

# Chromatogram baseline estimation and denoising using sparsity (BEADS) <sup>☆</sup>

Xiaoran Ning<sup>a</sup>, Ivan W. Selesnick<sup>a</sup>, Laurent Duval<sup>b,c</sup>

<sup>a</sup>*Polytechnic School of Engineering, New York University, 6 Metrotech Center, Brooklyn, NY 11201*

<sup>b</sup>*IFP Energies nouvelles, Technology Division, Rueil-Malmaison, France*

<sup>c</sup>*Université Paris-Est, LIGM, ESIEE Paris, France*

---

## Abstract

This paper jointly addresses the problems of chromatogram baseline correction and noise reduction. The proposed approach is based on modeling the series of chromatogram peaks as sparse with sparse derivatives, and on modeling the baseline as a low-pass signal. A convex optimization problem is formulated so as to encapsulate these non-parametric models. To account for the positivity of chromatogram peaks, an asymmetric penalty function is utilized. A robust, computationally efficient, iterative algorithm is developed that is guaranteed to converge to the unique optimal solution. The approach, termed Baseline Estimation And Denoising with Sparsity (BEADS), is evaluated and compared with two state-of-the-art methods using both simulated and real chromatogram data.

*Keywords:* baseline correction, baseline drift, sparse derivative, asymmetric penalty, low-pass filtering, convex optimization

---

## 1. Introduction

Several sources of uncertainties affect the quality and the performance of gas and liquid chromatography analysis [48, 1]. As with many other analytical chemistry methods (including infrared or Raman spectra [6]), chromatogram measurements are often considered as a combination of peaks, background and noise [35]. The two latter terms are sometimes merged under different denominations: drift noise, baseline wander, or spectral continuum. For instance in [5], the baseline drift is characterized as a “*colored*” noise, with a low-frequency dominance in the noise power spectrum. In the following, we restrict the term “baseline” to refer to the smoothest part of the trend or bias (*The portion of the chromatogram recording the detector response when only the mobile phase emerges from the column*, [34]), while we call “noise” the more stochastic part. Peak line shapes are of possibly various nature, from Gaussian to asymmetric empirical models [17, p. 97 sq.]. Meanwhile, they can easily be described as short-width, steep-sided up-and-down bumps. They therefore also possess relatively broad frequency spectra, albeit localized and behaving differently from the drift noise disturbance. Leaving peak artifacts (fronting and tailing, co-elution, etc.) aside, their quantitative analysis (peak area, width, height quantification) is thus hindered by the possibility to accurately remove both the smooth baseline and the random noise [29]. Indeed, these problems are often addressed independently, in two different steps (which could, in turn, “*introduce substantial levels of correlated noise*” [5]): a generally low-order approximation or smoothing for the baseline, and forms of filtering for the noise on the residual chromatogram with background removed.

First, although seemingly simple, the problem of baseline subtraction remains a long-standing issue, which can be traced back to [58, 38]. Recent overviews are presented in [42, 20, 27]. Spectral information processing [46, 47, 57] has been a major course of action. Methods based on linear and non-linear [36, 26, 41] filtering, or multiscale forms of filtering with wavelet transforms [9, 24, 7, 31] have been proposed. The relative overlap between the spectra of the peaks, the baseline, and the noise has led to alternative regression

---

<sup>☆</sup>This research is supported by the NSF under grant CCF-1018020.

*Email addresses:* xn211@nyu.edu (Xiaoran Ning), selesi@nyu.edu (Ivan W. Selesnick), laurent.duval@ifpen.fr (Laurent Duval)

LAST EDIT: 1:20pm, September 21, 2014

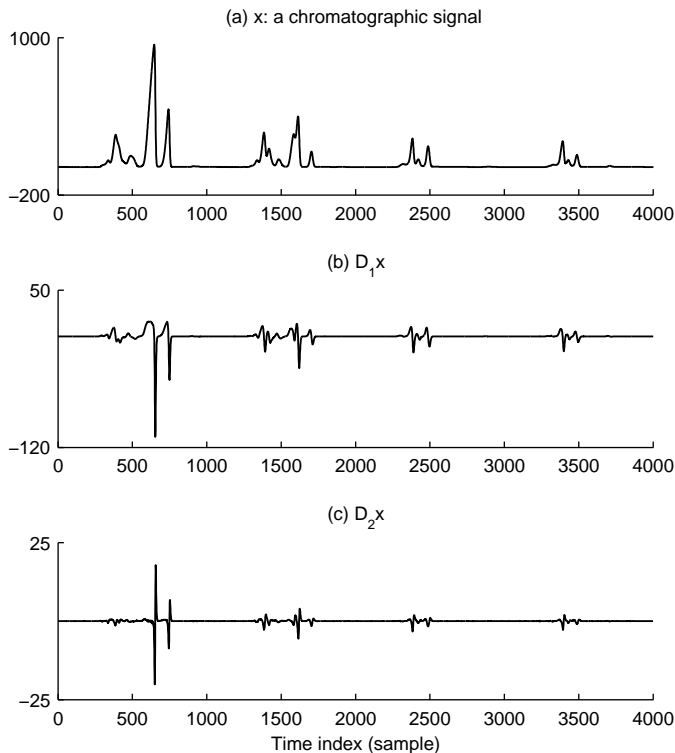


Figure 1: Gas chromatogram from two-dimensional gas chromatography [52], with flat baseline and low noise: (a) analytical signal  $\mathbf{x}$ , (b) first-order difference  $\mathbf{D}_1\mathbf{x}$  and (c) second-order difference  $\mathbf{D}_2\mathbf{x}$ . The chromatographic signal is sparse, as are its first- and second-order derivatives.

models, based on various constraints. The low-pass part of the baseline may be modeled by regular functions, such as low-degree polynomials [33, 59] or (cubic) spline models [19, 23, 12], in conjunction with manual identification, polynomial fitting or iterative thresholding methods [21]. Related algorithms based on signal derivatives [5, 11] have been devised. In many approaches, modeling and constraints are laid on the potential features of the baseline itself: shape, smoothness, transformed domain properties. Consequently, it appears beneficial to investigate generalized penalizations [13, 3, 33, 59], with less stringent models on either the signal, the background or the noise. This is the very motivation of this paper: a joint estimation of these three chromatogram components while avoiding overly restrictive parametric models. Specifically, in this work, the baseline is modeled as a low-pass signal, while chromatogram peaks of interest are deemed to be sparse up to second order derivatives, leaving random noise as a residual.

In the past decade, this concept of parsimony, or sparsity, has been an active and fruitful drive in signal processing and chemistry. It entails the possibility of describing a signal of interest with a restricted number of non-zero parameters or components. Sparsity trades accuracy (of the description) with concentration (of the decomposition). Many algorithms based on sparsity have been developed for reconstruction, denoising, detection, deconvolution. Most sparse modeling techniques arose from the “least absolute shrinkage and selection operator” (better known under the *lasso* moniker [50, 40]), basis pursuit methods [10], total variation [8], and compound regularization [2]. While the latter essentially promote sparsity, different problems require, simultaneously, other constraints, like signal smoothness or residual stochasticity.

More specifically, recent works in signal [33, 43, 44, 37] and image processing [22, 15, 16, 49, 4] have promoted a framework for decomposing potentially complex measurements into “sufficiently” distinct components. Such non-linear decompositions are termed “morphological component analysis”, “geometric separation” or “clustered sparsity” [28]. Such approaches are amenable to analytical chemistry issues, relying on morphological properties of baselines and chromatographic peaks. Figure 1(a) displays a chromatogram  $\mathbf{x}$  obtained from two-dimensional gas chromatography [52]. It consists of abrupt peaks returning to a relatively flat baseline, and hence exhibits a form of sparsity. Moreover, as illustrated in Fig. 1(b) and (c), the second



With initialization  $\mathbf{x}^{(0)}$  and under suitable assumptions, the MM update equation (4) produces a sequence  $\mathbf{x}^{(k)}$  converging to the minimizer of  $F(\mathbf{x})$ .

### 3. Baseline Estimation and Denoising: Problem Formulation

The proposed approach is based on modeling an  $N$ -point noise-free chromatogram data vector as

$$\mathbf{s} = \mathbf{x} + \mathbf{f}, \quad \mathbf{s} \in \mathbb{R}^N. \quad (6)$$

The vector  $\mathbf{x}$ , consisting of numerous peaks, is modeled as a sparse-derivative signal (i.e.,  $\mathbf{x}$  and its first several derivatives are sparse). The vector  $\mathbf{f}$ , representing the baseline, is a low-pass signal. We further model the observed (noisy) chromatogram data as

$$\mathbf{y} = \mathbf{s} + \mathbf{w} \quad (7)$$

$$= \mathbf{x} + \mathbf{f} + \mathbf{w}, \quad \mathbf{y} \in \mathbb{R}^N \quad (8)$$

where  $\mathbf{w}$  is a stationary white Gaussian process with variance  $\sigma^2$ . Our goal is to estimate the baseline,  $\mathbf{f}$ , and peaks,  $\mathbf{x}$ , simultaneously, from observation  $\mathbf{y}$ .

We assume that if peaks are absent, then the baseline can be approximately recovered from a noise-corrupted observation by low-pass filtering, i.e.,  $\mathbf{f} \approx \mathbf{L}(\mathbf{f} + \mathbf{w})$  where  $\mathbf{L}$  is a suitable low-pass filter. Hence, given an estimate  $\hat{\mathbf{x}}$  of the peaks, we may obtain an estimate  $\hat{\mathbf{f}}$  of the baseline by filtering  $\mathbf{y}$  in (7) with low-pass filter  $\mathbf{L}$ ,

$$\hat{\mathbf{f}} = \mathbf{L}(\mathbf{y} - \hat{\mathbf{x}}). \quad (9)$$

In this case, we may obtain an estimate  $\hat{\mathbf{s}}$  by adding  $\hat{\mathbf{x}}$ ,

$$\hat{\mathbf{s}} = \hat{\mathbf{f}} + \hat{\mathbf{x}} \quad (10)$$

$$= \mathbf{L}(\mathbf{y} - \hat{\mathbf{x}}) + \hat{\mathbf{x}} \quad (11)$$

$$= \mathbf{L}\mathbf{y} + \mathbf{H}\hat{\mathbf{x}} \quad (12)$$

where  $\mathbf{H}$  is the high-pass filter,

$$\mathbf{H} = \mathbf{I} - \mathbf{L}. \quad (13)$$

In order to obtain an estimate  $\hat{\mathbf{x}}$  of the chromatogram peaks from observed data  $\mathbf{y}$ , we will formulate an inverse problem with the quadratic data fidelity term  $\|\mathbf{y} - \hat{\mathbf{s}}\|_2^2$ . Note that

$$\|\mathbf{y} - \hat{\mathbf{s}}\|_2^2 = \|\mathbf{y} - \mathbf{L}\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\|_2^2 \quad (14)$$

$$= \|\mathbf{H}(\mathbf{y} - \hat{\mathbf{x}})\|_2^2. \quad (15)$$

Hence, the data fidelity term in the proposed formulation depends on the high-pass filter  $\mathbf{H}$ . Also note that the data fidelity term does not depend on the baseline estimate  $\hat{\mathbf{f}}$ . The optimization problem, formulated below, produces an estimate of the chromatogram peaks  $\hat{\mathbf{x}}$ . The baseline estimate is then obtained by (9).

#### 3.1. Low-pass and High-pass Filters

We take the filters  $\mathbf{L}$  and  $\mathbf{H}$  to be zero-phase, non-causal, recursive filters. In other words, they filter relatively symmetric signals (such as chromatogram peaks), without introducing shifts in peak locations. A procedure for defining such filters is given in [45]. A filter is specified by two parameters: its order  $2d$  and its cutoff frequency  $f_c$ . The high-pass filter  $\mathbf{H}$  described in [45] is of the form

$$\mathbf{H} = \mathbf{A}^{-1}\mathbf{B} \quad (16)$$

where  $\mathbf{A}$  and  $\mathbf{B}$  are banded convolution (Toeplitz) matrices. Expressing  $\mathbf{H}$  in terms of banded matrices leads in [45] to the development of algorithms that utilize fast solvers for banded linear systems [39, Sec. 2.4]. Being convolution matrices,  $\mathbf{A}$  and  $\mathbf{B}$  represent linear, time-invariant (LTI) systems, and  $\mathbf{H}$  represents a

cascade of LTI systems where the LTI system  $\mathbf{B}$  is followed by the LTI system  $\mathbf{A}^{-1}$ . Using the commutative property of LTI systems, in this paper we define  $\mathbf{H}$  as

$$\mathbf{H} = \mathbf{B}\mathbf{A}^{-1}. \quad (17)$$

Due to  $\mathbf{A}$  and  $\mathbf{B}$  being finite matrices, they are not exactly commutative. However, the difference between (16) and (17) is confined to the beginning and end of the signal and hence, for long signals, is negligible. In Sections 4.1 and 4.2, we will see that (17) serves our purposes better than (16), allowing the derivation of a computationally efficient optimization algorithm.

### 3.2. Compound sparse derivative modeling

According to (6), the first  $i$  derivatives,  $i = 0, \dots, M$ , of the estimated peaks,  $\hat{\mathbf{x}}$ , should be sparse. In sparse signal processing, sparse signal behaviour is typically achieved through the use of suitable non-quadratic regularization terms. Therefore, to obtain an estimate  $\hat{\mathbf{x}}$ , the following optimization problem is proposed:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left\{ F(\mathbf{x}) = \frac{1}{2} \|\mathbf{H}(\mathbf{y} - \mathbf{x})\|_2^2 + \sum_{i=0}^M \lambda_i R_i(\mathbf{D}_i \mathbf{x}) \right\}. \quad (18)$$

The assumption that the observed data is corrupted by additive white Gaussian noise (AWGN) is reflected in the use of a quadratic data fitting term as is classical. The quadratic data fidelity term is given by (15).

In (18),  $\mathbf{D}_i$  is the order- $i$  difference operator defined in Section 2. Functions  $R_i : \mathbb{R}^{N-i} \rightarrow \mathbb{R}$ , are of the form

$$R_i(\mathbf{v}) = \sum_n \phi(v_n) \quad (19)$$

where the function  $\phi : \mathbb{R} \rightarrow \mathbb{R}$ , referred to as a penalty function, is designed to promote sparsity. Substituting (19) in (18), we obtain

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left\{ F(\mathbf{x}) = \frac{1}{2} \|\mathbf{H}(\mathbf{y} - \mathbf{x})\|_2^2 + \sum_{i=0}^M \lambda_i \sum_{n=0}^{N_i-1} \phi([\mathbf{D}_i \mathbf{x}]_n) \right\} \quad (20)$$

where  $N_i$  denotes the length of  $\mathbf{D}_i \mathbf{x}$ . The constants  $\lambda_i \geq 0$  are regularization parameters. Increasing  $\lambda_i$  has the effect of making  $\mathbf{D}_i \mathbf{x}$  more sparse. More discussion of how to specify  $\lambda_i$  will be given in Section 5.

Compared with [45], this work introduces several modifications to adapt the approach therein to chromatograms. The novel features of the approach proposed here include: 1) An  $M$ -term compound regularizer to model chromatogram peaks. 2) The modeling of the positivity of chromatogram peaks by non-symmetric penalties. 3) An improved algorithm based on MM (in contrast to ADMM as in [45]) for compound regularization. The improved algorithm converges faster and does not require any user-specified step-size parameter as does the earlier algorithm of [45].

### 3.3. Symmetric penalty functions

For many applications, samples of the target signal  $\mathbf{x}$  and its derivative  $\mathbf{D}_i \mathbf{x}$  are positive or negative with equal probability, or this information is not available. In such cases, the penalty function should be symmetric about  $x = 0$ . One such function is the absolute value function,

$$\phi_A(x) = |x| \quad (21)$$

which leads to  $\ell_1$  norm regularization. While widely used, one drawback of (21) is that it is non-differentiable at zero, which can lead to numerical issues, depending on the utilized optimization algorithm. To address this issue, we utilize a smoothed (differentiable) approximation of the  $\ell_1$  penalty function, for example, the hyperbolic function

$$\phi_B(x) = \sqrt{|x|^2 + \epsilon} \quad (22)$$

or

$$\phi_C(x) = |x| - \epsilon \log(|x| + \epsilon). \quad (23)$$

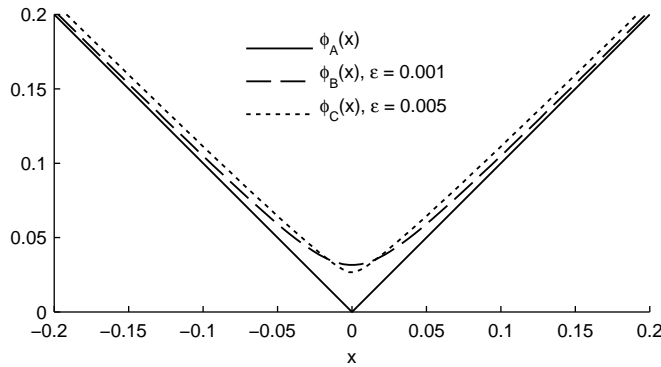


Figure 2: Penalty functions  $\phi_A(x)$ ,  $\phi_B(x)$ , and  $\phi_C(x)$ .

As the constant  $\epsilon > 0$  approaches zero, the functions  $\phi_B$  and  $\phi_C$  approach the absolute value function. When  $\epsilon = 0$ , then  $\phi_B$  and  $\phi_C$  reduce to the absolute value function. Functions  $\phi_A$ ,  $\phi_B$ , and  $\phi_C$  are illustrated in Fig. 2 for comparison. The penalty functions and their first-order derivatives are listed in Table 1.

In order that the smoothed penalty functions maintain the effective sparsity-promoting behavior of the original non-differentiable penalty function,  $\epsilon$  should be set to a sufficiently small value. On the other hand,  $\epsilon$  should be large enough so as to avoid the afore-mentioned numerical issues arising in some optimization algorithms (in particular, the MM algorithm developed below). Fortunately, we have found that the numerical issues are reliably avoided with  $\epsilon$  small enough so that its impact on the optimal solution is negligible. The same holds true for the asymmetric functions discussed below. We have found that  $\epsilon = 10^{-5}$  works well with both  $\phi_B$  and  $\phi_C$ .

### 3.4. Asymmetric penalty functions

For some applications, the signal  $\mathbf{x}$  may be known to be sparse in an asymmetric manner. For example, it is known that chromatogram data have positive peaks above a relatively flat baseline. In such cases, it is preferable to use an asymmetric penalty function that penalizes positive and negative values differently, as in [14, 33, 32]. We start with the function  $\theta : \mathbb{R} \rightarrow \mathbb{R}$  defined by

$$\theta(x; r) = \begin{cases} x, & x \geq 0 \\ -rx, & x < 0 \end{cases} \quad (24)$$

where  $r > 0$  is a parameter. The function  $\theta(x; r)$  is convex in  $x$  and penalizes the positive and negative amplitude asymmetrically. If  $r = 1$ , then  $\theta(x; r) = \phi_A(x) = |x|$ .

The definition (24) has the same drawback as (21), that is, it is non-differentiable at  $x = 0$ . To alleviate this, a differentiable version of (24) is also proposed. In contrast to the differentiable symmetric penalties  $\phi_B$  and  $\phi_C$ , the function we propose is of the form

$$\theta_\epsilon(x; r) = \begin{cases} x, & x > \epsilon \\ f(x), & |x| \leq \epsilon \\ -rx, & x < -\epsilon. \end{cases} \quad (25)$$

The function  $\theta_\epsilon(x; r)$  differs from  $\theta(x; r)$  in a way that, in the small interval  $[-\epsilon, \epsilon]$ , a function  $f(x)$  is defined. In Section 4.2, we will set  $f(x)$  to be a fixed second-order polynomial function and with this modification, 1)  $\theta_\epsilon(x; r)$  will remain convex in  $(-\infty, +\infty)$ ; 2)  $\theta_\epsilon(x; r)$  will be continuously differentiable on  $(-\infty, +\infty)$ .

Asymmetric penalty functions are also used in algorithm *backcor* [33, 32] to reflect the positivity of the peaks. We note that *backcor* uses non-convex penalties, while the method described here uses convex penalties (although, it can be modified to use non-convex penalties).

Table 1: Symmetric penalty functions and their derivatives.

	$\phi(x)$	$\phi'(x)$
$\phi_A(x)$	$ x $	$\text{sign}(x)$
$\phi_B(x)$	$\sqrt{ x ^2 + \epsilon}$	$\frac{x}{\sqrt{ x ^2 + \epsilon}}$
$\phi_C(x)$	$ x  - \epsilon \log( x  + \epsilon)$	$\frac{x}{ x  + \epsilon}$

## 4. Algorithms

### 4.1. Symmetric penalty functions

The first form of the proposed approach, BEADS, is defined through the minimization of the objective function  $F$  in (20), where  $\phi$  is a differentiable symmetric penalty function as in Table 1. In this section, we use the MM procedure (4) to derive an iterative algorithm for this optimization. Hence, we seek a majorizer  $G(\mathbf{x}, \mathbf{v})$  of  $F(\mathbf{x})$ . First, we find a majorizer  $g(x, v)$  for  $\phi(x) : \mathbb{R} \rightarrow \mathbb{R}$  such that

$$g(x, v) \geq \phi(x), \quad (26a)$$

$$g(v, v) = \phi(v), \quad \text{for all } x, v \in \mathbb{R}. \quad (26b)$$

Since  $\phi(x)$  is symmetric, we set  $g(x, v)$  to be an even second-order polynomial,

$$g(x, v) = mx^2 + b \quad (27)$$

where  $m$  and  $c$  are to be determined so as to satisfy (26). From (26), we have

$$g(v, v) = \phi(v) \quad \text{and} \quad g'(v, v) = \phi'(v), \quad (28)$$

that is,  $g(x, v)$  and its derivative should agree with  $\phi(x)$  at  $x = v$ . Using (27) and (28), (26) becomes

$$mv^2 + b = \phi(v) \quad \text{and} \quad 2mv = \phi'(v). \quad (29)$$

Solving for  $m$  and  $b$  we obtain

$$m = \frac{\phi'(v)}{2v} \quad \text{and} \quad b = \phi(v) - \frac{v}{2}\phi'(v). \quad (30)$$

Substituting (30) in (27), we obtain

$$g(x, v) = \frac{\phi'(v)}{2v}x^2 + \phi(v) - \frac{v}{2}\phi'(v) \quad (31)$$

which gives

$$\sum_n g(x_n, v_n) = \sum_n \left[ \frac{\phi'(v_n)}{2v_n}x_n^2 + \phi(v_n) - \frac{v_n}{2}\phi'(v_n) \right] \quad (32)$$

$$= \frac{1}{2}\mathbf{x}^\top [\Lambda(\mathbf{v})] \mathbf{x} + c(\mathbf{v}) \quad (33)$$

$$\geq \sum_{n=0}^{N_i-1} \phi(x_n) \quad (34)$$

where  $\Lambda(\mathbf{v})$  denotes a diagonal matrix with diagonal elements

$$[\Lambda(\mathbf{v})]_{n,n} = \frac{\phi'(v_n)}{v_n} \quad (35)$$

and  $c(\mathbf{v})$  is the scalar,

$$c(\mathbf{v}) = \sum_n \left[ \phi(v_n) - \frac{v_n}{2} \phi'(v_n) \right]. \quad (36)$$

Using (32) to (36), we can write

$$\begin{aligned} \sum_{i=0}^M \lambda_i \sum_{n=0}^{N_i-1} g([\mathbf{D}_i \mathbf{x}]_n, [\mathbf{D}_i \mathbf{v}]_n) &= \sum_{i=0}^M \left[ \frac{\lambda_i}{2} (\mathbf{D}_i \mathbf{x})^\top [\Lambda(\mathbf{D}_i \mathbf{v})] (\mathbf{D}_i \mathbf{x}) + c_i(\mathbf{v}) \right] \\ &\geq \sum_{i=0}^M \lambda_i \sum_{n=0}^{N_i-1} \phi([\mathbf{D}_i \mathbf{x}]_n) \end{aligned} \quad (37)$$

where  $\Lambda(\mathbf{D}_i \mathbf{v})$  are diagonal matrices,

$$[\Lambda(\mathbf{D}_i \mathbf{v})]_{n,n} = \frac{\phi'([\mathbf{D}_i \mathbf{v}]_n)}{[\mathbf{D}_i \mathbf{v}]_n} \quad (38)$$

and  $c_i(\mathbf{v})$  are scalars,

$$c_i(\mathbf{v}) = \sum_n \left[ \phi([\mathbf{D}_i \mathbf{v}]_n) - \frac{[\mathbf{D}_i \mathbf{v}]_n}{2} \phi'([\mathbf{D}_i \mathbf{v}]_n) \right]. \quad (39)$$

Equality holds when  $\mathbf{x} = \mathbf{v}$ . Equation (37) implies that

$$G(\mathbf{x}, \mathbf{v}) = \frac{1}{2} \|\mathbf{H}(\mathbf{y} - \mathbf{x})\|_2^2 + \sum_{i=0}^M \left[ \frac{\lambda_i}{2} (\mathbf{D}_i \mathbf{x})^\top [\Lambda(\mathbf{D}_i \mathbf{v})] (\mathbf{D}_i \mathbf{x}) \right] + c(\mathbf{v}) \quad (40)$$

is a majorizer for  $F$  in (20). Minimizing  $G(\mathbf{x}, \mathbf{v})$  with respect to  $\mathbf{x}$  leads to the explicit solution

$$\mathbf{x} = \left( \mathbf{H}^\top \mathbf{H} + \sum_{i=0}^M \lambda_i \mathbf{D}_i^\top [\Lambda(\mathbf{D}_i \mathbf{v})] \mathbf{D}_i \right)^{-1} \mathbf{H}^\top \mathbf{H} \mathbf{y}. \quad (41)$$

Equation (41) explains why we use a differentiable penalty function. Suppose that the penalty function  $\phi$  were taken to be the absolute value function, i.e.,  $\phi(x) = |x|$ , then (38) becomes

$$[\Lambda(\mathbf{D}_i \mathbf{v})]_{n,n} = \frac{1}{[\mathbf{D}_i \mathbf{v}]_n} \text{sign}([\mathbf{D}_i \mathbf{v}]_n). \quad (42)$$

As the iterative algorithms progresses, some values  $[\mathbf{D}_i \mathbf{v}]_n$  go to zero due to the sparsifying property of the penalty. Consequently, ‘divide-by-zero errors’ are encountered in the implementation of (42). This is due to the absolute value function being non-differentiable at zero. The use of smoothed (differentiable) penalty functions such as those in Table 1, avoids this issue. For example, if we let  $\phi = \phi_B$ , then (38) becomes

$$[\Lambda(\mathbf{D}_i \mathbf{v})]_{n,n} = \frac{1}{\sqrt{|[\mathbf{D}_i \mathbf{v}]_n|^2 + \epsilon}} \quad (43)$$

and if we let  $\phi = \phi_C$ , we obtain

$$[\Lambda(\mathbf{D}_i \mathbf{v})]_{n,n} = \frac{1}{|[\mathbf{D}_i \mathbf{v}]_n| + \epsilon}. \quad (44)$$

Another issue in implementing (41) resides in the computational complexity of solving the linear system represented by the matrix inverse. The computational cost increases dramatically with the length of the signal  $\mathbf{y}$ . To address this, recall from (17) that we take the high-pass filter  $\mathbf{H}$  to have the form  $\mathbf{H} = \mathbf{B} \mathbf{A}^{-1}$ . Hence, (41) can be written as

$$\mathbf{x} = \left( \mathbf{A}^{-\top} \mathbf{B}^\top \mathbf{B} \mathbf{A}^{-1} + \sum_{i=0}^M \lambda_i \mathbf{D}_i^\top [\Lambda(\mathbf{D}_i \mathbf{v})] \mathbf{D}_i \right)^{-1} \mathbf{A}^{-\top} \mathbf{B}^\top \mathbf{B} \mathbf{A}^{-1} \mathbf{y} \quad (45)$$

$$= \mathbf{A} \left( \mathbf{B}^\top \mathbf{B} + \mathbf{A}^\top \left( \sum_{i=0}^M \lambda_i \mathbf{D}_i^\top [\Lambda(\mathbf{D}_i \mathbf{v})] \mathbf{D}_i \right) \mathbf{A} \right)^{-1} \mathbf{B}^\top \mathbf{B} \mathbf{A}^{-1} \mathbf{y} \quad (46)$$

$$= \mathbf{A} \mathbf{Q}^{-1} \mathbf{B}^\top \mathbf{B} \mathbf{A}^{-1} \mathbf{y} \quad (47)$$



Table 2: Algorithm to minimize cost function (20).

---

Input: $\mathbf{y}$ , $\mathbf{A}$ , $\mathbf{B}$ , $\lambda_i$ , $i = 0, \dots, M$
1. $\mathbf{b} = \mathbf{B}^\top \mathbf{B} \mathbf{A}^{-1} \mathbf{y}$
2. $\mathbf{x} = \mathbf{y}$ (Initialization)
Repeat
3. $[\mathbf{\Lambda}_i]_{n,n} = \frac{\phi'([\mathbf{D}_i \mathbf{x}]_n)}{[\mathbf{D}_i \mathbf{x}]_n}, \quad i = 0, \dots, M,$
4. $\mathbf{M} = \sum_{i=0}^M \lambda_i \mathbf{D}_i^\top \mathbf{\Lambda}_i \mathbf{D}_i$
5. $\mathbf{Q} = \mathbf{B}^\top \mathbf{B} + \mathbf{A}^\top \mathbf{M} \mathbf{A}$
6. $\mathbf{x} = \mathbf{A} \mathbf{Q}^{-1} \mathbf{b}$
Until converged
8. $\mathbf{f} = \mathbf{y} - \mathbf{x} - \mathbf{B} \mathbf{A}^{-1} (\mathbf{y} - \mathbf{x})$
Output: $\mathbf{x}$ , $\mathbf{f}$

---

where

$$\mathbf{Q} = \mathbf{B}^\top \mathbf{B} + \mathbf{A}^\top \left( \sum_{i=0}^M \lambda_i \mathbf{D}_i^\top [\mathbf{\Lambda}(\mathbf{D}_i \mathbf{v})] \mathbf{D}_i \right) \mathbf{A}. \quad (48)$$

Note that  $\mathbf{Q}$  is a banded matrix. (Sums and product of banded matrices are banded.) Hence,  $\mathbf{x}$  can be obtained with high computational efficiency and low memory using fast solvers for banded systems [39, Sec. 2.4]. Note that this can not be achieved when  $\mathbf{H}$  is written  $\mathbf{H} = \mathbf{A}^{-1} \mathbf{B}$  as in [45].

#### 4.2. Asymmetric and symmetric penalty functions

To account for the positivity of the chromatogram peaks, we propose a second form of BEADS which uses an asymmetric penalty that penalizes negative values of  $\mathbf{x}$  more than positive values. Hence, in this section we use the MM approach to derive an algorithm solving the problem:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left\{ F(\mathbf{x}) = \frac{1}{2} \|\mathbf{H}(\mathbf{y} - \mathbf{x})\|_2^2 + \lambda_0 \sum_{n=0}^{N-1} \theta_\epsilon(x_n; r) + \sum_{i=1}^M \lambda_i \sum_{n=0}^{N_i-1} \phi([\mathbf{D}_i \mathbf{x}]_n) \right\}. \quad (49)$$

The penalty  $\theta_\epsilon(x_n; r)$ , given by (25), is a differentiable version of the asymmetric penalty function  $\theta(x_n; r)$ . The function  $\phi$  is a differentiable symmetric penalty functions such as  $\phi_B$  or  $\phi_C$  in Table 1.

To obtain a majorizer for  $F$ , we first find a majorizer for  $\theta(x; r) : \mathbb{R} \rightarrow \mathbb{R}$ , defined in (24). We seek a majorizer as illustrated in Fig. 3, i.e., a function of the form

$$g(x, v) = ax^2 + bx + c \quad (50)$$

which is an upper bound of  $\theta(x; r)$  that not only coincides with  $\theta(x; r)$  at  $x = v$ , but also coincides with  $\theta(x; r)$  at  $x = s$  on the opposite side of zero as illustrated in Fig. 3. That is,

$$g(v, v) = \theta(v; r), \quad g'(v, v) = \theta'(v; r), \quad (51)$$

$$g(s, v) = \theta(s; r), \quad g'(s, v) = \theta'(s; r). \quad (52)$$

The differentiation is with respect to the first argument. Note that  $a, b, c, s$  are all functions of  $v$ . Solving for them gives

$$a = \frac{1+r}{4|v|}, \quad b = \frac{1-r}{2}, \quad c = \frac{(1+r)|v|}{4}, \quad s = -v. \quad (53)$$

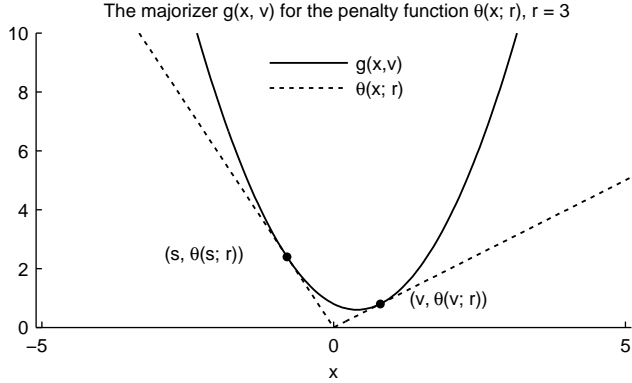


Figure 3: Asymmetric penalty function and its majorizer. ( $v = 0.8, r = 3$ )

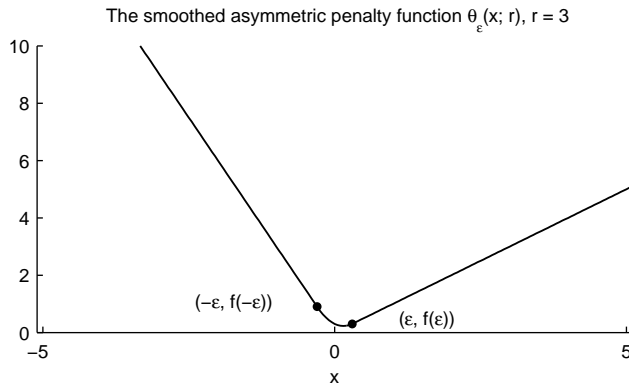


Figure 4: Continuously differentiable asymmetric penalty function  $\theta_\epsilon(x; r)$  in (55). The function is a second-order polynomial on  $[-\epsilon, \epsilon]$ .

Substituting (53) in (50), we obtain a majorizer for  $\theta(x; r)$ ,

$$g(x, v) = \frac{1+r}{4|v|}x^2 + \frac{1-r}{2}x + \frac{(1+r)|v|}{4}. \quad (54)$$

Again, an issue with (54) is that, as  $v$  approaches zero, ‘divide-by-zero’ numerical errors arises. To address this issue, we define  $\theta_\epsilon(x; r)$  to be a continuously differentiable approximation to  $\theta(x; r)$ . In a neighborhood of  $x = 0$ , we define  $\theta_\epsilon(x; r)$  to be the second order polynomial (54) with  $v = \epsilon$ , i.e.,

$$\theta_\epsilon(x; r) = \begin{cases} x, & x > \epsilon \\ \frac{1+r}{4\epsilon}x^2 + \frac{1-r}{2}x + \epsilon\frac{1+r}{4}, & |x| \leq \epsilon \\ -rx, & x < -\epsilon \end{cases} \quad (55)$$

where  $\epsilon > 0$  is a small constant. The new function  $\theta_\epsilon(x; r)$ , illustrated in Fig. 4, behaves similarly to  $\theta(x; r)$  but is continuously differentiable.

The majorizer given by (54) is still valid for (55) in the domain of  $(-\infty, \epsilon)$  and  $(-\epsilon, +\infty)$ . In the domain  $[-\epsilon, \epsilon]$ , we use  $\theta_\epsilon(x; r)$  itself as its majorizer. Hence, a majorizer of  $\theta_\epsilon(x; r)$  is found to be

$$g_0(x, v) = \begin{cases} \frac{1+r}{4|v|}x^2 + \frac{1-r}{2}x + |v|\frac{1+r}{4}, & |v| > \epsilon \\ \frac{1+r}{4\epsilon}x^2 + \frac{1-r}{2}x + \epsilon\frac{1+r}{4}, & |v| \leq \epsilon. \end{cases} \quad (56)$$

A proof that  $g_0(x, v)$  is a majorizer of  $\theta_\epsilon(x; r)$  is given in Appendix A.

Using (56), we then have

$$\sum_{n=0}^{N-1} g_0(x_n, v_n) = \mathbf{x}^\top [\boldsymbol{\Gamma}(\mathbf{v})] \mathbf{x} + \mathbf{b}^\top \mathbf{x} + c(\mathbf{v}) \quad (57)$$

$$\geq \sum_{n=0}^{N-1} \theta_\epsilon(x_n; r) \quad (58)$$

where  $\boldsymbol{\Gamma}(\mathbf{v})$  is a diagonal matrix with diagonal elements

$$[\boldsymbol{\Gamma}(\mathbf{v})]_{n,n} = \begin{cases} \frac{1+r}{4|v_n|}, & |v_n| \geq \epsilon \\ \frac{1+r}{4\epsilon}, & |v_n| \leq \epsilon \end{cases} \quad (59)$$

and  $\mathbf{b}$  is a vector with elements

$$[\mathbf{b}]_n = \frac{1-r}{2} \quad (60)$$

and  $c(\mathbf{v})$  is a scalar that does not depend on  $\mathbf{x}$ .

Using (37) and (57), we find that a majorizer for  $F$  in (49) is given by:

$$G(\mathbf{x}, \mathbf{v}) = \frac{1}{2} \|\mathbf{H}(\mathbf{y} - \mathbf{x})\|_2^2 + \lambda_0 \mathbf{x}^\top [\boldsymbol{\Gamma}(\mathbf{v})] \mathbf{x} + \lambda_0 \mathbf{b}^\top \mathbf{x} + \sum_{i=1}^M \left[ \frac{\lambda_i}{2} (\mathbf{D}_i \mathbf{x})^\top [\Lambda(\mathbf{D}_i \mathbf{v})] (\mathbf{D}_i \mathbf{x}) \right] + c(\mathbf{v}). \quad (61)$$

Minimizing  $G(\mathbf{x}, \mathbf{v})$  with respect to  $\mathbf{x}$  yields

$$\mathbf{x} = \left[ \mathbf{H}^\top \mathbf{H} + 2\lambda_0 \boldsymbol{\Gamma}(\mathbf{v}) + \sum_{i=1}^M \lambda_i \mathbf{D}_i^\top [\Lambda(\mathbf{D}_i \mathbf{v})] \mathbf{D}_i \right]^{-1} (\mathbf{H}^\top \mathbf{H} \mathbf{y} - \lambda_0 \mathbf{b}). \quad (62)$$

Using  $\mathbf{H} = \mathbf{B}\mathbf{A}^{-1}$  as in (45), we can write (62) as

$$\mathbf{x} = \mathbf{A}\mathbf{Q}^{-1} (\mathbf{B}^\top \mathbf{B} \mathbf{A}^{-1} \mathbf{y} - \lambda_0 \mathbf{A}^\top \mathbf{b}) \quad (63)$$

where  $\mathbf{Q}$  is the banded matrix,

$$\mathbf{Q} = \mathbf{B}^\top \mathbf{B} + \mathbf{A}^\top \mathbf{M} \mathbf{A}, \quad (64)$$

and  $\mathbf{M}$  is the banded matrix,

$$\mathbf{M} = 2\lambda_0 \boldsymbol{\Gamma}(\mathbf{v}) + \sum_{i=1}^M \lambda_i \mathbf{D}_i^\top [\Lambda(\mathbf{D}_i \mathbf{v})] \mathbf{D}_i. \quad (65)$$

As above, the system of equations represented by  $\mathbf{Q}$  in (63) is banded and thus a fast solver for banded systems can be used to implement the MM update equation.

Using the above equations, the MM iteration takes the form:

$$\mathbf{M}^{(k)} = 2\lambda_0 \boldsymbol{\Gamma}(\mathbf{x}^{(k)}) + \sum_{i=1}^M \lambda_i \mathbf{D}_i^\top [\Lambda(\mathbf{D}_i \mathbf{x}^{(k)})] \mathbf{D}_i. \quad (66)$$

$$\mathbf{Q}^{(k)} = \mathbf{B}^\top \mathbf{B} + \mathbf{A}^\top \mathbf{M}^{(k)} \mathbf{A} \quad (67)$$

$$\mathbf{x}^{(k+1)} = \mathbf{A}[\mathbf{Q}^{(k)}]^{-1} (\mathbf{B}^\top \mathbf{B} \mathbf{A}^{-1} \mathbf{y} - \lambda_0 \mathbf{A}^\top \mathbf{b}) \quad (68)$$

The complete BEADS algorithm for solving cost function (49) is detailed in Table 3. The run-time of BEADS, with a fixed number of 30 iterations, as implemented in MATLAB, is tabulated in Table 4. Note that applying the algorithm to a signal of length 10,000 requires less than one second of computation. Run-times were measured using MATLAB version 2010b on an i7 3.2GHz PC with 8 GB of RAM.

The convergence property of the MM approach is discussed in [25, 30]. In particular, when the objective function is strictly convex, as is the case in (20) and (49), the MM algorithm is guaranteed to converge to the unique optimum.

Table 3: BEADS algorithm to minimize cost function (49).

---

Input:  $\mathbf{y}$ ,  $r \geq 1$ ,  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\lambda_i$ ,  $i = 0, \dots, M$

1.  $[\mathbf{b}]_n = \frac{1-r}{2}$
2.  $\mathbf{d} = \mathbf{B}^T \mathbf{B} \mathbf{A}^{-1} \mathbf{y} - \lambda_0 \mathbf{A}^T \mathbf{b}$
3.  $\mathbf{x} = \mathbf{y}$  (Initialization)

Repeat

4.  $[\mathbf{\Gamma}]_{n,n} = \begin{cases} \frac{1+r}{4|x_n|}, & |x_n| \geq \epsilon \\ \frac{1+r}{4\epsilon}, & |x_n| \leq \epsilon \end{cases}$
5.  $[\mathbf{\Lambda}_i]_{n,n} = \frac{\phi'([\mathbf{D}_i \mathbf{x}]_n)}{[\mathbf{D}_i \mathbf{x}]_n}, \quad i = 0, \dots, M,$
6.  $\mathbf{M} = 2\lambda_0 \mathbf{\Gamma} + \sum_{i=1}^M \lambda_i \mathbf{D}_i^T \mathbf{\Lambda}_i \mathbf{D}_i$
7.  $\mathbf{Q} = \mathbf{B}^T \mathbf{B} + \mathbf{A}^T \mathbf{M} \mathbf{A}$
8.  $\mathbf{x} = \mathbf{A} \mathbf{Q}^{-1} \mathbf{d}$

Until converged

9.  $\mathbf{f} = \mathbf{y} - \mathbf{x} - \mathbf{B} \mathbf{A}^{-1} (\mathbf{y} - \mathbf{x})$

Output:  $\mathbf{x}$ ,  $\mathbf{f}$

---

Table 4: Run-time (in sec.) of BEADS for  $N$ -point data.

$N$	$10^2$	$10^3$	$10^4$	$10^5$
BEADS	0.040	0.124	0.677	7.266

## 5. Experiments

### 5.1. Baseline correction of simulated chromatograms in Gaussian noise

This example illustrates the use of BEADS for the removal of baselines in simulated chromatograms. The algorithm is compared with two other methods, *airPLS* [59] and *backcor* [32, 33]. We conduct two experiments. In both experiments, the chromatogram peaks are generated as described in [27] (similar also peak simulation of [59, 32, 33]). Namely, the signal  $\mathbf{x}$  is created as a superposition of Gaussian functions with different amplitudes, positions and widths. For Experiments 1 and 2, we simulate the baseline in two ways, respectively:

1. Type 1 simulated baseline: as in [27, 33], a baseline signal is generated as the sum of an order- $p$  polynomial and a sinusoidal signal of frequency  $f$ . Specifically, for each realization, the order  $p$  and frequency  $f$  are uniformly distributed in a prescribed range.
2. Type 2 simulated baseline: a baseline is generated as a stationary random process with a power spectrum limited to a low-pass range of  $[0, f_c]$  Hz. Specifically, such a signal is obtained by applying a low-pass filter with cut-off frequency  $f_c$  to a white Gaussian process.

Table 5: Experiment 1. The mean and standard deviation (std) of SNR. The table shows the result when input SNR is 0dB, 10dB and 20dB.

	0 dB		10 dB		20 dB	
	mean	std	mean	std	mean	std
BEADS	28.1	8.52	32.64	8.02	38.33	6.74
backcor	24.91	9.75	31.27	8.33	36.47	6.53
airPLS	20.26	9.65	22.54	10.15	26.71	7.76

Sample realizations are illustrated in Figs. 5 and 6 for Experiments 1 and 2, respectively. White Gaussian noise is added to each realization. We generate 500 realizations and vary the variance of the signal such that the SNR ranges from  $-5$ dB to  $25$ dB. For each realization and SNR level, we apply the BEADS, airPLS, and backcor algorithms to estimate the baseline. The accuracy of the baseline estimation is evaluated by computing the SNR of the output, i.e, the energy of the generated baseline divided by the energy of the difference between the generated and the estimated baselines, measured in decimal.

For Type 1 simulated baselines (Experiment 1), the results are shown in Fig. 7 and Table 5. Algorithm backcor and BEADS outperform airPLS. Since the morphology of Type 1 baselines is relatively simple, backcor and BEADS perform quite similarly; however, BEADS yields a slightly smaller error on average. For Type 2 simulated baselines (Experiment 2), the results are shown in Fig. 8 and Table 6. As in Experiment 1, airPLS yields the greatest error and BEADS the least error, on average; but, the improvement of BEADS in comparison with backcor and airPLS is more significant than in Experiment 1. BEADS is better able to estimate the more challenging Type 2 baseline, because it models the baseline as a low-pass signal rather than as a parametric function (e.g., polynomial).

To clarify the use of BEADS as applied in this example: we have set the low-pass filter  $\mathbf{L} = \mathbf{I} - \mathbf{H}$  to be a second order filter ( $d = 1$  in [45]) with a cut-off frequency of  $f_c = 0.0035$  cycles/sample. Although the variance of the noise can affect the choice of  $f_c$ , in practice, we have the effect insignificant.

We model the chromatogram peaks as having two sparse derivatives, i.e., we set  $M = 2$  in the cost function  $F$  in (49). We use the differentiable penalty  $\phi_B$  with  $\epsilon = 10^{-5}$ . The regularization parameters  $\lambda_i$ ,  $i = 0, 1, 2$  should match the sparsity of  $\mathbf{x}$  and its derivatives. For each derivative order,  $i$ : the sparser  $\mathbf{D}_i \mathbf{x}$ , the larger the corresponding  $\lambda_i$  should be set. For an individual signal, a simple rule is to choose  $\lambda_i$  inversely proportional to  $\|\mathbf{D}_i \mathbf{x}\|_1$  for  $i = 0, 1, 2$ , and to set the proportionality constant  $\alpha$  according to the noise variance. In this experiment, we have set  $\lambda_i$  to be inversely proportional to the empirical mean of  $\|\mathbf{D}_i \mathbf{x}\|_1$ ,  $i = 0, 1, 2$ , where the mean is computed over the 500 realizations. We manually tune the proportionality constant  $\alpha$  so as to minimize the average SNR. It is also conceivable that other approaches such as cross-validation or bootstrapping can be used to select the  $\lambda_i$  parameters.

The backcor algorithm requires two user-specified parameters: a threshold value and the order of the approximation polynomial. To set these parameters, we optimized them, via a search, to minimize the SNR. The algorithm, airPLS, also requires two user-specified parameters which we likewise set so as to minimize the SNR.

### 5.2. Baseline correction: Poisson observation process

With the variety of chromatographic detectors [53, p. 277–337], noise distributions may more aptly be characterized by Poisson statistics, under the denomination of shot noise [54, 55]. Gaussian fluctuations can be regarded as a limit of a Poisson process. In modern detectors the noise can often be modeled as Poisson distributed, proportional to the square root of intensity (peaks + baseline). More specifically, if at time  $n$  we denote the Poisson observation  $y(n)$ , the peak signal  $x(n)$  and the baseline  $f(n)$ , then  $y(n)$  may be modeled as  $y(n) = (P(n)/c)^2$  where  $c$  is a proportionality constant and  $P(n) \sim \text{Poisson}(\lambda(n))$  is a Poisson random variable with mean  $\lambda(n) = c\sqrt{x(n) + f(n)}$ .

Although the proposed algorithm is developed under the assumption that noise is additive Gaussian, we also test its performance when the observed data follows a Poisson model. A simulated signal is shown in

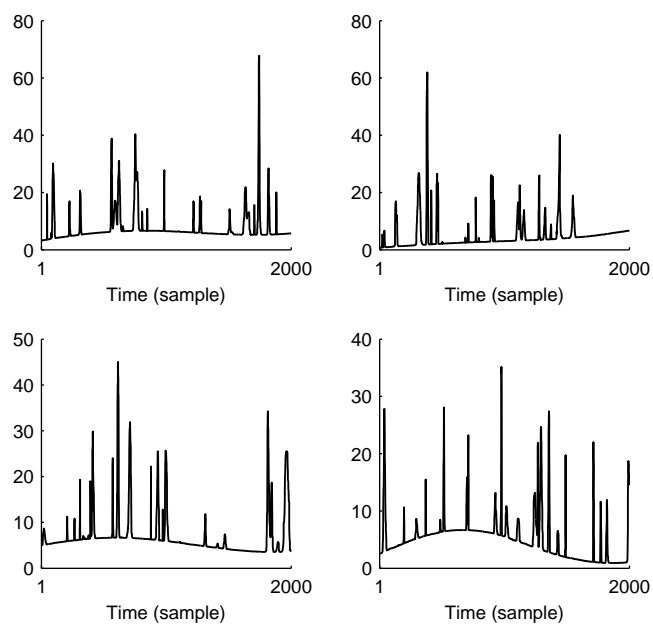


Figure 5: Simulated chromatograms with Type 1 baseline [27].

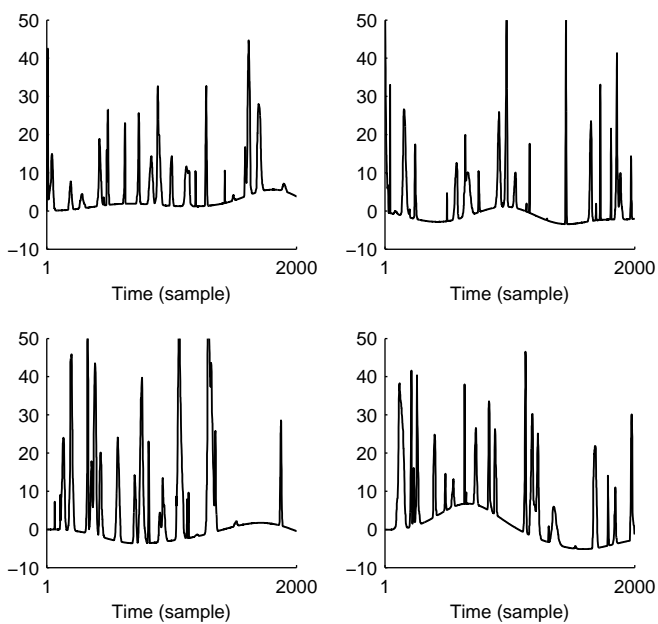


Figure 6: Simulated chromatograms with Type 2 baseline.

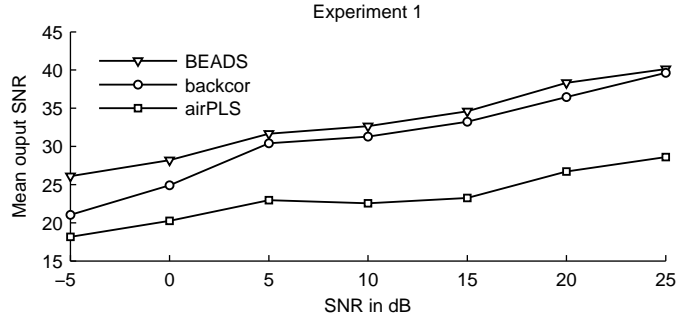


Figure 7: Experiment 1. Output SNR as a function of input SNR.

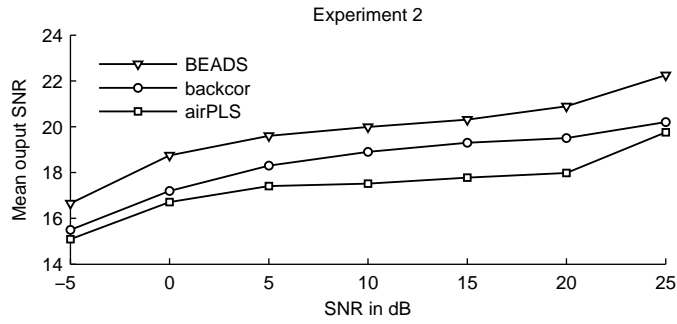


Figure 8: Experiment 2. Output SNR as a function of input SNR.

Table 6: Experiment 2. The mean and standard deviation (std) of SNR. The table shows the result when input SNR is 0dB, 10dB and 20dB.

	0 dB		10 dB		20 dB	
	mean	std	mean	std	mean	std
BEADS	18.75	3.71	19.99	3.17	20.89	3.32
backcor	17.20	4.57	18.93	3.74	19.54	3.18
airLPS	16.71	4.80	17.52	5.54	17.98	4.82

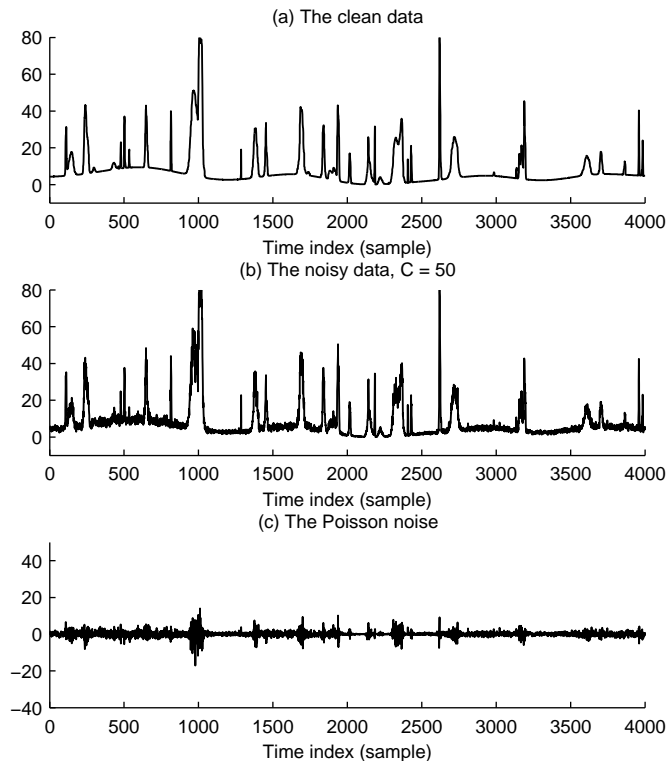


Figure 9: An example of chromatogram data (peaks + baseline) in Poisson noise.

Fig. 9(a); Fig. 9(b) and (c) show the Poisson observation and the difference between the observation and the noise-free simulated signal (peaks + baseline).

With this Poisson noise model, two experiments (3 and 4) are conducted following the same setup as Experiments 1 and 2 respectively. The noise level is quantified by the constant  $c$  and performance is evaluated using SNR. The results are detailed in Fig. 10 and Fig. 11. Similar to Experiment 1 and 2, BEADS uniformly outperforms the two methods to which it is compared.

### 5.3. Baseline correction of real chromatogram data

This example illustrates baseline correction using the proposed method, BEADS, as applied to real chromatogram data. The chromatogram data, from [59], is shown in gray in Fig. 12. The algorithms, BEADS, backcor, and airPLS, are applied to estimate the baseline. The parameters for backcor and airPLS were manually set so as to obtain their best possible result. Figs. 12(a)-(c) display the estimated baseline produced by each of the three methods, and (d)-(f) show the corresponding estimated peaks (obtained by subtracting the estimated baseline from the data). The three methods are able to capture the baseline trend. However, close examination shows that BEADS exhibits less distortion in some regions. In the interval 2200-2500, backcor overestimates the baseline, while airPLS slightly underestimates the baseline.

### 5.4. Joint baseline correction and denoising

Some methods, e.g., backcor and airPLS, have been developed specifically for baseline removal; while other methods have been devised for the reduction of random noise. However, in many operational scenarios, both baseline drift and noise are present in the measured data. In this case, BEADS can be used to jointly perform baseline correction and noise reduction.

For illustration, white Gaussian noise is added to the chromatogram signal from [59] (Fig. 13(a)). The new signal exhibits both baseline drift and additive noise. The output obtained using BEADS is illustrated in Fig. 13(b)-(d). (We have used BEADS parameters  $r = 6$ ,  $f_c = 0.006$  cycles/sample,  $d = 1$  and  $\phi = \phi_C$ .)



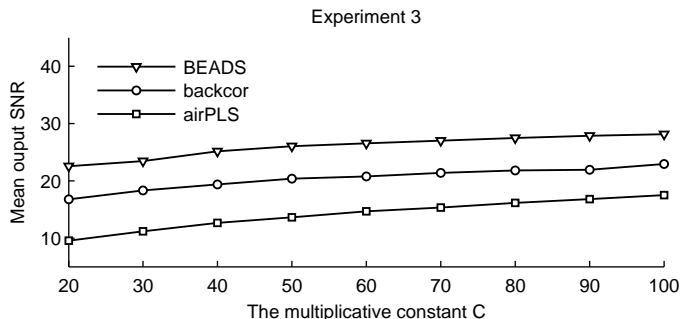


Figure 10: Experiment 3. Output SNR as a function of C.

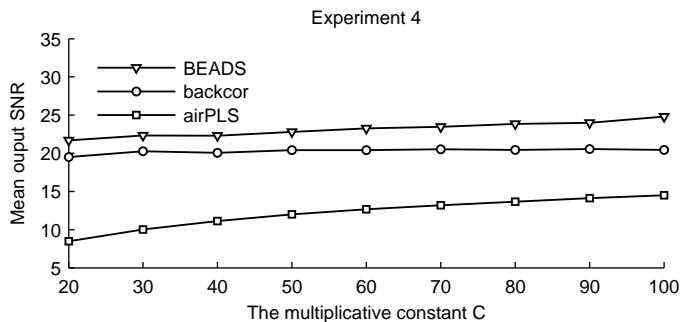


Figure 11: Experiment 4. Output SNR as a function of C.

Note that the estimated chromatogram peaks, illustrated in Fig. 13(b), are well delineated. The baseline is well estimated as illustrated in Fig. 13(c). The residual constitutes an estimate of the noise. The cost function history of the iterative algorithm, shown in Fig. 14, indicates that the algorithm converges in about 20 iterations.

## 6. Conclusion

This paper addresses the problems of chromatogram baseline correction and noise reduction. The approach, denoted ‘BEADS,’ is based on formulating a convex optimization problem designed to encapsulate non-parametric models of the baseline and the chromatogram peaks. Specifically, the baseline is modeled as a low-pass signal and the series of chromatogram peaks is modeled as sparse and as having sparse derivatives. Moreover, to account for the positivity of chromatogram peaks, both asymmetric and symmetric penalty functions are utilized (symmetric ones for the derivatives). In order that the iterative optimization algorithm be computationally efficient, use minimal memory, and can be applied to very long data series, we formulate the problem in terms of banded matrices. As such, the algorithm leverages the availability of fast solvers for banded linear systems and the majority of the computational work of the algorithm resides therein. Furthermore, due to the formulation of the problem as a convex problem and the properties of the majorization-minimization (MM) approach by which the iterative algorithm is derived, the proposed algorithm is guaranteed to converge regardless of its initialization.

The baseline correction performance is evaluated on simulated chromatogram data and compared with two state-of-art methods. In particular, the proposed method, BEADS, outperforms methods based on polynomial modeling when low-order polynomials are not efficient representations of the baseline. Experiments suggest that BEADS can better estimate the baselines of real chromatograms. Finally, since BEADS jointly estimates the baseline and the chromatogram peaks, it can be used to perform both baseline correction and noise reduction simultaneously.

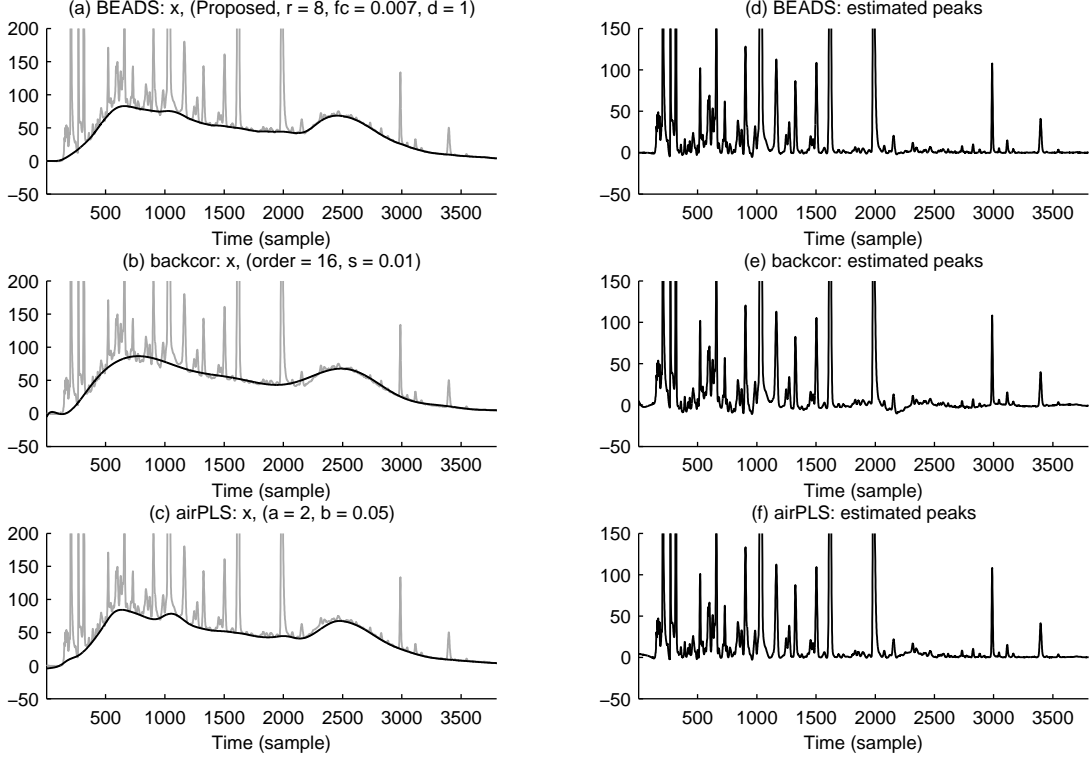


Figure 12: Left column: Baselines as estimated by each method. Right column: Estimated peaks (obtained by subtraction of estimated baseline from data). All three methods yield reasonable estimates of the baseline, yet BEADS appears to exhibit the least distortion.

The proposed method also provides a more general framework for different types of signals. By adjusting regularization parameters  $\lambda_i$  and penalty functions, the model can be customized for signals of different kinds. For example, an electrocardiography (ECG) signal is only sparse in its first several derivatives (the signal itself is not sparse) [37], therefore, by setting  $\lambda_0 = 0$  and choosing proper  $\lambda_i$ ,  $i = 1, 2, 3$  and using symmetric penalty functions, BEADS can be used for ECG baseline estimation and denoising.

BEADS uses symmetric/asymmetric convex penalty function to promote positivity of the estimated peaks, however, other penalty functions, such as non-convex penalties, may provide further improvements. In addition, due to the growing interest on hyphenated techniques in analytical chemistry [51, 56], with an increasing need laid on repeatability, extensions to two-dimensional chromatography are advisable. These will all be considered in future work.

## Appendix A.

We prove that  $g_0(x, v)$  in (56) is a majorizer of  $\theta_\epsilon(x; r)$ . Define

$$f(x) = \frac{1+r}{4\epsilon}x^2 + \frac{1-r}{2}x + \epsilon\frac{1+r}{4}, \quad |x| \leq \epsilon. \quad (\text{A.1})$$

For  $\theta_\epsilon(x; r)$  to be a majorizer of  $g_0(x, v)$ , we need to show that

$$g_0(x, v) = \frac{1+r}{4v}x^2 + \frac{1-r}{2}x + v\frac{1+r}{4} \geq f(x), \quad v > \epsilon \quad (\text{A.2})$$

$$g_0(x, v) = -\frac{1+r}{4v}x^2 + \frac{1-r}{2}x - v\frac{1+r}{4} \geq f(x), \quad v < -\epsilon. \quad (\text{A.3})$$

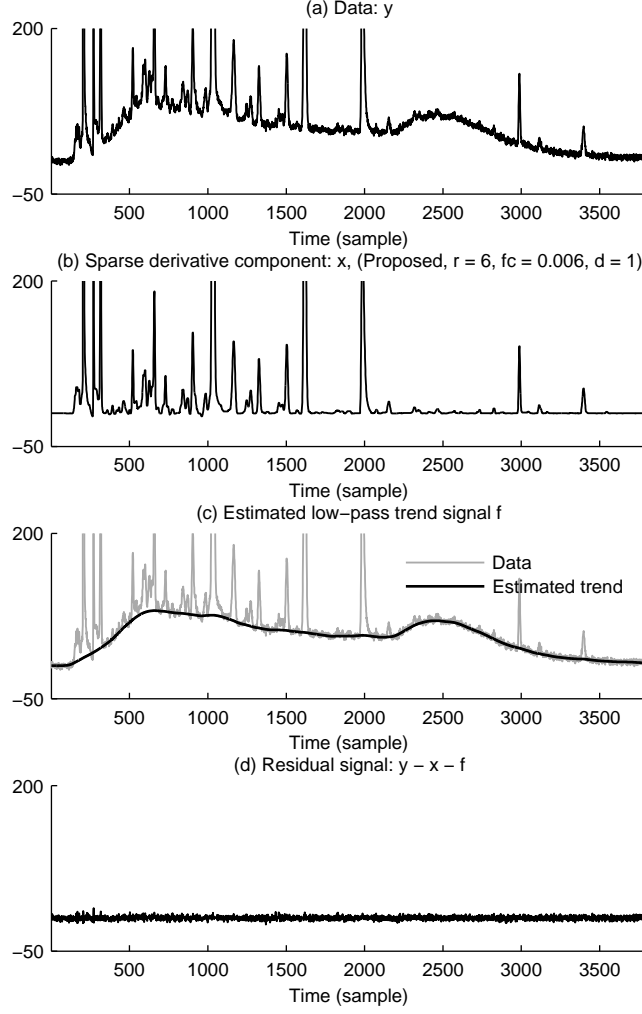


Figure 13: Processing of noisy chromatogram data using BEADS. (a) Chromatogram data with additive noise. (b) Estimated peaks. (c) Estimated baseline. (d) Residual.

If  $v > \epsilon$ , then

$$g_0(x, v) - f(x) = \left( \frac{1+r}{4v} - \frac{1+r}{4\epsilon} \right) x^2 + (v - \epsilon) \frac{1+r}{4}, \quad (\text{A.4})$$

which can be written as

$$g_0(x, v) - f(x) = \frac{(1+r)(v - \epsilon)(v\epsilon - x^2)}{4v\epsilon} > 0, \quad (\text{A.5})$$

where we have used  $v > \epsilon$  and  $|x| \leq \epsilon$ .

If  $v < -\epsilon$ , then

$$g_0(x, v) - f(x) = \left( -\frac{1+r}{4v} - \frac{1+r}{4\epsilon} \right) x^2 - (v + \epsilon) \frac{1+r}{4}, \quad (\text{A.6})$$

which can be written as

$$g_0(x, v) - f(x) = -\frac{(1+r)(v + \epsilon)(v\epsilon + x^2)}{4v\epsilon} > 0, \quad (\text{A.7})$$

where we have used  $v < -\epsilon$  and  $|x| \leq \epsilon$ .

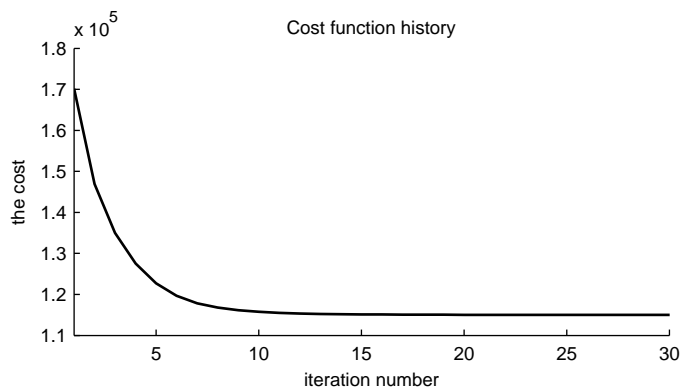


Figure 14: Cost function history of the BEADS algorithm.

## References

- [1] V. J. Barwick. Sources of uncertainty in gas chromatography and high-performance liquid chromatography. *J. Chrom. A*, 849(1):13–33, Jul. 1999.
- [2] J. M. Bioucas-Dias and M. A. T. Figueiredo. An iterative algorithm for linear inverse problems with compound regularizers. *Proc. Int. Conf. Image Process.*, pages 685–688, Oct. 12–15, 2008.
- [3] H. F. M. Boelens, R. J. Dijkstra, P. H. C. Eilers, F. Fitzpatrick, and J. A. Westerhuis. New background correction method for liquid chromatography with diode array detection, infrared spectroscopic detection and Raman spectroscopic detection. *J. Chrom. A*, 1057(1-2):21–30, 2004.
- [4] L. M. Briceño-Arias, P. L. Combettes, J.-C. Pesquet, and N. Pustelnik. Proximal algorithms for multi-component image processing. *J. Math. Imaging Vis.*, 41(1):3–22, Sep. 2011.
- [5] C. D. Brown, L. Vega-Montoto, and P. D. Wentzell. Derivative preprocessing and optimal corrections for baseline drift in multivariate calibration. *Appl. Spectrosc.*, 54(7):1055–1068, 2000.
- [6] S. D. Brown, R. Tauler, and B. Walczak, editors. *Comprehensive Chemometrics: Chemical and Biochemical Data Analysis*. Elsevier, 2009.
- [7] S. Cappadona, F. Levander, M. Jansson, P. James, S. Cerutti, and L. Pattini. Wavelet-based method for noise characterization and rejection in high-performance liquid chromatography coupled to mass spectrometry. *Anal. Chem.*, 80(13):4960–4968, 2008.
- [8] T. F. Chan, S. Osher, and J. Shen. The digital TV filter and nonlinear denoising. *IEEE Trans. Image Process.*, 10(2):231–241, Feb. 2001.
- [9] F.-T. Chau and A. K.-M. Leung. Application of wavelet transform in processing chromatographic data. In B. Walczak, editor, *Wavelets in Chemistry*, volume 22 of *Data Handling in Science and Technology*, chapter 9, pages 205–223. Elsevier, 2000.
- [10] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.
- [11] R. Danielsson, D. Bylund, and K. E. Markides. Matched filtering with background suppression for improved quality of base peak chromatograms and mass spectra in liquid chromatography-mass spectrometry. *Anal. Chim. Acta*, 454(2):167–184, 2002.
- [12] J. J. de Rooi and P. H. C. Eilers. Mixture models for baseline estimation. *Chemometr. Intell. Lab. Syst.*, 117:56–60, Aug. 2012.
- [13] P. H. C. Eilers. A perfect smoother. *Anal. Chem.*, 75(14):3631–3836, May 2003.

- [14] P. H. C. Eilers. Unimodal smoothing. *J. Chemometrics*, 19(5-7):317–328, May 2005.
- [15] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Appl. Comp. Harm. Analysis*, 19(3):340–358, 2005.
- [16] M. J. Fadili, J.-L. Starck, J. Bobin, and Y. Moudden. Image decomposition and separation using sparse representations: An overview. *Proc. IEEE*, 98(6):983–994, Jun. 2010.
- [17] A. Felinger, editor. *Data analysis and signal processing in chromatography*. Elsevier, 1998.
- [18] M. A. T. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak. Majorization-minimization algorithms for wavelet-based image restoration. *IEEE Trans. Image Process.*, 16(12):2980–2991, Dec. 2007.
- [19] R. Fischer, K. Hanson, V. Dose, and W. von der Linden. Background estimation in experimental spectra. *Phys. Rev. E*, 61(2):1152–1160, Feb. 2000.
- [20] M. Fredriksson, P. Petersson, M. Jörntén-Karlsson, B.-O. Axelsson, and D. Bylund. An objective comparison of pre-processing methods for enhancement of liquid chromatography-mass spectrometry data. *J. Chrom. A*, 1172(2):135–150, 2007.
- [21] F. Gan, G. Ruan, and J. Mo. Baseline correction by improved iterative polynomial fitting with automatic threshold. *Chemometr. Intell. Lab. Syst.*, 82(1-2):59–65, 2006.
- [22] L. Granai and P. Vanderghenst. Sparse decomposition over multi-component redundant dictionaries. In *IEEE Workshop Multimedia Signal Process.*, pages 494–497, Siena, Italy, 29 Sep.-1 Oct., 2004.
- [23] S. Gulam Razul, W. J. Fitzgerald, and C. Andrieu. Bayesian model selection and parameter estimation of nuclear emission spectra using RJMCMC. *Nucl. Instrum. Meth. Phys. Res. A*, 497(2-3):492–510, Feb. 2003.
- [24] Y. Hu, T. Jiang, A. Shen, W. Li, X. Wang, and J. Hu. A background elimination method based on wavelet transform for Raman spectra. *Chemometr. Intell. Lab. Syst.*, 85(1):94–101, 2007.
- [25] D. R. Hunter and K. Lange. A tutorial on MM algorithms. *Am. Stat.*, 58(1):30–37, Feb. 2004.
- [26] M. A. Kneen and H. J. Annegarn. Algorithm for fitting XRF, SEM and PIXE X-ray spectra backgrounds. *Nucl. Instrum. Methods Phys. Res. Sect. B*, 109:201–213, 1996.
- [27] Ł. Komsta. Comparison of several methods of chromatographic baseline removal with a new approach based on quantile regression. *Chromatographia*, 73:721–731, 2011.
- [28] G. Kutyniok. Geometric separation by single-pass alternating thresholding. *Appl. Comp. Harm. Analysis*, 36(1):23–50, Jan. 2014.
- [29] J. M. Laeven and H. C. Smit. Optimal peak area determination in the presence of noise. *Anal. Chim. Acta*, 176:77–104, 1985.
- [30] K. Lange, D. R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *J. Comput. Graph. Stat.*, 9(1):1–20, Mar. 2000.
- [31] Y. Liu, W. Cai, and X. Shao. Intelligent background correction using an adaptive lifting wavelet. *Chemometr. Intell. Lab. Syst.*, 125:11–17, Jun. 2013.
- [32] V. Mazet, D. Brie, and J. Idier. Baseline spectrum estimation using half-quadratic minimization. In *Proc. Eur. Sig. Image Proc. Conf.*, pages 305–308, Vienna, Austria, Sep. 7-10, 2004.
- [33] V. Mazet, C. Carteret, D. Brie, J. Idier, and B. Humbert. Background removal from spectra by designing and minimising a non-quadratic cost function. *Chemometr. Intell. Lab. Syst.*, 76(2):121–133, 2005.
- [34] A. D. McNaught and A. Wilkinson. *IUPAC. Compendium of Chemical Terminology*. Wiley Blackwell, 2nd edition, Aug. 2009. The "Gold Book".

- [35] D. A. McNulty and H. J. H. MacFie. The effect of different baseline estimators on the limit of quantification in chromatography. *J. Chemometrics*, 11(1):1–11, Jan. 1997.
- [36] A. W. Moore and J. W. Jorgenson. Median filtering for removal of low-frequency background drift. *Anal. Chem.*, 65(2):188–191, 1993.
- [37] X. Ning and I. W. Selesnick. ECG enhancement and QRS detection based on sparse derivatives. *Biomed. Signal Process. Contr.*, 8(6):713–723, Nov. 2013.
- [38] G. A. Pearson. A general baseline-recognition and baseline-flattening algorithm. *J. Magn. Reson.*, 27(2):265–272, 1977.
- [39] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes: The Art of Scientific Computing*. Cambridge University Press, 3rd edition, 2007.
- [40] M. A. Rasmussen and R. Bro. A tutorial on the lasso approach to sparse modeling. *Chemometr. Intell. Lab. Syst.*, 119:21–31, Oct. 2012.
- [41] A. F. Ruckstuhl, M. P. Jacobson, R. W. Field, and J. A. Dodd. Baseline subtraction using robust local regression estimation. *J. Quant. Spectrosc. Radiat. Transf.*, 68(2):179–193, 2001.
- [42] G. Schulze, A. Jirasek, M. M. L. Yu, A. Lim, R. F. B. Turner, and M. W. Blades. Investigation of selected baseline removal techniques as candidates for automated implementation. *Appl. Spectrosc.*, 59(5):545–574, May 2005.
- [43] I. W. Selesnick. Resonance-based signal decomposition: A new sparsity-enabled signal analysis method. *Signal Process.*, 91(12):2793–2809, Dec. 2011.
- [44] I. W. Selesnick, S. Arnold, and V. R. Dantham. Polynomial smoothing of time series with additive step discontinuities. *IEEE Trans. Signal Process.*, 60(12):6305–6318, Dec. 2012.
- [45] I. W. Selesnick, H. L. Graber, D. S. Pfeil, and R. L. Barbour. Simultaneous low-pass filtering and total variation denoising. *IEEE Trans. Signal Process.*, 62(5):1109–1124, Mar. 2014.
- [46] H. C. Smit. Specification and estimation of noisy analytical signals: Part I. Characterization, time invariant filtering and signal approximation. *Chemometr. Intell. Lab. Syst.*, 8(1):15–27, 1990.
- [47] H. C. Smit. Specification and estimation of noisy analytical signals: Part II. Curve fitting, optimum filtering and uncertainty determination. *Chemometr. Intell. Lab. Syst.*, 8(1):29–41, 1990.
- [48] H. C. Smit and H. Steigstra. Noise and detection limits in signal-integrating analytical methods. In L. A. Currie, editor, *Detection in Analytical Chemistry: Importance, Theory, and Practice*, ACS Symposium Series, chapter 7, pages 126–148. American Chemical Society, 1988.
- [49] J.-L. Starck, M. Elad, and D. L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE Trans. Image Process.*, 14(10):1570–1582, Oct. 2005.
- [50] R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.
- [51] C. Vendeuvre, F. Bertoncini, L. Duval, J.-L. Duplan, D. Thiébaud, and M.-C. Hennion. Comparison of conventional gas chromatography and comprehensive two-dimensional gas chromatography for the detailed analysis of petrochemical samples. *J. Chrom. A*, 1056(1-2):155–162, 2004.
- [52] C. Vendeuvre, R. Ruiz-Guerrero, F. Bertoncini, L. Duval, and D. Thiébaud. Comprehensive two-dimensional gas chromatography for detailed characterisation of petroleum products. *Oil Gas Sci. Tech.*, 62(1):43–55, 2007.
- [53] Modern practice of gas chromatography, 4th Edition. *Wiley-Interscience*, 2004.

- [54] W. Schottky. Über spontane Stromschwankungen in verschiedenen Elektrizitätsleitern. *Proc. Camb. Phil. Soc.*, 15: 117–136, 1909.
- [55] N. Campbell. The Study of Discontinuous Phenomena. *Ann. Phys.*, 362: 541–567, 1918.
- [56] C. Vendevre, R. Ruiz-Guerrero, F. Bertoncini, L. Duval, D. Thiébaud, and M.-C. Hennion. Characterisation of middle-distillates by comprehensive two-dimensional gas chromatography (GC  $\times$  GC): A powerful alternative for performing various standard analysis of middle-distillates. *J. Chrom. A*, 1086(1-2):21–28, 2005.
- [57] P. D. Wentzell and C. D. Brown. Signal processing in analytical chemistry. In R. A. Meyers, editor, *Encyclopedia of Analytical Chemistry*. John Wiley & Sons Ltd, 2000.
- [58] J. D. Wilson and C. A. J. McInnes. The elimination of errors due to baseline drift in the measurement of peak areas in gas chromatography. *J. Chrom. A*, 19:486–494, 1965.
- [59] Z.-M. Zhang, S. Chen, and Y.-Z. Liang. Baseline correction using adaptive iteratively reweighted penalized least squares. *Analyst*, 135(5):1138–1146, 2010.