and Fryzlewicz, P. (2019), “Narrowest-Over-Threshold Detection of Multiple Change Points and Change-Point-Like Features,” *Journal of the Royal Statistical Society, Series B*, 81, 649–672. [653]
- Blake, A. (1989), “Comparison of the Efficiency of Deterministic and Stochastic Algorithms for Visual Reconstruction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, 2–12. [652,653]
- Blake, A., and Zisserman, A. (1987), *Visual Reconstruction*, Cambridge: MIT Press. [651,652,653]
- Boysen, L., Kempe, A., Liebscher, V., Munk, A., and Wittich, O. (2009), “Consistencies and Rates of Convergence of Jump-Penalized Least Squares Estimators,” *The Annals of Statistics*, 37, 157–183. [652,653]
- Cheney, E. (1998), *Introduction to Approximation Theory*. 1998. reprint of the 2nd de(1982) ed. [656]
- Craven, P., and Wahba, G. (1979), “Smoothing Noisy Data with Spline Functions: Estimating the Correct Degree of Smoothing by the Method of Generalized Cross-Validation,” *Numerische Mathematik*, 31, 377–403. [653]
- Culliford, E., and Balu, N. (2022), “Meta Shares Sink 20% as Facebook Loses Daily Users for the First Time,” *Reuters*. Available at <https://www.reuters.com/technology/facebook-owner-meta-forecasts-q1-revenue-behind-estimates-2022-02-02/>. [661]
- De Boor, C. (2001), *A Practical Guide to Splines - Revised Edition* (Vol. 27), New York: Springer-Verlag. [651,653,656,658]
- De Hoog, F., and Hutchinson, M. (1986), “An Efficient Method for Calculating Smoothing Splines Using Orthogonal Transformations,” *Numerische Mathematik*, 50, 311–319. [654,656]
- Donoho, D. L., and Johnstone, J. M. (1994), “Ideal Spatial Adaptation by Wavelet Shrinkage,” *Biometrika*, 81, 425–455. [654,658]



- Dougherty, R. L., Edelman, A. S., and Hyman, J. M. (1989), "Nonnegativity-, Monotonicity-, or Convexity-Preserving Cubic and Quintic Hermite Interpolation," *Mathematics of Computation*, 52, 471–494. [656]
- Drobyshev, A., Machka, C., Horsch, M., Seltmann, M., Liebscher, V., de Angelis, M., and Beckers, J. (2003), "Specificity Assessment from Fractionation Experiments (Safe): A Novel Method to Evaluate Microarray Probe Specificity based on Hybridisation Stringencies," *Nucleic Acids Research*, 31, 1–10. [651]
- Fearnhead, P. (2006), "Exact and Efficient Bayesian Inference for Multiple Changepoint Problems," *Statistics and Computing*, 16, 203–213. [653]
- Frick, K., Munk, A., and Sieling, H. (2014), "Multiscale Change Point Inference," *Journal of the Royal Statistical Society, Series B*, 76, 495–580. [651,653]
- Friedrich, F., Kempe, A., Liebscher, V., and Winkler, G. (2008), "Complexity Penalized M-estimation," *Journal of Computational and Graphical Statistics*, 17, 201–224. [652,653,655]
- Fryzlewicz, P. (2014), "Wild Binary Segmentation for Multiple Changepoint Detection," *The Annals of Statistics*, 42, 2243–2281. [653]
- Golub, G. H., Heath, M., and Wahba, G. (1979), "Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter," *Technometrics*, 21, 215–223. [653]
- Green, P. J., and Silverman, B. W. (1993), *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*, Boca Raton, FL: CRC Press. [653]
- Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Vol. 2), New York: Springer. [651,653,659]
- Haynes, K., Fearnhead, P., and Eckley, I. A. (2017), "A Computationally Efficient Nonparametric Approach for Changepoint Detection," *Statistics and Computing*, 27, 1293–1305. [653]
- Hotz, T., Schutte, O. M., Sieling, H., Polupanow, T., Diederichsen, U., Steinem, C., and Munk, A. (2013), "Idealizing Ion Channel Recordings by a Jump Segmentation Multiresolution Filter," *IEEE Transactions on NanoBioscience*, 12, 376–386. [651]
- Hupé, P., Stransky, N., Thiery, J., Radvanyi, F., and Barillot, E. (2004), "Analysis of Array CGH Data: From Signal Ratio to Gain and Loss of DNA Regions," *Bioinformatics*, 20, 3413–3422. [651]
- Hutchinson, M. (1986), "Algorithm 642: A Fast Procedure for Calculating Minimum Cross-Validation Cubic Smoothing Splines," *ACM Transactions on Mathematical Software (TOMS)*, 12, 150–153. [654]
- Jackson, B., Scargle, J. D., Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., Gwin, E., Sangtrakulcharoen, P., Tan, L., and Tsai, T. T. (2005), "An Algorithm for Optimal Partitioning of Data on an Interval," *IEEE Signal Processing Letters*, 12, 105–108. [652,653,655]
- Jeong, S., Park, T., and van Dyk, D. A. (2021), "Bayesian Model Selection in Additive Partial Linear Models via Locally Adaptive Splines," *Journal of Computational and Graphical Statistics*, 31, 324–336. [653]
- Joo, C., Balci, H., Ishitsuka, Y., Buranachai, C., and Ha, T. (2008), "Advances in Single-Molecule Fluorescence Methods for Molecular Biology," *Annual Review of Biochemistry*, 77, 51–76. [651]
- Killick, R., Fearnhead, P., and Eckley, I. (2012), "Optimal Detection of Changepoints with a Linear Computational Cost," *Journal of the American Statistical Association*, 107, 1590–1598. [652,653,654,657,658]
- Kim, K.-R., Dryden, I. L., Le, H., and Severn, K. E. (2021), "Smoothing Splines on Riemannian Manifolds, with Applications to 3d Shape Space," *Journal of the Royal Statistical Society, Series B*, 83, 108–132. [653]
- Kleinberg, J., and Tardos, É. (2006), *Algorithm Design*, Delhi: Pearson Education India. [651,652,655]
- Koo, J.-Y. (1997), "Spline Estimation of Discontinuous Regression Functions," *Journal of Computational and Graphical Statistics*, 6, 266–284. [653]
- Little, M. A., and Jones, N. S. (2011a), "Generalized Methods and Solvers for Noise Removal from Piecewise Constant Signals. I. Background Theory," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 467, 3088–3114. [653]
- (2011b), "Generalized Methods and Solvers for Noise Removal from Piecewise Constant Signals. II. New Methods," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 467, 3115–3140. [653]
- Loeff, L., Kerssemakers, J. W., Joo, C., and Dekker, C. (2021), "Autostepfinder: A Fast and Automated Step Detection Method for Single-Molecule Analysis," *Patterns*, 2, 100256. [651]
- Mallat, S. (2008), *A Wavelet Tour of Signal Processing: The Sparse Way*, New York: Academic Press. [654]
- Meng, C., Yu, J., Chen, Y., Zhong, W., and Ma, P. (2022), "Smoothing Splines Approximation Using Hilbert Curve Basis Selection," *Journal of Computational and Graphical Statistics*, 31, 802–812. [653]
- Mumford, D., and Shah, J. (1985), "Boundary Detection by Minimizing Functionals," in *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. 17), pp. 137–154. [653]
- (1989), "Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems," *Communications on Pure and Applied Mathematics*, 42, 577–685. [652,653]
- Page, E. S. (1954), "Continuous Inspection Schemes," *Biometrika*, 41, 100–115. [653]
- Reinsch, C. H. (1967), "Smoothing by Spline Functions," *Numerische Mathematik*, 10, 177–183. [653,654,656,657]
- Schoenberg, I. J. (1964), "Spline Functions and the Problem of Graduation," *Proceedings of the American Mathematical Society*, 52, 947–950. [653]
- Sharma, V., and Vengattil, M. (2018), "Zuckerberg Loses More than \$15 Billion in Record Facebook Fall," *Reuters*, available at <https://www.reuters.com/article/us-facebook-results-stock-idUSKBN1KG1TN>. [661]
- Silverman, B. W. (1984), "Spline Smoothing: The Equivalent Variable Kernel Method," *The Annals of Statistics*, 12, 898–916. [653]
- (1985), "Some Aspects of the Spline Smoothing Approach to Nonparametric Regression Curve Fitting," *Journal of the Royal Statistical Society, Series B*, 47, 1–21. [651,652,653,655,659,661]
- Snijders, A., Nowak, N., Segraves, R., Blackwood, S., Brown, N., Conroy, J., et al. (2001), "Assembly of Microarrays for Genome-Wide Measurement of DNA Copy Number by CGH," *Nature Genetics*, 29, 263–264. [651]
- Sowa, Y., Rowe, A., Leake, M., Yakushi, T., Homma, M., Ishijima, A., and Berry, R. (2005), "Direct Observation of Steps in Rotation of the Bacterial Flagellar Motor," *Nature*, 437, 916–919. [651]
- Storath, M., and Weinmann, A. (2014), "Fast Partitioning of Vector-Valued Images," *SIAM Journal on Imaging Sciences*, 7, 1826–1852. [654,657]
- Storath, M., Weinmann, A., and Demaret, L. (2014), "Jump-Sparse and Sparse Recovery Using Potts Functionals," *IEEE Transactions on Signal Processing*, 62, 3654–3666. [653]
- Storath, M., Weinmann, A., and Unser, M. (2017), "Jump-Penalized Least Absolute Values Estimation of Scalar or Circle-Valued Signals," *Information and Inference: A Journal of the IMA*, 6, 225–245. [653]
- Storath, M., Kiefer, L., and Weinmann, A. (2019), "Smoothing for Signals with Discontinuities Using Higher Order Mumford-Shah Models," *Numerische Mathematik*, 143, 423–460. [652]
- Tickle, S. O., Eckley, I., Fearnhead, P., and Haynes, K. (2020), "Parallelization of a Common Changepoint Detection Method," *Journal of Computational and Graphical Statistics*, 29, 149–161. [653]
- Truong, C., Oudre, L., and Vayatis, N. (2020), "Selective Review of Offline Change Point Detection Methods," *Signal Processing*, 167, 107299. [658]
- Unser, M. (1999), "Splines: A Perfect Fit for Signal and Image Processing," *IEEE Signal Processing Magazine*, 16, 22–38. [653]
- (2002), "Splines: A Perfect Fit for Medical Imaging," in *Medical Imaging 2002: Image Processing* (Vol. 4684), pp. 225–236, International Society for Optics and Photonics. [653]
- van den Burg, G. J. J., and Williams, C. K. I. (2020), "An Evaluation of Change Point Detection Algorithms." Available at <https://arxiv.org/abs/2003.06222>. [653]
- Wahba, G. (1990). *Spline Models for Observational Data*, Philadelphia, PA: SIAM. [653]
- Wasserman, L. (2004), *All of Statistics – A Concise Course in Statistical Inference*, New York: Springer. [661]
- Weinmann, A., and Storath, M. (2015), "Iterative Potts and Blake-Zisserman Minimization for the Recovery of Functions with Discontinuities from Indirect Measurements," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471, 20140638. [653]
- Weinmann, A., Demaret, L., and Storath, M. (2016), "Mumford-Shah and Potts Regularization for Manifold-Valued Data," *Journal of Mathematical Imaging and Vision*, 55, 428–445. [653]
- Weinmann, A., Storath, M., and Demaret, L. (2015), "The  $L^1$ -Potts Functional for Robust Jump-Sparse Reconstruction," *SIAM Journal on Numerical Analysis*, 53, 644–673. [655]

- Whittaker, E. T. (1923), "On a New Method of Graduation," *Proceedings of the Edinburgh Mathematical Society*, 41, 63–75. [653]
- Wilhelm, A. (2013), "Facebook q2 Earnings Beat Expectations with \$1.81b in Revenue, Up 53%, Mobile Hits 41% of Ad Revenue," *Techcrunch*. Available at <https://techcrunch.com/2013/07/24/facebook-q2-earnings-beats-with-1-81b-in-revenue-up-53-mobile-hits-41-of-ad-revenue/>. [661]
- Winkler, G. (2003), *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction* (Vol. 27), Berlin: Springer. [652]
- Winkler, G., and Liebscher, V. (2002), "Smoothers for Discontinuous Signals," *Journal of Nonparametric Statistics*, 14, 203–222. [652,653,655,658]
- Winkler, G., Wittich, O., Liebscher, V., and Kempe, A. (2005), "Don't Shed Tears over Breaks," *Jahresbericht DMV*, 107, 57–87. [651,653]
- Wittich, O., Kempe, A., Winkler, G., and Liebscher, V. (2008), "Complexity Penalized Least Squares Estimators: Analytical Results," *Mathematische Nachrichten*, 281, 582–595. [657]
- Yang, G., Zhang, B., and Zhang, M. (2021), "Estimation of Knots in Linear Spline Models," *Journal of the American Statistical Association*, 118, 639–650. [653]
- Yao, Y.-C. (1988), "Estimating the Number of Change-Points via Schwarz Criterion," *Statistics & Probability Letters*, 6, 181–189. [653]
- Zhang, N. R., and Siegmund, D. O. (2007), "A Modified Bayes Information Criterion with Applications to the Analysis of Comparative Genomic Hybridization Data," *Biometrics*, 63, 22–32. [653]
- Zhao, K., Wulder, M. A., Hu, T., Bright, R., Wu, Q., Qin, H., Li, Y., Toman, E., Mallick, B., Zhang, X., et al. (2019), "Detecting Change-Point, Trend, and Seasonality in Satellite Time Series Data to Track Abrupt Changes and Nonlinear Dynamics: A Bayesian Ensemble Algorithm," *Remote Sensing of Environment*, 232, 111181. [653,660]
- Zou, C., Yin, G., Feng, L., and Wang, Z. (2014), "Nonparametric Maximum Likelihood Approach to Multiple Change-Point Problems. *The Annals of Statistics*, 42, 970–1002. DOI:10.1214/14-AOS1210. [653]